

## PEMODELAN PERUBAHAN KEBUTUHAN PELANGGAN MENGGUNAKAN CASE TOOL

Nia Saurina SST., M.Kom

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

[niasaurina@yahoo.com](mailto:niasaurina@yahoo.com)

### Abstrak

Perubahan kebutuhan memberikan dampak yang signifikan terhadap peningkatan biaya dan waktu pada pengembangan perangkat lunak. Salah satu perubahan kebutuhan tersebut terjadi karena pelanggan sebagai *user* mempunyai karakteristik yang berbeda. Keterbatasan resource yang tersedia untuk merealisasikan perubahan kebutuhan menyebabkan perubahan kebutuhan perlu dikelola. Cara yang digunakan untuk mengelola perubahan kebutuhan adalah dengan menganalisa dependensi beserta bobot perubahan kebutuhan tersebut dan memberikan prioritas terhadap perubahan kebutuhan yang direalisasikan agar optimal sesuai dengan resource yang tersedia. Perubahan kebutuhan dari pelanggan dapat mengakibatkan kekacauan pada keterselesaian proyek. Labelled Transition System dapat menangkap alur dan perubahan transisi pada perubahan kebutuhan. Untuk memetakan perubahan kebutuhan serta menelusuri dampaknya maka dilakukan pemodelan perubahan kebutuhan. Penelitian ini mengajukan Labelled Transition System (LTS) untuk memodelkan perubahan kebutuhan. LTS merupakan model yang efektif untuk menggambarkan perubahan perilaku sistem dan dikembangkan untuk memodelkan perubahan kebutuhan perangkat lunak, sehingga mempermudah stakeholder untuk mengamati alur perubahan kebutuhan beserta dampaknya. Dengan model LTS tersebut dapat memberikan solusi dalam menyusun strategi realisasi perubahan kebutuhan. Selain itu dengan adanya case tool dapat menentukan prioritas kebutuhan sesuai dengan karakter pelanggan.

**Kata Kunci:** LTS, perubahan kebutuhan, pemodelan, dependensi dan bobot perubahan.

### Abstract

*Changing needs of a significant impact on the improvement of software development costs. Limitations of available resources to realize the need for change causes a change needs to be managed. Among the methods used to manage it is to analyze the dependencies along with weight changes and give priority to the need for change to be realized in order to optimally match the available resources. PT. Arcadia Global Media is a company engaged in the field of Field Services in the field of Media Placement Service (Indoor / Outdoor), Production of Signage, Booth & Maintenance and Design & Digital Printing. Changing needs of customers can lead to chaos on keterselesaian project. Labeled Transition System can capture the flow and changes in the transition to the changing needs. To map the changing needs and explore the impact it done modeling requirements change. This study propose a labeled transition System (LTS) to model the changing needs. LTS is an effective model to describe the change in behavior of the system and was developed to model the changing needs of the software, making it easier for stakeholders to observe the flow changes and their impact needs. With LTS models are to provide solutions to the changing needs of the realization strategy.*

**Keywords :** LTS , changing needs , modeling , dependencies and weight changes .

## 1. PENDAHULUAN

### 1.1 LATAR BELAKANG

Requirements management adalah sebuah proses untuk mendokumentasikan, menganalisa, melacak dan mengatur prioritas sebuah kebutuhan yang kemudian dikomunikasikan dengan stakeholder. Tujuan dari manajemen kebutuhan adalah untuk memastikan bahwa dokumen kebutuhan yang ada sudah sesuai dengan ekspektasi customer, internal dan external stakeholder. Manajemen kebutuhan dimulai dengan analisis dan elisitasi tujuan dan batasan sebuah organisasi. Selanjutnya dilanjutkan

dengan kegiatan mendukung perencanaan dari sebuah kebutuhan dan mengintegrasikannya dengan organisasi.

Manajemen kebutuhan meliputi komunikasi antara project team dengan stakeholder dan penyesuaian jika terjadi perubahan spesifikasi kebutuhan. Untuk mencegah terjadinya sebuah kebutuhan menimpa kebutuhan yang lain maka sebuah komunikasi yang terus menerus dan penggunaan tools untuk manajemen kebutuhan perlu dilakukan.

Salah satu alasan kenapa sebuah tim pengembang membutuhkan manajemen kebutuhan adalah karena menginginkan proyek yang mereka kerjakan berjalan dengan sukses. Bisa dikatakan bahwa menghindari kegagalan dalam suatu proyek adalah suatu motivasi untuk memajemen kebutuhan. Selain itu, manajemen kebutuhan dapat meningkatkan kemungkinan sebuah proyek menjadi sukses serta memunculkan kelebihan-kelebihan yang menjadi faktor dalam mendukung suksesnya suatu proyek.

