



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - KS 091336

**DEVELOPMENT OF MOBILE CLIENT APPLICATION
FOR TRAFFIC REPORT SYSTEM BASED ON
ANDROID OPERATING SYSTEM**

GABRIEL SURYADI KUSUMA
NRP 5206 100 100

SUPERVISOR

Dr.Eng. Febriliyan Samopa, S.Kom, M.Kom
Faizal Johan Atletiko, S.Kom

INFORMATION SYSTEM DEPARTMENT
Information Technology Faculty
Sepuluh Nopember Institute of Technology
Surabaya 2010

**PEMBUATAN APLIKASI MOBILE CLIENT TRAFFIC
REPORT SYSTEM BERBASIS SISTEM OPERASI
ANDROID**

TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

GABRIEL SURYADI KUSUMA
5206 100 100

Surabaya, 1 Agustus 2010

**KETUA
JURUSAN SISTEM INFORMASI**



Ir. A. HOLIL NOOR ALLI, M.KOM.
NIP 1966 0602 1992 031002

**PEMBUATAN APLIKASI MOBILE CLIENT TRAFFIC
REPORT SYSTEM BERBASIS SISTEM OPERASI
ANDROID**

TUGAS AKHIR

Disusun untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada

Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

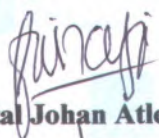
GABRIEL SURYADI KUSUMA
5206 100 100

Disetujui Tim Penguji:

Tanggal Ujian : 28 Juli 2010
Periode Wisuda : 101


Dr.Eng. Febriliyan Samopa,S.Kom,M.Kom

(Pembimbing I)


Faizal Johan Atletiko, S.Kom

(Pembimbing II)


Bambang Setiawan, S.Kom, M.T

(Penguji I)


Edwin Riksakomara S.kom, M.T

(Penguji II)

**PEMBUATAN APLIKASI MOBILE CLIENT TRAFFIC
REPORT SYSTEM
BERBASIS SISTEM OPERASI ANDROID**

Nama Mahasiswa : GABRIEL SURYADI KUSUMA
NRP : 5206 100 100
Jurusan : SISTEM INFORMASI FTIF-ITS
**Dosen Pembimbing I : Dr.Eng. Febriliyan Samopa, S.Kom,
M.Kom**
Dosen Pembimbing II : Faizal Johan Atletiko, S.Kom

ABSTRAK

Tingkat kepadatan arus lalu-lintas di daerah perkotaan yang tidak pasti dan kerap berubah secara drastis mengakibatkan kekhawatiran dari pengguna kendaraan bermotor di jalan atas kemungkinan munculnya kemacetan arus lalu-lintas. Bagi masyarakat modern yang memiliki aktifitas yang sangat banyak tentu saja kemacetan lalu-lintas adalah suatu hal yang ingin dihindari. Cara terbaik untuk menghindari kemacetan adalah dengan mengetahui daerah mana saja yang sedang terjadi kemacetan lalu-lintas dan mengetahui tingkat kemacetannya. Namun hal tersebut bukanlah perkara mudah karena belum ada sistem yang memungkinkan pengguna jalan raya dalam mengakses informasi kemacetan arus lalu-lintas tersebut.

Informasi yang disediakan oleh server Traffic Report System sendiri adalah data informasi arus lalu-lintas di sekitar pengguna yang mengakses Traffic Report System. Untuk menentukan posisi pengguna, penulis digunakan GPS dengan pertimbangan tingkat keakuratan informasi yang tinggi.

Aplikasi mobile client Traffic Report System yang dibangun di atas sistem operasi android ini dapat mengirimkan laporan

kepada server berupa area yang dipantau serta tingkat kemacetan arus lalu-lintasnya. Informasi yang diolah server kemudian dikembalikan ke aplikasi grafis untuk diterjemahkan ke dalam tampilan antar muka yang memudahkan pengguna dalam memahami informasi yang diberikan server. Sistem informasi yang dimasukkan kepada server merupakan kumpulan dari informasi pengguna lain sehingga arus informasi akan berjalan dari pengguna kepada pengguna.

Kata kunci : Client Traffic Report System, GPS, Sistem Operasi Android

ABSTRAK

Untuk memantau arus lalu-lintas di daerah pedesaan yang tidak memiliki jaringan internet, maka diperlukan sistem pemantauan yang dapat diakses secara langsung ke server. Sistem ini akan memantau kondisi lalu-lintas di daerah pedesaan dan mengirimkan informasi ke server. Informasi ini akan diolah server dan dikembalikan ke aplikasi grafis untuk diterjemahkan ke dalam tampilan antar muka yang memudahkan pengguna dalam memahami informasi yang diberikan server. Sistem informasi yang dimasukkan kepada server merupakan kumpulan dari informasi pengguna lain sehingga arus informasi akan berjalan dari pengguna kepada pengguna.

Informasi yang di-stalakan oleh server traffic report system adalah data lalu-lintas dan informasi di sekitar pengguna yang digunakan Traffic Report System. Untuk memudahkan pengguna dalam memahami informasi yang diberikan server, maka diperlukan tampilan antar muka yang memudahkan pengguna dalam memahami informasi yang diberikan server.

Informasi yang di-stalakan oleh server traffic report system adalah data lalu-lintas dan informasi di sekitar pengguna yang digunakan Traffic Report System. Untuk memudahkan pengguna dalam memahami informasi yang diberikan server, maka diperlukan tampilan antar muka yang memudahkan pengguna dalam memahami informasi yang diberikan server.

DEVELOPMENT OF MOBILE CLIENT APPLICATION FOR TRAFFIC REPORT SYSTEM BASED ON ANDROID OPERATING SYSTEM

Student : GABRIEL SURYADI KUSUMA
NRP : 5206 100 100
Department : SISTEM INFORMASI FTIF-ITS
Supervisor I : Dr.Eng. Febriliyan Samopa, S.Kom,
M.Kom
Supervisor II : Faizal Johan Atletiko, S.Kom

Abstrak

Traffic density in a metropolis area keep changing erratically into a level that the urban citizen would worry about. For a modern citizens with a lot of work stuff, this unpredictable traffic density, which oftenly lead into a traffic jam, is a phenomenon, no matter how ought to be avoided. A hypothetical method to avoid a traffic jam is knowing which area currently suffered traffic jam in which level of density. But practically, this method is difficult to implemented, since such system that allows user to access the traffic jam information is not existed yet.

The Traffic Report System Server, actually is providing information about the traffic density level for any user accesing the Traffic Report System Server. GPS technology is used for defines the user current position. It is choosen among any other location based technology for its high accuracy in providing user current location.

Main usage of the mobile client application built in android operating system is to send report to server about traffic density in such area. This information itself come from the application user thus the system built from user, and for user. After gaining some information the mobile client application would convert the

user-interface information itu a data, which replied by user that converting it into a XML file..

Keyword : Client Traffic Report System, GPS, Android Operating System

Supervisor I : Dr. Tjandjaja, S.T., M.Pd., S.Kom.
Supervisor II : Fauzan Johari, S.T., S.Kom.
M.Kom
Department : Sistem Informatika FTIF-ITS
NIP : 5206 100 100
Student : GABRIEL SURYADI KRAMA

Abstrak

The traffic data in an metropolitan area keeps changing every day. It is a fact that the urban citizen would worry about. For a modern citizen with a lot of cars, the unpredictable traffic jam is a headache, which should lead into a traffic jam is a phenomenon, no one can be thought to be avoided. An important method to avoid a traffic jam is knowing which an currently affected traffic jam in which area of domain. The present the method is difficult to implement, even the system that want to access the traffic jam information is not easy to get.

The Traffic Report System, actually is providing information about the traffic domain level for any time occurring. The Traffic Report System, GPS technology is used for helping the user know position. It is needed more any other location based technology for the high accuracy in providing user current location.

Main focus of the mobile client application built in android operating system is to send report to server about traffic density in each area. This information will come from the application user that the system built from user and for user after knowing how information the mobile client application would connect the

KATA PENGANTAR

Segala puji dan syukur penulis tuturkan ke hadirat Allah Tritunggal, Bapa di surga, yang telah memberikan kekuatan dan kehidupan untuk penulis sehingga penulis dapat menyelesaikan laporan tugas akhir dengan judul :

PEMBUATAN APLIKASI MOBILE CLIENT TRAFFIC REPORT SYSTEM BERBASIS SISTEM OPERASI ANDROID

Tugas akhir ini tidak akan pernah terwujud tanpa bantuan dan dukungan dari berbagai pihak yang telah meluangkan waktu, jiwa dan pikirannya bagi penulis untuk menunaikan tugas akhir ini. Pada kesempatan ini penulis mengucapkan terima kasih kepada pihak-pihak di bawah ini, yaitu:

- Bapak Febriliyan Samopa, selaku dosen pembimbing I yang telah meluangkan waktu dan pikiran di tengah kesibukan beliau untuk membimbing dan mengarahkan penulis dalam mengerjakan tugas akhir ini.
- Bapak Faizal Johan, selaku dosen pembimbing II yang telah memberikan saran dan membagikan pengetahuan beliau untuk membantu penulis menyelesaikan tugas akhir ini.
- Orang tua yang teramat sangat saya cintai, Iyus, Martin, Dian, dan Tika yang telah dan akan menjadi saudara yang amat saya sayangi. Terima kasih untuk doa dan semangat yang telah kalian berikan.
- ANONIMS, keluarga dan sahabat yang telah mengisi hidup saya dengan kebahagiaan dan mengajarkan jutaan hal lain yang jauh lebih penting daripada jutaan materi. *We're not rascal nor radical, but this is just the way we are!!*
- KMSI yang telah memberi saya waktu dan kesempatan untuk berkembang menjadi mahasiswa sepenuhnya. *They*

can change our name, ruin our castle, but they can't take our FREEDOM.

- Bogi dan Yulianto sebagai partner MTRS, Husni yang telah menyumbangkan ilmunya, Hatta yang telah merelakan printernya untuk di-*gasak* bersama-sama, Bagit, Pradipta dan Rizal yang telah menyumbangkan suaranya untuk seluruh penghuni Lab E-Business, DSS, dan PPSI yang telah berjuang bersama, bernyanyi bersama, *celubh* bersama, hingga melakukan segala kegiatan produktif bersama. Terima kasih untuk hari-hari indah tersebut. *Alon-alon, sing penting ora kelakon.*
- Penunggu dan sesepuh GW23A, Mas Ervan, Andre, Lukman, Adit, dan Galih yang telah banyak membantu saya dalam akomodasi peralatan hidup sehari-hari di kos.
- Pasukan Thole 2, yang telah memberikan saya ruang lapang di tengah kesibukan yang mendera. Persahabatan tanpa batas.
- Seluruh pihak yang telah membantu penulis dalam mengerjakan tugas akhir ini.

Penulis pun menyadari bahwa tugas akhir ini masih belum sempurna dengan segala kekurangan di dalamnya. Oleh karena itu penulis memohon maaf atas segala kekurangan dan kekeliruan yang ada di dalam tugas akhir ini. Penulis membuka pintu selebar-lebarnya bagi pihak-pihak yang ingin memberikan kritik dan saran bagi penulis untuk menyempurnakan tugas akhir ini. Semoga tugas akhir ini dapat bermanfaat bagi seluruh pembaca

Surabaya 28 Juli 2010

(Penulis)

DAFTAR ISI

ABSTRAK.....	iii
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan.....	2
1.3 Manfaat.....	2
1.4 Permasalahan.....	2
1.5 Batasan Permasalahan.....	3
1.6 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Aplikasi Mobile.....	5
2.2 Android Operating System.....	6
2.3 Eclipse.....	11
2.4 Google Maps.....	13
2.4 <i>Global Positioning System (GPS)</i>	14
BAB III METODOLOGI.....	17
3.1 Perumusan Permasalahan.....	18
3.2 Studi Literatur.....	18
3.3 Perancangan aplikasi <i>mobile client Traffic Report System</i>	18
3.4 Pembuatan aplikasi <i>mobile client Traffic Report System</i>	19
3.5 Testing.....	19
3.6 Analisis hasil pengujian aplikasi.....	20
3.7 Evaluasi dan revisi hasil pengujian.....	20
3.8 Pembuatan laporan.....	20
BAB IV PERANCANGAN SISTEM.....	21
4.1 Domain Model.....	21
4.2 Requirements.....	22
4.2.1 Analisis Kebutuhan.....	22

4.2.2	Kebutuhan fungsional dan non-fungsional	23
4.3	Use Case Diagram.....	23
4.4	Robustness Diagram	24
4.5	Sequence Diagram	26
4.6	Data Model.....	27
4.7	Class Diagram	27
4.8	Test Case	27
BAB V IMPLEMENTASI DAN UJI COBA SISTEM		29
5.1	Perangkat Implementasi Sistem	29
5.2	Implementasi Aplikasi <i>Mobile Client Traffic Report System</i>	29
5.3	Transfer data dengan web service	32
1.	Kondisi jalan	32
2.	Daftar jalan.....	32
3.	Pengguna.....	33
5.4	Implementasi kustomisasi <i>Google Maps</i>	33
5.5	Uji Coba Fungsionalitas Sistem	36
5.5.1	Test Case Pengguna melakukan login ke aplikasi	36
5.5.2	Test Case Pengguna melakukan pendaftaran	36
5.5.3	Test Case Pengguna melakukan proses unregistrasi	36
5.5.4	Test Case Menginputkan informasi jalan raya.....	36
5.5.5	Test Case Melihat map.....	37
5.5.6	Test Case Melihat map sesuai input pengguna	37
BAB VI.....		43
KESIMPULAN DAN SARAN.....		43
6.1	Kesimpulan	43
6.2	Saran	43
LAMPIRAN A GUI STORY BOARD.....		1
LAMPIRAN B DOMAIN MODEL		1
LAMPIRAN C USE CASE DIAGRAM		1
LAMPIRAN D USE CASE DESKRIPSI.....		1

LAMPIRAN E ROBUSTNESS DIAGRAM	1
LAMPIRAN F SEQUENCE DIAGRAM	1
LAMPIRAN G CLASS DIAGRAM	1
LAMPIRAN H TEST CASE.....	1

DAFTAR GAMBAR

Gambar 2.1 Tampilan emulator android	8
Gambar 2.2 Arsitektur Android	10
Gambar 2.3 Tampilan aplikasi Eclipse SDK	11
Gambar 2.4 Tampilan halaman DDMS	12
Gambar 2.5 Contoh Google Maps yang telah dikustomisasi	13
Gambar 2.6 Contoh perangkat yang menggunakan GPS.....	15
Gambar 3. 1 Flowchart metodologi	17
Gambar 5.1 potongan kode untuk menjalankan halaman login..	30
Gambar 5.2 potongan kode untuk menjalankan halaman login..	34
Gambar 5.3 potongan kode untuk menjalankan halaman login..	35
Gambar 5.4 potongan kode XML untuk menampilkan halaman pembuka.....	39
Gambar 5.5 hasil kode XML yang.....	40
menjadi antarmuka pengguna	40
Gambar 5.6 hasil keluaran file XML kondisi jalan.....	40
Gambar 5.7 hasil keluaran file XML daftar jalan	41
Gambar 5.8 hasil keluaran file XML daftar jalan	41

DAFTAR TABEL

Tabel 5.1 Spesifikasi perangkat implementasi sistem 38



BAB I

PENDAHULUAN

1.1 Latar Belakang

Setiap individu di masa sekarang ini memiliki kewajiban untuk menuntaskan tugas beserta aktivitas yang sangat padat. Namun di lain pihak masih banyak permasalahan yang menimbulkan hambatan bagi seseorang dalam menuntaskan tugas serta aktivitas, salah satu contoh hambatan yang paling sering timbul adalah kemacetan arus lalu-lintas yang sulit diprediksi karena tidak memiliki pola atau interval waktu kepadatan.

Kemacetan arus lalu-lintas ini sendiri telah menjadi suatu permasalahan yang ingin dihindari oleh para pengguna kendaraan bermotor di jalan raya terutama di kota-kota besar. Oleh karena itu dibutuhkan suatu sistem yang dapat membantu pengguna kendaraan bermotor dalam menghindari kemacetan arus lalu-lintas. Bantuan tersebut dapat berupa informasi mengenai kondisi arus lalu-lintas dari area yang akan dilalui oleh pengguna kendaraan bermotor.

