

RANCANG BANGUN *MOBILE PHONE POSITIONING* SYSTEM PADA PLATFORM ANDROID

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Pada
Jurusan Teknik Informatika

Oleh :

ERZI HIDAYAT
NIM : 10851002900



FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2013

LEMBAR PENGESAHAN
RANCANG BANGUN MOBILE PHONE POSITIONING SYTEM
PADA PLATFORM ANDROID

TUGAS AKHIR

Oleh :

ERZI HIDAYAT
10851002900

Telah dipertahankan di depan sidang dewan penguji
Sebagai salah satu syarat untuk memperoleh gelar sarjana Teknik Informatika
Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau
Di Pekanbaru, pada tanggal, 11 Juli 2013

Pekanbaru, 11 Juli 2013

Mengesahkan,

Dekan



Dra. Hj. Yenita Morena, M.Si
NIP. 19601125 198503 2 002

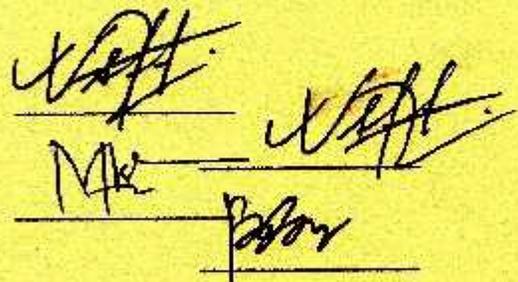
Ketua Jurusan



Dr. Okfalisa, ST, M.Sc
NIP. 19771028 200312 2 004

DEWAN PENGUJI

Ketua : Nazruddin Safaat H, M.T.
Sekretaris : Nazruddin Safaat H, M.T.
Anggota I : M. Irsyad, M.T.
Anggota II : Benny Sukma Negara, M.T.



APPLICATION DESIGN MOBILE PHONE POSITIONING SYSTEM ON ANDROID PLATFORM

ERZI HIDAYAT
10851002900

Final Exam Date: 11 July 2013

Graduation Ceremony Period: November 2013

*Information Engineering Department
Faculty of Sciences and Technology
State Islamic University of Sultan Syarif Kasim Riau*

ABSTRACT

Positioning of the presence of moving objects by using SMS (Short Message Service) and voice call are the ways which can be carried out to find the presence of mobile devices, however such location determination is not effective if the user does not give a respond, especially when the position is needed periodically. Mobile Positioning application is designed by utilizing the LBS (Location Based Service) technology Android platform-based, the IMEI (International Mobile Equipment Identity) code of the cellphone is used as user's unique code which differs between one device and another. The system consists of two applications which intergrated between position sender and position controller applications. Position data is sent by position sender application periodically on server online for then be caught by position controller application which shown in Google maps. The examination is carried out to two separate smartphone devices, the constructed examination results of the system show that the recipient application successfully show the location of position of sender application.

Key words: Android, Mobile Positioning, LBS, Google maps, samartphone, IMEI.

RANCANG BANGUN *MOBILE PHONE POSITIONING* SYSTEM PADA PLATFORM ANDROID

ERZI HIDAYAT
10851002900

Tanggal Sidang: 11 Juli 2013
Periode Wisuda: November 2013

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau

ABSTRAK

Penentuan posisi keberadaan benda bergerak dengan menggunakan fasilitas SMS (*Short Message Service*) dan *voice call* merupakan cara yang dapat dilakukan untuk mengetahui keberadaan perangkat ponsel, tetapi penentuan lokasi tersebut tidak efektif jika pengguna tidak memberikan respon, terutama jika posisi tersebut dibutuhkan secara berkala. Aplikasi *Mobile Positioning* dirancang dengan memanfaatkan teknologi *LBS (Location Based Service)* berbasis platform *Android*, kode *IMEI (International Mobile Equipment Identity)* handphone digunakan sebagai kode unik pengguna yang membedakan antara perangkat satu dengan lainnya. Sistem terdiri dari dua aplikasi yang saling terintegrasi antara aplikasi pengirim dan pengamat posisi. Data posisi dikirimkan oleh aplikasi pengirim posisi secara berkala pada *server online* untuk kemudian ditangkap oleh aplikasi pemantau posisi yang ditampilkan pada *Google maps*. Pengujian dilakukan terhadap dua perangkat *smartphone* yang terpisah, hasil pengujian sistem yang dibangun menunjukkan aplikasi penerima telah berhasil menampilkan posisi lokasi dari aplikasi pengirim.

Kata kunci: Android, Mobile Positioning, LBS, Google maps, smartphone, IMEI.

KATA PENGANTAR

Alhamdulillah Robbil'alamin, penulis ucapkan syukur yang setinggi-tinggi ke-hadirat Allah SWT, karena atas segala limpahan rahmat dan karuniahnya yang diberikan sehingga penulis dapat menyelesaikan penelitian sekaligus penulisan laporan tugas akhir ini. *Allahumma sholli'ala Muhammad wa'ala ali sayyidina Muhammad*, yang tidak lupa penulis haturkan juga untuk junjungan alam, kekasih Allah, Rasul Allah, dan tauladan kita yakni Nabi Muhammad SAW.

Laporan tugas akhir ini merupakan salah satu prasyarat untuk memenuhi persyaratan akademis dalam rangka meraih gelar kesarjanaan di Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sultan Syarif Kasim Riau (UIN SUSKA Riau). Selama menyelesaikan tugas akhir ini, penulis telah banyak mendapatkan bantuan, bimbingan, dan petunjuk dari banyak pihak baik secara langsung maupun tidak langsung. Dalam kesempatan ini penulis ingin mengucapkan terimakasih yang sebesar-besarnya kepada:

1. Prof. Dr. H. M. Nazir, selaku Rektor Universitas Islam Negeri Sultan Syarif Kasim Riau.
2. Dra. Yenita Morena, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau.
3. Novriyanto ST, M.sc, selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau.
4. Risky Maycandra, ST, M.sc, selaku Kordinator Tugas Akhir.
5. Nazruddin Safaat H. ST, MT selaku pembimbing Tugas Akhir, terimakasih atas petunjuk, arahan serta waktu yang telah diberikan kepada penulis utuk dapat menyelesaikan Tugas Akhir.
6. Bapak M. Irsyad, ST, MT, selaku dosen penguji 1, terimakasih atas ilmu-ilmunya, saran-sarannya, perbaikan-perbaikannya, dan masukan yang Bapak berikan untuk penyempurnaan laporan ini.

7. Bapak Benny Sukma Negara, ST, MT, selaku dosen penguji 2, terimakasih juga untuk ilmu-ilmunya, saran-sarannya, perbaikan-perbaikannya, dan masukan yang Bapak berikan untuk penyempurnaan laporan ini.
8. Dan terakhir, terimakasih pula penulis ucapkan untuk Almamater Teknik Informatika, Fakultas Sains dan Teknologi, UIN Sultan Syarif Kasim Riau serta pihak-pihak lain yang tidak dapat penulis sebutkan satu persatu. Terimakasih banyak atas bantuan dan dukungannya yang berharga.
Akhirnya, penulis menyadari dalam penulisan laporan ini masih terdapat kekurangan. Oleh karena itu, saran dan kritik sangat penulis harapkan untuk kemajuan penulis secara pribadi. Terimakasih.

Pekanbaru, Mei 2013

Penulis

DAFTAR ISI

HALAMAN JUDUL LAPORAN	i
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL.....	iv
LEMBAR PERNYATAAN	v
LEMBAR PERSEMBAHAN	vi
ABSTRACT	vii
ABSTRAK	viii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiv
DAFTAR LAMPIRAN.....	xvii
DAFTAR TABEL	xviii
DAFTAR SIMBOL	xix
DAFTAR ISTILAH	xx
BAB I PENDAHULUAN.....	I-1
1.1. Latar Belakang	I-1
1.2. Rumusan Masalah	I-2
1.3. Batasan Masalah.....	I-3
1.4. Tujuan Penelitian	I-3
1.5. Manfaat Penelitian	I-3
1.6. Sistematika Penelitian	I-4
BAB II LANDASAN TEORI	II-1
2.1. Teknologi <i>Mobile</i>	II-1
2.1.1. <i>Wireles Mobile Network</i>	II-1
2.1.2. Android.....	II-4
2.1.2.1. Pengertian android	II-4
2.1.2.2. Anatomi Android	II-5

2.1.2.3. Kelebihan Android.....	II-7
2.2. GPS.....	II-9
2.2.1. Metode Advanced Position	II-10
2.2.2. Contoh Penerapan GPS pada smartphone android.....	II-10
2.3.3. Contoh Penerapan GPS pada makhluk hidup	II-11
2.3. Pemetaan (<i>Google Map</i>).....	II-11
2.4. Metodologi Penelitian	II-12
2.4.1. <i>Waterfall</i>	II-12
2.4.2. <i>Unified Rational Unified Process (RUP)</i>	II-14
2.4.2.1. Pengertian RUP	II-14
2.4.2.2. Fase RUP	II-16
2.5. <i>Unified Modelling Language (UML)</i>	II-17
2.5.1. <i>Usecase Diagram</i>	II-18
2.5.2. <i>Sequance Diagram</i>	II-19
2.5.3. <i>Class Diagram</i>	II-20
2.5.4. <i>Activity Diagram</i>	II-21
2.6. <i>Location Based Services (LBS)</i>	II-22
2.6.1. Komponen LBS.....	II-22
2.7. Contoh Aplikasi penelitian mengenai LBS.....	II-24
BAB III METODOLOGI PENELITIAN	III-1
3.1. Tahapan <i>Waterfall</i>	III-1
3.1.1. Alur Tahapan <i>Waterfall</i>	III-1
3.2.2. Defenisi Kebutuhan Sistem.....	III-1
3.2.2. Analisa dan Perancangan	III-3
3.2.3. Pembuatan Aplikasi	III-3
3.2.4. Testing dan Pengujian Aplikasi.....	III-3
BAB IV ANALISA DAN PERANCANGAN.....	IV-1
4.1. Analisa.....	IV-1
4.1.1. Defenisi Kebutuhan Sistem.....	IV-2
4.1.1.1. Project Plan	IV-3
A. Gambaran Umum Sistem	IV-3

B.Deskripsi Kebutuhan Sistem	IV-5
C.Fungsi Sistem	IV-8
4.2. Perancangan Sistem	IV-8
4.2.1. Pemodelan UML (<i>Unified Modeling Language</i>)	IV-8
4.2.1.1. <i>Usecase Diagram</i>	IV-9
4.2.1.2. <i>Class Diagram</i>	IV-13
4.2.1.3. <i>Activity Diagram</i>	IV-18
4.2.1.4. <i>Sequence Diagram</i>	IV-19
4.3. Rancangan Struktur Menu Sistem.....	IV-21
4.4. Perancangan Antarmuka Pengguna Sistem.....	IV-21
BAB V IMPLEMENTASI DAN PENGUJIAN.....	V-1
5.1. Implementasi	V-1
5.1.1. Batasan Implementasi	V-1
5.1.2. Lingkungan Implementasi.....	V-2
5.1.3. Tahap-tahap Implementasi aplikasi Mobile position	V-2
A. Instalasi Aplikasi A	V-2
B. Instalasi Aplikasi User B	V-4
5.2. Pengujian Aplikasi	V-4
5.2.1. Pengujian <i>Blackbox</i> Aplikasi <i>mobile position user A</i>	V-4
5.2.2. Pengujian <i>Blackbox</i> Aplikasi <i>mobile position user B</i>	V-5
5.2.3. Pengujian Akses Aplikasi <i>Mobile Position</i>	V-8
5.2.4. Kesimpulan Pengujian	V-10
BAB VI PENUTUP	VI-1
6.1. Kesimpulan.....	VI-1
6.2. Saran.....	VI-1
DAFTAR PUSTAKA	xxi
LAMPIRAN	
RIWAYAT HIDUP	

BAB I

PENDAHULUAN

1.1. Latar Belakang

Penentuan posisi keberadaan suatu objek bergerak seringkali menjadi suatu permasalahan tersendiri, hal ini karena keberadaan objek selalu berubah dari suatu tempat ke tempat lainnya, mengetahui informasi posisi pada manusia menjadi suatu hal yang bisa dilakukan, dengan ditemukannya berbagai kemajuan teknologi saat sekarang ini, seperti penggunaan telepon genggam dengan mengirimkan SMS (*Short Message Service*) dan dengan menggunakan layanan *voice call*, akan tetapi kedua layanan tersebut tidak dapat dijadikan acuan utama karena bisa saja nomor tujuan memberikan informasi yang salah terhadap keberadaan posisinya (berbohong), hal tersebut juga akan sangat menyulitkan apabila informasi posisi tersebut dibutuhkan secara berkala (*periodic*).

LBS (*Location-Based Services*) merupakan metode yang dapat digunakan untuk membuat aplikasi penentuan posisi, gagasan ini muncul karena adanya perkembangan teknologi *mobile* yang menggabungkan berbagai komponen teknologi seperti *Global Positioning System* (GPS), *Geographic Information System* (GIS) dan teknologi *wireless*. Pada saat sekarang ini hampir semua perangkat ponsel cerdas (*smartphones*) sudah terintegrasi dengan GPS sehingga sangat memungkinkan untuk mengetahui posisi keberadaan perangkat *mobile* lain.

Teknologi selular yang berkembang sangat pesat mengakibatkan ponsel berevolusi menjadi perangkat pintar yang dapat menjalankan berbagai macam aplikasi (*multi tasking*) serta mempunyai sistem operasi tersendiri seperti sebuah komputer, sistem operasi ponsel cerdas yang sedang berkembang pesat belakangan ini yaitu Android. Menurut catatan *International Data Corporation* (IDC) sampai dengan 8 Agustus 2012, pangsa pasar Android yang didukung oleh Google telah mencapai 68,1 % dari pangsa pasar *smartphone* di seluruh dunia (idc.com).

Studi dan penelitian terkait penentuan lokasi dengan penggunaan GPS pada sistem operasi Android mulai marak dikembangkan oleh peneliti untuk menemukan lokasi, objek wisata dan juga tempat-tempat penting lainnya seperti lokasi ATM, hotel atau rumah makan yang ditampilkan pada peta digital. Dalam jurnal-jurnal juga banyak dipublikasikan penelitian layanan berbasis lokasi seperti *Implementation of Location Based Services in Android using GPS and Web Services* oleh Manav Singhal dan Anupam Shukla tahun 2012.

Diantara aplikasi yang ada di *Google play store* seperti aplikasi “Abang kat mana?”. Aplikasi ini juga menggunakan layanan berbasis lokasi, cara kerja aplikasi yaitu dengan mengirimkan SMS dengan format tertentu (“akm”) ke *handpone user* yang ingin diketahui, kemudian *user* akan menerima SMS balasan berupa *link Maps handphone* orang yang dituju. Permasalahan aplikasi penentuan posisi berbasis lokasi selama ini yaitu hanya mengirimkan lokasi pengguna pada saat dikirimkan SMS dan menunggu balasan dari *mobile* tujuan, kemudian ditampilkan pada saat merujuk kepada *link* yang diberikan oleh nomor tujuan. Belum memiliki layanan *track history* serta *auto sending* data GPS yang dapat memperlihatkan arah pergerakan *mobile* dari waktu ke waktu.

