



# Melek IT

Program Studi  
Teknik Informatika  
Universitas Wijaya Kusuma Surabaya

Jurnal Teknologi Informasi Dan Komunikasi

Volume 1 No 2 Juli 2012

ISSN 2252-9128



9 772252 912806

**PENENTUAN QUALITY OF SERVICE (QOS) VOIP PADA JARINGAN LAN DAN JARINGAN VLAN MENGGUNAKAN METODE E-MODEL.** Agus Samsu Rizal, Nia Saurina.

**SISTEM MONITORING DAN KONTROL LAMPU GEDUNG.** M. Ali Mashudi, Tjatorsari W.

**INTERFACING ARDUINO DENGAN ANDROID MENGGUNAKAN ANDROID DEBUG BRIDGE STUDI KASUS SISTEM OTOMASI RUMAH SEDERHANA.** Badarrudin, Maslihah.

**ANALISIS KOMUNIKASI DATA PADA KOMPUTER IPv4 DAN Ipv6 MENGGUNAKAN DUAL STACK TRANSITION MECHANISM (DSTM).** Adi Bagus Santoso, F. X Wmsnu Yudo Untoro

**RANCANGAN SISTEM NAVIGASI PARIWISATA MENGGUNAKAN WEB SERVICE DENGAN PLATFORM ANDROID.** Hasbullah, Nia Saurina.

**PENJUALAN TENDA ONLINE BERBASIS E-COMMERCE DI CV. HARAPAN JAYA TENDA SURABAYA.** Kartika Dwi Jayantiningsih, Tjatorsari W.

**SISTEM INFORMASI CUTI PEGAWAI PADA PT. PELABUHAN INDONESIA III (PERSERO) SURABAYA.** Rizal Fathony, Tjatorsari W.

**SISTEM Pencarian Parkir Kosong Pada Lahan Parkir Khusus Mobil Di Pusat Grosir Surabaya.** Setyo Purwo Sarwono, Tjatorsari W.

**SISTEM MANAJEMEN AKUN DENGAN MENGGUNAKAN OPENLDAP.** Windyarto Adi Candra, Tjatorsari W.

**RANCANG BANGUN SISTEM DAFTAR RIWAYAT HIDUP DOSEN MENGGUNAKAN METODE VIEWPOINT ORIENTED REQUIREMENT DEFINITION (VORD).** Christian Novianto, Nia Saurina.

**IMPLEMENTASI PERPINDAHAN AGEN PADA JARINGAN NIRKABEL MENGGUNAKAN METODE HANDOFF UNTUK VOIP.** Ligar Syamrama, Nia Saurina



## Daftar Isi

- (1) **PENENTUAN QUALITY OF SERVICE (QOS) VOIP PADA JARINGAN LAN DAN JARINGAN VLAN MENGGUNAKAN METODE E-MODEL.** Agus Samsu Rizal, Nia Saurina. (Hal. 1 - 18)
- (2) **SISTEM MONITORING DAN KONTROL LAMPU GEDUNG.** M. Ali Mashudi, Tjatusari W. (Hal. 19 – 32)
- (3) **INTERFACING ARDUINO DENGAN ANDROID MENGGUNAKAN ANDROID DEBUG BRIDGE STUDI KASUS SISTEM OTOMASI RUMAH SEDERHANA.** Badarrudin, Maslihah. (Hal. 33 – 48).
- (4) **ANALISIS KOMUNIKASI DATA PADA KOMPUTER IPv4 DAN Ipv6 MENGGUNAKAN DUAL STACK TRANSITION MECHANISM (DSTM).** Adi Bagus Santoso, F. X Wisnu Yudo Untoro. (Hal. 49 - 60)
- (5) **RANCANGAN SISTEM NAVIGASI PARIWISATA MENGGUNAKAN WEB SERVICE DENGAN PLATFORM ANDROID.** Hasbullah, Nia Saurina. (Hal. 61 - 76)
- (6) **PENJUALAN TENDA ONLINE BERBASIS E-COMMERCE DI CV. HARAPAN JAYA TENDA SURABAYA.** Kartika Dwi Jayantiningih, Tjatusari W. (Hal. 77 - 88)
- (7) **SISTEM INFORMASI CUTI PEGAWAI PADA PT. PELABUHAN INDONESIA III (PERSERO) SURABAYA.** Rizal Fathony, Tjatusari W. (Hal. 89 - 98)
- (8) **SISTEM PENCARIAN PARKIR KOSONG PADA LAHAN PARKIR KHUSUS MOBIL DI PUSAT GROSIR SURABAYA.** Setyo Purwo Sarwono, Tjatusari W. (Hal. 99 - 114)
- (9) **SISTEM MANAJEMEN AKUN DENGAN MENGGUNAKAN OPENLDAP.** Windyarto Adi Candra, Tjatusari W. (Hal. 115 - 124)
- (10) **RANCANG BANGUN SISTEM DAFTAR RIWAYAT HIDUP DOSEN MENGGUNAKAN METODE VIEWPOINT ORIENTED REQUIREMENT DEFINITION (VORD).** Christian Novianto, Nia Saurina. (Hal. 125 - 142)
- (11) **IMPLEMENTASI PERPINDAHAN AGEN PADA JARINGAN NIRKABEL MENGGUNAKAN METODE HANDOFF UNTUK VOIP.** Ligar Syamrama, Nia Saurina (Hal. 143 - 152)

(1)  
**PENENTUAN *QUALITY OF SERVICE (QoS) VOIP* PADA JARINGAN LAN DAN  
JARINGAN VLAN MENGGUNAKAN METODE *E-MODEL***

**Agus Samsu Rizal, Nia Saurina<sup>2</sup>, Noven Indra Prasetya<sup>3</sup>**

Program Studi Teknik Informatika, Fakultas Teknik, Univesitas Wijaya Kusuma Surabaya  
[as.rizalazam@gmail.com](mailto:as.rizalazam@gmail.com)

**ABSTRAK**

Pada saat ini, dunia telekomunikasi berkembang dengan cepat. Salah satu bentuk alat telekomunikasi yang telah dikembangkan dan berbiaya murah adalah *VoIP (Voice over Internet Protocol)*. Teknologi *VoIP* dapat berjalan pada berbagai macam jaringan, baik itu jaringan *LAN* maupun jaringan *VLAN*. Permasalahannya adalah kualitas layanan (*QoS*) teknologi *VoIP* yang berjalan pada kedua jaringan tersebut belum begitu teruji. Sehingga diperlukan suatu penelitian untuk menguji *Quality of Service (QoS) VoIP* pada kedua jaringan.

Oleh karena itu dalam tugas akhir ini akan dilakukan penelitian untuk menguji *Quality of Service VoIP* yang berjalan pada jaringan *LAN* maupun jaringan *VLAN*. Penelitian dilakukan untuk menentukan besarnya *packet loss* dan *delay* dari *VoIP* yang berjalan pada masing-masing jaringan. Pengujian dilakukan menggunakan *OPNET IT Guru*. Pemilihan *OPNET* sebagai media pengujian dikarenakan dengan menggunakan *OPNET* kondisi jaringan yang dibangun dapat diberikan berbagai parameter sehingga akan mirip dengan kondisi jaringan yang sebenarnya.

Setelah menguji tingkat *packet loss* dan *delay* dari *VoIP* yang berjalan pada masing-masing jaringan, dilakukan pengukuran kualitas *VoIP* secara umum menggunakan metode *E-Model*, yang selanjutnya akan diubah kedalam nilai *Mean Opinion Score (MOS)*.

**Kata kunci :** QoS, VoIP, E-Model, Mean Opinion Score, OPNET.

**PENDAHULUAN**

**Latar Belakang**

Saat ini, teknologi informasi dan komunikasi berkembang dengan pesat. Alat komunikasi konvensional seperti telepon atau semacamnya bukan merupakan satu-satunya alat komunikasi yang digunakan dalam kehidupan sehari-hari.

Dengan berkembangnya teknologi informasi dan komunikasi tersebut muncul suatu teknologi komunikasi berbasis IP yang disebut dengan *VoIP*. *VoIP* merupakan suatu teknologi komunikasi *audio visual* yang menggunakan jaringan komputer baik lokal maupun internet sebagai media komunikasinya dengan biaya yang lebih murah dibandingkan dengan teknologi komunikasi konvensional.

Teknologi jaringan komputer lokal yang biasanya digunakan untuk berkomunikasi menggunakan *VoIP* adalah jaringan *LAN* atau yang sering disebut *Local Area Network*. Akan tetapi penelitian M. Mirza (2010) dalam Tugas Akhirnya mengatakan bahwa teknologi *VoIP* yang berjalan pada jaringan *LAN* memiliki kualitas suara yang tidak begitu bagus.

Keadaan tersebut terjadi karena teknologi *VoIP* yang berjalan pada jaringan *LAN* rawan akan adanya gangguan, baik gangguan berupa *noise* ataupun gangguan berupa bercampurnya data suara dengan data-data lainnya. Gangguan-gangguan tersebut akan menyebabkan besar *packet loss* dan *delay* dari teknologi *VoIP* yang berjalan pada jaringan *LAN* cukup tinggi, sehingga akan mempengaruhi kualitas dari jaringan *VoIP*.

Oleh karena itu, para pengembang teknologi *VoIP* mencari solusi untuk mengatasi masalah tersebut. Salah satunya adalah dengan menggunakan jaringan selain *LAN*, yaitu jaringan *Virtual LAN (VLAN)*. Dengan menggunakan jaringan *VLAN*, kualitas suara yang dihasilkan akan lebih baik daripada menggunakan jaringan *LAN*. Hal ini dikarenakan jaringan *VLAN* merupakan jaringan yang jalurnya bersifat privat. Artinya, data-data yang berjalan dalam jalur jaringan *VLAN* merupakan data yang sejenis, sehingga data tersebut tidak akan bercampur dengan data lain yang akan mengakibatkan berkurangnya kualitas suatu data.

Kualitas layanan teknologi *VoIP* di dalam jaringan *LAN* dan *VLAN* belum begitu teruji. Apalagi bila digunakan dalam jaringan dengan *traffick* yang padat, dimana terdapat berbagai macam

*request*, paket data dan lain sebagainya yang berjalan dalam jalur suatu jaringan. Oleh karena itu diperlukan suatu penelitian dan pembuktian kualitas suara dari *VoIP*, baik yang berjalan dalam jaringan *LAN* maupun jaringan *VLAN*, sehingga akan diketahui kualitas layanan *VoIP* yang berjalan dari masing-masing jaringan. Untuk membuktikan kualitas layanan *VoIP* pada masing-masing jaringan tersebut dilakukan uji coba dan implementasi menggunakan *OPNET*. Karena *OPNET* dapat diberi parameterparameter tertentu, sehingga sistem jaringan yang dibangun akan terlihat seperti kondisi yang sebenarnya. Selanjutnya akan dilakukan penghitungan nilai *QoS VoIP* pada jaringan *LAN* dan jaringan *VLAN* menggunakan metode *E-Model*.

## TUJUAN

Adapun tujuan dalam penelitian adalah menghitung *Quality of Service (QoS)* atau kualitas layanan *VoIP* yang berjalan pada jaringan *LAN* dan jaringan *VLAN* menggunakan metode *E-Model*.

## TINJAUAN PUSTAKA

### *QoS (Quality of Services)*

*QoS (Quality of Service)* atau biasa disebut kualitas layanan dari jaringan adalah pengukuran performansi dan kemampuan untuk memberikan prioritas yang berbeda untuk berbagai aplikasi, pengguna, atau aliran data, atau untuk menjamin tingkat kinerja tertentu ke aliran data.

Pengukuran performansi tersebut merupakan salah satu upaya dalam peningkatan efisiensi dan efektifitas kerja suatu jaringan guna meningkatkan produktifitas kerja pada jaringan. Walaupun ada suatu nilai performansi, nilai tersebut hanyalah menggambarkan tingkat utilitas dari suatu performansi jaringan.

Dalam kaitannya dengan pengukuran performansi dibuat suatu skala penilaian yang dapat mengukur prestasi pencapaian target nilai performansi. Target itu sendiri ditetapkan setelah adanya data nilai performansi sebelumnya sehingga target sangat realistis. Baik tidaknya nilai performansi sangat dipengaruhi oleh berbagai macam hal.

### A. Parameter *QoS*

*QoS (Quality of Service)* merupakan kemampuan jaringan untuk menyediakan service yang lebih baik pada suatu trafik tertentu mulai berbagai macam teknologi meliputi jaringan IP, frame relay, ATM dan SDH. Beberapa arsitektur telah diajukan untuk mengelola *QoS* dalam jaringan

IP. Dua arsitektur utama yaitu *Integrated Services (IntServ)* dan *Differentiated Services (DiffServ)*.

*Integrated Services* bertujuan menyediakan sumber daya seperti bandwidth untuk trafik dari end to end. Ditujukan untuk aplikasi yang peka terhadap delay dan keterbatasan bandwidth seperti *video conference* dan *VoIP*. Arsitekturnya didasarkan pada system pemesanan sumber daya per aliran trafik. Sistem pemesanan sumber daya memerlukan protokol tersendiri. Salah satu protokol yang sering digunakan adalah RSVP. *IntServ* sesuai untuk *VoIP* dan video tetapi sangat tidak tepat untuk aplikasi semacam web yang aliran trafiknya banyak tetapi datanya kecil.

*Differentiated Services* bertujuan untuk membagi trafik atas kelas-kelas yang kemudian diberi perlakuan yang berbeda. *DiffServ* menyediakan diferensiasi layanan dengan membagi trafik atas kelas-kelas dan memperlakukan setiap kelas secara berbeda.

Secara khusus *QoS* mampu memberikan service yang lebih baik dengan cara :

1. Mendukung *dedicated bandwidth*.
2. Memperbaiki / memperkecil timbulnya *loss packet* dalam jaringan.
3. Menghindari timbulnya *congestion*.
4. Memberikan prioritas pada suatu *traffic* tertentu melalui jaringan.

Elemen *QoS* tergantung dari informasi yang ditransmisikan (*voice*, data atau video). Faktor yang mempengaruhi *QoS* pada jaringan IP ada bermacam-macam, diantaranya adalah seperti berikut ini :

#### a. *Throughput*

*Throughput* yaitu kecepatan (*rate*) transfer data efektif, yang diukur dalam bps. *Header* dalam paket data mengurangi nilai ini. *Throughput* dapat dihitung dengan melihat jumlah paket yang datang terhadap yang dikirim.

*Throughput (S)* adalah total waktu yang digunakan mengirim paket dengan sukses per satuan waktu tertentu yang dapat dihitung dengan:

$$S = \frac{\sum \text{paket sukses} \times \text{waktu}}{\text{transmisi Lama Pengamatan}}$$

*Offered Traffic (G)* adalah total waktu paket yang ditawarkan per satuan waktu yang dapat dihitung dengan :

$$G = \frac{\sum \text{paket muncul} \times \text{waktu transmisi}}{\text{Lama Pengamatan}}$$



### b. Jitter

*Jitter* merupakan masalah khas dari *connectionless network* atau *packet switched network*. *Jitter* didefinisikan sebagai variasi *delay* yang diakibatkan oleh panjang *queue* dalam suatu pengolahan data dan *reassemble* paket-paket data di akhir pengiriman akibat kegagalan sebelumnya.

Besarnya nilai *jitter* akan sangat dipengaruhi oleh variasi beban trafik dan besarnya tumbukan antar paket (*congestion*) yang ada dalam jaringan IP. Semakin besar beban trafik di dalam jaringan akan menyebabkan semakin besar pula peluang terjadinya *congestion* dengan demikian nilai *jitter*-nya akan semakin besar. Semakin besar nilai *jitter* akan mengakibatkan nilai *QoS* akan semakin turun. Untuk mendapatkan nilai *QoS* jaringan yang baik, nilai *jitter* harus dijaga seminimum mungkin

Terdapat empat kategori penurunan performansi jaringan berdasarkan nilai *peak jitter* sesuai dengan versi Tiphon, seperti yang ditunjukkan pada tabel berikut :

Tabel 1 : Kategori Performansi Jaringan IP Berdasarkan Parameter *Jitter* (Sumber : Tiphon).

Kategori Degradasi	Peak Jitter
Sangat Bagus	0 ms
Bagus	75 ms
Sedang	125 ms
Jelek	225 ms

### c. Packet Loss

*Packet loss* adalah jumlah paket yang hilang. Biasanya perangkat jaringan memiliki *buffer* untuk menampung data yang diterima. Jika terjadi kongesti yang cukup lama, *buffer* akan penuh, dan data baru tidak akan diterima. Semakin besar *packet loss* maka semakin buruk *QoS* pada jaringan.

*Packet loss* terjadi karena kegagalan transmisi paket IP mencapai tujuannya yang didasari oleh protokol UDP dimana bersifat *connectionless*. Jika paket gagal dikirim maka paket tersebut tidak akan dikirim lagi dan menjadi masalah besar jika *packet loss* yang terjadi sangat besar. Kegagalan paket tersebut mencapai tujuan, dapat disebabkan oleh beberapa kemungkinan, diantaranya yaitu:

- Terjadinya *overload* trafik didalam jaringan.
- Tabrakan (*congestion*) dalam jaringan.
- *Error* yang terjadi pada media fisik.
- Kegagalan yang terjadi pada sisi penerima, bisa disebabkan karena *overflow* yang terjadi pada *buffer*.

Di dalam implementasi jaringan IP, nilai *packet loss* ini diharapkan mempunyai nilai yang

minimum. Tetapi dalam beberapa kasus, nilai *packet loss* untuk aplikasi *voice* dan multimedia dapat ditoleransi sampai dengan 20% untuk single *Access Point*.

Secara umum terdapat empat kategori penurunan performansi jaringan berdasarkan nilai *packet loss* sesuai dengan versi Tiphon yaitu sangat bagus, bagus, sedang dan jelek. Kategori *packet loss* berdasarkan persentase besar *packet loss* berdasarkan Typhon yang dapat diterima seperti tampak pada tabel berikut :

Tabel 2 : Kategori Performansi Jaringan IP Berdasarkan Parameter *Packet Loss* (Sumber : Tiphon).

Kategori Degradasi	Packet Loss
Sangat Bagus	0 %
Bagus	3 %
Sedang	15 %
Jelek	25 %

Untuk menghindari problem *packet loss* antara lain dengan tidak mengirimkan *silence packet* (terutama dalam *network* dengan kecepatan rendah atau *congesty*), teknik *redudancy* (paket n diberi tambahan *header*, yaitu paket (n-1) dengan steram audio yang resolusinya lebih rendah dari paket n sebagai informasi *redudant*), teknik *interleaving* (pada *interleaving* terjadi pembagian masing-masing paket menjadi beberapa bagian, lalu disatukan kembali dengan potongan paket lain, sehingga paket yang dikirim merupakan gabungan dari potongan masingmasing paket).

### d. Delay

*Delay* adalah waktu tunda saat paket yang diakibatkan oleh proses transmisi dari satu titik lain yang menjadi tujuannya. *Delay* juga dapat diartikan interval waktu yang dibutuhkan oleh suatu paket data saat data mulai dikirim dan keluar dari proses antrian dari titik awal menuju titik tujuan. *Delay* didalam jaringan dapat digolongkan sebagai berikut:

- *Packetisasi delay*

*Delay* yang disebabkan oleh waktu yang diperlukan untuk proses pembentukan paket IP dari informasi user, yang harus melewati proses enkapsulasi melalui 7 layer OSI atau 4 layer TCP/IP. *Delay* ini hanya terjadi sekali saja, yaitu di source informasi.

- *Queuing delay*

*Delay* ini disebabkan oleh waktu proses yang diperlukan oleh *router* di dalam menangani transmisi paket di sepanjang jaringan. Umumnya *delay* ini sangat kecil, kurang lebih sekitar 100 ms.

- *Delay propagasi*

Proses perjalanan informasi selama media transmisi, misalnya SDH, coax atau tembaga, menyebabkan *delay* yang disebut dengan *delay propagasi*.

Rekomendasi ITU G.114 merekomendasikan standar *delay*, bahwa ada 3 *band* atau batasan *delay*, yaitu 0-150 ms, 150-400 ms dan >400 ms, seperti yang ditunjukkan oleh tabel dibawah ini :

Tabel 3 : Batasan *delay* rekomendasi ITU-T G.114

Delay Range	Keterangan
0 – 150 ms	Batas <i>delay</i> yang direkomendasikan.
150 – 400 ms	Batas <i>delay</i> yang masih bisa diterima, apabila <i>designer</i> berniat untuk mengurangi kualitas.
> 400 ms	Tidak direkomendasikan, meskipun dalam beberapa kasus yang luar biasa masih memungkinkan diterima.

## B. Metode Pengukuran Kualitas VoIP

Untuk menentukan kualitas layanan suara dalam jaringan IP dapat digunakan beberapa metode di bawah ini :

### a. Mean Opinion Score (MOS)

Pengukuran secara subjektif melibatkan manusia yang mendengarkan suara langsung atau rekaman suara dan memberi rating kepadanya. Salah satu metode subjektif yang paling banyak digunakan adalah MOS (*Mean Opinion Score*) seperti dideskripsikan pada rekomendasi ITU-T P.800.

Karena menggunakan subjek manusia, pengukuran secara subjektif menjadi sangat akurat dan berguna dalam mengevaluasi sebuah sistem telephony.

Rekomendasi ITU-T P.800 menerangkan secara detail cara melakukan pengukuran metode subjektif, yang secara ringkas dapat disimpulkan sebagai berikut:

Sebuah panelis yang terdiri dari beberapa orang mendengarkan sebuah sampel suara, memberi nilai pada setiap sampel dengan rentang standar nilai dari 1 (Buruk) sampai dengan 5 (Baik Sekali) seperti ditunjukkan pada Tabel 4. Nilai rata-rata yang diberikan panelis merupakan nilai MOS.

Metode pengukuran seperti MOS yang menggunakan subjek manusia merupakan pengukuran yang baik untuk menilai kualitas suara.

Walaupun begitu, metode subjektif memiliki beberapa kekurangan. Secara khusus, metode ini membutuhkan waktu yang lama dan biaya yang tidak sedikit. Beberapa peneliti dan organisasi tidak memiliki cukup sumber daya untuk melakukan test pengukuran seperti itu.

Tabel 4 : Standar Nilai MOS Rekomendasi ITU-T P.800

Nilai MOS	Opini
5	Sangat Baik
4	Baik
3	Baik
2	Cukup
1	Baik
	Tidak Baik
	Buruk

Metode MOS dirasakan kurang efektif untuk mengestimasi kualitas layanan suara untuk VoIP, hal ini dikarenakan :

1. Tidak tedapatnya nilai yang pasti terhadap parameter yang mempengaruhi kualitas layanan suara dalam VoIP.
2. Setiap orang memiliki standar yang berbeda-beda terhadap suara yang mereka dengar dengan hanya melalui percakapan.

### b. E-Model

E-Model merupakan ukuran objektif dari jaringan telekomunikasi yang diperkenalkan oleh ETSI pada ETR 250 dan distandarkan oleh ITU-T melalui G.107.

Hasil akhir yang didapat merupakan R factor, yang dapat memprediksikan nilai MOS. Metode ini direkomendasikan bagi para penyedia jasa layanan telekomunikasi untuk menjadi panduan dalam perencanaan jaringan dan mengontrol performansinya.

Nilai akhir estimasi E-Model disebut dengan R faktor. R faktor didefinisikan sebagai faktor kualitas transmisi yang dipengaruhi oleh beberapa parameter seperti *signal-to-noise ratio* dan *echo perangkat*, *codec* dan kompresi, *packet loss*, dan *delay*.

Menurut R. G. Cole dan J. H. Rosenbluth dalam papernya (Cole, r. g and Rosenbluth, J.H., 2000: np), persamaan matematis E-Model untuk hubungan VoIP dari PC ke PC adalah :

$$R = 94.2 - I_d - I_f \dots\dots\dots (1)$$

$I_d$  = Faktor penurunan kualitas suara yang disebabkan oleh pengaruh *one-way delay*.  
 $I_f$  = Faktor penurunan kualitas disebabkan oleh *packet loss*.

$$I_d = 0.024d + 0.11(d) \times H(d) \dots\dots\dots (2)$$

Untuk teknik kompresi G.107 nilai  $I_f$  sesuai dengan persamaan berikut ini:

$$I_f = 7 + 30 \ln (1+15e) \dots\dots\dots (3)$$

Maka secara umum persamaan nilai estimasi R factor menjadi:

$$R = 94.2 - [ 0.024d + 0.11(d) \times H(d) ] - [ 7 + 30 \ln (1 + 15e) ]$$

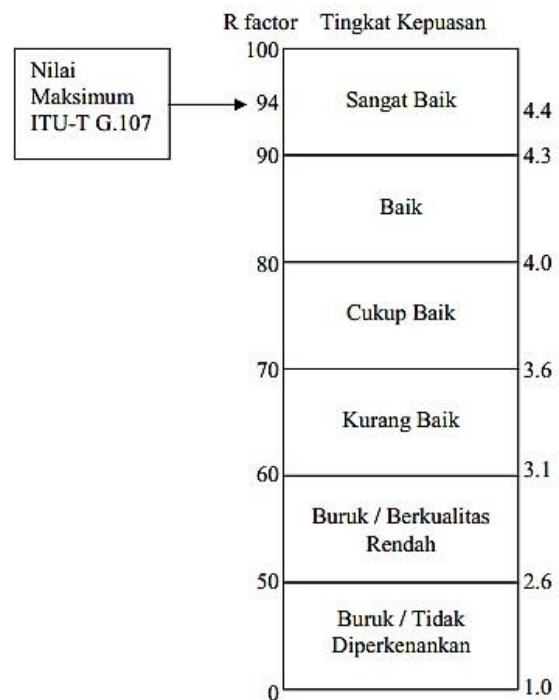
R = kualitas transmisi d = *one-way delay*  
H = fungsi tangga; H = 0, jika d ≤ 0, H = 1, jika d > 0.  
e = besarnya *packet loss*.

R Faktor berkisar antara 0 dan 100. Nilai R di bawah 50 dinilai tidak dapat diterima dan koneksi telepon pada umumnya tidak melampaui nilai di atas 94. Jadi nilai R yang baik berkisar antara 50 sampai dengan 94.

Selanjutnya nilai R dapat dikalkulasi untuk menentukan nilai estimasi MOS. Untuk mengubah nilai R menjadi estimasi MOS, digunakan ketentuanketentuan sebagai berikut :

Untuk  $R < 0$  : MOS = 1  
Untuk  $R > 100$  : MOS = 4.5  
Untuk  $0 < R < 100$  : MOS = Lihat gambar 2.1

Hubungan antara nilai R dengan nilai MOS ditunjukkan pada berikut :



Gambar 1 : Hubungan Nilai R Dengan MOS

Pada gambar 1 terlihat nilai R Faktor berkisar antara 0 dan 100 dan untuk nilai MOS berkisar antara 1,0 dan 4,4. Nilai R maksimum berdasarkan ITU-TG.107 tidak melampaui 94. Dan nilai R faktor di bawah 50 dinilai tidak dapat. Jadi nilai R yang bisa diterima berkisar antara 50 sampai dengan 94 atau kalau diubah dalam bentuk MOS, nilai MOS yang dapat diterima berkisar antara 2,6 dan 4,4.

### Voice Over Internet Protocol

Teknologi *Voice over Internet Protocol* (VoIP) adalah Teknologi yang menjadikan media internet untuk bisa melakukan komunikasi suara jarak jauh secara langsung. Sinyal suara *analog*, seperti yang anda dengar ketika berkomunikasi ditelepon diubah menjadi data digital dan dikirimkan melalui jaringan berupa paket-paket data secara *real time*.

Dalam komunikasi VoIP, pemakai melakukan hubungan telepon melalui *terminal* yang berupa PC atau telepon biasa. Dengan bertelepon menggunakan VoIP, banyak keuntungan yang dapat diambil diantaranya adalah dari segi biaya jelas lebih murah dari tarif telepon tradisional, karena jaringan IP bersifat global. Sehingga untuk hubungan Internasional dapat ditekan hingga 70%. Selain itu, biaya *maintenance* dapat ditekan karena *voice* dan data *network* terpisah, sehingga IP Phone dapat ditambah, dipindah dan diubah. Hal ini karena VoIP dapat dipasang disembarang *ethernet* dan IP

*address*, tidak seperti telepon *konvensional* yang harus mempunyai *port* tersendiri di Sentral atau *PBX* (*Private Branch Exchange*).

### Protokol-protokol Penunjang VoIP

Berikut adalah beberapa protokol-protokol yang menjadi penunjang jaringan *VoIP*, antara lain :

a. *TCP/IP* (*Transfer Control Protocol/Internet Protocol*)

Merupakan sebuah protokol yang digunakan pada jaringan internet. Standarisasi diperlukan agar antar komputer terjadi kesepakatan tentang tatacara pengiriman dan penerimaan data sehingga data dapat dikirimkan dan diterima dengan benar. Protokol ini terdiri dari dua bagian besar, yaitu *TCP* dan *UDP* serta dibawah lapisan tersebut ada protokol yang bernama *IP*. b. *SIP* (*Session Initiation Protocol*)

Merupakan protokol yang digunakan untuk *inisiasi*, modifikasi dan terminasi sesi komunikasi *VoIP*. *SIP* adalah protokol *Open Standard* yang dipublikasikan oleh *IETF*, *RFC 2543* dan *RFC 3261*. Selain digunakan untuk negosiasi sesi komunikasi *voice*, *SIP* juga dapat digunakan untuk negosiasi sesi komunikasi data media lain seperti video dan text. Disebutkan sebagai hanya melakukan “negosiasi sesi komunikasi” adalah karena *SIP* merupakan *signalling protocol*, bukan media *transfer protocol*. Artinya *SIP* tidak menghantar data media (*voice*, *video* dan *text*), melainkan hanya melakukan negosiasi sesi komunikasi saja dan memanfaatkan protokol lain seperti *RTP* sebagai media *transfer protocol*.

b. H.323

*VoIP* dapat berkomunikasi dengan sistem lain yang beroperasi pada jaringan *packet-switch*. Untuk dapat berkomunikasi dibutuhkan suatu standarisasi sistem komunikasi yang kompatibel satu sama lain. Salah satu standar komunikasi pada *VoIP* menurut rekomendasi *ITU-T* adalah *H.323* (1995-1996). Standar *H.323* terdiri dari komponen, protokol, dan prosedur yang menyediakan komunikasi multimedia melalui jaringan *packet-based*. Bentuk jaringan *packet-based* yang dapat dilalui antara lain jaringan internet, *Internet Packet Exchange (IPX)-based*, *Local Area Network (LAN)*, dan *Wide Area Network (WAN)*. *H.323* dapat digunakan untuk layanan-layanan multimedia seperti komunikasi suara (*IP telephony*), komunikasi video dengan suara (*video telephony*), dan gabungan suara, video dan data.

### Jaringan Komputer

Jaringan komputer merupakan sebuah sistem yang terdiri atas komputer-komputer yang

didesain untuk dapat berbagi sumber daya, berkomunikasi, dan dapat mengakses informasi. Tujuan dari jaringan komputer adalah agar setiap bagian dari jaringan komputer dapat meminta dan memberikan layanan. Pihak yang meminta/menerima layanan disebut *client* dan yang memberikan/mengirim layanan disebut *peladen server*. Desain ini disebut dengan sistem *client-server*, dan digunakan pada hampir seluruh aplikasi jaringan komputer.

#### A. Klasifikasi Jaringan Komputer

Berdasar geografinya, jaringan komputer terbagi menjadi *Local Area Network (LAN)*, *Metropolitan Area Network (MAN)*, dan *Wide Area Network (WAN)*.

*Local Area Network* merupakan jaringan milik pribadi di dalam sebuah gedung atau tempat yang berukuran 1-10 kilometer. *Local Area Network (LAN)* seringkali digunakan untuk menghubungkan komputer-komputer pribadi atau *workstation* dalam kantor suatu perusahaan atau pabrik-pabrik untuk bertukar informasi dan memakai satu sumber daya secara bersama-sama, misalkan mencetak printer dan lain-lain.

Untuk geografis yang lebih luas dari *LAN* dinamakan *Metropolitan Area Network (MAN)*. *MAN* merupakan perluasan dari jaringan *LAN* sehingga mencakup satu kota yang cukup luas, terdiri atas puluhan gedung yang berjarak 10-50 kilometer. Kabel transmisi yang digunakan adalah kabel serat optik (*Coaxial Cable*).

Sedangkan *Wide Area Network (WAN)* merupakan jaringan antarkota, antar propinsi, antar negara, bahkan antar benua. Jaraknya bisa mencakup seluruh dunia, misalnya jaringan yang menghubungkan semua bank di Indonesia, atau jaringan yang menghubungkan semua kantor Perwakilan Indonesia di seluruh dunia. Media transmisi utama adalah komunikasi lewat satelit, tetapi banyak yang mengandalkan koneksi serat optik antar negara.

Berdasarkan fungsinya, jaringan komputer terbagi menjadi jaringan *client server* dan jaringan *peer to peer*. Dalam jaringan *client server* pada dasarnya terdapat satu komputer yang disiapkan menjadi *server* dari komputer lainnya yang bertindak sebagai *client*. Semua permintaan layanan sumberdaya dari komputer *client* harus melewati komputer *server*, komputer *server* ini yang akan mengatur pelayanannya. Komputer yang bertindak sebagai *server* dapat lebih dari satu dengan pembagian tugas tertentu, misalnya *file server*, *database server* dan sebagainya.

Sedangkan pada jaringan *peer to peer* tidak ada komputer yang berfungsi khusus sebagai *client*

ataupun sebagai *server*. Semua komputer dapat berfungsi sebagai *client* dan *server* dalam satu saat bersamaan. Model jaringan ini biasanya hanya bisa diterapkan pada jumlah komputer yang tidak terlalu banyak, karena komunikasi akan menjadi rumit dan macet apabila jumlah komputer terlalu banyak.

Berdasarkan topologinya, jaringan komputer terbagi menjadi 5 macam, yaitu topologi *bus*, topologi *ring*, topologi *star*, topologi *tree* dan topologi *mesh*.

Pada topologi *bus* dua ujung jaringan harus diakhiri dengan sebuah terminator. Pabila ingin memperluas jaringan dapat meggunakan *barel connector*. Jaringan hanya terdiri dari satu saluran kabel yang menggunakan kabel BNC (*Bayonet Naur Connector*). Instalasi jaringan *bus* sangat sederhana, murah dan maksimal terdiri atas 5-7 komputer. Kesulitan yang sering dihadapi adalah kemungkinan terjadinya tabrakan data karena mekanisme jaringan relatif sederhana dan jika salah satu node putus maka akan mengganggu kinerja dan trafik seluruh jaringan.

Topologi *ring* (cincin) adalah topologi jaringan berbentuk rangkaian titik yang masing-masing terhubung ke dua titik lainnya, sehingga membentuk jalur melingkar membentuk cincin. Pada topologi cincin, komunikasi data dapat terganggu jika satu titik mengalami gangguan. *Jaringan FDDI (Fiber DistributedData Interface)* yang merupakan protokol standar komunika data menggunakan *fiber optic* mengantisipasi kelemahan ini dengan mengirim data searah jarum jam dan berlawanan dengan arah jarum jam secara bersamaan. Topologi ring digunakan dalam jaringan yang memiliki performance tinggi dan jaringan yang membutuhkan *bandwidth* untuk fitur yang *timesensitive* seperti video dan audio, atau ketika performance dibutuhkan saat komputer yang terhubung ke jaringan dalam jumlah yang banyak.

Topologi *star* digunakan dalam jaringan yang padat, ketika *endpoint* dapat dicapai langsung dari lokasi pusat, kebutuhan untuk perluasan jaringan, dan membutuhkan kehandalan yang tinggi. Topologi ini merupakan susunan yang menggunakan lebih banyak kabel daripada *bus* dan *ring* karena semua komputer dan perangkat terhubung ke *central point*. Jadi bila ada salah satu komputer atau perangkat yang mengalami kerusakan maka tidak akan mempengaruhi perangkat yang lainnya (jaringan).

Topologi *tree* disebut juga sebagai topologi jaringan bertingkat. Topologi ini biasanya digunakan untuk interkoneksi antar *central* dengan hirarki yang berbeda. Untuk hirarki yang lebih rendah digambarkan pada lokasi yang rendah dan semakin keatas mempunyai hirarki semakin tinggi.

Topologi jaringan jenis ini cocok digunakan pada sistem jaringan komputer.

Topologi *mesh* adalah suatu bentuk hubungan antar perangkat dimana setiap perangkat terhubung secara langsung ke perangkat lainnya yang ada di dalam jaringan. Akibatnya, dalam topologi mesh setiap perangkat dapat berkomunikasi langsung dengan perangkat yang dituju (*dedicated links*). Dengan demikian maksimal banyaknya koneksi antar perangkat pada jaringan bertopologi mesh ini dapat dihitung yaitu sebanyak  $n(n-1)/2$ , dimana  $n$  mewakili jumlah perangkat atau komputer dalam jaringan. Selain itu karena setiap perangkat dapat terhubung dengan perangkat lainnya yang ada di dalam jaringan maka setiap perangkat harus memiliki sebanyak  $n-1$  port *Input/Output (I/O ports)*.

### B. Virtual LAN

VLAN adalah suatu teknologi jaringan komputer yang tidak terbatas pada media fisik seperti LAN, hal ini mengakibatkan suatu jaringan komputer dapat dikonfigurasi secara *virtual* tanpa harus menyesuaikan media fisiknya. Penggunaan VLAN akan membuat pengaturan jaringan menjadi sangat fleksibel tanpa bergantung pada lokasi *workstation*. Dengan kata lain VLAN bisa membagi suatu *broadcast domain* sesuai dengan kelompok atau organisasi sehingga, dapat saling berkomunikasi sesuai dengan *port-port* yang sudah ditetapkan. Sedangkan *port-port* yang berada diluar VLAN kelompoknya tidak dapat melakukan komunikasi secara langsung karena VLAN tidak meneruskan *broadcast* yang berbeda *port* tersebut.

VLAN diklasifikasikan berdasarkan metode (tipe) yang digunakan untuk mengklasifikasikannya, baik menggunakan port, MAC address dan sebagainya. Semua informasi yang mengandung penandaan atau pengalamatan suatu vlan (*tagging*) di simpan dalam suatu database (tabel), jika penandaannya berdasarkan port yang digunakan maka database harus mengindikasikan port-port yang digunakan oleh VLAN. Untuk mengaturnya maka biasanya digunakan *switch/bridge* yang manageable atau yang bisa di atur. *Switch/bridge* inilah yang bertanggung jawab menyimpan semua informasi dan konfigurasi suatu VLAN dan dipastikan semua *switch/bridge* memiliki informasi yang sama. *Switch* akan menentukan kemana data-data akan diteruskan dan sebagainya atau dapat pula digunakan suatu software pengalamatan (*bridging software*) yang berfungsi mencatat/menandai suatu VLAN beserta *workstation* yang didalamnya untuk menghubungkan antar VLAN dibutuhkan router.

Perbedaan yang sangat jelas dari model jaringan *Local Area Network* dengan *Virtual Local*



Area Network adalah bahwa bentuk jaringan dengan model *Local Area Network* sangat bergantung pada letak fisik dari *workstation*, serta penggunaan *hub* dan *repeater* sebagai perangkat jaringan yang memiliki beberapa kelemahan. Sedangkan yang menjadi salah satu kelebihan dari model jaringan dengan VLAN adalah bahwa tiap-tiap *workstation* atau *user* yang tergabung dalam satu VLAN atau satu bagian dapat tetap saling berhubungan walaupun terpisah secara fisik.

Keanggotaan dalam suatu VLAN dapat di kelompokkan berdasarkan berbagai macam cara, antara lain :

#### a. Berdasarkan Port

Keanggotaan pada suatu VLAN dapat di dasarkan pada port yang di gunakan oleh VLAN tersebut. Sebagai contoh, pada bridge/switch dengan 4 port yaitu port 1, 2, dan 4 merupakan VLAN 1 sedang port 3 dimiliki oleh VLAN 2, seperti terlihat di bawah.

Port	1	2	3	4
VLAN	2	2	1	2

Kelemahannya adalah user tidak bisa bebas untuk berpindah pindah. Apabila terpaksa harus berpindah maka *network administrator* harus mengkonfigurasi ulang.

#### b. Berdasarkan MAC Address

Keanggotaan suatu VLAN dapat didasarkan pada MAC address dari setiap workstation atau komputer yang dimiliki oleh user. Switch mendeteksi atau mencatat semua MAC address yang dimiliki oleh setiap Virtual LAN. MAC address merupakan suatu bagian yang dimiliki oleh NIC (Network Interface Card) di setiap workstation. Kelebihannya apabila user berpindah pindah maka dia akan tetap terkonfigurasi sebagai anggota dari VLAN tersebut. Sedangkan kekurangannya adalah setiap mesin harus di konfigurasi secara manual, dan untuk jaringan yang memiliki ratusan workstation maka tipe ini kurang efisien untuk dilakukan.

MAC Add	132516617738	272389579355
VLAN	1	2

#### c. Berdasarkan tipe protokol yang digunakan

Keanggotaan VLAN juga bisa berdasarkan protokol yang digunakan. Seperti contoh berikut :

Protokol	IP	IPX
VLAN	1	2

#### d. Berdasarkan Alamat Subnet IP

Subnet IP address pada suatu jaringan juga dapat digunakan untuk mengklasifikasi atau mengelompokkan suatu VLAN.

IP Subnet	22.3.24	22.3.15
VLAN	1	2

Konfigurasi ini tidak berhubungan dengan routing pada jaringan dan juga tidak memperlakukan fungsi router. IP address digunakan untuk memetakan keanggotaan VLAN. Keuntungannya seorang user tidak perlu mengkonfigurasi ulang alamatnya di jaringan apabila berpindah tempat, hanya saja karena bekerja di layer yang lebih tinggi maka akan sedikit lebih lambat untuk meneruskan paket di banding menggunakan MAC addresses.

#### e. Berdasarkan aplikasi atau kombinasi lain

Sangat dimungkinkan untuk menentukan suatu VLAN berdasarkan aplikasi yang dijalankan, atau kombinasi dari semua tipe di atas untuk diterapkan pada suatu jaringan. Misalkan aplikasi FTP (*file transfer protocol*) hanya bias digunakan oleh VLAN 1 dan Telnet hanya bisa digunakan pada VLAN 2 dan sebagainya.

#### f. Keuntungan Menggunakan VLAN

Ada beberapa keuntungan didalam penggunaan Vlan. seperti yang anda ketahui di dunia jaringan komputer, bagaimana kita memperoleh kelebihan tanpa harus mengambil resiko yang besar. salah satunya menggunakan Vlan. berikut ialah keuntungan menggunakan Vlan antara lain:

1. *Security*: keamanan data dalam sebuah divisi dapat di buat sendiri, mengapa demikian, karena dari setiap segmennya bisa dipisah secara logika. selain itu lalu lintas datanya juga di batasi oleh segmennya.
2. *Cost reduction* : penghematan dari penggunaan *bandwith* yang ada, dan *up-grade* perluasan jaringan yang bisa menjadikan biaya mahal.
3. *Higher performance* : pembagian jaringan pada layer 2 ke dalam beberapa kelompok *broadcast domain* yang lebih kecil, yang tentunya akan mengurangi lalu lintas paket data yang tidak dibutuhkan dalam jaringan tersebut.
4. *Broadcast storm mitigation* : pembagian jaringan ke dalam *vlan-vlan* akan mengurangi banyaknya *device* yang ikut serta dalam pembuatan *broadcast storm*. mengapa demikian, karena adanya pembatasan broadcast domain.
5. *Improved IT staff efficiency* : vlan memudahkan manajemen jaringan karena pengguna yang membutuhkan sumber daya yang dibutuhkan berbagi dengan *segmen* yang sama.

#### OPNET

OPNET (*Optimized Network Engineering Tool* ) adalah salah satu software untuk network modelling yang sering digunakan dalam mendesain

atau optimasi suatu jaringan. *OPNET* memiliki banyak modul yang disesuaikan dengan equipment dari banyak vendor yang digunakan pada banyak perusahaan. Dukungan inilah yang mempermudah user ataupun designer dalam merancang maupun melakukan optimasi suatu jaringan. Sayangnya fitur-fitur yang lengkap tersebut dijual per modul dengan harga yang cukup mahal.

Salah satu solusinya adalah dengan menggunakan versi academic atau educational version. Pada versi ini kita bisa menginstal *OPNET* dengan fitur-fitur yang telah disediakan untuk kepentingan educational. Memang tidak selengkap versi berbayarnya, namun sudah cukup untuk belajar mendisain dan mengoptimasi suatu jaringan, entah itu jaringan telekomunikasi maupun jaringan paket, wired maupun wireless.

Software ini memiliki kelebihan-kelebihan untuk mendesign jaringan berdasarkan perangkat yang ada di pasaran, protocol, layanan dan teknologi yang sedang beredar dalam dunia telekomunikasi. Hasil simulasi dapat dibuat dalam beberapa skenario sehingga dapat dijadikan dasar di dalam perencanaan suatu jaringan berbasis paket.

Salah satu cara yang mudah untuk merencanakan suatu jaringan berbasis paket adalah dengan mensimulasikan suatu jaringan dengan kondisi yang serupa dengan kondisi eksisting. Jenis perangkat switch (router, switch), jaringan transmisi, layanan, protocol dan lainnya yang terdapat di dalam jaringan eksisting digambarkan di dalam *OPNET* untuk selanjutnya di simulasikan dalam waktu tertentu untuk dapat dilihat parameter yang dibutuhkan.

Dengan membuat beberapa model simulasi, prediksi kebutuhan jaringan (BW), kebutuhan *Quality of Service* suatu layanan, jenis perangkat yang tepat, dan lain-lain dapat digambarkan, sehingga hasil ini dapat dipergunakan untuk suatu perencanaan suatu jaringan berbasis IP. *OPNET* dapat dipergunakan untuk simulasi jaringan paket berbasis Internet Protocol (IP), Asynchronous Transfer Mode (ATM), Frame Relay ataupun TDM. Jenis layanan yang disimulasikan juga beragam, baik itu internet (WEB), *VoIP*, File transfer, video conference, video streaming dan lain-lain yang dapat di setting berdasarkan kebutuhan dari pengguna simulasi. Secara umum *OPNET* sudah cukup sebagai simulator berbasis paket yang handal dan dapat dikembangkan oleh penggunaannya.

## ANALISA DAN PERANCANGAN

Perencanaan yang dilakukan adalah membuat desain jaringan *VoIP* yang mewakili desain jaringan yang sebenarnya, baik yang berjalan

pada jaringan *LAN* maupun yang berjalan pada jaringan *VLAN*. Selain itu juga dilakukan pencatatan terhadap kebutuhan *hardware* maupun *software* yang dibutuhkan dalam menganalisa kualitas suara yang dihasilkan *VoIP* yang berjalan pada jaringan *LAN* dan jaringan *VLAN*.

### Kebutuhan Sistem

Kebutuhan sistem merupakan penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponen dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan dan hambatan yang terjadi, sehingga dapat diusulkan perbaikan. Tahap analisa merupakan tahap yang penting, karena kesalahan ditahap ini akan menyebabkan juga kesalahan pada tahap perancangan. Analisis sistem diperlakukan untuk memenuhi kebutuhan akan data dan prosedur suatu sistem yang sedang berjalan.

#### A. Kebutuhan Perangkat Keras (*Hardware*)

Dalam menentukan kualitas suara *VoIP* yang berjalan pada jaringan *LAN* maupun jaringan *VLAN*, membutuhkan beberapa perangkat keras (*hardware*) untuk mendukung penelitian ini. *Hardware* tersebut terdiri dari dua macam, yaitu hardware pada *OPNET* dan hardware lingkungan pengujian. Hardware lingkungan pengujian adalah Laptop Tosgiba Satellite C840, sedangkan hardware pada *OPNET* diantaranya adalah sebagai berikut :

1. Server *VoIP*
2. *Ethernet 16 Switch*
3. Kabel *Ethernet 10Base T*
4. *Workstation*

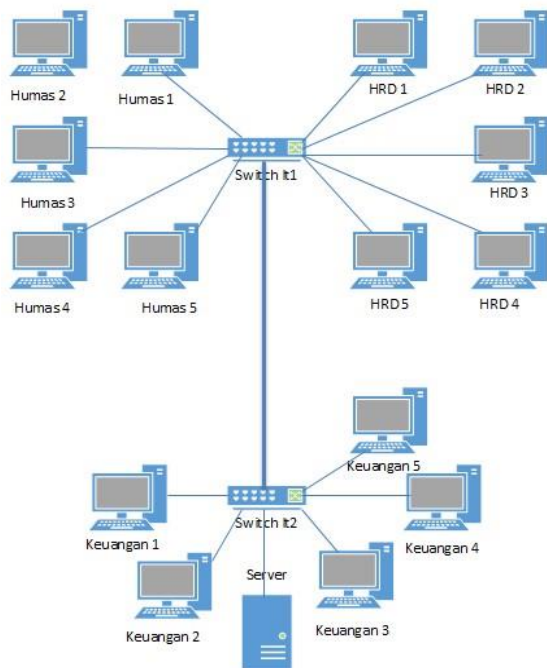
#### B. Kebutuhan Perangkat Lunak (*Software*)

Dalam menentukan kualitas suara *VoIP* yang berjalan pada jaringan *LAN* dan jaringan *VLAN*, membutuhkan beberapa *software* untuk mendukung penelitian ini, baik *software* pada lingkungan pengujian, maupun *software* pada aplikasi *OPNET*, diantaranya adalah sebagai berikut:

1. Sistem operasi Windows 7 Professional
2. *OPNET IT Guru*
3. *VoIP Profile OPNET*
4. *VoIP Application OPNET*
5. Microsoft Visual Studio 2008
6. Microsoft .Net Framework

### Desain Jaringan

Desain jaringan menjelaskan mengenai desain jaringan yang digunakan untuk pengujian *QoS VoIP* pada jaringan *LAN* dan jaringan *VLAN*. Desain jaringan tersebut dapat dilihat pada gambar 2 di bawah ini.



Gambar 2 : Desain Jaringan VoIP pada OPNET

Gambar 2 memperlihatkan sebuah desain jaringan komputer yang terdiri dari 2 lantai. Lantai pertama terdapat dua divisi yaitu Divisi Humas dan Divisi HRD dan terhubung kedalam jaringan dengan sebuah switch.. Pada Divisi Humas terdapat 5 buah *workstation* dan Divisi HRD juga terdapat 5 buah *workstation*.

Pada desain jaringan lantai 2 terdapat sebuah divisi yaitu Divisi Keuangan yang juga mempunyai 5 buah *workstation*. Selain Divisi keuangan, di lantai 2 juga terdapat sebuah *server* utama gedung 2 lantai tersebut. Semua *workstation* dan *server* yang ada di lantai 2 tersebut terhubung ke dalam jaringan menggunakan sebuah switch. Switch tersebut juga terhubung dengan switch di lantai 1 dan menghubungkan jaringan pada lantai 1 dengan jaringan pada lantai 2.

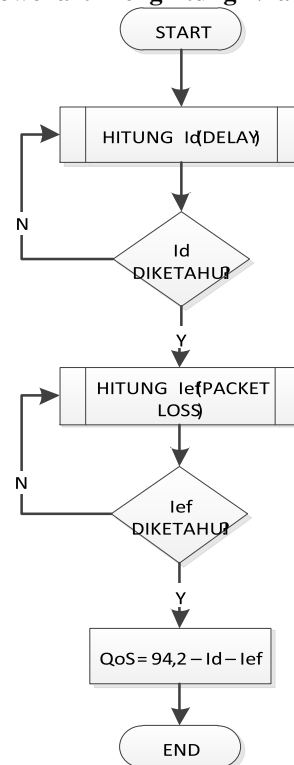
Desain jaringan tersebut digunakan untuk dua kali pengujian *QoS VoIP*. Pengujian *QoS VoIP* yang pertama adalah untuk menentukan besar *packet loss* dan *delay VoIP* yang berjalan pada jaringan LAN. Pengujian selanjutnya adalah untuk menentukan besar *packet loss* dan *delay VoIP* yang berjalan pada jaringan VLAN.

Pada pengujian jaringan VLAN, pembagian jaringan VLANnya berdasarkan pada asal divisi. Terdapat tiga VLAN yaitu VLAN Humas yang diidentifikasi sebagai VLAN11, VLAN HRD yang diidentifikasi sebagai VLAN22 dan VLAN Keuangan yang diidentifikasi sebagai VLAN33. Server utama termasuk dalam VLAN33.

## Desain Sistem

Desain sistem dibuat bertujuan untuk memberikan gambaran secara umum kepada pengguna dalam menentukan kualitas suara dari VoIP yang berjalan dalam jaringan LAN maupun yang berjalan dalam jaringan VLAN. Gambaran yang akan dipaparkan antara lain berupa flowchart seperti berikut :

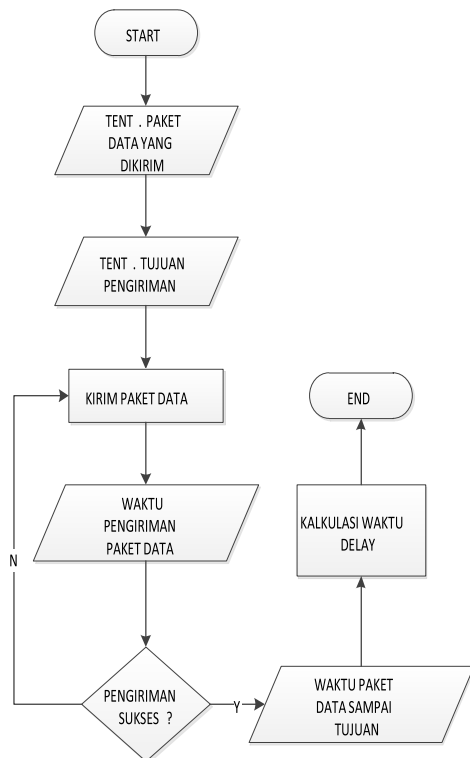
### A. Flowchart Menghitung Nilai QoS



Gambar 3 : Flowchart Penghitungan Nilai QoS

Gambar 3 menjelaskan proses menghitung nilai *Quality of Service (QoS)* menggunakan metode E-Model dengan memperhatikan besar *packet loss* dan besar *delay* dalam jaringan.

## B. Flowchart Delay

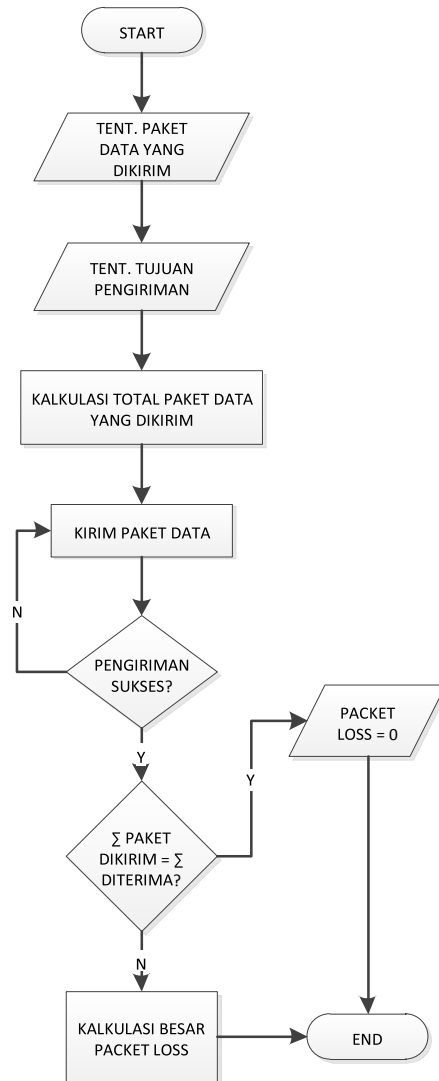


Gambar 4 : Flowchart Delay

Gambar 4 menjelaskan menjelaskan proses menentukan besar *delay* pengiriman suatu paket data pada sebuah jaringan. Proses dimulai dengan menentukan paket data dan tujuan pengiriman paket data tersebut. Selanjutnya waktu pengiriman paket data dihitung sejak paket data mulai dikirimkan. Selanjutnya setelah pengiriman sukses, dilihat juga waktu paket data tersebut sampai. Kemudian dikalkulasi besar *delay* dengan melihat tenggang waktu pengiriman paket dan waktu paket sampai tujuan.

## C. Flowchart Packet Loss

Gambar 5 menjelaskan proses menentukan besar *packet loss* suatu paket data dalam satu jaringan. Proses dimulai dengan menentukan paket data dan tujuan pengiriman paket data tersebut. Kemudian besar paket data yang akan dikirim tersebut dikalkulasi. Selanjutnya paket data dikirim pada jaringan. Apabila pengiriman sukses, kemudian dilihat apakah jumlah paket data yang dikirim sama dengan jumlah paket data ng diterima. Apabila sama, maka besar *packet loss* nya adalah 0. Apabila tidak sama maka dihitung besar *packet loss* nya.



Gambar 5 : Flowchart Packet Loss

## Skenario Pengujian

Skenario pengujian dibuat dengan tujuan agar penentuan *QoS VoIP* yang berjalan pada jaringan LAN dan jaringan VLAN berjalan sesuai dengan apa yang diharapkan. Ada dua skenario utama yang akan dijalankan.

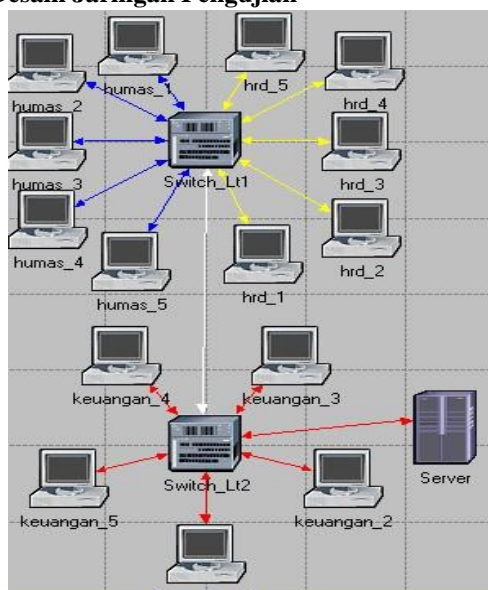
Skenario yang pertama akan menentukan nilai *packet loss* dan *delay* dari *VoIP* yang berjalan pada jaringan LAN. Desain jaringan yang telah dibuat akan diimplementasikan pada *OPNET IT Guru* dengan pengkondisian menggunakan jaringan LAN. Selanjutnya akan dianalisa berapa nilai *packet loss* dan *delay* dari *VoIP* yang berjalan pada jaringan LAN tersebut. Setelah nilai-nilai tersebut didapat, selanjutnya nilai tersebut dijadikan *input* untuk sistem perhitungan *QoS* menggunakan metode *E-Model* yang sudah dibuat. Hasilnya selanjutnya akan diubah ke dalam bentuk MOS (*Mean Opinion Score*).

Skenario yang kedua hampir mirip dengan skenario yang pertama. Perbedaannya terletak pada pengkondisian jaringannya. Desain jaringan yang telah dibuat sebelumnya akan diimplementasikan pada *OPNET Modeller 14.5* dengan pengkondisian menggunakan jaringan *VLAN*. Selanjutnya akan dianalisa berapa nilai *packet loss* dan *delay* dari *VoIP* yang berjalan pada jaringan *VLAN* tersebut. Setelah nilai-nilai tersebut didapat, selanjutnya nilai tersebut dijadikan *input* untuk sistem perhitungan *QoS* menggunakan metode *E-Model* yang sudah dibuat. Hasilnya selanjutnya akan diubah ke dalam bentuk *MOS (Mean Opinion Score)*.

## IMPLEMENTASI SISTEM

Implementasi merupakan tahap penerapan dan menjelaskan bagaimana langkah-langkah yang akan dilakukan untuk mengimplementasikan penelitian yang dilakukan sesuai dengan perancangan sistem yang telah dijelaskan.

### Desain Jaringan Pengujian



Gambar 6 : Desain Jaringan Pengujian

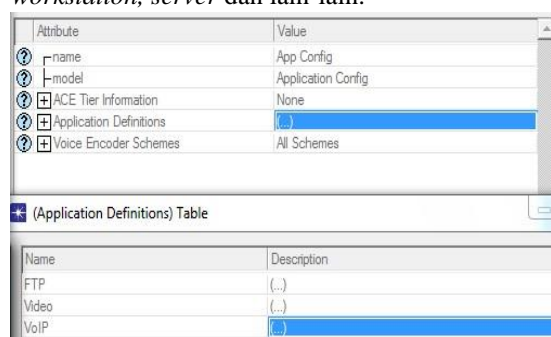
Gambar 6 memperlihatkan desain jaringan pengujian *QoS VoIP* pada jaringan *OPNET*. Penentuan *VLAN*nya didasarkan pada divisi masing-masing *workstation*. Karena terdapat 3 divisi, maka juga terdapat 3 *VLAN*, yaitu *VLAN* Divisi Humas, *VLAN* Divisi HRD dan *VLAN* divisi Keuangan. Pembagian *VLAN* lebih jelasnya lihat pada tabel 5 berikut :

Divisi	Nama VLAN	Identifler(VID)
Humas	VLAN11	11
HRD	VLAN22	22
Keuangan	VLAN33	33

Tabel 5 : Pembagian VLAN.

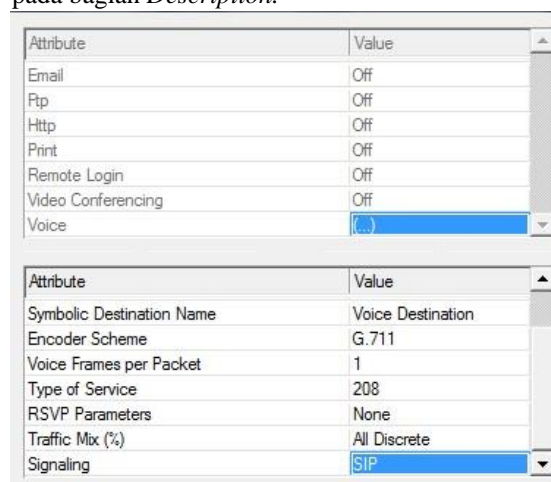
### Mengatur Application Configuration

*Application configuration* diperlukan dalam pembangunan *VoIP* pada jaringan LAN. *Application configuration* berfungsi untuk mendefinisikan berbagai *services* atau aplikasi yang akan dijalankan pada jaringan. *Application configuration* harus diatur terlebih dahulu sebelum kita melakukan *setting* pada *profile configuration*, *workstation*, *server* dan lain-lain.



Gambar 7 : Pengaturan Application Configuration

Gambar 7 menunjukkan pengaturan pada *application configuration*. Pada bagian "*Application Definitions*" tentukan aplikasi apa saja yang nantinya akan berjalan pada jaringan, yaitu *FTP*, *Video* dan *VoIP*. Masing-masing aplikasi tersebut juga harus ditentukan atributnya. Pengaturannya dilakukan dengan cara mengedit pada bagian *Description*.



Gambar 8 : Edit Attribute



Gambar 8 menunjukkan pengaturan *attribute* dari aplikasi *VoIP* yang berjalan pada jaringan. Atribut pada *VoIP* merupakan *voice* yang memiliki tipe *signaling* yaitu *SIP*, dengan *encoder scheme* *G.711*, *Voice Frame* tiap paketnya adalah 1 dan tipe servisnya adalah 208 yang merupakan tipe *Interactive Voice*.

### Mengatur Profile Configuration

Pengaturan *profile configuration* diperlukan untuk mendefinisikan aplikasi-aplikasi yang diperlukan oleh grup-grup tertentu, *client* ataupun *host*. Sebagai contoh, Divisi Humas memerlukan semua aplikasi yang telah kita atur pada *Application Configuration* pada Gambar 7 yaitu *FTP*, *video* dan *VoIP*.

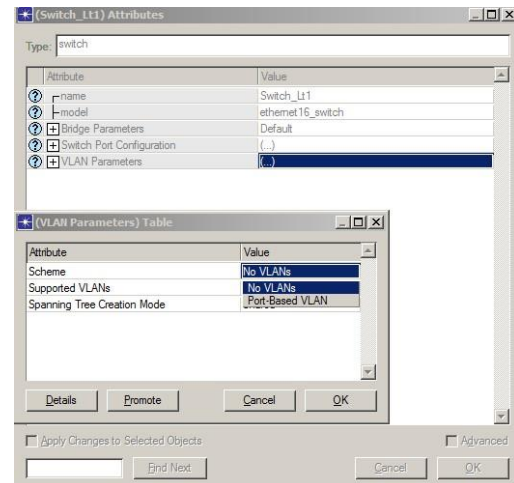
Profile Name	Applications	Operation Mode	Start Time (seconds)	Duration (seconds)	Repeatability
HRD	(...)	Simultaneous	uniform (100,110)	End of Simulation	Unlimited
Humas	(...)	Simultaneous	uniform (100,110)	End of Simulation	Unlimited
Keuangan	(...)	Simultaneous	uniform (100,110)	End of Simulation	Unlimited

Gambar 9 : Membuat Profil Masing-Masing Divisi

Gambar 9 menunjukkan cara pembuatan profil dari 3 divisi yang ada dalam jaringan, yaitu Divisi Humas, Divisi HRD dan Divisi Keuangan. Langkah-langkah pembuatannya adalah Edit pada *Profile Configuration* kemudian terlebih dahulu tentukan jumlah divisi yang ada pada bagian *rows* yaitu 3. Selanjutnya edit nama profil sesuai dengan nama-nama divisi yang ada dan ganti *operation mode* menjadi *simultaneous*. Tentukan durasi sampai simulasi berakhir (*end of simulation*) dan pengulangan tanpa batas (*unlimited*).

### Mengatur Switch

Pengaturan *switch* diperlukan untuk membuat *VLAN* yang diinginkan. Ada 3 buah *VLAN* yang dibuat, yaitu *VLAN11*, *VLAN22* dan *VLAN 33*.



Gambar 10 : Mengatur Switch

Gambar 10 memperlihatkan cara pengaturan *switch* agar dapat menjalankan *VLAN*. Klik pada salah satu *switch* kemudian pilih *Edit Similar Nodes* agar konfigurasi berlaku bagi kedua *switch*. Edit pada *VLAN Parameters*, atur skemanya yang awalnya *No VLANs* ubah menjadi *Port Based VLAN*. Pada bagian *Supported VLANs* tambahkan parameter-parameter *VLAN* yang diperlukan.

### Link Attributes

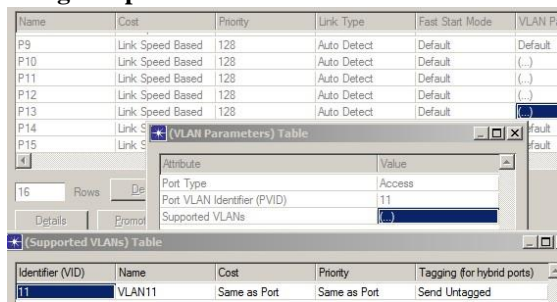
Untuk mengatur agar semua *client workstation* dapat tergabung menjadi salah satu anggota *VLAN* yang sudah dibuat, perlu diketahui *port* mana yang menghubungkan *workstation*, *server* dan *switch*.

name	model	port a	port b	Background Utilization	data rate
duplex_0	10BaseT Switch_L11 Ethernet (P0)	hrd_1 Ethernet (IF0 P0)	None	10,000,000	None
duplex_1	10BaseT Switch_L11 Ethernet (P1)	hrd_2 Ethernet (IF0 P0)	None	10,000,000	None
duplex_2	10BaseT Switch_L11 Ethernet (P10)	hrd_3 Ethernet (IF0 P0)	None	10,000,000	None
duplex_3	10BaseT Switch_L11 Ethernet (P11)	hrd_4 Ethernet (IF0 P0)	None	10,000,000	None
duplex_4	10BaseT Switch_L11 Ethernet (P12)	hrd_5 Ethernet (IF0 P0)	None	10,000,000	None
duplex_5	10BaseT Switch_L11 Ethernet (P13)	humas_1 Ethernet (IF0 P0)	None	10,000,000	None
duplex_6	10BaseT Switch_L11 Ethernet (P14)	humas_2 Ethernet (IF0 P0)	None	10,000,000	None
duplex_7	10BaseT Switch_L11 Ethernet (P15)	humas_3 Ethernet (IF0 P0)	None	10,000,000	None
duplex_8	10BaseT Switch_L11 Ethernet (P2)	humas_4 Ethernet (IF0 P0)	None	10,000,000	None
duplex_9	10BaseT Switch_L11 Ethernet (P3)	humas_5 Ethernet (IF0 P0)	None	10,000,000	None
duplex_0	10BaseT Switch_L12 Ethernet (P0)	keuangan_1 Ethernet (IF0 P0)	None	10,000,000	None
duplex_1	10BaseT Switch_L12 Ethernet (P1)	keuangan_2 Ethernet (IF0 P0)	None	10,000,000	None
duplex_2	10BaseT Switch_L12 Ethernet (P10)	keuangan_3 Ethernet (IF0 P0)	None	10,000,000	None
duplex_3	10BaseT Switch_L12 Ethernet (P11)	keuangan_4 Ethernet (IF0 P0)	None	10,000,000	None
duplex_4	10BaseT Switch_L12 Ethernet (P12)	keuangan_5 Ethernet (IF0 P0)	None	10,000,000	None
Server <-> Switch_L12	10BaseT Server Ethernet (IF0 P0)	Switch_L12 Ethernet (P13)	None	10,000,000	None
Switch_L11 <-> Switch_L12	10BaseT Switch_L11 Ethernet (P4)	Switch_L12 Ethernet (P14)	None	10,000,000	None

Gambar 11 : Link Attributes

Gambar 11 menunjukkan semua *link attributes* yang ada, baik dari *switch 1* maupun *switch 2*. Terlihat masing-masing *port* yang dihubungkan yang nantinya akan digunakan sebagai parameter pengaturan anggota *VLAN*.

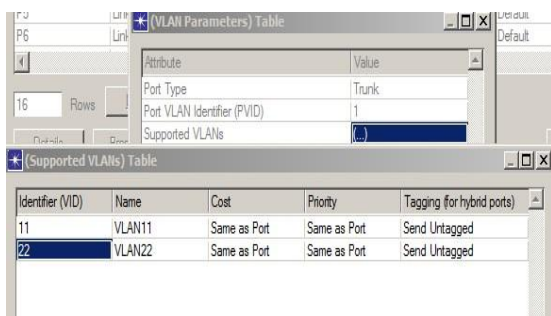
### Mengelompokkan *Client* Kedalam VLAN



Gambar 12 : Mengatur Kelompok VLAN11

Gambar 12 menunjukkan proses pengaturan *workstation* yang masuk dalam kelompok VLAN11 yang merupakan VLAN dari Divisi Humas. Pengaturan dilakukan di *switch* dengan cara Klik kanan pada *switch* lantai 1, kemudian pilih *Edit Attributes* dan klik dua kali pada *Switch Port Configuration*. Perhatikan kolom nama pada *Switch Port Configuration window*, edit *VLAN Parameters* pada nama *port-port* yang menghubungkan antara *switch 1* dengan *workstation* Divisi Humas yang merupakan *VLAN11* dengan melihat *link attributes* pada Gambar 4.16. Klik dua kali pada *VLAN Parameters*, ubah *Port Type* menjadi *Access* agar dapat menjadi *VLAN* dan pada *Supported VLANs* pilih VID dengan nilai 11 yang merupakan ID dari *VLAN* Divisi Humas. Pengaturan *VLAN 22* dan *VLAN 33* dilakukan dengan cara yang sama.

### Mengatur *Port Trunk*

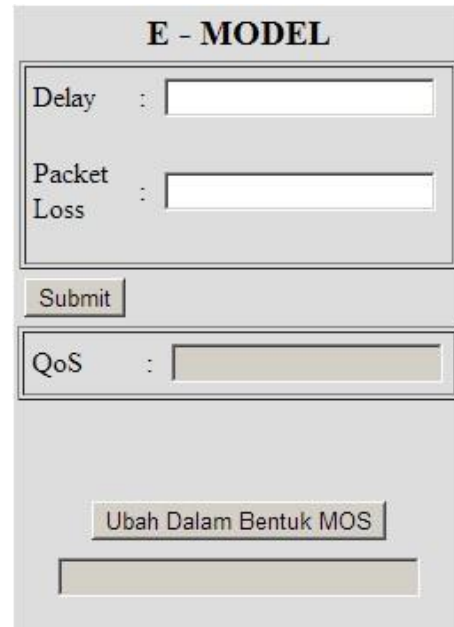


Gambar 13 : Mengatur *Port Trunk*

*Trunk* merupakan *port* yang menghubungkan *switch* lantai 1 dengan *switch* lantai 2. Dengan melihat *link attributes* pada Gambar 4.16 terdapat dua *trunk* yang masing-masing diatur pada tiap *switch*. Gambar 4.19 menunjukkan pengaturan pada *port trunk*. Klik kanan pada salah satu *switch*, kemudian pilih *Edit Attributes* dan klik dua kali pada *Switch Port Configuration*. Klik dua kali pada *VLAN Parameters*, ubah *Port Type* menjadi *Trunk* dan pada *Supported*

*VLANs* pilih VID dengan nilai 11 yang merupakan ID dari *VLAN* Divisi Humas dan VID dengan nilai 22 yang merupakan ID dari *VLAN* Divisi HRD.