Salah satu alasan kenapa sebuah tim pengembang membutuhkan manajemen kebutuhan adalah karena menginginkan proyek yang mereka kerjakan berjalan dengan sukses. Bisa dikatakan bahwa menghindari kegagalan dalam suatu proyek adalah suatu motivasi untuk memajemen kebutuhan. Selain itu, manajemen kebutuhan dapat meningkatkan kemungkinan sebuah proyek menjadi sukses serta memunculkan kelebihan-kelebihan yang menjadi faktor dalam mendukung suksesnya suatu proyek. Permasalahan dalam Merekayasa dan Memanajemen Kebutuhan Tentunya dalam mencapai sesuatu yang baik terkadang seseorang menemukan kendala-kendala baik yang memberikan pengaruh secara besar ataupun kecil. Sama seperti hal tersebut, terdapat masalah-masalah yang dihadapi dalam memajemen kebutuhan. Dalam survey yang dilakukan pada tahun 1996 terhadap pihak pengembang, manajer, dan personil penjamin kualitas ditemui empat masalah dasar dalam merekayasa kebutuhan [7], antara lain:

- Tidak bisa menelusuri perubahan kebutuhan (71%)
- Susah menuliskan kebutuhan (70%)
- Fitur-fitur kebutuhan di luar nalar (67%)
- Kebutuhan tidak terorganisasi dengan baik (54%).

PT. Arcadia Global Media merupakan perusahaan yang bergerak di bidang Bidang Layanan Jasa di bidang Media Placement Service (Indoor/Outdoor), Production of Signage, Booth & Maintenance serta Design & Digital Printing. Perubahan kebutuhan dari pelanggan dapat mengakibatkan kekacauan pada keterselesaian proyek. Labelled Transition System dapat menangkap alur dan perubahan transisi pada perubahan kebutuhan.

Untuk memetakan prioritas perubahan kebutuhan perangkat lunak serta menelusuri dampaknya dapat dilakukan dengan memodelkan perubahan kebutuhan tersebut. Penelitian ini mengajukan Labelled Transition System (LTS), metode untuk memodelkan perubahan kebutuhan pelanggan yang digunakan untuk membuat case tool, berfungsi untuk mempermudah stakeholder dalam mengamati alur perubahan kebutuhan beserta dampaknya. Dengan model LTS tersebut diharapkan dapat memberikan solusi dalam menyusun strategi realisasi perusahaan terhadap perubahan kebutuhan.

## 1.2 PERUMUSAN MASALAH

Perumusan masalah dalam penelitian ini membahas tentang merancang case tool sesuai dengan perubahan kebutuhan pelanggan di PT. Arcadia Global Media dengan menggunakan metode Labelled Transition System (LTS).

## 1.3 TUJUAN PENELITIAN

Tujuan dari penelitian ini adalah untuk mengetahui peta atau alur perubahan kebutuhan perangkat lunak beserta dampaknya yang dihasilkan melalui model Labelled Transition System (LTS).

## 1.4 MANFAAT PENELITIAN

Manfaat yang diperoleh dari hasil penelitian yaitu memberikan solusi dalam menyusun strategi realisasi perubahan kebutuhan.

## 2. TINJAUAN PUSTAKA

### 2.1 MANAJEMEN KEBUTUHAAN

Manajemen kebutuhan customer adalah salah satu masalah kunci dalam pengembangan sistem perangkat lunak dan produksinya. Untuk memulai pengembangan, spesifikasi kebutuhan harus disetujui terlebih dahulu oleh developer dan customer. Namun, sulit untuk membuat semua spesifikasi kebutuhan benar sejak awal, sehingga ketidak lengkapan sering terjadi. Sistem perangkat lunak dengan baseline set requirement juga mengalami permintaan berubah. Permintaan berubah meliputi semua informasi yang penting untuk melakukan modifikasi kebutuhan dalam rangka memenuhi fungsionalitas yang diharapkan. Manajemen perubahan adalah salah satu aspek paling penting untuk proyek pengembangan perangkat lunak yang sukses. Oleh karena itu, developer harus memiliki mekanisme yang efektif untuk mengelola proses perubahan. Developer juga harus mengetahui dampak yang mengikuti perubahan kebutuhan tersebut. Telah menjadi hipotesis bahwa, biaya perbaikan semakin meningkat seiring dengan proses perkembangan, perubahan kebutuhan membuat siklus pengembangan menjadi lebih sulit dan membutuhkan biaya untuk implementasinya.

### 2.2 PERKIRAAN RESIKO

Organisasi pengembang perangkat lunak banyak yang mengalami permasalahan yang kronis, mulai dari proyek yang tertunda hingga over budget. Permasalahan tersebut tidak hanya ditemui selama pengembangan baru saja, namun juga selama perubahan dan perbaikan kebutuhan dari proyek yang ada. Kesulitan yang dialami praktisioner perangkat lunak adalah menghindari adanya perubahan kebutuhan perangkat lunak serta meminimumkan resiko yang diakibatkannya.