Seiring perkembangan *smartphone* serta penelitian terkait GPS dengan penggunaan metode LBS maka penulis tertarik untuk membangun suatu aplikasi yang memberikan informasi posisi keberadaan *smartphone* dengan pemanfaatan teknologi GPS berbasis Android, sehingga dapat mempermudah pengguna untuk menemukan keberadaan seseorang pengguna *smartphone* Android yang memiliki fasilitas *track history*, posisi terakhir dan autentifikasi pengguna. Berdasarkan latar belakang diatas maka tugas akhir ini diberi judul ” RANCANG BANGUN *MOBILE PHONE POSITIONING SISTEM PADA PLATFORM ANDROID*”.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah diatas maka dapat dirumuskan bagaimana membangun aplikasi *mobile phone positioning* dengan menggunakan GPS pada *mobile device* berbasis Android untuk memberikan informasi keberadaan posisi *mobile* dengan menggunakan metode LBS.

1.3. Batasan Masalah

Untuk memberikan gambaran secara jelas dan menghasilkan hasil yang optimal dalam penulisan tugas akhir ini, maka akan diberikan batasan-batasan masalah. Batasan masalah dalam penelitian tugas akhir ini antara lain sebagai berikut:

1. Tampilan posisi koordinat GPS ditampilkan pada *maps smartphone* bukan web.
2. Data yang diterima dari *user* terdiri dari *latitude* dan *longitude*.
3. Menggunakan *Goggle maps* API sebagai sumber *maps*.
4. Tidak membahas analisa perancangan dari sisi *server*.

1.4. Tujuan Penelitian

Tujuan yang ingin dicapai penulis dari penelitian Tugas Akhir ini adalah membangun sebuah aplikasi yang mampu memberikan informasi posisi keberadaan perangkat *handphone* dengan *platform Android*, yang mampu menunjukkan rute pergerakan perangkat Android secara berkala, mampu memberikan jarak tempuh ke tempat tujuan pengguna yang diamati, mampu menunjukkan rute tempuh ke tempat pengguna yang diamati serta direpresentasikan pada *Google Maps*.

1.5. Manfaat Penelitian

Beberapa manfaat dari penelitian tentang aplikasi *Mobile Positionig System* ini adalah:

- a. Dapat memberikan informasi mengenai *tracking route* pengguna perangkat Android.
- b. Memberikan kemudahan dalam akses informasi pengguna untuk mengetahui keberadaan posisi, informasi dapat berupa: Rute jalur yang dilewati, jarak tempuh ke lokasi
- c. Membantu pengguna untuk melakukan pemantauan keberadaan pengguna perangkat Android.

1.6. Sistematika Penulisan

Berikut merupakan rencana susunan sistematika penulisan laporan Tugas Akhir yang akan dibuat:

Bab I Pendahuluan

Bab ini berisi penjelasan mengenai latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan sistematika penulisan dari Tugas Akhir yang dibuat.

Bab II Landasan Teori

Bab ini membahas tentang teori-teori umum dan khusus yang berhubungan dengan tugas akhir ini.

Bab III Metodologi Penelitian

Bab ini membahas langkah-langkah yang dilaksanakan dalam proses penelitian, yaitu pengamatan pendahuluan, tahapan identifikasi masalah, perumusan masalah, analisa aplikasi, perancangan aplikasi dan implementasi aplikasi beserta pengujian.

Bab IV Analisa dan Perancangan

Bab ini berisi pembahasan mengenai kebutuhan sistem, yang terdiri dari : Analisa kebutuhan sistem, *Unified Modeling Language* (UML), Struktur menu, *User interface*, Perancangan menggunakan pendekatan berorientasi objek.

Bab V Implementasi dan Pengujian

Bab ini berisi penjelasan mengenai implementasi dan pengujian dari aplikasi yang terdiri dari: batasan implementasi, lingkungan implementasi, hasil implementasi, pengujian aplikasi dan kesimpulan dari pengujian.

Bab VI Penutup

Bagian ini berisi kesimpulan yang dihasilkan dari pembahasan tentang pengembangan aplikasi serta saran pengembangan lebih lanjut.

BAB II

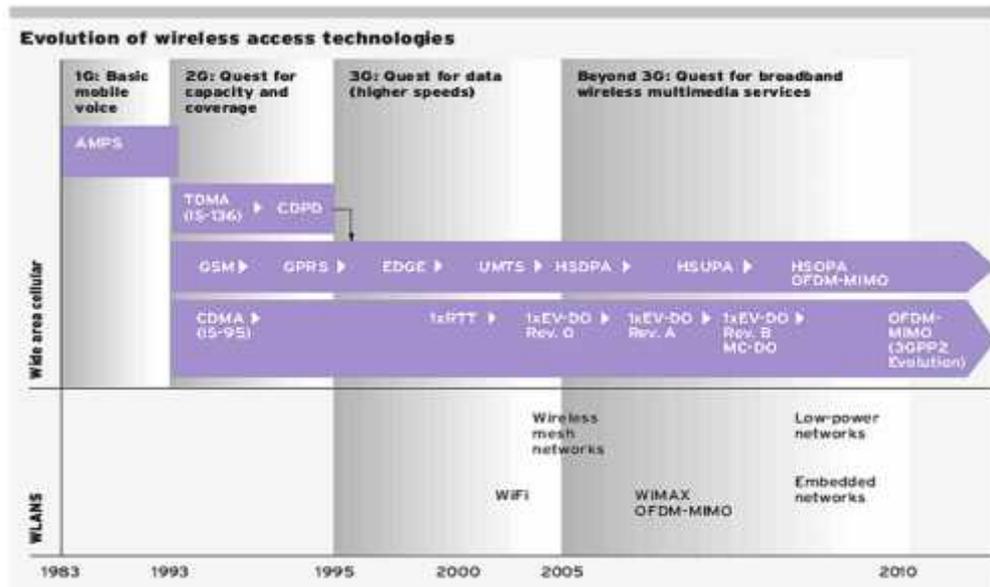
LANDASAN TEORI

2.1. Teknologi *Mobile*

Perkembangan teknologi *mobile* berkembang sangat pesat sejak pertama kali diluncurkan pada tahun 2005, hal ini tidak terlepas akan adanya kebutuhan manusia yang menginginkan stabilitas, kecepatan dan keakuratan informasi serta data. Salah satu perkembangan yang dirasakan saat ini adalah pada teknologi *wireless* dan sistem operasi *mobile* tersebut serta dukungan fitur-fitur pendukung lainnya seperti GPS pada *smartphone*.

2.1.1. *Wireless Mobile Network*

Wireless merupakan teknologi yang digunakan oleh *mobile device* untuk transfer data dan pesan layanan permintaan data dengan menggunakan jaringan *wireless*. Gambar di bawah ini memperlihatkan bahwa jaringan *wireless* untuk *mobile* dapat diklasifikasikan menjadi beberapa generasi.



Gambar 2.1 *Wireless Technology Roadmap*
(sumber: Siemens, Jan 2007).

Dari gambar di atas jaringan *wireless* untuk *mobile* saat ini dapat dibagi menjadi beberapa generasi yaitu:

1. Generasi pertama telekomunikasi bergerak (1G)

Pada dekade 70-an di ciptakan sebuah perangkat telepon bergerak dengan menggunakan sistem dengan menggunakan teknologi analog dengan kecepatan rendah dan suara sebagai objek utamanya. Standarisasi yang digunakan adalah semacam AMPS, TACS dan NMT. AMPS digolongkan dalam generasi pertama teknologi telekomunikasi bergerak yang menggunakan teknologi analog pada band frekuensi 800 Mhz dan menggunakan metode akses FDMA (*Frequency Division Multiple Access*). Pada generasi pertama ini data *bandwidth* yang mampu dilewatkan hanya sebesar 1,9 kbps dan dengan menggunakan sistem multiplexing FDMA.

2. Generasi kedua telekomunikasi bergerak (2G)

Sistem generasi kedua (2G) didesain pada dekade 80an dan mulai diimplementasikan pada awal dekade 90an atau sekitar 1991. Secara garis besar masih digunakan untuk layanan suara namun secara perlahan sudah mulai beralih ke teknologi digital. Sistem 2G menyediakan layanan komunikasi data dengan teknologi circuit-switched dengan kecepatan rendah. Secara umum layanan 2G yang berkembang, baik itu menggunakan standarisasi apa pun, mengedepankan layanan suara dengan sistem digital serta mulai memperkenalkan teknologi layanan pertukaran data. Layanan pertukaran data ini masih tergolong sederhana karena masih dibatasi *bandwidth* 14,4 kbps. Layanan ini sering dikenal masyarakat dengan istilah layanan pesan singkat atau SMS (*Short Message Service*).

3. Generasi ketiga telekomunikasi bergerak (3G)

Pada dekade 90an dua organisasi bekerja untuk mendefinisikan kelanjutan generasi telepon selular berikutnya. Mereka menyebutnya generasi ketiga (3G). Generasi ketiga ini hadir dengan tujuan mengeliminasi incompatibility hasil dari berbagai macam versi dari generasi kedua dan akhirnya menjadi sistem global yang sejati. Disamping mencoba

mengatasi incompatibility antar sistem sebuah sistem 3G diharapkan akan mendapatkan kualitas channel suara yang lebih baik dari generasi sebelumnya dan kemampuan dalam mengirimkan data *broadband* hingga 2 Mbps (setara dengan 1 E1 = 2,048 Mbps). Sayangnya kedua organisasi tersebut (masing-masing organisasi menelurkan standarisasi masing-masing yakni WCDMA dan CDMA2000) gagal dalam melakukan rekonsiliasi perbedaannya, hasil akhirnya kita bisa melihat pengenalan akan dua macam versi dari generasi ketiga telepon seluler. Bahkan sebagai tambahannya, China saat ini dalam tahapan untuk mengimplementasikan versi ketiga dari 3G. Dalam perjalanannya menuju generasi ketiga, diciptakanlah sebuah sistem yang disebut Sistem 2,5G. Sistem ini adalah kondisi perantara dari 2G ke 3G. Standarisasi yang dikenal pada generasi ini seperti EDGE, GPRS, 1XRTT. Pada dasarnya 2,5 G didesain untuk meningkatkan kapasitas kanal frekuensi radio dari teknologi 2G dan memperkenalkan layanan data hingga kecepatan 384 kbps. Aspek paling penting dari generasi 2,5 adalah kanal data dioptimasi untuk paket data yang akan memperkenalkan media akses ke internet melalui perangkat mobile entah itu telepon, laptop ataupun PDA. Berdasarkan perkembangan sejarah yang telah tertulis, indikasi munculnya sebuah generasi baru akan terjadi setiap satu dekade (sekitar sepuluh tahun), kenyataan ini memunculkan semangat untuk memulai kelahiran dari generasi keempat dalam komunikasi telepon bergerak.

4. Generasi ke-empat teknologi telekomunikasi bergerak (4G)

Teknologi 4G (*Fourth Generation*) adalah teknologi kelanjutan dari proses perkembangan teknologi telepon seluler (*mobile phone*). Sebelumnya masyarakat telah sangat mengenal dengan teknologi 2G (*Second Generation*) yang sangat ngetrend dengan teknologi voice call dan SMS. Baru-baru ini masyarakat dikenalkan dengan teknologi 3G (*Third Generation*) dengan andalannya teknologi *video call*. Di generasi keempat (4G) akan dapat menyediakan solusi IP yang komprehensif dimana suara, data, dan arus multimedia dapat sampai kepada pengguna kapan saja dan

dimana saja, pada rata-rata data lebih tinggi dari generasi sebelumnya. Belum ada definisi formal untuk 4G. Bagaimanapun, terdapat beberapa pendapat yang ditujukan untuk 4G, yakni: 4G akan merupakan sistem berbasis IP terintegrasi penuh. Ini akan dicapai setelah teknologi kabel dan nirkabel dapat dikonversikan dan mampu menghasilkan kecepatan 100Mb/detik dan 1Gb/detik baik dalam maupun luar ruang dengan kualitas premium dan keamanan tinggi. 4G akan menawarkan segala jenis layanan dengan harga yang terjangkau. Setiap *handset* 4G akan langsung mempunyai nomor IP v6 dilengkapi dengan kemampuan untuk berinteraksi internet telephony yang berbasis Session Initiation Protocol (SIP). Semua jenis radio transmisi seperti GSM, TDMA, EDGE, CDMA 2G, 2.5G akan dapat digunakan, dan dapat berintegrasi dengan mudah dengan radio yang dioperasikan tanpa lisensi seperti IEEE 802.11 di frekuensi 2.4GHz & 5-5.8Ghz, bluetooth dan selular. Integrasi voice dan data dalam channel yang sama. Integrasi voice dan data aplikasi SIP-enabled (*Martin. 2006*).

2.1.2. Android

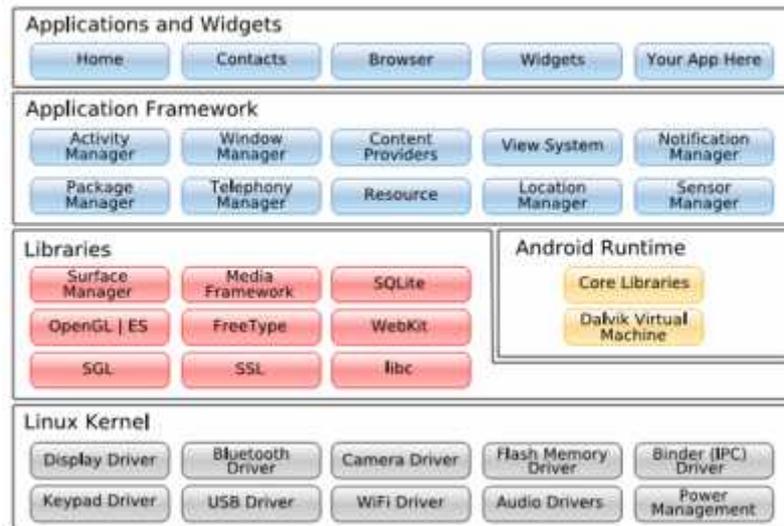
2.1.2.1. Pengertian android

Android adalah kumpulan perangkat lunak yang ditujukan bagi perangkat bergerak mencakup sistem operasi, *middleware*, dan aplikasi kunci. Android *Standart Development Kit* (SDK) menyediakan perlengkapan dan *Application Programming Interface* (API) yang diperlukan untuk mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java (Amiral, 2010).

Android dikembangkan oleh Google bersama *Open Handset Alliance* (OHA) yaitu aliansi perangkat selular terbuka yang terdiri dari 47 perusahaan *Hardware*, *Software* dan perusahaan telekomunikasi ditujukan untuk mengembangkan standar terbuka bagi perangkat selular.

2.1.2.2. Anatomi Android

Dalam paket sistem operasi Android terdiri dari beberapa unsur seperti tampak pada gambar 2.3. Secara sederhana arsitektur Android merupakan sebuah kernel Linux dan sekumpulan pustaka C / C++ dalam suatu *framework* yang menyediakan dan mengatur alur proses aplikasi (Amiral, 2010).