### Antarmuka Sistem Perhitungan Menggunakan Metode *E-Model*



Gambar 14 : Antarmuka Sistem Perhitungan *E-Model*

Gambar 14 menunjukkan antarmuka sistem perhitungan menggunakan metode *E-Model* yang dibuat. Terdapat dua inputan yaitu *Delay* dan *Packet Loss*. Apabila kedua inputan tersebut telah diisi, selanjutnya di tekan tombol submit, akan memunculkan nilai *R-Factor* (nilai *QoS*) berdasarkan metode *E-Model*. Selanjutnya nilai *QoS* tersebut akan diubah dalam bentuk *MOS* yang nantinya akan menghasilkan *output* baik, kurang baik, sangat baik dan lain-lain.

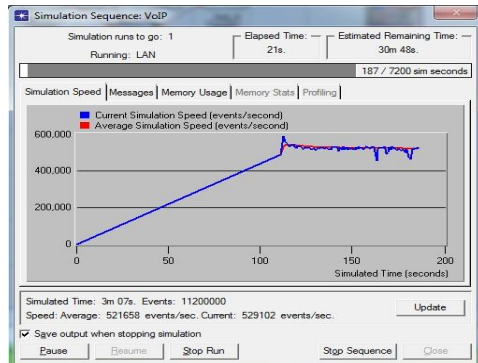
### HASIL DAN PEMBAHASAN

Pengujian skenario ini dilakukan sesuai dengan skenario pengujian yang sudah dibuat. Pengujian dilakukan dua kali, yaitu untuk menguji *QoS VoIP* pada jaringan *LAN*, dan untuk menguji *QoS VoIP* pada jaringan *VLAN*. Selanjutnya hasil pengujian dimasukkan ke dalam sistem perhitungan menggunakan metode *EModel* yang sudah dibuat.

#### Pengujian Skenario *LAN*

Pengujian skenario yang pertama adalah untuk menguji parameter kualitas layanan *VoIP* yang berjalan pada jaringan *LAN*. Pengujian ini bertujuan untuk mengetahui besar dari *delay* dan

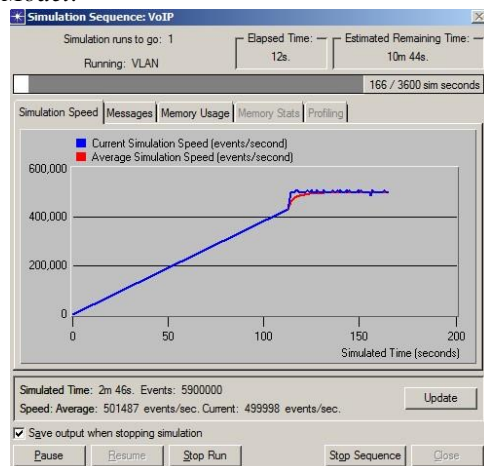
packet loss dari teknologi VoIP yang dibangun di atas jaringan LAN. Hasil tersebut nantinya akan digunakan sebagai inputan untuk menghitung kualitas layanan (QoS) VoIP yang berjalan pada jaringan LAN menggunakan metode penghitungan E-Model.



Gambar 15 : Proses Menjalankan Skenario LAN

### Pengujian Skenario VLAN

Pengujian skenario yang kedua adalah untuk menguji parameter kualitas layanan VoIP yang berjalan pada jaringan VLAN. Pengujian ini bertujuan untuk mengetahui besar dari delay dan packet loss dari teknologi VoIP yang dibangun di atas jaringan VLAN. Hasil tersebut nantinya akan digunakan sebagai inputan untuk menghitung kualitas layanan (QoS) VoIP yang berjalan pada jaringan VLAN menggunakan metode penghitungan E-Model.



Gambar 16 : Proses Menjalankan Skenario VLAN

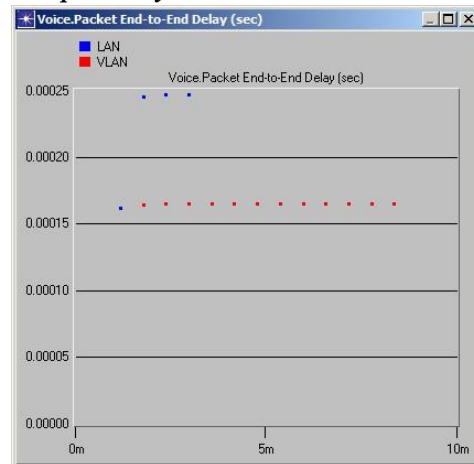
### Hasil Pengujian

Terdapat dua hasil pengujian yaitu hasil dari pengujian QoS VoIP pada jaringan LAN dan jaringan VLAN, yaitu output berupa nilai delay dan output berupa jumlah paket dikirim dan jumlah paket diterima yang digunakan untuk mengetahui besar packet loss.

Hasil-hasil pengujian tersebut nantinya akan digunakan sebagai inputan untuk sistem

penghitungan E-Model yang telah dibuat. Yang selanjutnya output nilai QoSnya diubah lagi ke dalam bentuk MOS sehingga akan diketahui kualitas layanan dari VoIP yang berjalan pada jaringan LAN maupun jaringan VLAN.

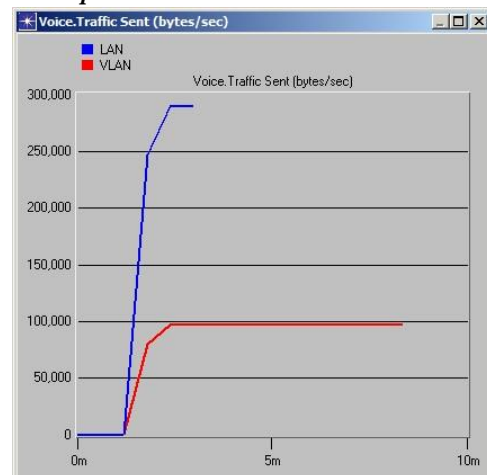
#### A. Output Delay



Gambar 17 : Output Delay

Gambar 17 menunjukkan output dari delay VoIP yang berjalan pada jaringan LAN maupun jaringan VLAN. Delay yang berjalan pada jaringan LAN ditunjukkan dengan warna biru dengan nilai 0,00025. Sedangkan delay yang berjalan pada jaringan VLAN ditunjukkan dengan warna merah dengan nilai 0,00016.

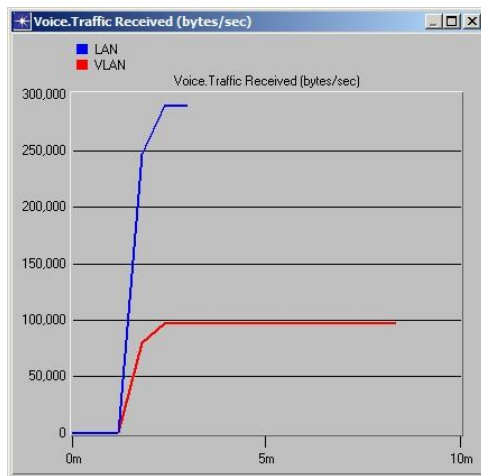
#### B. Output Packet Loss



Gambar 18 : Output Paket Dikirim

Gambar 18 menunjukkan output dari paket yang dikirim VoIP yang berjalan pada jaringan LAN maupun jaringan VLAN. Paket yang dikirim yang berjalan pada jaringan LAN ditunjukkan dengan warna biru dengan nilai 259.000. Sedangkan delay yang berjalan pada jaringan VLAN

ditunjukkan dengan warna merah dengan nilai 100000.



Gambar 19 : Output Paket Diterima

Gambar 19 menunjukkan output dari paket yang diterima *VoIP* yang berjalan pada jaringan LAN maupun jaringan VLAN. Paket yang diterima yang berjalan pada jaringan LAN ditunjukkan dengan warna biru dengan nilai 259.000. Sedangkan delay yang berjalan pada jaringan VLAN ditunjukkan dengan warna merah dengan nilai 100000.

Perbedaan besar *delay* dan *packet loss* antara *VoIP* yang berjalan pada jaringan LAN dan jaringan VLAN tidak terlihat dengan jelas (kecil) dapat disebabkan karena jaringan yang dibangun bersifat lokal.

### Penghitungan QoS

Penghitungan nilai *QoS VoIP* dilakukan menggunakan sistem penghitungan yang dibangun berbasis web berdasarkan metode *E-Model*.

### A. QoS VoIP Pada Jaringan LAN

**E - MODEL**

Delay : 0.00025

Packet Loss : 0

Submit

QoS : 87.1999665

Ubah Dalam Bentuk MOS

Baik

Gambar 20 : QoS VoIP Pada Jaringan LAN

Penghitungan *QoS VoIP* pada jaringan LAN dilakukan dengan memasukkan nilai *delay* dan nilai *packet loss* yang didapat dari OPNET ke dalam sistem penghitung. Selanjutnya didapat nilai *QoS* sebesar 87,1999665. Nilai tersebut kemudian diubah kedalam bentuk MOS menghasilkan nilai BAIK bagi kualitas *VoIP* yang berjalan pada jaringan LAN.

### B. QoS VoIP Pada Jaringan VLAN

**E - MODEL**

Delay : 0.00016

Packet Loss : 0

Submit

QoS : 87.19997856

Ubah Dalam Bentuk MOS

Baik

Gambar 21 : QoS VoIP Pada Jaringan VLAN

Penghitungan *QoS VoIP* pada jaringan *VLAN* dilakukan dengan memasukkan nilai *delay* dan nilai *packet loss* yang didapat dari OPNET ke dalam sistem penghitung. Selanjutnya didapat nilai *QoS* sebesar 87,19997856. Nilai tersebut kemudian diubah kedalam bentuk MOS menghasilkan nilai BAIK bagi kualitas *VoIP* yang berjalan pada jaringan *VLAN*.

### KESIMPULAN

Berdasarkan uraian pembahasan analisa dan pengujian yang telah dilakukan, maka dapat diambil kesimpulan terhadap Penentuan *Quality of Service (QoS) VoIP* Pada Jaringan *LAN* dan Jaringan *VLAN* Menggunakan Metode *E-Model* adalah sebagai berikut :

1. OPNET dapat digunakan sebagai alat untuk menentukan besar *delay* dan *packet loss* jaringan *VoIP*.
2. Dengan menggunakan metode *E-Model*, kualitas layanan *VoIP* yang berjalan pada jaringan *LAN* dan jaringan *VLAN* sama-sama BAIK.
3. Perbedaan terletak pada besar *delay* yang dihasilkan teknologi *VoIP* yang berjalan pada jaringan *LAN* dan jaringan *VLAN*, dimana *delay VoIP* pada *LAN* lebih besar daripada *delay VoIP* pada *VLAN*.

### DAFTAR PUSTAKA

- [1] Cole, R.G dan Rosenbluth, J.H. 2000. *Voice Over IP Performance Monitoring*. AT&T Laboratories. USA.
- [2] ITU-T Telecommunication Standardization Sector. (<http://www.itu.int/en/ITU-T/Pages/default.aspx>).
- [3] Kristof, Richard. 2001. *Administering Cisco QoS fo IP Network*. Syngress Publishing Inc. USA.
- [4] Local Area Network (LAN) dan Virtual Local Area Network (VLAN). (<http://www.library.binus.ac.id>)
- [5] Mirza, Muhammad. 2011. *Implementasi Teknologi VoIP Berbasis SIP Pada Jaringan VLAN*. Fakultas Teknik Universitas Wijaya Kusuma Surabaya
- [6] Purbo, Onno. W. 2007. *VoIP: Cikal Bakal "Telkom Rakyat" (Panduan Lengkap Setting VoIP)*. Prima Infosarana Media. Jakarta.
- [7] Purbo, Onno. W dan Thabratas Tharom. 2001. *VoIP: Voice Over Internet Protocol*. Elex Media Computindo. Jakarta.

- [8] Telecommunication and Internet Protocol Harmonization Over Network. (<http://etsi.org/WebSite/homepage.aspx>)
- [9] Verma, Pramode K dan Ling Wang. 2011. *Voice over IP Network: Quality os Service, Pricing and Security*. Springer. USA.



{Halaman sengaja dikosongkan}  
{Jurnal **Melek IT**}

(2)

## SISTEM MONITORING DAN KONTROL LAMPU GEDUNG

M Ali Mashudi, Tjatusari W<sup>2</sup>, Noven Indra Prasetya<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Wijaya Kusuma Surabaya

[Masyudi\\_ali@yahoo.co.id](mailto:Masyudi_ali@yahoo.co.id)

### ABSTRAK

Banyaknya titik lampu di setiap ruangan dalam suatu gedung, baik itu gedung perkantoran, perusahaan, dan kampus terdapat suatu permasalahan yaitu, mengetahui dan mengontrol lampu yang sudah mati atau masih menyala pada ruangan gedung yang tidak digunakan. Pada penelitian ini akan diimplementasikan sistem monitoring dan kontrol lampu gedung yang bertujuan untuk memberikan kemudahan pengawas atau pengelola gedung dalam memonitoring dan mengontrol lampu pada setiap ruangan lampu akan dimonitoring yang berbeda sehingga dalam pengontrolannya lebih mudah

Agar sistem yang akan dibangun dapat diimplementasikan, maka dibutuhkan beberapa perangkat *hardware*, dan *software* pendukung. Beberapa perangkat *hardware* yang digunakan yaitu *LDR* untuk membedakan intensitas cahaya dalam ruangan, *LED* untuk lampu, *arduino*. Sedangkan untuk *software*-nya adalah *softwareprocessing arduino* untuk menulis program yang akan dibuat ke dalam *arduino*. Di bagian Pada implementasi nantinya sistem ini akan dilakukan pada sebuah *maket* yang menggambarkan setiap ruangan dalam suatu gedung. Dengan adanya sistem ini, maka pengelola gedung dapat mengoptimalkan monitoring lampu sehingga dapat menginformasikan secara cepat kondisi lampu yang sedang menyala dan mati, menghasilkan suatu sistem monitoring yang dapat meng-handle beberapa tugas yang biasa dilakukan oleh petugas lapangan secara manual, dan juga membantu pengelola gedung dalam menekan biaya listrik lampu

**Kata Kunci :** Monitoring Lampu, Kontrol Lampu, Arduino

### PENDAHULUAN

#### Latar Belakang

Sistem kendali dapat dikatakan sebagai hubungan antara komponen yang membentuk sebuah konfigurasi sistem, yang akan menghasilkan tanggapan sistem yang diharapkan. Jadi harus ada yang dikendalikan, yang biasa disebut dengan kendali Masukan dan keluaran merupakan variabel. Keluaran merupakan hal yang dihasilkan oleh kendalian, artinya yang dikendalikan, sedangkan masukan adalah yang mempengaruhi kendalian, yang mengatur keluaran. Kedua dimensi masukan dan keluaran tidak harus sama.

Dengan pesatnya gedung – gedung perkantoran perusahaan dan kampus terdapat suatu permasalahan yaitu : terlalu banyak titik lampu yang sebgaiian sulit untuk mengetahui lampu yang sudah mati atau masih menyala pada ruangan gedung yang tidak digunakan. Dengan adanya permasalahan seperti ini kita akan memecahkan permasalahan pengelola gedung dengan membuat sistem control lampu gedung.

Dengan demikian dapat diatasi dengan membuat Sistem control lampu gedung, sehingga membantu pengelola gedung untuk tidak membutuhkan

pengawas dalam melakukan pengecekan dengan memasuki setiap ruangan dan juga membantu pengelola gedung dalam menekan biaya listrik lampu dan memudahkan pengawas mengontrol pada jam yang di tentukan untuk menyalakan (on) atau mematikan (off) lampu. Pada Tugas Akhir ini akan dibahas tentang implementasi web service berbasis sistem informasi geografis untuk pencarian rute menuju lokasi pariwisata dan estimasi biaya yang di keluarkan untuk sampai lokasi pariwisata.

### TINJAUAN PUSTAKA

#### Arduino

Arduino adalah pengendali mikro *singleboard* yang bersifat *open-source*, diturunkan dari *Wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardware*-nya memiliki prosesor Atmel AVR dan *software*-nya memiliki bahasa pemrograman sendiri.

Arduino merupakan *platform* yang terdiri dari *software* dan *hardware*. *Hardware* Arduino sama dengan microcontroller pada umumnya hanya pada *arduino* ditambahkan penamaan pin agar mudah diingat. *Software* Arduino merupakan *softwareopensource* sehingga dapat di *download*

secara gratis. *Software* ini digunakan untuk membuat dan memasukkan program ke dalam Arduino. Pemrograman Arduino tidak sebanyak tahapan microcontroller konvensional karena Arduino sudah didesain mudah untuk dipelajari, sehingga para pemula dapat mulai belajar microcontroller dengan Arduino.

### Kelebihan Arduino

Di bawah ini ada beberapa kelebihan Arduino.

1. *Open Source*  
*Hardware* maupun *software* Arduino adalah *open source*. Artinya kita bisa membuat tiruan atau *clone* atau *board* yang kompatibel dengan *board* Arduino tanpa harus membeli *board* asli buatan Italy. Kalau kita membuat *board* yang persis dengan desain asli, kita tidak akan dianggap membajak.
2. Tidak memerlukan chip programmer Chip pada Arduino sudah dilengkapi dengan *bootloader* yang akan menangani proses *upload* dari komputer. Dengan adanya *bootloader* ini kita tidak memerlukan chip programmer lagi, kecuali untuk menanamkan *bootloader* pada chip yang masih *blank*.
3. Koneksi USB  
Sambungan dari komputer ke *board* Arduino menggunakan USB, bukan serial atau parallel port. Sehingga akan mudah menghubungkan Arduino ke PC atau laptop yang tidak memiliki serial/parallel port.
4. Fasilitas chip yang cukup lengkap  
Arduino menggunakan chip AVR ATmega 168/328 yang memiliki fasilitas PWM, komunikasi serial, ADC, timer, interrupt, SPI dan I2C. Sehingga Arduino bisa digabungkan bersama modul atau alat lain dengan protokol yang berbeda-beda.
5. Ukuran kecil dan mudah dibawa Ukuran *board* Arduino cukup kecil, mudah di bawah kemana-mana bersama laptop atau dimasukkan ke dalam saku.
6. Bahasa pemrograman relatif mudah  
Walaupun bahasa pemrograman Arduino adalah bahasa C/C++, tetapi dengan penambahan *library* dan fungsi-fungsi standar membuat pemrograman Arduino lebih mudah dipelajari. Contoh, untuk mengirimkan nilai HIGH pada pin 10 pada Arduino, cukup menggunakan fungsi `digitalWrite(10, HIGH);` Sedangkan kalau menggunakan bahasa C aslinya adalah `PORTB |= (1 << 2);`
7. Tersedia *library* gratis Tersedia *library* yang sangat banyak untuk menghubungkan Arduino

dengan macam-macam sensor, aktuator maupun modul komunikasi.

Misalnya *library* untuk *mouse*, *keyboard*, servo, GPS, dsb. Berhubung Arduino adalah *open source*, maka *library-library* ini juga *open source* dan dapat di *download* gratis di website Arduino.

8. Pengembangan aplikasi lebih mudah dengan bahasa yang lebih mudah dan adanya *library* dasar yang lengkap, maka mengembangkan aplikasi elektronik relatif lebih mudah. Contoh, kalau kita ingin membuat sensor suhu. Cukup membeli sebuah IC sensor suhu (misalnya LM35) dan menyambungkan ke Arduino. Kalau suhu tersebut ingin ditampilkan pada LCD, tinggal membeli sebuah LCD dan menambahkan *library* LCD pada program yang sama, dan seterusnya.
9. Komunitas *open source* yang saling mendukung *Software* Linux, PHP, MySQL atau WordPress perkembangannya begitu pesat karena merupakan *software open source* dimana ada komunitas yang saling mendukung pengembangan proyek. Demikian juga dengan Arduino, pengembangan *hardware* dan *software* Arduino didukung oleh pencinta elektronika dan pemrograman di seluruh dunia. Contoh, *interface* USB pada Arduino Uno mengambil dari LUFA project. *Library* dan contoh-contoh program adalah sumbangan dari beberapa programmer mikrokontroler, seperti Tom Igoe, dsb.

### Bahasa Pemrograman Arduino

Arduino diciptakan untuk para pemula bahkan yang tidak memiliki basic bahasa pemrograman sama sekali karena menggunakan bahasa C++ yang telah dipermudah melalui *library*. Kenapa Arduino menggunakan bahasa C++, karena Bahasa C adalah bahasa yang sangat lazim dipakai sejak awal-awal komputer diciptakan dan sangat berperan dalam perkembangan *software*. Bahasa C++ telah membuat bermacam-macam sistem operasi dan compiler untuk banyak bahasa pemrograman, misalnya sistem operasi Unix, Linux, dsb. Bahasa C++ juga biasanya diajarkan di akademi dan perguruan tinggi selain bahasa pemrograman Basic atau Pascal dan Bahasa C++ adalah bahasa pemrograman yang sangat ampuh yang kekuatannya mendekati bahasa assembler. Bahasa C++ menghasilkan file kode objek yang sangat kecil dan dieksekusi dengan sangat cepat. Karena itu Bahasa C++ sering digunakan pada sistem operasi dan pemrograman mikrokontroler. Arduino menggunakan *Software Processing* yang

digunakan untuk menulis program kedalam Arduino. Processing sendiri merupakan penggabungan antara bahasa C++ dan Java. Software Arduino ini dapat di-install di berbagai operating sistem (OS) seperti: LINUX, Mac OS, Windows. Software IDE Arduino terdiri dari 3 (tiga) bagian:

1. Editor program, untuk menulis dan mengedit program dalam bahasa processing. Listing program pada Arduino disebut *sketch*.
2. *Compiler*, modul yang berfungsi mengubah bahasa processing (kode program) kedalam kode biner karena kode biner adalah satu-satunya bahasa program yang dipahami oleh microcontroller.
3. *Uploader*, modul yang berfungsi memasukkan kode biner kedalam memori microcontroller.

Struktur perintah pada arduino secara garis besar terdiri dari 2 (dua) bagian yaitu *void setup* dan *void loop*. Void setup berisi perintah yang akan dieksekusi hanya satu kali sejak arduino dihidupkan sedangkan void loop berisi perintah yang akan dieksekusi berulang-ulang selama arduino dinyalakan.

Kegunaan Arduino tergantung kepada kita yang membuat program. Arduino bisa digunakan untuk mengontrol LED, bisa juga digunakan untuk mengontrol helikopter. Contoh yang sudah pernah dibuat adalah MP3 player, pengontrol motor, mesin CNC, monitor kelembaban tanah, pengukur jarak, penggerak servo, balon udara, pengontrol suhu, monitor energi, stasiun cuaca, pembaca RFID, drum elektronik, GPS logger, monitoring bensin dan masih banyak lagi.

### Lampu LED

Kata LED merupakan singkatan dari *Light-emitting diode* (dioda cahaya) ialah suatu semikonduktor yang memancarkan cahaya monokromatik yang tidak koheren ketika diberi tegangan maju. Gejala ini termasuk bentuk elektroluminesensi. Warna yang dihasilkan bergantung pada bahan semikonduktor yang dipakai, dan bisa juga ultraviolet dekat atau inframerah dekat. Lampu LED memiliki beberapa keunggulan dari lampu konvensional yang sebelumnya kita gunakan (neon, bohlam dan lainnya) sehingga lampu Led layak disebut sebagai lampu masa depan dan mulai digunakan dari sekarang.

### Keuntungan dari penggunaan lampu LED

Beberapa keuntungan yang didapat apabila menggunakan lampu Led ketimbang lampu neon / pijar :

1. Dengan menggunakan lampu LED kita bisa menghemat tagihan listrik karena tahukah anda kalau lampu pijar hanya bisa mengubah sekitar 8 % dari konsumsi listrik yang digunakan menjadi cahaya. Bandingkan dengan lampu LED yang bisa mengubah dua kali lipatnya yaitu sekitar 1525%.
2. Led menghasilkan panas lebih sedikit. Dengan begitu selain hemat konsumsi listrik dari lampu juga hemat untuk konsumsi listrik pendingin.
3. Panas yang dihasilkan lampu yang sering kita gunakan selain tidak nyaman untuk mata juga bisa merubah warna dekorasi rumah. Apabila menggunakan Lampu Led hal tersebut bisa dihindari karena kualitas dari pencahayaan lampu dan manajemen panas LED lebih baik.
4. Lampu LED bebas dari bahaya merkuri jadi sangat aman digunakan dimana saja.
5. Lampu LED jauh lebih tahan lama, 60 kali dari lampu pijar dan 10 kali dari lampu neon.
6. Lampu LED memiliki desain yang elegan, bahkan bisa memperindah interior di ruangan.

### Prinsip Kerja LDR

Pada sisi bagian atas LDR terdapat suatu garis / jalur melengkung yang menyerupai bentuk kurva. Jalur tersebut terbuat dari bahan cadmium sulphida yang sangat sensitif terhadap pengaruh dari cahaya. Jalur cadmium sulphida yang terdapat pada LDR

Ketika cahaya mengenai cadmium sulphida, maka energi proton dari cahaya akan diserap sehingga terjadi perpindahan dari band valensi ke band konduksi. Akibat perpindahan elektron tersebut mengakibatkan hambatan dari cadmium sulphida berkurang dengan hubungan kebalikan dari intensitas cahaya yang mengenai LDR

### Karakteristik LDR

Karakteristik LDR terdiri dari dua macam yaitu Laju Recovery dan Respon Spektral:

1. Laju Recovery

Bila sebuah LDR dibawa dari suatu ruangan dengan level kekuatan cahaya tertentu kedalam suatu ruangan yang gelap, maka bisa kita amati bahwa nilai resistansi dari LDR tidak akan segera berubah resistansinya pada keadaan ruangan gelap

tersebut. Namun LDR tersebut hanya akan bisa mencapai harga di kegelapan setelah mengalami selang waktu tertentu. Laju recovery merupakan suatu ukuran praktis dan suatu kenaikan nilai resistansi dalam waktu tertentu. Harga ini ditulis dalam K /detik, untuk LDR type arus harganya lebih besar dari 200 K /detik (selama 20 menit pertama mulai dari level cahaya 100 lux), kecepatan tersebut akan lebih tinggi pada arah sebaliknya, yaitu pindah dari tempat gelap ke tempat terang yang memerlukan waktu kurang dari 10 ms untuk mencapai resistansi yang sesuai dengan level cahaya 400 lux.

## 2. Respon Spektral

LDR tidak mempunyai sensitivitas yang sama untuk setiap panjang gelombang cahaya yang jatuh padanya (yaitu warna). Bahan yang biasa digunakan sebagai penghantar arus listrik yaitu tembaga, aluminium, baja, emas, dan perak. Dari kelima bahan tersebut tembaga merupakan penghantar yang paling banyak digunakan karena mempunyai daya hantar yang baik.

## Resistor

Resistor adalah salah satu komponen elektronika yang berfungsi untuk menahan arus yang mengalir dalam suatu rangkaian/sistem elektronika. Resistor adalah komponen yang paling sering di gunakan dalam rangkaianrangkaian elektronika. Karena itu anda harus mampu mempelajari bagaimana menghitung nilai suatu resistor dan daya yang di gunakan apabila anda merancang suatu rangkaian yang menggunakan resistor. Dan anda harus mampu mengetahui rangkaian elektronika yang mengantong suatu resistor yang rusak atau terbakar.

Ada dua karakteristik resistor yang perlu di ketahui yaitu:

1. Nilai Resistansinya
  2. rating dayanya (Kemampuan untuk menahan arus yang mengalir pada resistor tersebut)
- Resistor mempunyai harga resistansi yang cukup banyak, mulai dari beberapa ohm di belakang koma sampai beberapa mega ohm didepan koma. Rating daya yang tertinggi da yang mencapai beberapa ratus watt dan yang terendah sampai mencapai 0,1 watt rating daya sangat penting , sebab ia menunjukkan daya maksimum yang bisa disipasikan tanpa menimbulkan panas-panas yang berlebihan yang dapat mengakibatkan kerusakan pada resistor tersebut. Disipasi artinya bahwa daya sebesar  $I^2R$  akan di buang kepadanya. Panas yang berlebihan dapat mengakibatkan terbakarnya resistor. Untuk memilih resistor dalam pemakaian apapun di butuhkan berbagai

macam pertimbangan yaitu: ukuran fisik, bentuknya, cara pemasangan dan penyambungannya pada rangkaian, harga resistansinya, disipasi dayanya, kemampuan menangani beban lebih, kehandalan, perubahan resistansi terhadap frekuensi dan terhadap tegangan yang jatuh padanya, ketahanan sebagai beban, pengaruh kondisi lingkungan dan umurnya.

## Personal Home Page (PHP)

PHP adalah singkatan dari "PHP Hypertext Preprocessor", yang merupakan sebuah bahasa scripting yang terpasang pada HTML. Sebagian besar sintaks mirip dengan bahasa C, Java dan Perl, ditambah beberapa fungsi PHP yang spesifik. Tujuan utama penggunaan bahasa ini adalah untuk memungkinkan perancang web menulis halaman web dinamik dengan cepat. Kelebihan PHP yang paling utama adalah pada konektivitasnya dengan *system database* di dalam web. Sistem *database* yang dapat didukung oleh PHP adalah : Oracle, MySQL, Sybase, PostgreSQL, dan lainnya. PHP dapat berjalan di berbagai sistem operasi seperti windows 98/NT, UNIX/LINUX, solaris maupun macintosh. PHP merupakan *software* yang *open source* yang dapat anda *download* secara gratis. *Software* ini juga dapat berjalan pada *web server* seperti PWS (Personal Web Server), Apache, IIS, AOLServer, fhttpd, phttpd dan sebagainya. PHP juga merupakan bahasa pemrograman yang dapat kita kembangkan sendiri seperti untuk menambah fungsifungsi baru. PHP mendukung komunikasi dengan layanan seperti protocol IMAP, SNMP, NNTP, POP3 dan bahkan HTTP. PHP dapat diinstal sebagai bagian atau modul dari apache *web server* atau sebagai CGI *script* yang mandiri.

## My Structured Query Language (MySQL)

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. MySQL adalah Relational Database Management System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat



komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh *user* maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh single user, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan *Interbase*.

### Data Flow Diagram

Sebelum mengimplementasi program, maka dilakukan pembuatan DFD atau Data Flow Diagram. DFD adalah suatu grafik yang menjelaskan sebuah sistem dengan menggunakan bentuk-bentuk atau symbol untuk menggambarkan aliran data dari proses-proses yang saling berhubungan. DFD menggambarkan input, proses, dan output yang terjadi dalam suatu sistem. DFD juga menggambarkan aliran data dalam sebuah sistem. Berdasarkan penjelasan di atas, dapat disimpulkan bahwa *Data Flow Diagram* (DFD) adalah diagram yang menggambarkan sistem-sistem yang saling berhubungan baik untuk sistem yang sudah ada maupun sistem yang baru. Tujuan menggunakan DFD yaitu untuk membuat dokumentasi dari sistem informasi yang ada, atau untuk menyusun dokumentasi untuk sistem informasi yang baru.

### Flowchart

Flowchart atau diagram alir merupakan sebuah diagram dengan simbol-simbol grafis yang menyatakan aliran algoritma atau proses yang menampilkan langkah-langkah yang disimbolkan dalam bentuk kotak, beserta urutannya dengan menghubungkan masing-masing langkah tersebut menggunakan tanda panah. Diagram ini bisa memberi solusi selangkah demi selangkah untuk penyelesaian masalah yang ada di dalam proses atau algoritma tersebut.

Tujuan utama dari penggunaan flowchart adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi

dan jelas dengan menggunakan simbol-simbol standart.

**Pedoman Dalam Membuat Flowchart** Pedoman yang perlu diperhatikan dalam membuat flowchart :

1. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
2. Kegiatan di dalam bagan alir harus ditunjukkan dengan jelas.
3. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhir.
4. Masing-masing kegiatan di dalam bagan alir sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan.
5. Masing-masing kegiatan di dalam bagan alir harus didalam urutan yang semstinya.
6. Kegiatan yang terpotong dan akan disambung ke tempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
7. Gunakan simbol-simbol alir yang standart.

### Jenis Flowchart

Terdapat 2 jenis flowchart yaitu sistem flowchart dan program flowchart.

1. **Sistem Flowchart**  
Sistem flowchart merupakan diagram alir yang menggambarkan suatu sistem peralatan komputer yang digunakan dalam proses pengolahan data serta hubungan antar peralatan tersebut. Sistem flow chart tidak digunakan untuk menggambarkan urutan langkah untuk memecahkan masalah, tetapi hanya untuk menggambarkan prosedur dalam sistem yang dibentuk.
2. **Flowchart Program**  
Merupakan bagan alir yang menggambarkan urutan logika dari suatu prosedur pemecahan masalah. Untuk menggambarkan flowchart program telah tersedia simbol-simbol standart. Berikut ini adalah gambar dari simbol-simbol standart yang digunakan pada flowchart program

## ANALISA DAN PERANCANGAN

Analisa sistem merupakan suatu kegiatan yang bertujuan untuk mempelajari serta mengevaluasi suatu bentuk permasalahan yang ada pada suatu sistem. Dalam analisa sistem akan di temukan masalah yang akan mempengaruhi kerja sistem. Agar sistem yang dirancang dapat berjalan sebagaimana sistemnya, perlu di lakukan analisa terhadap kerja sistem yang pada akhirnya bertujuan untuk pengembangan sistem.

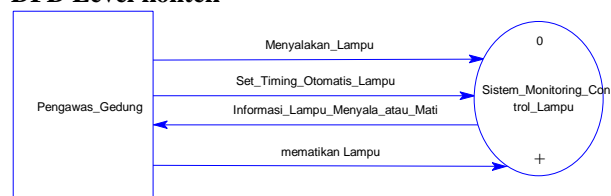
## Analisa Sistem

Seperti yang telah dijelaskan pada bab sebelumnya, sistem ini sebagai rencana untuk memonitoring lampu gedung yang merupakan sebuah inovasi dalam memonitoring ruangan gedung agar dapat mempermudah dalam memonitor sebuah ruangan. Pada implementasi nantinya sistem ini akan dilakukan pada sebuah maket yang menggambarkan setiap ruangan dalam suatu gedung. Setiap ruangan gedung akan di pasangi dengan satu LDR atau *Light Dependent Resistor* atau sensor cahaya. Pemasangan LDR di letakkan pada ruangan yang akan dimonitoring, fungsinya untuk membedakan intensitas cahaya dalam ruangan, karena dengan begitu dapat dijadikan penanda jika dalam keadaan intensitas cahaya lebih kecil atau hampir tidak ada cahaya dapat dikatakan atau di artikan lampu dalam ruangan tersebut mati sedangkan jika intensitas cahaya dalam ruangan terang maka dapat dikatakan lampu dalam ruangan tersebut menyala. LDR nantinya akan disambung dengan mikrokontoler, dengan menggunakan arduino. Dan nantinya akan di hubungkan dengan perangkat komputer pada ruangan pengawas gedung yang mengontrol atau memonitoring lampu, sehingga petugas penjaga dengan mudah mengetahui dan memonitoring setiap dalam sebuah gedung.

## Desain Sistem

Untuk memperjelaskan kebutuhan sistem maka akan dijelaskan desain kosep sistem yang akan dibuat berserta DFD

### DFD Level kontek

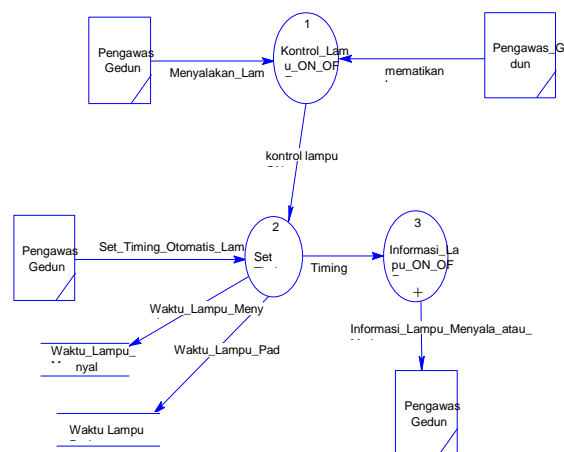


Gambar 3.1 level kontek

Keterangan:

DFD Level kontek memiliki 2 Entity, yang setiap mempunyai Tugas dan peranan fungsi yang berbeda – beda di dalam sistem monitoring control lampu Entity tersebut yaitu pengawas gedung.

## DFD Level 0

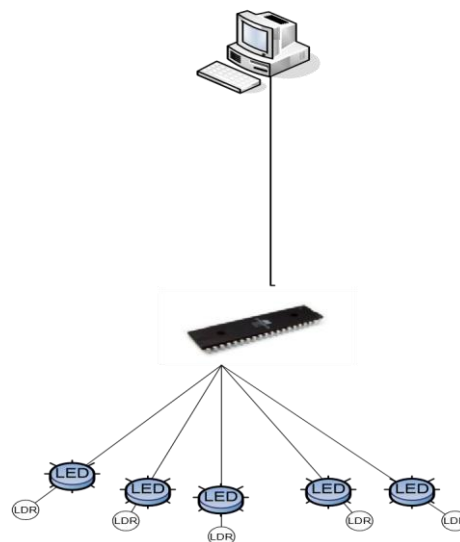


Gambar 3.2 level 0

Keterangan :

Pada gambar DFD level 0 ini memiliki 3 proses, yaitu, kontrol lampu on atau off dan Set timing, informasi lampu menyala lampu dan mematikan lampu dari 3 proses ini memiliki dengan ke 3 entity yang ada dan setiap relasi juga berhubungan dengan data store/penyimpanan data yaitu data store waktu\_lampu \_menyalah, waktu lampu padam

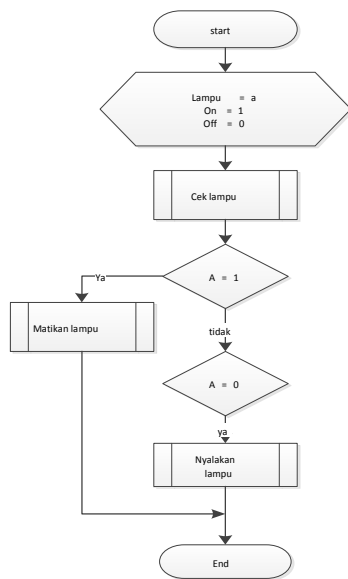
## Desain Sistem



Gambar 3.3 Desain Sistem

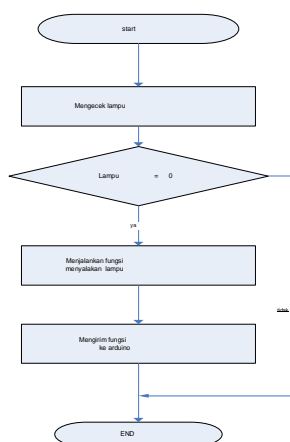
Berisi desain dari perangkat keras maupun lunak yang akan digunakan dalam melakukan simulasi sistem, penentuan perangkat lunak dan topologi yang akan digunakan, sekaligus pengaturan perangkat keras tersebut agar sesuai dengan topologi yang diinginkan.

### Flowchart monitoring lampu



Gambar 3.4 Flowchart Monitoring Lampu Gambar 3.4 flowchart monitor lampu. Proses monitoring lampu dilakukan dengan menginisialisasi lampu=a, on=1, off=0. Dilanjutkan dengan proses pengecekan lampu, apakah a=1 jika ya berarti lampu nyala sehingga terjadi proses matikan lampu, jika tidak apakah a=0 jika ya berarti lampu dalam keadaan menyala, sehingga terjadi proses mematikan lampu, dan jika tidak maka prosesnya selesai

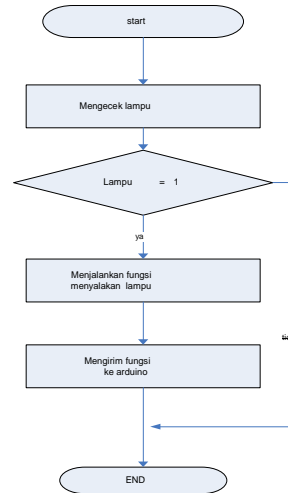
### Flowchart menyalakan lampu



Gambar 3.5 Flowchart menyalakan lampu Gambar 3.5 adalah flowchart saat menyalakan lampu. Arduino akan mengecek lampu. Jika lampu = 0 maka lampu tersebut dalam keadaan mati yang kemudian lampu tersebut dinyalakan. PC

mengirim fungsi ke arduino untuk menyalakan lampu sehingga lampu tersebut menyala.

### Flowchart mematikan lampu



Gambar 3.6 Flowchart mematikan lampu Gambar 3.6 adalah flowchart mematikan lampu. Arduino akan mengecek lampu. Jika lampu = 1 maka lampu tersebut dalam keadaan nyala yang kemudian lampu tersebut dimatikan. PC mengirim fungsi ke arduino untuk mematikan lampu sehingga lampu tersebut mati

## IMPLEMENTASI

### Kebutuhan Sistem

Dalam pembuatan Sistem monitoring dan kontrol lampu gedung diperlukan perangkat keras (*Hardware*) sebagai media untuk membuat sistem dan perangkat lunak (*Software*) untuk merancang dan membangun sistem.

### Perangkat keras ( *hardware* )

Adapun perangkat keras yang digunakan dalam perancangan dan pembuatan Sistem monitoring dan kontrol lampu gedung adalah :

1. Processor AMD turion x2 RM-70. 2.00GHz
2. Harddisk 35GB
3. RAM 1GB
4. VGA On-Board
5. Keyboard dan Mouse
6. Arduino Mega 2560
7. Lampu LED
8. LDR ( Light Dependent Resistor ) 10 K ohm
9. Resistore Ukuran 10 K ohm

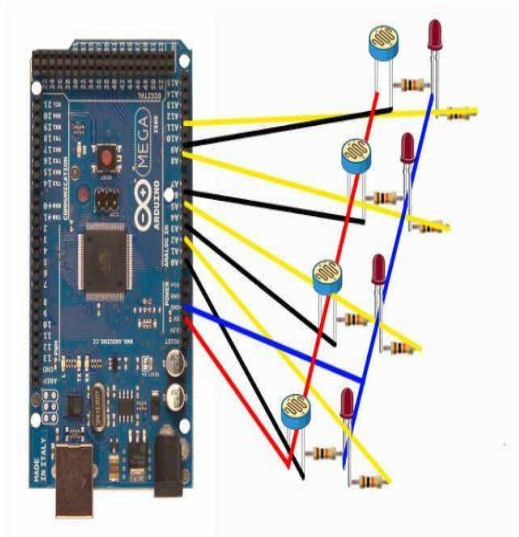
### Perangkat Lunak ( software )

Sedangkan perangkat lunak yang digunakan yaitu:

1. Sistem Operasi menggunakan Microsoft Windows 7
2. Arduino Software 1.0.3
3. Sybase Power Designer Versi 6.0 untuk pembuatan DFD
4. Microsoft Office Visio 2003 untuk membuat flowchart
5. MySQL sebagai Database penyimpanan data
6. Adobe Dreamweaver CS4 sebagai pembuat *form* aplikasi
7. Apache2 sebagai *web server*
8. Mozilla Firefox sebagai *web browser*

### Konfigurasi pada Arduino mega 2560 hardware

Berikut ini adalah skema rangkaian arduino mega 2560



Gambar 4.1 Rangkaian pada Arduino mega 2560

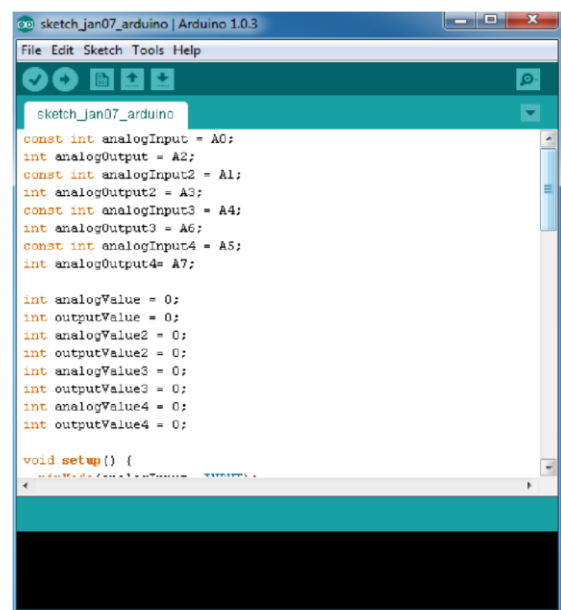
Pada gambar 4.1 merupakan rangkaian arduino mega 2560 yang terdiri dari sensor LDR ( 4buah ), lampu LED ( 4 buah ) dan resistor 1K ohm ( 8 buah ). Untuk satu lampu Led dan sensor LDR terdiri dari satu output dan langsung di sambungkan ke arduino pada port analog in, pada rangkaian ini, hasil output terdapat pada analog in 0, analog in 1, analog in 2, dan analog in 3. Sebelum mengirimkan output ke analog in kabel disambungkan dengan resistor dan diteruskan ke port GND pada arduino mega 2560. Dan salah satu kaki dari lampu LED dan LDR di sambungkan dengan power arduino mega 2560 inisialisasi komunikasi serial 9600 bps, input analog (analogRead) dan berfungsi untuk menulis nilai pada serial monitor.

### Konfigurasi pada Arduino 1.0.3 Software

Berikut ini adalah proses meng-*upload* *source code* dari arduino *software* ke arduino *hardware* setelah arduino *hardware* dihubungkan dengan PC

### Konfigurasi pada Windows 7 Degan Software Realter untuk Membaca Output Arduiono MEGA 2560

Agar Windows dapat membaca hasil output pada Arduino mega 2560 kami menggunakan software realtek



Gambar 4.2 Upload source code Arduino

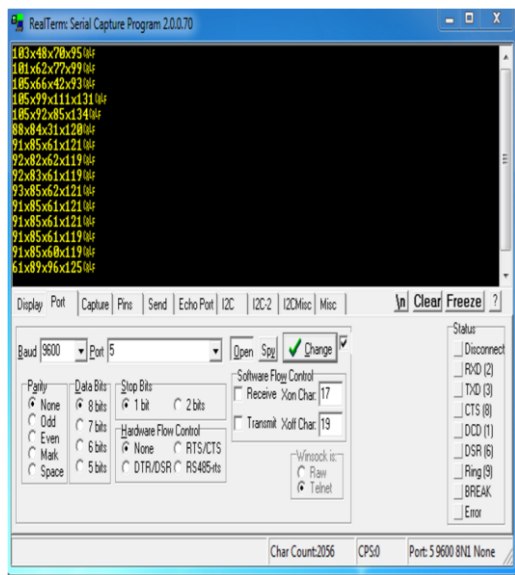
Pada gambar 4.2 merupakan interface dari Arduino Software yang berguna untuk melakukan *upload* atau transfer perintah-perintah berupa *source code* ke arduino *hardware*. Dengan menekan tombol upload pada arduino *software* (tombol di tunjukan dengan simbol arah panah mengarah ke kanan) maka perintah-perintah atau *source code* tersebut akan tertanam pada arduino Mega 2560 hardware.

Dan berikut ini adalah *source code* arduino secara keseluruhan.



Gambar 4.3 Source code Arduino

Pada gambar 4.3 dijelaskan tentang *source code* yang di *upload* ke dalam arduino mega 2560 *hardware*. Pada deklarasi pin input analog A0 sampai dengan pin input analog A3, pada deklarasi variable *analogValue* dan deklarasi *variable* *outputValue*, *analogInput* sebagai input,



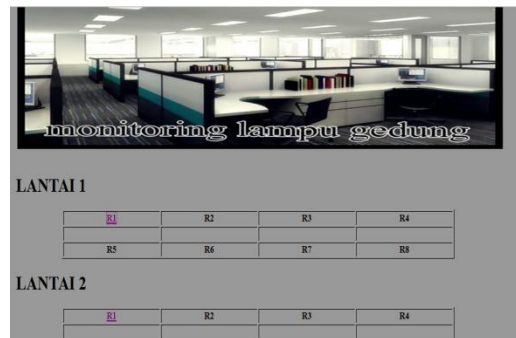
Gambar 4.4 Realterm untuk membaca sensor dari arduino ke windows 7

Pada gambar 4.4. dijelaskan tentang pada software realterm. menghubungkan tempat di mana port pada PC yang ditempati oleh kabel USB yang berhubungan dengan rangkaian arduino Mega 2560. Dan membaca dari hasil output dari rangkaian arduino tersebut.

## Desain Antarmuka

Pada sub-bab desain antarmuka, dijelaskan setiap desain antarmuka yang telah direncanakan sehingga pembaca bias lebih mudah dalam pemahaman menu dan fungsi masing-masing modul yang telah disediakan

## Monitoring lampu gedung



Gambar 4.5 monitoring lampu gedung

Pada gambar 4.5 merupakan monitoring lampu gedung yang digunakan untuk melihat lampu gedung di ruang berapa lampu nomor berapa masih menyala atau mati

## Form Ruang Gedung

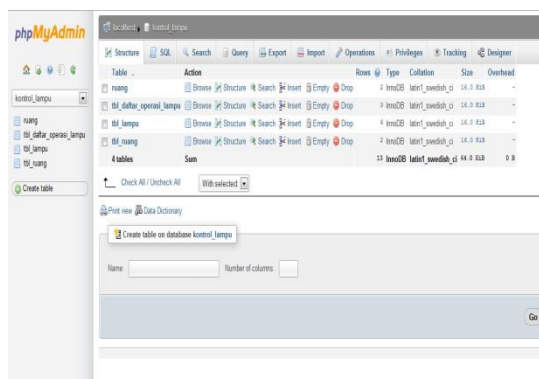


Gambar 4.6 Form Ruang Gedung keadaan lampu awal

Pada gambar 4.6 merupakan data ruang gedung yang digunakan untuk melihat berapa banyak lampu yang di gunakan atau menyala dan lampu tidak di gunakan dalam keadaan mati



## Database Kontro Lampu



Gambar 4.7 Database Kontrol Lampu

Pada gambar 4.7 merupakan database Kontrol Lampu yang digunakan untuk menyimpan data-data lampu dan ruang gedung. Diantaranya ada no ruang, lampu nomor berapa

## UJI COBA

Pada BAB V uji coba akan dilakukan pengujian supaya aplikasi yang telah dibuat bisa memenuhi tujuan pembuatan laporan tugas akhir ini. Skenario pengujian yang digunakan, akan disesuaikan dengan apa yang telah dijelaskan sebelumnya pada BAB III dan BAB IV.

Pada uji coba Sistem monitoring lampu gedung dilakukan beberapa pengujian sistem, setelah melalui tahap analisa dan perancangan sistem. Hasil uji coba digunakan untuk mengetahui apakah sistem dapat memenuhi tujuan yang akan dicapai seperti yang telah dijelaskan pada Bab Pendahuluan.

## Monitoring Lampu

Pada uji coba kali ini pada tempat ruangan gedung kita hanya membuat simulasi dengan membuat maket degan di umpamakan di dalam ruangan gedung yang hanya untuk 1 ruangan terdapat 4 (empat) lampu, dan berikut adalah gambar maket dalam 1 ruangan gedung



Gambar 5.1 maket dalam ruangan gedung

Pada gambar 5.1 maket ruangan gedung tersebut menempatkan lampu LED dan sensor LDR yang berjumlah 4 dan setiap ruangan terdapat 4 LED dan sensor LDR, dan berikut adalah gambar lampu LED dan sensor LDR yang terpasang pada miniatur tempat ruangan gedung



Gambar 5.2 Lampu LED dan Sensor LDR pada maket ruangan gedung

Pada gambar 5.2 ini untuk setiap ruangan gedung dipasang satu lampu LED dan sensor LDR guna mendeteksi pada ruangan tersebut sedang di gunakan atau kosong dengan menggunakan intensitas besarnya cahaya yang di tangkap oleh sensor LDR **Rangkaian pada arduino**

Rangkaian arduino dalam proses memonitoring lampu ruangan lagi menyalah atau mati ini kami menggunakan papan PCB, menggunakan beberapa kabel kecil yang berfungsi sebagai penghubung, dan berikut adalah gambar rangkaian pada arduino



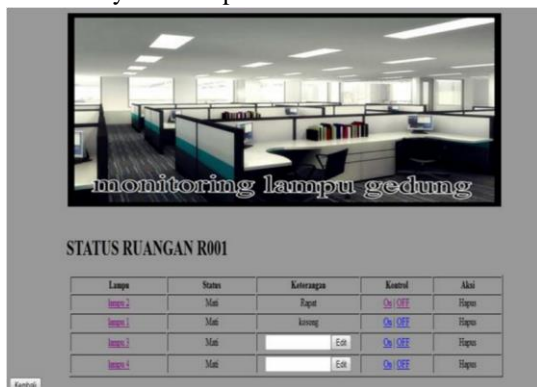
Gambar 5.3 Rangkaian pada arduino

Dalam gambar 5.3 alurnya adalah satu lampu LED dan sensor LDR mempunyai 2 (dua)

kaki, kaki yang pertama langsung dihubungkan dengan *port* power arduino sebesar 5V, dan pada kaki kedua ada percabangan, cabang pertama dihubungkan dengan resistor yang dengan nilai besaran resistor 10 K $\Omega$  dan dari resistor langsung diteruskan ke *port* GND pada arduino, sedangkan untuk percabangan kedua langsung dihubungkan dengan port analog in pada arduino yang berfungsi menghasilkan output. Untuk *port* analog in pada arduino MEGA 2560 *hardware* itu sendiri ada 6 port, diantaranya terdiri dari port analog in A0, port analog in A1, port analog in A2, port analog in A3, port analog in A4, dan port analog in A5

### Tampilan Monitoring Lampu Gedung

Untuk Tampilan Monitoring Lampu Gedung kami mengguna aplikasi berbasis web, guna untuk memudahkan pengawas gedung melakukan monitoring lampu tentang adanya tempat ruangan gedung yang lampu masih menyala dan mati, dari monitoring lampu tersebut pengawas gedung juga bisa meng-*input* data lampu masih menyala maupu mati.



Gambar 5.4 ruangan lampu keadaan mati

Pada gambar 5.4 tampilan monitoring lampu gedung pada menu sebelah kiri menunjuk tempat lampu . Pada gambar tersebut tercantum dalam 1 ruangan kondisi lampu. Jika ruangan itu lagi nyala lampu sedang di gunakan ruangan gedung, bila ada keterangan ruangan gedung di buat rapat maupun lainnya bilah keterangan udah lewat bias di hapus oleh pengawas gedung.

### Memonitoring Dan Kontrol Lampu Gedung

Memonitoring Lampu ini dilakukan jika terjadi adanya perubahan intensitas cahaya pada sensor LDR sehingga LDR mengirimkan output ke arduino dari arduino memonitoring melalui PC atau perangkat komputer, dan berikut adalah gambar dari ruangan gedung, rangkaian arduino dan PC atau perangkat komputer.



Gambar 5.5 Sistem monitoring dan kontrol lampu gedung

Pada gambar 5.5 semua sistem telah disambungkan, mulai dari maket ruangan gedung, rangkaian arduino hingga ke PC atau perangkat komputer yang berfungsi sebagai memonitoring

### Hasil Uji Coba Sistem

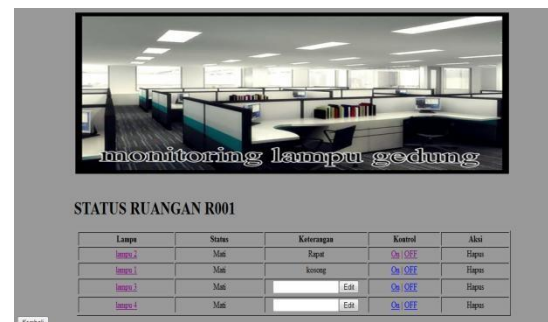
Berikut ini adalah hasil uji coba sistem yang telah kami lakukan berulang-ulang dengan berbagai variasi uji coba.

### Kondisi Ruangan Gedung

Pada saat kondisi ruangan gedung dengan kondisi tak ada satupun lampu yang menyala maka pada ruangan gedung tidak menandakan ruangan di gunakan.



Gambar 5.6 Kondisi ruangan sedang tidak di gunakan atau kosong



Gambar 5.7 monitoring lampu gedung



Pada gambar 5.7 Kondisi ruangan sedang tidak di gunakan atau kosong monitoring lampu gedung **Kondisi Ketika Dua Ruangan Terisi**

Pada saat kondisi ruangan terisi dengan kondisi dua lampu pada ruangan tersebut maka pada monitoring lampu menandakan bahwa lampu 1 dan 2 menyala.



Gambar 5.8 Monitoring lampu nyalah lampu 1 dan 2



Gambar 5.9 Monitoring ruangan 1 lampu nyalah lampu 1 dan 2

Pada Gambar 5.8 nampak Monitoring lampu nyalah lampu 1 dan 2

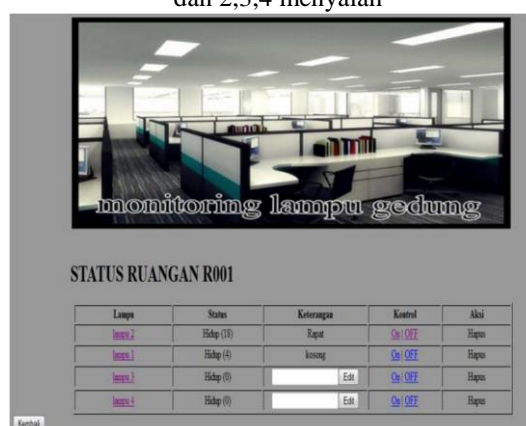
Gambar 5.9 Monitoring ruangan 1 lampu nyalah lampu 1 dan 2

### Kondisi Ketika Empat Lampu Nyala

Pada saat kondisi ruangan terisi dengan kondisi 4 lampu pada ruangan tersebut maka pada monitoring lampu menandakan bahwa lampu 1 dan 2,3,4 menyala.



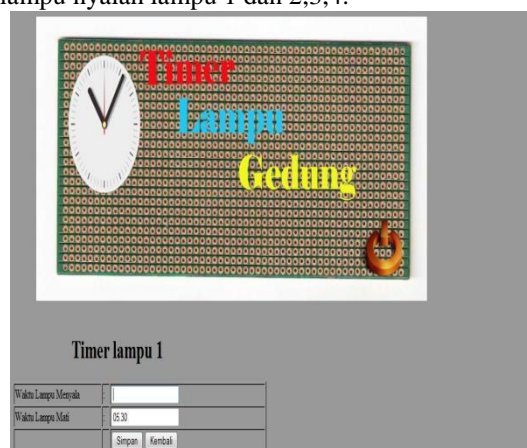
Gambar 5.10 Monitoring lampu nyalah lampu 1 dan 2,3,4 menyala



Gambar 5.11 Monitoring ruangan 1 lampu nyalah lampu 1 dan 2,3,4.

Pada Gambar 5.10 Monitoring lampu nyalah lampu 1 dan 2,3,

Gambar 5.11 Monitoring ruangan 1 lampu nyalah lampu 1 dan 2,3,4.



Gambar 5.12 timer lampu setiap lampu pada ruangan

Bila ruangan tersebut lagi digunakan sampek malam nanti user aka men timer waktu lampu akan mati jam 05.30 pagi

## PENUTUP

Secara umum sistem monitoring dan kontrol lampu gedung mengunaka web sebagai interfasenya degan hardware menggunakan arduino mega 2560 dapat membantu pengawas gedung untung memonitoring ruangan gedung dan perlantai Kegunaan dari aplikasi yang dibuat adalah untuk mengoptimalkan monitoring lampu gedung Namun tentu saja masih diperlukan beberapa penyempurnaan yang akan dipaparkan dalam pembahasan bab ini.

## Kesimpulan

Berdasarkan uraian pembahasan analisa dan pengujian yang telah dilakukan, maka dapat diambil kesimpulan terhadap sistem yaitu Sistem monitoring dan kontrol lampu degan hardware menggunakan arduino mega 2560 pengawas bias mengetahui lampu yang masih on atau off pada ruangan gedung.

## Saran

Saran untuk perkembangan lebih lanjut dalam masa mendatang pada Sistem monitoring dan kontrol lampu dapat di kembangkan degan secara detail pada titik lampu dan ruangan secara keseluruhan sama degan gedung – gedung perkantoran perusahaan dan kampus maupun lainnya.

## DAFTAR PUSTAKA

- [1] Author's Guide. *Datasheet AT89S51*. ([http://www.datasheet4u.com/html/A/T/8/AT89S51\\_ATMELCorporation.pdf.html](http://www.datasheet4u.com/html/A/T/8/AT89S51_ATMELCorporation.pdf.html)) Diakses pada tanggal 11 Oktober 2011
- [2] Bunafit, Nugroho. 2004. *JSP dan MySQL dengan editor Dreamweaver CS4*. Yogyakarta. Andi Yogyakarta.
- [3] Eko Putra, Afgianto. 2006. " *Belajar Mikrokontroler AT89C51/52/55 (Teori dan Aplikasi)* " Yogya Gava Media
- [4] Febriani, SKOM, MM. 2010. *Flowchart*. gunadarma.ac.id. Diakses tanggal 2011.
- [5] Syafii, M. 2004. *Membangun Aplikasi Berbasis PHP dan MySQL*. Andi Yogyakarta.

{Halaman sengaja dikosongkan}  
{Jurnal **Melek IT**}

## INTERFACING ARDUINO DENGAN ANDROID MENGGUNAKAN ANDROID DEBUG BRIDGE STUDI KASUS SISTEM OTOMASI RUMAH SEDERHANA

Badarrudin<sup>1</sup>, Maslihah<sup>2</sup>, Benny Y.V. Nasution<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Wijaya Kusuma Surabaya

[badar.rudin@gmail.com](mailto:badar.rudin@gmail.com)

### ABSTRAK

Teknologi *smart phone* berbasis android pada saat ini mengalami perkembangan yang pesat. Salah satu perkembangannya adalah mampu dikoneksikan dengan *embedded device*, contohnya arduino. Koneksi android dan arduino dapat dilakukan dengan dua arah, salah satunya adalah pengaturan dan pengontrolan *hardware*, misalnya lampu rumah, sensor gerak, dsb.

Pengontrolan arduino oleh perangkat android membutuhkan *user interface* agar dapat dimengerti dan mempermudah pengguna untuk berinteraksi dengan arduino. Permasalahan yang muncul adalah bagaimana cara komunikasi android ke arduino, dan sebaliknya. Media komunikasi yang digunakan adalah media kabel USB, dengan ADB (*Android Debug Bridge*) sebagai protokol komunikasi untuk menjembatani perangkat android dan *embedded device*.

Dalam penulisan ini, dibahas metode *interfacing* android dengan arduino, yang akan diujicobakan untuk pengaturan rumah sederhana. *Interface* pada android digunakan untuk mengendalikan arduino dan perangkat pendukungnya. Untuk mematikan dan menghidupkan lampu, dilakukan dengan menekan tombol pada interface yang telah dibangun. Untuk pembacaan data sensor, data hasil pembacaan ditampilkan menggunakan *interface* android. Dari interaksi android dan arduino dapat ditunjukkan tata cara komunikasi data antar perangkat.

**Kata kunci :** Android, Arduino, dan Android Debug Bridge

### PENDAHULUAN

#### Latar Belakang

Perkembangan teknologi telepon seluler (ponsel) cerdas yang sangat pesat membuat para pengembang perangkat lunak berlomba-lomba untuk mengembangkan aplikasi ponsel berbasis android. Hal ini dikarenakan aplikasi pengembang android yang dikenal sebagai Android SDK, menyediakan *tools* dan *API* untuk para pengembang aplikasi secara gratis. Dalam pengembangan aplikasi android diantaranya dalam segi kamera, GPS (*Global Positioning System*), kompas dan juga bisa digabungkan dengan *embedded device* seperti *accelerometer*.

Teknologi android telah banyak digunakan di negara-negara maju dan berkembang. Tetapi tidak semuanya dimanfaatkan sebagai sarana untuk mengendalikan perangkat yang tertanam *embedded device*. Di dalam *Embedded device* terdapat sebuah sistem operasi (*firmware*) yang dihubungkan pada *port-in* USB dengan kabel data pada android. Perangkat android dan *embedded device* masing-masing terdapat ADB (*Android Debug Bridge*), pada pertukaran data pada *embedded device* dengan android dijemput oleh ADB khusus untuk *embedded device* sehingga bisa

menghubungkan kedua perangkat tersebut, pengendalian *embedded device* oleh perangkat android bisa dilakukan dengan membuat tampilan (*user interfaces*) yang bagus dan mudah untuk dimengerti sehingga mempermudah bagi para pengguna (*users*) dalam pengaturan *embedded device* dan memperoleh hasil keluaran (*output*) data sesuai dengan keinginan pada perangkat android.

Dengan adanya perangkat android yang bisa dihubungkan dengan *embedded device*, Mampu menjadi teknologi alternatif untuk dapat dikembangkan dengan menggunakan sensor yang terdeteksi oleh android dan dapat mengatur menyalakan dan mematikan lampu. Oleh karena itu, penelitian ini membahas cara pembuatan desain *interface* pada android yang bisa dikomunikasikan dengan *embedded device*, yang akan diujicobakan untuk otomasi rumah sederhana.

### TINJAUAN PUSTAKA

#### Gambaran Sistem Otomasi Rumah Sederhana

Di dalam rumah terdapat beberapa ruangan yang perangkat-perangkatnya masih dijalankan secara *manual*, seperti dalam menyalakan /mematikan lampu kita masih dengan menggunakan saklar, dimana jika menyalakan/ mematikan lampu

kita harus pergi ketempat saklar masing-masing lampu, dan juga belum menerapkan sistem pendeteksian manusia pada ruangan rumah yang berguna untuk memberitahukan bahwa ada manusia yang memasuki ruangan rumah tersebut. Dengan menerapkan Sistem Otomasi Rumah Sederhana kita tidak perlu untuk berpindah-pindah ruangan untuk menyalakan/ mematikan lampu, karena telah menggunakan pengontrol utama.

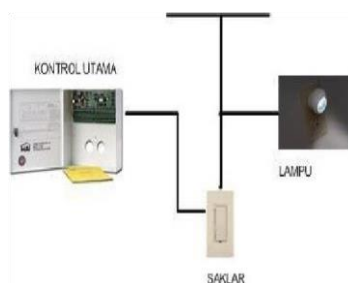
Sistem Otomasi Rumah Sederhana ini telah membanyak membantu penghuni rumah untuk menghemat tenaga dan waktu, serta untuk meminimalisasikan kerja manusia.

Sistem otomasi Rumah (SOR) adalah sebutan lain dari kata pada istilah Home Automation System. Sistem otomasi rumah adalah pengaturan perangkat-perangkat listrik rumah tangga, misalnya penerangan, musik atau temperatur ruang, yang dikendalikan lewat alat pengatur secara terpusat. Alat pengatur ini dilengkapi dengan fasilitas sistem penjadwalan pekerjaan (scheduler) untuk menghidup/matikan peralatan listrik rumah tangga secara otomatis.

### Cara Kerja Sistem Otomasi Rumah

Untuk memahami cara kerja SOR, kita akan melihat cara kerja saklar, bagaimana sebuah lampu dapat dikontrol oleh saklar. Dalam gambar 2.1 terlihat sebuah lampu dan saklar terhubung bersama ke kabel distribusi listrik utama. Bila Saklar dinyalakan maka akan ada aliran listrik dari sumber listrik ke lampu melalui saklar, sehingga lampu akan menyala. Inilah prinsip kerja sederhana, bagaimana cara menyalakan lampu melalui saklar.

Sedangkan apa yang dilakukan oleh SOR dapat dilihat pada gambar dibawah ini Kontrol Utama' SOR mengontrol lampu melalui titik saklar ini. Sehingga seperti halnya saklar juga bisa mengontrol lampu.



Gambar 2.1. Kontrol Utama Mengontrol lampu melalui Saklar

Karena 'Kontrol Utama' sudah bisa mengendalikan Lampu, maka berbagai alat dan teknologi akses dapat bergabung ke kontrol utama.

Dalam gambar 2.3 dapat kita lihat ber macam-macam alat dan teknologi akses dapat mengontrol lampu melalui kontrol utama. Seperti halnya lampu ini, titik beban listrik lainnya juga dapat dikendalikan oleh 'Kontrol Utama' dan berbagai alat dan teknologi aksesnya.



Gambar 2.2. Mengontrol Lampu melalui Kontrol Utama

### Serial Komunikasi (USB)

Berbagai jenis perangkat komunikasi bisa menggunakan keuntungan dari USB. Spesifikasi ini menyediakan model untuk perangkat telekomunikasi, seperti telepon, modem analog, dan perangkat jaringan. Ini menggambarkan Spesifikasi:

1. Antarmuka dalam Class Komunikasi
2. Komunikasi Class Perangkat
3. Data Class Interface

Kerangka untuk membangun perangkat komunikasi:

1. Perakitan struktur USB yang relevan dan benar dalam konfigurasinya.
2. Komunikasi Kelas interface dan penggunaannya.
3. Data Kelas interface dan penggunaannya.
4. Penggunaan jenis kelas tambahan atau antarmuka vendor tertentu.

Implementasi perangkat komunikasi, seperti telepon dan modem analog.

### Karakteristik fungsi

Bagian ini menjelaskan karakteristik fungsi dari Perangkat Komunikasi. Komunikasi Antarmuka dan data Interface, termasuk:

a. Organisasi Perangkat :

1. Persyaratan hasil akhir .
2. Membangun antarmuka dari hasil akhir.
3. Membangun konfigurasi dari berbagai antarmuka, beberapa diantaranya ditentukan oleh spesifikasi perangkat lain.
4. Mengidentifikasi kelompok interface dalam konfigurasi yang membuat unit fungsi dan menetapkan master antarmuka bagi kelompok masing-masing.

b. Perangkat operasi

Meskipun spesifikasi ini mendefinisikan kedua komunikasi Antarmuka dan data Interface, mereka adalah dua system komunikasi yang berbeda. Semua perangkat komunikasi harus memiliki antarmuka untuk mengelola perangkat sebagai perangkat komunikasi menggunakan kode Komunikasi. Selain itu, perangkat ini memiliki beberapa jumlah antarmuka lain yang digunakan untuk transmisi data aktual. Data Interface mengidentifikasi interface transmisi data saat data tidak cocok dengan struktur atau model penggunaan perangkat jenis lain, seperti Audio.

### Protocol Data Wrapper

Untuk mendukung protokol dalam perangkat, data dan perintah antara host dan perangkat harus mempertahankan permintaan mereka. Hal ini memastikan bahwa antrian protokol yang dirancang untuk dijalankan dalam sistem operasi ini secara real time dapat dibagi menjadi dua bagian yang berjalan pada masing-masing perangkat. Oleh karena itu, perintah dan data untuk protokol harus dimultiplexing ke antarmuka menggunakan kabel komunikasi yang sama. Kabel komunikasi ini juga memiliki fasilitas untuk mengirim data ke setiap lapisan stack. Setiap protokol menentukan bagaimana mendefinisikan intruksi protokol secara khusus dan bidang data akan melintasi ujung antarmuka. Host dan perangkat mengizinkan fitur-fitur wrapper pada saat protokol dari data yang ditetapkan. Ini adalah tugas dari host dan perangkat untuk berkomunikasi satu sama lain melalui beberapa antarmuka lainnya, seperti Antarmuka Komunikasi untuk menentukan protokol. Wrapper tersebut tidak digunakan jika tidak ada protokol yang ditetapkan. Ini adalah pengaturan untuk menggunakan wrapper jika protokol yang ditetapkan bisa menggunakannya, itu adalah hal yang harus dilakukan jika menggunakan wrapper dan jika protokol membutuhkan hal itu. Untuk mengaktifkan berbagai jenis antrian protokol yang ditemukan pada perangkat komunikasi, dua bentuk umum telah didefinisikan untuk header wrapper data

### Sensor Gerak PIR (Passive InfraRed)

Sensor gerak PIR (*Passive Infra Red*) adalah sensor yang berfungsi untuk pendeteksi gerakan yang bekerja dengan cara mendeteksi adanya perbedaan atau perubahan suhu sekarang dan sebelumnya. Sensor gerak menggunakan modul pir sangat simpel dan mudah diaplikasikan karena Modul PIR hanya membutuhkan tegangan input DC 5V cukup efektif untuk mendeteksi gerakan hingga

jarak 5 meter. Ketika tidak mendeteksi gerakan, keluaran modul adalah LOW. Dan ketika mendeteksi adanya gerakan, maka keluaran akan berubah menjadi HIGH. Adapun lebar pulsa HIGH adalah  $\pm 0,5$  detik. Sensitivitas Modul PIR yang mampu mendeteksi adanya gerakan pada jarak 5 meter memungkinkan kita membuat suatu alat pendeteksi gerak dengan keberhasilan lebih besar. Dengan output yang hanya memberikan 2 logika High dan Low ini kita dapat membuat aplikasi sensor gerak yang bervariasi. Misal kita ingin langsung aplikasikan pada alarm, kita tinggal membuat rangkaian driver untuk mengaktifkan alarm tersebut. Atau misal ingin digunakan untuk mengaktifkan lampu, maka tinggal dibuat driver untuk memberikan sumber tegangan ke lampu. Modul sensor gerak PIR memiliki output yang langsung bisa dihubungkan dengan komponen digital TTL atau CMOS dan juga dapat langsung dihubungkan ke mikrokontroler.