Traffic Report System adalah suatu sistem yang dapat memberikan informasi kepada penggunanya mengenai kondisi arus lalu lintas disekitar mereka. Masukan informasi untuk sistem ini sendiri berasal dari pengguna yang melaporkan kondisi arus lalu-lintas yang dipantau. *Server Traffic Report System* kemudian mengolah informasi yang masuk dari pengguna dan meneruskan informasi tersebut ke pengguna lainnya. Aplikasi yang dibuat penulis bermanfaat untuk menerjemahkan informasi dari *server Traffic Report System* ke dalam sebuah tampilan antar muka yang serupa dengan Google Maps sehingga informasi arus lalu lintas dapat di pahami dengan baik oleh pengguna *Traffic Report System*. Dari sistem yang dibuat ini, pengguna tidak hanya menghindar dari kemacetan saja akan tetapi juga dapat membantu

mengurangi kepadatan arus lalu lintas dengan tidak mendatangi lokasi tersebut.

1.2 Tujuan

Tujuan dari tugas akhir ini adalah membuat aplikasi mobile sebagai client untuk *Traffic Report System* yang dapat menampilkan informasi grafis serupa Google Maps kepada pengguna telepon seluler berbasis Android.

1.3 Manfaat

Manfaat dari tugas akhir ini adalah :

1. Pengguna aplikasi *mobile client traffic report system* memungkinkan pengguna untuk mengetahui kondisi lalu lintas di sekitar pengguna.
2. Menciptakan komunitas masyarakat yang terhubung lewat aplikasi untuk berbagi informasi lalu lintas.

1.4 Permasalahan

Permasalahan yang akan diselesaikan dalam tugas akhir ini adalah:

1. Bagaimana mengubah *input* pengguna berupa data alfabetik dan grafik ke dalam angka sebagai masukan data kepada *server* dan juga sebaliknya mengenai cara untuk mengolah *output* dari *server* ke dalam tampilan grafik serupa Google Maps.
2. Bagaimana menghubungkan aplikasi android dengan web service dalam hal transfer data dan informasi berupa XML.
3. Bagaimana menggunakan teknologi GPS sehingga dapat memberikan informasi akurat dan mudah diperbaharui

mengenai posisi pengguna dan informasi jalan yang dibutuhkan pengguna.

4. Bagaimana meningkatkan presisi informasi lalu lintas untuk menghasilkan informasi kondisi lalu lintas yang akurat.

1.5 Batasan Permasalahan

Batasan permasalahan dalam tugas akhir ini adalah:

1. Aplikasi yang dikembangkan didesain untuk mengirim dan menerima data melalui penggunaan file XML.
2. Aplikasi yang dikembangkan hanya untuk perangkat klien yang berbasis sistem operasi Android.
3. Aplikasi yang dibangun berupa *dependant application* yang terhubung dengan *web service traffic report system* dengan kustomisasi *Google Maps* di dalamnya.
4. Aplikasi yang dibangun masih terbatas pada proses pengembangan dan penelitian.

1.6 Sistematika Penulisan

Sistematika penulisan buku tugas akhir dibagi menjadi 6 bab sebagai berikut.

BAB I

PENDAHULUAN

Bab ini berisi latar belakang, tujuan, manfaat, permasalahan, batasan permasalahan, tinjauan pustaka, metodologi, dan sistematika penulisan.

BAB II

TINJAUAN PUSTAKA

Bab ini menjelaskan istilah-istilah yang digunakan pada penulisan buku tugas akhir ini.

BAB III METODOLOGI

Bab ini membahas alur dan tata pengerjaan tugas akhir dari awal sampai selesainya tugas akhir.

BAB IV PERANCANGAN SISTEM

Bab ini menjelaskan rancangan aplikasi yang dibuat berdasarkan kebutuhan sistem. Rancangan tersebut digunakan untuk pembangunan aplikasi pada tugas akhir ini.

BAB V IMPLEMENTASI DAN UJI COBA SISTEM

Bab ini menjelaskan pembangunan aplikasi yang sesuai dengan rancangan. Selain itu, dijelaskan pula uji coba sistem dalam menjaga performa aplikasi.

BAB VI KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang dapat diambil dari tugas akhir ini dan saran demi kepentingan pengembangan selanjutnya dari sistem.

BAB II TINJAUAN PUSTAKA

Pada bagian ini akan dijelaskan teknologi dan konsep yang terlibat dalam pembuatan tugas akhir ini. Teknologi dan konsep yang akan dibahas antara lain adalah aplikasi mobile, *Android Operating System*, *Eclipse*, *Google Maps*, dan *Global Positioning System* atau yang dikenal dengan GPS

2.1 Aplikasi Mobile

Secara harfiah aplikasi mobile dapat diartikan sebagai sebuah aplikasi yang dapat dijalankan walaupun pengguna berpindah atau karena pengguna berpindah[1]. Namun secara padanan kata, aplikasi mobile dapat diartikan sebagai perangkat lunak yang dapat dibawa dalam peralatan portabel seperti *USB Flash Drive*, *Personal Digital Asisstants* (PDA), ataupun telepon seluler.

Berdasar teknologi yang digunakan aplikasi mobile ini dapat dibagi menjadi 4 macam yaitu:

1. Server Request Based
Perangkat portabel meminta akses data secara langsung kepada *server*. Teknologi interkoneksi yang digunakan adalah WAP dengan menggunakan WML (bahasa khusus hasil *markup* dari XML).
2. SMS Based
Menggunakan *Short Message Service* pada perangkat portabel untuk berhubungan dengan sebuah *server* khusus yang melayani aplikasi ini.

3. **SIM-Card Based**
Aplikasi dijalankan di atas *SIM Card* sebagai landasan *platformnya*. Menu baru akan muncul dalam perangkat portabel seiring instalasi aplikasi ke dalam *SIM Card*.
4. **Mobile Equipment Based**
Hampir sama dengan *SIM-Card Based*, namun tipe yang satu ini menggunakan perangkat portabel atau dengan kata lain *operating system (OS)* yang berjalan di dalam perangkat tersebut sebagai landasan tempat aplikasi berjalan. Aplikasi yang dibuat penulis tergolong ke dalam tipe *Mobile Equipment Based*.

2.2 Android Operating System

Android adalah sebuah *software* dengan di dalamnya meliputi sistem operasi, *middleware* dan *key application*. Android sendiri berjalan di atas modifikasi linux kernel dengan Java sebagai bahasa pemrograman bagi pihak pengembang yang ingin membuat aplikasi Android.

Selanjutnya aplikasi yang akan dibuat penulis menggunakan *Android Software Development Kit* dan *Eclipse Application Programming Interface* sebagai media penulisan *code* untuk pengembangan aplikasi.

Beberapa keuntungan dari menggunakan Android adalah fitur *library* yang ditulis dengan bahasa C namun memiliki *Dalvik Virtual Machine (built-in virtual machine* pada Android) untuk membantu pengguna dalam mengimpor *library* baru yang ditulis dalam berbagai bahasa pemrograman yang kompatibel dengan *Dalvik Virtual Machine*. Adapun fitur yang menjadi keunggulan di sistem operasi Android adalah :

- Framework Aplikasi yang mendukung penggantian ataupun penggunaan kembali komponen

- Emulator yang berfungsi sebagai perangkat maya yang digunakan sebagai media simulasi aplikasi pada saat tahap pengembangan sampai proses instalasi ke perangkat nyata.

Emulator memiliki komponen yang identik dengan perangkat telepon genggam yang berbasis Android OS. Setiap versi dari Android akan menghasilkan tampilan serta sistem yang berbeda pada setiap emulator. Hal ini memudahkan pihak pengembang dalam membuat aplikasi sehubungan dengan perbedaan yang cukup mencolok dari tiap versi Android OS.

- Hak akses bagi *third-party developer* untuk mengakses *Core Mobile Device Functionality* dengan standar API. Hak akses seperti ini tidak diberikan oleh setiap *brand handset* di dunia.
- SDK yang dapat diakses secara gratis, serta *keystore* yang dapat digunakan untuk memasukkan aplikasi yang dibangun ke dalam perangkat *handheld* secara gratis pula.
- Kombinasi data antara aplikasi mobile dengan internet. Sebagian besar API untuk android berada di dunia internet sehingga proses transaksi data antara dunia *offline* dan *online* dapat digunakan.
- *Dalvik Virtual Machine* (DVM) sebagai mesin virtual yang menjalankan fungsi pada android dan berfungsi pula sebagai *debugger* pada saat proses *running* aplikasi. Setiap aplikasi Android, berjalan pada proses independen yang diinstansiasi oleh *Dalvik Virtual Machine*. DVM inilah yang merupakan yang melakukan pengaturan terhadap sistem *multi-threading* dan *low-level memory management*. Dalvik Virtual Machine berfungsi sama seperti Java Virtual Machine dan BREW runtime. Dalvik

Virtual Machines mengeksekusi file dalam format .dex
(*Dalvik Executable*),



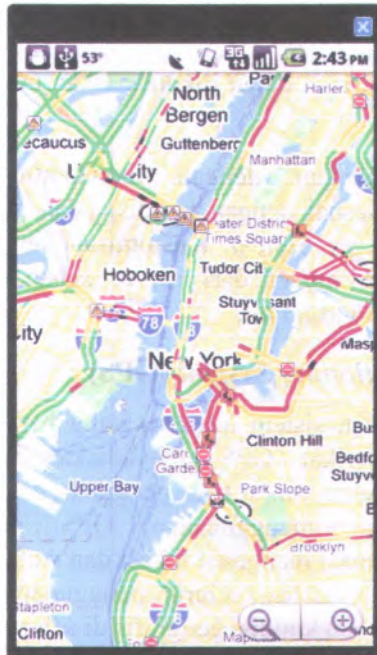
Gambar 2.1 Tampilan emulator android

Android sendiri memiliki arsitektur yang unik seperti ditampilkan pada gambar 2.2. Arsitektur android ini memiliki penjelasan sebagai berikut:

- Mula-mula Android akan diluncurkan dengan ketentuan seperti pada lapisan *Application Framework*. Aplikasi ini seperti seperti *email client*, *SMS program*, *calendar*, *browser*, *contacts manager* ditulis menggunakan bahasa pemrograman Java.
- Pada *layer* kedua (Arsitektur Framework API), pihak pengembang dapat melakukan kustomisasi pada aplikasi ini yang telah tersedia pada *layer* satu. Pihak pengembang memiliki hak yang sama untuk mengakses

2.4 Google Maps

Aplikasi *client* yang akan dibuat dalam Tugas Akhir ini menonjolkan tampilan grafis sebagai media informasi, maka dari itu aplikasi ini membutuhkan Google Maps sebagai visualisasi informasi kepadatan arus lalu lintas. Contoh kustomisasi *Google Maps* dapat dilihat pada gambar 2.5



Gambar 2.5 Contoh Google Maps yang telah dikustomisasi

Google Maps sendiri adalah sebuah layanan peta virtual yang diciptakan oleh Google. Google Maps bersifat gratis dan dapat dikustomisasi sesuai kebutuhan pengguna. Untuk memodifikasi Google Maps, maka dapat digunakan Google Maps API yang dapat dimasukkan ke dalam ADT milik android. API ini bersifat gratis dan dapat diciptakan sebanyak mungkin yang diinginkan

oleh pengguna. Namun sifat API ini unik untuk setiap domain yang ada sehingga tidak mungkin satu API digunakan untuk dua domain yang berbeda.

Google sebagai pengembang Android dan Google Maps memberikan kemudahan kepada pihak pengembang dengan menciptakan *library* yang menghubungkan sistem operasi dan aplikasi peta tersebut. Google Maps yang telah terintegrasi sebagai *library* di Android juga memiliki banyak fitur manajemen lokasi untuk Google Maps, seperti *Geocode* atau nilai lotitude dan langitude yang nantinya dapat dikonversikan menjadi lokasi dari pengguna aplikasi ini.

Kustomisasi dilakukan dengan membentuk *layer* untuk pewarnaan jalan sesuai dengan kondisi jalan pada informasi di *server Traffic Report System*. Kustomisasi ini melibatkan data pada output *XML object* dari server sebagai *overlay* untuk tampilan pada peta virtual ini.

2.4 Global Positioning System (GPS)

GPS adalah sebuah sistem navigasi yang berbasis pada satelit milik Amerika Serikat. GPS menyediakan layanan navigasi, penentuan posisi serta waktu bagi pengguna perseorangan secara gratis[4]. Dengan menggunakan GPS seseorang dapat memperoleh informasi mengenai lokasi dan waktu dengan tingkat keakuratan yang tinggi bagi seluruh penggunanya di segala jenis cuaca, waktu, ataupun kondisi geografis di seluruh dunia.

GPS sendiri pada dasarnya terdiri dari 3 bagian utama yaitu satelit yang mnejadi orbit di Bumi, stasiun kontrol dan monitoring di Bumi, dan *GPS receivers* yang dimiliki oleh manusia sebagai pengguna servis GPS ini. Informasi yang diberikan oleh GPS receiver ini terdiri dari penjabaran lokasi dengan 3 dimensi (latitude, longitude, dan altitude), jika dibutuhkan maka dapat diperoleh kondisi waktu bagi pihak yang membawa perangkat GPS tersebut.

2.3 Eclipse

Dalam aplikasi yang dibuat oleh penulis akan digunakan IDE (*Integrated Development Environment*) untuk java berupa Eclipse. Hal ini dikarenakan Eclipse telah terintegrasi dengan *development tools* dari Android. Eclipse sendiri merupakan *open source software* yang bersifat *portable* dan dapat digunakan untuk menulis beberapa bahasa pemrograman lainnya seperti C, C++, COBOL, Python, Perl, PHP sebagai *plugin*.



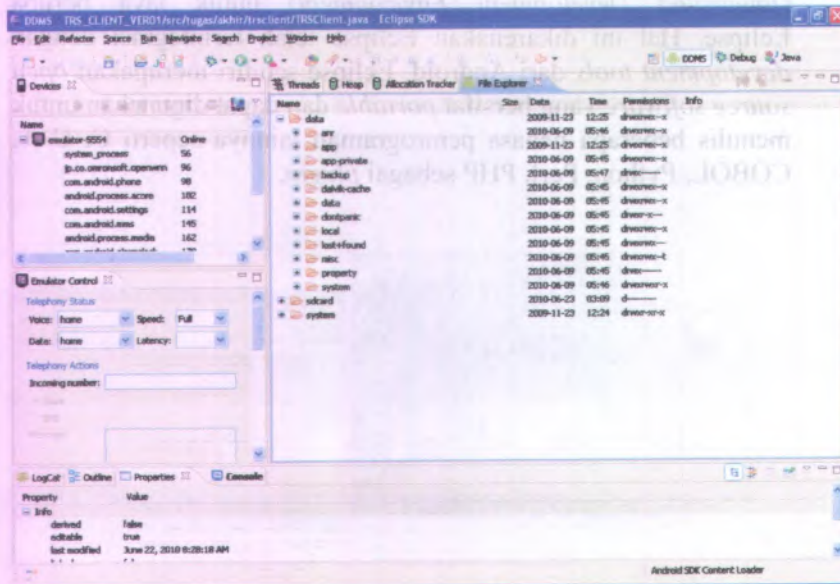
Gambar 2.3 Tampilan aplikasi Eclipse SDK

Hubungan yang telah terintegrasi antara Eclipse dan Android dapat membantu pengembang aplikasi dalam platform Android yang ingin menggunakan fitur *library* pada Java. Untuk menghubungkan Eclipse dengan Android ini cukup memasukkan Android ADT sebagai bagian dari project eclipse yang dibuat. Sifatnya hampir sama dengan *plug-ins* atau *add-ons*

Eclipse juga memiliki fitur DDMS (*Dalvik Debug Monitor System*) yang diperoleh setelah diintegrasikan dengan Android. DDMS ini berfungsi sebagai tampilan yang membantu pihak pengembang android dalam memonitor proses *debugging* pada aplikasinya. Sampai buku tugas akhir ini ditulis belum ada



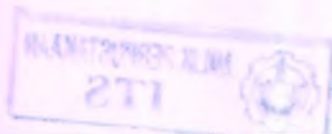
aplikasi IDE lain yang dapat berintegrasi dengan android sebaik Eclipse.



