



Melek IT

Program Studi
Teknik Informatika
Universitas Wijaya Kusuma Surabaya

Jurnal Teknologi Informasi Dan Komunikasi

Volume 2 No 2 Juli 2013

ISSN 2252-9128



PERANCANGAN SISTEM BANDWIDTH MANAGEMENT PADA PROXY SERVER MENGGUNAKAN HIERARCHY TOKEN BUCKET. Indra Kumiawan, F.X. Wisnu Yudo Untoro.

SISTEM OPTIMALISASI CACHE PROXY MENGGUNAKAN METODE REPLACEMENT PADA SQUID PROXY SERVER. Erwin Damei Hadi Putra, Tjatusari Widiartin.

IMPLEMENTASI APLIKASI PENGENDALIAN JARAK JAUH KAMERA DSLR VIA BLUETOOTH BERBASIS ANDROID. Abednego Eko Harjiyanto, Guendra Kusuma Wardhana.

IMPLEMENTASI WEB SERVICE BERBASIS SISTEM INFORMASI GEOGRAFIS UNTUK PENCARIAN LOKASI RUMAH MAKAN. Wira Lamba, Nia Saurina.

SISTEM OPTIMALISASI BANDWIDTH MENGGUNAKAN METODE LOAD BALANCING PADA SQUID PROXY SERVER. Ayu Dian Pertiwi, Anang Kukuh A, Beny Y.V. Nasution.

SISTEM INFORMASI MANAJEMEN RAPAT. Achmad Nashihuddin, Nonot Wisnu Karyanto.

HYBRID ARTIFICIAL BEE COLONY- DIVERSITY SEBAGAI ALGORITMA PEMBELAJARAN PADA NEURAL NETWORK. Shofiya Syidada, Abidatul Izzah, Nuru Aini.

IMPUTASI MISSING DATA MENGGUNAKAN METODE K-NEAREST NEIGHBOUR DENGAN OPTIMASI ALGORITMA GENETIKA. Abidatul Izzah, Nur Hayatin.

PREDIKSI CACAT PERANGKAT LUNAK PADA KODE SUMBER DENGAN SVM DAN SELEKSI ATRIBUT. Ratih Kartika Dewi, Sarwosri.

PEMBUATAN MODEL PENILAIAN KELAYAKAN ASPEK MANUSIA PADA PROYEK PERANGKAT LUNAK UNTUK MENINGKATKAN KUALITAS PRODUK PERANGKAT LUNAK. Parma Hadi R, Sarwosri.

SELEKSI FITUR TASK-STATE FMRI MENGGUNAKAN NON-OVERLAPPING CROSS-CORRELATION. Agus Eko Minarno, Fitri Bimantoro, Arrie Kumiawardhani.



Daftar Isi

1. **PERANCANGAN SISTEM *BANDWIDTH MANAGEMENT* PADA *PROXY SERVER* MENGGUNAKAN *HIERARCHY TOKEN BUCKET*.** Indra Kurniawan, F.X. Wisnu Yudo Untoro (Hal. 1-18)
2. **SISTEM OPTIMALISASI *CACHE PROXY* MENGGUNAKAN METODE *REPLACEMENT* PADA *SQUID PROXY SERVER*** Erwin Darni Hadi Putra, Tjatusari Widiartin. (Hal. 19 - 36)
3. **IMPLEMENTASI APLIKASI PENGENDALIAN JARAK JAUH KAMERA DSLR VIA BLUETOOTH BERBASIS ANDROID.** Abednego Eko Harjiyanto , Guendra Kusuma Wardhana (Hal. 37 - 46)
4. **IMPLEMENTASI WEB SERVICE BERBASIS SISTEM INFORMASI GEOGRAFIS UNTUK PENCARIAN LOKASI RUMAH MAKAN.** Wira Lamba, Nia Saurina (Hal. 47-62)
5. **SISTEM OPTIMALISASI *BANDWIDTH* MENGGUNAKAN METODE *LOAD BALANCING* PADA *SQUID PROXY SERVER*.** Ayu Dian Pertiwi, Anang Kukuh A, Beny Y.V. Nasution (Hal. 63-72)
6. **SISTEM INFORMASI MANAJEMEN RAPAT.** Achmad Nashihuddin, Nonot Wisnu Karyanto (Hal. 73-88)
7. **HYBRID *ARTIFICIAL BEE COLONY- DIVERSITY* SEBAGAI ALGORITMA PEMBELAJARAN PADA *NEURAL NETWORK*** Shofiya Syidada, Abidatul Izzah, Nuru Aini (Hal. 89-94)
8. **IMPUTASI *MISSING DATA* MENGGUNAKAN METODE *K-NEAREST NEIGHBOUR* DENGAN OPTIMASI ALGORITMA GENETIKA.** Abidatul Izzah, Nur Hayatin (Hal. 95-102)
9. **PREDIKSI CACAT PERANGKAT LUNAK PADA KODE SUMBER DENGAN SVM DAN SELEKSI ATRIBUT.** Ratih Kartika Dewi, Sarwosri (Hal. 103-110)
10. **PEMBUATAN MODEL PENILAIAN KELAYAKAN ASPEK MANUSIA PADA PROYEK PERANGKAT LUNAK UNTUK MENINGKATKAN KUALITAS PRODUK PERANGKAT LUNAK.** Parma Hadi Rantelinggi, Sarwosri (Hal. 111-118)
11. **SELEKSI FITUR TASK-STATE FMRI MENGGUNAKAN *NON-OVERLAPPING CROSS-CORRELATION*** Agus Eko Minamo, Fitri Bimantoro, Arrie Kurniawardhani (Hal. 119-130)

(1)

PERANCANGAN SISTEM *BANDWIDTH MANAGEMENT* PADA *PROXY SERVER* MENGUNAKAN *HIERARCHY TOKEN BUCKET*

Indra Kurniawan¹, F. X. Wisnu Yudo Untoro²

Program Studi Teknik Informatika, Universitas Wijaya Kusuma Surabaya, Jl. Dukuh Kupang XXV/ 54
Email : indra300990@gmail.com

ABSTRAK

Kebutuhan akan akses internet dewasa ini sangat tinggi. Dimana dalam suatu jaringan yang terhubung dengan internet, kecepatan *upload* maupun *download* merupakan hal yang sangat penting untuk memperlancar transmisi data.

Dalam sebuah jaringan *hotspot*, sering kali terjadi adanya dominasi *bandwidth* antar pengguna yang diakibatkan salah satu atau beberapa pengguna belum ter-authentikasi melakukan *download* sehingga akan mengganggu pengguna lain. Munculah ide untuk membangun sebuah sistem *bandwidth management* pada *proxy server* menggunakan *hierarchy token bucket*. Dimana di dalam sistem ini berfungsi mengelola pengguna yang belum ter-authentikasi agar dapat di bedakan mana yang dosen mana yang mahasiswa dan mengelola *bandwidth* (*bandwidth management*) yang tersedia dalam jaringan tersebut menggunakan *hierarchy token bucket* (*htb*) dimana salah satunya yaitu dengan membatasi pemakaian kapasitas *bandwidth* tiap pengguna.

Di harapkan setelah menggunakan sistem ini penggunaan *bandwidth* yang ada lebih efektif, lebih optimal dan tidak terjadi lagi perebutan *bandwidth* oleh *user*.

Kata Kunci : *bandwidth, bandwidth management, internet, hierarchy token bucket.*

PENDAHULUAN

Latar Belakang

Kebutuhan akan akses internet dewasa ini sangat tinggi. Dimana dalam suatu jaringan yang terhubung dengan internet, kecepatan *upload* maupun *download* merupakan hal yang sangat penting untuk memperlancar transmisi data. Banyak hal yang dapat mempengaruhi kecepatan dua proses tersebut, diantaranya yaitu besarnya *bandwidth* yang digunakan jaringan tersebut dan seberapa efektif *bandwidth* tersebut bisa dimanfaatkan. *Bandwidth* adalah suatu ukuran dari banyaknya informasi yang dapat mengalir dari satu tempat ke tempat lain dalam satu waktu tertentu. *Bandwidth* dapat dipakai untuk mengukur aliran data analog maupun aliran data digital. Sekarang telah menjadi umum jika kata *bandwidth* lebih banyak digunakan untuk mengukur aliran data digital. Satuan *bandwidth* adalah *bits per second* atau sering disingkat *bps*. Jadi semakin besar *bandwidth* yang digunakan oleh suatu jaringan, seharusnya transmisi data pada jaringan tersebut akan menjadi semakin

cepat. Misalnya penggunaan *bandwidth* di jaringan hotspot kampus, Dengan adanya jaringan hotspot kampus bisa di gunakan untuk mencari informasi dengan mudah dan cepat.

Penggunaan *bandwidth* di sebuah jaringan hotspot kampus sering kali kurang dimanfaatkan secara optimal. Hal ini dapat disebabkan oleh adanya satu atau lebih pengguna yang belum ter-authentikasi menghabiskan kapasitas *bandwidth* dalam jaringan tersebut untuk *download* dengan menggunakan aplikasi *download manager* yang dapat menyita kapasitas *bandwidth*. Sehingga akibatnya pengguna lain tidak mendapatkan porsi *bandwidth* yang sesuai atau memadai. Keadaan yang seperti itu bisa mengganggu aktivitas *user* yang memang menggunakan internet sesuai kebutuhannya, seperti mencari informasi terbaru untuk menambah wawasan dan pengetahuan mereka seputar perkuliahan. Atau mengakses aplikasi yang sebenarnya hanya memerlukan *bandwidth* yang kecil, membutuhkan waktu yang lama untuk mengakses aplikasi yang diinginkan.

Dalam sebuah jaringan hotspot, sering kali terjadi adanya dominasi *bandwidth* antar pengguna

yang diakibatkan salah satu atau beberapa pengguna belum ter-authentikasi melakukan *download* sehingga akan mengganggu pengguna lain. Munculah ide untuk merancang sebuah sistem *bandwidth management* pada *proxy server* menggunakan *hierarchy token bucket*. Dimana di dalam sistem ini berfungsi mengelola pengguna yang belum ter-authentikasi agar dapat di bedakan mana yang dosen mana yang mahasiswa dan mengelola *bandwidth (bandwidth management)* yang tersedia dalam jaringan tersebut dimana salah satunya yaitu dengan membatasi pemakaian kapasitas *bandwidth* tiap pengguna. Dengan demikian jika ada pengguna yang mengakses internet yang membutuhkan kapasitas *bandwidth* yang besar, maka pengguna lain tidak akan terganggu, karena tiap-tiap pengguna sudah mempunyai kapasitas *bandwidth* masing masing yang dapat dipakai untuk mengakses internet.

Di harapkan setelah menggunakan sistem ini penggunaan *bandwidth* yang ada lebih efektif, lebih optimal dan tidak terjadi lagi perebutan *bandwidth* oleh *user*. Dan sarana dan prasarana internet di kampus tersebut lebih baik dan nyaman untuk di gunakan oleh *user*.

TINJAUAN PUSTAKA

Pengertian Bandwidth Management

Penggunaan Internet bersama pastinya mempengaruhi *bandwidth* dan kecepatan transfer data antar komputer. Oleh karena itu dibutuhkan manajemen *bandwidth* agar tidak terjadi penguasaan *bandwidth* secara tunggal. Dapat dibayangkan bagaimana jika terjadi penguasaan *bandwidth*, hal ini akan mempengaruhi koneksi pengguna lain. Tanpa manajemen *bandwidth*, setiap komputer akan secara otomatis memperluas *bandwidth*-nya tergantung kebutuhan yang terbesar. Sehingga, apabila ada yang *download* video dan *file* berukuran besar maupun *streaming* yang akan berakibat melambatnya koneksi pada komputer lain. Hal ini disebut dengan *fair usage* atau penggunaan yang adil.

Pengertian Proxy Server

Proxy dapat dipahami sebagai pihak ketiga yang berdiri ditengah-tengah antara kedua pihak yang saling berhubungan dan berfungsi sebagai perantara, sedemikian sehingga pihak pertama dan pihak kedua tidak secara langsung berhubungan, akan tetapi masing-masing berhubungan dengan perantara, yaitu *proxy*. Sebuah analogi; bila seorang mahasiswa meminjam buku di perpustakaan, kadang si mahasiswa tidak diperbolehkan langsung mencari

dan mengambil sendiri buku yang kita inginkan dari rak, tetapi kita meminta buku tersebut kepada petugas, tentu saja dengan memberikan nomor atau kode bukunya, dan kemudian petugas tersebut yang akan mencarinya dan mengambilkan bukunya.

Dalam kasus diatas, petugas perpustakaan tersebut telah bertindak sebagai perantara atau *Proxy*. Petugas tersebut juga bisa memastikan dan menjaga misalnya, agar mahasiswa hanya bisa meminjam buku untuk mahasiswa, dosen boleh meminjam buku semua buku, atau masyarakat umum hanya boleh meminjam buku tertentu. Mungkin proses tersebut menjadi lebih lama dibandingkan bila kita langsung mencari dan mengambil sendiri buku yang kita inginkan.

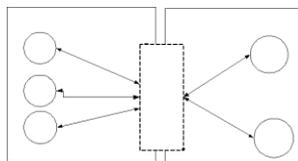
Namun bila saja setiap kali petugas mencari dan mengambil buku untuk seseorang, si petugas juga membuat beberapa salinan dari buku tersebut sebelum memberikan bukunya kepada orang yang meminta, dan menyimpannya di atas meja pelayanan, maka bila ada orang lain yang meminta buku tertentu, sangat besar kemungkinan buku yang diminta sudah tersedia salinannya diatas meja, dan si petugas tinggal memberikannya langsung. Hasilnya adalah layanan yang lebih cepat dan sekaligus keamanan yang baik.

Analogi diatas menjelaskan konsep dan fungsi dasar dari suatu *proxy* dalam komunikasi jaringan komputer dan internet. *Proxy server* mempunyai 3 fungsi utama yaitu *Connection Sharing*, *Filtering* dan *Caching*. Masing masing fungsi akan dijelaskan lebih lanjut dibawah. *Proxy* dalam pengertiannya sebagai perantara, bekerja dalam berbagai jenis protokol komunikasi jaringan dan dapat berada pada level-level yang berbeda pada hirarki *layer* protokol komunikasi jaringan. Suatu perantara dapat saja bekerja pada *layer Data-Link*, *layer Network* dan *Transport*, maupun *layer Aplikasi* dalam hirarki *layer* komunikasi jaringan menurut *OSI*. Namun pengertian *proxy server* sebagian besar adalah untuk menunjuk suatu *server* yang bekerja sebagai *proxy* pada *layer Aplikasi*, meskipun juga akan dibahas mengenai *proxy* pada level sirkuit.

Dalam suatu jaringan lokal yang terhubung ke jaringan lain atau internet, pengguna tidak langsung berhubungan dengan jaringan luar atau internet, tetapi harus melewati suatu *gateway*, yang bertindak sebagai batas antara jaringan lokal dan jaringan luar. *Gateway* ini sangat penting, karena jaringan lokal harus dapat dilindungi dengan baik dari bahaya yang mungkin berasal dari internet, dan hal tersebut

akan sulit dilakukan bila tidak ada garis batas yang jelas jaringan lokal dan internet. *Gateway* juga bertindak sebagai titik dimana sejumlah koneksi dari pengguna lokal akan terhubung kepadanya, dan suatu koneksi ke jaringan luar juga terhubung kepadanya.

Dengan demikian, koneksi dari jaringan lokal ke internet akan menggunakan sambungan yang dimiliki oleh *gateway* secara bersama-sama (*connection sharing*). Dalam hal ini, *gateway* adalah juga sebagai *proxy server*, karena menyediakan layanan sebagai perantara antara jaringan lokal dan jaringan luar atau internet. Diagram berikut menggambarkan posisi dan fungsi dari *proxy server*, diantara pengguna dan penyedia layanan:



Gambar 2.1 diagram posisi dan fungsi *proxy server*.

Pada gambar 2.1 diatas di jelaskan bahwa *proxy server* adalah sebagai *gateway* atau jembatan antara jaringan lokal dan jaringan luar atau internet.

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*.

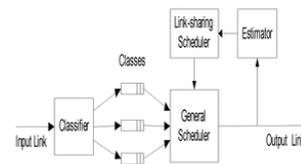
Hierarchy Token Bucket

Hierarchical Token Bucket (HTB) merupakan teknik penjadwalan paket yang baru-baru ini diperkenalkan bagi *router* berbasis *Linux*, dikembangkan pertama kali oleh Martin Devera pada akhir 2001 untuk diproyeksikan sebagai pilihan (atau pengganti) mekanisme penjadwalan yang saat ini masih banyak dipakai yaitu *CBQ*.

Sebagai catatan, apabila nilai *ceil* sama dengan nilai *base link*, maka akan memiliki fungsi yang sama seperti parameter *bounded* pada *CBQ*, di mana kelas-kelas tidak diijinkan untuk meminjam *bandwidth*. Sedangkan jika nilai *ceil* diset tak terbatas atau dengan nilai yang lebih tinggi seperti kecepatan *link* yang dimiliki, maka akan didapat fungsi yang sama seperti kelas *non-bounded*.

Istilah *bandwidth management* sering diartikan dengan istilah *traffic control*, yang dapat didefinisikan sebagai pengalokasian yang tepat dari suatu *bandwidth* untuk mendukung kebutuhan atau keperluan aplikasi atau suatu layanan jaringan. Istilah *bandwidth* dapat didefinisikan sebagai kapasitas atau daya tampung suatu *channel* komunikasi (*medium* komunikasi) untuk dapat dilewati sejumlah *traffic* informasi atau data dalam satuan waktu tertentu. Umumnya *bandwidth* dihitung dalam satuan *bit*, *kbit* atau *bps* (*byte per second*). Pengalokasian *bandwidth* yang tepat dapat menjadi salah satu metode dalam memberikan jaminan kualitas suatu layanan jaringan (*QoS = Quality Of Services*).

Teknik antrian *Hierarchical Token Bucket* mirip dengan *CBQ* hanya perbedaannya terletak pada opsi, *Hierarchical Token Bucket* lebih sedikit opsi saat konfigurasi serta lebih presisi. Teknik antrian *Hierarchical Token Bucket* memberikan kita fasilitas pembatasan trafik pada setiap level maupun klasifikasi, *bandwidth* yang tidak terpakai bisa digunakan oleh klasifikasi yang lebih rendah. Kita juga dapat melihat *Hierarchical Token Bucket* seperti suatu struktur organisasi dimana pada setiap bagian memiliki wewenang dan mampu membantu bagian lain yang memerlukan, teknik antrian *Hierarchical Token Bucket* sangat cocok diterapkan pada perusahaan atau institusi pendidikan dengan banyak struktur organisasi.



Gambar 2.2 *hierarchy token bucket*.

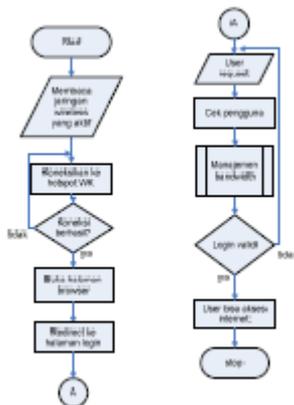
ANALISA DAN PERANCANGAN

Perencanaan Konsep Sistem

Sistem *bandwidth management* pada *proxy server* menggunakan *hierarchy token bucket* yang akan dibangun mempunyai fitur yaitu membuat autentikasi pengguna *hotspot* yang di integrasikan dengan manajemen *bandwidth*. salah satu cara yaitu

dengan membuat beberapa tingkatan dalam autentikasi tersebut. Tingkatan tersebut di bagi menjadi dua yaitu dosen dan mahasiswa. Masing-masing tingkatan tersebut sudah mendapatkan kapasitas *bandwidth* yang di tentukan.

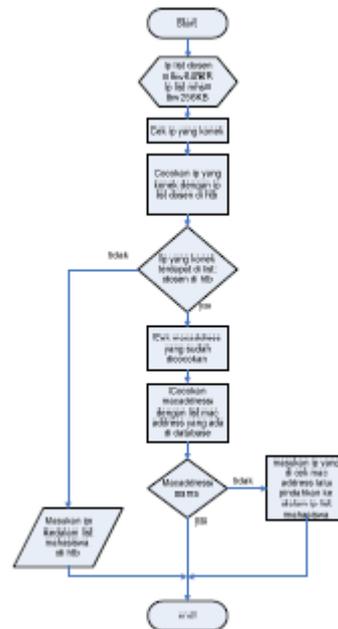
Apabila ada pengguna yang menghabiskan kapasitas *bandwidth* dalam jaringan tersebut dengan cara *men-download* menggunakan aplikasi *download manager* dan akibatnya pengguna lain tidak mendapatkan porsi *bandwidth* yang sesuai atau memadai, bisa di atasi dengan sistem ini. Dengan demikian jika ada pengguna yang mengakses internet yang membutuhkan kapasitas *bandwidth* yang besar, maka pengguna lain tidak akan terganggu, karena tiap-tiap pengguna sudah mempunyai kapasitas *bandwidth* masing-masing yang dapat dipakai untuk mengakses internet. Dan di harapkan setelah menggunakan sistem ini penggunaan *hotspot* lebih optimal dan tidak terjadi perebutan *bandwidth* kembali oleh *user* yang menggunakan fasilitas tersebut. Dan sarana dan prasarana internet kampus lebih baik dan nyaman untuk digunakan. Untuk lebih jelasnya bisa melihat desain *flowchart* sistem manajemen *bandwidth* berikut:



Gambar 3.1 *flowchart* sistem manajemen *bandwidth*

Pada gambar 3.1 diatas merupakan runtutan cara kerja sistem manajemen *bandwidth* sebagai berikut, pertama setiap *user* yang masuk kedalam *hotspot* kita lewat *wireless* dan mencoba untuk *browsing* internet, semuanya akan *redirect* ke *login* yang meminta klien untuk memasukkan *username* dan *password*. Ketika *username* dan *password* telah dimasukkan maka sistem akan mengecek apakah ada *username* dan *password* yang dimasukkan oleh si *user* bersangkutan. Lalu ada proses manajemen *bandwidth* dimana di dalamnya ada proses lagi yaitu proses untuk manajemen *bandwidth* untuk *user*. Sistem akan memproses

apakah *user* tersebut login sebagai dosen atau mahasiswa. Jika sudah, maka sistem akan memberikan izin sehingga si *user* bisa *surfing* di internet, dan jika tidak, maka si akan meminta login ulang dan begitu seterusnya. Disini penjelasan proses manajemen *bandwidth*:



Gambar 3.2 *flowchart* proses manajemen *Bandwidth*

Pada gambar 3.2 proses manajemen *bandwidth* dimana di Perencanaan kebutuhan sistem di tentukan parameter bahwa ip list dosen di *htb* mendapatkan *bandwidth* 640KB, dan ip list mahasiswa di *htb* mendapatkan *bandwidth* 256KB. Pertama sistem akan mengecek ip yang konek. Setelah itu sistem akan mencocokkan ip yang konek dengan ip list dosen. Disini ip list dosen sebagai acuannya karena disini ip list dosen lebih besar kapasitas *bandwidth* yang di dapat. Jika ip yang konek tidak terdapat maka sistem akan langsung memasukan kedalam ip list mahasiswa di dalam *htb*, dan Jika ip yang konek terdapat di ip list dosen maka sistem akan mengecek *mac address* ip tersebut. Setelah di cek, maka *mac address* tersebut akan di cocokkan dengan data *mac address* di dalam *database*. Jika *mac address* tersebut cocok maka biarkan. Dan jika tidak cocok maka sistem akan menghapus ip tersebut yang sudah terdapat di dalam ip list dosen pindahkan ke dalam ip list mahasiswa. Setelah itu selesai.

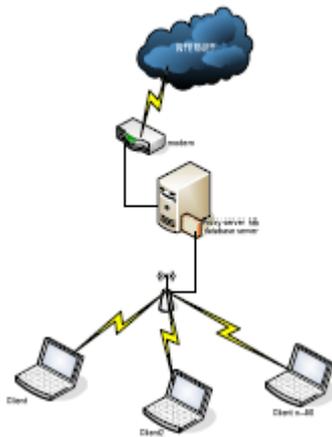
Perencanaan kebutuhan sistem

Berisi perancangan (desain) dari perangkat keras maupun lunak yang akan digunakan dalam

melakukan simulasi sistem *hotspot*, penentuan perangkat lunak dan topologi yang akan digunakan, sekaligus pengaturan perangkat keras tersebut agar sesuai dengan topologi yang diinginkan.

Desain Jaringan yang di gunakan

Topologi jaringan komputer nirkabel yang akan digunakan penulis terhadap studi literatur yang telah dilakukan yaitu topologi dengan konsep *Portal*, dimana konsep dari topologi ini ialah topologi jaringan yang umum digunakan untuk *hotspot*. *Hotspot* menjadi portal untuk akses bagi *pc client*.



Gambar 3.3 desain jaringan *proxy server*

Pada gambar 3.3 diatas terdapat *server proxy* yang terkoneksi dengan modem internet. Dimana fungsi *proxy server* yaitu sebagai *gateway* atau batas jaringan local dengan jaringan luar. Didalam *server proxy* juga ada *htb* (*hierarchy token bucket*) yang berfungsi sebagai aplikasi untuk me-manajemen *bandwidth* pengguna. Didalam *server proxy* tersebut juga terdapat *Mysql* sebagai *database servernya*. Internet tersebut disebar oleh *server proxy* menggunakan *access point*, sehingga internet yang yang disebar tersebut bisa di akses oleh *client* menggunakan *wifi*.

Hardware yang digunakan

Ada beberapa spesifikasi *hardware* dan peralatan jaringan yang di gunakan pada pengujian tugas akhir ini yaitu:

1. Router TPLINK MR3420



Gambar 3.4 Router *Wireless* TP-Link MR3420
Tampak Depan & Belakang.

Spesifikasi : *USB 2.0 Port for LTE/ HSDPA/ HSDPA/ UMTS/ EVDO USB Modem 1 10/100Mbps WAN Port, 4 10/100Mbps LAN Ports, support the auto-Negotiation and auto-MDI/MDIX.*

2. Modem smartfren CE682 REV A
3. *Server* laptop HP COMPAQ PRESARIO CQ40

Software Yang Digunakan

Selain kebutuhan *hardware* yang telah di jelaskan diatas, ada beberapa *software* yang dibutuhkan untuk dapat dilakukannya pengujian sistem *bandwidth management* menggunakan *hierarchy token bucket* tanpa ada masalah. *Software-software* yang digunakan adalah sebagai berikut:

1. Laptop *server*
 - a. sistem operasi : *linux ubuntu server 10.04*
 - b. aplikasi : *squid , MySQL, HTB*
2. Laptop klien
 - a. sistem operasi : *Windows 7 ultimate SP1 x32*

Perencanaan pengujian

Pada penelitian yang dilakukan di jaringan *wireless* yang ada. Penulis menemukan kriteria untuk di lakukan pengujian. Kriteria tersebut di bagi menjadi 3 bagian:

1. Pengujian *login user* pada *proxy server*.

Pada pengujian ini adalah memeriksa apakah *user* yang telah terautentikasi dengan benar sehingga dapat melakukan akses internet keluar cara kerjanya saat seorang pengguna berusaha untuk melakukan *browsing* ke Internet, *proxy* akan memaksa pengguna yang belum terautentikasi untuk menuju ke *Authentication web* dan akan di beri *prompt login*. Jika *username* dan *password* yang di masukkan benar maka *user* bisa mengakses internet luar.

2. Pengujian manajemen user

Pengujian ini dilakukan untuk mengetahui apakah bisa jika ada klien yang ingin masuk jika jumlah klien yang di tentukan itu sudah penuh.

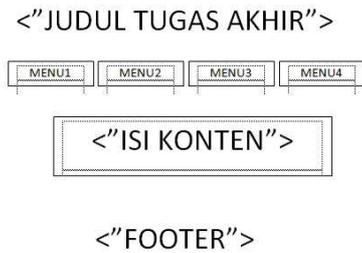
3. Pengujian *download* klien

Pengujian kali ini bertujuan agar bisa

menunjukkan seberapa besar komputer klien jika mendownload menggunakan IDM (*Internet Download Manager*).

Perencanaan antarmuka

Pada perencanaan antarmuka, digunakan sebagai *format* desain yang digunakan untuk setiap tampilan menu yang nantinya dijadikan sebagai *template* utama. Bisa dilihat pada gambar 3.5:



Gambar 3.5 Desain Perencanaan Antarmuka.

Pada gambar 3.5, desain perencanaan antarmuka memiliki 4 bagian yang diantaranya adalah sebagai berikut:

1. Judul tugas akhir (berisi judul dari tugas akhir).
2. MENU1-4 (berisi menu-menu dari modul yang tersedia).
3. ISI KONTEN (berisi konten dari masing-masing modul).
4. FOOTER (berisi identitas pembuat).

IMPLEMENTASI SISTEM

Konfigurasi Pada komputer Server

Komputer server yang akan dibuat ini supaya *client* bisa terhubung dengan server atau untuk menerima permintaan dari *client*, sehingga di lakukan persiapan supaya komputer server dalam pengujian bisa sesuai dengan yang diinginkan. Adapun persiapan tersebut antara lain.

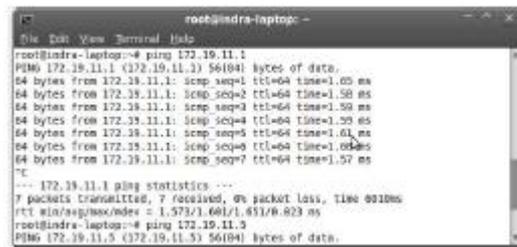
Pemasangan IP Address Server

Pada komputer server ini akan diisikan *IP Address* sebagai *IP* fisik dari komputer server tersebut. Adapun isi dari *file/etc/network/interfaces* yang terletak di *Terminal Console* sebagai berikut.

```
1. Auto lo
2. iface lo inet loopback
3. auto eth0
4. iface eth0 inet static
5.     address 103.29.231.26
6.     netmask 255.255.255.0
7.     gateway 103.29.231.17
8. auto eth1
9. iface eth1 inet static
10.     address 172.19.11.2
11.     netmask 255.255.255.0
12.     gateway 172.19.11.1
```

Gambar 4.1 IP Address eth0

Penjelasan 4.1 adalah, *eth0* dan *eth1* merupakan *interfaces card* yang sesungguhnya pada komputer server. Untuk menguji berhasil apa tidaknya pemasangan *IP address* pada komputer server maka akan dilakukan perintah sebagai berikut :



Gambar 4.2 Ping 172.19.11.1

Konfigurasi Router

Pada komputer server ini membutuhkan sebuah *router* untuk menyebarkan internet yang nantinya akan di akses oleh *user*. Disini *router* di konfigurasi DHCP (*Dynamic Host Configuration Protokol*). Dimana nanti *user* yang mengakses *router* tersebut akan mendapatkan *ipaddress* secara otomatis. supaya penulis bisa menggunakan komputer server sesuai dengan konsep yang sudah dibuat. Adapun langkah-langkah untuk mengkonfigurasi *router* tersebut adalah sebagai berikut :

1. buka *browser* dan akses *ipaddress router* yaitu 172.19.11.1, maka akan muncul tampilan *tplink router* . Pilih DHCP – DHCP Setting. Untuk lebih jelas bisa di gambarkan dengan gambar sebagai berikut:

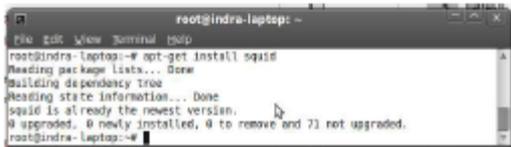


Gambar 4.3 Setting DHCP

Pemasangan Proxy Server

Penggunaan *Proxy Server* atau yang biasa disebut dengan *application firewall*, yang bertujuan perantara antara jaringan luar dan jaringan lokal. sehingga dengan menggunakan *proxy* memudahkan kita mengatur / mengontrol akses internet dan me-manajemen *bandwidth* bagi *user*. Adapun langkah-langkah yang dilakukan untuk membuat *Proxy Server*, sebagai berikut :

1. Instalasi *Proxy Server* :
 - `root@indra-laptop:~# apt-get install squid` untuk lebih jelasnya bisa lihat gambar berikut:



Gambar 4.4 Proses Install Squid

- a. Konfigurasi *squid* agar menjadi mode *ncsa authentication* dengan cara sebagai berikut:

1. Buat *username* dan *password* `root@indra-laptop:~# htpasswd /etc/squid/passwd namauser`
Output: `New password: Re-type new password:`
 Adding password for user `namauser`

2. Setelah itu pastikan bahwa *squid* bisa membaca *file passwd* tersebut dengan cara :

```
root@indra-laptop:~# chmod o+r /etc/squid/passwd
root@indra-laptop:~# dpkg -L squid | grep ncsa_auth
```

Output: `/usr/lib/squid/ncsa_auth`

- b. Ubah dan menambahkan perintah di bawah ini pada *file gedit /etc/squid/squid.conf* yang terletak di *Terminal Console* :

1. `auth_param basic program /usr/lib/squid/ncsa_auth`
2. `/etc/squid/passwd`
3. `auth_param basic children 5`
4. `auth_param basic realm Squid proxy-caching web server`
5. `auth_param basic credentialsttl 2 hours`
6. `auth_param basic casesensitive off`
7. `acl ncsa_users proxy_auth REQUIRED`
8. `http_access allow ncsa_users`

Gambar 4.5 Squid.conf

Gambar 4.5 merupakan perintah untuk merubah konfigurasi *squid* menjadi mode *ncsa authentication* pada komputer *server* tersebut.

- c. *Restart Squid*

```
root@indra-laptop:~# /etc/init.d/squid restart
```

Pemasangan Hierarchy Token Bucket

Penggunaan *HTB* atau *Hierarchy Token Bucket* bertujuan untuk mengatur pembagian *bandwidth*, pembagian dilakukan secara hirarki yang dibagi-bagi kedalam kelas sehingga mempermudah pengaturan *bandwidth*. Adapun langkah-langkah untuk install *Hierarchy Token Bucket*, sebagai berikut :

- a. Instalasi *Hierarchy Token Bucket* :

di karenakan *package htb* ini tidak tersedia di repositori maka harus *download* secara manual. *Copy* kan file *htb* tersebut ke dalam `/usr/local/src`.

1. Setelah itu ekstrak file tersebut dan akan muncul *etc, install, folder, sbin*.
 - `root@indra-laptop:~# tar -zxvf HTB-tools-0.3.0a-i486-1.tgz`
2. pindahkan isi semua folder *sbin* nya *htb tools* di *sbin server*
 - `root@indra-laptop:~# cd /usr/local/src/sbin`
 - `root@indra-laptop:~# mv htb /sbin`
 - `root@indra-laptop:~# mv htbgen /sbin`
 - `root@indra-laptop:~# mv q_checkcfg/sbin`
 - `root@indra-laptop:~# mv q_parser /sbin`
 - `root@indra-laptop:~# mv q_show /sbin`
3. setelah itu masuk ke folder *etc* nya *htb tools*, pindahkan *folder htb* ke */etc* nya *server* dengan cara sebagai berikut:
 - `root@indra-laptop:~# cd /usr/local/src/etc`
 - `root@indra-laptop:~# mv htb /etc`
4. setelah itu hilangkan tulisan *new* yang ada di dalam *folder htb* yang susah kita pindahkan tadi dengan cara sebagai berikut:

- `root@indra-laptop:~# mv /etc/htb/eth0-qos.cfg.new /etc/htb/eth0-qos.cfg`
 - `root@indra-laptop:~# mv /etc/htb/eth1-qos.cfg.new /etc/htb/eth1-qos.cfg`
5. setelah itu masuk ke folder `rc.d` di dalam `etc` `htb tools` dengan cara sebagai berikut:
- `root@indra-laptop:~# cd /usr/local/src/etc/rc.d`
6. masukan file `rc.htb new` ke dalam folder `init.d` di server kita dan di ganti jadi `rc.htb` dengan cara sebagai berikut:
- `root@indra-laptop:~# mv rc.htb.new /etc/init.d/rc.htb`
7. `chmod` file `rc.htb` dengan cara sebagai berikut:
- `root@indra-laptop:~# chmod 755 /etc/init.d/rc.htb`
- b. Ubah dan menambahkan perintah di bawah ini pada file `/etc/htb/eth1-qos.cfg` yang terletak di `Terminal Console` :

```
1. class wijaya_kusuma {
2.   bandwidth 640;
3.   limit 1024;
4.   burst 2;
5.   priority 1;
6.   client dosen {
7.     bandwidth 256;
8.     limit 640;
9.     burst 2;
10.    priority 1;
11.    src {
12.      172.19.11.4/24; 172.19.11.5/24;
13.    };
14.  };
15.  client mahasiswa {
16.    bandwidth 128;
17.    limit 256;
18.    burst 2;
19.    priority 1;
20.    src {
21.      172.19.11.3/24;172.19.11.6/24;
22.      172.19.11.7/24;
23.    };
24.  };
25.  };
26.  class default { bandwidth 8; };
```

Gambar 4.6 `eth0-qos.cfg`

Perintah pada gambar 4.6 adalah untuk pengaturan `bandwidth` yang di bagi menjadi kelas wijaya kusuma dan di dalam kelas tersebut ada kategori dosen dan kategori mahasiswa.

- c. Jika sudah di konfigurasi maka tinggal menjalankan `htb` toolnya dengan cara sebagai berikut:

```
root@indra-laptop:~# etc/init.d/rc.htb start_eth0
```

- d. jika ingin melihat `traffic` nya bisa di lihat dengan cara sebagai berikut:

```
root@indra-laptop:~# /etc/init.d/rc.htb show_eth0
```

Penggabungan Squid Dan Hierarchy token bucket

di dalam server ini perlu adanya penggabungan antara `squid` dan `htb tools`. Adapun proses `penggabungan` dan langkah-langkah konfigurasinya pada komputer server adalah sebagai berikut:

1. Ubah dan tambahkan perintah di bawah ini pada file `/etc/squid/squid.conf` yang terletak di `Terminal Console` :

```
1. zph_mode tos
2. zph_local 0x30
3. zph_parent 0
4. zph_option 136
```

Gambar 4.7 Konfigurasi `squid.conf`

Perintah pada gambar 4.7 adalah untuk menggabungkan `squid zph` dengan `htb tools`.

2. Setelah itu `restart squid` dengan cara sebagai berikut:

- `root@indra-laptop:~# squid -k reconfigure`

3. konfigurasi `htb-tools`

- `root@indra-laptop:~# q_parser eth0 10000 10000 /etc/htb/eth0-qos.cfg > /home/admin/eth0-qos.sh`

4. Ubah dan tambahkan perintah di bawah ini pada file `/home/admin/eth0-qos.sh` yang terletak di `Terminal Console` :

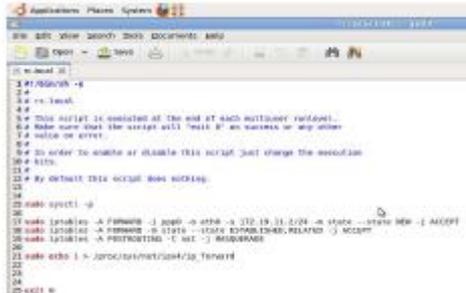
```
1. $TC qdisc add dev $DEV parent
2. 1:0x10 handle 0x10: pfifo limit 5
3. $TC class add dev $DEV parent
4. 1:1 classid 1:0x15 htb rate 10Mbit
5. $TC filter add dev $DEV parent
6. 1:0 protocol ip prio 1 u32 match
7. ip protocol 0x6 0xff match ip tos
   0x30 0xff flowid 1:15
```

Gambar 4.8 `eth0-qos.sh`

Perintah pada gambar 4.8 adalah untuk mengkonfigurasi `htb-tools`.

Konfigurasi *rc.local*

Didalam *rc.local* perlu adanya konfigurasi agar internet yang ada bisa di teruskan ke user atau pengguna. Konfigurasi tersebut bisa di lihat pada gambar 4.9 berikut ini:



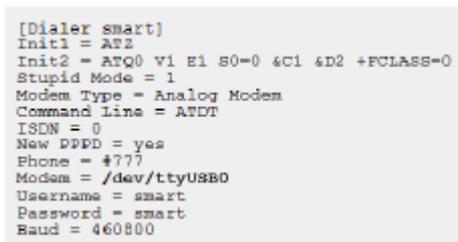
Gambar 4.9 Konfigurasi *rc.local*

Pada gambar 4.9 dapat di jelaskan bahwa *script* tersebut menjalankan perintah internet yang di hasilkan oleh modem akan di teruskan ke koneksi LAN yang mempunyai ip 172.19.11.2 .

Koneksi Modem

Pada komputer *server* perlu internet jadi konfigurasi koneksi modem sebagai berikut:

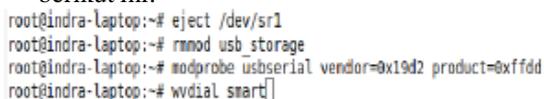
1. Install package *wvdial*
`apt-get install wvdial`
2. konfigurasi *wvdial* dengan mengetikan perintah sebagai berikut:
`gedit /etc/wvdial.conf`
3. kemudian ganti isi dari *default* konfigurasinya dengan konfigurasi *dialer smart* pada gambar 4.10 berikut ini:



Gambar 4.10 konfigurasi *Dialer Smart*.

Pada gambar 4.10 adalah konfigurasi *dialer smart* yang nantinya akan di gunakan untuk mengkoneksikan *modem smart*.

4. Lakukan perintah untuk mengkoneksikan modem *smart* bisa dilihat pada gambar 4.11 berikut ini:



Gambar 4.11 perintah untuk mengkoneksikan modem *smart*.

Pada gambar 4.11 adalah perintah untuk mengkoneksikan modem *smart*. Langkah pertama adalah `eject /dev/sr1` dan `rmmod usb_storage` dan `modprobe usbserial vendor=0x19d2 product=0xffdd` agar modem bisa terbaca sebagai fungsi modem. Langkah kedua adalah `wvdial smart` untuk mengkoneksikan modem ke internet.

Konfigurasi Pada Komputer Klien

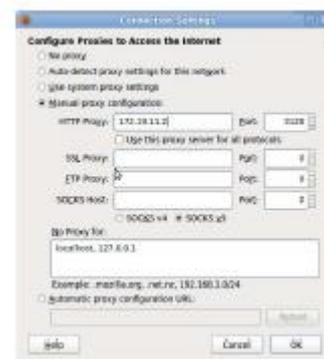
Pada komputer klien harus dilakukan konfigurasi agar komputer klien bisa terkoneksi kedalam *proxy server*. adapun langkah-langkahnya sebagai berikut:

1. Buka *browser*, disini diambil contoh *firefox* bisa dilihat pada gambar 4.12 berikut ini:



Gambar 4.12 *Firefox Preference*

Pada gambar 4.12 adalah untuk menkonfigurasi *proxy* klien. Pilih *tab advanced* setelah itu pilih lagi *tab network* dan terakhir pilih *tab connection setting*. Setelah itu akan muncul *popup menu* kembali seperti gambar 4.13 berikut ini:



Gambar 4.13 *Connection Setting*

Pada gambar 4.13 adalah tampilan konfigurasi *proxy* untuk mengakses internet. Konfigurasi default adalah *auto detect proxy settings for this network* untuk *firefox*. Tapi disini ganti dengan *manual proxy configuration*. Dan isi *http proxy* nya 172.19.11.2 ini adalah ip dari komputer *server*. lalu isi *port* yang akan digunakan yaitu 3128. Setelah itu klik *ok*.

Desain Antarmuka

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

Form tambah user wireless



Gambar 4.14 form tambah user wireless.

Pada gambar 4.14. merupakan form tambah user dimana di tengah ada form inputan untuk memasukan username, password dan macaddress. Dimana form ini adalah form untuk menambah user wireless dimana nanti yang akan dipakai klien untuk login proxy server.

Form view data user



Username	Password	Mac	Aktif
68120037	aad	70:5a:9e:99:45:12	1

Gambar 4.15 Form View Datauser

Pada gambar 4.15 merupakan form untuk melihat siapa saja user yang sudah tersimpan di dalam database. Di form view datauser ada tabel Username, password, dan macaddress disitu nanti akan muncul user siapa saja yang sudah tersimpan. Di form view data user juga ada tab aksi dimana disitu fungsinya juga bisa menghapus user yang ada.

Form menjalankan squid



Gambar 4.16 Form menjalankan squid.

Pada gambar 4.16 merupakan form untuk menjalankan squid. Disini lebih memudahkan pengguna sistem ini karena mereka tidak perlu menjalankan squid proxy server menggunakan perintah yang di jalankan melalui terminal linux. Disitu juga ada button untuk menghentikan squid.

Form menjalankan HTB



Gambar 4.17 Form menjalankan HTB.

Pada gambar 4.17 merupakan form untuk menjalankan HTB. Disini pengguna sistem bisa menjalankan HTB dengan mengklik button start htb dan menghentikan htb dengan mengklik stop htb.

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 login user pada proxy server

Pada pengujian ini adalah Pengujian login user pada proxy server. dimana user memasukan username dan password.

a. Tujuan pengujian

Tujuan dari scenario pengujian 1 ini dibuat adalah memeriksa apakah user yang telah terautentikasi dengan benar sehingga dapat melakukan akses internet keluar.

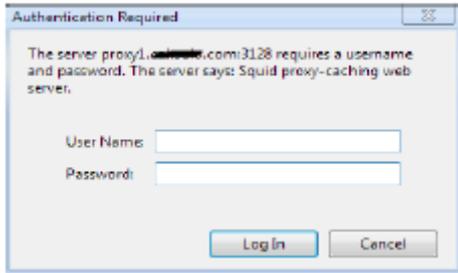
b. Tahapan pengujian

Komputer klien mengkoneksikan ke wireless wijaya kusuma. Setelah itu computer klien mengkonfigurasi pada browser ke konfigurasi proxy agar bisa mendapatkan akses internet.

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 penilain tersebut :

1. Pengujian berhasil

Pengujian dikatakan berhasil jika klien ketika ingin mengakses suatu website muncul popup login proxy yang akan terdapat form inputan username dan password. Bisa dilihat pada gambar 5.1 berikut ini:



Gambar 5.1 login proxy server.

Pada gambar 5.1 merupakan tampilan *form login proxy*. Dimana klien harus memasukan *username* dan *password* untuk bisa mendapatkan akses internet. Jika *username* dan *password* yang di masukan benar maka *browser* akan otomatis *me-redirect* ke halaman *website* yang akan di akses oleh klien.

2. Pengujian gagal

Pengujian gagal apabila klien memasukan *username* dan *password* yang salah. Akan muncul *form login* kembali, jika di *cancel* atau di *close* akan muncul *error cache access denied*. Bisa dilihat pada gambar berikut ini:



Gambar 5.2 error cache acces denied.

Pada gambar 5.2 diatas merupakan tampilan jika *login* gagal dan pengguna mengklik *close* atau *cancel form login*. Akan muncul tampilan *error cache access denied*.

Pengujian manajemen user

Pada pengujian kali ini *computer server* melakukan uji coba manajemen *user*. Ketika jumlah klien yang sudah di tentukan penuh dan ada dosen yang ingin masuk menggunakan fasilitas ini.

a. Tujuan pengujian

Pengujian ini dilakukan untuk mengetahui apakah bisa jika ada klien yang ingin masuk jika jumlah klien yang di tentukan itu sudah penuh.

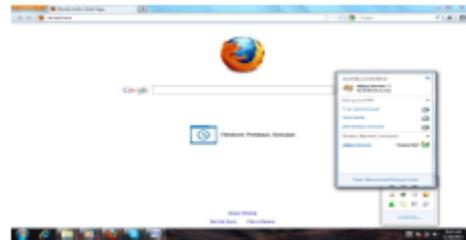
b. Tahapan pengujian

Jika jumlah klien yang sudah di tentukan, misal *ip range wifi router* yang sudah di konfigurasi *dhcp* 172.19.11.3 sampai 172.19.11.7.

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 penilain tersebut :

1. Pengujian berhasil

Pengujian di katakan berhasil jika pengguna dosen atau mahasiswa yang ingin masuk menggunakan fasilitas ini bisa terkoneksi dengan *wifi wijaya kusuma*. Bisa dilihat pada gambar berikut ini:

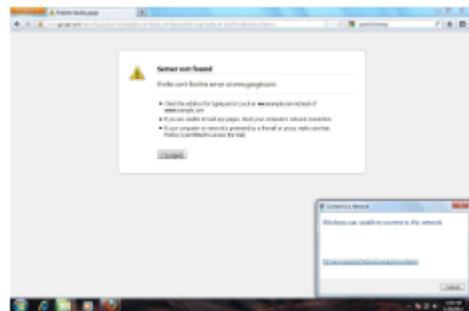


Gambar 5.3 terkoneksi dengan hotspot wijaya kusuma.

Pada gambar 5.3 diatas merupakan tampilan jika *user* atau pengguna belum penuh dan bisa terkoneksi dengan jaringan *hotspot wijaya kusuma*.

2. Pengujian gagal

Pengujian dikatakan gagal apabila pengguna dosen yang ingin masuk menggunakan fasilitas ini tidak bisa terkoneksi dan mendapatkan *ip*. Bisa dilihat pada gambar berikut ini:



Gambar 5.4 unable to connect to this network.

Pada gambar 5.4 diatas merupakan tampilan jika ada *user* atau pengguna yang ingin terkoneksi dengan *hotspot wijaya kusuma* dan di *hotspot* tersebut *user* atau pengguna sudah penuh 5 pengguna, maka *user* tersebut tidak akan bisa terkoneksi dengan jaringan *hotspot* tersebut.

Pengujian download klien

Pada pengujian kali ini komputer klien akan melakukan uji coba *download* menggunakan *IDM (Internet Download Manager)*.

a. Tujuan pengujian

Pengujian kali ini bertujuan agar bisa menunjukkan seberapa besar komputer klien jika *download* menggunakan *IDM (Internet Download Manager)*.

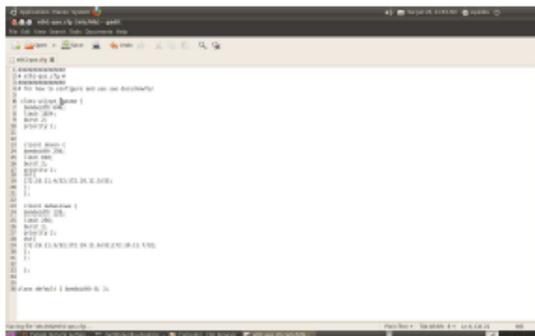
b. Tahapan pengujian

Pada pengujian kali ini akan di ambil *sample* atau contoh 5 kali pengujian. Bisa di lihat pada tabel di bawah ini:

No	Skema	Batas <i>maximum download</i>	<i>Bandwidth</i> Tanpa dibatasi <i>HTB</i>	Berhasil / gagal
1	2 dosen 1 mahasiswa	Dosen = 80KB/sec Mahasiswa = 30KB/sec	544KB/sec	berhasil
2	2 dosen 2 mahasiswa	Dosen = 80KB/sec Mahasiswa = 30KB/sec	544KB/sec	berhasil
3	2 dosen 3 mahasiswa	Dosen = 80KB/sec Mahasiswa = 30KB/sec	544KB/sec	berhasil
4	1 dosen 3 mahasiswa	Dosen = 80KB/sec Mahasiswa = 30KB/sec	544KB/sec	berhasil
5	4 dosen 1 mahasiswa	Dosen = 80KB/sec Mahasiswa = 30KB/sec	544KB/sec	berhasil

Tabel 5.1 ruang lingkup uji coba.

Pada tabel 5.1 diatas merupakan ruang lingkup uji coba dimana ada skema pengujian, batas *maximum download*, *bandwidth* tanpa dibatasi oleh *HTB* dan ada berhasil atau gagalnya pengujian tersebut. Skema pengujian diatas telah ditentukan parameter pengujiannya dengan parameter 2 dosen dan 3 mahasiswa. Dan bisa di lihat pada gambar berikut ini:



Gambar 5.5 konfigurasi *htb*

Pada gambar 5.5 adalah konfigurasi umum *htb* untuk di ambil contoh pengujian. Disitu bisa dilihat ada class wijaya kusuma yang mempunyai *bandwidth minimum* 640KB dan mempunyai *bandwidth maximum* 1024KB. Dimana didalam class tersebut ada 2 klien yaitu klien dosen dan klien mahasiswa. Klien dosen mempunyai *bandwidth minimum* 256KB dan mempunyai *bandwidth maximum* 640KB. Di klien mahasiswa mempunyai *bandwidth minimum* 128KB dan mempunyai *bandwidth maximum* 256KB. Dan di



dalam klien klien tersebut sudah di tentukan *ip range* yang nantinya akan di pakai untuk dosen dan mahasiswa. Dan pengujiannya sebagai berikut:

1. Pengujian pertama di ambil contoh 2 dosen dan 1 mahasiswa

Pada kali ini yang *login* bersamaan yaitu 2 dosen dan 1 mahasiswa dimana 2 dosen tersebut menggunakan *ip address* 172.19.11.4 dan 172.19.11.5 dan 1 mahasiswa tersebut 172.19.11.3. ini adalah gambar dosen pertama ketika melakukan *download* menggunakan *IDM* bisa dilihat pada gambar berikut ini:



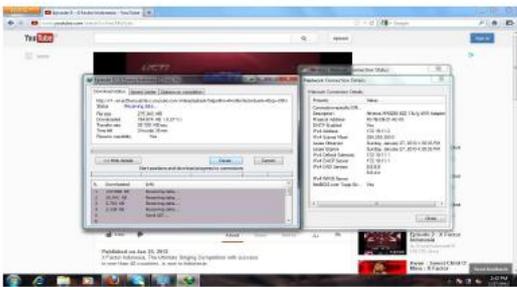
Gambar 5.6 *download idm* dosen 1

Pada gambar 5.6 diatas merupakan proses *download* dosen 1 dengan menggunakan *ip address* 172.19.11.4. dia mendapatkan batas *maximum download* dengan kecepatan 79KB/sec. sama halnya dengan dosen 2 bisa dilihat pada gambar berikut:



Gambar 5.7 download idm dosen 2

Pada gambar 5.7 diatas merupakan proses download dosen 2 dengan menggunakan ip address 172.19.11.5. dia juga mendapatkan batas maximum download dengan kecepatan 80KB/sec. berbeda dengan mahasiswa 1 bisa dilihat pada gambar berikut ini;



Gambar 5.8 download idm mahasiswa 1

Pada gambar 5.8 diatas merupakan proses download mahasiswa 1 dengan menggunakan ip address 172.19.11.3. yang mendapatkan batas maximum download dengan kecepatan 30KB/sec.

2. Pengujian kedua di ambil contoh 2 dosen dan 2 mahasiswa

Pada kali ini yang login bersamaan yaitu 2 dosen dan 2 mahasiswa dimana 2 dosen tersebut menggunakan ip address 172.19.11.4 dan 172.19.11.5 dan 2 mahasiswa tersebut 172.19.11.3 dan 172.19.11.6. ini adalah gambar dosen 1 ketika melakukan download menggunakan IDM bisa dilihat pada gambar berikut ini:



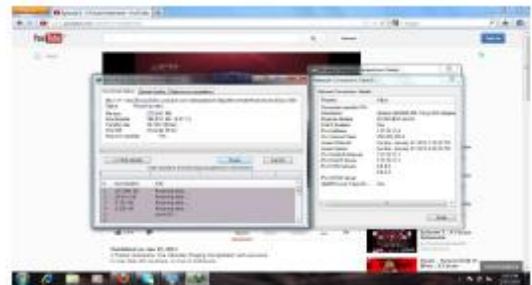
Gambar 5.9 download idm dosen 1

Pada gambar 5.9 diatas merupakan proses download dosen 1 dengan menggunakan ip address 172.19.11.4. dia mendapatkan batas maximum download dengan kecepatan 79KB/sec. sama halnya dengan dosen 2 bisa dilihat pada gambar berikut:



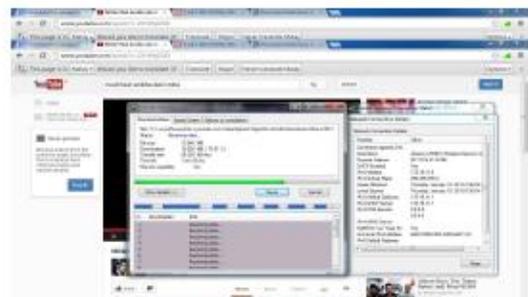
Gambar 5.10 download idm dosen 2

Pada gambar 5.10 diatas merupakan proses download dosen 2 dengan menggunakan ip address 172.19.11.5. dia juga mendapatkan batas maximum download dengan kecepatan 80KB/sec. berbeda dengan mahasiswa 1 bisa dilihat pada gambar berikut ini;



Gambar 5.11 download idm mahasiswa 1

Pada gambar 5.11 diatas merupakan proses download mahasiswa 1 dengan menggunakan ip address 172.19.11.3. yang mendapatkan batas maximum download dengan kecepatan 30KB/sec. dan mahasiswa 2 bisa di lihat pada gambar berikut ini:



Gambar 5.12 download idm mahasiswa 2

Pada gambar 5.12 diatas merupakan proses download idm mahasiswa 2 yang menggunakan

ip address 172.19.11.6. dan mendapatkan batas *maximum download* dengan kecepatan 28KB/sec.

3. Pengujian ketiga di ambil contoh 2 dosen 3 mahasiswa

Pada kali ini yang *login* bersamaan yaitu 2 dosen dan 3 mahasiswa dimana 2 dosen tersebut menggunakan *ip address* 172.19.11.4 dan 172.19.11.5 dan 2 mahasiswa tersebut 172.19.11.3 dan 172.19.11.6 dan 172.19.11.7. ini adalah gambar dosen 1 ketika melakukan *download* menggunakan *IDM* bisa dilihat pada gambar berikut ini:



Gambar 5.13 *download idm* dosen 1

Pada gambar 5.13 diatas merupakan proses *download* dosen 1 dengan menggunakan *ip address* 172.19.11.4. dia mendapatkan batas *maximum download* dengan kecepatan 79KB/sec. sama halnya dengan dosen 2 bisa dilihat pada gambar berikut:



Gambar 5.14 *download idm* dosen 2

Pada gambar 5.14 diatas merupakan proses *download* dosen 2 dengan menggunakan *ip address* 172.19.11.5. dia juga mendapatkan batas *maximum download* dengan kecepatan 80KB/sec. berbeda dengan mahasiswa 1 bisa dilihat pada gambar berikut ini;



Gambar 5.15 *download idm* mahasiswa 1

Pada gambar 5.15 diatas merupakan proses *download* mahasiswa 1 dengan menggunakan *ip address* 172.19.11.3. yang mendapatkan batas *maximum download* dengan kecepatan 30KB/sec. dan mahasiswa 2 bisa di lihat pada gambar berikut ini:



Gambar 5.16 *download idm* mahasiswa 2

Pada gambar 5.16 diatas merupakan proses *download idm* mahasiswa 2 yang menggunakan *ip address* 172.19.11.6. dan mendapatkan batas *maximum download* dengan kecepatan 28KB/sec. dan mahasiswa 3 bisa dilihat pada gambar sebagai berikut:

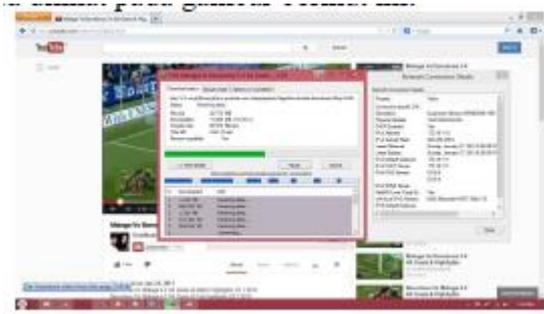


Gambar 5.17 *download idm* mahasiwa 3

Pada gambar 5.17 diatas merupakan proses *download idm* mahasiswa 3 yang menggunakan *ip address* 172.19.11.7. dan mendapatkan batas *maximum download* dengan kecepatan 30KB/sec.

4. Pengujian keempat di ambil contoh 1 dosen 3 mahasiswa

Pada kali ini yang *login* bersamaan yaitu 1 dosen dan 3 mahasiswa dimana 1 dosen tersebut menggunakan *ip address* 172.19.11.5 dan 2 mahasiswa tersebut 172.19.11.3 dan 172.19.11.7. ini adalah gambar dosen 1 ketika melakukan *download* menggunakan *IDM* bisa dilihat pada gambar berikut ini:



Gambar 5.18 *download idm* dosen 1

Pada gambar 5.18 diatas merupakan proses *download* dosen 1 dengan menggunakan *ip address* 172.19.11.5. dia mendapatkan batas *maximum download* dengan kecepatan 80KB/sec. berbeda dengan mahasiswa 1 bisa dilihat pada gambar berikut ini;



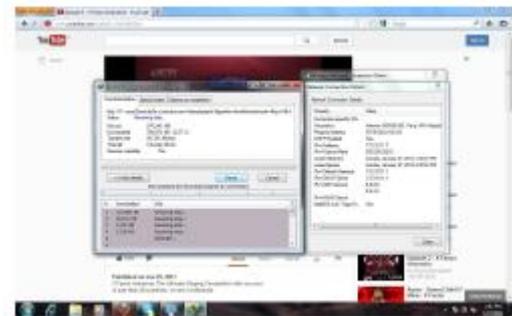
Gambar 5.19 *download idm* mahasiswa 1

Pada gambar 5.19 diatas merupakan proses *download* mahasiswa 1 dengan menggunakan *ip address* 172.19.11.3. yang mendapatkan batas *maximum download* dengan kecepatan 30KB/sec. dan mahasiswa 2 bisa di lihat pada gambar berikut ini:



Gambar 5.20 *download idm* mahasiswa 2

Pada gambar 5.20 diatas merupakan proses *download idm* mahasiswa 2 yang menggunakan *ip address* 172.19.11.6. dan mendapatkan batas *maximum download* dengan kecepatan 28KB/sec. dan mahasiswa 3 bisa dilihat pada gambar sebagai berikut:



Gambar 5.21 *download idm* mahasiswa 3

Pada gambar 5.21 diatas merupakan proses *download idm* mahasiswa 3 yang menggunakan *ip address* 172.19.11.7. dan mendapatkan batas *maximum download* dengan kecepatan 30KB/sec.