Gambar 2.2 Anatomi Android
(sumber: Ed Burnette, 2008).

a. Linux Kernel

Android dibangun di atas kernel Linux 2.6. Namun secara keseluruhan Android bukanlah Linux, karena dalam Android tidak terdapat paket standar yang dimiliki oleh Linux lainnya. Linux merupakan sistem operasi terbuka yang handal dalam manajemen memori dan proses. Oleh karenanya pada Android hanya terdapat beberapa servis yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan driver. Kernel Linux menyediakan driver layar, kamera, keypad, WiFi, Flash Memory, audio, dan IPC (*Interprocess Communication*) untuk mengatur aplikasi dan lubang keamanan.

b. Libraries

Android menggunakan beberapa paket pustaka yang terdapat pada C/C++ dengan standar *Berkeley Software Distribution* (BSD) hanya setengah dari

yang aslinya untuk tertanam pada kernel Linux. Beberapa *library* diantaranya:

1. *Media Library* untuk memutar dan merekam berbagai macam format audio dan video.
2. *Surface Manager* untuk mengatur hak akses layer dari berbagai aplikasi.
3. *Graphic Library* termasuk didalamnya *SGL* dan *OpenGL*, untuk tampilan 2D dan 3D.
4. *SQLite* untuk mengatur relasi database yang digunakan pada aplikasi.
5. *SSL* dan *WebKit* untuk browser dan keamanan internet.
6. *Library* tersebut bukanlah aplikasi yang berjalan sendiri, namun hanya dapat digunakan oleh program yang berada di level atasnya. Sejak versi Android 1.5, pengembang dapat membuat dan menggunakan *library* sendiri menggunakan *Native Development Toolkit* (NDK).

c. Android Runtime

Pada Android tertanam paket *library* inti yang menyediakan sebagian besar fungsi Android. Inilah yang membedakan Android dibandingkan dengan sistem operasi lain yang juga mengimplementasikan Linux. *Android Runtime* merupakan mesin virtual yang membuat aplikasi Android menjadi lebih tangguh dengan paket *library* yang telah ada. Dalam Android Runtime terdapat 2 bagian utama, diantaranya :

1. *Library* Inti, Android dikembangkan melalui bahasa pemrograman Java, tapi Android Runtime bukanlah mesin virtual Java. *Library* inti Android menyediakan hampir semua fungsi yang terdapat pada *library* Java serta beberapa *library* khusus Android.
2. Mesin Virtual Dalvik, Dalvik merupakan sebuah mesin virtual yang dikembangkan oleh Dan Bornstein yang terinspirasi dari nama sebuah perkampungan yang berada di Iceland. Dalvik hanyalah interpreter mesin virtual yang mengeksekusi file dalam format

Dalvik Executable (*.dex). Dengan format ini Dalvik akan mengoptimalkan efisiensi penyimpanan dan pengalamatan memori pada file yang dieksekusi. Dalvik berjalan di atas kernel Linux 2.6, dengan fungsi dasar seperti *threading* dan manajemen memori yang terbatas.

d. Application Framework

Kerangka aplikasi menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi Android. Selain itu, juga menyediakan abstraksi generic untuk mengakses perangkat, serta mengatur tampilan *user interface* dan sumber daya aplikasi. Bagian terpenting dalam kerangka aplikasi Android adalah sebagai berikut (*Burnette, 2008*):

1. *Activity Manager*, berfungsi untuk mengontrol siklus hidup aplikasi dan
2. menjaga keadaan "Backstack" untuk navigasi penggunaan.
3. *Content Providers*, berfungsi untuk merangkum data yang memungkinkan
4. digunakan oleh aplikasi lainnya, seperti daftar nama.
5. *Resource Manager*, untuk mengatur sumber daya yang ada dalam
6. program. Serta menyediakan akses sumber daya diluar kode program,
7. seperti karakter, grafik, dan file layout.
8. *Location Manager*, berfungsi untuk memberikan informasi detail mengenai lokasi perangkat Android berada.
9. *Notification Manager*, mencakup berbagai macam peringatan seperti, pesan masuk, janji, dan lain sebagainya yang akan ditampilkan pada *statusbar*.

2.1.2.3. Kelebihan Android

Sudah banyak *platform* untuk perangkat selular saat ini, termasuk didalamnya Symbian, iPhone, Windows Mobile, BlackBerry, Java Mobile Edition, Linux Mobile (LiM0), dan banyak lagi. Namun ada beberapa hal yang

menjadi kelebihan Android. Walaupun beberapa fitur-fitur yang ada telah muncul sebelumnya pada platform lain, Android adalah yang pertama menggabungkan hal seperti berikut (Amiral, 2009):

1. Keterbukaan (*Open Source*), Bebas pengembangan tanpa dikenakan biaya terhadap sistem karena berbasis Linux dan *open source*. Pembuat perangkat menyukai hal ini karena dapat membangun *platform* yang sesuai yang diinginkan tanpa harus membayar *royalty*. Sementara pengembang *software* menyukai karena Android dapat digunakan diperangkat manapun dan tanpa terikat oleh vendor manapun.
2. Arsitektur komponen dasar Android terinspirasi dari teknologi internet *Mashup*. Bagian dalam sebuah aplikasi dapat digunakan oleh aplikasi lainnya, bahkan dapat diganti dengan komponen lain yang sesuai dengan aplikasi yang dikembangkan.
3. Banyak dukungan service, kemudahan dalam menggunakan berbagai macam layanan pada aplikasi seperti penggunaan layanan pencarian lokasi, database SQL, browser dan penggunaan peta. Semua itu sudah tertanam pada Android sehingga memudahkan dalam pengembangan aplikasi.
4. Siklus hidup aplikasi diatur secara otomatis, setiap program terjaga antara satu sama lain oleh berbagai lapisan keamanan, sehingga kerja sistem menjadi lebih stabil. Pengguna tak perlu khawatir dalam menggunakan aplikasi pada perangkat yang memorinya terbatas.
5. Dukungan grafis dan suarat terbaik, dengan adanya dukungan 2D grafis dan animasi yang diilhami oleh *Flash* menyatu dalam 3D menggunakan *OpenGL* memungkinkan membuat aplikasi maupun game yang berbeda.
6. Portabilitas aplikasi, aplikasi dapat digunakan pada perangkat yang ada saat ini maupun yang akan datang. Semua program ditulis dengan menggunakan bahasa pemrograman Java dan dieksekusi oleh mesin virtual Dalvik, sehingga kode program portabel antara ARM, X86, dan arsitektur lainnya. Sama halnya dengan dukungan masukan seperti penggunaan

Keyboard, layar sentuh, *trackball* dan resolusi layar semua dapat disesuaikan dengan program.

2.2. GPS

GPS (*Global Positioning System*) merupakan suatu sistem penentuan lokasi berdasarkan sinyal satelit yang akan menghasilkan informasi berupa koordinat, latitude, longitude dan lokasi dalam peta. Sistem GPS ini hasil gabungan program U.S. Navy's TIMATION dan proyek U.S. Air Force's 621B di bawah tanggung jawab JPO (*Joint Program Office*). Sistem ini awalnya dirancang untuk penggunaan dalam beberapa sistem pertahanan utama di Amerika Serikat. Namun dalam perkembangannya penggunaan GPS meliputi berbagai bidang Militer, Navigasi, Sistem Informasi Geografis dan Pelacakan kendaraan.

Global Positioning System (GPS) digunakan untuk menentukan posisi suatu objek di atas permukaan bumi, berdasarkan posisi astronomis garis lintang dan garis bujur. GPS ini merupakan teknologi yang baru untuk smart phone yang terintegrasi dengan satelite yang dapat memberikan akurasi dari 5 sampai 10 meter (*Manav Singhal, 2012*).

GPS adalah sistem satelit navigasi yang paling populer dan paling banyak diaplikasikan di dunia pada saat ini, baik di darat, laut, udara, maupun angkasa. Disamping aplikasi-aplikasi militer, bidang-bidang aplikasi GPS yang cukup marak saat ini antara lain meliputi *survey* pemetaan, geodinamika, geodesi, geologi, geofisik, transportasi dan navigasi, pemantauan deformasi, pertanian, kehutanan, juga di bidang olahraga dan rekreasi.

2.2.1. Metode Advanced Positioning

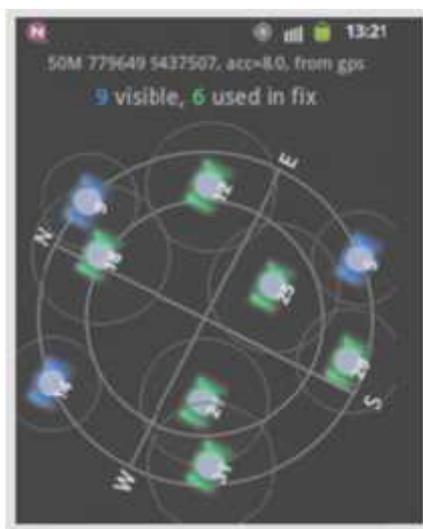
Pada umumnya menggunakan teknologi *Assisted-Global Positioning System* (A-GPS). A-GPS juga merupakan metode yang berbasis pada waktu. Pada metode ini, akan dilakukan pengukuran waktu tiba dari sebuah sinyal yang dikirim dari tiga buah satelit GPS. Hal ini berarti handset harus memiliki fasilitas untuk mengakses GPS. A-GPS juga menghasilkan akurasi secara vertikal dan estimasi jarak yang baik. Akurasinya pun sampai kurang dari 10m.



Gambar 2.3 Arsitektur *Advanced Positioning* Publikasi Penelitian tentang implementasi LBS di Android menggunakan GPS oleh IJCSI
(sumber: Manaf singhal dkk 2012).

2.2.2. Contoh Penerapan GPS pada *smartphone android*

Penerapan GPS berbasis Android adalah *GPS essentials*, aplikasi ini tersedia pada *Android Market* atau *Play Store* dengan versi berbayar dan gratis.



Gambar 2.4 Aplikasi *GPS essentials*
(sumber: <http://play.google.com>).

Aplikasi ini digunakan untuk mendeteksi penangkapan sinyal pada GPS yang online untuk:

- a. Temukan lokasi, untuk mengetahui lokasi dan GPS.

- b. *Internet access*, di gunakan untuk membaca peta dan mengirimkan *bug reports*.
- c. *USB storage contents*, untuk menulis *waypoints* dan fasilitas *track*
- d. *Read phone* alamat dan identitas, digunakan untuk memberikan penandaan pada gambar pada perangkat.
- e. *Pengambilan gambar dan video*, pada perangkat yang memiliki camera aplikasi ini dapat memberikan fasilitas pengambilan gambar.

2.2.3. Contoh Penerapan GPS pada makhluk hidup

Kemajuan teknologi GPS sangat pesat sehingga mengakibatkan pencarian memiliki ruang lingkup kecil dengan berjarak beberapa meter (4m) dari jarak benda sebenarnya, hal ini menyebabkan pegguan GPS tidak hanya terbatas pada ruanglingkup kendaraan, militer, geologi saja bahkan telah merambah ke dunia hewan, seperti pelacakan posisi keberadaan hewan peliharaan “*aplikasi A Dog GPS Collar Sistem*”, aplikasi bekerja dengan penanaman sebuah *chip* di bawah kulit anjing dengan sebuah operasi kecil. Aplikasi ini banyak digunakan oleh pemburu (<http://gpsmobil.harga-murah.com>).

Lain halnya dengan aplikasi GPS untuk menemukan keberadaan ikan pada nelayan, di daerah Provinsi Kalimantan Barat, aplikasi GPS digunakan untuk mendeteksi suhu permukaan laut karena adanya perbedaan antara satu dengan yang lain dengan deteksi panas oleh *phytoplankton* sehingga ikan-ikan berkumpul, pada radius beberapa meter alat ini dapat melihat bayangan dan gerombolan titik ikan (<http://www.equator-news.com>).

2.3. Pemetaan (Google Map)

Google Maps merupakan layanan dari google yang mempermudah pgunanya untuk melakukan kemampuan pemetaan untuk aplikasi yang dibuat. Sedangkan Google Maps API memungkinkan pengembangan untuk mengintegrasikan *Google Maps* ke dalam situs web. Dengan menggunakan *Google Maps* API memungkinkan untuk menanamkan situs *Google Maps* ke dalam situs eksternal, di mana situs data tertentu dapat dilakukan *overlay*.

Meskipun pada awalnya hanya *Java Script* API, API Maps sejak diperluas untuk menyertakan sebuah API untuk Adobe Flash aplikasi, layanan untuk mengambil gambar peta statis, dan layanan web untuk melakukan geocoding, menghasilkan petunjuk arah mengemudi, dan mendapatkan profil elevasi. Kelas kunci dalam perpustakaan Maps adalah *MapView*, sebuah subclass dari *ViewGroup* dalam standar perpustakaan Android. Sebuah *MapView* menampilkan peta dengan data yang diperoleh dari layanan *Google Maps* (Amri, 2012).

2.4. Metodologi Penelitian

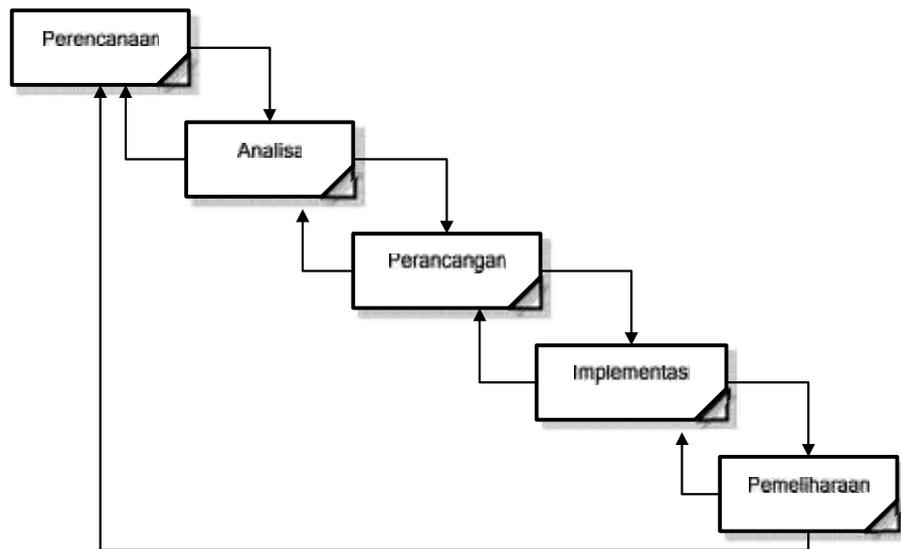
Penggunaan metode dalam sebuah penelitian diperlukan agar penelitian mempunyai tahapan proses yang jelas. Fungsi metode penelitian ini adalah sebagai kerangka acuan dalam melaksanakan penelitian agar penelitian terfokus kepada tujuan awal.

Dalam penelitian ini, akan menerapkan metode *waterfall model* dengan menggunakan pendekatan metode RUP pada setiap fasenya.

2.4.1. Waterfall

Metode *waterfall* ini adalah model klasik yang mengusung pengembangan perangkat lunak yang sistematis, berurutan atau sekuensial dimulai pada tingkat dan kemajuan sistem pada seluruh persyaratan dalam perancangan, analisis, perancangan (*desain*), implementasi (*testing*), hingga ke tahap pemeliharaan dalam membangun software (perangkat lunak). Berikut ini gambaran dari *Linear Sequential Model / waterfall model*.

Secara umum kerangka kerja model *Waterfall* adalah sebagai berikut : (Pressman,1997)



Gambar 4.2 Kerangka Kerja Pengembangan Sistem

Keterangan:

1. Perencanaan

Menyangkut studi kebutuhan pengguna, studi kelayakan baik secara teknis maupun secara teknologi serta penjadwalan pengembangan perangkat lunak. Dapat juga dikatakan sebagai defenisi kebutuhan sistem.

2. Analisa

Tahap dimana kita berusaha mengenali seluruh permasalahan yang muncul pada pengguna (*user*), mengenali komponen-komponen sistem, objek-objek, hubungan antar objek, dan sebagainya.

3. Perancangan

Merupakan tahap pencarian solusi dari permasalahan yang didapat dari tahap analisa. Perancangan dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap.

4. Implementasi

Tahap pengimplementasian rancangan sistem kesituasi nyata. Pada tahap ini dimulai proses pemilihan perangkat keras, penyusunan perangkat lunak aplikasi (*coding*), dan pengujian (*testing*). Apakah sistem sudah sesuai dengan kebutuhan. Jika belum, dilakukan proses iteratif, yaitu kembali ke tahap-tahap sebelumnya.