Gambar 2.4 Tampilan halaman DDMS

Halaman yang akan terintegrasi antara Eclipse dan Android dapat membantu pengembangan aplikasi dalam platform Android yang akan menggunakan fitur device pada saat ini. Hal ini terdapat dengan Android ini cukup membantu dalam pengembangan bagian dari project aplikasi yang dibuat dengan bantuan dengan fitur ini.

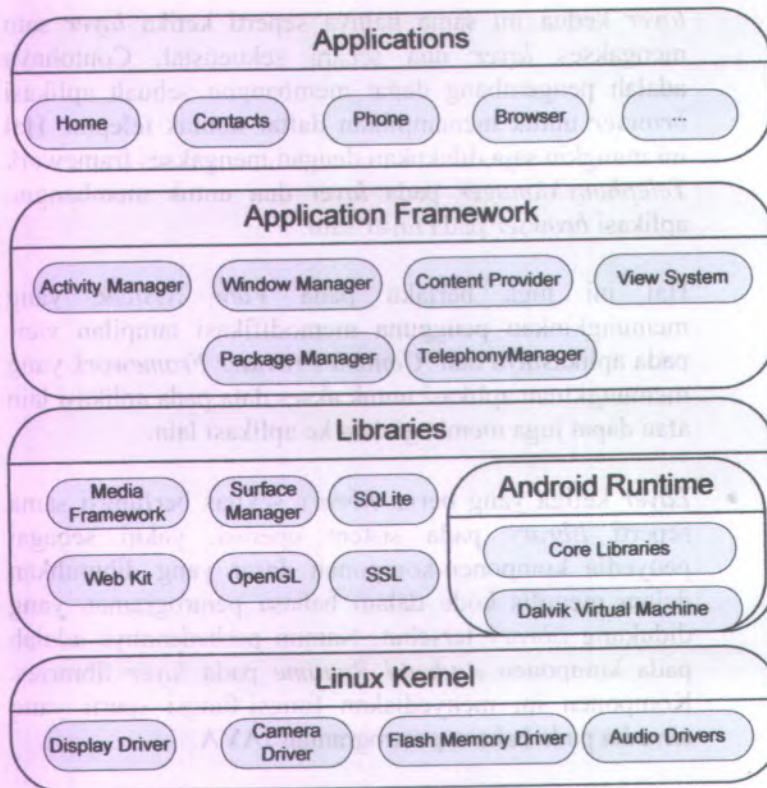
Eclipse juga memiliki fitur DDMS (Dalvik Debug Monitor Service) yang terintegrasi dengan Android. Fitur ini berfungsi sebagai tampilan yang membantu dalam pengembangan android dalam monitoring proses & device pada aplikasi. Fitur ini juga dapat membantu dalam pengembangan



layer kedua ini sama halnya seperti ketika *layer* satu mengakses *layer* dua secara sekuensial. Contohnya adalah pengembang dapat membangun sebuah aplikasi *browser* untuk menampilkan daftar kontak telepon. Hal ini mungkin saja dilakukan dengan mengakses framework *TelephonyManager* pada *layer* dua untuk membangun aplikasi *browser* pada *layer* satu.

Hal ini juga berlaku pada *View System*, yang memungkinkan pengguna memodifikasi tampilan *view* pada aplikasinya dan *Content Provider Framework* yang memungkinkan aplikasi untuk akses data pada aplikasi lain atau dapat juga membagi data ke aplikasi lain.

- *Layer* ketiga yang berisi *library* sekilas berfungsi sama seperti *library* pada sistem operasi, yakni sebagai penyedia komponen-komponen dasar yang dibutuhkan dalam menulis kode dalam bahasa pemrograman yang didukung *library* tersebut. Namun perbedaannya adalah pada komponen *Android Runtime* pada *layer* *libraries*. Komponen ini menyediakan fungsi-fungsi seperti yang tersedia pada bahasa pemrograman JAVA.
- Lapisan terbawah atau nomor empat adalah lapisan terpenting yaitu Kernel Linux versi 2.6 untuk melayani services inti operating system seperti security, memory management, process management, network stack dan driver model. Kernel juga bertindak sebagai abstraction layer antara hardware dan software stack.



Gambar 2.2 Arsitektur Android

Google Inc. sendiri sebagai salah satu perusahaan yang memiliki Android meramalkan Android sebagai *mobile OS* masa depan yang akan menjadi acuan semua pengembang aplikasi mobile dikarenakan kompatibilitasnya terhadap hampir semua lingkungan mobile yang pernah ada. Hal ini tentunya menjadikan pengembangan aplikasi berbasis platform Android sebagai suatu kegiatan yang prospektif.

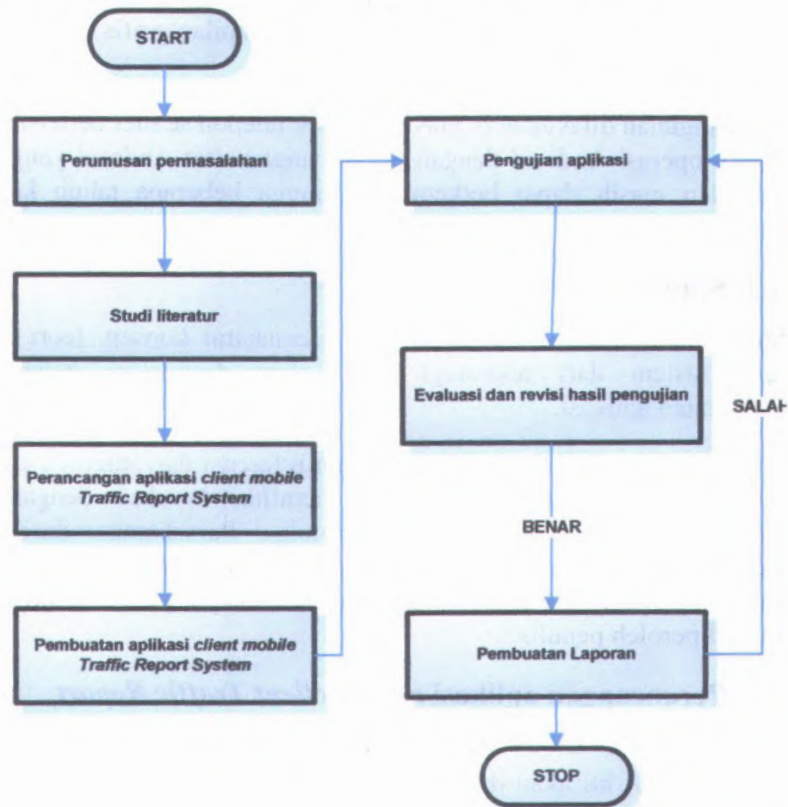


Gambar 2.6 Contoh perangkat yang menggunakan GPS

Sekalipun layanan GPS bersifat gratis, namun perangkat penerima layanan GPS ini tidaklah gratis. Pengguna harus membeli perangkat GPS yang dapat digunakan langsung untuk menangkap layanan GPS. Namun untuk fitur pada Android dan Google Maps, perangkat penerima GPS ini telah diberikan secara satu kesatuan dengan perangkat keras seperti telepon genggam.

BAB III METODOLOGI

Pembuatan tugas akhir ini terbagi menjadi beberapa alur dan tahap pengerjaan yang meliputi awal pembuatan aplikasi sampai pembuatan laporan untuk tugas akhir. Deskripsi dari alur tersebut dapat dilihat pada flowchart seperti pada gambar 3.1.



Gambar 3. 1 Flowchart metodologi

3.1 Perumusan Permasalahan

Permasalahan yang ditemukan penulis adalah bahwa pembangunan sistem informasi kemacetan lalu-lintas *Traffic Report System* belum memiliki aplikasi *client* yang bersifat *mobile*.

Maka dirumuskan suatu pembangunan aplikasi *client* untuk *server Traffic Report System* yang mampu mengirimkan informasi kepada *server* berupa kordinat untuk kemudian menampilkan informasi dari *server* berupa tampilan grafis Google Maps yang telah dikustomisasi.

Pembangunan dilakukan di atas perangkat telepon seluler berbasis sistem operasi Android dengan pertimbangan sifat Android yang *free* dan masih dapat berkembang hingga beberapa tahun ke depan.

3.2 Studi Literatur

Studi literatur akan dilakukan guna memahami konsep, teori, serta sistem dari teknologi yang akan digunakan dalam pembuatan aplikasi.

Literatur dalam proses ini dapat berupa referensi dari internet, e-book, paper, ataupun jurnal yang memiliki korelasi dengan teknologi aplikasi yang dibuat penulis. Pemahaman dasar mengenai pembangunan aplikasi dengan teknologi Android, GPS, dan *Google Maps* akan dipelajari secara bertahap melalui literatur yang diperoleh penulis.

3.3 Perancangan aplikasi *mobile client Traffic Report System*

Pada tahapan ini akan ditentukan konsep dan desain yang akan digunakan untuk aplikasi. Konsep dan desain dibangun sesuai dengan gambaran sistem yang telah digambarkan secara tidak

langsung pada bagian awal buku tugas akhir ini perihal aplikasi *mobile client Traffic Report System*.

3.4 Pembuatan aplikasi *mobile client Traffic Report System*

Setelah tahapan analisa dan desain telah selesai, maka tahapan selanjutnya adalah membangun aplikasi *mobile client Traffic Report System* pada emulator Android untuk selanjutnya diunggah ke dalam telepon seluler yang berbasis sistem operasi Android.

3.5 Testing

Pengujian dilakukan pada telepon seluler yang telah dimasukkan aplikasi. Selanjutnya pengujian ini dibagi ke dalam 2 jenis yaitu:

1. Pengujian fungsional

Fungsi dasar dari sistem akan diuji untuk menentukan keabsahan dari informasi yang beredar di dalam sistem. Informasi-informasi tersebut di antaranya adalah validasi posisi pengguna, konektivitas aplikasi ke server serta ketepatan pencitraan informasi server ke dalam tampilan grafis serupa Google Maps.

2. Pengujian non-fungsional

Performa aplikasi dan ketersesuaian aplikasi terhadap perangkat keras berbasis android, adalah bahan yang diujikan di dalam pengujian non-fungsional ini. Performa meliputi kemampuan aplikasi dalam mengakses map dan proses transfer data pada *server* dan ketersesuaian terhadap perangkat keras meliputi kemampuan dari aplikasi untuk diinstall ke dalam perangkat keras berbasis android di dunia nyata.

3.6 Analisis hasil pengujian aplikasi

Setelah melalui tahapan pengujian maka hasil pengujian tersebut akan analisis terkait performa aplikasi dan ketersesuaiannya dengan rancangan sistem di awal.

3.7 Evaluasi dan revisi hasil pengujian

Pada tahap ini hasil monitoring diteliti dan bila ada kesalahan dilakukan perbaikan sampai selesai. Setelah itu dibuat kesimpulan dan didokumentasikan.

3.8 Pembuatan laporan

Setiap langkah-langkah pengerjaan tugas akhir ini dari awal hingga akhir didokumentasikan dan ditulis dalam sebuah laporan yang sesuai dengan format tugas akhir sehingga menghasilkan laporan tugas akhir.

BAB IV PERANCANGAN SISTEM

Pada bab ini dijelaskan mengenai rancangan desain aplikasi yang dibuat berdasarkan kebutuhan sistem *mobile client* untuk aplikasi *Traffic Report System*. Desain ini akan digunakan untuk pembangunan aplikasi tugas akhir. Pada bab ini juga akan diuraikan hal-hal yang berkaitan dengan perancangan komponen klien pada aplikasi *Traffic Report System*. Proses perancangan tersebut menggunakan referensi *ICONIX process* dengan alur pembuatan *domain model*, *requirements*, *use case diagram* serta deskripsi, *robustness diagram*, *sequence diagram*, *data model*, *class diagram*, dan *test case*.

4.1 Domain Model

Domain model adalah proses pendefinisian model dengan proyeksi objek utama yang membangun aplikasi. Domain model ini sendiri pada dasarnya dapat berubah jika sistem yang membangun aplikasi turut mengalami perubahan. Penyesuaian ini ditujukan untuk meningkatkan kesesuaian alur sistem dengan objek dan komponen yang membangun sistem.

Domain model *Traffic Report System* pertama kali dibangun seperti yang terlihat pada lampiran A gambar B.1. Namun seiring dengan perkembangan proses pengembangan *Traffic Report System* diketahui bahwa domain model awal tersebut belum cukup untuk merepresentasikan objek yang terlibat dalam *Traffic Report System*. Maka dilakukan proses pemutakhiran dari domain model *Traffic Report System*.

Penjelasan dari setiap objek yang terdapat dalam domain model terbaru adalah sebagai berikut :

1. Pengguna

Objek yang melakukan pengecekan pengguna untuk kemudian nantinya akan direlasikan dengan objek Daftar Pengguna.

2. Daftar Pengguna

Objek yang akan mengelompokkan pengguna berdasarkan statusnya

3. Jalan

Objek yang digunakan untuk menampilkan data jalan secara *default* yang nantinya akan direlasikan dengan objek Daftar Jalan

4. Daftar Jalan

Objek ini digunakan untuk mengelompokkan jalan berdasarkan kondisinya

5. Kondisi Jalan

Objek ini berfungsi untuk memberikan informasi kondisi suatu jalan (relasi dengan daftar jalan) dan memberikan informasi mengenai siapa dan kapan informasi kondisi jalan akan diupdate (relasi daftar pengguna)

4.2 Requirements

Requirements merupakan daftar kebutuhan dari aplikasi *Traffic Report System* dengan terlebih dulu melakukan analisa untuk kemudian menurunkannya menjadi dua bagian yaitu kebutuhan fungsional dan kebutuhan non-fungsional. Selanjutnya *requirement* ini dibutuhkan untuk membuat *use case*.

4.2.1 Analisis Kebutuhan

Aplikasi klien hanya membutuhkan satu pengguna yang memiliki kebutuhan dalam melakukan proses melihat,

mencari dan memasukkan data sebagai kebutuhan informasi bagi pengguna lainnya dimana semua kebutuhan tersebut dapat dijalankan dalam *platform mobile* untuk dapat mengikuti mobilitas dari pengguna.

Untuk memenuhi kebutuhan pengguna tersebut maka dibutuhkan implementasi teknologi berupa sistem operasi Android untuk mewujudkan terciptanya aplikasi *mobile*, internet dan web service sebagai peladen bagi proses sistem informasi yang dibutuhkan pengguna, dan GPS sebagai perangkat pencari lokasi pengguna yang *mobile*.

4.2.2 Kebutuhan fungsional dan non-fungsional

Kebutuhan fungsional aplikasi didefinisikan sebagai berikut :

1. Memberikan informasi status jalan: informasi status jalan yang diinginkan dan informasi status jalan disekitar.
2. Manajemen pengguna: menyimpan data pengguna dan berhenti berlangganan.
3. Melakukan update informasi status jalan.
4. Mengidentifikasi posisi pengguna saat online.
5. Otomasi kadaluwarsa informasi status jalan.

Sedangkan untuk kebutuhan non-fungsional, mencakup pengiriman data ke *client* sedinamis mungkin dan pengolahan data secepat mungkin.

4.3 Use Case Diagram

Use case yang dirancang berdasarkan realisasi dari *requirement* yang telah dibuat sebelumnya. Terdapat 2 aktor utama dari aplikasi *server Traffic Report System* yaitu:

- a. Pengguna

Merupakan pihak yang dapat mengakses aplikasi, mengirimkan informasi lalu lintas dan mendapatkan layanan informasi dari server Traffic Report System.

b. Sistem

Merupakan pihak internal dari sistem keseluruhan yang dapat melakukan fungsi otomatisasi tampilan map.

Rancangan *use case* ini juga dibagi menjadi 3 bagian besar yaitu Use Case Akses Aplikasi, Use Case Fungsionalitas Use Case Sistem.