Boehm mendefinisikan resiko pengembangan perangkat lunak sebagai kemungkinan output yang tidak memuaskan. "output yang tidak memuaskan" relatif berbeda menurut perspektif masing-masing stakeholder. Untuk customer dan developer, output yang tidak memuaskan tersebut adalah over budget dan jadwal yang tertunda; untuk user adalah produk dengan fungsionalitas yang tidak sesuai, atau antarmuka, performance dan reabilitas yang tidak sesuai; sedangkan untuk maintainer adalah kualitas perangkat lunak yang rendah. Praktisioner memerlukan pendekatan standart untuk menangani resiko.

Salah satu aspek penting untuk penanganan resiko adalah dengan mengidentifikasi kategori resiko yang dapat diterapkan untuk tipe tertentu dari proyek pengembangan perangkat lunak yang ada. Beberapa kategori resiko yang ada dalam literatur diantaranya adalah resiko ukuran produk dan kompleksitas, resiko jadwal dan waktu, resiko fungsionalitas sistem, resiko manajemen kebutuhan, dan resiko jaminan kualitas sistem [1]. Oleh karena ada berbagai tipe resiko yang mungkin terjadi serta perlu ditangani, praktisioner harus memutuskan prioritas resiko yang paling penting untuk masing-masing proyek. Proses ini disebut prioritasasi resiko. Prioritasasi resiko adalah proses mengidentifikasi, menganalisa, dan mengatur resiko. Setelah mengembangkan perangkat lunak atau membuat perubahan untuk kebutuhan perangkat lunak yang ada, developer perlu mengetahui dampaknya terhadap perubahan kebutuhan lain yang mengikuti.

Terdapat beberapa pendekatan untuk membantu developer memperkirakan resiko dalam manajemen perubahan. Pendekatan pertama, analisa dampak, adalah aktivitas mengidentifikasi apa langkah modifikasi yang perlu dilakukan dalam mencapai target perubahan, atau mengidentifikasi potensi konsekuensi dari perubahan. Beberapa aspek dari analisa dampak mengharuskan developer memiliki wawasan terhadap sejumlah usaha terkait perubahan yang dibuat. Analisa dampak meliputi proses mengidentifikasi program yang mengacu pada variabel atau prosedur yang mengalami perubahan. Hal ini juga meliputi mengidentifikasi obyek dan relasi diantara obyek untuk menentukan obyek yang terpengaruh oleh perubahan. Analisa dampak juga membantu developer menjawab pertanyaan apa yang dibutuhkan untuk melakukan perubahan sekaligus meminimumkan efek samping yang tidak diharapkan. Meskipun minimalisasi ini tidak selalu memungkinkan, analisa dampak didesain untuk membantu developer mengontrol berbagai efek terkait perubahan [4].

Pendekatan kedua, untuk mengelola perubahan melalui klasifikasi perubahan. Mengklasifikasi perubahan kebutuhan adalah salah satu cara untuk

membantu praktisioner memahami kealamiah perubahan. Melalui pengklasifikasian perubahan permintaan secara akurat, developer dapat memanfaatkan secara umum pendekatan heuristik yang ada untuk menangani variasi tipe perubahan dan memprediksi dampak perubahan pada sistem.

Metode ketiga, analisa kausal, telah diterapkan dalam mengelola perubahan untuk mengidentifikasi permasalahan implementasi dalam proses perubahan kebutuhan, dan untuk mengidentifikasi penyebab dari permasalahan ini. Dengan menyusun analisa kausal, sebuah organisasi lebih mampu untuk memahami mengapa perubahan terjadi dalam perangkat lunak yang mereka kembangkan. Setelah memahami penyebab perubahan diharapkan organisasi menjadi lebih baik dalam mengelola perubahan kebutuhan dalam proyek selanjutnya.

### 2.3 LABELLED TRANSITION SYSTEM (LTS)

Labelled Transition System (LTS) adalah model dasar untuk merepresentasikan reaktifitas, kongkurensi dan komunikasi sistem dan sering digunakan dalam ilmu komputer sebagai model untuk menggambarkan behavior dari sistem [6]. Definisi sebuah Labelled Transition System adalah sebuah tuple  $(S, Act, \rightarrow, I, AP, L)$ , di mana:

$S$  adalah himpunan dari state.

$Act$  adalah himpunan dari action.

$S \times Act \times S$  yang merupakan sebuah relasi transisi.

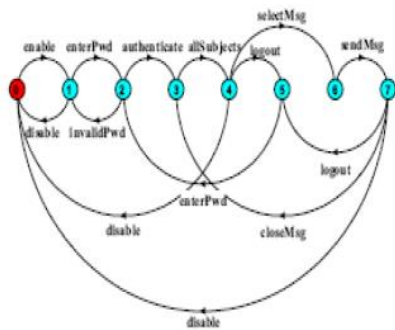
$I$  adalah himpunan bagian  $S$  ( $I \subseteq S$ ) dan merupakan himpunan dari state inisial.

$AP$  adalah himpunan dari atomic proposition.

$L : S \rightarrow 2^{AP}$  adalah sebuah fungsi labeling.