5. Pengujian kelima di ambil contoh 4 dosen 1 mahasiswa

Pada kali ini yang *login* bersamaan yaitu 4 dosen dan 1 mahasiswa dimana 4 dosen tersebut menggunakan *ip address* 172.19.11.4 dan 172.19.11.5 dan 172.19.11.3 dan 172.19.11.6. dan 1 mahasiswa tersebut menggunakan *ip address* 172.19.11.7. ini adalah gambar dosen 1 ketika melakukan *download* menggunakan *IDM* bisa dilihat pada gambar berikut ini :



Gambar 5.22 download idm dosen 1

Pada gambar 5.22 diatas merupakan proses download dosen 1 dengan menggunakan ip address 172.19.11.4. dia mendapatkan batas maximum download dengan kecepatan 79KB/sec. sama halnya dengan dosen 2 bisa dilihat pada gambar berikut:



Gambar 5.23 download idm dosen 2

Pada gambar 5.23 diatas merupakan proses download dosen 2 dengan menggunakan ip address 172.19.11.5. dia juga mendapatkan batas maximum download dengan kecepatan 80KB/sec. sama halnya dengan dosen 3 bisa dilihat pada gambar berikut ini;



Gambar 5.24 download idm dosen 3

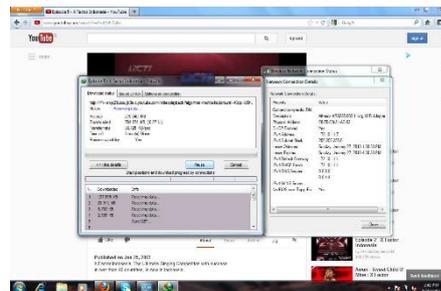
Pada gambar 5.24 diatas merupakan proses download dosen 3 dengan menggunakan ip address 172.19.11.3. yang mendapatkan batas maximum download dengan kecepatan 79KB/sec.

dan sama juga dengan dosen 4 bisa di lihat pada gambar berikut ini:



Gambar 5.25 download idm dosen 4

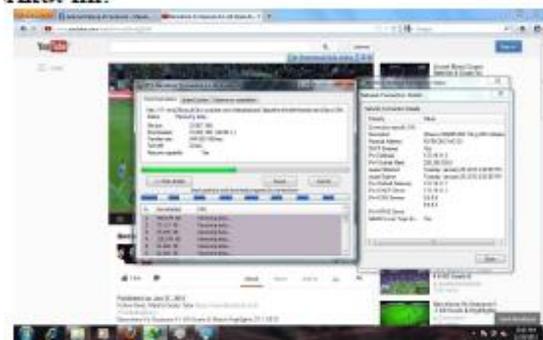
Pada gambar 5.25 diatas merupakan proses download idm dosen 4 yang menggunakan ip address 172.19.11.6. dan mendapatkan batas maximum download dengan kecepatan 79KB/sec. dan berbeda dengan mahasiswa 1 bisa dilihat pada gambar sebagai berikut:



Gambar 5.26 download idm mahasiswa 1

Pada gambar 5.26 diatas merupakan proses download idm mahasiswa 1 yang menggunakan ip address 172.19.11.7. dan mendapatkan batas maximum download dengan kecepatan 30KB/sec.

Dari pengujian diatas, dilakukan juga pengujian pengguna melakukan download menggunakan IDM dan pengguna tersebut tidak dibatasi oleh HTB. Bisa dilihat pada gambar berikut ini:



Gambar 5.27 download pengguna tanpa htb.

Pada gambar 5.27 diatas merupakan hasil *download* pengguna yang tidak dibatasi *bandwidth*nya oleh *HTB*. Sehingga pengguna bisa mendapatkan kecepatan *download* sebesar 544KB/sec.

PENUTUP

Secara umum Perancangan sistem manajemen *bandwidth* pada *proxy server* menggunakan *hierarchy token bucket* ini dapat berfungsi sebagaimana yang diharapkan. Namun tentu saja masih diperlukan beberapa penyempurnaan yang akan dipaparkan dalam pembahasan bab ini.

Kesimpulan

Berdasarkan uraian pembahasan perancangan dan pengujian yang telah dilakukan, maka dapat diambil kesimpulan terhadap sistem yaitu rancangan sistem manajemen *bandwidth* pada *proxy server* menggunakan *hierarchy token bucket* dapat digunakan sebagai alternatif yang baik untuk memamanajemen *bandwidth* di kampus.

Saran

Saran untuk perkembangan lebih lanjut dalam masa mendatang pada Perancangan sistem manajemen *bandwidth* pada *proxy server* menggunakan *hierarchy token bucket* ini yaitu dalam pengujiannya bisa lebih banyak pengguna lagi yang di ujikan. Dan agar username yang di daftarkan bisa digunakan di laptop pengguna lain.

DAFTAR PUSTAKA

- [1] yHuS_dCc_X, 2011. *Definisi Jaringan Komputer*. <http://sengkang-flash.blogspot.com/2011/01/defenisi-jaringan-komputer.html> diakses 3 juli 2012.
- [2] adi4adi. 2010. *HTB (Hierarchical Token Bucket)*.<http://www.scribd.com/doc/24030951/HTB> diakses 2 juli 2012.
- [3] acieeee, 2010. *Pengertian Proxy Server*. <http://acieeee.wordpress.com/2010/04/01/pengertian-proxy-server/> diakses 1 juli 2012
- [4] Croll, Alistair, 2000, *Managing Bandwidth Deploying QoS in Enterprise Networks*, Prentice Hall, New Jersey.
- [5] Devera, Martin, *HTB Manual User Guide* <http://luxik.cdi.cz/~devik/qos/>. Diakses 4 juli 2012.
- [6] *Howto Squid proxy authentication using ncsa_auth helper*, <http://www.cyberciti.biz/tips/linux-unix-squid-proxy-server-authentication.html>. Diakses 5 september 2012.

{Halaman sengaja dikosongkan}

{Jurnal **Melek IT**}

(2)

SISTEM OPTIMALISASI *CACHE PROXY* MENGGUNAKAN METODE *REPLACEMENT* PADA *SQUID PROXY SERVER*

Erwin Damei Hadi Putra¹, Tjatorsari Widiartin²

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Wijaya Kusuma Surabaya
Jl. Dukuh Kupang XXV/54, erwin.mellowdik@gmail.com

ABSTRAK

Koneksi internet yang lambat sering terjadi pada organisasi atau perusahaan yang memiliki *proxy server*. Tesis ini meneliti tentang optimalisasi performa dari *proxy server*. *Proxy server* perlu memiliki metode yang baik untuk melakukan pergantian objek pada. *Proxy server* menciptakan *log* dari setiap transaksi data yang terjadi. Parameter utama dari penelitian ini dan hasil dari penelitian ini berupa usulan metode dan topologi *proxy server* untuk optimalisasi *proxy*.

Dalam sebuah jaringan lokal yang ada dalam suatu perusahaan atau organisasi, terdapat satu atau lebih komputer yang terhubung di dalamnya. Lebih jauh, jika perusahaan atau organisasi tersebut ingin terhubung dengan dunia luar melalui internet, maka dibutuhkan satu atau lebih untuk menghubungkan komputer yang ada pada jaringan lokal agar dapat mengakses internet secara bersamaan. Fungsi tersebut dapat dimungkinkan dengan adanya sebuah *proxy server* sebagai jembatan penghubung.

Dalam prakteknya, akan ada beberapa *user* yang mengakses alamat yang sama, maka untuk menghemat yang digunakan, *proxy server* dapat menyimpan konten dari yang sebelumnya telah dikunjungi, untuk dapat digunakan kembali apabila di oleh *user* yang lain. Beberapa *proxy server* juga dapat melakukan fitur-fitur administratif seperti pembatasan hak akses terhadap user mengenai alamat yang boleh diakses dan tidak boleh diakses, pembatasan, autentikasi dan sebagainya. *Proxy server* dalam hal ini dapat berfungsi sebagai dari sebuah jaringan lokal menuju internet.

Kata kunci : *Cache Proxy*, Metode *Replacement*, *Squid Proxy Server*.

PENDAHULUAN

Latar Belakang

Komunikasi di dalam *internet* antara suatu *web browser (client)* dengan suatu *web server (server)* umumnya dilakukan secara langsung dengan menggunakan paradigma hubungan *client/server*. Seiring dengan perkembangan teknologi *internet*, dan meningkatnya reiko keamanan suatu komputer/jaringan komputer yang terhubung langsung dengan *internet*, maka dikembangkanlah suatu teknologi dalam jaringan komputer yang mampu bertindak sebagai perantara antara suatu komputer/jaringan komputer dengan *internet*, yang dikenal sebagai *proxy*.

Proxy adalah suatu program yang diletakkan di antara suatu komputer/jaringan komputer dengan *internet*. *Proxy* bertindak sebagai perantara (*middleman*) bagi kedua pihak. *Proxy* bertindak sebagai *server* bagi pengakses *internet (web browser)* dan *proxy* bertindak sebagai *client* bagi *web server*. Dalam penggunaan *internet* selalu membutuhkan *bandwidth* yang cukup besar, karena *request* dari semua *client* dapat membuat lalu lintas transfer data menjadi penuh, sehingga

dapat membuat transfer data ke *client* menjadi lama.

Maka dari itu dibutuhkan pengaturan *bandwidth* agar *request* dari *client* tidak membuat lalu lintas transfer data menjadi penuh disaat *client* melakukan *request* secara bersamaan. Untuk mengatasi masalah yang dihadapi dan supaya penggunaan *bandwidth* tidak terlalu besar, maka dibutuhkannya pengaturan *bandwidth* yang baik. Untuk itu dalam penelitian ini, dilakukan pembuatan *server proxy* yang berfungsi sebagai penyimpan *content* dari *server* yang pernah *request* oleh *client*. Sehingga pada saat *client* melakukan *request*, *bandwidth* yang digunakan tidak terlalu besar. Sistem ini dapat menghemat akan penggunaan *bandwidth* dan juga membuat *request* data menjadi lebih cepat. Dalam penelitian ini menggunakan metode *Replacement* yang menempatkan penyimpanan *cache* di luar *proxy* atau di *server* lain. Dengan menggunakan metode ini, dapat membuat *server proxy* menjadi lebih baik kinerjanya, karena apabila ruang hardisk di dalam *proxy server* tersebut penuh, maka dapat mengakibatkan kinerja *proxy server* menjadi lambat.

Tujuan

Berdasarkan permasalahan yang ada, maka tujuan yang diharapkan dalam membangun *cache proxy server* adalah sebagai berikut :

1. Mengendalikan kebutuhan *bandwidth* yang sangat besar.
2. Memberikan kecepatan akses kepada server dan client dengan adanya proses *caching*.
3. Mengevaluasi hasil kinerja dari *cache server*

TINJAUAN PUSTAKA

Jaringan Komputer

Menurut Tanenbaum (2003,p10), jaringan komputer adalah kumpulan dua atau lebih komputer yang masing-masing berdiri sendiri dan terhubung melalui sebuah teknologi.

A. Topologi Jaringan Komputer

Topologi dapat diartikan sebagai struktur dari sebuah jaringan. Topologi jaringan dapat dibagi menjadi dua yaitu :

1. *Physical Topology*, ada *layout* fisik, penempatan *device* dan media yang digunakan. Pada topologi ini terdapat 7 topologi yaitu *bus*, *ring*, *star*, *extended star*, *hierarchical*, *mesh*, dan *hybrid* . *Bus*, *Ring*, *Star* merupakan yang paling sering digunakan.
2. *Logical Topology*, dimana *host* mengakses satu media. *Logical Topology* adalah gambaran secara maya bagaimana *host* dapat berkomunikasi melalui medium. Bentuk umum yang biasa digunakan adalah *Broadcast* dan *Token Passing*.

B. Macam-macam Jaringan

Saat ini ada beberapa macam jaringan. Pembagian tersebut didasarkan pada luas wilayah yang ditangani oleh jaringan tersebut. Berikut ini adalah beberapa macam jaringan berdasarkan luas wilayahnya.

1. LAN (*Local Area Network*)

LAN adalah suatu jaringan yang mencakup daerah yang kecil, sehingga penirisan data berkecepatan tinggi serta memiliki *error rate* yang rendah. LAN biasanya digunakan dalam suatu gedung dengan cara menghubungkan host dengan peralatan lainnya.

2. WAN (*Wide Area Network*)

WAN adalah gabungan dari beberapa LAN yang terhitung dalam jarak jangkauan yang jauh, misalnya antar kota (Myhre, 2000, p2).

3. MAN (*Metropolitan Area Network*)

MAN dirancang untuk menghubungkan beberapa LAN yang berbeda dalam satu kota (Forouzan, 2003, p21).

Bandwidth

Bandwidth dapat diartikan sebagai berikut:

1. Kapasitas maksimum sebuah saluran jaringan. *Bandwidth* biasanya diukur dalam satuan *bit* per detik (*Bps*), kilo-*bit* per detik (*Kbps*), mega-*bit* per detik (*Mbps*), atau giga-*bit* per detik (*Gbps*).
2. Bagian dari spektrum frekuensi yang diperlukan untuk mentransmisi informasi yang dibutuhkan.

Internet

Internet adalah suatu jaringan computer yang satu dengan yang lain saling terhubung untuk keperluan komunikasi dan informasi. Sebuah *computer* dalam satu jaringan *internet* dapat berada di mana saja atau bahkan di seluruh Indonesia. Sering juga *internet* diartikan sebagai jaringan komputer diseluruh diartikan sebagai jaringan komputer di seluruh dunia yang berisikan informasi dan sebagai sarana komunikasi data yang berupa suara, gambar, *video*, dan juga *teks*. Halaman *web* pada *internet* terdiri dari dua jenis yaitu:

1. *Static web page*

Adalah halaman pada *internet* yang disampaikan kepada pengguna persis sesuai dengan apa yang disimpan dan bersifat statis. Akibatnya halaman web statis menampilkan informasi yang sama kepada semua pengguna dari semua *konteksnya*. Untuk mengganti konten dalam *web* statis sangat sulit sebab disimpan pada HTML bukan dalam *databases*.

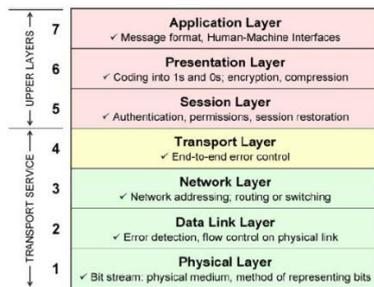
2. *Dynamic web page*

Adalah jenis halaman *web* yang telah dipersiapkan dengan informasi yang lebih dinamis. Hal ini tidak statis karena berubah dengan waktu (misalnya pada konten berita), pengguna (misal pada *login session*), interaksi pengguna (misalnya pada *web base game*), atau gabungan kombinasi dari semuanya.

OSI Model

Dahulu, komunikasi antara 2 vendor yang berbeda sangat sulit dilakukan karena komputer menggunakan protokol dan format data yang berbeda. Oleh karena itu, *International Organization for Standardization (ISO)* membuat suatu arsitektur komunikasi yang dikenal dengan *Open System Interconnection (OSI)*, yaitu model yang mendefinisikan standar untuk menghubungkan komputer dengan vendor-vendor berbeda.

Arsitektur OSI



Gambar 1. OSI Model

Model OSI dibagi menjadi 2 bagian yaitu :

1. Upper layers

Upper layers terdiri dari *application layer*, *presentation layer* dan *session layer* yang ketiganya merupakan *application-oriented* di mana bertanggung jawab untuk menangani aplikasi jaringan, representasi data, dan interaksi antar aplikasi dengan pengguna.

2. Lower Layer / Transport

Lower layer merupakan *network-oriented* yang akan berurusan dengan transmisi data, *covering* dan pemaketan, *routing*, *verification*, dan transmisi dari masing-masing bagian paket data.

Berikut adalah 7 layer OSI :

1. Physical Layer

Layer ini berfungsi mendefinisikan media transmisi jaringan, metode pensinyalan, sinkronisasi bit, arsitektur jaringan, topologi jaringan serta pengkabelannya. Pada layer ini juga didefinisikan tentang bagaimana NIC (*network interface card*) dapat berinteraksi dengan media kabel atau radio.

2. Data Link Layer

Layer ini memiliki fungsi menentukan bagaimana mengelompokkan bit-bit data menjadi suatu format yang disebut dengan *frame*. Pada layer ini juga terjadi koreksi terhadap error, *flow control*, pengalamatan perangkat keras seperti MAC Address (*Media Access Control Address*), dan menentukan bagaimana perangkat-perangkat jaringan seperti hub, bridge, repeater, dan switch layer beroperasi.

3. Network Layer

Network layer memiliki fungsi untuk mendefinisikan alamat-alamat IP, membuat header untuk paket-paket, dan melakukan *routing* melalui *internetworking* dengan menggunakan *router* dan *switch layer-3*.

4. Transport Layer

Transport layer memiliki fungsi untuk memecah data ke dalam paket-paket tersebut sehingga dapat disusun kembali pada sisi tujuan setelah diterima

dengan sukses (*acknowledgement*), dan melakukan transmisi ulang terhadap paket-paket yang rusak.

5. Session Layer

Session Layer berfungsi menyediakan mekanisme untuk mengontrol *dialog* antara aplikasi. Aplikasi-aplikasi pada lapisan sesi membuat dialog, mengatur dialog, dan mengakhiri dialog ketika sudah selesai, serta membuat kembali dialog antara aplikasi ketika dialog terputus karena alasan-alasan tertentu. *Session layer* juga menyediakan layanan yang memastikan bahwa komputer-komputer pengirim tidak membuat komputer penerima mengalami kelebihan data (*overwhelming*).

6. Presentation Layer

Layer ini berfungsi untuk mentranslasikan data yang hendak ditransmisikan oleh aplikasi ke dalam format yang ditransmisikan melalui jaringan. Protokol yang berada dalam level ini adalah perangkat lunak redirektor (*redirector software*), seperti layanan *workstation* dan juga *network shell* atau *Remote Desktop Protocol (RDP)*.

7. Application Layer

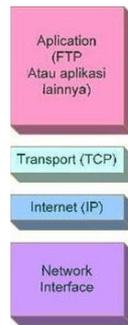
Layer ini berfungsi menyediakan cara bagi program-program aplikasi seperti file dan *print service*, *database service*, dan e-mail untuk mengakses lingkungan OSI. Lapisan ini berisikan mekanisme-mekanisme yang umumnya berguna untuk mendukung aplikasi-aplikasi yang didistribusikan di jaringan.

TCP / IP

Protokol diperlukan untuk mengatur bagaimana sebuah perangkat keras dapat berkomunikasi dengan perangkat keras lainnya. Agar perangkat-perangkat keras seperti *switch*, *router*, atau komputer dapat berhubungan satu sama lain, maka dibutuhkan protokol yang sama.

A. TCP

TCP/IP (*Transmission Control Protokol / Internet Protokol*) adalah standar komunikasi data yang digunakan oleh komunitas *internet* dalam proses tukar-menukar data dari satu komputer ke komputer lain di dalam jaringan *Internet*. Protokol TCP/IP dikembangkan pada akhir dekade 1970-an hingga awal 1980-an sebagai sebuah protokol standar untuk menghubungkan komputer-komputer dan jaringan untuk membentuk sebuah jaringan yang luas (WAN). TCP/IP merupakan sebuah standar jaringan terbuka yang bersifat independen terhadap mekanisme *transport* jaringan fisik yang digunakan, sehingga dapat digunakan di mana saja. Definisi Masing-masing Layer pada model TCP/IP sebagai berikut :



Gambar 2. TCP/IP

1. Network Interface

Berfungsi untuk meletakkan *frame-frame* jaringan di atas media jaringan yang digunakan. TCP/IP dapat bekerja dengan banyak teknologi *transport*, mulai dari teknologi *transport* dalam LAN (seperti halnya *Ethernet* dan *Token Ring*), MAN dan WAN (seperti halnya *dial-up* model yang berjalan di atas *Public Switched Telephone Network (PSTN)*, *Integrated Services Digital Network (ISDN)*, serta *Asynchronous Transfer Mode (ATM)*.

2. Internet

Berfungsi untuk melakukan pemetaan (*routing*) dan enkapsulasi paket-paket data jaringan menjadi paket-paket IP. Protokol yang bekerja dalam lapisan ini adalah *Internet Protocol (IP)*, *Address Resolution Protocol (ARP)*, *Internet Control Message Protocol (ICMP)*, dan *Internet Group Management Protocol (IGMP)*.

3. Transport

Berguna untuk membuat komunikasi menggunakan sesi koneksi yang bersifat *connection-oriented* atau *broadcast* yang bersifat *connectionless*. Protokol dalam lapisan ini adalah *Transmission Control Protocol (TCP)* dan *User Datagram Protocol (UDP)*.

4. Application

Merupakan *Layer* paling atas pada model *TCP/IP*, yang bertanggung jawab untuk menyediakan akses kepada aplikasi terhadap layanan jaringan *TCP/IP*. Protokol ini mencakup *protocol Dynamic Host Configuration Protocol (DHCP)*, *Domain Name System (DNS)*, *Hypertext Transfer Protocol (HTTP)*, *File Transfer Protocol (FTP)*, *Telnet*, *Simple Mail Transfer Protocol (SMTP)*, *Simple Network Management Protocol (SNMP)*, dan masih banyak protokol lainnya. Dalam beberapa implementasi *Stack Protocol*, seperti halnya *Microsoft TCP/IP*, protokol-protokol lapisan aplikasi berinteraksi dengan menggunakan antarmuka *Windows Sockets (Winsock)* atau *NetBios over CP/IP (NetBT)*.

B. UDP

UDP singkatan dari *User Datagram Protocol*, adalah salah satu protokol lapisan transpor *TCP/IP* yang mendukung komunikasi yang tidak andal (*unreliable*), tanpa koneksi (*connectionless*) antara *host-host* dalam jaringan yang menggunakan *TCP/IP*. Protokol ini didefinisikan dalam *RFC 76*. Karakteristik dari UDP adalah sebagai berikut :

1. *Connectionless* (tanpa koneksi) : Pesan-pesan UDP akan dikirimkan tanpa harus dilakukan proses negosiasi koneksi antara dua *host* yang hendak berukar informasi
2. *Unreliable* (tidak andal) : Pesan-pesan UDP akan dikirimkan sebagai datagram tanpa adanya nomor urut atau pesan *acknowledgment*. Protokol lapisan aplikasi yang berjalan di atas UDP harus melakukan pemulihan terhadap pesan-pesan yang hilang selama transmisi. Umumnya, protokol lapisan aplikasi yang berjalan di atas UDP mengimplementasikan layanan keandalan mereka masing-masing, atau mengirim pesan secara periodik atau dengan menggunakan waktu yang telah didefinisikan.
3. UDP menyediakan mekanisme untuk mengirim pesan-pesan ke sebuah protokol lapisan aplikasi atau proses tertentu di dalam sebuah *host* dalam jaringan yang menggunakan *TCP/IP*. *Header* UDP berisi *field Source Process Identification* dan *Destination Process Identification*.
4. UDP menyediakan penghitungan checksum berukuran 16-bit terhadap keseluruhan pesan UDP.

C. IP (Internet Protocol)

Internet Protocol (IP) adalah protokol network layer (layer ke3 dalam *OSI model*) yang mengandung informasi pengalamatan dan beberapa informasi kontrol yang membuat paket dapat diarahkan menuju tujuan dalam sebuah jaringan. *IP* merupakan protokol network layer yang utama dalam deretan protokol *TCP/IP*. (Dong,2007,p.255).

Alamat *IP* adalah representasi dari 32 *bit* bilangan biner, yang ditampilkan dalam bentuk desimal dengan titik. Contoh penulisan *IP* adalah 192.168.1.200.

Alamat *IP* versi 4 umumnya diekspresikan dalam notasi desimal bertitik (*dotted-decimal notation*), yang dibagi ke dalam empat buah *oktet* berukuran 8-bit. Dalam beberapa buku referensi, format bentuknya adalah *w.x.y.z*. Karena setiap *oktet* berukuran 8-bit, maka nilainya berkisar antara 0 hingga 255 (meskipun begitu, terdapat beberapa pengecualian nilai).

NAT

Menurut Mudji Basuki (2003) keterbatasan alamat IPv4 merupakan masalah pada jaringan global atau Internet. Untuk memaksimalkan penggunaan alamat IP yang diberikan oleh *Internet Service Provider (ISP)* dapat digunakan *Network Address Translation* atau *NAT*. *NAT* membuat jaringan yang menggunakan alamat lokal (*private*), alamat yang tidak boleh ada dalam tabel routing Internet dan dikhususkan untuk jaringan lokal/intranet, dapat berkomunikasi ke Internet dengan jalan „meminjam“ alamat IP Internet yang dialokasikan oleh ISP. Jadi *NAT* merupakan suatu metode untuk menghubungkan lebih dari satu komputer ke jaringan internet dengan menggunakan satu alamat IP.

IP Masquerade

IP Masquerade adalah salah satu fasilitas yang menerapkan bentuk *NAT* di *Linux* yang memungkinkan komputer yang tidak memiliki nomor IP resmi dapat tersambung ke internet melewati komputer *Linux*.

Cara kerja *IP Masquerade* dapat digambarkan sebagai berikut :

1. Mesin klien diatur dengan menempatkan mesin *Linux* sebagai *gateway*nya.
2. Jika ada paket dari klien ke mesin *Linux*, *IP Masquerade* akan membuat nomor port baru, header IP baru berdasarkan IP mesin *Linux*, dan menyimpan paket aslinya. Paket tersebut kemudian diteruskan ke internet..
3. Jika ada paket baru dari Internet, *IP Masquerade* akan mencocokkan dengan nomor port. Jika sesuai dengan nomor port yang telah dibuat sebelumnya, maka paket tersebut dibungkus kembali dengan header yang telah disimpan sebelumnya, dan diteruskan kepada klien.
4. Baik klien maupun host di Internet yang mengirimkan paket tidak pernah merasakan perbedaannya

Proxy Server

Menurut Mark Grennan (2000) *proxy server* sebagian besar digunakan untuk mengontrol, atau memantau, lalu lintas keluar pada jaringan internet. Sedangkan Keir Thommas (2006) *Proxy server* adalah sebuah komputer server yang melakukan dua hal yaitu pertama, menyediakan keamanan tambahan dengan menyediakan portal tunggal untuk mengakses semua halaman web dan menyimpan halaman yang sering diakses sehingga mempercepat koneksi akses internet. Jadi *Proxy server* adalah sebuah komputer server atau program komputer yang dapat bertindak

sebagai komputer lainnya untuk melakukan request terhadap konten dan menangani konten tersebut serta masalah lalu lintas jaringan dari Internet atau intranet. Perbedaan *proxy* dengan *gateway* adalah *gateway* merupakan peralatan yang biasa pada *router* yang berfungsi hanya untuk memberikan jalur (*path*) ke jaringan lain atau jaringan internet sedangkan *proxy* menangani tiap *request* maupun konten yang masuk dari jaringan lokal ke jaringan luar.

A. Jenis-jenis Proxy

Proxy server berdasarkan referensi G-net (2002), terdiri dari dua jenis:

1. Forwarded Proxy

Forwarded Proxy adalah tipe *proxy server* yang meneruskan request dari klien ke jaringan web server yang dituju.

2. Reverse Proxy

Reverse Proxy adalah tipe *proxy server* yang membantu web server untuk menanganin request maupun response pada web server. *Proxy server* ini muncul seolah-olah sebagai web server pada saat klien me-request ke *web server* maupun me-response request tersebut dari *web server*.

B. Kegunaan Proxy Server

Adapun kegunaan mendasar dari *proxy server* sendiri adalah :

1. Filtering

Proxy server dapat digunakan untuk membatasi atau penyaringan konten yang disediakan oleh administrator yang biasa digunakan untuk memastikan jaringan atau internet digunakan sesuai dengan aturan penggunaan yang diperbolehkan. Dalam beberapa kasus, klien dapat mengabaikan *filtering proxy* dengan menggunakan *bypass proxy*.

2. Pemblokiran URL

Pemblokiran URL memungkinkan *administrator* untuk menolak situs tertentu berdasarkan URL. *Proxy server* mengecek setiap *request* dengan daftar halaman yang ditolak sebelum ia memperbarui *request* tersebut. Jika URL diblokir, *proxy server* tidak akan meminta atau memberikan halaman tersebut.

3. Bypass Filtering

Fungsi *bypass filter* pada *proxy server* adalah menyediakan informasi yang terdapat pada *web server* yang dituju. Jadi, penyaringan konten dapat dihindari dengan menggunakan *proxy server* sebagai *bypass filtering*.

4. Caching

Proxy Server memiliki fungsi *caching* yang dapat menyimpan objek atau konten yang secara populer diakses oleh klien. Dengan ini, *proxy server* dapat meningkatkan kecepatan akses ke web browser. Walaupun demikian, *proxy server*

secara *default* hanya dapat meng-*cache* konten yang bersifat statis.

5. Pencatatan (logging) dan Penyadapan (Eavesdropping)

Proxy server dapat juga berfungsi sebagai mencatat kegiatan yang dilakukan klien dalam sebuah jaringan sehingga dapat mempunyai fungsi lain sebagai alat penyadapan. Walaupun demikian, proxy server menyadap data sensitif seperti password atau internet banking karena adanya enkripsi pada koneksi seperti SSL.

Caching pada Proxy Server (Cache Server)

Menurut Gunung Sarjono (2005) *Proxy Cache Server* mula-mula dikembangkan untuk menyimpan halaman *web* yang sering diakses. Dimana *proxy server* yang menangani konten dapat menghilangkan akses *internet* yang berlebihan untuk mengambil kembali halaman yang sama berulang-ulang. Namun umumnya *proxy server* secara *default* tidak dapat menyimpan *web* yang bersifat dinamis (*Squid Developer Team. 2000*), tetapi *internet* pada saat ini lebih banyak bersifat dinamis sehingga membuat *caching proxy server* tidak efektif kecuali pada organisasi besar ataupun *ISP*.

A. Defenisi Cache

Definisi *Cache* Menurut Merriam-Websterr *Dictionary* adalah tempat penyimpanan memori komputer dengan waktu akses yang sangat singkat yang yang sering digunakan untuk menunggu mendapatkan instruksi berikutnya. Jadi *Cache* adalah suatu tempat untuk menyimpan sesuatu secara sementara, mekanisme untuk mempercepat transfer data dengan cara menyimpan data yang telah di akses, dengan harapan jika data yang sama akan diakses, akses akan menjadi lebih cepat.

B. Mekanisme Caching

Mekanisme *caching* akan menyimpan obyek-obyek yang merupakan hasil permintaan dari para pengguna, yang didapat dari *internet*. Karena *proxy cache server* bertindak sebagai perantara, maka *cache server* mendapatkan obyek-obyek tersebut lebih dahulu dari sumbernya untuk kemudian diteruskan kepada peminta yang sesungguhnya. Dalam proses tersebut, *proxy cache server* juga sekaligus menyimpan obyek-obyek tersebut untuk dirinya sendiri dalam ruang *disk* yang disediakan (*cache*).

Squid

Squid adalah *high-performance proxy caching server* untuk *web client*, yang sudah mendukung *FTP*, *ghoper*, dan *HTTP data object*. Berbeda dengan *software caching* yang lama, *Squid* menangani semua permintaan tunggal (*single*), *non-blocking*, *I/O-driven* proses. *Squid* menyimpan meta data dan terutama *hot-object* yang di simpan

di RAM, menyimpan *DNS lookups*, mendukung *non-blocking DNS lookups*, dan implementasi *negative-caching* jika permintaan gagal.

A. Fitur Squid

Squid menyediakan banyak fitur yang dapat membantu para administrator warnet, perkantoran ataupun di kawasan institusi dapat menikmati beberapa fitur yang ditawarkan oleh *squid*, diantaranya adalah :

1. Dengan menggunakan *squid* kita dapat menggunakan *bandwidth* yang kecil di koneksi *internet* kita pada waktu berselancar di dunia maya.
2. Dapat mengurangi lamanya waktu untuk *me-load* halaman *website* sehingga dengan menggunakan *squid* ini kecepatan kita untuk *berinternet* akan semakin cepat.
3. Dapat melindungi jaringan *host* internal kita oleh *proxy web traffic*
4. Mengumpulkan statistik mengenai *web traffic* ke dalam jaringan kita.
5. Mampu melindungi pengguna terhadap situs yang tidak sesuai dengan penggunaanya khususnya untuk penggunaan *internet* di sekolah yang harus dilindungi situs yang diakses oleh siswa.
6. Memastikan hanyalah pengguna yang legal yang mengakses *internet* dalam suatu area tertentu.
7. Meningkatkan *privacy* dari pengguna dari pemfilteran informasi yang sensitif dari *web request*.
8. Mampu mengurangi beban di *web server* kita.
9. Serta dapat mengkonversi (*HTTPS*) *encrypted request* menjadi *unencrypted (HTTP) request*.

B. Squid Cache Server

Squid merupakan sebuah *daemon* yang digunakan sebagai *proxy server* atau *cache server*. *Squid* dapat mendukung *HTPP*, *FTP*, *HTTPS*, *Internet Gopher* dan banyak lainnya. Pada umumnya *Squid* didesain untuk berjalan di atas *OS* seperti *Unix*, tetapi dapat diimplementasikan dalam beberapa *platform OS* lainnya. *Squid* dirilis oleh *GNU General Public Lisence* dan bersifat Gratis dan *Open Source*.

C. Hubungan dan Hirarki antar Squid Proxy Server

Dalam penggunaannya, dimungkinkan untuk menggunakan lebih dari satu *proxy server* untuk melayani permintaan. Hal ini mungkin terjadi jika *cache server* melayani permintaan (*request*) dalam jumlah yang besar pada satu satuan waktu. Misalnya pada sebuah institusi pendidikan seperti universitas, perusahaan dengan cabang yang tersebar di berbagai daerah, atau mungkin *ISP (Internet*

Service Provider) yang melayani ribuan *client* setiap harinya. Dengan menggabungkan beberapa *cache server* dalam suatu *hierarki*, maka setiap permintaan akan dapat dilayani dengan baik tanpa penundaan (*delay*) yang terlalu lama.

D. Cara kerja dan Arsitektur SQUID

Cara kerja Squid dilakukan oleh proxy dan cache. Sebagai proxy, Squid merupakan perantara dalam transaksi web. Squid ini menerima request dari klien, kemudian menerima request tersebut sebagai request oleh Squid, lalu request akan di forward ke server. Request ini mungkin akan di logged, di reject dan bahkan akan dimodifikasi terlebih dahulu sebelum diforward.

E. Replacement Policy

Replacement Policy atau aturan penggantian objek berdasarkan jenis tempat penyimpanan pada *squid* terdiri dari dua yaitu:

1. *Memory Replacement Policy*

Yaitu aturan penggantian objek yang terdapat pada *memory*

2. *Cache Replacement Policy*

Yaitu aturan penggantian objek yang terdapat pada *harddisk*

Menurut Visolve Team (2009) aturan penggantian objek yang menentukan kapan sebuah objek akan tetap disimpan sebagai *cache*, dibuang, atau diganti dengan objek yang baru. Terdapat beberapa metode aturan penggantian objek yang dapat digunakan dalam *Squid* yaitu:

1. LRU (*Least Recently Used*)

Menurut Dilley, John; Arlitt, Martin; Perret, Stephane (1999) LRU yaitu metode penggantian objek *cache* dimana objek tidak pernah diakses dalam waktu yang sangat lama.

2. GDSF (*Greedy-Dual Size Frequency*)

Adalah sebuah aturan penggantian objek yang mengoptimasi *hit rate* dari sebuah *object* dengan cara menyimpan ukuran objek yang lebih kecil dalam *cache*. Sehingga *object* tersebut mempunyai kesempatan lebih besar mendapatkan HIT

3. LFUDA (*Least Frequently Used with Dynamic Aging*)

Adalah sebuah aturan penggantian objek yang menyimpan objek yang *popular* di dalam tempat penyimpanan dan mencegah objek lain yang kurang popular untuk dapat masuk ke dalam tempat penyimpanan.

F. Manajemen Objek

Manajemen objek merupakan salah satu hal yang sangat penting pada operasional *Squid*. *Metadata* disimpan dalam *memory* dan merujuk ke objek tertentu di dalam *harddisk*. Hal ini dilakukan untuk mempercepat proses pencarian objek tertentu hanya dengan melihat *metadata*-nya saja.

Iptraf

Iptraf adalah program network monitoring yang sangat bagus. Iptraf datang dengan fungsi spesifik dan tidak dipenuhi dengan fitur luar biasa. Namun, secara fungsional, iptraf hebat sekali. Program ini sangat profesional di bidangnya. Tak heran apabila program ini berhasil duduk di posisi kedua, hanya kalah sedikit (0.05 poin) dari juara Pertama. Kekalahan ini pun semata-mata karena masalah fitur saja, Yang barangkali tidak diperlukan oleh setiap administrator.

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 dan kami mengundang Anda untuk turut serta berpartisipasi mengembangkan Ubuntu.

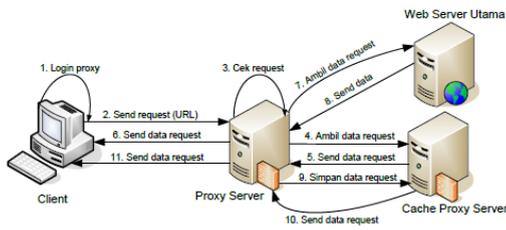
METODOLOGI PENELITIAN

Analisa Sistem

Seperti yang telah dijelaskan pada bab sebelumnya, bahwa *replacement* memiliki beberapa aturan dalam mengoptimalkan *cache* yang tersimpan dalam *proxy server*. Pada tugas akhir ini yang menjadi tema adalah penggunaan *internet*. Yang tidak jarang penggunaan *internet* ini sering memenuhi lalu lintas di *internet* dan penggunaan *bandwidth* yang tergolong cukup besar. Dengan terjadinya permasalahan tersebut banyak aktivitas yang sering terjadi dalam lalu lintas *internet*, terutama kegiatan yang bergantung pada *internet*, misalnya pencarian data/informasi di *internet* untuk kepentingan dalam mencari referensi yang dilakukan oleh setiap orang. Artinya, bagaimana cara agar penggunaan koneksi *internet* tidak sering terjadi dan penghematan *bandwidth* yang seminimal mungkin.

Dalam implementasi ini, menggunakan metode *replacement*. Dimana metode ini diterapkan dalam penempatan ruang penyimpanan *cache*, karena jika *cache* ditempatkan dalam satu *server proxy* dapat mengakibatkan kinerja dari *proxy* dapat menjadi lambat. Proses lambat ini dikarenakan ruang *harddisk* menjadi penuh. Serta bagaimana mengimplementasikan konsep *replacement* sebagai solusi permasalahan tersebut.

A. Metode Cache Replacement pada Proxy Server



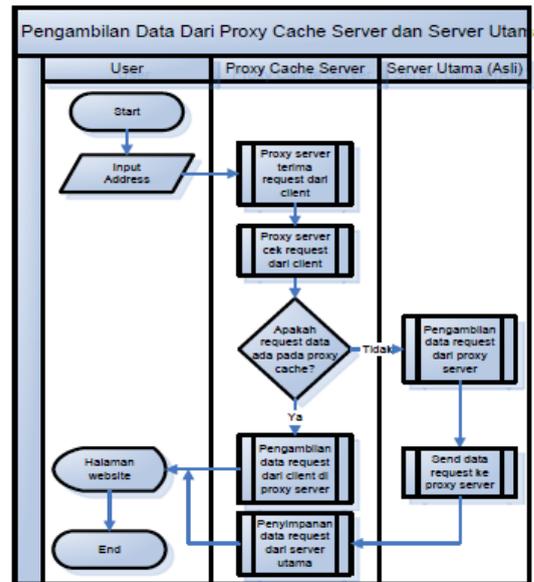
Gambar 3. Metode cache replacement pada proxy server

Penjelasan prinsip kerja metode cache replacement pada proxy server :

1. Client melakukan login proxy dengan menggunakan browser.
2. Client melakukan request website berupa alamat URL dengan menggunakan web browser ke proxy server.
3. PC proxy server melakukan cek request dari client.
4. Jika data request dari client tersedia di proxy server, maka selanjutnya proxy server akan melakukan pengambilan data ke cache proxy server.
5. Setelah data telah di ambil, maka selanjutnya cache proxy server melakukan pengiriman data kembali ke proxy server.
6. Proxy server melanjutkan pengiriman data ke client.
7. Jika data yang direquest oleh client tidak tersedia pada cache proxy server, maka proxy server melakukan pengambilan data ke web server utama.
8. Web server utama melakukan pengiriman data ke proxy server.
9. Selanjutnya proxy server mengirim data yang diterima dari web server utama ke cache proxy server untuk disimpan.
10. Cache proxy server mengirim data ke proxy server.
11. Setelah proxy server menerima data dari cache proxy server, maka selanjutnya proxy server mengirim data ke client.

B. Proses Bisnis

Proses bisnis ini menggambarkan mengenai proses yang terjadi pada sistem penyimpanan data dan sistem request data. Dengan adanya input pada sistem akan menghasilkan output pada sistem ini.



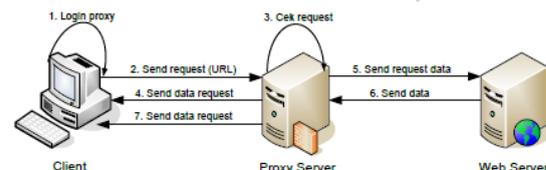
Gambar 4. Proses Bisnis

Pada gambar 4 menggambarkan proses bisnis yang terjadi pada cache proxy. Pada awalnya user melakukan input address dengan menggunakan browser. Setelah user menginputkan address, proxy cache server menerima request dari user yang selanjutnya akan dilakukan pencarian data request sesuai yang diminta oleh user. Jika data tersebut ada di proxy cache server, maka langkah selanjutnya akan dilakukan pengambilan data dan dikirim ke user dalam bentuk halaman website. Dan jika data yang di request tidak ada didalam proxy cache server, maka request data akan diteruskan ke server utama (asli). Setelah melakukan request data ke server utama, selanjutnya data tersebut akan disimpan ke dalam proxy cache server dan dikirim ke user

Desain Sistem

Desain sistem dibuat bertujuan untuk memberikan gambaran secara umum kepada user antara lain adalah proses bisnis dan flowchart sistem proxy cache server.

A. Alur Proses Sistem Cache Proxy



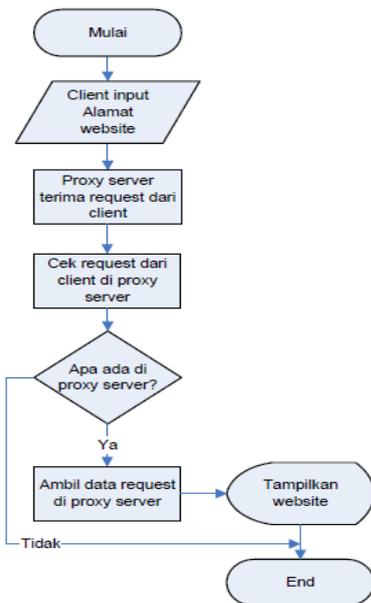
Gambar 5. Alur Proses Sistem Cache Proxy

Pada gambar 5 menjelaskan bagaimana alur dan proses dari proxy cache server dari awal proses

hingga akhir. Untuk lebih detailnya berikut ini adalah langkah – langkah dari alur proses sistem *cache* :

1. *Client* melakukan login *proxy* dengan menggunakan browser.
2. *Client* melakukan *request website* berupa alamat URL dengan menggunakan *web browser* ke *proxy server*.
3. *PC proxy server* melakukan cek *request* dari *client*.
4. Jika data *request* dari *client* tersedia di *proxy server*. Maka selanjutnya *proxy server* akan melakukan pengiriman data ke *client*.
5. Dan jika data tidak tersedia dalam *Proxy server* maka *proxy server* melanjutkan *request* dari *client* ke *server* utama asli
6. *Server* utama/asli akan melakukan pengiriman data ke *proxy server*
7. Selanjutnya *proxy server* akan mengirim data *request* ke *client*.

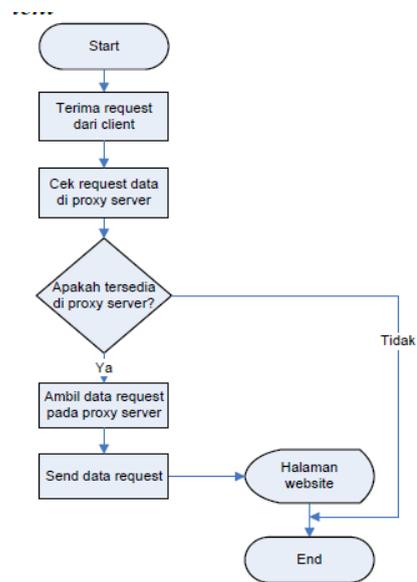
B. Flowchart Proxy Server Terima Request dari Client



Gambar 6. Flowchart Proxy Server Terima Request dari Client

Pada gambar 6 menjelaskan tentang alur proses *proxy server* terima *request* dari *client*.

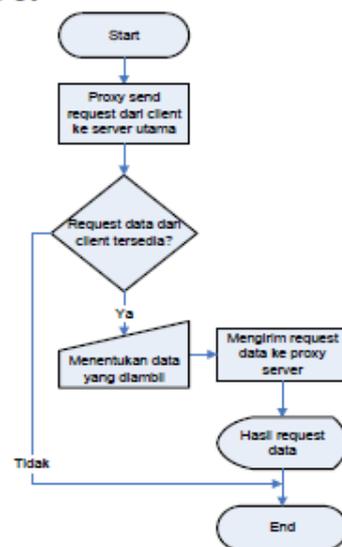
C. Flowchart Proxy Server Cek Request dari Client



Gambar 7. Flowchart Proxy Server Cek Request dari Client

Pada gambar 7 menjelaskan tentang alur proses *Proxy Server* Cek *Request* dari *Client*.

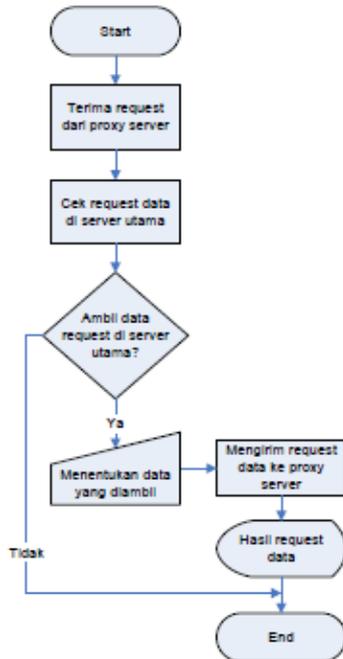
D. Flowchart Pengambilan Data Request dari Proxy Server



Gambar 8. Flowchart Pengambilan Data Request dari Proxy Server

Pada gambar 8 menjelaskan tentang alur proses Pengambilan Data *Request* dari *Proxy Server*

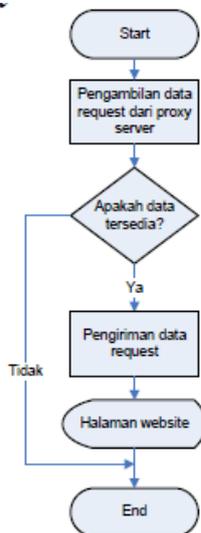
E. *Flowchart Send Data Request ke Proxy Server*



Gambar 9. *Flowchart Send Data Request ke Proxy Server*

Pada gambar 9 menjelaskan tentang alur proses *Send Data Request* ke *Proxy Server*.

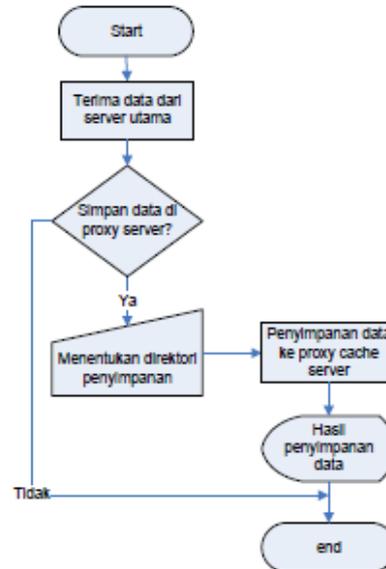
F. *Flowchart Pengambilan Data Request dari Client di Proxy Server*



Gambar 10. *Flowchart Pengambilan Data Request dari Client di Proxy Server*

Pada gambar 10 menjelaskan tentang alur proses *Pengambilan Data Request* dari *Client* di *Proxy Server*.

G. *Flowchart Penyimpanan Data Request dari Server Utama*



Gambar 11. *Flowchart Penyimpanan Data Request dari Server Utama*

Pada gambar 11 menjelaskan tentang alur proses *Penyimpanan Data Request* dari *Server Utama*.

Skenario Pengujian

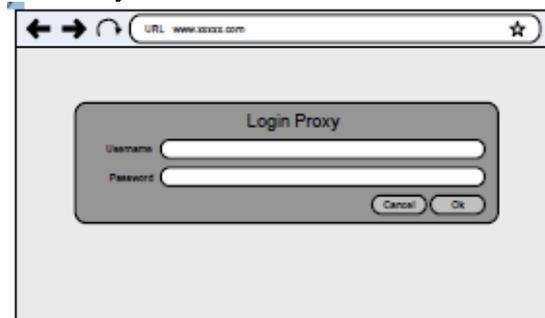
Skenario pengujian yang akan dilakukan dalam penelitian ini yaitu sebagai berikut:

1. Koneksi ke *proxy server*.
2. Mengendalikan kebutuhan *bandwidth*
3. Mengatur kecepatan akses pada *proxy server* dan *client*.
4. Mengevaluasi hasil kinerja dari *proxy server*.

Desain Perencanaan Antarmuka

Pada perencanaan antarmuka, digunakan sebagai format desain yang digunakan untuk setiap tampilan sistem yang nantinya dijadikan sebagai tampilan utama.

A. Desain Perencanaan Antarmuka Login Proxy



Gambar 12. *Desain Perencanaan Antarmuka Login Proxy*

Pada gambar 12 menjelaskan tentang Desain Perencanaan Antarmuka *Login Proxy*.

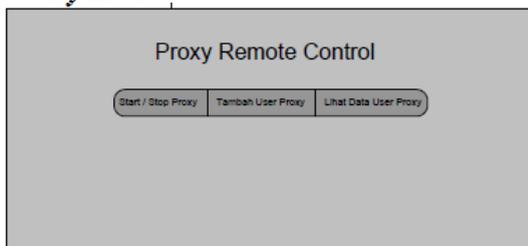
B. Desain Perencanaan Antarmuka *Error Login Proxy*



Gambar 13. Desain Perencanaan Antarmuka *Error Login Proxy*

Pada gambar 13 menjelaskan tentang Desain Perencanaan Antarmuka *Error Login Proxy*.

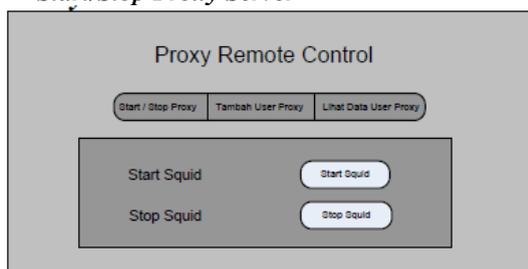
C. Desain Perencanaan Antarmuka *Control Proxy Server*



Gambar 14. Desain Perencanaan Antarmuka *Control Proxy Server*

Pada gambar 14 menjelaskan tentang Desain Perencanaan Antarmuka *Control Proxy Server*.

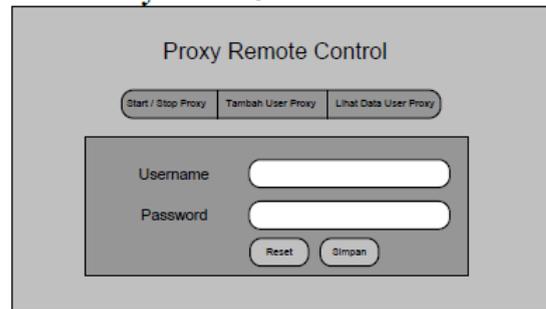
D. Desain Perencanaan Antarmuka *Start/Stop Proxy Server*



Gambar 15. Desain Perencanaan Antarmuka *Start/Stop Proxy Server*

Pada gambar 15 menjelaskan tentang Desain Perencanaan Antarmuka *Start/Stop Proxy Server*.

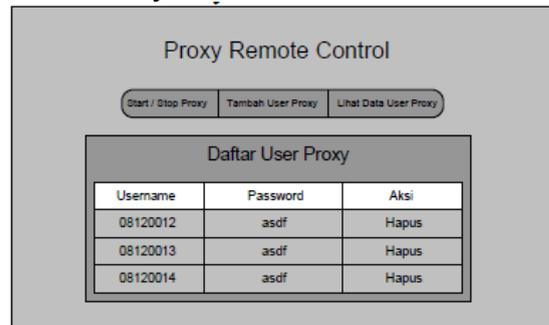
E. Desain Perencanaan Antarmuka *Tambah User Proxy*



Gambar 16. Desain Perencanaan Antarmuka *Tambah User Proxy*

Pada gambar 16 menjelaskan tentang Desain Perencanaan Antarmuka *Tambah User Proxy*.

F. Desain Perencanaan Antarmuka *Lihat Data User Proxy*

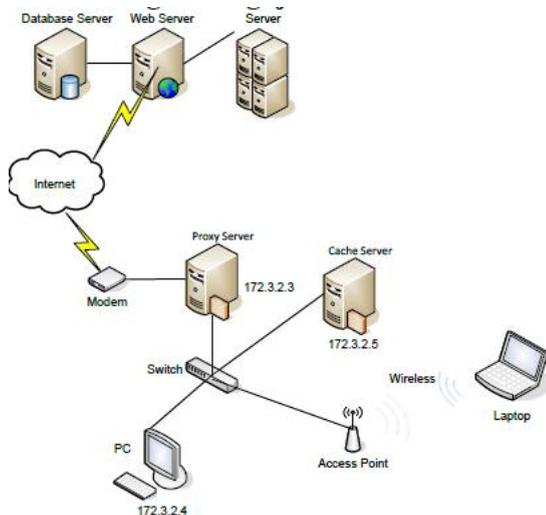


Gambar 17. Desain Perencanaan Antarmuka *Lihat Data User Proxy*

Pada gambar 17 menjelaskan tentang Desain Perencanaan Antarmuka *Lihat Data User Proxy*.

IMPLEMENTASI SISTEM

Desain Jaringan Pengujian



Gambar 18 Desain Jaringan Pengambilan Data dari Proxy Cache Server dan Permintaan Data ke Server Utama

Pada gambar 18 menjelaskan tentang Desain Jaringan Pengambilan Data dari Proxy Cache Server dan Permintaan Data ke Server Utama.

Konfigurasi IP pada Server

Pada sub-bab konfigurasi IP pada PC Proxy Server, dijelaskan pengaturan konfigurasi pada IP Proxy Server. Server ini terkoneksi dengan internet yang akan diteruskan pada PC Client yang terhubung pada switch. Cara setting yang digunakan untuk mengatur IP pada Proxy Server yaitu dengan perintah `nano /etc/network/interfaces`.

```

Berkas Ubah Tampilan Cari Alat Dokumen Bantuan
Buka Simpan Batalkan
interfaces
auto eth0
iface eth0 inet static
address 172.3.2.3
netmask 255.255.255.0
    
```

Gambar 19 Konfigurasi IP eth0 pada Proxy Server

Pada gambar 19 menjelaskan IP eth0 dengan `inet static` pada Proxy server. Pada eth0 memiliki IP 172.3.2.3 dengan `netmask 255.255.255.0`.

```

wlan0 Link encap:Ethernet HWaddr 90:4c:e5:24:b2:fd
inet addr:172.11.12.100 Bcast:172.11.12.255 Mask:255.255.255.0
inet6 addr: fe80::924c:e5ff:fe24:b2fd/64 Scope:Link
IP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:178 errors:0 dropped:0 overruns:0 frame:0
TX packets:69 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:57309 (57.3 KB) TX bytes:9661 (9.6 KB)

root@trin-lapt-0pc:~#
    
```

Gambar 20 Konfigurasi IP Wlan0 pada Proxy Server

Pada gambar 20 menjelaskan IP Wlan0 dengan `inet DHCP` pada Proxy server. Pada Wlan0 memiliki IP 172.11.12.100 dengan `netmask 255.255.255.0`.

Konfigurasi IP pada cache server

Pada sub-bab konfigurasi IP pada Cache Server, dijelaskan pengaturan konfigurasi pada Cache Server. Server ini terkoneksi dengan internet yang akan diteruskan pada PC Client dan Proxy Server yang terhubung pada switch. Cara setting yang digunakan untuk mengatur IP pada Cache Server yaitu dengan perintah `nano /etc/network/interfaces`.

```

Berkas Ubah Tampilan Cari Alat Dokumen Bantuan
Buka Simpan Batalkan
interfaces
auto eth0
iface eth0 inet static
address 172.3.2.5
netmask 255.255.255.0
    
```

Gambar 21 Konfigurasi IP eth0 pada Cache Server

Pada gambar 21 menjelaskan IP eth0 dengan `inet static` pada Cache server. Pada eth0 memiliki IP 172.3.2.5 dengan `netmask 255.255.255.0`.

Konfigurasi IP pada Client

Pada sub-bab konfigurasi IP pada Client, dijelaskan pengaturan konfigurasi pada Client. Client ini terkoneksi dengan internet yang akan diteruskan pada PC Cache Server dan Proxy Server yang terhubung pada switch. Cara setting yang digunakan untuk mengatur IP pada Client yaitu dengan perintah `nano /etc/network/interfaces`.

```

Berkas Ubah Tampilan Cari Alat Dokumen Bantuan
Buka Simpan Batalkan
interfaces
auto eth0
iface eth0 inet static
address 172.3.2.4
netmask 255.255.255.0
    
```

Gambar 22 Konfigurasi IP eth0 pada Client

Pada gambar 22 menjelaskan IP eth0 dengan `inet static` pada Cache server. Pada eth0 memiliki IP 172.3.2.4 dengan `netmask 255.255.255.0`.

Konfigurasi IP Forward pada Proxy Server

Pada sub-bab konfigurasi IP forward pada server dijelaskan pengaturan konfigurasi pada IP Forward pada server. Untuk melakukan IP forward konfigurasi dilakukan pada `nano /etc/sysctl.conf`.

```

#net.ipv4.ip_forward=1
    
```

Gambar 23 Perintah IP forward

Pada gambar 23 menjelaskan bahwa pada awalnya, perintah IP forward konfigurasi untuk mengaktifkan IP forward yaitu dengan cara menghilangkan tanda “#” pada awal baris. Pada awal

konfigurasi terdapat tanda “#” di awal baris, sehingga IP *forward* masih belum aktif.

```
net.ipv4.ip_forward=1
```

Gambar 24 IP Forward aktif

Pada gambar 24 menunjukkan bahwa perintah untuk melakukan IP *forward* telah aktif yang ditandai dengan tidak adanya tanda “#” pada awal baris. IP *forward* digunakan untuk meneruskan arus yang pada *ethernet* awal kepada *ethernet* tujuan. Dengan mengaktifkan IP *forward*, maka dapat melakukan *routing*.

Konfigurasi Internet Forwarding IPTables pada Server

Pada sub-bab konfigurasi *internet forwarding IPTables* pada *server* dijelaskan pengaturan *internet* yang diteruskan dari *Wlan0* menuju *eth0* melalui *IPTables*. Untuk melakukan *internet forwarding*, konfigurasi dilakukan pada *nano letc/rc.local*.

```
sudo systemctl -p
sudo iptables -A FORWARD -i wlan0 -o eth0 -s 172.3.2.0/24 -m state --state NEW -j ACCEPT
sudo iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
sudo iptables -A POSTROUTING -t nat -j MASQUERADE
sudo echo 1 > /proc/sys/net/ipv4/ip_forward
```

Gambar 25 Konfigurasi IPTables.

Pada gambar 25 menjelaskan konfigurasi *IPTables* untuk melakukan *internet forwarding*. *Internet forwarding* yang dikonfigurasi berasal dr *wlan0* yang akan diteruskan menuju *eth0* dengan *network* tujuan 172.3.2.0/24. Dengan dilakukan konfigurasi *IPTables* pada di *rc.local* maka konfigurasi *IPTables* tersebut akan berlaku permanen sehingga tidak akan hilang meskipun *server* di *restart*.

Konfigurasi Create Password Proxy

Pada sub-bab *create password* pada *proxy server* dilakukan dalam perintah *shell command* dengan mengetik perintah “*htpasswd -c /etc/squid/squid/passwd (username)*”.

```
root@squidku-laptop: ~
File Edit View Terminal Tabs Help
root@squidku-laptop: ~
squidku@squidku-laptop:~$ sudo bash
root@squidku-laptop:~# htpasswd -c /etc/squid/passwd squidku
New password:
Re-type new password:
Adding password for user squidku
root@squidku-laptop:~#
```

Gambar 26 Create User dan Password Proxy.

Pada gambar 26 menjelaskan perintah untuk membuat *user* dan *password proxy*. *User* dan *password* ini digunakan untuk *login* ke *proxy* yang fungsinya untuk mendapatkan akses *internet*.

Perintah Start Squid

Pada sub-bab *start squid* pada *proxy server* dilakukan dalam perintah *shell command.n* Perintahnya */etc/init.d/squid start*.

```
root@squidku-laptop:~# /etc/init.d/squid start
Rather than invoking init scripts through /etc/init.d, use the service(8)
utility, e.g. service squid start
Since the script you are attempting to invoke has been converted to an
Upstart job, you may also use the start(8) utility, e.g. start squid
squid start/running, process 30690
root@squidku-laptop:~#
```

Gambar 27 Start Squid.

Pada gambar 27 menjelaskan hasil dari perintah *start squid*. Hasil dari perintah tersebut adalah “*squid start/running, process 30690*”.

Perintah Stop Squid

Pada sub-bab *stop squid* pada *proxy server* dilakukan dalam perintah *shell command*. Perintahnya */etc/init.d/squid stop*.

```
root@squidku-laptop:~# /etc/init.d/squid stop
Rather than invoking init scripts through /etc/init.d, use the service(8)
utility, e.g. service squid stop
Since the script you are attempting to invoke has been converted to an
Upstart job, you may also use the stop(8) utility, e.g. stop squid
squid stop/waiting
root@squidku-laptop:~#
```

Gambar 28 Stop Squid.

Pada gambar 28 menjelaskan hasil dari perintah *stop squid*. Hasil dari perintah tersebut adalah “*squid stop/waiting*”.

Konfigurasi Squid

Pada sub-bab konfigurasi *squid*, dijelaskan pengaturan konfigurasi pada *rule squid*. Cara setting yang digunakan *nano/etc/squid/squid.conf*.

A. Konfigurasi Login Proxy

Pada sub-bab konfigurasi *login proxy*, dijelaskan tentang perintah yang dimasukkan dalam *squid*. Untuk membuka *squid.conf* dapat dilakukan dengan perintah *nano/etc/squid/squid.conf*.

```
##-----##
# Login Proxy
#-----#
auth_param basic program /usr/lib/squid/nsc_auth /etc/squid/passwd
auth_param basic children 5
auth_param basic realm Squid proxy-caching web server
auth_param basic credentials_ttl 2 hours
auth_param basic casesensitive off

acl ncsa_users proxy_auth REQUIRED
httpd_access allow ncsa_users
```

Gambar 29 Konfigurasi Login Proxy.

Pada gambar 29 menjelaskan tentang fungsi dari perintah yang ada di dalam *squid.conf*. Pada perintah “*auth_param basic program*” dijelaskan tentang direktori dari perintah *authentication* yang di butuhkan oleh *squid*. Dan pada “*acl ncsa_users proxy_auth REQUIRED*” dijelaskan tentang perintah user *proxy* yang wajib memasukkan *username* dan *password* yang diminta. Sedangkan pada “*http access allow ncsa users*” dijelaskan bahwa yang berhak mengakses adalah *client* yang telah melakukan *login ncsa*.