4.4 Robustness Diagram

Robustness diagram dibuat didalam semua use case yang ada, hal ini untuk menjelaskan alur dari *use case* yang dirancang. Adapun *robustness diagram* yang terdapat dalam rancangan desain aplikasi *mobile client Traffic Report System* adalah:

1. Pengguna melakukan login ke aplikasi

Pengguna mengakses halaman login pada aplikasi untuk kemudian status dari pengguna akan dirubah menjadi aktif atau terlihat *online* pada sistem server. Hal ini ditujukan untuk pengembangan lebih lanjut dari aplikasi *Traffic Report System* itu sendiri.

Pada proses ini tidak ada input dari pengguna yang dilakukan secara manual, hal ini dikarenakan tujuan dari login pada aplikasi hanyalah mengubah status pengguna dari offline menjadi online.

2. Pengguna melakukan pendaftaran

Pengguna membuka halaman pendaftaran. *Field* yang harus diisi adalah nama, dan nomor telepon seluler. *Primary key* untuk pendaftaran adalah nomor IMEI yang secara otomatis akan diambil oleh sistem pada aplikasi.

3. Pengguna melakukan proses unregistrasi

Pengguna mengakses halaman unregistrasi. Selanjutnya permintaan pengguna akan diproses oleh servis pada server. Aplikasi klien hanya mengantarkan *request* kepada server dan menampilkan pesan ketika proses unregistrasi telah berhasil.

4. Melihat informasi jalan raya sesuai input

Pengguna mengakses halaman dialog input yang berisi field nama jalan yang akan dicari informasi lalu lintasnya. Selanjutnya input akan diverifikasi oleh aplikasi klien dan kemudian informasi tersebut dikirim ke server untuk kemudian diproses.

Hasil proses kemudian ditampilkan ke dalam peta beserta tampilan informasi lalu lintasnya. Nama jalan yang dimasukkan oleh pengguna akan menjadi titik pusat dari map.

5. Melihat map

Pengguna akan melihat tampilan map beserta tampilan informasi lalu lintas dengan posisi pengguna saat itu sebagai titik pusat dari map.

6. Mengedit interval auto refresh map

Interval update dibutuhkan karena kenyataan bahwa posisi pengguna pasti akan berubah. Dengan menggunakan GPS dapat diperoleh lokasi pengguna secara *real time*, namun hal tersebut akan berdampak pada penggunaan daya baterai berlebih dari perangkat telepon seluler. Maka diberikan fitur untuk menentukan interval update sistem secara berkala sehingga fitur GPS akan menyala pada interval yang telah ditetapkan pengguna untuk kemudian dimatikan ketika posisi pengguna telah terupdate.

7. Menginputkan informasi jalan raya

Pengguna mengakses halaman seperti pada *robustness*. Melihat Map dengan fitur tambahan berupa menu tingkat kemacetan dan menu *touchscreen* untuk menginputkan lokasi kemacetan secara dinamis.

8. Mengupdate tampilan map sesuai interval settingan pengguna

Sistem secara berkala akan melakukan pengecekan mengenai interval *update* sesuai *setting* awal dari pengguna.

4.5 Sequence Diagram

Sequence diagram memuat alur dalam use case dengan pendeskripsian yang mengarah pada pemrograman aplikasi. Sehingga sebelum merancang sequence diagram terlebih dahulu harus mengerti tentang teknologi yang akan diterapkan.

4.6 Data Model

Data Model merupakan desain interaksi antara domain model dengan *database*. Domain model tidak berelasi secara langsung dengan tabel-tabel pada database akan tetapi melalui *data access object* (DAO). Dalam desain rancangan sistem aplikasi *server Traffic Report System* terdapat 3 objek yang berelasi dengan *database* yaitu: pengguna, kondisi jalan, dan jalan. Untuk setiap domain model yang berelasi terhubung dengan DAO yang berbeda-beda.

4.7 Class Diagram

Class diagram adalah rancangan akhir yang menghasilkan suatu desain objek yang nantinya akan diimplementasikan pada pemrograman aplikasi. Untuk pengerjaan tugas akhir ini, *class diagram* merupakan hasil perkembangan dari *update* domain model dan *sequence diagram* sehingga menghasilkan desain objek yang lebih detail dan lengkap sehingga memudahkan dalam pemrograman aplikasi *server Traffic Report System*.

4.8 Test Case

Test case dirancang untuk menjaga performa aplikasi agar sesuai dengan desain yang dibuat. Dalam hal ini, *test case* akan dijalankan dengan beberapa skenario yang sesuai dengan alur pada use case diagram.

BAB V IMPLEMENTASI DAN UJI COBA SISTEM

Pada bab ini dijelaskan pembangunan aplikasi yang telah disesuaikan dengan desain aplikasi sebagaimana tertulis pada bab IV. Selain itu, dijelaskan pula proses uji coba sistem dalam mengukur performa aplikasi. Definisi untuk lingkup sistem yang dibahas di bab ini adalah sistem untuk klien.

5.1 Perangkat Implementasi Sistem

Proses pembuatan dan implementasi sistem dilakukan pada sebuah *Personal Computer* untuk selanjutnya diunggah ke *handheld smartphone* berbasis sistem operasi Android. Detail dari spesifikasi perangkat dapat dilihat pada tabel 5.1

5.2 Implementasi Aplikasi *Mobile Client Traffic Report System*

Aplikasi *mobile client Traffic Report System* pada tugas akhir ini dibangun di atas *platform* android sebagai sistem operasinya. Seperti telah dijelaskan dalam perangkat implementasi sistem mengenai perangkat keras dan lunak yang digunakan dalam pembangunan aplikasi klien ini.

Implementasi aplikasi ini sendiri dijalankan serta ditelusuri sementara dengan menggunakan aplikasi emulator. Pembuatan aplikasi dilakukan dengan memanfaatkan perangkat pengembangan Android (*Android SDK*) dengan *style* pemrograman berbasis objek yang mengikuti Java. Berikut adalah potongan kode dari aplikasi untuk modul login.

```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.firstpage);

    request = sfClient.getInstance().createRequest();

    //---Get IMEI---
    telephonyManager =
        (TelephonyManager) getSystemService(Context.TELEPHONY_S
        ERVICE);
        IMEI = telephonyManager.getDeviceId(); {
        textIMEI = (TextView)
        findViewById(R.id.textIMEI);
        textIMEI.setText(IMEI);

        butLogin = (Button)
        findViewById(R.id.butLogin);
        butLogin.setOnClickListener(new
        View.OnClickListener() {

        public void onClick(View v) {
        try{
        sfRequest request =
        sfClient.getInstance().createRequest();

        //---WEB SERVICE URL---
        request.setUrl("http://ebiz.is.its.ac.id:8080/trs/rest
        /pengguna/ubah/"+textIMEI.getText()+"/Online");

        request.setMethod("GET");
        request.execute();
        }catch(Exception err){
        Toast.makeText(getApplicationContext(),
        err.getMessage(), Toast.LENGTH_LONG).show();
        }
        Intent intent = new Intent
        (MainActivity.this, HomeActivity.class);
        startActivity(intent); }
        });
    }

```

Gambar 5.1 potongan kode untuk menjalankan halaman login

Secara garis besar *user interface* atau tampilan aplikasi android dengan menggunakan pemrograman murni sangatlah dimungkinkan. Akan tetapi Android telah mendukung konsep MVC (*Model View Controller*) dengan membagi dua komponen dasar aplikasi tersebut ke dalam file-file yang berbeda.

Untuk komponen view atau tampilan user interface, Android memiliki file dengan ekstensi XML yang mengatur layout dasar untuk aplikasi android dan puluhan widget milik Android maupun Java. Contoh dari file XML pada aplikasi ini dapat dilihat pada gambar 5.2. Hasil dari file XML tersebut akan menampilkan *layout interface* seperti yang dapat dilihat pada gambar 5.3.

Pada dasarnya aplikasi yang telah dibangun dapat dijalankan di dalam emulator. Namun bagaimanapun juga perilaku dari emulator tidak mungkin bisa identik sempurna dengan perilaku dari *handheld smartphone* mengingat resource yang menjalankan keduanya jauh berbeda (*processor personal computer* tempat emulator berjalan tentu jauh berbeda dengan *processor* yang ada di dalam *handheld smartphone*).

Oleh karena itu aplikasi yang telah dibangun di atas perangkat *personal computer* harus dapat diunggah ke dalam perangkat *handheld smartphone*. Android menyediakan fitur untuk menggabungkan semua *project* yang dibutuhkan dalam membangun sebuah aplikasi menjadi satu file berekstensi APK. File APK inilah yang akan di-*install* ke dalam *handheld smartphone*.

Proses instalasi sendiri dilakukan secara otomatis oleh *handheld smartphone* tergantung metode yang digunakan dalam mengunggah file APK tersebut. Jika menggunakan kabel data, maka pengguna harus mencari file APK tersebut ke dalam perangkat *smartphone* untuk kemudian memilih untuk melakukan proses instalasi. Namun jika menggunakan *bluetooth* maka

smartphone akan menampilkan notifikasi yang meminta user untuk melakukan instalasi secara langsung

5.3 Transfer data dengan web service

Transfer data untuk koneksi dengan Web Service sepenuhnya menggunakan *HTTP Method : GET* dimana pihak klien bersifat “pasif” dengan tidak mengirimkan parameter tersendiri melainkan memasukkan semua parameter yang dibutuhkan ke dalam variabel di URL. URL untuk mengakses *web service* adalah <http://ebiz.is.its.ac.id:8080/trs/rest/> dengan *sub folder* di dalamnya sebagai berikut:

1. Kondisi jalan

Digunakan untuk mendapatkan informasi mengenai kondisi jalan. Informasi yang didapatkan adalah geocode titik awal jalan, geocode titik akhir jalan, nama, dan status sebagai penanda tingkat kemacetan (kondisi) jalan tersebut. Secara detail URL yang digunakan adalah <http://ebiz.is.its.ac.id:8080/trs/rest/kondisi/carigeo/longitude/latitude>, dengan nilai longitude dan latitude sesuai posisi pengguna saat mengakses XML ini. Hasil keluaran dari XML kondisi jalan dapat dilihat pada gambar 5.4

2. Daftar jalan

Digunakan untuk mendapatkan informasi mengenai nama jalan yang berada di sekitar pengguna. Klien mengirimkan lokasi pengguna saat mengakses URL ini untuk kemudian mendapatkan kembalian berupa nama jalan yang berada di daerah tersebut. Radius dan pemilihan nama jalan yang ditampilkan merupakan kebijakan server. URL yang digunakan adalah <http://ebiz.is.its.ac.id:8080/trs/rest/jalan/carigeo/longitude/latitude>}, dengan nilai longitude dan latitude sesuai

posisi pengguna saat mengakses XML ini. Hasil keluaran dari XML daftar jalan dapat dilihat pada gambar 5.5.

3. Pengguna

Digunakan untuk mendapatkan informasi mengenai pengguna. Informasi yang diperoleh antara lain: IMEI, nama, status, nomor telepon. Use case akses aplikasi seluruhnya menggunakan layanan informasi agar dapat berjalan dengan baik. URL yang digunakan adalah <http://ebiz.is.its.ac.id:8080/trs/rest/pengguna/cari/id>, dengan nilai id adalah IMEI milik pengguna. Hasil keluaran dari XML pengguna dapat dilihat pada gambar 5.6

5.4 Implementasi kustomisasi *Google Maps*

Kustomisasi Google Maps menggunakan *overlay libraries* yang ada pada Android. Implementasi *overlay* ini dapat dilihat pada:

1. Pengguna mengirimkan informasi kemacetan

Pengguna menentukan jalan yang terjadi perubahan tingkat kemacetan dengan menekan titik awal kemacetan dan titik akhir kemacetan. Setiap titik yang ditekan oleh user akan ditandai dengan *icon marker* yang merupakan hasil overlay dari Google Maps. Fungsi *getLatitude* dan *getLongitude* digunakan untuk menentukan lokasi yang dipilih pengguna. Detail dari kode program untuk mencari posisi pengguna ditampilkan pada gambar 5.3.

```

public void onLocationChanged(Location
meloc) {
    // TODO Auto-generated method
    stub
    Toast.makeText (
        getApplicationContext(), "Location Changed
: Latid : " + meloc.getLatitude()+ " Longit : "
+ meloc.getLongitude(),
        Toast.LENGTH_LONG).show();

        getPoint=new
        GeoPoint ((int) (meloc.getLatitude()*1000000.0),
        (int) (meloc.getLongitude()*1000000.0));
        mc = map.getController();
        mc.animateTo(getPoint);
        mc.setZoom(14);
        map.setStreetView(true);
        map.setSatellite(false);

        mlocManager.removeUpdates(mlocListener);
    }

```

Gambar 5.2 potongan kode untuk menjalankan halaman login

2. Pengguna melihat kondisi jalan sekitar
Pengguna memilih menu *View Map Around* yang akan menampilkan Google Maps dengan titik pusat pada lokasi pengguna saat mengakses menu tersebut. Overlay dilakukan pada jalan sekitar yang ada di dekat pengguna. *Overlay* menggunakan class *ItemizedOverlay*.

```

Paint paint = new Paint();
paint.setAntiAlias(true);
Point point = new Point();

projection.toPixels(gp1, point);
if (mode == 1){
    if (defaultColor == 999){
        paint.setColor(Color.BLUE);
    }
    else {
        paint.setColor(defaultColor);
    }
    RectF oval = new RectF(point.x -
mRadius, point.y - mRadius, point.x + mRadius, point.y + mRadius);
    canvas.drawOval(oval, paint);
}
else if (mode == 2){
    if (defaultColor == 999){
        paint.setColor(Color.RED);
    }
    else {
        paint.setColor(defaultColor);
    }
    Point point2 = new Point();
    projection.toPixels(gp2, point2);
    paint.setStrokeWidth(5);
    paint.setAlpha(120);
    canvas.drawLine(point.x, point.y, point2.x,
point2.y, paint);
}else if (mode == 3){
    if (defaultColor == 999){
        paint.setColor(Color.GREEN);
    }else{
        paint.setColor(defaultColor);
    }
    Point point2 = new Point();
    projection.toPixels(gp2, point2);
    paint.setStrokeWidth(5);
    paint.setAlpha(120);
    canvas.drawLine(point.x, point.y, point2.x,
point2.y, paint);
    RectF oval = new RectF(point2.x - mRadius, point2.y
- mRadius, point2.x + mRadius, point2.y + mRadius);

    paint.setAlpha(255);
    canvas.drawOval(oval, paint);
}
}
return super.draw(canvas, mapView, shadow,
when);
}

```

Gambar 5.3 potongan kode untuk menjalankan halaman login

5.5 Uji Coba Fungsionalitas Sistem

Uji coba fungsionalitas sistem terdiri dari 6 skenario yang merupakan representasi dari 6 *use case*. Berikut adalah detail dari setiap uji coba.

5.5.1 Test Case Pengguna melakukan login ke aplikasi.

Pengujian dilakukan dengan menguji kemampuan aplikasi untuk mengakses halaman utama setelah berhasil melalui validasi login dengan *primary key* IMEI. Validasi dilakukan dengan mencoba memasukkan IMEI yang sudah ada dan belum ada dalam database server. Tabel skenario uji coba dapat dilihat pada lampiran H.1 dan H.2.

5.5.2 Test Case Pengguna melakukan pendaftaran

Pengujian dilakukan dengan menguji kemampuan aplikasi untuk menerima masukan dari pengguna untuk melakukan pendaftaran. Masukan pengguna menggunakan data yang telah dan belum dimasukkan untuk menguji kemunculan notifikasi. Tabel skenario uji coba dapat dilihat H.3 dan H.4.

5.5.3 Test Case Pengguna melakukan proses unregistrasi

Pengujian dilakukan dengan menguji kemampuan aplikasi dalam menangani proses unregistrasi atau penghapusan daftar pelanggan di dalam server. Tabel skenario uji coba dapat dilihat H.5 dan H.6.