Labelled Transition System (LTS) digunakan untuk memodelkan perilaku sistem yang dibuat serta menggambarkan komunikasi antar komponen dalam sistem tersebut. LTS adalah tahapan transisi sistem dimana setiap transisi tersebut diberi label untuk mempermudah pengamatan aliran perubahan sistem. Label yang ada menginterpretasikan komunikasi yang bersinggungan antar komponen [5].

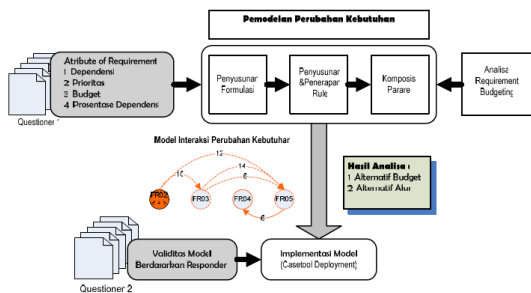
Labelled Transition System (LTS) adalah salah satu model yang menyediakan animasi grafis yang efektif berdasarkan aturan yang diberikan untuk model perilaku sistem [2]. Model perilaku sistem adalah diskripsi tahapan kejadian selama system dijalankan. Model tersebut juga dapat mendeskripsikan bagaimana komponen sistem, lingkungan, dan juga user berinteraksi dalam rangka menjalankan fungsional sistem [3]. Gambar 2.1 menunjukkan contoh LTS untuk skenario user membuka email.



Gambar 1 Contoh LTS [5]

### 3. METODE PENELITIAN

Sistematika metoda penelitian secara global ditunjukkan pada Gambar 2. Diawali dengan melakukan studi literatur dan analisa dasar teori hingga implementasi model. Studi literatur dilakukan untuk mendapatkan informasi dari literatur yang digunakan, mengetahui perkembangan penelitian yang telah dilakukan sebelumnya, serta metode yang ada, sedangkan analisa dasar teori dilakukan untuk menguji model dengan menggunakan turunan dari teori serta rumus yang dapat dibuktikan kebenarannya secara ilmiah.



Gambar 2. Sistematika Metode Penelitian

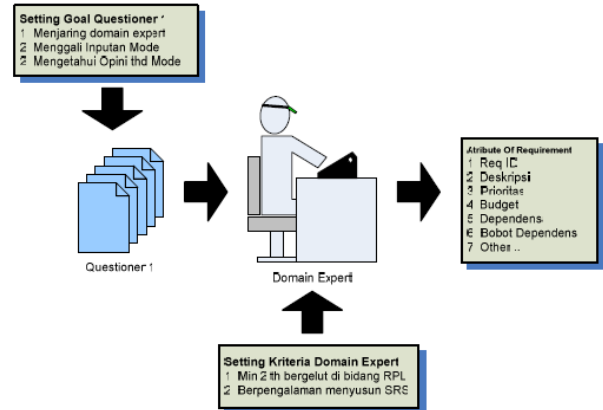
#### 3.1 IDENTIFIKASI KEBUTUHAN

Identifikasi kebutuhan dilakukan dengan tujuan menggali informasi dari responden. Adapun informasi yang digali adalah daftar spesifikasi kebutuhan perangkat lunak (Software Requirement Specification) beserta variabel yang perlu untuk dieksplorasi dalam dokumen SRS tersebut serta opini responden terhadap model perubahan kebutuhan perangkat lunak sebagai langkah tindak lanjut untuk mempermudah stakeholder dalam merealisasikan perubahan spesifikasi kebutuhan perangkat lunak agar optimal sesuai dengan resource (budget) yang tersedia.

Responden dalam penelitian ini adalah para ahli (domain expert) dalam bidang rekayasa perangkat lunak (software engineering). Kriteria domain expert dalam penelitian ini adalah mereka yang minimal telah dua tahun bergelut di bidang rekayasa perangkat lunak serta berpengalaman dalam membuat dokumen Software Requirement Specification (SRS).

Penggalan data dalam penelitian ini dilakukan dengan menggunakan teknik wawancara dan

pengisian questioner oleh domain expert. Adapun data yang diperoleh digunakan sebagai inputan dari model perubahan kebutuhan yang dibuat. Data yang diperlukan sebagai inputan diantaranya adalah: requirement ID, deskripsi, prioritas, budget, analisa dependensi, besar bobot dependensi antar requirement, serta variabel tambahan dari responden bila memungkinkan. Adapun lebih jelasnya dapat dilihat pada Gambar 3.



Gambar 3 Mekanisme Penggalan Data Awal

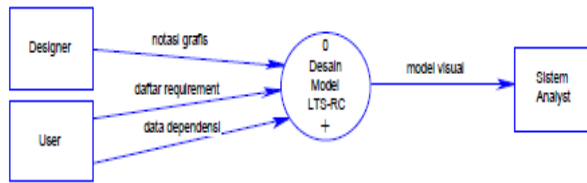
Responden yang berpartisipasi dalam penelitian ini diseleksi dari kalangan Software Engineer yang kemudian diklaim sebagai domain expert dalam penelitian ini karena telah memenuhi kriteria yang diberikan. Tabel 1 menunjukkan data responden yang terlibat dalam penelitian.