### Protocol Data Wrapper

Untuk mendukung protokol dalam perangkat, data dan perintah antara host dan perangkat harus mempertahankan permintaan mereka. Hal ini memastikan bahwa antrian protokol yang dirancang untuk dijalankan dalam sistem operasi ini secara real time dapat dibagi menjadi dua bagian yang berjalan pada masing-masing perangkat. Oleh karena itu, perintah dan data untuk protokol harus dimultiplexing ke antarmuka menggunakan kabel komunikasi yang sama. Kabel komunikasi ini juga memiliki fasilitas untuk mengirim data ke setiap lapisan stack. Setiap protokol menentukan bagaimana mendefinisikan intruksi protokol secara khusus dan bidang data akan melintasi ujung antarmuka. Host dan perangkat mengizinkan fitur-fitur wrapper pada saat protokol dari data yang ditetapkan. Ini adalah tugas dari host dan perangkat untuk berkomunikasi satu sama lain melalui beberapa antarmuka lainnya, seperti Antarmuka Komunikasi untuk menentukan protokol. Wrapper tersebut tidak digunakan jika tidak ada protokol yang ditetapkan. Ini adalah pengaturan untuk menggunakan wrapper jika protokol yang ditetapkan bisa menggunakannya, itu adalah hal yang harus dilakukan jika menggunakan wrapper dan jika protokol membutuhkan hal itu. Untuk mengaktifkan berbagai jenis antrian protokol yang ditemukan pada perangkat komunikasi, dua bentuk umum telah didefinisikan untuk header wrapper data

### Sensor Gerak PIR (Passive InfraRed)

Sensor gerak PIR (*Passive Infra Red*) adalah sensor yang berfungsi untuk pendeteksi

gerakan yang bekerja dengan cara mendeteksi adanya perbedaan atau perubahan suhu sekarang dan sebelumnya. Sensor gerak menggunakan modul pir sangat simpel dan mudah diaplikasikan karena Modul PIR hanya membutuhkan tegangan input DC 5V cukup efektif untuk mendeteksi gerakan hingga jarak 5 meter. Ketika tidak mendeteksi gerakan, keluaran modul adalah LOW. Dan ketika mendeteksi adanya gerakan, maka keluaran akan berubah menjadi HIGH. Adapun lebar pulsa HIGH adalah  $\pm 0,5$  detik. Sensitifitas Modul PIR yang mampu mendeteksi adanya gerakan pada jarak 5 meter memungkinkan kita membuat suatu alat pendeteksi gerak dengan keberhasilan lebih besar. Dengan output yang hanya memberikan 2 logika High dan Low ini kita dapat membuat aplikasi sensor gerak yang berfariatif. Misal kita ingin langsung aplikasikan pada alarm, kita tinggal membuat rangkaian driver untuk mengaktifkan alarm tersebut. Atau misal ingin digunakan untuk mengaktifkan lampu, maka tinggal dibuat driver untuk memberikan sumber tegangan ke lampu. Modul sensor gerak PIR memiliki output yang langsung bisa dihubungkan dengan komponen digital TTL atau CMOS dan juga dapat langsung dihubungkan ke mikrokontroler.

### **User Interface**

Dalam sebuah aplikasi Android, user interface dibangun dengan menggunakan *View* dan *ViewGroup* objek. Ada banyak jenis *view* dan *viewGroup*, masing-masing adalah keturunan dari *View* kelas. *View objects* adalah unit dasar dari ekspresi antarmuka pengguna pada platform Android. Kelas *View* berfungsi sebagai dasar untuk subclass disebut "widget", yang menawarkan objek UI sepenuhnya dilaksanakan, seperti bidang teks dan tombol. Kelas *ViewGroup* berfungsi sebagai dasar untuk subclass disebut "layout", yang menawarkan berbagai jenis arsitektur tata letak, seperti linear, tabular dan relatif.

*View objects* adalah struktur data yang sifatnya menyimpan parameter tata letak dan konten untuk area persegi tertentu dari layar. *view objects* menangani pengukuran sendiri, tata letak, gambar, perubahan fokus, gulir, dan kunci / isyarat interaksi untuk daerah persegi panjang pada layar di mana ia berada. Sebagai objek dalam antarmuka pengguna, *view* juga merupakan titik interaksi bagi pengguna dan penerima dari peristiwa interaksi.

### **Android SDK**

Android SDK merupakan sebuah tool development yang memungkinkan membuat aplikasi untuk platform Android. Android SDK sendiri mencakup

dari sampel project dengan kode sumbernya, tool development, sebuah emulator, dan librari (baca: file pendukung) yang dibutuhkan untuk membangun aplikasi Android. Aplikasi ini ditulis menggunakan bahasa pemrograman Java, dan dijalankan pada Dalvik Debug Monitor Server (ddms), yang merupakan mesin virtual yang dirancang khusus untuk penggunaan embedded yang berjalan di atas kernel Linux.

### **ADT (Android Development Tool)**

Google menyediakan Android Development Tools (ADT) untuk mengembangkan aplikasi Android dengan Eclipse. ADT adalah seperangkat komponen (plug-in) yang menambah IDE Eclipse dengan kemampuan pengembangan Android. ADT berisi semua fungsi yang diperlukan untuk membuat, mengkompilasi, debug dan menyebarkan aplikasi Android dari IDE Eclipse. ADT juga memungkinkan untuk membuat dan memulai AVD. Android Development Tools (ADT) menyediakan editor khusus untuk file sumber, misalnya file layout. Editor ini memungkinkan untuk beralih antara representasi XML dari file dan user interface yang lebih kaya melalui tab di bagian bawah editor.

### **ADB**

ADB adalah *Android Debug Bridge* dari Google. Ini adalah program command-line yang dijalankan dari PC, dengan ponsel yang terhubung melalui USB, yang bertujuan untuk menjalankan perintah yang harus dilakukan dengan telepon. Misalnya, seseorang dapat menginstal APK pada telepon dengan menggunakan perintah 'adb install apkname.apk'. Untuk menggunakan ADB, diperlukan Google debugging driver USB yang sudah diinstal pada PC, dan USB debugging sudah diaktifkan pada telepon.

### **SQLite Android**

SQLite adalah database Open Source yang tertanam ke Android. SQLite mendukung fitur database relasional standar seperti sintaks SQL, transaksi dan pernyataan siap. Selain itu hanya memerlukan sedikit memori pada saat runtime (sekitar 250 KByte). SQLite mendukung tipe data TEXT (mirip ke String di Jawa), INTEGER (mirip dengan yang lama di Jawa) dan REAL (mirip dengan ganda di Java). Semua jenis lain harus dikonversi ke pada bidang ini sebelum menyimpannya dalam database.

SQLite tersedia pada setiap perangkat Android. Menggunakan SQLite database di Android tidak memerlukan setup database atau



administrasi. Hanya dengan menentukan SQL untuk bekerja dengan database dan database dikelola secara otomatis. Bekerja dengan database bisa lambat. Oleh karena itu dianjurkan untuk melakukan tugas tersebut di latar belakang, misalnya melalui AsyncTask . Jika aplikasi membuat sebuah database, maka database ini disimpan di direktori.

### SQLiteOpenHelper

Membuat dan meng-upgrade database dalam aplikasi Android, biasanya menggunakan subclass SQLiteOpenHelper . Di kelas ini, perlu mengganti metode onCreate() untuk membuat database dan onUpgrade() untuk meng-upgrade database dalam kasus perubahan dalam skema database.

Primary key dari tabel database selalu harus menggunakan identifier \_id karena beberapa fungsi Android bergantung pada standar ini. Sebuah praktek terbaik adalah untuk menciptakan per meja kelas terpisah yang menentukan statis onCreate() dan onUpdate() metode. Metode ini kemudian disebut dalam metode yang sesuai SQLiteOpenHelper. Penerapan cara ini SQLiteOpenHelper tidak akan bisa besar bahkan jika memiliki beberapa tabel.

### SQLite Database

SQLite database di Android dan menyediakan metode untuk membuka, query, update dan menutup database. Para execSQL() metode memungkinkan untuk mengeksekusi SQL secara langsung. Objek ContentValues memungkinkan untuk mendefinisikan kunci / nilai untuk insert dan update. Kuncinya adalah kolom dan nilai adalah nilai untuk kolom ini. Query dapat dibuat melalui metode.rawQuery() yang menerima SQL atau query() yang menyediakan sebuah antarmuka untuk menentukan data yang dinamis atau SQLiteQueryBuilder .

### Mikrokontroler

Mikrokontroler adalah suatu chip berupa IC (Integrated Circuit) yang dapat menerima sinyal input, mengolahnya dan memberikan sinyal output sesuai dengan program yang diisikan ke dalamnya. Sinyal input mikrokontroler berasal dari sensor yang merupakan informasi dari lingkungan sedangkan sinyal output ditujukan kepada aktuator yang dapat memberikan efek ke lingkungan. Jadi secara sederhana mikrokontroler dapat diibaratkan sebagai otak suatu perangkat/produk yang mampu berinteraksi dengan lingkungan sekitarnya.

### Arduino

Arduino adalah pengendali mikro singleboard yang bersifat open-source, diturunkan dari Wiring platform, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. Hardwarenya memiliki prosesor Atmel AVR dan softwarenya memiliki bahasa pemrograman sendiri. Arduino adalah kit mikrokontroler yang serba bisa dan sangat mudah penggunaannya. Untuk membuatnya diperlukan chip programmer (untuk menanamkan bootloader Arduino pada chip). Arduino merupakan single board hardware yang open-source dan juga softwarenya pun dapat kita nikmati secara open source juga. Disini software arduino dapat dijalankan di multi platform, yaitu linux, windows, atau juga mac. Hardware arduino merupakan mikrokontroler yang berbasis AVR dari ATMEGA yang didalamnya sudah diberi bootloader dan juga sudah terdapat standart pin I/O nya.

Saat ini Arduino sudah sangat populer dan sudah banyak dipakai untuk membuat proyek-proyek seperti drum digital, pengontrol LED, web server, MP3 player, pengendali robot, pengendali motor, sensor suhu/kelembaban, pengontrol kamera, dsb. Arduino terdiri dari hardware berupa Arduino Board dan software berupa arduino IDE (Integrated Development Environment). Arduino dihubungkan dengan komputer melalui koneksi USB. Setelah itu kita bisa mulai menulis program menggunakan Arduino IDE untuk ditanam pada Arduino Board tersebut.

## ANALISA DAN PERANCANGAN

### Perencanaan metode komunikasi data

Sistem komunikasi pada android ke Arduino bekerja seperti client server. Pada android sebagai server dan arduino sebagai client, dalam komunikasi pertukaran data pada android ke arduino dijumpai dengan ADB ( *Android Debug Bridge* ) yang dihubungkan melalui kabel USB.

Hubungan android dan arduino membutuhkan ADB sebagai penerjemah antara bahasa android ke bahasa arduino, begitu juga sebaliknya. Agar ADB bisa diakses pada android yaitu membuka shell dalam android agar sistem dalam android bisa diakses penuh, kemudian memulai akses ADB pada android.

Pengiriman data pada android berupa data angka yang diproses dengan ADB yang dikirimkan pada arduino melewati port local TCP yang akan diterima oleh arduino, data yang telah dikirim oleh android diproses oleh ADB arduino untuk diterjemahkan dengan bahasa arduino berupa data byte.

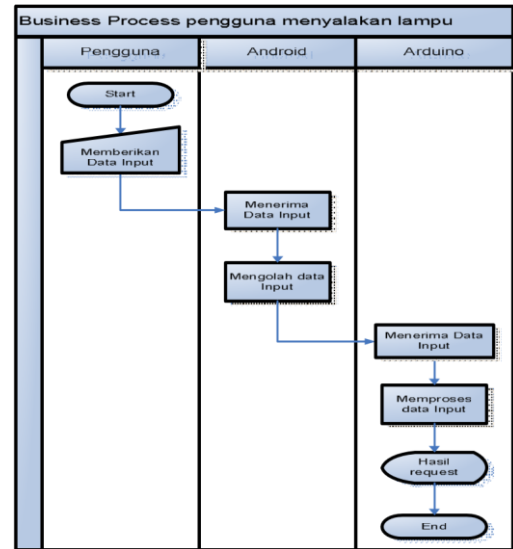
Pengontrolan ini dapat dilakukan dengan 2 arah :

1. Android mengirim data pada arduino Android mengirim data melalui *interface* untuk mengontrol arduino dengan memfungsikan arduino untuk menghidupkan dan menyalakan lampu serta mengaktifkan alarm sensor gerak yaitu *passive infrared sensor (PIR)*. Dalam pengiriman data android ke arduino berupa Array data, android memberikan data berupa 1 atau 0, bahwa data yang bernilai 1 berfungsi untuk menyalakan lampu dan data bernilai 0 berfungsi untuk mematikan lampu yg terdapat pada arduino.
2. Android menerima data dari Arduino Arduino mengirim data ke android yang berfungsi untuk mengetahui kondisi sensor apakah terdeteksi atau tidak terdeteksi gerakan, jika sensor terdeteksi gerakan maka android memberikan bunyi alarm dan menampilkan visual pada *interface* android.

Arduino mendapatkan data dari sensor dan sensor memberikan data berupa data HIGH atau LOW, kemudian arduino mengelola data dari sensor dengan merubah data High atau Low menjadi data digital, bahwa data High yaitu angka 1, dan data Low yaitu angka 0. Pengiriman data arduino pada android ini menggunakan Array data yang terdiri dari 2 indek Array, yaitu indek [0] mewakili nama port dan indek [1] mewakili kondisi atau status dari port.

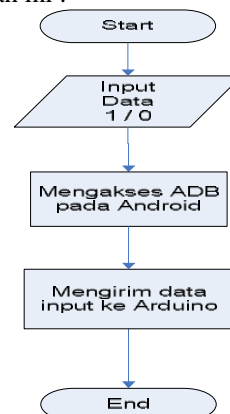
Lebih jelasnya dalam pembahasan ini bisa dilihat dalam Business Process dan Flowcart di bawah ini :

1. Proses mengirim dan pengolahan data Android ke arduino untuk menyalakan lampu dijelaskan pada Business Process di bawah ini



Gambar 3.1. Business Process Pengguna Menyalakan Lampu

Pada Gambar 3.1. Ketika pengguna menekan tombol pada interface, maka interface memberikan inputan kepada android, kemudian android menerima inputan dan mengelola inputan data untuk diterjemahkan dari bahasa android menjadi bahasa arduino, setelah diproses maka android mengirim hasil data inputan ke arduino, setelah itu arduino menerima hasil inputan dan memproses data request dari android agar menghasilkan data output yang diujicobakan untuk menghidupkan lampu LED. Pada Gambar 3.1. Business Proses dijelaskan dengan beberapa Flowcart seperti di bawah ini: Pengguna memberikan inputan pada android melalui *user interface* untuk menjalankan fungsi pada Arduino, untuk lebih jelasnya ada pada gambar Flowcart 3.2. di bawah ini :



Gambar 3.2. Flowcart Input Data pada Android

Pada Gambar 3.2 ketika pengguna memberikan inputan pada Android, dan data itu berupa angka 1 atau 0 setelah pengguna memberikan inputan, android mengakses ADB server, kemudian data dari android dikirim ke arduino.

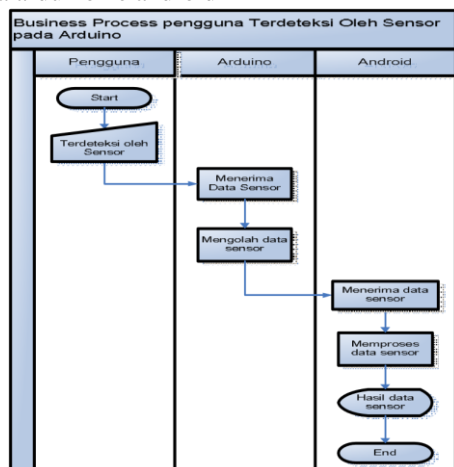
Memproses data fungsi dari android yang telah dijelaskan pada gambar Flowcart 3.2. untuk dikelola pada Arduino agar fungsi bisa dijalankan



Gambar 3.3. Flowcart Memproses Data Fungsi pada Arduino

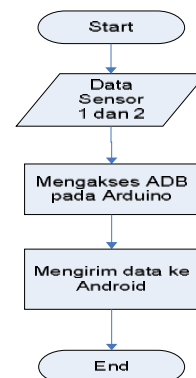
Setelah mendapatkan hasil data fungsi dari android maka pada Arduino menerima fungsi kemudian memproses data fungsi tersebut agar dijalankan oleh arduino untuk menyalakan atau mematikan lampu sesuai dengan inputan yang telah diberikan pada android.

## 2. Proses mengirim dan pengolahan data sensor pada arduino ke android



Gambar 3.4. Business Process Pengguna Terdeteksi oleh Sensor

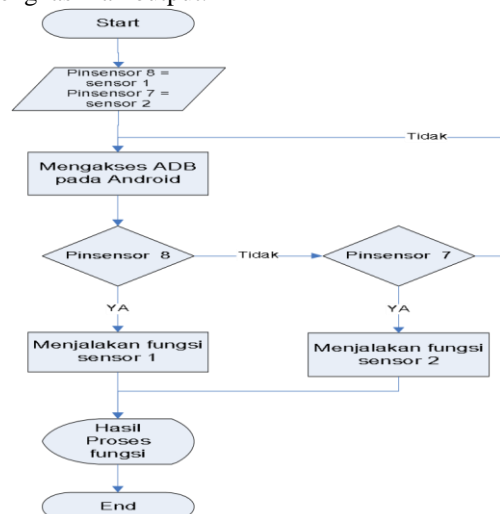
Pada Gambar 3.4. Pengguna terdeteksi oleh Sensor yang ada pada Arduino maka Arduino menerima inputan sensor dan mengelola inputan sensor untuk diterjemahkan dari bahasa arduino menjadi bahasa android, setelah diproses maka arduino mengirim hasil data inputan ke android, setelah itu android menerima hasil inputan dan memproses data request dari arduino agar menghasilkan data output yang ditampilkan pada interface android. Pada Gambar 3.4. diatas dijelaskan dengan beberapa Flowcart seperti di bawah ini: Data sensor yang telah diberikan oleh pengguna akan diproses dan dikirimkan pada Android



Gambar 3.5. Flowcart Mengirim data Arduino pada Android

Pada Gambar 3.5. pengguna memberikan inputan melalui Sensor dan inputan data sensor diteruskan pada Arduino, setelah itu Arduino mengakses ADB untuk memproses inputan agar bisa dimengerti oleh android.

Pada pengolahan fungsi data sensor yang terdapat pada gambar 3.5. akan diproses kembali pada Android agar input data fungsi menghasilkan output.



Gambar 3.6. Flowcart Memproses Data Fungsi

### Arduino pada Android

Pada Gambar 3.6. Pada proses ini android menerima data dari arduino yaitu pinsensor 8 dan pinsensor 7, jika android menerima data dari arduino maka ADB server terhubung dan android mengelola pinsensor 8 atau pinsensor 7, maka sensor 1 atau sensor 2 akan memberikan kondisi pada android untuk menjalankan bunyi alarm dan menampilkan visual pada interface android.

### Perencanaan Database

Pada aplikasi komunikasi data arduino dan android perlu direncanakan pembuatan database yang terdiri dari 2 tabel.

#### 1. Tabel Log Kejadian

ID	Integer auto increment (10)
Log	Varchar (255)

Tabel 3.1 Rancangan Table Log Kejadian

Kolom Log berfungsi sebagai menyimpan kejadian yang terjadi, yang berisi kolom ID dan kolom Log

#### 2. Tabel Alarm

ID	Integer auto increment (10)
Alarm	Varchar (1)

Tabel 3.2 Rancangan Table Alarm

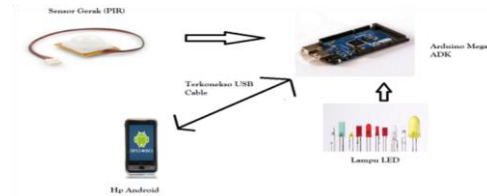
Kolom alarm berfungsi sebagai kolom untuk menyimpan status alarm, jika bernilai 1 alarm On dan jika 0 maka alarm Off

### Perencanaan Kebutuhan Sistem

Kebutuhan skenario meliputi kebutuhan *hardware* maupun *software* yang dipergunakan pada masing-masing skenario sesuai dengan konsep yang telah dibuat.

### Arsitektur Hardware

Pada system ini dibutuhkan beberapa komponen hardware yang saling terhubung, yaitu : sensor gerak *PIR*, lampu LED, arduino mega ADK, power supply 5V/1A dan handphone yang berbasis android. Untuk memperjelas konsep sistem dapat dilihat pada gambar di bawah ini.



Gambar 3.7. Arsitektur Sistem

Pada Gambar 3.7. menjelaskan tentang hubungan antar hardware pada system. Pada Arduino Mega ADK dipasang lampu LED dengan konektivitas serial. Begitu juga dengan sensor gerak *PIR*. Sedangkan Android dengan Arduino Mega ADK menggunakan *Port-in* USB sebagai pertukaran data. Android meminta data pada Arduino Mega ADK, dan Arduino Mega ADK dapat mengirim data ke android.

### Spesifikasi Hardware

Ada beberapa spesifikasi hardware dan peralatan yang digunakan pada pengujian Penelitian ini termasuk dalam peralatan mikrokontroler seperti Arduino Mega ADK pada gambar 3.10 yaitu sebagai berikut :

1. Arduino Mega ADK
2. *Smart Phone* Android
3. Kabel USB
4. Sensor *PIR* (*Passive Infra Red*)
5. Lampu LED
6. Kabel Pelangi
7. Kabel power
8. Power Supply 5V

### Spesifikasi Software

Selain kebutuhan hardware yang telah dijelaskan, dibutuhkan juga software untuk dapat dilakukannya pengujian. Software-software yang digunakan adalah sebagai berikut:

1. Sistem operasi android Versi 2.3.3
2. Android SDK
3. Arduino Development GUI Versi 1.0.2
4. Eclipse Helios

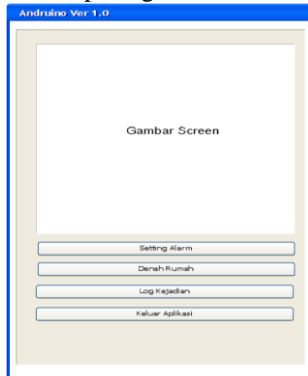
### Perencanaan Desain *interface*

Perancangan desain *interface* merupakan upaya menggambar tampilan dari system yang akan dibuat. Perlunya perancangan desain *interface* agar tampilan system dapat menemukan kenyamanan dan fungsional dari system.

### Desain menu sistem

Menu system merupakan fasilitas yang ada pada system. Terdapat empat menu yang tersusun berjenjang kebawah, yaitu menu setting alarm, menu denah rumah, menu log kejadian dan menu keluar

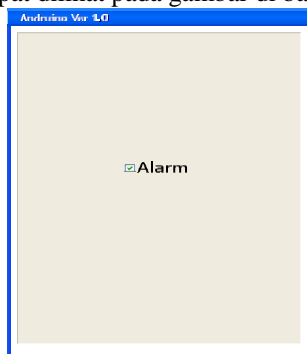
dari system. Pada bagian atas terdapat gambar yang berisi nama aplikasi atau logo aplikasi. Agar lebih jelas, dapat dilihat pada gambar dibawah ini.



Gambar 3.8 Desain Menu

### Desain halaman setting alarm

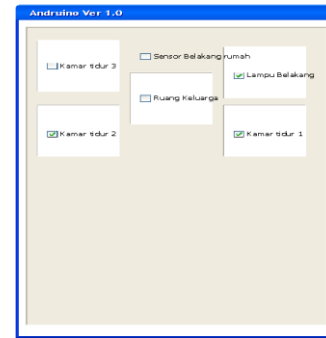
Menu setting alarm mempunyai fungsi untuk mematikan atau menghidupkan alarm. Setting dilakukan dengan memberi centang untuk menghidupkan alarm dan menghilangkan centang untuk mematikan alarm. Desain tampilan setting alarm dapat dilihat pada gambar di bawah ini.



Gambar 3.9 Desain Setting Alarm

### Desain halaman denah rumah

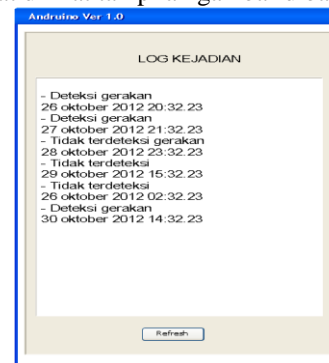
Menu denah rumah menampilkan posisi lampu dan sensor gerak yang dipasang didalam rumah. Melalui menu ini pengguna dapat mematikan lampu atau menghidupkan lampu. Begitu juga pengguna dapat mematikan sensor gerak atau mematakannya. Tanda check menandakan hidup, sedangkan uncheck menandakan mati. Tampilan dari menu denah rumah dapat dilihat pada gambar di bawah ini.



Gambar 3.10 Desain Denah

### Desain halaman log kejadian

Menu log kejadian menjadi tempat pencatatan aktifitas yang tertangkap oleh sensor gerak. Pada bagian bawah terdapat tombol refresh untuk mendapatkan log kejadian terbaru. Agar lebih jelas dapat dilihat tampilan gambar dibawah ini.



Gambar 3.11. Desain Log Kejadian

## IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari rancangan sistem *aplikasi* yang telah dibuat pada bab sebelumnya. Bagian implementasi sistem *aplikasi* ini implementasi proses, dan implementasi antarmuka.

### Spesifikasi Pendukung Sistem

Dalam pembuatan aplikasi sistem ini, dibutuhkan perangkat sistem pendukung seperti perangkat keras (*hardware*) dan perangkat lunak (*software*). Dibutuhkan spesifikasi hardware yang memadai, sehingga aplikasi dapat berjalan dengan baik.

### Perangkat Keras (*Hardware*)

Di dalam perancangan dan pembangunan aplikasi, penulis membutuhkan perangkat keras. Perangkat keras yang dimaksud untuk menunjang pembuatan *aplikasi* adalah dengan spesifikasi sebagai berikut :

1. Handphone dengan spesifikasi minimal operating system android froyo, ram 512 Kb dan penyimpanan memory minimal 1GB.
2. Arduino Mega ADK 256
3. Kabel koneksi USB
4. Kabel pelangi
5. Lampu led sebagai simulasi lampu rumah
6. Sensor PIR sebagai pendeteksi gerak

### Perangkat Lunak (Software)

Perangkat lunak yang digunakan dalam perancangan dan pembuatan aplikasi android dan arduino adalah sebagai berikut :

- a. Java editor Eclipse Helios
- b. Arduino Development GUI Versi 1.0
- c. ADT
- d. Android Platform tool Api 8
- e. Java JDK 1.6
- f. ADB Library

### Koneksi dengan Arduino Mega ADK

Agar perangkat dapat terhubung dengan perangkat mikro arduino mega adk, dibutuhkan koneksi yang menjembatani. Penghubung kedua perangkat ini dinamakan Android debug bridge (ADB). ADB dapat diartikan sebagai penerjemah kedua bahasa yang berbeda antar kedua perangkat. ADB dalam arduino berbentuk

Libraries yang terletak dalam folder libraries pada Arduino Development GUI Versi 1.0.2, kemudian libraries tersebut dipanggil oleh program yang telah dibuat, agar ADB tersebut dapat dijalankan pada program arduino yang dibuat. Berikut adalah potongan code di bawah ini, pemanggilan ADB yang terdapat dalam libraries pada aplikasi Arduino Development GUI Versi 1.0.2.

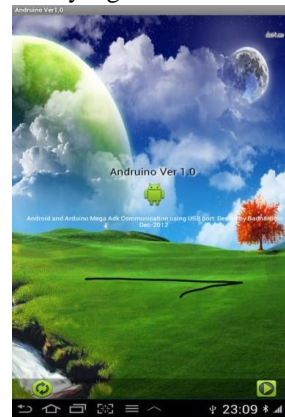
```
#include <SPI.h>
#include <Adb.h>
void adbEventHandler(Connection
    * connection, adb_eventType
    event, uint16_t length, uint8_t * data)
{
    if(event=
        ADB_CONNECTION_RECEIVE)
```

Dan Berikut adalah potongan code koneksi menggunakan ADB.

```
Server mServer = null; deklarasi untuk
server mServer = new Server(4568); setting
port TCP IP pada port 4568 mServer.start();
untuk menjalankan Server
mServer.addListener(this); membuat server
untuk mendeteksi koneksi
secara otomatis pada port USB
```

### Implementasi Desain Antarmuka

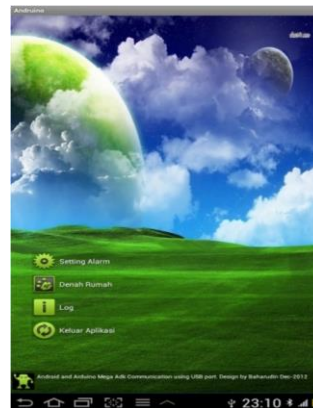
Setelah perancangan desain antarmuka yang dilakukan pada bab 3, maka pada bab ini menjelaskan implementasi bagaimana penerapan desain antarmuka yang telah dibuat.



Gambar 4.1 Splash Screen Aplikasi.

### Halaman menu aplikasi

Halaman menu aplikasi menampilkan menu-menu yang tersedia pada aplikasi. Terdapat 4 menu yaitu menu setting alarm, denah rumah, log kejadian dan keluar aplikasi. Tampilan dari halaman menu aplikasi dapat dilihat pada gambar di bawah ini.



Gambar 4.2 Halaman Menu Aplikasi

### Halaman setting alarm

Di bagian ini pengguna dapat mematikan atau menghidupkan alarm. Alarm "ON" mempunyai tanda tombol berwarna hijau dan terdapat tulisan "ON", sedangkan jika alarm "OFF"



tombol berwarna merah dengan tulisan "OFF". Tombol refresh di bagian bawah layar berfungsi untuk menampilkan perubahan setting terakhir. Tampilan halaman menu setting alarm dapat dilihat pada gambar di bawah ini.

#### Halaman Splash Screen

Halaman splash screen muncul pertama kali saat aplikasi dijalankan. Berfungsi sebagai jeda waktu aplikasi untuk menyiapkan resource berjalannya sistem. Pada halaman ini terdapat keterangan nama aplikasi dan deskripsi aplikasi. Tampilan dari halaman Splash Screen dapat dilihat pada gambar di bawah ini.



Gambar 4.3 Halaman Setting Alarm

Jika tombol *dicheck*, maka variabel bernilai 1 disimpan ke database, jika tombol *diuncheck* maka variabel bernilai 0 akan disimpan ke database alarm. Dapat dilihat pada tabel di bawah ini

RecNo	_id	status
Click here to define a filter		
1	1	1

Tabel 4.1 Tabel Status Alarm On

Pada tabel di atas bahwa kolom status bernilai 1 maka menandakan bahwa alarm sedang diaktifkan

RecNo	_id	status
Click here to define a filter		
1	1	0

Tabel 4.2 Tabel Status Alarm Off

Pada tabel di atas jika tombol di tekan menjadi Off, maka kolom status berubah menjadi nilai 0 maka menandakan bahwa alarm sedang dinonaktifkan.

#### Halaman denah rumah

Halaman denah rumah berfungsi sebagai media mematikan maupun menghidupkan sensor gerak dan lampu. Pada halaman ini digambarkan denah dari rumah dengan tombol untuk menghidupkan atau mematikan lampu. Terdapat 5 area yang dapat dimatikan atau dihidupkan lampunya. Sensor gerak berada pada halaman

belakang rumah. Tombol berwarna hijau mempunyai arti bahwa lampu sedang menyala dan sensor gerak tidak mendeteksi adanya gerakan, sedangkan warna merah menerangkan bahwa lampu dalam keadaan mati dan sensor gerak mendeteksi adanya gerakan.

Di bagian kiri bawah terdapat keterangan dari sensor gerak. Jika ada gerakan yang terdeteksi, maka tulisan akan menerangkan "Gerakan terdeteksi". Sedangkan jika tidak mendeteksi gerakan, tulisan berubah menjadi "Tidak ada gerakan terdeteksi!". Tampilan dari halaman denah rumah dapat dilihat pada gambar di bawah ini.



Gambar 4.4 Halaman Denah Rumah

#### Halaman log kejadian

Halaman ini mencatat setiap perubahan yang terjadi pada sensor gerak. Jika terjadi gerakan, maka sistem akan mencatat gerakan terjadi pada tanggal dan jam kejadian. Log kejadian dicatat setiap terjadi pendeteksian gerakan oleh sensor. Tampilan dari halaman logi kejadian dapat dilihat pada gambar dibawah ini.



Gambar 4.5 Halaman Log Kejadian

Berikut adalah tampilan Tabel database Log Kejadian yang berada dalam sistem android yang dapat dilihat pada gambar di bawah ini.



RecNo	_id	action	time
Click here to define a filter			
1	1	Deteksi gerakan halaman depan,L1 0 ,L2 0 ,L3 0 ,L4 0 ,L5 0	Jan 19, 2013 04:05:10 PM
2	2	Deteksi gerakan halaman depan,L1 0 ,L2 0 ,L3 0 ,L4 0 ,L5 0	Jan 19, 2013 4:05:10 PM
3	3	Deteksi gerakan halaman depan,L1 0 ,L2 0 ,L3 0 ,L4 0 ,L5 0	Jan 19, 2013 4:07:30 PM
4	4	Deteksi gerakan halaman belakang,L1 1 ,L2 0 ,L3 1 ,L4 0 ,L5 0	Jan 19, 2013 4:10:16 PM
5	5	Deteksi gerakan halaman belakang,L1 1 ,L2 0 ,L3 1 ,L4 0 ,L5 0	Jan 19, 2013 4:11:22 PM
6	6	Deteksi gerakan halaman belakang,L1 1 ,L2 0 ,L3 1 ,L4 0 ,L5 1	Jan 19, 2013 4:11:39 PM

Tabel 4.3 Tabel Database Log Kejadian

Pada Tabel di atas adalah record data kejadian dari hasil Log Kejadian, data tersebut tersimpan jika ada pendeteksian dari sensor, data yang terdapat pada kolom action yang berisi keberadaan dan kondisi sensor yaitu deteksi gerakan halaman belakang atau deteksi halaman depan dan juga berisi kondisi lampu L1 0 yaitu LED 1 dalam keadaan mati dan L1 1 yaitu LED 1 dalam keadaan hidup, begitu juga pada L2, L3, L4, L5 dan pada kolom Time yang berisi waktu pendeteksian sensor.

### Implementasi Arduino dan Android

Arduino terdapat 5 output yang berupa lampu LED dan 2 input untuk sensor. Pada lampu LED telah ditempatkan pada port yang terdapat pada arduino, yaitu pada port 13, 12, 11, 10, 9 dan sensor ditempatkan pada port ke 8 dan 7.

Implementasi pada Arduino dapat dijelaskan pada gambar di bawah ini, pada gambar tersebut bahwa lampu 2 dan 3 menyala. Berarti port 10 dan 11 yang terdapat pada arduino bernilai 1, dan port yang lain bernilai 0, hal itu dapat dilihat pada lampu yang lain mati.

Di bawah ini adalah potongan coding pada arduino yang menggunakan bahasa pemrograman C, menjelaskan mengenai proses eksekusi data berdasarkan data yang masuk melalui serial port (USB). Pada coding tersebut jelaskan juga bagaimana proses saat sensor mendeteksi gerakan dan mengirim data melalui port USB ke Android



Gambar 4.6 Android Mengontrol Arduino

### Impelementasi Pengiriman data dari Arduino ke Android

Implementasi pengiriman data arduino ke android ini mengenai informasi sensor 1 ditempatkan pada port 8 yang dideklarasikan dengan BUTTON1 dan sensor 2 ditempatkan pada port 7 yang dideklarasikan dengan BUTTON2, Pengiriman data ini menggunakan Array data terdiri dari 2 indek Array, yaitu indek [0] mewakili nama port dan indek [1] mewakili kondisi atau status dari port.

### Proses Data Fungsi pada Arduino

Penjelasan proses data fungsi ini yang terdapat pada arduino yaitu data yang telah dikirim dari Android berupa masukan kemudian diolah pada arduino yang akan menghasilkan keluaran berupa menghidupkan dan mematikan lampu.

### Proses Android menerima data dari arduino

Pada proses ini menjelaskan tentang androis memproses data dari arduino, data dari arduino yang diterima oleh android berupa array data dengan 2 indek array, kemudian data indek [0] berfungsi sebagai keberadaan sensor yaitu sensor 1 atau sensor 2, dan data indek [1] berfungsi sebagai kondisi sensor zsaqwdyaitu sensor aktif atau nonaktif.

## UJI COBA DAN EVALUASI

Pada bab ini membahas tentang uji coba dan evaluasi proses kelancaran kinerja program untuk mencapai keberhasilan sesuai yang diinginkan agar program berjalan optimal dan *implementatif*.

### Uji Coba Sistem

Pada bagian ini akan dijelaskan mengenai proses uji coba dari *aplikasi* yang telah dibuat berdasarkan dari *desain* sistem yang telah dijelaskan sebelumnya. Uji coba ini dilakukan untuk melihat bahwa *aplikasi* yang telah dibuat sesuai dengan yang diharapkan, mulai dari awal hingga akhir proses dan tidak terjadi error pada aplikasi saat digunakan.

### Uji Coba Setting Alarm

Proses setting alarm merupakan bagian yang penting pada aplikasi ini. Alarm berfungsi untuk memberitahu pengguna jika ada gerakan yang terdeteksi oleh sensor gerak dengan membunyikan suara pada android. Proses alarm harus dipastikan berjalan lancar, agar tidak terjadi gerakan yang tidak diketahui oleh pengguna.

Pada uji coba kali ini, dilakukan setting untuk menyalakan alarm. Jika alarm menyala warna pada tombol berwarna hijau dengan tulisan “On” pada bagian atas tombol. Hasil dari uji coba menyalakan alarm dapat dilihat pada gambar di bawah ini.



Gambar 5.1 Uji coba menyalakan alarm

Posisi alarm saat ini telah menyala dengan tanda warna hijau pada tombol. Kemudian dilakukan mematikan alarm dengan menekan tombol alarm kembali. Hasil uji coba mematikan alarm dapat dilihat pada gambar di bawah ini.



Gambar 5.2 Hasil Uji coba mematikan alarm.

Gambar di atas menunjukkan bahwa alarm berhasil dimatikan dan bunyi alarm tidak lagi aktif.

### Uji Coba Mati Hidup Lampu dan Sensor

Proses ini adalah proses utama pada aplikasi. Uji coba pertama adalah melakukan deteksi gerak dengan sensor gerak. Sensor gerak didiamkan tanpa ada gerakan sama sekali. Hasilnya dapat dilihat pada gambar di bawah ini.



Gambar 5.3 Hasil Uji coba tidak ada gerakan pada sensor.

Gambar di atas menunjukkan keterangan “tidak ada gerakan terdeteksi” dan sensor berwarna hijau menandakan tidak ada gerakan yang terdeteksi oleh sensor. Uji coba selanjutnya adalah melakukan gerakan di depan sensor gerak. Hasil yang didapat seperti gambar di bawah ini.



Gambar 5.4 Hasil Uji coba Mendeteksi Gerakan di halaman depan

Gambar di atas menunjukkan bahwa sensor mendeteksi gerakan pada halaman depan rumah dengan keterangan tulisan “Gerakan Terdeteksi” dan warna sensor pada denah halaman depan berubah menjadi merah. Pada pendeteksian sensor pada halaman belakang, dapat dilihat pada gambar di bawah ini.



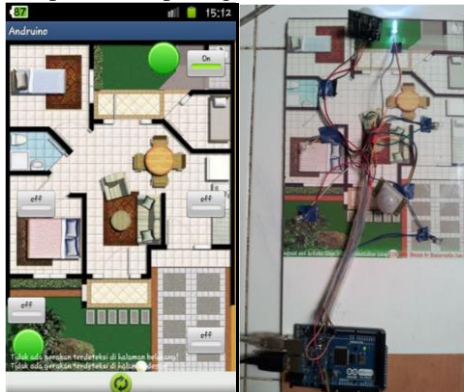
Gambar 5.5 Hasil Uji coba mendeteksi gerakan di halaman belakang

Gambar di atas menunjukkan bahwa sensor mendeteksi gerakan pada halaman belakang rumah dengan keterangan tulisan “Gerakan Terdeteksi” dan warna sensor pada denah berubah menjadi merah. Setiap kejadian pada sensor gerak tercatat oleh system dan disimpan ke dalam database. Data log yang tercatat dapat dilihat pada menu log kejadian. Hasil dari log kejadian jika sensor mendeteksi suatu gerakan halaman belakang atau halaman depan, dan log juga menampilkan kondisi lampu. Hasil dari Log Kejadian dapat dilihat pada gambar di bawah ini.



Gambar 5.6 Hasil Uji coba pencatatan log kejadian

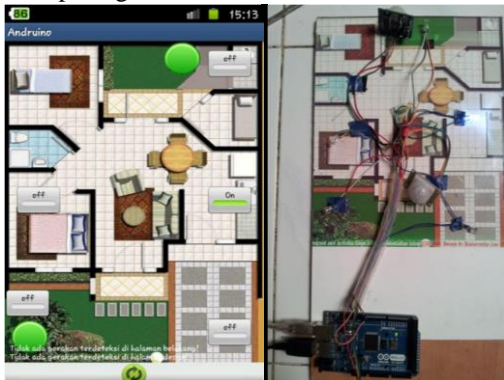
Uji coba selanjutnya adalah menyalakan lampu di rumah dengan proses uji coba lampu dinyalakan dengan menggunakan android. Hasil uji coba dapat dilihat pada gambar di bawah ini.



Gambar 5.7 Hasil Uji coba lampu Halaman Belakang

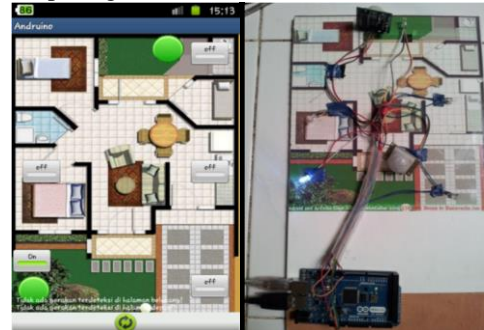
Gambar di atas menunjukkan bahwa pada *interface* android yang terdapat tombol di halaman belakang dirubah menjadi On maka lampu pada denah halaman belakang menyala.

Uji coba berikutnya adalah menyalakan lampu pada denah ruang tengah, yang awalnya tombol Off dirubah menjadi On dan hasilnya lampu pada ruang tengah menyala. Hasil uji coba dapat dilihat pada gambar di bawah ini.



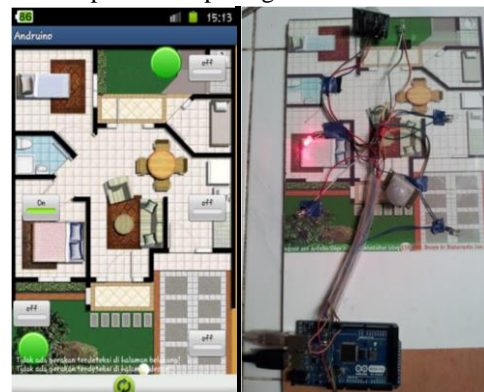
Gambar 5.8 Hasil Uji coba lampu Ruang Tengah

Uji coba berikutnya adalah menyalakan lampu pada denah ruang Garasi, yang awalnya tombol Off dirubah menjadi On dan hasilnya lampu pada ruang garasi menyala. Hasil uji coba dapat dilihat pada gambar di bawah ini.



Gambar 5.10 Hasil Uji coba lampu Halaman Depan

Gambar di atas menunjukkan bahwa lampu pada halaman depan menyala. Pada uji coba yang terakhir ini yang menunjukkan hasil uji coba dalam menyalakan lampu pada Kamar Tidur, yang awalnya tombol Off dirubah menjadi On dan hasilnya lampu pada Kamar Tidur menyala. Hasil uji coba dapat dilihat pada gambar di bawah ini.



Gambar 5.11 Hasil Uji coba lampu Kamar Tidur

Pada gambar di atas menunjukkan bahwa lampu pada kamar tidur menyala, pada hasil uji coba di atas dimulai dari *interface* pendeteksian sensor dan meyalakan atau mematikan lampu telah berhasil berjalan dengan baik dan hal ini menunjukkan system berhasil berkomunikasi antara Android dengan arduino.

## PENUTUP

Pada bab terakhir ini, beberapa kesimpulan dan saran yang bisa diberikan berdasarkan penerapan sistem Komunikasi antar Android dengan Arduino.



## Kesimpulan

Berdasarkan pada hasil penerapan dan uji coba aplikasi android ke arduino pada bab sebelumnya, dapat ditarik kesimpulan sebagai berikut :

1. Android dan arduino dapat berkomunikasi secara 2 arah dengan menggunakan ADB melalui USB
2. Aplikasi *interface* pada android dapat menjadi control pada arduino, agar lebih mudah dalam pengaturan arduino.
3. Aplikasi *interface* android dapat mematikan dan menyalakan lampu yang berada di arduino
4. Aplikasi *interface* android dapat digunakan pada sistem operasi android minimal Android versi 1.5 (*Cupcake*) sampai versi android akhir yaitu 4.0 (*Ice Cream*), tanpa memprogram ulang aplikasi.

## Saran

1. Disarankan untuk komunikasi USB dalam jarak lebih jauh akan sulit tercapai, oleh karena itu perlu ditambahkan sebuah converter USB to RS232, karena RS232 bisa mencapai jarak ratusan meter
2. Disarankan untuk penggunaan komunikasi android dengan arduino lebih dari satu, maka diperlukan sebuah converter USB to RS486, karena RS486 bisa untuk mencabang beberapa Arduino.

## DAFTAR PUSTAKA

- [1] Akbarul, Arif. 2012. *24 jam Pintar Pemrograman Android*. Andi. Yogyakarta
- [2] Irawan Budhi. 2007. *Linux Kernel Documentation*. [www.kernel.org](http://www.kernel.org) Diakses tanggal 12 Juni 2012
- [3] Mark L. Murphy. 2010. *Android TM Programming Tutorial* [www.github.com/commonsguy](http://www.github.com/commonsguy) Diakses tanggal 12 Juni 2012
- [4] Putra, Eko. 2009. *Belajar Mikrokontroler AT89c51/52/55 Teori dan Aplikasi*. Gaya Media. Reto Meier. 2012. *Profesional Android Application Development*. [www.wrox.com](http://www.wrox.com) Diakses tanggal 05 Juni 2012
- [5] Raharjo, Budi. 2010. *Pemrograman C++*. Informatika. Bandung
- [6] Raharjo, Budi. 2006. *Pemrograman C dan mplementasinya*. Informatika. Bandung
- [7] Afaat H, Nazruddin. 2011. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Informatika. Bandung
- [8] Setiawan, Sulhan. 2009. *Mudah dan Menyenangkan Belajar Mikrokontroler*. Andi Publisher. Yogyakarta
- [9] Safaat, Nazrudin. 2011. *Android: Pemrograman Aplikasi Mobile Smartphone dan Tablet PC*. Informatika. Bandung
- [10] Winoto, Ardi. 2008. *Mikrokontroler AVR ATMEGA8/32/16/8535 dan Pemrograman Dengan Bahasa C*. Informatika. Bandung

{Halaman sengaja dikosongkan}  
{Jurnal **Melek IT**}

(4)  
**ANALISIS KOMUNIKASI DATA PADA KOMPUTER IPv4 DAN IPv6 MENGGUNAKAN  
DUAL STACK TRANSITION MECHANISM (DSTM)**

**Adi Bagus Santoso<sup>1</sup>, F.X. Wisnu Yudo Untoro<sup>2</sup>**

Program Studi Teknik Informatika, Fakultas Teknik, Univesitas Wijaya Kusuma Surabaya

Jl. Dukuh Kupang XXV/ 54 Surabaya 60225

Email: [b\\_guez44@yahoo.com](mailto:b_guez44@yahoo.com)

**ABSTRAK**

Perkembangan teknologi jaringan komputer saat ini semakin berkembang pesat seiring semakin meningkatnya pengguna yang memanfaatkan jaringan komputer. Dalam suatu jaringan komputer terdapat sistem pengalamatan yang dinamai dengan *IP address*.

*IP address* yang umum digunakan saat ini adalah IPv4. Pertambahan *user* yang semakin banyak menyebabkan habisnya IPv4 yang tersedia di internet. Untuk mengatasi permasalahan tersebut dikembangkan jenis IPv6. Karena sistem ini masih tergolong baru maka apabila ada *user* pengguna IPv4 bisa berdampingan dengan IPv6, diperlukan suatu sistem *Dual Stack Transition Mechanism* (DSTM) untuk mengkoneksikan kedua IP tersebut dalam sebuah jaringan komputer.

Pembuatan aplikasi ini menggunakan *Dual Stack Transition Mechanism* (DSTM) yang nantinya akan diuji melalui parameter *throughput* dan *delay* guna mengetahui gambaran antara IPv4 dan IPv6 dimasa depan.

**Kata Kunci :** Dual Stack Transition Mechanism (DSTM), IPv4, IPv6

**PENDAHULUAN**

**Latar Belakang**

Perkembangan teknologi saat ini sudah sangat modern, terutama perkembangan teknologi jaringan komputer semakin pesat seiring dengan kebutuhan *user* yang memanfaatkan layanan ini. Pada sistem jaringan komputer terdapat IP (*Internet Protocol*). IP yang dikenal secara umum saat ini adalah IP versi 4 (IPv4) yang ketersediaannya semakin berkurang, sehingga dikembangkan IP versi 6 (IPv6).

IP versi 6 (IPv6) merupakan *protocol* internet baru yang dikembangkan pada tahun 1994 oleh *Internet Engineering Task Force* (IETF) untuk menggantikan IP versi 4 (IPv4) yang saat ini tengah mendekati ambang batas alokasi alamatnya. Ruang alamat IPv4 ini diperkirakan akan habis pada tahun 2011. Tujuan utama dikembangkan IPv6 adalah untuk meningkatkan ruang alamat internet sehingga mampu mengakomodasi perkembangan internet yang semakin pesat.

Sejak pelepasan IPv4 dirancang untuk mendukung  $2^{32}$  *IP address*. Di lain pihak IPv6, yang mampu mendukung  $2^{128}$  *IP address* dapat menampung  $2^{96}$  kali jumlah alamat yang dapat disediakan oleh IPv4. Tetapi, itu bukanlah satu-satunya keuntungan dari IPv6. IPv6 juga memiliki banyak kelebihan lain, *addressing* dan

*routing* yang lebih baik, *header* yang lebih sederhana, dukungan untuk kualitas dan *classes of service* ditingkat paket, keamanan yang lebih baik dalam hal autentifikasi, integritas pesan dan privasi, berkurangnya jumlah administrasi karena auto konfigurasi yang lebih baik dan peningkatan dukungan terhadap mobilitas.

Karena sistem ini masih tergolong baru maka apabila ada *user* pengguna IPv4 bisa berdampingan dengan IPv6, oleh karena itu menjadi permasalahan bagaimana jaringan IPv4 mampu berinteraksi dengan jaringan IPv6. Salah satu cara yaitu dengan dilakukan migrasi IPv4 ke IPv6 atau bisa menggunakan mekanisme lain. Walaupun terlihat sederhana, migrasi ini tidak mudah dilakukan, karena bila hal ini dilakukan maka semua bagian dari jaringan, termasuk diantaranya *firewall*, *server*, dan *workstation* yang digunakan juga harus di-*up grade*. Dalam mekanisme transisi yang memungkinkan keduanya bisa berhubungan. Mekanisme itu misalnya dengan menggunakan *dual protocol stack*, translasi, *tunneling*, atau menggantinya menjadi IPv6 sepenuhnya.

Dalam tugas akhir ini penulis menggunakan teknik *DualStack Transition Mechanism* (DSTM) merupakan solusi yang ditujukan untuk jaringan untuk dilakukan teknik tersebut maka akan di analisa dari jaringan IPv4 dan IPv6 untuk

mengetahui perbandingan dari kedua jaringan tersebut dengan parameter *throughput* dan *delay*.

## TINJAUAN PUSTAKA

### Internet Protocol version 4 (IPv4)

IPv4 adalah sebuah jenis pengalamatan jaringan yang digunakan di dalam protokol jaringan TCP/IP yang menggunakan protokol IP versi 4. Panjang totalnya adalah 32-bit, dan secara reoritis dapat mengalami hingga 232 *host* komputer di dunia. Alamat IPv4 umumnya diekspresikan dalam notasi desimal bertitik (*dotted-decimal notation*), yang dibagi kedalam empat buah oktet berukuran 8-bit sehingga nilainya berkisar antara 0 hingga 255.

### IPv4 Addressing

Alamat IP yang dimiliki oleh sebuah *host* dapat dibagi dengan menggunakan *subnet mask* jaringan kedalam dua buah bagian, yakni: 1. *Network Identifier/NetID* atau *network address* (alamat jaringan) yang digunakan khusus untuk mengidentifikasi alamat jaringan di mana *host* berada. Dalam banyak kasus, sebuah alamat *network identifier* adalah sama dengan segmen jaringan fisik dengan batasan yang dibuat dan didefinisikan oleh *router* IP. Meskipun demikian, ada beberapa kasus di mana beberapa jaringan logis terdapat di dalam sebuah segmen jaringan fisik yang sama dengan menggunakan sebuah praktek yang disebut sebagai *multinetting*. Semua sistem di dalam sebuah jaringan fisik yang sama harus memiliki alamat *network identifier* yang sama. *Network identifier* juga harus bersifat unik dalam sebuah *internetwork*. Jika semua node di dalam jaringan logis yang sama tidak dikonfigurasi dengan menggunakan *network identifier* yang sama, maka terjadilah masalah yang disebut dengan *routing error*. Alamat *network identifier* tidak boleh bernilai 0 atau 255. 2. *Host Identifier/Host ID* atau *Host address* (alamat *host*) yang digunakan khusus untuk mengidentifikasi alamat *host* (dapat berupa workstation, server atau sistem lainnya yang berbasis teknologi TCP/IP) di dalam jaringan. Nilai *host identifier* tidak boleh bernilai 0 atau 255 dan harus bersifat unik di dalam *network identifier*/segmen jaringan di mana ia berada.

Alamat IPv4 terbagi menjadi beberapa jenis, yakni sebagai berikut:

1. *Alamat Unicast*, merupakan alamat IPv4 yang ditentukan untuk sebuah antarmuka jaringan yang dihubungkan ke sebuah *internetwork* IP. Alamat *unicast* digunakan dalam komunikasi *point-to-point* atau *one-to-one*.

2. *Alamat Broadcast*, merupakan alamat IPv4 yang didesain agar diproses oleh setiap *node* IP dalam segmen jaringan yang sama. Alamat *broadcast* digunakan dalam komunikasi *one-to-everyone*.
3. *Alamat Multicast*, merupakan alamat IPv4 yang didesain agar diproses oleh satu atau beberapa *node* dalam segmen jaringan yang sama atau berbeda. Alamat *multicast* digunakan dalam komunikasi *one-to-many*.

Alamat IP versi 4 dibagi ke dalam beberapa kelas, dilihat dari oktet pertamanya, seperti terlihat pada tabel 2.1. Sebenarnya yang menjadi pembeda kelas IP versi 4 adalah pola biner yang terdapat dalam oktet pertama

(utamanya adalah *bit-bit awal/high-order bit*), tapi untuk lebih mudah mengingatnya, akan lebih cepat diingat dengan menggunakan representasi desimal.

Kelas Alamat IP	Oktet Pertama (desimal)	Oktet Pertama (biner)	Digunakan Oleh
Kelas A	1 – 126	0xxx xxxx	Alamat <i>unicast</i> untuk jaringan skala besar
Kelas B	128 – 191	1xxx xxxx	Alamat <i>unicast</i> untuk jaringan skala menengah hingga skala besar
Kelas C	192 – 223	110x xxxx	Alamat <i>unicast</i> untuk jaringan skala kecil
Kelas D	224 – 239	1110 xxxx	Alamat <i>multicast</i> (bukan alamat <i>unicast</i> )
Kelas E	240 – 255	1111 xxxx	Direservasikan, umumnya digunakan sebagai alamat percobaan (eksperimen); (bukan alamat <i>unicast</i> )

Tabel 2.1 Pembagian Kelas Dalam IPv4

#### 1. Kelas A

Alamat-alamat kelas A diberikan untuk jaringan skala besar. Nomor urut *bit* tertinggi di dalam alamat IP kelas A selalu diset dengan nilai 0 (nol). Tujuh *bit* berikutnya untuk melengkapi oktet pertama akan membuat sebuah *network identifier*. 24 *bits* sisanya (atau tiga oktet terakhir) merepresentasikan *host identifier*. Ini mengizinkan kelas A memiliki hingga 126 jaringan, dan 16,777,214 *host* tiap jaringannya. Alamat dengan oktet awal 127 tidak diizinkan, karena digunakan untuk mekanisme *Inter process Communication* (IPC) di dalam mesin yang bersangkutan.

#### 2. Kelas B

Alamat-alamat kelas B dikhususkan untuk jaringan skala menengah hingga skala besar. Dua *bit* pertama di dalam oktet pertama alamat IP kelas B selalu diset ke bilangan biner 10. 14 *bit* berikutnya (untuk melengkapi dua oktet pertama), akan



membuat sebuah *network identifier*. 16 bits-nya (dua oktet terakhir) merepresentasikan *host identifier*. Kelas B dapat memiliki hingga 16,384 jaringan, dan 65,534 *host* untuk setiap jaringannya.

### 3. Kelas C

Alamat IP kelas C digunakan untuk jaringan berskala kecil. Tiga *bit* pertama di dalam oktet pertama alamat kelas C selalu diset ke nilai biner 110. 21 *bit* selanjutnya (untuk melengkapi tiga oktet pertama) akan membentuk sebuah *network identifier*. 8 bits-nya (sebagai oktet terakhir) akan merepresentasikan *host identifier*. Ini memungkinkan pembuatan total 2,097,152 buah jaringan, dan 254 *host* untuk setiap jaringannya.

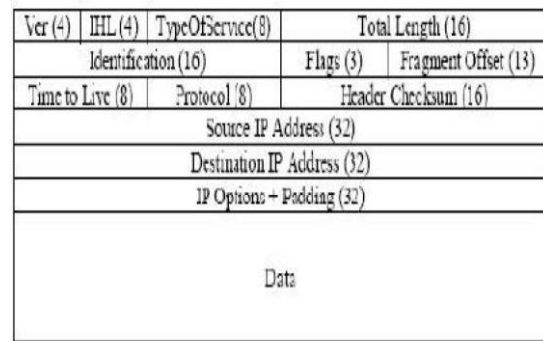
### 4. Kelas D

Alamat IP kelas D disediakan hanya untuk alamat-alamat *IP multicast*, sehingga berbeda dengan tiga kelas di atas. Empat *bit* pertama di dalam IP kelas D selalu diset ke bilangan biner 1110. 28 bits-nya digunakan sebagai alamat yang dapat digunakan untuk mengenali *host*.

### 5. Kelas E

Alamat IP kelas E disediakan sebagai alamat yang bersifat eksperimental atau percobaan dan dicadangkan untuk digunakan pada masa depan. Empat *bit* pertama selalu diset kepada bilangan biner 1111. 28 sisanya digunakan sebagai alamat yang dapat mengenali *host*.

**Struktur Header Paket IPv4** paket-paket data dalam protokol IPv4 dikirimkan dalam bentuk datagram. Sebuah paket IPv4 terdiri atas *header IP* dan muatan IP (*payload*). *Header IP* menyediakan dukungan untuk memetakan jaringan (*routing*), identifikasi muatan IP, ukuran *header IP* dan paket IP, dukungan fragmentasi, dan juga *IP options*. Sedangkan *payload IP* berisi informasi yang dikirimkan. *Payload IP* memiliki ukuran bervariasi, berkisar dari 8 *byte* hingga 65515 *byte*. Sebelum dikirimkan di dalam saluran jaringan, paket IP akan dibungkus (*encapsulation*) dengan *header protocol* lapisan antarmuka jaringan dan *trailer*-nya, untuk membuat sebuah frame jaringan. Setiap paket terdiri dari beberapa field yang memiliki fungsi tersendiri dan memiliki informasi yang berbeda-beda. Pada gambar 2.10 dapat dilihat struktur dari paket IPv4.



Gambar 2.2 Struktur Paket IPv4

### Internet Protocol version 6 (IPv6)

Sistem pengalamatan IPv6 disebut juga dengan IPng (*Internet Protocol, next generation*) karena merupakan generasi terbaru pengganti IPv4 sebagai standar IP. IPv6 menggunakan sistem pengalamatan 128 *bits*, 4 kali lebih besar daripada IPv4 yang artinya mampu menghasilkan alokasi alamat sebesar  $2^{(128-32)}$  kali lebih besar daripada IPv4. Sistem pengalamatan ini dipetakan secara heksa (16 *bits*) untuk mempermudah pembacaannya. Setiap 16 *bits* tersebut ditampilkan dalam bentuk *section* secara heksadesimal 4 digit dengan dipisahkan oleh tanda titik dua. Walaupun ditampilkan secara heksadesimal, IPv6 dirasa terlalu rumit untuk diingat karena panjangnya mencapai 32 digit angka. Selain itu, IPv6 seringkali terdiri dari banyak angka nol sehingga dianggap kurang efisien. Pada kasus tersebut, IPv6 memiliki kelonggaran untuk memperpendek alamatnya dengan ketentuan sebagai berikut:

1. Angka nol yang mengawali setiap *section* dapat dihilangkan.
2. *Section* minimal memiliki satu digit angka.
3. *Section* yang berurutan dan hanya terdiri dari angka nol dapat diganti dengan tanda titik dua yang ditulis rangkap. Ketentuan ini hanya berlaku satu kali penulisan. Sebagai contoh penulisannya, dapat dilihat pada Gambar ini :



Gambar 2.3 Contoh penulisan alamat IPv6

Pada dasarnya, IPv6 terdiri dari 2 bagian utama yaitu prefiks yang menunjukkan tipe

pengalamatan dan sisanya mengikuti *system* yang digunakan prefiks tersebut. *Provider based unicast address* merupakan tipe prefix yang umum digunakan sebagai pengalamatan *unicast* pada *host* yang spesifik. Pengalamatan *unicast* memungkinkan suatu *host* berkomunikasi dengan satu *host* yang lain. *Provider-based unicast address* menggunakan prefiks 3 *bits* berupa “010” dengan diikuti sistem pengalamatannya sebagai berikut (Forouzan 2003):

1. *Registry identifier*, 5 *bits* penunjuk agensi pusat IPv6 yang telah mengalokasikan alamatnya. Sebagai contoh, untuk kawasan Asia-Pasifik dengan agensi pusat APNIC menggunakan kode 10100.
2. *Provider identifier*, menunjukkan ISP (*Internet Service Provider*) yang digunakan. Umumnya menggunakan 16 *bits*.
3. *Subscriber identifier*, menunjukkan kode berlangganan terhadap ISP tertentu. Umumnya menggunakan 24 *bits*.
4. *Subnet identifier*, menunjukkan *subnet* (sub jaringan) spesifik yang berada dibawah manajemen pengguna. Umumnya menggunakan 32 *bits*.
5. *Node identifier*, menunjukkan alamat spesifik suatu *host* di bawah *subnet* tertentu. Umumnya menggunakan 48 *bits*.

#### Format Prefix

Dalam IPv4, sebuah alamat dalam notasi dotted-decimal format dapat direpresentasikan dengan menggunakan angka prefiks yang merujuk kepada subnet mask. IPv6 juga memiliki angka prefiks, tapi tidak digunakan untuk merujuk kepada subnet mask, karena memang IPv6 tidak mendukung subnet mask.

Prefiks adalah sebuah bagian dari alamat IP, di mana *bit-bit* memiliki nilai-nilai yang tetap atau *bit-bit* tersebut merupakan bagian dari sebuah rute atau subnet identifier. Prefiks dalam IPv6 direpresentasikan dengan cara yang sama seperti halnya prefiks alamat IPv4, yaitu [alamat] / [angka panjang prefiks]. Panjang prefiks menentukan jumlah *bit* terbesar paling kiri yang membuat prefiks *subnet*. Sebagai contoh, prefiks sebuah alamat IPv6 dapat direpresentasikan yaitu : 3ffe:2900:d005:f28b::/64 Pada contoh tersebut, 64 *bit* pertama dari alamat tersebut dianggap sebagai prefiks alamat, sementara 64 *bit* sisanya dianggap sebagai interface ID.

#### Struktur Header IPv6

*Datagram* IPv6 terbagi menjadi dua bagian utama yaitu *header* dan *payload*. *Header* IPv6 memiliki ukuran yang tetap yakni 40 *bytes*. Akan tetapi, terdapat *header* tambahan (*extension*) untuk meningkatkan fungsionalitasnya di bagian *payload*. Dengan demikian, *payload* berisikan data paket beserta *header* tambahan tersebut.

VER 4 bits	PRI 4 bits	Flow Label 24 bits	
Payload Length 16 bits		Next Header 8 bits	Hop Limit 8 bits
Source Address 128 bits			
Destination Address 128 bits			
Payload (Extension headers + Data packet)			

Gambar 2.4 Struktur header IPv6.

#### Perbedaan IPv4 Dan IPv6

Perbedaan antara IPv4 dan IPv6 sebagai berikut.

##### IPv4

1. Kapasitas alamat 32 *bit*
2. IPv4 tidak mempunyai kemampuan auto renumbering
3. Bisa melakukan NAT (*Network Address Translation*).
4. Tidak semua produk dari IPv4 bisa mengimplementasikan IPSEC.
5. Alamat IPv4 terbagi menjadi 2 jenis yaitu *unicast address*, dan *multicast address*.
6. IP header IPv4 terdiri dari *version*, *IHL*, *Type of service*, *Total Length*, *Identification*, *Flags*, *Fragment Offset*, *Time to Live*, *Protocol*, *Header checksum*, *Source address*, *Destination address*, *Option*, *Padding*.
7. Alamat sumber (*source address*) dan alamat tujuan (*destination address*) sebesar 32 *bit*.
8. Fragmentasi dikerjakan oleh both *router* dan *host* pengirim.
9. Checksum termasuk *Header* pada IPv4
10. Options termasuk *header* pada IPv4.
11. Menggunakan *host address* (A) resource record didalam *Domain Name System* (DNS) untuk memetakan nama *host* pada pengalamatan IPv4.

##### IPv6

1. Kapasitas perluasan alamat 128 *bit*.
2. IPv6 mempunyai kemampuan autorenumbering (penomoran kembali alamat IP secara otomatis).
3. Tidak bisa melakukan NAT.

4. Setiap produk yang menggunakan IPv6 bisa mengimplementasikan IPSEC sehingga tidak perlu di upgrade. IPSEC adalah fitur yang dimiliki oleh IPv6 namun oleh beberapa *developer* diaplikasikan ke dalam IPv4. (Lihat IPSEC).
5. Alamat IPv4 terbagi menjadi 3 yaitu *unicastaddress*, *multicastaddress*, dan *anycast address*.
6. IP header IPv6 terdiri dari *Version*, *Traffic class*, *Flow Label*, *Payload length*, *Next Header*, *Hop Limit*, *Source address*, *Destinationaddress*.
7. Alamat sumber (*source address*) dan alamat tujuan (*destination address*) sebesar 128 bit.
8. Fragmentasi tidak dikerjakan oleh *router*, hanya oleh *host* pengirim.
9. Checksum tidak termasuk *header* pada IPv6.
10. Semua optional data diusulkan untuk ekstensi *header* IPv6.
11. Menggunakan *hostaddress* (AAAA) resource record didalam Domain Name system (DNS) untuk memetakan nama *host* pada pengalamatan IPv6. AAAA adalah tipe *record* baru untuk menyimpan sebuah alamat IPv6 dengan nilai tipe 28.

### Mekanisme Transisi

Mekanisme transisi secara umum didefinisikan sebagai sekumpulan teknik yang berupaya agar node IPv6 dapat saling berkomunikasi dengan node IPv4 yang sudah ada sebelumnya. Mekanisme ini terbagi menjadi empat kategori berdasarkan teknik yang digunakan, yaitu mekanisme *hybrid* (*dual* IPv4/IPv6), *aplication-layer gateways*, penerjemahan protokol, dan *tunneling*. Masingmasing kategori tersebut memiliki cara kerja dan tujuan yang berbeda-beda. *Tunneling* sangat diandalkan sebagai mekanisme transisi pada saat IPv6 mulai dikembangkan.

Teknik yang digunakan yakni menghubungkan IPv4 dan IPv6 dengan cara enkapsulasi-dekapsulasi paket. Secara umum *tunneling* berupa IPv6-over-IPv4 yaitu membungkus paket IPv6 ke dalam paket IPv4 untuk kemudian dibuka kembali. Mekanisme ini sangat sesuai dalam kondisi jaringan yang didominasi IPv4 dan keberadaan node IPv6 yang menyebar tidak beraturan untuk saling berkomunikasi. Akan tetapi mekanisme ini kurang sesuai jika suatu jaringan didominasi IPv6. DSTM sebagai salah satu mekanisme *tunneling* terbaru menggunakan sistem yang berkebalikan yaitu IPv4-over-IPv6. jalur

husus paket IPv4. Paket IPv4 yang akan dikirim oleh DSTM *client* akan dibungkus dalam paket IPv6

### DualStack Transition Mechanism (DSTM)

*DualStack Transition Mechanism* (DSTM) merupakan salah satu mekanisme transisi *tunneling* (IPv4-over-IPv6) dengan membungkus paket IPv4 ke dalam bentuk paket IPv6 di sisi *host* IPv6 untuk kemudian dibuka kembali di batas akhir IPv6 ke IPv4 dan dikirim menuju *host* dalam jaringan IPv4 (Bound 2002). Begitu pula sebaliknya untuk arah yang berlawanan. DSTM *client* (IPv6) dapat berkomunikasi dengan *host* IPv4 dengan cara meminta alamat IPv4 terlebih dahulu ke *server* DSTM. *Server* DSTM memberikan IPv4 secara dinamis kepada DSTM *client* yang kemudian dipetakan (*address mapping*) dengan alamat IPv6-nya dalam *cache* (penyimpanan sementara). Setelah mendapatkan IPv4 tersebut, DSTM *client* akan membangun *Dynamic Tunnel Interface* (DTI) sebagai menuju DSTM *Tunnel End Point* (DSTM TEP) untuk dibuka kembali dan dikirim sesuai tujuannya di jaringan IPv4. DSTM juga memungkinkan komunikasi sebaliknya antara *host* dalam jaringan IPv4 dengan jaringan IPv6. Dengan membaca *address mapping* dalam *cache* yang telah disediakan secara temporal sebelumnya, komunikasi data dapat segera berjalan. Namun jika *address mapping* tidak ada atau sudah hilang, maka perlu berhubungan dengan *Domain Name System* (DNS) untuk mengarahkan alamat sebenarnya dalam jaringan IPv6.

### Delay

*Delay* merupakan lamanya waktu yang dibutuhkan oleh data informasi untuk sampai ke tempat tujuan data informasi tersebut dikirim. *Delay* pada suatu jaringan akan menentukan langkah apa yang akan kita ambil ketika kita memanajemen suatu jaringan. Ketika *delay* besar, dapat diketahui jaringan tersebut sedang sibuk atau kemungkinan yang lain adalah kapasitas jaringan tersebut yang kecil sehingga bisa melakukan tindakan pencegahan agar tidak terjadi *overload*. Misalkan dengan memindahkan sebagian aliran data ke jalur lain atau memperbesar kapasitas jaringan kita, berikut tabel parameter kategori *delay*

Nilai Delay	Kategori
< 150 ms	Sangat Bagus
150 ms – 300 ms	Bagus
300 ms – 450 ms	Sedang
> 450 ms	Jelek

Tabel 2.2 Parameter Kategori *Delay*

### **Throughput**

*Throughput* adalah ukuran dari *transferbit* di media selama jangka waktu tertentu. Karena sejumlah faktor, *throughput* biasanya tidak sesuai dengan *bandwidth* yang ditentukan dalam implementasi lapisan fisik seperti Ethernet. Banyak faktor yang mempengaruhi *throughput*. Diantara faktor-faktor tersebut jumlah lalu lintas, jenis lalu lintas, dan jumlah perangkat jaringan ditemui pada jaringan yang diukur. Dalam topologi multi-access seperti Ethernet, node bersaing untuk akses media dan penggunaannya. Oleh karena itu, *throughput* masing node terdegradasi penggunaan media meningkat.

Dalam *internetwork* atau jaringan dengan beberapa segmen, *throughput* tidak bisa lebih cepat dari link paling lambat path dari sumber ke tujuan. Bahkan jika semua atau sebagian besar segmen memiliki *bandwidth* yang tinggi, itu hanya akan mengambil satu segmen dalam jalur dengan *throughput* yang rendah untuk menciptakan hambatan ke *throughput* seluruh jaringan

### **FileTransferProtocol**

Protokol pengiriman berkas atau biasa di sebut *file transfer protocol* adalah sebuah protocol internet yang berjalan di dalam lapisan aplikasi yang merupakan standar untuk pengiriman berkas (*file*). komputer antarmesinmesin dalam sebuah antar jaringan. FTP merupakan salah satu protokol Internet yang paling awal dikembangkan, dan masih digunakan hingga saat ini untuk melakukan pengunduhan (*download*) dan pengunggahan (*upload*) berkasberkas komputer antara klien FTP dan *server* FTP. Sebuah Klien FTP merupakan aplikasi yang dapat mengeluarkan perintah-perintah FTP ke sebuah *server* FTP, sementara *server* FTP adalah sebuah *Windows Service* atau *daemon* yang berjalan di atas sebuah komputer yang merespons perintah-perintah dari sebuah klien FTP. Perintahperintah FTP dapat digunakan untuk mengubah direktori, mengubah modus pengiriman antara biner dan ASCII, mengunggah berkas komputer ke *server* FTP, serta mengunduh berkas dari *server* FTP.