5. Pemeliharaan

Tahap ini dimulai saat melakukan pengoperasian sistem dan melakukan perbaikan-perbaikan/perubahan-perubahan kecil terhadap sistem jika diperlukan. Jika sistem tidak dapat dikembangkan lagi, maka akan kembali ke tahap pertama, yaitu perencanaan untuk pembuatan sistem yang baru.

2.4.2. *Rational Unified Process (RUP)*

Untuk mengembangkan aplikasi *mobile positioning* sistem, pada tahapan metodologi penelitian menggunakan metode pengembangan perangkat lunak *Rational Unified Process (RUP)*.

2.4.2.1. **Pengertian RUP**

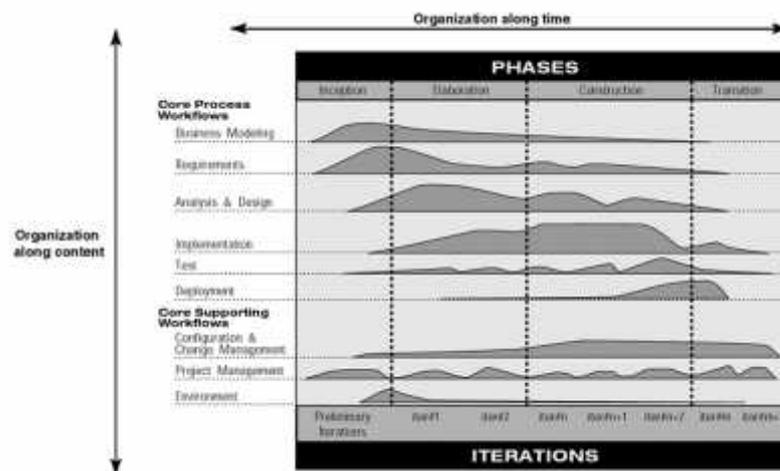
Rational Unified Process adalah sebuah Proses Rekayasa Perangkat Lunak. RUP menyediakan pendekatan disiplin untuk memberikan tugas dan tanggung jawab dalam organisasi pengembang perangkat lunak. Tujuannya untuk memastikan perangkat lunak yang berkualitas tinggi dan sesuai kebutuhan penggunaannya dalam anggaran dan jadwal yang dapat diprediksi.

RUP mengarahkan kita terhadap pengembangan perangkat lunak secara praktis dan efektif. Terdapat 6 *best practice* atau disebut juga *basic principle* dalam metode RUP, antara lain

1. *Develop software iteratively*, bertujuan untuk mengurangi resiko pada awal proyek.
2. *Manage requirements*, bertujuan untuk mengatur kebutuhan yang diperlukan selama proyek.
3. *Use component-based architectures* untuk membangun komponen arsitektur sebuah proyek.
4. *Visually model software*, bertujuan untuk merancang sebuah model visual perangkat lunak, untuk mendapatkan struktur dan perilaku dari aritektur perangkat lunak.
5. *Continuously verify software quality*,

6. *Control changes to software.* Kemampuan untuk mengatur serta mengubah perangkat lunak saat dibutuhkan.

RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML). Melalui gambar 2.2 dibawah dapat dilihat bahwa RUP memiliki 2 dimensi, yaitu:



Gambar 2.13 Struktur Proses dua Dimensi RUP

(sumber: Kruchten, 2000).

Dimensi pertama digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek dinamis dari pengembangan perangkat lunak. Aspek ini dijabarkan dalam tahapan pengembangan atau fase. Setiap fase akan memiliki suatu *major milestone* yang menandakan akhir dari awal dari phase selanjutnya. Setiap phase dapat berdiri dari satu atau beberapa iterasi. Dimensi ini terdiri atas *Inception*, *Elaboration*, *Construction*, dan *Transition*.

Dimensi kedua digambarkan secara vertikal. Dimensi ini mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan kedalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing*, *what*, *how* dan *when*. Dimensi ini terdiri atas *Business Modeling*, *Requirement*,

Analysis and Design, Implementation, Test, Deployment, Configuration dan Change Management, Project Management, Environment.

2.4.2.2. Fase RUP

Fase-fase pada RUP berdasarkan waktu pengerjaan proyek dapat dibagi menjadi 4 fase, yaitu *Inception, Elaboration, Construction* dan *Transition* (Rational Team, 2001).

1. Fase *Inception*

Fase *inception* merupakan fase untuk mengidentifikasi masalah, untuk itu diperlukan juga identifikasi entitas dari luar yang berhubungan dengan sistem. Pada fase ini melibatkan semua identifikasi *use case* dan gambarannya. Selain itu juga termasuk kriteria keberhasilan proyek, perkiraan resiko, perkiraan terhadap *resource* yang dibutuhkan dan merencanakan penjadwalan *milestone*.

2. Fase *Elaboration*

Tujuan dari fase *elaboration* (pengembangan) adalah menganalisa area permasalahan, mengembangkan rencana proyek, dan menghilangkan unsur-unsur yang memiliki resiko besar terhadap proyek.

3. Fase *Construction*

Selama fase konstruksi, semua komponen dan fitur yang dikembangkan terintegrasi ke dalam produk dan secara menyeluruh semua fitur telah diuji. Di lain sisi, proses konstruksi adalah sebuah proses *manufacturing*, dimana terdapat penekanan dalam mengelola *resource* dan mengatur operasi untuk mengoptimalkan jadwal dan kualitas. Pada tahap ini pola pikir (*mindset*) mengalami perubahan dari pengembangan *intellectual property* pada fase *Inception* dan *Elaboration*, menjadi pengembangan *deployable product*.

4. Fase *Transition*

Tujuan dari fase ini adalah untuk transisi dari produk perangkat lunak ke pengguna akhir. Apabila produk telah di luncurkan kepada pengguna, maka isu-isu akan muncul dari pengguna. Nantinya isu ini akan digunakan untuk tahap perbaikan terhadap produk.

2.5. *Unified Modelling Language (UML)*

Unified Modelling Language (UML) adalah sebuah bahasa yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C (*Dharwiyanti, 2003*).

UML merupakan sebuah standar dalam pembuatan dari model-model yang mempresentasikan sistem *software* dan bisnis yang beorientasi objek. UML menggabungkan diagram-diagram terbaik yang diterapkan oleh *software developer* selama 40 tahun belakangan. Pemodelan standarisasi UML tidak mendikte bagaimana untuk menerapkan pemodelannya. Standar pendekatan ini menyediakan kebebasan besar bagi para *developer* untuk menerapkan gaya dan tekniknya untuk memastikan konsistensi dalam pekerjaan mereka (*Pender, 2002*).

Dalam aplikasi-aplikasi bisnis, metodologi-metodologi pengembangan aplikasi yang menggunakan bahasa pemrograman terstruktur(*structured programming language*), DFD, dan ERD kurang dapat beradaptasi dengan kebutuhan dan harapan pengguna (*user's needs and expectations*), dengan alasan itu para pakar dibidang perancangan perangkat lunak pada sekitar tahun 1980-1990 mulai bekerja dengan bahasa pemrograman berorientasi objek (OOP[*Object Oriented Programming*]) seperti C++ dan Java.

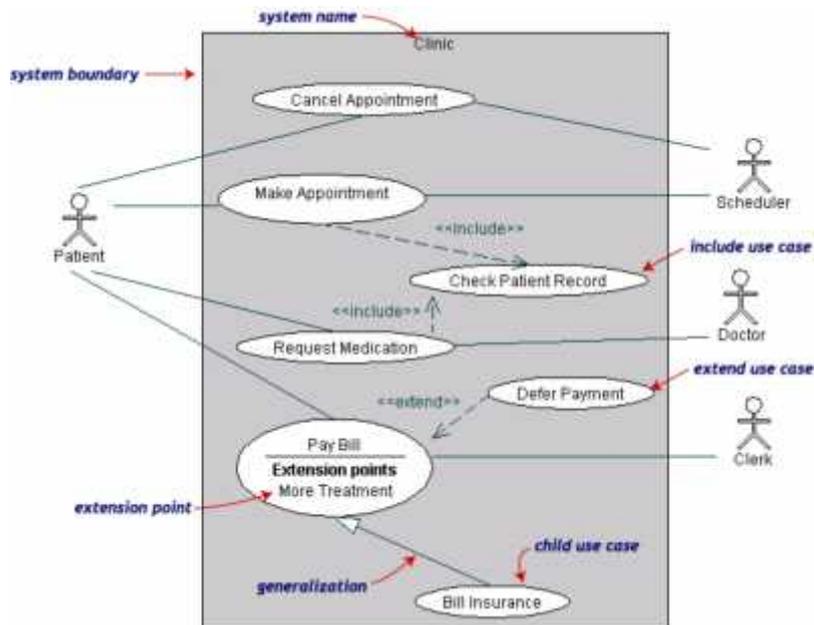
Oleh karena itu, UML merupakan kolaborasi antara metoda Booch, OMT(*object Modeling Technique*), serta OOSE(*Object Oriented Software Engineering*) dan beberapa metoda lainnya merupakan metodologi yang paling sering digunakan saat ini untuk mengadaptasi maraknya penggunaan bahasa "pemrograman berorientasi objek "(OOP) (*Adi Nugroho, 2009*).

2.5.1. *Usecase Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-*create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu (*Dharwiyanti, 2003*).

Use case diagram dapat sangat membantu bila kita sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *test case* untuk semua *feature* yang ada pada sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah *use case* dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang *common*.

Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain.



Gambar 2.4 Usecase Diagram
(sumber: Dharwiyanti,2003).

2.5.2. Sequence Diagram

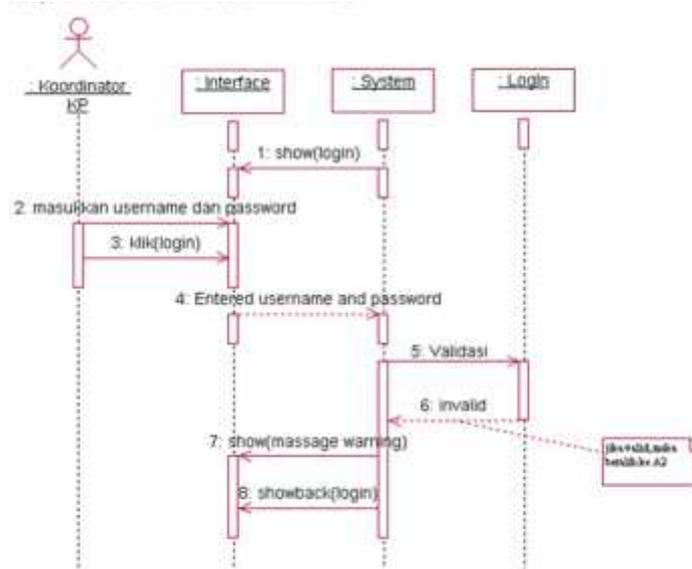
Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu.

Diawali dari apa yang *trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain

berikutnya, *message* akan dipetakan menjadi operasi/metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. Untuk objek-objek yang memiliki sifat khusus,

standar UML mendefinisikan *icon* khusus untuk objek *boundary*, *controller* dan *persistent entity*.



Gambar 2.5 Sequence Diagram
(sumber: Dharwiyanti,2003).

2.5.3. Class Diagram

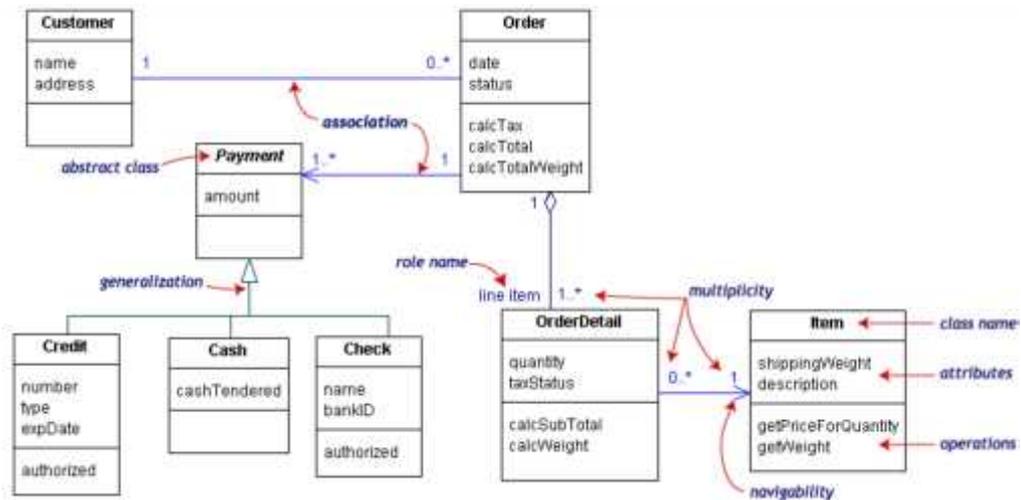
Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

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

1. Nama (dan *stereotype*)
2. Atribut
3. Metoda

Atribut dan metoda dapat memiliki salah satu sifat berikut:

- a. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan
- b. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
- c. *Public*, dapat dipanggil oleh siapa saja



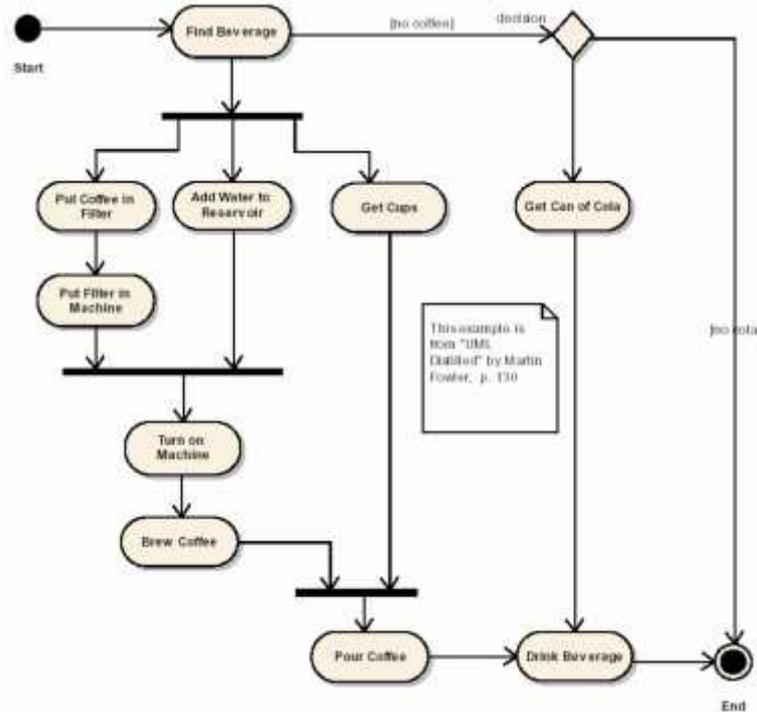
Gambar 2.9 Contoh Class Diagram
 (sumber: Dharwiyanti,2003).

2.5.4. Activity Diagram

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Sama seperti *state*, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan aktivitas. *Decision* digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (*fork* dan *join*) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal.

Activity diagram dapat dibagi menjadi beberapa *object swimlane* untuk menggambarkan objek mana yang bertanggung jawab untuk aktivitas tertentu.