B. Konfigurasi Cache dan Object

Pada sub-bab konfigurasi *cache* dan *object*, dijelaskan tentang perintah yang dimasukkan dalam *squid*. Untuk membuka *squid.conf* dapat dilakukan dengan perintah *nano/etc/squid/squid.conf*.

```
#-----#  
# Cache & Object  
#-----#  
cache_mem 8 MB  
cache_swap_low 98  
cache_swap_high 99  
max_filedes 8192  
maximum_object_size 128 MB  
minimum_object_size 0 KB  
maximum_object_size_in_memory 128 KB  
  
ipcache_size 4096  
ipcache_low 98  
ipcache_high 99  
fqdn_cache_size 4096
```

Gambar 30 Konfigurasi Cache dan Object.

Pada gambar 30 menjelaskan tentang fungsi dari perintah yang ada di dalam *squid.conf*. Penjelasan antara lain sebagai berikut:

1. *Cache_mem* 8 MB dijelaskan *squid* akan mempergunakan maksimum RAM untuk memberikan kepuasan pada *request* yang muncul setiap kali dari *user*. Ukuran 8MB ini akan memberikan ruang pada memori fisik untuk terus menempatkannya pada memori sampai penuh. Jika kondisi telah mencapai puncaknya, maka *squid* akan memilih pada lokasi memori yang akan di*swap* pada *harddisk*.
2. *Cache_swap_low* 98 dijelaskan settingan ini akan membuat masa untuk *squid* melakukan pembersihan *cache*. Jika anda memiliki *cache* pada *harddisk* berukuran 10GB, pada penggunaan 9GB *squid* akan melakukan pembersihan *cache*.
3. *Cache_swap_high* 99 dijelaskan jika settingan ini akan membuat sisi agresif *squid* akan bekerja. Begitu mencapai kondisi nilai 99% dari kapasitas, maka *squid* akan melakukan pembersihan *cache*.
4. *Maximum_object_size* 128 MB dijelaskan dengan option ini, ukuran file maksimum yang disimpan oleh *squid* *cache* bisa dibatasi. Dengan kata lain objek yang lebih besar dari bilangan ini tidak akan disimpan ke dalam *disk* yang sudah disisihkan buat *cache*.
5. *Minimum_object_size* 0 KB dijelaskan dengan option ini maka objek yang lebih kecil dari bilangan yang disebutkan tidak akan disimpan ke dalam *disk* yang sudah disisihkan buat *cache*.

C. Konfigurasi Direktori Cache

Pada sub-bab konfigurasi direktori *cache*, dijelaskan tentang perintah yang dimasukkan dalam *squid*. Untuk membuka *squid.conf* dapat dilakukan dengan perintah *nano/etc/squid/squid.conf*.

```
#-----#  
# direktori cache  
#-----#  
cache_dir aufs /home/squidku/cache 7699 16 256  
#cache_dir aufs /home/proxy2 7900 16 256  
#cache_dir aufs /home/proxy3 7900 16 256  
  
cache_access_log /var/log/squid/access.log  
cache_log /var/log/squid/cache.log  
cache_store_log none  
pid_filename /var/run/squid.pid  
cache_swap_log /var/log/squid/swap.state  
dns_nameservers /etc/resolv.conf  
emulate_httpd_log off  
hsts_file /etc/hosts  
half_closed_clients off  
negative_ttl 1 minutes
```

Gambar 31 Konfigurasi Direktori Cache.

Pada gambar 31 menjelaskan tentang fungsi dari perintah yang ada di dalam *squid.conf*. Penjelasan antara lain sebagai berikut:

1. *Cache_dir aufs /home/squidku/cache* 7000 16 256 dijelaskan option pada *cache_dir* menentukan sistem penyimpanan seperti apa yang akan digunakan (*aufs*), nama direktori tempat penyimpanan *cache* (*/home/squidku/cache*), ukuran *disk* dalam *megabytes* yang digunakan oleh direktori tempat penyimpanan *cache* (7000 Mbytes), jumlah subdirektori pertama yang akan dibuat di bawah */home/squidku/cache* (16), dan jumlah subdirektori kedua yang akan diciptakan di bawah subdirektori pertama tadi (256).
2. *Cache_log /var/log/squid/cache.log* dijelaskan option ini berfungsi untuk menempatkan hasil kerja dari sistem *squid* yang telah dijalankan, direktori *log* terdapat pada */var/log/squid/cache.log*.

D. Konfigurasi Port

Pada sub-bab konfigurasi *port*, dijelaskan tentang perintah yang dimasukkan dalam *squid*. Untuk membuka *squid.conf* dapat dilakukan dengan perintah *nano/etc/squid/squid.conf*.

```
#-----#  
# Rules: Safe Port  
#-----#  
acl all src 0.0.0.0/0.0.0.0  
acl manager proto cache_object  
acl localhost src 127.0.0.1/255.255.255.255  
acl to_localhost dst 127.0.0.0/8  
acl ssl_ports port 443 563 873 # https snexus rsync  
acl safe_ports port 80 # http  
acl safe_ports port 20 21 # ftp  
acl safe_ports port 70 # gopher  
acl safe_ports port 210 # wais  
acl safe_ports port 1025-65535 # unregistered ports  
acl safe_ports port 631 # cups  
acl safe_ports port 10000 # webmin  
acl safe_ports port 901 # SWAT  
acl safe_ports port 200 # http-mgmt  
acl safe_ports port 488 # gss-http  
acl safe_ports port 591 # filemaker  
acl safe_ports port 777 # multiling http  
acl safe_ports port 873 # rsync  
acl safe_ports port 110 # POP3  
acl safe_ports port 25 # SMTP  
acl safe_ports port 2095 2096 # webmail from cpanel  
acl safe_ports port 2082 2083 # cpanel
```

Gambar 32 Konfigurasi Port.

Pada gambar 32 menjelaskan tentang fungsi dari perintah yang ada di dalam *squid.conf*. Penjelasan antara lain sebagai berikut:

1. *Acl all src 172.2.2.0/24* dijelaskan option ini berfungsi untuk membatasi akses *internet*. Yang dapat menggunakan akses ini hanya *client* yang menggunakan ip dengan *network* 172.2.2.0 saja.
2. *Acl localhost src 127.0.0.1/255.255.255.255* dijelaskan

option ini berfungsi untuk memberikan hak akses untuk localhost.

3. *Acl safe_ports port* dijelaskan option ini berfungsi untuk membuka hak akses untuk *port* yang telah tercantum dalam *squid.conf*.

E. Konfigurasi Refresh Pattern

Pada sub-bab konfigurasi *refresh pattern*, dijelaskan tentang perintah yang dimasukkan dalam *squid*. Untuk membuka *squid.conf* dapat dilakukan dengan perintah *nano/etc/squid/squid.conf*.

```
#-----#
# Refresh Pattern
#-----#
refresh_pattern ^ftp: 1440 20% 10080
refresh_pattern ^gopher: 1440 0% 1440
refresh_pattern -i \.(gif|png|jpg|jpeg|ico)$ 10080 90% 43200 override-
expire ignore-no-cache ignore-private
refresh_pattern -i \.(iso|avi|wav|mp3|mp4|mpeg|mpg|swf|flv|x-flv)$ 43200
90% 43200 override-expire ignore-no-cache ignore-private
refresh_pattern -i \.(deb|rpm|exe|ram|bin|pdf|ppt|doc|tif)$ 10080 90%
43200 override-expire ignore-no-cache ignore-private
refresh_pattern -i \.(zip|gz|jar|lha|lzh|tar|tgz|cab|rar)$ 10080 95% 43200
override-expire ignore-no-cache ignore-private
refresh_pattern -i \.(html|htm|css|js|php|asp|aspx|cgi) 1440 40% 40320
refresh_pattern . 0 20% 4320
```

Gambar 33 Konfigurasi *Refresh Pattern*.

Pada gambar 33 menjelaskan tentang fungsi dari perintah yang ada di dalam *squid.conf*. Dengan memaksa sedikit supaya obyek lebih intensif di lokal *squid* dan waktu simpan ditambah sebelum proses validasi terjadi (mis. Validasi terjadi per 3jam dengan penyimpanan obyek terlalu lama 3 bulan, untuk ftp bisa lebih lama lagi).

F. Konfigurasi Allow Access

Pada sub-bab konfigurasi *allow access*, dijelaskan tentang perintah yang dimasukkan dalam *squid*. Untuk membuka *squid.conf* dapat dilakukan dengan perintah *nano/etc/squid/squid.conf*.

```
#-----#
# ALLOWED ACCESS
#-----#
acl squidku src 172.3.2.0/24
http_access allow squidku
http_access allow localhost
http_access deny all
http_reply_access allow all
icp_access allow squidku
icp_access allow localhost
icp_access deny all
always_direct deny all
reply_header_max_size 20 KB
reply_body_max_size 10240000 allow magic_words2
```

Gambar 34 Konfigurasi *Allow Access*.

Pada gambar 34 menjelaskan tentang fungsi dari perintah yang ada di dalam *squid.conf*. Penjelasannya antara lain sebagai berikut:

1. *Acl squidku src 172.3.2.0/24* dijelaskan option ini berfungsi untuk memberikan hak akses untuk *client* yang menggunakan ip dengan *network 172.3.2.0/24*.
2. *http_access allow squidku* dijelaskan option ini memberikan hak akses pada *client* untuk mengakses *website*.
3. *http_access allow localhost* dijelaskan option ini memberikan hak akses pada *localhost* untuk mengakses *website*.

4. *http_access deny all* dijelaskan option ini menolak hak akses kecuali pengguna berada pada *network 172.3.2.0/24*.
5. *Reply_header_max_size 20 KB* dijelaskan option ini hanya membuka atau menerima web yang *contentnya* hanya berukuran 20 KB.
6. *Reply_body_max_size 10240000* dijelaskan option ini hanya membuka atau menerima web yang *contentnya* hanya berukuran 10240000.

G. Konfigurasi Cache CGI dan Administrative

Pada sub-bab konfigurasi *allow access*, dijelaskan tentang perintah yang dimasukkan dalam *squid*. Untuk membuka *squid.conf* dapat dilakukan dengan perintah *nano/etc/squid/squid.conf*.

```
#-----#
# Cache CGI & Administrative
#-----#
cache_mgr erwin.mellowdik@gmail.com
cache_effective_user proxy
cache_effective_group proxy
coredump_dir /var/spool/squid
shutdown_lifetime 10 seconds
logfile_rotate 14
```

Gambar 35 Konfigurasi *Cache CGI dan Administrative*.

Pada gambar 35 menjelaskan tentang fungsi dari perintah yang ada di dalam *squid.conf*. Penjelasannya antara lain sebagai berikut:

1. *Cache_mgr erwin.mellowdik@gmail.com* dijelaskan option ini berfungsi untuk memberikan informasi data atau *contact* dari admin jika terjadi kesalahan disaat melakukan *login proxy*.
2. *Cache_effective_user proxy* dijelaskan jika *squid cache* dijalankan oleh root, maka user yang akan menjalankannya akan diubah sesuai UID/GID user yang disebutkan pada gambar 4.18. Pada contoh option diatas jika *squid* dijalankan oleh root, maka *squid cache* akan dijalankan dengan UID/GID *squid*.

HASIL DAN PEMBAHASAN

Skenario Pengujian

Skenario pengujian yang akan dilakukan dalam penelitian ini yaitu sebagai berikut:

1. Koneksi ke *proxy server*.
2. Mengendalikan kebutuhan *bandwidth*.
3. Memberikan kecepatan akses pada *proxy server* dan *client*.
4. Mengevaluasi hasil kinerja dari *proxy server*.

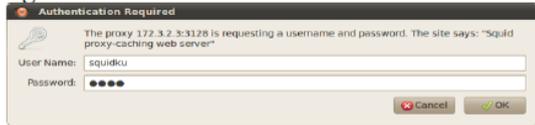
Pengujian Koneksi ke Proxy Server

Pada sub-bab pengujian koneksi *proxy server*, dijelaskan mengenai cara mendapatkan akses *internet* dari *proxy server*. Mendapatkan akses

internet dapat melalui *login proxy* yang tersedia dalam setiap *browser*.

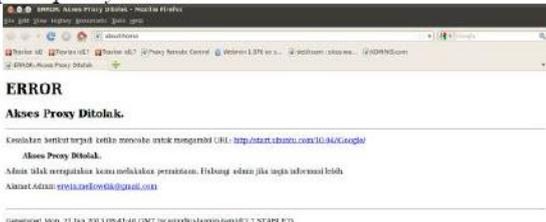
Koneksi ke Proxy Server Melalui Login Proxy

Pada sub-bab pengujian koneksi ke *proxy server* melalui *login proxy*, dijelaskan mengenai *login* melalui *browser*.



Gambar 36 Login Proxy melalui Browser

Pada gambar 36 merupakan gambar mengenai cara *login proxy* untuk mendapatkan akses internet dari *proxy server*. Langkah ini sangat penting karena untuk mendapatkan akses internet maka *client* wajib memasukkan *username* dan *password* yang telah di daftarkan pada *proxy server*.



Gambar 37 Tampilan Error ketika Client tidak melakukan Login Proxy

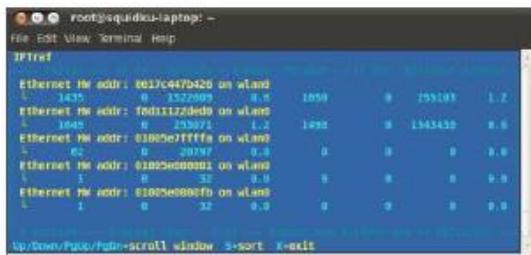
Pada gambar 37 merupakan gambar hasil dari pesan *error* apabila *client* tidak melakukan *login proxy* untuk mendapatkan akses internet.

Pengujian Pengendalian Kebutuhan Bandwidth

Pada sub-bab pengujian pemakaian *bandwidth* melalui pemantauan kebutuhan *bandwidth* di waktu melakukan browsing ke situs yang memiliki ukuran *content* yang besar.

Uji Coba Iptraf

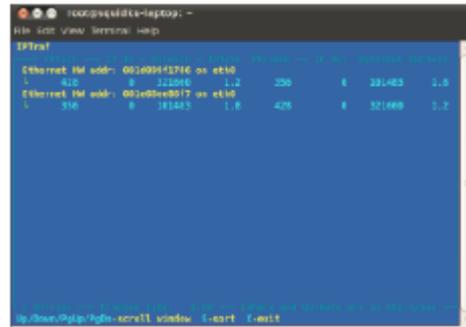
Pada sub-bab uji coba browsing dan mengukur kebutuhan *bandwith* melalui *iptraf*.



Gambar 5.3 Kebutuhan Bandwidth sebelum tersimpan di cache sever.

Pada gambar 5.3 merupakan gambar hasil *record* kebutuhan *bandwidth* saat pertama kali melakukan browsing ke situs *kompas.com* dan data dari *kompas.com* belum tersimpan pada *cache server*. Hasil *record* menunjukkan angka 255103

ByteOut bandwidth yang di butuhkan untuk menampilkan situs *kompas.com* pada *MCAAddress* 0017c447b426 dari *wlan0 proxy server*



Gambar 5.4 Kebutuhan Bandwidth sesudah tersimpan di cache sever

Pada gambar 5.4 merupakan gambar hasil *record* kebutuhan *bandwidth* saat dilakukan perulangan *request* ke *website* *kompas.com* oleh *client* dimana data dari *kompas.com* telah tersimpan dalam *cache server*. Hasil *record* menunjukkan angka 101486 *ByteOut bandwidth* yang di butuhkan untuk menampilkan situs *kompas.com* yang datanya sudah tersimpan dalam *cache server* pada *MCAAddress* 001d099f1786 dari *eth0 client*.

Pengujian Kecepatan Akses Pada Proxy Server dan Client

Pada sub-bab pengujian kecepatan akses melalui pemantauan di waktu melakukan browsing ke situs yang memiliki ukuran *content* yang besar.

Uji Coba Browsing

Pada sub-bab uji coba browsing ini dengan menggunakan metode pengukuran waktu *load web* yang akan di akses.

Tabel 1 Uji coba kecepatan akses

No	Situs	Waktu Sebelum Disimpan	Waktu Sesudah Disimpan
1	www.kompas.com	41 detik	15 deik
2	www.detik.com	46 detik	13 detik
3	www.id.yahoo.com	32 detik	8 detik
4	www.tabloidpulsa.co.id	39 detik	12 detik

Pada tabel 1 merupakan hasil *record* kecepatan akses *website* yang dilakukan oleh *client*. Dalam tabel terdapat beberapa situs yang telah di akses oleh *client* dan juga terdapat waktu yang dibutuhkan untuk membuka *website* yang dituju. Di dalam tabel terdapat perbedaan waktu akses untuk membuka *website*, perbedaan yang terjadi bisa sampai separuh dari hasil membuka *website*.

Mengevaluasi Hasil Kinerja Dari Proxy Server

Pada sub-bab mengevaluasi hasil kinerja *proxy server* melalui pemantauan *iptraf* diwaktu melakukan browsing.

Uji Coba Browsing dan Iptraf

Pada sub-bab uji coba browsing dan iptraf ini dengan menggunakan metode pengukuran waktu *load web* dan pengukuran kebutuhan *bandwidth* yang dibutuhkan disaat sebelum *cache* disimpan dan sesudah *cache* disimpan.

Tabel 5.2 Mengevaluasi Hasil Kinerja Proxy Server

No	Situs	Waktu Sebelum Disimpan	Bandwidth yang dibutuhkan	Waktu Sesudah Disimpan	Bandwidth yang dibutuhkan
1	kompas.com	41 detik	255103 <i>ByteOut</i>	15 deik	101486 <i>ByteOut</i>
2	detik.com	46 detik	293209 <i>ByteOut</i>	13 detik	132878 <i>ByteOut</i>
3	id.yahoo.com	32 detik	191137 <i>ByteOut</i>	8 detik	95619 <i>ByteOut</i>
4	tabloidpula.com	39 detik	197408 <i>ByteOut</i>	12 detik	97954 <i>ByteOut</i>

Pada tabel 5.2 merupakan hasil *record* kecepatan akses *website* yang dilakukan oleh *client* dan *bandwidth* yang dibutuhkan. Dalam tabel terdapat beberapa situs yang telah di akses oleh *client* dan juga terdapat waktu yang dibutuhkan untuk membuka *website* yang dituju serta besarnya pemakaian *bandwidth*. Di dalam tabel terdapat perbedaan waktu akses dan *bandwidth* untuk membuka *website*, perbedaan yang terjadi bisa sampai separuh dari hasil membuka *website*.

KESIMPULAN DAN SARAN

Secara umum kebutuhan *bandwidth* sangat perlu diperhatikan, karena semakin besar *bandwidth* yang digunakan, maka semakin besar pula lalu lintas transfer data yang terjadi dalam *internet*, sehingga dapat membuat transfer data ke *client* menjadi lama. Namun tentu saja masih diperlukan beberapa penyempurnaan yang akan dipaparkan dalam pembahasan bab ini.

Kesimpulan

Berdasarkan uraian dan pembahasan analisa dan pengujian yang telah dilakukan, maka dapat di ambil kesimpulan bahwa kebutuhan *bandwidth* sangat besar disaat *client* melakukan *request* data. Kebutuhan *bandwidth* tersebut dapat dikurangi dengan cara mengoptimalkan *cache proxy*, sehingga kebutuhan *bandwidth* tidak terbuang dengan sia-sia disaat *client* melakukan *request* secara bersamaan. Dengan cara mengoptimalkan *cache proxy*, hasil yang diperoleh adalah *client* dapat mengakses *website* yang diminta dengan cepat. Sehingga dapat menghemat biaya untuk setiap penggunaan *internet*. Dan juga dapat

mengurangi kepadatan lalu lintas transfer data yang ada pada *internet*.

Saran

Saran untuk perkembangan lebih lanjut dalam masa mendatang memberikan hak akses bagi setiap user dan dari pemberian hak akses tersebut bisa membedakan dari segi jabatan untuk memmanagement akan kebutuhan *bandwidth*. Sehingga *bandwidth* yang di pakai tidak terlalu besar dan juga dapat menghemat biaya.

DAFTAR PUSTAKA

Chendramata, A. 2009. *Jaringan Informasi Berbasis Open Source*. Jakarta: Direktorat Sistem Informasi Perangkat Lunak dan Konten Direktorat Jendral Aplikasi Telematika Departemen Komunikasi dan Informatika.

Kurusetra 2012. *Optimasi Squid Proxy Server*. [Online] Tersedia : <http://kurusetra.web.id/2012/06/07/optimasisquid-proxy-server/> [7November 2012]

Sembiring, J. 2001. *Jaringan Komputer Berbasis Linux*. Jakarta: Elex Media Komputindo.

Syafrizal, M. 2005. *Pengantar Jaringan Komputer*. Yogyakarta: Andi Offset.

Tajidyakub 2012, *Squid Cache, Tproxy, Mikrotik, Alternatif Untuk Perpaduan Jaringan Sederhana*. [Online] Tersedia : <http://www.tajidyakub.net/2007/08/08/squidcache-tproxy-dan-mikrotikalternatifperpaduan-untuk-jaringan-sederhana/> [20 November 2012]

Wagito. 2005. *Jaringan Komputer, Teori dan Implementasi Berbasis Linux*. Yogyakarta: Gaya Media.

Wahana Komputer. 2006. *Pengelolaan Jaringan Komputer di Linux*. Jakarta: Salemba Infotek.

Wahyudi 2012. Cara Instal dan Optimasi Squid Proxy Server. [Online] Tersedia : <http://wahyudi83.wordpress.com/2012/06/05/cara-instal-dan-optimasi-squid-proxy-server/> [6 November 2012].

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

(3)

IMPLEMENTASI APLIKASI PENGENDALIAN JARAK JAUH KAMERA DSLR VIA BLUETOOTH BERBASIS ANDROID

Abednego Eko Harjiyanto, Guendra Kusuma Wardhana

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Wijaya Kusuma Surabaya Jl. Dukuh Kupang
54 Surabaya, abed_ekoharjiyanto@yahoo.com

ABSTRAK

Dalam dunia fotografi, remote kontrol untuk kamera DSLR (*Digital Single-Lens Reflex*) belum dapat memenuhi kebutuhan para *photographer*. Biasanya remote kontrol tersebut hanya bisa digunakan untuk mengambil gambar saja dan itupun menggunakan kabel. Untuk para pecinta photographi yang suka foto pun sulit jika ingin mengambil foto mereka sendiri.

Remote Control Camera DSLR adalah aplikasi yang dapat mengontrol kamera DSLR yang beroperasi pada ponsel yang berbasis Android secara jarak jauh. Aplikasi dapat mengontrol berbagai merek dan jenis kamera DSLR, untuk melakukan pengaturan pada kamera dan pengambilan gambar dapat dilakukan melalui ponsel Android yang sudah terinstal aplikasi ini. Yang dapat dilakukan oleh aplikasi adalah mengatur *white balance*, *shutter speed*, *ISO*, dan lain – lain.

Dalam penggunaannya aplikasi ini dapat membantu para *photographer* untuk melakukan pengaturan kamera dengan lebih mudah karena semua menu pengaturan sudah *user friendly*. Untuk dapat menggunakan aplikasi ini, pengguna harus menggunakan ponsel pintar berbasis Android dan menggunakan perangkat tambahan yang dipasang pada kamera DSLR. Jika aplikasi sudah terinstall dan perangkat telah terpasang, aplikasi ini siap untuk dioperasikan.

Kata kunci : *Android, kamera DSLR, remote control*

PENDAHULUAN

Latar Belakang

Remote Kontrol (atau beberapa menyebut dengan *Shutter Release Cable*) dalam dunia fotografi sangat berguna untuk mengambil gambar dengan kecepatan sangat rendah. Goncangan yang diakibatkan tekanan pada badan kamera saat tombol melepas (memencet) tombol shutter bisa mengakibatkan gambar menjadi blur. Biasanya *remote control* dipakai saat mengambil foto malam hari atau gambar macro dengan kamera diletakkan di atas tripod atau dudukan statis.

Remote kontrol untuk kamera ini masih menggunakan kabel dan hanya dapat digunakan untuk melakukan pengambilan gambar saja. Untuk melakukan pengaturan pada kamera tetap harus dilakukan secara manual dari kamera. Penggunaan remote kontrol dengan kabel juga memiliki batas jarak yang dekat tergantung panjang kabel tersebut. Remote kontrol untuk *Shutter Release* juga sudah ada yang menggunakan teknologi *Wi-Fi*, tetapi juga hanya bisa digunakan untuk pengambilan gambar saja dan tidak dapat *Live View*.

Dalam Tugas Akhir ini akan dibahas bagaimana cara mengontrol kamera DSLR (*Digital Single-Lens Reflex*) dengan ponsel pintar tanpa menggunakan kabel dan memaksimalkan

fungsi remote kontrol untuk memudahkan kinerja seorang *photographer*. Remote kontrol yang akan dibuat dalam bentuk aplikasi yang berjalan pada ponsel Android dan sebuah perangkat keras yang terpasang pada kamera DSLR. Ponsel yang berbasis Android sering disebut sebagai ponsel pintar yang memungkinkan dapat melakukan segala sesuatu untuk membantu kinerja manusia. Sehingga tidak mustahil bila dapat digunakan untuk mengontrol sebuah kamera DSLR yang tentunya harus melalui sebuah aplikasi. Aplikasi ini akan membantu dalam teknik pengambilan gambar macro karena dapat mengurangi efek getaran pada kamera.

TINJAUAN PUSTAKA

Pengendalian Jarak Jauh

Pada umumnya, pengendali jarak jauh digunakan untuk memberikan perintah dari kejauhan kepada televisi atau barang-barang elektronik lainnya seperti sistem stereo dan pemutar DVD. Remote control untuk perangkat-perangkat ini biasanya berupa benda kecil nirkabel yang dipegang dalam tangan dengan sederetan tombol untuk menyesuaikan berbagai pengaturan, seperti misalnya saluran televisi, nomor trek, dan volume suara. Malah, pada kebanyakan peranti

modern dengan kontrol seperti ini, remote controlnya memiliki segala kontrol fungsi sementara perangkat yang dikendalikan itu sendiri hanya mempunyai sedikit kontrol utama yang mendasar. Kebanyakan remote berkomunikasi dengan perangkatnya masing-masing melalui sinyal- sinyal infra merah dan beberapa saja melalui sinyal radio. Remote control biasanya menggunakan baterai AAA yang kecil atau AA sebagai catu dayanya.

Android

Android adalah sistem operasi berbasis Linux untuk perangkat mobile seperti ponsel pintar dan komputer tablet. Hal ini dikembangkan oleh *Open Handset Alliance* yang dipimpin oleh Google.

Google membeli pengembang awal perangkat lunak Android Inc pada tahun 2005. Pada tahun 2007 diumumkan pembukaan dari distribusi Android dengan *Open Handset Alliance* sebagai pendirinya, sebuah konsorsium dari 86 perangkat keras, perangkat lunak, dan perusahaan telekomunikasi yang ditujukan untuk memajukan standar terbuka perangkat mobile. Google merilis kode Android sebagai open-source di bawah *Apache License*. *Android Open Source Project (AOSP)* bertugas sebagai pemeliharaan dan pengembangan lebih lanjut dari Android.

Android memiliki komunitas pengembang aplikasi ("*apps*") yang besar, sehingga memperluas fungsionalitas dari perangkat. Aplikasi dapat diunduh dari situs pihak ketiga atau melalui toko online seperti *Google Play* (sebelumnya *Android Market*). Pada Februari 2012 ada lebih dari 450.000 aplikasi yang tersedia untuk Android, dan perkiraan jumlah download aplikasi dari *Android Market* pada Desember 2011 melebihi 10 miliar.

Android tercatat sebagai platform ponsel pintar terlaris di seluruh dunia pada pertengahan 2010 oleh *Canalys* dengan lebih dari 300 juta perangkat Android digunakan pada Februari 2012. Menurut Andy Rubin pada Februari 2012 ada lebih dari 850.000 perangkat Android diaktifkan setiap hari.

Akuisisi oleh Google

Google mengakuisisi *Android Inc* pada tanggal 17 Agustus 2005, menjadikan Android Inc sebagai anak perusahaan yang sepenuhnya dimiliki oleh karyawan Google Inc. Pendiri Android Inc, termasuk Andy Rubin, Rich Miner dan Chris White, menjadi karyawan di

perusahaan setelah akuisisi. Tidak banyak yang diketahui tentang Android Inc pada saat akuisisi, tetapi banyak diasumsikan bahwa Google merencanakan untuk memasuki pasar ponsel dengan langkah ini.

Open Handset Alliance

Pada tanggal 5 November 2007, *Open Handset Alliance*, sebuah konsorsium dari beberapa perusahaan yang termasuk Broadcom Corporation, Google, HTC, Intel, LG, Marvell Technology Group, Motorola, Nvidia, Qualcomm, Samsung Electronics, Sprint Nextel, T-Mobile dan Texas Instrumen bergabung. Tujuan dari Open Handset Alliance adalah untuk mengembangkan standar terbuka untuk perangkat mobile. Pada hari yang sama, Open Handset Alliance juga meluncurkan produk pertama mereka, Android, sebuah platform perangkat mobile dibangun berdasarkan Linux kernel versi 2.6.

Android Proyek Open Source

Android Open Source Project (AOSP) dipimpin oleh Google, dan bertugas dengan pemeliharaan dan pengembangan Android. Tujuan dari Proyek Android Open Source untuk menciptakan produk nyata yang memberikan pengalaman lebih bagi pengguna mobile. AOSP juga mengelola *Android Compatibility Program* mendefinisikan "perangkat Android yang kompatibel" sebagai salah satu yang dapat menjalankan aplikasi yang dibuat oleh pengembang pihak ketiga menggunakan SDK Android dan NDK, untuk mencegah tidak kompatibelnya Android. Program kompatibilitas juga bersifat opsional dan bebas biaya, dengan Test Suite Kompatibilitas juga gratis dan open source.

Design

Android terdiri dari sebuah kernel berbasis kernel Linux, dengan middleware, libraries dan API yang ditulis dalam bahasa C dan perangkat lunak aplikasi yang berjalan pada application framework mencakup Java-compatible libraries didasarkan pada *Apache Harmony*. Android menggunakan *Dalvik virtual machine* dengan *just-in-time* untuk menjalankan kompilasi *Dalvik dex-code (Dalvik executable)*, yang biasanya diterjemahkan dari Java bytecode.

Keamanan

Aplikasi Android berjalan pada *sandbox*, sebuah daerah terisolasi dari sistem operasi yang

tidak memiliki akses ke seluruh sumber daya sistem, kecuali izin akses diberikan oleh pengguna saat aplikasi diinstal. Sebelum menginstal aplikasi, *Play Store* menampilkan semua izin yang diperlukan. Sebuah permainan mungkin perlu mengaktifkan getaran, misalnya, tetapi tidak perlu untuk membaca pesan atau mengakses buku telepon. Setelah meninjau hak akses ini, pengguna dapat memutuskan apakah akan memasang aplikasi tersebut.

Android telah dikritik karena memberikan sistem izin tidak efektif. Pada *Permissions Android Demystified*, Felt, Chin, Hanna, Song, dan Wagner mengamati "... Sistem perizinan pada saat instalasi tidak efektif jika pengembang secara rutin meminta izin secara berlebihan. Aplikasi *Overprivileged* mengirim kepada pengguna untuk peringatan izin yang tidak perlu dan meningkatkan dampak dari bug atau kerentanan.. ". Para penulis kemudian melakukan survei aplikasi *overprivileged*, termasuk implementasi referensi Google menggunakan alat *Stowaway*. Dr. Android and Mr. Hide: *Fine-grained security policies on unmodified Android*, Jeon, Micinski, Vaughan, et al. berkomentar pada hak akses yang menyatakan "penyimpangan dari hak istimewa setidaknya meningkatkan ancaman dari kerentanan dan malware. Untuk mengatasi masalah ini, kami menyajikan sebuah sistem baru yang dapat menggantikan hak akses platform yang lebih baik".

Beberapa perusahaan keamanan telah merilis perangkat lunak antivirus untuk perangkat Android, khususnya AVG Technologies, Avast, F-Secure, Kaspersky, McAfee dan Symantec.

Kamera DSLR (Digital Single-Lens Reflex)

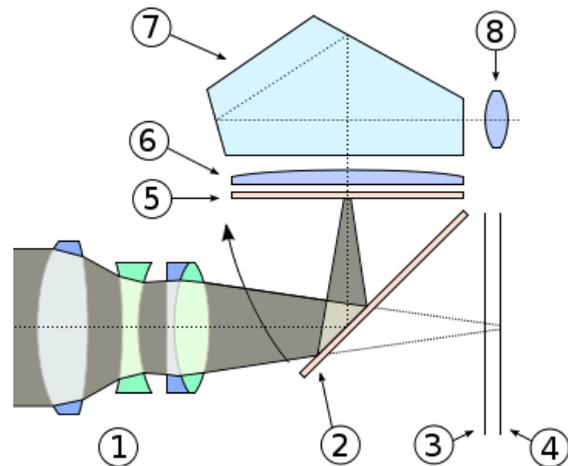
Kamera DSLR adalah kamera digital menggabungkan bagian-bagian dari kamera *Single-Lens Reflex (SLR)* dengan kamera digital untuk menggantikan film fotografi. Kamer DSLR dilengkapi fitur seperti *live preview*, video perekam *High Definition (HD)* dengan *autofocus* deteksi kontras atau integrasi ergonomis seperti tombol *film speed (ISO)*.

Saat ini kamera DSLR secara luas digunakan oleh fotografer masih amatir dan profesional. DSLR saat ini menawarkan berbagai macam lensa yang berdedikasi dan peralatan fotografi lainnya, menggunakan format sensor gambar yang lebih besar, menyediakan berbagai dinamis lebih tinggi dan sinyal terhadap noise.

Resolusi terendah yang dimiliki kamera

digital *SLR (Single Lens Reflex)* adalah 5.1 megapiksel. Seperti halnya pada kamera SLR analog, kamera digital SLR juga memiliki kualitas gambar terbaik karena menggunakan lensa optik dan sistem kendali manual. Selain kendali yang diberikan secara manual, kamera ini juga memiliki sistem kendali otomatis yang dibantu oleh mikro prosesor yang cukup canggih.

Cara kerja kamera DSLR:



Gambar 2.2 Skema kerja *Single-Lens Reflex*

Cahaya yang masuk, setelah melewati jajaran lensa (1) akan dipantulkan oleh cermin yang dipasang pada posisi kemiringan 45 derajat (2) dan diproyeksikan ke *matte focusing screen* (5). Melalui *condensing lens* (6) dan pantulan di dalam *pentaprism* (7), gambar objek kemudian diteruskan ke lensa mata manusia (8). Ketika kita menekan tombol shutter (bidik) maka cermin (2) akan melipat ke arah panah, *focal plane shutter* (3) membuka dan kemudian gambar akan ditangkap oleh sensor (4) dan diteruskan ke prosesor gambar kemudian disimpan di media penyimpanan (MMC dan sebagainya).

Tujuan Penelitian

Tujuan dari penulisan Tugas Akhir ini adalah:

- Dapat merubah pengaturan pada kamera *DSLR* seperti ISO, Arpeture, Shutter speed dan White balance melalui ponsel Android.
- Membantu kinarja fotografer.

IMPLEMENTASI SISTEM

Perangkat Keras (*Hardware*)

Perangkat keras yang dibutuhkan untuk dapat menjalankan aplikasi ini adalah:

- Ponsel pintar Android minimal versi 2.3

(Gingerbread)

- Kamera tipe DSLR (Digital Single Lens Reflector) dengan merek tertentu.
- Sebuah perangkat tambahan berupa arduino (prototype).

Perangkat Lunak (*Software*)

Perangkat lunak yang dibutuhkan untuk dapat menjalankan aplikasi ini adalah sistem operasi Android versi 2.3 (Gingerbread) sampai 4.0.1 (Ice Cream Sandwich).

Tampilan Awal



Gambar 4.1. Tampilan awal

Saat pengguna/user membuka aplikasi ini, akan muncul tampilan awal aplikasi seperti pada gambar 4.1. Pada tampilan tersebut terdapat pilihan pengaturan *shutter speed*, *ISO*, *white balance*, dan lain – lain.

1.1.1. Tampilan Pilihan Aperture



Gambar 4.2. Tampilan pilihan *aperture*

Tampilan seperti pada gambar 4.2 akan muncul bila user/pengguna memilih pengaturan *aperture* pada tampilan awal. Menu ini akan merubah nilai *aperture* pada kamera.

1.1.2. Tampilan Pilihan Shutter Speed



Gambar 4.3. Tampilan pilihan *shutter speed*

Tampilan seperti pada gambar 4.3 akan muncul bila user/pengguna memilih pengaturan *shutter speed* pada tampilan awal. Menu ini akan merubah nilai *shutter speed* pada kamera.

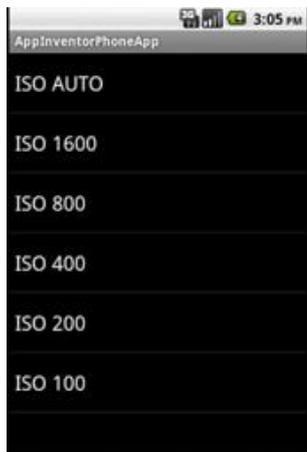
Tampilan Pilihan White Balance



Gambar 4.4. Tampilan pilihan *white balance*

Tampilan seperti pada gambar 4.4 akan muncul bila user/pengguna memilih pengaturan *white balance* pada tampilan awal. Menu ini akan merubah jenis *white balance* pada kamera.

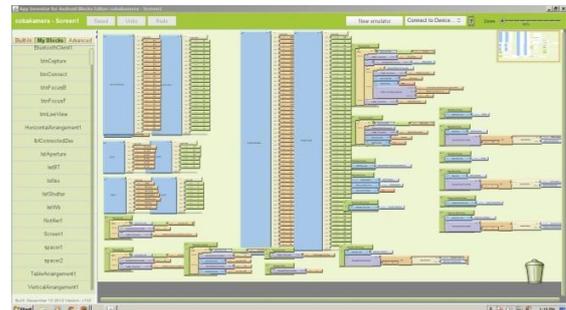
Tampilan Pilihan ISO



Gambar 4.5. Tampilan pilihan *ISO*

Tampilan seperti pada gambar 4.5 akan muncul bila user/pengguna memilih pengaturan *ISO* pada tampilan awal. Menu ini akan merubah nilai *ISO* pada kamera.

Kode Blok Program



Gambar 4.6. Kode blok aplikasi pengendali kamera *DSLR*

Gambar 4.6 adalah kode blok secara keseluruhan dari aplikasi pengendalian jarak jauh kamera - *DSLR* yang dijalankan pada ponsel pintar Android. Keluaran dari kode blok ini berupa file dengan jenis *.apk* yang akan dijalankan pada ponsel Android.

Kode Blok Capture (Menggambil Gambar)

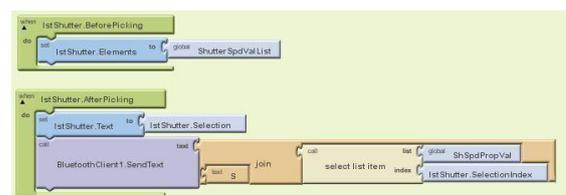


Gambar 4.7. kode blok *capture*

Gambar 4.7. adalah kode blok untuk *capture* atau pengambilan gambar. Saat tombol “Capture” ditekan aplikasi akan mengirim perintah berupa teks “C” melalui Bluetooth kepada Arduino. Pada Arduino perintah “C” akan diterjemahkan oleh kode dibawah ini.

```
case 'C': //Capture!!!  
Serial.println("Capture!");  
Eos.Capture();  
delay(500);  
break;
```

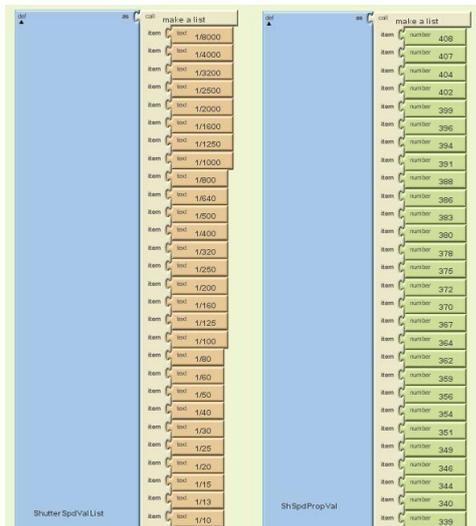
Kode Blok Shutter Speed



Gambar 4.8. Kode blok *shutter speed*

Pada kode gambar 4.8 bekerja saat pengguna memilih menu pengaturan *shutter speed*. Kode blok pertama (gambar 4.8 atas) berfungsi untuk memunculkan *list* atau pilihan nilai *shutter speed* saat pengguna memilih menu pengaturan *shutter speed*. Setelah pilihan nilai ditampilkan pengguna dapat memilih nilai *shutter speed*. Kode blok kedua (gambar 4.8 bawah) berfungsi saat pengguna memilih nilai *shutter speed*, kode blok ini akan mengirim perintah berupa *number* atau *integer* dan perintah berupa teks “S” kepada Arduino melalui Bluetooth. Kode blok *list* nilai *shutter speed* dapat dilihat pada gambar 4.9.

Gambar 4.9 kiri adalah kode blok yang menampilkan pilihan nilai *shutter speed*, sedangkan gambar 4.9 kanan adalah nilai *number* atau *integer* yang dikirim pada Arduino.

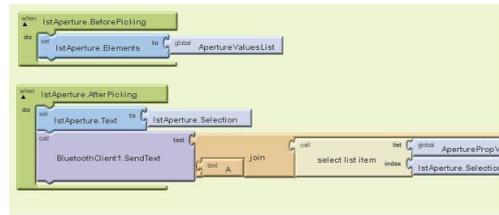


Gambar 4.9. Kode blok pilihan nilai pada *shutter speed*

Perintah teks “S” yang dikirim oleh aplikasi kepada Arduino diterjemahkan pada kode dibawah ini.

```
case 'S': //Set ShutterSpeed
    Eos.SetProperty(EOS_DPC_ShutterSpeed, (uint16_t)readFloatSerial());
    delay(1000);
    Serial.println("Shutter Speed Changed!");
    break;
```

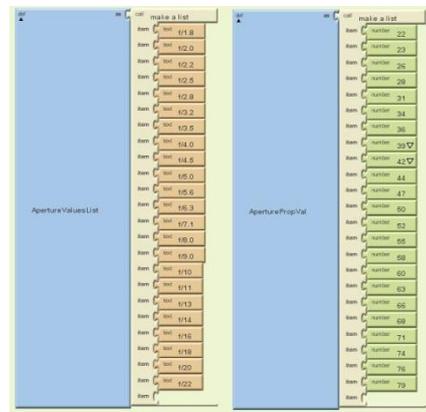
Kode Blok Aperture



Gambar 4.10. Kode blok *Aperture*

Pada kode gambar 4.10 bekerja saat pengguna memilih menu pengaturan *aperture*. Kode blok pertama (gambar 4.10 atas) berfungsi untuk memunculkan *list* atau pilihan nilai *aperture* saat pengguna memilih menu pengaturan *aperture*. Setelah pilihan nilai ditampilkan pengguna dapat memilih nilai *aperture*. Kode blok kedua (gambar 4.10 bawah) berfungsi saat pengguna memilih nilai *aperture*, kode blok ini akan mengirim perintah berupa *number* atau *integer* dan perintah berupa teks “A” kepada Arduino melalui Bluetooth. Kode blok *list* nilai *aperture* dapat dilihat pada gambar 4.11.

Gambar 4.11 kiri adalah kode blok yang menampilkan pilihan nilai *aperture*, sedangkan gambar 4.11 kanan adalah nilai *number* atau *integer* yang dikirim pada Arduino.

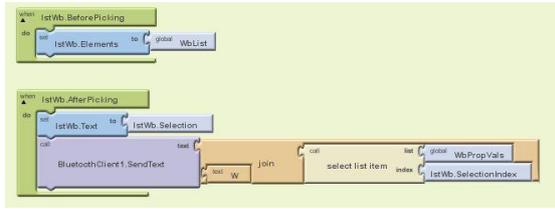


Gambar 4.11. Kode blok *list* nilai *aperture*

Perintah teks “A” yang dikirim oleh aplikasi kepada Arduino diterjemahkan pada kode dibawah ini.

```
case 'A': //Set Aperture
    Eos.SetProperty(EOS_DPC_Aperture, (uint16_t)readFloatSerial());
    delay(1000);
    Serial.println("Aperture value Changed!");
    break;
```

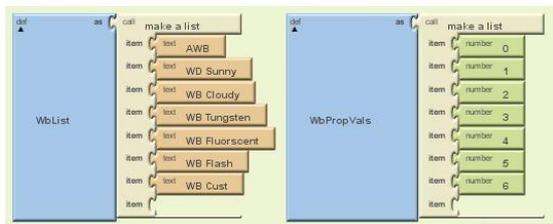
Kode Blok White Balance



Gambar 4.12. Kode blok *white balance*

Pada kode gambar 4.12 bekerja saat pengguna memilih menu pengaturan *white balance*. Kode blok pertama (gambar 4.12 atas) berfungsi untuk memunculkan *list* atau pilihan jenis *white balance* saat pengguna memilih menu pengaturan *white balance*. Setelah pilihan jenis ditampilkan pengguna dapat memilih jenis *white balance*. Kode blok kedua (gambar 4.12 bawah) berfungsi saat pengguna memilih jenis *white balance*, kode blok ini akan mengirim perintah berupa *number* atau *integer* dan perintah berupa teks “W” kepada Arduino melalui Bluetooth. Kode blok *list* jenis *white balance* dapat dilihat pada gambar 4.12.

Gambar 4.13 kiri adalah kode blok yang menampilkan pilihan jenis *white balance*, sedangkan gambar 4.13 kanan adalah nilai *number* atau *integer* yang dikirim pada Arduino.

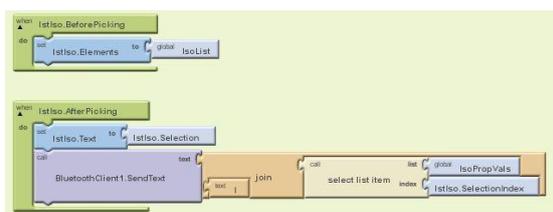


Gambar 4.13. Kode blok *list* jenis *white balance*

Perintah teks “W” yang dikirim oleh aplikasi kepada Arduino diterjemahkan pada kode dibawah ini.

```
case 'W': //Set Whitebalance
  Eos.SetProperty(EOS_DPC_WhiteBalance, (uint16_t)readFloatSerial());
  delay(1000);
  Serial.println("White Balance Changed!");
  break;
```

Source Code ISO



Gambar 4.14. Kode blok *ISO*

Pada kode gambar 4.14 bekerja saat pengguna memilih menu pengaturan *ISO*. Kode blok pertama (gambar 4.14 atas) berfungsi untuk memunculkan *list* atau pilihan nilai *ISO* saat pengguna memilih menu pengaturan *ISO*. Setelah pilihan jenis ditampilkan pengguna dapat memilih nilai *ISO*. Kode blok kedua (gambar 4.14 bawah) berfungsi saat pengguna memilih nilai *ISO*, kode blok ini akan mengirim perintah berupa *number* atau *integer* dan perintah berupa teks “I” kepada Arduino melalui Bluetooth. Kode blok *list* nilai *ISO* dapat dilihat pada gambar 4.15.

Gambar 4.15 kiri adalah kode blok yang menampilkan pilihan jenis *white balance*, sedangkan gambar 4.15 kanan adalah nilai *number* atau *integer* yang dikirim pada Arduino.

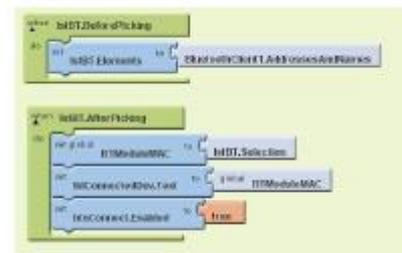


Gambar 4.15. Kode blok *list* nilai *ISO*

Perintah teks “I” yang dikirim oleh aplikasi kepada Arduino diterjemahkan pada kode dibawah ini.

```
case 'I': //Set Iso
  Eos.SetProperty(EOS_DPC_Iso, (uint16_t)readFloatSerial());
  delay(1000);
  Serial.println("ISO Changed!");
  break;
```

Source Code Mencari Perangkat Bluetooth



Gambar 4.16. Kode blok untuk mencari perangkat Bluetooth

Kode pada gambar 4.16 berfungsi untuk mencari dan menampilkan perangkat Bluetooth yang tersambung pada kamera. Untuk dapat terkoneksi harus “pairing” terlebih dahulu antara ponsel Android dengan modul Bluetooth.

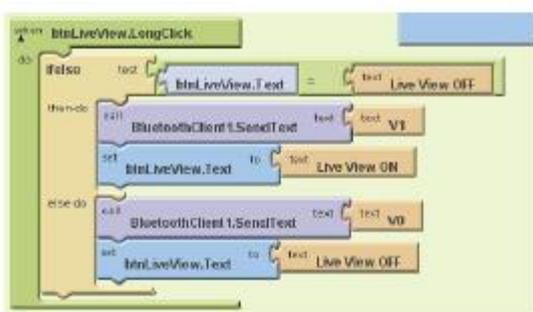
Source Code Koneksi Bluetooth



Gambar 4.17. Kode blok untuk koneksi Bluetooth

Kode pada gambar 4.17 berfungsi untuk koneksi antara ponsel Android dengan perangkat yang tersambung pada kamera. Untuk memutus jaringan Bluetooth antara ponsel Android dan mikro-kontroler pengguna harus menahan tombol koneksi Bluetooth.

Source Code Live View



Gambar 4.18. Kode blok untuk live view

Kode pada gambar 4.18 berfungsi untuk menampilkan live view atau tampilan gambar yang ditangkap oleh kamera secara streaming.

HASIL DAN PEMBAHASAN

Perangkat Uji Coba

Perangkat Keras

Dalam uji coba yang dilakukan adalah menggunakan perangkat sebagai berikut:

- Kamera DSLR Canon EOS 650D
- Arduino Meda ADK
- Bluetooth modul HC-06
- HTC Evo 3D

Perangkat Lunak

Perangkat lunak yang digunakan dalam uji coba adalah sebagai berikut:

- Sistem Operasi Android versi 4.0.1 (Ice Cream Sandwich)

- Arduino IDE versi 0.0.22, Arduino IDE versi 0.0.23, Arduino IDE versi 1.0.1, Arduino IDE versi 1.0.3
- Sistem Operasi Windows 7 Ultimate x64

Uji Coba

Uji coba pertama dilakukan menggunakan kamera Canon EOS 650D dan menggunakan Arduino IDE versi 1.0.1, tetapi terjadi error saat *compiling* kode yang akan di *upload* kedalam Arduino Mega ADK seperti pada gambar 5.1.



Gambar 5.1 Peringatan *Error Compiling*

Dibawah ini adalah kode yang terdapat pada YanisEOS (1).pde yang di *upload* pada Arduino Mega ADK.

```
#include <inttypes.h>
#include <avr/pgmspace.h>

// #include <Spi.h>
#include <Max3421e.h>
#include <Max3421e_constants.h>
#include <Max_LCD.h>
#include <Usb.h>

#include <ptp.h>
#include <canoneos.h>

#define DEV_ADDR 1

// Canon EOS 400D
#define DATA_IN_EP 1
#define DATA_OUT_EP 2
#define INTERRUPT_EP 3

#define CONFIG_NUM 1
#define SerPri Serial.print
#define SerPrIn Serial.println
#define SerAva Serial.available
#define SerRea Serial.read
uint16_t x;
class CamStateHandlers : public PTPStateHandlers
{
  bool stateConnected;
public:
  CamStateHandlers() : stateConnected(false) {};
  virtual void OnDeviceDisconnectedState(PTP *ptp);
  virtual void OnDeviceInitializedState(PTP *ptp);
} CamStates;
```

```
CanonEOS Eos(DEV_ADDR, DATA_IN_EP,
DATA_OUT_EP, INTERRUPT_EP,
CONFIG_NUM, &CamStates);
void
CamStateHandlers::OnDeviceDisconnectedState(PTP
*ptp)
{
if (stateConnected)
{
stateConnected = false;
Notify(PSTR("Camera disconnected\r\n"));
}
}

void
CamStateHandlers::OnDeviceInitializedState(PTP
*ptp)
{
if (!stateConnected)
stateConnected = true;
while(stateConnected){
//Serial.println("Reading");
//delay(1000);
readSerialCommand();
}
// if (rc != PTP_RC_OK)
// Message(PSTR("Error: "), rc);
delay(5000);
}

void setup()
{
Serial.begin( 115200 );
Serial.println("Start");
Eos.Setup();
delay( 200 );
}
void loop()
{
Eos.Task();
}
void readSerialCommand() {
char queryType;
if (SerAva()) {
queryType = SerRea();
switch (queryType) {
case 'C': //Capture!!!
Serial.println("Capture!");
Eos.Capture();
delay(500);
break;
case 'O': //ViewFinder Output. 1 = LCD. 2 = AV.
//Eos.SetDevicePropValue(EOS_DPC_CameraOut
put, (uint16_t)readFloatSerial());
delay(1000);
break;
case 'V': //Liveview ON/OFF
if((uint16_t)readFloatSerial() == 0){
Eos.SwitchLiveView(false);
Serial.println("Live View OFF!");
}
else {
Eos.SwitchLiveView(true);
Serial.println("Live View ON!");
}
delay(1000);
break;
case 'I': //Set Iso
Eos.SetProperty(EOS_DPC_Iso,
(uint16_t)readFloatSerial());
delay(1000);
Serial.println("ISO Changed!");
break;
case 'S': //Set ShutterSpeed
Eos.SetProperty(EOS_DPC_ShutterSpeed,
(uint16_t)readFloatSerial());
delay(1000);
Serial.println("Shutter Speed Changed!");
break;
case 'W': //Set Whitebalance
Eos.SetProperty(EOS_DPC_WhiteBalance,
(uint16_t)readFloatSerial());
delay(1000);
Serial.println("White Balance Changed!");
break;
case 'A': //Set Aperture
Eos.SetProperty(EOS_DPC_Aperture,
(uint16_t)readFloatSerial());
delay(1000);
Serial.println("Aperture value Changed!");
break;
case 'F': //MoveFocus
Eos.MoveFocus(3);
break;
case 'B': //MoveFocus
Eos.MoveFocus(0x8003);
break;
}
}
}

float readFloatSerial() {
byte index = 0;
byte timeout = 0;
char data[128] = "";
do {
if (SerAva() == 0) {
delay(10);
timeout++;
}
else {
data[index] = SerRea();
```

```
timeout = 0;
index++;
}
}
while ((data[constrain(index-1, 0, 128)] != ';') &&
(timeout < 5) && (index < 128));
return atof(data);
}
```

Ternyata ada beberapa library yang bermasalah dan tidak *support* dengan Arduino Mega ADK sehingga harus menunggu *update* terbaru library yang akan dipakai. Library tersebut adalah :

- Arduino Camera Control yang berisi library perintah – perintah dan kode dasar yang digunakan dalam kamera DSRL.
- USB Host Shield versi 2.0 yang *support* dengan USB pada Arduino Mega ADK.

Setelah mendapat update library yang baru, ada beberapa kode pada YanisEOS (1).pde, perubahannya adalah penambahan kode dibawah ini.

```
void setup();
void loop();
```

Saat dilakukan *upload* ulang pada Arduino Mega ADK tetap terjadi *error* seperti pada gambar 5.1. Lalu uji coba dilakukan kembali setelah keluar Arduino IDE 1.0.3, tetapi hasilnya sama, begitu juga dengan Arduino IDE versi 0.0.23 dan 0.0.22 seperti yang disarankan saat bertanya pada forum resmi Arduino. Untuk dapat menyelesaikan aplikasi ini sementara masih menunggu *update* library dan Arduino IDE yang baru serta masih menunggu saran dan jalan keluar sesuai pertanyaan yang penulis kirim pada forum resmi Arduino.

Kesimpulan

Berdasarkan uji coba yang dilakukan, maka dapat diambil kesimpulan:

1. Banyak diperlukan *library* dan perangkat yang memadai untuk dapat menyempurnakan aplikasi ini.

Saran

- Untuk dapat menyempurnakan aplikasi ini minimal dibutuhkan 2 sampai 3 orang untuk dapat dibagi tugas antara pembuatan aplikasi di Android dan kode untuk mikro-kontroler yang benar – benar berpengalaman dalam bidang mikro-kontroler dan bahasa C++.

- Harus dapat menyediakan anggaran biaya yang lebih untuk suatu perancangan perangkat dan aplikasi baru.
- Harus selalu terkoneksi dengan dunia informatika yang luas agar dapat cepat mendapat update terbaru tentang materi dan library yang diperlukan.

Daftar Pustaka

- Android.[http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system)). Diakses pada tanggal 14 April 2012.
- Arduino.<http://en.wikipedia.org/wiki/Arduino>. Diakses pada tanggal 14 April 2012.
- Arduino.<http://arduino.cc/hu/Guide/Introduction>. Diakses pada tanggal 14 April 2012.
- BluetoothBasics.<http://www.bluetooth.com/Pages/Basics.aspx>. Diakses pada tanggal 14 April 2012.

(4)
**IMPLEMENTASI WEB SERVICE BERBASIS SISTEM INFORMASI GEOGRAFIS
UNTUK PENCARIAN LOKASI RUMAH MAKAN**

Wira Lamba¹, Nia Saurina²

Program Studi Teknik Informatika, Fakultas Teknik, Universitas Wijaya Kusuma Surabaya
wiralamba@gmail.com

ABSTRAK

Pencarian lokasi rumah makan selama ini masih dilakukan secara manual. Pencarian lokasi rumah makan menggunakan peta dalam bentuk *hard copy*. Hal tersebut tentunya sangat menyulitkan khususnya bagi orang pendatang atau orang luar daerah dalam mencari lokasi rumah makan.

Pada penelitian Tugas Akhir ini akan diimplementasikan *web service* berbasis sistem informasi geografis (SIG) untuk pencarian lokasi rumah makan pada daerah Surabaya yang bertujuan untuk memberikan kemudahan dalam mencari lokasi rumah makan. *Web service* digunakan sebagai suatu fasilitas untuk menyediakan layanan data lokasi rumah makan dalam bentuk informasi peta digital menggunakan peta dasar *googlemaps*. Agar sistem yang akan dibuat dapat di implementasikan, maka dibutuhkan beberapa perangkat, aplikasi pendukung dan data rumah makan di Surabaya. Perangkat yang digunakan salah satunya yaitu Android. Sedangkan untuk aplikasinya adalah Android SDK untuk pengembangan aplikasi *web service client* pada *smartphone* Android.

Dengan adanya sistem ini, maka pencarian lokasi rumah makan tidak perlu dilakukan secara manual dan juga tidak menyulitkan untuk orang pendatang atau orang luar daerah dalam mencari lokasi rumah makan.

Kata kunci : *Web Service, Sistem Informasi Geografis, Android, Lokasi Rumah Makan*

PENDAHULUAN

Latar Belakang

Pencarian lokasi rumah makan selama ini masih dilakukan secara manual yaitu dengan cara melihat pada peta dalam bentuk *hard copy* maupun dengan bertanya secara langsung kepada orang lain. Namun data lokasi yang di dapat dari *hard copy* tersebut maupun bertanya jarang kita temui dan sangat memakan waktu. Hal tersebut tentunya sangat menyulitkan khususnya bagi orang pendatang atau orang luar daerah dalam mencari lokasi rumah makan.

Berkembangnya teknologi dalam kurunwaktu belakangan ini membuat setiap orang dengan mudah mengakses informasi-informasi dengan cepat dan akurat. Pencarian lokasi rumah makan yang dulunya dilakukan secara manual kini mulai dilakukan dengan cara yang lebih modern. Banyak cara yang dapat dilakukan dalam mencari lokasi rumah makan, salah satunya dalam bentuk peta digital dimana akan ditampilkan data dalam suatu informasi berbasis geografis.

Metode yang lebih modern untuk mencari posisi lokasi rumah makan yaitu dengan aplikasi *web service* berbasis sistem informasi geografis. Aplikasi ini menampilkan lokasi rumah

makan dalam bentuk peta digital. *Web service* digunakan sebagai suatu fasilitas yang disediakan oleh suatu web site untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (*service*) yang disediakan oleh suatu sistem yang menyediakan *web service*.

Agar sistem yang akan dibuat dapat diimplementasikan, maka dibuat aplikasi *web service* berbasis sistem informasi geografis pada *smartphone* Android sebagai *web service* klien atau sebagai pengguna aplikasi pencarian lokasi rumah makan.

Oleh karena itu pada Tugas Akhir ini akan dibahas tentang Implementasi *Web Service* Berbasis Sistem Informasi Geografis Untuk Pencarian Lokasi Rumah Makan. Lokasi rumah makan yang terdeteksi akan diberikan label berupa informasi atau no telepon.

Tujuan

Tujuan dari pembuatan Tugas Akhir Implementasi *Web Service* Berbasis Sistem Informasi Geografis Untuk Pencarian Lokasi Rumah Makan adalah sebagai berikut :

1. Mengetahui penggunaan dan cara kerja GPS untuk penentuan lokasi.

2. Untuk mengintegrasikan google map dan GPS di Android sebagai sebuah sistem berbasis GIS.
3. Mampu mengimplementasikan web service untuk pencarian lokasi rumah makan.
4. Mengetahui pengembangan aplikasi Android untuk web service.

TINJAUAN PUSTAKA

Web Service

Perkembangan internet ternyata mampu mempengaruhi proses evolusi perkembangan sistem terdistribusi (*distributed sistem*), dari sistem yang berdasarkan pada *platform proprietary* (DCOM, CORBA, RMI) menuju *platform* yang lebih terbuka yaitu web. Momentum perkembangan ini diawali dengan kehadiran XML (*eXtensible Markup Language*) untuk mendukung pertukaran data berbasis web secara lebih fleksibel. Keberadaan standard XML kemudian ikut mendorong penetapan standard protokol komunikasi untuk mendukung mekanisme transfer data pada sistem terdistribusi melalui jaringan internet yang dikenal sebagai *Simple Object Access Protocol* (SOAP) pada tahun 1998. Pada akhirnya kedua teknologi ini (XML dan SOAP) kemudian mendasari perkembangan arsitektur teknologi yang memungkinkan untuk menerapkan konsep integrasi data pada jaringan internet berbasis web, yang kemudian dikenal sebagai Web Service.

World Wide Web Consortium (W3C) mendefinisikan web service sebagai “ sebuah software aplikasi yang dapat teridentifikasi oleh URI dan memiliki interface yang didefinisikan, dideskripsikan, dan dimengerti oleh XML dan juga mendukung interaksi langsung dengan *software* aplikasi yang lain dengan menggunakan message berbasis XML melalui protokol internet.

Metode REST

Disamping metode SOAP, terdapat metode lain yang berorientasi pada sumberdaya informasi (resource) dalam Web Services. Konsep Web Services dengan metode yang diberi nama REpresentational State Transfer (REST). Konsep REST adalah konsep untuk mengakses suatu resource (sumber informasi) serta metode yang digunakan untuk melakukan pertukaran resource dari klien ke server. Metode REST didasari oleh empat prinsip utama teknologi, yaitu:

1. Uniform Resource Identifier (URI)

REST Web Services mencari sekumpulan sumberdaya yang mengidentifikasi interaksi antar klien.