5.5.4 Test Case Menginputkan informasi jalan raya

Pengujian dilakukan dengan menguji kemampuan aplikasi dalam membaca input dari user dan

mengirimkan data tersebut kepada server. Tabel skenario uji coba dapat dilihat H.7 dan H.8

5.5.5 Test Case Melihat map

Pengujian dilakukan dengan menguji kemampuan aplikasi dalam melakukan kustomisasi map berupa pewarnaan pada jalan sesuai tingkat kemacetan. Tabel skenario uji coba dapat dilihat pada H.11 dan H.12.

5.5.6 Test Case Melihat map sesuai input pengguna

Pengujian dilakukan untuk melihat area map yang disesuaikan dengan pilihan pengguna. Pengujian dilakukan dengan memasukkan data nama yang tidak valid dan data nama yang valid untuk menguji sistem dalam melakukan proses pencarian yang valid. Tabel skenario uji coba dapat dilihat pada H.9 dan H.10.

Tabel 5.1 Spesifikasi perangkat implementasi sistem

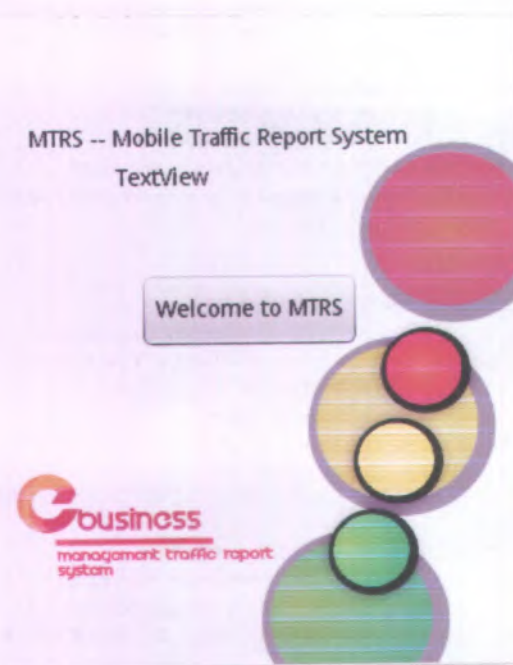
Perangkat Keras	Spesifikasi	Detil Spesifikasi
<i>Personal Computer</i>	Sistem Operasi	Microsoft Windows XP Professional (5.1, Build 2600)
	Prosesor	Intel®Pentium® Dual CPU E2160
	Memori	2040 MB RAM
<i>Handheld Smartphone</i>	Sistem Operasi	Android 2.1 (Eclair)
	Merek	Samsung Galaxy Spica I5700
	CPU	800 MHz
	Memori	128 MB RAM
	GSM Provider	Indosat IM2
	GPS	Ada, mendukung A-GPS
	Storage	180 MB Internal, 1024 MB Eksternal.
<i>Android Development Kit</i>	IDE	Eclipse Version: 3.5.2
	Java SDK	Java 1.6
	Android SDK	SDK Platform Android 2.1
	Google API Library	Google APIs by Google Inc, Android API 7, Revision 1

```

<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
  android:id="@+id/widget0"
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  android:background="@drawable/background"
  xmlns:android="http://schemas.android.com/apk/res/android"
>
  <Button
    android:id="@+id/butWelcome"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Welcome to MTRS"
    android:layout_x="83px"
    android:layout_y="189px"
  >
</Button>
  <TextView
    android:id="@+id/widget29"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="MTRS -- Mobile Traffic Report
System"
    android:layout_x="16px"
    android:layout_y="97px"
    android:textColor="#ff000000"
  >
</TextView>
  <TextView
    android:id="@+id/textFlexible"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="TextView"
    android:layout_x="70px"
    android:layout_y="121px"
    android:textColor="#ff000000"
  >
</TextView>
</AbsoluteLayout>

```

Gambar 5.4 potongan kode XML untuk menampilkan halaman pembuka



Gambar 5.5 hasil kode XML yang menjadi antarmuka pengguna

```

- <xmlKondisi>
  - <daftarKondisi>
    <akLatitude>-7.302544</akLatitude>
    <akLongitude>112.746933</akLongitude>
    <awLatitude>-7.300926</awLatitude>
    <awLongitude>112.740916</awLongitude>
    <keterangan>tecam</keterangan>
    <status>macet</status>
  </daftarKondisi>
</xmlKondisi>

```

Gambar 5.6 hasil keluaran file XML kondisi jalan

```

- <xmlJalan>
  <daftarJalan xsi:nil="true"/>
  <daftarJalan>kayoon</daftarJalan>
  <daftarJalan>Prof. Dr. Moestopo</daftarJalan>
  <daftarJalan>Jalan Walikota Mustajab</daftarJalan>
  <daftarJalan>Jalan Anggrek</daftarJalan>
  <daftarJalan>Jalan Pemuda</daftarJalan>
  <daftarJalan>Gubeng</daftarJalan>
  <daftarJalan>biliton</daftarJalan>
  <daftarJalan>Jalan Gub. Suryo</daftarJalan>
  <daftarJalan>Kertajaya</daftarJalan>
  <daftarJalan>Yos Sudarso</daftarJalan>
  <daftarJalan>Panglima Sudirman</daftarJalan>
  <daftarJalan>Jalan Sulawesi</daftarJalan>
  <daftarJalan>Dharmawangsa</daftarJalan>
</xmlJalan>

```

Gambar 5.7 hasil keluaran file XML daftar jalan

```

- <xmlPengguna>
  - <daftarPengguna>
    <id>0000000000000000</id>
    <nama>L3briga </nama>
    <nomor>085733043304</nomor>
    <status>Online</status>
  </daftarPengguna>
</xmlPengguna>

```

Gambar 5.8 hasil keluaran file XML daftar jalan

BAB VI KESIMPULAN DAN SARAN

Bab ini akan membahas kesimpulan mengenai hasil akhir dari pengerjaan tugas akhir ini dan juga saran berisikan informasi yang dibutuhkan untuk pengembangan tugas akhir ini di masa mendatang.

6.1 Kesimpulan

Kesimpulan yang diperoleh setelah menyelesaikan pengerjaan tugas akhir ini adalah:

1. Aplikasi telah berhasil dibuat dengan baik dan dapat dipasang ke dalam aplikasi perangkat keras berbasis android berupa *smartphone*.
2. Perbedaan jenis map yang digunakan dalam aplikasi berakibat data koordinat yang diberikan oleh server tidak bisa menghasilkan informasi yang presisi.
3. Aplikasi klien yang dibangun di tugas akhir ini memiliki ketergantungan penuh terhadap *server* bahkan sejak aplikasi mulai dijalankan. Jika server sedang offline maka pengguna tidak bisa mengakses aplikasi.

6.2 Saran

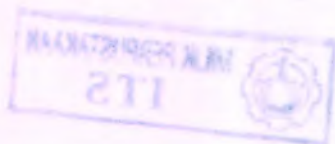
Untuk proses pengembangan aplikasi dalam masa mendatang, berikut adalah saran yang ditelaah oleh penulis dalam proses pengembangan aplikasi di masa kini.

1. Aplikasi sebaiknya ditambahkan fungsi *thread* yang memungkinkan sistem untuk memberikan notifikasi kepada pengguna mengenai sistem yang sedang berjalan namun belum termuat sepenuhnya (*loading process*). Praktik penggunaan thread dapat diimplementasikan saat halaman map sedang melakukan proses *searching* lokasi dengan GPS yang kerap memakan waktu lama sehingga dibutuhkan suatu notifikasi yang berjalan simultan



dengan proses *searching* tersebut untuk memberitahukan pengguna bahwa proses pencarian lokasi masih dalam proses *searching*.

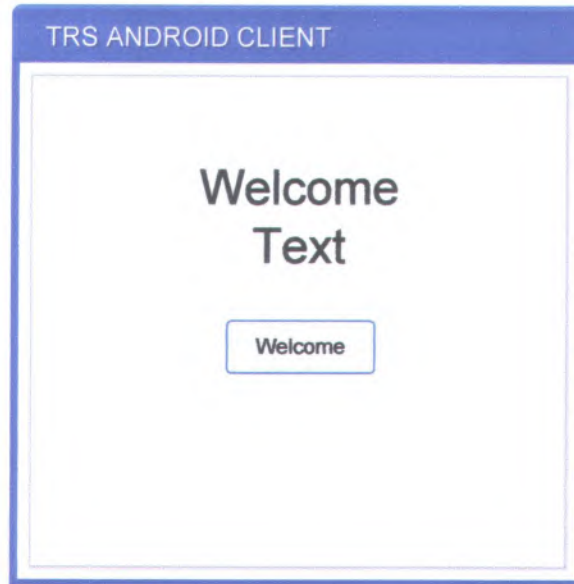
2. File KML yang memberikan data koordinat *geocode* pada map sebaiknya tidak hanya diproses dengan *streaming* langsung dari internet namun dimasukkan ke dalam *database* agar dapat diproses meskipun dalam kondisi *server* yang offline.
3. Penambahan fitur yang menyerupai fitur-fitur orisinal milik *Google Maps* seperti:
 - a. *Google Buzz* untuk menampilkan status pada *Buzz* milik *friends contact*. Informasi yang ditampilkan dapat berupa data keterangan pribadi sang pengguna mengenai kemacetan jalan yang dipantau.
 - b. *Google Latitude* untuk menampilkan lokasi teman dalam *friend contact Google Buzz* di dalam map. Informasi lokasi teman bisa ditampilkan dalam bentuk *clickable marker* untuk menampilkan informasi teman tersebut.
 - c. *Google Goggles* yang dapat melakukan pencarian dengan menggunakan gambar *image mining*. Penggunaan diarahkan kepada pencarian jalan dengan menggunakan image yang dimiliki pengguna semisal pengguna berada di jalan yang tidak diketahuinya, maka pengguna dapat mengunggah *image* via aplikasi untuk kemudian dilakukan proses pencarian.



DAFTAR PUSTAKA

- [1].Anonymous,Aplikasi Portable,http://id.wikipedia.org/wiki/Aplikasi_portabel
- [2].Anonymous, Ponsel Android terpopuler kedua di 2013, <http://inilah.com/news/read/teknologi/2010/01/26/306742/ponsel-android-terpopuler-kedua-di-2013>, Januari 2010
- [3].Anonymous,KMLAugmentations, <http://www.gps.gov/systems/augmentations/index.html> Juni 2010
- [4]. Anonymous, Global Positioning System Serving the World, <http://www.gps.gov/index.html>, Mei 2010
- [5].Ferrydiansyah,Application Mobile, Mobile Application Working Group , Februari 2002
- [6].GoogleDocs,KMLTutorial http://code.google.com/apis/kml/documentation/kml_tut.html Juni 2010
- [7].Indoandroid, Apa itu Android?, <http://indoandroid.blogspot.com/2009/01/apa-itu-android.html>, Januari 2010
- [8]Plusminus, Map Tutorial, <http://www.anddev.org/map-tutorials-f18/> Mei 2010

A.1 Halaman Pembuka



Gambar A. 1 Halaman Utama


A-4

A.2 Halaman Pengantar Registrasi



Gambar A. 22 Halaman Pengantar Registrasi

A.3 Halaman Pendaftaran



TRS ANDROID CLIENT

Name

Phone Number

IMEI

Gambar A.3 Halaman Registrasi

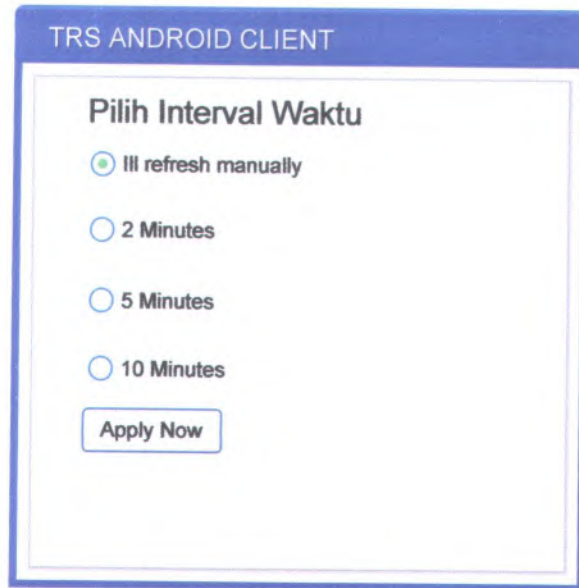
A-6

A.4 Halaman Tampilan Map



Gambar A.4 Halaman Tampilan Map

A.5 Halaman Konfigurasi



Gambar A. 5 Halaman Konfigurasi

A.6 Halaman Cari Kondisi Jalan



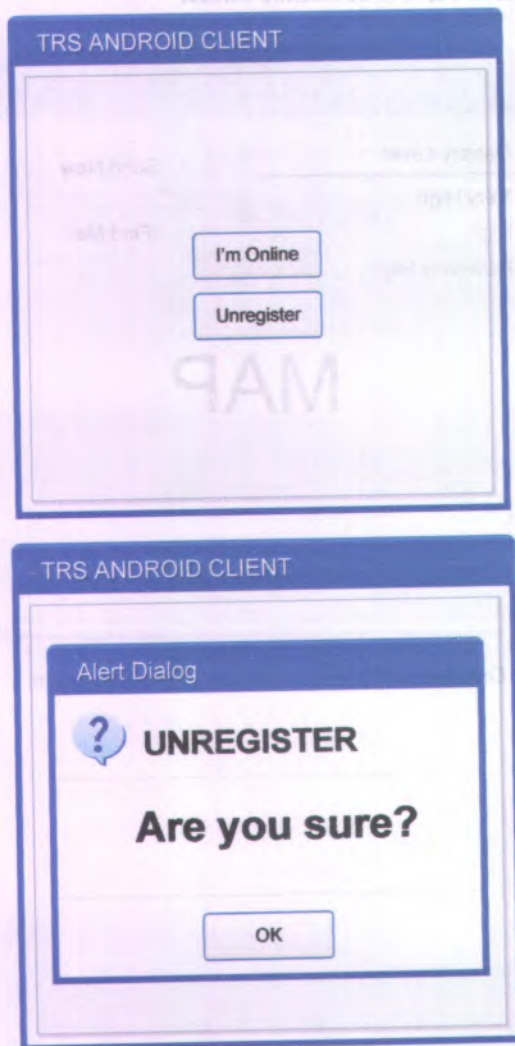
Gambar A. 6 Halaman Cari Kondisi Jalan

A.7 Halaman Kirim Kondisi Jalan

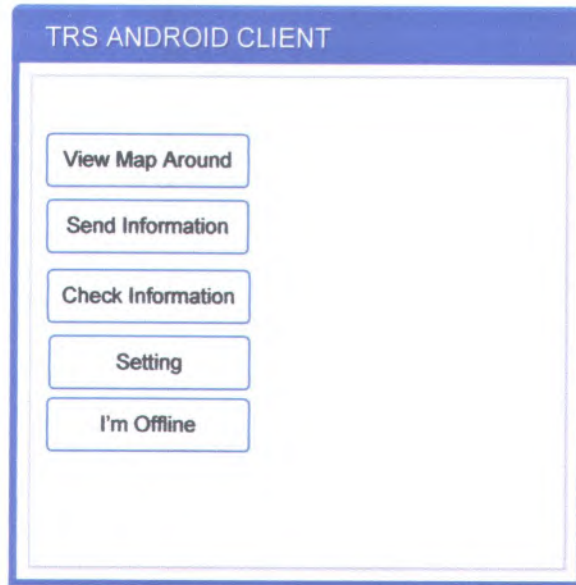


Gambar A. 7 Halaman Kirim Kondisi Jalan

A.7 Halaman Login dan Unregister

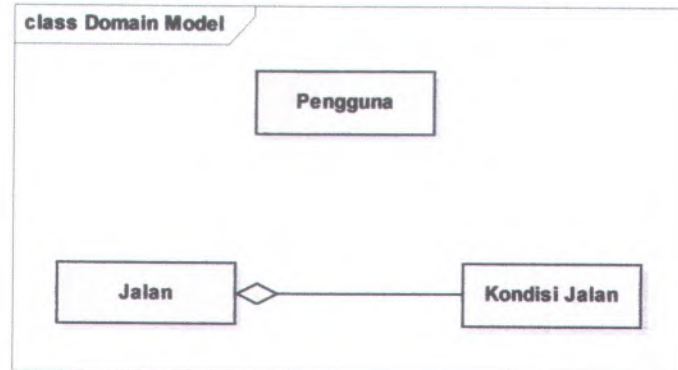


Gambar A. 8 Halaman Login dan Unregister

A.8 Halaman Utama**Gambar A. 9 Halaman Utama**

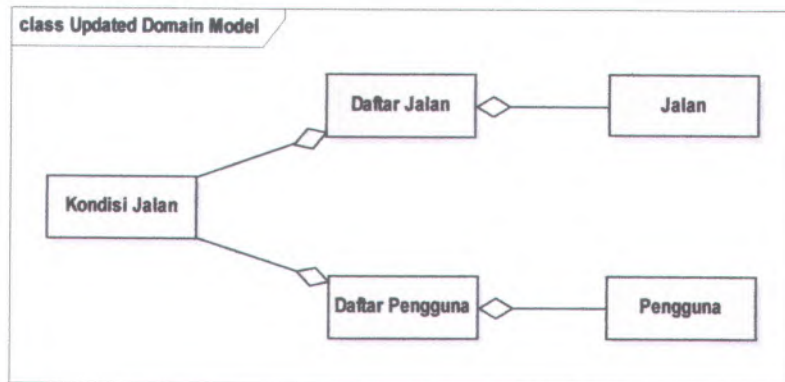
Lampiran B menggambarkan desain model yang akan dibangun pada tugas akhir ini. Desain model yang dirancang pada lampiran ini, berguna sebagai acuan dalam penentuan obyek pada pemrograman aplikasi Bali Travel Guide di iPhone.