Tabel 1 Data Responden

No	Profesi bidang	Jumlah
Software Engineerin		
1	Project Officer	1
2	Estimator	4
3	Design	3
4	Workshop	2
Total Responden		10 orang

#### 3.2 DFD dari LTS-RC

Data Flow Diagram (DFD) dari model LTS-RC yang berbasis notasi grafis merupakan penggambaran secara grafis dari sistem yang terdapat dalam model LTS-RC dengan menggunakan sejumlah simbol untuk menggambarkan bagaimana aliran data terhubung antar proses di dalam sistem tersebut. Adapun DFD tersebut dimulai dari DFD level 0 untuk kemudian dijabarkan dalam DFD level 1 yang merupakan sub proses dari proses sebelumnya. DFD level 0 menjelaskan secara garis besar interaksi antar proses yang terdapat dalam suatu sistem dengan entitas-entitas yang berhubungan dengan proses tersebut.



Gambar 4 DFD Level 0

Gambar 4 menjelaskan hubungan antara proses membuat desain model dengan beberapa entitas, yaitu: entitas designer, entitas user, dan entitas sistem analyst. Entitas designer dan entitas user berhubungan dengan proses input data untuk membuat desain model LTS-RC, sedangkan entitas sistem analyst berhubungan dengan analisa output yang dihasilkan oleh model LTS-RC berupa model interaksi perubahan kebutuhan yang merupakan visualisasi dari model yang dibuat.

DFD level 1 merupakan subproses dari proses membuat desain model pada DFD level 0. Adapun DFD level 1 dari sistem dapat dilihat pada Gambar 4.



Gambar 5. DFD level 1 Proses Membuat Desain Model LTS-RC

Gambar 5 merupakan dekomposisi dari proses membuat desain model LTS-RC yang ada pada DFD level 0. Dari gambar tersebut dapat diuraikan penjelasan bahwa jika daftar kebutuhan (list of kebutuhan) beserta atributnya, termasuk data dependensi kebutuhan telah diinputkan oleh user maka dimodelkan ke dalam model jaringan (network model) yang untuk kemudian dinamakan Label Transition System for Kebutuhan Change (LTS-RC) yang berbasis pada notasi grafis.

Desain model dapat dilakukan dengan menggunakan notasi yang disediakan pada template notasi grafis (default notation). Dari model LTS-RC

user dapat mengetahui visualisasi dari interaksi diantara kebutuhan sesuai dengan data yang diinputkan.

Dalam proses analisa budget diawali oleh user yang menginputkan data budget untuk setiap kebutuhan maupun data budget setiap perubahan kebutuhan perangkat lunak yang direalisasikan. Kemudian dari database budget yang dibutuhkan maka dianalisa besar budget yang dibutuhkan dengan menggunakan prinsip kombinasi linear variabel. Hasil analisa budget berupa list dari alternatif budget untuk merealisasikan perubahan kebutuhan setelah user menentukan kebutuhan pertama yang dipilih untuk dirubah.

Pada model LTS-RC terdapat proses untuk mendesain detil data model yang berfungsi untuk mempermudah user dalam menganalisa perubahan kebutuhan yang direalisasikan. Sedangkan validasi data model dilakukan berdasarkan rule yang ada pada model jaringan.

### 3.3 Rule LTS-RC

Rule dalam LTS-RC adalah rules yang digunakan dalam proses transformasi dari data list kebutuhan hingga terbentuk model jaringan yang dinamakan LTS-RC, juga digunakan untuk proses validasi data model yang dibuat oleh perancang, sehingga data model tersebut dapat memberikan makna visualisasi grafis yang mudah dipahami oleh user. Berikut ini adalah rule LTS RC yang digunakan dalam transformasi data tersebut:

Template notasi grafis yang tersedia adalah:

Obyek notasi grafis: node must, node optional, symbol minimum, symbol maximum, symbol optimum, symbol total, initial state

Relasi dari objek: edge must, edge optional

Notasi node must adalah node yang melambangkan kebutuhan dari sistem software yang memiliki prioritas harus (must). Node ini berisi reqID (identitas kebutuhan). Setiap kebutuhan ini memiliki atribut berupa deskripsi, prioritas, budget, dependensi, dan bobot dependensi. Atribut dapat diakses dengan klik kiri pada node yang bersangkutan [11].

Notasi node optional adalah node yang melambangkan kebutuhan dari sistem software yang memiliki prioritas pilihan (optional). Node ini berisi reqID (identitas kebutuhan). Setiap kebutuhan ini memiliki atribut berupa deskripsi, prioritas, budget, dependensi, dan bobot dependensi. Atribut dapat diakses dengan klik kiri pada node yang bersangkutan.

Simbol minimum melambangkan analisa budget minimum yang diperlukan untuk merealisasikan perubahan kebutuhan.



Simbol maximum melambangkan analisa budget maximum yang diperlukan untuk merealisasikan perubahan kebutuhan.