Sebuah *server* FTP diakses dengan menggunakan Universal Resource Identifier (URI) dengan menggunakan format `ftp://namaserver` Klien FTP dapat menghubungi *server* FTP dengan membuka URI tersebut. FTP menggunakan protokol Transmission Control Protocol (TCP) untuk komunikasi data antara klien dan *server*, sehingga di antara kedua komponen tersebut akan dibuatlah sebuah sesi komunikasi sebelum pengiriman data

dimulai. Sebelum membuat koneksi, port TCP nomor 21 di sisi *server* akan "mendengarkan" percobaan koneksi dari sebuah klien FTP dan kemudian akan digunakan sebagai port pengatur (*control port*) untuk membuat sebuah koneksi antara klien dan *server*, untuk mengizinkan klien untuk mengirimkan sebuah perintah FTP kepada *server* dan juga mengembalikan respons *server* ke perintah tersebut. Sekali koneksi kontrol telah dibuat, maka *server* akan mulai membuka port TCP nomor 20 untuk membentuk sebuah koneksi baru dengan klien untuk mengirim data aktual yang sedang dipertukarkan saat melakukan pengunduhan dan pengunggahan.

FTP hanya menggunakan metode autentikasi standar, yakni menggunakan *username* dan *password* yang dikirim dalam bentuk tidak terenkripsi. Pengguna terdaftar dapat menggunakan *username* dan *password*-nya untuk mengakses, men-*download*, dan meng-*upload* berkas-berkas yang ia kehendaki. Umumnya, para pengguna terdaftar memiliki akses penuh terhadap beberapa direktori, sehingga mereka dapat membuat berkas, membuat direktori, dan bahkan menghapus berkas. Pengguna yang belum terdaftar dapat juga menggunakan metode anonymous login, yakni dengan menggunakan nama pengguna anonymous dan *password* yang diisi dengan menggunakan alamat e-mail atau kosong.

## **ANALISA DAN PERANCANGAN**

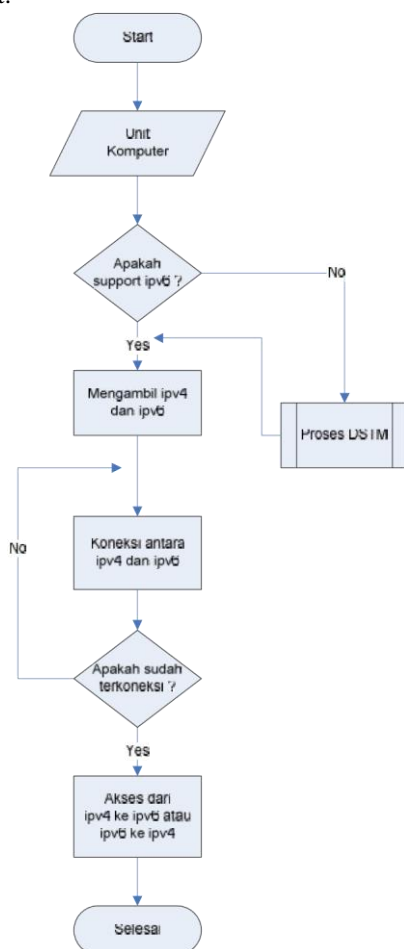
Perancangan sistem dibuat untuk memenuhi kebutuhan pengguna mengenai gambaran yang jelas bagaimana suatu sistem dibentuk. Perancangan sistem ini merupakan tahapan persiapan sistem sebelum diimplementasikan.

### **Perencanaan konsep sistem**

Pada konsep ini merupakan terdapat sistem jaringan menggunakan IPv4 dan ada juga *system* jaringan menggunakan IPv6 karena ada yang berbeda maka bagaimana agar kedua sistem ini bisa berkomunikasi data antara satu sistem dengan sistem lain. Karena sistem ini masih tergolong baru maka apabila ada *user* pengguna IPv4 bisa berdampingan dengan IPv6, oleh karena itu menjadi permasalahan bagaimana jaringan IPv4 mampu berkomunikasi data dengan jaringan IPv6. Dalam implementasinya maka dibutuhkan aplikasi tambahan yaitu dibutuhkan suatu mekanisme tersendiri.

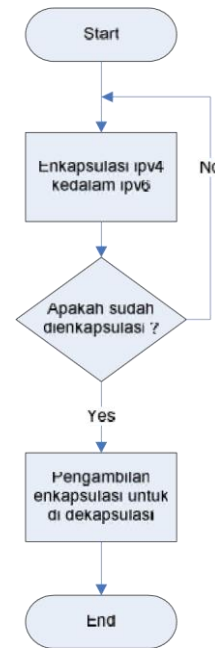
Mekanisme tersebut ada dengan cara DSTM (*DualStack Transition Mechanism*) merupakan suatu mekanisme pengambilan IPv4 untuk diubah

menjadi IPv6 yaitu dengan cara *enkapsulasi* atau pembungkusan IPv4 yang akan di ambil untuk dijadikan IPv6 setelah itu akan dilakukan *dekapsulasi* yaitu pelepasan IP apabila sudah menjadi IPv6 dan akan dilakukan koneksi antar kedua sistem yang berbeda tersebut yaitu IPv4 koneksi dengan IPv6. Apabila sudah bisa terkoneksi kedua sistem maka akan dilakukan uji coba dengan parameter *throughput* dan *delay* dengan parameter tersebut bisa mengetahui hasil perbandingan dari kedua sistem yang dicapai dan perbandingan dari kedua masing-masing sistem tersebut untuk apakah baik dan buruk dari kedua sistem. Untuk lebih jelasnya bisa melihat desain *flowchart* sistem berikut:



Gambar 3.1 *flowchart* sistem

Pada gambar 3.1 cara kerja sebagai berikut, terdapat komputer IPv4 yang akan melakukan koneksi dengan IPv6 dilanjutkan dengan melakukan apakah support IPv6 komputer tersebut, apabila belum maka akan dilakukan proses DSTM (*DualStack Transition Mechanism*) tapi apabila sudah support IPv6 langsung melakukan pengambilan IPv4 untuk dilakukan koneksi antara IPv4 ke IPv6 atau IPv6 ke IPv4.



Gambar 3.2 *flowchart* DSTM

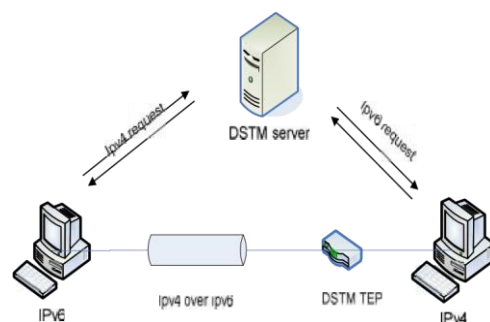
Pada gambar 3.2 cara kerja sebagai berikut, pertama melalui enkapsulasi IPv4 kedalam IPv6 jadi pengambilan suatu IP untuk diterjemahkan kedalam IPv6, apabila sudah di enkapsulasi maka diambil IP tersebut untuk dilakukan dekapsulasi IP yang sudah diterima untuk disampaikan dan selesai.

### Perencanaan kebutuhan sistem

Berisi perancangan (desain) dari perangkat keras maupun lunak yang akan digunakan dalam melakukan simulasi sistem, penentuan perangkat lunak dan topologi yang akan digunakan, sekaligus pengaturan perangkat keras tersebut agar sesuai dengan topologi yang diinginkan.

### Desain Jaringan

Desain jaringan komputer nirkabel yang akan dibuat ini merupakan desain yang ingin dicapai dalam pengerjaan Tugas Akhir berikut merupakan design jaringan yang telah dibuat :



Gambar 3.3 Desain Jaringan

Pada gambar 3.3 dapat di gambarkan sebagai berikut IPv6 dapat berkomunikasi dengan *host* IPv4



dengan cara meminta alamat IPv4 terlebih dahulu ke *serverDSTM (DualStack Transition Mechanism)*. *Server DSTM (DualStack Transition Mechanism)* memberikan IPv4 secara dinamis kepada IPv6 yang kemudian dipetakan (*address mapping*) dengan alamat IPv6-nya dalam *cache* (penyimpanan sementara). Setelah mendapatkan IPv4 tersebut, IPv6 akan mengirimkan jalur khusus paket IPv4. Paket IPv4 yang akan dikirim oleh IPv6 akan dibungkus dalam paket IPv6 menuju *DSTM Tunnel EndPoint (DSTM TEP)* untuk dibuka kembali dan dikirim sesuai tujuannya di jaringan IPv4.

#### **Hardware yang digunakan**

Mulai dari tahap analisa sampai dengan tahap implementasi dalam ini menggunakan perangkat *laptop* dengan spesifikasi sebagai berikut:

##### **Hardware:**

1. Laptop AMD Turion Dual Core 2.0 GHz
2. Harddisk 320 GB
3. RAM 2 GB
4. Keyboard

#### **Software yang digunakan**

Adapun untuk kebutuhan *software* mulai tahap analisa sampai implementasi ini menggunakan beberapa *software* berikut:

##### **Software :**

1. Sistem Operasi windows
2. Aplikasi *DualStack Transition Mechanism (DSTM)*
3. *Wireshark*
4. *Xlight FTP server*

#### **Perencanaan pengujian komunikasi data**

##### **Perencanaan**

Pada perencanaan dibutuhkan 4 uji coba komunikasi data yang berbeda-beda yaitu sebagai berikut :

1. Pengujian komunikasi data IPv4 ke IPv4.

Pada sistem ini dilakukan hanya dengan koneksi antara IPv4 dan IPv4, komputer 1 disetting IPv4 dan komputer 2 disetting IPv4 setelah kedua sistem ini terkoneksi maka digunakan parameter *throughput* dan *delay* untuk menguji dan akan dilihat hasil dari pengujian tersebut.

2. Pengujian komunikasi data IPv6 ke IPv6.

Pada sistem ini dilakukan hanya dengan koneksi antara IPv6 dan IPv6, komputer 1 disetting IPv6 dan komputer 2 disetting IPv6 setelah kedua sistem ini terkoneksi maka digunakan parameter *throughput* dan *delay* untuk menguji dan akan dilihat hasil dari pengujian tersebut.

3. Pengujian komunikasi data IPv4 ke IPv6.

Pada sistem ini dilakukan dengan koneksi antara IPv4 dan IPv6, komputer 1 disetting IPv4 tanpa menghilangkan IPv6 yang dia punya dan komputer 2 disetting IPv6 tanpa menghilangkan IPv4 yang dia punya maka setelah itu digunakan parameter *throughput* dan *delay* untuk menguji dan akan dilihat hasil dari pengujian tersebut.

4. Pengujian komunikasi data IPv6 ke IPv4.

Pada sistem ini dilakukan dengan koneksi antara IPv4 dan IPv6, komputer 1 disetting IPv4 tanpa menghilangkan IPv6 yang dia punya dan komputer 2 disetting IPv6 tanpa menghilangkan IPv4 yang dia punya maka setelah itu digunakan parameter *throughput* dan *delay* untuk menguji dan akan dilihat hasil dari pengujian tersebut.

#### **Pengujian**

Pengujian dalam hal ini menggunakan cara *download file* dari komputer 1 ke komputer 2 pada konfigurasi yang berbeda-beda. Agar mendapatkan hasil dari trafik yang didapat maka diperlakukan parameter, parameter tersebut adalah ukuran paket *file* yang akan di *download* masing-masing berukuran 5, 10, 20, 30, dan 40 *Megabyte* dalam keragaman pengukuran paket tersebut bisa dilihat nantinya pada table trafik yang didapat dan bisa dilihat perbedaan dari tiap ukuran *Megabyte* yang sudah ditentukan dengan *transfer* data dan *download* data.

## **HASIL DAN PEMBAHASAN**

#### **Pengujian Sistem**

Pengujian sistem ini dilakukan sesuai dengan skenario yang sudah dibuat pada bab 3, dimana nantinya di bab ujicoba ini mempunyai beberapa bentuk dari pengujian antara lain Tujuan Pengujian Skenario dan Tahapan Pengujian.

#### **Pengujian skenario komunikasi data IPv4 ke IPv4**

Pada pengujian komunikasi data IPv4 ini ada 2 komputer yang akan saling berkomunikasi data, yang 2 komputer disetting dengan IPv4. Dimana *Client* dan *server* nantinya akan berkomunikasi data yaitu dengan cara *download file* dengan size yang berbeda-beda

Dari langkah-langkah pengujian yang telah dilakukan tersebut, terdapat 2 cara penilaian pengujian dimana cara tersebut nantinya bisa dijadikan acuan untuk mengetahui apakah pengujian



ini dapat dinyatakan berhasil atau gagal. Berikut adalah 2 cara penilaian tersebut :

1. Pengujian Berhasil

Pengujian dikatakan berhasil jika *client* dan *server* dapat melakukan *transfer* data berupa *download file* dan apabila request IP yang dituju mendapatkan balasan request. Dan *server* bisa mengetahui hasil dan grafik *transfer* yang didapatkan dari komunikasi data tersebut.

2. Pengujian Gagal

Pengujian dikatakan gagal jika *client* tidak mendapatkan respon dari komputer *server* berupa request IP dan request untuk *download file* dari *server*, sehingga *client* tidak bisa berkomunikasi data dengan *server*.

### **Pengujian skenario komunikasi data IPv6 ke IPv6**

Pada pengujian komunikasi data IPv6 ini ada 2 komputer yang akan saling berkomunikasi data, yang 2 komputer disetting dengan IPv6. Dimana *Client* dan *server* nantinya akan berkomunikasi data yaitu dengan cara *download file* dengan size yang berbeda-beda.

Dari langkah-langkah pengujian yang telah dilakukan tersebut, terdapat 2 cara penilaian pengujian dimana cara tersebut nantinya bisa dijadikan acuan untuk mengetahui apakah pengujian ini dapat dinyatakan berhasil atau gagal. Berikut adalah 2 cara penilaian tersebut :

1. Pengujian Berhasil

Pengujian dikatakan berhasil jika *client* dan *server* dapat melakukan *transfer* data berupa *download file* dan apabila request IP yang dituju mendapatkan balasan request. Dan *server* bisa mengetahui hasil dan grafik *transfer* yang didapatkan dari komunikasi data tersebut.

2. Pengujian Gagal

Pengujian dikatakan gagal jika *client* tidak mendapatkan respon dari komputer *server* berupa request IP dan request untuk *download file* dari *server*, sehingga *client* tidak bisa berkomunikasi data dengan *server*.

### **Pengujian skenario komunikasi data IPv4 ke IPv6**

Pada pengujian komunikasi data IPv6 ini ada 2 komputer yang akan saling berkomunikasi data, komputer disetting dengan IPv4 dan IPv6 mempunyai dua ip dalam 1 komputer. Dimana *Client* dan *server* nantinya akan berkomunikasi data yaitu dengan cara *download file* dengan size yang berbeda-beda.

Dari langkah-langkah pengujian yang telah dilakukan tersebut, terdapat 2 cara penilaian pengujian dimana cara tersebut nantinya bisa dijadikan acuan untuk mengetahui apakah pengujian ini dapat dinyatakan berhasil atau gagal. Berikut adalah 2 cara penilaian tersebut :

1. Pengujian Berhasil

Pengujian dikatakan berhasil jika *client* dan *server* dapat melakukan *transfer* data berupa *download file* dan apabila request IP yang dituju mendapatkan balasan request. Dan *server* bisa mengetahui hasil dan grafik *transfer* yang didapatkan dari komunikasi data tersebut.

2. Pengujian Gagal

Pengujian dikatakan gagal jika *client* tidak mendapatkan respon dari komputer *server* berupa request IP dan request untuk *download file* dari *server*, sehingga *client* tidak bisa berkomunikasi data dengan *server*. Apabila komputer di setting IPv6 untuk komputer satu dan IPv4 untuk komputer dua maka tidak bisa melakukan koneksi di karena kedua ip saling berkaitan satu sama lain, maka dari itu harus ada 2 ip yaitu IPv4 dan IPv6 di setiap komputer.

### **Pengujian skenario komunikasi data IPv6 ke IPv4**

Pada pengujian komunikasi data IPv6 ini ada 2 komputer yang akan saling berkomunikasi data, komputer disetting dengan IPv4 dan IPv6 mempunyai dua IP dalam 1 komputer. Dimana *Client* dan *server* nantinya akan berkomunikasi data yaitu dengan cara *download file* dengan size yang berbeda-beda.

Dari langkah-langkah pengujian yang telah dilakukan tersebut, terdapat 2 cara penilaian pengujian dimana cara tersebut nantinya bisa dijadikan acuan untuk mengetahui apakah pengujian ini dapat dinyatakan berhasil atau gagal. Berikut adalah 2 cara penilaian tersebut :

1. Pengujian Berhasil

Pengujian dikatakan berhasil jika *client* dan *server* dapat melakukan *transfer* data berupa *download file* dan apabila request IP yang dituju mendapatkan balasan request. Dan *server* bisa mengetahui hasil dan grafik *transfer* yang didapatkan dari komunikasi data tersebut.

2. Pengujian Gagal

Pengujian dikatakan gagal jika *client* tidak mendapatkan respon dari komputer *server* berupa request IP dan request untuk *download file* dari *server*, sehingga *client* tidak bisa berkomunikasi data dengan *server*. Apabila komputer di setting IPv6 untuk komputer satu dan IPv4 untuk komputer

dua maka tidak bisa melakukan koneksi di karena kedua ip saling berkaitan satu sama lain, maka dari itu harus ada 2 ip yaitu IPv4 dan IPv6 di setiap komputer.

### Analisa Pengujian

Pada tahap ini dilakukan analisa pengujian yang didasarkan pada penilaian pengujian yang ada diskenario-skenario di atas yang terdiri dari 4 skenario, dengan parameter *throughput* dan *delay* berikut merupakan hasil dari keseluruhan uji coba pada skenario tersebut :

No	Skenario	throughput	delay
1	IPv4 ke IPv4	5Mb	58.256
		10Mb	23.847
		20Mb	17.788
		30Mb	15.236
		40Mb	13.271
2	IPv6 ke IPv6	5Mb	38.908
		10Mb	22.394
		20Mb	17.519
		30Mb	14.417
		40Mb	13.516

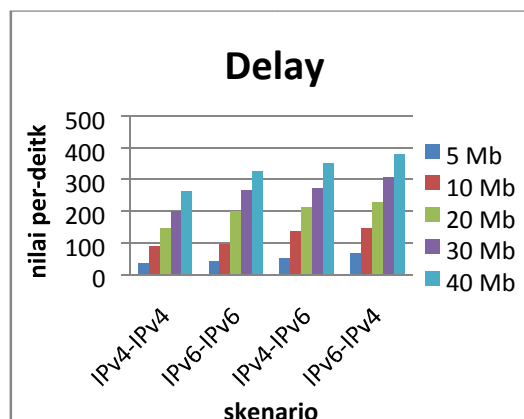
Table 5.1 hasil keseluruhan skenario 1 dan 2

No	Skenario	throughput	delay
3	IPv4 ke IPv6	5Mb	41.237
		10Mb	20.502
		20Mb	13.144
		30Mb	14.945
		40Mb	14.357
4	IPv6 ke IPv4	5Mb	31.445
		10Mb	18.596
		20Mb	11.231
		30Mb	9.897
		40Mb	9.171

Table 5.2 hasil keseluruhan skenario 3 dan 4

### Analisa Pengujian delay

Dari hasil uji coba maka akan dilakukan analisa di data atau hasil yang di peroleh dari *delay* dengan ukuran berbeda-beda, berikut analisa dari *delay* :



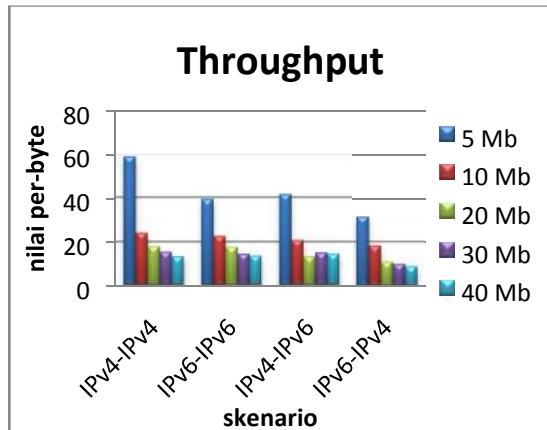
Gambar 5.33 Diagram hasil delay

Pada gambar 5.33 merupakan hasil dari *delay* yang sudah di uji coba di skenario-skenario sebelumnya dalam bentuk diagram. Dapat dilihat pada diagram bahwa ada perbedaan-perbedaan dalam nilai yang diperoleh di setiap masing-masing pengujian. Pada pengujian IPv6 ke IPv4 mendapatkan lebih besar nilai dari pada pengujian yang lain dan IPv4 ke IPv4 mendapatkan nilai yang lebih kecil dari pada pengujian yang lain.

Pada *delay* ini ditunjukkan nilai yang besar pada uji coba IPv6 ke IPv4 dikarenakan adanya mekanisme tambahan DSTM yang mempengaruhi *transfer* data yang dikirim atau diterima sehingga nilai *delay* lebih besar dari yang tidak ada mekanisme tambahan DSTM dalam *transfer* data yang diterima. Dan untuk uji coba yang IPv4 ke IPv6 juga ada mekanisme tambahan DSTM juga mempengaruhi *transfer* data yang dikirim atau diterima sehingga mendapatkan nilai *delay* lebih besar dari yang tidak ada mekanisme tambahan DSTM dalam *transfer* data yang diterima. Antara uji coba IPv4 ke IPv6 dan IPv6 ke IPv4 mengalami sedikit peningkatan nilai, kedua uji coba ini yang menggunakan mekanisme tambahan DSTM lebih bagus IPv4 ke IPv6 karena lebih kecil nilai yang didapat. Dan untuk yang tidak dipengaruhi mekanisme tambahan DSTM seperti IPv4 ke IPv4 dan IPv6 ke IPv6 lebih bagus IPv4 ke IPv4 karena nilai yang diperoleh lebih kecil dan dari kesimpulan diantara kesemua uji coba yang sudah dilakukan nilai yang paling bagus didapat yaitu IPv4 ke IPv4 dan yang paling kurang bagus yaitu IPv6 ke IPv4.

### Analisa Pengujian throughput

Dari hasil uji coba maka akan dilakukan analisa di data atau hasil yang di peroleh dari *throughput* dengan ukuran berbeda-beda, berikut analisa dari *throughput* :



Gambar 5.34 Diagram hasil *Throughput*

Pada gambar 5.34 merupakan hasil dari *throughput* yang sudah di uji coba di scenarioskenario sebelumnya dalam bentuk diagram. Dapat dilihat pada diagram bahwa ada perbedaanperbedaan dalam nilai *Throughput* yang diperoleh di setiap masing-masing pengujiannya. Untuk nilai *Throughput* dengan nilai ukuran *file* 5Mb mendapatkan nilai yang sangat tinggi disetiap uji coba yang dilakukan merupakan nilai yang didapat sangat cepat dalam hal mensampaikan data dan merupakan nilai yang sangat bagus. Dan ukuran *file* yang selanjutnya seperti ukuran 10Mb, 20Mb, 30Mb, 40Mb mengalami banyak penurunan disetiap ukuran *file* yang berbedabeda, itu semua dikarena ukuran *file* yang semakin besar dapat di simpulkan semakin besar ukuran *file* untuk mensampaikan data maka nilai yang didapat juga akan mengalami penurunan di setiap hasilnya.

Pada uji coba IPv6 ke IPv4 memiliki nilai yang sangat kecil di antara semua uji coba yang dilakukan dan itu merupakan nilai yang tidak bagus dan untuk IPv4 ke IPv4 memiliki nilai yang cukup besar diantara uji coba yang lain dan itu merupakan nilai yang sangat bagus dalam mendapatkan nilai *throughput*.

Tidak hanya ukuran *file* yang mempengaruhi turunnya nilai *throughput* dalam IPv6 ke IPv4 dan IPv4 ke IPv6 mendapatkan nilai kecil juga adanya pengaruh mekanisme tambahan DSTM seperti halnya uji coba pada *transferfile* sebelumnya juga karena adanya mekanisme tambahan DSTM. Untuk yang tidak menggunakan mekanisme tambahan DSTM mendapatkan nilai yang cukup bagus karena tidak ada mekanisme tambahan DSTM tersebut, untuk IPv4 ke IPv4 mendapat nilai yang sangat bagus dan tinggi diantara uji coba yang lainnya, IPv6 ke IPv6 mengalami penurunan sedikit dengan uji coba IPv4 ke IPv4 dan itu semua uji coba tanpa menggunakan mekanisme tambahan DSTM.

## PENUTUP

### Kesimpulan

Dari pengujian yang dilakukan pada skenario-skenario yang telah di uji coba dapat disimpulkan bahwa sekenario yang menggunakan IPv4 ke IPv4 merupakan yang sangat bagus komunikasi datanya antara komputer 1 dengan komputer 2 mengalami *throughput* dan *delay* yang sangat kecil dan itu merupakan hasil yang sangat baik dan untuk sekenario yang menggunakan IPv6 ke IPv4 mendapatkan nilai yang sangat besar diantara nilai hasil uji coba yang lain dan itu merupakan hasil yang sangat tidak bagus.

### Saran

Berdasarkan apa yang sudah dilakukan pada Tugas Akhir ini maka didapatkan beberapa Saran yang mungkin bisa digunakan untuk pengembangan, yaitu :

1. Karena uji coba menggunakan parameter uji coba *throughput* dan *delay*. bisa menggunakan parameter uji coba yang lain..
2. Diharapkan juga bisa diimplementasikan di *operation system* lain dan metode lain.

## DAFTAR PUSTAKA

- [1] Bound J. 2004. *Dual Stack Transition Mechanism*. IETF draft-bound-dstm-exp-01.txt.
- [2] Houston G. 2005. *IPv4 AddresReport*. [Http://www.potaroo.net/tools/ipv4/index.html](http://www.potaroo.net/tools/ipv4/index.html)
- [3] Jogiyanto. 1995. *Analisa dan Perancangan Sistem Informasi*. Andy Offset. Yogyakarta
- [4] Postel J. 1981. *Internet Protocol*. Request of Comment 791. IETF
- [5] Ruiz PM. 2002. *Dual Stack Transition Mechanism*. [www.ipv6es.com/02/docs/pedro\\_ruiz\\_2.pdf](http://www.ipv6es.com/02/docs/pedro_ruiz_2.pdf)

{ Halaman sengaja dikosongkan }  
{ Jurnal **Melek IT** }

(5)  
**RANCANGAN SISTEM NAVIGASI PARIWISATA MENGGUNAKAN WEB SERVICE  
DENGAN PLATFORM ANDROID**

**Hasbullah<sup>1</sup>, Nia Saurina<sup>2</sup>, Beny Y.V. Nasution<sup>3</sup>**

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Wijaya Kusuma Surabaya

[Ab.boets@gmail.com](mailto:Ab.boets@gmail.com)

**ABSTRAK**

Kebutuhan informasi jalur menuju daerah wisata sangat penting bagi wisatawan. Terutama untuk wisatawan yang menggunakan fasilitas angkutan umum menuju lokasi pariwisata. Tak dapat dihindari mereka akan di pusingkan karena banyaknya jalan dan rumitnya lalu lintas kota Surabaya. Jauh dekat tempat pariwisata bergantung pada jalan yang dilewatinya, demikian juga dengan banyak dan sedikitnya biaya yang dikeluarkan bergantung pada jalur dan fasilitas transportasi yang digunakan.

Dengan adanya permasalahan tersebut maka akan dibangun sebuah sistem rute pariwisata yang menampilkan rute menuju lokasi pariwisata dalam bentuk peta digital. Selain menampilkan rute aplikasi ini juga menampilkan estimasi biaya yang di keluarkan untuk sampai di lokasi pariwisata. Dengan adanya sistem ini diharapkan bisa membantu para wisatawan untuk lebih mengenal Surabaya dari informasi- informasi yang disediakan dan dapat melakukan perjalanan wisata dengan efektif dan mudah.

**Kata kunci :** Pariwisata, Rute, Estimasi Biaya

**PENDAHULUAN**

**Latar Belakang**

Surabaya sebagai ibu kota propinsi Jawa Timur merupakan pusat budaya, pendidikan, pariwisata, maritim, industri, perdagangan. Sedangkan Surabaya sendiri tidak memiliki sumber alam yang bisa digali. Dengan sedikitnya potensi yang dimiliki kota Surabaya maka sektor pariwisata lah yang perlu diutamakan. Untuk mendapatkan informasi menuju lokasi pariwisata tidaklah semudah sekedar melihat sebuah peta. Maka dari itu kebutuhan informasi mengenai jalur untuk menuju lokasi pariwisata sangatlah besar. Begitu juga dalam hal kepariwisataan, kebutuhan informasi jalur menuju daerah wisata sangatlah penting bagi wisatawan. Terutama untuk wisatawan yang menggunakan fasilitas angkutan umum menuju lokasi pariwisata. Tak dapat dihindari mereka akan di pusingkan karena banyaknya jalan dan rumitnya lalu lintas kota Surabaya. Jauh dekat tempat pariwisata bergantung pada jalan yang dilewatinya, demikian juga dengan banyak sedikitnya biaya yang dikeluarkan bergantung pada jalur dan fasilitas transportasi yang digunakan. Untuk menghemat biaya wisatawan harus menentukan lokasi pariwisata dan mencari rute menuju lokasi.

Perkembangan teknologi ikut mendukung perkembangan aplikasi Sistem Informasi Geografis. Aplikasi Sistem Informasi Geografis berbasis web kini dikembangkan menuju arah aplikasi terdistribusi, dimana pemakai tidak terikat pada suatu komputer khusus untuk mendapatkan informasi data spasial. Menentukan lokasi

pariwisata dan mencari rute terpendek kini mulai di wujudkan dengan cara yang lebih mudah dan modern, salah satunya dalam bentuk peta digital dimana akan di tampilkan data dalam suatu informasi berbasis geografis.

Metode yang lebih modern untuk mencari rute menuju lokasi pariwisata dengan aplikasi web service berbasis sistem informasi geografis. Aplikasi ini menampilkan rute menuju lokasi pariwisata dalam bentuk peta digital. Selain menampilkan rute aplikasi ini juga menampilkan estimasi biaya yang di keluarkan untuk sampai di lokasi pariwisata. Karena pentingnya informasi tentang perjalanan wisata. maka dalam Tugas Akhir kali ini dibuat aplikasi perjalanan wisata menggunakan web service dengan platform android yang diharapkan bisa membantu para wisatawan untuk lebih mengenal Surabaya dari informasi-informasi yang disediakan dan dapat melakukan perjalanan wisata dengan efektif dan mudah.

Pada Tugas Akhir ini akan dibahas tentang implementasi web service berbasis sistem informasi geografis untuk pencarian rute menuju lokasi pariwisata dan estimasi biaya yang di keluarkan untuk sampai lokasi pariwisata.

**TINJAUAN PUSTAKA**

**Web Service**

Web service adalah sistem software yang dirancang untuk mendukung interoperabilitas mesin-ke-mesin yang dapat berinteraksi melalui jaringan. Web service memiliki antarmuka yang dijelaskan dalam format mesin-processable (khusus WSDL).

Sistem lain berinteraksi dengan Web service dalam cara ditentukan oleh deskripsi dengan menggunakan pesan SOAP, biasanya disampaikan menggunakan HTTP dengan serialisasi XML dalam hubungannya dengan Web lainnya yang terkait standar. Dalam pengertian yang sederhana, XML Web Services dapat di definisikan sebagai aplikasi yang diakses oleh aplikasi yang lain. Mungkin orang berpendapat itu semacam web site, tetapi itu bukan demikian..

Web service bertujuan untuk meningkatkan kolaborasi antar pemrogram dan perusahaan, yang memungkinkan sebuah fungsi di dalam Web Service dapat dipinjam oleh aplikasi lain tanpa perlu mengetahui detail pemrograman yang terdapat di dalamnya.

Beberapa alasan mengapa digunakannya web service adalah sebagai berikut:

1. Web service dapat digunakan untuk mentransformasikan satu atau beberapa bisnis logic atau class dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. Web service memiliki kemudahan dalam proses deployment-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. Web service cukup di-upload ke web server dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.
3. Web service berjalan di port 80 yang merupakan protokol standar HTTP, dengan demikian web service tidak memerlukan konfigurasi khusus di sisi firewall.

Web service dapat diimplementasikan baik untuk lingkungan internal (intranet) maupun untuk lingkungan publik (internet). Dalam lingkungan internal, web service memberikan kemudahan dan solusi baru untuk mengintegrasikan berbagai platform aplikasi yang ada dalam organisasi atau perusahaan.

### **GPS (Global Positioning System)**

Gps adalah Sistem radio navigasi dan penentuan posisi menggunakan satelit yang dimiliki dan dikelola oleh Amerika Serikat. GPS didesain untuk memberikan posisi dan kecepatan 3D serta informasi mengenai waktu secara kontinyu, dan dapat digunakan oleh banyak orang sekaligus. Saat ini GPS mulai banyak diaplikasikan di Indonesia, terutama terkait dengan aplikasi-aplikasi yang menuntut informasi tentang posisi.

Beberapa kemampuan GPS antara lain dapat memberikan informasi tentang posisi, kecepatan, dan waktu secara cepat, akurat, murah, dimana saja di bumi ini tanpa tergantung cuaca. Hal

yang perlu dicatat bahwa GPS adalah satu-satunya sistem navigasi ataupun sistem penentuan posisi dalam beberapa abad ini yang memiliki kemampuan handal seperti itu. Ketelitian dari GPS dapat mencapai beberapa mm untuk ketelitian posisinya, beberapa cm/s untuk ketelitian kecepatannya dan beberapa nanodetik untuk ketelitian waktunya.

Komponen utama dari receiver GPS adalah :

1. Antena dengan pre-amplifier ;
2. Bagian RF (Radio Frequency) dengan pengidentifikasi sinyal dan pemroses sinyal;
3. Pemroses mikro untuk pengontrol receiver, data sampling, dan pemroses data
4. Osilator presisi 24
5. Catu daya
6. Unit perintah dan tampilan
7. Memori serta perekam data

Cara kerja sistem GPS pada dasarnya adalah menentukan jarak antara posisi satelit-satelit GPS pada orbitnya di angkasa luar ke alat penerima GPS. Dengan minimal 4 signal satelit yang diterima pada alat penerima GPS, maka alat penerima GPS dapat menghitung, dengan tingkat ketelitian tertentu,

### **GIS (Geographic Information System)**

Geographic Information System (GIS) adalah suatu sistem informasi yang dirancang untuk bekerja dengan data yang bereferensi spasial atau berkoordinat geografi atau dengan kata lain suatu SIG adalah suatu sistem basis data dengan kemampuan khusus untuk menangani data yang bereferensi keruangan (spasial) bersamaan dengan seperangkat operasi kerja. Sistem Informasi geografi adalah suatu sistem Informasi yang dapat memadukan antara data grafis (spasial) dengan data teks (atribut) objek yang dihubungkan secara geografis di bumi (*georeference*).

Secara umum ada dua jenis data yang diolah pada GIS :

Data Spasial adalah sebuah data yang berorientasi geografis, memiliki sistem koordinat tertentu sebagai dasar referensinya dan mempunyai dua bagian penting yang membuatnya berbeda dari data lain, yaitu informasi lokasi (spasial) dan informasi deskriptif (*attribute*) yang dijelaskan berikut ini :

1. Informasi lokasi (spasial), berkaitan dengan suatu koordinat baik koordinat geografi (lintang dan bujur) dan koordinat XYZ, termasuk diantaranya informasi datum dan proyeksi.
2. Informasi deskriptif (atribut) atau informasi non spasial, suatu lokasi yang memiliki beberapa keterangan yang berkaitan



dengannya, contohnya : jenis vegetasi, populasi, luasan, kode pos, dan sebagainya.  
Data Atribut : berupa data yang terpusat.

GIS dapat menghasilkan gambar dua atau tiga dimensi dari sebuah area yang menampilkan kekhasan alam seperti alam (bukit, pegunungan, sungai) ciri buatan seperti jalan dan garis. Gambar GIS digunakan untuk membuat perkiraan ukuran menyatukan data dan menguji rencana dengan bantuan komputer. Database GIS yang berisi kumpulan set informasi dinamakan *layer*. Tiap *layer* mempresentasikan sebuah tipe data geografis khusus. Contohnya satu *layer* mungkin memuat informasi jalan-jalan di suatu area. *Layer* lainnya mungkin berisi informasi data ketinggian tersebut. GIS dapat mengkombinasikan *layer – layer* ini kedalam suatu gambar yang menampilkan jalan dan ketinggian yang saling berhubungan satu dengan yang lain.

Fungsi – fungsi GIS secara khusus antara lain:

1. *Network Analyst*: modul perangkat lunak yang digunakan untuk melakukan analisis yang berhubungan dengan jaringan ( routing )
2. *Internet Map Server* : modul perangkat lunak yang digunakan untuk mempublikasikan peta ke jaringan internet ( web based GIS ) sehingga dapat diakses dengan menggunakan program browser.
3. *Analyst* : modul perangkat lunak yang digunakan untuk membuat menganalisa dan menampilkan data permukaan berupa gambar seperti dataran tinggi maupun permukaan digital

### Location Based Service

Location Based Service (LBS) atau dalam bahasa Indonesia diartikan sebagai Layanan Berbasis Lokasi merupakan suatu layanan yang bereaksi aktif terhadap perubahan entitas posisi sehingga mampu mendeteksi letak objek dan memberikan layanan sesuai dengan letak objek yang telah diketahui. LBS ini bekerja memanfaatkan lokasi dari devices untuk menyediakan informasi mengenai lokasi dengan memanfaatkan Global Positioning System (GPS).

### Google Maps

Google Maps merupakan layanan web milik Google yang menyediakan pencarian peta online, informasi jalan, serta berbagai data geografis lainnya. Layanan web ini selain menyediakan platform pencarian geografis lokal, juga menyediakan informasi lokal lain seperti keadaan jalan, panduan arah berkendara, ataupun direktori bisnis yang cukup lengkap.

Google Maps menawarkan peta yang dapat digeser (panned), diperbesar (zoom in), diperkecil

(zoom out), dapat diganti dalam beberapa mode (maps, satelit, hybrid, dan lain- lain), fitur pencarian rute (routing), penunjuk arah dari satu objek peta ke objek yang lain (direction) dan juga pencarian tempat (place). Sampai saat ini, Google Maps masih berada dalam tahap beta, dan masih terus dikembangkan dengan data yang selalu diperbarui secara berkala.

### Eclipse

Eclipse adalah sebuah IDE (Integrated Development Environment) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (platform-independent).

Berikut ini adalah sifat dari Eclipse:

1. Multi-platform: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
2. Multi-language: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lain seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.
3. Multi-role: Selain sebagai IDE untuk pengembangan aplikasi. Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak seperti dokumentasi, pengujian perangkat lunak, pengembangan web, dan lain sebagainya.

Pada saat ini, Eclipse merupakan salah satu IDE favorit karena gratis dan open source. Open source berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan membuat komponen yang disebut plug-in.

### Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Fungsinya sama seperti sistem operasi Symbian di Nokia, iOS di Apple dan BlackBerry. Android tidak terikat ke satu merek Handphone, beberapa merek HP terkenal yang sudah memakai Android antara lain Samsung, Sony Ericsson, HTC, Nexus, Motorola, dan lain-lain.

Android pertama kali dikembangkan oleh perusahaan bernama Android Inc yang kemudian pada tahun 2005 di akuisisi oleh raksasa Internet Google. Android dibuat dengan basis kernel Linux yang telah dimodifikasi, dan untuk setiap release-nya diberi kode nama berdasarkan nama hidangan makanan.

## Java

Java adalah bahasa pemrograman tingkat tinggi yang berorientasi objek dan program java tersusun dari bagian yang disebut kelas. Kelas terdiri atas metode-metode yang melakukan pekerjaan dan mengembalikan informasi setelah melakukan tugasnya. Para pemrogram Java banyak mengambil keuntungan dari kumpulan kelas di pustaka kelas Java, yang disebut dengan *Java Application Programming Interface* (API). Kelas-kelas ini diorganisasikan menjadi sekelompok yang disebut paket (*package*). Java API telah menyediakan fungsionalitas yang memadai untuk menciptakan *applet* dan aplikasi canggih. Jadi ada dua hal yang harus dipelajari dalam Java, yaitu mempelajari bahasa Java dan bagaimana menggunakan kelas pada Java API. Kelas merupakan satu-satunya cara menyatakan bagian eksekusi program, tidak ada cara lain. Pada Java program *javac* untuk mengkompilasi file kode sumber Java menjadi kelas-kelas *bytecode*. File kode sumber mempunyai ekstensi \*.java. Kompilator *javac* menghasilkan file *bytecode* kelas dengan ekstensi \*.class. Interpreter merupakan modul utama sistem Java yang digunakan aplikasi Java dan menjalankan program *bytecode* Java.

Sebagai tipe maka pada saat jalan, program menciptakan objek-objek yang merupakan instan-instan kelas. Kelas dapat mewarisi kelas lain. Java tidak mengijinkan pewarisan jamak namun menyelesaikan kebutuhan pewarisan jamak dengan fasilitas antarmuka yang lebih elegan.

Seluruh objek diprogram harus dideklarasikan lebih dulu sebelum digunakan. Ini merupakan keunggulan Java yaitu *Statically Typed*. Pemaksaan ini memungkinkan kompilator Java menentukan dan melaporkan terjadinya pertentangan (ketidakkompatibelan) tipe yang merupakan barikade awal untuk mencegah kesalahan yang tidak perlu (seperti mengurangi variabel bertipe integer dengan variabel bertipe string). Pencegahan sedini mungkin diharapkan menghasilkan program yang bersih. Kebaikan lain fitur ini adalah kode program lebih dapat dioptimasi untuk menghasilkan program berkinerja tinggi.

## PHP (*Hypertext preprocessor*)

PHP adalah bahasa *scripting server-side*, artinya bahasa yang digunakan pada server dengan tanpa perlu melakukan kompilasi tetapi cukup menuliskan tulisan dalam bentuk ASCII-nya saja. PHP sangat mirip dengan penulisan bahasa C, juga mempunyai karakteristik yang mirip dengan perl. Bahasa PHP ini sangat mudah untuk dipelajari.

Dengan menggunakan PHP, kita dapat melakukan *update database*, membuat *database*, dan melakukan perhitungan matematis (termasuk fungsi *trigonometri* yang kompleks). Kita juga dapat membuat dan menghapus file-file secara acak di system kita, bergantung pada level sekuriti yang menjalankan PHP. Kita juga dapat membuat koneksi jaringan *internet* dan melayani koneksi tersebut. Secara teori, kita juga dapat membuat sebuah *web server* yang canggih menggunakan PHP.

## MY SQL

MY SQL adalah suatu software sistem manajemen *database*. *Database* adalah suatu koleksi data yang terstruktur. MySQL merupakan *Relational Database Management system* (RDBMS) yang didistribusikan secara gratis di bawah lisensi *General Public License* (GPL). Sistem manajemen *database* seperti MySQL diperlukan untuk menambahkan, mengakses, memproses data yang disimpan di komputer. MySQL menggunakan *Standart Structured Query Language* (SQL), yaitu bahasa standart yang paling banyak digunakan untuk mengakses *database*. Alasan penggunaan MySQL sebagai *software database server* adalah MySQL dari awal didesain untuk menangani *database* yang cukup sangat besar, lebih cepat dari solusi yang ada. Konektivitas, kecepatan dan *security* yang baik membuat MySQL sangat cocok digunakan di internet.

## ANALISA DAN PERANCANGAN

### Analisa Sistem

Permasalahan pada penelitian ini adalah Untuk mendapatkan informasi menuju lokasi pariwisata tidaklah semudah sekedar melihat sebuah peta. Maka dari itu kebutuhan informasi mengenai jalur untuk menuju lokasi pariwisata sangatlah besar. Begitu juga dalam hal kepariwisataan, kebutuhan informasi jalur menuju daerah wisata sangatlah penting bagi wisatawan. Terutama untuk wisatawan yang menggunakan fasilitas angkutan umum menuju lokasi pariwisata. Serta bagaimana mengimplementasikan web service dengan android sebagai solusi permasalahan tersebut.

### Analisa Kebutuhan Data

Kebutuhan data yang diperlukan dalam membuat aplikasi ini terbagi menjadi 2, yaitu : Kebutuhan Data Fungsional dan Kebutuhan Data Non-Fungsional.

### Kebutuhan Data Fungsional

Sistem yang akan dibangun merupakan sistem untuk pemberian informasi tentang pariwisata yang terdapat disurabaya dengan memberikan alternatif jalur terpendek menuju lokasi pariwisata serta memberikan informasi biaya yang akan dikeluarkan untuk sampai dilokasi pariwisata. Sehingga dibutuhkan data yang akan digunakan sebagai inputan informasi tentang pariwisata yang ada disurabaya.

Untuk lebih jelasnya lagi akan dijelaskan kebutuhan data fungsional, yaitu :

1. Nama Tempat Pariwisata, adalah nama dari tempat pariwisata tersebut.
2. Alamat Tempat Pariwisata, yaitu penjelasan tentang alamat lokasi atau jalan dari tempat pariwisata tersebut berada
3. Koordinat Tempat Pariwisata, yaitu posisi lokasi tempat pariwisata tersebut berada fungsi dari koordinat ini adalah untuk menandakan titik yang akan ditampilkan pada Map lokasi penentuan rute terpendek menuju Tempat Pariwisata.
4. Penentuan biaya, yaitu prediksi biaya yang dikeluarkan dengan menggunakan alat transportasi umum lyn untuk sampai ke lokasi pariwisata.

Data-data yang telah dijelaskan di atas akan diproses pada web service server.

Pada tabel 3.1 di bawah ini adalah data beberapa tempat pariwisata

Jenis Pariwisata	Nama Pariwisata	Alamat	Telepon
1. Wisata Hiburan	- Kebun Binatang Surabaya	Jl. Setail no.1 Surabaya	(031) 5678703
	- Pantai Ria Kenjeran	Jl. Sukolilo Lor Surabaya	(031) 3818778
2. Wisata Religi	- Taman Hiburan Remaja	Jl. Kusuma Bangsa No. 116-118, Surabaya	(031) 5316557
	- Masjid Cheng ho	Jl. Gading No.2, Surabaya	(031) 549 0879
3. Wisata Sejarah	- Sunan Ampel	Kelurahan Ampel - Kec Semampir, Surabaya	(031) 749 5610
	- Tugu Pahlawan	Jl. Pahlawan Bubutan, Surabaya	(031) 3571100
4. Wisata Belanja	- Monumen Kapal Selam	Jl. Pemuda, Surabaya	(031) 549 0410
	- Tunjungan Plaza	Jl. Jenderal Basuki Rahmat 2-12 . Surabaya	(031) 531 1088
	- Delta	Jl. Pemuda No. 33-37 Surabaya	(031) 547 3968
	- Royal Plaza	Jl. Raya Ahmad Yani No. 16-18, Surabaya	(031) 8270866
	- Ciputra World	Jl. Mayor Jenderal Sungkono, Surabaya	(031) 5632828

Tabel 3.1 Tabel Data Tempat Pariwisata

Data pada tabel 3.1 adalah data dari tempat pariwisata, dimana pada tabel ini terdapat 4 jenis pariwisata, data tabel di atas yang akan diakses oleh user untuk mendapatkan informasi tentang tempat pariwisata, data tersebut yang pada akhirnya client pada android dapat melakukan proses pemanggilan method atau fungsi yang ada pada web service untuk mendapatkan informasi lokasi pariwisata.

Setelah mendapatkan data pariwisata selanjutnya di buatlah data biaya angkutan umum yang akan digunakan menuju lokasi pariwisata untuk lebih memperkuat data yang telah ada.

### Kebutuhan Data Non-Fungsional

Kebutuhan data non-fungsional terbagi menjadi dua yaitu kebutuhan hardware, dan kebutuhan software.

1. Kebutuhan perangkat keras (Hardware)

Kebutuhan hardware yang dibutuhkan yaitu,

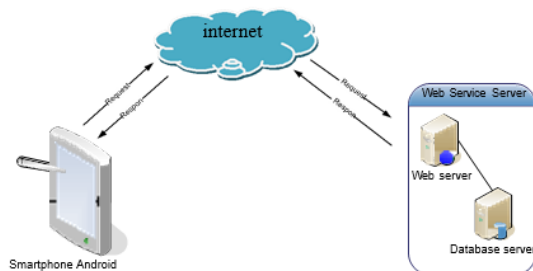
1. Laptop
2. Smartphone Android

3. Web Server
  4. Database Server
  5. Router
  6. Koneksi internet
2. Kebutuhan perangkat lunak (Software)
- Kebutuhan software yang dibutuhkan yaitu,
1. Java Development Kit, untuk kompilasi kode-kode program.
  2. Eclipse 3.5, sebagai perangkat lunak yang digunakan untuk membangun aplikasi client pada android.
  3. Android SDK, untuk pengembang aplikasi android.
  4. Android Development Tool, sebagai plugin android pada eclipse.

Android Virtual Device, sebagai emulator untuk menjalankan sistem android.

### Arsitektur Sistem

Pada Arsitektur Sistem ini menjelaskan tentang konsep perencanaan pada sistem. Seperti yang akan dijelaskan pada gambar dibawah



Gambar 3.1 Gambar Konsep Sistem

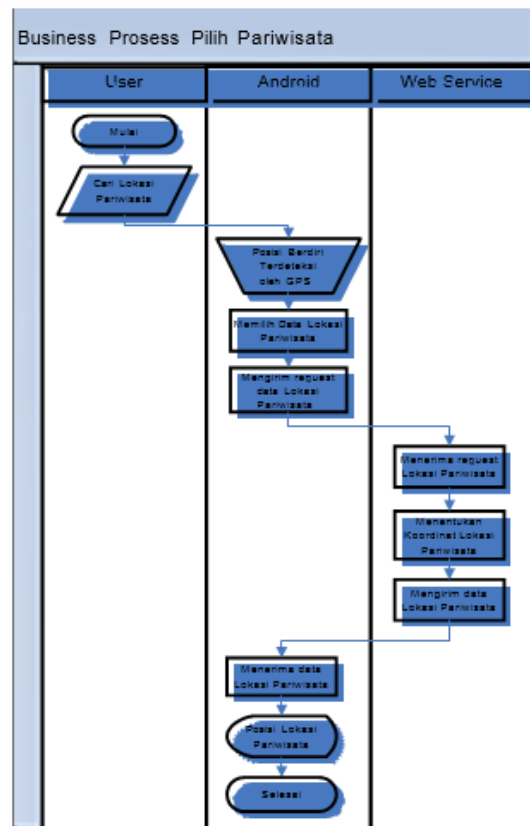
Gambar 3.1 bisa dilihat bahwa smartphone Android yaitu sebagai service request yaitu aplikasi yang bertindak sebagai client web service. Client Android me-request data lokasi pariwisata dan rute dari satelit GPS ke web service. Di dalam web service terdapat web server dan database server. Data request kemudian diproses dan web service memberikan respon data lokasi pariwisata dan rute. Android menerima data respon yang diberikan web service kemudian ditampilkan dalam bentuk interface, sehingga lokasi pariwisata dan rute menuju lokasi dapat ditampilkan dalam bentuk maps pada client android.

### Proses Bisnis

Untuk memperjelas gambaran yang ada, maka akan dibuat rangkaian kerja dalam bentuk business process. Yang mana akan dijelaskan pada tiga business process :

### 1. Pilih pariwisata

1. Bisnis proses yang pertama adalah pilih pariwisata yang mana akan dijelaskan pada gambar dibawah ini.

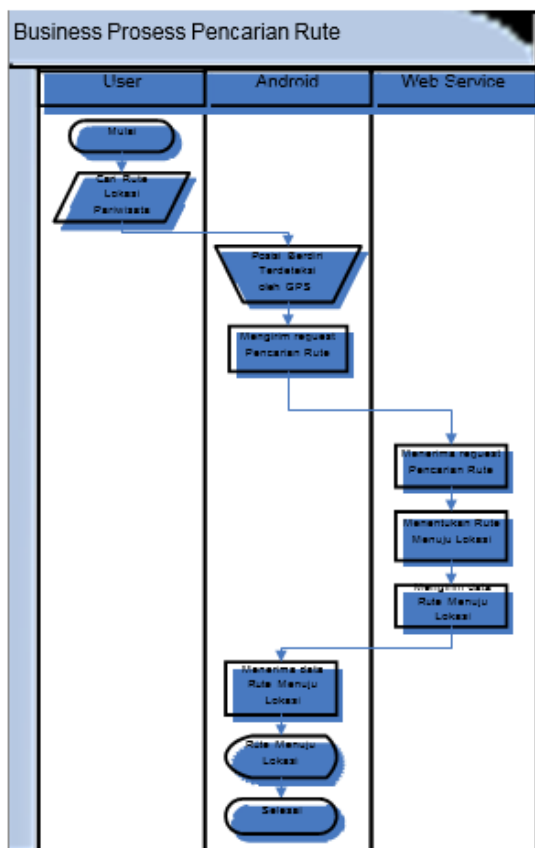


Gambar 3.2 Proses Bisnis Pilih Lokasi Pariwisata

Gambar 3.2 menjelaskan tentang proses pilih lokasi pariwisata. Dimana pada posisi berdiri client telah terdeteksi oleh GPS. Dengan memanfaatkan GPS yang ada pada smartphone android untuk mendapatkan titik koordinat dimana posisi berdiri client. Setelah itu client memilih data lokasi pariwisata yang ada pada android. Kemudian android melakukan request data lokasi pariwisata kepada web service. Web service menerima request dari android, kemudian web service mencari koordinat lokasi pariwisata. Setelah koordinat lokasi pariwisata sudah ditentukan kemudian web service mengirim data lokasi pariwisata ke android. Android menerima data lokasi pariwisata, data yang diterima kemudian di proses oleh android untuk menampilkan posisi lokasi pariwisata yang berupa output dalam bentuk tampilan maps

### 2. Pencarian rute

Dari bisnis proses yang telah dijelaskan sebelumnya tentang pilih pariwisata, maka bisnis proses selanjutnya adalah pencarian rute yang mana



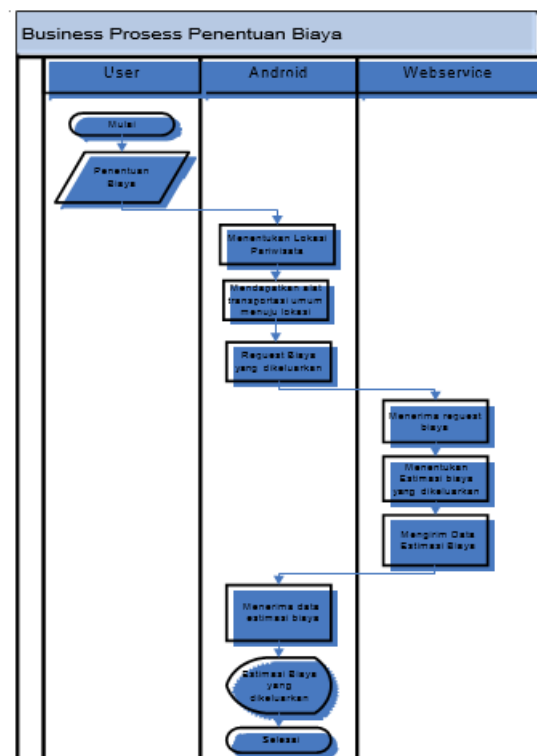
akan dijelaskan pada gambar dibawah ini.

Gambar 3.3 Bisnis Proses Pencarian Rute Lokasi Pariwisata

Gambar 3.3 menjelaskan tentang proses pencarian rute lokasi pariwisata. Dimana pada posisi berdiri telah terdeteksi oleh GPS. Dengan memanfaatkan GPS yang ada pada smartphone android untuk mendapatkan titik koordinat dimana posisi berdiri client. Android melakukan request pencarian rute terpendek menuju lokasi pariwisata kepada web service. Web service menerima request dari android, kemudian web service menentukan rute menuju lokasi pariwisata, setelah sudah ditentukan rute maka webservice mengirim data rute menuju lokasi pariwisata. Android menerima data rute menuju lokasi pariwisata, data yang diterima kemudian di proses oleh android untuk menampilkan rute menuju lokasi pariwisata yang berupa output dalam bentuk tampilan maps.

### 3. Penentuan Biaya

Bisnis proses selanjutnya adalah penentuan biaya yang mana akan dijelaskan pada gambar dibawah ini

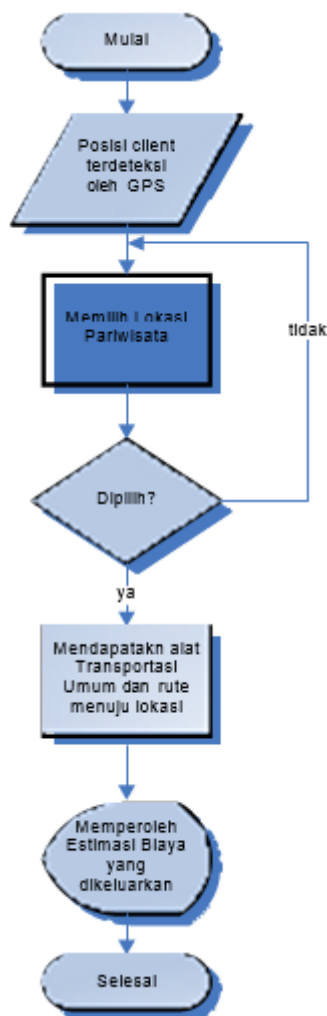


Gambar 3.4 Proses Bisnis Penentuan Biaya

Gambar 3.4 menjelaskan tentang proses penentuan biaya. Dimana client terlebih dahulu menentukan lokasi pariwisata, setelah menentukan lokasi pariwisata kemudian mendapatkan alat transportasi umum yang akan digunakan menuju lokasi pariwisata. Android kemudian mengirim request estimasi biaya yang dikeluarkan. Web service menerima request dari android, kemudian web service menentukan estimasi biaya yang akan dikeluarkan, setelah sudah ditentukan estimasi biaya maka webservice mengirim data estimasi biaya. Android menerima data request estimasi biaya, data yang diterima kemudian di proses oleh android untuk menampilkan estimasi yang berupa output estimasi biaya yang dikeluarkan.

### Diagram Proses Aplikasi

Dari semua bisnis proses yang telah dijelaskan sebelumnya tentang pilih pariwisata, pencarian rute serta penentuan biaya maka dibuatlah diagram proses aplikasi pada gambar dibawah ini.



Gambar 3.5 Diagram Proses Aplikasi

Gambar 3.5 menjelaskan tentang diagram proses aplikasi. Dimana flowchart ini menjelaskan tentang gambaran umum aplikasi, Dimana pada posisi client berdiri telah terdeteksi oleh GPS. Dengan memanfaatkan GPS yang ada pada smartphone android untuk mendapatkan titik koordinat dimana posisi berdiri klien yang merupakan titik koordinat client berada yang mana akan menjadi acuan untuk mencari lokasi pariwisata yang sudah di simpan pada database server yang ada di web service. Kemudian client memilih lokasi pariwisata. Jika lokasi pariwisata telah dipilih maka akan mendapatkan alat transportasi umum serta rute yang akan ditampilkan android dalam bentuk peta google maps. Jika lokasi sudah dipilih dan telah mendapatkan alat transportasi umum serta rute menuju lokasi maka akan memperoleh estimasi biaya yang dikeluarkan, atau yang diterima kemudian di proses oleh android untuk menampilkan lokasi pariwisata serta rute yang

berupa output dalam bentuk tampilan maps, serta estimasi biaya yang dikeluarkan untuk sampai dilokasi pariwisata

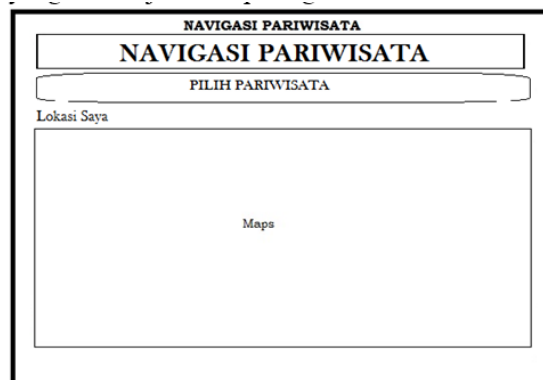
### Perancangan Antarmuka

*Interface* atau antarmuka merupakan tampilan dari suatu program aplikasi yang berperan sebagai media komunikasi yang digunakan sebagai sarana berdialog antara program dengan *client*. Sistem yang akan dibangun diharapkan menyediakan *interface* yang mudah dipahami dan digunakan oleh *client*.

Untuk pengembangan perancangan antarmuka pada aplikasi web service *client* menggunakan Android SDK (*Software Development Kit*). Untuk menjalankan atau menggunakan Android SDK harus di install ADT (*Android Developer Tools*). ADT adalah plugin untuk Eclipse yang menyediakan seperangkat alat yang terintegrasi dengan IDE Eclipse, yang berfungsi untuk membantu mengembangkan aplikasi Android yang akan dibuat. ADT menyediakan akses GUI untuk banyak baris perintah alat SDK serta desain alat UI untuk *prototyping* cepat (*emulator*), perancangan, dan pembangunan antarmuka pengguna aplikasi. Berikut ini adalah perancangan antarmuka pada sistem yang akan dibuat.

### Perancangan Antarmuka Android

Perancangan antarmuka yang pertama adalah halaman utama navigasi pariwisata seperti yang akan dijelaskan pada gambar dibawah ini.

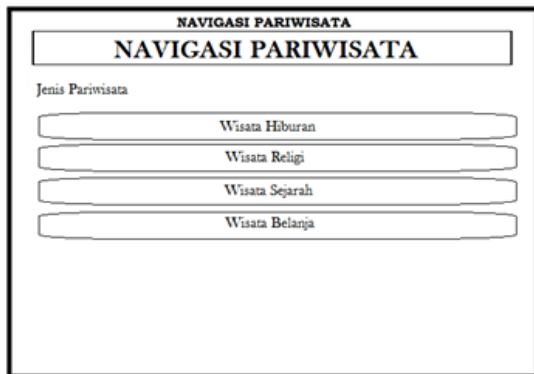


Gambar 3.6 Antarmuka Halaman Utama Navigasi Pariwisata

Gambar 3.6 adalah perancangan antarmuka halaman utama navigasi pariwisata. Perancangan antarmuka ini terdapat beberapa bagian yaitu gambar Icon navigasi pariwisata dari aplikasi yang dibuat, gambar maps akan menampilkan posisi berdiri telah terdeteksi oleh GPS dalam bentuk peta digital. sedangkan untuk

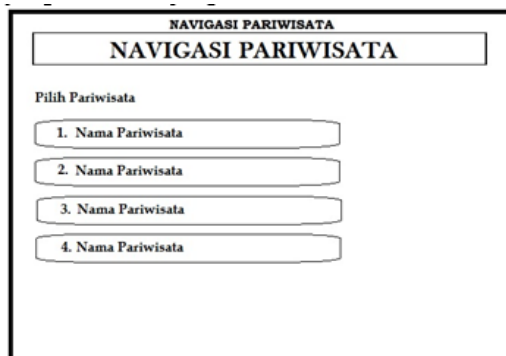


tombol pilih pariwisata adalah untuk memilih tempat pariwisata.



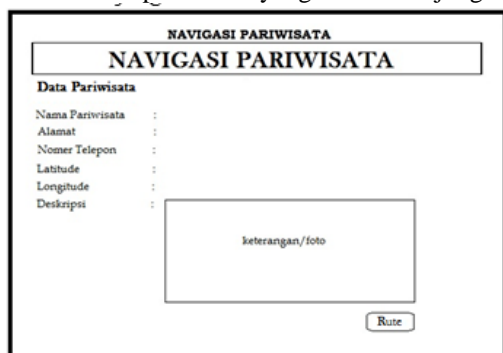
Gambar 3.7 Antarmuka jenis pariwisata

Gambar 3.7 adalah perancangan antarmuka untuk memilih jenis pariwisata. Perancangan antarmuka ini terdapat beberapa bagian yaitu gambar Icon navigasi pariwisata dari aplikasi yang dibuat, sedangkan tombol-tombol jenis pariwisata untuk memilih jenis pariwisata yang akan dikunjungi.



Gambar 3.8 Antarmuka memilih tempat pariwisata

Gambar 3.8 adalah perancangan antarmuka untuk memilih tempat pariwisata. Perancangan antarmuka ini terdapat beberapa bagian yaitu gambar Icon navigasi pariwisata dari aplikasi yang dibuat, sedangkan tombol-tombol pilih pariwisata untuk memilih pariwisata yang akan dikunjungi.

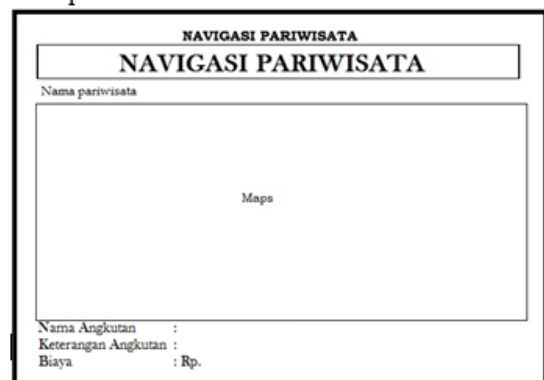


Gambar 3.9 Antarmuka keterangan pariwisata

Gambar 3.9 adalah gambar perancangan Keterangan Tempat Pariwisata. Tampilan ini akan

tampil setelah sudah memilih tempat pariwisata. Bagianya terdiri dari :

1. Nama Pariwisata, berisi nama dari pariwisata yang telah dipilih
2. Alamat, berisi informasi alamat pariwisata
3. Nomer telepon, berisi tentang nomer telepon lokasi pariwisata
4. Latitude, berisi tentang posisi koordinat lokasi pariwisata
5. Longitude, berisi tentang posisi koordinat lokasi pariwisata
6. Deskripsi, berisi tentang informasi-informasi dari tempat pariwisata yang telah dipilih
7. Tombol Rute adalah tombol perintah menampilkan lokasi serta rute menuju lokasi pariwisata.



Gambar 3.10 Antarmuka Halaman Rute

Gambar 3.10 adalah perancangan antarmuka halaman rute. Perancangan antarmuka ini terdapat beberapa bagian yaitu gambar Icon navigasi pariwisata dari aplikasi yang dibuat, gambar maps akan menampilkan lokasi pariwisata serta rute menuju lokasi pariwisata. Sedangkan keterangan dibawah maps adalah keterangan nama angkutan yang digunakan menuju lokasi serta keterangan trayek dan keterangan biaya.

## IMPLEMENTASI

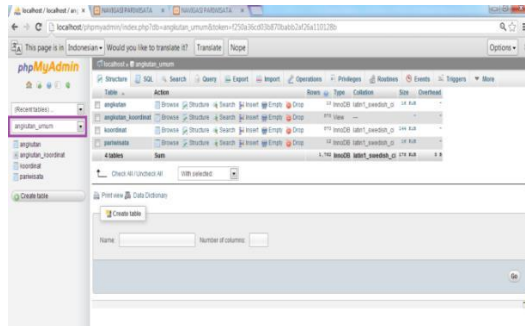
Pada bab IV ini, akan dijelaskan bagaimana langkah-langkah yang dilakukan untuk mengimplementasikan penelitian yang sedang dilakukan sesuai dengan perancangan sistem yang telah dijelaskan pada bab III.

### Konfigurasi Web Service Server

Pada sub-bab konfigurasi web service server, dijelaskan perancangan basis data yaitu membuat dan merancang kumpulan data pariwisata database pariwisata, database yang digunakan adalah mysql.

### Create Database Angkutan Umum

Database yang digunakan adalah mysql. Pada database mysql akan dibuat sebuah database yang menyimpan data Pariwisata di Surabaya

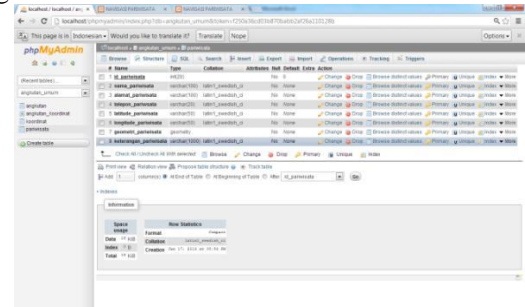


Gambar 4.1 Database angkutan\_umum

Pada Gambar 4.1, merupakan database yang dibuat dengan nama database angkutan\_umum (kotak ungu). Di dalam database angkutan\_umum terdapat tabel data pariwisata, yang diberi nama tabel pariwisata.

### Create Pariwisata

Implementasi tabel pariwisata berdasarkan struktur tabel pada tabel 4.1, pada database angkutan\_umum dapat dilihat pada gambar 4.2.



Gambar 4.2 Struktur Tabel Pariwisata

Struktur dari tabel pariwisata dapat dijelaskan, sebagai berikut :

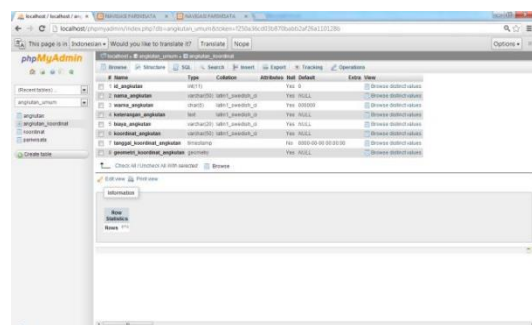
1. Id\_pariwisata, adalah primary key dari data pariwisata.
2. Nama\_pariwisata, adalah nama dari pariwisata.
3. Alamat\_pariwisata, yaitu penjelasan tentang alamat lokasi pariwisata berada.
4. Latitude\_pariwisata, merupakan titik koordinat lokasi pariwisata berdasarkan garis bujur (Y).
5. Longitude\_pariwisata, merupakan titik koordinat lokasi pariwisata berdasarkan garis lintang (X).
6. Lokasi\_pariwisata, merupakan gabungan dari

titik koordinat X dan Y dengan tipe data geometry.

keterangan\_pariwisata, yaitu keterangan tentang pariwisata tersebut, dapat berupa foto ataupun informasi lainnya.

### Create Angkutan Koordinat

Implementasi tabel angkutan\_koordinat berdasarkan struktur tabel pada tabel 4.2, pada database angkutan\_umum dapat dilihat pada gambar 4.3.



Gambar 4.3 Tabel Angkutan\_koordinat

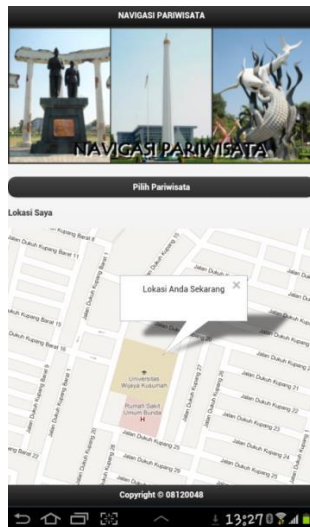
Struktur tabel Angkutan\_koordinat dapat dijelaskan, sebagai berikut :

1. Id\_angkutan, adalah primary key dari data angkutan.
2. Nama\_angkutan, adalah nama dari angkutan.
3. Warna\_angkutan, yaitu penjelasan tentang warna jalur angkutan.
4. Keterangan\_angkutan, yaitu penjelasan trayek angkutan.
5. Biaya\_angkutan, adalah penjelasan tentang biaya yang di dikeluarkan.
6. Koordinat\_angkutan, merupakan gabungan dari titik koordinat X dan Y.
7. Tanggal\_koordinat\_angkutan, merupakan penjelasan tentang tanggal koordinat.
- Geometry\_koordinat\_angkutan, angkutan, merupakan gabungan dari titik koordinat X dan Y dengan tipe data geometry

### Antarmuka Web Service Klien

Pada sub-bab konfigurasi web service pada klien, dijelaskan desain antarmuka yang telah direncanakan pada smartphone Android, sehingga klien sebagai pengguna bisa lebih mudah dalam pemahaman menu dan fungsi masing-masing modul ataupun button yang telah disediakan.

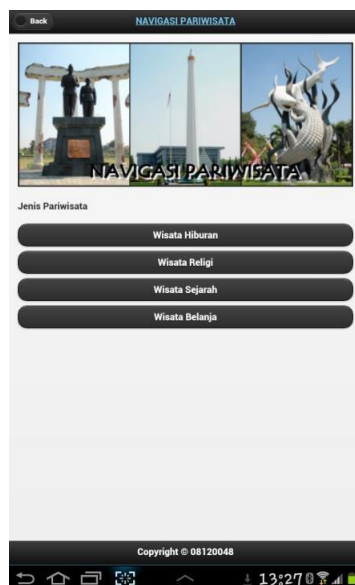
### Antarmuka Halaman Utama



Gambar 4.4 Antarmuka Halaman Utama Navigasi Pariwisata

Pada gambar 4.4 merupakan antarmuka halaman utama pada smartphone Android ketika membuka Aplikasi Navigasi Pariwisata. Tombol button pilih pariwisata pada gambar 4.4 Antarmuka Halaman utama berisi tentang kategori pariwisata. Pada halaman utama ini lokasi user telah terdeteksi oleh GPS dengan adanya pemberitahuan Lokasi Anda Sekarang.