B.1 Domain Model Awal



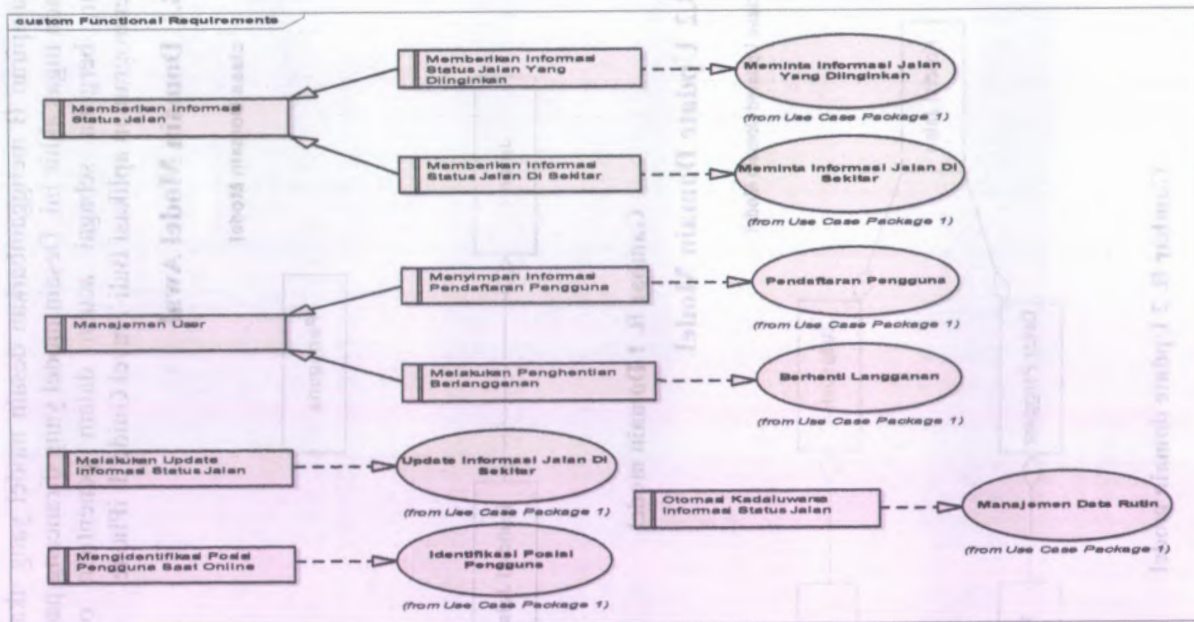
Gambar B. 1 Domain model

B.2 Update Domain Model



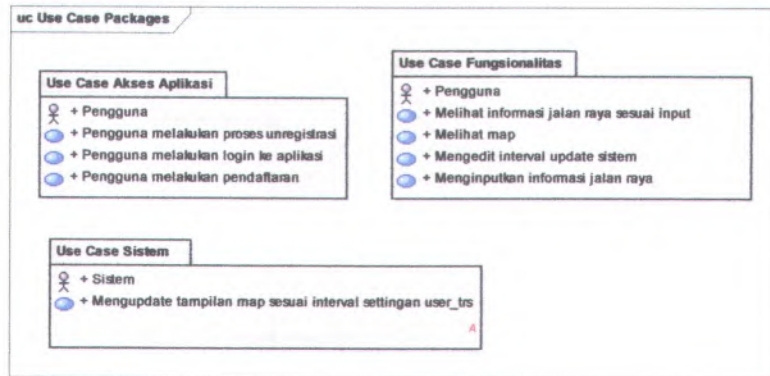
Gambar B. 2 Update domain model

B.3 Requirements



Gambar B.3 Requirements

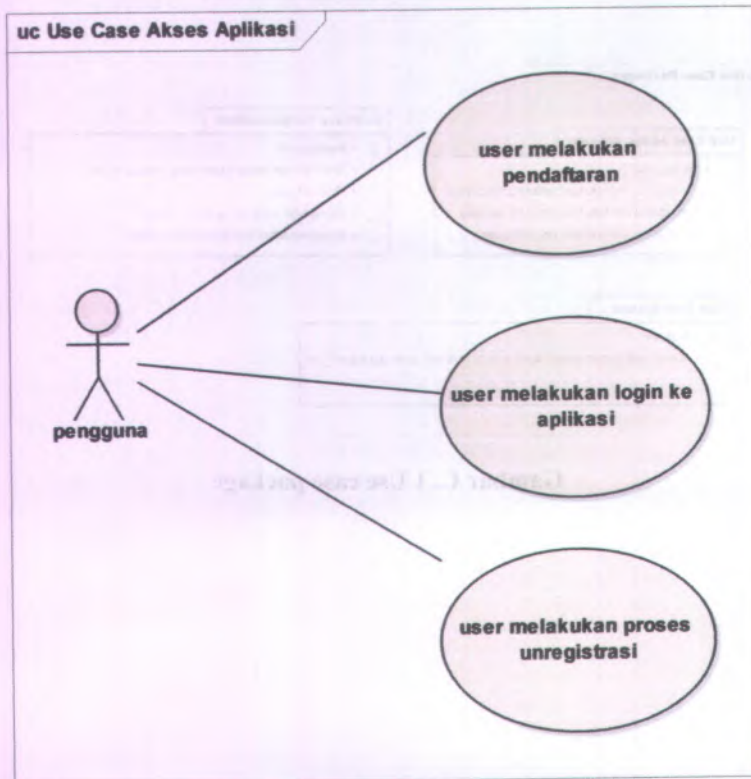
Lampiran C menampilkan diagram *use case* yang digunakan dalam pembangunan rancangan aplikasi pada tugas akhir ini. Use case dibagi menjadi 3 bagian seperti yang ditampilkan pada use case package di bawah ini.



Gambar C. 1 Use case package

C.1 Use Case Akses Aplikasi

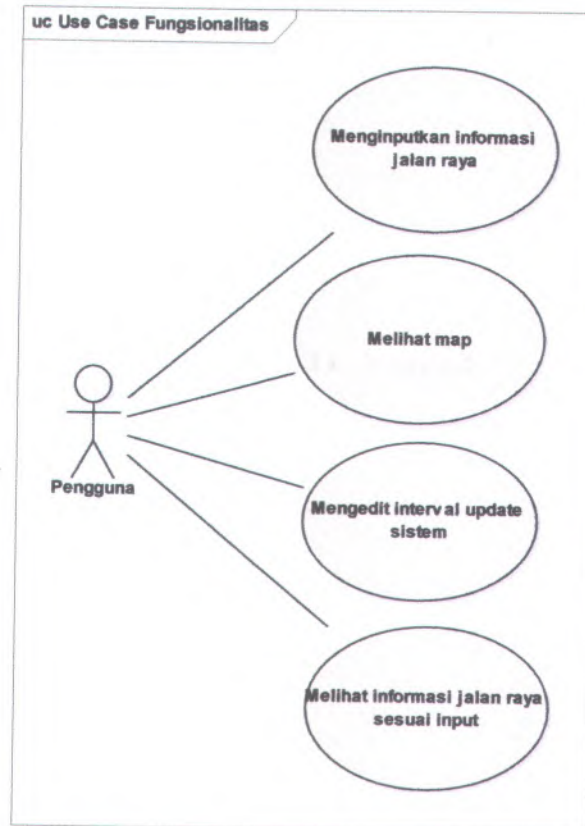
Use case akses aplikasi menampilkan rancangan pemenuhan kebutuhan dalam pengaksesan aplikasi.



Gambar C. 2 Use case akses aplikasi

C.2 Use Case Fungsionalitas

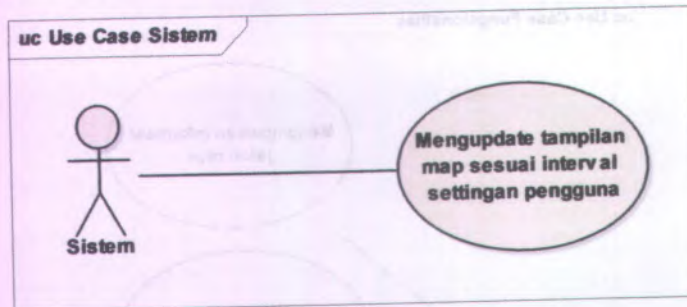
Use case fungsionalitas menampilkan rancangan pemenuhan kebutuhan pengguna atas fitur yang harus disediakan aplikasi untuk membangun sistem.



Gambar C. 3 Use case fungsionalitas

C.3 Use Case Sistem

Use case sistem melibatkan sistem sebagai aktor dalam hubungannya dengan kemampuan melakukan update atas tampilan map



Gambar C. 4 Use case Sistem

D.1 Use Case Akses Aplikasi

Pada bagian *use case* akses aplikasi terdapat 3 *use case*, yaitu *Pengguna melakukan login ke aplikasi*, *Pengguna melakukan pendaftaran*, dan *Pengguna melakukan proses unregistrasi*.

D.1.1 Pengguna melakukan login ke aplikasi

Tabel D. 1 Deskripsi Pengguna melakukan login ke aplikasi

Nama use case: <i>Pengguna melakukan login ke aplikasi</i>
Basic Flow: Pengguna menekan tombol login. Sistem melakukan pengecekan atas nomor IMEI. Jika IMEI telah terdaftar maka status pengguna akan diubah menjadi <i>online</i> dan dikirim ke server, dan jika IMEI belum terdaftar maka tidak akan ada data yang dikirim ke server.
Alternate Flow: Pengguna gagal melakukan login karena koneksi dengan server terputus. Notifikasi kegagalan hubungan dengan server ditampilkan ke pengguna

D.1.2 Pengguna melakukan pendaftaran

Tabel D. 2 Deskripsi Pengguna melakukan pendaftaran

Nama use case: <i>Pengguna melakukan pendaftaran</i>
Basic Flow: Pengguna menekan tombol daftar untuk kemudian diarahkan ke halaman pendaftaran. Pengguna mengisi field nama, dan nomor telepon kemudian menekan tombol daftar yang akan mengirimkan data tersebut ke server beserta data IMEI yang diperoleh oleh sistem secara otomatis. Pengguna mendapatkan kembalian berupa notifikasi proses pendaftaran telah berhasil
Alternate Flow: Pengguna tidak mengisi informasi yang diminta pada field. Muncul peringatan yang menampilkan bahwa ada <i>field</i> yang belum terisi. Pengguna akan diarahkan kembali ke halaman pendaftaran

D.1.3 Pengguna melakukan proses unregistrasi

Tabel D. 3 Deskripsi Pengguna melakukan proses unregistrasi

<p>Nama use case: Pengguna melakukan proses unregistrasi</p> <p>Basic Flow:</p> <p>Pengguna menekan tombol unregister. Jendela dialog ditampilkan untuk melakukan pengecekan kembali mengenai keputusan pengguna untuk melakukan proses unregistrasi. Pengguna menekan tombol OK pada jendela dialog. Data pengguna berupa IMEI akan dikirimkan kepada server. Server menghapus data pelanggan yang memiliki IMEI tersebut.</p> <p>Alternate Flow:</p> <p>-</p>
--

<p>Basic Flow:</p> <p>Pengguna menekan tombol unregister. Jendela dialog ditampilkan untuk melakukan pengecekan kembali mengenai keputusan pengguna untuk melakukan proses unregistrasi. Pengguna menekan tombol OK pada jendela dialog. Data pengguna berupa IMEI akan dikirimkan kepada server. Server menghapus data pelanggan yang memiliki IMEI tersebut.</p> <p>Alternate Flow:</p> <p>-</p>
--

<p>D.1.2 Pengguna melakukan proses unregistrasi</p> <p>Table D.1 Deskripsi Pengguna melakukan proses unregistrasi</p> <p>Basic Flow:</p> <p>Pengguna menekan tombol unregister. Jendela dialog ditampilkan untuk melakukan pengecekan kembali mengenai keputusan pengguna untuk melakukan proses unregistrasi. Pengguna menekan tombol OK pada jendela dialog. Data pengguna berupa IMEI akan dikirimkan kepada server. Server menghapus data pelanggan yang memiliki IMEI tersebut.</p> <p>Alternate Flow:</p> <p>-</p>
--

<p>D.1.3 Pengguna melakukan proses unregistrasi</p> <p>Table D.3 Deskripsi Pengguna melakukan proses unregistrasi</p>

D.2 Use Case Fungsionalitas

Pada *use case fungsionalitas* terdapat 4 macam *use case*, yaitu *use case Melihat informasi jalan raya sesuai input*, *Melihat map*, *Mengedit interval update sistem* dan *Menginputkan informasi jalan raya*.

D.2.1 Use case Melihat informasi jalan raya sesuai input

Tabel D. 4 Deskripsi use case Melihat informasi jalan raya sesuai input

Nama use case: <i>Melihat informasi jalan raya sesuai input</i>
Basic Flow: Pengguna menekan tombol Check Information. Pengguna memasukkan nama jalan yang akan dicek informasi jalan rayanya. Sistem mengirimkan nama jalan ke server yang kemudian akan diolah server untuk mengecek nama jalan yang memiliki kemiripan dengan masukan pengguna. Pengguna diarahkan ke halaman tampilan map yang menunjukkan jalan sesuai masukan pengguna sebagai titik pusat dari map beserta kondisinya.
Alternate Flow: <ul style="list-style-type: none"> • Pengguna tidak memasukkan nama jalan. Muncul notifikasi bahwa pengguna belum mendefinisikan nama jalan • Pengguna memasukkan nama jalan. Nama jalan diolah oleh server namun tidak ditemukan nama jalan yang memiliki kemiripan. Muncul notifikasi bahwa nama jalan tidak ditemukan

D.2.2 Use case Melihat map

Tabel D. 5 Deskripsi use case Melihat map

Nama use case: <i>Melihat map</i>
Basic Flow: Pengguna menekan tombol Lihat Map. Pengguna diarahkan kepada halaman tampilan map dengan posisi pengguna saat itu (<i>current position</i>) menjadi titik pusat dari peta
Alternate Flow: -

D.2.3 Use Case Mengedit interval update sistem

Tabel D. 6 Deskripsi use case Mengedit interval update sistem

Nama use case: Mengedit interval update sistem

Basic Flow:

Pengguna menekan tombol konfigurasi. Pengguna diarahkan ke halaman konfigurasi dengan pilihan untuk mengaktifkan *auto-update*. Pengguna memilih satu dari pilihan interval *waktu*. Aplikasi menyimpan settingan autoupdate pengguna.