Simbol optimum melambangkan analisa budget optimum yang diperlukan untuk merealisasikan perubahan kebutuhan, yang didapat dengan mencari selisih terkecil budget per bahan kebutuhan terhadap budget yang tersedia [10].

Simbol total melambangkan analisa budget total yang diperlukan untuk merealisasikan perubahan kebutuhan, yang resume dari seluruh alternative budget untuk kebutuhan yang berada pada alur perubahan kebutuhan yang terbentuk.

Obyek-obyek notasi grafis (berupa node) dapat dihubungkan dengan menggunakan notasi relasi (berupa edge) dengan ketentuan sebagai berikut:

Antara notasi node must dengan node must hanya dapat dihubungkan oleh edge must dengan arah sesuai dengan hubungan implikasi dalam dependensi antar kebutuhan. Seperti ilustrasi pada Tabel 2 nomor 1.

Antara node must dengan node optional hanya dapat dihubungkan oleh edge must jika arah implikasi dari node optional ke node must, sebaliknya jika arah implikasi dari node must ke node optional hanya dapat dihubungkan dengan edge optional. Seperti ilustrasi pada Tabel 2 nomor 2.

Antara notasi node optional dengan node optional hanya dapat dihubungkan oleh edge optional dengan arah sesuai dengan hubungan implikasi dalam dependensi antar kebutuhan. Seperti ilustrasi pada Tabel 2 nomor 3.

Adanya angka pada edge menunjukkan bobot dependensi perubahan kebutuhan. Seperti ilustrasi pada Tabel 2 nomor 4.

Relasi kombinasi merupakan gabungan dari dua atau lebih relasi implikasi, dimana satu kebutuhan yang mempengaruhi dua atau lebih perubahan pada kebutuhan yang lain. Seperti ilustrasi yang terdapat pada Tabel 2 no 5.

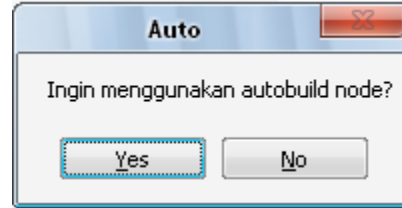
Tabel 2 Relasi Implikasi

No	Relasi Implikasi	Makna Relasi
1		Jika FR01 berubah maka akan menyebabkan FR03 berubah
2		Jika FR01 berubah maka akan menyebabkan FR03 berubah (optional)
3		Jika FR01 berubah maka akan menyebabkan FR03 berubah (optional)
4		Relasi terboboti
5		Jika FR01 berubah maka akan menyebabkan FR03 dan FR04 berubah

#### 4. IMPLEMENTASI CASE TOOL PERUBAHAN KEBUTUHAN PELANGGAN

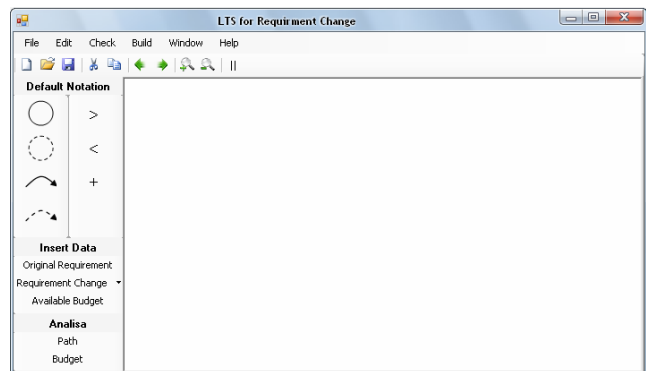
Antarmuka merupakan sarana untuk komunikasi antara user dengan komputer. Antarmuka dibuat sesederhana mungkin sehingga memudahkan user dalam mengaplikasikannya. Antarmuka ini

berfungsi untuk memudahkan user sebagai system analyst untuk memodelkan perubahan kebutuhan perangkat lunak.



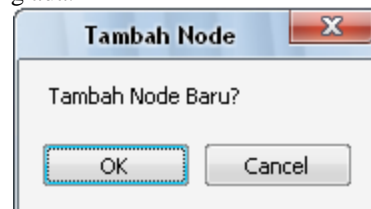
Gambar 6 Kotak Dialog Auto

Sebagai awalan dari implementasi case tool LTS-RC maka user mengisi kotak dialog yang berisi pilihan apakah user ingin menggunakan aotubuilt node untuk memodelkan perubahan kebutuhan atau tidak. Kotak dialog ini dapat dilihat pada Gambar 6. Jika user memilih tidak maka secara otomatis keluar dari sistem, sedangkan jika user memilih ya maka muncul antarmuka untuk memodelkan perubahan kebutuhan perangkat lunak seperti yang tertera pada Gambar 6.