### Antarmuka Jenis Pariwisata



Gambar 4.5 Antarmuka jenis pariwisata

Pada gambar 4.5 Merupakan Antarmuka jenis pariwisata. Yang mana terdapat tombol yang berisi jenis pariwisata.

1. Wisata Hiburan, berisi tentang wisata hiburan yang terdapat di Surabaya.
2. Wisata Religi, berisi tentang tempat religi di Surabaya.
3. Wisata Sejarah, berisi tentang tempat yang bersejarah yang terdapat di Surabaya.
4. Wisata Belanja, berisi tentang wisata belanja seperti mall yang terdapat di Surabaya.

### Antarmuka Pilih Pariwisata



Gambar 4.6 Antarmuka memilih tempat pariwisata

Pada gambar 4.6 Merupakan Antarmuka memilih tempat pariwisata. Yang mana terdapat tombol yang berisi tempat pariwisata yang terdapat Surabaya.

### Antarmuka Keterangan Tempat Pariwisata



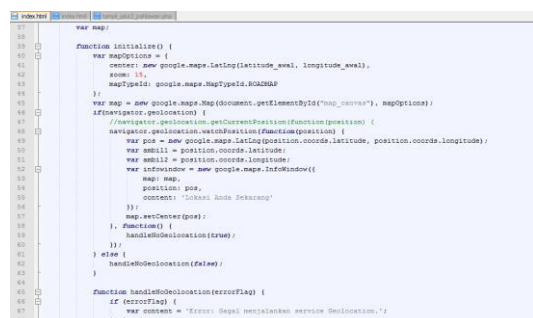
Gambar 4.6 Antarmuka Keterangan Tempat Pariwisata

Gambar 4.6 adalah Antarmuka Keterangan Tempat Pariwisata. Tampilan ini akan tampil setelah sudah memilih tempat pariwisata. Bagianannya terdiri dari :

1. Nama Pariwisata, berisi nama dari pariwisata yang telah dipilih
2. Alamat, berisi informasi alamat pariwisata
3. Nomer telepon, berisi tentang nomer telepon lokasi pariwisata
4. Latitude, berisi tentang posisi koordinat lokasi pariwisata
5. Longitude, berisi tentang posisi koordinat lokasi pariwisata
6. Deskripsi, berisi tentang informasi-informasi dari tempat pariwisata yang telah dipilih
7. Tombol Rute adalah tombol perintah
8. menampilkan lokasi serta rute menuju lokasi pariwisata.

### Konfigurasi GPS

Smartphone Android mempunyai fungsi sebagai GPS dimana dapat terkoneksi dengan satelit navigasi sehingga client dapat mengerti posisi keberadaannya.



Gambar 4.7 Source Code Konfigurasi GPS

Pada gambar 4.7 adalah kutipan source code untuk mendapatkan posisi koordinat latitude dan posisi koordinat longitude dari GPS android, dan kemudian posisi tersebut akan ditampilkan dalam bentuk peta digital menggunakan peta google.maps.

### Konfigurasi Data Pariwisata

Pada sub bab Konfigurasi Data Pariwisata, dijelaskan proses request, respons data Pariwisata. seperti yang telah dijelaskan sebelumnya pada gambar flowchart di bab III.

### Request dan Respons Data Rute Angkutan

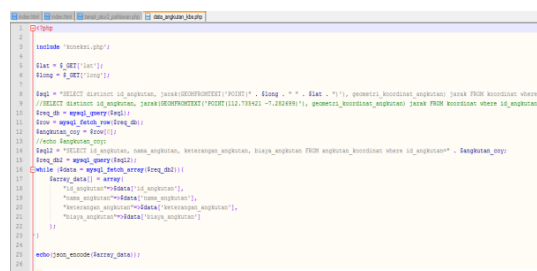
Untuk mencari rute angkutan umum menuju lokasi pariwisata, diperlukan konfigurasi request data pada web service klien. Di bawah ini adalah gambar 4.8 yaitu kutipan source code request

data rute angkutan pada web service klien.



Gambar 4.8 Source Code Request Rute Angkutan

Pada gambar 4.8 web service klient me request Rute Angkutan pada web service server (database) dengan alamat url [http://localhost/www/php/tampil\\_jalur2\\_kbs.php](http://localhost/www/php/tampil_jalur2_kbs.php). Untuk mendapatkan lokasi rute angkutan menuju lokasi pariwisata Kebun Binatang Surabaya. Gambar 4.9 adalah kutipan source code data angkutan Kebun Binatang Surabaya pada web service server.



Gambar 4.9 Select Data rute angkutan Kebun Binatang Surabaya

Web service server me-respons request data rute kebun binatang surabaya dengan meng-execute data rute kebun binatang yang ada pada database dengan menambahkan variabel koordinat longitude (\$long) dan latitude (\$lat) dari GPS (koordinat klien) dan menambahkan parameter jarak terdekat dengan posisi client. Setelah data rute kebun binatang diproses, web service server me-respons request dalam bentuk tipe data JSON.

### UJI COBA

Pada BAB V uji coba akan dilakukan pengujian supaya aplikasi yang telah dibuat bisa memenuhi tujuan pembuatan laporan tugas akhir ini. Skenario pengujian yang digunakan, akan disesuaikan dengan apa yang telah dijelaskan sebelumnya pada BAB III dan BAB IV.

### Pengujian Deteksi Lokasi Klien

Pada sub bab pengujian deteksi lokasi klien, akan menjelaskan mengenai tampilan yang



akan muncul pada awal halaman utama. Halaman utama tersebut akan menampilkan lokasi awal klien.



Gambar 5.1 Antarmuka Halaman Utama Navigasi Pariwisata

Pada gambar 5.1 merupakan antarmuka halaman utama pada smartphone Android ketika membuka Aplikasi Navigasi Pariwisata. Tombol button pilih pariwisata pada gambar 5.1 Antarmuka Halaman utama berisi tentang kategori pariwisata. Pada halaman utama ini lokasi user telah terdeteksi oleh GPS dengan adanya pemberitahuan Lokasi Anda Sekarang.

### Pengujian Memilih Pariwisata

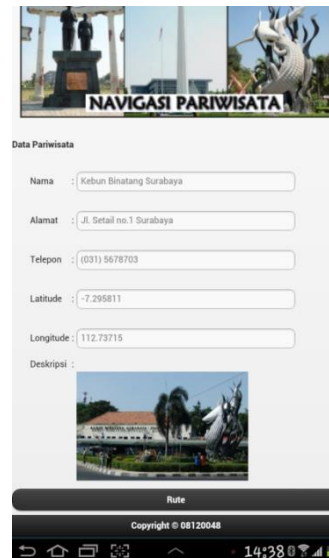
Pada sub bab pengujian memilih pariwisata, akan menjelaskan mengenai tampilan yang akan muncul pada menu pilih pariwisata.



Gambar 5.2 Antarmuka Halaman Pilih Pariwisata

Pada gambar 5.3 merupakan antarmuka halaman pilih pariwisata hiburan. Pada gambar 5.3

terdapat beberapa pilihan menu pariwisata. Diantarnya adalah menu pariwisata Kebun Binatang Surabaya, Pantai Ria Kenjeran, dan Taman Hiburan Remaja.

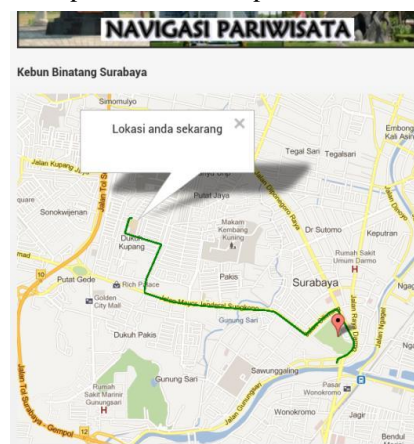


Gambar 5.3 Antarmuka Halaman Data Pariwisata Kebun Binatang

Pada gambar 5.3 merupakan antarmuka halaman data pariwisata. Pada menu ini menampilkan data dari pilihan pariwisata yang telah di pilih sebelumnya. Data yang muncul adalah Nama pariwisata, alamat, telepon, latitude, longitude maupun gambar dari lokasi pariwisata yang dipilih.

### Pengujian Menampilkan Hasil Rute Angkutan Umum

Pada sub bab pengujian menampilkan hasil rute angkutan umum, akan menjelaskan mengenai tampilan yang akan muncul setelah melakukan pemilihan lokasi pariwisata.



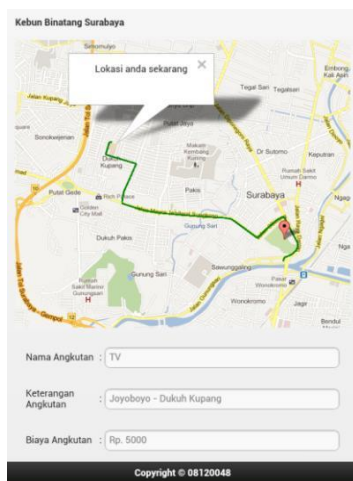
Gambar 5.4 Antarmuka Halaman Rute Pariwisata Kebun Binatang

Pada gambar 5.4 merupakan antarmuka

halaman rute pariwisata. Pada menu ini menampilkan rute pariwisata kebun binatang surabaya dari pilihan pariwisata kebun binatang yang telah di pilih sebelumnya. Map akan menampilkan rute angkutan kota yang dapat digunakan dari lokasi awal klien, menuju tempat tujuan lokasi yang akan telah ditentukan.

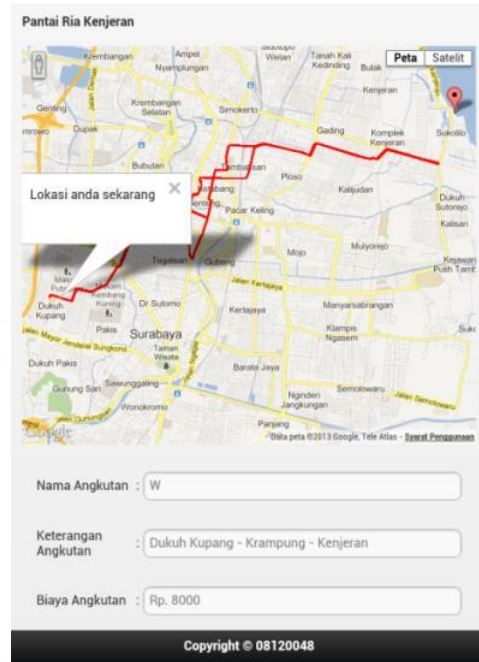
### Pengujian Biaya Angkutan

Pada sub bab pengujian biaya angkutan, akan menjelaskan mengenai tampilan yang akan muncul pada menu biaya angkutan.



Gambar 5.5 Antarmuka Halaman Rute Pariwisata Kebun Binatang

Pada gambar 5.5 merupakan antarmuka halaman rute angkutan pariwisata. Pada menu ini menampilkan rute dari pilihan pariwisata yang telah di pilih sebelumnya yaitu pariwisata kebun binatang. Map akan menampilkan rute angkutan kota yang dapat digunakan dari lokasi awal klien, menuju tempat tujuan lokasi yang akan telah ditentukan. Dalam hal ini tujuan telah ditentukan menuju pariwisata Kebun Binatang Surabaya. Biaya yang diperlukan untuk menggunakan angkutan kota yang telah tampil, juga akan muncul beserta nama angkutan yang dapat dipergunakan dan keterangan angkutan tersebut.



Gambar 5.5 Antarmuka Halaman Rute Pariwisata Pantai Ria Kenjeran

Pada gambar 5.5 merupakan antarmuka halaman rute angkutan pariwisata. Pada menu ini menampilkan rute dari pilihan pariwisata yang telah di pilih sebelumnya yaitu pariwisata kenjeran. Map akan menampilkan rute angkutan kota yang dapat digunakan dari lokasi awal klien, menuju tempat tujuan lokasi yang akan telah ditentukan. Dalam hal ini tujuan telah ditentukan menuju pariwisata Pantai Ria Kenjeran. Biaya yang diperlukan untuk menggunakan angkutan kota yang telah tampil, juga akan muncul beserta nama angkutan yang dapat dipergunakan dan keterangan angkutan tersebut.

### Simpulan

Berdasarkan uraian pembahasan analisa dan pengujian yang telah dilakukan, maka dapat diambil kesimpulan terhadap sistem yaitu Rancangan Sistem Rute Pariwisata Menggunakan Web Service Dengan Platform Android hanya bisa untuk menampilkan satu rute angkutan umum .

### Saran

Saran untuk perkembangan lebih lanjut dalam masa mendatang pada Rancangan Sistem Rute Pariwisata Menggunakan Web Service Dengan Platform Android dapat di kembangkan untuk menampilkan 2 rute angkutan umum atau lebih sistem ini juga dapat dikembangkan untuk pindah angkutan umum.



## DAFTAR PUSTAKA

- [1] Abay. (2011). Eclipse. <http://bayduaenam.blogspot.com/2011/06/eclipse.html> Diakses tanggal 21 Oktober 2012
- [2] Abidin, H.Z. 2007. Penentuan Posisi dengan GPS dan Aplikasinya. Jakarta : PT Pradnya Paramita.
- [3] Ardiansyah, M.Cs. 2011. *Mengenal Location Based Service*. URL: <http://blog.uad.ac.id/ardi/2011/04/04/mengenal-location-based-service> Diakses tanggal 16 Oktober 2012
- [4] Android SDK Developer Guide, <http://developer.android.com/guide/index.html>, Diakses tanggal 19 Oktober 2012
- [5] Author's Guide. *Android Developer Tools* URL: <http://developer.android.com/tools/help/adt.html> Diakses tanggal 16 Oktober 2012
- [6] Author's Guide. *GPS(Global Positioning System)* [http://en.wikipedia.org/wiki/Global\\_Positioning\\_System](http://en.wikipedia.org/wiki/Global_Positioning_System), Diakses tanggal 15 Oktober 2012
- [7] Author's Guide. *Web Service* <http://elib.unikom.ac.id/download.php?id=132941>. Diakses tanggal 15 Oktober 2012
- [8] Basofi, A, Fariza, A, Hardiansyah, M, Puspita Ayu, O, "Web Gis Untuk Informasi Pelayanan Umum Di Kotamadya Surabaya", Penerbit Proceeding Of The 9th Industrial Electronics Seminar 2007.
- [9] Budiyo, Eko , 2002, Sistem Informasi Geografis Menggunakan ArcView GIS, Andi , Yogyakarta.
- [10] Kelompok Keilmuan Geodesi Fakultas Ilmu dan Teknologi Kebumihan. Studi Ionosfer Menggunakan GPS, <URL: [http://geodesy.gd.itb.ac.id/?page\\_id=91](http://geodesy.gd.itb.ac.id/?page_id=91) >. Diakses tanggal 15 Oktober 2012
- [11] Muhamad Nur Taufiq, Nur Aini Rakhmawati, S.Kom, M.Sc.Eng, Radityo Prasetyanto Wibowo, S.Kom. *Penerapan Teknologi Mashup Pada Aplikasi Pariwisata Berbasis Web Nusantara View: Modul Akomodasi*. Surabaya: Paper ITS.
- [12] Safaat H, Nazruddin. 2011. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Informatika. Bandung
- [13] Saputro Haris, *Manajemen Database MySQL menggunakan Mysql Front*, elex Media komputindo, 2003, Jakarta.

{ Halaman sengaja dikosongkan }  
{ Jurnal **Melek IT** }

(6)  
**PENJUALAN TENDA ONLINE BERBASIS E-COMMERCE DI CV. HARAPAN JAYA  
TENDA SURABAYA**

**Kartika Dwi Jayantiningsih , Tjatusari W., S.Kom, M.Kom<sup>2</sup>**

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Wijaya Kusuma Surabaya

[lolycandy@gmail.com](mailto:lolycandy@gmail.com)

**ABSTRAK**

E-Commerce merupakan salah satu konsep yang cukup berkembang dalam dunia internet. Penggunaan sistem ini sebenarnya dapat menguntungkan banyak pihak, baik pihak konsumen, produsen maupun penjual (retail). Konsep online shopping menyediakan banyak kemudahan dan kelebihan jika dibandingkan dengan konsep belanja yang konvensional. Selain proses transaksi bisa menjadi lebih cepat, di internet telah disediakan hampir semua barang yang biasanya dijual secara lengkap.

Berdasarkan perkembangan teknologi khususnya teknologi internet, tugas akhir ini menerapkan sebuah konsep e-commerce pada penjualan tenda. Penerapan teknologi pada internet akan memberikan aksesibilitas dan fleksibilitas yang lebih, dalam melakukan transaksi maupun mendapatkan informasi mengenai tenda yang akan dipesan lebih cepat, praktis, familiar dan simple.

Penerapan teknologi ini diusulkan sebagai tindakan kongkret untuk melakukan perluasan pemasaran pebisnis pabrik tenda, dan menjadi sebuah keunggulan kompetitif yang sangat representatif karena sudah tidak waktunya lagi pebisnis pabrik tenda menunggu untuk mendapatkan konsumen, dan menaikkan omset pendapatan pabrik.

**Kata Kunci :** E-Commerce, Online Shopping, Teknologi, Transaksi, Konsumen.

**PENDAHULUAN**

**Latar Belakang**

*E-Commerce* merupakan salah satu konsep yang cukup berkembang dalam dunia internet. Penggunaan sistem ini sebenarnya dapat menguntungkan banyak pihak, baik pihak konsumen, produsen maupun penjual (retail). Konsep *online shopping* menyediakan banyak kemudahan dan kelebihan jika dibandingkan dengan konsep belanja yang konvensional. Selain proses transaksi bisa menjadi lebih cepat, di internet telah disediakan hampir semua barang yang biasanya dijual secara lengkap.

Di sisi lain, Bisnis tenda di Indonesia khususnya Jawa Timur pun telah berkembang cukup baik. Ditengah kebutuhan pasar yang meningkat CV. Harapan Jaya Tenda bersaing dengan lain CV lain untuk meningkatkan order pesanan pelanggan. Mobilitas masyarakat yang merupakan kandidat konsumen bisnis tenda. *website* yang hanya diakses melalui PC (*personal computer*) memberikan kemudahan dalam melakukan transaksi yang transparan sehingga produk dapat langsung tersampaikan ke pelanggan. Maka dari itu diharapkan sebuah solusi alternative yang memungkinkan melakukan transaksi ataupun

mengakses informasi tanpa tergantung pada sumber informasi dan lokasi akses sehingga memberikann layanan yang lebih fleksibel bagi konsumen. Sebuah penerapan teknologi yang melengkapi model yang telah ada dan diharapkan memberikan kemudahan yang mendorong terjadinya sebuah transaksi pada bisnis pabrik tenda.

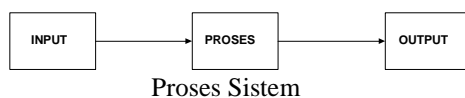
Berdasarkan perkembangan teknologi khususnya teknologi internet, tugas akhir ini menerapkan sebuah konsep *e-commerce* pada penjualan tenda. Penerapan teknologi pada internet akan memberikan aksesibilitas dan fleksibilitas yang lebih, dalam melakukan transaksi maupun mendapatkan informasi mengenai tenda yang akan dipesan lebih cepat, praktis, familiar dan simple.

Penerapan teknologi ini diusulkan sebagai tindakan kongkret untuk melakukan perluasan pemasaran pebisnis pabrik tenda, dan menjadi sebuah keunggulan kompetitif yang sangat representatif karena sudah tidak waktunya lagi pebisnis pabrik tenda menunggu untuk mendapatkan konsumen, dan menaikkan omset pendapatan pabrik.

## TINJAUAN PUSTAKA

### Pengertian Sistem

Pengertian Sistem Informasi diperoleh dari pemahaman tentang data dan informasi, karena data merupakan bentuk dasar dari informasi sedangkan informasi merupakan hasil dari pengolahan data. Sistem Informasi merupakan gabungan dari kata Sistem dan Informasi. Sistem berarti gabungan dari beberapa sub sistem dan informasi berarti data yang bermanfaat bagi penerima. Adapun yang dimaksud dengan data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata. Atau data adalah representasi dunia nyata yang mewakili suatu objek seperti manusia (pegawai, mahasiswa, pelanggan), hewan, peristiwa, konsep, keadaan dll. yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi atau kombinasinya. Informasi adalah data yang telah diproses menjadi bentuk yang memiliki arti bagi penerima dan dapat berupa fakta, suatu nilai yang bermanfaat. Jadi ada suatu proses transformasi data menjadi suatu informasi == *input* - proses – *output*.



### Internet Commerce

Internet commerce adalah penggunaan internet yang berbasis teknologi informasi dan komunikasi untuk perdagangan. Kegiatan komersial ini seperti iklan dalam penjualan produk dan jasa. Transaksi yang dapat dilakukan di internet antara lain pemesanan/pembelian barang dimana barang akan dikirim melalui pos atau sarana lain setelah uang ditransfer ke rekening penjual. Penggunaan internet sebagai media pemasaran dan saluran penjualan terbukti mempunyai keuntungan antara lain untuk beberapa produk tertentu lebih sesuai ditawarkan melalui internet; harga lebih murah mengingat membuat situs di internet lebih murah biayanya dibandingkan dengan membuka outlet retail di berbagai tempat, internet merupakan media promosi perusahaan dan produk yang paling tepat dengan harga yang relatif lebih murah, serta pembelian melalui internet akan diikuti dengan layanan pengantaran barang sampai di tempat pemesan.

### Toko Online

Toko online adalah toko yang dibuat dan dijalankan secara online dengan menggunakan media komputer dan koneksi internet itulah yang

paling tepat untuk menggambarkan apa arti dari toko online itu sendiri .

Dari segi bahasa, toko online berasal dari dua suku kata, Toko dan Online. Menurut Kamus Besar Bahasa Indonesia, toko berarti sebuah tempat atau bangunan permanen untuk menjual barang-barang (makanan, minuman, dan sebagainya). Sedangkan online yang terjemahan bahasa indonesianya adalah dalam jaringan atau disingkat daring menurut Wikipedia adalah keadaan di saat seseorang terhubung ke dalam suatu jaringan ataupun sistem yang lebih besar. Jadi berangkat dari dua pengertian secara bahasa tersebut kita dapat mengartikan toko online sebagai tempat terjadinya aktifitas perdagangan atau jual beli barang yang terhubung ke dalam suatu jaringan dalam hal ini jaringan internet.

Salah satu manfaat positif yang bisa dirasakan untuk perdagangan atau bisnis, membangun toko online. Ada keuntungan dan kurang toko online dibandingkan dengan toko offline atau toko tradisional. Antara lain :

#### 1. Modal Usaha

Bisa dibilang untuk memulai membuka toko online memerlukan modal banyak, hanya 1 set komputer + modem + internet, 1 buah camera digital, beberapa sample produk dan Paket hosting + domain saja. Tidak mencapai Rp. 10 juta untuk memulai usaha toko online, bayangkan jika kita membuka toko offline Rp. 10 juta mungkin hanya cukup untuk sewa tempat selama 1 tahun.

#### 2. Persediaan toko online tidak perlu membeli persediaan (zero stock). Artinya kita tidak perlu banyak mengeluarkan investasi / modal kerja baik untuk gudang, tempat display maupun membeli persediaan itu sendiri.

#### 3. Pasar yang luas

Jika ingin menembus dunia internetlah tempatnya. Karena dengan internet pesan kita bisa sampai ke pelosok dunia dimanapun itu berada.

#### 4. Kemudahan promosi

Melakukan bisnis saat ini mutlak perlu adanya promosi dan marketing. Dalam dunia internet tempat melakukan promosi sangatlah banyak, asal kita rajin browsing dan cari-cari informasi semua bisa kita dapat di internet

#### 5. Keluwesan waktu

Tidak perlu waktu khusus dalam membuka toko online. Seseorang pekerja juga bisa menyisihkan waktunya pada malam hari atau hari libur untuk membuka toko online, jadi tidak perlu mengganggu waktu kerjanya.

6. Tidak perlu tenaga kerja khusus  
Beda halnya toko offline yang perlu tenaga kerja untuk melayani pelanggan atau menunggu toko, di toko online kita tidak perlu membayar pekerja untuk itu. Cukup cantumkan no telepon, id yahoo messenger, alamat email dan alamat websitenya, kemudian tunggu saja sang calon pembeli yang tertarik dengan produk kita.

Selain kelebihan, pastinya toko online juga memiliki kekurangan. Kekurangan yang paling utama adalah kepercayaan pelanggan. Mengingat kepercayaan pelanggan sangatlah penting sehingga setiap pertanyaan yang datang harus dijawab dengan jelas sehingga tidak muncul keraguan bagi pelanggan.

### **Flowchart**

*Flowchart* adalah penggambaran secara grafik dari langkah-langkah dan urutan prosedur dari suatu program. *Flowchart* menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut.

## **ANALISA DAN PERANCANGAN**

Analisa dan Perancangan Sistem ini merupakan tahapan perencanaan dan perancangan desain pada sistem. Perencanaan yang dilakukan adalah membuat analisa sistem, analisa kebutuhan, pengguna sistem, prosedur pengguna sistem, dan perancangan sistem.

### **Analisa Sistem**

Pada pembahasan ini menjelaskan deskripsi tentang sistem penjualan tenda yang dihadapi oleh CV. Harapan Jaya baik segi penyampaian informasi produk, pemesanan barang, pembayaran barang sampai dengan proses laporan penjualan dari admin kepada pelanggan dan pemilik usaha. Transaksi – transaksi yang dapat dilakukan yaitu :

1. Informasi barang, yaitu pengolahan data category produk dan produk barang untuk admin dan informasi barang untuk pelanggan.
2. Pemesanan barang, yaitu proses transaksi order dan transaksi return barang yang dilakukan pelanggan
3. Tagihan pengiriman, yaitu proses aliran data berupa alamat pelanggan yang bertujuan lokasi pengiriman barang.
4. Pembayaran, yaitu proses aliran data pembayaran pelanggan dan konfirmasi pembayaran dari staff admin ke instansi.
5. Laporan, yaitu proses data laporan transaksi pelanggan dan history pemesanan dari staff admin ke pemilik usaha.

### **Kebutuhan Data**

Pada sistem penjualan tenda online berbasis *e-commerce* memiliki empat kebutuhan data sesuai dengan kebutuhan penggunaannya antara lain :

#### **A. Pelanggan**

1. Info spesifikasi barang
2. Order pelanggan
3. Registrasi pelanggan
4. Registrasi tagihan
5. Info alamat pembayaran
6. Registrasi pembayaran
7. Return
8. History pemesanan

#### **B. Staff Admin**

1. Data spesifikasi barang
2. Data stok barang
3. Info stok barang
4. Info registrasi pelanggan
5. Info order pelanggan
6. Info registrasi tagihan
7. Info registrasi pembayaran
8. Data konfirmasi pembayaran
9. Data alamat pembayaran
10. Info cross check
11. Data history pemesanan
12. Data laporan

#### **C. Instansi**

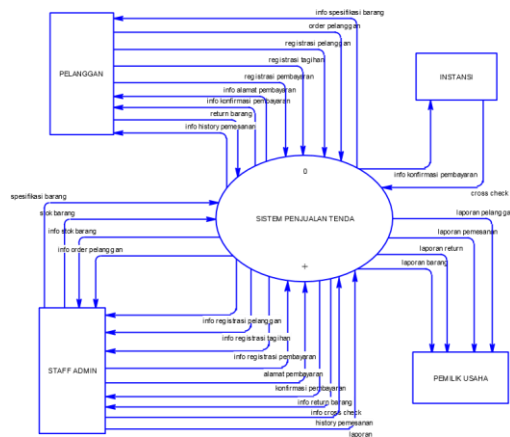
1. Info konfirmasi pembayaran
2. Data cross check

#### **D. Pemilik Usaha**

1. Laporan pelanggan
2. Laporan pemesanan
3. Laporan return
4. Laporan barang

### **DFD**

Sistem penjualan tenda diatas menjelaskan terdapat empat entitas yaitu, pelanggan, instansi, staff admin dan pemilik usaha dimana aliran data input dan output yang mengarah ke sistem saling bersangkutan.



Gambar Data Flow Diagram Level 0

Pada DFD level 1 terdapat lima proses yaitu, proses informasi barang, proses pemesanan barang, proses pembayaran, proses pengiriman, dan proses laporan penjualan dijelaskan sebagai berikut :

1. Proses Informasi Barang

Staff admin menginput data stok barang dan spesifikasi barang pada proses diolah menjadi informasi stok barang pada admin dan spesifikasi barang pada pelanggan.

2. Proses Pemesanan Barang

Pelanggan menginput data order pelanggan, registrasi pelanggan dan return barang pada proses pemesanan barang. Dari proses dioutputkan sebagai info order pelanggan, registrasi pelanggan dan return barang pada proses pemesanan barang.

3. Proses Tagihan Pengiriman

Pelanggan menginput data registrasi tagihan pada proses tagihan pengiriman. Dari proses diolah menjadi info tagihan kepada staff admin. Proses tersebut di simpan pada data store customer dan address.

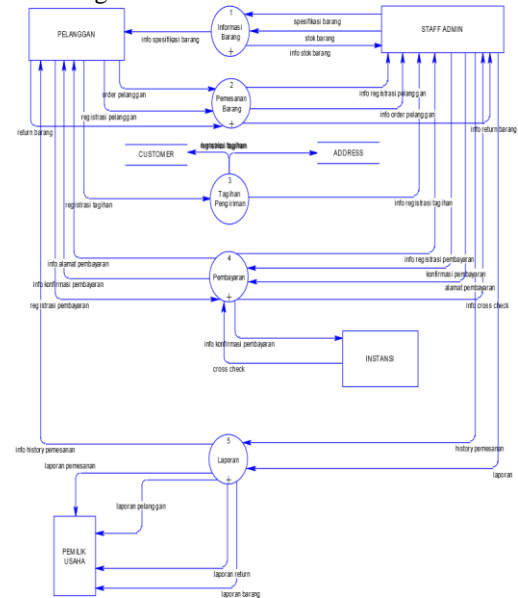
4. Proses pembayaran

Staff admin menginput data alamat pembayaran pada proses pembayaran dari proses di infokan kepada pelanggan. Pelanggan menginput data registrasi pembayaran dari proses diolah menjadi info registrasi pembayaran pada staff admin. Setelah itu, staff admin menginput data konfirmasi pembayaran pada proses pembayaran, dari proses di infokan kepada pelanggan dan instansi. Instansi melakukan cross check data lalu di infokan kembali kepada staff admin.

5. Proses Laporan

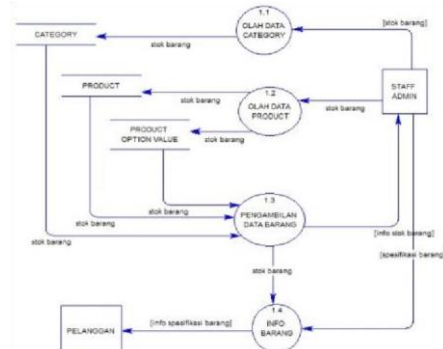
Staff admin menginput data history pemesanan pada sistem di infokan kepada pelanggan. Staff admin juga menginput data laporan pada proses di info kan menjadi laporan pelanggan,

laporan pemesanan, laporan return dan laporan barang.



Data Flow Diagram Level 1

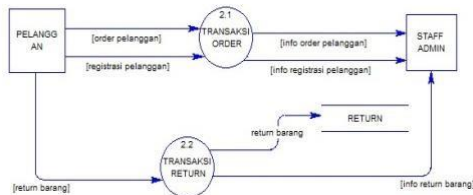
Pada DFD level 2 Informasi barang menjelaskan decompose dari proses informasi barang. Terdapat empat proses yaitu olah data category, olah data product, pengambilan data dan info barang. Pada proses olah data stok staff admin menginput data stok category dan product dari proses simpan pada data store category, product, dan product option value. Ketiga data store tersebut memberikan data stok barang pada proses pengambilan data barang lalu di informasikan kepada staff admin. Pada proses info barang staff admin menginput data spesifikasi barang lalu di informasikan kepada pelanggan.



Gambar DFD Level 2 Informasi Barang Pada DFD Level 2 pemesanan barang menjelaskan proses decompose dari proses pemesanan barang. Terdapat dua proses yaitu transaksi order dan transaksi return. Pada proses transaksi order pelanggan menginput data order pelanggan pada proses di infokan kepada staff admin. Pelanggan juga menginput data registrasi pelanggan dari

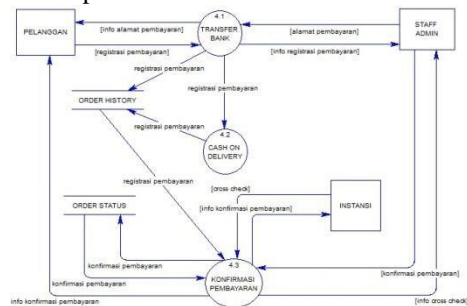


proses di infokan kepada staff admin. Pada proses transaksi return pelanggan menginput data return barang dari proses di infokan kepada staff admin. Pada proses transaksi return data disimpan pada data store return.



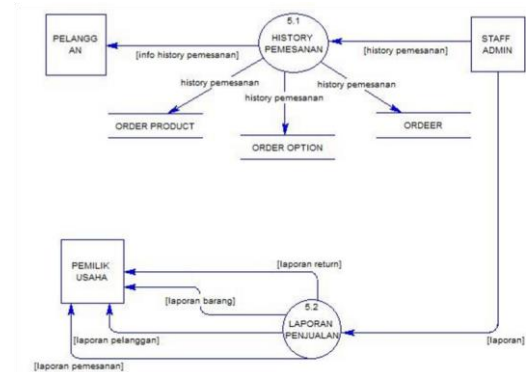
Gambar DFD Level 2 pemesanan barang

Pada Gambar DFD Level 2 pembayaran menjelaskan decompose dari proses pembayaran. Terdapat tiga proses yaitu proses transfer bank, cash on delivery dan konfirmasi pembayaran. Pada proses pembayaran staff admin menginput alamat pembayaran pada proses di infokan kepada pelanggan. Pelanggan melakukan input registrasi pembayaran transfer bank dan cash on delivery pada proses di infokan kepada staff admin selanjutnya registrasi tersebut disimpan pada data store order history. Pada proses konfirmasi pembayaran mengambil data registrasi pembayaran. Staff admin menginput konfirmasi pembayaran pada proses konfirmasi pembayaran lalu di simpan pada data store order status. Pada data store order status memberikan info pada proses konfirmasi pembayaran lalu di informasikan kepada pelanggan dan instansi. Instansi menginput proses cross check data dari proses di infokan kembali kepada staff admin.



Gambar DFD Level 2 pembayaran

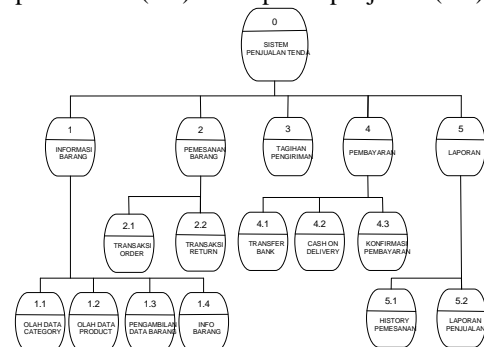
Pada DFD Level 2 Laporan menjelaskan decompose dari proses laporan. Terdapat dua proses yaitu proses history pemesanan dan laporan penjualan. Pada proses ini pada staff admin menginput data history pemesan dari proses di infokan kepada pelanggan. Proses history pemesanan disimpan pada data store order, order product dan order option. Pada proses laporan penjualan staff admin menginput data laporan. Dari proses di infokan kepada pemilik usaha berupa laporan pelanggan, laporan barang, laporan pemesanan dan laporan return.



Gambar DFD Level 2 Laporan

### Kebutuhan Proses

Pada Diagram Berjenjang Sistem Penjualan Tenda menjelaskan kebutuhan proses pada sistem penjualan tenda pada level 0 terdapat proses sistem informasi penjualan tenda(0). Pada level satu terdapat lima proses yaitu proses informasi barang(1), proses pemesanan barang(2), proses tagihan pengiriman(3), proses pembayaran(4), proses laporan penjualan(5). Pada level dua terdapat empat proses dari proses informasi barang yaitu olah data category(1.1), olah data product (1.2), pengambilan data barang (1.3) dan info barang(1.4). Pada level dua pemesanan barang yaitu transaksi order(2.1) dan transaksi return (2.2). Pada level dua proses pembayaran yaitu transfer bank (4.1), cash on delivery (4.2) dan konfirmasi pembayaran(4.3). Pada level dua proses pembayaran yaitu history pemesanan (5.1) dan laporan penjualan(5.2).



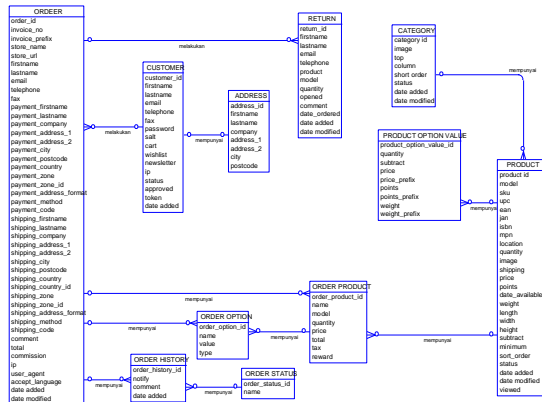
Gambar Diagram Berjenjang Sistem Penjualan Tenda

### Conceptual Data Model (CDM)

Pada CDM (*Conceptual Data Model*) menjelaskan sepuluh hubungan antar table yang saling berhubungan, yaitu :

1. Satu *costumer* mempunyai satu *address*.
2. Satu *costumer* melakukan banyak order.

3. Satu customer melakukan banyak return.
4. Satu order mempunyai banyak return, order *history*, order *option* dan order *product*.
5. Banyak order *history* mempunyai satu status order.
6. Satu *product* mempunyai banyak return.
7. Banyak *product* mempunyai satu *category* dan order *product*.
8. Satu *product* mempunyai banyak *product option value*.



Gambar CDM (Conceptual Data Model)

### Physical Data Model (PDM)

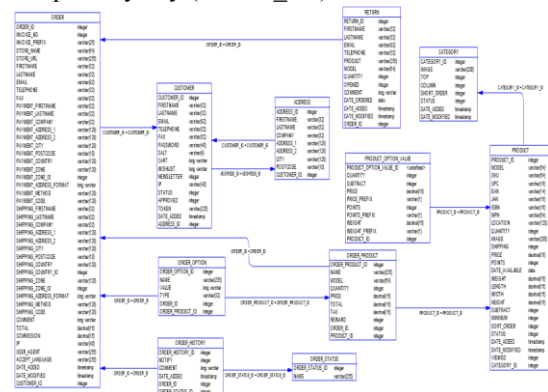
Pada PDM (*Physical Data Model*) menjelaskan bentuk representasi fisik dari tabel-tabel yang telah dirancang pada sistem penjualan tenda online. Di dalam PDM tidak hanya dijelaskan relasi yang menghubungkan antar tabel yang satu dengan yang lain, tetapi dijelaskan juga atribut-atribut yang menjadi penghubung antar tabel. Terbentuk pula relasi-relasi yang tidak hanya satu *foreign key* tetapi ada dua karena setiap atribut saling berhubungan satu sama lain. Pada table *customer*, atribut dari table *customer* dan address saling berhubungan dengan atribut address dan atribut barang yang membentuk dua foreign key (“*customer\_id*” dan “*address\_id*”) dan satu primary key (“*customer\_id*”) serta dua foreign key (“*customer\_id*” dan “*address\_id*”) dan satu primary key (“*address\_id*”).

Table order, atribut dari table order berhubungan dengan atribut customer yang membentuk satu foreign key (“*customer\_id*”) dan satu primary key (“*order\_id*”).

Table order option, atribut dari table order option berhubungan dengan atribut order product dan order yang membentuk dua foreign key (“*order\_product\_id*” dan “*order\_id*”) dan satu primary key (“*order\_option\_id*”).

Table order history, atribut dari table order history berhubungan dengan atribut order dan order status yang membentuk dua foreign key (“*order\_id*” dan “*order\_status\_id*”) dan satu primary key (“*order\_history\_id*”).

Table order product, atribut dari table order product berhubungan dengan atribut order dan product yang membentuk dua foreign key (“*order\_id*” dan “*product\_id*”) dan satu primary key (“*order\_product\_id*”). Table product option value, atribut dari table product option value berhubungan dengan atribut product yang membentuk satu foreign key (“*product\_id*”) dan satu primary key (“*product\_option\_value\_id*”). Table return, atribut dari table return berhubungan dengan atribut order, product, dan *customer* yang membentuk tiga foreign key (“*order\_id*”, “*product\_id*” dan “*customer\_id*”) dan satu primary key (“*return\_id*”).

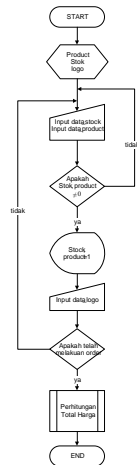


Gambar PDM (Physical Data Model)

### Desain Algoritma

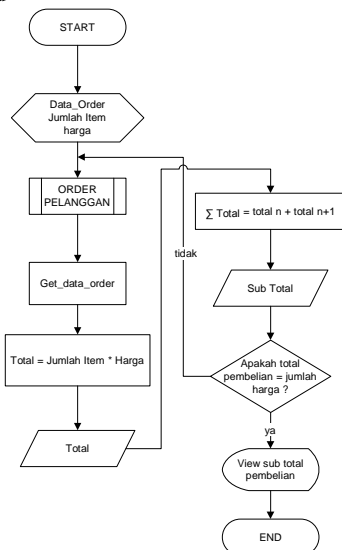
Keterangan Algoritma Order Pelanggan :

1. Start
2. Preparation data product, stock, logo
3. Input data\_stock , input data\_product
4. Decision apakah stock, product ≠ 0
5. Jika tidak, kembali ke proses input data data\_stock , input data\_product. Jika ya, display stock, product =1
6. Input data\_logo
7. Decision apakah telah melakukan order ?
8. Jika tidak, kembali ke Input data\_stock , input data\_product. Jika ya, proses perhitungan total harga
9. End



Gambar Algoritma Order Pelanggan  
Keterangan Algoritma Total Harga :

1. Start
2. Preparation data\_order, jumlah item, harga
3. Proses data order pelanggan
4. Proses get\_data\_order
5. Proses perhitungan (total=jumlah item\*harga)
6. Data total
7. Proses perhitungan sub total ( $\Sigma$  Total=total n + total n+1)
8. Data sub total
9. Decision “apakah total pembelian = jumlah harga ?”
10. Display sub total pembelian
11. End



Gambar Algoritma Total Harga

## IMPLEMENTASI PERANGKAT LUNAK

Perangkat keras komputer atau hardware sangat mendukung dalam kinerja sistem. Semakin baik spesifikasi perangkat keras yang ada maka

kinerja sistem pun akan semakin baik. Spesifikasi perangkat keras yang digunakan adalah sebagai berikut :

1. Processor : AMD Turion (tm) X2 DualCore Mobile RM-70 (2CPUs)
2. RAM : 1536 MB
3. VGA : NVIDIA GEFORCE 9100 M G
4. Harddisk : 160 GB
5. Keyboard dan mouse

## Program File

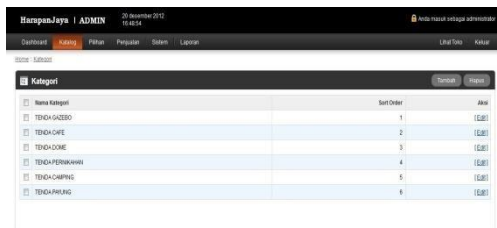
Pada perancangan Sistem Penjualan Tenda Online Berbasis *E-Commerce* didapat file program dari software Adobe Macromedia CS3, diantaranya sebagai berikut :

1. *Costumer.php* dan *Customer\_group.php*  
Halaman yang berfungsi untuk mengolah data *costumer* baru dan member.
2. *Category.php*  
Adalah halaman yang berfungsi untuk mengolah data kategori produk.
3. *Product.php*  
Adalah halaman yang berfungsi untuk mengolah data produk barang.
4. *Cart.php* dan *checkout.php*  
Halaman yang berfungsi untuk menampilkan transaksi barang yang sudah dibeli oleh pelanggan.
5. *Order.php*  
Halaman yang berfungsi untuk menampilkan *histori* data dan transaksi pelanggan.
6. *Return.php*  
Halaman yang berfungsi untuk mengolah data return barang.
7. *Sale.php*  
Halaman yang berisi tentang rekapan laporan data pelanggan dan transaksi pembelian barang.

## Uji Coba

Berikut merupakan uji coba pengolahan data category yang dilakukan oleh admin. Pada proses *insert* terdapat beberapa form yang berisi informasi barang berdasarkan sub category pada bagian category produk. Beberapa diantaranya adalah form nama category dan upload gambar. Form tersebut bertujuan agar pelanggan dapat mengetahui informasi barang yang dijual oleh perusahaan. Pada proses *update*, admin mengedit data yang sebelumnya telah *terinsert*. Admin dapat merubah data informasi dan gambar. Pada proses

delete, admin dapat menghapus data yang tidak terpakai. Hasil olah data category



Nama Kategori	Start Order	Aktif
TENDA-GAZERO	1	<input type="checkbox"/>
TENDA-CAFE	2	<input type="checkbox"/>
TENDA-DOME	3	<input type="checkbox"/>
TENDA-PERKAWINAN	4	<input type="checkbox"/>
TENDA-CAMPING	5	<input type="checkbox"/>
TENDA-RIVERO	6	<input type="checkbox"/>

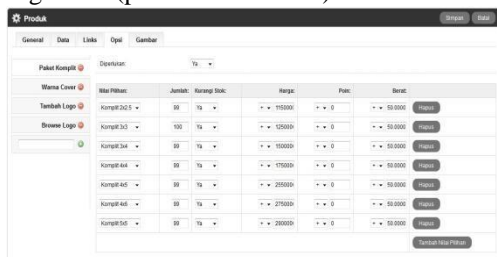
Gambar Hasil Olah Data Kategori Produk

Berikut merupakan uji coba pengolahan data produk barang yang dilakukan oleh admin. Pada proses *insert* terdapat beberapa form yang berisi informasi barang berdasarkan sub category pada bagian produk barang. Beberapa diantaranya adalah form nama produk, model, spesifikasi, harga dan upload gambar. Form tersebut bertujuan agar pelanggan dapat mengetahui informasi barang berdasarkan spesifikasi dan harga sesuai dengan kebutuhan pelanggan.

Pada sub opsi terdapat penggolongan harga berdasarkan ukuran, spesifikasi produk, rangka dan cover.

Pada gambar Tampilan Opsi Pada Produk didapat keterangan diantaranya :

- Pada pilihan paket komplit terdapat nilai pilihan spesifikasi ukuran barang . Pada nilai pilihan terdapat ukuran 2x2.5, komplit 3x3, komplit 3x4, komplit 4x4, komplit 4x5, komplit 4x6, dan komplit 5x5. Pada jumlah terdapat stok dari masing – masing paket. Pada harga terdapat nominal harga barang. Dan pada berat berdasarkan jumlah akumulasi kilogram (Kg) dari tiap ukuran.
- Pada warna cover terdapat nilai pilihan warna sesuai spesifikasi dari tiap warna cover.
- Pada tambah logo, terdapat nilai pilihan logo dan harga senilai Rp. 200.000.
- Pada Browse logo terdapat panel browse file gambar (pada halaman user).



Paket Komplit	Warna Cover	Jumlah	Ruang Toko	Harga	Paket	Berat
Komplit 2x2.5	Putih	99	1x	110000	1	50.000
Komplit 3x3	Putih	100	1x	120000	1	50.000
Komplit 3x4	Putih	99	1x	130000	1	50.000
Komplit 4x4	Putih	99	1x	140000	1	50.000
Komplit 4x5	Putih	99	1x	150000	1	50.000
Komplit 4x6	Putih	99	1x	160000	1	50.000
Komplit 5x5	Putih	99	1x	170000	1	50.000

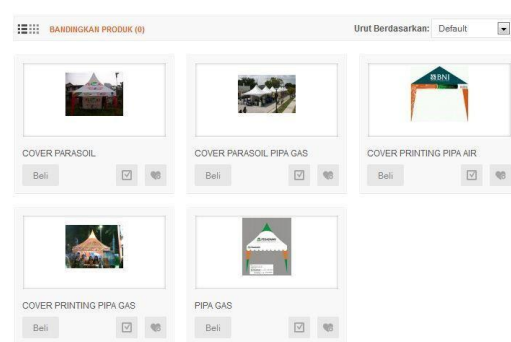
Gambar Tampilan Opsi Pada Produk

Pada halaman ini data category produk dan produk barang yang sebelumnya diolah oleh admin ditampilkan pada halaman user yaitu halaman yang

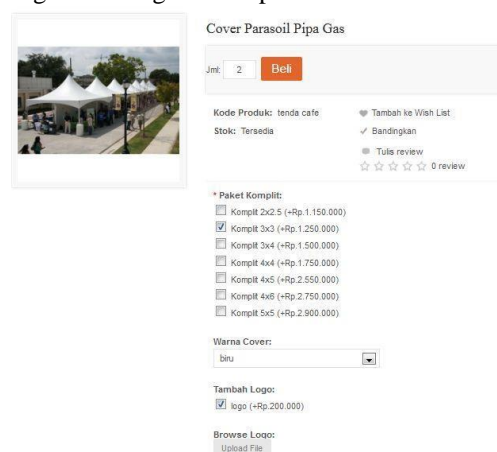
dapat diakses oleh pelanggan. Uji coba pada halaman user dilakukan pada kategori tenda café.



Gambar Hasil Tampilan Kategori Produk Pada Hasil Tampilan Kategori Produk menampilkan informasi produk tenda café. Pelanggan dapat mengetahui segala fungsi dan informasi mengenai tenda café.



Gambar Hasil Tampilan Produk Barang Gambar Hasil Tampilan Produk Barang menampilkan produk barang dari tenda café. Terdapat lima produk berdasarkan spesifikasinya. Terdapat button beli yang berfungsi agar pelanggan dapat melakukan transaksi pembelian sesuai dengan ukuran dan spesifikasinya. Eksekusi dari button beli dimaksudkan agar pelanggan dapat mengetahui harga dari tiap ukuran.



Gambar Tampilan Pembelian Barang

Sebagai contoh kasus pelanggan membeli dua unit tenda café dengan spesifikasi cover parasoil. Terdapat gambar (\*) pada paket pilihan,

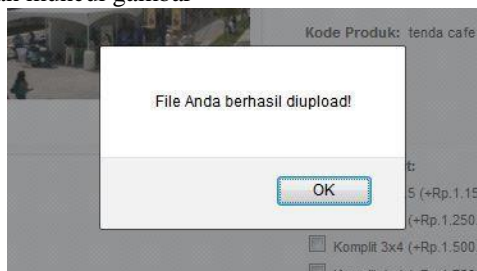
dimaksudkan pelanggan harus memilih salah satu dari ukuran paket komplit. Komplit 3x3 (+Rp. 1.250.000) menjelaskan pelanggan memilih paket komplit dengan ukuran 3x3 meter seharga Rp. 1.250.000.

Pelanggan dapat memilih warna cover sesuai dengan kebutuhan. Pada List Pilihan Warna menjelaskan terdapat tiga pilihan warna yaitu merah, kuning dan biru. Pada contoh kasus ini pelanggan memilih warna biru. Selain itu pelanggan juga mencentang pada tambah logo artinya pelanggan menambah logo / gambar pada cover tenda. Logo (+Rp.200.000) menjelaskan harga dari tambah logo seharga Rp. 200.000.



Gambar List Pilihan Warna

Untuk menentukan gambar logo pelanggan melakukan aksi upload file. Aksi ini dimaksudkan pelanggan dapat mengambil file gambar pada computer. Jika sudah melakukan proses upload akan muncul gambar



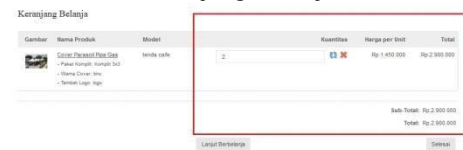
Gambar Tampilan Hasil Upload

Halaman ini sistem melakukan aksi kalkulasi keranjang belanja berdasarkan jumlah item dikalikan dengan harga. Pada contoh kasus sebelumnya pelanggan membeli dua item tenda model tenda café ukuran 3x3 meter seharga Rp. 1.250.000 dengan nama produk cover parasol pipa gas, warna biru dan tambah logo. Tertampil pada gambar Tampilan Keranjang Belanja



Gambar Tampilan Keranjang Belanja Proses akumulasi harga tenda dengan harga per unit Rp. 1.250.000 ukuran 3x3 meter tambah logo Rp. 200.000 menjadi Rp. 1.450.000 harga per unit. Pelanggan membeli 2 unit tenda menjadi 2 unit X

Rp. 1.450.000 menjadi Rp. 2.900.000 harga total. Tertampil pada gambar Hasil Kalkulasi Keranjang Belanja



Gambar Hasil Kalkulasi Keranjang Belanja Halaman ini adalah penyelesaian dari proses kalkulasi keranjang belanja. Terdapat dua pilihan, jika pelanggan belum memiliki akun maka pelanggan harus mengisi form pelanggan baru.

Gambar Form Pelanggan Baru

Dan jika pelanggan sudah memiliki akun pelanggan hanya mengisi e-mail dan password untuk login.

Gambar Form Member

Setelah mengisi data pelanggan proses selanjutnya detail penagihan. Pada aksi ini terdapat dua alternative tagihan yaitu sesuai alamat dengan akun atau penagihan dengan alamat lain

Gambar Tampilan Detail Tagihan Sesuai Akun



Gambar Tampilan Detail Tagihan Alamat lain

Langkah selanjutnya adalah memilih metode pembayaran pada contoh kasus ini pelanggan memilih Transfer Bank dan menambahkan komentar “TOLONG CEPAT BANG !!!”.

Gambar Tampilan Metode Pembayaran

Langkah berikutnya adalah konfirmasi pemesanan. Disini menampilkan rangkuman dari pesanan pelanggan antara lain : nama produk, model, kuantitas, harga dan total pembayaran. Serta instansi bank yang dituju.

Gambar Tampilan Konfirmasi Pemesanan

Tampilan akhir jika pelanggan sudah melakukan semua tahap maka akan tampil Sukses.

Gambar Tampilan Sukses Setelah Check Out

Pada halaman ini admin memiliki hak akses dalam mengolah histori pemesanan. Pada halaman ini admin dapat mengetahui produk barang, spesifikasi dan jumlah item yang dibeli oleh pelanggan. Seperti pada gambar Histori Pemesanan Hak akses ini admin mengubah status

pemesanan tersebut sudah pada tahap. Terdapat lima tahap status pemesanan yaitu :

- Pending, status dimana pelanggan baru melakukan pemesanan.
- Dalam proses, status sedang dalam proses pengiriman dan admin akan mencantumkan no resi sesuai dengan biro perjalanan yang dituju.
- Sedang diproses, status sedang dalam tahap produksi.
- Dibatalkan, proses dimana barang tersebut dicancel oleh pihak pelanggan.
- Complete, proses telah melakukan pengiriman.

Admin dapat melakukan pencetakan faktur pada data histori sebagai laporan untuk pemilik usaha.

Gambar Histori Pemesanan

Hasil laporan dari seluruh penjualan dapat dilihat pada halaman admin. Dan selanjutnya diserahkan pada pemilik usaha.

Gambar Hasil Laporan Pesanan

Gambar Hasil Laporan Return

Gambar Hasil Laporan Pelanggan

Gambar Hasil Laporan Barang



## KESIMPULAN

Setelah dilakukan pengujian dan analisa program, maka dapat diperoleh kesimpulan sebagai berikut :

1. Sistem dapat melakukan olah data dan menampilkan barang sesuai dengan spesifikasi barang dan ukuran.
2. Sistem dapat melakukan kalkulasi harga pembelian pelanggan.
3. Sistem dapat menampilkan history pelanggan, sehingga admin dapat dengan mudah memonitoring pesanan dari pelanggan.
4. Sistem dapat menampilkan laporan penjualan, sehingga memudahkan admin dan pemilik usaha dalam memonitoring penjualan barang.

## DAFTAR PUSTAKA

- [1] Jogiyanto. 1995. *Analisa dan Perancangan Sistem Informasi*. Andy Offset. Yogyakarta
- [2] Kadir, Abdul. 2009. *Dasar Perancangan dan Implementasi Database Relational*. Andi. Yogyakarta.
- [3] Kadir, Abdul. 2004. *Dasar Aplikasi Database MySQL*. Andi. Yogyakarta.
- [4] Kristianto, Andri. 2008. *Perancangan Sistem Informasi dan aplikasinya, Edisi Revisi*. Gaya Media. Yogyakarta.
- [5] Fatansyah, Ir. 2004. *Basis Data*. Bandung : Informatika.
- [6] Jogiyanto. 2004, *Analisis dan Desain Sistem Informasi*, Bandung : Penerbit Informatika.

{Halaman sengaja dikosongkan}  
{Jurnal **Melek IT**}

(7)

## SISTEM INFORMASI CUTI PEGAWAI PADA PT. PELABUHAN INDONESIA III (PERSERO) SURABAYA

Rizal Fathony , Tjatusari .W<sup>2</sup>, Guendra Kusuma .W<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Wijaya Kusuma Surabaya

### ABSTRAK

*Cuti dimaksudkan untuk memberikan istirahat kepada karyawan sebagai usaha pemulihan kesegaran jasmani dan rohani yang bertujuan untuk meningkatkan produktifitas kerja kepada pegawai selama jangka waktu tertentu. Dalam pengolahan data cuti saat ini terdapat masalah apabila ada seorang karyawan yang ingin mengajukan cuti, bagian SDM harus memeriksa terlebih dahulu data cuti karyawan dan juga mengetahui sisa cuti yang belum diambil oleh karyawan. Hal ini tentu saja memerlukan waktu pencarian yang lebih lama dalam penentuan izin pengajuan cuti. Berdasarkan latar belakang tersebut, penelitian ini bertujuan untuk membangun perangkat lunak yang dapat mengelola data cuti pegawai dan pencarian terhadap data pegawai yang telah mengambil cuti serta data-data yang berhubungan dengan cuti pegawai. Pembuatan aplikasi ini menggunakan bahasa pemrograman PHP dan MySQL sebagai basis data.*

**Kata kunci :** cuti pegawai, sistem cuti, sistem informasi, sistem informasi cuti pegawai,

### PENDAHULUAN

#### Latar Belakang

Cuti adalah keadaan tidak masuk kerja yang diijinkan dalam jangka waktu tertentu. Cuti merupakan hak semua pegawai, pelaksanaan cuti hanya dapat ditunda dalam jangka waktu tertentu apabila terdapat kepentingan yang mendesak. Cuti merupakan salah satu bagian dari lingkup Manajemen waktu. Manajemen waktu adalah pemanfaatan waktu karyawan untuk melaksanakan kegiatan yang berkaitan dengan kepentingan PT. Pelabuhan Indonesia III (Persero) Surabaya.

Pemanfaatan hak cuti harus sesuai dengan kebijakan atau peraturan perundangan yang berlaku di PT. Pelabuhan Indonesia III (Persero) Surabaya. Sehingga PT. Pelabuhan Indonesia III (Persero) Surabaya mampu mencapai target – targetnya. Dalam pengolahan data cuti saat ini terdapat masalah apabila ada seorang karyawan yang ingin mengajukan cuti, bagian SDM harus memeriksa terlebih dahulu data cuti karyawan dan juga mengetahui sisa cuti yang belum diambil oleh karyawan. Hal ini tentu saja memerlukan waktu pencarian yang lebih lama dalam penentuan izin pengajuan cuti.

Penulis bermaksud membuat suatu sistem untuk para karyawan di PT. Pelindo III (Persero) Surabaya, yaitu mengelola data cuti pegawai dan pencarian terhadap data pegawai yang telah mengambil cuti serta data-data yang berhubungan dengan cuti pegawai. Selain itu juga, sistem informasi ini dapat mengurangi beban tugas Direktorat Personalia dan Umum (Dirpum). Penulis tertarik untuk mencoba menentengahkan

permasalahan tersebut dalam sebuah penelitian yang bertemakan : “**SISTEM INFORMASI CUTI PEGAWAI PADA PT. Pelabuhan Indonesia III (Persero) SURABAYA BERBASIS WEB**”.

### TINJAUAN PUSTAKA

#### Pengertian Sistem

Sistem berkembang sesuai dengan konteks di mana pengertian sistem itu digunakan. Berikut akan diberikan beberapa definisi sistem secara umum:

Sekumpulan *object-object* yang saling berelasi dan berinteraksi serta hubungan antar objek bisa dilihat sebagai satu kesatuan yang dirancang untuk mencapai satu tujuan.

Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variable-variable yang saling terorganisasi, saling berinteraksi, dan saling bergantung sama lain.

Banyak ahli mengajukan konsep sistem dengan deskripsi yang berbeda, tetapi pada prinsipnya hampir sama dengan konsep dasar sistem umumnya. *Schronderberg* (1971) dalam *Suradinata* (1996) secara ringkas menjelaskan bahwa sistem adalah:

- Komponen-komponen yang saling berhubungan satu sama lain.
- Suatu keseluruhan tanpa memisahkan komponen pembentuknya.
- Bersama-sama dalam mencapai tujuan.
- Memiliki input dan output yang dibutuhkan oleh sistem lainnya.
- Terdapat proses yang mengubah input menjadi output.

- Memiliki aturan.
- Memiliki sub sistem yang lebih kecil.
- Memiliki deferensiasi antar subsistem.
- Memiliki tujuan yang sama meskipun mulainya berbeda.

### Pengertian Cuti

Cuti adalah keadaan tidak masuk kerja yang diijinkan dalam jangka waktu tertentu. Cuti merupakan hak semua pegawai, oleh sebab itu pelaksanaan cuti hanya dapat ditunda dalam jangka waktu tertentu apabila terdapat kepentingan yang mendesak. Cuti merupakan salah satu bagian dari lingkup Manajemen waktu. Yang dimaksud dengan manajemen waktu adalah pemanfaatan waktu yang dilakukan karyawan sebaik mungkin untuk melaksanakan kegiatan yang berkaitan dengan kepentingan perusahaan sehingga dicapai sinergi yang optimal antara kepentingan perusahaan dan pegawai.

### Aturan Cuti Pegawai PT. Pelindo III (Persero)

Berdasarkan pasal 30 tentang cuti pegawai pada buku Perjanjian Kerja Bersama (PKB) antara PT. Pelabuhan Indonesia III (Persero) Surabaya dengan Serikat Pekerja Pelabuhan Indonesia III, yang berisi.

#### Pasal 30

##### Cuti Pegawai

- (1) Kepada pegawai yang bekerja secara terus menerus dalam jangka waktu tertentu berhak atas cuti
- (2) Jenis – jenis cuti terdiri dari :
  - a. Cuti tahunan, yaitu cuti yang diberikan kepada pegawai yang telah bekerja 12 (dua belas) bulan terus menerus berhak atas cuti tahunan paling lama 12 (dua belas) hari kerja;
  - b. Cuti besar, yaitu cuti yang diberikan kepada pegawai yang telah bekerja paling sedikit 6 (enam) tahun terus menerus berhak atas cuti besar paling lama 3 (tiga) bulan;
  - c. Cuti sakit, yaitu cuti yang diberikan kepada pegawai yang menderita sakit lebih dari 2 (dua) hari kerja berhak atas cuti sakit paling lama 1 (satu) tahun;
  - d. Cuti bersalin, yaitu cuti yang diberikan kepada pegawai untuk melakukan persalinan berhak atas cuti bersalin selama paling lama 3 (tiga) bulan dengan ketentuan pelaksanaan 1 bulan sebelum dan 2 bulan setelah kelahiran;
  - e. Cuti karena alasan penting, yaitu cuti yang diberikan kepada pegawai karena keluarganya (istri/suami, anak, orang tua, kakak/adik kandung, mertua/menantu) sakit keras atau meninggal dunia selama paling lama 12 (dua belas) hari kerja;

- f. Cuti diluar tanggungan Perusahaan, yaitu cuti yang diberikan kepada pegawai yang telah bekerja paling sedikit 5 (lima) tahun secara terus menerus dan hanya diberikan bagi pegawai yang melaksanakan pendidikan tanpa ikatan dinas maksimal 3 tahun.
- (3) Pegawai yang mengambil cuti tahunan diberikan bantuan uang cuti paling sedikit sebesar 1 (satu) kali penghasilan ditambah tunjangan jabatan structural/ fungsional.
  - (4) Bagi pegawai yang tidak sempat mengambil cuti tahunan dalam tahun berjalan karena mengambil masa persiapan pension dapat diberikan uang cuti tahunan.
  - (5) Pegawai yang mengambil cuti besar tidak berhak mengambil cuti tahunan tetapi dapat diberikan uang cuti besar sebesar 2 (dua) kali penghasilan ditambah tunjangan jabatan structural/ fungsional.
  - (6) Ketentuan pelaksanaan cuti pegawai pada pasal ini, lebih lanjut akan diatur dan ditetapkan dengan peraturan pengusaha.

### Prosedur Cuti

Prosedur untuk pengajuan cuti di PT. Pelabuhan Indonesia III (Persero) Surabaya adalah sebagai berikut:

1. Pegawai mengisi form cuti.
2. Pegawai menyerahkan form cuti yang telah diisi kepada Senior Manajernya.
3. Senior manajer meng-acc form cuti dengan mempertimbangkan alasan cuti dan tanggal cuti yang diajukan.
4. Jika Senior Manajer menolak pegawai diminta untuk mengajukan cuti baru dengan tanggal yang telah diubah.
5. Jika Senior Manajer menyetujui (dengan menandatangani form) selanjutnya pegawai menyerahkan form tersebut kepada Dirpum.
6. Dirpum meng-acc form cuti tersebut.
7. Dirpum mengecek kelayakan cuti yang diajukan oleh pegawai dengan mempertimbangkan jatah cuti yang dapat diambil, jumlah pemohon cuti pada subdit yang sama jika melebihi cuti harus ditunda terlebih dahulu, masa kerja pegawai, dan tanggal cuti yang tidak boleh diambil cuti.
8. Dirpum menolak cuti jika cuti yang diajukan tidak layak, Dirpum menyetujui cuti jika cuti telah memenuhi syarat.

### Bisnis Rule

*Business Rules* merupakan aturan – aturan yang berhubungan dengan interaksi dan suatu kejadian dari sistem yang akan dikerjakan.

1. Setiap pengguna dalam sistem adalah pegawai PT. Pelabuhan Indonesia III (Persero) Surabaya

dengan hak akses masing – masing ditentukan oleh jabatan.

2. Setiap data yang di-input dalam sistem harus valid.
3. Setiap pengguna dalam sistem diberikan *username* dan *password*, yang dapat diubah sendiri oleh pengguna.
4. Setiap pengguna sistem menerima “warning message sisa cuti Tahunan” setelah berhasil login ke dalam sistem.
5. Setiap pengguna dapat melihat kalender event yang diterbitkan oleh admin sebagai acuan untuk cuti.
6. Pegawai dapat melihat history cuti. Untuk melakukan penundaan cuti dan melihat status permohonan cuti.
7. Untuk memperpanjang masa cuti, pegawai harus membuat lagi permohonan cuti.
8. Cuti Tahunan tidak bisa melebihi dari 12 hari dalam setahun. Pegawai tidak bisa menambah masa cuti tahunan jika sudah melewati batas cuti.
9. Untuk jenis cuti sakit, cuti bersalin, dan cuti karena alasan penting bisa diambil tanpa melihat masa kerja.
10. Senior Manajer dapat melihat history cuti yang berisi list permohonan cuti dari anggota subditnya (untuk melakukan proses acc) dan list untuk data cuti-nya sendiri (untuk melakukan penundaan dan melihat status cutinya).
11. Senior Manajer dapat melihat status anak buahnya apakah sedang cuti atau tidak.
12. Admin dapat melihat history cuti yang berisi list permohonan cuti yang telah mendapat persetujuan dari SM subdit (untuk melakukan proses acc) dan list untuk data cuti-nya sendiri (untuk melakukan penundaan dan melihat status cutinya).
13. Admin dapat mencetak SK cuti.
14. Admin dapat membuat/mengedit kalender event.

### Pengertian Unified Modeling Language (UML)

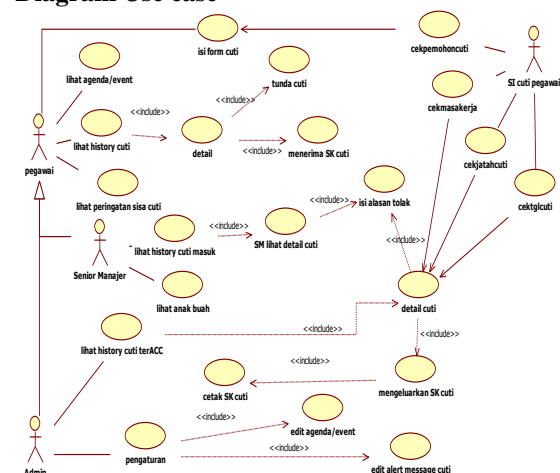
UML (Unified Modeling Language) adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan artifact (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. Artifact dapat berupa model, deskripsi atau perangkat lunak) dari system perangkat lunak. UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan.

## ANALISA DAN PERANCANGAN

*Analisa sistem* merupakan kegiatan penguraian suatu sistem informasi yang utuh dan nyata ke dalam bagian-bagian atau komponen-komponen komputer yang bertujuan untuk mengidentifikasi serta mengevaluasi masalah-masalah yang muncul, hambatan-hambatan yang mungkin terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga mengarah kepada suatu solusi untuk perbaikan maupun pengembangan ke arah yang lebih baik dan sesuai dengan kebutuhan serta perkembangan teknologi.

Sistem Informasi Cuti Pegawai pada PT. Pelabuhan Indonesia III (Persero) Surabaya mengelola data cuti pegawai dan pencarian terhadap data pegawai yang telah mengambil cuti serta data-data yang berhubungan dengan cuti pegawai. Analisa sistem dalam pembuatan Sistem Informasi Cuti Pegawai pada PT. Pelabuhan Indonesia III (Persero) Surabaya menggunakan *Use case* dan *Sekuensial diagram*:

### Diagram Use case



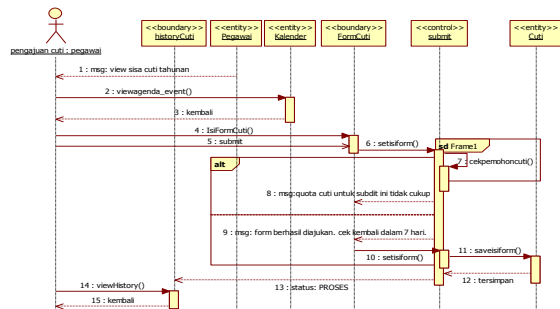
Gambar 1. Use case SI Cuti Pegawai.

Keterangan:

Use case diagram menggambarkan fungsionalitas *system* atau syarat – syarat yang harus dipenuhi *system* dari pandangan pemakai.

### Diagram Sekuensial

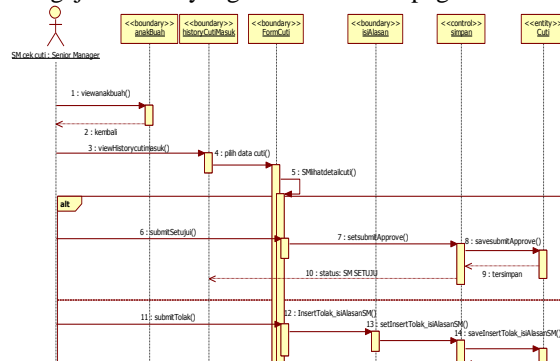
*Diagram* sekuensial atau *sequence diagram* digunakan untuk menunjukkan aliran fungsionalitas dalam *usecase*.



Gambar 2. Diagram sekuensial pengajuan cuti.

Keterangan:

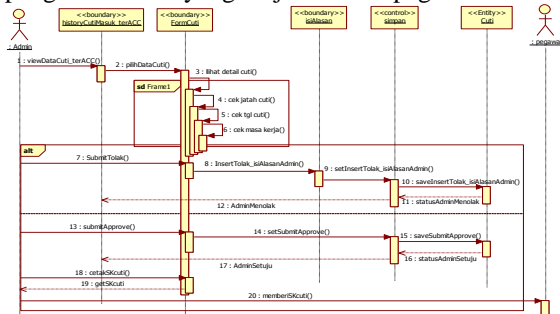
Menjelaskan bagaimana proses mengajukan cuti yang dilakukan oleh pegawai.



Gambar 3. Diagram Sekuensial SM cek cuti.

Keterangan:

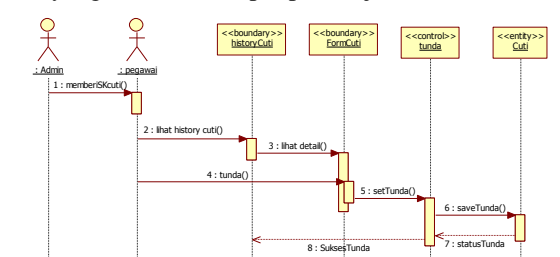
Menjelaskan bagaimana SM melakukan pengecekan cuti yang diajukan oleh pegawai.



Gambar 4. Diagram Sekuensial Admin cek cuti.