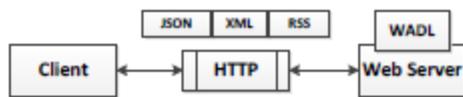
2. Uniform interface
Sumberdaya yang dimanipulasi CRUD (Create, Read, Update, Delete) menggunakan operasi PUT, GET, POST, dan DELETE.
3. Self-descriptive messages
Sumberdaya informasi tidak terikat, sehingga dapat mengakses berbagai format konten (HTML, XML, PDF, JPEG, Plain text dan lainnya).
4. Stateful interactions through hyperlinks

Setiap interaksi dengan suatu sumberdaya bersifat stateless, yaitu request messages tergantung jenis kontennya.

Metode REST Web Services dianggap sederhana karena menggunakan format standar yang umum seperti HTTP, HTML, XML, URI, MIME. Penerapan REST Web Services sama dengan membangun web site dinamis, dengan kehandalan yang diharapkan dalam melakukan uji coba cukup melalui aplikasi Web Browser tanpa membutuhkan software khusus. Namun jika diperlukan untuk proses pengambilan data, maka konten hasil eksekusi Web Services berupa teks dapat digunakan sebagai data yang dapat diolah melalui pembentukan struktur.

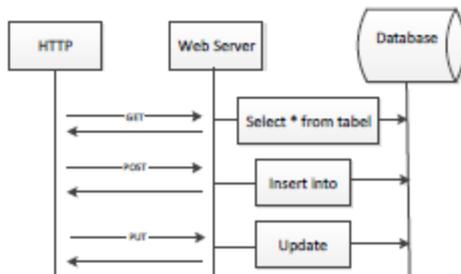
Data dalam berbagai format teks, seperti XML atau HTML. Hal ini dapat dilakukan dalam kode program yang menggunakan utilitas komunikasi data melalui koneksi socket protokol HTTP. Umumnya utilitas ini tersedia dalam pustaka komunikasi pada beberapa bahasa pemrograman seperti Java, VisualBasic, Delphi, PHP, ASP, maupun JSP. Pustaka komunikasi yang tersedia pada beberapa bahasa pemrograman tidak seluruhnya mampu untuk melewati parameter URI dengan metode POST, apalagi bila Web Services yang dituju berada pada protokol HTTPs. Hanya beberapa bahasa pemrograman yang telah menyediakan pustaka komunikasi data dengan kemampuan yang optimal. Sebut saja salah satunya adalah pustaka komunikasi data yang dimiliki PHP, meskipun PHP sendiri menyediakan pustaka SOAP namun dengan konsep Open Source pustaka lain yang mendukung REST juga tersedia dengan nama Klien URL (CURL). CURL bukan satu-satunya pustaka komunikasi data di dalam PHP, masih banyak lagi pustaka lain yang menyediakan kemampuan untuk komunikasi data. Namun pada umumnya CURL sudah terbukti dapat digunakan untuk memenuhi proses komunikasi data di beberapa protokol, seperti HTTP, HTTPS, FTP, GOPHER, TELNET, DICT, FILE, dan

LDAP. Dengan demikian penggunaan pustaka fungsi komunikasi data seperti PHP-CURL dapat mewujudkan mekanisme REST Web Services. Namun untuk menggunakan fungsi-fungsi yang dipublikasikan melalui Web Services dengan metode REST, bisa dilakukan melalui pustaka fungsi komunikasi data apa saja yang tersedia pada masing-masing bahasa pemrograman. Karena pada dasarnya metode REST tidak memperlakukan utilitas apa yang digunakan untuk berkomunikasi, melainkan konten apa yang dapat diproses melalui HTTP atau HTTPS. Secara eksplisit proses interaksi antara REST-Klien dan REST-Server pada metode REST dapat terlihat dalam gambar 1.



Gambar 1. Arsitektur REST Web Service

Berdasarkan gambar arsitektur REST maka dijelaskan mekanisme transfer data pada metode rest sesuai gambar dibawah ini.



Gambar 2. Mekanisme REST Web Service

Dengan demikian REST-Klien hanya perlu menyertakan parameter berupa nama fungsi yang akan diproses dan nilai parameter yang diminta secara langsung dalam bentuk URI. Pengiriman URI ini dapat dilakukan dengan dua metode umum, yaitu POST dan GET. Perbedaannya kalo POST dilakukan secara implisit pada URL, tapi GET dilakukan secara eksplisit pada URL. Melalui metode ini REST-Server cukup membaca metode yang dilewatkan kedalam URI, kemudian menterjemahkan seluruh isi parameter berupa nama umum suatu fungsi yang dipetakan ke dalam fungsi sebenarnya yang diproses berdasarkan nilai parameter permintaan.

GIS (Geographic Information System)

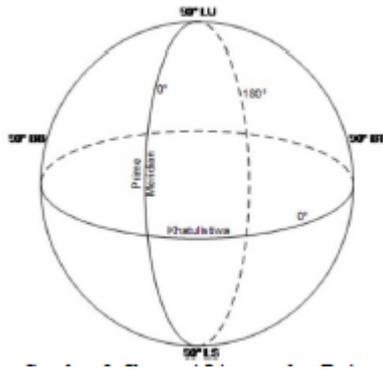
Gis atau yang umum dikenal sebagai (Geographic Information Sytem) merupakan sistem yang berbasis komputer yang digunakan untuk menyimpan dan memanipulasi informasi – informasi geografis. GIS (Geographic Information Sytem) dirancang untuk mengumpulkan, menyimpan, menganalisis objek dan kejadian dimana letak geografis merupakan karakteristik yang penting dan kritis untuk dianalisis. Dengan demikian GIS merupakan sistem komputer yang memiliki empat kemampuan berikut dalam menangani data yang bereferensi geografis (a) masukan, (b) penyimpanan dan pemanggilan data, (c) keluaran (d) analisa data

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.

Sistem Koordinat Geografis

Sistem koordinat geografis menjelaskan keberadaan setiap titik di permukaan bumi dengan koordinat lintang dan bujur. Lintang adalah sudut yang dibuat dari pusat bumi terhadap garis khatulistiwa, dimulai dari 0° di khatulistiwa hingga 90° di kutub utara atau selatan. Bujur adalah sudut yang dibuat dari pusat bumi terhadap garis prime meridian, dari 0° pada Prime Meridian hingga kurang dari 180° belahan bumi timur.



Gambar 3. Ilustrasi Lintang dan Bujur

A. Konversi Jarak dan Koordinat

Panjang lingkaran meridional (utara-selatan) 40.007,86 km, dan panjang lingkaran ekuatorial (khatulistiwa) adalah 40.075,02 km. Dengan perhitungan sederhana, maka didapatkan :

Jarak yang diwakili 1° bujur = $40.075,02 / 360 = 111,3195$ km

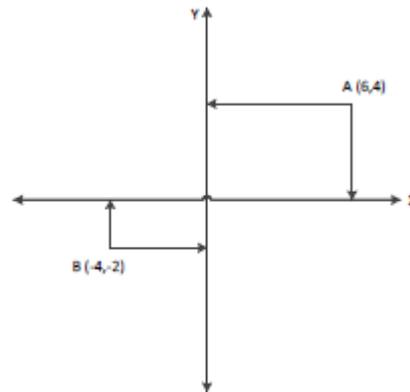
Jarak yang diwakili 1° lintang = $40.007,86 / 360 = 111,1329$ km

Jarak yang diwakili oleh derajat lintang selalu sama untuk derajat bujur berapapun. Sedangkan jarak yang diwakili oleh satu derajat bujur tidak sama pada derajat lintang yang berbeda. Misalkan untuk lintang 0° (khatulistiwa), satu derajat bujur mewakili 111,3195 km. Namun untuk lintang 60°, dimana panjang lingkaran bujur menjadi lebih pendek, satu derajat bujur tidak mewakili 111,3195 km. Rumusan untuk menghitung jarak yang diwakili oleh satu derajat bujur adalah :

Jarak = $111,3195 \times \cos(y)$, dengan y = derajat lintang dimana lingkaran bujur itu berada. Contoh untuk lintang 60°, maka jarak yang diwakili oleh setiap derajat bujur adalah : $111,3195 \times \cos(60) = 55,65975$ km.

B. Rumus Phytagoras

Rumus phytagoras digunakan untuk pencarian jarak antara dua titik dengan menganggap bidang di antara dua titik tersebut sebagai permukaan datar. Contohnya, dasar penentuan tempat titik dua garis lurus yang letaknya saling tegak lurus, dua garis lurus mana dinamakan salib sumbu.



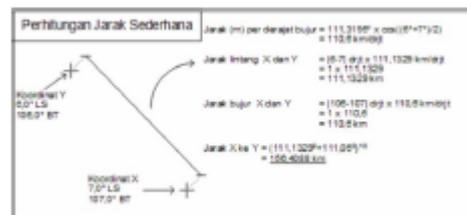
Gambar 4. Menghitung Jarak Dua Garis Lurus

Untuk menghitung jarak antara titik dapat digunakan rumus pythagoras;

$$\begin{aligned}
 d^2 &= dy^2 + dx^2 \\
 &= (x_2 - x_1)^2 + (y_2 - y_1)^2 \\
 d &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \\
 &= \sqrt{(6 - (-4))^2 + (4 - (-2))^2} \\
 &= \sqrt{(10)^2 + (6)^2} \\
 &= \sqrt{136} = 11,66 \text{ (jarak antara titik A dan titik B)}
 \end{aligned}$$

Menghitung jarak antara dua koordinat di bumi relatif sulit jika ingin menghitung secara detail, karena faktor yang harus dipertimbangkan adalah kurva yang terbentuk akibat bentuk bumi yang bulat, serta tidak konsistennya jarak yang diwakili oleh setiap derajat bujur pada derajat lintang tertentu. Untuk menghitung secara kasar, dapat dilakukan dengan menghitung jarak lintang antara dua titik, kemudian untuk menghitung jarak per derajat bujurnya dapat diambil dari rata-rata lintangnya. Jarak lintang dan bujur kedua titik tersebut kemudian dimasukkan kedalam rumus phytagoras untuk dihitung jaraknya. Gambar dibawah ini adalah contoh perhitungan jarak sederhana.

Gambar 5. Perhitungan Jarak Sederhana



Pengertian Rumah Makan

Rumah mempunyai arti bangunan untuk tempat tinggal; bangunannya (seperti gedung), dan kata makan mempunyai arti memasukkan makanan pokok ke dalam mulut, kemudian mengunyah dan menelannya; memasukkan sesuatu ke dalam mulut dan mengunyah-ngunyahnya. Rumah Makan adalah kedai tempat makan atau tempat yang menjual makanan. (www.kbbi.web.id).

Makanan cepat saji mempunyai arti makanan yang pengolahannya dan penyajiannya dilakukan dengan serba cepat. Rumah makan cepat saji adalah kedai tempat makan atau tempat yang menjual makanan yang pengolahannya dan penyajiannya dilakukan dengan serba cepat, misalnya KFC, Hokben, dan lain-lain. Warung adalah tempat menjual makanan, minuman, kelontong dan sebagainya; kedai; lepau. (www.kbbi.web.id)

Pengertian Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak.

Pengertian 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 mempergunakan kelas pada Java API. Kelas merupakan satu-satunya cara menyatakan bagian eksekusi program. 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.

ANALISA DAN PERANCANGAN

Analisa Masalah

Permasalahan pada tugas akhir ini adalah masih banyak pencarian lokasi rumah makan yang dilakukan secara manual, baik itu melihat pada peta dalam bentuk *hard copy* maupun dengan bertanya secara langsung kepada orang lain. Hal tersebut tentunya sangat menyulitkan khususnya bagi orang pendatang atau orang luar daerah. 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.

A. Kebutuhan Data Fungsional

Sistem yang akan dibangun merupakan sistem untuk menginformasikan lokasi rumah makan yang ada di kota Surabaya dari lokasi terdekat klien berada berdasarkan koordinat yang didapat dari GPS yang sudah ada pada *smartphone* Android. Sehingga dibutuhkan data yang akan digunakan sebagai inputan untuk proses pencarian lokasi Rumah Makan.

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

1. Nama Rumah Makan, adalah nama dari rumah makan tersebut.
2. Alamat Rumah Makan, yaitu penjelasan tentang alamat lokasi atau jalan dari rumah makan tersebut berada.
3. No telepon Rumah Makan, yaitu no telepon dari rumah makan yang dapat dihubungi.
4. Koordinat Rumah Makan, yaitu posisi lokasi rumah makan tersebut berada fungsi dari koordinat ini adalah untuk menandakan titik yang akan ditampilkan pada Map lokasi pencarian Rumah Makan.
5. Kategori Rumah Makan, yaitu yang membedakan rumah makan tersebut.

B. Kebutuhan Data Non-Fungsional

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

1. Kebutuhan *Hardware*
Kebutuhan hardware yang dibutuhkan yaitu,
 1. Laptop
 2. Web Server
 3. Database Server
 4. Router

5. Koneksi internet
 6. *Smartphone* Android
2. Kebutuhan *Software*
Kebutuhan *software* yang dibutuhkan yaitu,
1. Java Development Kit, untuk kompilasi kode-kode program.
 2. Eclipse Juno Service Release 1, sebagai perangkat lunak yang digunakan untuk membangun aplikasi klien pada android.
 3. Android SDK, untuk pengembang aplikasi android.
 4. Android Development Tool, sebagai plugin android pada eclipse.
 5. Android Virtual Device, sebagai emulator untuk menjalankan sistem android.

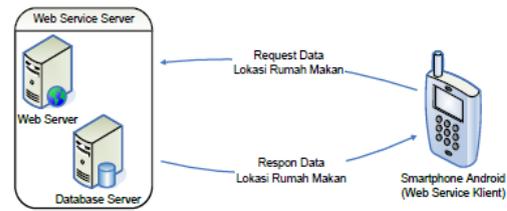
Analisa Sistem

Ada dua jenis web service yang dapat digunakan dalam implementasi yang akan dibuat, yaitu REST dan SOAP. Pada Tugas Akhir ini akan menggunakan web service dengan standar REST (*Representational State Transfer*) karena, REST lebih sederhana dan simpel sehingga mudah untuk dikembangkan dan dipelajari dan tidak bergantung pada *tools*. REST bekerja dengan bernavigasi melalui link-link HTTP untuk melakukan aktivitas tertentu, seakan-akan terjadi perpindahan *state* satu sama lain. Tipe data yang digunakan adalah json dengan menggunakan pustaka komunikasi atau bahasa pemrograman .php dengan menyertakan parameter berupa nama fungsi yang akan diproses yaitu POST untuk menambahkan data rumah makan dan GET untuk mencari rumah makan terdekat sehingga informasi yang diterima lebih mudah dibaca, diparsing dan ditampilkan dalam bentuk maps disisi klien.

Untuk memperjelas analisa sistem maka akan dijelaskan tentang arsitektur dari sistem yang akan dibuat, beserta dengan *bussines process*, desain jaringan, dan flowchart.

A. Arsitektur Sistem

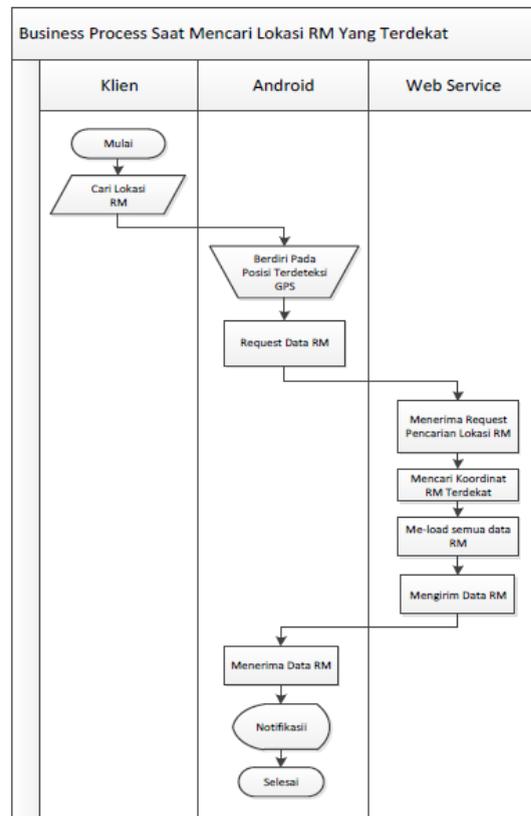
Pada Arsitektur Sistem akan dijelaskan tentang konsep perencanaan dari sistem. Untuk lebih jelasnya bisa dilihat pada desain konsep perencanaan pencarian lokasi rumah makan di bawah ini.



Gambar 6. Desain Konsep Sistem

Gambar 6. bisa dilihat bahwa *smartphone* Android yaitu sebagai *service request* yaitu aplikasi yang bertindak sebagai klien web service. Klien Android me-*request* data lokasi rumah makan terdekat berdasarkan lokasi koordinat yang di dapat dari satellite GPS ke web service. Di dalam web service pada server terdapat web server dan database server. Data *request* kemudian diproses dan web service memberikan respon data lokasi rumah makan. Android menerima data respon yang diberikan web service kemudian ditampilkan dalam bentuk *interface*, sehingga lokasi rumah makan dapat ditampilkan dalam bentuk maps pada klien android.

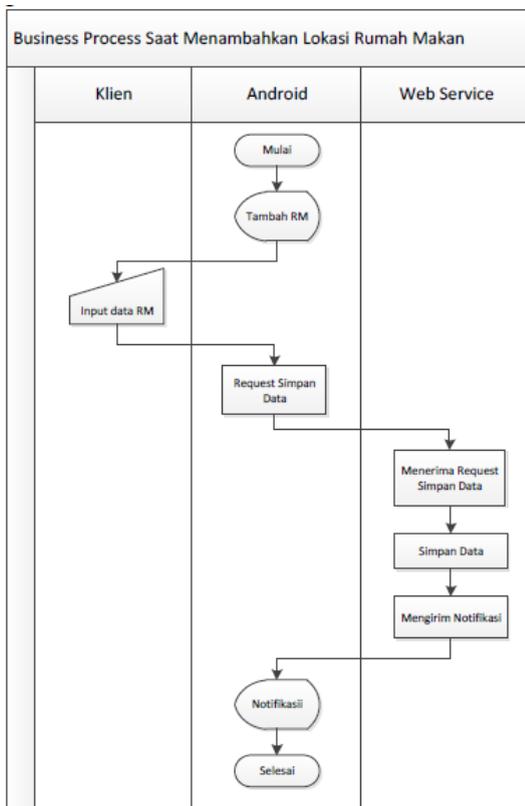
Untuk memperjelas gambaran yang ada, maka akan dibuat rangkaian kerja dalam bentuk *business process* yang dapat dilihat pada gambar 7 di bawah ini.



Gambar 7. Proses Bisnis Klien Mencari Lokasi Rumah Makan

Gambar 7 menjelaskan tentang proses klien mencari lokasi rumah makan terdekat dari posisi klien berada. Dimana Android berdiri pada posisi yang terdeteksi oleh GPS yaitu pada ruang terbuka, klien memanfaatkan GPS yang ada pada *smartphone* android untuk mendapatkan titik koordinat klien. Android kemudian melakukan *request* data lokasi rumah makan kepada web service. Web service menerima *request* dari android, kemudian mencari rumah makan terdekat. Untuk mendapatkan data lokasi rumah makan terdekat web service me-*load* semua data rumah makan. Web service kemudian mengirim data rumah makan yang telah di-*loading* berdasarkan lokasi terdekat dan mengirim data rumah makan tersebut ke Android. Android akan menerima data dari web service, data yang diterima kemudian di proses oleh android untuk menampilkan semua lokasi rumah makan yang terdekat dengan klien yang berupa *output* dalam bentuk tampilan *maps*.

Dari bisnis proses yang telah dijelaskan sebelumnya pada gambar 7 tentang proses pencarian lokasi rumah makan, aplikasi yang akan dibuat juga dapat menambahkan lokasi rumah makan yang dimana akan dijelaskan pada gambar 8.

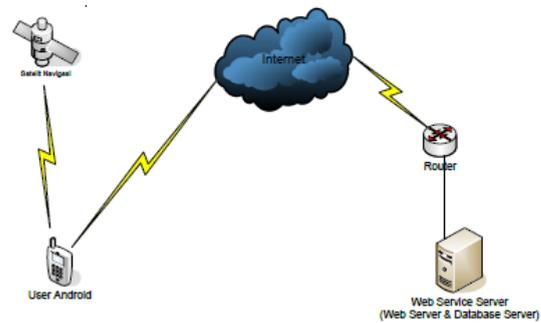


Gambar 8. Proses Bisnis Menambahkan Lokasi Rumah Makan

Gambar 8 menjelaskan tentang alur dari proses menambahkan lokasi rumah makan pada aplikasi Android. Android akan menandai posisi lokasi berdasarkan GPS yang ada pada android tersebut. Setelah menandai posisi lokasi, klien sebagai pengguna aplikasi akan melakukan manual *input* data rumah makan yaitu nama rumah makan, alamat rumah makan, dan keterangan tentang rumah makan. Setelah data dimasukkan, aplikasi Android akan me-*request* proses untuk menyimpan data. Web service akan menerima *request* simpan data rumah makan, dan melakukan proses simpan data. Jika data berhasil dimasukkan atau gagal maka akan dikirim notifikasi dalam bentuk tampilan informasi data berhasil disimpan atau gagal menyimpan data.

B. Analisa Desain Jaringan

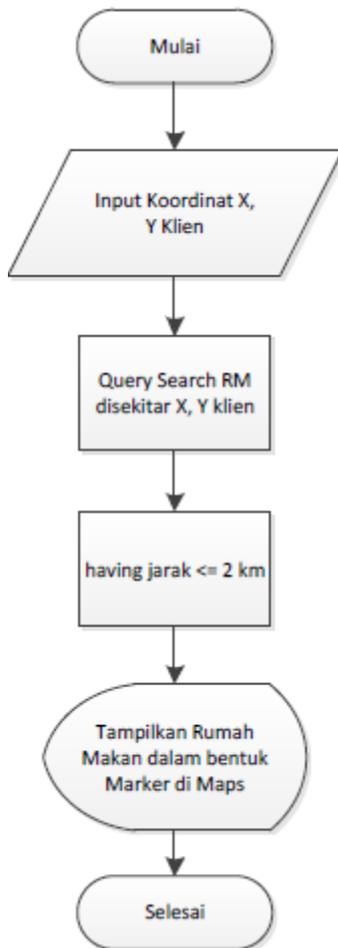
Gambar 9 di bawah ini adalah gambar desain jaringan yang akan digunakan pada tugas akhir berdasarkan konsep sistem yang telah dijelaskan sebelumnya pada analisa arsitektur sistem.



Gambar 9. Desain Jaringan

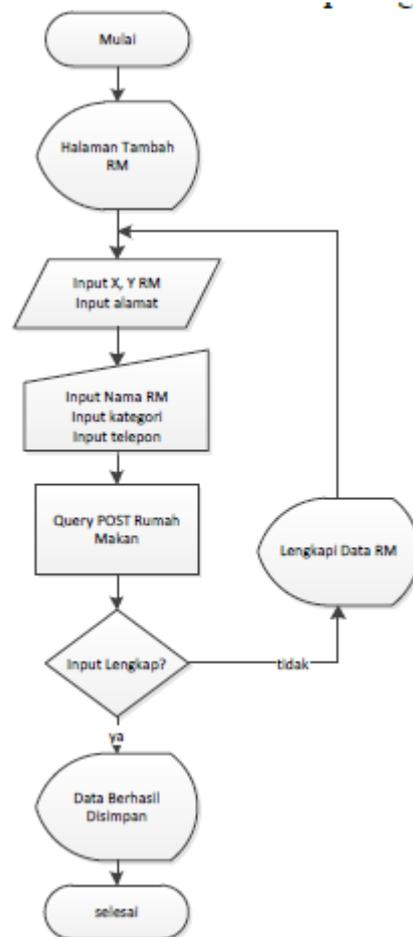
Pada gambar 9 dapat dilihat bahwa klien android terkoneksi dengan satelit navigasi dan internet. Android mempunyai fungsi sebagai GPS dimana dapat terkoneksi dengan satelit navigasi sehingga klien dapat mengerti posisi keberadaannya. Posisi dari klien ini akan dijadikan acuan untuk mencari rumah makan terdekat. Untuk mencari rumah makan terdekat, klien android harus terhubung dengan internet untuk me-*request* data rumah makan yang terdekat dari posisinya saat itu. Sedangkan Pada web service server, di dalamnya terdapat web server dan database server yang terhubung ke router. Sehingga pada saat web service klien yaitu klien android me-*request* data rumah makan ke web service server android harus terhubung ke internet dan web service server akan mengirim data atau merespon data *request* tersebut melalui router yang terhubung ke internet.

C. Flowchart



Gambar 10. Flowchart Pencarian Rumah Makan

Pada Gambar 10 Menjelaskan tentang alur pencarian lokasi rumah makan berdasarkan posisi klien. Input data X, Y klien yang merupakan titik koordinat klien berada akan menjadi acuan untuk mencari rumah makan yang sudah di simpan pada database server yang ada di web service. Setelah data di-inputkan, akan dilakukan proses *query search* rumah makan di sekitar koordinat X, Y klien, dimana di dalam proses ini dilakukan perhitungan jarak berdasarkan titik koordinat X, Y klien dan titik koordinat X, Y rumah makan yang ada pada database di web service server. Proses selanjutnya adalah menambahkan ukuran atau besar jarak antara klien dan rumah makan yaitu besar jaraknya kurang dari sama dengan 2 km. Hasil dari proses, dimana akan muncul data rumah disekitar klien yang ditampilkan dalam bentuk marker rumah makan di peta google.maps



Gambar 11. Flowchart Menambahkan Rumah Makan

Gambar 11 menjelaskan tentang flowchart menambahkan data rumah makan. Pada halaman tambah rumah makan titik koordinat X dan titik koordinat Y diambil dari GPS dan di-inputkan secara otomatis dan input alamat dilakukan secara otomatis yaitu titik koordinat di *convert* menjadi sebuah alamat. Klien melakukan *input* manual data rumah makan, yaitu nama rumah makan, kaategori rumah makan, nomor telepon rumah makan. Data yang telah di-input akan disimpan di database web service server dengan menggunakan perintah *query POST* rumah makan. Jika data yang dimasukan lengkap maka data berhasil disimpan yang dimana akan muncul notifikasi data berhasil disimpan dalam bentuk *interface*, dan jika data yang dimasukan belum lengkap maka muncul notifikasi lengkapi data rumah makan, yaitu klien harus melakukan *input* data ulang.

IMPLEMENTASI SISTEM

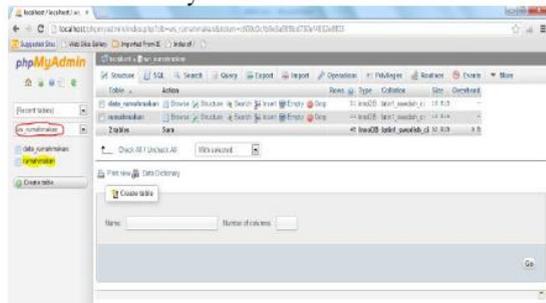
Implementasi merupakan tahap penerapan sistem untuk siap dioperasikan. Tahap ini juga termasuk menuliskan kode program yang dijalankan oleh komputer. Adapun implementasi dari Implementasi Web Service Berbasis Sistem Informasi Geografis Untuk Pencarian lokasi Rumah Makan sebagai berikut :

Konfigurasi Web Service Server

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

A. Create Database Rumah Makan

Pada database mysql akan dibuat sebuah database yang menyimpan data-data rumah makan di Surabaya.



Gambar 12. Database Rumah Makan

Pada Gambar 12, merupakan database yang dibuat dengan nama database ws_rumahmakan (lingkaran merah). Di dalam database ws_rumahmakan terdapat tabel data rumah makan, yang diberi nama tabel rumah makan.

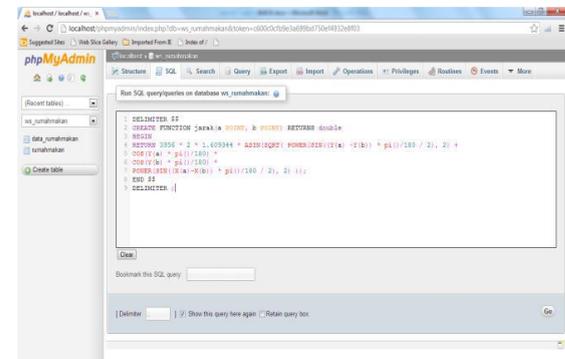
Implementasi tabel rumahmakan berdasarkan struktur tabel yang telah dijelaskan pada Analisa dan Perancangan Sistem, pada database ws_rumahmakan dapat dilihat pada gambar 13.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(11)		No		AUTO_INCREMENT		Change Drop Browse distinct values Primary Unique Index More
2	nama	varchar(20)	latin1_swedish_ci	No				Change Drop Browse distinct values Primary Unique Index More
3	alamat_nm	varchar(200)	latin1_swedish_ci	No				Change Drop Browse distinct values Primary Unique Index More
4	latitude	varchar(20)	latin1_swedish_ci	Yes	NULL			Change Drop Browse distinct values Primary Unique Index More
5	longitude	varchar(20)	latin1_swedish_ci	Yes	NULL			Change Drop Browse distinct values Primary Unique Index More
6	lokasi	geometry		Yes	NULL			Change Drop Browse distinct values Primary Unique Index More
7	telepon	varchar(14)	latin1_swedish_ci	No				Change Drop Browse distinct values Primary Unique Index More
8	kategori	varchar(20)	latin1_swedish_ci	No				Change Drop Browse distinct values Primary Unique Index More

Gambar 13. Tabel Rumah Makan

B. Create Function Jarak

Pada database ws_rumahmakan akan dibuat function jarak, yaitu untuk menghitung jarak antara koordinat X (longitude), Y (latitude) dari posisi klien berada dan koordinat X (longitude), Y (latitude) rumah makan yang telah disimpan di database.



Gambar 14. Function Jarak

Pada gambar 14 adalah function jarak yang dibuat dengan rumus dasar pythagoras yang di dapat dari mysql dalam perhitungan geometry spasial. Dapat dijelaskan angka 3956 adalah radius bumi yang dihitung berdasarkan satuan mil. Untuk merubah jarak dalam bentuk satuan kilometer maka dibuat perhitungan dengan 1.609344 (satuan mil dirubah ke satuan kilometer).

C. Input Data Rumah Makan

Pada implementasi Aplikasi Pencarian Rumah Makan, akan di input-kan data-data rumah makan sesuai dengan tabel rumahmakan yang telah dibuat. Pada implementasi ini dilakukan proses input data dengan contoh 50 data rumah makan di Surabaya. Berikut adalah beberapa data rumah makan yang telah di input-kan pada database ws_rumahmakan.

nama	latitude	longitude	lokasi	telepon	kategori
1 Rumah Makan Bujangrene	-7.269550	112.73982	Jl. Raya Gubeng 44, Surabaya	(031) 503 0263	Rumah Makan
2 Restoran Rian Bakar Cempur Pakel Cakle	-7.267574	112.739553	Jalan Mayor Jenderal Sungsono 206-209 Pakel Cakle	(031) 568 8792	Restoran
3 Restoran Baw Ams	-7.283508	112.73228	Jalan Deller Gatot, Tegayun	(031) 567 8426	Restoran
4 Rumah Makan Kebab Kebab	-7.268482	112.742609	Jl. Ploke Intenno 23-25, Surabaya, 60205	(031) 563 2215	Rumah Makan
5 Rumah Makan Bumbu Jawa	-7.276450	112.746581	Jl. Sichevski 34, Surabaya, 60264	(031) 552 3086	Rumah Makan
6 Rumah Makan Ampara	-7.275444	112.747179	Jl. Unip Surokerto 4, Surabaya, 60271	(031) 531 2342	Rumah Makan
7 Rumah Makan Campur Sari	-7.278849	112.731133	Jl. Raden Ajeng Katen 141, Surabaya, 60264	(031) 568 1909	Rumah Makan
8 Rumah Makan Koki Pak Boo	-7.272889	112.756198	Gubung Anlangga V, Surabaya, 60286	(031) 567 4058	Rumah Makan
9 Rumah Makan Ampol	-7.266459	112.748331	Jl. Walkota Mustang 33-8, Surabaya, 60272	(031) 546 6462	Rumah Makan
10 Restoran Rii	-7.269111	112.748157	Jalan Komandya Besar Pahlu Mohammad Dary 7, Tig	(031) 567 6343	Restoran
11 Rumah Makan Gang Jangkrik	-7.294454	112.711447	Perumahan Permata Mayjen Sungsono I Blok A2/L May	(031) 567 8766	Rumah Makan
12 Rumah Makan Sulawesi	-7.291222	112.763938	Jalan Kumpang Baru Raya	(031) 567 8766	Rumah Makan
13 Rumah Makan Halim	-7.297229	112.758276	Jalan Mayor Jenderal Sungsono 216	(031) 567 4299	Rumah Makan
14 Rumah Makan Kampung Bekas	-7.299114	112.76073	Jl. Hydrant Semplo 18, Surabaya, 60232	(031) 567 4651	Rumah Makan
15 Rumah Makan Chang Swo	-7.262509	112.752471	Komplek RRM 116-168, Nagel Jaya Selatan, Surabaya	(031) 564 3953	Rumah Makan

Gambar 15. Input Data Rumah Makan

Konfigurasi Antarmuka Pada Web Service Klien

Pada sub-bab konfigurasi web service

pada klien, dijelaskan desain antarmuka yang telah direncanakan pada *smartphone* Android, sehingga klien sebagai pengguna lebih mudah dalam pemahaman menu dan fungsi masing-masing modul ataupun *button* yang telah disediakan.

A. Antarmuka Home



Gambar 16. Antarmuka Home

Pada gambar 16 merupakan antarmuka awal pada *smartphone* Android ketika membuka Aplikasi Pencarian Rumah Makan.

Diantara header dan footer pada gambar 16 Antarmuka Home berisi tentang informasi singkat Aplikasi Pencarian Rumah Makan dan beberapa *button* dalam bentuk *images* sesuai dengan fungsinya masing-masing.

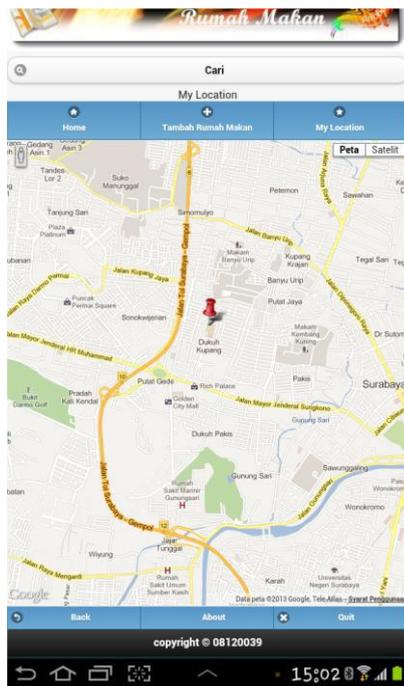
B. Antarmuka Tambah Rumah Makan



Gambar 17. Antarmuka Tambah Rumah Makan

Pada gambar 17 merupakan antarmuka tambah data rumah makan. Data yang dimasukkan berupa nama rumah makan, dan telepon rumah makan. Sedangkan untuk masukkan posisi koordinat rumah makan berupa latitude dan longitude di dapat dari koneksi *device* android dengan GPS dan alamat rumah makan adalah hasil *convert* koordinat dalam bentuk alamat.

C. Antarmuka My Location



Gambar 18. Antarmuka My Location

Antarmuka My Location adalah antarmuka yang menampilkan posisi klien berada, dan posisi rumah makan terdekat dalam bentuk peta digital.

Pada gambar 18 Antarmuka My Location berisi peta digital, peta yang digunakan adalah peta google.maps. Ketika posisi klien pada *smartphone* Android terdeteksi oleh GPS, posisi klien akan ditampilkan pada peta dalam bentuk marker. Dari posisi klien ini akan dijadikan acuan untuk mencari lokasi rumah makan terdekat dengan memberi parameter jarak kurang dari 2 km.

D. Antarmuka Search Rumah Makan



Gambar 19. Antarmuka Search Rumah Makan

Pada Gambar 19 menjelaskan tentang *search* rumah makan. Klien sebagai pengguna aplikasi dapat mencari rumah makan berdasarkan nama rumah makan, kategori rumah makan, atau berdasarkan jarak dari posisi (kurang dari sama dengan 2 km).

E. Antarmuka About



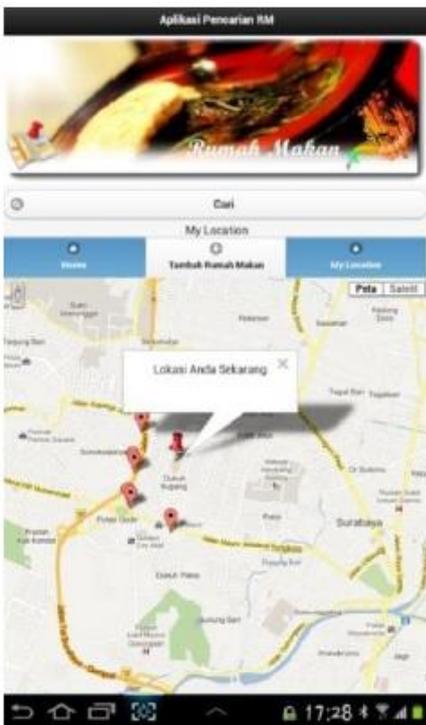
Gambar 20. Antarmuka About

Pada Gambar 20 menjelaskan tentang antarmuka *About*. Antarmuka ini berisi informasi dan penjelasan tentang aplikasi yang dibuat yaitu latar belakang dari pembuatan aplikasi, konsep aplikasi, hingga kesimpulan dari pembuatan aplikasi pencarian lokasi rumah makan.

HASIL DAN PEMBAHASAN

Pada Hasil Dan Pembahasan 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 Analisa dan Perancangan dan Implementasi Sistem.

Pengujian Deteksi Lokasi Klien dan Rumah Makan



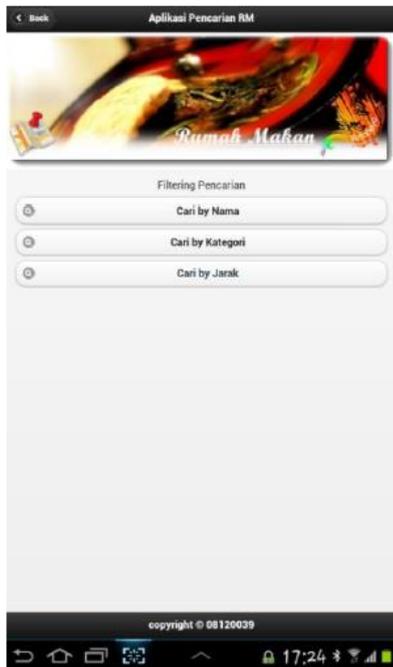
Gambar 21. Uji Lokasi Klien



Gambar 22. Uji Rumah Makan Berdasarkan Lokasi Klien

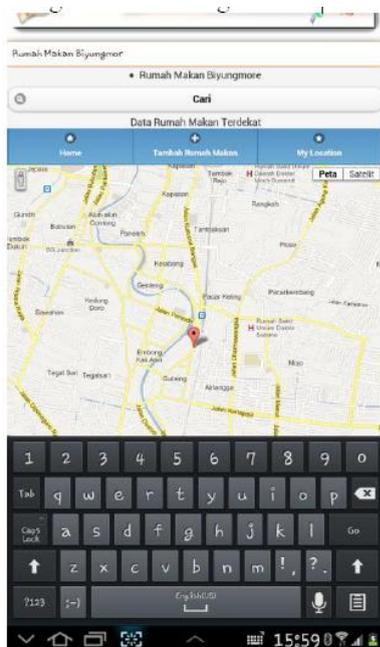
Pada gambar 21 dan 22 bisa dilihat bahwa uji lokasi klien dan rumah makan yang terdekat dari posisi klien terdeteksi. Posisi lokasi klien terdeteksi oleh GPS yang ada pada *smartphone* Android di Dukuh Kupang tepatnya di Universitas Wijaya Kusuma Surabaya. Posisi Koordinat X, Y klien dijadikan acuan untuk mencari lokasi rumah makan terdekat dari klien dengan radius 2 km. Posisi rumah makan yang terdekat dari posisi klien berada ditampilkan dalam bentuk marker yang berbeda dengan posisi klien berada. Ketika salah satu marker rumah makan dipilih, maka muncul informasi tentang rumah makan tersebut dalam bentuk infowindow.

Pengujian Search Rumah Makan



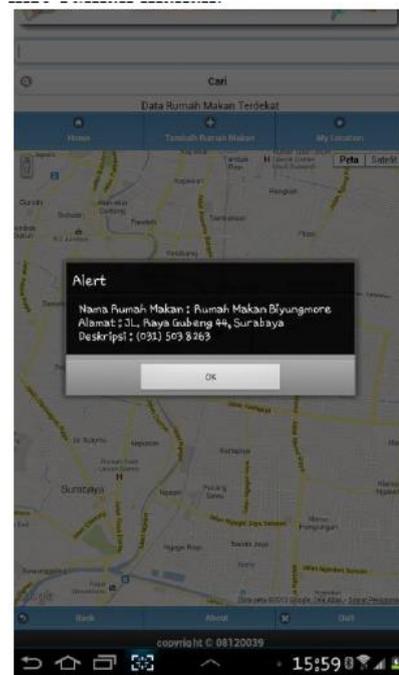
Gambar 23. Uji Search Rumah Makan

Gambar 23 menjelaskan tentang uji search rumah makan, dimana pencarian rumah makan di filtering berdasarkan pencarian nama, pencarian kategori, atau pencarian berdasarkan jarak yaitu jarak kurang dari sama dengan 2 km.



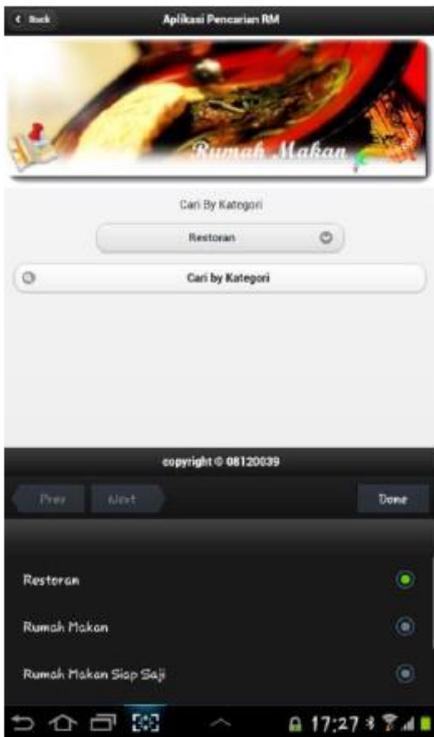
Gambar 24. Uji Search Rumah Makan by Nama

Gambar 24 adalah hasil uji coba search rumah makan by nama. Klien sebagai pengguna aplikasi memasukkan nama rumah makan yang dicari. Contoh di atas adalah mencari lokasi rumah makan dengan nama rumah makan Biyungmore, ketika rumah makan tersebut dipilih (button cari) maka posisi dari rumah makan Biyungmore akan muncul pada peta. Untuk mendapatkan informasi tentang rumah makan yang dicari dapat dilihat pada gambar 25 uji search info rumah makan.



Gambar 25. Uji Search info Rumah Makan

Gambar 25 uji search info rumah makan adalah kelanjutan dari gambar 23 uji search rumah makan by nama. Gambar 25 menjelaskan tentang informasi dari rumah makan yang dicari, ketika rumah makan yang telah dipilih ditampilkan pada peta dalam bentuk marker dan dipilih maka muncul informasi tentang rumah makan tersebut. Contoh di atas adalah informasi tentang rumah makan Biyungmore, yaitu informasi nama rumah makan, alamat, dan no telepon rumah makan.



Gambar 26. Uji Search Rumah Makan by Kategori

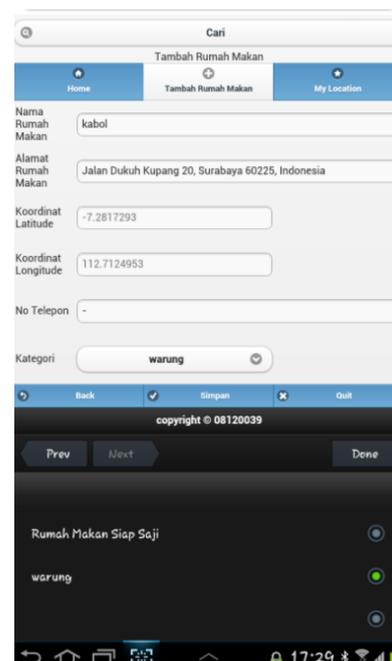
Gambar 26 menjelaskan pencarian rumah makan berdasarkan kategori yang dipilih, contoh pada gambar 26 di atas adalah pencarian rumah makan berdasarkan kategori rumah makan, yaitu kategori restoran. Ketika kategori restoran dipilih kemudian klien melakukan pencarian maka hasil dari pencarian kategori rumah makan dapat dilihat pada gambar 27.



Gambar 27. Data Kategori Rumah Makan

Gambar 27 adalah hasil pencarian rumah makan berdasarkan kategori yang dipilih pada gambar 26. Contoh pada gambar 27 adalah data kategori rumah makan yang dipilih berdasarkan nama kategori restoran.

Pengujian Tambah Data Rumah Makan



Gambar 28. Uji Tambah Data Rumah Makan

Pada Gambar 28 merupakan uji tambah data rumah. Aplikasi yang dibuat dapat menambahkan rumah makan berdasarkan posisinya saat itu. Contoh di atas adalah tambah data rumah makan dengan nama : Kabol, alamat : jalan Dukuh Kupang hasil dari *convert* koordinat ke alamat, koordinat latitude dan koordinat longitude adalah posisi klien berada. Koordinat ini di dapat dari GPS yang ada pada *smartphone* Android yang di inputkan secara langsung (otomatis) pada *text input* dan tidak dapat dirubah nilai dari koordinat.

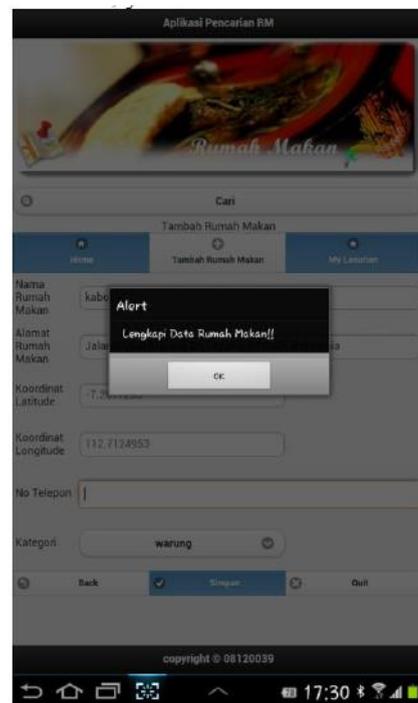


Gambar 29. Uji Sukses Tambah Data Rumah Makan

Gambar 29 adalah gambar uji sukses tambah rumah makan. Ketika klien menambahkan rumah makan dengan mengisi data rumah makan secara lengkap kemudian menyimpan data tersebut pada web service server, maka muncul notifikasi berupa pemberitahuan data berhasil dimasukkan. Pembuktian bahwa data tersebut berhasil dimasukkan pada database web service server dapat dilihat pada gambar 30.

Gambar 30. Uji Sukses Tambah Data Rumah Makan di Database Web Service Server

Gambar 30 menjelaskan tentang uji sukses tambah data pada database. Data rumah makan yang telah dimasukkan seperti yang telah dijelaskan sebelumnya pada gambar 28 uji tambah data rm, yaitu data rumah makan Kabol.



Gambar 31. Uji Gagal Tambah Data Rumah Makan

Pada gambar 31 adalah uji coba gagal tambah data RM. Dalam uji coba ini data yang dimasukkan tidak lengkap (no telepon tidak dimasukkan). Ketika data akan disimpan, muncul pemberitahuan dalam bentuk alert lengkapi data rumah makan.

PENUTUP

Kesimpulan

Simpulan hasil penelitian yang telah dilakukan pada BAB I sampai pada BAB V, bahwa Aplikasi Pencarian Lokasi Rumah Makan yang sudah dibangun memiliki beberapa hasil penelitian yang bisa disimpulkan, diantaranya adalah:

1. Mengetahui penggunaan dan cara kerja GPS untuk penentuan lokasi, dimana data yang di dapat berupa data spasial (latitude, longitude),
2. Perhitungan jarak antara posisi klien dan posisi rumah makan dilakukan berdasarkan rumus phytagoras dan haversian yaitu perhitungan jarak sederhana, dimana diimplementasikan pada function jarak di database rumah makan,
3. Dapat mengintegrasikan peta googlemaps dan GPS di android sebagai sebuah sistem berbasis GIS, yaitu uji coba pencarian lokasi rumah makan berdasarkan posisi klien berhasil dilakukan, dimana posisi klien terdeteksi oleh GPS di kampus UWKS yang ditampilkan dalam bentuk peta di *smartphone* android.
4. Penelitian yang dilakukan untuk mengetahui pengembangan aplikasi android untuk web service berhasil dilakukan, dimana proses *request* data rumah makan berdasarkan lokasi klien pada web service server berhasil dilakukan dimana muncul lokasi rumah makan terdekat di kampus UWKS dengan radius jarak 2 km. Proses menambahkan rumah makan berdasarkan lokasi klien berada berhasil di simpan pada database web service server.

Saran

Pada kesimpulan yang telah didapatkan pada maka dapat disarankan kajian dan perbaikan aplikasi yang bisa disampaikan, diantaranya adalah:

1. Aplikasi yang dibuat dapat dikembangkan lagi baik dalam fitur menampilkan rute arah dari posisi klien ke rumah makan disekitarnya,
2. Aplikasi yang dibuat dapat dikembangkan dalam implementasinya di daerah lain, di Indonesia,
3. Aplikasi yang dibuat dapat dikembangkan dalam bentuk tampilan peta, yaitu dengan menggunakan peta sendiri bukan berdasarkan peta googlemaps, dan juga dapat dilakukan pengembangan dalam bentuk peta 3D,
4. Pengembangan dalam hal database dapat ditambahkan tentang manajemen admin dalam mengelolah data rumah makan, sehingga data yang dimasukkan benar-benar nyata sesuai dengan di lapangan.

DAFTAR PUSTAKA

- Aprilia Rachmawati, Yuliana Setyowati, S.Kom., M.Kom., SCJP, Arna Fariza, S.Kom., M.Kom. *Pencarian Lokasi Fasilitas Umum Terdekat Pada Perangkat Mobile Dengan Penentuan Posisi User Menggunakan Cell ID*. Surabaya: Paper Teknik Informatika ITS
- Ardiansyah, M.Cs. 2011. *Mengenal Location Based Service*. URL: <http://blog.uad.ac.id/ardi/2011/04/04/mengenal-location-based-service-lbs/>. Diakses tanggal 3 Mei2012.
- Author's Guide. *Android Developer Tools* URL: <http://developer.android.com/tools/help/adt.htm> 1 Diakses tanggal 23 Juni 2012.
- Author's Guide. *SOAP vs REST* URL: <http://www.adityarizki.net/2012/06/soap-vs-rest-mana-yang-lebih-baik/>. Diakses tanggal 22 Juni 2012.
- Egi Adtya Pratama Putra. 2011. *Pembangunan Aplikasi Event Calender Pada Platform Android Dengan Dukungan Web Service*. Bandung: Perpustakaan UNIKOM.
- Fadilah Fahrul Hardiansyah, Yuliana Setiowati, S.Kom, M.Kom, Kholid Fathoni, S.Kom. *AUGMENTED REALITY UNTUK MENGETAHUI FASILITAS UMUM BERBASIS ANDROID*. Surabaya: Paper Teknologi Informasi ITS.
- Muhamad Nur Taufix, Nur Aini Rakhmawati, S.Kom, M.Sc.Eng, Radityo Prasetianto Wibowo, S.Kom. *Penerapan Teknologi Mashup Pada Aplikasi Pariwisata Berbasis Web Nusantara View: Modul Akomodasi*. Surabaya: Paper ITS.
- Soetam Rizky Wicaksono. *IMPLEMENTASI EXTENSIBLE MARKUP LANGUAGE WEB SERVICE PADA SECURE REMOTE FORM FIELD FILLIN DALAM DOKUMEN MICROSOFT WORD*. Surabaya: Paper Sistem Informasi STIKOM

(5)
**SISTEM OPTIMALISASI BANDWIDTH MENGGUNAKAN METODE LOAD BALANCING
PADA SQUID PROXY SERVER**

Ayu Dian Pertiwi¹, Anang Kukuh A², Beny Y.V. Nasution³,
Program Studi Teknik Informatika, Fakultas Teknik, Universitas Wijaya Kusuma Surabaya
Jl. Dukuh Kupang XXV/ 54, ayoedhian@gmail.com

ABSTRAK

Salah satu hal yang perlu di perhatikan dalam pemanfaatan dan penggunaan teknologi informasi adalah mengenai optimalisasi dalam pemanfaatan sumber daya jaringan yang ada. Meningkatnya beban pada *server*, jumlah *traffic* di internet dan ditambah lagi dengan permasalahan konektivitas menjadi perhatian penting dan harus segera dicarikan solusinya. Dan salah satu solusinya untuk mengurangi beban *server* adalah melakukan pembagian beban.

Load balancing adalah mengoptimalkan *bandwidth* yang tersedia pada dua buah jalur koneksi internet atau lebih secara merata dan membagi beban kumulatif pada sebuah jaringan. Dalam penggunaan *load balancing* ini saya menggunakan dua buah koneksi internet, yang menggunakan *mobile broadband* internet dan koneksi internet UWKS sebagai *provider* utama. Pembuatan *load balancing* ini menggunakan *pc router* debian.

Penamaan *Debian router* adalah komputer yang difungsikan sebagai *router* untuk mengatur jaringan. Yaitu adanya aktifitas *server* dan *client* dalam melakukan transaksi data. Dengan membuat *load balancing* menggunakan *mobile broadband* internet dan koneksi internet UWKS yang didalamnya dapat meningkatkan performansi jaringan dengan baik karena *network administrator* sebagai admin jaringan dapat mengatur penggunaan *bandwidth* yang ada sehingga dapat menyeimbangkan beban koneksi yang ada.

Kata kunci : *Load balancing, Bandwidth, Proxy Server.*

PENDAHULUAN

Jaringan komputer saat ini sangat dibutuhkan untuk menghubungkan berbagai instansi pemerintahan, kampus, dan bahkan untuk bisnis dimana banyak sekali perusahaan yang memerlukan informasi dan data-data dari kantor-kantor lainnya dan dari rekan kerja, afiliasi bisnis, dan konsumen. Satu hal yang perlu diperhatikan dalam pemanfaatan dan penggunaan teknologi informasi adalah mengenai optimalisasi dalam pemanfaatan sumber daya jaringan yang ada. Meningkatnya beban pada server, jumlah *traffic* di internet dan ditambah lagi dengan permasalahan konektivitas menjadi perhatian penting dan harus segera dicarikan solusinya.

Perancangan infrastruktur dan pembuatan topologi jaringan merupakan salah satu hal yang sangat penting untuk diperhatikan karena ke dua hal tersebut akan menentukan hasil akhir dari jaringan yang dibangun. Karena kesalahan perancangan dan topologi jaringan akan membuat jaringan yang dibangun tidak bekerja secara optimal atau bahkan dapat menimbulkan permasalahan baru.

Laboratorium komputer di Teknik Informatika Universitas Wijaya Kusuma, pada infrastruktur dan topologi jaringannya ditemukan beberapa permasalahan yang harus segera dicari solusinya. Permasalahannya adalah tidak jarang jalur internet mengalami error atau bahkan down.

Dengan terjadinya permasalahan tersebut banyak aktivitas yang akan terganggu, terutama pada kegiatan yang bergantung pada penggunaan internet, misalnya pencarian data/informasi di internet untuk kepentingan bahan ajar dan perkuliahan yang dilakukan oleh mahasiswa maupun dosen pengajar.

Sehingga dibutuhkan sebuah solusi agar permasalahan tersebut dapat teratasi, dan mampu bekerja dan beroperasi dalam waktu yang panjang. Efektivitas dari solusi tersebut juga menjadi perhatian, maka dari itu dalam pembuatan tugas akhir ini akan dibuat sebuah router untuk mengakses jaringan yang menggunakan beberapa koneksi internet yang kemudian diseimbangkan atau yang biasanya disebut dengan layanan *load balancing*.

Mengingat bahwa *bandwidth* yang rutin dipakai hanya dari satu provider saja, sedangkan *bandwidth* yang lainnya hanya digunakan pada kegiatan tertentu saja. Dengan penerapan konsep *load balancing* ini, diharapkan mampu menyediakan kenyamanan bagi setiap mahasiswa ataupun pengajar yang memanfaatkan fasilitas internet.

Tujuan

Tujuan dari Topik Khusus ini adalah:

1. Membuat *PC router* menggunakan *linux ubuntu*.

2. Merancang dan menerapkan *load balancing* dengan koneksi yang berasal dari provider internet yang berbeda.
3. Pembagian beban pada server.

TINJAUAN PUSTAKA

Proxy server merupakan suatu komputer/*hardware* yang menghubungkan dua jaringan. Umumnya *proxy server* digunakan untuk menghubungkan jaringan internal (*Private IP*) dengan jaringan eksternal (Public IP), hubungan antara *proxy server*, jaringan *internal*, dan jaringan *eksternal* bisa digambarkan secara kasar seperti berikut ini:

[jaringan internal] - eth0-<proxy>-eth1- [jaringan eksternal]

Private IP tidak bisa mengakses data pada *Public IP*, sehingga diperlukan *proxy server* untuk menghubungkan keduanya. Data yang dikirimkan dari komputer *internal* akan diubah IP asalnya menjadi IP *proxy server*, dan tanggapan yang didapat akan dikembalikan pada komputer pada jaringan internal yang melakukan request.

Pada awalnya, *proxy server* dikembangkan untuk menyimpan halaman web yang sering diakses. Saat pertama kali munculnya internet, koneksi sangat lambat, ruang lingkungannya masih relatif kecil, dan halaman web masih statis. Pada waktu itu internet hanya terdiri dari beberapa ribu situs yang sebagian besar untuk para ilmuwan dan akademis. Setiap kali ada berita penting pada suatu situs web maka banyak ilmuwan dari organisasi yang sama akan mengunjungi situs tersebut. *Proxy* dalam pengertiannya sebagai perantara, bekerja dalam berbagai jenis protokol komunikasi jaringan dan dapat berada pada level-level yang berbeda pada hirarki layer protokol komunikasi jaringan. Suatu perantara dapat saja bekerja pada layer Data-Link, layer Network dan Transport, maupun layer Aplikasi dalam hirarki layer komunikasi jaringan menurut OSI. Namun pengertian *proxy server* sebagian besar adalah untuk menunjuk suatu server yang bekerja sebagai *proxy* pada layer Aplikasi, meskipun juga akan dibahas mengenai *proxy* pada level sirkuit.

Dalam suatu jaringan lokal yang terhubung ke jaringan lain atau internet, pengguna tidak langsung berhubungan dengan jaringan luar atau internet, tetapi harus melewati suatu *gateway*, yang bertindak sebagai batas antara jaringan lokal dan jaringan luar. *Gateway* ini sangat penting, karena

jaringan lokal harus dapat dilindungi dengan baik dari bahaya yang mungkin berasal dari internet, dan hal tersebut akan sulit dilakukan bila tidak ada garis batas yang jelas jaringan lokal dan internet. *Gateway* juga bertindak sebagai titik dimana sejumlah koneksi dari pengguna lokal akan terhubung kepadanya, dan suatu koneksi ke jaringan luar juga terhubung kepadanya. Dengan demikian, koneksi dari jaringan lokal ke internet akan menggunakan sambungan yang dimiliki oleh *gateway* secara bersama-sama (*connection sharing*). Dalam hal ini, *gateway* adalah juga sebagai *proxy server*, karena menyediakan layanan sebagai perantara antara jaringan lokal dan jaringan luar atau internet.

Fungsi Proxy Server

Proxy Server memiliki banyak manfaat antara lain melakukan blocking URL (*Uniform Resource Locator*), memfilter content, mengecek konsistensi, memblokir *routing*, *logging* dan *alerting*, *caching*, dan *Reverse Proxy Load Balancing*.

1. Pemblokiran URL
Pemblokiran URL memungkinkan administrator untuk menolak situs tertentu berdasarkan URL. *Proxy server* mengecek setiap request dengan daftar halaman yang ditolak sebelum ia memperbarui request tersebut. Jika URL diblokir, *proxy server* tidak akan meminta atau memberikan halaman tersebut.
2. Pemfilteran Content
Karena *proxy server* memperbarui semua muatan protokol dan protokol spesifik, *proxy server* dapat digunakan untuk mencari muatan content yang mencurigakan. Ini berarti layanan HTTP *proxy* dapat dikonfigurasi untuk menguraikan ActiveX control, Java applet, atau bahkan gambar berukuran besar jika dirasa hal-hal tersebut bisa menyebabkan masalah keamanan.
3. *Caching*
Selain fasilitas-fasilitas diatas, *proxy server* juga dapat memberikan peningkatan kinerja yang penting. Pertama, *proxy server* dapat menyimpan data yang sering direquest untuk meningkatkan kinerja dengan menghilangkan akses berlebih ke jaringan eksternal (yang kecepatannya lebih lambat). Kedua, *proxy server* dapat menyeimbangkan beban layanan ke sejumlah server internal.
4. *Logging dan Alerting*
Proxy server juga menyediakan fasilitas *logging* dan *alerting*. *Proxy server*

memastikan bahwa semua content mengalir melalui satu tempat, yang bisa menjadi tempat pemeriksaan data jaringan. Kebanyakan *proxy server* akan mencatat penggunaan *proxy server* oleh user dan dapat dikonfigurasi untuk mencatat situs yang dikunjungi. Ini akan memungkinkan untuk mengatur ulang cara browsing user jika dicurigai adanya aktivitas yang ilegal atau tidak semestinya.

Fasilitas alert disediakan oleh beberapa *proxy server* untuk memberi peringatan atas serangan yang terjadi, meskipun *proxy server* umumnya bukan sasaran penyerangan. Namun, fasilitas ini bisa digunakan untuk meningkatkan kewaspadaan pada interface eksternal, yang sering kali dicoba untuk dieksploitasi oleh hacker.

Kelemahan *Proxy Server*

Selain fungsi-fungsi yang telah disebutkan di atas, ternyata *proxy server* juga mempunyai kelemahan, antara lain *proxy* dapat membuat *single point of failure*, sehingga *software client* sering kali harus dapat bekerja dengan *proxy server* (hanya sistem firewall dan *proxy server* tingkat lanjut yang dapat bekerja transparan pada jaringan), *proxy server* harus ada untuk semua layanan, *proxy server* tidak melindungi inti sistem operasi, dan masalah *proxy server* terhadap kinerja.