Alternate Flow: -

D.2.4 Use case Menginputkan informasi jalan raya

Tabel D. 7 Deskripsi use case Menginputkan informasi jalan raya

Nama use case: Menginputkan informasi jalan raya

Basic Flow:

Pengguna menekan tombol Kirim Info. Pengguna diarahkan ke halaman tampilan map. Pengguna menekan dua titik sebagai tanda daerah yang mengalami perubahan kondisi jalan dan memilih tingkat kemacetan. Sistem mengkonversi dua titik tersebut ke dalam geocode. Data dari pengguna selanjutnya dikirimkan ke server untuk diolah.

Alternate Flow:

Pengguna tidak mengisi tingkat kemacetan. Muncul peringatan yang menyatakan bahwa tingkat kemacetan belum diisi oleh pelanggan.

D.3 Use Case Sistem

Pada fungsionalitas pengaturan terdapat satu *use case* *Mengupdate tampilan map sesuai interval settingan pengguna*.

D.3.1 Use case Mengupdate tampilan map sesuai interval settingan pengguna

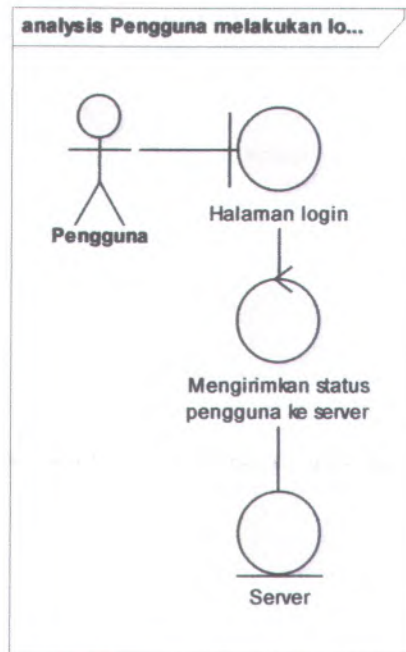
Tabel D. 8 Deskripsi Mengupdate tampilan map sesuai interval settingan pengguna

Nama <i>use case</i> : <i>Mengupdate tampilan map sesuai interval settingan pengguna</i>
Basic Flow: Sistem mengambil nilai konfigurasi pengguna. Sistem melakukan cek terhadap nilai konfigurasi pengguna. Sistem melakukan pembaharuan tampilan map setiap jumlah angka pada settingan pengguna dalam satuan menit.
Alternate Flow: -

E.1 Use Case Akses Aplikasi

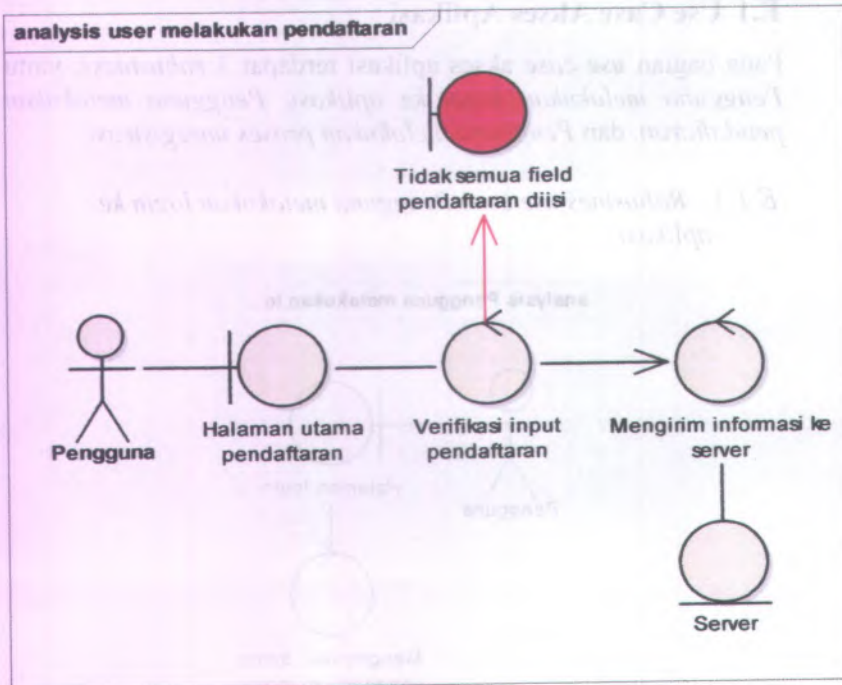
Pada bagian *use case* akses aplikasi terdapat 3 *robustness*, yaitu *Pengguna melakukan login ke aplikasi*, *Pengguna melakukan pendaftaran*, dan *Pengguna melakukan proses unregistrasi*.

E.1.1 Robustnes use case Pengguna melakukan login ke aplikasi



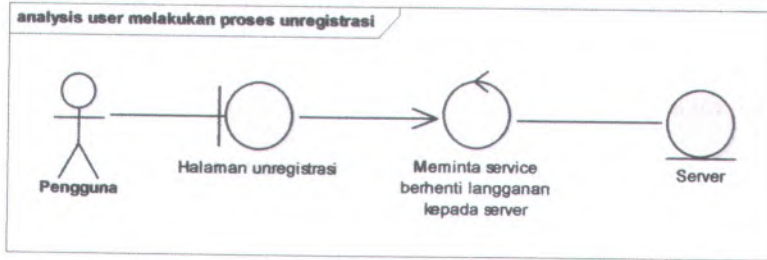
Gambar E. 1 Robustnes use case Pengguna melakukan login ke aplikasi

E.1.2 Robustnes use case Pengguna melakukan pendaftaran



Gambar E. 2 Robustnes use case Pengguna melakukan pendaftaran

E.1.3 Robustnes use case Pengguna melakukan proses unregistrasi

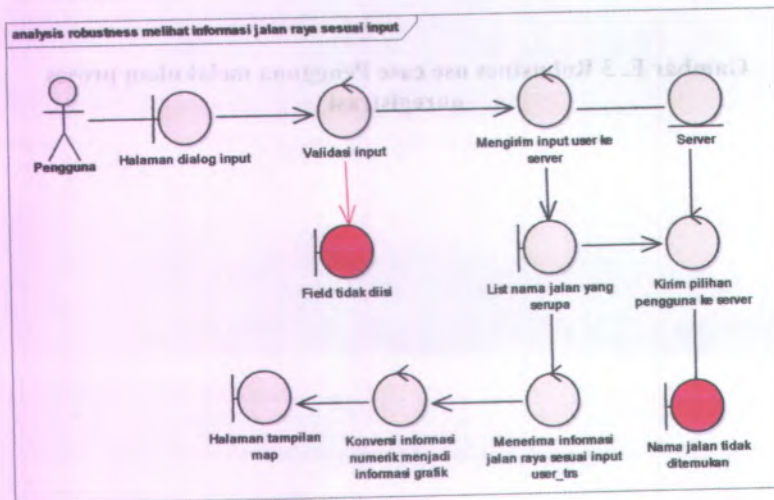


Gambar E. 3 Robustnes use case Pengguna melakukan proses unregistrasi

E.2 Use Case Fungsionalitas

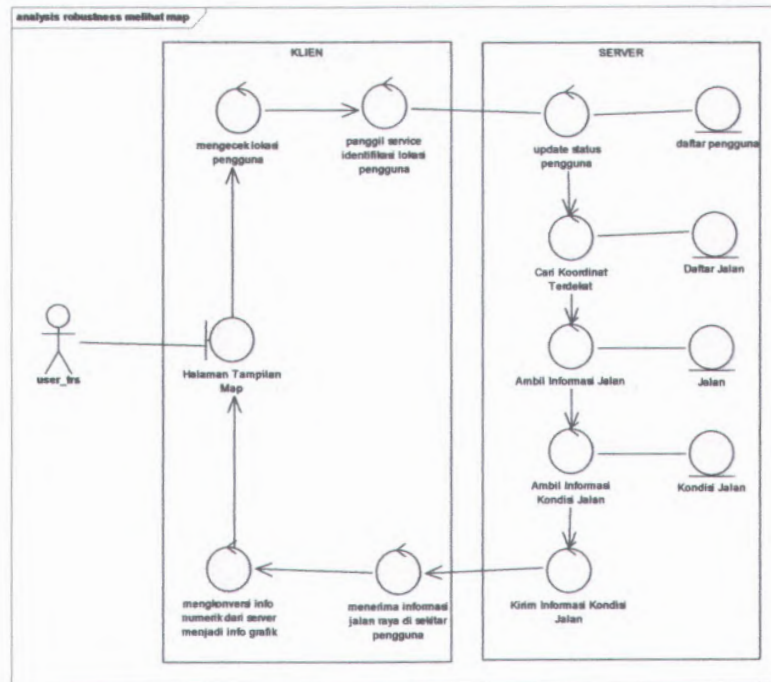
Pada *use case* fungsionalitas terdapat 4 macam *use case*, yaitu *use case* Melihat informasi jalan raya sesuai input, Melihat map Mengedit interval update sistem, dan Menginputkan informasi jalan raya.

E.2.1 Robustnes use case Melihat informasi jalan raya sesuai input



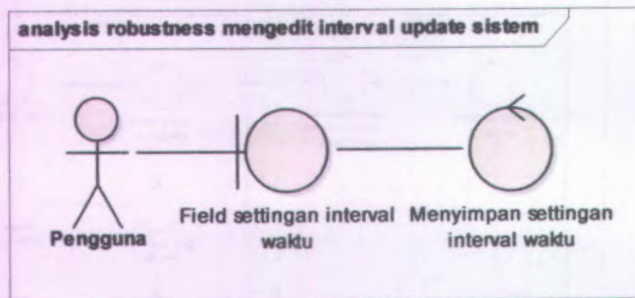
Gambar E. 4 Robustnes use case Melihat informasi jalan raya sesuai input

E.2.2 Robustnes use case Melihat map



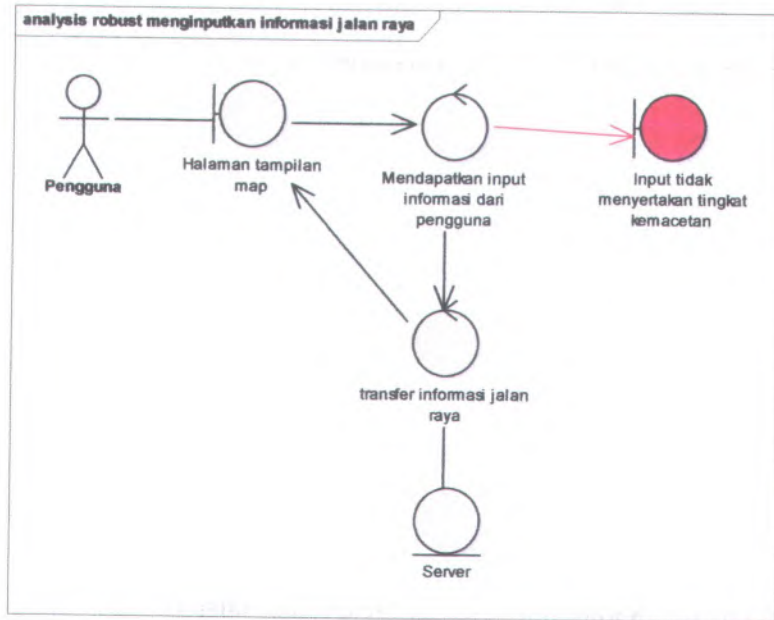
Gambar E. 5 Robustnes use case Melihat map

E.2.3 Robustnes use case Mengedit interval update sistem



Gambar E. 6 Robustnes use case Mengedit interval update sistem

E.2.4 Robustnes use case Menginputkan informasi jalan raya

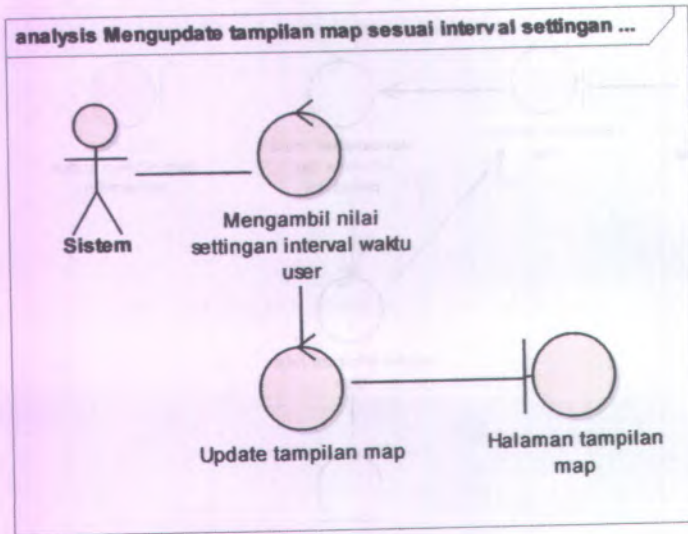


Gambar E. 7 Robustnes use case Menginputkan informasi jalan raya

E.3 Use Case Sistem

Pada *use case* sistem terdapat satu *use case* Mengupdate tampilan map sesuai interval settingan pengguna.

E.3.1 Robustnes use case Mengupdate tampilan map sesuai interval settingan pengguna

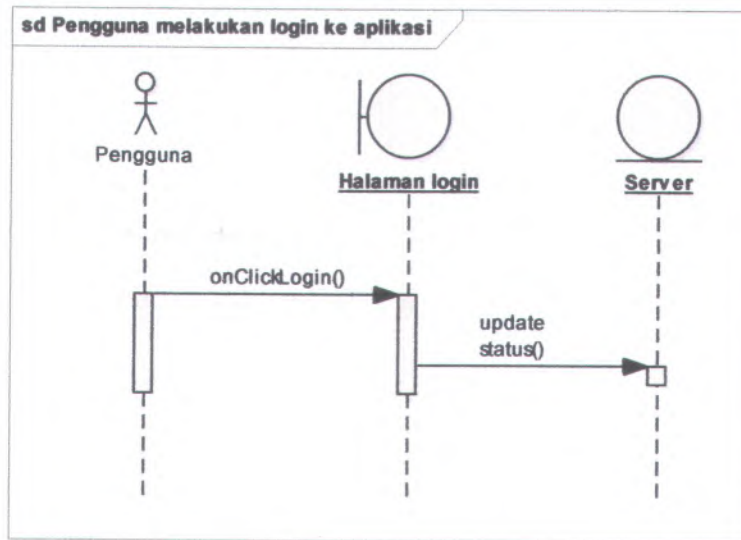


Gambar E. 8 Robustnes use case Mengupdate tampilan map sesuai interval settingan pengguna

F.1 Use Case Akses Aplikasi

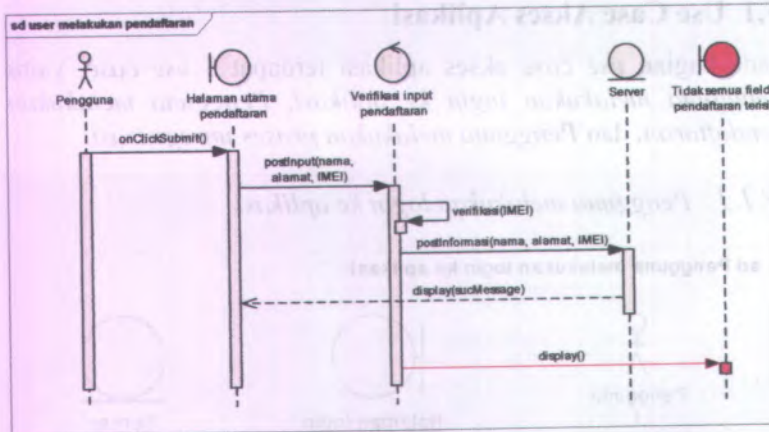
Pada bagian *use case* akses aplikasi terdapat 3 *use case*, yaitu *Pengguna melakukan login ke aplikasi*, *Pengguna melakukan pendaftaran*, dan *Pengguna melakukan proses unregistrasi*.