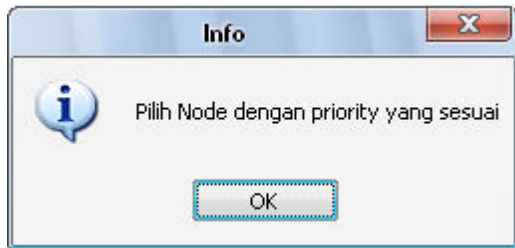
Gambar 7 Screenshot LTS-RC

Pada Gambar 7 terdapat beberapa notasi yang dapat digunakan oleh user untuk memodelkan perubahan kebutuhan perangkat lunak secara manual sesuai data yang dimiliki. Dengan menggunakan notasi grafis yang tersedia maka user secara otomatis memodelkan kebutuhan yang dimiliki sesuai dengan data yang ada.



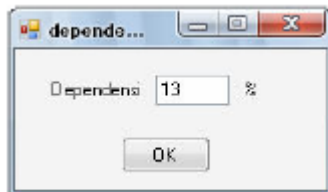
Gambar 8 Kotak Dialog Tambah Node

Jika user ingin menambahkan node atau kebutuhan baru maka user bisa langsung meng-klik pallet node untuk kemudian jika ada komentar untuk menambahkan node baru seperti pada Gambar 8 maka pilih OK dan node dengan ID baru secara otomatis tampil pada layar, dan demikian seterusnya.



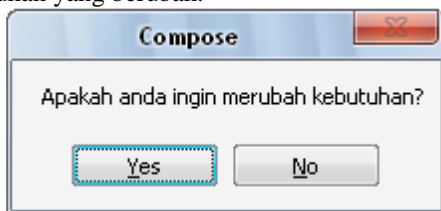
Gambar 9 Kotak Dialog Info

Setelah node terbentuk maka user membubuhkan edge pada setiap node sesuai dengan data dependensi yang ada. Jika edge yang dipilih tidak sesuai dengan prioritasnya maka muncul warning sistem seperti tertera pada Gambar 9.



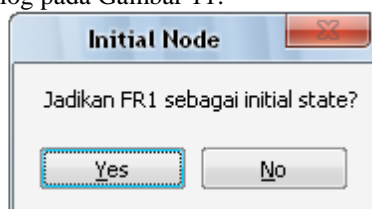
Gambar 10 Form Dependensi

Setelah user menambahkan edge dengan priority yang sesuai pada setiap node maka secara otomatis muncul kotak dialog seperti tertera pada Gambar 10. Kotak dialog tersebut digunakan untuk memberikan besarnya bobot dependensi untuk perubahan kebutuhan yang direalisasikan. Nilai yang diinputkan digunakan untuk perhitungan besarnya budget yang dibutuhkan bila salah satu node sebagai kebutuhan yang berubah.



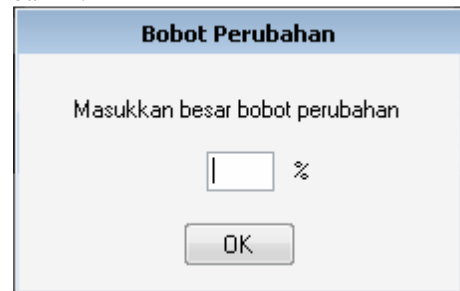
Gambar 11 Kotak Dialog Compose

Setelah seluruh node beserta bobot dependensi terbentuk maka untuk mengetahui peta perubahan kebutuhan yang direalisasikan beserta besarnya budget yang dibutuhkan maka langkah selanjutnya adalah memilih menu compose ( || ) maka muncul kotak dialog pada Gambar 11.



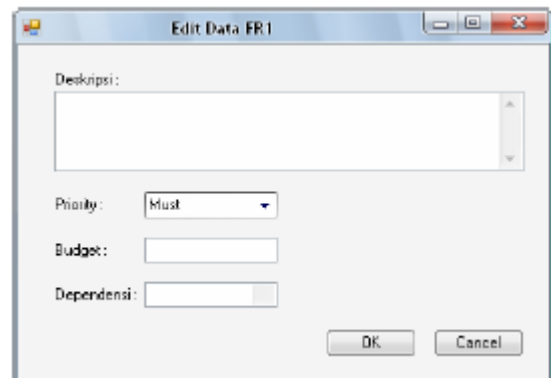
Gambar 12 Kotak Dialog Initial Node

Ketika user memilih satu initial state maka selanjutnya user mengisikan besar bobot perubahan yang diinginkan (%). Hal ini mempengaruhi besar perubahan untuk kebutuhan yang masih berhubungan. Form bobot perubahan tertera pada Gambar 12.



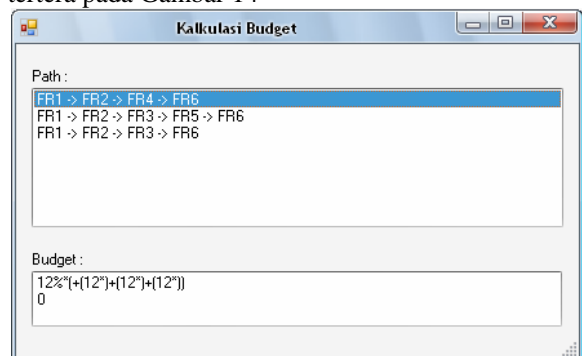
Gambar 13 Form Bobot Perubahan

Untuk melakukan editing data sebelum compose maka klik kanan node kemudian pilih menu edit dan user dapat merubah segala informasi yang ada pada form editing data. Seperti pada Gambar 13.