Proses *load balancing* sebenarnya merupakan proses fleksibel yang dapat diciptakan dengan berbagai cara dan metode. Proses ini tidak dapat dilakukan oleh sebuah perangkat tertentu atau sebuah software khusus saja. Cukup banyak cara dan pilihan untuk mendapatkan jaringan yang dilengkapi dengan sistem *load balancing*. Cara kerja dan prosesnya pun berbeda-beda satu dengan yang lainnya. Namun, cara yang paling umum dan banyak digunakan adalah dengan mengandalkan konsep *Virtual server* atau *Virtual IP*.

Istilah *Virtual server* atau *Virtual IP* sebenarnya merupakan istilah bebas, karena mungkin saja sistem lain menggunakan konsep yang sama namun dengan istilah yang berbeda. Secara umum, konsep dari *Virtual server* atau *Virtual IP* ini adalah sebuah alamat IP, sebuah nama, atau bias juga dikatakan sekelompok alamat IP yang bertugas sebagai jembatan penghubung antara pengakses dari luar dengan sekelompok server atau perangkat jaringan yang berada dibelakangnya.

Tujuan dibuatnya sistem perwakilan tersebut adalah agar ketika nama atau alamat IP tersebut diakses dari luar, yang dapat melayani permintaan

tersebut tidak terbatas hanya satu perangkat server saja. Sekelompok server atau perangkat jaringan yang diwakilinya memiliki kemampuan untuk menjawab permintaan-permintaan tersebut. Sebagai hasilnya, permintaan-permintaan tersebut terdistribusi ke beberapa server sehingga beban proses kerja server-server tersebut tidak terlalu berat. Hal ini membuat servis dan layanan yang diberikan server tersebut ke si pengguna dapat berjalan lebih baik dan berkualitas.

SQUID PROXY SERVER

Menurut Dodi Maryanto (2001), *Squid Proxy Server* digunakan untuk melakukan efisiensi *bandwidth* internet pada sebuah organisasi yang terhubung ke jaringan internet. Hal ini diperlukan mengingat antara lain kecenderungan pengguna/sekelompok pengguna internet yang mengakses halaman web tertentu secara terus-menerus pada suatu kurun waktu.

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.

Squid layaknya sebuah pembantu untuk proses mempercepat kegiatan *browsing*, dimana *squid* dapat menyimpan semua obyek-obyek dari sebuah atau beberapa *website* ke dalam media lokal dan akan di berikan ketika ada permintaan setelah sebelumnya *squid* melakukan proses kalkulasi. Dalam pengadopsian *squid* untuk kehidupan nyata sebenarnya relatif, maksudnya disini relatif adalah *squid* tergantung dari beberapa faktor. antara lain: berapa kecepatan prosesor yang di gunakan sebagai server *squid*, berapa banyak client yang dilayani oleh *squid*, berapa cepat *squid* bisa mencapai client, dan masih banyak lagi faktor-faktor teknis dan non-teknis yang bisa mempengaruhi performance *squid*.

Sedangkan *squid* adalah sebuah *software* yang dapat menjalankan fungsi *proxy* pada Ubuntu. *Proxy server* adalah komputer server dan/atau program komputer yang dapat bertindak sebagai

komputer lainnya, fungsinya untuk melakukan *request* terhadap content / halaman / objek dari Internet atau intranet. *Proxy Server* bertindak sebagai *gateway* ke dunia Internet untuk setiap komputer klien, *Proxy server* tidak terlihat (*software*) oleh komputer klien. Pengguna yang berinteraksi dengan Internet melalui sebuah proxy server dia tidak akan mengetahui bahwa sebuah proxy server sedang menangani *request*/permintaan content/halaman/objek yang dilakukannya. Web server (*website server*) yang menerima *request* dari proxy server akan menginterpretasikan *request-request* tersebut seolah-olah *request* itu datang secara langsung dari komputer klien, bukan dari *proxy server*.

Hal tersebut tidak hanya terjadi di perusahaan yang menggunakan internet, tetapi juga termasuk di warnet-warnet. Hal ini akan membebani bandwidth karena setiap permintaan halaman web akan diperlakukan dengan cara yang sama, yaitu web server yang diminta akan memberikan halaman web yang diminta secara utuh, terlepas dari apakah halaman tersebut pernah diminta sebelumnya, dalam waktu yang relatif singkat.

Cara Kerja Squid Proxy Server

Squid proxy server melakukan tugasnya dengan menyimpan halaman- halaman web (selanjutnya disebut object) yang pernah menjadi request oleh client pada suatu jaringan internal, sehingga apabila ada permintaan terhadap object yang sama dan pernah diakses sebelumnya dalam perbedaan waktu yang singkat atau jika ada object yang tadinya disimpan itu belum kadaluarsa, maka squid proxy server tidak akan meneruskan permintaan itu ke web server di jaringan internet, melainkan akan memberikan object yang disimpannya kepada client yang membutuhkan. Object/Internet Object adalah file , dokumen atau respon terhadap sebuah permintaan terhadap berbagai layanan seperti FTP atau HTTP.

Hal ini tentunya akan memberikan penghematan bandwidth karena object yang sama tidak perlu secara diambil berulang-ulang dari jaringan internet. Penghematan ini bervariasi tergantung dari beberapa faktor misalnya: jumlah pengguna yang terhubung ke jaringan internet melalui squid proxy server, perilaku pengguna dalam mengakses halaman web, kesamaan/kemiripan tugas dan hobi dari masing-masing pengguna. Secara umum, penghematan dapat berkisar antara 30-70%. Dengan pertimbangan bahwa harga bandwidth

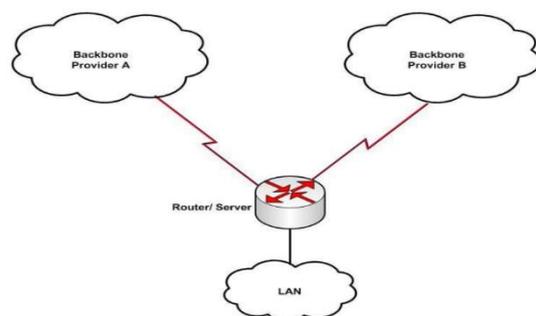
dihitung dalam dollar Amerika, maka secara ekonomik penghematan itu akan sangat besar artinya bagi organisasi tersebut.

Dalam dokumentasinya, Squid proxy server tidak hanya melayani akses HTTP, tetapi juga gopher dan FTP. Dari sisi keamanan, Squid proxy server mengakomodasi SSL (Secure Socket Layer), access control yang dapat dikonfigurasi dan pencatatan akses (logging) secara penuh. Untuk mengakomodasi jaringan serta permintaan yang semakin besar, Squid proxy server juga dapat disusun membentuk hirarki atau hubungan tertentu untuk memaksimalkan penghematan bandwidth.

Sistem Load Balancing

Sistem *load balancing* yang sederhana memang hanya mampu membuat sebuah perwakilan nama atau alamat IP untuk mewakili beberapa IP dari server-server dibelakangnya, namun perangkat yang memang dikhususkan menangani sistem load balancing kompleks dapat melakukan perwakilan hanya terhadap servis-servis yang dibuka oleh server dibelakangnya.

Dalam sistem *load balancing*, proses pembagian bebannya memiliki teknik dan algoritma tersendiri. Pada perangkat *load balancing* yang kompleks biasanya disediakan bermacam- macam algoritma pembagian beban ini. Tujuannya adalah untuk menyesuaikan pembagian beban dengan karakteristik dari *server-server* yang ada di belakangnya. Untuk dapat mengimplementasikan system ini diperlukan suatu perangkat tambahan baik berupa router Cisco atau menggunakan solusi router dari Mikrotik yang lebih ekonomis namun powerful.



Gambar 2.2 Solusi *load balancing* dengan dua *backbone provider*

Secara umum, algoritma-algoritma pembagian beban yang banyak digunakan saat ini adalah :

1. *Round Robin*

Algoritma Round Robin merupakan algoritma yang paling sederhana dan banyak digunakan oleh perangkat *load balancing*. Algoritma ini membagi beban secara bergiliran dan berurutan dari satu *server* ke server lain sehingga membentuk putaran.

2. *Ratio*

Ratio (rasio) sebenarnya merupakan sebuah parameter yang diberikan untuk masing-masing *server* yang akan dimasukkan kedalam sistem *load balancing*. Dari parameter *Ratio* ini, akan dilakukan pembagian beban terhadap *server-server* yang diberi rasio. Server dengan rasio terbesar diberi beban besar, begitu juga dengan server dengan rasio kecil akan lebih sedikit diberi beban.

3. *Fastest*

Algoritma yang satu ini melakukan pembagian beban dengan mengutamakan *server-server* yang memiliki respon yang paling cepat. Server di dalam jaringan yang memiliki respon paling cepat merupakan server yang akan mengambil beban pada saat permintaan masuk.

4. *Least Connection*

Algoritma *Least connection* akan melakukan pembagian beban berdasarkan banyaknya koneksi yang sedang dilayani oleh sebuah server. Server dengan pelayanan koneksi yang paling sedikit akan diberikan beban yang berikutnya akan masuk.

Sistem Load Balancing

Seperti telah dijelaskan di atas, sistem *load balancing* sebenarnya dapat dibuat dengan banyak cara. Pembuatannya tidak terikat oleh sebuah *operating system* saja, atau hanya dapat dibuat oleh sebuah perangkat saja. Namun secara garis besar cara pembuatan sistem *load balancing* terbagi menjadi tiga kategori besar, yaitu *load balancing* dengan menggunakan *DNS round robin*, *Integrated load balancing*, dan *Dedicated load balancing*. Ketiga jenis ini memiliki cara kerja yang unik dan berbeda satu sama lain, tetapi tetap menuju suatu hasil akhir yang sama, yaitu menciptakan sebuah sistem yang lebih menjamin kelangsungan hidup jaringan di belakangnya dan membuatnya lebih skalabel.

1. *DNS round robin*

Metode yang paling sederhana untuk menciptakan sistem *load balancing* adalah dengan menggunakan metode *DNS Round robin*. Metode ini sebenarnya merupakan sebuah fitur dari aplikasi

bernama *BIND* (*Berkeley Internet Name Domain*). Ini merupakan aplikasi open source khusus untuk membangun server *DNS* yang tampaknya sudah menjadi semacam standar yang digunakan di mana-mana. Sistem *DNS round robin* banyak mengandalkan teknik input penamaan yang teratur rapi dan dipadukan dengan sistem perputaran *round robin*.

Seperti yang telah kita ketahui, *DNS* merupakan sebuah sistem penamaan terhadap perangkat-perangkat komputer. Penamaan ini dibuat berdasarkan alamat *IP* dari perangkat tersebut. Sebuah perangkat yang memiliki alamat *IP* dapat diberi nama dan dapat diakses menggunakan namanya saja jika anda memiliki *DNS Server*. Sistem penamaan tersebut banyak sekali manfaatnya, misalnya hanya sekedar untuk lebih mudah diakses atau diproses lebih lanjut.

Dari sistem penamaan ini dapat dibuat sebuah sistem *load balancing* sederhana dan murah yang memanfaatkan sifat alami dari program *BIND*, yaitu sistem perputaran *round robin*. Pada sebuah record *DNS* yang berisikan informasi penamaan, anda dapat memasukkan beberapa nama lain untuk diwakili oleh sebuah nama utama. Beberapa nama lain itu memiliki masing-masing record sendiri yang juga mewakili alamat *IP* dari masing-masing perangkat jaringan. Jadi setelah proses inputan penamaan selesai, anda dapat mendapatkan sebuah nama lain yang mewakili beberapa perangkat jaringan seperti *server*.

Disinilah kuncinya, ketika ada yang mengakses nama utama, *DNS server* akan menghubungi oleh si pencari. Setelah menerima permintaan, *DNS server* akan mencari record dari nama utama tersebut. Ternyata di dalam record tersebut terdapat beberapa nama lain yang berhubungan dengan nama utama. Pada kondisi inilah, *DNS server* akan menjalankan sistem perputaran *round robin* untuk mengaliri informasi nama-nama lain mana saja yang diberikan kepada para pemintanya.

Disinilah, sistem *load balancing* sebenarnya sudah terjadi. Alamat *IP* dari server yang mewakili oleh nama lain tersebut akan diberikan kepada para peminta secara bergantian sesuai algoritma *round robin*. Ini akan menjadikana beban terbagi secara bergiliran ke server lain dengan sendirinya.

Sistem *load balancing* ini terbilang mudah dan sederhana. Namun ada beberapa kelemahan yang signifikan. Problem yang sering terjadi adalah ketika ada *DNS server* lain di internet

yang masih meng-cache hasil pencarian yang pertama.

Jadi jika kali pertama server DNS lain tersebut mendapatkan informasi IP dari domain utama perusahaan tersebut adalah IP 1.1.1.2, maka DNS itu akan tidak mengetahui IP dari domain utama tersebut. Ini membuat para pengguna server DNS ini juga tidak mengetahui sistem load balancing yang ada.

2. Integrated load balancing

Integrated load balancing biasanya merupakan solusi load balancing tambahan dari sebuah aplikasi atau operating system. Biasanya aplikasi atau *operating system* yang memiliki fitur ini adalah yang memiliki kemampuan beroperasi sebagai *server*. Sistem load balancing bukan merupakan fungsi utama. Oleh sebab itu, biasanya fitur, performa, dan kemampuannya cukup sederhana dan digunakan untuk sistem berskala kecil menengah. Fasilitasnya juga lebih banyak bersifat general saja, jarang yang spesifik. Meski demikian, fitur ini amat berguna jika digunakan pada jaringan yang tepat.

Salah satu integrated load balancing ini dapat anda temukan di microsoft windows 2000 *advance server* yang merupakan fitur tambahan. Pada operating system yang memiliki kemampuan jaringan yang hebat ini, anda dapat melakukan konfigurasi sistem load balancing dengan cukup mudah. Selain itu, fitur yang diberikan untuk keperluan ini juga terbilang cukup lengkap.

3. Component Load Balancing (CLB)

Teknologi load balancing ini menyediakan sistem load balance terhadap komponen-komponen yang mendukung jalannya sebuah software atau aplikasi. Aplikasi atau *software* yang dapat di-load balance adalah yang komponen-komponennya menggunakan COM+. Dengan melakukan *load balancing* terhadap komponen-komponen COM+ yang ada di beberapa *server*, maka jalannya sebuah aplikasi lebih terjamin dan lebih skalabel melayani para pengguna aplikasi.

4. Network Load Balancing (NLB)

Network load balancing merupakan fasilitas yang memungkinkan mesin windows 2000 *advance server* melakukan load balancing terhadap aplikasi yang berjalan berdasarkan jaringan IP. Aplikasi yang berjalan di atas IP seperti HTTP/HTTPS, FTP, SMTP, dan masih banyak lagi dapat dengan mudah load balancing dengan menggunakan fasilitas ini. Dengan menggunakan NLB, anda dapat membuat group cluster server yang dilengkapi dengan sistem load balancing terhadap semua service TCP. Untuk

semua proses tersebut, dikenal dengan istilah virtual server yang bertindak sebagai satu titik pusat pengaksesan server dibawahnya. Dengan adanya fasilitas ini, service dan layanan yang dijalankan oleh server ini lebih terjamin kelancarannya. Sangat ideal untuk digunakan untuk keperluan servis *fron-end*, seperti web server agar masalah-masalah seperti bottleneck pada server terkurangi.

Keunggulan Load Balancing

Ada beberapa keuntungan pada saat menggunakan system *load balancing*, diantaranya adalah :

1. Asymetric Load

Rasio dapat dibuat dengan menentukan koneksi yang menjadi *primary* yang dianggap paling baik backbonenya dan terbaik dalam path routingsnya, jadi kita dapat membuat mesin untuk mencari best path determination dan routing yang terpendek dan terbaik untuk sampai ketujuan.

2. Aktivitas berdasarkan Prioritas

Disaat load jaringan lagi peek, server akan dapat membagi aktivitas berdasarkan prioritas dan ke link cadangan.

3. Proteksi dari serangan DDoS

Karena kita dapat membuat features seperti SYN Cookies dan delayed-binding (suatu metode di back-end server pada saat terjadi proses TCP *handshake*) pada saat terjadi serangan SYN Flood

4. Kompresi HTTP

Memungkinkan data untuk bisa mentransfer objek HTTP dengan dimungkinkannya penggunaan utilisasi kompresi gzip yang berada di semua web browser yang modern.

5. TCP Buffering

Dapat membuat respon buffer dari server dan berakibat dapat memungkinkan task akses lebih cepat

6. HTTP Caching

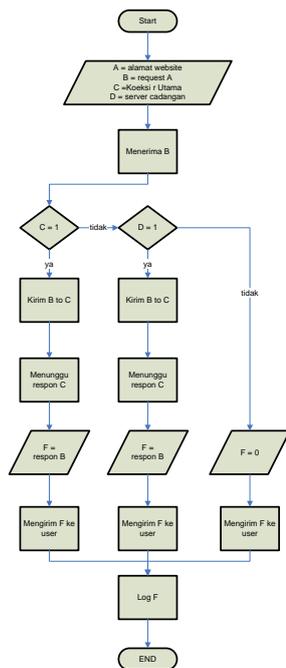
Dapat menyimpan content yang static, dengan demikian request dapat di handel tanpa harus melakukan kontak ke web server diluar jaringan yang berakibat akses terasa semakin cepat.

7. Content Filtering

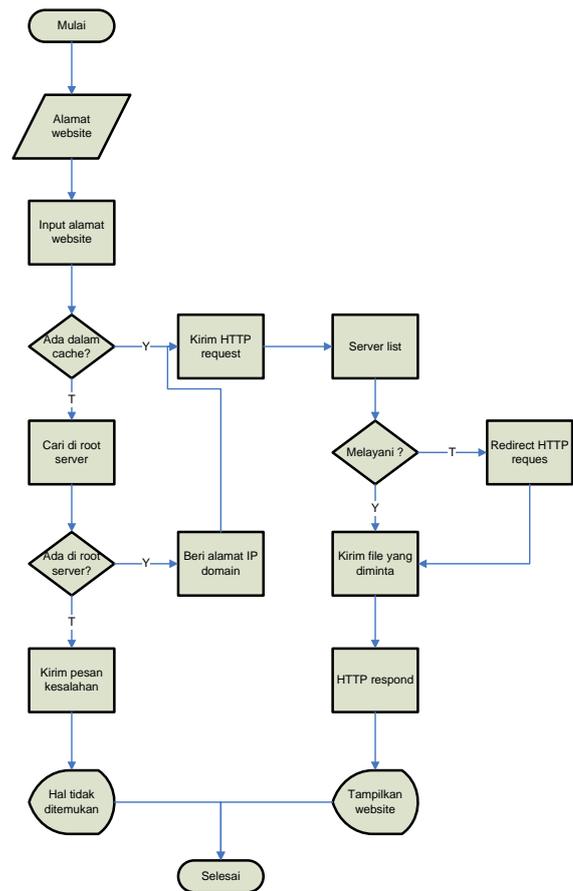
Beberapa load balancing dapat melakukan perubahan trafik pada saat dijalankan.

8. **HTTP Security**
Beberapa *system load balancing* dapat menyembunyikan HTTP error pages, menghapus identifikasi header server dari respon HTTP, dan melakukan enkripsi *cookies* agar user tidak dapat memanipulasinya.
9. **Spam Filtering**
Spam merupakan penyalahgunaan dalam pengiriman berita elektronik untuk menampilkan berita iklan dan keperluan lainnya yang mengakibatkan ketidaknyamanan bagi para pengguna web. Bentuk berita spam yang umum dikenal meliputi: spam surat elektronik, spam Usenet newsgroup, spam mesin pencari informasi web (web search engine spam), spam blog, spam berita pada telepon genggam, spam forum Internet, dan lain lain.

Pada gambar 3.6 menjelaskan PC Router menerima request dari client yang ditujukan pada isp pertama, disini pc router menangani proses dimana terjadi *request* (permintaan) dari user terlalu banyak sehingga menyebabkan koneksi internet pertama down bahkan tidak dapat diakses, yang kemudian oleh pc router akan dialihkan ke isp cadangan. Koneksi internet aktif diasumsikan sama dengan 1 dan koneksi internet *lost* (putus) sama dengan 0. User akan menerima hasil tanggapan dari pc router server bahwa koneksi aktif atau tidaknya koneksi internet



Gambar 3.6 Flowchart Load Balancing

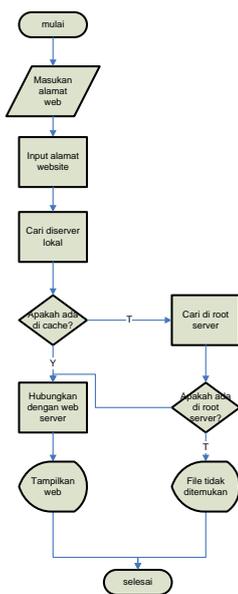


Gambar 3.7 flowchart Terima Request URL

Pada gambar 3.7 menjelaskan tentang alur proses terima permintaan URL oleh server. Pada saat client meminta permintaan website maka, sistem akan mengecek pada cache server. diteruskan kirim HTTP permintaan dan mengecek server, jika iya maka file yang diminta akan direspon dengan memunculkan alamat website yang diminta.

Dan jika pada cache server tidak ada server secara otomatis akan mencari pada server luar, jika server luar tidak ada maka akan server akan memunculkan halaman pemberitahuan file tidak ditemukan. Dan jika server luar memberikan maka akan mengirimkan IP domain dan mengecek server. Jika iya maka server akan menampilkan file yang diminta oleh client.

Pada flowchart 3.8 menjelaskan proses cek permintaan pada cache server. jika client meminta permintaan server akan mencari pada cache server jika tidak ada maka server akan menghubungkan dan mencari pada server luar. Jika file sudah ditemukan, maka server akan menampilkan permintaan client.



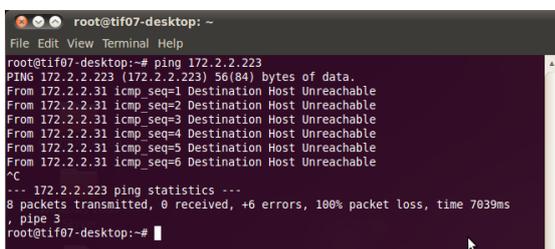
Gambar 3.8 flowchart cek URL pada cache

HASIL DAN PEMBAHASAN

Hasil Dan Pembahasan akan dilakukan pengujian supaya aplikasi yang telah dibuat bisa memenuhi tujuan pembuatan laporan tugas akhir ini.

Uji Coba Salah satu koneksi internet down / putus

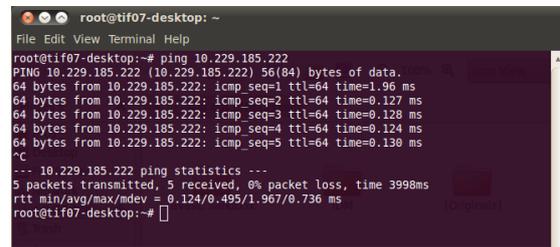
Percobaan ini dilakukan dengan cara memutuskan koneksi internet dari ISP dari kampus. Jika koneksi internet terputus, maka komputer router otomatis akan mengarahkan request internet dari client dari lokal ke ISP cadangan atau koneksi smart.



Gambar 5.1 Ping IP address Kampus Lost

Pada gambar 5.1 terlihat bahawa jika kita ping IP internet kampus tidak bisa, disini diasumsikan sebagai ISP dari kampus lost atau terputus. Dan pada gambar 5.2 ping IP dari smart dan berhasil. Terlihat ketika koneksi kampus terputus maka ketika dilakukan tracert, terlihat bahwa router load

balancing mengarahkan request ke jalur koneksi smart dan client masih tetap mengakses website seperti terlihat pada gambar berikut.



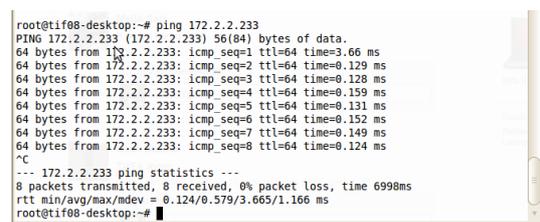
Gambar 5.2 Ping IP address Koneksi smart



Gambar 5.3 Akses Internet dari Koneksi Smart

Pada gambar 5.3 ini akan muncul permintaan dari client melalui koneksi smart. Dan selanjutnya ketika koneksi Smart terputus, maka komputer router akan secara otomatis mengarahkan request pada koneksi kampus.

Pada gambar 5.4 menunjukkan koneksi kampus dalam keadaan normal dan berjalan. Dan apabila koneksi koneksi smart terputus maka secara otomatis koneksi internet akan melalui koneksi kampus. Tanpa client mengetahui bahwa koneksi provider smart sedang down / bahkan terputus.



Gambar 5.4 Ping IP address Kampus

Dan terlihat pada gambar 5.5 client akan tetap bisa menikmati koneksi internet tanpa mengetahui bahwa salah satu koneksi terputus



Gambar 5.5 Akses Internet dari Koneksi Kampus

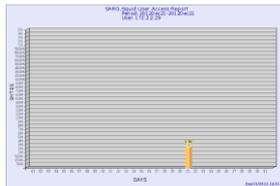
Uji Coba Monitoring aktifitas internet client dengan SARG

Dengan aplikasi SARG server bisa memonitoring aktivitas dari client, diantaranya durasi client koneksi internet, berapa banyak bandwidth yang digunakan, berapa banyak client menggunakan cache memory. Pada tampilan SARG dibawah ini menunjukkan beberapa aktivitas client yang terhubung pada server router.



Gambar 5.6 Tampilan SARG

Pada gambar 5.6 adalah tampilan dari SARG, pada tabel user id adalah aktifitas IP client maupun server yang termonitoring oleh sarg. Dan adapula tampilan grafik yang menunjukkan penggunaan bandwidth pada salah satu client seperti terlihat pada gambar 5.7 dibawah ini.



Gambar 5.7 Grafik SARG client

SARG Squid Analysis Report Generator

Squid User Access Report
Period: 2012Dec21-2012Dec21
User: 172.2.2.51
Sort: BYTES, reverse
User Report

ACCESSED SITE	CONNECT	BYTES	%BYTES	IN-CACHE-OUT	ELAPSED TIME	MILISEC	%TIME
salebrowsing-cache.google.com	15	1.71M	95.76%	0.00%	100.00%	00:00:35	35.663 83.67%
www.google.com	18	69.56K	3.88%	5.47%	94.53%	00:00:13	13.889 24.80%
salebrowsing.clients.google.com	1	4.15K	0.23%	0.00%	100.00%	00:00:00	613 1.09%
clients1.google.com	5	1.28K	0.07%	40.76%	59.24%	00:00:03	3.590 6.41%
ssl.gstatic.com	4	984	0.05%	100.00%	0.00%	00:00:02	2.255 4.03%
TOTAL	43	1.79M	16.24%	0.30%	99.70%	00:00:56	56.010 3.02%
AVERAGE	488	3.67M				00:10:18	618.081 33.33%

Generated by sarg-2.2.6 Dec-01-2009 on Dec21/2012 14:31

Gambar 5.7 SARG untuk client

Pada gambar 5.7 dimunculkan aktifitas dari client dengan IP 172.2.2.51. Disini client dengan IP address tersebut, pada tanggal 21 desember telah membuka beberapa situs internet. Diantaranya adalah google.com, softbroasing.com, dll. Dan terlihat cache in 0% dan cache out 100% dan lain-lain.

Uji Coba Pengujian Kecepatan Load Balancing

Dan uji coba selanjutnya adalah uji coba tes kecepatan upload dan download pada client, menggunakan layanan speedtest.net yang ada pada browser. Berikut ini adalah hasil dari ujicoba kecepatan tanpa melewati server router load balancing.



Gambar 5.8 Pengujian kecepatan tanpa Load Balancing

Terlihat ketika tanpa menggunakan sistem load balancing speed download 4.29Mb/s dan untuk upload 0.70Mb/s. Dan apabila client sudah terhubung pada server dan menggunakan sistem load balancing hasil kecepatan terlihat berbeda download nya mencapai 0.19Mb/s dan untuk uploadnya 0.06Mb/s.



Gambar 5.9 Pengujian kecepatan menggunakan Load Balancing

KESIMPULAN DAN SARAN

Sistem load balancing pada squid proxy ini difungsikan sebagai solusi adanya kondisi dimana banyaknya traffic dan seringnya terjadi down terlebih putus pada koneksi internet di laboratorium teknik wijaya kusuma. Sistem load balancing ini menggunakan 2 koneksi internet yang berbeda, dan untuk mengoptimalkan bandwidth yang tersedia. Namun setelah implementasi system ada kekurangan.

Kesimpulan

Berdasarkan uraian dan pembahasan analisa dan pengujian yang telah dilakukan, maka dapat di ambil kesimpulan bahwa sistem load balancing pada Laboratorium Teknik Informatika

yang menggunakan *mobile broadband* internet sangat bergantung pada koneksi internet yang ada.

Saran

Kekurangan lain di sistem ini yaitu belum terdapat pembagian bandwidth. Juga untuk mendapatkan sinyal *broadband* terbaik kita harus meminimalkan penghambat yang menghambat sinyal seperti dinding, besi, atau baja. Penulis menyadari masih banyak kekurangan dalam melakukan penelitian dan pembuatan laporan ini. Sebagai penutup, penulis berharap semoga laporan ini bermanfaat bagi penulis sendiri dan semua yang sedang mempelajari tentang jaringan komputer khususnya *load balancing*

DAFTAR PUSTAKA

- [1] Abas Ali Pangeran, 2008, *Menjadi Administrator Jaringan Nirkabel*, Yogyakarta, ANDI Yogyakarta.
- [2] Dony Ariyus & Rum Andri K.R, 2008, *Komunikasi Data*, Yogyakarta, ANDI Yogyakarta.
- [3] Hubert, Bert, and cs, *Linux Advance Routing & Traffic Control*, <http://www.lartc.org/>, 2004 (diakses 24 November 2012)
- [4] Purbo, Onno, Adnan Basalamah, Ismail Fahmi dan Achmad Husni Thamrin, *TCP/IP Standar, Desain, dan Implementasi*, Buku Pintar Internet, PT Elex Media Komputindo.
- [5] Santoso, Budi, 2010. *Manajemen Bandwith Internet Dan Intranet*, Jayadi, M.
- [6] Tanenbaum, Andrew S .(1996) *Jaringan Komputer Edisi Bahasa Indonesia Jilid 1*. Prehallindo : Jakarta

(6)
SISTEM INFORMASI MANAJEMEN RAPAT

Achmad Nashihuddin, Nonot Wisnu Karyanto
Program Studi Teknik Informatika, Fakultas Teknik, Universitas Wijaya Kusuma Surabaya
huddin_mer@yahoo.com

ABSTRAK

Rapat pada suatu perusahaan, biasanya dilakukan untuk menemukan suatu solusi dari permasalahan atau untuk menentukan rencana kerja perusahaan. Dalam mengadakan rapat penyelenggara sering menghadapi berbagai permasalahan, diantaranya kurangnya informasi tentang rapat yang akan diselenggarakan, sehingga jumlah kehadiran peserta tidak sesuai dengan apa yang diharapkan. Karena itu, diperlukan suatu sistem yang mampu menangani seluruh kegiatan rapat, sehingga rapat bisa berjalan dengan lancar. Sistem informasi manajemen rapat merupakan suatu aplikasi yang bisa memudahkan karyawan dalam mengajukan rapat, merekap seluruh data pengajuan rapat, mengatur jadwal peserta, mengatur jadwal ketersediaan ruang rapat, mendata konfirmasi kehadiran peserta rapat dan menyimpan notulen atau hasil keputusan rapat. diharapkan dengan system ini seluruh agenda rapat bisa berjalan dengan baik dan menghasilkan keputusan yang disepakati.

Kata kunci : Web, Manajemen Rapat

PENDAHULUAN

Latar Belakang

Rapat diperlukan dalam aktivitas kerja siapapun. Rapat yang ditangani dengan baik, akan membantu peserta yang hadir menemukan solusi dari situasi yang berat, tindakan yang disepakati, dan melakukan sesuatu yang positif. Apabila rapat tidak ditangani dengan baik, hal ini bisa membuat waktu dan biaya terbuang dengan percuma. Pada dasarnya, setiap orang ingin hadir dan keluar dari ruang rapat secepat mungkin dengan keputusan yang *relevan*.

Dalam mengadakan rapat penyelenggara sering menghadapi berbagai permasalahan. Diantaranya kurangnya informasi tentang rapat yang akan diselenggarakan, sehingga jumlah kehadiran peserta tidak sesuai dengan apa yang diharapkan. Apabila jumlah kehadiran peserta kurang hal ini akan memberikan pengaruh terhadap pengambilan keputusan. Sehingga keputusan rapat yang dihasilkan tidak sesuai dengan harapan.

Sehubungan dengan hal berikut, dapat diberikan suatu solusi dengan merancang sistem informasi manajemen rapat berdasarkan sistem manajemen rapat manual yang sudah ada yang masih kurang efektif dan efisien, Dengan adanya aplikasi ini diharapkan bisa memudahkan karyawan dalam mengajukan rapat, merekap seluruh data pengajuan rapat, mengatur jadwal peserta, mengatur ruang rapat, mendata konfirmasi kehadiran peserta rapat dan menyimpan notulen atau hasil keputusan rapat, sehingga rapat

menghasilkan suatu keputusan yang sudah disepakati.

TINJAUAN PUSTAKA

Pengertian Rapat

Rapat merupakan kumpulan beberapa orang atau organisasi yang akan membicarakan suatu masalah atau kepentingan bersama untuk memberikan penjelasan, memecahkan suatu persoalan dan sekaligus mengadakan perundingan demi memperoleh suatu hasil yang disepakati atau disetujui bersama. Jadi rapat merupakan bentuk komunikasi yang dihadiri oleh beberapa orang untuk membicarakan dan memecahkan permasalahan tertentu, di mana melalui rapat berbagai permasalahan dapat dipecahkan dan berbagai kebijaksanaan organisasi dapat dirumuskan (Samsir Rambe, 1991).

Tata Tertib Rapat

Agar rapat bisa mencapai maksud dan tujuannya, hendaknya rapat harus dikelola dengan baik dan harus mengetahui tata tertib rapat yang memenuhi kriteria. Beberapa tata tertib rapat berdasarkan buku etika komunikasi karangan Samsir Rambe diantaranya sebagai berikut :

1. Tepat waktu dalam memulai rapat.
2. Agenda rapat dirumuskan atau disusun dengan baik sehingga peserta rapat dapat mengetahui susunan acara rapat.
3. Setiap peserta saling menghargai pendapat yang dikemukakan peserta lain.
4. Adanya partisipasi dari peserta rapat.

- Bersifat terbuka, artinya bersedia menerima kritik dan saran dari peserta lain tanpa emosi. Dengan tidak melihat siapa yang berbicara, tapi setiap peserta mau mendengar pendapat orang lain.
- Setiap argumen atau pertanyaan yang diajukan disampaikan secara singkat, jelas dan lugas
- Selalu ada kesimpulan yang diambil berdasarkan argumen-argumen yang disetujui bersama.

Tugas Pokok Kegiatan Rapat

PT. Pelabuhan Indonesia III adalah perusahaan penyedia jasa dipelabuhan. Dalam mengatur kegiatan rapat ada beberapa tugas pokok atau *job description* seperti berikut :

- Unit Fungsional
 - Mengajukan permohonan rapat dan menentukan tanggal pelaksanaan rapat
 - Melakukan penundaan rapat apabila jumlah kehadiran tidak sesuai dengan ketentuan rapat.
 - Menentukan peserta rapat dengan ketentuan peserta tidak terikat dengan agenda rapat lain, cuti atau tugas keluar kota
 - Menyiapkan materi rapat dan menyimpulkan notulen atau hasil rapat.
 - Mengirimkan informasi penyelenggaraan rapat kepada peserta rapat.
- Unit Umum Konsumsi .
 - Memonitor jadwal rapat.
 - Mencetak daftar permintaan konsumsi
 - Melihat dan mencetak bukti daftar hadir rapat

Pengertian UML

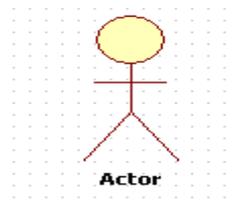
UML (*Unified Modeling Language*) adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan artifact (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. Artifact dapat berupa model, deskripsi atau perangkat lunak) dari system perangkat lunak, seperti pada pemodelan bisnis dan *system non perangkat lunak* lainnya (Sholiq,2005).

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. Bagian-bagian diagram utama dari UML adalah :

1. 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.



Gambar 2.1 aktor pada usecase

Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

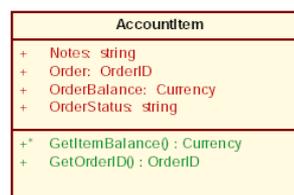


Gambar 2.2 usecase

2. Class Diagram

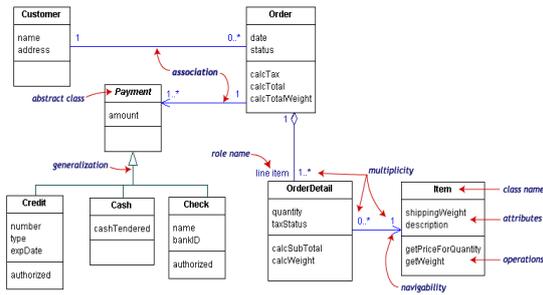
Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.



Gambar 2.4 Diagram class

Contoh diagram class



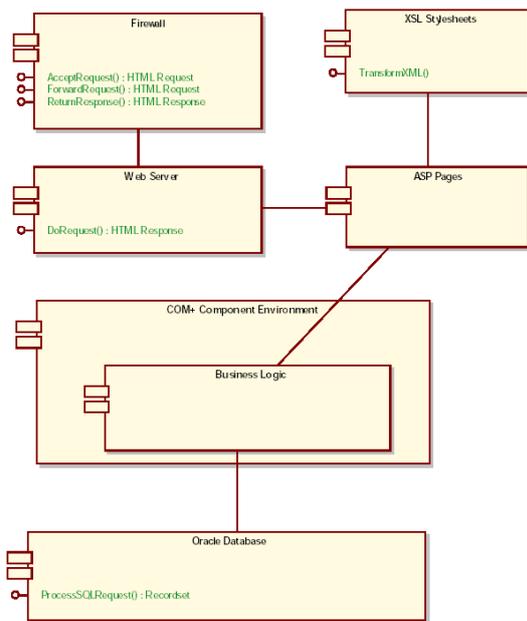
Gambar 2.5 contoh diagram class

3. 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.

Contoh diagram component

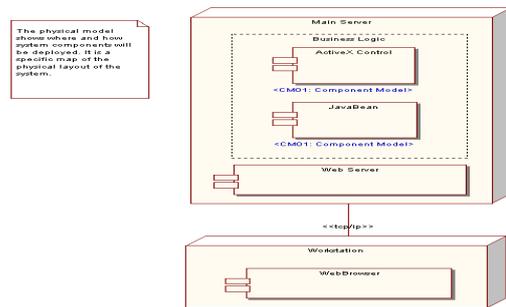


Gambar 2.6 contoh diagram komponen

4. 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 2.7 contoh diagram deployment

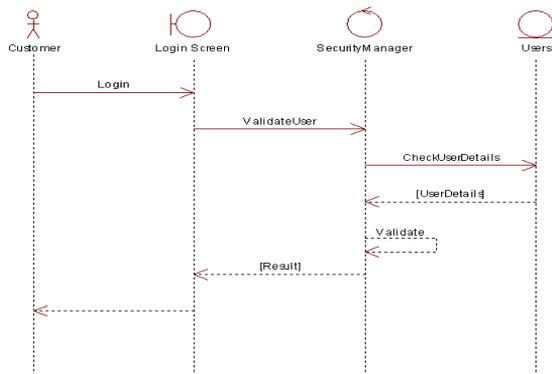
5. Sequence Diagram

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 2.8 contoh diagram sequential

6. Statechart diagram

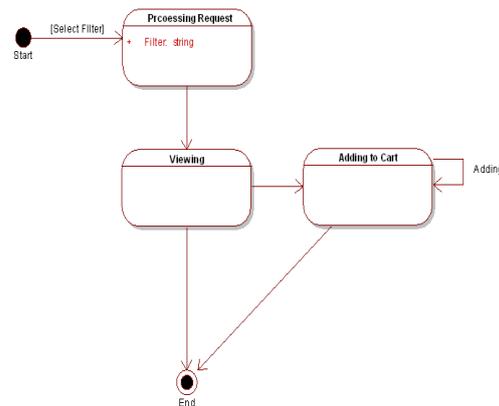
Statechart *diagram* menggambarkan transisi dan perubahan keadaan (dari satu *state* ke *state* lainnya) suatu objek pada sistem sebagai akibat dari *stimuli* yang diterima. Pada umumnya *statechart diagram* menggambarkan *class* tertentu (satu *class* dapat memiliki lebih dari satu *statechart diagram*).

Dalam UML, *state* digambarkan berbentuk segiempat dengan sudut membulat dan memiliki nama sesuai kondisinya saat itu. Transisi antar *state* umumnya memiliki kondisi *guard* yang merupakan syarat terjadinya transisi yang bersangkutan, dituliskan dalam kurung siku. *Action* yang dilakukan sebagai akibat dari *event* tertentu dituliskan dengan diawali garis miring. Titik awal dan akhir digambarkan berbentuk lingkaran berwarna penuh dan berwarna setengah.

Untuk menentukan apakah sebuah *class* mempunyai perilaku dinamik yang signifikan, dilakukan dengan memperlihatkan dua hal yaitu :

- *Attribut* : jika sebuah kelas mempunyai sebuah attribute yang bernama status atau *attribute* yang mempunyai nilai berbeda yang digunakan menyimpan berbagai macam kondisi yang mungkin, maka ini adalah cara yang baik mempertimbangkan bahwa objek – objek dalam sebuah kelas mempunyai perilaku dinamik.
- *Relasi* : cara berikutnya adalah memeriksa relasi antar kelas. Perhatikan *multiplicity* dalam sebuah relasi nol. Nol menunjukkan bahwa ada objek tertentu yang tidak terlibat dalam relasi.

Contoh *statechart diagram* :



Gambar 2.9 contoh diagram statechart

Definisi Web

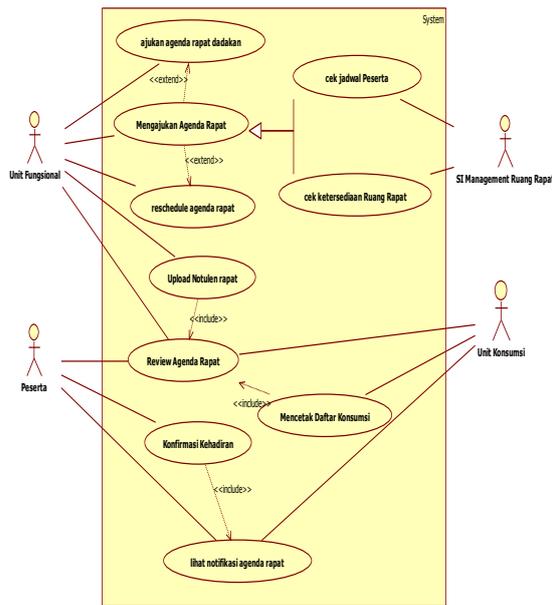
Secara terminologi, *website* adalah kumpulan dari halaman - halaman situs, yang terangkum dalam sebuah *domain* atau *subdomain*, yang tempatnya berada di dalam *World Wide Web* (*WWW*) di dalam Internet. Sebuah halaman web yang berupa dokumen yang ditulis dalam format *HTML* (*Hyper Text Markup Language*), yang diakses melalui *HTTP*, yaitu sebuah protokol yang menyampaikan informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser* (Andri Kristianto, 2008).

ANALISA DAN PERANCANGAN

Analisa Sistem

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. Analisa sistem dalam pembuatan aplikasi management ruang rapat menggunakan *usecase*, *Sequential diagram*.

Use Case Diagram



Gambar 3.1 Use Case SI Manajemen Rapat

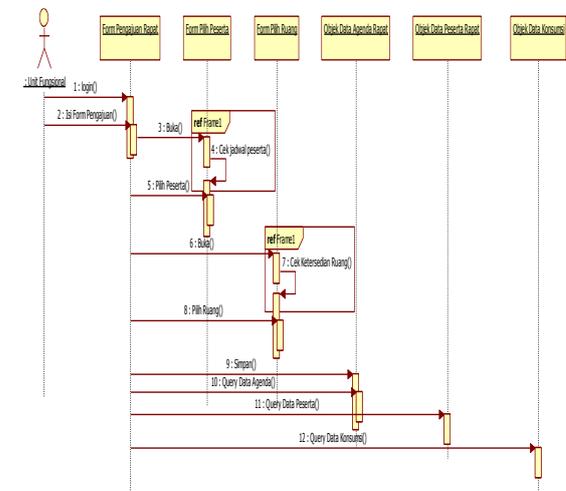
Usecase diagram menyajikan interaksi antara *actor* dan *usecase*. dimana *actor* dapat berupa orang, peralatan atau system lain yang berinteraksi dengan *system* yang sedang dibangun. Sedangkan *usecase* menggambarkan fungsionalitas *system* atau syarat – syarat yang harus dipenuhi system dari pandangan pemakai.

Pada *system* Manajemen rapat ini terdapat 4 aktor yaitu unit fungsional, unit konsumsi, peserta dan SI management ruang rapat . Aktor unit fungsional menggunakan beberapa *usecase* diantaranya mengajukan agenda rapat, review jadwal rapat, *upload* notulen rapat dan *reschedule* agenda rapat. terdapat relasi generalisasi antara *usecase* mengajukan agenda rapat dengan *usecase* cek jadwal peserta rapat dan *usecase* cek ketersediaan ruang rapat. dimana *usecase* cek jadwal peserta rapat dan *usecase* cek ketersediaan ruang rapat menjadi dasar atau bagian dari *usecase* mengajukan agenda rapat. terdapat relasi *extend* antara *usecase* mengajukan agenda rapat dengan *usecase* *reschedule* agenda rapat, yang berarti ketika *usecase* mengajukan agenda rapat sedang berjalan *usecase* *reschedule* agenda rapat bisa berjalan jika diinginkan oleh actor unit fungsional. *Usecase* mengajukan agenda rapat juga mempunyai relasi *extend* dengan ajukan agenda rapat dadakan. Terdapat juga relasi *include* antara *usecase* review agenda rapat dengan *upload* notulen rapat, yang berarti apabila aktor unit fungsional menjalankan *usecase* *upload* notulen rapat maka *usecase* review agenda rapat juga harus dijalankan. Aktor unit konsumsi mempunyai relasi dengan *usecase* review

jadwal rapat dan *usecase* mencetak daftar konsumsi. Aktor peserta mempunyai relasi dengan *usecase* review agenda rapat , *usecase* konfirmasi kehadiran dan *usecase* lihat notifikasi agenda rapat. *usecase* konfirmasi kehadiran mempunyai relasi *include* dengan *usecase* konfirmasi kehadiran.

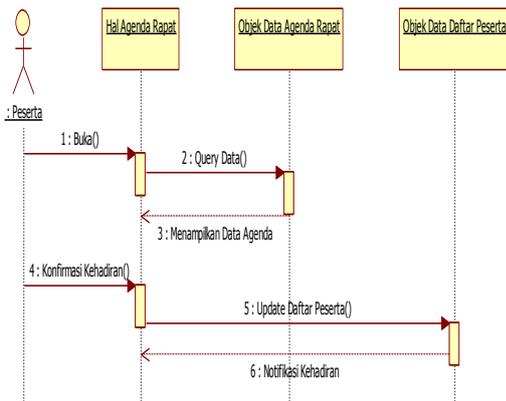
Diagram Sequential

Diagram sekuensial atau *sequence* diagram digunakan untuk menunjukkan aliran fungsionalitas dalam *usecase*. Diagram ini memperlihatkan interaksi yang menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu.



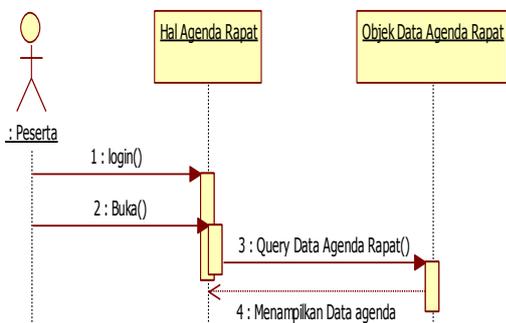
Gambar 3.2 Diagram sekuensial pengajuan rapat

Pada gambar 3.2 dijelaskan unit fungsional melakukan input pada objek *form* pengajuan rapat. form pengajuan rapat ini menjadi *interface* bagi actor unit fungsional. Selanjutnya actor akan memilih daftar peserta rapat diobjek *form* pilih peserta. Diobjek *form* pilih peserta ini akan dilakukan proses cek jadwal peserta, untuk mengetahui peserta mana yang bisa diundang sesuai dengan tanggal pengajuan rapat. setelah proses memilih peserta actor akan memilih daftar ruang rapat. diobjek *form* pilih ruang akan dilakukan cek ketersediaan ruang rapat. apabila semua proses selesai objek *form* pengajuan rapat akan melakukan *query* simpan ke objek data agenda rapat, data peserta rapat dan konsumsi untuk menyimpan data kedalam *database*.



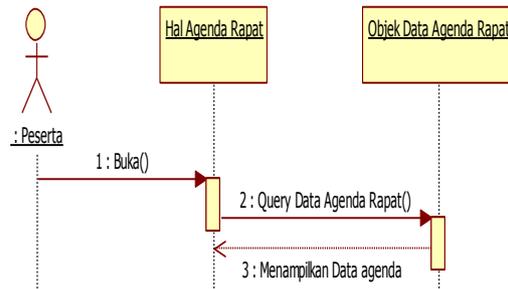
Gambar 3.3 Diagram sekuensial konfirmasi kehadiran

Pada gambar 3.3 dijelaskan peserta mengirimkan *message* buka ke objek halagendarapat untuk melihat daftar agenda, kemudian objek data agenda rapat menampilkan daftar rapat. setelah daftar agenda rapat ditampilkan peserta akan melakukan konfirmasi kehadiran. Objek data daftar peserta akan melakukan *query update* daftar peserta data kehadiran dan setelah itu objek data daftar peserta melakukan notifikasi berhasilnya penyimpanan data.



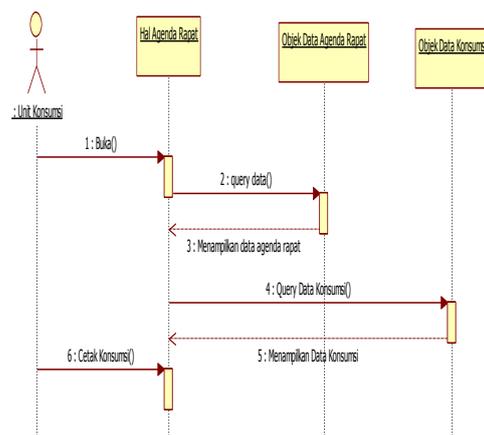
Gambar 3.4 Diagram sekuensial melihat jadwal kegiatan

Pada gambar 3.4 dijelaskan bahwa peserta membuka halaman jadwal kegiatan. Kemudian *query* data dilakukan dan diterima oleh objek data jadwal kegiatan. setelah *query* diterima maka data jadwal kegiatan akan ditampilkan di halaman jadwal kegiatan



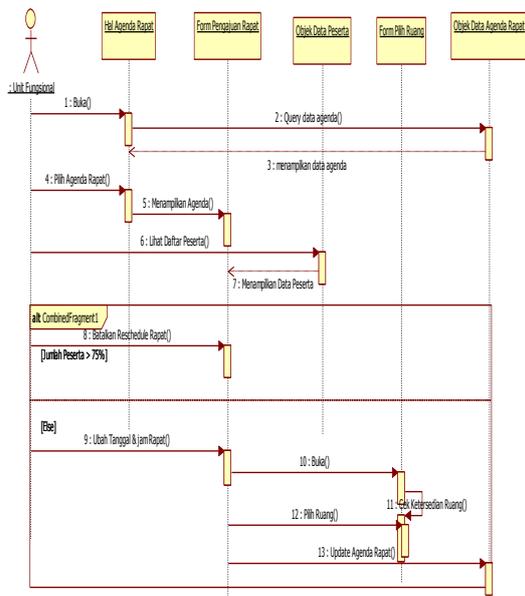
Gambar 3.5 Diagram sekuensial review jadwal rapat

Pada gambar 3.5 dijelaskan bahwa peserta membuka halaman agenda. Kemudian *query* data dilakukan dan diterima oleh objek data agenda rapat. setelah *query* diterima maka data agenda rapat akan ditampilkan di halaman agenda rapat.



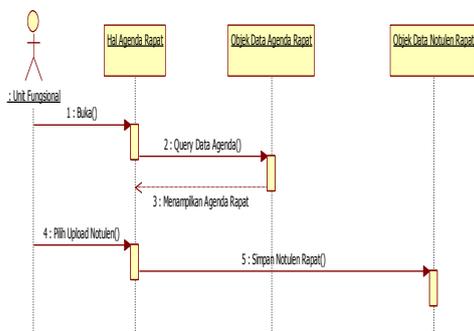
Gambar 3.6 Diagram sekuensial mencetak daftar konsumsi

Pada gambar 3.6 dijelaskan bahwa unit konsumsi membuka halaman agenda. Kemudian *query* data dilakukan dan diterima oleh objek data agenda rapat. setelah *query* diterima maka data agenda akan ditampilkan di halaman agenda. Kemudian unit konsumsi melihat daftar konsumsi dan mencetak daftar konsumsi.



Gambar 3.7 Diagram sekuensial reschedule jadwal agenda

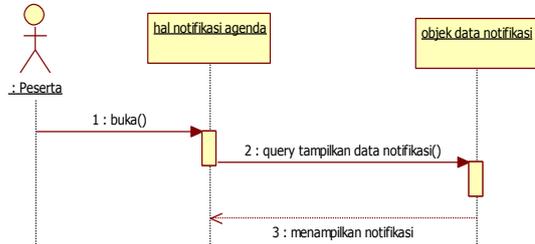
Pada gambar 3.7 dijelaskan bahwa unit fungsional membuka halaman agenda rapat. setelah itu data agenda rapat akan ditampilkan. Untuk menjadwalkan ulang agenda rapat unit fungsional memilih salah satu agenda rapat dan data agenda rapat tersebut akan ditampilkan di form pengajuan rapat. setelah data agenda ditampilkan unit fungsional akan melihat daftar peserta. Apabila kehadiran peserta sudah diatas 75% maka reschedule akan dibatalkan. Tetapi apabila daftar kehadiran peserta rapat kurang dari 75% maka unit fungsional akan menata ulang jadwal agenda rapat. Selanjutnya proses update dijalankan untuk menyimpan data agenda ke dalam basis data.



Gambar 3.8 Diagram sekuensial upload notulen rapat

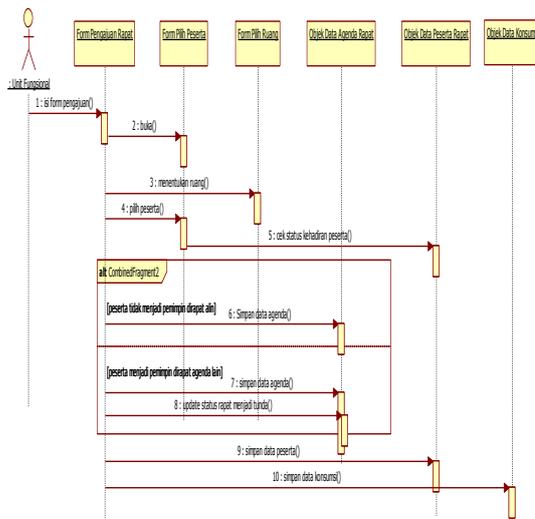
Pada gambar 3.8 dijelaskan bahwa unit fungsional membuka halaman agenda rapat. setelah

itu data agenda rapat akan ditampilkan. Selanjutnya unit fungsional memilih upload notulen dan akan dilakukan query simpan notulen rapat ke objek data notulen rapat.



Gambar 3.9 Diagram sekuensial lihat notifikasi agenda rapat

Pada gambar 3.9 dijelaskan bahwa peserta membuka halaman notifikasi agenda. Kemudian query data dilakukan dan diterima oleh objek data notifikasi. setelah query diterima maka data notifikasi agenda rapat akan ditampilkan di halaman notifikasi agenda.



Gambar 3.10 Diagram sekuensial ajukan agenda rapat dadakan

Pada gambar 3.10 dijelaskan unit fungsional melakukan input pada objek form pengajuan rapat. form pengajuan rapat ini menjadi interface bagi actor unit fungsional. Selanjutnya actor akan memilih daftar peserta rapat diobjek form pilih peserta. setelah proses memilih peserta actor akan memilih daftar ruang rapat. diobjek form pilih ruang akan dilakukan cek ketersediaan ruang rapat yang bisa dipakai untuk melakukan rapat dadakan. apabila semua proses selesai objek form pengajuan

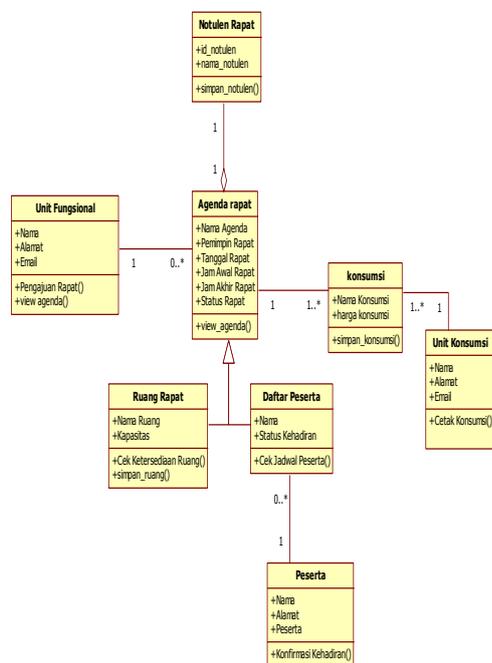
rapat akan melakukan query simpan ke objek data agenda rapat, data peserta rapat dan konsumsi untuk menyimpan data kedalam database. Dan agenda rapat yang pemimpinnnya diundang untuk menghadiri rapat dadakan, oleh sistem statusnya akan diubah menjadi tunda.

Perancangan Sistem

Perancangan sistem dibangun untuk memenuhi kebutuhan user mengenai gambaran yang jelas tentang perancangan sistem yang akan dibangun serta diimplementasikan. Perancangan sistem yang digunakan dalam membangun aplikasi management ruang rapat adalah 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 . Disamping itu diagram class juga memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari class – class yang ada dan relasinya satu dengan yang lain.



Gambar 3.12 Diagram class manajemen rapat

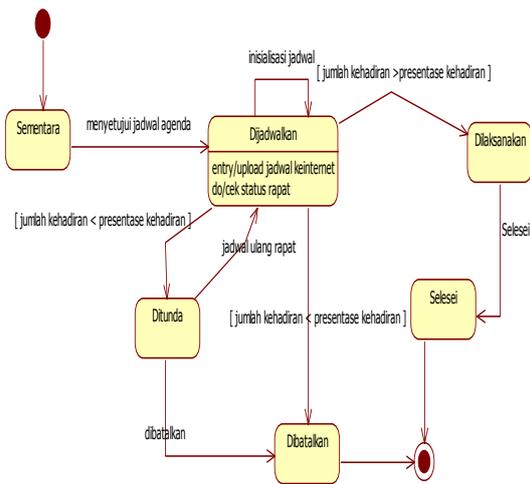
Diagram class dibentuk dalam bentuk bujur sangkar yang terbagi dalam 3 bagian. Bagian pertama menunjukkan nama kelas. Bagian kedua

menunjukkan attribute yang memuat informasi misalnya class Objek form pengajuan rapat mempunyai attribute agenda rapat, tanggal rapat. Bagian ketiga menunjukkan operasi dari sebuah class . dimana operasi adalah tingkah laku yang disediakan oleh class.

Pada gambar 3.12 diatas terlihat bahwa agenda rapat harus berasal dari satu unit fungsional. Tetapi unit fungsional bisa mengajukan banyak agenda rapat. peserta tidak harus berasosiasi dengan daftar peserta, tetapi daftar peserta harus terkait dengan satu peserta. Peserta tidak harus berasosiasi dengan jadwal kegiatan . Ruang rapat dan daftar peserta generalisasi dengan agenda rapat karena Ruang rapat dan daftar peserta mewarisi atribut dan operasi yang didefinisikan secara public oleh agenda rapat . notulen rapat diintegrasikan dengan agenda rapat karena notulen rapat merupakan bagian dari agenda rapat.

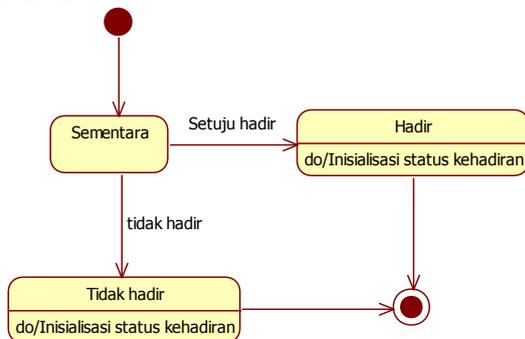
Statechart Diagram

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. Pada diagram class yang telah dibuat diatas terdapat dua kelas yang memiliki perilaku dinamis yaitu class agenda rapat dan daftar peserta. Perilaku dinamik kedua class tersebut di tunjukkan dengan attribut yang mempunyai nilai berbeda yang digunakan untuk menyimpan berbagai macam kondisi. Pada class agenda rapat terdapat attribut status rapat yang dapat menyatakan beberapa kondisi diantaranya sementara, dijadwalkan, dilaksanakan, ditunda, dibatalkan dan selesai. Sementara pada class daftar peserta attribut yang bisa menyimpan beberapa kondisi adalah status kehadiran. Kondisi tersebut diantaranya sementara, hadir dan tidak hadir.



Gambar 3.13 Diagram statechart untuk class agenda rapat

Pada gambar 3.13 diatas dapat dijelaskan suatu agenda rapat dapat dimulai dikeadaan / state sementara. Jadwal agenda disetujui agenda rapat bergerak ke kondisi dijadwalkan. Pada kondisi ini apabila jumlah kehadiran rapat lebih dari presentase kehadiran maka kondisi menjadi dilaksanakan. Tetapi apabila jumlah kehadiran kurang dari presentase kehadiran maka kondisi akan menjadi ditunda dan agenda rapat bisa direschedule atau dibatalkan. Agenda rapat menjadi selesai setelah rapat selesai dilaksanakan dan status rapat menjadi selesai.



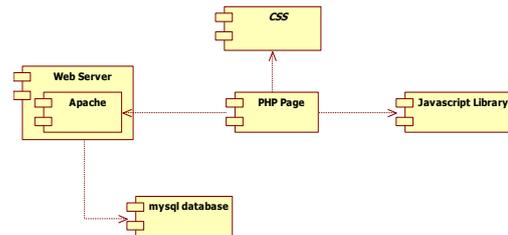
Gambar 3.14 Diagram statechart untuk class daftar peserta

Pada gambar 3.14 diatas dapat dijelaskan suatu agenda rapat dapat dimulai dikeadaan / state sementara. Apabila seseorang setuju untuk hadir maka state / kondisi akan bergerak ke kondisi hadir dan status kehadiran menjadi hadir. Tetapi apabila seseorang meminta tidak hadir maka state akan bergerak ke kondisi tidak hadir dan status kehadiran menjadi tidak hadir.