Keterangan:

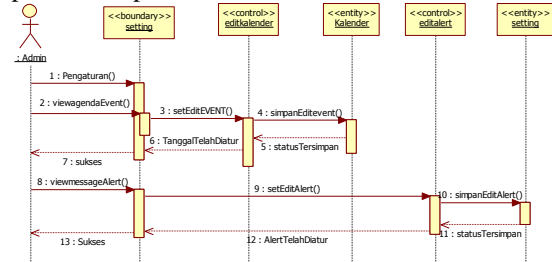
Menjelaskan bagaimana admin melakukan cek cuti dan mencetak skcuti terhadap permohonan cuti yang sudah mendapat persetujuan dari SM.



Gambar 5. Diagram Sekuensial Tunda Cuti.

Keterangan:

Menjelaskan bagaimana user melakukan penundaan permohonan cuti.



Gambar 6. Diagram Sekuensial pengaturan.

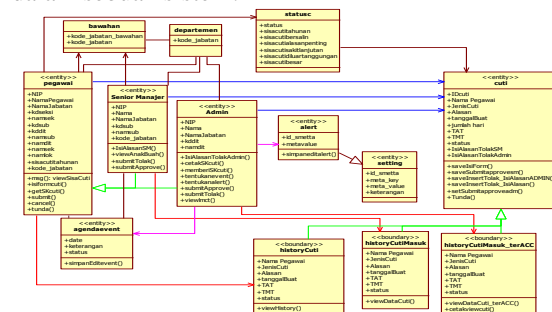
Keterangan:

Menjelaskan bagaimana alur ketika admin melakukan pengaturan terhadap komponen sistem cuti.

**Perancangan sistem** dibangun untuk memenuhi kebutuhan *user* mengenai gambaran yang jelas tentang perancangan sistem yang akan dibangun serta diimplementasikan dengan menggunakan *Class Diagram*, *State machine Diagram*, *Componen Diagram*, *deployment Diagram* dan *database diagram*.

## Class Diagram

*Class diagram* adalah *diagram* yang menunjukkan interaksi antar *class* / relasi antar *class* dalam sebuah sistem.



Gambar 7. Class Diagram SI Cuti Pegawai.

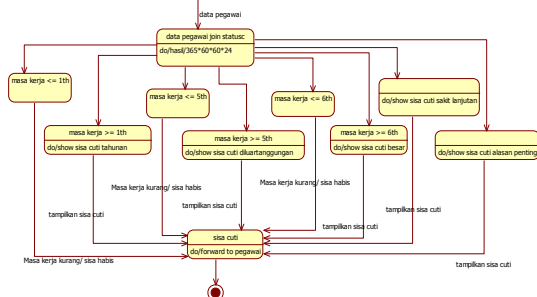
Keterangan:

Menjelaskan hubungan antara *class* yang satu dengan *class* yang lainnya.



## Statechart Diagram

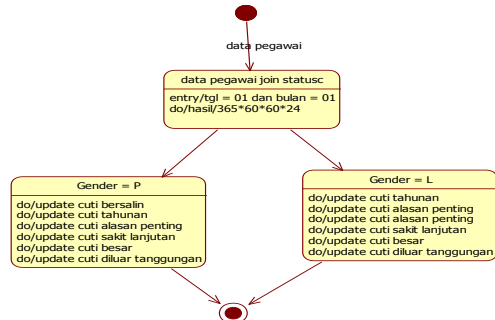
*Diagram statechart* menunjukkan siklus hidup sebuah objek tunggal, dari saat dibuat sampai objek tersebut dihapus. *Diagram* statechart merupakan cara tepat untuk memodelkan perilaku dinamis sebuah *class*.



Gambar 8. *Statechart Diagram* sisa cuti.

Keterangan:

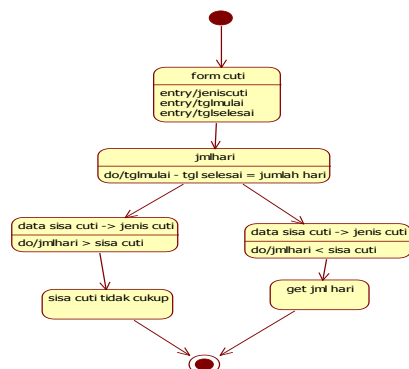
Menjelaskan bagaimana sistem menampilkan sisa cuti.



Gambar 9. *Statechart Diagram* reset cuti.

Keterangan:

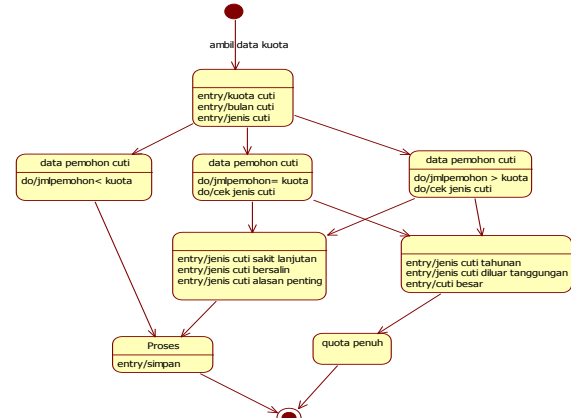
Menjelaskan bagaimana sistem melakukan pengembalian terhadap sisa cuti yang telah habis setiap tahunnya.



Gambar 10. *Statechart Diagram* proses hitung jumlah hari.

Keterangan:

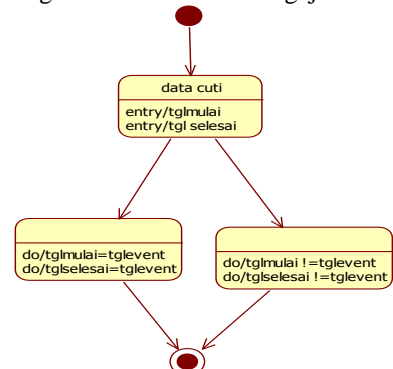
Menjelaskan bagaimana sistem melakukan perhitungan jumlah hari pada form permohonan cuti.



Gambar 11. *Statechart Diagram* cek kuota cuti.

Keterangan:

Menjelaskan bagaimana sistem menghitung kuota cuti untuk mengajukan cuti.



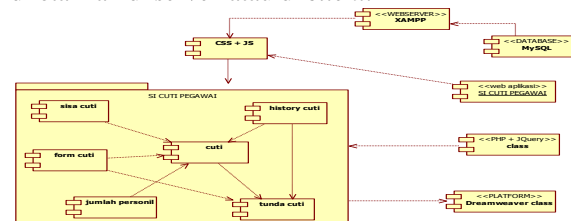
Gambar 12. *Statechart Diagram* cek tanggal cuti.

Keterangan:

Menjelaskan sistem mengecek tanggal yang diajukan oleh pemohon cuti apakah terdapat *event* atau tanggal yang tidak boleh diambil cuti.

## Diagram Komponen

Komponen adalah bagian fisik dari sistem. Karena perancangan komponen tidak bisa dilepaskan dari *platform* implementasi. Sistem cuti pegawai adalah aplikasi berbasis *web* dengan *database server* dimana logika bisnis bisa diletakkan di *server* atau di *client*.

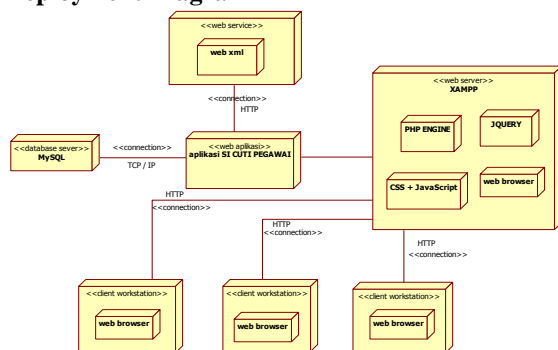


Gambar 13 *Diagram Komponen* SI Cuti Pegawai.

Keterangan:

Komponen terhubung oleh garis putus-putus yang menampilkan hubungan dependensi antar komponen.

## Deployment Diagram

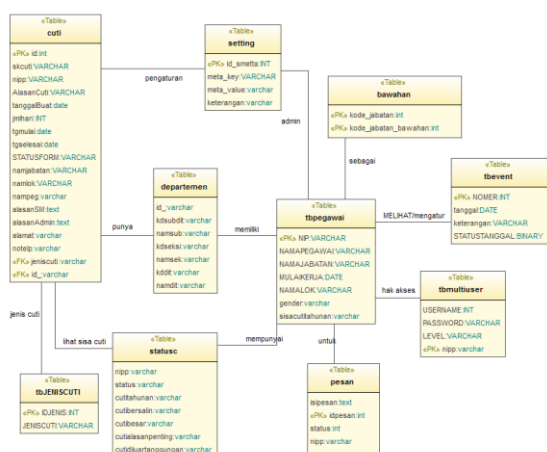


Gambar 14. *Deployment Diagram* SI Cuti Pegawai

Keterangan:

Web service cukup diunggah ke web server seperti pada gambar dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi yaitu client. Web service berjalan di port 80 yang merupakan protokol standar HTTP.

## Database Diagram



Gambar 15. *Database Diagram* SI Cuti Pegawai.

Keterangan:

Menjelaskan bagaimana relasi antar tabel pada database cuti.

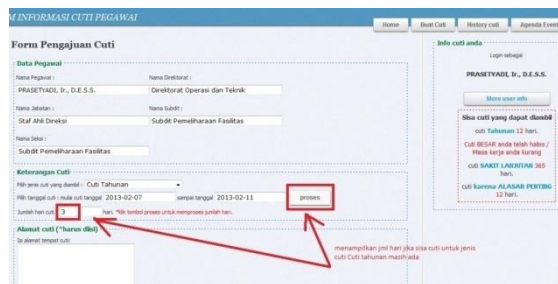
## IMPLEMENTASI PERANGKAT LUNAK

Implementasi merupakan tahap penerapan sistem untuk dapat dioperasikan. Tahapan ini dilakukan setelah perancangan selesai dilakukan dan selanjutnya akan diimplementasikan. Pada bab ini akan dilakukan pembahasan meliputi tahap koneksi database dan implementasi interface system.

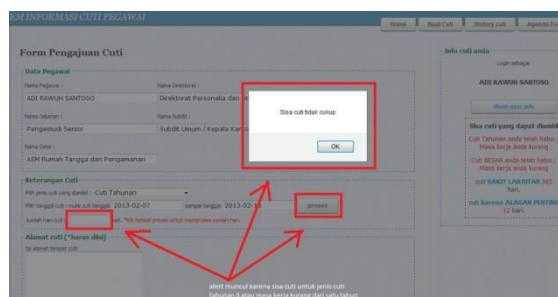
### 1. Uji coba untuk Cuti Tahunan

Berikut ini merupakan implementasi dari aturan cuti pasal 30 ayat 2 butir a. Yang berisi, Cuti tahunan, yaitu cuti yang diberikan kepada pegawai yang telah bekerja 12 (dua belas) bulan terus

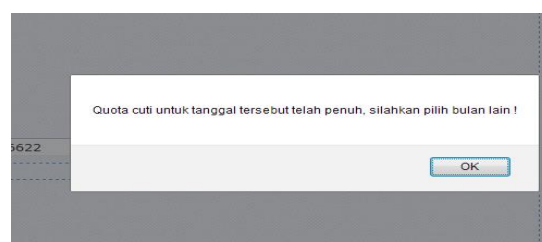
menerus berhak atas cuti tahunan paling lama 12 (dua belas) hari kerja.



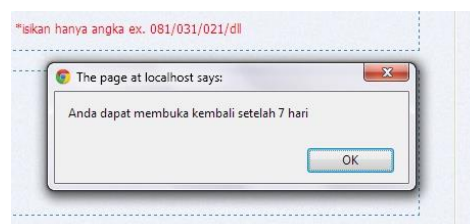
Gambar 16. Cuti tahunan - proses sukses.



Gambar 17. Cuti tahunan – proses gagal.



Gambar 18. Kuota cuti penuh.



Gambar 19. form cuti berhasil submit.

Untuk jenis – jenis cuti yang lain kecuali untuk jenis cuti sakit lanjutan, karena alasan penting dan bersalin, proses pengajuan cuti kurang lebih sama seperti proses pengajuan cuti tahunan untuk langkah-langkahnya.

### 2. Uji coba untuk Cuti Besar

Berikut ini merupakan implementasi dari aturan cuti pasal 30 ayat 2 butir b. Yang berisi, Cuti besar, yaitu cuti yang diberikan kepada pegawai yang telah bekerja paling sedikit 6 (enam) tahun terus menerus berhak atas cuti besar paling lama 3 (tiga) bulan.

Gambar 20. User Info.

Keterangan:

Dijelaskan bahwa pegawai bernama Sriyana Rusmala Dewi, SE dapat mengambil Cuti besar.

### 3. Uji coba untuk Cuti Sakit Lanjutan

Berikut ini merupakan implementasi dari aturan cuti pasal 30 ayat 2 butir c. Yang berisi, Cuti sakit, yaitu cuti yang diberikan kepada pegawai yang menderita sakit lebih dari 2 (dua) hari kerja berhak atas cuti sakit paling lama 1 (satu) tahun. Yang dimaksud cuti sakit disini adalah cuti sakit yang apabila telah melewati ijin karena sakit yang disertai surat keterangan sakit dari dokter sebut saja Cuti Sakit Lanjutan.

Gambar 21. User Info.

Keterangan:

Prosesnya tidak seperti pada uji coba cuti tahunan. Tentunya jumlah hari akan lebih besar ketimbang cuti Tahunan. Sistem tidak melakukan pengecekan jumlah kuota pada form cuti.

### 4. Uji coba untuk Cuti Bersalin

Berikut ini merupakan implementasi dari aturan cuti pasal 30 ayat 2 butir d. Yang berisi, Cuti bersalin, yaitu cuti yang diberikan kepada pegawai untuk melakukan persalinan berhak atas cuti bersalin selama paling lama 3 (tiga) bulan dengan ketentuan pelaksanaan 1 bulan sebelum dan 2 bulan setelah kelahiran.

Gambar 22. Jenis cuti bersalin.

Keterangan:

Jenis cuti bersalin ditampilkan pada pegawai perempuan.

Gambar 23. Jenis cuti pada pegawai laki – laki.

Keterangan:

Berarti pada saat pegawai laki – laki memilih jenis cuti, untuk jenis cuti bersalin tidak berada pada pilihan jenis cuti.

### 5. Uji coba untuk Cuti karena alasan penting

Gambar 24. user info

Keterangan:

Dijelaskan bahwa pegawai tersebut dapat mengambil jenis cuti karena alasan penting. System tidak melakukan pengecekan kuota cuti.

### 6. Uji coba untuk Cuti diluar tanggungan perusahaan

Berikut ini merupakan implementasi dari aturan cuti pasal 30 ayat 2 butir f. Yang berisi, Cuti diluar tanggungan Perusahaan, yaitu cuti yang diberikan kepada pegawai yang telah bekerja paling sedikit 5 (lima) tahun secara terus menerus dan hanya diberikan bagi pegawai yang melaksanakan pendidikan tanpa ikatan dinas maksimal 3 tahun.

Gambar 25. user info.

Keterangan:

Dijelaskan bahwa pegawai tersebut dapat mengambil jenis cuti diluar tanggungan.

## 7. Uji coba pegawai tidak dapat melakukan cuti

Info cuti anda  
Login sebagai  
PRASETYADI, Ir., D.E.S.S.  
More user info  
Sisa cuti yang dapat diambil  
cuti Tahunan 12 hari.  
Cuti BESAR anda telah habis / Masa kerja anda kurang  
Cuti DI LUAR TANGGUNGAN anda telah habis / Masa kerja anda kurang  
cuti SAKIT LANJUTAN 365 hari.  
cuti karena ALASAN PENTING 12 hari.

Gambar 26. user info.

Keterangan:

Dijelaskan bahwa pegawai tersebut tidak dapat mengambil cuti karena sisa cuti habis/ masa kerja belum memenuhi syarat.

## 8. Uji coba SM melakukan pengaturan kuota cuti

Berikut ini adalah gambar dimana SM melakukan pengaturan kuota cuti untuk subditnya.

Untuk mengubah kuota cuti yang dapat diambil untuk subdit.  
3 SIMPAN  
Pesan yang tampil saat ini adalah  
"kuota cuti untuk subdit saat ini 5 orang maksimal untuk satu subdit".

Gambar 27. Kuota cuti.

Keterangan:

dijelaskan bahwa SM bisa menentukan kuota cuti untuk bawahannya. Sehingga jika jumlah pemohon cuti melebihi kuota. Akan muncul alert yang berisi bahwa pegawai tersebut tidak dapat mengambil jenis cuti yang sama karena kuota cuti sudah penuh.

Untuk mengubah kuota cuti yang dapat diambil untuk subdit.  
3 SIMPAN  
Pesan yang tampil saat ini adalah  
"kuota cuti untuk subdit saat ini 3 orang maksimal untuk satu subdit".

Gambar 28. Kuota sukses diatur.

Keterangan:

Dijelaskan bahwa SM berhasil mengatur banyaknya jumlah kuota cuti untuk mengatur banyaknya jumlah pemohon cuti yang dapat mengambil cuti.

## 9. Uji coba SM melakukan persetujuan cuti

Form Pengajuan Cuti  
Data Pegawai  
Nama Pegawai: RIZA ROSDIANA, ST.  
Nama Direktur: Direktorat Personalia dan Umum  
Nama Jabatan: Asisten Sekjen Kesenjahteraan  
Nama Subdit: Subdit Kesenjahteraan dan Kinerja SDM  
Keterangan Cuti: jenis cuti yang diambil: Cuti Tahunan, tanggal cuti: mulai cuti tanggal 2013-02-08 sampai tanggal 2013-02-12, jumlah hari cuti: 3 hari.  
alamat cuti: and / telp 21223  
alasan cuti: andal  
SETUJUI TOLAK

Gambar 29. Cuti masuk di SM.

Form Pengajuan Cuti  
Data Pegawai  
Nama Pegawai: RIZA ROSDIANA, ST.  
Nama Direktur: Direktorat Personalia dan Umum  
Nama Jabatan: Asisten Sekjen Kesenjahteraan  
Nama Subdit: Subdit Kesenjahteraan dan Kinerja SDM  
Keterangan Cuti: jenis cuti yang diambil: Cuti Tahunan, tanggal cuti: mulai cuti tanggal 2013-02-08 sampai tanggal 2013-02-12, jumlah hari cuti: 3 hari.  
alamat cuti: and / telp 21223  
alasan cuti: andal  
SETUJUI TOLAK

Gambar 30. detail cuti.

The page at localhost says:  
Form pengajuan cuti sukses disetujui  
OK

Gambar 31. Alert form telah disetujui oleh SM.

isi alasan tolak  
Isi alasan penolakan pengajuan cuti:  
CONFIRM Cancel  
SETUJUI TOLAK

Gambar 32. Tolak cuti oleh SM.

Sukses menolak  
OK

Gambar 33. isi alasan sukses.

## 10. Uji coba Admin melakukan pengaturan cuti

Form Pengaturan Cuti  
Data Pegawai  
Nama Pegawai: NOVIANINGRUM, A.M.D.  
Nama Direktur: Direktorat Personalia dan Umum  
Nama Jabatan: Asisten Sekjen Kesenjahteraan  
Nama Subdit: Subdit Kesenjahteraan dan Kinerja SDM  
Keterangan Cuti: jenis cuti yang diambil: Cuti Tahunan, tanggal cuti: mulai cuti tanggal 2013-02-08 sampai tanggal 2013-02-12, jumlah hari cuti: 3 hari.  
alamat cuti: and / telp 21223  
alasan cuti: andal  
SETUJUI TOLAK

Gambar 34. Pengaturan.



Keterangan:

Admin dapat menentukan/ mengatur alert (pesan hari) yang tampil saat sukses melakukan pengajuan cuti dan menentukan tanggal – tanggal mana saja yang tidak dapat diambil cuti oleh pegawai.

### 11. Uji coba Admin melakukan approve/tolak form cuti

Klik ini untuk melihat siapa saja yang belum pernah mengambil cuti.

Klik untuk melihat detail.

Tampilkan 10 data

Nama	Jenis cuti	Mulai	Selesai	Tglbuat	Status cuti
RIZA ROSDIANA, ST.	Cuti Tahunan	2013-02-08	2013-02-12	2013-02-07	SM menolak
DIONET WIRAWAN, SH.	Cuti karena alasan penting	2013-02-08	2013-02-12	2013-02-08	PROSES
DIONET WIRAWAN, SH.	Cuti Sakit Lanjutan	2013-02-08	2013-02-15	2013-02-08	PROSES
RIZA ROSDIANA, ST.	Cuti Bersalin	2013-02-13	2013-06-05	2013-02-08	PROSES
PRASETYADI, Ir., D.E.S.S.	Cuti karena alasan penting	2013-02-08	2013-02-12	2013-02-08	PROSES
ANJAR SULISTYORINI, S.Psi.	Cuti Tahunan	2013-02-08	2013-02-12	2013-02-07	PROSES
RONY KURNIAWAN, S.Sos.	Cuti Tahunan	2013-02-08	2013-02-12	2013-02-07	PROSES

7 dari 7 data

PRINT FORM

Gambar 35. daftar cuti.

Keterangan:

Pada gambar dijelaskan bahwa admin mampu melihat daftar cuti dengan status Proses (cuti yang masih membutuhkan persetujuan SM), SM setuju (daftar cuti yang telah mendapat persetujuan oleh SM), SM Menolak (permohonan cuti yang ditolak oleh SM), Admin Setuju (cuti yang telah mendapat persetujuan oleh Admin) dan Admin menolak (cuti yang ditolak oleh SM), dan Tunda (cuti yang ditunda oleh Pegawai). Tombol “2” untuk mencetak daftar cuti.

Pegawai yang belum pernah mengambil cuti

Tampilkan 10 data

Nipp	Nama	Lokasi	Cuti	Sisa cuti
16711871	HASNA RIEP KAHARIYAH, SH., MH.	ASM Kinerja SDM dan Hubungan Industrial		12
469081469	PRASETYADI, Ir., D.E.S.S.	Subdit Pemeliharaan Fasilitas		12
560303114	FADJAR SANTOSO, CAPT. MSc.	Subdit Pelayanan Terminal		12
560502177	CHUS NURAINI	Aseskep Bidang Hub. Kalimantan & Monitoring GCG		12
560601740	HARWIJONO, Drs., MM.	Subdit Kemitraan dan Bina Lingkungan		12
560602180	LIARI MK, Drs., MM.	ASM Akuntansi Aktiva Tetap		12
560602189	ENDANG HARIJATI, S. Sos	ASM Tata Usaha Perusahaan		12
560702000	YULIANTO, S. Sos	Pengawas Bidang III		12
560802127	MOCHAMAD CHOIRON	ASM Rumah Tangga dan Pengamanan		12
560802232	ARIADITI, S. Sos	ASM Anggaran		12

Menampilkan 1 sampai 10 dari 228 data

Gambar 36. daftar pegawai yang belum pernah mengambil cuti (tombol 1).

**Form Pengajuan Cuti**

**Validasi**

- cek jumlah cuti: ☒ Jumlah cuti masih ada [detail](#)
- cek tanggal cuti: ☒ Tanggal cuti bisa diambil [detail](#)
- cek masa kerja pegawai: ☒ Masa kerja memenuhi syarat [detail](#)

FORM CUTI INI TELAH MENDAPAT PERSETUJUAN DARI SM

**Data Pegawai**

Nama Pegawai: ANJAR SULISTYORINI, S.Psi. Nama Subdit: Subdit Kesejahteraan dan Kinerja SDM  
Nama Setai: ASM Kinerja SDM dan Hubungan Industrial

**Keterangan Cuti**

SK Cuti: 2820130208-030239-02761103817-03  
Jenis cuti yang diambil: Cuti Tahunan  
Tanggal cuti: mulai cuti tanggal 2013-02-08 sampai tanggal 2013-02-12  
Jumlah hari cuti yang diambil: 3 hari.

**Alasan cuti**

adefa

[SETUJUI](#) [TOLAK](#)

Gambar 37. Detail cuti oleh Admin.

**Form Pengajuan Cuti**

FORM CUTI INI TELAH MENDAPAT PERSETUJUAN DARI ADMIN

**Data Pegawai**

Nama Pegawai: RIZA ROSDIANA, ST. Nama Subdit: Subdit Kesejahteraan dan Kinerja SDM  
Nama Setai: ASM Kesejahteraan

**Keterangan Cuti**

SK Cuti: 2820130208-024959-02761103817-03  
Jenis cuti yang diambil: Cuti Bersalin  
Tanggal cuti: mulai cuti tanggal 2013-03-13 sampai tanggal 2013-06-05  
Jumlah hari cuti yang diambil: 60 hari.

**Alasan cuti**

adefa

[PRINT FORM](#)

Gambar 38. Admin setuju siap print.

Keterangan:

Merupakan detail cuti dari salah satu daftar cuti pada gambar. Terdapat label biru jika mendapat persetujuan dari SM. Jika admin menolak cuti tersebut maka admin harus mengisi alasan seperti pada gambar tolak cuti oleh SM.

[Print](#)

**PT. PELABUHAN INDONESIA III (PERSERO)**

KANTOR PUSAT  
Jl. Perak Timur No. 610 Surabaya

Surat Keputusan Cuti : 2820130208-024959-02761103817-03  
Perihal : Pemohonan Cuti Bersalin

Yth. (nama sm)  
PT. Pelabuhan Indonesia III (Persero)  
di S U R A B A Y A

Yang bertanda tangan di bawah ini:

Nama : RIZA ROSDIANA, ST.  
NIPP : 177102495  
Pangkat / Golongan : Analis Junior Kesejahteraan  
Nama / Kelas Jabatan : ASM Kesejahteraan  
Unit Kerja : Subdit Kesejahteraan dan Kinerja SDM

Dengan ini mengajukan permohonan Cuti Bersalin selama 60 hari kerja, terhitung tanggal 2013-03-13 s.d. 2013-06-05. Dan selama menjalankan Cuti Bersalin alamat saya sd.

Demikian permohonan ini saya buat untuk dapat dipertimbangkan sebagaimana mestinya.

**KEPUTUSAN PEJABAT YANG BERWENANG MEMBERIKAN CUTI**  
Senior Manager Kesejahteraan dan Kinerja SDM

(md)

Jl. Perak Timur No. 610, Surabaya, 60615, PO Box 1123  
Telp. 031-3298631 s.d 37, Fax. 031-3295204, 3295207  
website: www.pp3.co.id E-mail: portina3@pp3.co.id

Gambar 39. Print Preview

### DAFTAR PUSTAKA

- [1] Bonnie, Soeherman. Marion Pinontoan., (2008), "Designing Information System", Jakarta: PT Elex Media Komputindo.
- [2] Dennis, Alan., Wixom, Barbara Haley, (2005), "System Analysis and Design with UML

- Version 2.0”, Addison-Wesley, Massachusetts.
- [3] Fatansyah, Ir. (2004), *”Basis Data”*, Bandung: Informatika.
- [4] Hadi, Mulyo., (2009), *”Dreamwebaver CS4”*, Palembang : Maxicom. ISBN 978-979-1398-80-0.
- [5] Jogyanto. HM (1992), *”Sistem Informasi Manajemen”*, Andy Offset Yogyakarta, Yogyakarta.
- [6] Jogyanto H.M., (1995), *”Analisis dan Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis”*, Andi Offset, Yogyakarta.
- [7] Jogyanto. 1995. *Analisa dan Perancangan Sistem Informasi*. Andy Offset. Yogyakarta.
- [8] Jogyanto, (2008), *”Metode Penelitian Sistem Informasi”*, Edisi Soft Cover, Andi publisher, Yogyakarta.
- [9] Jogyanto, (2004), *”Analisis dan Desain Sistem Informasi”*, Informatika, Bandung.
- [10] Kadir, Abdul., (2004), *”Dasar Aplikasi Database MySQL”*, Andi. Yogyakarta.
- [11] Kadir, Abdul., (2009), *”Dasar Perancangan dan Implementasi Database Relational”*, Andi, Yogyakarta.
- [12] Kristianto, Andri., (2008), *”Perancangan Sistem Informasi dan aplikasinya”*, Edisi Revisi, Gaya Media, Yogyakarta.
- [13] McLeod, Reymond, (2000), *”Sistem Informasi Manajemen”*, PT. Prenhallindo, Jakarta.
- [14] Munawar, (2005), *”Pemodelan Visual dengan UML”*, Graha Ilmu, Yogyakarta.
- [15] Murdick, Robert. E, Ross, Joel . (1997), *”Sistem Informasi untuk Manajemen Modern*. Erlangga, Jakarta.
- [16] Samani, Mansur. (2007), *”Sertifikasi Guru Menuju profesional Pendidik”*, Jakarta: Bumi Aksara.
- [17] Scott GM. (1996), *”Prinsip-Prinsip Sistem Informasi Manajemen”*, Jakarta: PT Raja Grafindo Persada.
- [18] Yuniar Banowosari, Lintang (2007), *”Object Oriented Database”*, Gunadarma, Jakarta.



(8)

## SISTEM PENCARIAN PARKIR KOSONG PADA LAHAN PARKIR KHUSUS MOBIL DI PUSAT GROSIR SURABAYA

Setyo Purwo Sarwono<sup>1</sup>, Tjatusari Widiartin<sup>2</sup>, Noven Indra Prasetya<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Wijaya Kusuma Surabaya  
Jl. Dukuh Kupang XXV/ 54, [wowoqdino88@gmail.com](mailto:wowoqdino88@gmail.com)

### ABSTRAK

Seiring dengan banyaknya jumlah pengunjung suatu *mall* atau pusat perbelanjaan tentunya harus diimbangi dengan berbagai fasilitas yang memadai, salah satunya adalah fasilitas parkir. Dalam kenyataannya di lapangan pengunjung merasa kerepotan ketika akan memarkir mobilnya, pengunjung harus berkeliling dari tiap-tiap lantai untuk mencari parkir yang kosong.

Untuk mengatasi masalah tersebut penulis ingin membuat suatu sistem pencarian parkir kosong pada lahan parkir mobil. Sistem ini menggunakan arduino yang nantinya akan di sambungkan dengan sensor LDR untuk mendeteksi parkir tersebut terisi atau kosong. Sehingga proses pencarian parkir tersebut dapat di permudah, dipercepat dan lebih praktis sehingga pengunjung dapat menuju lokasi parkir yang kosong tanpa harus mencarinya

**Kata kunci :** Parkir Mobil, Arduino, LDR

### PENDAHULUAN

#### Latar Belakang

Omzet pada suatu *mall* atau pusat perbelanjaan itu ditentukan oleh jumlah pengunjungnya, jika omzet pada suatu *mall* atau pusat perbelanjaan itu menurun dikarenakan berkurangnya jumlah pengunjung pada suatu *mall* atau pusat perbelanjaan tersebut, salah satunya adalah Pusat Grosir Surabaya. Menurunnya jumlah pengunjung disebabkan oleh beberapa faktor, salah satunya adalah kurang nyamannya tempat area parkir. Pengunjung batal mengunjungi *mall* atau pusat perbelanjaan dikarenakan tidak mengetahui sebenarnya *mall* atau pusat perbelanjaan tersebut ada parkir kosong atau disatu sisi lainnya pengunjung kesulitan mencari keberadaan atau tempat tata letak parkir kosong yang dilakukan secara manual.

Dengan adanya permasalahan seperti ini penulis ingin membantu, yaitu dengan membuat perancangan sistem pencarian tempat parkir kosong pada suatu *mall* atau pusat perbelanjaan, yang berguna untuk mengetahui letak adanya parkir kosong.

Sistem pencarian parkir kosong pada lahan parkir mobil ini dapat membantu pengunjung yang kesulitan mencari parkir kosong pada suatu *mall* atau pusat perbelanjaan, dengan menggunakan aplikasi pencarian parkir kosong diharapkan dapat memberikan kemudahan dan kenyamanan bagi pengunjung sehingga dapat meningkatkan jumlah

pengunjung dan secara tidak langsung dapat menaikkan omzet dari *mall* atau pusat perbelanjaan itu sendiri.

Sistem pencarian tempat parkir kosong ini berfungsi untuk memudahkan proses pencarian lokasi kosong pada suatu area parkir mobil atau memudahkan pengunjung mengetahui tata letak tempat parkir kosong tanpa harus melakukan pencarian secara manual yaitu memutar dari tiap-tiap lantai, memberikan informasi kepada pengunjung tentang adanya tempat area parkir kosong dan mengoptimalkan layanan pengguna lahan parkir lebih mudah dan efisien. Semua proses tersebut akan dipermudah, dipercepat, lebih praktis.

### TINJAUAN PUSTAKA

#### Pengertian Sistem

Sistem adalah suatu jaringan kerja dari prosedur– prosedur yang saling berhubungan, berkumpul, bersama–sama untuk melakukan suatu kegiatan atau menyelesaikan suatu sasaran tertentu . Dalam perkembangannya sistem dibedakan menjadi dua yaitu sistem terbuka dan sistem tertutup. Sistem terbuka adalah sistem yang terhubung dengan arus sumber daya luar dan tidak mempunyai elemen pengendali. Sedangkan sistem tertutup tidak mempunyai elemen pengontrol dan dihubungkan pada lingkungan sekitarnya. Sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu:

1. **Komponen Sistem**  
Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling kerjasama membentuk satu kesatuan.
2. **Sifat-sifat**  
Sistem itu terdiri untuk menjalankan fungsi tertentu dan mempunyai sistem yang lain secara keseluruhan.
3. **Batasan (*Boundary*) Sistem** Merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya.
4. **Lingkungan Luar Sistem (*Environment*)** Adalah apapun diluar batasan sistem yang mempengaruhi operasi sistem, lingkungan luar sistem dapat bersifat menguntungkan dan juga dapat bersifat merugikan sistem tersebut.
5. **Penghubung (*Interface*) Sistem** Merupakan media penghubung antara suatu subsistem dengan subsistem lainnya.
6. **Masukkan Sistem (*Input*)**  
Adalah energi yang dimasukkan kedalam sistem, masukan dapat berupa masukan perawatan (*Maintenace Input*), dan masukan sinyal (*Signal Input*), maintenance input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Sedangkan signal input adalah energi yang diproses untuk mendapatkan keluaran.
7. **Keluaran Sistem (*Output*)**  
Adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan keluaran dapat merupakan masukan untuk subsistem yang lain kepada supra sistem.
8. **Sasaran Sistem**  
Suatu sistem pasti memiliki sasaran atau tujuan (*Goal*). Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan sistem dan keluaran yang akan di hasilkan sistem.

### **LDR (Light Dependent Resistor)**

Light Dependent Resistor atau yang biasa disebut LDR adalah jenis resistor yang nilainya berubah seiring intensitas cahaya yang diterima oleh komponen tersebut. Biasa digunakan sebagai detektor cahaya atau pengukur besaran konversi cahaya. Light Dependent Resistor, terdiri dari sebuah cakram semikonduktor yang mempunyai dua buah elektroda pada permukaannya. Pada saat gelap atau cahaya redup, bahan dari cakram tersebut menghasilkan elektron bebas dengan jumlah yang relatif kecil. Sehingga hanya ada sedikit elektron untuk mengangkut muatan elektrik. Artinya pada saat cahaya redup LDR menjadi konduktor yang buruk, atau bisa disebut juga LDR memiliki

resistansi yang besar pada saat gelap atau cahaya redup.

Pada saat cahaya terang, ada lebih banyak elektron yang lepas dari atom bahan semikonduktor tersebut. Sehingga akan ada lebih banyak elektron untuk mengangkut muatan elektrik. Artinya pada saat cahaya terang LDR menjadi konduktor yang baik, atau bisa disebut juga LDR memiliki resistansi yang kecil pada saat cahaya terang. LDR memanfaatkan bahan semikonduktor yang karakteristik listriknya berubah-ubah sesuai dengan cahaya yang diterima. Bahan yang digunakan adalah Kadmium Sulfida (CdS) dan Kadmium Selenida (CdSe). Bahan-bahan ini paling sensitif terhadap cahaya dalam spektrum tampak, dengan puncaknya sekitar 0,6  $\mu\text{m}$  untuk CdS dan 0,75  $\mu\text{m}$  untuk CdSe. Sebuah LDR CdS yang typikal memiliki resistansi sekitar 1 M $\Omega$  dalam kondisi gelap gulita dan kurang dari 1 K $\Omega$  ketika ditempatkan dibawah sumber cahaya terang.

Karakteristik LDR terdiri dari dua macam yaitu Laju Recovery dan Respon Spektral:

1. **Laju Recovery**  
Bila sebuah LDR dibawa dari suatu ruangan dengan level kekuatan cahaya tertentu kedalam suatu ruangan yang gelap, maka bisa kita amati bahwa nilai resistansi dari LDR tidak akan segera berubah resistansinya pada keadaan ruangan gelap tersebut. Namun LDR tersebut hanya akan bisa mencapai harga di kegelapan setelah mengalami selang waktu tertentu. Laju recovery merupakan suatu ukuran praktis dan suatu kenaikan nilai resistansi dalam waktu tertentu. Harga ini ditulis dalam K /detik, untuk LDR type arus harganya lebih besar dari 200 K /detik (selama 20 menit pertama mulai dari level cahaya 100 lux), kecepatan tersebut akan lebih tinggi pada arah sebaliknya, yaitu pindah dari tempat gelap ke tempat terang yang memerlukan waktu kurang dari 10 ms untuk mencapai resistansi yang sesuai dengan level cahaya 400 lux.
2. **Respon Spektral**  
LDR tidak mempunyai sensitivitas yang sama untuk setiap panjang gelombang cahaya yang jatuh padanya (yaitu warna). Bahan yang biasa digunakan sebagai penghantar arus listrik yaitu tembaga, aluminium, baja, emas, dan perak. Dari kelima bahan tersebut tembaga merupakan penghantar yang paling banyak digunakan karena mempunyai daya hantar yang baik.

## Arduino

Arduino merupakan platform yang terdiri dari software dan hardware. Hardware Arduino sama dengan microcontroller pada umumnya hanya pada arduino ditambahkan penamaan pin agar mudah diingat. Software Arduino merupakan software open source sehingga dapat didownload secara gratis. Software ini digunakan untuk membuat dan memasukkan program ke dalam Arduino. Pemrograman Arduino tidak sebanyak tahapan microcontroller konvensional karena Arduino sudah didesain mudah untuk dipelajari, sehingga para pemula dapat mulai belajar microcontroller dengan Arduino.

Arduino merupakan platform *open source* baik secara hardware dan software. Arduino terdiri dari microcontroller megaAVR seperti ATmega8, ATmega168, ATmega328, ATmega1280, dan ATmega 2560 dengan menggunakan Kristal osilator 16 MHz, namun ada beberapa tipe Arduino yang menggunakan Kristal osilator 8 MHz. Catu daya yang dibutuhkan untuk mensupply minimum sistem Arduino cukup dengan tegangan 5 VDC. Port arduino Atmega series terdiri dari 20 pin yang meliputi 14 pin I/O digital dengan 6 pin dapat berfungsi sebagai output PWM (*Pulse Width Modulation*) dan 6 pin I/O analog. Kelebihan Arduino adalah tidak membutuhkan flash programmer external karena di dalam chip microcontroller Arduino telah diisi dengan *bootloader* yang membuat proses *upload* menjadi lebih sederhana. Untuk koneksi terhadap komputer dapat menggunakan RS232 to TTL Converter atau menggunakan Chip USB ke Serial converter seperti FTDI FT232.

Arduino board sendiri telah tersedia dalam banyak jenis baik yang sudah berkoneksi USB maupun serial. Contoh Arduino yang terkoneksi dengan USB seperti : Arduino Uno, Arduino Duemilanove, Arduino Diecimila, Arduino NG Rev. C , Arduino FIO, dan Arduino LilyPad. Untuk LilyPad memiliki ukuran sebesar kancing baju dan anti air sehingga dapat dicuci. Sedangkan Arduino Severino merupakan contoh untuk yang terkoneksi secara serial. Untuk para pemula yang bingung memilih jenis board yang cocok, dapat memilih Arduino Duemilanove atau Arduino UNO karena kedua jenis ini yang paling banyak digunakan. Namun jika ingin berkreasi lebih maka dapat membuat board sendiri dengan menyesuaikan kebutuhan dan dana yang ada. Selain Arduino board, juga terdapat perangkat tambahan yang disebut *shield* untuk pengembangan Arduino. Dengan shield ini maka tidak perlu lagi repot menyolder karena semua sudah didesain sesuai dengan pin arduino. Contoh shield seperti : *Ethernet*

*shield* untuk mengkoneksikan arduino dengan LAN, Xbee untuk memungkinkan beberapa arduino berkomunikasi secara wireless.

Arduino diciptakan untuk para pemula bahkan yang tidak memiliki basic bahasa pemrograman sama sekali karena menggunakan bahasa C++ yang telah dipermudah melalui library. Kenapa Arduino menggunakan bahasa C++, karena Bahasa C adalah bahasa yang sangat lazim dipakai sejak awal-awal komputer diciptakan dan sangat berperan dalam perkembangan software. Bahasa C++ telah membuat bermacam-macam sistem operasi dan compiler untuk banyak bahasa pemrograman, misalnya sistem operasi Unix, Linux, dsb. Bahasa C++ juga biasanya diajarkan di akademi dan perguruan tinggi selain bahasa pemrograman Basic atau Pascal dan Bahasa C++ adalah bahasa pemrograman yang sangat ampuh yang kekuatannya mendekati bahasa assembler. Bahasa C++ menghasilkan file kode objek yang sangat kecil dan dieksekusi dengan sangat cepat. Karena itu Bahasa C++ sering digunakan pada sistem operasi dan pemrograman mikrokontroler. Arduino menggunakan Software Processing yang digunakan untuk menulis program ke dalam Arduino. Processing sendiri merupakan penggabungan antara bahasa C++ dan Java. Software Arduino ini dapat di-install di berbagai operating sistem(OS) seperti: LINUX, Mac OS, Windows. Software IDE Arduino terdiri dari 3 (tiga) bagian:

1. Editor program, untuk menulis dan mengedit program dalam bahasa processing. Listing program pada Arduino disebut sketch.
2. Compiler, modul yang berfungsi mengubah bahasa processing (kode program) ke dalam kode biner karena kode biner adalah satu-satunya bahasa program yang dipahami oleh microcontroller.
3. Uploader, modul yang berfungsi memasukkan kode biner ke dalam memori microcontroller.

Struktur perintah pada arduino secara garis besar terdiri dari 2 (dua) bagian yaitu void setup dan void loop. Void setup berisi perintah yang akan dieksekusi hanya satu kali sejak arduino dihidupkan sedangkan void loop berisi perintah yang akan dieksekusi berulang-ulang selama arduino dinyalakan.

Kegunaan Arduino tergantung kepada kita yang membuat program. Arduino bisa digunakan untuk mengontrol LED, bisa juga digunakan untuk mengontrol helikopter. Contoh yang sudah pernah dibuat adalah MP3 player, pengontrol motor, mesin CNC, monitor kelembaban tanah, pengukur jarak, penggerak

servo, balon udara, pengontrol suhu, monitor energi, stasiun cuaca, pembaca RFID, drum elektronik, GPS logger, monitoring bensin dan masih banyak lagi.

## Resistor

Resistor adalah salah satu komponen elektronika yang berfungsi untuk menahan arus yang mengalir dalam suatu rangkaian/sistem elektronika. Resistor adalah komponen yang paling sering di gunakan dalam rangkaian-rangkaian elektronika. Karena itu anda harus mampu mempelajari bagaimana menghitung nilai suatu resistor dan daya yang di gunakan apabila anda merancang suatu rangkaian yang menggunakan resistor. Dan anda harus mampu mengetahui rangkaian elektronika yang mengantung suatu resistor yang rusak atau terbakar. Ada dua karakteristik resistor yang perlu di ketahui yaitu:

1. Nilai Resistansinya
2. rating dayanya (Kemampuan untuk menahan arus yang mengalir pada resistor tersebut)

Resistor mempunyai harga resistansi yang cukup banyak, mulai dari beberapa ohm di belakang koma sampai beberapa mega ohm di depan koma. Rating daya yang tertinggi di dapat yang mencapai beberapa ratus watt dan yang terendah sampai mencapai 0,1 watt rating daya sangat penting, sebab ia menunjukkan daya maksimum yang bisa disipasikan tanpa menimbulkan panas yang berlebihan yang dapat mengakibatkan kerusakan pada resistor tersebut. Disipasi artinya bahwa daya sebesar  $I^2R$  akan di buang kepadanya. Panas yang berlebihan dapat mengakibatkan terbakarnya resistor.

Untuk memilih resistor dalam pemakaian apapun di butuhkan berbagai macam pertimbangan yaitu: ukuran fisik, bentuknya, cara pemasangan dan penyambungannya pada rangkaian, harga resistansinya, disipasi dayanya, kemampuan menangani beban lebih, kehandalan, perubahan resistansi terhadap frekuensi dan terhadap tegangan yang jatuh padanya, ketahanan sebagai beban, pengaruh kondisi lingkungan dan umurnya.

Dalam prakteknya resistor hanya terbagi dua jenis yaitu :

1. Resistor tetap (Fixed resistor) Resistor jenis ini mempunyai nilai resistansi yang tetap (tidak berubah)

Adapun beberapa jenis resistor tetap antara lain:

1. Resistor Wirewound.

Sesuai dengan namanya, resistor ini terbuat dari lilitan kawat, di mana kawatnya terbuat dari jenis

nikel dan logam campuran untuk memperoleh nilai resistansi dengan koefisien temperature yang rendah, maka resistor ini dapat mempertahankan harga resistansi, toleransi, rating daya dan konfigurasi fisik yang lebih besar dibandingkan dengan tipe-tipe resistor yang lainnya.

Daya yang tersedia untuk resistor jenis ini yaitu mulai dari 5 watt sampai beberapa watt, dan range nilai resistansi mulai dari kurang dari satu ohm hingga beberapa ohm.

### 2. Resistor Film

Resistor ini juga terbagi ada beberapa jenis berdasarkan jenis bahan filmnya, yaitu:

- Resistor Film yang terbuat dari logam

- Resistor film yang terbuat dari karbon (resistor inilah yang paling sering digunakan dalam rangkaian elektronika). Resistor Film dari karbon tersedia di pasaran dengan nilai dari 1  $\Omega$  sampai 1000 M $\Omega$ . Rating dayanya dari 1/8 watt hingga 6 watt.

### 2. Resistor Variabel (variable Resistor)

Resistor ini mempunyai nilai resistansi dapat berubah-ubah.

### 3. Resistor Non linier

Resistor yang nilai hambatannya tidak linier karena pengaruh faktor lingkungan, misalnya suhu dan cahaya.

## Flowchart

Flowchart atau diagram alir merupakan sebuah diagram dengan simbol-simbol grafis yang menyatakan aliran algoritma atau proses yang menampilkan langkah-langkah yang disimbolkan dalam bentuk kotak, beserta urutannya dengan menghubungkan masing masing langkah tersebut menggunakan tanda panah. Diagram ini bisa memberi solusi selangkah demi selangkah untuk penyelesaian masalah yang ada di dalam proses atau algoritma tersebut.

Tujuan utama dari penggunaan flowchart adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai, rapi dan jelas dengan menggunakan simbol-simbol standart.

Pedoman yang perlu diperhatikan dalam membuat flowchart :

- a. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman.
- b. Kegiatan di dalam bagan alir harus ditunjukkan dengan jelas.
- c. Harus ditunjukkan dari mana kegiatan akan dimulai dan dimana akan berakhir.

- d. Masing-masing kegiatan di dalam bagan alir sebaiknya digunakan suatu kata yang mewakili suatu pekerjaan.
- e. Masing-masing kegiatan di dalam bagan alir harus didalam urutan yang semstinya.
- f. Kegiatan yang terpotong dan akan disambung ke tempat lain harus ditunjukkan dengan jelas menggunakan simbol penghubung.
- g. Gunakan simbol-simbol alir yang standart.

Terdapat 2 jenis flowchart yaitu sistem flowchart dan program flowchart.

#### 1. Sistem Flowchart

Sistem flowchart merupakan diagram alir yang menggambarkan suatu sistem peralatan komputer yang digunakan dalam proses pengolahan data serta hubungan antar peralatan tersebut. Sistem flow chart tidak digunakan untuk menggambarkan urutan langkah untuk memecahkan masalah , tetapi hanya untuk menggambarkan prosedur dalam sistem yang dibentuk.

#### 2. Flowchart Program

Merupakan bagan alir yang menggambarkan urutan logika dari suatu prosedur pemecahan masalah. Untuk menggambarakan flowchart program telah tersedia simbol-simbol standart. Berikut ini adalah gambar dari simbol-simbol standart yang digunakan pada flowchart program.

### Data Flow Diagram (DFD)

Sebelum mengimplementasi program, maka dilakukan pembuatan DFD atau Data Flow Diagram. DFD adalah suatu grafik yang menjelaskan sebuah sistem dengan menggunakan bentuk-bentuk atau symbol untuk menggambarkan aliran data dari proses-proses yang saling berhubungan . DFD menggambarkan input, proses, dan output yang terjadi dalam suatu sistem. DFD juga menggambarkan aliran data dalam sebuah sistem. Berdasarkan penjelasan di atas, dapat disimpulkan bahwa *Data Flow Diagram* (DFD) adalah diagram yang menggambarkan sistem sistem yang saling berhubungan baik untuk sistem yang sudah ada maupun sistem yang baru.

## ANALISA DAN PERANCANGAN

### Analisa Sistem

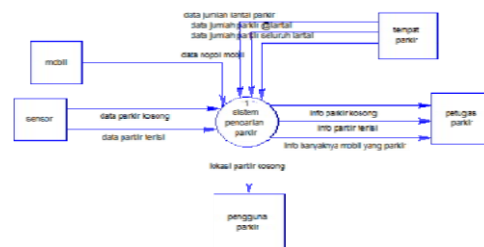
Sistem pencarian ini ini didefinisikan sebagai rencana untuk melakukan pencarian parkir kosong pada suatu area parkir mobil. diperlukan beberapa komponen untuk membangun sistem ini, yaitu LDR atau Light Dependent Resistor atau sensor cahaya adalah suatu bentuk komponen yang

mempunyai perubahan resistansi yang besarnya tergantung pada cahaya. Arduino, komponen ini adalah kit elektronik atau papan rangkaian elektronik open source yang di dalamnya terdapat komponen utama yaitu sebuah chip mikrokontroler dengan jenis AVR. Mikrokontroler itu sendiri adalah chip atau IC (integrated circuit) yang bisa diprogram menggunakan komputer. Dan yang terakhir adalah perangkat komputer yang berfungsi sebagai display parkir kosong.

Pada implementasi nantinya pada lokasi atau satu tempat parkir mobil akan di pasangi dengan satu LDR atau Light Dependent Resistor atau sensor cahaya, pemasangan LDR di letakkan tepat di mana jika terdapat mobil yang sedang parkir dapat membuat intensitas cahaya menjadi lebih kecil atau hampir tidak ada cahaya, karena dengan begitu dapat dijadikan penanda jika dalam keadaan intensitas cahaya lebih kecil atau hampir tidak ada cahaya dapat dikatakan atau diartikan jika parkir tersebut telah terisi, dan jika ada intensitas cahaya yang terang maka parkir tersebut kosong atau belum terisi, dari LDR nantinya akan disambung dengan mikrokontoler, dengan menggunakan arduino. Dan nantinya akan dihubungkan dengan perangkat komputer yang nantinya diletak pada loket parkir masuk, sehingga nanti pada petugas jaga loket parkir masuk dapat mencantumkan tata letak parkir kosong pada tiket sehingga pengguna parkir tidak perlu susah-susah mencari lagi tempat parkir kosong dan pengunjung akan menuju lahan parkir kosong sesuai petunjuk atau letak parkir kosong sesuai yang tercantum pada tiket tersebut.

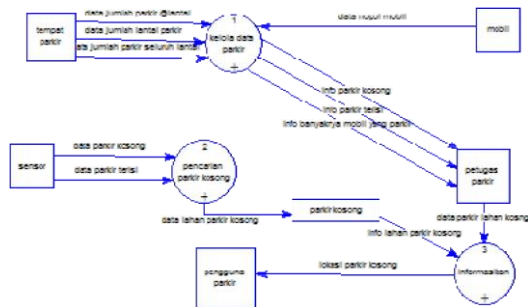
### DFD

Pada Gambar 3.1 dibawah ini merupakan DFD level 0 dari sistem pencarian parkir kosong pada lahan parkir khusus mobil di pusat grosir Surabaya



Gambar 3.1 DFD Level 0

Pada gambar 3.1 DFD level 0 ini memiliki 3 entity, yang setiap entity memiliki tugas dan peranan fungsi yang berbeda-beda di dalam sistem pencarian parkir kosong. Entity tersebut yaitu mobil, tempat parkir, sensor, petugas parkir dan pengguna parkir.



Gambar 3.2 DFD Level 1

Pada gambar 3.2 DFD level 1 ini memiliki 2 proses, yaitu, pencarian parkir kosong dan informasikan, dari 3 proses ini memiliki dengan ke 5 entity yang ada dan mempunyai 1 data store.

### DFD Level 2 Pencarian Parkir Kosong

Berikut ini adalah DFD Level 2 pencarian parkir kosong, untuk lebih jelasnya bisa melihat gambar di bawah ini :

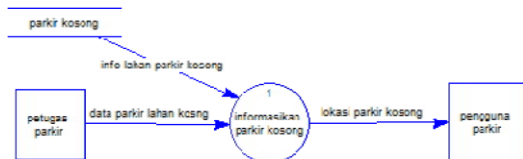


Gambar 3.3 DFD Level 2 pencarian parkir kosong

Pada gambar 3.3 DFD level 2 pencarian parkir kosong ini menghasilkan data store/penyimpanan data yaitu data store parker kosong.

### DFD Level 2 Proses Informasikan

Berikut ini adalah DFD Level 2 proses informasikan, untuk lebih jelasnya bisa melihat gambar di bawah ini :



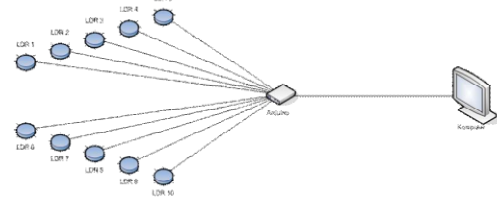
Gambar 3.4 DFD Level 2 proses informasikan

Pada gambar 3.4. DFD level 2 proses informasikan ini memiliki 1 data store/penyimpanan data yaitu data parkir kosong yang kemudian data tersebut diinfokan kepada pengguna parkir.

### Gambar Desain Sistem

Berisi desain dari perangkat keras maupun lunak yang akan digunakan dalam melakukan simulasi sistem, penentuan perangkat lunak dan topologi yang akan digunakan,

sekaligus pengaturan perangkat keras tersebut agar sesuai dengan topologi yang diinginkan.



Gambar 3.5 Desain sistem

Gambar 3.5 merupakan desain sistem pencarian tempat parkir kosong, dimisalkan dengan adanya 8 tempat parkir mobil, di masing-masing tempat di pasang satu sensor LDR yang berfungsi mendeteksi adanya mobil parkir atau tidak, setelah dihubungkan langsung ke arduino dan langsung di sambungkan ke perangkat komputer yang berfungsi sebagai display info parkir kosong.

### Flowchart Pengguna Parkir

Berikut ini adalah flowchart pengguna parkir, untuk lebih jelasnya bisa melihat gambar di bawah ini :



Gambar 3.6 Flowchart Pengguna parkir

Pada gambar 3.6 menjelaskan tentang alur pengguna parkir. Pengguna parkir memasuki loket parkir, setelah membayar dan memperoleh tiket parkir, pada tiket parkir telah dicantumkan lokasi lahan parkir kosong, sehingga pengguna parkir dengan mudah langsung menuju tempat parkir kosong dan memarkir mobilnya. Pengguna keluar dan menyerahkan tiket parkir pada loket keluar.

### Flowchart Pencarian Lahan Parkir Kosong

Berikut ini adalah flowchart pencarian lahan parkir kosong, untuk lebih jelasnya bisa melihat gambar di bawah ini :





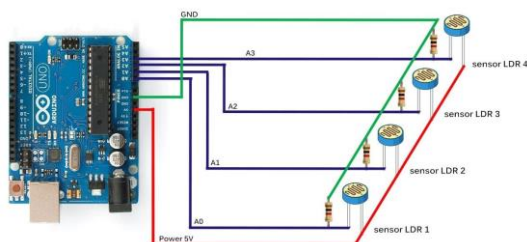
Gambar 3.7 Flowchart Pencarian lahan parkir kosong

Pada gambar 3.7 menjelaskan tentang pencarian lahan parkir kosong. Pada proses pencarian ini pertama pada sensor pada jika intensitas cahaya  $< 50$  maka sensor mendeteksi tempat tersebut gelap yang berarti tempat tersebut telah terisi, dan jika intensitas cahaya  $= 50$  atau  $> 50$  maka sensor mendeteksi tempat tersebut terang yang berarti tempat tersebut masih kosong, selanjutnya sistem secara otomatis melakukan update lahan parkir kosong dan di tampilkan pada display, setelah parkir terisi akan melakukan pengecekan lahan parkir kosong dan secara otomatis melakukan update lahan parkir kosong lagi dan hal tersebut dilakukan secara berulangulng.

## IMPLEMENTASI PERANGKAT LUNAK

### Konfigurasi pada Arduino UNO Hardware

Berikut ini adalah skema rangkaian arduino UNO:



Gambar 4.1 Rangkaian pada Arduino UNO

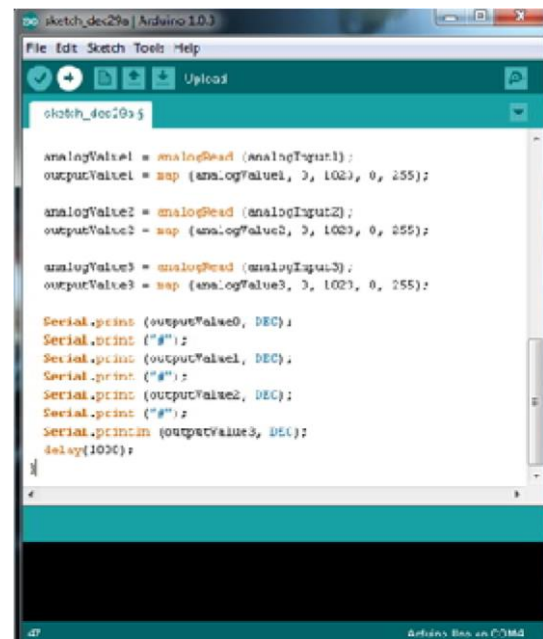
Pada gambar 4.1 merupakan rangkaian arduino UNO yang terdiri dari sensor LDR (4buah) dan resistor 1K ohm (4buah). Untuk satu sensor terdiri dari satu output dan langsung di sambungkan ke arduino pada port analog in, pada

rangkaian ini, hasil output terdapat pada analog in 0, analog in 1, analog in 2, dan analog in 3.

Sebelum mengirimkan output ke analog in kabel disambungkan dengan resistor dan diteruskan ke port GND pada arduino UNO. Dan salah satu kaki dari LDR di sambungkan dengan power arduino UNO.

### Konfigurasi pada Arduino 1.0.3 Software

Berikut ini adalah proses meng-upload *source code* dari arduino *software* ke arduino *hardware* setelah arduino *hardware* dihubungkan dengan PC:



Gambar 4.2 Upload source code Arduino

Pada gambar 4.2 merupakan interface dari Arduino Software yang berguna untuk melakukan *upload* atau transfer perintah-perintah berupa *source code* ke arduino *hardware*. Dengan menekan tombol upload pada arduino *software* (tombol di tunjukan dengan simbol arah panah mengarah ke kanan) maka perintah-perintah atau *source code* tersebut akan tertanam pada arduino UNO hardware.

Dan berikut ini adalah *script* arduino secara keseluruhan.

```

1  const int analogInput0 = A0;
2  const int analogInput1 = A1;
3  const int analogInput2 = A2;
4  const int analogInput3 = A3;
5
6  int analogValue0 = 0;
7  int outputValue0 = 0;
8
9  int analogValue1 = 0;
10 int outputValue1 = 0;
11
12 int analogValue2 = 0;
13 int outputValue2 = 0;
14
15 int analogValue3 = 0;
16 int outputValue3 = 0;
17
18 void setup() {
19   pinMode(analogInput0, INPUT);
20   pinMode(analogInput1, INPUT);
21   pinMode(analogInput2, INPUT);
22   pinMode(analogInput3, INPUT);
23   Serial.begin(9600);
24 }
25
26 void loop() {
27   analogValue0 = analogRead(analogInput0);
28   outputValue0 = map(analogValue0, 0, 1023, 0, 255);
29
30   analogValue1 = analogRead(analogInput1);
31   outputValue1 = map(analogValue1, 0, 1023, 0, 255);
32
33   analogValue2 = analogRead(analogInput2);
34   outputValue2 = map(analogValue2, 0, 1023, 0, 255);
35
36   analogValue3 = analogRead(analogInput3);
37   outputValue3 = map(analogValue3, 0, 1023, 0, 255);
38
39   Serial.print(outputValue0, DEC);
40   Serial.print("#");
41   Serial.print(outputValue1, DEC);
42   Serial.print("#");
43   Serial.print(outputValue2, DEC);
44   Serial.print("#");
45   Serial.println(outputValue3, DEC);
46   delay(1000);
47 }

```

Gambar 4.3 Script Arduino

Pada gambar 4.3 dijelaskan tentang *source code* yang di *upload* ke dalam arduino UNO *hardware*. Pada baris 1 - 4 merupakan deklarasi pin input analog A0 sampai dengan pin input analog A3, pada baris 6 merupakan deklarasi variable *analogValue* dan pada baris 7 merupakan deklarasi ariable *outputValue*, pada baris 19 – 22 pin *analogInput* sebagai input, pada baris 23 merupakan inisialisasi komunikasi serial 9600 bps, pada baris 27 merupakan fungsi dari membaca input analog (*analogRead*) dan pada baris 39 – 45 berfungsi untuk menulis nilai pada serial monitor.

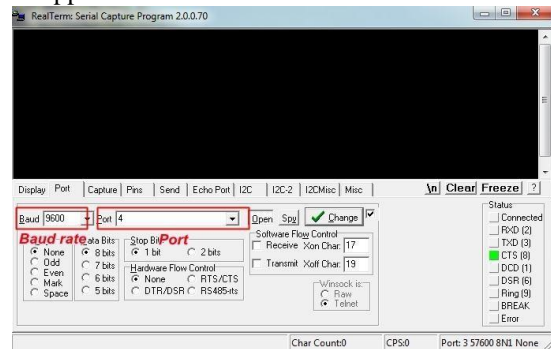
### Konfigurasi Realterm

Membaca serial port dan menampilkan/mengolah pada php di windows tidak bisa langsung melalui script pada php. Pada Windows hal ini sulit dilakukan berikut adalah cara membaca Output data dari arduino melalui jalur serial port dengan menggunakan Realterm, berikut adalah langkah-langkahnya :

1. Jalankan RealTerm, pada port pilih dimana USB to Serial pada Arduino berada.
2. Baud rate dipilih 9600. Lalu klik Open. Nanti di layar monitor keluar angka-angka.
3. Pada Capture pilih File, taruh file txt pada folder dimana file php akan dijalankan.

Misal pada D:\xampp\htdocs\arduino\capture.txt”.

4. Bisa dipilih Start overwrite atau Start append.



Gambar 4.4 Interface RealTerm

5. Copy paste script berikut:

```

1  <?php
2  $line = '';
3
4  $f = fopen('capture.txt', 'r');
5  $cursor = -1;
6
7  fseek($f, $cursor, SEEK_END);
8  $char = fgetc($f);
9
10 /**
11  * Trim trailing newline chars of the file
12  */
13 while ($char == "\n" || $char == "\r") {
14   fseek($f, $cursor--, SEEK_END);
15   $char = fgetc($f);
16 }
17
18 /**
19  * Read until the start of file or first newline char
20  */
21 while ($char != false && $char != "\n" && $char != "\r") {
22   /**
23    * Prepend the new char
24    */
25   $line = $char . $line;
26   fseek($f, $cursor--, SEEK_END);
27   $char = fgetc($f);
28 }
29
30 echo $line;
31 ?>

```

Gambar 4.5 Script untuk membaca port arduino

- 6.



Gambar 4.6 Display Parkir Kosong

Jika overwrite data sebelumnya akan dihapus, Jika append, maka akan ditambahkan pada baris kosong yang digunakan untuk melihat atau mendisplay parkir yang kosong, pada gambar tersebut di jelas kan pada lantai 1 terdapat 2 (dua) tempat parkir dan pada lantai 2 juga terdapat 2 (dua) tempat parkir. Pada text field sebelah kanan digunakan untuk input data parkir yaitu berupa nomor polisi mobil yang parkir. Pada menu bawah ada “parkir kosong” disini menginfokan jumlah keseluruhan parkir yang kosong dan rinciannya ada pada “lantai 1” dan “lantai 2”. Pada kolom pencarian “posisi parkir” tersebut digunakan untuk mencari letak mobil yang parkir jika pengunjung lupa tempat memarkir mobilnya.

7. Simpan sebagai “index.php”.

### Desain Antarmuka

Pada sub-bab desain antarmuka, dijelaskan setiap desain antarmuka yang telah direncanakan sehingga pembaca bias lebih mudah dalam pemahaman menu dan fungsi masing-masing modul yang telah disediakan.



Nomor Polisi	Lantai	Block	Tanggal Parkir	Aksi	Aksi
S 1111 TA	Lantai 1	block 1	2013-01-22	Edit	Hapus
W 4444 TA	Lantai 2	block 1	2013-01-22	Edit	Hapus
S 8989 OK	Lantai 1	block 1	2013-01-24	Edit	Hapus
S 1234 TY	Lantai 1	block 2	2013-01-24	Edit	Hapus
B 777 IK	Lantai 2	block 1	2013-01-24	Edit	Hapus
B 677 R	Lantai 2	block 2	2013-01-24	Edit	Hapus
S 88 WR	Lantai 1	block 1	2013-01-24	Edit	Hapus
DK 909 K	Lantai 2	block 2	2013-01-24	Edit	Hapus
R 789 OP	Lantai 2	block 1	2013-01-24	Edit	Hapus
S 1236 LA	Lantai 1	block 1	2013-01-25	Edit	Hapus
L 4545 KS	Lantai 1	block 2	2013-01-25	Edit	Hapus
S 2903 JC	Lantai 2	block 1	2013-01-25	Edit	Hapus
L 2957 JH	Lantai 2	block 2	2013-01-25	Edit	Hapus
L 5679 SL	Lantai 1	block 1	2013-01-25	Edit	Hapus
K 9867 SK	Lantai 2	block 2	2013-01-25	Edit	Hapus
L 3456 LH	Lantai 1	block 2	2013-01-25	Edit	Hapus

Gambar 4.7 Form Data Parkir

Pada gambar 4.7 merupakan data parkir yang digunakan untuk menampilkan data mobil yang parkir.

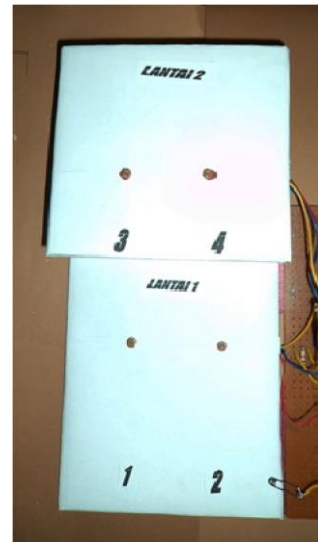
## HASIL DAN PEMBAHASAN

### Uji Coba Sistem

Pada uji coba Sistem Pencarian Parkir Kosong Pada Lahan Parkir Khusus Mobil dilakukan beberapa pengujian sistem, setelah melalui tahap analisa dan perancangan sistem. Hasil uji coba digunakan untuk mengetahui apakah sistem dapat memenuhi tujuan yang akan dicapai seperti yang telah dijelaskan pada Bab Pendahuluan.

### Tempat Parkir

Pada uji coba kali ini pada tempat parkir kita hanya membuat simulasi dengan membuat miniatur tempat parkir mobil yang hanya berkapasitas untuk 2(dua) lantai dan 4 (empat) mobil dengan rincian 1(satu) lantai parkir mobil berisi 2(dua) mobil saja, dan berikut adalah gambar miniatur tempat parkir mobil.



Gambar 5.1 Miniatur tempat parkir mobil

Pada gambar 5.1 miniatur parkir tersebut menempatkan sensor LDR yang berjumlah 4(empat) dan setiap lantai dipasang 2(dua) sensor LDR, dan berikut adalah gambar sensor LDR yang terpasang pada miniatur tempat parkir mobil.

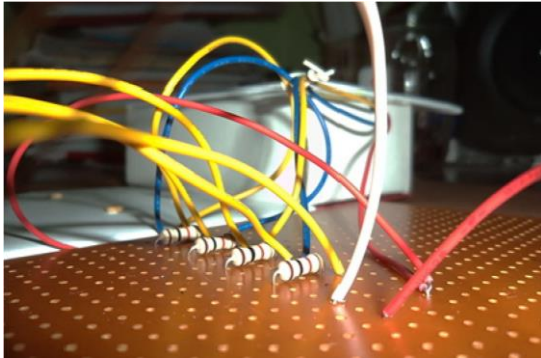


Gambar 5.2 Sensor LDR pada miniatur parkir mobil

Pada gambar 5.2 ini untuk setiap satu tempat parkir mobil dipasang satu sensor LDR guna mendeteksi ada tidaknya mobil yang parkir pada tempat parkir tersebut dengan menggunakan intensitas besarnya cahaya yang di tangkap oleh sensor LDR.

### Rangkaian pada Arduino

Rangkaian arduino dalam proses pencarian parkir kosong ini kami menggunakan papan PCB, 4 (empat) buah resistor dengan nilai sebuah resistor 10 K $\Omega$  dan menggunakan beberapa kabel kecil yang berfungsi sebagai penghubung, dan berikut adalah gambar rangkaian pada arduino.

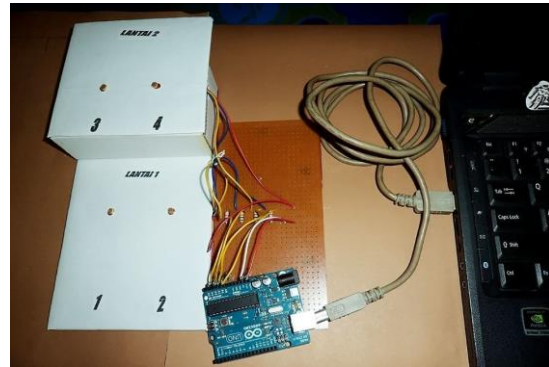


Gambar 5.3 Rangkaian pada Arduino

Dalam gambar 5.3 alurnya adalah satu sensor LDR mempunyai 2(dua) kaki, kaki yang pertama langsung dihubungkan dengan port power arduino sebesar 5V, dan pada kaki kedua ada percabangan, cabang pertama dihubungkan dengan resistor yang dengan nilai besaran resistor 10 K $\Omega$  dan dari resistor langsung diteruskan ke port GND pada arduino, sedangkan untuk percabangan kedua langsung dihubungkan dengan port analog in pada arduino yang berfungsi menghasilkan output. Untuk port analog in pada arduino UNO hardware itu sendiri ada 6 port, diantaranya terdiri dari port analog in A0, port analog in A1, port analog in A2, port analog in A3, port analog in A4, dan port analog in A5.

### Pencarian Parkir Kosong Mobil

Pencarian ini dilakukan jika terjadi adanya perubahan intensitas cahaya pada sensor LDR sehingga LDR mengirimkan output ke arduino dari arduino di-display melalui PC atau perangkat computer, dan berikut adalah gambar dari miniatur lantai, rangkaian arduino dan PC atau perangkat komputer.



Gambar 5.4 Sistem Pencarian Parkir Kosong

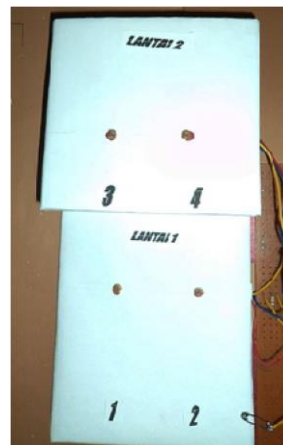
Pada gambar 5.4 semua sistem telah disambungkan, mulai dari miniatur parkir mobil, rangkaian arduino hingga ke PC atau perangkat komputer yang berfungsi sebagai *display*.

### Hasil Uji Coba Sistem

Berikut ini adalah hasil uji coba sistem yang telah kami lakukan berulang-ulang dengan berbagai variasi uji coba.

### Kondisi Parkir Kosong

Pada saat kondisi parkir kosong dengan kondisi tak ada satupun mobil yang parkir maka pada display tidak menandakan bahwa parkir tersebut terisi.



Gambar 5.5 Kondisi Parkir Dalam keadaan Kosong



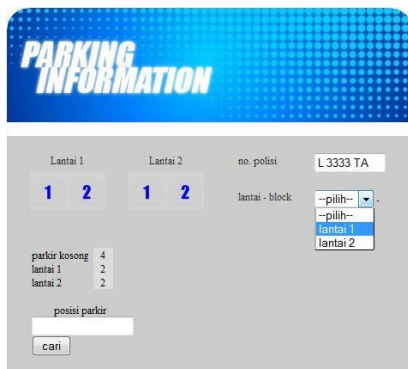


Gambar 5.6 Display Parkir Ketika Parkir Kosong

Pada gambar 5.5 nampak kondisi kosong tak terisi pada lahan parkir dan pada gambar 5.6 display parkir ketika lahan parkir kosong tak terisi dan nampak pada jumlah parkir kosong masih berjumlah 4(empat) dengan rincian lantai 1(satu) berjumlah 2(dua) tempat parkir kosong dan lantai 2(dua) berjumlah 2(dua) tempat parkir kosong.