Gambar 2.10 Activity Diagram Tanpa Swimlane
(sumber: Dharwiyanti,2003)

2.6. Location Based Services (LBS)

Layanan berbasis lokasi memberikan layanan secara personal kepada pengguna perangkat bergerak yang telah disesuaikan dengan lokasi pengguna. Layanan ini menggunakan teknologi *Geographic Information System* (GIS) juga beberapa teknologi dari sisi *client* seperti GPS, atau sisi *server* seperti posisi layanan jaringan operator *provider*. Ciri utama dari LBS aplikasi menggunakan waktu nyata (*real time*).

2.6.1. Komponen LBS

Terdapat lima komponen pendukung utama dalam Layanan Berbasis Lokasi, antara lain:

1. Piranti *Mobile*

Piranti *Mobile* adalah layanan satu komponen penting dalam LBS. Piranti ini berfungsi sebagai alat bantu bagi pengguna untuk meminta informasi.

Hasil dari informasi yang diminta dapat berupa teks, suara, gambar juga dapat berupa koordinat posisi *mobile*. Piranti *Mobile* yang dapat digunakan bisa berupa PDA, *smartphone*, dan *laptop*.

2. Jaringan Komunikasi

Komponen kedua adalah jaringan komunikasi. Komponen ini berfungsi sebagai jalur penghubung yang dapat mengirimkan data-data yang dikirim oleh pengguna dari piranti *mobile*-nya untuk kemudian dikirim ke penyedia layana dan kemudian hasil permintaan tersebut dikirimkan kembali oleh penyedia layanan kepada pengguna.

3. Komponen *Positioning* (Penunjuk Posisi).

Setiap layana yang diberikan oleh penyedia layana biasanya akan berdasarkan pada posisi pengguna yang meminta layanan tersebut. Oleh karena itu diperlukan komponen yang berfungsi memproses yang akan menentukan posisi pengguna tersebut pada saat itu juga. Posisi tersebut dapat di menggunakan GPS.

4. Penyedia layanan dan aplikasi.

Penyedia layanan merupakan komponen LBS yang memberikan berbagai macam layanan agar dapat mengetahui posisi pada saat itu, maka aplikasi dan penyedia layana langsung memproses permintaan tersebut, mulai dari menghitung dan menentukan posisi pengguna, menemukan rute jalan, mencari data *Yellow Pages* sesuai dengan permintaan.

5. Penyedia data dan konten

Penyedia layana tidak selalu menyimpan seluruh data dan informasi yang diolahnya. Karena bisa jadi berbagai macam data dan informasi yang diolah tersebut berasal dari pengembang atau pihak ketiga yang memang memiliki otoritas untuk menyimpannya. Sebagai contoh basis data geografis dan lokasi bisa saja bersal dari badan-badan milik pemerintah atau data-data perusahaan.



Gambar 2.11 Komponen LBS
 (sumber: Dario Ortuno, 2011).

2.7. Contoh Aplikasi Penelitian Mengenai LBS

Saat ini penelitian mengenai penentuan *geolocalic* banyak dilakukan untuk menentukan lokasi objek yang ada di bumi, menemukan posisi keberadaan *divice* lainnya berdasarkan GPS. Berikut contoh penelitian mengenai LBS yang pernah dilakukan sebelumnya seperti aplikasi “Abang kat mana?”, Aplikasi ini tersedia di *google play store* dengan *link*: play.google.com/store/apps/details?id=com.cr.akm, aplikasi ini memungkinkan pengguna mengetahui keberadaan *handphone* lain dengan mengirimkan format SMS tertentu ke nomor *handphone* orang yang ingin diketahui keberadannya kemudian pengirim akan mendapatkan *link* yang di tujukan ke *Google map* yang apabila *link* tersebut di *click* akan menampilkan posisi keberadaan si penerima SMS tersebut, dengan syarat aplikasi ini haruslah terinstal pada penerima SMS. Pada intinya aplikasi ini dapat memberikan pemberitahuan dimana *handphone* (Android) berada berdasarkan pada lokasi GPS yang diterima oleh *handphone* tersebut pada saat itu juga. Jika sinyal GPS tidak ada, aplikasi ini akan memberitahu lokasi terakhir kali sinyal GPS berada.

BAB III

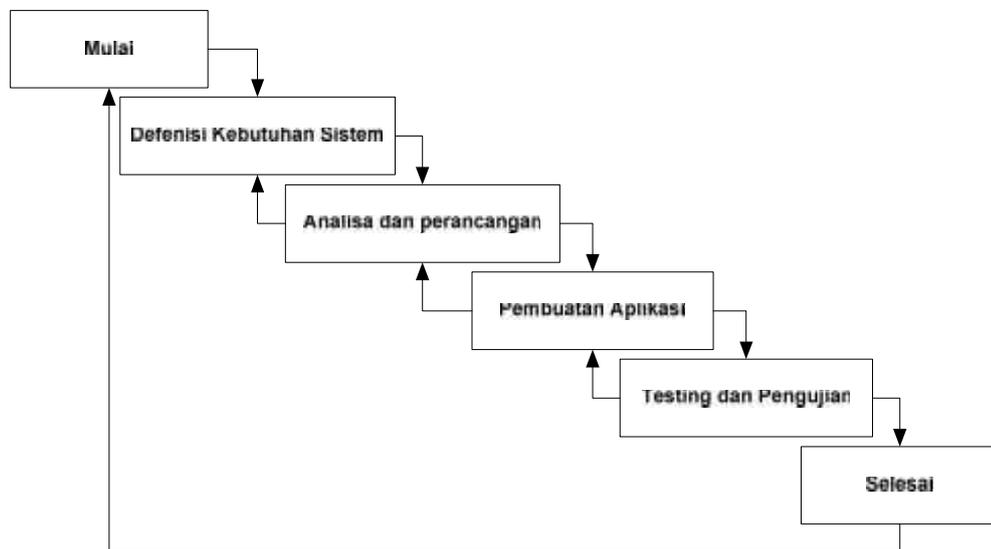
METODOLOGI PENELITIAN

3.1. Tahapan *Waterfall*

Bab ini akan dijelaskan mengenai tahapan-tahapan dalam pembuatan aplikasi *Mobile Position* dengan menerapkan metode *waterfall* menggunakan pendekatan metode RUP pada setiap tahap alur *waterfall model*.

3.1.1. Alur Tahapan *Waterfall*

Alur tahapan *Waterfall* yang akan digunakan dalam membuat rancang bangun aplikasi *Mobile Position* pada *Android* ditunjukkan pada Gambar 3.1 dibawah ini:



Gambar 3.1. Tahapan Penelitian dengan *Waterfall*

3.1.1.1. Defenisi Kebutuhan Sistem

Pada fase ini akan dilakukan tugas-tugas sebagai berikut:

- a. Pengenalan masalah, memahami permasalahan yang terjadi, mengapa

diperlukan aplikasi *Mobile Positioning System* pada perangkat *smartphone* dengan sistem operasi *Android*, yaitu:

- i. Pengaruh teknologi yang berkembang pesat membuat manusia membutuhkan sarana informasi yang lebih maju.
 - ii. Dibutuhkannya waktu yang lebih efisien saat ingin mengetahui informasi posisi keberadaan seseorang.
 - iii. Dibutuhkan penelitian untuk mengetahui konsep dasar, struktur, dan sistem kerja, serta pengkodean dalam merancang dan membangun aplikasi *Mobile Positioning System* dengan menggunakan GPS internal pada perangkat *handphone* dengan sistem operasi *Android*.
- b. Pembuatan proposal untuk pembangunan aplikasi *mobile positioning* dengan GPS pada platform *Android*, yaitu mencakup latar belakang permasalahan, pokok permasalahan, tujuan dan batasan permasalahan.
- c. Studi literatur. Mencakup penelusuran teori-teori yang berhubungan dengan penelitian, yang bersumber dari buku, jurnal, artikel internet, dan penelitian-penelitian sejenis yang dapat mendukung pemecahan masalah dalam penelitian yang dilakukan, mencakup:
1. Defenisi *mobile positioning* dengan menggunakann GPS pada sistem operasi *Android*.
 2. Pemrograman *Android*, termasuk sejarah, lingkungan operasional *Android*, keunggulan dan pengkodean untuk sistem operasi *Android*.
 3. *Object Oriented Analysis and Design* (OOAD), mencakup deskripsi dan struktur OOAD, serta pendekatannya dengan *Unified Modelling Language* (UML).
 4. *Rational Unified Process* (RUP), mencakup deskripsi dan fase-fase RUP.

3.1.1.2. Analisa dan Perancangan

Pada fase ini akan dilakukan tugas-tugas sebagai berikut:

- a. Pembuatan UML (*Unified Modelling Language*)
- b. Perancangan struktur menu
- c. Perancangan *prototype* antar muka aplikasi

3.1.1.3. Pembuatan Aplikasi

Pada fase ini akan dilakukan tugas-tugas sebagai berikut:

- a. Pengkodean aplikasi yang berpedoman pada model *use-case* menggunakan bahasa pemrograman Java.
- b. Melakukan pengujian terhadap kesalahan-kesalahan yang mungkin terjadi selama proses pengkodean.

3.1.1.4. Testing dan Pengujian Aplikasi

- a. Melakukan pengujian terhadap aplikasi yang dibuat.
- b. Menyimpulkan fitur-fitur tambahan guna pengembangan dari aplikasi *Mobile Positioning* untuk kedepannya.

BAB IV

ANALISA DAN PERANCANGAN

Bab ini merupakan bagian dari tahapan analisa dan perancangan sistem, pada tahapan ini akan dilakukan analisa sejalan dengan pembuatan deskripsi arsitektur yang dibutuhkan aplikasi juga pembuatan UML (*Unified Modelling Language*), perancangan struktur menu aplikasi yang akan dibangun dan perancangan *prototype* antarmuka aplikasi yang akan dibangun.

4.1. Analisa

Penentuan lokasi keberadaan objek bergerak mejadi permasalahan tersendiri bagi orang yang menginginkan posisi lokasi objek bergerak tersebut, karena keberadaannya selalu berpindah dari suatu tempat ketempatlainnya, penentuan lokasi tersebut pada manusia menjadi suatu hal yang mungkin untuk dilakukan, dengan ditemukannya berbagai macam teknologi diberbagai bidang diantaranya yaitu penggunaan telepon genggam (*handphone*) pada teknologi telekomunikasi, seseorang dapat mengetahui informasi posisi keberadaan orang yang lainnya dengan menggunakan pesan singkat SMS (*Short Message Service*) dan layanan *voice call*, akan tetapi kedua layanan tersebut tidak dapat dijadikan acuan utama karena bisa saja orang yang membalaskan pesan atau berbicara memberikan informasi yang salah terhadap keberadaan posisi lokasinya, hal tersebut juga akan sangat menyulitkan pengguna apabila informasi posisi tersebut dibutuhkan dalam jangka waktu tertentu secara berkala.

Perkembangan teknologi *mobile* yang menggabungkan berbagai komponen teknologi seperti *Global Positioning System* (GPS), *Geographic Information System* (GIS) dan teknologi *wireless* pada saat sekarang telah dapat dilakukan. Hampir semua perangkat ponsel cerdas (*smartphones*) sudah

terintegrasi dengan GPS sehingga sangat memungkinkan untuk mengetahui posisi keberadaan perangkat *mobile* lain yang dipresentasikan pada *Google Maps*.

Diantara perangkat *smartphone* yang terus berkembang yaitu *smartphone* yang menggunakan sistem operasi Android, terdapat lebih dari ribuan aplikasi berbasis Android pada *Play Store* yang telah digunakan oleh pengguna *handphone* sehingga penulis tertarik untuk membangun sebuah aplikasi penentuan posisi berbasis Android.

Aplikasi penentuan posisi ini akan menggunakan teknologi GPS sebagai penentuan posisi keberadaan perangkat *handphone* pengguna. Akan tetapi berbagai macam keterbatasan yang dimiliki oleh perangkat GPS-*internal* pada *smartphone* memiliki berbagai macam keterbatasan sehingga penggunaan GPS pada aplikasi posisi ini digunakan hanya untuk mendapatkan informasi koordinat *latitude* dan *longitude* saja.

Penggunaan teknologi *wireless* untuk mengirimkan koordinat tersebut ke *database* juga memiliki beberapa kendala, seperti terdapat beberapa lokasi yang tidak dapat dijangkau oleh sinyal yang disediakan oleh *provider* jaringan telepon selular, contohnya di daerah pegunungan, dikawasan luar perkotaan yang sulit untuk dijangkau oleh sinyal.

Begitu juga pada penggunaan peta Google, layanan peta yang diberikan oleh perusahaan Google juga memiliki beberapa keterbatasan, seperti terdapat beberapa lokasi, jalan, dan kawasan yang belum tergambar pada peta Google, dengan begitu posisi koordinat pada GPS tidak terpresentasikan pada peta Google sehingga lokasi yang terdapat pada peta dan pada kondisi *real*-nya akan berbeda.

Dengan berbagai macam keterbatasan tersebut diharapkan aplikasi ini masih mampu memberikan informasi posisi keberadaan pengguna pada kawasan tertentu yang terdata oleh peta Google.

4.1.1. Defenisi Kebutuhan Sistem

Tahap ini memiliki beberapa tahapan yang akan dilaksanakan dalam pembuatan sistem, antara lain pengenalan masalah, pembuatan proposal, studi literatur, *project plan* dan pembuatan model.

Untuk tahapan pengenalan masalah, pembuatan proposal dan studi literatur sudah dibahas sebelumnya pada bab 1, bab 2, dan bab 3.

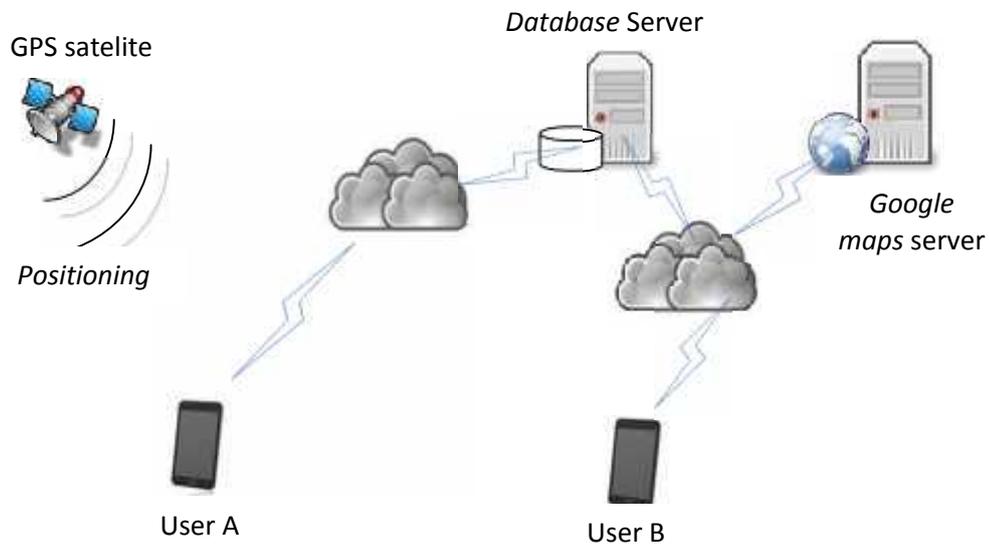
4.1.1.1. Project Plan

Pada *project plan* ini, akan dijelaskan mengenai gambaran umum sistem, deskripsi kebutuhan sistem, dan fungsi sistem yang nantinya akan dibuat.