Komponen Diagram

Terdapat beberapa komponen penyusun system informasi manajemen rapat. komponen tersebut diantaranya class css yang menjadi komponen untuk memperindah interface. Class php page sebagai penyusun halaman aplikasi. Class javascript library sebagai compile javascript. Class webserver sebagai komponen yang menampilkan php page di browser. Class mysql database sebagai komponen yang bertugas untuk menyiapkan hak akses kepada banyak pengguna terhadap data – data yang diperlukan oleh pengguna dan melayani proses penyimpanan data. Class page php memiliki relasi dependensi dengan class css, class webserver dan class library javascript. Hal ini berarti class php page tidak bisa dikompilasi sebelum class web server, class css dan class javascript library selesai dikompilasi.

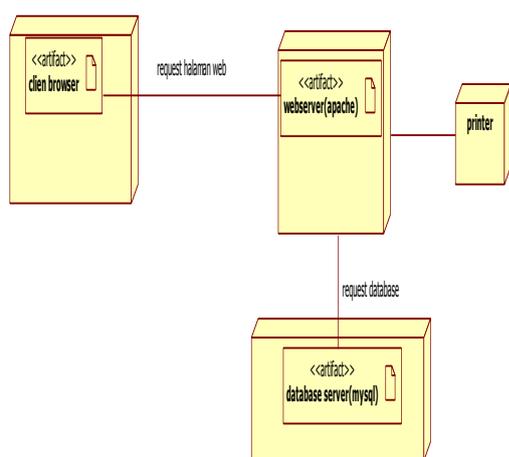
Class webserver memiliki relasi dependensi dengan class mysql database. Hal ini berarti class webserver tidak bisa dikompilasi sebelum class mysql database dikompilasi. Untuk selengkapnya dijelaskan pada gambar 3.15



Gambar 3.15 Diagram komponen manajemen rapat

Deployment Diagram

Diagram deployment menggambarkan bagaimana konfigurasi perangkat keras dimana sistem akan diinstalasikan. Diagram ini memberi simbol untuk segala bentuk perangkat keras, proses – proses dan hubungan perangkat dengan proses seperti: komputer, printer, router, switch, jaringan, dan sebagainya.

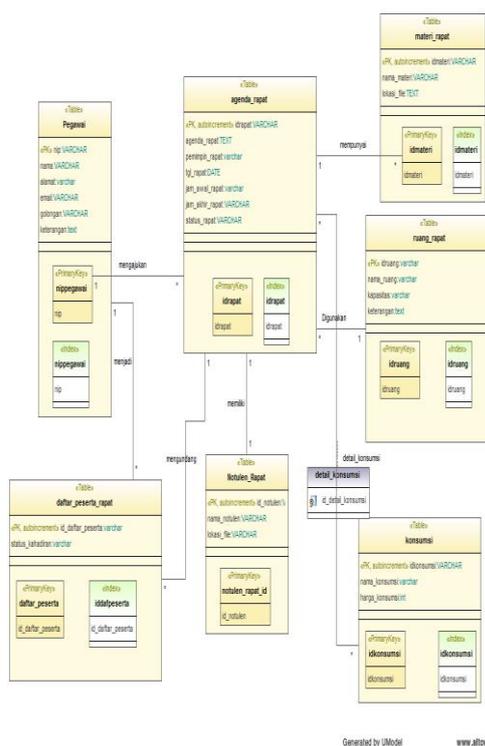


Gambar 3.16 Diagram Deployment management rapat

Client browser digunakan untuk menampilkan halaman web. Webservice menangani permintaan / request dari browser dan secara dinamis men-generate halaman web dan kode program untuk dijalankan dan ditampilkan di client. Database server bertugas untuk menyiapkan hak akses kepada banyak pengguna terhadap data – data yang diperlukan oleh pengguna dan melayani proses penyimpanan data. Sedangkan printer adalah peralatan yang digunakan untuk mencetak data sesuai dengan permintaan pengguna.

Database Diagram

Database diagram merupakan suatu model untuk menjelaskan hubungan antar data dalam **basis data** berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Gambar berikut menunjukkan bagaimana UML memodelkan sebuah aplikasi berkaitan dengan database model.



Gambar 3.17 Diagram database manajemen rapat

Pada gambar 3.17 diatas dapat dijelaskan hubungan relasinya seperti berikut :

1. tabel pegawai berasosiasi dengan tabel agenda rapat dengan nama relasi asosiasi mengajukan yang berarti pegawai bisa mengajukan banyak agenda rapat.
2. Tabel pegawai berasosiasi dengan tabel daftar peserta rapat dengan nama relasi asosiasi menjadi yang berarti pegawai bisa menjadi anggota di daftar peserta rapat.
3. Tabel agenda rapat berasosiasi dengan tabel daftar peserta rapat dengan nama relasi asosiasi mengundang yang berarti agenda rapat bisa mengundang banyak daftar peserta rapat.
4. Tabel agenda rapat berasosiasi dengan tabel notulen rapat dengan nama relasi asosiasi memiliki yang berarti agenda rapat bisa memiliki notulen rapat.
5. Tabel agenda rapat berasosiasi dengan tabel konsumsi yang berarti agenda rapat bisa memiliki banyak konsumsi dan konsumsi juga bisa digunakan di banyak agenda rapat.
6. Tabel ruang rapat berasosiasi dengan tabel agenda rapat dengan nama relasi asosiasi digunakan yang berarti ruang rapat bisa digunakan untuk banyak agenda rapat .
7. Tabel agenda rapat berasosiasi dengan tabel materi rapat dengan nama relasi asosiasi mempunyai yang berarti agenda rapat bisa mempunyai banyak materi rapat.

HASIL DAN PEMBAHASAN

Dalam menggunakan sistem informasi manajemen rapat user membutuhkan hardware dan software untuk mendukung dalam kinerja dan proses sistem yang akan dibuat agar kinerja sistem bisa maksimal.

Kebutuhan Perangkat Keras (*Hardware*)

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 : Intel(R) Pentium(R) Core IM i3-350M processor
2. RAM : 2 GB (sebaiknya lebih)
3. VGA : On-Board
4. Keyboard dan mouse

Dengan spesifikasi perangkat keras komputer yang digunakan dapat membantu dalam pengolahan data.

Kebutuhan Perangkat Lunak (*Software*)

Aplikasi sistem informasi manajemen rapat ini memerlukan perangkat lunak (*Software*) agar mampu beroperasi sesuai dengan yang diharapkan. Adapun perangkat lunak (*Software*) yang digunakan, yaitu :

- a. Sistem Operasi : windows, linux, mac os .
- b. Web browser seperti Mozilla Firefox,opera,google chrom .

Uji Coba System

Dalam melakukan pengujian sistem, yang dilakukan sesuai dengan ketentuan pengajuan rapat seperti dijelaskan pada sub bab 2.1.3 telah menghasilkan beberapa skenario meliputi:

Memberikan kemudahan dalam mengajukan rapat

Untuk mengajukan rapat, unit fungsional bisa masuk kehalaman form pengajuan rapat. Unit fungsional memasukkan mulai tanggal dan sampai tanggal. Memasukkan perkiraan kehadiran peserta dalam satuan persen. Hal ini akan memberikan output berupa jumlah daftar kehadiran peserta dalam satuan persen yang sesuai dengan keinginan unit fungsional dan system akan melakukan proses kalkulasi dengan menampilkan ruang rapat yang sesuai dengan jumlah peserta yang diundang unit fungsional. Selanjutnya unit fungsional memilih pemimpin rapat dan daftar peserta dengan cara

memilih langsung per department atau perseorangan seperti pada gambar 5.1.

Gambar 5.1 form pengajuan rapat

Untuk memperjelas proses simulasi pengajuan rapat perhatikan data agenda rapat pada gambar 5.2. Data tersebut diambil dari database manajemen rapat.

ID_RUANG	NAMA_RUANG	KAPASITAS	LOKASI
r007	Ruang Microsoft	5	lantai 4 gedung A pelindo
r005	Ruang Twitter	10	lantai 2 kantor cabang
r006	Ruang Google	15	lantai 1
r004	Ruang Facebook	20	Lantai 3 kantor cabang
r003	Ruang Wikipedia	25	lantai 4 kantor pusat

Gambar 5.2 gambar daftar ruang rapat

ID_RAPAT	RUANG_RAPAT	AGENDA_RAPAT	JENIS_KONSUMSI	PEMIMPIN_RAPAT	TGL_RAPAT
09	r005	rapat agenda 1	2	Daniel Agger	2012-11-15
010	r005	rapat agenda 2	2	Achmad Nashihuddin	2012-11-17
011	r007	rapat agenda 3	1	Siti Aisah	2012-11-18

Gambar 5.3 gambar daftar agenda rapat

Pada gambar 5.3 didapat keterangan diantaranya :

- a. Tanggal 15 november 2012
Pada tanggal 15 november 2012 di database manajemen rapat terdapat agenda rapat dengan agenda rapat agenda 1. Yang dipimpin oleh Daniel agger dan diadakan diruang twitter dengan kode ruang r005. Daftar peserta berjumlah 6 orang diantaranya Steven Gerrard, Luis Suares, Daniel Agger,Fernando Torres, Juan Mata dan John Terry.
- b. Tanggal 17 november 2012
Pada tanggal 17 november 2012 didatabase manajemen rapat terdapat agenda rapat dengan agenda rapat agenda 2. Yang dipimpin oleh

achmad nashihuddin dan diadakan diruang twitter dengan kode ruang r005. Daftar peserta berjumlah 6 orang diantaranya Lukas Podolski, Oliver Giroud, Aaron Ramsey, Siti Fatimah, Theo Walcott dan Achmad Nashihuddin.

c. Tanggal 18 november 2012

Pada tanggal 18 november 2012 didatabase manajemen rapat terdapat agenda rapat dengan agenda rapat agenda 3. Yang dipimpin oleh siti aisah dan diadakan diruang Microsoft dengan kode ruang r007. Daftar peserta berjumlah 5 orang diantaranya Fernando Torres, Juan Mata, John Terry, Gary Cahill dan Siti Aisah.

Berdasarkan data diatas sebagai percobaan unit fungsional akan mengajukan rapat dengan memasukkan mulai tanggal 15-11-2012, sampai tanggal 20-11-2012, perkiraan kehadiran peserta diatas 80 %, pemimpin rapat Achmad Nashihuddin dan daftar peserta diantaranya Achmad Nashihuddin, Glen Johnson, Steven Gerrard, Luis Suares dan Daniel Agger seperti terlihat pada gambar 5.4.

No	Nama Peserta	Jabatan	Hapus Peserta
1	Daniel Agger	CEO	Hapus
2	Achmad Nashihuddin	Staff IT	Hapus
3	Glen Jonshon	Sekretaris	Hapus
4	Steven Gerrard	Dirut	Hapus
5	Luis Suarez	CEO	Hapus

Terdapat Peserta Sebanyak 5 orang

Gambar 5.4 gambar daftar peserta agenda rapat

Mulai Tanggal*: 2012-11-15 Sampai Tanggal*: 2012-11-20 Perkiraan Kehadiran Peserta Diatas*: 90 %

Pilih Peserta Per Departmen*: --Pilih Departmen-- Pilih Departmen

Pilih Peserta Per Orang: Pilih Peserta Klik Pilih Peserta untuk memilih peserta per orang

Pilih Pemimpin Rapat: Achmad Nashihuddin Pilih Pemimpin Rapat

Teknologi Informasi / (Hapus)

Proses

Tanggal yang bisa anda pilih dari tanggal 2012-11-15 sampai Tanggal 2012-11-20 dengan presentasi kehadiran >= 80 %

Peserta Yang anda undang sebanyak 5 Orang

- 16 November 2012 dengan Kehadiran sebanyak 5 orang atau 100 % Ruang Yang Bisa Anda gunakan Ruang Microsoft
- 18 November 2012 dengan Kehadiran sebanyak 5 orang atau 100 % Ruang Yang Bisa Anda gunakan Ruang Twitter
- 19 November 2012 dengan Kehadiran sebanyak 5 orang atau 100 % Ruang Yang Bisa Anda gunakan Ruang Microsoft
- 20 November 2012 dengan Kehadiran sebanyak 5 orang atau 100 % Ruang Yang Bisa Anda gunakan Ruang Microsoft

Pilih Jadwal

Gambar 5.5 gambar output hari & ruang rapat

Berdasarkan data peserta yang diinputkan seperti pada gambar 5.4 diatas diperoleh beberapa output seperti pada gambar 5.5 diantaranya.

- 16 november 2012 dengan kehadiran 100 % diruang microsoft.
- 18 november 2012 dengan kehadiran 100 % diruang twitter.
- Berdasarkan data didatabase ruang twitter memiliki kapasitas 10 orang. Seharusnya dengan peserta sebanyak 5 orang ruang yang dipakai adalah ruang microsoft yang berkapasitas 5 orang. Tetapi disebabkan pada tanggal 18 november 2012 ruang microsoft sudah digunakan, maka ruang dialihkan ke ruang twitter.
- 19 november 2012 dengan kehadiran 100 % diruang microsoft.
- 20 november 2012 dengan kehadiran 100 % diruang microsoft
- Tanggal 15 november tidak ditampilkan karena pada tanggal 15 november presentase kehadiran di bawah 80 %.
- Tanggal 17 november tidak ditampilkan karena pada tanggal 17 november pemimpin rapat telah menghadiri agenda rapat.

Dari data output di atas bisa membantu unit fungsional dalam mengambil keputusan untuk menentukan hari dan ruang agenda rapat. sebagai contoh implementasi, unit fungsional memilih tanggal 16 november 2012. Maka selanjutnya unit fungsional akan masuk kehalaman detail agenda rapat seperti pada gambar 5.6 berikut .

The screenshot shows a web form titled 'Form Pengajuan Rapat'. It includes sections for 'Waktu & Ruang Rapat' (Meeting Time & Room), 'Perengkapan & Peserta Rapat' (Meeting Equipment & Participants), and 'Detail Rapat' (Meeting Details). The 'Detail Rapat' section features a rich text editor for the agenda, with the text 'rapat perundingan anggaran belanja perusahaan tahun 2013'. Other fields include meeting date (2012-11-16), start time (12:00), end time (15:00), room (Ruang Microsoft), organizer (Achmad Nashihuddin), and a list of participants.

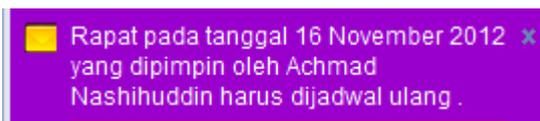
Gambar 5.6 gambar detail agenda rapat

Apabila telah selesai melengkapi semua form pada detail rapat, unit fungsional bisa mengklik simpan. Selanjutnya akan muncul konfirmasi kalau rapat telah diajukan dan data tersimpan didatabase seperti pada gambar 5.7.



Gambar 5.7 gambar konfirmasi

Setelah rapat selesai diajukan dan daftar peserta rapat tidak sesuai dengan presentase kehadiran maka unit fungsional akan menerima notifikasi reschedule seperti pada gambar 5.8 berikut .



Gambar 5.8 notifikasi reschedule

Pada gambar 5.8 untuk melihat detail notifikasi pesan, unit fungsional bisa meng-klik

pesan tersebut, maka akan muncul halaman detail notifikasi reschedule seperti pada gambar 5.9



Gambar 5.9 detail notifikasi reschedule

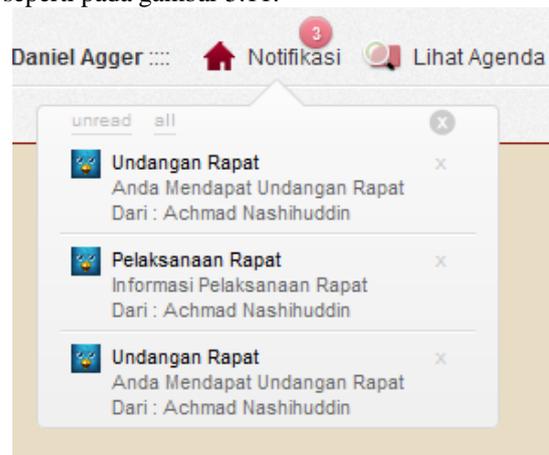
Untuk melakukan reschedule rapat, unit fungsional bisa meng-klik tombol reschedule, dan akan muncul halaman reschedule rapat seperti pada gambar 5.10

The screenshot shows a form titled 'Reschedule Jadwal Rapat'. It has fields for 'Tentukan Tanggal' (Specify Date), 'Mulai Tanggal' (Start Date), and 'Sampai Tanggal' (End Date). The organizer is listed as 'Achmad Nashihuddin'. There is a 'Proses' (Process) button and a 'Kembali' (Back) button in the top right corner.

Gambar 5.10 halaman reschedule

Memberikan informasi jadwal pengadaaan rapat

Setelah unit fungsional mengajukan agenda rapat seperti dijelaskan pada sub bab 5.2.1, maka peserta akan mendapatkan informasi pengadaaan rapat berupa notifikasi undangan rapat seperti pada gambar 5.11.



Gambar 5.11 notifikasi undangan rapat

Untuk melihat detail notifikasi user bisa mengklik link subjek pada pesan notifikasi. Maka

user akan masuk kehalaman detail notifikasi seperti pada gambar 5.12.



Gambar 5.12 detail notifikasi

Pada gambar 5.12 peserta bisa melihat data agenda rapat secara menyeluruh. Untuk menghadiri rapat peserta bisa mengklik tombol hadir dan akan keluar konfirmasi seperti pada gambar 5.13.



Gambar 5.13 konfirmasi kehadiran rapat

Tetapi, apabila tidak berkenan untuk hadir, peserta bisa mengklik tombol tidak hadir dan memberikan alasan atas ketidakhadirannya tersebut. Seperti ditampilkan pada gambar 5.14



Gambar 5.14 Alasan ketidakhadiran rapat

Mengatur jadwal peserta rapat

Pada sub bab 5.2.1 tentang kemudahan dalam mengajukan rapat, dijelaskan bahwa setiap kali unit fungsional ingin mengajukan rapat seperti pada gambar 5.1 sampai gambar 5.5, system akan selalu memberikan informasi tentang jumlah

kehadiran peserta sesuai dengan dengan tanggal yang diinputkan, seperti pada gambar 5.15 berikut.

Tanggal yang bisa anda pilih dari tanggal 2012-11-20 sampai Tanggal 2012-11-30 dengan presentasi kehadiran >= 0 %
Peserta Yang anda undang sebanyak 7 Orang
<input type="radio"/> 20 November 2012 dengan Kehadiran sebanyak 7 orang atau 100 %. Ruang Yang Bisa Anda gunakan Ruang Twitter
<input type="radio"/> 21 November 2012 dengan Kehadiran sebanyak 7 orang atau 100 %. Ruang Yang Bisa Anda gunakan Ruang Twitter
<input type="radio"/> 22 November 2012 dengan Kehadiran sebanyak 7 orang atau 100 %. Ruang Yang Bisa Anda gunakan Ruang Twitter
<input type="radio"/> 23 November 2012 dengan Kehadiran sebanyak 7 orang atau 100 %. Ruang Yang Bisa Anda gunakan Ruang Twitter
<input type="radio"/> 24 November 2012 dengan Kehadiran sebanyak 7 orang atau 100 %. Ruang Yang Bisa Anda gunakan Ruang Twitter
<input type="radio"/> 25 November 2012 dengan Kehadiran sebanyak 2 orang atau 28.57 %. Ruang Yang Bisa Anda gunakan Ruang Twitter
<input type="radio"/> 26 November 2012 dengan Kehadiran sebanyak 6 orang atau 85.71 %. Ruang Yang Bisa Anda gunakan Ruang Twitter
<input type="radio"/> 27 November 2012 dengan Kehadiran sebanyak 7 orang atau 100 %. Ruang Yang Bisa Anda gunakan Ruang Twitter
<input type="radio"/> 28 November 2012 dengan Kehadiran sebanyak 2 orang atau 28.57 %. Ruang Yang Bisa Anda gunakan Ruang Twitter
<input type="radio"/> 29 November 2012 dengan Kehadiran sebanyak 2 orang atau 28.57 %. Ruang Yang Bisa Anda gunakan Ruang Microsoft
<input type="radio"/> 30 November 2012 dengan Kehadiran sebanyak 7 orang atau 100 %. Ruang Yang Bisa Anda gunakan Ruang Twitter

Gambar 5.15 informasi jadwal kehadiran peserta

Mengatur jadwal ketersediaan ruang rapat

Pada gambar 5.16 dijelaskan bahwa setiap kali unit fungsional ingin mengajukan rapat seperti pada gambar 5.1 sampai gambar 5.5 pada sub bab 5.2.1 tentang kemudahan dalam mengajukan rapat, system akan selalu memberikan informasi tentang ketersediaan ruang rapat. kalau ruang pada tanggal yang diinputkan sudah dipakai maka sistem akan mengalihkan keruang yang tidak dipakai. Selain itu, ruang rapat akan otomatis disesuaikan dengan jumlah kehadiran peserta.

Tanggal yang bisa anda pilih dari tanggal 2012-11-20 sampai Tanggal 2012-11-30 dengan presentasi kehadiran >= 0 %
Peserta Yang anda undang sebanyak 7 Orang
<input type="radio"/> 20 November 2012 dengan Kehadiran sebanyak 7 orang atau 100 %. Ruang Yang Bisa Anda gunakan Ruang Twitter
<input type="radio"/> 21 November 2012 dengan Kehadiran sebanyak 7 orang atau 100 %. Ruang Yang Bisa Anda gunakan Ruang Twitter
<input type="radio"/> 22 November 2012 dengan Kehadiran sebanyak 7 orang atau 100 %. Ruang Yang Bisa Anda gunakan Ruang Twitter
<input type="radio"/> 23 November 2012 dengan Kehadiran sebanyak 7 orang atau 100 %. Ruang Yang Bisa Anda gunakan Ruang Twitter
<input type="radio"/> 24 November 2012 dengan Kehadiran sebanyak 7 orang atau 100 %. Ruang Yang Bisa Anda gunakan Ruang Twitter
<input type="radio"/> 25 November 2012 dengan Kehadiran sebanyak 2 orang atau 28.57 %. Ruang Yang Bisa Anda gunakan Ruang Twitter
<input type="radio"/> 26 November 2012 dengan Kehadiran sebanyak 6 orang atau 85.71 %. Ruang Yang Bisa Anda gunakan Ruang Twitter
<input type="radio"/> 27 November 2012 dengan Kehadiran sebanyak 7 orang atau 100 %. Ruang Yang Bisa Anda gunakan Ruang Twitter
<input type="radio"/> 28 November 2012 dengan Kehadiran sebanyak 2 orang atau 28.57 %. Ruang Yang Bisa Anda gunakan Ruang Twitter
<input type="radio"/> 29 November 2012 dengan Kehadiran sebanyak 2 orang atau 28.57 %. Ruang Yang Bisa Anda gunakan Ruang Microsoft
<input type="radio"/> 30 November 2012 dengan Kehadiran sebanyak 7 orang atau 100 %. Ruang Yang Bisa Anda gunakan Ruang Twitter

Gambar 5.16 informasi ketersediaan ruang rapat

Memberikan informasi kehadiran peserta rapat

Setelah peserta mendapatkan notifikasi undangan rapat seperti pada gambar 5.8 , maka peserta akan melakukan konfirmasi kehadiran maupun ketidakhadiran. Data tersebut oleh system akan diinformasikan ke unit fungsional, sehingga unit fungsional mengetahui siapa yang hadir dan siapa yang tidak hadir. Hal ini sangat penting sebagai dasar pengambil keputusan bagi unit fungsional, apakah ingin melaksanakan rapat atau

menjadwal ulang rapat. hal ini ditampilkan seperti pada gambar 5.17.



Nik	Nama Karyawan	Jabatan	Kehadiran	Alasan Tidak Hadir
0915	Daniel Agger	CEO	Hadir	-
0911	Achmad Hashihuddin	Staff IT	Hadir	-
0912	Glen Jonshon	Sekretaris	Hadir	-
0913	Steven Gerrard	Dirut	Hadir	-
0914	Luis Suarez	CEO	Hadir	-

Gambar 5.17 informasi kehadiran peserta rapat

Memberikan informasi tentang konsumsi untuk rapat

Dalam mengajukan agenda rapat seperti pada gambar 5.6 unit fungsional akan menentukan konsumsi yang dibutuhkan dalam rapat. Data konsumsi yang dimasukkan akan diinformasikan ke unit konsumsi berupa notifikasi permintaan konsumsi. Unit konsumsi bertugas untuk menyediakan konsumsi. Seperti ditampilkan pada gambar 5.18.



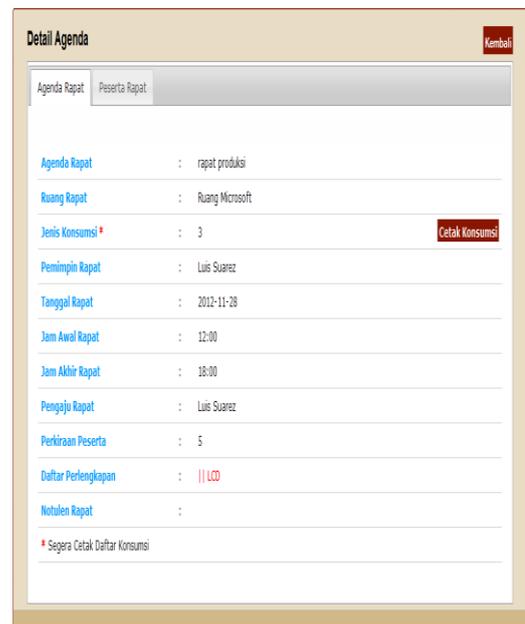
Gambar 5.18 notifikasi permintaan konsumsi

Untuk melihat detail notifikasi user bisa mengklik link subjek notifikasi. Selanjutnya akan masuk kehalaman detail notifikasi permintaan konsumsi seperti pada gambar 5.19.



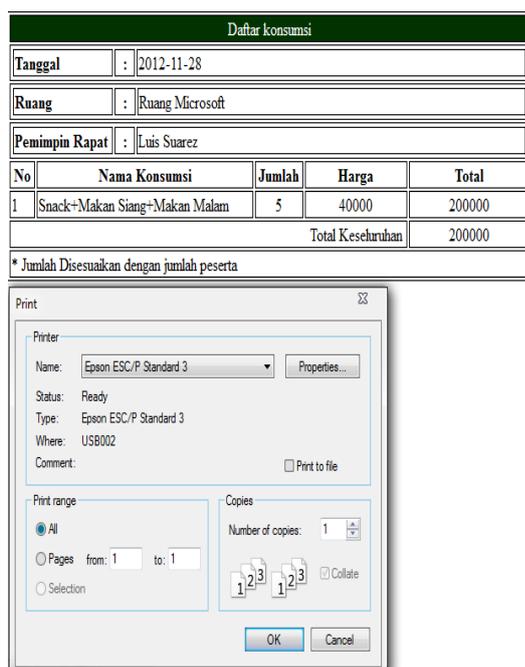
Gambar 5.19 halaman detail notifikasi permintaan konsumsi

Untuk melihat detail agenda, unit konsumsi bisa mengklik tombol lihat agenda. Selanjutnya akan masuk kehalaman detail agenda seperti pada gambar 5.20.



Gambar 5.20 halaman detail agenda

Untuk mencetak daftar konsumsi pada gambar 5.20 unit konsumsi bisa mengklik tombol cetak konsumsi seperti ditampilkan pada gambar 5.21.



Gambar 5.21 cetak daftar konsumsi

Memberikan informasi tentang notulen rapat

Setelah rapat selesai dilaksanakan, maka unit fungsional akan meng upload file notulen rapat. hasil upload notulen bisa didownload oleh peserta seperti pada gambar 5.22.

Notulen Rapat : || Tutorial Membuat Game dengan
Unity.pdf

Tambah Notulen

Upload Notulen Rapat (.doc / .pdf / .zip / .ppt)

Browse

Gambar 5.22 notulen rapat

KESIMPULAN DAN SARAN

Kesimpulan

Berdasarkan uraian pembahasan analisa dan pengujian yang telah dilakukan, maka dapat diambil kesimpulan terhadap system informasi manajemen rapat yaitu sebagai berikut :

1. Bisa memberikan kemudahan kepada unit fungsional dalam mengajukan agenda rapat
2. Bisa mengatur jadwal peserta rapat dan mengatur ketersediaan ruang rapat.

3. Dapat memberikan kemudahan kepada unit konsumsi dalam mengelola konsumsi rapat.

Saran

Saran saran yang dapat diambil dari kesimpulan diatas adalah sebagai berikut :

1. Aplikasi ini sangat cocok apabila digunakan untuk mengelola kegiatan rapat diperusahaan yang memiliki banyak karyawan dan memiliki banyak ruang rapat digedung perusahaannya.

DAFTAR PUSTAKA

- [1] Kadir, Abdul. 2006. *Pemodelan Sistem Informasi Berorientasi Objek Dengan UML*. Andi. Yogyakarta.
- [2] Kadir, Abdul. 2008. *Dasar Pemrograman Web Dinamis dengan PHP*, Penerbit Andi, Jakarta.
- [3] Kristianto, Andri. 2008. *Perancangan Sistem Informasi dan aplikasinya, Edisi Revisi*. Gaya Media. Yogyakarta.
- [4] Munawar. 2009. *Pemodelan Visual dengan UML*. Andi. Yogyakarta.
- [5] Rambe, Samsir. 1999. *Etika Komunikasi*. Angkasa. Bandung
- [6] Sholiq, 2005, *Pemodelan Sistem Informasi Berorientasi Objek dengan UML*, Penerbit Graha Ilmu Surabaya.
- [7] Netcraft Web Server Survey, 2005.
- [8] <http://jquery.com>. diakses bulan juli 2012
- [9] Tugas Pokok Kegiatan Rapat PT. Pelindo III Surabaya

(7)

**HYBRID ARTIFICIAL BEE COLONY- DIVERSITY
SEBAGAI ALGORITMA PEMBELAJARAN PADA NEURAL NETWORK****Shofiya Syidada^{a1}, Abidatul Izzah^{b2}, Nuru Aini^{b3}**^aTeknik Informatika, Universitas Wijaya Kusuma Surabaya^bTeknik Informatika, ITS Surabayaemail : cpya12@gmail.com¹, a.za.syaifa@gmail.com², nurroo.ainee@gmail.com³**ABSTRAK**

Algoritma *Artificial Bee Colony (ABC)* merupakan algoritma evolusional yang membutuhkan sedikit kontrol parameter tetapi mampu mencapai konvergensi dengan cepat. Namun algoritma ABC bisa mengalami *konvergensi* dini jika dihadapkan pada kasus berdimensi tinggi. Algoritma *Diversity-guided Evolutionary Algorithm (DGEA)* memanfaatkan ukuran *diversitas* populasi untuk mencegah *konvergensi* dini. *Neural network (NN)* merupakan algoritma yang memiliki kemampuan pembelajaran. Salah satu metode untuk pembelajaran NN adalah algoritma optimasi ABC. Penelitian ini bertujuan untuk mengkombinasikan algoritma ABC-DGEA menjadi ABC-Diversity sebagai algoritma pembelajaran pada NN untuk melakukan klasifikasi. Performa pembelajaran NN dengan ABC-Diversity akan dibandingkan dengan pembelajaran NN dengan ABC serta Backpropagasi. Hasil akurasi klasifikasi dengan pembelajaran algoritma ABC-Diversity pada dataset XOR, Iris dan Wine berturut-turut adalah 83%, 94%, dan 74%, pembelajaran dengan algoritma ABC pada dataset yang sama adalah 77%, 88%, dan 54%, sedangkan nilai akurasi dengan menggunakan pembelajaran backpropagasi adalah 59%, 77%, dan 23%.

Kata kunci : *Artificial Bee Colony, Diversity, Klasifikasi, Neural Network*

I. Pendahuluan

Algoritma ABC adalah algoritma *swarm* yang diinspirasi dari perilaku lebah madu. Algoritma ini mempunyai banyak kelebihan diantaranya struktur yang sederhana, mudah digunakan, kemampuan untuk mencapai konvergensi dengan cepat serta dapat diaplikasikan untuk memecahkan masalah optimasi yang kompleks. Algoritma ABC menunjukkan performa yang lebih baik jika dibandingkan dengan *Genetic Algorithm (GA)* dan PSO pada kasus optimasi fungsi meskipun ABC hanya membutuhkan sedikit kontrol parameter (Karaboga dan Akay, 2009).

Lee (2011) telah mengusulkan kombinasi algoritma ABC dengan strategi diversity algoritma DGEA (*Diversity Guided Evolutionary Algorithm*) serta menguji performa algoritma ini agar tidak mencapai konvergensi dini dan terjebak pada local minimum dengan menggunakan fungsi *benchmark*. DGEA memanfaatkan ukuran diversity dari suatu populasi. Strategi ini digunakan pada algoritma

evolusional dalam menentukan fase eksplorasi dan eksploitasi.

Neural Network (NN) telah banyak dimanfaatkan untuk berbagai aplikasi analisis data, deteksi kesalahan, aplikasi pengambilan keputusan, dan klasifikasi. NN mempunyai kemampuan untuk melakukan pembelajaran dengan memetakan data input terhadap data output atau targetnya. Berbagai varian NN banyak bermunculan yang meliputi model struktur jaringan, metode pembelajaran yang digunakan dan kombinasi dengan berbagai algoritma yang lain.

Salah satu metode pembelajaran NN yang populer adalah Backpropagasi (BP). BP mampu memetakan fungsi non-linier dengan derajat akurasi sesuai yang diharapkan (Irani dan Nasimi, 2011). Namun BP memiliki beberapa kekurangan antara lain BP adalah mudah terjebak pada lokal minimum sehingga kesulitan dalam menemukan solusi global dan kecepatan konvergensi yang lambat (Zhang, 2007).

NN dengan struktur *feed forward* menggunakan algoritma *Artificial Bee Colony* (ABC) untuk mempercepat konvergensi dari pembelajaran BP (Jin dan Shu, 2013). Pembelajaran NN dengan ABC-BP untuk mendapatkan bobot terbaik sehingga bisa melakukan generalisasi lebih baik (Qiongshuai dan Shiqing, 2011). Selain itu telah dimanfaatkan pula algoritma ABC yang dikombinasikan dengan *Levenberg-Marquardt* (LM) sebagai algoritma pembelajaran pada NN. Algoritma *hybrid* ini digunakan sebagai pembelajaran bobot pada *Feed forward Neural Network* (FNN) (Karaboga, 2011).

Pembelajaran NN dengan menggunakan ABC mempunyai dua keuntungan yaitu dapat memetakan input dengan output dari permasalahan non-linier atau yang disebut dengan *supervised*. Kedua, cepat dalam konvergensi serta generalisasi yang lebih baik. Namun, pembelajaran dengan algoritma ABC belum optimal dalam memecahkan permasalahan berdimensi tinggi karena dapat mengalami konvergensi dini (Yan, 2012).

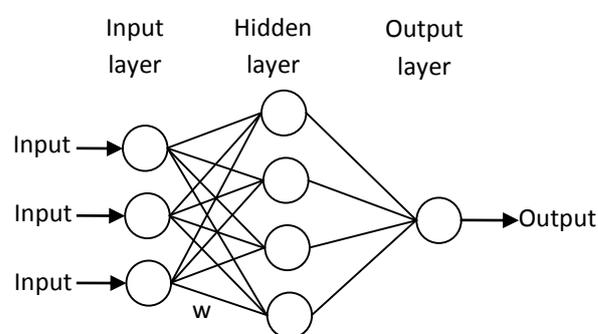
Makalah ini mencoba mengkombinasikan kelebihan algoritma ABC dan DGEA menjadi algoritma ABC-Diversity sebagai algoritma pembelajaran pada NN untuk melakukan klasifikasi. Dengan adanya perhitungan ukuran diversitas diharapkan ABC-Diversity ini mampu memperbaiki kinerja ABC untuk menghindari konvergensi dini pada dataset yang memiliki dimensi tinggi.

II. Neural Network

NN terinspirasi dari susunan sel saraf (neuron) manusia. NN merupakan algoritma *supervised learning* yang memetakan data input terhadap data output atau targetnya. NN mempunyai arsitektur *parallel processing* yang mampu memodelkan permasalahan non linier kompleks. Struktur NN terdiri dari lapisan input, lapisan hidden, dan lapisan output. Masing-masing node antar lapisan dihubungkan dengan bobot dan dipengaruhi oleh bias. Salah satu kelebihan NN adalah mampu memodelkan permasalahan non linier kompleks yang sulit dipecahkan dengan menggunakan persamaan matematis biasa (Jin dan Shu, 2013). Struktur jaringan NN dapat dilihat pada Gambar 1.

III. Artificial Bee Colony

Algoritma *Artificial Bee Colony* (ABC) merupakan simulasi perilaku kawanan lebah madu pada saat mencari makan. Pada algoritma ABC, sumber makanan lebah merepresentasikan sebuah solusi permasalahan dan jumlah nektar dari sumber makanan merepresentasikan nilai *fitness* dari solusi yang berkaitan. Koloni lebah terdiri dari tiga kelompok yaitu *employed bee*, *onlookers bee* dan *scouts bee*. *Employed bee* bertugas untuk menghitung jumlah nektar yang terdapat pada masing-masing sumber makanan, *onlookers bee* bertugas untuk memilih sumber makanan yang akan diambil nektarnya, sedangkan *scout bee* bertugas untuk mencari sumber makanan baru.



Gambar 1. Contoh Struktur Neural Network

Karaboga dan Akay (2009) melakukan beberapa tahap untuk menerapkan algoritma ABC. Bila diketahui SN adalah jumlah solusi yang tersedia, maka tahap pertama adalah dengan melakukan inialisasi sejumlah n sumber makanan, yang dinyatakan dengan x_i dimana $i = 1, \dots, SN$ dan n adalah jumlah populasi. Kemudian melakukan evaluasi nilai fitness dari populasi.

Langkah yang kedua adalah melakukan perulangan sampai ditemukan parameter yang sesuai untuk menghentikannya. Selama perulangan ini akan dicari solusi baru v_{ij} dari x_{ij} , dari tetangga yang berdekatan. Posisi solusi baru ini akan diubah menggunakan persamaan (1):

$$v_{ij} = x_{ij} + \alpha(x_{ij} - x_{kj}), \quad (1)$$

dimana $k = 1, \dots, n$ dan $j = 1, \dots, D$ adalah indeks yang dipilih secara random dan berbeda dengan nilai i , α nilai random yang dihasilkan secara

adaptif. Parameter ini yang mengontrol pencarian solusi baru di sekitar tetangga x_{ij} . Kemudian dilakukan evaluasi nilai fitness dengan memilih dan membandingkan jumlah makanan di posisi x_{ij} dan v_{ij} menggunakan persamaan (2):

$$P_i = \begin{cases} v_i, & \text{jika } (f(x_i) \geq f(v_i)) \\ x_i, & \text{jika } (f(x_i) \leq f(v_i)). \end{cases} \quad (2)$$

Untuk memilih posisi jumlah makanan yang lebih banyak yang ditemukan *onlooker* O_b didefinisikan sebagai berikut :

$$O_b = \arg_{P_i} \min f(P_i) \quad 1 \leq i \leq n, \quad (3)$$

dimana P_i adalah nilai fitness yang paling baik dari solusi i dan n adalah jumlah sumber makanan yang tersedia.

Langkah selanjutnya adalah menentukan solusi baru dari *scout* menggunakan persamaan (4):

$$x_i^j = x_{min}^j + rand(0,1) * (x_{max}^j - x_{min}^j), \quad (4)$$

nilai dari semua komponen vector x_i harus ada dalam rentang $[x_{min}, x_{max}]$. Dalam persamaan (4) tersebut x_{min} dan x_{max} merupakan batas atas dan batas bawah dari range pencarian x_{ij} pada tiap dimensinya.

Solusi terbaik yang telah ditemukan disimpan. Perulangan tersebut akan berakhir setelah solusi optimal ditemukan dari pencarian sudah dilakukan.

IV. Algoritma DGEA

DGEA merupakan algoritma yang digunakan untuk memperbaiki performa dari algoritma evolusi untuk mengetahui kondisi kekonvergenan (Wei, 2011). Strategi ini digunakan untuk menepis anggapan bahwa diversity atau keberagaman data populasi cenderung menyebabkan terjadinya konvergensi yang terlalu cepat dan terjebak pada lokal optimum. DGEA melakukan pengukuran diversity data untuk memilih fase eksplorasi atau eksploitasi dalam algoritma ABC. Ukuran *diversity*

data diperoleh dengan menggunakan persamaan (5) berikut:

$$diversity(P) = \frac{1}{|L||P|} \sum_{i=1}^{|P|} \sqrt{\sum_{j=1}^N (s_{ij} - \bar{s}_j)^2} \quad (5)$$

dimana P menyatakan populasi data, N adalah dimensi permasalahan, $|L|$ adalah panjang diagonal pada ruang pencarian dengan $S \subseteq \mathbb{R}^N$, dimana tiap variabel yang dicari x_k dalam range $x_{min} < x_k < x_{max}$, s_{ij} adalah nilai ke- j dari individu ke- i , dan \bar{s}_j adalah nilai rata-rata \bar{s} yang ke- j .

Menurut Ursen (2002), jika d_{high} merupakan suatu ambang batas dari ukuran diversitas, jika $P > d_{high}$ menunjukkan bahwa ukuran diversitas populasi tinggi sehingga dapat menyebabkan konvergensi yang lambat dan perlu dilakukan eksplorasi lebih lanjut. Namun jika $P < d_{high}$ menunjukkan bahwa kondisi diversitas populasi rendah sehingga dapat menyebabkan konvergensi dini dan perlu dilakukan fase eksploitasi.

V. Algoritma Hybrid ABC-Diversity

Algoritma DGEA kemudian digunakan untuk menentukan fase eksplorasi dan eksploitasi pada algoritma ABC untuk kasus berdimensi tinggi. Pada ABC, untuk menentukan ukuran diversitas populasi sumber makanan lebah sejumlah SN dapat ditentukan dengan persamaan (6) sebagai berikut:

$$diversity(P) = \frac{1}{|L||SN|} \sum_{i=1}^{|SN|} \sqrt{\sum_{j=1}^N (s_{ij} - \bar{s}_j)^2}, \quad (6)$$

Sedangkan batas diversitas dari populasi akan berkurang setiap waktu yang dinyatakan dengan persamaan (7) :

$$line(t) = \frac{MCN-C}{MCN} (w_{initial} - w_{final}) + w_{initial}, \quad (7)$$

dimana C menyatakan iterasi saat ini, MNC menyatakan jumlah iterasi maksimum atau epoch dan nilai dari $w_{initial}$ adalah 0.5 dan nilai w_{final} adalah 0.

Bila nilai diversitas (P) lebih kecil dari nilai $line(t)$ maka akan dilakukan fase eksplorasi dengan menggunakan persamaan (8). Dimana $i, r1, r2, r3 \in \{1, \dots, SN\}$ adalah indeks random

yang dipilih, dan $j \in \{1, \dots, D\}$ merupakan nilai random antara [0,1].

$$v_{ij} = x_{r1j} + F(x_{r2j} - x_{r3j}). \quad (8)$$

Sebaliknya, bila P lebih besar dari nilai $line(t)$ maka akan dilakukan fase eksploitasi. Pada fase eksploitasi untuk menentukan posisi solusi baru menggunakan persamaan (9) bila sumber makanan adalah solusi yang terbaik. Sedangkan persamaan (10) digunakan untuk mencari solusi baru yang lain. Dimana $i, k \in \{1, \dots, SN\}$, $j \in \{1, \dots, D\}$, k dan j adalah indeks yang dipilih secara random, ω adalah nilai random antara [-0.5, 0.5] dan φ nilai random antara [0,1].

$$v_{gbest} = x_{gbest} + \omega \cdot x_{gbest}, \quad (9)$$

$$v_{ij} = x_{ij} + \varphi(x_{gbest} - x_{kj}). \quad (10)$$

VI. Pembelajaran dengan ABC-Diversity

Berikut ini adalah metode yang digunakan untuk mengimplementasikan algoritma ABC-Diversity sebagai algoritma pembelajaran pada NN untuk melakukan klasifikasi. Pertama, adalah menentukan parameter-parameter yang diperlukan, antara lain jumlah lebah, jumlah neuron hidden dan nilai maksimum epoch yang diperlukan.

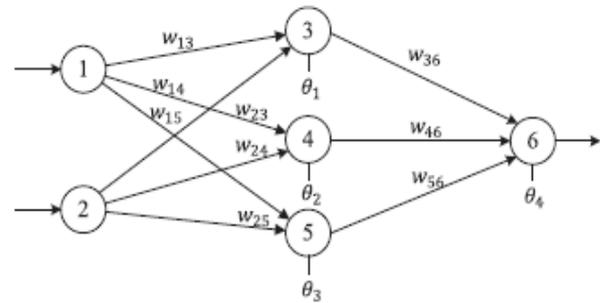
Berikutnya, adalah membagi data yang digunakan. Data yang akan digunakan untuk klasifikasi adalah data XOR, dataset Iris, dan dataset Wine. Dataset tersebut kemudian dibagi menjadi dua yaitu 90% dari total data menjadi data latih dan 10% dari total data menjadi data uji.

Tahap selanjutnya adalah melakukan normalisasi data. Data yang digunakan dinormalisasi dengan mengubah data ke dalam interval antara 0 hingga 1.

Kemudian melakukan pembelajaran NN. Pada langkah ini dilakukan beberapa prosedur yaitu membuat arsitektur jaringan NN (gambar 2), membangkitkan data populasi dengan yang merupakan inisialisasi bobot dan bias secara random, serta menggunakan populasi kombinasi bobot dan bias tersebut pada data latih dalam jaringan NN sehingga diperoleh nilai minimum error.

Pada tahap akhir akan dilakukan klasifikasi data uji menggunakan kombinasi bobot dan bias

yang dihasilkan dari pembelajaran ABC-Diversity dengan nilai error minimal.



Gambar 2. Struktur jaringan NN untuk pengkodean bobot dan bias

Pengkodean bobot dan bias yang digunakan adalah pengkodean matriks dimana $food\ source\ (i)$ menyatakan solusi ke- i , W_1 merupakan bobot dari layer $input$ ke layer $hidden$, W_2 adalah bobot dari layer $hidden$ ke layer $output$, B_1 adalah bias pada layer $input$ dan B_2 merupakan bias pada layer $hidden$. Matrik bobot dan bias direpresentasikan seperti pada persamaan (11) berikut:

$$food\ source\ (i) = [W_1\ B_1\ W_2\ B_2], \quad \text{dengan}$$

$$W_1 = \begin{bmatrix} w_{13} & w_{23} \\ w_{14} & w_{24} \\ w_{15} & w_{25} \end{bmatrix}; B_1 = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} w_{36} \\ w_{46} \\ w_{56} \end{bmatrix}; B_2 = \theta_4.$$

(11)

Fungsi aktivasi yang digunakan di hidden layer sesuai dengan persamaan (12) :

$$f(s_j) = \frac{1}{1 + e^{-\sum(w_{ij}x_{ij} - \theta_j)}}; j = 1, 2, \dots, h. \quad (12)$$

Sedangkan fungsi fitness yang digunakan adalah persamaan (13) :

$$E = \sum_{k=1}^q \frac{\sum_{i=1}^m (o_i^k - d_i^k)^2}{q}. \quad (13)$$

VII. HASIL DAN UJI COBA

A. Hasil Uji Coba

Data yang digunakan dalam uji coba ini adalah dataset XOR, iris, dan wine. Dataset XOR merupakan problem nonlinear seperti yang terlihat pada tabel 1:

Tabel 1. Tabel XOR

Input	Output
0 0	0
0 1	1
1 0	1
1 1	0

Sedangkan distribusi dataset iris dan dataset wine seperti yang tertera pada tabel 2.

Tabel 2. Distribusi kelas masing-masing dataset

Dataset	Jumlah instances	Jumlah Kelas	Distribusi kelas: class 1, class 2, class 3
XOR	40	2	15, 25
Iris	150	3	50, 50, 50
Wine	178	3	56, 71, 48

Uji coba dilakukan dengan 2 skenario. Pertama, membagi dataset menjadi dua bagian yang sama untuk data latih dan data uji. Skenario kedua dengan metode *10-fold cross validation*, yaitu membagi masing-masing dataset menjadi 10 *fold* yang sama. Kemudian kombinasi 9 *fold* yang berbeda akan digabung dan digunakan sebagai data pelatihan, sedangkan 1 *fold* yang lain digunakan sebagai data uji. Untuk dataset iris, masing-masing *fold* terdiri dari 15 instance sehingga unruk satu kali pembelajaran data latih berjumlah 135 instance, 15 instance yang tersisa digunakan untuk data uji pada. Sedangkan untuk dataset wine, 8 *fold* berjumlah 17 instance, sedang 2 *fold* terdiri dari 18 instances tersisa.

Dalam pengujian ini dilakukan 30 replikasi dengan parameter jumlah lebah = 60, jumlah neuron hidden = 2, maxEpoch = 50 untuk skenario pertama dan 100 untuk skenario kedua.

Tabel 3. Perbandingan rata-rata akurasi dan MSE untuk pengujian model

Dataset		XOR	Iris	Wine
ABCD	Akurasi (%)	81.17	94.64	76.70
	MSE	0.19	0.01	0.06
ABC	Akurasi (%)	82.00	93.43	71.07
	MSE	0.18	0.03	0.08
BP	Akurasi (%)	54.58	82.69	34.79
	MSE	0.45	0.04	0.41

Metode pembelajaran *hybrid ABC* dan *diversity* akan dibandingkan dengan metode training ABC dan backpropagation. Tabel 3 adalah rata-rata akurasi dan Mean Square Error (MSE) yang diperoleh dari pengujian performa model pada skenario pertama. Sedangkan tabel 4 menunjukkan rata-rata akurasi yang diperoleh dari pengujian performa model dengan *10-fold cross validation*.

B. Pembahasan

Dari pengamatan secara rata-rata metode training backpropagation memiliki akurasi paling rendah. Dalam pengujian data hasil 30 replikasi antara metode hybrid ABC dan Diversity dengan ABC diatas digunakan uji statistik ANOVA dengan taraf kepercayaan 95%. Dengan menggunakan software SPSS 17, diperoleh nilai p-value untuk dataset wine adalah 0.014, sedangkan XOR dan iris masing-masing 0.631 dan 0.820. Jika diambil nilai $\alpha = 0.05$, maka hal ini menunjukkan bahwa metode training ABCD dan ABC memiliki perbedaan yang signifikan untuk kasus dataset wine yang berdimensi tinggi, akan tetapi tidak begitu memiliki perbedaan untuk dataset XOR dan iris yang memiliki dimensi kecil.

Tabel 4. Perbandingan rata-rata akurasi tiga metode training untuk cross validation

Dataset	Akurasi ABCD (%)	Akurasi ABC (%)	Akurasi BP (%)
XOR	83.00	77.00	59.00
Iris	93.99	87.99	76.67
Wine	73.97	53.83	22.90

Hal ini menunjukkan bahwa ABC-Diversity mampu mengatasi kelemahan algoritma ABC untuk dataset berdimensi tinggi serta mampu memberikan nilai akurasi yang lebih baik.

V. Kesimpulan

Hasil pembelajaran menggunakan algoritma ABC-Diversity memberikan hasil akurasi yang lebih baik untuk kasus dataset yang berdimensi tinggi jika dibandingkan dengan ABC dan BP. Namun memiliki ABC-Diversity mempunyai nilai akurasi yang sama dengan ABC untuk dataset berdimensi kecil.

VI. Ucapan Terima Kasih

Ucapan terima kasih kami sampaikan kepada Bapak Dr. Agus Zainal Arifin, S.Kom, M.Kom yang telah membimbing pembuatan paper ini sampai selesai.

algorithm for feedforward neural network training, Applied Mathematics and Computation, vol. 185, pp. 1026–10

VII. Daftar Pustaka

- [1] Irani, Rasoul dan Nasimi, Reza, 2011, *Application of artificial bee colony-based neural network in bottom hole pressure prediction in underbalanced drilling*, Journal of Petroleum Science and Engineering, 78, 6-12.
- [2] Jin, F., Shu, G., 2013, *Back Propagation Neural Network Based on Artificial Bee Colony Algorithm*, 978-1-4673-1773-3/12
- [3] Karaboga, D., Akay, B., 2009, *A comparative study of Artificial Bee Colony algorithm* Applied Mathematics and Computation, 214, 108–132.
- [4] Lee, W. P., & Cai, W. T., 2011, *A novel artificial bee colony algorithm with diversity strategy*. In Natural Computation (ICNC), Seventh International Conference on (Vol. 3, pp. 1441-1444).
- [5] Qiongshuai, L., & Shiqing, W., 2011, *A hybrid model of neural network and classification in wine*. In Computer Research and Development (ICCRD), 2011 3rd International Conference on (Vol. 3, pp. 58-61).
- [6] R. Ursem, “*Diversity-guided evolutionary algorithms*,” Parallel Problem Solving from Nature—PPSN VII, pp. 462–471, 2002
- [7] Xu, Y., Fan, P., & Yuan, L. (2013), *A Simple and Efficient Artificial Bee Colony Algorithm*. Mathematical Problems in Engineering, 2013.
- [8] Yan, dkk. , 2012, *A new approach for data clustering using hybrid artificial bee colony algorithm*, Neurocomputing, 97, 241–250
- [9] Zhang, J., T. Lok, and M. Lyu, 2007, *A hybrid particle swarm optimization backpropagation*

(8)
**IMPUTASI MISSING DATA MENGGUNAKAN METODE K-NEAREST NEIGHBOUR
DENGAN OPTIMASI ALGORITMA GENETIKA**

Abidatul Izzah¹⁾ Nur Hayatin²⁾

¹⁾ Teknik Informatika ITS Surabaya

Jl. Teknik Kimia Kampus Teknik Informatika ITS, Surabaya Indonesia

email : aza.syafa@gmail.com

²⁾ Teknik Informatika Universitas Muhammadiyah Malang

Jl. Raya Tlogomas 246, Malang Indonesia

email : noorhayatin@umm.ac.id

ABSTRAK

Salah satu permasalahan yang ada pada dataset adalah ketiadaan nilai pada data untuk atribut tertentu atau yang sering disebut dengan missing data. Metode yang paling mudah dan populer adalah *K-Nearest Neighbour (KNN)*. Namun, metode ini memiliki beberapa kelemahan salah satunya adalah pemilihan nilai *k* yang tidak tepat dapat menurunkan kinerja klasifikasi. Penelitian ini bertujuan menangani missing data dengan teknik imputasi menggunakan gabungan algoritma KNN dan Algoritma Genetika (KNN-GA). GA digunakan untuk mengoptimasi nilai *k* pada KNN sehingga dapat menghasilkan nilai estimasi yang baik dengan MSE sekecil mungkin. Pengujian performansi dilakukan dengan membandingkan nilai MSE dan akurasi hasil klasifikasi antara metode KNN-GA dengan metode imputasi yang lain, yaitu : KNN, Mean, dan Median. Hasil yang diperoleh menunjukkan bahwa secara rata-rata metode imputasi KNN-GA memiliki nilai MSE terendah dan hasil akurasi klasifikasi yang tinggi.

Kata kunci : Algoritma Genetika, Imputasi, KNN, KNN-GA

1. Pendahuluan

Pada beberapa aplikasi yang ada di dunia nyata terdapat banyak sekali kasus hilangnya nilai pada dataset atau ketiadaan nilai pada data untuk atribut tertentu. Permasalahan hilangnya nilai pada data ini lebih sering disebut dengan *missing data*. Penyebab terjadinya *missing data* adalah tidak adanya respon terhadap unit maupun item, hal ini merupakan permasalahan yang terjadi pada sebagian survey skala besar (Mawarsari, 2012). Selain itu banyak juga kasus dimana ditemukannya *database* yang setengah dari dataset hilang. Sehingga sangat sulit melakukan penambahan data, dimana metode analisis yang tersedia hanya dapat bekerja dengan data yang lengkap (Farhangfari dkk, 2008). *Missing data* merupakan kelemahan umum pada banyak skenario klasifikasi pola (Laencina dkk, 2009) dan salah satu masalah yang dapat mempengaruhi hasil dari sistem prediksi data yang efektif (Malarvizhi dkk, 2012). Sehingga penanganan terhadap *missing data* sangat penting dan dibutuhkan teknik penanganan khusus untuk memperkirakan nilai data yang hilang. Terdapat 3 metode yang dapat digunakan untuk penanganan *missing data*: *Case Deletion*, *Parameter Estimation*, dan *Imputation Techniques* (Little dan Rubin, 2002). Paper ini akan membahas

imputation techniques (teknik imputasi) yakni metode penanganan *missing data* berdasarkan informasi yang tersedia pada dataset yang bertujuan untuk memprediksi nilai yang valid sebagai pengganti nilai yang hilang. *Missing data* akan menjadi masalah penting pada kasus klasifikasi dataset. Secara umum metode imputasi dan teknik klasifikasi adalah dua hal yang berbeda namun poin penting dari klasifikasi adalah bagaimana mendapatkan data pelatihan yang baik. Karena selain pemilihan metode yang tepat, akurasi hasil dari klasifikasi dipengaruhi oleh karakteristik dan kelengkapan *instance* dari sebuah data (Acuna, 2004). Sehingga dengan melakukan imputasi terhadap *missing data* maka hal tersebut secara langsung dapat mempengaruhi hasil klasifikasi. Sebaliknya, ketika sebuah kasus *missing data* diabaikan maka dapat dipastikan akan menjadikan sulit memperoleh akurasi yang tinggi untuk hasil klasifikasi walaupun digunakan algoritma klasifikasi yang paling handal sekalipun (Zhang, 2011). Beberapa penelitian menunjukkan bahwa penanganan *missing data* dengan menggunakan metode imputasi dapat meningkatkan akurasi klasifikasi dibandingkan dengan tanpa imputasi (Farhangfari dkk, 2008) (Laencina dkk, 2009).

Karena nilai yang akan diimputasikan pada *missing data* didapatkan berdasarkan estimasi, sehingga dibutuhkan pemilihan metode imputasi yang tepat agar estimasi tersebut dapat mendekati data asli. Metode yang umum dilakukan ketika terdapat *missing data* adalah dengan membuang *missing data* tersebut. Namun, metode ini dapat menghilangkan informasi-informasi penting pada data yang kemungkinan ada pada data yang dihilangkan tersebut. Pengembangan dari metode imputasi telah banyak diteliti. Beberapa metode imputasi yang populer adalah : *Mean*, *Median*, klusterisasi, dan prediksi. Imputasi dengan menggunakan metode klusterisasi melakukan imputasi *missing data* dengan cara membagi dataset menjadi dua klaster, yaitu klaster yang berisi data dengan nilai komplit dan klaster yang berisi *missing data*. Selanjutnya, klaster yang berisi data komplit digunakan untuk mendapatkan nilai estimasi dengan cara menghitung nilai *Mean* atau modus seluruh data yang ada pada klaster tersebut. Nilai estimasi inilah yang akan digunakan untuk imputasi klaster yang berisi *missing data* (Malarvizhi dkk, 2012). Sedangkan metode imputasi dengan model prediksi menggunakan sistem prediktif untuk memperkirakan nilai yang akan diimputasikan pada *missing data* (Malarvizhi dkk, 2012).

Penelitian terkait dengan imputasi *missing data* dan klasifikasi dataset telah banyak dilakukan. Salah satunya penelitian (Acuna, 2004) membandingkan beberapa metode imputasi dengan menggunakan 12 dataset untuk mengevaluasi hasil klasifikasi. Terdapat 4 metode yang diujicobakan, yaitu: metode *Case Deletion*, *Mean*, *Median*, dan K-NN. Untuk pengklasifikasian data penelitian ini menggunakan LDA untuk pengklasifikasian parametrik dan KNN untuk non-parametrik. Penelitian (Farhangfari dkk, 2008) melakukan evaluasi tentang pengaruh dari metode imputasi terhadap kinerja klasifikasi. Penelitian tersebut menggunakan metode *Mean* yaitu melakukan imputasi *missing data* dengan menggunakan nilai rata-rata dari beberapa data yang telah ada. Selain itu digunakan pula metode klusterisasi untuk menangani *missing data* seperti penggunaan algoritma *K-Means Clustering* yang mengkluster data untuk melakukan imputasi (Malarvizhi dkk, 2012). Penelitian ini juga menganalisis kinerja dari sistem prediksi dengan membandingkan algoritma *K-Means Clustering* dan KNN Classifier untuk imputasi data. Hasil dari pengujian menunjukkan bahwa algoritma k-NN memiliki akurasi lebih baik dibandingkan *K-Means Clustering* untuk imputasi data. Beberapa penelitian juga menggunakan metode hybrid untuk menyelesaikan permasalahan imputasi *missing data*. Salah satunya adalah penelitian (Mawarsari, 2012) yang menggunakan

metode hibrida KNN-GA yaitu metode gabungan antara *k-Nearest Neighbour* (KNN) dan Algoritma Genetik (GA) yang digunakan sebagai seleksi dan pembobotan variabel untuk imputasi pada *missing data*. Penelitian ini juga melakukan perbandingan kinerja metode KNN-GA dengan metode imputasi lain yaitu KNN dan metode hibrida *Artificial Neural Networks* dan Algoritma Genetika (ANN-GA) dengan cara mengukur nilai RMSE dan waktu running dari masing-masing metode. Penelitian ini menggunakan data hasil Survei IBS 2008 kategori perusahaan industry. Pola *missing data* pada kasus ini adalah pola *missing data* univarian dengan asumsi mekanisme *Missing Completely at Random* (MCAR). Hasil yang didapatkan dari penelitian ini adalah bahwa imputasi *missing data* dengan menggunakan metode KNN yang menggunakan nilai bobot variabel dari metode hibrida KNN-GA dapat menghasilkan nilai RMSE yang lebih kecil daripada imputasi KNN dengan menggunakan bobot koefisien korelasi variabel.

Dalam penanganan *missing data*, KNN merupakan metode yang paling mudah dan populer. Namun, metode ini memiliki beberapa kelemahan salah satunya adalah pemilihan nilai *k* yang tidak tepat dapat menurunkan kinerja klasifikasi. Sehingga diusulkan algoritma genetika untuk mengoptimasi nilai *k* pada KNN sehingga dapat menghasilkan nilai estimasi yang baik dengan MSE sekecil mungkin. Dengan demikian, akan didapatkan hasil klasifikasi dengan akurasi yang tinggi. Penelitian ini bertujuan untuk menangani *missing data* dengan teknik imputasi menggunakan gabungan algoritma KNN dan GA (KNN-GA).

2. Metodologi

Penelitian ini menggunakan dataset lengkap yang digunakan untuk mengukur performansi beberapa metode imputasi yang akan diujicobakan. Selanjutnya dengan mengikuti metode MCAR akan dibangkitkan *missing data* dari dataset komplit tersebut dengan proporsi 10%, 20%, dan 30% dari keseluruhan data. Langkah berikutnya adalah melakukan imputasi terhadap *missing data* dengan menggunakan 4 metode imputasi yang telah dipilih, yaitu: KNN-GA, KNN, *Mean*, dan *Median*. Dari hasil imputasi akan dianalisa performansi dengan menggunakan MSE dan akurasi klasifikasi dengan menggunakan metode *Multi Layer Perceptron* (MLP). Gambar 1 menunjukkan alur diagram dari penelitian.