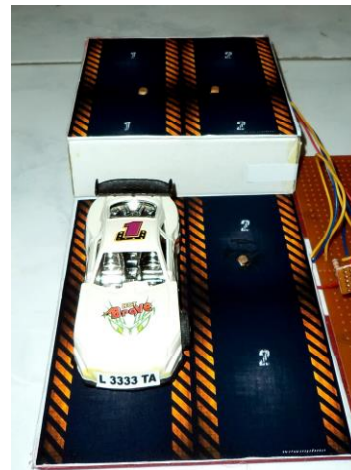
#### Kondisi 1 (satu) Parkir Terisi

Pada saat kondisi salah 1 (satu) tempat parkir terisi, di mulai dari tempat loket parkir masuk. Terdapat satu mobil yang akan parkir dengan nomor polisi L 3333 TA, data nomor polisi tersebut di-input-kan pada form no. polisi dan pada lantai-block dipilih pada lantai 1(satu) block 1(satu).



Gambar 5.7 Input nomor polisi dan posisi parkir

Pada gambar 5.7 adalah proses menginputkan nomor polisi mobil dan menentukan tempat parkir, setelah nomor polisi mobil dan tempat parkir kosong di inputkan kemudian diberi tiket parkir yang berisi tempat parkir kosong pada lantai 1(satu) block 1(satu), setelah itu mobil menuju tempat parkir sesuai dengan lokasi parkir yang tertera pada loket parkir. Dan mobil telah sampai pada lokasi tempat parkir lantai 1(satu) block 1(satu).



Gambar 5.8 Parkir lantai 1 block 1 terisi Pada gambar 5.8 mobil dengan nopol L 3333 TA telah menempati parkir pada lantai 1(satu) block 1(satu), secara otomatis sensor LDR mendeteksi parkir tersebut telah terisi dan pada *display* lantai 1(satu) block 1(satu) menandakan terisi.



Gambar 5.9 Display parkir terisi

Pada gambar 5.9 menunjuk parkir pada lantai 1(satu) block 1(satu) telah terisi dengan menandakan warna menjadi biru dan pada jumlah parkir kosong menjadi 3(tiga) dengan rincian sisa parkir kosong pada lantai 1(satu) sisa 1(satu) parkir kosong sedangkan untuk lantai 2(dua) jumlah parkir kosong 2(dua).

#### Kondisi 2 (dua) Parkir Terisi

Pada saat kondisi 2 (dua) tempat parkir terisi, dengan catatan satu tempat parkir telah terisi terlebih dahulu, di mulai dari tempat loket parkir masuk. Terdapat satu mobil yang akan parkir dengan nomor polisi B 2222 TA, data nomor polisi tersebut di-input-kan pada form no. polisi dan pada lantai-block dipilih pada lantai 1(satu) block 2(dua).

The screenshot shows the 'PARKING INFORMATION' system interface. It has a blue header with the title. Below the header, there are input fields for 'Lantai 1', 'Lantai 2', 'no. polisi', and 'lantai - block'. The 'no. polisi' field contains 'B 2222 TA'. The 'lantai - block' dropdown menu is open, showing options 'lantai 1', 'lantai 2', and 'lantai 3'. Below these fields, there is a table showing the number of empty parking spaces: 'parkir kosong' (3), 'lantai 1' (1), and 'lantai 2' (2). There is also a 'posisi parkir' field and a 'cari' button.

Gambar 5.10 Input nomor polisi dan posisi parkir

Pada gambar 5.10 adalah proses menginputkan nomor polisi mobil dan menentukan tempat parkir, setelah nomor polisi mobil dan tempat parkir kosong di inputkan kemudian diberi tiket parkir yang berisi tempat parkir kosong pada lantai 1(satu) block 2(dua), setelah itu mobil menuju tempat parkir sesuai dengan lokasi parkir yang tertera pada loket parkir. Dan mobil telah sampai pada lokasi tempat parkir lantai 1(satu) block 2(dua).



Gambar 5.11 Parkir lantai 1 block 2 terisi

Pada gambar 5.11 mobil dengan nopol B 2222 TA telah menempati parkir pada lantai 1(satu) block 2(dua), secara otomatis sensor LDR mendeteksi parkir tersebut telah terisi dan pada display lantai 1(satu) block 1(satu) dan block 2(dua) menandakan terisi.

The screenshot shows the 'PARKING INFORMATION' system interface. It has a blue header with the title. Below the header, there are input fields for 'Lantai 1', 'Lantai 2', 'no. polisi', and 'lantai - block'. The 'no. polisi' field is empty. The 'lantai - block' dropdown menu is open, showing options 'lantai 1', 'lantai 2', and 'lantai 3'. Below these fields, there is a table showing the number of empty parking spaces: 'parkir kosong' (2), 'lantai 1' (0), and 'lantai 2' (2). There is also a 'posisi parkir' field and a 'cari' button.

Gambar 5.12 Display parkir terisi

Pada gambar 5.12 menunjuk parkir pada lantai 1(satu) block 1(satu) dan block 2(dua) telah terisi dengan menandakan warna menjadi biru dan pada jumlah parkir kosong menjadi 2(dua) dengan rincian sisa parkir kosong pada lantai 1(satu) adalah 0(nol) sedangkan untuk lantai 2(dua) jumlah parkir kosong 2(dua).

### Kondisi 3 (tiga) Parkir Terisi

Pada saat kondisi 3 (tiga) tempat parkir terisi, dengan catatan 2(dua) tempat parkir telah terisi terlebih dahulu, di mulai dari tempat loket parkir masuk. Terdapat satu mobil yang akan parkir dengan nomor polisi S 1111 TA, data nomor polisi tersebut di-inputkan pada form no. polisi dan pada lantai-block dipilih pada lantai 2(dua) block 1(satu).

The screenshot shows the 'PARKING INFORMATION' system interface. It has a blue header with the title. Below the header, there are input fields for 'Lantai 1', 'Lantai 2', 'no. polisi', and 'lantai - block'. The 'no. polisi' field contains 'S 1111 TA'. The 'lantai - block' dropdown menu is open, showing options 'lantai 1', 'lantai 2', and 'lantai 3'. Below these fields, there is a table showing the number of empty parking spaces: 'parkir kosong' (2), 'lantai 1' (0), and 'lantai 2' (2). There is also a 'posisi parkir' field and a 'cari' button.

Gambar 5.13 Input nomor polisi dan posisi parkir

Pada gambar 5.13 adalah proses menginputkan nomor polisi mobil dan menentukan tempat parkir, setelah nomor polisi mobil dan tempat parkir kosong di inputkan kemudian diberi tiket parkir yang berisi tempat parkir kosong pada lantai 2(dua) block 1(satu), setelah itu mobil menuju tempat parkir sesuai dengan lokasi parkir yang tertera pada loket parkir. Dan mobil telah sampai



pada lokasi tempat parkir lantai 2(dua) block 1(satu).



Gambar 5.14 Parkir lantai 2 block 1 terisi

Pada gambar 5.14 mobil dengan nopol S 1111 TA telah menempati parkir pada lantai 2(dua) block 1(satu), secara otomatis sensor LDR mendeteksi parkir tersebut telah terisi dan pada *display* lantai 1(satu) block 1(satu), block 2(dua) dan lantai 2(dua) block 1(satu) menandakan terisi.



Gambar 5.15 Display parkir terisi

Pada gambar 5.15 menunjuk parkir pada lantai 1(satu) block 1(satu), block 2(dua) dan lantai 2(dua) block 1(satu) telah terisi dengan menandakan warna menjadi biru dan pada jumlah parkir kosong menjadi 1(satu) dengan rincian sisa parkir kosong pada lantai 1(satu) adalah 0(nol) sedangkan untuk lantai 2(dua) jumlah parkir kosong 1(satu).

#### Kondisi 4 (empat) Parkir Terisi

Pada saat kondisi 4(empat) tempat parkir terisi, dengan catatan 3(tiga) tempat parkir telah terisi terlebih dahulu, di mulai dari tempat loket parkir masuk. Terdapat satu mobil yang akan parkir dengan nomor polisi W 4444 TA, data

nomor polisi tersebut di-input-kan pada form no. polisi dan pada lantai-block dipilih pada lantai 2(dua) block 2(dua).



Gambar 5.16 Input nomor polisi dan posisi parkir

Pada gambar 5.16 adalah proses menginputkan nomor polisi mobil dan menentukan tempat parkir, setelah nomor polisi mobil dan tempat parkir kosong di inputkan kemudian diberi tiket parkir yang berisi tempat parkir kosong pada lantai 2(dua) block 2(dua), setelah itu mobil menuju tempat parkir sesuai dengan lokasi parkir yang tertera pada loket parkir. Dan mobil telah sampai pada lokasi tempat parkir lantai 2(dua) block 2(dua).



Gambar 5.17 Parkir lantai 2 block 2 terisi

Pada gambar 5.17 mobil dengan nopol W 4444 TA telah menempati parkir pada lantai 2(dua) block 2(dua), secara otomatis sensor LDR mendeteksi parkir tersebut telah terisi dan pada *display* lantai 1(satu) block 1(satu), block 2(dua) dan lantai 2(dua) block 1(satu), block 2(dua) menandakan terisi.

Gambar 5.18 Display parkir terisi

Pada gambar 5.18 menunjuk parkir pada lantai 1(satu) block 1(satu), block 2(dua) dan lantai 2(dua) block 1(satu) block 2(dua) telah terisi dengan menandakan warna menjadi biru dan pada jumlah parkir kosong menjadi 0(nol).

### Mencari Tempat Posisi Mobil Parkir

Ketika pengunjung lupa dimana tempat memarkir mobilnya, pada halaman *display* parkir kosong menyediakan form cari yang berfungsi untuk mencari posisi mobil parkir dengan cara menginputkan nomor polisi. Dimisalkan pengguna parkir ingin mengetahui posisi mobil yang parkir dengan nomor polisi mobil B 2222 TA.

Gambar 5.19 Pencarian Posisi Parkir

Pada Gambar 5. 19 merupakan pencarian posisi parkir dengan menggunakan nomor polisi pada mobil yang telah parkir. Dan berikut adalah hasil pencarian.

Nomor Polisi	Lantai	Block
B 2222 TA	lantai 1	block 2

Gambar 5.20 Hasil Pencarian Posisi Parkir

Pada gambar 5.20 merupakan hasil pencarian posisi parkir mobil pada mobil dengan

nomor polisi 2222 TA menempati parkir pada lantai 1(satu) block 2(dua).

### Merubah Manual Posisi Tempat Parkir Mobil

Keadaan ini dimana ketika pengunjung parkir memarkir mobilnya tidak sesuai dengan lokasi parkir yang di cantumkan di loket parkir, sehingga data yang diinputkan oleh petugas parkir harus dirubah/di-*edit*. Sebagai contoh, ada mobil dengan nomor polisi L 3456 LH yang seharusnya parkir pada lantai 1(satu) block 2(dua), tetapi dalam kenyataan di lapangan parkir pada lantai 2(dua) block 1(satu), dan disini petugas parkir perlu merubah data yang sebelumnya telah te-*input*.

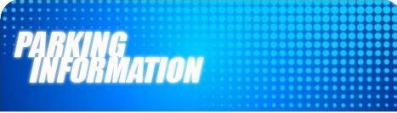
Nomor Polisi	Lantai	Block	Tanggal Parkir	Aksi	Aksi
S 1111 TA	lantai 1	block 1	2013-01-22	Edit	Hapus
W 4444 TA	lantai 2	block 1	2013-01-22	Edit	Hapus
S 8989 OK	lantai 1	block 1	2013-01-24	Edit	Hapus
S 1234 TY	lantai 1	block 2	2013-01-24	Edit	Hapus
B 777 IK	lantai 2	block 1	2013-01-24	Edit	Hapus
B 677 R	lantai 2	block 2	2013-01-24	Edit	Hapus
S 88 WR	lantai 1	block 1	2013-01-24	Edit	Hapus
DK 909 K	lantai 2	block 2	2013-01-24	Edit	Hapus
R 789 OP	lantai 2	block 1	2013-01-24	Edit	Hapus
S 1236 LA	lantai 1	block 1	2013-01-25	Edit	Hapus
L 4545 KS	lantai 1	block 2	2013-01-25	Edit	Hapus
S 2903 JC	lantai 2	block 1	2013-01-25	Edit	Hapus
L 2957 JH	lantai 2	block 2	2013-01-25	Edit	Hapus
L 5679 SL	lantai 1	block 1	2013-01-25	Edit	Hapus
K 9867 SK	lantai 2	block 2	2013-01-25	Edit	Hapus
L 3456 LH	lantai 1	block 2	2013-01-25	Edit	Hapus

Gambar 5.21 Data Parkir Mobil

Pada gambar 5.21 merupakan data parkir, dengan tujuan untuk merubah tempat lokasi parkir pada mobil dengan nomor polisi L 3456 LH, yang sebelumnya lantai 1 (satu) block 2(dua) menjadi lantai 2(dua) block 1(satu) dengan cara mengklik tulisan edit pada menu aksi.

Gambar 5.22 Halaman untuk merubah lokasi tempat parkir

Pada gambar 5.22 merupakan halaman untuk merubah lokasi tempat parkir, yang sebelumnya pada lantai 1 (satu) block 2(dua) menjadi lantai 2(dua) block 1(satu), kemudian klik tombol simpan.

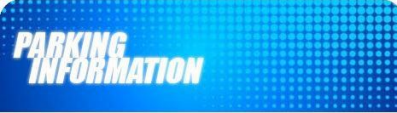


Nomor Polisi	Lantai	Block	Tanggal Parkir	Aksi	Aksi
S 1111 TA	lantai 1	block 1	2013-01-22	Edit	Hapus
W 4444 TA	lantai 2	block 1	2013-01-22	Edit	Hapus
S 8989 OK	lantai 1	block 1	2013-01-24	Edit	Hapus
S 1234 TY	lantai 1	block 2	2013-01-24	Edit	Hapus
B 777 IK	lantai 2	block 1	2013-01-24	Edit	Hapus
B 677 R	lantai 2	block 2	2013-01-24	Edit	Hapus
S 88 WR	lantai 1	block 1	2013-01-24	Edit	Hapus
DK 909 K	lantai 2	block 2	2013-01-24	Edit	Hapus
R 789 OP	lantai 2	block 1	2013-01-24	Edit	Hapus
S 1236 LA	lantai 1	block 1	2013-01-25	Edit	Hapus
L 4545 KS	lantai 1	block 2	2013-01-25	Edit	Hapus
S 2903 JC	lantai 2	block 1	2013-01-25	Edit	Hapus
L 2957 JH	lantai 2	block 2	2013-01-25	Edit	Hapus
L 5679 SL	lantai 1	block 1	2013-01-25	Edit	Hapus
K 8667 SK	lantai 2	block 2	2013-01-25	Edit	Hapus
L 3456 LH	lantai 2	block 1	2013-01-25	Edit	Hapus

Gambar 5.23 Data parkir setelah lokasi tempat parkir dirubah

Pada gambar 5.23 merupakan data parkir setelah mobil dengan nomor polisi L 3456 LH yang sebelumnya berada pada lantai 1 (satu) block 2(dua) menjadi lantai 2(dua) block 1(satu).

#### Report Parkir Mobil

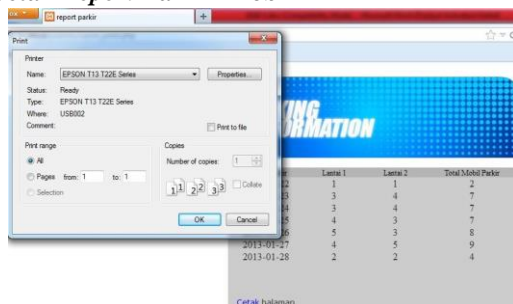


Tanggal Parkir	Lantai 1	Lantai 2	Total Mobil Parkir
2013-01-22	1	1	2
2013-01-23	3	4	7
2013-01-24	3	4	7
2013-01-25	4	3	7
2013-01-26	5	3	8
2013-01-27	4	5	9
2013-01-28	2	2	4

Gambar 5.24 Report parkir

Pada gambar 5.24 ini adalah report parkir yang berfungsi untuk mengetahui jumlah banyaknya mobil yang parkir setiap harinya.

#### Cetak Report Parkir Mobil



Gambar 5.25 Print report parkir

Pada gambar 5.25 ini adalah menu print report parkir yang berfungsi untuk mencetak halaman report parkir.

PARKING INFORMATION			
2012-01-22	1	1	2
2012-01-23	3	4	7
2012-01-24	3	4	7
2012-01-25	4	3	7
2012-01-26	5	3	8
2012-01-27	4	5	9
2012-01-28	2	2	4

Gambar 5.26 Hasil Cetak report parkir

Gambar 5.26 Cetak report parkir menunjukkan hasil cetak report parkir yang sesuai dengan tanggal parkir.

## PENUTUP

### Kesimpulan

Setelah dilakukan pengujian dan analisa program, maka dapat diperoleh kesimpulan sebagai berikut :

1. Sistem dapat melakukan pencarian parkir kosong dan memberikan informasi kepada pengunjung tentang adanya tempat parkir kosong.
2. Sistem dapat memberikan informasi mengetahui tata letak tempat parkir kosong tanpa harus memutar dari tiap-tiap lantai, dan melakukan pencarian letak dimana mobil tersebut di parkir.

### Saran

Dari beberapa simpulan yang telah diambil, maka dapat dikemukakan saran-saran yang akan sangat membantu untuk pengembangan sistem pencarian parkir kosong ini adalah :

1. Menambahkan sistem kontrol otomatis dalam sistem ini karena dalam sistem ini sistem kontrolnya masih manual.
2. Lebih dikembangkan lagi untuk penentuan posisi parkirnya setelah mobil terparkir dan sensor mendeteksi adanya mobil parkir.

## DAFTAR PUSTAKA

- [1] Artanto, Dian (2012). Interaksi Arduino dan LabVIEW. Jakarta: Elex Media Komputindo.
- [2] Celulux (2011). Rangkaian LDR dengan Arduino. From <http://cicink.wordpress.com/2011/10/05/rangkaian-ldr-dengan-arduino/#more-366>, 20 Oktober 2012

- [3] Himatekom. Pedoman-pedoman Dalam Membuat Flowchart. From <http://www.scribd.com/doc/11592833/PedomanpedomanDalamMembuatFlowchart>, 10 Agustus 2012
- [4] Kendall E, Kendall J. (2007). Analisis dan Perancangan Sistem. Klaten: PT Indeks. Klaten.
- [5] Orion. About Led. From <http://www.orionled.com/index/about>, 3 Juli 2012
- [6] RiderSystem. LDR (Light Dependent Resistor). From <http://www.rider-system.net/2011/10/ldr-light-dependentsistor.html>, 3 Juli 2012
- [7] Sulaiman, Arif (2012). Arduino. From <http://buletin.Balai elektronik .com/?p=163>, 16 Juli 2012

(9)

## SISTEM MANAJEMEN AKUN DENGAN MENGGUNAKAN OPENLDAP

Windyarto Adi Candra<sup>1</sup>, Tjatusari W.<sup>2</sup>, Beny Y.V. Nasution<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Wijaya Kusuma Surabaya  
Jl. Dukuh Kupang XXV/ 54 Surabaya 60225  
Email: [windyartoadi@gmail.com](mailto:windyartoadi@gmail.com)

### ABSTRAK

Sistem manajemen akun ini digunakan untuk memudahkan user dalam mengakses aplikasi yang berada dalam sebuah jaringan lokal, dengan menggunakan satu account saja. Diharapkan sistem ini dapat memberikan kemudahan kepada user dalam mengakses aplikasi dalam jaringan komputer, dengan menggunakan satu account. Kombinasi antara username dan password pada setiap account yang dimiliki oleh user berbeda, sehingga menyebabkan user kesulitan masuk ke dalam account yang dia miliki.

Dengan adanya sistem ini yang menerapkan metode *Single Sign On (SSO)* dalam OpenLDAP (*Lightweight Directory Access Protocol*), bertujuan agar dapat mempermudah user dalam mengakses berbagai aplikasi – aplikasi dalam sebuah jaringan komputer dengan menggunakan satu account

**Kata kunci :** OpenLDAP, Single Sign On (SSO)

### PENDAHULUAN

#### Latar Belakang

Akun atau yang biasa disebut account merupakan identitas yang dimiliki oleh user atau pengguna yang terdaftar dalam internet, maupun dalam sebuah jaringan komputer. Fungsi account adalah untuk memberi hak akses kepada user dalam mengakses sumber daya yang ada dalam sebuah aplikasi. *Account – account* itu biasanya terdapat dalam email, *social network*, aplikasi dalam jaringan komputer, dan lain – lain. Untuk menggunakan account, user diminta memasukkan username dan password untuk proses verifikasi.

Proses ini dibutuhkan, untuk menghindari penyalahgunaan account. Setelah proses berhasil, user bisa mengakses sumber daya yang dibutuhkan.

User biasanya memiliki lebih dari satu account. Dalam setiap account, ada kalanya user memiliki kombinasi username dan password yang berbeda. Hal inilah yang menyebabkan user lupa kombinasi username dan password pada account yang dimiliki, untuk mengakses sumber daya yang diperlukan.

Dari permasalahan tersebut, pada penelitian ini akan membahas sebuah sistem yang dapat membantu manajemen account yang dimiliki user, di mana sistem ini akan memudahkan user masuk ke dalam aplikasi dalam jaringan lokal. Tanpa harus melakukan login ulang ketika

mengakses berbagai macam aplikasi dalam jaringan lokal.

#### Tujuan

Adapun tujuan dalam penelitian adalah Menyediakan akses untuk user, masuk ke dalam berbagai aplikasi dalam jaringan local melalui autentikasi account tunggal.

### METODOLOGI PENELITIAN

Dalam metode penelitian ini menjelaskan beberapa teknik pengerjaan Tugas Akhir. Metode penelitian yang digunakan adalah sebagai berikut :

#### 1. Studi Literatur

Sebelum melaksanakan penelitian dan melakukan implementasi, terlebih dulu melakukan studi dan mencari referensi tentang metode *Single Sign On (SSO)* dengan menggunakan OpenLDAP dalam sebuah jaringan komputer.

#### 2. Analisa Sistem

Pada tahap ini adalah menganalisa cara kerja atau alur dari metode SSO yang diterapkan pada aplikasi-aplikasi yang berada pada sebuah jaringan komputer dan terhubung dengan server OpenLDAP.



### 3. Perancangan

Pada tahap ini dilakukan perancangan desain Sistem Manajemen Akun Dengan Menggunakan OpenLDAP sesuai dengan konsep dan tahapan dalam membangun sebuah jaringan.

### 4. Implementasi

Pada tahap ini dilakukan implementasi metode SSO dalam sebuah jaringan komputer dengan menggunakan OpenLDAP. Sehingga bisa sesuai dengan apa yang diinginkan.

### 5. Uji Coba

Tahap ini akan dilakukan proses uji coba Sistem Manajemen Akun supaya bisa berjalan sesuai dengan konsep yang diinginkan. Sehingga dapat diketahui hasil dari uji coba yang telah dilakukan.

## TINJAUAN PUSTAKA

### A. Metode Single Sign On (SSO)

Teknologi *Single Sign On* (sering disingkat menjadi SSO) adalah teknologi yang mengizinkan pengguna jaringan agar dapat mengakses sumber daya dalam jaringan hanya dengan menggunakan satu akun pengguna saja.

Teknologi ini sangat diminati, khususnya dalam jaringan yang sangat besar dan bersifat heterogen (di saat sistem operasi serta aplikasi yang digunakan oleh komputer adalah berasal dari banyak vendor, dan pengguna dimintai untuk mengisi informasi dirinya ke dalam setiap platform yang berbeda tersebut yang hendak diakses oleh pengguna). Dengan menggunakan SSO, seorang pengguna hanya cukup melakukan proses autentikasi sekali saja untuk mendapatkan izin akses terhadap semua layanan yang terdapat di dalam jaringan.

Selain mendatangkan manfaat, SSO juga dapat mendatangkan kerugian. Dari cara pandang seperti ini, beberapa pengamat memperkirakan bahwa penggunaan SSO dapat menghemat biaya untuk memelihara password yang rumit yang dapat mencapai ratusan dolar setiap pengguna tiap tahun. Tetapi, implementasi SSO dalam sebuah jaringan yang heterogen adalah rumit, sehingga banyak administrator jaringan kurang begitu giat dalam mengimplementasikannya.

Contoh dari sistem SSO adalah protokol Kerberos, yang telah dimasukkan ke dalam sistem operasi Windows 2000 ke atas. Protokol yang sama dapat juga digunakan di dalam keluarga sistem

operasi UNIX. Novell juga telah menawarkan fungsi SSO miliknya sendiri, yang disebut sebagai Novell *Single Sign On* (NSSO) yang dapat digunakan dalam lingkungan Windows/NetWare. Beberapa perusahaan, seperti *Entrust Technologies* dan *RSA Security* menawarkan fungsi SSO yang berbasis kriptografi kunci publik.

### B. Active Directory (AD)

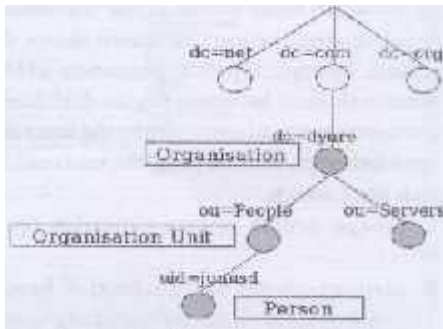
AD adalah implementasi LDAP directory services oleh Microsoft, dan digunakan dalam lingkungan Microsoft. Tujuannya adalah untuk menyediakan servis authentication dan otorisasi secara terpusat untuk komputer berbasis Windows. AD memuat informasi mengenai komputer, user, grup, printer dan network resources lainnya yang bisa diakses oleh user dan aplikasi. AD menyediakan cara konsisten untuk penamaan, pencarian, akses, pengelolaan, dan keamanan untuk objek-objek tersebut.

### C. Lightweight Directory Active Protocol (LDAP)

LDAP merupakan singkatan dari Lightweight Directory Access Protocol (Protokol Akses Direktori Ringan). Artinya, ini adalah protokol kelas ringan untuk mengakses servis direktori, yang berdasarkan pada protokol servis direktori X.500. LDAP Berjalan melalui protokol TCP/IP. Model informasi LDAP adalah berdasarkan entri. Sebuah entri adalah koleksi atribut yang mempunyai nama yang terbedakan (*Distinguished Name/DN*) secara global. DN ini digunakan sebagai referensi ke entri yang secara unik berbeda dengan nilai DN yang lainnya. Setiap atribut entri mempunyai sebuah tipe dengan satu nilai atau lebih. Tipe biasanya string singkatan khusus, seperti atau "mail" atau "cn" untuk *common name*, untuk alamat e-mail.

Sintaks dari nilai bergantung kepada tipe atribut. Contoh, atribut cn mungkin berisi kata-kata "Junus Djunawidjaja". Atribut mail mungkin berisi alamat email "junus@dynre.com". Atribut jpegPhoto mungkin terdiri sebuah foto dalam format binary JPEG.





Gambar 1 Pohon Direktori LDAP.

Dalam LDAP entri direktori disusun dalam sebuah hirarki struktur seperti pohon (*tree*). Struktur pohon LDAP pada umumnya saat sekarang ini berdasarkan nama domain internet. Pendekatan penamaan servis direktori mirip dengan penamaan pada DNS ini yang paling populer. Gambar 1 menunjukkan sebuah contoh pohon direktori LDAP menggunakan penamaan berdasarkan domain.

Selain secara struktur pohon dengan penamaan internet, juga dapat berupa struktur pohon dengan cara penamaan tradisional. Struktur ini merefleksikan geografis atau lingkup organisasi. Entri-entri mewakili negara-negara, terlihat di atas dari pohon (*tree*). Di bawah mereka adalah entri yang menyatakan provinsi dan organisasi nasional. Di bawah nya lagi mungkin entri yang menyatakan unit organisasi, orang, printer, dokumen, dan lainlain.

LDAP dapat mengontrol atribut-atribut yang diperlukan dan diizinkan dalam sebuah entri, melalui penggunaan atribut special yang dinamakan *objectClass*. Angka-angka dari atribut *objectClass* menyatakan aturan (*rule*) schema yang ditaati oleh entri.

Sebuah entri direferensi oleh nama yang berbeda dari yang lain, yang dibentuk dari nama entri itu sendiri. Ini dinamakan nama relative yang membedakan (*Relative Distinguished Name/RDN*) dan menggabungkan nama-nama dari entri-entri di atasnya atau sebelumnya. Sebagai contoh, entri untuk "Junus Djunawidjaja" pada penamaan berdasarkan Internet contoh di atas mempunyai RDN: uid=junusd dan DN dari uid=junusd, ou=People,dc=dynre,dc=com.

#### D. OPENLDAP

OpenLDAP adalah software *open source* yang diimplementasikan dari *Lightweight Directory Access Protocol* (LDAP) yang dikembangkan oleh OpenLDAP Project. Project ini dimulai pada tahun 1998 oleh Kurt Zeilenga. *Project* dimulai dengan

menggandakan LDAP *source* di University of Michigan dengan bantuan dari pengembang dan evaluator dari LDAP protokol.

OpenLDAP mempunyai daemon slapd dan slurpd. Daemon slapd yang berdiri sendiri dapat dilihat sebagai sebuah servis direktori X.500 kelas ringan. Ini tidak mengimplementasi Protokol Akses Direktori kelas berat X.500. Sebagai direktori *server* kelas ringan, slapd hanya mengimplementasi sebagian kecil dari model X.500.

Sedangkan daemon slurpd digunakan mereplikasi informasi direktori dari daemon slapd. *Client* tidak dapat mengupdate informasi direktori yang ada pada slurpd secara langsung. slurpd akan mereferensi ke slapd jika ada permintaan untuk update informasi. Guna dari replikasi direktori menggunakan slurpd untuk memperingan beban pada daemon utama slapd, serta untuk redundansi pada saat slapd tidak berjalan. Jadi, mesin-mesin *client* akan tetap dapat mengakses informasi direktori melalui slurpd.

#### E. POSTFIX

Postfix adalah *mail transfer agent* bebas dan terbuka. Postfix merupakan *mail transfer agent default* untuk sejumlah sistem operasi bertipe Unix. Postfix didistribusikan menggunakan Lisensi Umum IBM 1.0 yang merupakan lisensi perangkat lunak bebas tetapi tidak kompatibel dengan GPL.

Postfix adalah program alternatif bagi program *Sendmail* yang telah banyak digunakan. Postfix mencoba untuk menjadi cepat, mudah dikelola, dan diharapkan aman, dan dalam waktu yang bersamaan dapat menjadi program yang *sendmail compatible* sehingga tidak mengecewakan penggunaanya.

#### F. SQUID

Squid adalah sebuah daemon yang digunakan sebagai *proxy server* dan *web cache*. Squid memiliki banyak jenis penggunaan, mulai dari mempercepat *server web* dengan melakukan *caching* permintaan yang berulang-ulang, *caching* DNS, *caching* situs web, dan *caching* pencarian komputer di dalam jaringan untuk sekelompok komputer yang menggunakan sumber daya jaringan yang sama, hingga pada membantu keamanan dengan cara melakukan penyaringan (*filter*) lalu lintas. Meskipun seringnya digunakan untuk protokol HTTP dan FTP, Squid juga menawarkan dukungan terbatas untuk beberapa protokol lainnya termasuk *Transport Layer Security* (TLS), *Secure Socket Layer* (SSL), *Internet Gopher*, dan HTTPS.

### Personal Home Page (PHP)

PHP: *Hypertext Preprocessor* adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk memprogram situs web dinamis. PHP dapat digunakan untuk membangun sebuah CMS. Berikut contoh program php :

```
<?php echo "Hello  
World"  
?>
```

### MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus di mana penggunaannya tidak cocok dengan penggunaan GPL.

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (*Structured Query Language*).

### SAMBA

Samba adalah program yang dapat menjembatani kompleksitas berbagai platform system operasi Linux(UNIX) dengan mesin Windows yang dijalankan dalam suatu jaringan komputer. Samba merupakan aplikasi dari UNIX dan Linux, yang dikenal dengan SMB(Service Message Block) protocol. Banyak sistem operasi seperti Windows dan OS/2 yang menggunakan SMB untuk menciptakan jaringan *client/server*. Protokol Samba memungkinkan *server* Linux/UNIX untuk berkomunikasi dengan mesin *client* yang menggunakan OS Windows dalam satu jaringan.

Samba merupakan software yang bekerja di sistem operasi linux, unix dan windows yang menggunakan protokol network smb (*server message block*). Smb adalah sebuah protokol komunikasi data yang juga digunakan oleh Microsoft dan OS/2 untuk menampilkan fungsi

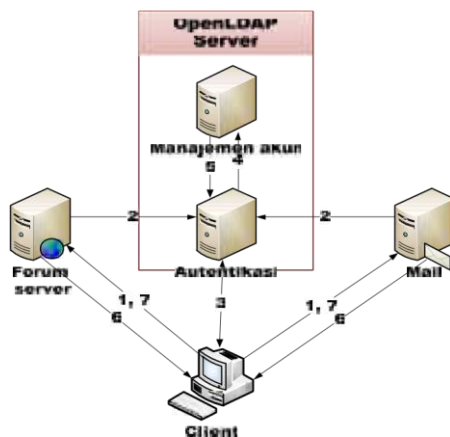
jaringan *client-server* yang menyediakan sharing file dan printer serta tugas-tugas lainnya yang berhubungan. Samba mampu membagi file dengan komputer yang menggunakan sistem operasi linux, unix dan windows dengan sistem *peer to peer*. Selain membagi file samba bisa menjembatani fungsi-fungsi sistem *client-server* seperti penggunaan PDC (*primary domain control*), DHCP (*dynamic host configuration protocol*), DNS (*domain name system*), FTP (*file transfer protocol*), *webserver*, *mail server*, *telnet*, *ssh* dan banyak lagi.

### Perbandingan “Sistem Manajemen Akun Dengan Menggunakan OpenLDAP”, Dengan Sistem Yang Sudah Ada.

Sebelumnya sudah pernah ada sistem yang menggunakan OpenLDAP dalam manajemen akun, namun perbedaan mendasar Sistem Manajemen Akun Dengan Menggunakan OpenLDAP dengan sistem yang sudah ada, yaitu :

1. Sistem yang sudah ada menggunakan satu akun user untuk masuk ke dalam beberapa aplikasi dalam sebuah jaringan komputer, di mana aplikasi – aplikasi tersebut membutuhkan autentikasi akun user untuk mengakses aplikasi tersebut. Namun ketika masuk dalam setiap aplikasi yang berbeda, user harus memasukkan kembali *username* dan *password*
2. Namun dalam Sistem Manajemen Akun Dengan Menggunakan OpenLDAP, user menggunakan satu akun user untuk masuk dalam aplikasi berbeda dalam sebuah jaringan komputer, di mana aplikasi – aplikasi tersebut membutuhkan autentikasi akun user untuk mengakses aplikasi tersebut. Ketika user masuk dalam beberapa aplikasi yang berbeda, user tidak perlu lagi memasukkan kembali *username* dan *password*. Jadi user cukup melakukan satu kali login akun user.

Berikut ini adalah gambar kinerja sistem yang sudah pernah dibuat dengan OpenLDAP .



Gambar 2 Prinsip Kerja Sistem Manajemen Akun Yang Sudah Pernah Dibuat.

Penjelasan prinsip kerja sistem manajemen akun yang sudah pernah dibuat :

1. User mengakses aplikasi sebelum proses autentikasi.
2. Redirect untuk proses autentikasi.
3. User login, memasukkan *username* dan *password*.
4. Identifikasi query.
5. User valid.
6. User mendapat service tiket dari OpenLDAP server.
7. User mengakses aplikasi setelah proses autentikasi.

Terlihat pada gambar 2, ketika user mengakses aplikasi berbeda user harus melakukan login ulang. Namun dalam Sistem Manajemen Akun Dengan Menggunakan OpenLDAP, user cukup melakukan satu kali login untuk mengakses aplikasi berbeda dalam sebuah jaringan komputer. Di mana aplikasi tersebut membutuhkan autentikasi *user account*.

## ANALISA DAN PERANCANGAN

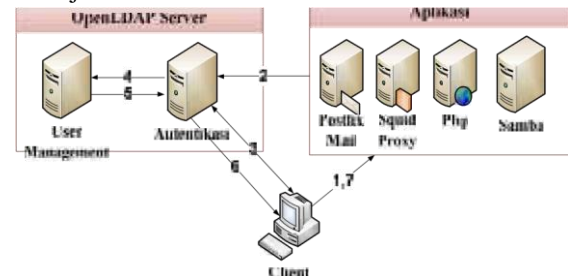
### Perancangan Konsep Sistem

Dalam implementasi sistem manajemen akun ini, menggunakan OpenLDAP sebagai *directory service* dengan menggunakan metode *Single Sign On* dan berjalan dalam lingkup LAN (*Local Area Network*). Dan aplikasi – aplikasi yang digunakan dalam implementasi sistem manajemen akun ini antara lain Postfix Mail, Squid Proxy, Samba dan aplikasi Php.

Kemudian untuk skenario pengujiannya yaitu, ketika User telah masuk ke dalam sebuah aplikasi, maka OpenLDAP Server mengirim sesi login kepada aplikasi – aplikasi yang terhubung dengan

OpenLDAP Server tersebut. Setelah proses validasi berhasil, maka user tidak perlu melakukan login ulang ketika ingin mengakses sumber daya pada aplikasi yang berbeda, di mana aplikasi tersebut membutuhkan autentikasi *user account*.

Berikut ini gambar alur proses dalam sistem manajemen akun :



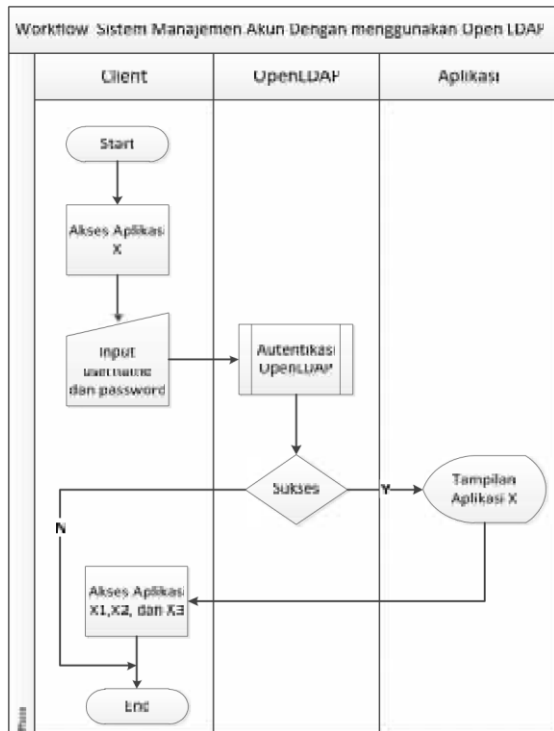
Gambar 3 Alur Proses SSO Pada Sistem Manajemen Akun.

Pada gambar 3 menjelaskan bagaimana alur dan proses dari load SSO dari awal proses hingga akhir. Untuk lebih detailnya berikut ini adalah langkah – langkah dari alur dan proses metode SSO dalam sistem manajemen akun dengan menggunakan OpenLDAP:

1. User mengakses aplikasi sebelum proses autentikasi.
2. Redirect untuk proses autentikasi.
3. User login, memasukkan *username* dan *password*.
4. Identifikasi query.
5. Validasi user.
6. User mendapat service tiket dari OpenLDAP server.
7. User mengakses aplikasi setelah proses autentikasi.

### Workflow Sistem Manajemen Akun

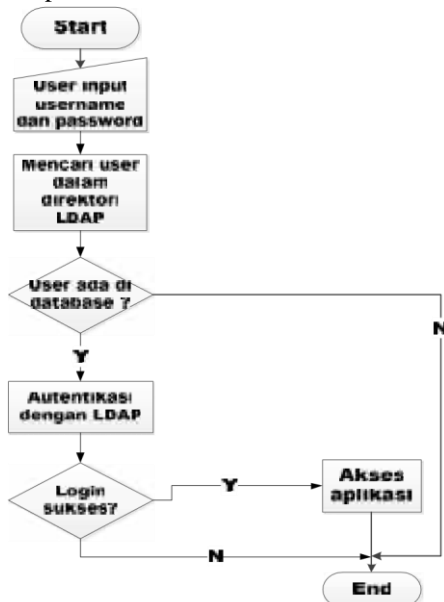
Berikut adalah *workflow* dari sistem manajemen akun dengan menggunakan OpenLDAP



Gambar 4 *Workflow* Sistem Manajemen Akun Dengan Menggunakan OpenLDAP.

#### Flowchart Login

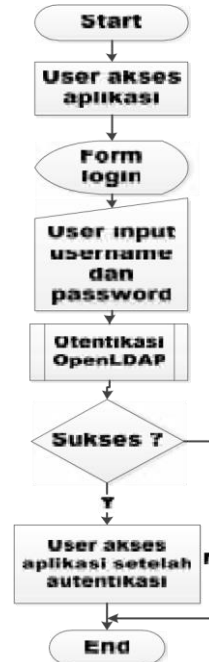
Berikut ini *flowchart* proses autentikasi dalam server OpenLDAP:



Gambar 5 *Flowchart* Autentikasi Pada OpenLDAP

**Flowchart akses aplikasi X, X, X2 dan X3 (X, X1, X2, X3 = Postfix mail/ Squid Proxy/ Samba/ Aplikasi Php ).**

Berikut adalah *flowchart* user untuk akses aplikasi X, X, X2 dan X3 (X, X1, X2, X3 = Postfix mail/ Squid Proxy/ Samba/ Aplikasi Php ).



Gambar 6 *Flowchart* Akses Aplikasi X, X, X2 Dan X3 (X, X1, X2, X3 = Postfix mail/ Squid Proxy/ Samba/ Aplikasi Php ).

#### Skenario Pengujian

Skenario pengujian yang akan dilakukan dalam penelitian ini yaitu sebagai berikut:

1. Masuk ke dalam salah satu aplikasi (Postfix mail/ Squid Proxy/ Samba/ Aplikasi Php).
2. Setelah proses autentikasi pada OpenLDAP berhasil, user bisa mengakses aplikasi yang lainnya (Postfix mail/ Squid Proxy/ Samba/ Aplikasi Php )tanpa harus melakukan login ulang.

#### HASIL DAN PEMBAHASAN

Uji coba diperlukan untuk mengetahui sistem telah benar atau belum. Sehingga dapat diketahui dari hasil uji coba, apakah sistem ini dapat berjalan dengan baik, atau belum berjalan dengan baik. Adapun skenario yang akan dilakukan dalam pengujian Sistem Manajemen Akun Dengan Menggunakan OpenLDAP ini, sebagai berikut :

User yang telah terdaftar dalam server OpenLDAP, akan masuk ke dalam salah satu aplikasi dalam jaringan lokal ( Postfix, Squid proxy, Aplikasi Php dan Samba) , di mana aplikasi ini membutuhkan autentikasi user.

Ketika proses autentikasi berhasil maka user dapat aplikasi-aplikasi yang lain tanpa melakukan login ulang.

Terdapat 2 cara penilaian pengujian dimana cara tersebut nantinya bisa dijadikan acuan untuk mengetahui apakah pengujian ini dapat dinyatakan berhasil atau gagal. Berikut adalah 2 cara penilainya :

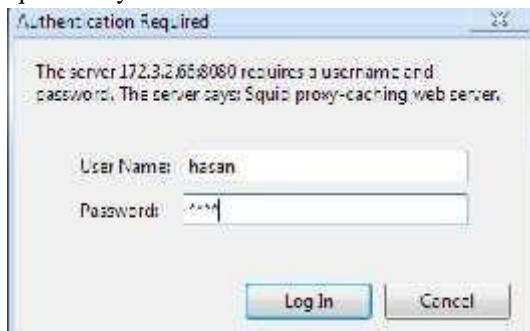
### Pengujian Berhasil

Pengujian dikatakan berhasil apabila user yang terdaftar pada OpenLDAP telah masuk pada salah satu aplikasi, dapat mengakses aplikasi-aplikasi yang lain tanpa harus melakukan login ulang. Berikut gambar dari hasil uji coba:



Gambar 7 User Yang Terdaftar Dalam OpenLDAP

Dilakukan uji coba apakah user “hasan” bisa masuk ke dalam aplikasi-aplikasi yang menggunakan autentikasi OpenLDAP. Pertama mengakses internet, dan terlebih dahulu melalui Squid Proxy.

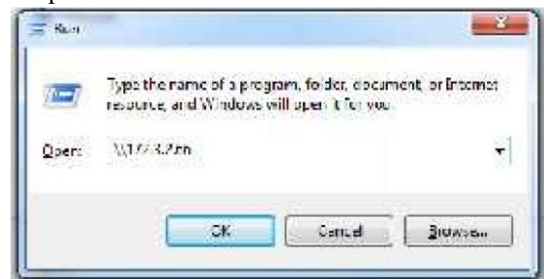


Gambar 8 Login Squid proxy Untuk Akses Internet



Gambar 9 Akses Internet Lewat Web Browser

Percobaan selanjutnya dengan mengakses *file sharing* dengan menggunakan aplikasi samba, dari pc linux ubuntu melalui windows7.



Gambar 10 Mengakses File Sharing Dengan Samba

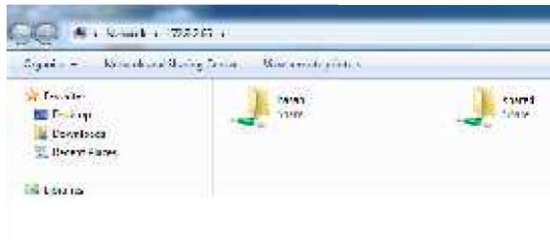
Setelah itu user diminta memasukkan username dan password untuk mengakses file yang dishare melalui aplikasi samba.



Gambar 11 Form Login Samba

Setelah sukses melalui proses autentikasi user dapat mengakses *file sharing*.





Gambar 12 Akses File Sharing Setelah Sukses Login Samba

### Pengujian Gagal

Pengujian dikatakan gagal apabila user yang terdaftar pada OpenLDAP telah masuk pada salah satu aplikasi, tidak dapat mengakses aplikasiaplikasi yang lain tanpa harus melakukan login ulang. Berikut gambar dari hasil uji coba:



Gambar 13 Form Login Postfix

Ketika user “hasan” telah berhasil masuk pada aplikasi samba dan squid, user diminta memasukkan email dan password untuk mengakses postfix admin. Seharusnya ketika user mengakses aplikasi Postfix, user langsung dapat mengaksesnya.

Hal ini disebabkan karena autentikasi aplikasi Postfix dengan server OpenLDAP belum terhubung, jadi proses autentikasi dibutuhkan lagi dalam mengakses aplikasi Postfix.

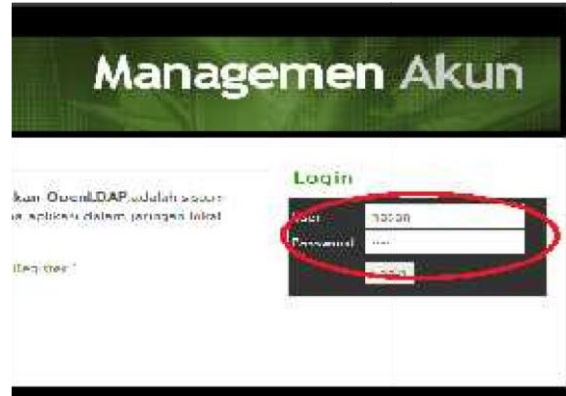


Gambar 14 Form Login Aplikasi Php

Ketika user “hasan” telah berhasil masuk pada aplikasi samba dan squid, user diminta memasukkan email dan password untuk mengakses

aplikasi Php. Seharusnya ketika user mengakses aplikasi Php, user langsung dapat mengaksesnya.

Hal ini disebabkan karena autentikasi aplikasi Php dengan server OpenLDAP belum terhubung, jadi proses autentikasi dibutuhkan lagi dalam mengakses aplikasi Php.



Gambar 15 Form Login Sistem Manajemen Akun

Ketika user “hasan” masuk pada Sistem Manajemen Akun, muncul pesan “login gagal” seperti pada gambar 16.



Gambar 16 Login Gagal

Hal ini disebabkan karena Sistem Manajemen Akun yang berbasis web dengan server OpenLDAP belum terhubung, jadi proses autentikasi gagal ketika user “hasan” masuk ke dalam Sistem Manajemen Akun.

## KESIMPULAN DAN SARAN

### Kesimpulan

Dari uji coba yang telah dilakukan dengan menggunakan skenario uji coba, maka dapat ditarik beberapa kesimpulan. Diantaranya:

1. Metode *Single Sign On (SSO)* dapat berjalan dalam Sistem Manajemen Akun Dengan Menggunakan OpenLDAP.
2. Dengan menggunakan OpenLDAP sebagai server direktori aktif, dapat menjadi alternatif yang baik dalam manajemen akun – akun yang ada dalam beberapa aplikasi jaringan lokal atau LAN (*Local Area Network*).



## Saran

Berdasarkan hasil uji coba ada beberapa saran untuk perkembangan Sistem Manajemen Akun Dengan Menggunakan OpenLDAP di masa mendatang, antara lain:

1. Implementasi Sistem Manajemen Akun dengan menggunakan algoritma.
2. Pembuatan tampilan aplikasi yang *user friendly*.

## DAFTAR PUSTAKA

- [1] Arif, Andi Hakim. 2011. *SINKRONISASI OPENLDAP PADA ZIMBRA DAN DRUPAL DENGAN STUDI KASUS SINGLE*
- [2] *ACCOUNT MAIL SERVER DAN FORUM*, Jurnal Program Studi Teknik Komputer Politeknik Telkom. Bandung.
- [3] InfoLINUX, 2005. *Integrasi User Account Dengan LDAP*: Majalah InfoLINUX Edisi 07/2005. Jakarta.
- [4] Kurusetra 2012. *Optimasi squid proxy server*. [Online] Tersedia :  
<http://kurusetra.web.id/2012/06/07/optimasi-squid-proxy-server/> [23 November 2012].
- [5] Kurusetra 2012. *Single Sign On Server Ubuntu 12.04*. [Online] Tersedia:  
<http://kurusetra.web.id/2012/10/15/single-sign-on-server-pada-ubuntu-12-04> [01 Januari 2013].
- [6] Linode Library 2012. *Email with Postfix, Dovecot and MySQL on Ubuntu 10.04 LTS (Lucid)* [Online] Tersedia:  
<http://library.linode.com/email/postfix/dovecot-mysql-ubuntu-10.04-lucid> [28 Desember 2012].
- [7] Shabri's Blog 2012. *PENGERTIAN, FUNGSI DAN KEUNGGULAN SAMBA* [Online] Tersedia:  
<http://shabri.cyber4rt.com/2012/05/pengertian-fungsi-dan-keunggulan-samba.html> [1 Januari 2013].
- [8] Servicenow 2012. *External Authentication (Single Sign-On - SSO)*. [Online] Tersedia:  
[http://wiki.servicenow.com/index.php?title=External\\_Authentication\\_\(Single\\_Sign-On\\_\\_SSO\)](http://wiki.servicenow.com/index.php?title=External_Authentication_(Single_Sign-On__SSO)) [25 September 2012].
- [9] The Tuxnetworks Linux Bible 2012. *HOWTO: SAMBA + LDAP on 10.04 Lucid Part 1* [Online] Tersedia:  
<http://tuxnetworks.blogspot.com/2010/07/howto-samba-ldap-on-1004-lucid-short.html> [5 Januari 2013].
- [10] The Tuxnetworks Linux Bible 2012. *HOWTO: SAMBA + LDAP on 10.04 Lucid Part 2* [Online] Tersedia:  
<http://tuxnetworks.blogspot.com/2010/07/howto-samba-ldap-on-1004-lucid-short.html> [5 Januari 2013].
- [11] Wahyudi 2012. *Cara Instal dan Optimasi Squid Proxy Server*. [Online] Tersedia :  
<http://wahyudi83.wordpress.com/2012/06/05/cara-instal-dan-optimasi-squid-proxyserver/> [12 September 2012]

{ Halaman sengaja dikosongkan }  
{ Jurnal **Melek IT** }

(10)

## RANCANG BANGUN SISTEM DAFTAR RIWAYAT HIDUP DOSEN MENGGUNAKAN METODE *VIEWPOINT ORIENTED REQUIREMENTS DEFINITION (VORD)*

Christian Novianto, Nia Saurina

Program Studi Teknik Informatika, Fakultas Teknik, Univesitas Wijaya Kusuma Surabaya  
[Christ.lanostra@gmail.com](mailto:Christ.lanostra@gmail.com)

### ABSTRAK

Pada beberapa instansi pemerintahan desa sampai saat ini dalam melakukan layanan surat-menyurat terhadap kepentingan penduduk masih dilakukan secara manual, untuk mengurangi kesalahan penulisan, maka validitas dan pencarian akurasi data penduduk dibutuhkan. Metode pencarian heuristic (*generate and test*) memiliki pengetahuan untuk membimbing pencarian menuju ke keadaan tujuan yaitu bergerak ke belakang menuju suatu keadaan awal. Dengan pencarian data menggunakan heuristic maka sistem akan mampu menghasilkan data yang akurat.

Sistem cerdas layanan pembuatan surat dinas Desa dengan menggunakan metode heuristic generate and test ini merupakan sistem yang di tujukan untuk kepentingan instansi pemerintahan desa. Sistem ini dibuat dengan pemrograman berbasis web dan dikembangkan dengan menggunakan DFD, ERD dan *Flowchart* untuk metode analisisnya, dan PHP untuk bahasa pemrogramannya, serta *SQLyog Enterprise* untuk database sebagai penyimpanannya. Dibuat untuk melakukan pencarian data nik sebagai isian data surat guna membantu instasi pemerintah Desa dalam melakukan tugas dalam pelayanan pembuatan surat dinas Desa.

**Kata kunci :** Metode Heuristic Generate And Test, DFD dan ERD, Flowchart, PHP, SQLyog Enterprise

### PENDAHULUAN

#### Latar Belakang

Daftar riwayat hidup dosen adalah suatu tempat atau *form* pengisian data diri dosen yang berisi tentang catatan Secara umum dan merupakan kumpulan dokumen seseorang, kelompok, lembaga, organisasi, perusahaan atau sejenisnya yang bertujuan untuk mendokumentasikan perkembangan suatu proses dalam mencapai tujuan yang telah ditetapkan. Untuk mencapai tujuan yang telah ditetapkan pada aturan yang telah ditentukan oleh UU Nomor 14 Tahun 2005 dan sesuai dengan prosedur Tri Dharma Perguruan Tinggi. Adapun prosedur Tri Dharma Perguruan Tinggi meliputi penelitian, pengajaran, dan pengabdian masyarakat. Saat ini belum ada *sistem* yang mengingatkan dosen melakukan Tri Dharma berdasarkan tingkat kewajaran yang telah ditetapkan oleh UU Nomor 14 Tahun 2005. Metode *Viewpoint Oriented Requirements Definition (VORD)* Menurut Kotonya (1996) dan Sommerville (1998) metode *Viewpoint-Oriented Requirements Definition (VORD)* adalah metode yang mendefinisikan persyaratan berorientasi sudut pandang. Metode *VORD* telah dirancang sebagai kerangka kerja berorientasi layanan untuk elisitasi dan analisis persyaratan. Sudut pandang dan informasi layanan pada *VORD* dikumpulkan dengan menggunakan *form* standar.

Metode *VORD* juga menggunakan berbagai notasi diagramatik termasuk diagram hierarki, sudut pandang dan skenario *event*.

Pada tugas akhir ini akan membuat rancang bangun sistem daftar riwayat hidup dosen menggunakan metode vord dengan tujuan mengingatkan dosen untuk bekerja sesuai peraturan yang telah ditentukan.

### TINJAUAN PUSTAKA

#### Sistem Informasi

Menurut Robert A. Leitch dan K. Roscoe Davis (1983), sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Menurut O'Brien (2005), sistem informasi adalah rangkaian orang, prosedur dan sumber daya yang mengumpulkan, mengubah dan menyebarkan informasi dalam sebuah organisasi dan sebuah sistem yang menerima sumber daya sebagai *input* dan memprosesnya kedalam produk informasi sebagai *outputnya*.

### Daftar Riwayat Hidup

Pengertian Riwayat hidup adalah catatan singkat tentang gambaran diri seseorang. Selain berisi data pribadi, gambaran diri itu paling tidak harus di isi keterangan tentang pendidikan atau keahlian dan pengalaman. Dengan data itu riwayat hidup akan memberikan gambaran atau kualifikasi seseorang.

Dari segi penampilannya riwayat hidup tidak mempunyai bentuk standard. Riwayat hidup ditulis seperti karangan singkat, diawali oleh judul dan ditutup oleh rangkaian tanggal, tanda tangan dan nama. Sebenarnya riwayat hidup termasuk surat keterangan, dalam hal ini keterangan pribadi. Isi riwayat hidup dapat dikelompokkan dalam empat sampai lima subjudul. Data Pribadi yang perlu dicantumkan dalam data pribadi yang terpenting saja. Data pribadi dalam riwayat hidup meliputi nama, tempat dan tanggal lahir, jenis kelamin, status perkawinan, jati diri (nomor KTP) dan alamat (domisili pada saat melamar). Adapun data lain seperti kewarganegaraan, suku bangsa, agama, tinggi dan berat badan pencatumannya supaya di sesuaikan dengan situasi dan kondisi. Jika pencantuman itu berdampak positif baru di cantumkan, tetapi apabila mubazir lebih baik tidak perlu di cantumkan.

Daftar riwayat hidup dapat digunakan untuk mendokumentasikan perkembangan dosen. Karena menyadari proses pendataan sangat penting untuk Universitas, Daftar riwayat hidup dapat digunakan oleh dosen untuk melihat kemajuan mereka sendiri terutama dalam hal perkembangan, sikap keterampilan dan ekspresinya terhadap sesuatu.

Daftar riwayat hidup mencakup berbagai contoh pekerjaan dosen yang tergantung pada keluasan tujuan. Contoh pekerjaan dosen ini memberikan dasar bagi pertimbangan bagi kemajuan belajarnya dan dapat dikomunikasikan dengan dosen. Sehingga Daftar riwayat hidup dapat digunakan untuk mendokumentasikan perkembangan dosen dalam setiap kegiatan dan proses pembelajaran. Secara umum, dalam dunia pendidikan Daftar riwayat hidup merupakan kumpulan hasil karya dosen atau catatan mengenai dosen yang didokumentasikan secara baik dan teratur. Daftar riwayat hidup dapat berbentuk tugas-tugas yang dikerjakan dosen, jabatan dosen, catatan hasil observasi dosen, pelatihan profesional, pengalaman berupa jabatan, pengalaman mengajar, pengalaman membimbing Dosen, karya tulis ilmiah dan lain-lain.

### Konsep Pemrograman Web

Dalam dunia internet kita selalu mendengar istilah *web*. Istilah ini sebenarnya adalah istilah pendek dari *World Wide Web* (disingkat *www*). *Web* ini adalah suatu ruang informasi dimana sumber-sumber daya yang berguna diidentifikasi oleh pengenalan *global* yang disebut *Uniform Resource Identifier* (URI). *Web* sering dianggap identik dengan internet dan merupakan bagian dari internet tersebut. Halaman *web* biasanya diatur dalam koleksi material yang berkaitan yang disebut dengan situs *web*. Monitor menampilkan halaman *web* dari suatu *server web* yang diambil informasinya oleh *browser web*.

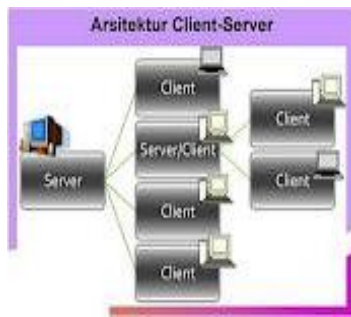
Halaman *web* merupakan *file* teks murni (*plain text*) yang berisi sintaks-sintaks HTML yang dapat dibuka atau dilihat dengan internet *browser*. Kegiatan pemindahan halaman untuk pindah ke dokumen lain atau bahkan mengirim informasi kembali kepada *server* untuk berinteraksi dengannya sering disebut dengan *surfing* atau berselancar. Halaman-halaman dari sebuah situs internet atau *web* diakses dari sebuah URL (*Uniform Resource Locator*) yang menjadi akar (*root*) yang disebut *homepage* dan biasanya disimpan dalam *server* yang sama.

### Internet WebBrowser

*Browser* adalah sebuah program aplikasi atau *software* yang me-request dokumen dari komputer- komputer yang terkoneksi internet (*server*) di seluruh dunia dan menampilkan informasi dari dokumen tersebut pada *window browser*. *Browser* menampilkan sesuai intruksi (format) HTML yang dibuat pada dokumen tersebut.

Konsep pemrograman *web* adalah membuat aplikasi berbasis *web*. Aplikasi berbasis *web* dibuat dengan memanfaatkan mekanisme dan aplikasi yang sudah ada pada sistem *web* (*www*). Sistem *web* sebenarnya merupakan aplikasi yang berarsitektur *client-server*.

Pemrograman *web* menggunakan *protocol* HTTP (*HyperTextTransferProtocol* ) dalam komunikasi antara *client* dan *server* yang mempunyai fungsi untuk mengambil atau menjalankan isi *file* dokumen *web* di sisi *server* dan menampilkannya di sisi *client*.



Gambar 1 Arsitektur *Client server*

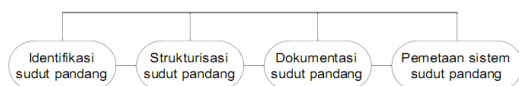
*Server* berfungsi menyediakan dokumen *web*, mengeksekusi dokumen *web* yang bersifat *server side* (php, asp, jsp) dan menerjemahkan ke dalam bentuk yang bisa dipahami oleh *web browser* di *client*, memiliki *service* (program) *web server* (IIS, Apache dan sebagainya) memberikan *respon* ke *client*. *Client* berfungsi mengambil dokumen *web*, mengeksekusi dokumen *web* yang bersifat *client side* (javascript, html, css). Memiliki *web browser* untuk menerjemahkan dokumen *web* ke bentuk yang “manusiawi”, mengirim *request* ke *server*.

### Metode *Viewpoint-Oriented Requirements Definition (VORD)*

Menurut Kotonya (1996) dan Sommerville (1998) metode *Viewpoint-Oriented Requirements Definition (VORD)* adalah metode yang mendefinisikan persyaratan berorientasi sudut pandang. Metode VORD telah dirancang sebagai kerangka kerja berorientasi layanan untuk elisitasi dan analisis persyaratan. Sudut pandang dan informasi layanan pada VORD dikumpulkan dengan menggunakan *form* standar. Metode VORD juga menggunakan berbagai notasi diagramatik termasuk diagram hierarki, sudut pandang dan skenario *event*.

### Tahap-Tahap Metode VORD

Pada gambar 2.3 adalah tahap-tahap yang ada dalam metode VORD (Ian Sommerville, 2003).



Gambar 2 Tahap-Tahap Metode VORD (Ian Sommerville, 2003)

Dari tahap-tahap utama metode VORD, sebagaimana diperlihatkan seperti pada gambar 2.3, adalah:

1. Identifikasi sudut pandang, yang mencakup pencarian sudut pandang yang menerima layanan sistem dan pengidentifikasian layanan-layanan khusus yang diberikan bagi setiap sudut pandang.
2. Strukturisasi sudut pandang, yang mencakup pengelompokan sudut pandang yang berhubungan menjadi suatu hierarki.
3. Layanan-layanan yang umum diberikan pada tingkatan yang lebih tinggi pada hierarki dan diwarisi oleh sudut pandang tingkat rendah.
4. Dokumentasi sudut pandang, yang mencakup penyempurnaan deskripsi sudut pandang dan layanan yang teridentifikasi.
5. Pemetaan sistem sudut pandang, yang mencakup pengidentifikasian objek pada desain berorientasi objek dengan menggunakan informasi layanan yang dicakup dalam sudut pandang dan layanan.

Menurut Ian Sommerville (2003) metode VORD menggunakan *form* standar dan *form* layanan dalam melakukan tahapan-tahapan metode VORD. Contoh *form* standar yang dipetakan pada tabel 2.1 sebagai berikut:

Tabel 1 *Form Template* Sudut Pandang

Template sudut pandang	
Referensi	Nama sudut pandang.
Atribut	Atribut yang menyediakan informasi sudut pandang.
Event	Referensi ke satu set skenario <i>event</i> yang mendeskripsikan bagaimana sistem bereaksi.
Layanan	Referensi ke satu set deskripsi layanan.
Sub-Vp	Nama sub- <i>viewpoint</i> (sub-sudut pandang).

Contoh *form* layanan yang dipetakan pada tabel 2 sebagai berikut:

Tabel 2 *Form Template* Layanan

Template layanan	
Referensi	Nama layanan.
Atribut	Alasan mengapa layanan ini disediakan.
Event	Referensi ke sejumlah spesifikasi layanan. Ini bisa dinyatakan dengan notasi yang berbeda-beda.
Layanan	Daftar nama sudut pandang yang menerima layanan.
Sub-Vp	Referensi ke sejumlah objek sistem yang menyediakan layanan tersebut.

### Analisis Sudut Pandang dan Layanan

Langkah-langkah yang harus diperhatikan dalam analisis sudut pandang dan layanan seperti dijelaskan sebagai berikut (Ian Sommerville, 2003):

1. Mengidentifikasi sudut pandang yang mungkin. Satu pendekatan yang dipakai adalah pendekatan *brainstroming*, di mana

layanan dan entitas potensial yang berinteraksi dengan sistem diidentifikasi. Hasil *brainstorming* dari sudut pandang yang mungkin dituliskan dalam diagram gelembung (*bubble diagram*).

2. Mengidentifikasi sudut pandang dari *bubble diagram* hasil *brainstorming*.
3. Mengalokasikan identifikasi layanan ke dalam identifikasi sudut pandang. Layanan yang tidak teralokasikan dapat menggambarkan sudut pandang yang belum teridentifikasi pada sesi *brainstorming* awal.
4. Menyediakan input bagi sudut pandang layanan.
5. Menunjukkan kesamaan sudut pandang dan memakai ulang informasi sudut pandang, hierarki inherensi mengeluarkan sudut pandang yang memberikan layanan yang umum. Layanan data dan informasi kontrol diwarisi oleh sub-sudut-pandang.
6. Menemukan informasi yang lebih rinci mengenai layanan yang disediakan, data yang dibutuhkan dan bagaimana pengontrolannya.
7. Membuat *template* sudut pandang menjadi *scenario event* yang mendeskripsikan bagaimana sistem berperilaku dalam tanggapannya terhadap berbagai *event*. Tujuan dari *template* sudut pandang dan layanan dan *scenario event* dikembangkan untuk semua sudut pandang dan layanan yang teridentifikasi. Informasi kemudian dapat diperiksa silang untuk menemukan *error* pada analisis dan konflik persyaratan.

### Skenario

Skenario adalah deskripsi sesi interaksi contoh yang sangat berguna untuk menambahkan detail garis besar deskripsi persyaratan (Ian Sommerville, 2003). Skenario yang sangat umum mencakup sebagai berikut:

Deskripsi status sistem pada awal skenario.

1. Deskripsi aliran sistem yang normal pada skenario.
2. Deskripsi mengenai apa yang bisa salah dan bagaimana penanganannya.
3. Informasi mengenai kegiatan lain yang bisa berlangsung pada saat yang sama.
4. Deskripsi status sistem setelah berakhirnya skenario.

Adapun menurut Sudarmawan dan Dony Ariyus (2007:168) skenario adalah suatu uraian interaksi manusia dengan mesin. Skenario membantu proses desain fokus pada keperluan *user* yang berbeda secara teknis. Skenario dihubungkan dengan kasus penggunaan yang dijelaskan dengan

tingkat teknis interaksi. Skenario bisa dipahami oleh orang yang tidak mempunyai latar belakang teknis sehingga dapat digunakan untuk aktivitas desain. Dengan skenario yang sesuai, kapanpun dapat dideskripsikan suatu interaksi sistem dari perspektif *user*. Skenario dapat membantu membatasi keruwetan pada lapisan teknologi (di mana dia berada) dan dapat dicegah dengan menggunakan *user interface*.

Skenario *event* digunakan pada VORD untuk mendokumentasi perilaku sistem jika dihadapkan pada kumpulan *event* tertentu.

### PHP

PHP (*Hypertext preprocessor*) yang merupakan bahasa pemrograman berbasis *web* yang memiliki kemampuan untuk memproses data dinamis. PHP dikatakan sebagai sebuah *server-side embedded script language* artinya sintaks-sintaks dan perintah yang kita berikan akan sepenuhnya dijalankan oleh *server* tetapi disertakan pada halaman HTML biasa. Aplikasi-aplikasi yang dibangun oleh PHP pada umumnya akan memberikan hasil pada *web browser*, tetapi prosesnya secara keseluruhan dijalankan di *server*.

Pada prinsipnya *server* akan bekerja apabila ada permintaan dari *client*. Dalam hal ini *client* menggunakan kode-kode PHP untuk mengirimkan permintaan ke *server*.

Ketika menggunakan PHP sebagai *server-side embedded script language* maka *server* akan melakukan hal-hal sebagai berikut :

1. Membaca permintaan dari *client/browser*
2. Mencari halaman *page* di *server*
3. Melakukan instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman/*page*.
4. Mengirim kembali halaman tersebut kepada *client* melalui internet atau intranet.

### MY SQL

MySQL adalah suatu *software* sistem manajemen database. Database adalah suatu koleksi data yang terstruktur. MySQL merupakan *Relational Database Management sistem* (RDBMS) yang didistribusikan secara gratis di bawah lisensi *General Public License* (GPL). Sistem manajemen database seperti MySQL diperlukan untuk menambahkan, mengakses, memproses data yang disimpan di komputer. MySQL menggunakan *Standart Structured Query Language* (SQL), yaitu bahasa standar yang paling banyak digunakan untuk mengakses database. Alasan penggunaan MySQL sebagai *software* database *server* adalah MySQL dari awal didesain



untuk menangani database yang cukup sangat besar, lebih cepat dari solusi yang ada. Konektivitas, kecepatan dan *security* yang baik membuat *MySQL* sangat cocok digunakan di internet.

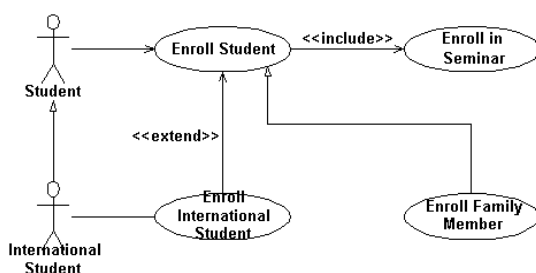
### Unified Modeling Language (UML)

UML (Unified Modeling Language) adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan artifact (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. *Artifact* dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya (Cernosek dan Naiburg, 2004).

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. Use case diagram

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya.

*Use case diagram* dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan *client*, dan merancang *test case* untuk semua *feature* yang ada pada sistem.



Gambar 3 Contoh *usecase diagram*

Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di- *include* akan dipanggil setiap kali *use case* yang meng- *include* dieksekusi secara normal.

Sebuah *use case* dapat di-*include* oleh

lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*. Sebuah *use case* juga dapat meng-*extend usecase* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.

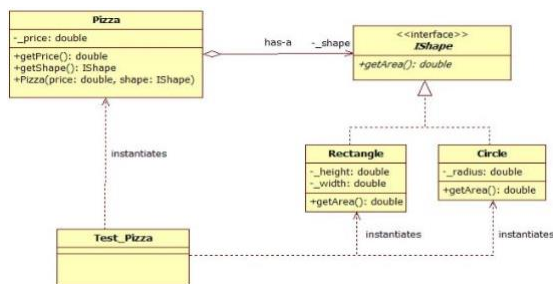
### Class Diagram

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut / property ) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda / fungsi).

*Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya.
3. *Public*, dapat dipanggil oleh siapa saja.  
Hubungan antar class diantaranya :
  - a. Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain. Panah *navigability* menunjukkan arah *query* antar *class*.
  - b. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas..”).
  - c. Pewarisan, yaitu hubungan hirarkis antar *class*. *Class* dapat diturunkan dari *class* lain dan mewarisi semua atribut dan metoda *class* asalnya dan menambahkan fungsionalitas baru, sehingga ia disebut anak dari *class* yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
  - d. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di-*passing* dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

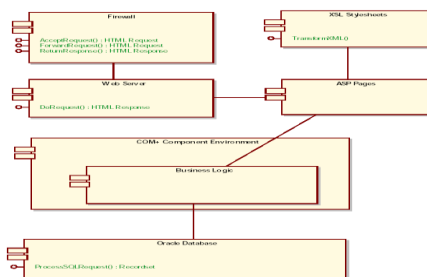


Gambar 3 Contoh diagram class

## Component Diagram

*Component diagram* menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source*

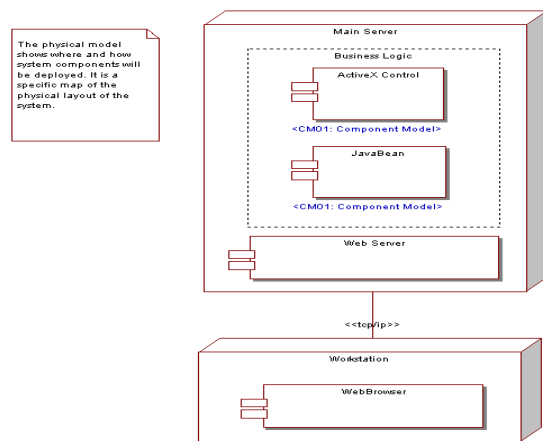
*code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen – komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.