A. Gambaran Umum Sistem

Aplikasi *mobile position* yang akan dibahas dalam penelitian ini adalah aplikasi yang berbasis GPS sebagai alat untuk mendapatkan nilai lokasi yang akan memberikan informasi posisi keberadaan perangkat *mobile* yang akan dipresentasikan pada *Google maps*. Bahasa pemrograman pada perangkat menggunakan Java sedangkan pada sisi *server* menggunakan PHP juga SQL untuk *query* data dari *database*. Aplikasi ini memiliki dua program utama yang saling bergantung satu dengan lainnya, aplikasi yang pertama digunakan oleh *phone* yang ingin di ketahui keberadaannya, program akan dirancang berjalan sebagai *background service* untuk memberikan nilai lokasi dalam durasi waktu tertentu secara berkala pada *server online* dan disimpan ke dalam *database*.

Aplikasi yang kedua dijalankan pada perangkat *mobile* yang ingin mengetahui posisi keberadaan perangkat *mobile* pertama, aplikasi khusus dibangun untuk mendapatkan nilai lokasi dari perangkat *phone* pertama yang telah tersimpan pada *database* online. Untuk lebih jelasnya deskripsi arsitektur sistem ini dapat di lihat pada gambar 4.1. dibawah ini.



Gambar 4.1. Arsitektur sistem.

Dari gambar 4.1 diatas dapat dilihat proses kerja aplikasi yang akan dibuat, terdiri dari *user A* dan *user B* yang masing-masing dipasangkan aplikasi yang berbeda sebagai berikut:

1. Aplikasi user A

Aplikasi berjalan sebagai *service* yang memiliki fungsi sebagai pengirim nilai koordinat melalui *httpconnection*, untuk mendapatkan nilai koordinat maka GPS sangat berperan penting untuk mendapatkan nilai posisi, kemudian nilai koodinat tersebut akan dikirimkan ke *server online* secara terus menerus berdasarkan rentang durasi waktu yang telah ditentukan.

2. Aplikasi user B

Aplikasi digunakan untuk mengambil nilai lokasi pada *server online* yang terhubung dengan koneksi internet sebagai data utama untuk menampilkan data lokasi terbaru dari *user A*, aplikasi inilah yang akan digunakan untuk mengetahui posisi *user A* pada *Google maps*.

B. Deskripsi Kebutuhan Sistem

Untuk membangun sebuah sistem yang efisien, kebutuhan sistem merupakan hal yang harus diperhatikan. Mengetahui kebutuhan sistem akan membantu dalam pembangunan sistem.

1. Sistem yang akan dibangun

Untuk kebutuhan sistem yang akan dibangun yakni kebutuhan sistem dari perangkat Android menggunakan bahasa pemrograman Java, pada kebutuhan perangkat ini akan di jelaskan analisa pemilihan ID pada perangkat pengguna.

Kebutuhan sistem pada perangkat Android adalah:

1. Bahasa pemrograman yang digunakan adalah *Java*.
2. Pada aplikasi *user B*, sebelum masuk ke halaman utama *user* harus terlebih dahulu melakukan otentifikasi pengguna dengan mengirimkan *username* dan *password*.
3. *User B* dapat melihat posisi terbaru dari *user A* dan mengetahui jarak tempuh dari *user A* dan *user B*.
4. *User B* dapat melakukan *track history* posisi pergerakan dari *user A* secara bersamaan di tampilkan pada *Google maps*.

2. Analisa pemilihan ID pengguna

Penggunaan nomor unik atau ID sebagai pengenalan utama pengguna sangat dibutuhkan sebagai pembeda antar pengguna, sehingga tidak terjadi ketimpangan data pada *system* dan *database*.

Analisa pemilihan ID yang dapat dipakai sebagai kode unik terdiri dari:

1. Nomor *handphone* pengguna.
Nomor *handphone* merupakan deretan digit angka unik yang tertanam pada setiap kartu *provider* pengguna.
2. Kode fisik *handphone* pengguna (IMEI).
IMEI (*International Mobile Equipment Identity*) adalah 15 kode unik. IMEI ini digunakan oleh GSM jaringan untuk mengidentifikasi perangkat valid.

Penggunaan nomor kartu *provider* sebagai ID akan mengalami masalah dikarenakan setiap perangkat *handphone* bisa berganti kartu yang jg akan merubah nomor *handphone* sedangkan nomor IMEI *handphone* tidak dapat diganti. Dari kedua kode unik tersebut yang paling memungkinkan di pakai sebagai ID pengguna adalah IMEI.

3. Batasan kebutuhan sistem

Aplikasi *mobile phone positioning* ini merupakan aplikasi yang berjalan dilingkungan perangkat bersistem operasi *Android* dengan menggunakan komponen *GPS internal*, *Google maps* dan layanan internet dari *provider*. Terdapat beberapa keterbatasan yang ditemui pada komponen perangkat ini, sehingga perlu diperhatikan untuk menjadi acuan dalam pengembangan aplikasi ini, yaitu diantaranya:

a. Perangkat android

1. Sumber daya yang terbatas, hingga saat ini perangkat *Android* yang banyak beredar memiliki kapasitas memori terbatas. Kapasitas memori terendah yang banyak beredar saat ini adalah 250Mb
2. Sumber daya baterai yang secara efektif hanya mampu bertahan selama kurang lebih 200 jam dalam keadaan *standby*.
3. Tampilan antar muka dengan pengguna sangat berpengaruh terhadap waktu tunggu aplikasi hingga aplikasi benar-benar siap digunakan, semakin banyak komponen yang digunakan akan semakin lama pula waktu tunggu yang dibutuhkan.

Dari keterbatasan-keterbatasan pada perangkat *Android*, maka diusulkan beberapa alternatif untuk meningkatkan performa aplikasi terhadap keterbatasan yang ada, diantaranya:

1. Merancang aplikasi yang menggunakan memori seefektif mungkin, sehingga tidak mengganggu siklus operasi *Android* dan aplikasi lain.
2. Merancang aplikasi dengan pemanfaatan sumber daya seefisien mungkin namun tidak mengurangi fungsi dan performa aplikasi.

3. Merancang aplikasi dengan antarmuka yang sederhana namun tetap menarik dan ramah bagi pengguna.

b. GPS internal

Penggunaan perangkat GPS internal pada *handphone* bersistem operasi Android memiliki keterbatasan dari sisi efektifitas pengiriman posisi, yaitu GPS internal harus selalu dalam keadaan aktif untuk dapat mengirimkan nilai posisi perangkat, sedangkan pada pengaturan GPS internal dapat dinonaktifkan sehingga pengiriman posisi pada perangkat *handphone* tidak berjalan.

c. Google maps

Google maps yang akan digunakan pada penelitian ini merupakan sumber pemetaan gratis yang diberikan oleh Google, layanan peta ini memiliki beberapa keterbatasan dimana tidak semua tempat atau lokasi yang terdata pada pemetaan lokasi, dengan keterbatasan ini terdapat beberapa tempat-tempat lokasi yang tidak tergambarkan sehingga akan memberikan informasi yang tidak akurat pada lokasi koordinat pada gambaran pemetaan *google maps*.

d. Layanan internet provider

Layanan internet *provider* untuk berkomunikasi dengan *server* database posisi online dibutuhkan agar data posisi GPS internal dapat di simpan, akan tetapi terdapat beberapa kendala layanan ini, yaitu apabila perangkat *handphone* berada pada lokasi yang tidak terjangkau oleh sinyal BTS provider sehingga dapat mengakibatkan data posisi tidak dapat dikirimkan melalui jalur internet.

4. Analisa fungsional

Analisa fungsional merupakan penjelasan mengenai fitur-fitur yang akan ada pada aplikasi *mobile positioning*. Fitur-fitur tersebut antara lain:

1. *User B* dapat melakukan *driving direction* terhadap *user A* untuk menentukan jalur tempuh dari lokasi *user B* ke *user A*

2. *User B* dapat melakukan *walking direction* terhadap *user A* untuk menentukan jalur tempuh pejalan kaki secara *turn-by-turn* dari lokasi *user B* ke *user A*.
3. Ketika *user B* meng-*click mark* (tanda) posisi pada *Google maps* maka aplikasi akan melakukan *pop-up* yang menampilkan keterangan dari mark tersebut.
4. *Line tracking history position* pada menu *tracking* untuk memberi garis rute pergerakan posisi *user A*.

C. Fungsi Sistem

Secara umum fungsi sistem yang akan dibangun dari sisi perangkat Android memiliki fungsi-fungsi sebagai berikut:

1. Menampilkan posisi user A terbaru dalam rentang waktu tertentu.
2. Menampilkan pengukuran jarak tempuh dari User A ke User B.
3. Menampilkan keterangan posisi lokasi dari user A.
4. Menampilkan pergerakan user A secara menyeluruh dengan *track history*.

4.2. Perancangan Sistem

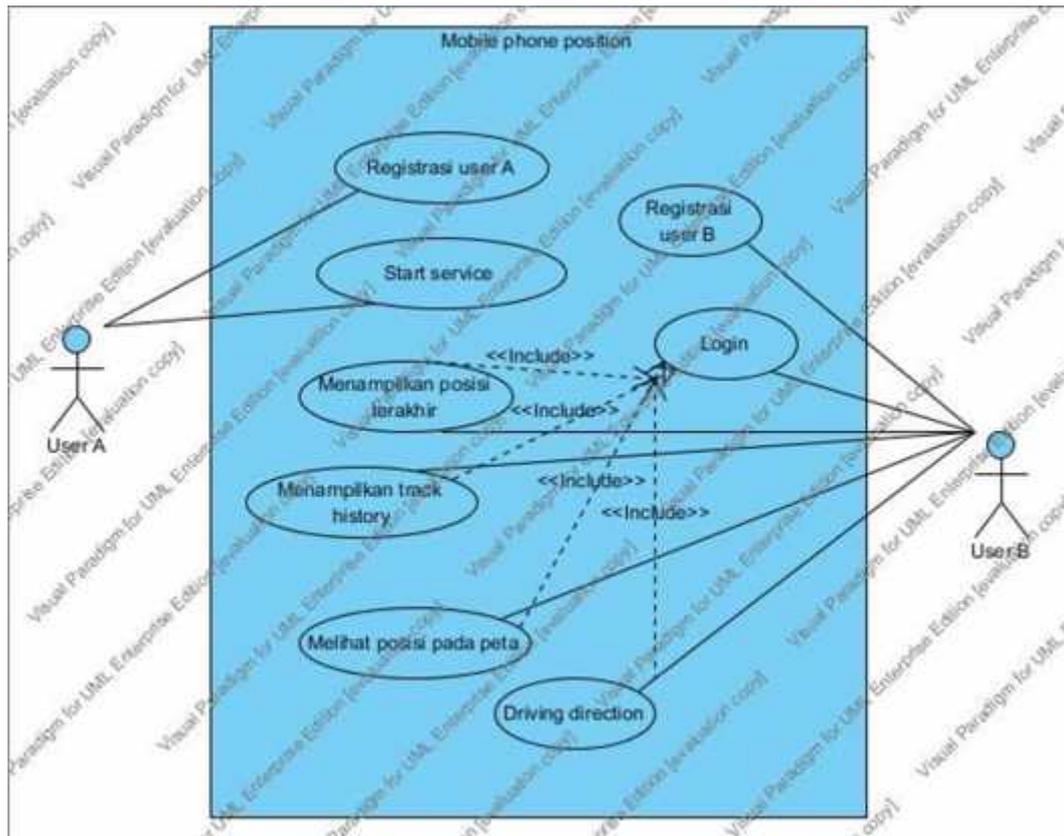
Setelah dilakukan beberapa tahapan dalam analisa sistem, maka dapat dilakukan beberapa perancangan aplikasi *mobile position* dengan GPS berbasis *platform* Android. Perancangan-perancangan yang akan dijelaskan dalam laporan ini terdiri dari perancangan model dalam bentuk UML (*Unified Modeling Language*) yaitu: *Usecase Diagram*, *Class Diagram*, *Activity Diagram*, *Sequence Diagram*. Selain itu juga ada perancangan *interface* sistem yang terdiri dari perancangan *prototype* dan struktur menu.

4.2.1. Pemodelan UML (*Unified Modeling Language*)

Analisa perancangan pada sistem ini menggunakan UML, yaitu terdiri dari *usecase diagram*, *class diagram*, *activity diagram* dan *sequence diagram*.

4.2.1.1. Usecase Diagram

Usecase yang akan dirancang yaitu *usecase diagram* untuk pengaksesan melalui perangkat Android. Gambar 4.2 dibawah ini menjelaskan aliran *usecase diagram* pengaksesan melalui perangkat Android.



Gambar 4.2. Aliran Use case diagram

Dari gambar 4.3 dapat dilihat sistem ini terdiri dari 2 aktor dan 7 *usecase*. Untuk lebih jelasnya, Spesifikasi dari *usecase diagram* (pengaksesan melalui perangkat Android) dapat di lihat pada tabel dibawah ini.

Tabel 4.1. Spesifikasi *use case* mendaftarkan user A.

Usecase	Mendaftarkan userA
Aktor Utama	User A

Deskripsi Singkat	Proses mengirim data user A ke server online
Kondisi Awal	Halaman menu user
Kondisi Akhir	User terdaftar di server.
<i>Main success scenario</i>	<ol style="list-style-type: none"> 1. <i>User A</i> memilih menu daftar user. 2. Tampil form daftar user 3. <i>User A</i> mengisi isian form dan meng-klik tombol submit. 4. Tampil pesan user terdaftar di server. 5. <i>User A</i> sukses terdaftar server.

Tabel 4.2. Spesifikasi *use case* start pengiriman posisi

Usecase	Start pengiriman posisi
Aktor Utama	User A
Deskripsi Singkat	Proses memulai pengiriman posisi user A secara berkala pada server online
Kondisi Awal	Halaman menu user
Kondisi Akhir	Service berjalan.
<i>Main success scenario</i>	<ol style="list-style-type: none"> 1. <i>User A</i> memilih menu <i>start service</i>. 2. Tampil list menu <i>stop service</i>. 3. <i>Service</i> berjalan secara background.

Tabel 4.3. Spesifikasi Mendaftarkan User B

Usecase	Mendaftarkan UserB
Aktor Utama	User B
Deskripsi Singkat	Proses pengiriman data user B ke server
Kondisi Awal	Halaman login
Kondisi Akhir	Berhasil mendaftar ke server
<i>Main success scenario</i>	User B memilih tombol daftar pengguna, form pengisian data tampil, pengguna mengisi data yang dibutuhkan dan mengklik tombol simpan, data b
<i>Alternate</i>	Jika input text isian masih kosong maka sistem akan menampilkan peringatan.

Tabel 4.4. Spesifikasi *use case* Log in

Usecase	<i>Log in</i>
Aktor Utama	<i>User B</i>
Deskripsi Singkat	Sistem memverifikasi pengguna untuk masuk ke aplikasi
Kondisi Awal	Sistem menampilkan <i>form login</i> sistem
Kondisi Akhir	Berhasil masuk ke sistem
<i>Main success scenario</i>	<i>User B</i> memasukkan inputan <i>username</i> dan <i>password</i> kemudian menu utama akan ditampilkan.
<i>Alternate</i>	Jika kombinasi <i>username</i> dan <i>password</i> salah atau inputan isian masih kosong maka sistem akan memberikan peringatan.

Tabel 4.5. Spesifikasi *use case* menampilkan posisi terakhir.

Usecase	Menampilkan posisi terakhir
Aktor Utama	User B
Deskripsi singkat	Proses menampilkan posisi lokasi terakhir user A
Kondisi Awal	Halaman utama
Kondisi Akhir	Tampil rincian posisi terakhir user A.
<i>Main success scenario</i>	<ol style="list-style-type: none"> 1. <i>User B</i> memilih menu posisi M. 2. Tampil <i>list</i> anggota user A. 3. <i>User B</i> memilih salah satu anggota. 4. Tampil detail posisi anggota yang dipilih.