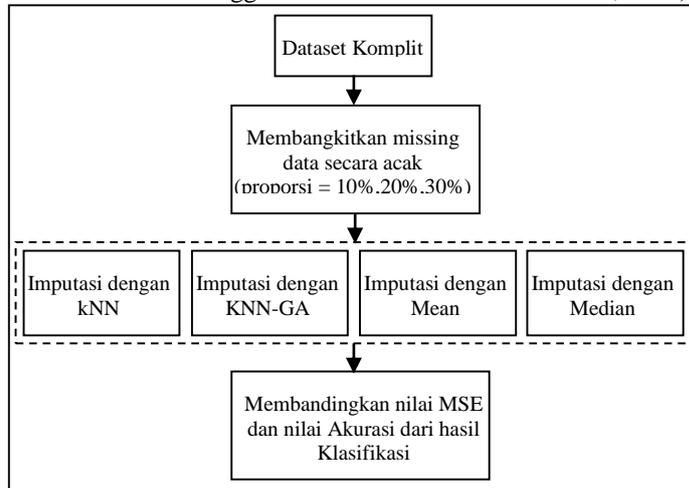
2.1 Mekanisme *Missing data*

Missing data adalah suatu kondisi dimana data tidak ada atau data hilang. Terdapat 3

mekanisme penghilangan data, yaitu (Farhangfari dkk, 2008) :

- 1) *Missing Completely at Random* (MCAR), yaitu jika distribusi data yang hilang pada suatu atribut tidak tergantung pada data pengamatan atau *missing data*. Metode ini akan menggunakan

dataset komplit kemudian membangkitkan *missing data* secara acak berdasarkan proporsi tertentu. Keuntungan dari metode ini adalah memudahkan para peneliti untuk estimasi komputasi dari model yang diusulkan (Pigott dan Therese, 2011).



Gambar 1. Diagram Alir Penelitian

- 2) *Missing at Random* (MAR), yaitu jika distribusi data yang hilang pada suatu atribut tergantung pada data pengamatan tetapi tidak tergantung pada *missing data*.
- 3) *Not Missing at Random* (NMAR), jika distribusi data yang hilang pada suatu atribut tergantung pada *missing data*.

2.2 Imputasi dengan Metode Mean dan Median

Imputasi dengan menggunakan *mean* dan *median* merupakan metode imputasi yang paling sering digunakan dan sederhana (Acuna, 2004). *Mean* melakukan imputasi *missing data* dengan cara menghitung nilai rata-rata dari seluruh nilai data yang diketahui. Metode ini hanya digunakan untuk data numerik. Bentuk formal teknik perhitungan *mean* dapat dilihat pada persamaan (1). Sedangkan jika data yang akan diimputasi merupakan data kategorikal maka dapat menggunakan *modus* (nilai yang frekuensi kemunculannya tinggi). Sedangkan *median* adalah metode imputasi dengan cara mencari nilai tengah dari seluruh nilai data yang diketahui. Metode ini optimal ketika digunakan pada data yang memiliki distribusi *skewed*. Bentuk formal dari teknik perhitungan median ditunjukkan oleh persamaan (2). Misalkan x adalah data berjumlah n maka persamaan umum mencari *Mean* data tersebut adalah:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (1)$$

Sedangkan persamaan umum mencari *Median* data adalah:

$$\tilde{x} = \begin{cases} \frac{n+1}{2}, & \text{untuk } n \text{ ganjil} \\ \frac{1}{2} \left[\frac{n}{2} + \left(\frac{n+1}{2} \right) \right], & \text{untuk } n \text{ genap} \end{cases} \quad (2)$$

2.3 Imputasi dengan Metode KNN

Salah satu metode yang sering digunakan untuk masalah imputasi *missing data* adalah KNN. Metode ini merupakan metode yang sederhana dan fleksibel karena dapat digunakan baik pada variabel dengan data kontinu maupun data diskrit (Mawarsari, 2012). Langkah penting dalam metode ini adalah menentukan nilai k dimana sejumlah k tetangga terdekat dari dataset akan dijadikan sebagai nilai estimator. Pemilihan sejumlah tetangga didasarkan pada jarak salah satu *instance* dengan seluruh *instance* yang ada pada dataset. Dari informasi tetangga tersebut diperoleh estimasi nilai yang kemudian digunakan sebagai nilai imputasi pada data yang hilang. Perhitungan dalam menentukan nilai imputasi tergantung pada jenis data, untuk data kontinu digunakan rata-rata dari tetangga terdekat, sedangkan untuk data kualitatif nilai imputasi diambil dari nilai yang seringkali keluar (Laencina dkk, 2009).

Misalkan, Jika D merupakan suatu dataset dengan *missing data*, maka D_c adalah subdata yang komplit sedangkan D_m adalah subdata dimana setiap *instance* mengandung atribut yang hilang sehingga $D = D_c \cup D_m$. Algoritma KNN dapat dilihat pada Gambar 2.

Algoritma KNN

1. Tentukan nilai k
2. Tentukan jarak Euclidian antar *instance* pada dataset D_m dan dataset D_c
3. Imputasi data hilang dengan rata-rata k tetangga terdekat di D_c

Gambar 2. Algoritma KNN

Keuntungan KNN adalah karena metode ini dapat digunakan untuk data yang bersifat kualitatif maupun kuantitatif, tanpa perlu membuat model prediksi, algoritma sederhana, KNN dibutuhkan untuk pertimbangan struktur korelasi data (Acuna, 2004). Sedangkan kelemahan KNN adalah adanya pemilihan fungsi jarak, dapat menggunakan Euclidean, Manhattan, Mahalanobis dan Pearson. Selain itu Algoritma KNN akan mencari data yang paling mirip pada seluruh data pada dataset. Hal ini mengakibatkan kompleksitas waktu yang dibutuhkan algoritma ini menjadi cukup tinggi. Kelemahan yang ketiga adalah Pemilihan nilai k yang tidak tepat dapat menurunkan kinerja klasifikasi (Acuna, 2004).

2.4 Metode Optimasi dengan Algoritma Genetika

Algoritma genetika adalah metode optimasi adaptif yang berdasar pada prinsip biologis. Dalam algoritma genetika, populasi kromosom merepresentasikan sekumpulan solusi dari permasalahan. Dasar-dasar teoritis dari algoritma genetika didasarkan pada proses seleksi reproduksi dan rekombinasi suatu kromosom. Adanya proses *crossover* dan mutasi yang terjadi pada kromosom dijadikan sebagai konsep dasar operator algoritma genetika dengan peluang tertentu. (Jurasovic, 2010). Dalam algoritma genetika proses *crossover* dan mutasi dilakukan berdasarkan pada peluang *crossover* (p_c) dan peluang mutasi (p_m). Prosedur algoritma genetika dapat dilihat pada Gambar 4.

Algoritma Genetika

- 1) Inisialisasi n populasi secara random
- 2) Evaluasi nilai fitness setiap kromosom
- 3) Buat populasi baru dengan mengulangi langkah sebagai berikut:
 - i. Seleksi
 - ii. *Crossover*
 - iii. Mutasi
- 4) Update populasi.
- 5) Jika kondisi terpenuhi berhenti dan kembali ke solusi terbaik di populasi sekarang

Gambar 4. Algoritma Genetika

2.5 Metode Hybrid KNN-GA

Imputasi *missing data* dengan menggunakan KNN dan GA dilakukan dengan membagi dataset menjadi dua yaitu data dengan *instance* yang memiliki atribut yang lengkap dan *instance* yang mengandung minimal satu atribut yang hilang. Jika D merupakan suatu dataset dengan *missing data*, maka dataset D_c adalah subdata yang komplit sedangkan D_m adalah subdata dimana setiap *instance* mengandung atribut yang hilang sehingga $D = D_c \cup D_m$.

Langkah pertama yang dilakukan adalah membangkitkan kromosom k secara acak. Kromosom dikodekan dengan string biner. Misalkan representasi kromosom untuk nilai $k = 9$ dapat dilihat pada Gambar 3.

1	0	0	1
---	---	---	---

Gambar 3. Representasi Kromosom

Langkah selanjutnya adalah menentukan jarak antara *instance* pada D_m dengan k tetangga pada D_c menggunakan konsep jarak Euclidian. Jika N adalah jumlah atribut yang terisi pada data hilang dengan $x_i = (x_i^1, x_i^2, \dots, x_i^N)$ dan $y_i = (y_i^1, y_i^2, \dots, y_i^N)$ merupakan dua pasangan data, maka jarak Euclid dapat ditentukan dengan persamaan :

$$d = \sqrt{\sum_{i=1}^N (x_i - y_i)^2} \tag{3}$$

Evaluasi dilakukan dengan menghitung nilai fitness setiap kromosom. Fungsi fitness yang digunakan adalah fungsi yang berbanding terbalik dengan *Mean Square Error* (MSE) antara nilai imputasi dengan nilai real. Jika $r_i = (r_i^1, r_i^2, \dots, r_i^N)$ merupakan dataset real dan $o_i = (o_i^1, o_i^2, \dots, o_i^N)$ merupakan dataset hasil imputasi maka MSE dapat ditentukan dengan persamaan :

$$MSE = \frac{\sum_{i=1}^N (r_i - o_i)^2}{N} \tag{4}$$

Nilai k hasil dari optimasi GA digunakan untuk menentukan sejumlah tetangga yang akan digunakan sebagai estimator nilai *missing data*. Proses imputasi dilakukan dengan menghitung rata-rata atribut k tetangga.

3. Hasil dan Pembahasan

3.1 Dataset

Data yang digunakan dalam uji coba ini adalah dataset iris, wine, dan glass yang diambil dari UCI *Machine Learning Repository*. Dataset tersebut merupakan dataset yang lengkap yang tidak terdapat *missing value* pada data. Karakteristik dataset tersebut dapat dilihat pada Tabel 1.

Tabel 1. Dataset

Dataset	Instance	Atribut	Kelas
Iris	150	4	3
Wine	178	13	3
Glass	214	10	7

3.2 Hasil Pengujian dan Pembahasan

Skenario yang dilakukan pada uji coba penelitian ini adalah dengan cara menghilangkan beberapa data dengan mekanisme MCAR dari dataset untuk kemudian dilakukan proses imputasi dengan menggunakan metode KNN-GA dan metode lain sebagai pembanding. Dalam uji coba ini digunakan full dataset (apriori) karena akan dilakukan analisa performansi sehingga dapat dibandingkan data asli dengan data hasil imputasi. Pengujian pada penelitian ini bertujuan untuk mengetahui performansi dari KNN-GA sebagai metode imputasi data. Secara garis besar uji coba pada penelitian ini dibagi menjadi dua skenario utama. Skenario pertama adalah uji performansi model dengan akurasi imputasi dengan cara menghitung nilai MSE untuk mengetahui perbandingan nilai data dari data asli dengan data hasil imputasi. Skenario selanjutnya uji performansi dengan akurasi hasil klasifikasi.

Pengujian dilakukan pada beberapa metode imputasi untuk menganalisa performa dari metode imputasi yang paling superior. Pada penelitian ini terdapat 4 metode yang akan diujicobakan yaitu KNN-GA, KNN, *Mean*, dan *Median*. Sesuai dengan skenario awal, proses yang akan dilakukan untuk imputasi data dari complete data adalah membuat *missing data*. Dengan cara menghilangkan beberapa *instance* secara acak, berturut-turut dengan proporsi 10%, 20%, dan 30% (Langkamp dkk, 2010).

Dalam pengujian ini akan dilakukan 20 replikasi untuk tiap proporsi pada masing-masing dataset. Dimana dari setiap replikasi akan dihitung nilai MSE sehingga dari 20 replikasi akan didapatkan nilai rata-rata MSE untuk tiap metode. Untuk metode KNN-GA, parameter yang akan digunakan adalah parameter pada algoritma genetika. Parameter tersebut adalah $p_{size} = 10$, $p_c = 0.6$, $p_m = 0.1$, $maxgen = 100$. Sedangkan untuk metode KNN nilai k ditentukan secara acak. Tabel 2 menunjukkan nilai rata-rata MSE yang diperoleh dari 20 replikasi.

Tabel 2. Perbandingan rata-rata MSE

Dataset	Proporsi Missing data	Rata-rata MSE			
		KNN-GA	KNN	Mean	Median
Iris	10 %	0.00283	0.00827	0.02192	0.02396
	20 %	0.02386	0.03757	0.05028	0.05430

Wine	30 %	0.04616	0.05664	0.07515	0.08446
	10 %	6.40706	59.5957	103.535	119.232
	20 %	0.02769	5.53316	1.89650	0.02646
Glass	30 %	28.0994	69.8221	89.689	55.9172
	10 %	0.00467	0.09343	0.0300	0.03088
	20 %	1.23007	3.82732	9.3970	9.27990
	30 %	2.72497	7.49991	11.275	12.0208

Dari pengamatan secara rata-rata metode imputasi KNN-GA memiliki MSE paling rendah. Dalam pengujian data hasil 20 replikasi antara metode hybrid KNN-GA dengan KNN, *Mean*, dan *Median* diatas digunakan uji statistik ANOVA dengan taraf kepercayaan 95%. Dengan menggunakan software SPSS 17, diperoleh nilai p-value untuk perbandingan metode KNN-GA dengan metode imputasi KNN, Mean, dan Median adalah sebagai berikut:

Tabel 3. Perbandingan p-value MSE

Dataset	Proporsi Missing Data	p-value		
		KNN	Mean	Median
Iris	10 %	0.055	0.000	0.000
	20 %	0.033	0.000	0.000
	30 %	0.138	0.000	0.000
Wine	10 %	0.000	0.000	0.000
	20 %	0.000	0.000	0.998
	30 %	0.000	0.000	0.000
Glass	10 %	0.000	0.010	0.008
	20 %	0.000	0.000	0.000
	30 %	0.000	0.000	0.000

Jika diambil nilai $\alpha = 0.05$, maka hal ini menunjukkan bahwa metode imputasi KNN-GA tidak memiliki perbedaan yang signifikan untuk kasus dataset iris dengan 10% missing data, 30% missing data, dan untuk dataset wine 20% missing data. Dari sini bisa disimpulkan bahwa secara rata-rata metode KNN-GA memiliki performa yang bagus dengan menghasilkan MSE yang paling kecil jika dibandingkan dengan tiga metode pembandingnya. Namun, pada kasus tertentu KNN-GA memberikan nilai MSE yang sama dengan metode imputasi KNN dan Median. Hal ini disebabkan proses pemilihan data hilang adalah secara random sehingga adakalanya nilai nilai yang hilang merupakan median itu sendiri.

Selanjutnya, uji performansi model dengan menggunakan metode klasifikasi akan melakukan proses klasifikasi dari dataset yang telah diimputasi untuk mengetahui pengaruh metode imputasi terhadap akurasi dari hasil klasifikasi. Uji performa

model dengan melihat hasil klasifikasi data Metode imputasi hybrid KNN GA akan dibandingkan dengan metode imputasi KNN, *Mean*, dan *Median*. Dalam pengujian ini dilakukan 20 replikasi dengan parameter popsize = 10, $P_c = 0.6$, $P_m = 0.1$, maxgen = 100. Berikut ini adalah nilai akurasi klasifikasi:

Tabel 4. Perbandingan akurasi klasifikasi

Dataset	Proporsi Missing data	Akurasi klasifikasi (%)			
		KNN - GA	KNN	Mean	Median
Iris	10 %	95.800	95.699	95.067	95.133
	20 %	93.683	93.916	91.367	91.450
	30 %	95.900	94.833	92.234	92.467
Wine	10 %	97.499	97.416	97.359	97.416
	20 %	95.309	95.281	95.393	95.365
	30 %	96.404	96.433	96.489	96.433
Glass	10 %	88.551	87.967	87.757	87.850
	20 %	85.257	84.743	84.299	84.322
	30 %	92.710	91.799	91.215	91.238

Dari pengamatan secara rata-rata metode imputasi KNN-GA memiliki akurasi paling tinggi. Dalam pengujian data hasil 20 replikasi antara metode hybrid KNN-GA dengan KNN, mean, dan median diatas digunakan uji statistik ANOVA dengan taraf kepercayaan 95%. Dengan menggunakan software SPSS 17, pada perbandingan metode KNN-GA dengan metode imputasi KNN, Mean, dan Median adalah sebagai berikut:

Tabel 5. Perbandingan p-value akurasi klasifikasi

Dataset	Proporsi Missing Data	p-value		
		KNN	Mean	Median
Iris	10 %	0.955	0.679	0.707
	20 %	0.921	0.328	0.346
	30 %	0.133	0.000	0.000
Wine	10 %	0.862	0.772	0.862
	20 %	0.985	0.954	0.970
	30 %	0.983	0.950	0.983
Glass	10 %	0.845	0.790	0.814
	20 %	0.927	0.865	0.868
	30 %	0.487	0.256	0.263

Jika diambil nilai $\alpha = 0.05$, maka hal ini menunjukkan bahwa metode imputasi KNN-GA hanya memiliki perbedaan yang signifikan dengan metode Mean dan Median untuk kasus dataset iris dengan 30% missing data. Dari sini bisa disimpulkan bahwa secara rata-rata metode KNN-GA memiliki performa yang bagus dengan menghasilkan akurasi yang paling tinggi jika dibandingkan dengan tiga metode pembandingnya. Namun, perbedaan ini tidak signifikan sehingga jika berpedoman pada hasil tes statistik maka performa hasil klasifikasi dari empat metode ini adalah sama.

4. Kesimpulan dan Saran

Penelitian ini menggunakan KNN-GA untuk melakukan imputasi *missing data*. Dataset yang digunakan adalah komplit dataset sehingga digunakan mekanisme MCAR untuk membangkitkan *missing data* dengan proporsi 10%, 20% dan 30%. Pengujian data menunjukkan bahwa secara signifikan KNN-GA memiliki nilai rata-rata MSE paling rendah dibandingkan 3 metode imputasi lain yaitu KNN, mean dan median. Sedangkan pada pengujian selanjutnya menunjukkan bahwa KNN-GA memiliki akurasi hasil klasifikasi paling tinggi dibandingkan 3 metode imputasi lain namun tidak signifikan. Dari penelitian ini dapat disimpulkan bahwa metode KNN-GA memiliki kinerja yang baik dalam melakukan imputasi.

5. Ucapan Terima Kasih

Ucapan terima kasih kami sampaikan kepada Ibu Dr. Chastine Fatichah, S.Kom, M.Kom atas masukan dan bimbingan yang telah diberikan sehingga terselesaikannya pembuatan paper ini.

Daftar Pustaka

- [1] Acuna, dkk., 2004, *The treatment of missing values and its effect on classifier accuracy*, Classification, Clustering, and Data Mining Applications. Springer Berlin Heidelberg, hal. 639-647.
- [2] Farhangfar, K., dkk, 2008, *Impact of imputation of missing values on classification error for discrete data*, Pattern Recognition, Vol.41, hal. 3692-3705.
- [3] Jurasovic, K., 2010, *Genetic algorithm for optimizing service distributions*, Neurocomputing, Vol. 73, hal 661-668.
- [4] Laencina, G., dkk, 2009, *K nearest neighbours with mutual information for simultaneous classification and missing data imputation*, Neurocomputing, Vol.72, hal. 1483-1493.
- [5] Langkamp, dkk., 2010, *Techniques for handling missing data in secondary analyses of large surveys*, Academic Pediatric Association, Vol. 10. No. 3, hal 205-210
- [6] Little, R. J. dan Rubin, D.B. 2002. *Statistical analysis with missing data*. Second Edition. John Wiley and Sons, New York.

- [7] Malarvizhi, T. 2012, *K-nn classifier performs better than k-means clustering in missing value imputation*, IOSR Journal of Computer Engineering, Vol.6, No. 5, hal 12-15.
- [8] Mawarsari, U., 2012, *Imputasi missing data dengan k-nearest neighbour dan algoritma genetika*, Tesis Pascasarjana Jurusan Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA), Institut Teknologi Sepuluh Nopember, Surabaya.
- [9] Pigott D.T., 2011. *A review of methods for missing data*. Educational Research and Evaluation, Vol. 7, No.4, hal. 353-383.
- [10] Zhang, dkk., 2011, *Missing data imputation by utilizing information within incomplete instances*, The Journal of Systems and Software, Vol.84, No. 3, hal. 452–459.

Abidatul Izzah, memperoleh gelar Sarjana Sains (S.Si), Program Studi Matematika FSAINTEK UNAIR Surabaya – Jawa Timur, lulus tahun 2012. Saat ini tercatat sebagai mahasiswa program Magister Teknik Informatika ITS sekaligus pengajar di lembaga bimbingan belajar Sony Sugema College (SSC) Kediri-Jawa Timur.

Nur Hayatin, memperoleh gelar Sarjana Sains Terapan (S.ST), Program Studi Teknik Informatika PENS Surabaya – Jawa Timur, lulus tahun 2007. Saat ini tercatat sebagai mahasiswa program Magister Teknik Informatika ITS sekaligus pengajar di Universitas Muhammadiyah Malang - Jawa Timur.

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

(9)
PREDIKSI CACAT PERANGKAT LUNAK PADA KODE SUMBER DENGAN SVM DAN SELEKSI ATRIBUT

Ratih Kartika Dewi¹⁾, Sarwosri²⁾

Jurusan Teknik Informatika ITS Surabaya

Jl. Teknik Kimia Kampus Teknik Informatika Institut Teknologi Sepuluh Nopember

Surabaya (031) 5939214

e-mail :ratihkartikadewi@yahoo.com¹⁾, sarwosri@if.its.ac.id²⁾

ABSTRAK

Kegagalan sebuah perangkat lunak muncul pada saat perangkat lunak tersebut digunakan oleh pengguna. Kegagalan tersebut disebabkan adanya cacat pada perangkat lunak. Prediksi cacat perangkat lunak diperlukan untuk efisiensi dalam hal kepuasan konsumen, biaya yang diperlukan untuk membangun perangkat lunak dan waktu hidup yang mengindikasikan perangkat lunak dapat digunakan sesuai fungsinya. Cacat perangkat lunak paling banyak ditemukan pada modul, dalam hal ini *function* dan *method* pada kode sumber. Karakteristik internal sebuah perangkat lunak dapat dilihat pada *component-level design metrics* yang terdapat pada kode sumber yang dapat diketahui melalui atribut yang terdapat dalam metrik Mc Cabe dan *Halstead*. Penelitian ini menganalisis dampak pemilihan atribut pada prediksi cacat perangkat lunak dengan SVM, pemilihan atribut secara manual dipilih berdasarkan metrik dasar Mc Cabe dan Halstead sedangkan pemilihan atribut otomatis dengan CFS. Seleksi atribut secara manual cocok digunakan untuk data mentah yang belum diketahui nilai dari masing-masing atribut secara numerik karena menghasilkan akurasi, *precision* dan *recall* yang baik serta atribut yang dimasukkan konsisten, yaitu 8 atribut yang telah diseleksi secara manual. Seleksi atribut dengan CFS cocok digunakan untuk data yang sudah diolah dan memiliki nilai dari masing-masing metrik secara numerik karena berapa saja atribut yang dimasukkan, CFS akan menyeleksi atribut untuk kemudian diproses lebih lanjut.

Kata Kunci : kualitas perangkat lunak, cacat, kode sumber, klasifikasi, seleksi atribut.

1. PENDAHULUAN

Kegagalan sebuah perangkat lunak muncul pada saat perangkat lunak tersebut digunakan oleh pengguna. Kegagalan tersebut disebabkan adanya cacat pada perangkat lunak. Cacat pada perangkat lunak dapat muncul pada semua tahap proses pembuatan perangkat lunak, mulai dari identifikasi kebutuhan, perancangan, sampai dengan dengan pembuatan kode sumber (Pressman, 2010).

Prediksi cacat perangkat lunak diperlukan untuk efisiensi dalam hal kepuasan konsumen, biaya yang diperlukan untuk membangun perangkat lunak dan waktu hidup yang mengindikasikan perangkat lunak dapat digunakan sesuai fungsinya. Prediksi cacat perangkat lunak yang efektif pada modul perangkat lunak dapat membuat developer lebih fokus pada jaminan kualitas perangkat lunak (Elish, 2008).

Cacat perangkat lunak paling banyak ditemukan pada modul, dalam hal ini *function* dan *method* pada kode sumber (Elish, 2008). Karakteristik internal sebuah perangkat lunak dapat dilihat pada *component-level design metrics* yang terdapat pada kode sumber (Zhang, 2007). Hal tersebut dapat diketahui melalui atribut yang terdapat dalam metrik Mc Cabe dan Halstead.

Dataset NASA (NASA, 2004) adalah dataset yang banyak digunakan untuk prediksi cacat perangkat lunak. Pada dataset ini terdapat 21 metrik yang digunakan untuk prediksi defect pada kode sumber. Metrik tersebut dijelaskan lebih rinci pada tabel 2 dan 3.

Prediksi cacat perangkat lunak dapat dilakukan dengan metode klasifikasi. Support Vector Machine (SVM) adalah metode klasifikasi yang menggunakan hyperplane untuk memisahkan kelas yang berbeda (Wu, 2008). Pada penelitian mengenai prediksi cacat pada modul perangkat lunak dengan SVM (Elish, 2008), SVM menghasilkan recall paling baik, yaitu 99,4 – 100 %, Kelemahan penelitian (Elish, 2008) adalah dalam hal akurasi, SVM tidak menghasilkan akurasi lebih baik daripada 8 metode lain, yaitu 84,6 – 93,3 %. Hal tersebut dapat disebabkan masih adanya korelasi yang kuat antar atribut dalam 21 metrik untuk deteksi cacat perangkat lunak sehingga memerlukan metode untuk seleksi atribut. Misalnya pada program length (L) memiliki korelasi dengan total number of operand and operator (N) dan volume on minimal implementation (V).

Dalam penelitian ini diajukan metode untuk prediksi cacat perangkat lunak pada kode

sumber dengan SVM dan seleksi atribut sehingga diharapkan dapat dibangun sebuah model untuk prediksi cacat perangkat lunak pada kode sumber yang efektif. Penelitian ini menganalisis dampak pemilihan atribut pada prediksi cacat perangkat lunak dengan SVM, pemilihan atribut secara manual dipilih berdasarkan metrik dasar Mc Cabe dan Halstead sedangkan pemilihan atribut otomatis dengan CFS.

2. TINJAUAN PUSTAKA

Penelitian mengenai prediksi cacat perangkat lunak pada kode sumber dengan SVM dan seleksi atribut ini memiliki tinjauan pustaka mengenai cacat pada perangkat lunak, Support Vector Machine dan seleksi atribut, sebagai berikut:

2.1 Prediksi Cacat pada Modul Perangkat Lunak dengan SVM

Pada penelitian mengenai prediksi cacat pada modul perangkat lunak dengan SVM (Elish, 2008), SVM menghasilkan recall paling baik, yaitu 99,4 – 100 %. Kelemahan penelitian (Elish, 2008) adalah dalam hal akurasi, SVM tidak menghasilkan akurasi lebih baik daripada 8 metode lain yang dibandingkan dalam penelitian tersebut, yaitu Logistic Regression (LR), K-Nearest Neighbor (KNN), Multi-layer Perceptrons (MLP), Radial Basis Function (RBF), Bayesian Belief Networks (BBN), Naïve Bayes (NB), Random Forests (RF), dan Decision Trees (DT), yaitu 84,6 – 93,3 %.

Tabel 1 Hasil pengujian (Elish, 2008)

Data set	Jumlah atribut	Akurasi (%)	Precision (%)	Recall (%)
CM1	21	90,69	90,66	100
KC1	22	84,59	84,95	99,40
KC3	40	93,28	87,3	99,58
PC1	22	93,10	93,65	99,58

2.2 Cacat Perangkat Lunak pada Kode Sumber

Manajemen kualitas perangkat lunak adalah *umbrella activity* yang diaplikasikan pada semua tahap dalam proses perangkat lunak. Pendekatan sistematis untuk peningkatan kualitas perangkat lunak dapat dilakukan dengan mengeliminasi akar penyebab kecacatan pada perangkat lunak. IEEE Standard 100-1992 mendefinisikan cacat perangkat lunak sebagai sebuah anomali (Pressman, 2010). Tujuan utama prediksi cacat perangkat lunak adalah menemukan kesalahan yang ada selama proses sehingga tidak menjadi produk yang cacat setelah digunakan oleh pengguna. Cacat perangkat lunak dapat terjadi pada semua proses perangkat lunak mulai elisitasi kebutuhan, perancangan sampai pembangunan kode sumber pada proses perangkat lunak. Beragam kesalahan pada proses perangkat lunak kebanyakan disebabkan:

Tabel 1. Jenis kesalahan pada proyek perangkat lunak

Jenis kesalahan	Keterangan
Kesalahan spesifikasi	Spesifikasi yang ada tidak sesuai kebutuhan konsumen
Kesalahan komunikasi pada pelanggan	Ambiguitas dalam komunikasi dengan stakeholder
Deviasi tujuan produk dari spesifikasi	Terdapat perbedaan dengan spesifikasi kebutuhan awal
Violasi dari standar pemrograman	Adanya pelanggaran standar pemrograman
Kesalahan representasi data	Data yang dikumpulkan tidak representative (tidak mencerminkan keadaan sebenarnya)
Komponen antarmuka yang tidak konsisten	Misalnya saat menekan tombol menu A, ternyata yang muncul jendela menu B
Kesalahan logika desain	Kesalahan pada pembangunan desain logika pemrograman, misalnya kesalahan rule pada IF
Kesalahan dalam pengujian	Kurang optimal dalam tahap pengujian perangkat lunak
Dokumentasi yang tidak lengkap	Dokumentasi yang kurang lengkap menyebabkan adanya informasi yang terlewatkan
Kesalahan translasi dari desain ke bahasa pemrograman	Kesalahan penerjemahan dari desain ke bahasa pemrograman
Antarmuka komputer yang ambigu	Antarmuka yang membingungkan bagi pengguna

Penelitian mengenai prediksi cacat perangkat lunak dengan pemilihan atribut yang paling berpengaruh secara manual telah dilakukan (Zhang, 2007, 2009) dengan memilih atribut untuk prediksi cacat perangkat lunak sehingga dapat meningkatkan kualitas perangkat lunak yang dibangun.

Penelitian ini menyatakan bahwa ada hubungan karakteristik eksternal perangkat lunak (kualitas) dengan atribut internalnya (misal: jumlah baris pada kode sumber). Belum ada teori untuk prediksi cacat perangkat lunak pada kode sumber sehingga penelitian ini berkembang dengan pembelajaran mesin, seperti dilakukan (Khosgoftaar, 2004) menggunakan dataset NASA dengan 21 atribut (tanpa pemilihan atribut) dan 25 teknik klasifikasi menunjukkan performa prediksi yang kurang baik, begitu juga penelitian (Menzies, 2007). Oleh karena itu (Zhang, 2007, 2009) melakukan penelitian prediksi perangkat lunak dengan memilih pengukuran atribut baris kode, volum, dan kompleksitas siklomatis (Zhang, 2007) dan (Zhang, 2009) menggunakan atribut total baris kode untuk memprediksi cacat perangkat lunak dan kedua penelitian tersebut menunjukkan perbaikan performa untuk prediksi cacat perangkat lunak.

Cacat perangkat lunak paling banyak ditemukan pada modul, dalam hal ini *function* dan *method* pada kode sumber (Elish, 2008). Karakteristik internal sebuah perangkat lunak dapat dilihat pada *component-level design metrics* yang terdapat pada kode sumber (Zhang, 2007). Hal

tersebut dapat diketahui melalui atribut yang terdapat dalam metrik Mc Cabe dan Halstead.

Tabel 2. Metriks Mc Cabe

METRIKS	KETERANGAN
Kompleksitas siklomatis V(g)	Pengukuran jumlah jalur yang linier bebas, yaitu jika tidak ada jalur yang merupakan kombinasi linier dari jalur lain pada flowgraph
Kompleksitas essensial EV(g)	Jumlah flowgraph yang tereduksi melalui dekomposisi subflowgraph dengan one-entry one-exit subflowgraph
Kompleksitas desain IV(g)	Kompleksitas siklomatis dari modul dengan flowgraph yang tereduksi
Total baris kode (LOC)	Atribut jumlah baris kode sumber
Jumlah baris statement (LOCcode)	Jumlah baris yang berisi kode
Jumlah baris comment (LOCcomment)	Jumlah baris yang berisi komentar
Jumlah baris kosong (LOBblank)	Jumlah baris yang kosong
Jumlah barisstatementdan comment (LOCcodeAndComment)	LOCcode + LOCcomment

Tabel 3. Metriks Halstead

METRIKS	KETERANGAN
Jumlah operator unik (UniqOp)	Jumlah operator unik. Misalnya perintah return, max
Jumlah operand unik (UniqOpnd)	Jumlah operand unik. Misalnya x, y, dan z
Jumlah operator (TotalOp)	Jumlah operator +, - dan operator lainnya
Jumlah of operand dan operator (N)	Jumlah operand dan operator
Volum pada implementasi minimal (V)	(2+ jumlah operand unik) * log2 (2+ jumlah operand unik)
Panjang program (L)	V / N
Kesulitan (D)	1 / L
intelligent count (I)	L * V
Usaha untuk penulisan program (E)	V / L
Perkiraan usaha (B)	Perkiraan usaha (E)
Waktu untuk menulis program (T)	E/18

Pada dataset NASA terdapat atribut tambahan untuk prediksi cacat perangkat lunak, yaitu Total percabangan (BranchCount).

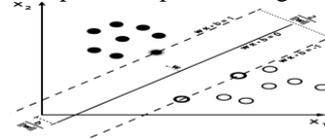
2.3 Support Vector Machine

Support Vector Machine (SVM) adalah sebuah teknik untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi. SVM berada dalam satu kelas dengan Artificial Neural Network (ANN) dalam hal fungsi dan kondisi permasalahan yang bisa diselesaikan. Hal yang berbeda adalah ANN menemukan solusi local optimal sedangkan SVM menemukan solusi global optimal. Teknik ini menarik perhatian peneliti karena performanya yang meyakinkan dalam prediksi.

Dalam SVM kita berusaha menemukan fungsi pemisah, dalam hal ini classifier/ hyperplane

yang optimal untuk memisahkan kelas berbeda. Hyperplane terbaik terletak di tengah-tengah antara dua set objek dari dua kelas, dalam hal ini memaksimalkan margin atau jarak antara dua kelas yang berbeda. SVM merupakan algoritma klasifikasi dengan langkah sebagai berikut:

a.) Melakukan plot data pada bidang koordinat



Gambar 2. Pemetaan data

b.) Memaksimalkan margin, dengan persamaan:

$$\begin{aligned} X_i \cdot w + b &\geq +1 \text{ for } y_i = +1 \\ X_i \cdot w + b &\leq -1 \text{ for } y_i = -1 \\ Y_i (x_i \cdot w + b) - 1 &\geq 0 \end{aligned} \quad (3.1)$$

c.) Optimasi nilai y sebagai constrained optimization problem dengan lagrangian

d.) Untuk mengatasi error pada klasifikasi SVM menggunakan slack variabel

$$\begin{aligned} X_i \cdot w + b &\geq +1 - \xi \text{ for } y_i = +1 \\ X_i \cdot w + b &\leq -1 + \xi \text{ for } y_i = -1 \\ Y_i (x_i \cdot w + b) - 1 + \xi &\geq 0 \end{aligned} \quad (3.2)$$

e.) Pada beberapa kasus, SVM memisahkan 2 kelas secara non-linier, dengan memetakan data ke ruang yang berdimensi lebih tinggi menggunakan fungsi kernel, misalnya kernel polinomial

$$K(x,y) = (x^T y + 1)^d \quad (3.3)$$

2.4 Seleksi Atribut

Correlation based Feature Selection (CFS) mengidentifikasi fitur yang relevan, artinya tidak ada ketergantungan yang kuat dengan fitur lainnya. Seleksi fitur dapat memperbaiki performa akurasi atau paling tidak menghasilkan akurasi yang sama dengan metode tanpa seleksi fitur. Metode yang digunakan dalam CFS adalah menentukan pasangan fitur dan menentukan bobot fitur dengan algoritma RELIEF (Hall, 1999). Seleksi atribut dengan CFS menggunakan algoritma sebagai berikut:

Seleksi atribut dengan CFS menggunakan algoritma sebagai berikut:

a.) Penghitungan nilai korelasi antar fitur dan fitur-kelas dengan Relief.

b.) Merit menyatakan himpunan fitur yang berkorelasi tinggi dengan kelas, tetapi tidak berkorelasi dengan masing-masing fitur lain. rcf adalah nilai tengah dari korelasi fitur-kelas dan rff adalah nilai tengah dari korelasi fitur-fitur.

$$\text{Merit } k = \frac{k \text{ rcf}}{\sqrt{k + k(k + 1) \text{ rff}}} \quad (3.4)$$

c.) CFS dapat dihitung dari pengembangan merit, yaitu dengan nilai merit terbaik (nilainya paling besar)

Penelitian mengenai pemilihan atribut secara otomatis berkembang dalam ranah

pembelajaran mesin, salah satu metode untuk pemilihan atribut secara otomatis adalah CFS (Correlation based Feature Selection). Penelitian (Hall, 1999) menunjukkan kebaikan penggunaan CFS terdapat pada kompleksitas waktu, yaitu bahwa kompleksitas waktu CFS sangat rendah.

2.5 Teknik Pengujian Sistem

Pengujian akurasi dilakukan untuk mengetahui akurasi sistem. Prosedur pengujianya adalah memasukkan data testing tanpa kelas ke sistem, kemudian hasil prediksi sistem dicocokkan kesesuaiannya dengan kelas pada data testing. Perhitungan untuk pengujian akurasi dapat dijabarkan dengan *weighted accuracy* sebagai berikut (Tan, 2006):

$$w \cdot \frac{TP}{TP+FP} + (1-w) \cdot \frac{TN}{TN+FN} \quad (3.6)$$

Keterangan:

w= weight/ bobot, bernilai 1.

TP = true positif, data yg diprediksi positif dan benar.

FP = false positif, data yg diprediksi positif tetapi salah.

FN = false negatif, data yg diprediksi negatif tetapi salah.

TN = true negatif, data yg diprediksi negatif dan benar.

Precision mendefinisikan komponen cacat yang diprediksikan benar dibandingkan dengan total komponen yang diprediksikan cacat (TP dan FP)

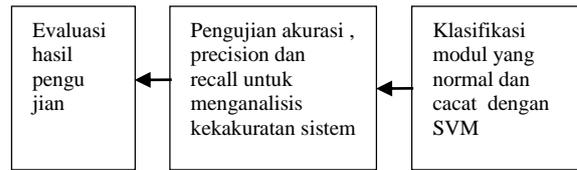
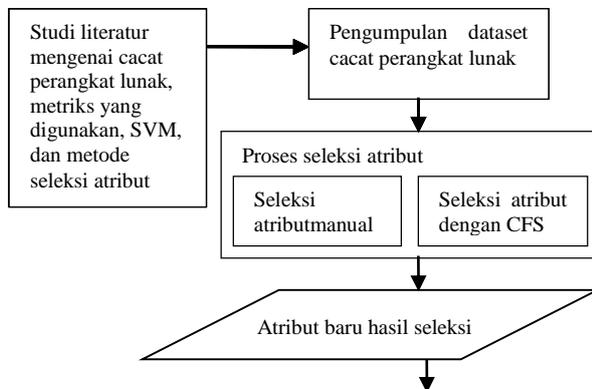
$$Precision = \frac{TP}{TP+FP}$$

Recall mendefinisikan komponen cacat yang diprediksikan benar dibandingkan dengan total komponen yang cacat (TP dan FN)

$$Recall = \frac{TP}{TP+FN}$$

3. METODE PENELITIAN

Metode dalam penelitian ini disusun sebagai berikut:



Gambar 1 Diagram Blok Sistem

3.1 Studi literatur

Studi literatur mengenai cacat perangkat lunak, metrik yang digunakan, SVM dan metode seleksi atribut. Kegagalan sebuah perangkat lunak muncul pada saat perangkat lunak tersebut digunakan oleh pengguna.

Kegagalan tersebut disebabkan adanya cacat pada perangkat lunak. Prediksi cacat perangkat lunak diperlukan untuk efisiensi dalam hal kepuasan konsumen, biaya yang diperlukan untuk membangun perangkat lunak dan waktu hidup yang mengindikasikan perangkat lunak dapat digunakan sesuai fungsinya. Prediksi cacat perangkat lunak yang efektif pada modul perangkat lunak dapat membuat developer lebih fokus pada jaminan kualitas perangkat lunak. Cacat perangkat lunak paling banyak ditemukan pada modul, dalam hal ini function dan method pada kode sumber. Hal tersebut dapat diketahui melalui atribut yang terdapat dalam metrik McCabe dan Halstead.

Pada penelitian mengenai prediksi cacat pada modul perangkat lunak dengan SVM (Elish, 2008), SVM menghasilkan recall paling baik, yaitu 99,4 – 100 %, Kelemahan penelitian (Elish, 2008) adalah dalam hal akurasi, SVM tidak menghasilkan akurasi lebih baik daripada 8 metode lain, yaitu 84,6 – 93,3 %. Dalam penelitian ini diajukan metode untuk prediksi cacat perangkat lunak pada kode sumber dengan SVM dan seleksi atribut sehingga diharapkan dapat dibangun sebuah model untuk prediksi cacat perangkat lunak pada kode sumber yang efektif dan akurat.

3.2 Pengumpulan Data

Dataset cacat perangkat lunak yang akan diteliti (NASA) Promise terdiri dari beberapa dataset, yaitu:

- CM1, modul berupa fungsi
- PC1, modul berupa fungsi
- KC1, modul berupa method
- KC3, modul berupa method

Dataset ini masing-masing memiliki 21 atribut seperti yang terdapat pada tabel 2 dan 3. Proses pengumpulan dataset dilakukan dengan proses sebagai berikut:

- Melakukan studi literatur mengenai penelitian cacat pada perangkat lunak dan meneliti dataset yang digunakan

- b) Mengunduh dataset NASA dari <http://nasa-softwaredefectdatasets.wikispaces.com/>. Tanggal akses: 12 Maret 2013
- c) Memilih dataset CM1, PC1, KC1, dan KC3 karena merupakan dataset yang digunakan pada penelitian sebelumnya (Elish, 2008).

3.3 Proses Seleksi Atribut

Seleksi atribut dilakukan dengan menggunakan 2 cara, yaitu secara manual dan metode Correlation based Feature Selection:

3.3.1 Seleksi atribut secara manual

Seleksi atribut secara manual menghasilkan metrik dasar dan metrik turunan Mc Cabe dan Halstead. Metrik dasar Mc Cabe, yaitu kompleksitas siklomatis, kompleksitas essensial, kompleksitas desain, dan total baris kode. Metrik turunan Mc Cabe, yaitu jumlah baris statement dan comment, jumlah baris statement, jumlah baris comment, dan jumlah baris kosong.

Metrik dasar Halstead, yaitu jumlah operator unik, jumlah operand unik, jumlah operator dan jumlah operand. Metrik turunan Halstead yaitu jumlah operand dan operator, volum pada implementasi minimal, panjang program, kesulitan, intelligent count, usaha untuk penulisan program, perkiraan usaha, waktu untuk menulis program, dan jumlah baris statement dan comment.

3.3.2 Seleksi atribut dengan CFS

Seleksi atribut secara otomatis dengan CFS diimplementasikan menggunakan WEKA seperti pada gambar 2, yaitu pada *filter attribute selection*. AttributeSelection-E

"weka.attributeSelection.CfsSubsetEval"-
S"weka.attributeSelection.BestFirst -D 1 -N 5"
Seleksi atribut secara manual dan otomatis yang telah dilakukan menghasilkan atribut baru yang cocok untuk prediksi cacat perangkat lunak pada kode sumber.

3.4 Klasifikasi dengan SVM

Klasifikasi menggunakan SVM diimplementasikan menggunakan WEKA seperti pada gambar 3, yaitu pada *function* pada tab *classification*.

SMO -C 1.0 -L 0.0010 -P 1.0E-12 -N o -V -1 -W 1 -K

"weka.classifiers.functions.supportVector.PolyKernel -C 250007 -E 1.0"

3.5 Pengujian

Pengujian sistem dilakukan dengan pengujian akurasi, precision dan recall untuk mengetahui keakuratan sistem yang dibangun, seperti pada gambar 4. Pengujian sistem dilakukan dengan 3 skenario, yaitu:

- a) Skenario 1: Penghitungan manual dengan seleksi atribut manual (tanpa CFS)

- b) Skenario 2: Penghitungan manual dengan seleksi atribut CFS

Nilai korelasi atribut2 dengan E	Nilai korelasi atribut2 dengan Loc
0.29274 12 i	0.08064 15 IOBlank
0.24914 8 d	0.07845 3 ev(g)
0.23472 11 b	0.05595 4 iv(g)
0.23459 6 v	0.04994 8 d
0.23126 4 iv(g)	0.0491 14 IOComment
0.22642 2 v(g)	0.04807 2 v(g)
0.22421 19 total_Opnd	0.04776 1 loc
0.22313 5 n	0.04771 20 branchCount
0.22293 18 total_Op	0.04592 91
0.21862 3 ev(g)	0.04089 19 total_Opnd
0.21327 1 loc	0.04025 16 uniq_Op
0.21324 20 branchCount	0.03949 17 uniq_Opnd
0.20518 13 IOCode	0.0393 6 v
0.18512 16 uniq_Op	0.03927 11 b
0.17966 14 IOComment	0.0391 12 i
0.17941 15 IOBlank	0.0391 10 e
0.15702 17 uniq_Opnd	0.03796 18 total_Op
0.13142 21 defects	0.03764 5 n
0.12361 9 i	0.03075 21 defects
-0.00107 7 i	-0.00478 7 i

- c) Skenario 3: Penghitungan otomatis dengan WEKA seleksi atribut manual dan CFS. Implementasi skenario 3 dapat dilihat pada gambar 2, 3 dan 4.

4. HASIL DAN PEMBAHASAN

Langkah awal penelitian ini adalah studi literatur dan pengumpulan data. Penelitian ini menganalisis dampak pemilihan atribut pada prediksi cacat perangkat lunak dengan SVM, pemilihan atribut secara manual dipilih berdasarkan metrik dasar Mc Cabe dan Halstead sedangkan pemilihan atribut otomatis dengan CFS.

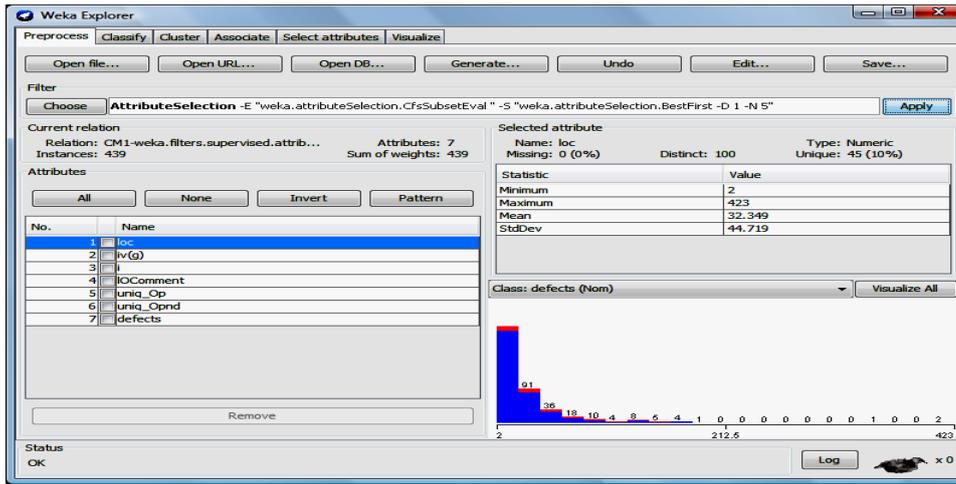
Pengumpulan data didapatkan dari Promise NASA dataset, yaitu dataset CM1, PC1, KC1, dan KC3. Langkah selanjutnya adalah seleksi atribut dari 21 atribut metrik Mc Cabe dan Halstead. Seleksi atribut dilakukan secara otomatis dengan CFS dan secara manual. CFS yang dilakukan pada dataset yang digunakan dalam penelitian ini menghasilkan seleksi atribut:

Tabel 9. Seleksi atribut dengan CFS

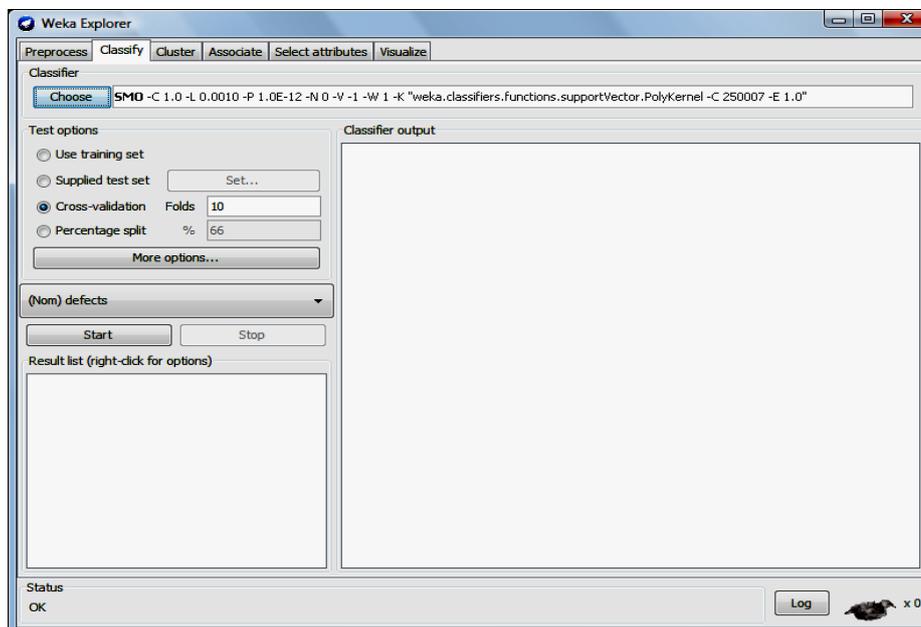
Dataset	Seleksi atribut CFS
CM1	loc, iv(g), i, lo comment, uniq_op, uniq_opnd
KC1	v, l, d, i, locomment, iobblank, uniq_opnd, branchCount
KC3	loc_blank, loc_code_and_comment, halstead_length, c
PC1	i, locomment, locCodeAndComment, IOBlank, uniq_opnd

Seleksi atribut yang dilakukan secara manual dilakukan dengan memisahkan metrik dasar dan metrik turunan Mc Cabe dan Halstead yang memiliki korelasi antar 1 atribut dengan lainnya. Setelah dilakukan seleksi atribut, maka dilakukan klasifikasi cacat perangkat lunak dengan Support Vector Machine (SVM) Pemilihan atribut terbaik untuk prediksi cacat perangkat lunak dilakukan dengan menguji akurasi, precision dan recall pada masing-masing dataset.

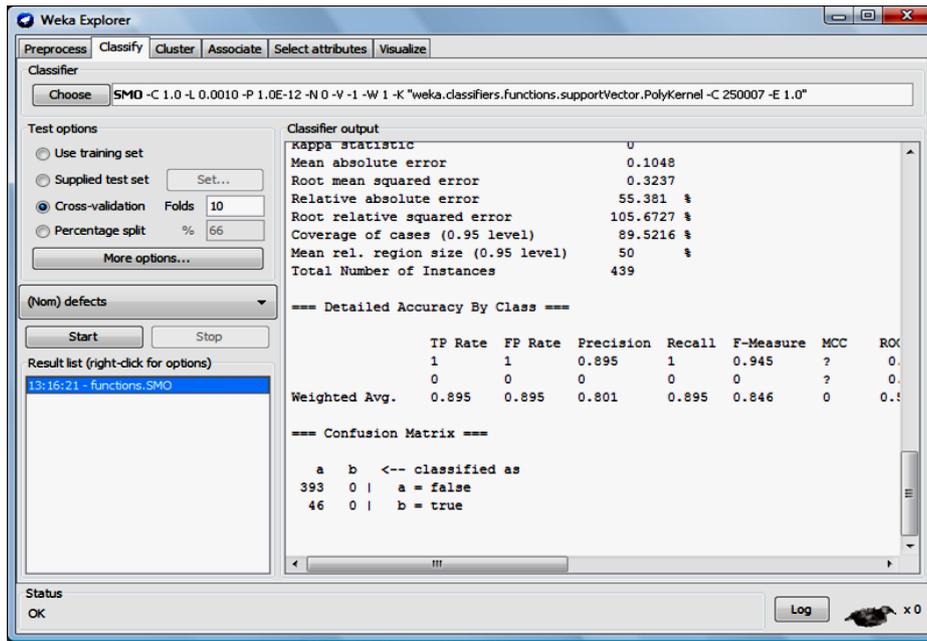
Hasil pengujian skenario 1, 2 dan 3 disajikan dalam tabel 12



Gambar 2. Tampilan CFS



Gambar 3. Tampilan SVM



Gambar 4 Tampilan hasil pengujian

Data set	Jumlah atribut	Skenario 1			Skenario 2		
		Akurasi	Precision	Recall	Akurasi	Precision	Recall
CM1	21	89,5	89,5	100	89,5	89,5	100
KC1	22	75,2	75,1	100	75,2	75,1	100
KC3	40	87,1	87,1	100	87,4	87,3	100
PC1	22	93,3	93,3	100	93,3	93,3	100

Tabel 12. Pengujian Akurasi, Precision dan Recall Skenario 1 dan 2

Data set	Skenario3								
	SVM			SVM+ S.A Manual			SVM+ CFS		
	Akurasi	Precision	Recall	Akurasi	Precision	Recall	Akurasi	Precision	Recall
CM1	89,5	89,5	100	89,5	89,5	100	89,5	89,5	100
KC1	75,9	76	99,2	75,2	75,1	100	75,2	75,1	100
KC3	87,4	87,3	100	87,1	87,1	100	87,4	87,3	100
PC1	93,2	93,3	99,9	93,3	93,3	100	93,3	93,3	100

Tabel 13. Pengujian Akurasi, Precision dan Recall Skenario 3

Dari hasil pengujian di atas, akurasi, precision dan recall hasil seleksi atribut, baik secara manual maupun CFS menunjukkan nilai yang lebih baik daripada SVM saja. Hal ini terlihat dalam nilai yang ditunjukkan pada dataset KC1 dan PC1.

Berdasarkan pengujian yang dilakukan, akurasi, precision dan recall untuk dataset CM1, KC1, dan PC1 menghasilkan nilai yang sama, perbedaannya terletak pada jumlah atribut hasil seleksi, yaitu apabila menggunakan seleksi atribut manual maka jumlah atribut tetap 8, sedangkan

dengan CFS jumlah atribut hasil seleksi tidak tetap, tergantung dengan penghitungan korelasi dari nilai masing-masing data.

Pada dataset KC3 mengandung 21 atribut metrik Mc Cabe dan Halstead ditambah atribut lain seperti jumlah keputusan, kerapatan keputusan, kompleksitas siklomatis ternormalisasi hingga berjumlah 40 atribut. Seleksi atribut dengan CFS menghasilkan akurasi dan precision lebih baik daripada pemilihan atribut secara manual, dengan recall sama, yaitu 100%.

5. KESIMPULAN

Prediksi cacat perangkat lunak telah dilakukan dengan SVM dan seleksi atribut. Seleksi atribut untuk prediksi cacat perangkat lunak secara manual cocok digunakan untuk data mentah yang belum diketahui nilai dari masing-masing atribut secara numerik karena menghasilkan akurasi, precision dan recall yang baik serta atribut yang dimasukkan konsisten, yaitu 8 atribut yang telah diseleksi secara manual. Seleksi atribut dengan CFS cocok digunakan untuk data yang sudah diolah dan memiliki nilai dari masing-masing metrik secara numerik karena berapa saja atribut yang dimasukkan, CFS akan menyeleksi atribut untuk kemudian diproses lebih lanjut.

6. DAFTAR PUSTAKA

- [1] Elish, Karim O, et al., 2008, *Predicting defect-prone software modules using support vector machines*. The Journal of Systems and Software 81 (2008) 649–660.
- [2] Hall, Mark A., 1999, *Correlation-based Feature Selection for Machine Learning*, The University of Waikato
- [3] Khoshgoftaar, T.M and N. Seliya, *The necessity of Assuring Quality in Software Measurement Data*, Proc. 10th International Symp. Software Metrics (METRICS '04), IEEE CS Press, 2004, pp.119-130
- [4] Menzies T., J. DiStefano, A. Orrego and R. Chapman, *Assessing Predictors of Software Defects*, Proc. Workshop Predictive Software Models, 2004.
- [5] Pressman, Roger S, 2010, *Software Engineering: A Practitioner's Approach*, Mc Graw Hill.
- [6] Promise NASA Dataset. <http://nasa-softwaredefectdatasets.wikispaces.com/>. Tanggal akses: 12 Maret 2013
- [7] Tan, Pang-Ning , et all. 2006. *Introduction to Data Mining*. Pearson Addison Wesley.
- [8] Zhang, Hongyu, et all, 2007, *Predicting Defective Software Components from Code Complexity Measures*, IEEE International Symposium on Pacific Rim Dependable Computing
- [9] Zhang, Hongyu, et all, 2009, *An Investigation of the Relationships between Lines of Code and Defects*, IEEE Proceeding ICSM

(10)
**PEMBUATAN MODEL PENILAIAN KELAYAKAN ASPEK MANUSIA
 PADA PROYEK PERANGKAT LUNAK UNTUK MENINGKATKAN KUALITAS PRODUK
 PERANGKAT LUNAK**

Parma Hadi Rantellinggi¹⁾, Sarwosri²⁾

¹⁾Program Studi Teknik Informatika Universitas Sains dan Teknologi Jayapura
 Jl. Raya Sentani Padang Bulan, Abepura, Jayapura
 e-mail : parma.hadi12@mhs.if.its.ac.id

²⁾Jurusan Teknik Informatika ITS Surabaya
 Jl. Teknik Kimia Kampus Teknik Informatika Institut Teknologi Sepuluh Nopember
 Surabaya (031) 5939214
 e-mail : sarwosri@if.its.ac.id

ABSTRAK

Untuk membangun suatu perangkat lunak yang berkualitas ada tiga aspek yang dibutuhkan yaitu aspek teknis, aspek manajerial dan aspek manusia. Saat ini banyak penelitian tentang proyek perangkat lunak, tetapi lebih menekankan pada aspek teknis dan aspek manajerial, padahal aspek manusia juga memegang peranan, dalam penulisan ini dibahas mengenai aspek manusia, karena aspek manusia juga sangat berpengaruh dalam menghasilkan perangkat lunak yang berkualitas, aspek manusia mempelajari tentang sisi psikologi individu yang akan terlibat dalam proyek perangkat lunak. Dalam langkah awal dilakukan terlebih dahulu pemetaan *skill* kompetensi aspek teknis, manajerial dan aspek manusia dalam profesi yang terlibat dalam membangun perangkat lunak, setelah itu dilakukan penyebaran kuisioner untuk mendapatkan informasi kelayakan standar yang telah dibuat untuk sisi aspek manusia. Hasil dari pengolahan data tersebut nantinya akan dijadikan acuan untuk menilai tiap-tiap individu dalam proyek perangkat lunak atau untuk perekrutan tenaga kerja yang baru dalam memenuhi kebutuhan tenaga kerja pada suatu proyek perangkat lunak. Kuisioner disebarkan secara acak dan dipilih empat profesi yaitu *programmer*, *interface designer*, *Project manager* dan *software tester* sebagai sampel untuk mendapatkan informasi. Hasil yang diperoleh menunjukkan 93,7% setuju dengan model yang dirancang dan 6,3% tidak setuju dengan model yang dirancang. Model ini bisa diterapkan untuk merekrut tenaga kerja dalam suatu proyek perangkat lunak untuk meningkatkan kualitas produk perangkat lunak.

Kata Kunci : Aspek manusia, Penilaian Kelayakan, Skill kompetensi, Proyek perangkat lunak, Kualitas

1. PENDAHULUAN

Dalam membangun suatu proyek perangkat lunak ada tiga aspek utama yang semuanya bertujuan untuk menghasilkan perangkat lunak yang berkualitas tinggi, yang pertama adalah aspek teknis. Aspek yang kedua adalah aspek manajerial dan aspek yang ketiga adalah aspek manusia.

Banyak ditemui penelitian tentang proyek perangkat lunak tetapi lebih menekankan pada aspek teknis dan manajerial. Padahal untuk menghasilkan perangkat lunak yang baik bukan hanya memperhatikan dua aspek ini saja, tetapi harus memperhatikan semua aspek termasuk aspek manusia yang berpengaruh dalam rekayasa perangkat lunak.

Dalam penulisan ini akan membahas pentingnya tindakan yang terkait dengan aspek manusia dalam proyek perangkat lunak, karena ada beberapa faktor yang mempengaruhi seperti kemampuan kognitif, psikologi, kerja sama tim, kemampuan komunikasi antar anggota tim dan manajemen konflik, yang dapat mempengaruhi suatu kegiatan tim proyek

perangkat lunak dalam menghasilkan perangkat lunak yang berkualitas.

Untuk itu diperlukan standar penilaian untuk menilai tiap-tiap individu dalam proyek perangkat lunak untuk mengetahui kemampuan tiap individu, atau untuk perekrutan tenaga kerja yang baru untuk memenuhi kebutuhan tenaga kerja pada suatu proyek perangkat lunak.

2. TINJAUAN PUSTAKA

2.1 ASPEK TEKNIS

Pada aspek teknis umumnya membahas mengenai kegiatan seperti desain, implementasi dan pengujian. Salah satu pendekatan yang di gunakan untuk pengukuran aspek teknis dari sistem perangkat lunak biasanya dengan atribut yang terkait dengan kualitas. Sebagai contoh *International Standard Organization* (ISO) merincikan atribut kualitas perangkat lunak sebagai berikut fungsi, keandalan, kemudahan penggunaan, kemampuan perangkat lunak untuk dipelihara dan portabilitas. Setiap atribut ini terdiri dari 2 – 5

subkarakteristik yang lebih spesifik (Tessey, 2002).

Menurut McCall yang dikutip oleh Roger S. Pressman dalam bukunya *Rekayasa Perangkat Lunak*, menjelaskan bahwa ada 3 aspek yang penting dalam produk perangkat lunak meyangkut kualitasnya, yang pertama *product operation* melingkupi *correctness, reliability, usability, integrity, efficiency*, yang kedua *product transition* melingkupi *portability, reusability, interoperability*, dan yang ketiga *product revision* melingkupi *maintainability, flexibility, testability*.

2.2 ASPEK MANAJERIAL

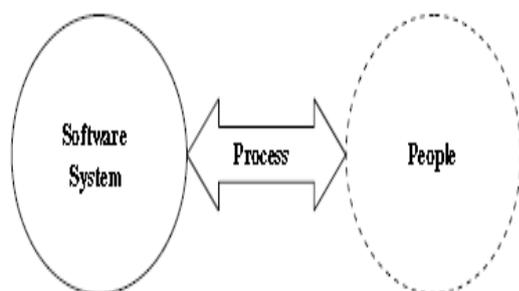
Kualitas perangkat lunak seringkali banyak dipengaruhi oleh keputusan-keputusan manajemen seperti juga banyak dipengaruhi keputusan-keputusan yang berkaitan dengan teknologi yang digunakan untuk menghasilkan perangkat lunak yang berkualitas.