Gambar 14 Form Edit Data FR

Jika stakeholder ingin mengetahui alternatif path atau budget maka dari notasi yang disediakan pada model LTS-RC maka user mendapatkan informasi tersebut pada form kalkulasi budget seperti yang tertera pada Gambar 14



Gambar 15 Form Kalkulasi Budget

Untuk mengetahui besar budget optimum maka user terlebih dahulu mengisi form ketersediaan budget seperti yang terdapat pada Gambar 15.

Gambar 16 Form Available Budget

Selanjutnya informasi yang diinputkan oleh user terekam dalam form kebutuhan seperti pada Gambar 16.

Nama	Deskripsi	Priority	Budget	Dependensi
FR1	Sistem bisa memvalidasi pin	Must	2	
FR2	Sistem bisa melakukan pengecekan s...	Must	1	
FR3	Sistem bisa melakukan pengecekan u...	Must	1	
FR4	Sistem bisa melakukan pembayaran	Must	3	
FR5	Sistem bisa melakukan transfer	Must	3	

Gambar 17 Form Kebutuhan

## 5. KESIMPULAN DAN SARAN

### 5.1 KESIMPULAN

LTS dapat digunakan untuk memetakan perubahan kebutuhan perangkat lunak dengan menggunakan data dependensi antar kebutuhan. Dengan model LTS maka stakeholder dapat mengetahui prioritas perubahan kebutuhan serta perubahan kebutuhan lain yang turut direalisasikan sebagai dampak dari kebutuhan awal yang dirubah. Selain itu, stakeholder dapat mengetahui perkiraan budget sebagai konsekuensi dari pilihan perubahan awal yang direalisasikan.

### 5.2 Saran

Adapun saran untuk penelitian selanjutnya adalah:

Menambahkan variabel yang lebih kompleks untuk memodelkan interaksi antar kebutuhan perangkat lunak diantaranya adalah: Budget, Use Case Trace, Comment, Kebutuhan ID Predecessor, Estimated Completion Time (likely, optimistic and pessimistic), time, deadline, type of kebutuhan, tujuan pembuatan perangkat lunak, constrain perangkat lunak, team, transaksi, serta even driven

Menganalisa tipe dependensi yang lain (cost related dependency, time related dependency) sebagai variabel pemodelan.

Menyempurnakan casetool LTS agar efektif dan efisien dalam mengilustrasikan model perubahan kebutuhan serta untuk menelusuri dampak perubahan tersebut dalam sistem perangkat lunak.

## DAFTAR PUSTAKA

- [1] Baier, C. dan Katoen, J.-P. 2008. *Principles of Model-Checking*. MIT Press, Cambridge.
- [2] A. Di Pierro, C. Hankin, and H. Wiklicky. 2014. *Approximate Non-Interference*. Journal of Computer Security, 12(1):37–82.
- [3] Boudol, G. 1992. *Asynchrony and the  $\pi$ -calculus*. Rapport de Recherche 1702. INRIA Sophia-Antipolis. URL <http://www.inria.fr/RRRT/RR-1702.html>.
- [4] De Nicola, R. dan Loreti, M. 2007. *Multi Labelled Transition Systems: A Semantic Framework for Nominal Calculi*. Electr. Notes Theor. Comput. Sci.. Vol. 169, Hal 133-146.
- [5] Sommerville, I. 2001. *Software Engineering*. Edisi 6. Yuhilza Hanum. Erlangga, Yogyakarta.
- [6] Uchitel, S. dan Kramer, J. dan Magee, J. 2003. *Behaviour model elaboration using partial labelled transition systems*. Pada Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering Series = ESEC/FSE-11. Helsinki, Finland.
- [7] D. Dams and K. Namjoshi. 2004. *The Existence of Finite Abstractions for Branching Time Model Checking*. In Proc. of the Nineteenth Annual IEEE Symposium on Logic in Computer Science, pages 335–344, 13-17 July, Turku, Finland. IEEE Computer Society Press.
- [8] Marino Miculan and Marco Peressotti. 2013. *Weak bisimulations for labelled transition systems weighted over semirings*. In Proceedings of the Second international conference on Theoretical Aspects of Computing, pages 454–468. Springer-Verlag.
- [9] B. Klin and V. Sassone. 2012. *Structural operational semantics for stochastic and weighted transition systems*. Journal of Automata, Languages and Combinatorics, 8(2):187–218.
- [10] M. Huth, R. Jagadeesan, and D. A. Schmidt. 2013. *A domain equation for refinement of partial systems*. Accepted for publication in the journal Mathematical Structures in Computer Science. In press; 1 February.
- [11] C. Jard and T. Jérón. 2005. *Theory, principles and algorithms*. International Journal on Software Tools for Technology Transfer, 7(4):297–315.