Tabel 4.6. Spesifikasi *use case* menampilkan *track history*.

Usecase	Track history
Aktor Utama	<i>User B</i>
Deskripsi singkat	Proses menampilkan keseluruhan pergerakan User A
Kondisi Awal	Halaman utama

Kondisi Akhir	Tampil list <i>track history</i> .
<i>Main success scenario</i>	<ol style="list-style-type: none"> 1. <i>User B</i> memilih menu <i>track history</i>. 2. Tampil list anggota. 3. <i>User B</i> memilih salah satu anggota. 4. Tampil list daftar semua posisi user.

Tabel 4.7. Spesifikasi *use case* menampilkan posisi user A pada peta.

Usecase	Melihat posisi pada peta
Aktor Utama	<i>User B</i>
Deskripsi singkat	Proses menampilkan posisi lokasi user A dan user B pada peta Google
Kondisi Awal	Halaman utama.
Kondisi Akhir	Tampil posisi user A pada <i>map google</i> .
<i>Main success scenario</i>	<ol style="list-style-type: none"> 1. <i>User B</i> memilih menu posisi M. 2. Tampil list anggota. 3. <i>User B</i> memilih salah satu anggota. 4. Tampil detil posisi user A. 5. <i>User B</i> menekan tombol lihat peta. 6. Posisi user A tampil pada <i>Google maps</i>

Tabel 4.8. Spesifikasi *use case driving direction* peta.

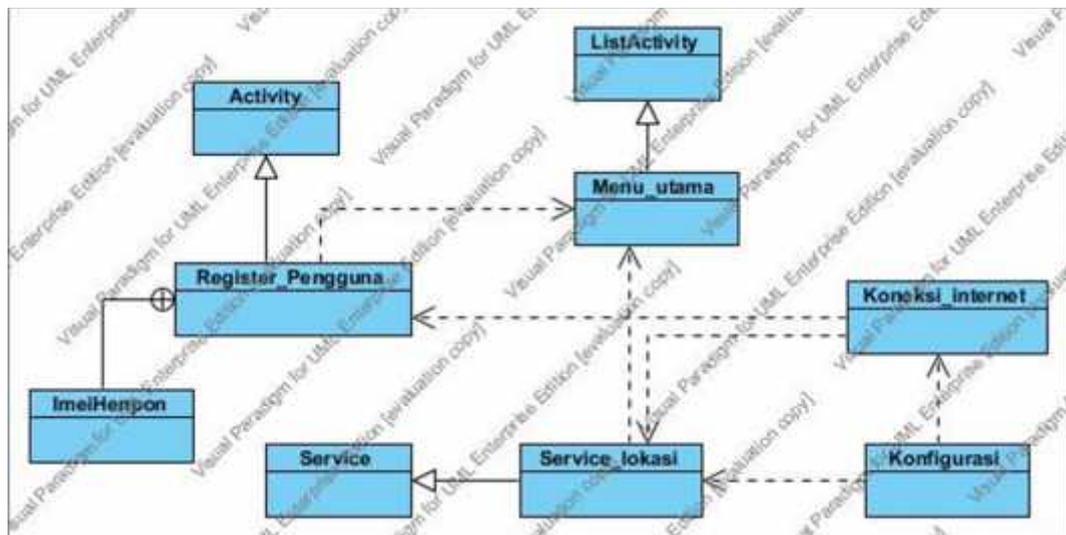
Usecase	<i>Driving direction</i>
Aktor Utama	<i>User B</i>
Deskripsi singkat	Proses menampilkan rute jalan dari lokasi user A ke user B pada peta Google
Kondisi Awal	Detil posisi user A tampil
Kondisi Akhir	Tampil rute jalan dari user A ke user B pada <i>map google</i>
<i>Main success scenario</i>	User B memilih menu lihat peta kemudian tampil <i>Google maps</i> dengan marker posisi <i>user A</i> dan posisi <i>user B</i> , User B memilih menu <i>driving direction</i> , kemudian <i>driving direction</i> ditampilkan.

4.2.1.2. Class Diagram

Class Diagram menggambarkan struktur dan deskripsi *class*, *package*, dan objek yang saling terhubung. Pada class diagram ini dibedakan menjadi dua bagian yaitu *class diagram user A* dan *class diagram user B* sebagai berikut:

1. Class Diagram User A

Class diagram yang dijelaskan pada analisa ini adalah *class diagram* sistem yang terpasang pada perangkat Android *user A*. Gambar 4.3 dibawah ini menjelaskan *class diagram* sistem yang terpasang pada perangkat Android *user A*.



Gambar 4.3. Class diagram sistem pada user A

Berikut detail masing-masing *class* yang berisi nama *class*, *atribut*, dan *method*, yang dapat di lihat pada tabel 4.3 di bawah ini.

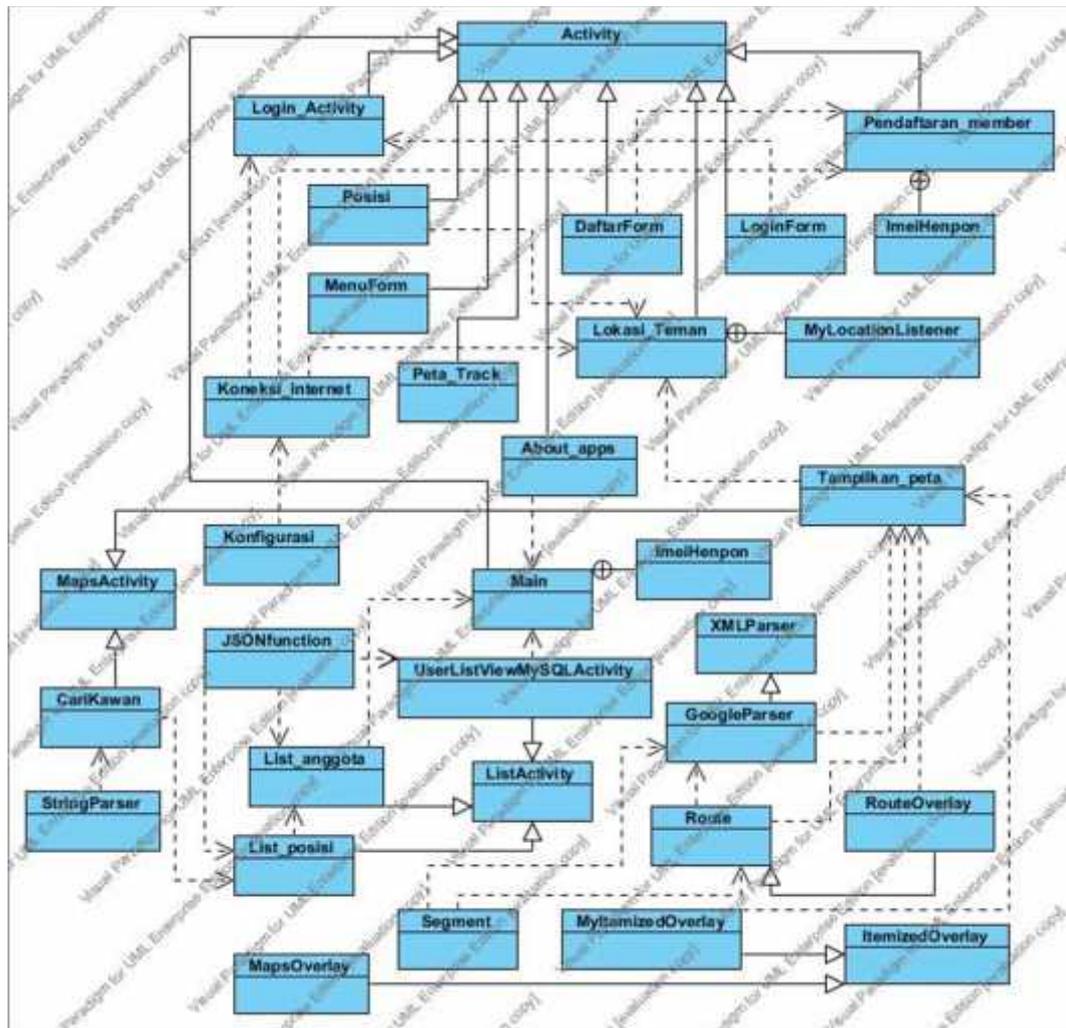
Tabel 4.9. Detail *class diagram* aplikasi *user A*

No	Class	Atribut	Method
1	Menu_utama	-	- onCreate(Bundle savedInstanceState) : void - onStart():void - onItemClick(ListView l, View v, int position, long id):void

			<ul style="list-style-type: none"> - tampilkanPilihan(String pilihan):void - isMyServiceRunning():boolean - createMenu(boolean check):void
2	Konfigurasi	<ul style="list-style-type: none"> - URL : String - WAKTU : int 	-
3	Koneksi_internet	<ul style="list-style-type: none"> - Text : String 	<ul style="list-style-type: none"> - Test():String - postData(JSONObject json, String queryString):String
4	Register_pengguna	<ul style="list-style-type: none"> - btnSimpan: Button - btnBatal :Button - imei : TextView - user : TextView - imeiTujuan :TextView - imeiKu : String - context: Context 	<ul style="list-style-type: none"> - onCreate(Bundle savedInstanceState): void - reset():void - ImeiHenpon(Context context) - getImeiNumber():String
5	Service_lokasi	<ul style="list-style-type: none"> - lm : LocationManager - locationListener : <li style="padding-left: 20px;">LocationListener - lon : String - lat : String 	<ul style="list-style-type: none"> - onBind(Intent intent): Ibinder - onCreate(): void - onDestroy():void - onStart(Intent intent, int startid):void

2. Class Diagram User B

Class diagram yang dijelaskan pada analisa ini adalah *class diagram* sistem yang terpasang pada perangkat Android *user B*. Gambar 4.4. dibawah ini menjelaskan *class diagram* sistem yang terpasang pada perangkat Android *user B*.



Gambar 4.4. Class diagram pada user B.

Berikut detail masing-masing *class* yang berisi nama *class*, *atribut*, dan *method*, yang dapat di lihat pada tabel 4.8 di bawah ini.

Tabel 4.10. Detail *class diagram* aplikasi user B.

No	Class	Attribut	Method
1	Main	- MenuForm form - Context contex	- onCreate(Bundle savedInstanceState): void - onCreateOptionsMenu(Menu menu) : boolean - onClick : void
2	MenuForm	- activity : Activity - buttonPosisi : Button	- getActivityActivity() - getbuttonPosisi();

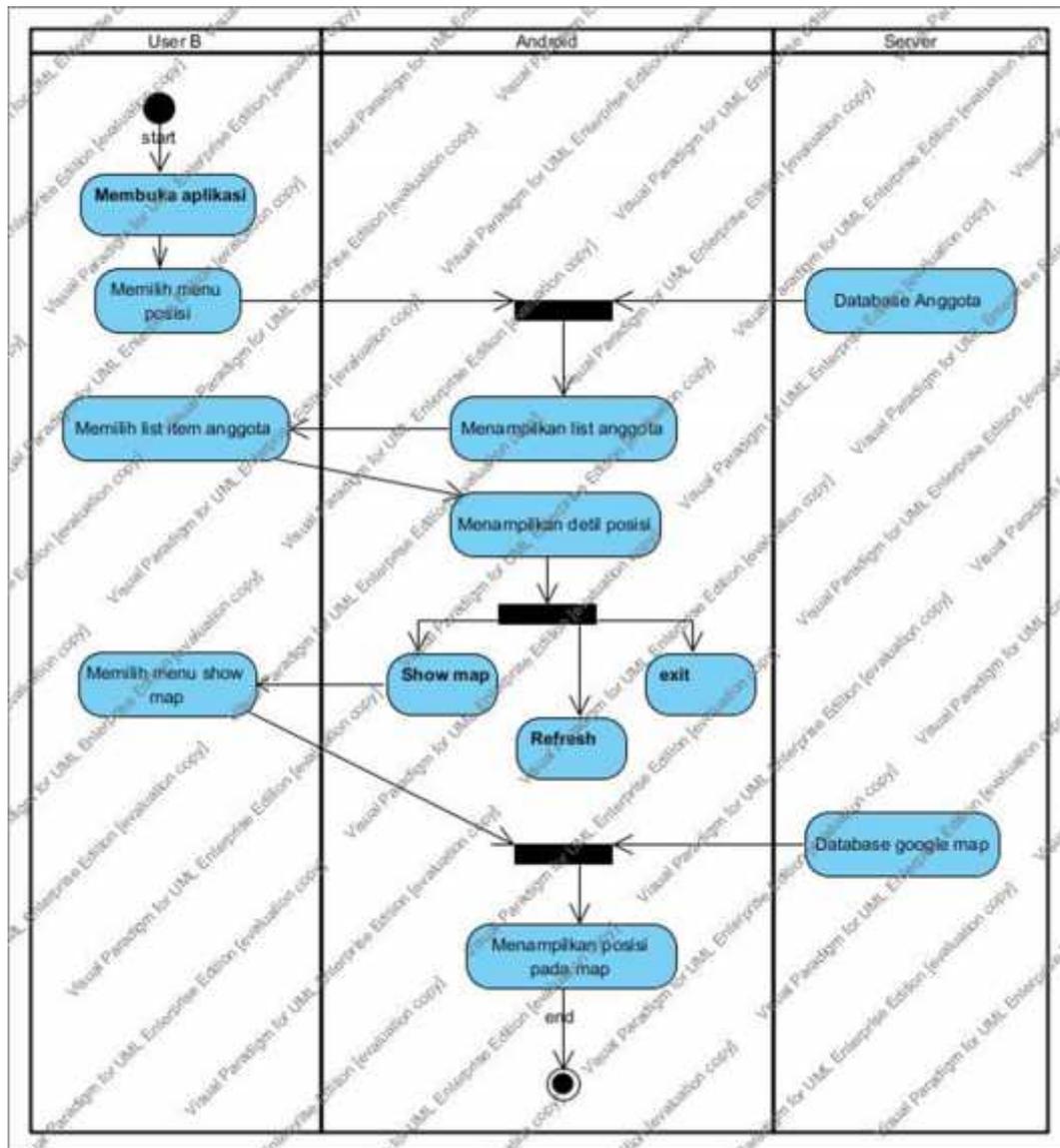
		<ul style="list-style-type: none"> - buttonTrack: Button - buttonAbout : Button - buttonExit: Button 	<ul style="list-style-type: none"> - getbuttonTrack(); - getbuttonAbout() - getbuttonExit()
3	About_apps	-	- onCreate(Bundle savedInstanceState): void
4	UserListViewMySql Activity	-	<ul style="list-style-type: none"> - onCreate(Bundle savedInstanceState): void - onItemClick(AdapterView<?>parent, View view, int Position, Long id) : void
5	Lokasi_Teman	<ul style="list-style-type: none"> - lat : String - lon : String - nama_teman : String - locationManager : LocatioManager - yourLat: String - yourLon : String - jarak : String - textlokasi :TextView - textlokasisaya : TextView - posisi : Posisi 	<ul style="list-style-type: none"> - onCreate(Bundle savedInstanceState): void - updateposisi: void - onproviderDisabled(String provider): void - onProvideEnabled(String provider) : void - onStatusChanged(String provider, int status, Bundle xtras): void
6	JSONFunction	<ul style="list-style-type: none"> - is :InputStream - result : String - jArray : JSONObject 	- getJSONFromURL(String url)
7	Koneksi_internet	- text : String	<ul style="list-style-type: none"> - test() :String - postData(JSONObject json, String): String
8	Posisi	<ul style="list-style-type: none"> - activity : Activity - btnLihatPeta: Button - buttonRefresh: Button - btnExit: Button 	<ul style="list-style-type: none"> - Posisi(Activity activity) - getActivity - getBtnLihatPeta() - getBtnRefresh() - getBtnExit()
9	Tampilkan_Peta	- mapView : Mapview	<ul style="list-style-type: none"> - onCreate(Bundle savedInstanceState) : void - tampilkanMarker() : void - isRouteDisplayed(): boolean
10	Konvigurasi	<ul style="list-style-type: none"> - URL : String - WAKTU : int 	-
11	MyItemOverlay	<ul style="list-style-type: none"> - items : ArrayList<OverlayItem> - context : Context 	<ul style="list-style-type: none"> - MyItemezeOverlay (Drawable default Marker, Context context) - OverlayItem createItem(int index);