Aspek manajerial ini berhubungan dengan keputusan-keputusan yang akan memiliki efek yang signifikan terhadap kualitas produk perangkat lunak yang akan di hasilkan, sebagai contoh keputusan-keputusan tersebut seperti keputusan-keputusan prakiraan, keputusan – keputusan penjadwalan dan keputusan – keputusan yang berorientasi risiko (Pressman, 2010).

2.3 ASPEK MANUSIA

Pada sub bab ini akan di bahas mengenai aspek yang ketiga yaitu aspek manusia, dimana aspek ini berkaitan dengan komunikasi antar rekan kerja, kolaborasi dengan pengguna dan proses belajar. (Hazzan dan Hadar, 2008) Ilustrasinya dapat dilihat pada gambar.1.

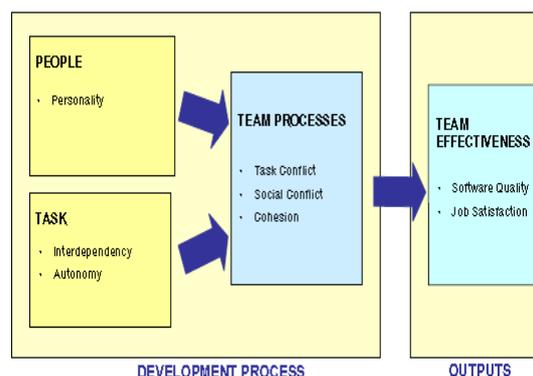
Setelah itu untuk mengeksplorasi informasi maka digunakan penilaian yang di bagi menjadi 4 bagian yaitu *the cognitive – individual cell, the social – individual cell, the cognitive – team cell dan the social – team cell* (Hazzan dan Hadar, 2008).



Gambar 1. Ilustrasi sebuah hubungan antara tiga komponen (Hazzan dan Hadar, 2008)

Dalam penelitian lainnya dijelaskan bagaimana menganalisis hubungan antar tiga karakteristik

pengembang yang didalamnya ada lima faktor utama yaitu personal, spesifikasi tugas dan interaksi dalam tim dengan dua karakteristik keluaran yaitu kualitas pengembangan perangkat lunak dan kepuasan anggota tim (Acuña et al, 2009). Penelitian ini menggunakan model seperti pada gambar.2 untuk penelitian empiris. Model ini membagi variabel yang diteliti kedalam empat komponen dasar yaitu orang, tugas, proses tim dan efektivitas tim.



Gambar 2. Modifikasi model input, proses dan output (Acuña et al, 2009)

Selain menggunakan studi empiris penelitian ini menggunakan *quasi experiment* (eksperimen semu) sehingga lebih mudah untuk menggeneralisasi hasil penelitian terhadap lingkungan kerja dimana proses penelitian berlangsung.

Di dalam proses penelitian, kelompok akan menjawab angket yang diberikan, tapi angket tidak di alokasikan secara acak. Selain itu sudah ada gambaran hipotesa yang dibuat dan juga penjelasan mengenai parameter ,variabel, partisipan, subjek, instrumen pengukuran dan prosedur eksperimen semu yang digunakan.

Selanjutnya menurut Atkinson et al dalam buku psikologi industri dan organisasi karangan Sutarto Wijono, memberi arti psikologi sebagai berikut psikologi sebagai studi ilmiah mengenai proses perilaku dan proses mental.

Psikologi industri/organisasi adalah suatu studi ilmiah tentang perilaku dan proses mental manusia dalam industri atau organisasi dengan berorientasi pada sistem kegiatan yang terkoordinasi dari suatu kelompok orang yang bekerja secara kooperatif untuk mencapai tujuan yang sama di bawah otoritas dan kepemimpinan tertentu (Wiyono, 2010).

Karena orientasi psikologi industri/organisasi lebih fokus pada individunya agar dapat mengembangkan potensi dan kompetensi secara optimal di tempat kerja. Banyak faktor yang mempengaruhi pengembangan sumber daya manusia dalam organisasi, aspek-aspek yang lebih potensial yang terus-menerus berkembang dalam

suatu organisasi seperti melatih potensi sumberdaya manusia agar lebih berprestasi dan produktif, mengelola frustrasi, konflik, dan stres kerja. Meningkatkan kepuasan kerja karyawan, memotivasi karyawan agar lebih produktif, dan membina hubungan komunikasi antar atasan-bawahan (Wiyono, 2010).

Dari perspektif sosial, psikologi industri /organisasi juga menambah keefektifan pekerja, dapat memperbaiki kualitas produksi dan dapat mengurangi biaya materi atau barang-barang, mengurangi perbaikan dan biaya penempatan ulang dengan meningkatkan efisiensi organisasi, dan dapat berhasil mengurangi setiap kegiatan seperti menunggu dalam satu jalur tertentu (Wiyono, 2010).

2.4 ASPEK PENILAIAN

Pada penulisan ini aspek penilaian terdiri dari dua aspek penilaian yaitu penilaian kompetensi keahlian dan penilaian psikologi. Dimana penilaian kompetensi berfungsi untuk menilai kemampuan atau kompetensi tiap individu dalam suatu tim proyek perangkat lunak atau untuk perekrutan tenaga kerja yang baru sehingga dapat menemukan orang yang pas pada posisi yang sedang di butuhkan untuk terlibat dalam suatu proyek perangkat lunak.

Misalnya suatu proyek perangkat lunak membutuhkan seorang *programmer* yang baru maka kriteria kompetensi untuk mendapatkan programer yang baik adalah mempunyai kemampuan dalam satu atau lebih bahasa pemrograman, teliti, kemampuan logika yang baik, disiplin, kreatif, memiliki kemampuan memecahkan masalah dan memiliki kemampuan yang baik dalam menganalisa.

Selain itu programer tersebut juga akan dinilai pada aspek psikologi dimana akan dinilai kemampuan seperti interaksi antar individu, faktor emosional, kerjasama antar tim, kemampuan memimpin dan siap dipimpin, yang nantinya semua kriteria itu akan dinilai untuk mendapatkan seorang programer yang baik.

3. METODE PENELITIAN

3.1 IDENTIFIKASI ASPEK PSIKOLOGI

Pada tahap ini akan diidentifikasi dahulu apa saja yang mempengaruhi seorang karyawan yang bekerja pada suatu proyek perangkat lunak. Tujuan umumnya yaitu untuk mengetahui tingkat kepuasan kerja seorang karyawan dengan pekerjaan atau tugas yang di berikan atasan kepada karyawan tersebut, selain itu tujuannya agar bagaimana sebuah perusahaan dapat menentukan standar psikologi pada calon karyawan yang akan di rekrut agar mendapat karyawan yang sesuai dengan yang dibutuhkan untuk di tempatkan pada suatu tugas atau pekerjaan.

Aspek Psikologi umumnya berhubungan

dengan kejiwaan karyawan meliputi minat, ketentraman kerja, sikap terhadap kerja, bakat dan keterampilan. Kepuasan kerja mempunyai peranan penting terhadap prestasi kerja karyawan, ketika seorang karyawan merasakan kepuasan dalam bekerja maka karyawan tersebut akan berupaya semaksimal mungkin dengan segenap kemampuan yang dimiliki untuk menyelesaikan tugasnya, yang akhirnya akan menghasilkan kinerja dan pencapaian yang baik bagi perusahaan. Kepuasan kerja mempunyai pengaruh yang cukup besar terhadap produktivitas organisasi baik secara langsung maupun tidak langsung. Ketidakpuasan merupakan titik awal dari masalah-masalah yang muncul dalam organisasi seperti kemangkiran, konflik manager-pekerja dan perputaran karyawan. Dari sisi pekerja, ketidakpuasan dapat menyebabkan menurunnya motivasi, menurunnya moral kerja, dan menurunnya tampilan kerja

3.2 ANALISIS PENENTUAN FAKTOR KOMPETENSI ASPEK MANUSIA

Dalam tahap ini merupakan tahap di mana akan di lakukan observasi kompetensi pada aspek manusia menyangkut psikologinya, dimana sudah di tentukan dahulu faktor – faktor kompetensi apa saja yang harus dimiliki oleh tiap – tiap profesi yang bekerja dalam proyek perangkat lunak dari aspek manusianya.

Langkah selanjutnya akan dilakukan penyebaran kuisioner kepada pihak – pihak yang berkecimpung dalam proyek perangkat lunak, untuk diisi dan juga untuk mendapatkan saran atau masukan dari pihak – pihak tersebut, dimana dalam kuisioner tersebut bisa diamati apakah standar yang sudah dibuat memenuhi kriteria yang di butuhkan pada umumnya di perusahaan yang bergerak dalam proyek perangkat lunak atau belum, jika belum maka akan didapatkan informasi tambahan mengenai standar aspek manusia untuk bisa diketahui bahwa masih banyak aspek manusia lainnya yang dibutuhkan selain standar yang dibuat.

3.3 PEMETAAN HUBUNGAN ASPEK MANUSIA DENGAN ASPEK TEKNIS DAN MANAJERIAL

Pada tahap ini akan di kelompokkan dulu skill kompetensi pada setiap profesi yang terlibat dalam suatu proyek perangkat lunak agar terlihat jelas kompetensi apa saja yang harus ada pada tiap individu mulai dari *Bisnis Analis*, *Sistem Analis*, *Infrastruktur Analis*, *Change Manager Analist*, *Database Manager Analist*, *Data Base Designer*, *Algoritmik Designer*, *User Interface Designer*, *Arsitek Designer*, *Programmer*, *Project Manager* dan *Software Tester*.

Setelah mendapatkan skill kompetensi tiap profesi maka dibuat tabel pengelompokan skill

kompetensi nya yang di bagi kedalam 3 aspek yaitu aspek teknis, aspek manajerial dan aspek manusia dan dapat dilihat contohnya pada tabel 1, tujuannya agar dapat menunjukkan bahwa tiap skill kompetensi dapat di sederhanakan ke dalam 3 aspek sebagai contoh skill profesi seorang *programmer* dari segi aspek teknis harus menguasai satu atau lebih bahasa pemrograman, kemampuan logika yang baik dan kemampuan yang baik dalam menganalisa, sedangkan dari aspek manajerial seorang *programmer* harus mampu memecahkan masalah, dan dari aspek manusia seorang *programmer* harus teliti, disiplin, bertahan dalam tekanan, rajin, bekerja kerja, melaksanakan tugas, patuh dan kreatif. Contoh lainnya skill individu seorang *software tester* dari aspek teknis harus memiliki kemampuan analisis yang baik, memiliki kemampuan teknis (*hard skill*) dan mampu beradaptasi dan belajar hal baru, dari aspek manajerial seorang *software tester* harus mampu mengorganisasi dan memprioritaskan, mampu berkomunikasi dan mampu menganalisa strategi bisnis dan mampu bekerja tanpa petunjuk atasan, dan dari aspek manusia seorang *software tester* harus memiliki kecerdasan, memiliki kepekaan, suka mencoba, betah dan terampil, paham teknis, teliti, pantang menyerah, aktif dan jeli.

No	Profesi	Aspek Teknis	Aspek Manajerial	Aspek Manusia
1.	Programmer	1. Menguasai satu atau lebih bahasa pemrograman 2. Kemampuan logika yang baik 3. Kemampuan yang baik dalam menganalisa	1. Kemampuan memecahkan masalah	1. Teliti 2. Disiplin 3. Kreatif 4. Bertahan dalam tekanan 5. Bekerja keras 6. Rajin 7. Melaksanakan tugas 8. Patuh
2.	Software Tester	1. Kemampuan analisis yang baik 2. Kemampuan teknis (<i>hard skill</i>) 3. Mampu beradaptasi dan belajar hal baru	1. Mampu mengorganisasi dan memprioritaskan 2. Mampu berkomunikasi 3. Mampu menganalisa strategi bisnis 4. Mampu bekerja tanpa ada petunjuk dari atasan	1. Memiliki kecerdasan 2. Punya kepekaan 3. Suka mencoba 4. Betah dan terampil 5. Paham Teknis 6. Teliti 7. Pantang Menyerah 8. Aktif 9. Jeli
3.	Interface Designer	1. Mampu memahami warna, tata letak, tipografi, dan ikonografi. 2. Memiliki pemahaman tentang psikologi dan demografi pengguna agar dapat merancang antarmuka	1. Memiliki kemampuan organisasi yang baik dan kemampuan untuk mengelola beberapa proyek pada saat yang sama.	1. Mempunyai jiwa seni 2. Paham konsep human common interaction 3. Menguasai tools desain 4. Paham grafis

Tabel 1. Contoh Pengelompokan Skill kompetensi ke dalam aspek teknis, manajerial dan aspek manusia

3.4 PENGUJIAN DAN EVALUASI HASIL PEMETAAN

Pada tahap ini akan dilakukan evaluasi hasil pemetaan yang telah dibuat dengan cara menyebarkan kuisioner kepada para praktisi dalam proyek perangkat lunak. Penyebaran kuisioner ini dilakukan di dua tempat.

Kemudian hasil kusioner tersebut akan dievaluasi, yang nantinya akan diperoleh hasil apakah standar pemetaan yang telah dibuat sudah sesuai dengan standar atau kebutuhan yang biasanya dipakai oleh para praktisi dalam proyek perangkat lunak.

3.5 PERBAIKAN KOMPONEN ASPEK MANUSIA

Pada tahap ini akan dilakukan perbaikan komponen pada aspek manusia, dari hasil pengumpulan informasi yang di peroleh dari kuisioner yang telah disebarkan.

Data yang diperoleh nanti akan diolah, untuk lebih jelasnya akan di jelaskan pada bab 4, kemudian hasilnya akan digunakan sebagai refisi atau perbaikan dan untuk menjadi acuan yang dapat digunakan dalam menarik kesimpulan pada Bab 5.

4. UJI COBA

Pada tahap penyebaran kusioner, disebarkan secara acak dan diambil empat profesi dalam proyek perangkat lunak sebagai sampel untuk mendapatkan informasi tentang kelayakan aspek manusia dalam proyek perangkat lunak. Empat profesi tersebut adalah *interface designer*, *programmer*, *project manager* dan *software tester*.

Kuisioner yang telah diisi dan dikumpulkan, kemudian diolah sehingga mendapatkan hasil seperti tabel dibawah ini.

Tabel 2. Hasil pengolahan kuisioner

<i>Interface Designer</i>		
<i>Aspek Manusia</i>	<i>Ya</i>	<i>Tidak</i>
1. Mempunyai jiwa seni	6	
2. Paham konsep human common interaction	6	
3. Menguasai tool desain	6	
4. Paham grafis	6	
Persentase	100%	
<i>Programmer</i>		
<i>Aspek Manusia</i>	<i>Ya</i>	<i>Tidak</i>
1. Orang yang Teliti	6	
2. Orang yang Disiplin	6	
3. Orang yang Kreatif	6	
4. Orang yang bertahan bekerja dalam tekanan	5	1
5. Bekerja keras	5	1
6. Orang yang rajin	6	

<i>No</i>	<i>Profesi</i>	<i>Aspek Teknis</i>	<i>Aspek Manajerial</i>	<i>Aspek Manusia</i>
4.	Project Manager	1.Kemampuan presentasi 2.Kemampuan manajemen data 3.Familiar dengan <i>Software Development Life Cycle (SDLC)</i> 4.Memiliki pengetahuan dalam pengembangan aplikasi 5.Memiliki pengetahuan dalam database	2.Kemampuan untuk memberikan umpan balik dan memberikan solusi untuk memecahkan masalah desain dan kegunaannya. 3.Keterampilan dan kemampuan negosiasi untuk memberikan umpan balik dan datang dengan solusi untuk memecahkan masalah desain 1.Berpikir kritis dan mampu memecahkan masalah 2.Kemampuan perencanaan dan pengorganisasian 3.Kemampuan dalam pengambilan keputusan 4.Kemampuan mempengaruhi dan memimpin 5.Kemampuan bernegosiasi 6.Kemampuan dalam manajemen konflik 7.Kemampuan evaluasi	1.Kemampuan dalam berkomunikasi 2.Dapat bekerja sama 3.Mampu menyesuaikan diri 4.Mampu memimpin 5.Kemampuan meyakinkan dan mempengaruhi 6.Visioner 7.Plannya bagus 8.Berpengalaman dalam hal proyek sebelumnya
			7.Bertanggung jawab dalam melaksanakan tugas	6
			8. Patuh	5
			Persentase	93,7%
				6,3%
			<i>Project Manager</i>	
			<i>Aspek Manusia</i>	<i>Ya</i>
			1. Kemampuan dalam berkomunikasi	6

2. Dapat bekerja sama	6	
3. Mampu menyesuaikan diri	6	
4. Mampu memimpin	6	
5. Kemampuan meyakinkan dan mempengaruhi	5	1

6. Visioner	6	
7. Mempunyai plan yang bagus	5	1
8. Berpengalaman	5	1
Persentase	93,8%	6,3%
<i>Software Tester</i>		
<i>Aspek Manusia</i>	<i>Ya</i>	<i>Tidak</i>
1. Memiliki kecerdasan	6	
2. Punya kepekaan	6	
3. Suka mencoba	6	
4. Betah dan terampil	6	
5. Paham teknis	6	
6. Orang yang teliti	6	
7. Pantang menyerah	6	
8. Aktif	6	
9. Jeli	6	
Persentase	100%	

Dari gambaran tabel di atas dapat dijelaskan bahwa untuk *interface designer*, kriteria aspek manusia yang di tawarkan disetujui oleh semua responden, begitu juga dengan kriteria yang di tawarkan pada *software tester*.

Tetapi pada *programmer*, tabel 2 menunjukkan 93,7 % setuju dengan semua kriteria aspek manusia yang di tawarkan, sedangkan 6,3% tidak setuju untuk kriteria yang di tawarkan yaitu bertahan bekerja dalam tekanan, bekerja keras dan patuh, begitu pula dengan *project manager* 93,7 % setuju dengan kriteria aspek manusia yang di tawarkan dan 6,3 % tidak setuju untuk kriteria yang di tawarkan yaitu kemampuan meyakinkan dan mempengaruhi, mempunyai plan yang bagus dan berpengalaman.

Dari hasil kuisioner dapat dilihat bahwa hampir semua responden setuju dengan kriteria aspek manusia yang dirancang.

Hasil analisis, bisa di gambarkan bahwa jika ingin mendapatkan atau merekrut *interface designer* dan *software tester* maka standar yang diusulkan dapat dijadikan acuan. *programmer* dapat bekerja lebih baik jika bekerja tanpa tekanan, bisa bekerja sesuai *deadline* dan dapat bertanggung jawab dalam mengatasi permasalahan. *Project manager* tidak harus mempunyai kemampuan meyakinkan dan mempengaruhi asalkan memiliki kecepatan dan *agility* dalam masa pengembangan proyek dan memiliki sifat mendengar dan memberi masukan pada bawahannya.

Jika ingin mendapatkan atau merekrut tenaga kerja yang baru maka kriteria aspek manusia harus juga di perhatikan, karena ada hubungan antara aspek teknis dan aspek manajerial.

5. KESIMPULAN DAN SARAN

5.1 KESIMPULAN

1. Pemanfaatan dari model yang diusulkan, dapat digunakan sebagai standar aspek manusia untuk mengetahui sisi psikologi tiap individu atau untuk merekrut tenaga kerja pada suatu proyek perangkat lunak. Karena untuk menghasilkan perangkat lunak yang baik tidak hanya dipengaruhi oleh aspek teknis dan aspek manajerial saja, tetapi aspek manusia juga harus diperhatikan.
2. Dari hasil uji coba dari model yang di usulkan 93,7 % responden setuju dengan kriteria aspek manusia yang di tawarkan pada profesi *programmer* dan *project manager*, sedangkan 6,3 % responden tidak setuju untuk kriteria yang ditawarkan. Pada profesi *interfece designer* dan *software tester* semua responden setuju dengan kriteria aspek manusia yang ditawarkan.

5.2 SARAN

1. Standar yang telah dibuat dapat dijadikan acuan sebagai standar untuk menilai tiap-tiap individu pada profesi yang terlibat dalam proyek perangkat lunak, karena hasil penilaian responden dari empat profesi yang terlibat dalam proyek perangkat lunak yang diambil sebagai sampel menunjukkan rata-rata setuju dengan standar aspek manusia yang diusulkan.
2. Studi tentang aspek manusia dalam membangun perangkat lunak ini dapat diteliti lebih lanjut dengan menambahkan variabel umur dan jenis kelamin dalam pengumpulan data pada tiap – tiap profesi yang terlibat dalam proyek perangkat lunak, sebab pendapat tiap – tiap individu bisa lebih bervariasi dan bisa dipengaruhi oleh faktor umur dan jenis kelamin.

6. DAFTAR PUSTAKA

- [1] Acuña, S. T., Gómez, M., Juristo, N (2009), *How do personality, team process and taks characteristic relate to job satisfaction and software quality?*. Information and Software Technology 51, 627 – 639
- [2] Hazzan, O., Hadar, I (2008), *Why and how*

human – related measures support software development process?. The Journal of Systems and Software 81, 1248 – 1252

- [3] Pressman, R. S (2012). *Rekayasa Perangkat Lunak – Buku Satu, Pendekatan Praktisi Edisi 7*. Yogyakarta : Panerbit ANDI
- [4] Tasse, G (2002). *The Economic Impacts of Inadequate Infrastructure for Software Testing*. National Research Triangle Park, NC
- [5] Wijono, S (2010). *Psikologi Industri dan Organisasi: Dalam Suatu Bidang Gerak Psikologi Sumber Daya Manusia*. Jakarta :Kencana

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

(11)

SELEKSI FITUR TASK-STATE FMRI MENGGUNAKAN NON-OVERLAPPING CROSS-CORRELATION

Agus Eko Minarno¹, Fitri Bimantoro², Arrie Kurniawardhani²

¹Jurusan Teknik Informatika

Universitas Muhammadiyah Malang

Jl. Tlogomas 246 Malang (0341) 464318

²Jurusan Teknik Informatika ITS Surabaya

Jl. Teknik Kimia Kampus Teknik Informatika Institut Teknologi Sepuluh Nopember

Surabaya (031) 5939214

agoes.minarno@gmail.com , beema.yichi@gmail.com , emailkuwardhani@gmail.com

ABSTRAK

Pada penelitian sebelumnya teridentifikasi beberapa *voxel* yang *overlapping* pada saat *resting-state* dan *task state*. Metode untuk menentukan *connectivity map* melibatkan daerah yang *overlapping*, kurang optimal dalam menggambarkan pola task-state sebagai ciri. Pada penelitian ini, diusulkan sebuah metode baru untuk pemilihan fitur untuk meningkatkan akurasi dan mengurangi waktu komputasi dalam menentukan *voxel* yang signifikan pada saat *task-state* menggunakan metode *non-overlapping area*. Pemilihan fitur untuk menentukan *voxel* yang signifikan menggunakan *cross correlation*, dan mengambil *voxel* dengan nilai korelasi yang berada diatas korelasi rata-rata. Tahapan berikutnya adalah penentuan nilai ambang batas (threshold) untuk menentukan jumlah *voxel* yang dipilih sebagai fitur. Fitur yang terpilih akan diberi label sesuai dengan stimulus yang diberikan, yaitu picture dan sentence, kemudian diseleksi untuk mendapatkan *voxel* yang *non-overlapping* antara *stimulus picture* dengan *stimulus sentence*. Hasil rata-rata dari 6 subyek, metode yang melibatkan *overlapping area* menggunakan *classifier SVM* diperoleh *precision*, *recall*, dan *accuracy* masing-masing 94.2%, 95.1% , 94.6% dan waktu komputasi 0.021 detik. Sedangkan metode *non-overlapping area* memiliki *precision*, *recall*, dan *accuracy* masing-masing 95.0%, 95.3% , 95.1% dan waktu komputasi 0.019 detik. Pemilihan fitur menggunakan metode *non-overlapping area* memiliki akurasi dan waktu komputasi yang lebih baik dari metode yang melibatkan *overlapping area*, dalam menentukan *connectivity map*.

Kata kunci : *feature selection, task-state, cross-correlation, voxel-based selection, non-overlapping*

1. PENDAHULUAN

Penentuan daerah otak yang aktif pada saat *task-state* telah banyak dilakukan dengan berbagai metode. Beberapa penelitian telah dilakukan untuk mengetahui daerah yang terkoneksi secara fungsional untuk menghasilkan *connectivity map*. Metode yang digunakan dalam menghasilkan *connectivity map* antara lain menggunakan pendekatan *region-based*, *seed-based*, dan *voxel-based*. Perbandingan *region-based* dan *voxel-based* telah dilakukan, masing-masing metode memiliki kekurangan dan kelebihan dalam menentukan *connectivity map*. Pendekatan metode *voxel-based* lebih rinci dalam menunjukkan daerah yang aktif dan memberikan akurasi yang tinggi, namun memiliki tingkat kompleksitas yang lebih tinggi dibandingkan dengan metode *region-based* yang

memberikan akurasi rendah namun memiliki tingkat kompleksitas yang lebih rendah pula (Hayasaka et al., 2010).

Penelitian tentang metode pengukuran *functional connectivity* menggunakan *resting-state fMRI* telah dilakukan sebelumnya. Metode pengukuran yang diuji adalah *cross-coherence*, *partial coherence*, *cross-correlation*, dan *partial cross-correlation*. Pada penelitian tersebut menunjukkan bahwa metode *cross-correlation* lebih stabil dibandingkan dengan metode lainnya (Fiecas et al., 2012).

Pada penelitian lain, metode *cross-correlation* digunakan untuk menentukan bagian dari otak yang terkoneksi secara fungsional atau menentukan *connectivity maps* dari *region of*

interest (ROIs). Metode *inter-voxel cross-correlation* tersebut telah berhasil digunakan untuk menentukan ROIs dari *resting-state*. Pendekatan yang digunakan untuk menentukan ROIs pada *resting-state* menggunakan metode *inter-voxel cross-correlation* dengan bantuan *task-state* untuk menghasilkan *connectivity map*. Bagian yang teridentifikasi terdiri dari daerah *resting-state*, *task-state*, dan daerah *overlapping*. Bagian yang teridentifikasi tersebut berada pada bagian kiri dan kanan otak pada daerah *motor cortices* (Golestani et al., 2011). Penelitian tersebut menghasilkan *connectivity map* yang menggambarkan adanya daerah *overlapping* antara *resting-state* dan *task-state*. Metode pada (Golestani et al., 2011) diaplikasikan pada *starplus dataset*. Metode *inter voxel cross corelation* digunakan untuk mengukur kedekatan ciri antar beberapa voxel, sehingga dapat diketahui voxel mana saja yang aktif saat mendapat stimulus *picture* dan voxel mana saja yang aktif saat melihat *sentence*. Kemudian mencari daerah yang *non-overlapping* untuk dijadikan fitur yang paling signifikan.

Pada pemilihan fitur, mereduksi dimensi fitur untuk mengurangi kompleksitas komputasi merupakan masalah penting pada data fMRI, karena data yang dimiliki fMRI merupakan data 4 dimensi yang memiliki jumlah rata-rata lebih dari 5000 fitur pada dataset Starplus. Oleh karena itu, dibutuhkan metode untuk mereduksi fitur pada data fMRI. Dengan adanya voxel yang *overlapping* dan *non-overlapping* antara *resting-state* dan *task-state* maka voxel tersebut dapat dimanfaatkan untuk dijadikan fitur. Voxel yang *overlapping* merupakan voxel yang aktif ketika kedua stimulus ditampilkan, sementara voxel yang *non-overlapping* merupakan voxel yang aktif hanya pada saat salah satu stimulus ditampilkan. Tujuan pemilihan fitur *non-overlapping* untuk mendapatkan voxel dapat memberikan perbedaan pada saat stimulus *picture* dan stimulus *sentence*.

Pada penelitian ini, diusulkan metode baru untuk pemilihan fitur dengan memanfaatkan fitur pada daerah *non-overlapping* dan mengembangkan *inter voxel cross corelation*, untuk menentukan *connectivity map* menggunakan Starplus dataset.

2. MATERIAL DAN METODE

2.1 DATASET

Functional Magnetic Resonance Imaging (fMRI) menggunakan teknik berupa citra “*snapshot*”, dimana setiap snapshotnya memiliki informasi irisan otak dalam satuan waktu tertentu. Citra “*snapshot*” tersebut memiliki informasi tentang perubahan sinyal BOLD, yang menggambarkan aktivitas syaraf (Baxter PR et al., 2007). Informasi tentang sinyal BOLD disimpan pada bagian terkecil dari citra snapshot otak disebut *voxel (volume element)*. Data fMRI diperoleh dengan cara membandingkan fungsi pada beberapa kondisi di waktu yang berbeda atau pada subyek yang berbeda, untuk mengevaluasi besaran relatif dari berbagai macam respon. Sinyal BOLD yang diperoleh dari kondisi yang berbeda tersebut kemudian dianalisa dengan berbagai macam metode untuk mencari korelasi dan *functional connectivity* pada jaringan syaraf. Kemunculan fMRI diharapkan dapat digunakan untuk membantu dalam perencanaan sebelum melakukan pembedahaan, diagnosis awal, mengetahui perkembangan terapi, dan mengidentifikasi kelainan fungsi otak.

Functional connectivity adalah suatu organisasi, hubungan saling keterkaitan, dan kinerja yang menggabungkan beberapa wilayah otak yang berbeda, untuk mengetahui interaksi dari beberapa wilayah otak yang terangkai secara simultan saat ada stimulus tertentu. Seperti *resting state* merupakan korelasi antar wilayah pada komponen-komponen jaringan syaraf yang berbeda di daerah visual, *language*, motor, dan sistem memori, dimana korelasi tersebut akan berubah selama ada *task* yang berkelanjutan (Baxter PR et al., 2007). Analisis *functional connectivity* merupakan salah satu hal yang penting dalam memahami perubahan aktivitas syaraf. Dari analisis *functional connectivity* tersebut, diharapkan dapat memberikan informasi berupa fungsi-fungsi apa saja yang sedang aktif di daerah otak, sehingga aktivitas apa yang sedang dilakukan oleh seseorang, atau emosi apa yang sedang dirasakan seseorang dapat diketahui, kemudian digambarkan pada *connectivity map*.

Data fMRI yang digunakan pada penelitian kali ini merupakan Starplus dataset. Dataset berisi sinyal BOLD yang merepresentasikan aktivitas otak. Dataset dihasilkan dari percobaan yang dilakukan pada 6 subyek. Masing-masing subyek diberikan stimulus yang sama, yaitu berupa *picture*

dan *sentence*. Stimulus *picture* dan *sentence* yang ditampilkan pada masing-masing subyek ditunjukkan pada Gambar 1. Antara stimulus *sentence* dan *picture* saling berkaitan satu sama lain. Stimulus *sentence* bertujuan untuk mendeskripsikan stimulus *picture*. Dari *picture* yang ditampilkan, apakah benar “tanda plus berada diatas bintang” atau “bintang berada diatas tanda plus” (Kartik V et al., 2009).

Pengambilan data starplus dilakukan dengan langkah-langkah sebagai berikut. Selama empat detik awal, diberikan stimulus pertama berupa *picture*. Empat detik kedua, stimulus pertama dihapus dan digantikan dengan *blank screen*. Empat detik ketiga, stimulus kedua berupa *sentence* ditampilkan. Stimulus *sentence* dapat ditampilkan selama 4 detik atau hingga subyek menekan tombol. Tujuan menekan tombol tersebut untuk menyatakan apakah stimulus kedua menjelaskan stimulus pertama dengan benar atau tidak. Pada akhir percobaan ditampilkan *blank screen*.

Hasil dari percobaan tersebut disimpan dalam matrik meta data yang terdiri dari 54 trial. Setiap trial terdiri dari $m \times n$ dimana m adalah jumlah snapshot dan n adalah jumlah voxel. Nilai dari matrik $m \times n$ adalah nilai BOLD hasil percobaan, yang merepresentasikan aktivitas otak, seperti yang terlihat pada Tabel 1.

Stimulus *picture* dan *sentence* ditampilkan secara berulang pada interval tertentu dalam 54 trial, dimana masing-masing trial terdiri atas 54 snapshot citra otak, seperti yang ditunjukkan pada Tabel 2. Snapshot yang terdapat pada setiap trial dihasilkan setiap 500 milidetik sepanjang percobaan. Setiap snapshot memiliki kurang lebih 5000 voxel yang berisi informasi nilai BOLD pada waktu ke- t milidetik. Lima ribu voxel tersebut merupakan gabungan voxel dari 25 hingga 30 Region Of Interest (ROI).

Pada trial 3-6, 8-11, 13-20, serta 22-25, subyek diberikan stimulus pertama berupa stimulus *picture* yang kemudian diikuti stimulus kedua berupa stimulus *sentence*. Sedangkan pada trial 30-33, 35-38, 40-47, serta 49-52, subyek diberikan stimulus pertama berupa stimulus *sentence* yang kemudian diikuti stimulus kedua berupa stimulus *picture*.

2.2 NON-OVERLAPPING CROSS CORRELATION

Pengujian dilakukan dengan dua skenario. Pertama pengujian pada masing-masing subyek dan kedua pengujian pada penggabungan keenam subyek. Pembagian pengujian tersebut untuk melihat apakah hasil yang diperoleh sama antara pengujian yang dilakukan pada masing-masing subyek dengan pengujian dengan cara penggabungan keenam subyek, karena masing-masing subyek memiliki jumlah voxel yang berbeda. Pengujian pada masing-masing subyek dilakukan dengan mencari fitur pada masing-masing subyek. Metode yang digunakan untuk memilih fitur adalah *cross correlation*. Fitur yang terpilih merupakan voxel yang paling signifikan dalam merepresentasikan aktivitas otak saat melihat *picture* dan saat melihat *sentence*. Voxel yang digunakan dalam pengujian hanya yang menggunakan snapshot 1 hingga snapshot 8.

$$V_i = \begin{bmatrix} v_{11} & v_{12} & v_{13} & \dots & v_{1m} \\ v_{21} & v_{22} & v_{23} & \dots & v_{2m} \\ v_{31} & v_{32} & v_{33} & \dots & v_{3m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ v_{n1} & v_{n2} & v_{n3} & \dots & v_{nm} \end{bmatrix} \dots\dots\dots (1)$$

$$\mu_v = [\mu_1, \mu_2, \mu_3, \dots, \mu_m] \dots\dots\dots (2)$$

$$\mu_t = \begin{bmatrix} \mu_{v11} & \mu_{v12} & \mu_{v13} & \dots & \mu_{v1m} \\ \mu_{v21} & \mu_{v22} & \mu_{v23} & \dots & \mu_{v2m} \\ \mu_{v31} & \mu_{v32} & \mu_{v33} & \dots & \mu_{v3m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \mu_{vt1} & \mu_{vt2} & \mu_{vt3} & \dots & \mu_{vtm} \end{bmatrix} \dots\dots\dots (3)$$

$$k = \begin{bmatrix} k_{11} & k_{12} & k_{13} & \dots & k_{1m} \\ k_{21} & k_{22} & k_{23} & \dots & k_{2m} \\ k_{31} & k_{32} & k_{33} & \dots & k_{3m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ k_{m1} & k_{m2} & k_{m3} & \dots & k_{mm} \end{bmatrix} \dots\dots\dots (4)$$

$$\mu_{kv} = [\mu_{k1}, \mu_{k2}, \mu_{k3}, \dots, \mu_{km}] \dots\dots\dots (5)$$

$$Corr = r_{xy} = \frac{S_{xy}}{S_x \cdot S_y} \dots\dots\dots (6)$$

$$Cov = S_{xy} = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{n - 1} \dots\dots\dots (7)$$

$$S_x = \sqrt{\frac{\sum (X_i - \bar{X})^2}{n-1}} \dots\dots\dots (8)$$

$$S_y = \sqrt{\frac{\sum (Y_i - \bar{Y})^2}{n-1}} \dots\dots\dots (9)$$

snapshot ke 3	BOLD	BOLD	BOLD	BOLD
snapshot ke 4	BOLD	BOLD	BOLD	BOLD
...	BOLD	BOLD	BOLD	BOLD
snapshot ke m	BOLD	BOLD	BOLD	BOLD

Dimana X adalah nilai rata-rata voxel ke *i* dan Y adalah voxel ke *i+1* dari *snapshot* 1 hingga *snapshot* 8. Pada masing-masing *trial* ke *i* *snapshot* 1 hingga 8 dihitung nilai mean μ_v per voxel, sehingga didapatkan hasil μ_v berukuran 1 x *m*, dan untuk masing-masing *trial* pada stimulus *picture* dan *sentence* akan memiliki matriks μ_v yang digabung menjadi matriks μ_t dengan ukuran *t* x *m*, dimana *t* merupakan 20 *trial* dari stimulus *picture* dan 20 *trial* dari stimulus *sentence*.

Langkah selanjutnya adalah mencari korelasi masing-masing voxel terhadap voxel yang lain berdasarkan μ_t . Hasil dari korelasi tersebut berupa matriks *k* berukuran *m* x *m*.

+

(a) *Picture stimulus*

The Plus is above the Star

(b) *Stimulus sentence yang benar*

The Star is above the Plus

(c) *Stimulus sentence yang salah*

Gambar 1. *Stimulus picture dan sentence*

Tabel 1. Snapshot Starplus dataset

	voxel ke 1	voxel ke 2	...	voxel ke n
snapshot ke 1	BOLD	BOLD	BOLD	BOLD
snapshot ke 2	BOLD	BOLD	BOLD	BOLD

Tabel 2. Starplus dataset

Trial	Snapshot Trial 3
Trial 1 : Rest	Snapshot 1 : Picture
Trial 2 : Rest	Snapshot 2 : Picture
Trial 3 : Picture	Snapshot 3 : Picture
Trial 4 : Picture	Snapshot 4 : Picture
Trial 5 : Picture	Snapshot 5 : Picture
Trial 6 : Picture	Snapshot 6 : Picture
Trial 7 : Rest	Snapshot 7 : Picture
Trial 8 : Picture	Snapshot 8 : Picture
Trial 9 : Picture	Snapshot 9 : Rest
Trial 10 : Picture	Snapshot 10 : Rest
Trial 11 : Picture	Snapshot 11 : Rest
Trial 12 : Rest	Snapshot 12 : Rest
Trial 13 : Picture	Snapshot 13 : Rest
Trial 14 : Picture	Snapshot 14 : Rest
Trial 15 : Picture	Snapshot 15 : Rest
Trial 16 : Picture	Snapshot 16 : Rest
Trial 17 : Picture	Snapshot 17 : Sentence
Trial 18 : Picture	Snapshot 18 : Sentence
Trial 19 : Picture	Snapshot 19 : Sentence
Trial 20 : Picture	Snapshot 20 : Sentence
Trial 21 : Rest	Snapshot 21 : Sentence
Trial 22 : Picture	Snapshot 22 : Sentence
Trial 23 : Picture	Snapshot 23 : Sentence
Trial 24 : Picture	Snapshot 24 : Sentence
Trial 25 : Picture	Snapshot 25 : Rest
Trial 26 : Rest	Snapshot 26 : Rest
Trial 27 : Rest	Snapshot 27 : Rest
Trial 28 : Rest	Snapshot 28 : Rest
Trial 29 : Rest	Snapshot 29 : Rest
Trial 30 : Sentence	Snapshot 30 : Rest
Trial 31 : Sentence	Snapshot 31 : Rest
Trial 32 : Sentence	Snapshot 32 : Rest
Trial 33 : Sentence	Snapshot 33 : Rest
Trial 34 : Rest	Snapshot 34 : Rest
Trial 35 : Sentence	Snapshot 35 : Rest
Trial 36 : Sentence	Snapshot 36 : Rest
Trial 37 : Sentence	Snapshot 37 : Rest
Trial 38 : Sentence	Snapshot 38 : Rest
Trial 39 : Rest	Snapshot 39 : Rest
Trial 40 : Sentence	Snapshot 40 : Rest
Trial 41 : Sentence	Snapshot 41 : Rest
Trial 42 : Sentence	Snapshot 42 : Rest
Trial 43 : Sentence	Snapshot 43 : Rest
Trial 44 : Sentence	Snapshot 44 : Rest
Trial 45 : Sentence	Snapshot 45 : Rest
Trial 46 : Sentence	Snapshot 46 : Rest
Trial 47 : Sentence	Snapshot 47 : Rest
Trial 48 : Rest	Snapshot 48 : Rest
Trial 49 : Sentence	Snapshot 49 : Rest
Trial 50 : Sentence	Snapshot 50 : Rest
Trial 51 : Sentence	Snapshot 51 : Rest
Trial 52 : Sentence	Snapshot 52 : Rest
Trial 53 : Rest	Snapshot 53 : Rest
Trial 54 : Rest	Snapshot 54 : Rest

Dari matriks k kemudian dicari nilai rata-rata korelasi per voxel μ_{kv} dan dicari μ_k yaitu nilai rata-rata dari semua μ_{kv} . Voxel dengan μ_{kv} yang kurang

dari μ_k akan dihapus dari matriks μ_i sejumlah n , sehingga nilai m akan menjadi $m = m - n$. Selanjutnya proses ini akan berlangsung terus menerus hingga didapat nilai *threshold* yang diinginkan. Bila *threshold* sama dengan 100, maka diperoleh 200 voxel hasil *cross correlation* dari stimulus *picture* (100) dan *sentence* (100). Dari 200 voxel tersebut dicari voxel yang *overlapping* antara voxel *picture* dan *sentence*. Jika terdapat voxel-voxel yang *overlapping*, maka voxel tersebut akan direduksi. Sedangkan voxel-voxel yang *non-overlapping* akan menjadi fitur yang signifikan sebagai masukan *classifier* untuk menguji kehandalan fitur tersebut.

Untuk skenario kedua, penggabungan keenam subjek dilakukan dengan cara menggabungkan voxel *non-overlapping*, hasil dari masing-masing subyek. Voxel-voxel tersebut digabungkan dengan cara mencari voxel-voxel yang sering muncul pada keenam subyek. Voxel-voxel yang kemunculannya lebih besar atau sama dengan dua, akan menjadi fitur yang signifikan sebagai masukan *classifier*.

Classifier yang digunakan pada penelitian ini adalah *Support Vector Machine* (SVM). Data *training* yang digunakan sebagai masukan *classifier* sebanyak 80% dari data keseluruhan. Sedangkan data *testing* sebanyak 20% dari data keseluruhan. Data keseluruhan yang digunakan berjumlah 1920 buah untuk pengujian dengan cara penggabungan keenam subyek, yang berasal dari 8 snapshot x 20 trial x 2 task x 6 subyek. Sedangkan untuk pengujian pada masing-masing subyek, data keseluruhan yang digunakan berjumlah 320 buah data.

Untuk mengevaluasi hasil uji coba dari metode *non-overlapping cross correlation* tersebut digunakan teknik *precision*, *recall*, dan akurasi.

2.3 KOMPLEKSITAS NON-OVERLAPPING

Pada metode (Golestani et al., 2011), kompleksitas didefinisikan dengan $O(n)$, karena pengurangan jumlah fitur dilakukan dengan mencari rata-rata dari semua fitur, kemudian rata-rata korelasi dari fitur yang paling rendah akan diabaikan. Jika terdapat 5000 fitur dan nilai *threshold* yang menjadi batasan jumlah fitur adalah 100, maka terdapat 5000 rata-rata korelasi dengan 4900 iterasi. Sedangkan pada metode *non-overlapping cross correlation* yang diusulkan,

pengurangan jumlah fitur berdasarkan nilai rata-rata korelasi yang lebih kecil dari nilai rata-rata korelasi akan diabaikan. Dengan proses ini dapat mengurangi jumlah fitur dan waktu hingga separuh dari jumlah fitur pada tiap iterasi. Kompleksitas dari metode yang diusulkan didefinisikan dengan $O(\log(n))$.

3. HASIL DAN PEMBAHASAN

Hasil pengujian pada masing-masing subyek ditunjukkan pada Tabel 3 hingga Tabel 8. Tabel 9 merupakan hasil rata-rata dari masing-masing subyek. Sedangkan pengujian dengan penggabungan keenam subyek ditunjukkan pada Tabel 10. Pada Tabel 3 hingga Tabel 8 merupakan hasil pengujian metode dilakukan dengan cara membandingkan hasil uji coba antara metode yang tidak melakukan reduksi pada voxel yang *overlapping* antara *stimulus sentence* dan *stimulus picture*, dengan metode yang melakukan reduksi pada voxel yang *overlapping*. Seperti ditunjukkan di tabel dengan skenario pengujian masing-masing subyek, hasil pengujian dapat dianalisis bahwa metode dengan cara mereduksi voxel yang *overlapping* memiliki akurasi mendekati atau lebih baik dari metode dengan cara tanpa mereduksi voxel yang *overlapping*.

Metode dengan cara mereduksi voxel yang *overlapping* memiliki nilai akurasi rata-rata sekitar 95.1% dan waktu komputasi 0.019 detik, sedangkan metode dengan cara tanpa mereduksi voxel yang *overlapping* memiliki akurasi sekitar 94.6% dan waktu akurasi 0.023 detik. Metode yang diusulkan lebih unggul karena dapat mengurangi jumlah voxel yang digunakan sebagai fitur untuk mencapai nilai akurasi yang sama pada metode tanpa mereduksi voxel yang *overlapping*, selain itu juga dapat mengurangi waktu komputasi. Namun, pada skenario penggabungan keenam subyek, diperoleh hasil bahwa metode dengan cara mereduksi voxel yang *overlapping* memiliki nilai akurasi rata-rata kurang baik dari metode dengan cara tanpa mereduksi voxel yang *overlapping*. Metode dengan cara mereduksi voxel yang *overlapping* memiliki nilai akurasi rata-rata sekitar 93.7%, sedangkan metode dengan cara tanpa mereduksi voxel yang *overlapping* memiliki nilai akurasi rata-rata sekitar 97.4%.

Nilai akurasi rata-rata pada metode dengan cara mereduksi voxel yang *overlapping* mengalami penurunan di skenario penggabungan keenam subyek karena lebih banyak voxel yang direduksi. Beberapa informasi voxel dari masing-masing subyek ada yang tidak dijadikan fitur, sehingga ada informasi yang hilang. Selain itu disebabkan juga oleh perbedaan ukuran anatomi daerah otak pada masing-masing subyek (Golestani et al., 2011). Sehingga letak voxel-voxel dengan fungsi sama dapat saja bergeser antara satu subyek dengan subyek yang lain. Hal ini mengakibatkan voxel yang terpilih menjadi fitur pada skenario penggabungan keenam subyek dengan metode mereduksi voxel yang *overlapping* tidak dapat mewakili informasi dengan baik. Sedangkan pada metode tanpa mereduksi voxel yang *overlapping* mengalami kenaikan di skenario penggabungan keenam subyek karena pada metode ini tidak melakukan reduksi. Voxel-voxel yang dilibatkan menjadi fitur bertambah menjadi sangat banyak, dimana semua fitur dari masing-masing subyek dijadikan fitur. Sehingga dengan fitur yang sangat banyak tersebut dapat mewakili informasi lebih rinci. Pada analisis nilai *threshold* yang digunakan antara 100 hingga 1000, dapat dilihat bahwa semakin besar nilai *threshold* yang digunakan semakin tinggi nilai akurasi yang diperoleh. Hal ini menunjukkan bahwa rata-rata nilai akurasi meningkat sebanding dengan banyaknya jumlah fitur yang digunakan. Hasil uji coba tersebut sudah seharusnya diperoleh, karena semakin banyak fitur yang menjadi masukkan *classifier* berarti semakin detail dalam membantu mengenali suatu objek atau *stimulus*. Dari semua tabel diatas dapat dianalisis pula bahwa nilai *threshold* rata-rata yang dapat digunakan adalah 400, karena pada nilai *threshold* tersebut rata-rata terjadi penurunan nilai akurasi pertama setelah terjadi kenaikan nilai akurasi yang *continue*. Nilai *threshold* tersebut berarti menunjukkan nilai *threshold* yang optimal.

Metode dengan cara mereduksi voxel yang *overlapping* yang dikembangkan dengan memanfaatkan metode *inter voxel cross correlation* memiliki beberapa kelebihan. Pertama metode yang diusulkan dapat mengurangi jumlah fitur, dimana untuk mulai mencapai akurasi 94% cukup dengan

jumlah fitur 302, sedangkan pada metode tanpa mereduksi voxel yang *overlapping* dapat mencapai akurasi tersebut dengan jumlah fitur 920. Kedua, metode yang diusulkan cukup sederhana, tidak ada perhitungan matematis yang rumit, sehingga dapat mengurangi waktu komputasi, kompleksitas, dan mudah untuk dipahami.

Namun, metode yang diusulkan terdapat beberapa kekurangan. Pertama, metode tersebut belum mengklasifikasi secara optimal pada skenario penggabungan keenam subjek. Hal ini disebabkan karena belum dapat memetakan secara tepat voxel-voxel yang mewakili fungsi yang sama di subyek yang berbeda. Ada beberapa fitur di subyek pertama yang tidak dimiliki di subyek kelima dan keenam, atau dengan kata lain tidak terdapat irisan fitur untuk keenam subyek. Kedua, metode yang diusulkan belum dapat menentukan voxel aktif secara tepat, dengan *cross correlation*.

Tabel 3. Hasil Evaluasi Subyek-05710-v7.mat

Subyek-05710-v7.mat										
Fitur Overlapping + Non-Overlapping						Fitur Non-Overlapping				
Threshold	Waktu	Fitur	Precision	Recall	Akurasi	Waktu	Fitur	Precision	Recall	Akurasi
100	0.012	183	87.6%	93.4%	90.7%	0.012	166	89.9%	89.2%	89.5%
200	0.014	351	94.6%	91.0%	92.6%	0.013	302	96.1%	92.5%	94.2%
300	0.017	508	93.8%	94.5%	94.2%	0.015	416	92.2%	92.2%	92.2%
400	0.020	677	93.8%	94.5%	94.2%	0.017	554	94.6%	93.1%	93.8%
500	0.025	849	96.1%	93.9%	95.0%	0.019	698	95.3%	92.5%	93.8%
600	0.024	993	96.9%	95.4%	96.1%	0.021	786	95.3%	92.5%	93.8%
700	0.028	1143	97.7%	94.0%	95.7%	0.023	886	98.4%	92.0%	95.0%
800	0.030	1303	98.4%	95.5%	96.9%	0.025	1006	96.9%	94.7%	95.7%
900	0.042	1453	98.4%	96.2%	97.3%	0.026	1106	97.7%	95.5%	96.5%
1000	0.035	1601	97.7%	96.2%	96.9%	0.040	1202	96.1%	94.7%	95.3%

Tabel 4. Hasil Evaluasi Subyek-04799-v7.mat

Subyek-04799-v7.mat										
Fitur Overlapping + Non-Overlapping						Fitur Non-Overlapping				
Threshold	Waktu	Fitur	Precision	Recall	Akurasi	Waktu	Fitur	Precision	Recall	Akurasi
100	0.012	169	87.6%	91.1%	89.5%	0.012	138	80.6%	86.0%	83.7%
200	0.014	320	86.8%	85.5%	86.0%	0.013	240	91.5%	89.4%	90.3%
300	0.015	474	92.2%	90.8%	91.5%	0.014	348	91.5%	90.1%	90.7%
400	0.019	630	94.6%	95.3%	95.0%	0.015	460	93.0%	93.8%	93.4%
500	0.022	770	93.8%	95.3%	94.6%	0.018	540	93.0%	93.8%	93.4%
600	0.025	920	93.0%	95.2%	94.2%	0.019	640	95.3%	93.9%	94.6%
700	0.027	1066	94.6%	96.8%	95.7%	0.020	732	93.0%	94.5%	93.8%
800	0.029	1208	95.3%	96.9%	96.1%	0.021	816	95.3%	94.6%	95.0%
900	0.033	1352	96.1%	96.9%	96.5%	0.023	904	96.1%	93.9%	95.0%
1000	0.034	1495	97.7%	97.7%	97.7%	0.024	990	94.6%	94.6%	94.6%

Tabel 5. Hasil Evaluasi Subyek-04820-v7.mat

Subyek-04820-v7.mat										
Threshold	Fitur Overlapping + Non-Overlapping					Fitur Non-Overlapping				
	Waktu	Fitur	Precision	Recall	Akurasi	Waktu	Fitur	Precision	Recall	Akurasi
100	0.013	198	89.1%	89.1%	89.1%	0.013	196	89.1%	89.1%	89.1%
200	0.015	388	90.7%	89.3%	89.9%	0.015	376	90.7%	87.3%	88.8%
300	0.018	579	89.9%	95.9%	93.0%	0.018	558	88.4%	95.8%	92.2%
400	0.021	761	91.5%	95.9%	93.8%	0.024	722	92.2%	97.5%	95.0%
500	0.025	944	93.0%	96.0%	94.6%	0.024	888	92.2%	97.5%	95.0%
600	0.029	1117	92.2%	96.7%	94.6%	0.025	1034	92.2%	96.7%	94.6%
700	0.032	1299	92.2%	97.5%	95.0%	0.028	1198	93.0%	95.2%	94.2%
800	0.039	1461	92.2%	97.5%	95.0%	0.034	1322	90.7%	97.5%	94.2%
900	0.035	1626	92.2%	99.2%	95.7%	0.032	1452	90.7%	98.3%	94.6%
1000	0.039	1790	93.0%	99.2%	96.1%	0.035	1580	93.0%	97.6%	95.3%

Tabel 6. Hasil Evaluasi Subyek-04847-v7.mat

Subyek-04847-v7.mat										
Threshold	Fitur Overlapping + Non-Overlapping					Fitur Non-Overlapping				
	Waktu	Fitur	Precision	Recall	Akurasi	Waktu	Fitur	Precision	Recall	Akurasi
100	0.012	200	93.8%	97.6%	95.7%	0.016	200	93.8%	97.6%	95.7%
200	0.015	395	95.3%	97.6%	96.5%	0.015	390	94.6%	97.6%	96.1%
300	0.019	590	96.1%	96.1%	96.1%	0.019	580	96.1%	96.1%	96.1%
400	0.021	778	96.1%	97.6%	96.9%	0.020	756	96.9%	97.7%	97.3%
500	0.023	964	96.9%	97.7%	97.3%	0.024	928	96.1%	98.4%	97.3%
600	0.026	1136	97.7%	100.0%	98.8%	0.025	1072	99.2%	99.2%	99.2%
700	0.030	1308	98.4%	100.0%	99.2%	0.028	1216	98.4%	99.2%	98.8%
800	0.035	1476	98.4%	99.2%	98.8%	0.031	1352	99.2%	98.5%	98.8%
900	0.036	1640	99.2%	99.2%	99.2%	0.033	1480	99.2%	99.2%	99.2%
1000	0.039	1808	99.2%	99.2%	99.2%	0.035	1616	99.2%	98.5%	98.8%

Tabel 7. Hasil Evaluasi Subyek-05675-v7.mat

Subyek-05675-v7.mat										
Fitur Overlapping + Non-Overlapping						Fitur Non-Overlapping				
Threshold	Waktu	Fitur	Precision	Recall	Akurasi	Waktu	Fitur	Precision	Recall	Akurasi
100	0.015	186	93.0%	85.7%	88.8%	0.012	172	91.5%	85.5%	88.0%
200	0.015	361	91.5%	87.4%	89.1%	0.013	322	92.2%	86.2%	88.8%
300	0.017	530	94.6%	89.7%	91.9%	0.017	460	94.6%	91.0%	92.6%
400	0.020	704	93.0%	87.6%	89.9%	0.019	608	93.0%	88.2%	90.3%
500	0.022	877	93.0%	90.9%	91.9%	0.020	754	93.0%	88.2%	90.3%
600	0.025	1047	95.3%	91.8%	93.4%	0.028	894	93.8%	89.6%	91.5%
700	0.029	1211	95.3%	91.1%	93.0%	0.024	1022	95.3%	92.5%	93.8%
800	0.031	1371	96.9%	91.2%	93.8%	0.027	1142	94.6%	90.4%	92.2%
900	0.035	1526	96.9%	91.2%	93.8%	0.029	1252	96.1%	92.5%	94.2%
1000	0.039	1688	96.9%	91.2%	93.8%	0.031	1376	93.8%	93.8%	93.8%

Tabel 8. Hasil Evaluasi Subyek-05680-v7.mat

Subyek-05680-v7.mat										
Fitur Overlapping + Non-Overlapping						Fitur Non-Overlapping				
Threshold	Waktu	Fitur	Precision	Recall	Akurasi	Waktu	Fitur	Precision	Recall	Akurasi
100	0.013	193	90.7%	94.4%	92.6%	0.013	186	89.9%	92.8%	91.5%
200	0.014	385	93.8%	97.6%	95.7%	0.014	370	93.0%	97.6%	95.3%
300	0.018	561	92.2%	98.3%	95.3%	0.017	522	92.2%	97.5%	95.0%
400	0.020	739	90.7%	96.7%	93.8%	0.032	678	89.1%	96.6%	93.0%
500	0.023	916	93.0%	98.4%	95.7%	0.021	832	93.0%	97.6%	95.3%
600	0.026	1073	93.0%	98.4%	95.7%	0.026	946	93.0%	99.2%	96.1%
700	0.031	1229	93.8%	98.4%	96.1%	0.028	1058	93.8%	97.6%	95.7%
800	0.033	1381	95.3%	98.4%	96.9%	0.030	1162	93.8%	99.2%	96.5%
900	0.035	1545	93.0%	99.2%	96.1%	0.030	1290	93.0%	98.4%	95.7%
1000	0.038	1683	94.6%	98.4%	96.5%	0.033	1366	93.0%	99.2%	96.1%

Tabel 9. Rata-rata evaluasi skenario per subyek

Rata-rata Evaluasi Skenario Per Subyek										
Fitur Overlapping + Non-Overlapping						Fitur Non-Overlapping				
Subyek	Waktu	Fitur	Precision	Recall	Akurasi	Waktu	Fitur	Precision	Recall	Akurasi
1	0.014	351	94.6%	91.0%	92.6%	0.013	302	96.1%	92.5%	94.2%
2	0.025	920	93.0%	95.2%	94.2%	0.019	640	95.3%	93.9%	94.6%
3	0.025	944	93.0%	96.0%	94.6%	0.024	888	92.2%	97.5%	95.0%
4	0.026	1136	97.7%	100.0%	98.8%	0.025	1072	99.2%	99.2%	99.2%
5	0.017	530	94.6%	89.7%	91.9%	0.017	460	94.6%	91.0%	92.6%
6	0.018	561	92.2%	98.3%	95.3%	0.017	522	92.2%	97.5%	95.0%
Rata-rata	0.021	740	94.2%	95.1%	94.6%	0.019	647	95.0%	95.3%	95.1%

Tabel 10. Hasil evaluasi skenario gabungan subyek

Hasil Evaluasi Skenario Gabungan Subyek										
Fitur Overlapping + Non-Overlapping						Fitur Non-Overlapping				
Threshold	Waktu	Fitur	Precision	Recall	Akurasi	Waktu	Fitur	Precision	Recall	Akurasi
100	0.025	956	96.1%	94.7%	95.3%	0.013	49	78.3%	75.9%	76.7%
200	0.043	1693	97.7%	96.9%	97.3%	0.017	151	94.6%	92.4%	93.4%
300	0.049	2291	97.7%	96.9%	97.3%	0.014	317	91.5%	93.7%	92.6%
400	0.058	2804	98.4%	96.9%	97.7%	0.018	477	92.2%	91.5%	91.9%
500	0.067	3210	98.4%	96.9%	97.7%	0.020	687	96.1%	94.7%	95.3%
600	0.068	3528	99.2%	97.0%	98.1%	0.022	860	96.1%	94.7%	95.3%
700	0.092	3775	99.2%	96.2%	97.7%	0.028	1079	98.4%	95.5%	96.9%
800	0.082	3971	99.2%	95.5%	97.3%	0.036	1309	99.2%	96.2%	97.7%
900	0.096	4131	99.2%	96.2%	97.7%	0.039	1566	100.0%	96.3%	98.1%
1000	0.087	4244	99.2%	96.2%	97.7%	0.042	1785	100.0%	98.5%	99.2%
Rata-rata	0.067	3060	98.4%	96.4%	97.4%	0.025	828	94.7%	92.9%	93.7%

Hal ini dikarenakan metode ini hanya mencari voxel yang berkorelasi tinggi, namun belum dapat dipastikan apakah voxel yang berkorelasi tinggi merupakan voxel yang aktif. Ketiga metode yang diusulkan masih terdapat perulangan yang cukup banyak untuk data yang lebih besar.

4. KESIMPULAN

Dari berbagai macam pengujian pada metode yang melakukan reduksi pada voxel yang *overlapping* dengan mengembangkan metode *inter voxel cross correlation*, dapat ditarik beberapa kesimpulan. Pertama, pada skenario masing-masing subyek metode yang diusulkan lebih unggul dari metode yang tidak melakukan reduksi pada voxel yang *overlapping*. Kedua, skenario pengujian pada penggabungan keenam subyek dengan metode yang diusulkan memiliki nilai akurasi rata-rata yang kurang baik dibandingkan dengan pengujian pada masing-masing subyek.

Beberapa kelemahan telah disampaikan pada sub bab sebelumnya, terutama dalam pemilihan voxel yang aktif secara signifikan. Pendekatan menggunakan bantuan resting-state dalam mencari voxel yang aktif dan pemilihan fitur menggunakan non-overlapping cross correlation untuk menentukan voxel yang aktif berdasarkan rentang antara task-state dan resting-state dapat diteliti lebih lanjut.

5. UCAPAN TERIMA KASIH

Ucapan terima kasih kami sampaikan kepada Bapak Agus Zainal Arifin, Ibu Handayani Tjandrasa, Ibu Nanik Suciati, Ibu Chastine Fatichah, Ibu Isye Ariesanti, Ibu Anggraini Dwi Sensusiaty yang telah membimbing kami dalam pembuatan makalah ini.

6. DAFTAR PUSTAKA

- [1] Baxter PR, Victoria LM, et al., 2007. Assessing *functional connectivity* in the human brain by fMRI. *Journal of Magnetic Resonance Imaging* 25 1347–1357.
- [2] Fiecas, Mark, Hernando Ombao, Dan van Lunen, Richard Baumgartner, Alexandre Coimbra, and Dai Feng, 2012. Quantifying temporal correlations: A test-retest evaluation of

functional connectivity in resting-state fMRI." *NeuroImage*.

- [3] Golestani, Ali-Mohammad, and Bradley G. Goodyear, 2011. Regions of interest for resting-state fMRI analysis determined by inter-voxel cross-correlation. *NeuroImage* 56, no. 1 : 246-251.
- [4] Hayasaka, Satoru, and Paul J. Laurienti, 2010. Comparison of characteristics between region- and *voxel-based* network analyses in resting-state fMRI data. *Neuroimage* 50, no. 2: 499.
- [5] Kartik V, Sourav K, 2009. Analysis of FMRI Data. Indian Institute of Technology Kanpur : Intruduction to Cognitive Science.
- [6] <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-81>