F.1.1 Pengguna melakukan login ke aplikasi



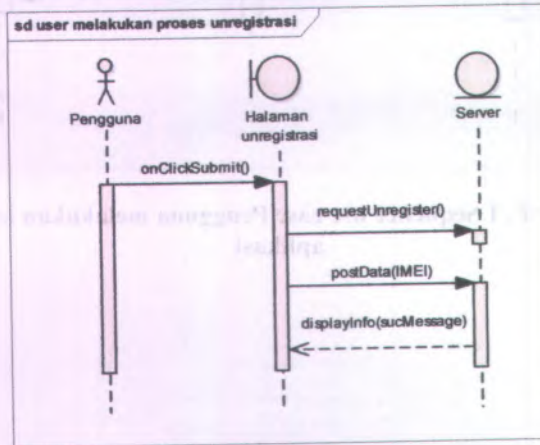
Gambar F. 1 Sequence use case Pengguna melakukan login ke aplikasi

F.1.2 Sequence use case Pengguna melakukan pendaftaran



Gambar F. 2 Sequence use case Pengguna melakukan pendaftaran

F.1.3 Sequence use case Pengguna melakukan proses unregistrasi

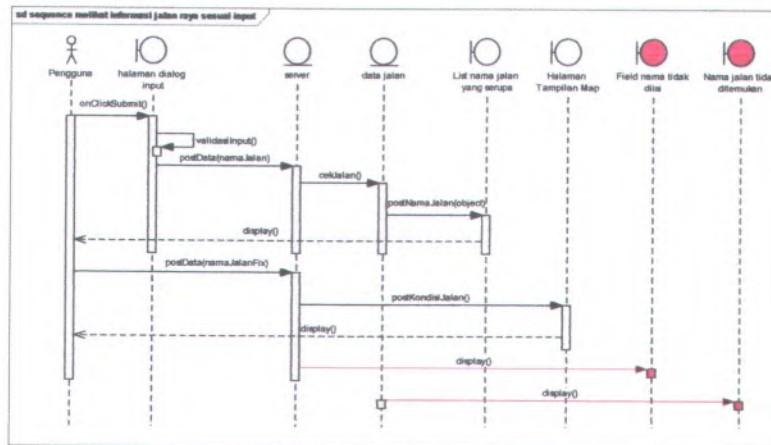


Gambar F. 3 Sequence use case Pengguna melakukan proses unregistrasi

F.2 Use Case Fungsionalitas

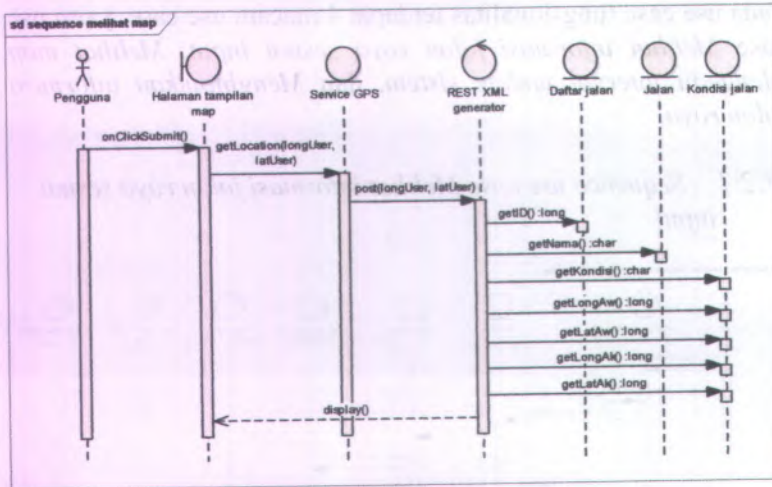
Pada use case fungsionalitas terdapat 4 macam use case, yaitu use case *Melihat informasi jalan raya sesuai input*, *Melihat map Mengedit interval update sistem*, dan *Menginputkan informasi jalan raya*.

F.2.1 Sequence use case *Melihat informasi jalan raya sesuai input*



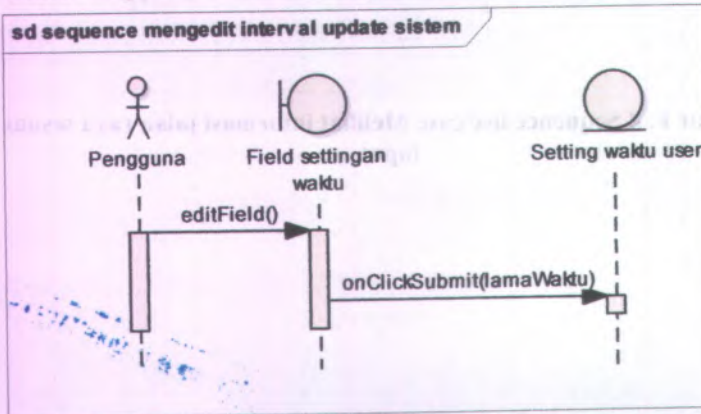
Gambar F. 4 Sequence use case *Melihat informasi jalan raya sesuai input*

F.2.2 Sequence use case Melihat map



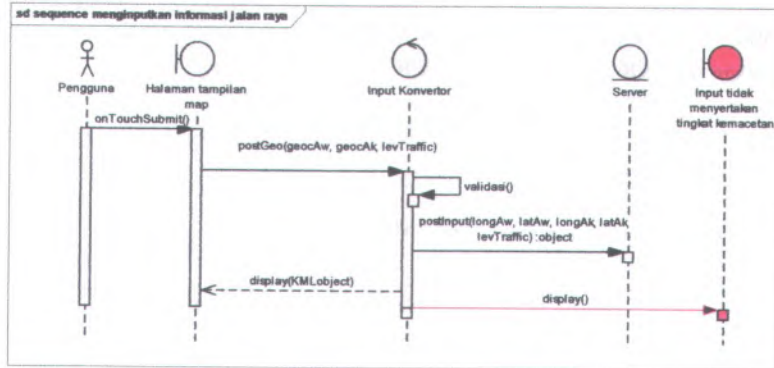
Gambar F. 5 Sequence use case Melihat map

F.2.3 Sequence use case Mengedit interval update sistem



Gambar F. 6 Sequence use case Mengedit interval update sistem

F.2.4 Sequence use case Menginputkan informasi jalan raya



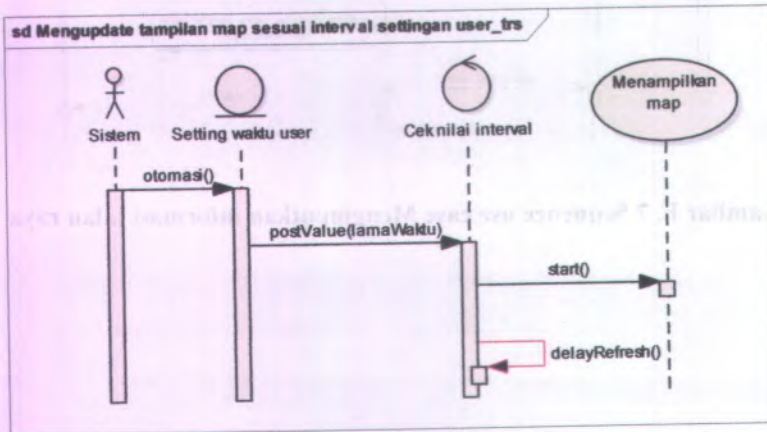
Gambar F. 7 Sequence use case Menginputkan informasi jalan raya



F.3 Use Case Sistem

Pada use case pengaturan terdapat satu *Mengupdate tampilan map sesuai interval settingan pengguna*.

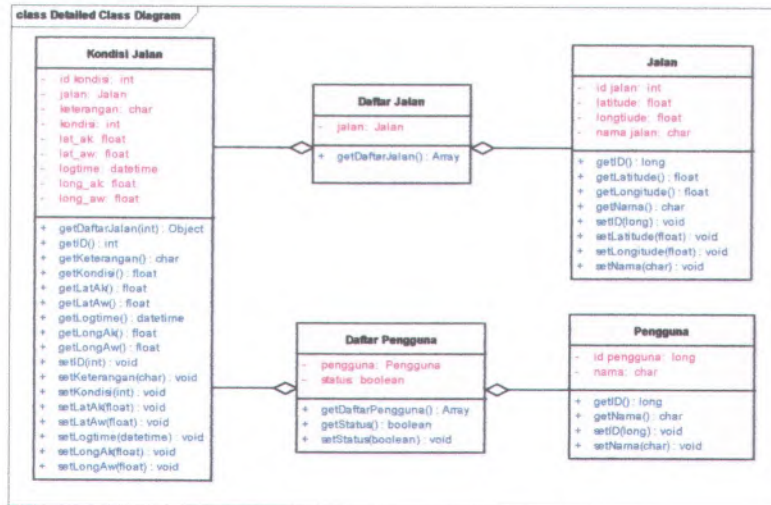
F.1.1 Mengupdate tampilan map sesuai interval settingan pengguna



Gambar F. 8 Sequence use case Mengupdate tampilan map sesuai interval settingan pengguna



G.1 Class Diagram



Gambar G. 1 Class diagram

H.1 Skenario Use Case Pengguna melakukan login ke aplikasi

No	Skenario	Starting Flow	Alternate
1	Menampilkan Halaman Pembuka	Basic Flow	
2	Pengguna Login	Basic Flow	
3	Menampilkan halaman utama	Basic Flow	

H.2 Skenario Test Case Pengguna melakukan login ke aplikasi

No	Skenario	Retrieved Halaman Utama	Retrieved Validasi Login	Hasil
1	Menampilkan Halaman Pembuka	N/A	N/A	Halaman pembuka berhasil ditampilkan
2	Pengguna Login	N/A	Yes	Validasi dilakukan dengan memasukkan IMEI yang sudah ada dan belum ada dalam database server. Notifikasi berhasil ditampilkan
3	Menampilkan halaman utama	Yes	N/A	Halaman Utama berhasil ditampilkan

H.3 Skenario Use Case Pengguna melakukan pendaftaran

No	Skenario	Starting Flow	Alternate
1	Menampilkan Halaman Pembuka	Basic Flow	
2	Pengguna Memilih Menu Register	Basic Flow	
3	Pengguna Pendaftaran Berhasil	Basic Flow	
4	Pengguna Pendaftaran Gagal	Basic Flow	A1

H.4 Skenario Test Case Pengguna melakukan pendaftaran

No	Skenario	<i>Retrieved Menu Register</i>	<i>Success Notification</i>	<i>Failure Notification</i>	<i>Hasil</i>
1	Menampilkan Halaman Pembuka	N/A	N/A	N/A	Halaman pembuka berhasil ditampilkan
2	Pengguna Memilih Menu Register	Yes	N/A	N/A	Halaman register berhasil ditampilkan
3	Pengguna Pendaftaran Berhasil	Yes	Yes	N/A	Notifikasi berhasil ditampilkan
4	Pengguna Pendaftaran Gagal	Yes	N/A	Yes	Notifikasi berhasil ditampilkan

H.5 Skenario Use Case Pengguna melakukan proses unregistrasi

No	Skenario	Starting Flow	Alternate
1	Menampilkan Halaman Unregistrasi	Basic Flow	
2	Pengguna Memilih Menu Unregister	Basic Flow	
3	Pengguna Unregistrasi Berhasil	Basic Flow	

H.6 Skenario Test Case Pengguna melakukan proses unregistrasi

No	Skenario	Retrieved Halaman Utama	Retrieved Validasi Login	Hasil
1	Menampilkan Halaman Unregistrasi	N/A	N/A	Halaman pembuka berhasil ditampilkan
2	Pengguna Memilih Menu Unregister	N/A	Yes	Validasi dilakukan dengan memasukkan IMEI yang sudah ada dan belum ada dalam database server. Notifikasi berhasil ditampilkan

No	Skenario	Retrieved Halaman Utama	Retrieved Validasi Login	Hasil
3	Pengguna Unregistrasi Berhasil	Yes	N/A	Halaman Utama berhasil ditampilkan

H.7 Skenario Use Case Melihat map

No	Skenario	Starting Flow	Alternate
1	Menampilkan Halaman Tampilan Map dan menampilkan informasi lalu lintas di sekitar pengguna	Basic Flow	
2	Menampilkan informasi lalu lintas di sekitar pengguna	Basic Flow	

H.8 Skenario Test Case Melihat map

No	Skenario	Retrieved Informasi	Hasil
1	Menampilkan Halaman Tampilan Map	N/A	Halaman tampilan map dapat ditampilkan
2	Menampilkan informasi lalu lintas di sekitar pengguna	N/A	Informasi dapat ditampilkan meskipun belum memberikan data dengan presisi yang tinggi

H.9 Skenario Use Case Melihat map sesuai input pengguna

No	Skenario	Starting Flow	Alternate
1	Menampilkan Halaman Cari Kondisi Jalan	Basic Flow	
2	Pencarian Berhasil	Basic Flow	
3	Pencarian Gagal	Basic Flow	A1

H.10 Skenario Test Case Melihat map sesuai input pengguna

No	Skenario	Retrieved Hasil Pencarian	Hasil
1	Menampilkan Halaman Cari Kondisi Jalan	Basic Flow	Halaman cari kondisi jalan berhasil ditampilkan
2	Pencarian Berhasil	Basic Flow	Pengguna memasukkan masukan yang benar dan memperoleh hasil pencarian
3	Pencarian Gagal	Basic Flow	Pengguna memasukkan masukan yang tidak valid hingga mengakibatkan pencarian gagal

H.11 Skenario Use Case Mengirim informasi

No	Skenario	Starting Flow	Alternate
1	Menampilkan Halaman Tampilan Map	Basic Flow	
2	Pengguna Memasukkan Informasi Jalan dan Tingkat Kemacetan	Basic Flow	
3	Pengiriman Informasi Berhasil	Basic Flow	

H.12 Skenario Test Case Mengirim informasi

No	Skenario	Error Message	Hasil
1	Menampilkan Halaman Tampilan Map	N/A	Map telah berhasil ditampilkan
2	Pengguna Memasukkan Informasi Jalan dan Tingkat Kemacetan	N/A	Proses penentuan data <i>geocode</i> telah berhasil dilakukan
3	Pengiriman Informasi Berhasil	N/A	Informasi yang dimasukkan pengguna dapat dikirimkan ke server

H.13 Skenario Use Case Mengedit interval update sistem

No	Skenario	Starting Flow	Alternate
1	Menampilkan Halaman Setting	Basic Flow	
2	Pengguna Memilih Interval Update	Basic Flow	
3	Interval Update Disimpan	Basic Flow	

H.14 Skenario Test Case Mengedit interval update sistem

No	Skenario	Interval Saved	Hasil
1	Menampilkan Halaman Setting	N/A	Halaman setting berhasil ditampilkan
2	Pengguna Memilih Interval Update	N/A	Tampilan interval waktu dan dapat dipilih oleh pengguna
3	Interval Update Disimpan	Yes	Interval update telah tersimpan dalam <i>variable</i> yang dapat dikirimkan ke aktifitas lain

H.15 Skenario Use Case Mengupdate tampilan map sesuai interval setingan pengguna

No	Skenario	Starting Flow	Alternate
1	Melakukan Refresh sesuai Interval Pengguna	Basic Flow	

H.16 Skenario Test Case Mengupdate tampilan map sesuai interval setingan pengguna

No	Skenario	Retrieved Refresh	Hasil
1	Melakukan Refresh sesuai Interval Pengguna	<i>Yes</i>	Sistem berhasil melakukan auto update sesuai dengan interval waktu yang dimasukkan pengguna

BIODATA PENULIS



Penulis dilahirkan di Jakarta pada tanggal 20 Desember 1987. Merupakan anak ketiga dari lima bersaudara. Menempuh pendidikan formal di SD Pamardi Yuwana Bhakti, SLTP Pamardi Yuwana Bhakti Bekasi, dan SMA 64. Melanjutkan jenjang perkuliahan di Jurusan Sistem Informasi Institut Teknologi Sepuluh Nopember angkatan 2006 dan menyandang NRP 5206100100. Penulis memilih *E-business* sebagai bidang minat yang digunakan untuk

menyelesaikan kewajiban tugas akhir. Penulis aktif mengikuti kegiatan kemahasiswaan dengan tergabung di Keluarga Mahasiswa Sistem Informasi (KMSI) dan tercatat sebagai anggota Departemen Kewirausahaan untuk tahun pertama, dan ketua Departemen Kewirausahaan pada tahun kedua.