Gambar 4 Contoh diagram komponen

## Deployment Diagram

*Deployment / physical diagram* menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisik. Sebuah *node* adalah *server*, *workstation*, atau piranti keras lain yang digunakan untuk men-deploy komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam *diagram* ini.



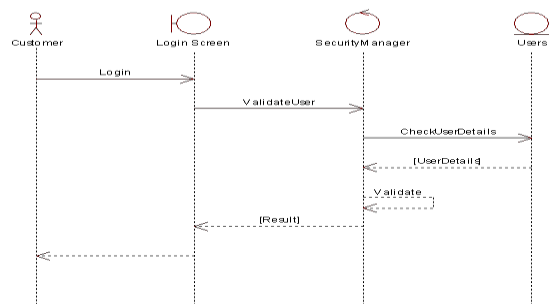
Gambar 5 Contoh diagram deployment

## Sequence Diagram

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

*Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men- *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara *internal* dan *output* apa yang dihasilkan.

Masing - masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*.



Gambar 6 Contoh diagram sekuensial

## Peraturan Beban Kerja Dosen, Penelitian, Penmas

PEDOMAN PENGHITUNGAN BEBAN KERJA DOSEN DAN EVALUASI PELAKSANAAN TRI DHARMA PERGURUAN TINGGI

No	Kegiatan	SKS Maks	Masa Berlaku	Bukti Fisik	Keterangan														
1	2	3	4	5	6														
1. BIDANG PENDIDIKAN DAN PENGAJARAN:																			
1.1	Memberi kuliah pada tingkat SD dan S1 terhadap setiap kelompok yang terdiri dari sebanyak-banyaknya 40 orang mahasiswa selama 1 semester pada kegiatan kuliah reguler maupun kuliah semester sisipan (KSS) dengan ketentuan:	1	1 th	Surat Tugas/ SK/ SKMT dan presentasi mahasiswa (dapat ditambah dengan daftar nilai Akhir (DPNSA))	Efektifitas tetap maka termasuk UTS & UAS adalah 12-14 kali/semester														
	<table><tr><th>Jumlah Dosen</th><th>Jumlah Mahasiswa</th></tr><tr><td>1</td><td>1-40 41-80 81-120</td></tr><tr><td>1</td><td>1,00 1,50 2,00</td></tr><tr><td>2</td><td>0,5 0,75 1,00</td></tr><tr><td>3</td><td>0,33 0,50 0,67</td></tr><tr><td>4</td><td>0,25 0,38 0,50</td></tr><tr><td>5</td><td>0,20 0,30 0,40</td></tr></table>	Jumlah Dosen	Jumlah Mahasiswa	1	1-40 41-80 81-120	1	1,00 1,50 2,00	2	0,5 0,75 1,00	3	0,33 0,50 0,67	4	0,25 0,38 0,50	5	0,20 0,30 0,40				
Jumlah Dosen	Jumlah Mahasiswa																		
1	1-40 41-80 81-120																		
1	1,00 1,50 2,00																		
2	0,5 0,75 1,00																		
3	0,33 0,50 0,67																		
4	0,25 0,38 0,50																		
5	0,20 0,30 0,40																		
	Jika lebih dari 1 sks maka hasil perhitungan seperti tabel di atas dikalikan dengan jumlah sks yang dilaksanakan. Ketidurnan dosen dihitung secara proporsional.																		

Gambar 7 contoh tabel penghitungan beban kerja dosen

## ANALISA DAN PERANCANGAN

### Kebutuhan Pengguna

#### 1. Dosen

Dosen dapat mengisi form DRH (daftar riwayat hidup) yang meliputi data pribadi, kegiatan mengajar, penelitian dan pengabdian masyarakat. Kemudian sistem dapat menyimpan DRH dosen tersebut sehingga dosen dapat mencetak data pribadinya. Apabila ada dosen ingin menuliskan DRH nya maka dosen harus menghubungi admin terlebih dahulu, hal ini dilakukan untuk meminimalisasi pengguna yang bukan berstatus dosen

#### 2. Kaprodi

Kaprodi dapat melihat laporan data diri, melihat seluruh kegiatan mengajar, riwayat penelitian dan pengabdian masyarakat yang dilakukan dosen, dan mencetak data dosen yang di butuhkan dalam bentuk dokumen

#### 3. Admin

Admin dapat menambah data pengguna, melihat seluruh kegiatan mengajar, riwayat penelitian dan pengabdian masyarakat yang dilakukan dosen, dan mencetak data dosen yang di butuhkan dalam bentuk dokumen.

### Analisa Sistem Dengan Metode VORD

Dalam rancang bangun sistem pelaporan daftar riwayat hidup dosen dengan menggunakan metode *Viewpoint*. Berdasarkan Evaluasi Pembukuan Admin dilakukan dengan cara mengimplementasikan hasil dari rancangan ke sebuah sistem baru dengan bentuk komputerisasi, dimana segala aspek pekerjaan yang terlibat didalamnya akan mendapatkan manfaat teknologi komputer secara optimal, misalnya Admin atau Kaprodi yang mencari sebuah data dari Dosen yang dibutuhkan dapat langsung mencari dari sebuah *web* yang telah tersedia, Admin dan Kaprodi dapat melihat data dosen dengan langsung mengetikan

pangkat, Jabatan maka data akan langsung keluar. Dengan adanya komputerisasi sistem maka diharapkan segala hambatan yang terjadi pada cara yang manual dapat diatasi dengan baik.

### Perancangan Sistem

Perancangan sistem dibuat bertujuan untuk memberikan gambaran secara umum kepada pengguna dalam pembuatan rancangan sistem yang baru untuk mempermudah dalam pengolahan data. Sehingga nantinya diharapkan aplikasi yang dibuat lebih baik dari pengolahan data yang masih manual.

### VORD

Tabel 3. View Point

<u>Viewpoint Dierct</u>	
Operator	Admin Kaprodi Dosen
Sistem	KHS
<u>Viewpoint Indierct</u>	
Organisasi	Program studi tehnik infrmatika
Regulatory	Peraturan Pemerintah Republik Indonesia Nomor 37 Tahun 2009 tentang Dosen Surat Keputusan Menkowasbangpan Nomor 38 Tahun 1999 tentang Jabatan Fungsional Dosen dan Nilai Angka Kreditnya Keputusan Direktur Jenderal Pendidikan Tinggi Departemen Pendidikan Dan Kebudayaan Republik Indonesia No. 48/D3/Kep/1983 Tentang Beban Tugas Tenaga Pengajar Pada Perguruan Tinggi
Environmen	1. Server 2. Sistem Operasi

### Identifikasi Viewpoint

Identifikasi *viewpoint* meliputi penemuan *viewpoint* berdasarkan layanan yang akan diterima oleh setiap *viewpoint*. Di dalam tahap ini, dilakukan identifikasi terhadap kebutuhan dengan langkah sebagai berikut :

- Mendaftar semua stakeholder yang ada di dalam Program studi ke dalam sistem DRH
- Mengidentifikasi mana yang termasuk ke dalam *viewpoint* dan *service* (layanan) di dalam kebutuhan sistem perangkat lunak.

Tabel 4. Identifikasi Viewpoint

Nama pengguna	Deskripsi
Admin	<ol style="list-style-type: none"> <li>1. Menambahkan pengguna seperti memberi hak akses pada setiap pengguna untuk membedakan pengguna.</li> <li>2. Melihat data pengguna untuk mengecek data pengguna</li> <li>3. Mengubah data sesuai dengan pangkat/golongan pengguna</li> <li>4. Mencetak data pengguna yang telah di butuhkan</li> </ol>
Kaprodi	<ol style="list-style-type: none"> <li>1. Melihat data pengguna untuk mengecek data pengguna</li> <li>2. Mengedit data pribadi dan mengubah data yang baru saja di perbarui</li> <li>3. Mencetak data pengguna dan data pribadi yang telah di butuhkan</li> </ol>
Dosen	<ol style="list-style-type: none"> <li>1. Melihat data pribadi apabila ingin melakukan pengecekan sesudah di isikan</li> <li>2. Mengedit data pribadi dan mengubah data yang baru saja di perbarui</li> <li>3. Mencetak data pribadi yang telah di butuhkan</li> </ol>
KHS	Sistem yang memanfaatkan teknologi informasi dan komunikasi untuk memfasilitasi administrasi mahasiswa dalam sebuah perkuliahan
Program studi	Program studi mempunyai tugas tugas yang melaksanakan fungsi pelayanan di bidang pendidikan
PP nomor 37 tahun 2009	tentang Dosen, perlu menetapkan Peraturan Menteri Pendidikan Nasional tentang Sertifikasi Pendidik Untuk Dosen
PP Nomor 38 Tahun 1999	Surat Keputusan Menkowsabangan Nomor 38 Tahun 1999 tentang Jabatan Fungsional Dosen dan Nilai Angka Kreditnya

Kep. Dir. Jend. Pendidikan Tinggi Dep. Pend. Dan Kebudayaan RI No. 48/D3/Kep /1983	Tentang Beban Tugas Tenaga Pengajar Pada Perguruan Tinggi
--	---

Tabel 5. Daftar Layanan Viewpoint

Nama pengguna	Deskripsi
Admin	<ol style="list-style-type: none"> <li>1. Menambahkan hak akses</li> <li>2. Melihat data pengguna</li> <li>3. Mencetak data pengguna</li> <li>4. Mengubah data</li> </ol>
Kaprodi	<ol style="list-style-type: none"> <li>1. Melihat data pengguna</li> <li>2. Mengedit data pribadi</li> <li>3. Mencetak data pengguna</li> </ol>
Dosen	<ol style="list-style-type: none"> <li>1. Melihat data pribadi</li> <li>2. Mengubah data pribadi</li> <li>3. Mencetak data pribadi</li> </ol>

Tabel 6. Atribut Viewpoint

Nama pengguna	Layanan	Deskripsi	Nama attribute
Dosen	- Memas ukan data pribadi	<ul style="list-style-type: none"> <li>- Mengisi form pengisian daftar riwayat hidup</li> <li>- Pengajaran</li> <li>- Penelitian</li> <li>- Pengmas</li> </ul>	<ul style="list-style-type: none"> <li>- Nik</li> <li>- Nama</li> <li>- Jenis kelamin</li> <li>- Tempat lahir</li> <li>- Tanggal lahir</li> <li>- Alamat email</li> <li>- Status perkawinan</li> <li>- Pangkat /golongan</li> <li>- Jabatan fungsional</li> <li>- Status dosen</li> <li>- Perguruan tinggi</li> </ul>

			<ul style="list-style-type: none"> <li>- Alamat perguruan tinggi</li> <li>- Telp</li> <li>- Alamat rumah</li> <li>- Telp</li> <li>- Foto</li> </ul>
--	--	--	---

	- mengu bah	- mengubah data pribadi yang mau di perbarui	- nama - alamat rumah - alamat email
	- Mencet ak data	- Mencatak data pribadi	- Pilih jenis laporan - Cetak data
	- Melihat data	- Melihat data pribadi	- Lihat data
Admin	- Menam bah penggu na	- Admin dapat menambahkan pengguna baru dan member hak akses pada penegguna	- Pilih - Tambah penggu na - Simpan
	Melihat data	- Sistem dapat menampilkan data dosen yang diinginkan	- Data lengkap/ profile dosen
	Mengubah data	- Mengubah data dosen.	- NIK - Pangkat /gol - jabatan
	Mencetak data	- Sistem dapat menghasilkan data berupa document	- Cetak laporan - Pilih cetak laporan
kapro di	Melihat data Dosen dan data pribadi	- Melihat data Daftar riwayat hidup keseluruhan Dosen	- Lihat profil Dosen
	Mengubah	- mengubah data pribadi yang mau di perbarui	- pilih edit - ubah data yang mau di ubah - simpan
			perubah an

	Mencetak data	- Mencetak data yang dosen	- Pilih jenis laporan - Cetak laporan
--	------------------	-------------------------------	---

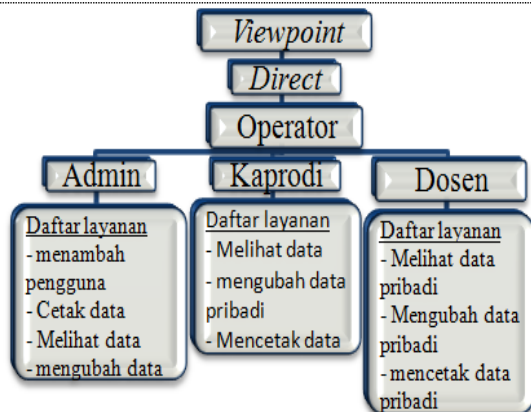
### Strukturisasi Viewpoint

Strukturisasi *viewpoint* ini meliputi pengelompokan *viewpoint* menjadi satu hirarki. Layanan umum digambarkan pada level yang paling tinggi dan diwarisi oleh *viewpoint* di level yang lebih rendah. Adapun tahapan struktrisasi ini seperti yang ditunjukkan pada gambar 3.14 Strukturisasi *viewpoint*.



Gambar 8. Strukturisasi Viewpoint

## Hirarki Viewpoint



Gambar 9. Hirarki viewpoint

## Dokumentasi Viewpoint

Dokumentasi Viewpoint meliputi cara untuk mendiskripsikan setiap viewpoint dan layanan yang telah ditentukan sebelumnya. Tujuan dari Viewpoint documentation ini adalah untuk memetakan kebutuhan sistem atau layanan terhadap masing-masing viewpoint sesuai dengan viewpoint structuring. Pemetaan kebutuhan tersebut meliputi kebutuhan fungsional dan non-fungsional sistem. Hasil dari viewpoint documentation digunakan sebagai acuan pembuatan use case.

Tabel 7 Dokumentasi Viewpoint

No	Viewpoint	Deskripsi	Tipe
----	-----------	-----------	------

1	Kaprodi	<ol style="list-style-type: none"> <li>Kaprodi dapat melakukan login</li> <li>Melihat data Dosen</li> <li>Meminta hak akses pada admin secara manual</li> <li>Dalam bidang pengajaran, penelitian, dan pengabdian kepada masyarakat dapat dilakukan minimal 3 sks</li> <li>Mencetak data Dosen</li> <li>Mengunggah data Dosen</li> </ol>	Fs Fs Ns Ns Fs Fs
2	Admin	<ol style="list-style-type: none"> <li>Admin melakukan login</li> <li>Mengisi Daftar riwayat hidup Dosen</li> <li>Memperbarui Daftar riwayat hidup Dosen</li> <li>Mengunggah foto</li> <li>Melihat data dosen</li> <li>Memberi hak akses masih dalam proses manual</li> </ol>	Fs Fs Fs Fs Ns Ns



3	Dosen	1. Pengguna dapat melakukan login 2. Melihat data Pribadi 3. Apabila login gagal lebih 4. Mencetak data Pribadi 5. Dalam bidang pengajaran, penelitian, dan pengabdian kepada masyarakat dapat dilakukan minimal 3 sks	Fs  Fs Fs Ns
4	KHS	1. Dapat menyimpan biodata pengguna sesuai dengan identitas 2. Dapat melakukan validasi terhadap data pengguna dengan waktu tidak lebih dari 5 detik 3. Dapat menampilkan beban kerja dosen	Fs  Ns  Fs
5	PP No 37	1. Dapat menampilkan beban kerja dosen 2. Pengajaran, penelitian, dan penmas dapat dilakukan minimal 3 sks max 9 sks	Fs  Ns

6	PP No 38	1. Menampilkan informasi jabatan fungsional dan angka kreditnya dari	FS  Ns
		dosen 2. Dosen yang memiliki jabatan fungsional yang dapat melakukan kegiatan pengajaran, penelitian, dan penmas	
7	Keputusan Direktur Jenderal Pendidikan Tinggi Departemen Pendidikan Dan Kebudayaan RI No. 48/D3/Kep/19 83	1. Menampilkan beban kerja tenaga pengajar pada perguruan tinggi 2. Dapat menyimpan kegiatan pendidikan tidak lebih dari 5 sks 3. Dapat menyimpan kegiatan penelitian tidak lebih dari 3 sks 4. Dapat menyimpan kegiatan penmas tidak lebih dari 3 sks	Fs  Ns  Ns  Ns

#### Pemetaan Viewpoint

Tahap ini meliputi pengimplementasian dokumentasi *viewpoint* menjadi *object-oriented design* dengan menggunakan informasi layanan yang dicakup dalam *viewpoint*.

Tabel 8 *Template* Operator Kaprodi  
*Template* Kaprodi

Referensi	Kaprodi
<i>Atribute</i>	1. Nik 2. Nama 3. Jenis kelamin 4. Tempat lahir 5. Tanggal lahir 6. Alamat email 7. Status perkawinan
	8. Pangkat/golongan 9. Jabatan fungsional 10. Status dosen 11. Perguruan tinggi 12. Alamat perguruan tinggi 13. Telp 14. Fax
<i>Event</i>	1. Melihat Daftar Riwayat Hidup Dosen 2. Mencari data dosen 3. Mengubah data pribadi 4. Cetak data Dosen atau data pribadi
<i>Layanan</i>	1. Melihat data dosen dan pribadi 2. Merubah data pribadi 3. Mencetak data dosen dan pribadi

Tabel 9 *Template* Layanan Kaprodi Melihat Data

<i>Template</i> Layanan untuk Kaprodi	
Referensi	Melihat Data
Dasar pemikiran	
Spesifikasi	1. Login 2. Melihat data pribadi 3. Log out
Viewpoint	Kaprodi
Persyaratan <i>in direct</i>	Meminta hak akses pada admin
Provider	

Tabel 10 *Template* layanan Kaprodi Mengubah Data

<i>Template</i> Layanan untuk Kaprodi	
Referensi	Mengedit/ mengubah Data
Dasar pemikiran	
Spesifikasi	1. Login 2. Pilih data edit 3. Ubah data pribadi 4. Simpan 5. Log out
Viewpoint	Kaprodi
Persyaratan <i>in direct</i>	Meminta hak akses pada admin
Provider	

Tabel 11 *Template* layanan Kaprodi Mencetak Data

<i>Template</i> Layanan untuk Kaprodi	
Referensi	Mencetak Data
Dasar pemikiran	
Spesifikasi	1. Login 2. Pilih data cetak 3. Cetak data 4. Log out
Viewpoint	Kaprodi
Persyaratan <i>in direct</i>	Meminta hak akses pada admin

### Skenario layanan

Skenario layanan sudut pandang mendeskripsikan urutan langkah-langkah dalam proses bisnis, baik yang dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor.

#### 1. Skenario Layanan Cetak Data (Kaprodi)

Tabel 12 Skenario Layanan Kaprodi Mencetak Data

Identifikasi	
<b>Nama Tujuan</b>	Mencetak data pribadi dan dosen Untuk mencetak data dosen yang diperlukan untuk kaprodi
<b>Deskripsi</b>	Menu ini menampilkan data dosen yang akan di cetak.
<b>Aktor</b>	kaprodi

Skenario Utama	
Aksi aktor	Reaksi sistem
1. kaprodi pilih menu lihat profil pribadi dan dosen	Sistem menampilkan daftar riwayat hidup dosen yang ingin di cetak
2. User pilih cetak.	Sistem akan mencetak daftar riwayat hidup dosen yang di pilih

## 2. Skenario Layanan Lihat Data Pribadi (Dosen)

Tabel 13 Skenario Layanan Lihat Data Pribadi Dosen

Identifikasi	
Nama	Melihat data pribadi
Tujuan	Untuk mengecek data pribadi yang kurang lengkap
Deskripsi	Menu ini menampilkan data pribadi
Aktor	Dosen
Skenario Utama	
Aksi aktor	Reaksi sistem
1. dosen pilih menu data profil	Sistem menampilkan daftar riwayat hidup dosen pribadi

## 3. Skenario Layanan Cetak Data (Admin)

Tabel 14 Skenario Layanan Admin Mencetak Data

Identifikasi	
Nama	Mencetak data dosen
Tujuan	Untuk mencetak data dosen yang diperlukan untuk Admin
Deskripsi	Menu ini menampilkan cetak data dosen.
Aktor	Admin
Skenario Utama	
Aksi aktor	Reaksi sistem
1. Admin pilih menu lihat profil dosen.	Sistem menampilkan daftar riwayat hidup dosen
2. User pilih cetak.	Sistem akan mencetak daftar riwayat hidup dosen yang di pilih

## 3.5. Use Case



Gambar 10 Use Case DRH Keterangan :

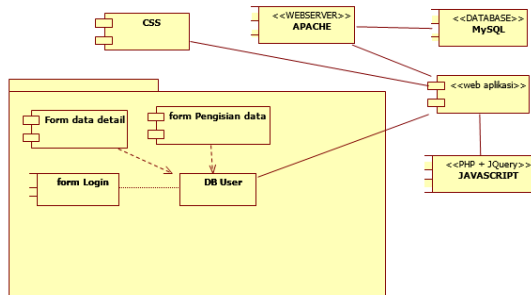
1. Use Case : Tampil data dosen, cari data dosen, isi data pribadi, data pelengkap, cetak data dosen, edit data pribadi, cetak data.
2. Aktor : terdapat 3 yaitu Admin, Kaprodi, Dosen  
Admin dan Kaprodi □ *association* (berhubungan) tampilan data dosen dan cetak data dosen: Admin dan Kaprodi bisa menampilkan data dosen yang di include dari cari data dosen dan admin juga bisa mencetak data dosen yang di tampilkan Kaprodi dan Dosen □ *association* (berhubungan) isi data pribadi, edit data pribadi, cetak data pribadi: kaprodi dan dosen mengisikan data pribadi yang extend dengan data pelengkap dan include dengan upload foto, kaprodi dan dosen bisa mengubah data yang telah di perbarui kemudian dapat di cetak.



→ Admin mengisikan teks Nik, Nama, dan Jabatan pada teks box cari, setelah proses pencarian sitem akan menampilkan hasil pencarian

→ Admin mencetak data pada form data pribadi, setelah di klik cetak maka system mencetak data pribadi Admin dan data dosen yang ingin di cetak

### 3.7 Component Diagram



Gambar 14 Component Diagram

## HASIL DAN PEMBAHASAN

Uji coba diperlukan untuk mengetahui sistem telah benar atau belum. Sehingga nantinya user atau pengguna bisa menggunakan sistem ini tanpa adanya kesalahan yang akan timbul dikemudian hari.

### 5.1 Tabel scenario uji coba

No	Layanan	UJI COBA			Keterangan
		Activiti	Rule	Action	
1	Admin	Menambah pengguna	Admin memasukkan NIK, Nama, Pass, dan Jabatan	Admin memberikan username dan pass untuk login dosen	
2	Admin	Melihat data dosen	Admin pilih menu data dosen	Sistem menampilkan data dosen	
3	Admin	Mencetak data dosen	Admin pilih layanan cetak	Laporan data dosen	

4	Admin	Ubah data dosen	Admin memilih data dosen setelah itu pilih	Sistem menyimpan data yang sudah	
			edit data dosen	di edit admin	
5	Kaprodi	Mencetak data dosen	Kaprodi pilih layanan cetak	Laporan data dosen	
6	Kaprodi	Ubah data pribadi	kaprodi memilih data pribadi setelah itu pilih	Sistem menyimpan data yang sudah	
7	Kaprodi	Mencetak data dosen	Kaprodi pilih layanan cetak	Laporan data dosen	
8	Dosen	Melihat data pribadi	Dosen memilih menu profile	Dosen dapat melihat data detail pribadi	
9	Dosen	Ubah data pribadi	Dosen memilih data pribadi setelah itu pilih	Sistem menyimpan data yang sudah	
10	Dosen	Mencetak data pribadi	Dosen pilih layanan cetak yang ada pada	Laporan data dosen hasil print/cetak	

## 5.2 Halaman Admin



Gambar 15 cetak data

Pada gambar 5.1 di jelaskan ketika data tampil apabila tombol cetak di klik maka data dosen yang tampil pada form data dosen pada halaman admin akan keluar gambar print yang di inginkan.



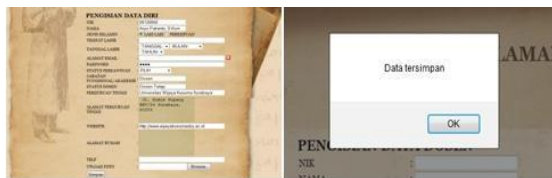
Gambar 16 Menampilkan dan melihat data dosen

Pada gambar 16 di jelaskan dari hasil pencarian data dosen keluar apabila data di tekan klik maka data akan tampil pada form data dosen dalam halaman admin.



Gambar 17 Mengubah data Dosen

Pada gambar 17 Dalam hasil pencarian admin dapat mengklik tombol edit pada kolom pilihan, disitu admin dapat megubah data dosen.



Gambar 18 Penambahan hak akses dosen dan simpan pada halaman admin

Pada gambar 18 di jelaskan admin menambahkan hak akses dosen dengan mengisi NIK, Nama, pangkat/golongan dan Jabatan sebagai

syarat pengisian data dosen dan data tersimpan

## 5.3 Halaman Kaprodi



Gambar 19 Pengisian data pribadi kaprodi

Pada gambar 19 di jelaskan kaprodi juga mengisi data untuk data pribadi.



Gambar 20 cetak data

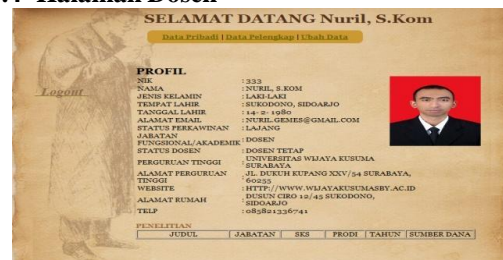
Pada gambar 20 di jelaskan ketika data tampil apabila tombol cetak di klik maka data pribadi dan data dosen yang tampil pada form data dosen pada halaman Kaprodi akan keluar gambar print yang di inginkan.



Gambar 21 Mengubah data Dosen

Pada gambar 21 Pada form ubah data di halaman kaprodi, kaprodi harus mengubah data yang ingin di ubah ketika sudah di ubah maka data akan dissimpan.

## 5.4 Halaman Dosen



Gambar 22 Tampil data pribadi dosen

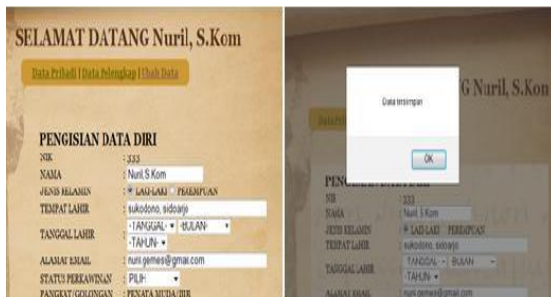


Pada gambar 22 di jelaskan form data pribadi pada halaman dosen bisa menampilkan data pribadi dosen secara keseluruhan.



Gambar 23 Cetak data

Pada gambar 23 di jelaskan apabila tombol cetak di klik maka data pribadi dosen yang tampil pada form data dosen pada halaman dosen akan keluar gambar print yang di inginkan.



Gambar 24 Mengubah data Dosen

Pada gambar 24 Pada form ubah data di halaman Dosen, dosen harus mengubah data yang ingin di ubah ketika sudah di ubah maka data akan dissimpan.

## SIMPULAN

Secara umum Rancang Bangun Sistem Pendukung Daftar Riwayat Hidup Dosen menggunakan metode *VORD* berfungsi sebagaimana yang diharapkan. Namun tentu saja masih diperlukan beberapa penyempurnaan yang akan dipaparkan dalam pembahasan bab ini.

### 1.1 Kesimpulan

Berdasarkan uraian pembahasan analisa dan pengujian yang telah dilakukan, maka dapat diambil kesimpulan terhadap sistem yaitu Rancang Bangun Sistem Daftar Riwayat Hidup Dosen dapat membantu tata usaha(TU) dalam pencarian data dosen.

### 1.2 Saran

Saran untuk perkembangan lebih lanjut dalam masa mendatang pada Rancang Bangun Sistem Pendukung Daftar Riwayat Hidup Dosen menggunakan metode *VORD*, dapat dikembangkan

dalam segi otomatisasi dan fitur-fitur pada form pengisian data diri dosen.

## DAFTAR PUSTAKA

- [1] M. Salem, Ahmed;. (2010). Requirement Analysis Through Viewpoints Oriented Requirement Model (VORD). IJACSA , 6-13.
- [2] Sommerville, Ian;. (2003). Software Engineering 6th edition. United Kingdom: Pearson Education.
- [3] Hadi, Mulyo. 2009. Dreamweaver CS4. Palembang : Maxicom. ISBN978-979-1398-80-0
- [4] Jogiyanto, HM. 2001. Analisa dan Desain SistemInformasi. Yogyakarta: Andi.
- [5] Jogiyanto. 1995. Analisa dan Perancangan SistemInformasi. Andy Offset. Yogyakarta
- [6] Kadir, Abdul. 2009. Dasar Perancangan dan ImplementasiDatabase Relational. Andi. Yogyakarta
- [7] Kadir, Abdul. 2004. Dasar Aplikasi Database MySQL. Andi. Yogyakarta

{ Halaman sengaja dikosongkan }  
{ Jurnal **Melek IT** }

(11)

## IMPLEMENTASI PERPINDAHAN AGEN PADA JARINGAN NIRKABEL MENGUNAKAN METODE HANDOFF UNTUK VOIP

Ligar Syamrama<sup>1</sup>, Nia Saurina<sup>2</sup>, Beny Y.V. Nasution<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Wijaya Kusuma Surabaya

Email: [ligar90@gmail.com](mailto:ligar90@gmail.com)

### ABSTRAK

*Access point* merupakan alat yang digunakan untuk menyebarkan sinyal *wifi*. Jangkauan dari sebuah *router access point* (agen) tidak terlalu besar. Apabila ingin membuat sebuah lokasi terhubung menjadi sebuah *Local Area Network (LAN)*, maka diperlukan banyak *access point*. Salah satu manfaat yang dapat dirasakan oleh pengguna dengan adanya *access point* adalah pengguna dapat berpindah-pindah tempat ketika terhubung dengan *access point*.

Namun ketika berpindah tempat kadang pengguna tidak menyadari bahwa kekuatan sinyal *wifi* yang terhubung menurun. Apalagi pengguna memanfaatkan sinyal *wifi* untuk VoIP, tentu tidak nyaman ketika pengguna sedang menelepon kemudian telepon terputus akibat sinyal *wifi* yang terputus.

Agar VoIP yang terhubung ke *wifi* tetap berjalan dan tidak terputus koneksinya maka digunakan metode *handoff*. Bisa menanganinya perpindahan pengguna yang terhubung ke *access point* sehingga koneksi tidak terputus ketika pengguna berpindah tempat, selama pengguna masih berada di jangkauan *access point* di sebuah *LAN*.

**Kata Kunci:** Access point, VoIP, Wifi, Handoff

### PENDAHULUAN

#### Latar Belakang

*Access Point* (agen) merupakan sebuah perangkat keras yang memungkinkan perangkat *wireless* (seperti laptop, ponsel) untuk terhubung ke dalam sebuah jaringan. Jangkauan dari sebuah *access point* (agen) tidak terlalu besar, sehingga apabila ingin membuat sebuah lokasi terhubung menjadi sebuah LAN, maka diperlukan banyak *access point* agar lokasi yang diinginkan dapat terhubung satu sama lain melalui *agen*.

*Handoff* merupakan sebuah metode dimana pengguna berpindah dari suatu area yang dijangkau oleh sinyal kemudian memasuki kawasan yang dijangkau oleh sinyal lain, maka koneksi akan dipindahkan ke agen kedua untuk mencegah terjadinya kegagalan panggilan ketika pengguna berpindah ke lokasi yang tidak dijangkau oleh agen yang pertama. *Handoff* selama ini dimanfaatkan untuk alat komunikasi *handphone* berbasis GSM (Global System for Mobile Communication) maupun CDMA (Code Division Multiple Access). Alat komunikasi seperti telepon atau semacamnya sering digunakan oleh semua orang, tetapi dengan menggunakan alat komunikasi seperti itu tidak akan selamanya menjadi yang utama. Solusi pengganti untuk melakukan komunikasi yaitu dengan menggunakan teknologi *Voice over Internet Protocol (VoIP)*.

VoIP (Voice over Internet Protocol) merupakan teknologi yang mampu melewati paket data secara *realtime* dengan jaringan *internet protocol*. VoIP dapat memanfaatkan infrastruktur *internet* yang sudah ada untuk berkomunikasi seperti layaknya menggunakan telepon biasa dan tidak dikenakan biaya telepon biasa untuk berkomunikasi dengan pengguna VoIP lainnya. Terdapat beberapa cara untuk memanfaatkan VoIP sebagai sarana percakapan, yaitu dengan menggunakan *internet* atau dengan menggunakan *wifi*. Dengan menggunakan *wifi*, pengguna tidak perlu mengeluarkan biaya tambahan, sehingga biaya untuk menelepon menggunakan VoIP pun murah bahkan gratis. Memang dengan menggunakan kabel dalam koneksi, kualitas suara dan kecepatannya lebih bagus, tetapi pengguna harus diam di satu tempat. Lain halnya dengan menggunakan *wifi*, pengguna bisa berpindah tempat selama masih ada sinyal *wifi* yang tertangkap, namun kejernihan suara lebih buruk jika dibandingkan dengan koneksi menggunakan kabel. Menggunakan *wifi* sebagai media untuk VoIP sangat tergantung pada pemancar *wifi*, yaitu *access point*. Banyak faktor yang mempengaruhi kekuatan sinyal, salah satunya yaitu jarak pengguna dan *access point* yang ada. Semakin jauh pengguna terhadap *access point*, semakin lemah pula kekuatan sinyal yang di dapat. Ketika sedang menelepon menggunakan VoIP dan terhubung ke

sebuah agen, kadang pengguna tidak sadar bahwa posisinya menelepon menjauhi agen. Pengguna harus mengkonfigurasi ulang koneksi ketika sinyal melemah dan akhirnya koneksi terputus.

Pengguna *Softphone* VoIP dengan media *wifi* jika koneksi terputus secara otomatis saluran telepon juga terputus, sehingga pengguna membutuhkan otomatisasi konfigurasi koneksi *wifi*. Oleh karena itu penelitian ini memberikan solusi mengenai implementasi VoIP untuk perpindahan agen pada jaringan nirkabel menggunakan metode *handoff*, sehingga dapat mengurangi waktu konfigurasi ulang.

Tujuan dari pembuatan Implementasi VoIP untuk perpindahan agen pada Jaringan Nirkabel dengan Metode *Handoff* adalah sebagai berikut :

1. Mencegah terjadinya kehilangan koneksi.
2. Mampu menjabarkan metode *handoff* pada jaringan nirkabel.
3. Mengimplementasikan *handoff* untuk digunakan pada VoIP.

## TINJAUAN PUSTAKA

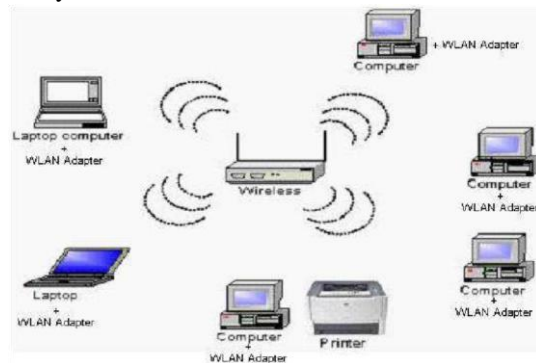
### *Handoff*

*Handoff* merupakan proses pengalihan *traffic* secara otomatis pada *Mobile Station* yang sedang digunakan untuk berkomunikasi tanpa terjadinya pemutusan hubungan, proses pengalihan *Mobile Station* (MS) dari suatu Agen ke Agen lain disebabkan oleh beberapa hal, yaitu adanya penghambat atau penghalang sehingga sinyal yang diterima oleh *Mobile Station* berkurang, dan pergerakan *Mobile Station* yang menjauhi Agen awal dan mendekati Agen baru. Sebuah agen memiliki nilai ambang batas (*threshold*), dimana *threshold* adalah batas bawah sebuah agen untuk berkomunikasi dengan *client*. Ketika kekuatan sinyal yang diterima oleh client dibawah atau lebih kecil.

### Jaringan Nirkabel

Jaringan nirkabel (*wireless network*) adalah bidang disiplin yang berkaitan dengan komunikasi antar sistem komputer tanpa menggunakan kabel. Jaringan nirkabel ini sering dipakai untuk jaringan komputer baik pada jarak yang dekat (beberapa meter) erat hubungannya dengan bidang telekomunikasi, teknologi informasi, dan teknik komputer. Jenis jaringan yang populer dalam kategori jaringan nirkabel ini meliputi: Jaringan kawasan lokal nirkabel (*wireless LAN/WLAN*), dan *wifi*. Jaringan nirkabel biasanya menghubungkan satu sistem komputer dengan sistem yang lain dengan menggunakan beberapa

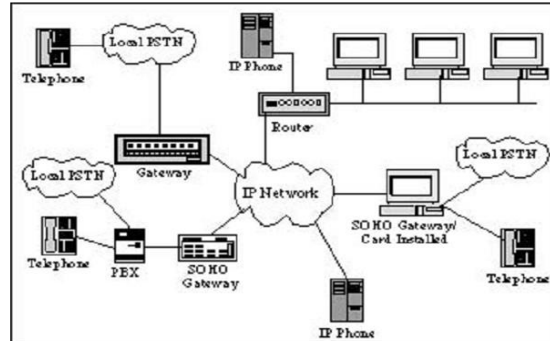
macam media transmisi tanpa kabel, seperti: gelombang radio, gelombang mikro, maupun cahaya infra merah.



Gambar 2.2 Desain Jaringan pada Jaringan Nirkabel

### Voice over Internet Protocol (VoIP)

Menurut Supriyanto (2009) dalam artikelnya yang berjudul “Membuat Layanan IP PBX”. Teknologi *Voice over Internet Protocol* (VoIP) adalah teknologi yang menjadikan media internet untuk bisa melakukan komunikasi suara jarak jauh secara langsung. Sinyal suara analog seperti yang Anda dengar ketika berkomunikasi di telepon diubah menjadi data digital dan dikirimkan melalui jaringan berupa paket-paket data secara *realtime*.



Gambar 2.3 Jaringan VoIP

Dalam komunikasi VoIP, pemakai melakukan hubungan telepon melalui terminal yang berupa PC atau telepon biasa. Dengan bertelepon menggunakan VoIP, banyak keuntungan yang dapat diambil diantaranya adalah dari segi biaya jelas lebih murah dari tarif telepon tradisional, karena jaringan IP bersifat global. Sehingga untuk hubungan Internasional dapat ditekan hingga 70%. Selain itu, biaya *maintenance* dapat ditekan karena *voice* dan data *network* terpisah, sehingga IP Phone dapat ditambah, dipindah dan diubah. Hal ini karena VoIP dapat dipasang di sembarang ethernet dan IP address, tidak seperti telepon konvensional yang harus mempunyai port tersendiri di Sentral atau PBX (*Private Branch Exchange*).

### Asterisk

Asterisk merupakan *software open source* yang biasanya digunakan untuk membangun suatu sistem layanan komunikasi. Asterisk sendiri memberikan kemudahan kepada pengguna untuk mengembangkan layanan telepon sendiri dengan kustomisasi yang seluas-luasnya diberikan kepada pihak pengguna. Dari pengertian *open source* sendiri berarti setiap pengembang dapat melihat dan mengubah *source code* yang ada, sehingga aplikasi-aplikasi yang ada dapat ditambahkan dengan mudah oleh setiap pengembang. Asterisk juga dapat dikatakan sebagai PBX yang lengkap dalam bentuk software, dan menyediakan semua fitur seperti PBX. Kelebihan Asterisk adalah dapat jalan di banyak *platform* OS, antara lain Linux, Windows, BSD, dan OS X, dan juga dapat melakukan koneksi dengan hampir semua standar yang berbasis telepon, dengan menggunakan *hardware* yang tidak begitu mahal sebagai *gateway*nya.

### Softphone

*Softphone* merupakan sebuah aplikasi yang memungkinkan untuk melakukan panggilan telepon secara langsung dari komputer maupun dari sebuah telepon seluler. Sebuah *softphone* seringkali digunakan bersama-sama dengan VoIP, sebuah bentuk panggilan telepon melalui komputer.

*Softphone* merupakan bagian dari kategori teknologi yang dikenal sebagai Computer Technology Integration (CTI). Dengan menggunakan *softphone*, panggilan dapat dibuat dari komputer atau telepon seluler ke saluran telepon (phone line), dari sebuah telepon ke komputer. Para pengguna *handphone* dengan sistem operasi Android dapat menggunakan sipdroid sebagai *softphone*.

### Android

*Android* adalah sistem operasi untuk telepon seluler yang berbasis Linux. *Android* menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak.

Android adalah sistem operasi untuk telepon seluler yang berbasis linux Fungsinya sama seperti sistem operasi Symbian di Nokia, iOS di Apple dan BlackBerry. Android tidak terikat ke satu merek *Handphone*, beberapa merek HP terkenal yang sudah memakai Android antara lain Samsung, Sony Ericsson, HTC, Nexus, Cross, Motorola, dan lain-lain.

Android pertama kali dikembangkan oleh perusahaan bernama Android Inc yang kemudian pada tahun 2005 di akuisisi oleh raksasa Internet Google. Android dibuat dengan basis kernel Linux yang telah dimodifikasi, dan untuk setiap *release*-nya diberi kode nama berdasar-kan nama hidangan makanan.

Keunggulan utama Android adalah gratis dan *open source*, yang membuat *smartphone* Android dijual lebih murah dibandingkan dengan Blackberry atau iPhone meski fitur (*hardware*) yang ditawarkan Android lebih baik.

Beberapa fitur utama dari Android antara lain WiFi *hotspot*, *Multi-touch*, *Multitasking*, GPS, *support* java, mendukung banyak jaringan (*GSM/EDGE*, *IDEN*, *CDMA*, *EV-DO*, *UMTS*, *Bluetooth*, *Wi-Fi*, *LTE*, and *WiMAX*) dan juga kemampuan dasar *handphone* pada umumnya.

### SCP (Secure Copy Protocol)

SCP merupakan perintah yang digunakan untuk melakukan copy file antara dua host via network atau jaringan komputer. SCP menggunakan SSH untuk mengirim data dan ototentikasi sehingga keamanan yang berlaku pada SSH, seperti data *encryption*, juga berlaku pada SCP.

Untuk menggunakan SCP, ada dua hal yang harus diperhatikan adalah SCP dan SSH telah terinstall serta SSH tersebut berjalan. Selain itu IP *address* atau *hostname* harus diketahui.

### Java

Java merupakan suatu teknologi perangkat lunak yang di dalamnya mencakup bahasa pemrograman. Selain itu Java juga merupakan suatu *platform* yang memiliki *virtualmachine* dan *library* yang diperlukan untuk menulis dan menjalankan suatu program. Bahasa pemrograman. Java dimulai dari sebuah tim pengembang *software* dari Sun *Microsystem* yang dipimpin oleh James Gosling dan Patrick Naughton. Pada tahun 1991, Sun *Microsystem* mengembangkan sebuah bahasa pemrograman yang berukuran kecil untuk diimplementasikan pada alat elektronik rumah tangga seperti *switchbox* TV kabel. Berhubung alat tersebut tidak memiliki banyak memori, maka bahasa yang digunakan harus sangat kecil dan menghasilkan kode yang kecil pula. Permasalahan lainnya adalah alat-alat tersebut memiliki CPU yang berbeda-beda karena dibuat oleh manufaktur yang berbeda. Jadi sangat diharuskan bahasa pemrograman tersebut tidak terikat pada sebuah arsitektur mesin tertentu saja.

Prinsip yang digunakan oleh tim dan Sun Micro-system, hanya saja pada Java mesin virtualnya disebut Java Virtual Machine (JVM). Akan tetapi karena mereka kebanyakan berlatar belakang Unix, mereka membuat bahasa pemrograman tersebut menggunakan bahasa C++ bukan Pascal. Jadi, jika JVM ada, maka kode *ash* program akan dapat dijalankan pada mesin yang berbeda-beda dan platform sistem operasi yang berbeda.

### API dan JDK

API adalah *Application Programming Interface* yaitu sebuah *layer* yang berisi classclass yang sudah didefinisikan dan antarmuka pemrograman yang akan membantu para pengembang aplikasi dalam perancangan sebuah aplikasi.

API memungkinkan para pengembang untuk dapat mengakses fungsi-fungsi sistem operasi yang diizinkan melalui bahasa Java. Pada saat ini dikenal ada tiga buah API dari Java, yaitu:

1. J2SE, Java 2 *Standard Edition* adalah API yang dapat digunakan untuk mengembangkan aplikasi-aplikasi yang bersifat *Client side standalone* atau *applet*.
2. J2EE, Java 2 *Enterprise Edition* adalah API yang digunakan untuk melakukan pengembangan aplikasi-aplikasi yang bersifat *serverside* seperti *Java Servlet*, dan *Java Server Pages*.
3. J2ME, Java 2 *Micro Edition* adalah API yang merupakan *subset* dari J2SE tetapi memiliki kegunaan untuk mengembangkan aplikasi pada *hand held device* seperti *Smart Phone* atau PDA tentu saja yang didalamnya telah ditanamkan interpreter Java.

Sun merilis tiap versi J2SE dengan sebuah JDK (*Java Development Kit*) dan JDK untuk J2SE 6.0 ini versinya adalah JDK 1.6. JDK sebenarnya adalah sekumpulan program kecil yang akan sangat membantu para pengembang aplikasi dalam merancang dan melakukan *testing* program. JDK harus diakses lewat *command line*. Anda jangan bingung oleh istilah JDK 1.6 atau JDK 6.0 karena hal ini mengacu pada hal yang sama. Selain JDK bawaan sendiri dari Sun, ada juga beberapa tool untuk pengembangan aplikasi dari pihak ketiga, contohnya:

1. Jbuilder dari Borland.
2. NetBeans Open Source dan Sun.
3. Sun ONE, yaitu versi komersial dari NetBeans yang dibuat oleh Sun.
4. Eclipse Open Source dari IBM.

### IDE

*Tool-tool* di atas juga memiliki IDE sendiri-sendiri. IDE (*Integrated Development Environment*), yaitu sebuah lingkungan pengembangan aplikasi yang lengkap dan dapat membantu proses pengembangan sebuah aplikasi menjadi lebih cepat. Pada IDE Anda dapat melakukan proses perancangan, kompilasi, *debugging*, *building*, dan dapat menggunakan fasilitas pertolongan *online*. Pada IDE para pengembang hanya perlu memencet tombol fungsi tertentu seperti (F5 atau F7, dan lain- lain) untuk melakukan prosesproses kompilasi, *debugging*, bahkan *building* sebuah aplikasi.

### Android SDK

Android SDK merupakan sebuah tool *development* yang memungkinkan Anda membuat aplikasi untuk *platform* Android. Android SDK sendiri mencakup dari sampel *project* dengan kode sumbernya, *tool development*, sebuah *emulator*, dan librari (baca: file pendukung) yang dibutuhkan untuk membangun aplikasi Android.

Aplikasi ini ditulis menggunakan bahasa pemrograman Java, dan dijalankan pada Dalvik Debug Monitor Server (DDMS), yang merupakan mesin virtual yang dirancang khusus untuk penggunaan *embedded* yang berjalan di atas kernel linux.

### Ubuntu

Ubuntu adalah sistem operasi lengkap berbasis Linux, tersedia secara bebas dan mempunyai dukungan baik yang berasal dari komunitas maupun tenaga ahli profesional. Ubuntu sendiri dikembangkan oleh komunitas sukarelawan

Ubuntu berasal dari bahasa Afrika yakni "*Humanity to Others*" yang berarti "Kemanusiaan Untuk Sesama". Atas dasar itulah diluncurkannya *Linux* Ubuntu yang dirilis pada tahun 2004. Sistem operasi ini adalah merupakan turunan dari sistem operasi *Linux* yang lain, yakni Debian.

Ubuntu itu sendiri dibuat dengan tujuan bahwa, Ubuntu selalu gratis tanpa adanya biaya lisensi, bersifat *open source*, dan siap untuk dipergunakan dalam kondisi yang stabil. Ubuntu didukung oleh perusahaan bernama Canonical, Ltd yang memiliki tujuan untuk membantu perkembangan, distribusi, dan promosi dari produk-produk yang bersifat *open source* (kode terbuka). Perusahaan ini bermarkas di Eropa dan dipimpin oleh seseorang bernama Mark Shuttleworth.

Sejak pertama kali diluncurkan, Ubuntu mendapat perhatian yang sangat besar dari pengguna *Linux* yang lain. Hal ini disebabkan



karena kestabilan yang dimiliki oleh Ubuntu itu sendiri. Selain itu kenyamanan dan kemudahan yang dimiliki Ubuntu menjadi daya tarik yang besar bagi pengguna *Linux* di seluruh belahan dunia.

### Openwrt

Menurut wiki OpenWRT (2011) yang dijelaskan pada <http://wiki.openwrt.org>, OpenWRT digambarkan sebagai distribusi *linux* untuk perangkat *embedded*. OpenWRT menyediakan *filesystem* yang sepenuhnya bisa dirubah sesuai dengan manajemen paket yang tersedia. Hal ini membebaskan dari pemilihan aplikasi dan konfigurasi yang disediakan oleh *vendor* dan memungkinkan untuk menyesuaikan perangkat melalui penggunaan paket sesuai dengan aplikasi apapun. Untuk para pengembang, OpenWRT adalah kerangka kerja untuk membangun aplikasi tanpa harus membangun *firmware* lengkap.

OpenWRT adalah sebuah proyek *open source* untuk menciptakan sebuah sistem operasi gratis (sebenarnya lebih tepat disebut *Firmware*) yang bisa di install (lebih tepatnya ditanam/diembedded) pada perangkat radio *wireless*. Karena dibuat dengan menggunakan kernel *Linux* maka OpenWRT bisa sebut sebagai salah satu distro *Linux* untuk perangkat *embedded* (*embedded devices*).

OpenWRT memiliki sistem file yang bisa dibuat ulang, yang memungkinkan untuk manajemen paket melalui sistem perintah pada terminal/konsole, hal tersebut memungkinkan pengguna untuk menginstall perangkat lunak baru untuk memenuhi kebutuhan individu mereka.

## ANALISA DAN PERANCANGAN

Perancangan sistem dibuat untuk memenuhi kebutuhan pengguna mengenai gambaran yang jelas bagaimana suatu sistem dibentuk. Perancangan sistem ini merupakan tahapan persiapan sistem sebelum diimplementasikan.

### Kebutuhan Perangkat Keras (*Hardware*)

Ada beberapa spesifikasi *hardware* dan peralatan jaringan yang digunakan pada pengujian tugas akhir ini:

1. *Access point* TP-Link WR1043ND
  - a. Processor: Atheros AR9132 @ 400MHz, Architecture: MIPS
  - b. *Interface*: 4 Port 1000/100/10 Switch + 1 Port 1000/100/10 WAN

- c. Memory : 8 MiB Flash + 32 MiB DDR RAM
  - d. 1 x USB 2.0 Host + 3 antena yang dapat dilepas
2. Kabel LAN socket RJ-45 (*Straight*) Laptop

### Kebutuhan Perangkat Lunak (*Software*)

Implementasi VoIP untuk perpindahan agen pada Jaringan Nirkabel dengan Metode *Handoff* ini memerlukan perangkat lunak (*Software*) agar mampu beroperasi sesuai dengan yang diharapkan. Adapun perangkat lunak (*Software*) yang digunakan, yaitu:

1. Softphone yang digunakan untuk menelepon VoIP menggunakan *handphone*
2. *Server* VoIP yang menggunakan Asterisk.

### Analisa Sistem

Jaringan nirkabel merupakan sebuah teknologi dalam jaringan yang memiliki keunggulan yaitu dalam penggunaan peralatan tidak harus terhubung secara fisik. Jadi alat yang akan dihubungkan kedalam jaringan nirkabel tidak membutuhkan kabel, hal ini tentu mempermudah penggunaannya untuk mengkoneksikan peralatan yang digunakan ke jaringan nirkabel yang ada.

Selain itu jaringan nirkabel memiliki kekurangan yaitu jarak dari sinyal yang diberikan oleh sebuah agen terkadang tidak dapat ditentukan secara jelas karena dipengaruhi oleh banyak faktor. Sehingga pengguna kadang tidak sadar bahwa ketika berpindah dari tempat yang satu ke tempat yang lain sinyal *wifi* yang diterima mulai melemah, bahkan sampai terputus koneksinya.

Untuk itu dibutuhkan suatu metode yang bisa menghubungkan koneksi terhadap jaringan nirkabel yang terputus untuk kembali tersambung ke media pemancar sinyal *wifi* yang sinyalnya lebih kuat secara otomatis. Sehingga pengguna tidak mengalami putus koneksi secara tiba-tiba selama terdapat sinyal *wifi* di sekitar pengguna.

Cara yang bisa dilakukan adalah dengan metode *handoff*. Dengan menggunakan metode *handoff* pengguna tidak akan mengalami putus koneksi terhadap jaringan nirkabel yang ada, karena metode *handoff* merupakan sebuah metode dimana sebuah alat akan menangkap sinyal paling bagus, sehingga secara otomatis koneksi juga akan berpindah ke jaringan nirkabel yang sinyal paling bagus.

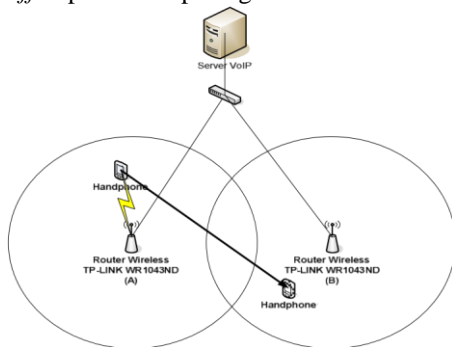
### Perancangan Sistem

Perancangan sistem dibuat bertujuan untuk memberikan gambaran secara umum kepada pengguna dalam pembuatan rancangan sistem yang

baru untuk mempermudah dalam pengaksesan *server*. Sehingga nantinya diharapkan sistem yang dibuat dapat membantu kinerja admin dalam melakukan *monitoring* dan manajemen jaringan. Perancangan sistem yang digunakan dalam sistem *monitoring* dan manajemen jaringan jarak jauh menggunakan android ini adalah dengan menggunakan desain jaringan, proses bisnis, *flowchart* dan desain Perencanaan Antarmuka.

### Desain Jaringan

Desain jaringan yang menjelaskan mengenai beberapa perangkat yang digunakan dalam implementasi VoIP untuk perpindahan agen pada jaringan nirkabel menggunakan metode *handoff* dapat dilihat pada gambar 3.1.

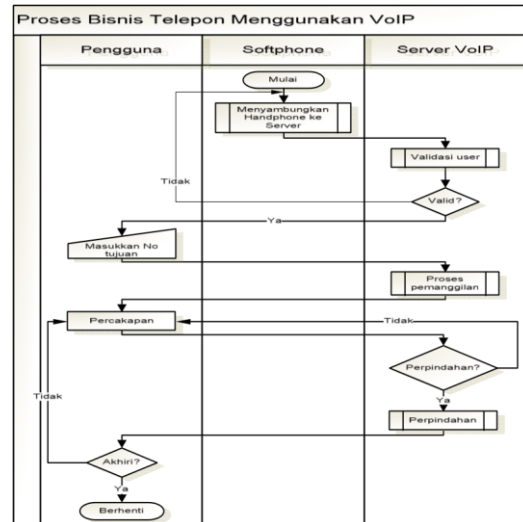


Gambar 3.1 Desain Jaringan

Pada gambar 3.2 Terdapat 2 agen untuk memancarkan sinyal *wifi*. *Handphone* tersambung ke agen A, kemudian pengguna berpindah tempat sampai melewati *range* agen A dan memasuki *range* sinyal agen B. Di dalam *handphone* sudah terdapat Softphone yang digunakan untuk menelepon via VoIP dan juga merupakan sebagai perangkat uji coba perpindahan koneksi dengan metode *handoff*.

### Proses Bisnis

Proses bisnis ini menggambarkan mengenai proses yang terjadi pada implementasi metode *handoff* untuk perpindahan agen pada jaringan nirkabel untuk VoIP.

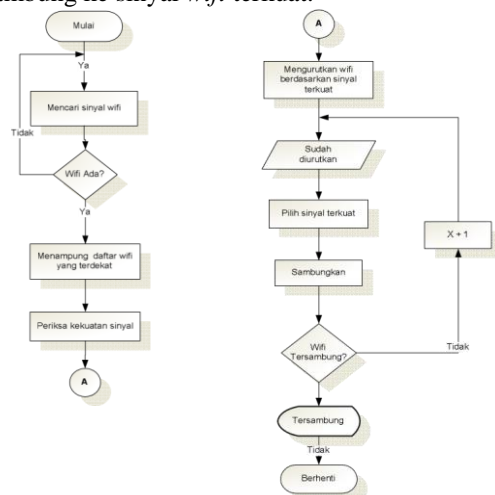


Gambar 3.2 Proses Bisnis

Pada gambar 3.2 dimulai dari softphone menyambungkan *wifi* dengan sinyal terkuat, kemudian dilakukan proses validasi *user* oleh *server*, jika memang *user* valid, langkah selanjutnya adalah pengguna memasukkan nomor tujuan. Setelah itu dilakukan proses pemanggilan nomor tujuan, setelah tersambung pengguna dapat melakukan percakapan. Jika sinyal *wifi* diketahui lemah dan ditemukan sinyal yang lebih kuat, maka dilakukan perpindahan koneksi.

### Flowchart Menyambungkan ke Wifi dengan sinyal terkuat

*Flowchart* *handphone* menyambungkan ke *wifi* terdekat menjelaskan bagaimana proses *handphone* melakukan pencarian sinyal *wifi* sampai tersambung ke sinyal *wifi* terkuat.

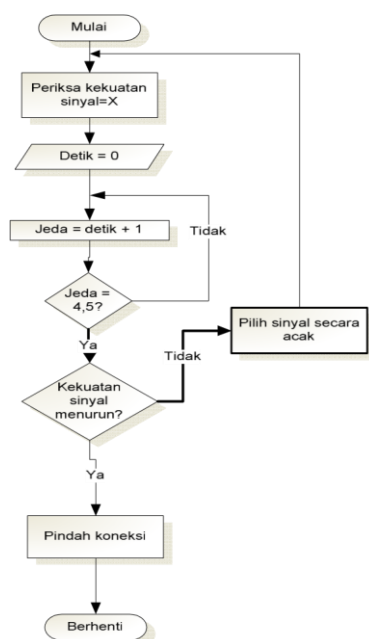


Gambar 3.3 Flowchart menyambungkan ke *wifi* dengan sinyal terkuat.

User menyalakan *wifi* yang ada di *Handphone*, jika *wifi* yang ada pada *handphone* menyala proses akan dilanjutkan yaitu pencarian sinyal *wifi*. Ketika sinyal *wifi* ditemukan lalu sinyal-sinyal yang ada ditampung dulu yang kemudian akan diperiksa kekuatan sinyal. Setelah diperiksa kekuatan sinyal, lalu sinyal-sinyal yang ada diurutkan berdasarkan sinyal terkuat. Sinyal yang sudah berurutan siap untuk dipilih sinyal yang paling kuat yang kemudian disambungkan. Jika *wifi* sudah tersambung akan dilakukan perulangan yang bertujuan untuk mencari sinyal terkuat.

### Flowchart Handoff

Flowchart *handoff* menjelaskan bagaimana proses perpindahan koneksi terjadi.



Gambar 3.4 Flowchart Handoff

## IMPLEMENTASI SISTEM

Pada bab ini, akan dijelaskan bagaimana langkah-langkah yang akan dilakukan untuk mengimple-mentasikan penelitian yang sedang dilakukan sesuai dengan perancangan sistem yang telah dijelaskan pada bab perancangan sistem.

### Konfigurasi IP pada Server

Pada sub-bab konfigurasi IP pada PC Server, dijelaskan pengaturan konfigurasi pada IP Server. Cara *setting* yang digunakan untuk mengatur IP pada server yaitu dengan perintah `gedit /etc/network/interfaces`.

```
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.1.11
netmask 255.255.255.0
gateway 192.168.1.11
```

Gambar 4.1. Konfigurasi IP pada Server

Pada gambar 4.1 menjelaskan IP pada server. Pada eth0 di atur IP static dengan IP 192.168.1.11 dengan *netmask* 255.255.255.0 dan *gateway* kepada IP 192.168.1.11.

### Konfigurasi Asterisk

Pada sub-bab ini dijelaskan pengaturan Asterisk sebagai server VoIP. Mulai dari konfigurasi untuk nama *user*, ID, serta *password* yang nantinya digunakan *user* untuk menggunakan akunnya.

### SIP.conf dalam Asterisk

```
[ligar]
type=friend
host=dynamic
musiconhold=default
context=voiptif
;context=demo
canreinvite=no
username=ligar
secret=ligar
callerid="ligar"
nat=no
dtmfmode=rfc2833
```

Gambar 4.2. Setting sip.conf

Pada gambar 4.2. terdiri dari *username*, *password* dan *callerid*. Ketiganya berguna untuk *user* sebagai data validasi. *User* harus memasukkan data pada *softphone* sipdroid agar dapat sesuai dengan data yang dimasukkan pada server asterisk yang ada.

### Setting extensions.conf dalam asterisk

```
[voiptif]
exten => 100,1,Dial(SIP/jojo)
exten => 101,1,Dial(SIP/ligar)
exten => 102,1,Dial(SIP/nico)
exten => 103,1,Dial(SIP/windi)
```

Gambar 4.3. Setting extensions.conf

Pada gambar 4.3. menjelaskan tentang nomor yang bisa dihubungi untuk *user* ketika akan melakukan panggilan ke nomor yang diinginkan sesuai dengan *user* lain.

## HASIL DAN PEMBAHASAN

### Skenario Pengujian I

Pada sistem ini dihasilkan sebuah scenario pengujian, sehingga dapat menjelaskan urutan sistem yang telah dirancang. Implementasi Perpindahan Agen Pada Jaringan Nirkabel Menggunakan Metode *Handoff* Untuk VoIP meliputi:

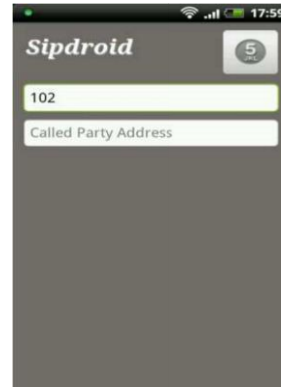
1. *Client* menghubungi admin untuk mendapatkan hak akses login.
2. Menghubungkan telepon seluler ke agen via *wifi*.
3. Menggunakan aplikasi Sipdroid untuk menelepon berbasis VoIP, menggunakan hak akses yang telah didapat dari admin.
4. Mulai melakukan aktifitas percakapan.
5. Server memerintahkan agen untuk selalu memeriksa kekuatan sinyal *client* yang terhubung ke agen.
6. Jika sinyal *client* mulai melemah, maka *server* melalui agen mengatur *handoff* untuk *client*.
7. Server mengambil data *client* yang terhubung dengan agen digunakan sebagai file log.

### Skenario Pengujian II

1. *Client* menghubungi admin untuk mendapatkan hak akses login.
2. Menghubungkan telepon seluler ke agen via *wifi*.
3. Menggunakan aplikasi Sipdroid untuk menelepon berbasis VoIP, menggunakan hak akses yang telah didapat dari admin.
4. Mulai melakukan aktifitas percakapan.
5. Softphone mencari sinyal terbaik dari agen untuk dipilih sebagai koneksi ke server VoIP.

### Hasil

Berikut merupakan dokumentasi hasil dan pembahasan dari implementasi perpindahan agen pada jaringan nirkabel menggunakan metode *handoff* untuk VoIP. Setelah melakukan analisa dan perancangan sistem, maka dilakukan uji coba. Tujuan dari uji program ini adalah untuk mengetahui bagaimana metode *handoff* jika di implementasikan pada jaringan *wifi* lokal.



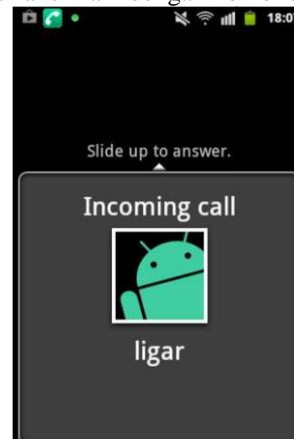
Gambar 5.1. Memasukkan Nomor Tujuan

Pengguna memasukkan nomor tujuan yang ingin dihubungi, dengan ketentuan nomor yang dihubungi telah terdaftar dalam server VoIP dan sama-sama dalam keadaan aktif.



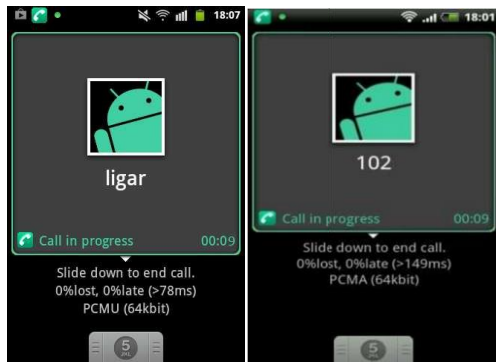
Gambar 5.2. Memanggil ke Nomor Tujuan

Pada gambar 5.2. sedang dilakukan proses pemanggilan akun lain dengan nomor tujuan 102.



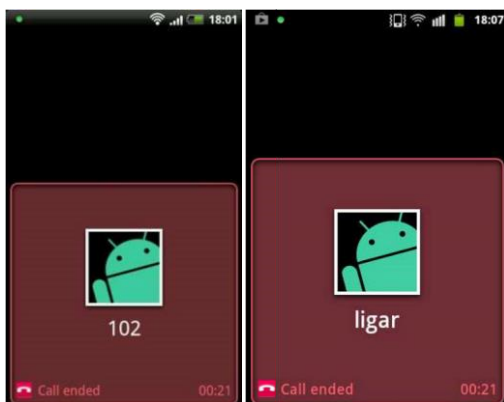
Gambar 5.3. Panggilan Masuk dari Ligar

Pada gambar 5.3. sedang ada panggilan masuk dari akun dengan nama Ligar.



Gambar 5.4. Proses Percakapan

Pada gambar 5.4. *user* sedang melakukan percakapan dengan menggunakan Sipsdroid dengan *user* lain yang bernama *ligar*.



Gambar 5.5. Panggilan dari 102 Diakhiri

Pada gambar 5.5. percakapan diakhiri oleh pengguna VoIP.

```
root@agustianra-router:~# cat /var/dhcp.leases |awk '{print $2_"$3"}' > /tmp/router1.text
root@agustianra-router:~# cat /tmp/router1.text
d4:e8:b2:48:94:e4 192.168.1.234
```

Gambar 5.6. Pencatatan Mac Address dan IP agen

Pada gambar 5.6. digunakan untuk melakukan pencatatan *mac address* dan IP client yang terhubung ke agen 1.

```
root@openmt:~# cat /var/dhcp.leases |awk '{print $2_"$3"}' >> /tmp/router2.text
root@openmt:~# cat /tmp/router2.text
d3:5d:4c:eb:8c:7c 192.168.1.151
00:26:82:30:7d:95 192.168.1.144
00:17:c4:41:3c:a3 192.168.1.111
d4:e8:b2:48:94:e4 192.168.1.234
root@openmt:~#
```

Gambar 5.7. Pencatatan Mac Address dan IP agen

Pada gambar 5.7. digunakan untuk melakukan pencatatan *mac address* dan IP client yang terhubung ke agen 2.

```
root@ligar-laptop:~# scp 192.168.1.1:/tmp/router1.text /var/www/log/data/
router1.text
root@ligar-laptop:~# scp 192.168.1.1:/tmp/router2.text /var/www/log/data/
router2.text
d4:e8:b2:48:94:e4 192.168.1.234
```

Gambar 5.8. SCP dari router 1

Pada gambar 5.8. dilakukan proses pengambilan data dari agen 1 untuk dijadikan file log di server.

```
root@ligar-laptop:~# scp 192.168.1.1:/tmp/router1.text /var/www/log/data/
router1.text
root@ligar-laptop:~# scp 192.168.1.1:/tmp/router2.text /var/www/log/data/
router2.text
d4:e8:b2:48:94:e4 192.168.1.234
```

Gambar 5.9. SCP dari router 2

Pada gambar 5.9. dilakukan proses pengambilan data dari agen 2 untuk dijadikan file log di server.

```
ScanResult bestscan = null;
if (mScanResult != null) {
    //hasil pemrosesan sinyal
    for (final ScanResult scan : mScanResult) {
        for (final WifiConfiguration config : configurations) {
            if (config.SSID != null && config.SSID.equals(scan.SSID)) {
                if (bestscan == null || scan.level > bestscan.level) { //scan level > bestscan level + memilih sinyal
                    bestscan = scan;
                }
            }
        }
    }
    if (config == null) {
        bestscan = null;
    }
}
```

Gambar 5.10. potongan kode dari Sipsdroid

Pada gambar 5.10. merupakan potongan kode dari Sipsdroid yang menunjukkan hasil dari kekuatan sinyal dari agen yang ada, lalu memilih agen dengan kekuatan sinyal paling besar.

## Pembahasan

Berikut dokumentasi hasil dan pembahasan dari Implementasi Perpindahan Agen Pada Jaringan Nirkabel Menggunakan Metode *Handoff* Untuk VoIP.

## Berdasarkan Skenario Pengujian 1

Setelah dilakukan pengujian dimana yang bertugas sebagai penentu sebuah *client* melakukan *handoff* atau tidak terhadap agen diketahui bahwa hal ini tidak bisa terjadi ketika diimplementasikan terhadap *wifi*. Hal ini disebabkan oleh kemampuan agen yang hanya bisa mengetahui kekuatan sinyal *client* yang terhubung kepada satu agen saja. Agen tidak bisa mengetahui kekuatan sinyal yang belum terhubung kepada agen, sehingga *server* pun juga tidak bisa melakukan perintah agar *client* berpindah ke agen yang memiliki sinyal yang lebih kuat.

## Berdasarkan Skenario Pengujian 2

Proses *handoff* dilakukan oleh aplikasi Sipsdroid, yang didalamnya disisipkan kode untuk

selalu memeriksa kekuatan sinyal, kemudian jika memang kekuatan sinyal yang ada lebih besar sinyal yang dipakai sekarang maka koneksi juga pindah ke agen yang sinyalnya lebih kuat.

## PENUTUP

### Simpulan

Berdasarkan uraian pembahasan analisa dan pengujian yang telah dilakukan, maka dapat diambil kesimpulan dari implementasi perpindahan agen pada jaringan nirkabel menggunakan metode *handoff* untuk VoIP yaitu sebagai berikut :

1. Dengan menambah jumlah agen, maka jangkauan area yang bisa mendapatkan sinyal *wifi* semakin luas.
2. Digunakannya metode *handoff* pengguna tidak harus melakukan konfigurasi ulang ketika sinyal *wifi* mulai melemah karena konfigurasi ulang dilakukan secara otomatis.
3. Implementasi *handoff* pada *wifi* lokal, tidak bisa jika pengaturan untuk berpindah *access point* (agen) dilakukan oleh *server*, hal ini disebabkan oleh keterbatasan agen yang hanya bisa mendeteksi kekuatan sinyal *client* (handphone) ketika *client* terhubung ke agen.

### Saran

Saran-saran yang dapat diambil dari kesimpulan diatas adalah sebagai berikut :

1. Sistem ini belum bisa menentukan apabila ada sinyal *wifi* yang lebih kuat tetapi bukan bagian dari agen yang telah ditentukan oleh *server*.
2. Untuk pengembangan, sistem ini dapat dijadikan referensi untuk meningkatkan kehandalan perpindahan koneksi antar agen.

## DAFTAR PUSTAKA

- [1] Ardiyansah Dian. 2006. *Teknologi Jaringan Komputer*. Jakarta
- [2] Febriani, SKOM, MM. 2010. *Flowchart*. gunadarma.ac.id. Diakses tanggal 2 November 2012.
- [3] Irawan Budhi. 2007. *Jaringan Komputer*. Graha Ilmu. Yogyakarta.
- [4] R.A, Agustian, (2011), *Rancang Bangun Sistem Keamanan Rumah dengan Menggunakan Perangkat Nirkabel Berbasis OpenWRT*, Tugas Akhir Teknik Informatika, Universitas Wijaya Kusuma Surabaya.
- [5] Suprianto.2009. *Membuat Layanan IP PBX*. Jakarta.
- [6] Wiki OpenWRT. 2012. *TP-Link WR1043ND*. Wiki.openwrt.org. Diakses Tanggal 26 Desember 2012.
- [7] Yair Amir, Claudiu Danilov, Michael Hilsdale, Raluca Mus aloiu-Elefteri, Nilo Rivera. *Fast Handoff for Seamless Wireless Mesh Networks*.8-12
- [8] www.ilmukomputer.org. Yogyakarta. Diakses tanggal 20 November 2012.