		- marker : Drawable	- Size() : Int - draw(Canvas canvas, Mapview mapview, boolean shadow) : void - addItem(overlayItem item) : void - onTop(int index) : Boolean
12	DaftarForm	- activity: Activity - btnSubmit: Button - btnBatal: Button - imeiDaftar:EditText - namaDaftar:EditText - usernameDaftar:EditText - passwordDaftar:EditText	- DaftarForm(Activity activity) - getActivity():Activity - getBtnSubmit():Button - getBtnBatal():Button - getImeiDaftar():EditText - getNamaDaftar()EditText - getUsernameDaftar():EditText - getPasswordDaftar():EditText
12	LoginForm	- btnLogin:Button - btnDaftar:Button - txtusername:EditText - textpassword:EditText - texterror:TextView - activity: Activity	- LoginForm(Activity activity) - getActivity():Activity - getBtnLogin():Button - getBtnDaftar():Button - getTextusername():EditText - getTextpassword():EditText
13.	Login_Activity	- form: LoginForm	- onCreate(Bundle savedInstanceState) : void - onClick(View v):void
14	Pendaftaran_Member	- form: DaftarForm - imei_member: String - context : Context	- onCreate(Bundle savedInstanceState) : void - onClick(View v):void - reset(): void - ImeiHenpon(Context_context) - getImeiNumber():String
14	UserListViewMySQLActivity	-	- onCreate(Bundle savedInstanceState) : void
15	List_anggota	-	- onCreate(Bundle savedInstanceState) : void
16.	List_Posisi	-	- onCreate(Bundle savedInstanceState) : void
17	GoogleParser	- distance: int	- GoogleParser(String feedUrl) - Parse() : Route - convertStreamtoString(final InputStream input):String

			- decodePolyline(final String poly):List<Geopoint>
18	XMLParser	- feedUrl : URL	- XMLParser(final String feedUrl) - InputStream getInputStream()
19	Route	- Points: List<GeoPoint> - Name: String - Segment:List<Segment> - Copyright: String - Warning :String - Country:String - Length:int - Polyline: String	- Route() - addPoint(final GeoPoint p):void - addPoints(final List<GeoPoint> point):void - getPoint():List<GeoPoint> - addSegment(final Segment s): void - getSegment()List<Segment> - setName(final String name): void - getName(): String - setCopyright(String Copyright): void - getCopyright():String - setWarning(String Warning): void - getWarning(): String - setCountry(String Country)
20	CariKawan	- mapView : MapView - latitude: double - logintude: double - saveName: String [] - saveLatitude: String[] - saveLongitude: String[] - locationmanager: LocationManager	- onKeyDown(int keyCode, KeyEvent event) :boolean - isRouteDisplay(): boolean - setPosition(double latitude, double longitude) : void - Call(String url): String - openHttpConnection(String urlString): inputStream - onCreate(Bundle savedInstanceState):void - onResume():void - onPause(): void - onProviderDisabled(String provider): void - onProviderEnabled(String provider): void - onLocationChanged(Location location): void
21	StringParser	-	- StringParser() - ArrayList<Object>Parse(String input);

4.2.1.3. Activity Diagram

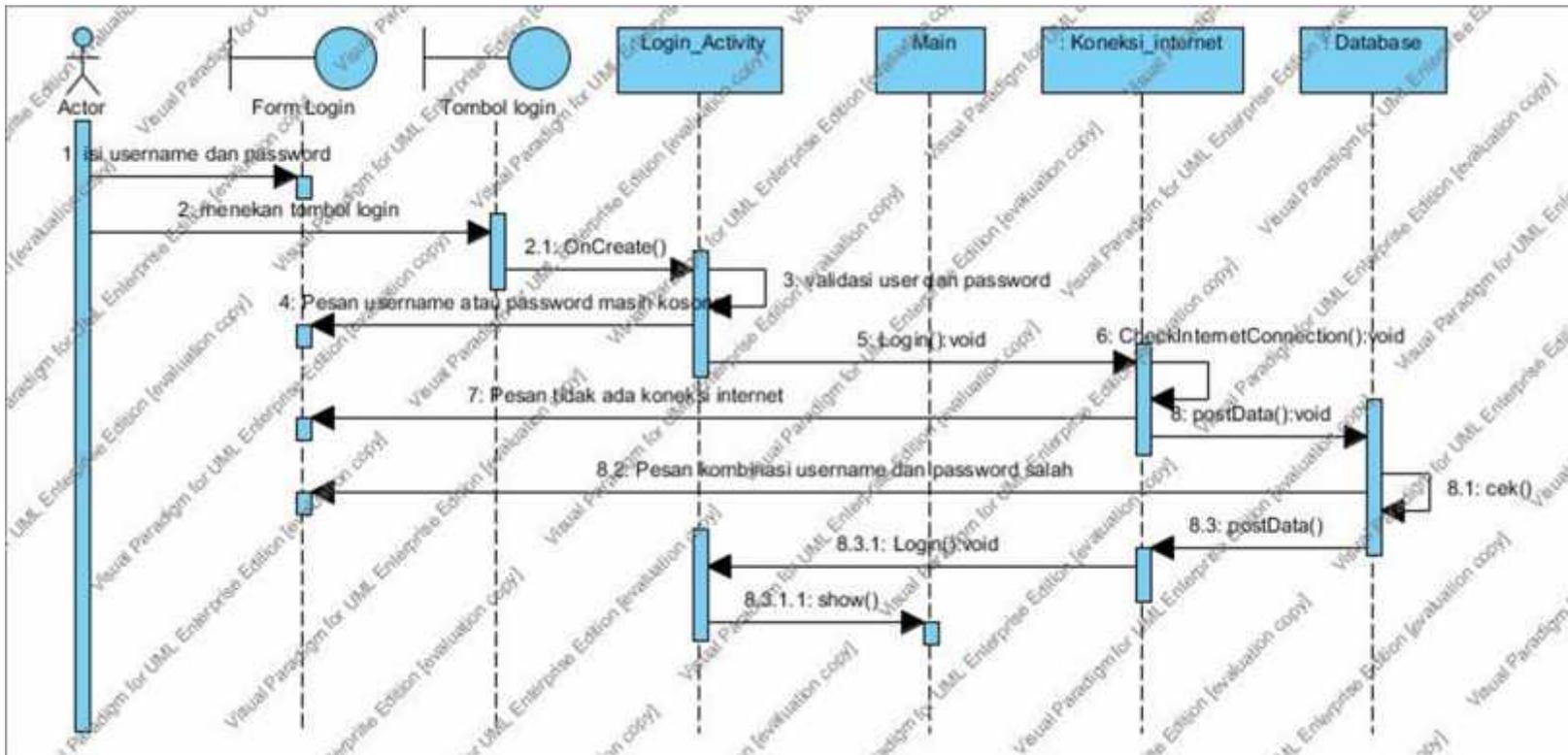
Activity diagram merupakan alur kerja pada setiap *usecase*. *Activity diagram* pada analisa ini mencakup *activity diagram* setiap *usecase*. Gambar 4.5. dibawah ini menjelaskan *activity* menampilkan poisisi user A dan user B pada peta secara bersamaan pada perangkat Android user B.



Gambar 4.5. Activity diagram menampilkan posisi A pada peta

4.2.1.4. Sequence Diagram

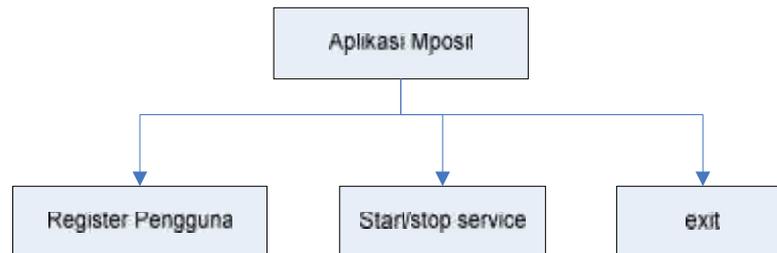
Sequence Diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display* dan sebagainya) berupa pesan yang digambarkan terhadap waktu. Gambar 4.7 dibawah ini menggambarkan *sequence diagram login* pada *server*. Untuk *Sequence diagram* lainnya dapat di lihat pada lampiran A.



Gambar 4.6. Sequence diagram untuk Log in

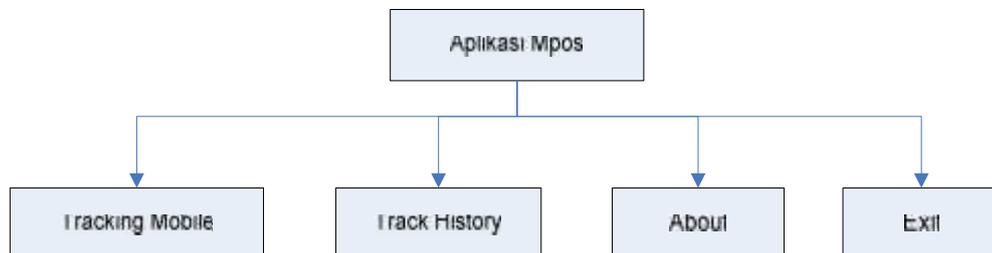
4.3. Rancangan Struktur Menu Sistem

Rancangan struktur menu merupakan tahapan untuk merancang bagaimana struktur menu yang akan dibangun. Berikut struktur menu dari sistem yang akan dibangun pada *user A* dapat dilihat pada gambar 4.7 di bawah ini.



Gambar 4.7. Rancangan Struktur menu di Android *user A*

. Berikut struktur menu dari sistem yang akan dibangun pada perangkat *user B* dapat dilihat pada gambar 4.8 di bawah ini.



Gambar 4.8. Rancangan Struktur menu di Android *user B*

4.4. Perancangan Antarmuka Pengguna Sistem

Rancangan antar muka pengguna sistem berfungsi sebagai landasan awal dalam merancang tampilan antar muka sistem. Pada analisa dan perancangan ini, antar muka untuk pengguna sistem pada perangkat Android antara lain, antarmuka Menu Utama, antar muka list anggota, pendaftaran anggota, detil posisi, lihat peta, track anggota, list menu user A, track history, antarmuka menu *about*. Gambar 4.9 di bawah ini menjelaskan perancangan antar muka *Home* pada perangkat Android.

Gambar 4.9. Perancangan menu daftar *user A*.

Deskripsi gambar 4.9. tentang perancangan antar muka pendaftaran anggota pada perangkat Android dapat dilihat pada tabel 4.11 di bawah ini.

Tabel 4.11. Deskripsi antar muka pendaftaran anggota pada perangkat Android A

No	Nama <i>Item</i>	Deskripsi
1.	<i>Imei mobile</i>	Merupakan sebuah <i>widget</i> berupa <i>EditText</i>
2.	Nama Pengguna	Merupakan sebuah <i>widget</i> berupa <i>EditText</i>
3.	<i>Imei Tujuan</i>	Merupakan sebuah <i>widget</i> berupa <i>EditText</i>
4.	<i>Submit</i>	Merupakan sebuah <i>widget</i> berupa <i>Button</i>
5.	<i>Cancel</i>	Merupakan sebuah <i>widget</i> berupa <i>Button</i>

BAB V

IMPLEMENTASI DAN PENGUJIAN

Bab ini menjelaskan tentang implementasi dan pengujian fungsi-fungsi aplikasi dengan metode *blackbox*, serta juga dilakukan pengujian aplikasi terhadap beberapa perangkat Android yang berbeda, kemudian dilakukan pengamatan dari hasil pengujian tersebut untuk mengetahui kekurangan aplikasi, juga dilakukan pengambilan kesimpulan.

5.1. Implementasi

Tahapan implementasi merupakan tahapan aplikasi yang telah dirancang, dianalisis, dan dibangun, lalu diuji kelayakannya untuk selanjutnya dioperasikan sebagaimana mestinya sesuai dengan fungsi dan kelayakannya. Berikut ini akan dijelaskan tentang implementasi dari analisis dan perancangan yang telah dilakukan terhadap aplikasi *mobile position* dengan GPS menggunakan sistem operasi Android.

5.1.1. Batasan Implementasi

Berikut adalah batasan implementasi pada pengembangan aplikasi *mobile position* dengan GPS menggunakan sistem operasi Android.

1. Sistem dibangun dengan menggunakan bahasa pemrograman Java.
2. Menggunakan *Database Server MySQL*.
3. Menggunakan *Database Server Apache*.
4. Lokasi pengujian adalah kota Pekanbaru
5. Waktu yang digunakan adalah waktu pada server *database*.
6. Waktu pengiriman posisi *user A* dalam rentang waktu 15 menit.

Untuk mengimplementasikan sistem ini dibutuhkan perangkat pendukung, perangkat tersebut berupa komponen perangkat keras dan perangkat lunak.

1. Perangkat keras

Processor	: <i>Intel(R) Pentium(R) CPU P6000 @1.87Ghz</i> (2 CPUs), ~1.9Ghz
Memori (RAM)	: 1024MB RAM
Kartu GSM	: Exis

2. Perangkat Lunak

Sistem Operasi	: <i>Windows 7 Enterprise 32-bit Operating System</i>
Bahasa Pemrograman	: Java dan PHP
Tools Pengembangan	: Eclipse Juno 4.2.1, Notepad ++ 5.0.2 . : <i>Java Development Kit 1.7.0</i> : <i>Android SDK, ADT 14.0.0</i> : <i>Android Virtual Device 2.2 (Froyo)</i> : <i>Google map API v2</i>
Browser	: <i>Mozilla firefox 14.0.1</i>
Server	: XAMPP (Apache 2, MySQL, PhpMyAdmin)
Pemodelan UML	: Visual paradigm v.9.0

5.1.2. Lingkungan Implementasi

Untuk lingkungan implementasi aplikasi ini dilakukan dengan menggunakan perangkat keras dan perangkat lunak diantaranya:

1. Perangkat keras : Samsung Galaxy young
2. Perangkat lunak : Sistem operasi Android Gingerbread (2.3.6).

Dalam implementasi aplikasi ini dipasang di sebuah penyedia layanan *hosting* dengan domain <http://kuliah62.pusku.com> untuk mengaksesnya menggunakan *mozilla firefox*.

5.1.3. Tahap-tahap Implementasi aplikasi Mobile position

A. Instalasi Aplikasi User A

Seperti yang telah dijelaskan pada bab analisa dan perancangan, sistem ini terdiri dari dua aplikasi yang saling membutuhkan yaitu sistem *user A* untuk mengirimkan koordinat dan sistem *user B* untuk menampilkan posisi koordinat

berdasarkan data posisi yang dikirimkan oleh *user A* ke *server*. Untuk memberikan identitas sebagai ID, sistem pada *user A* harus terlebih dahulu mendaftarkan nomor *Imei*. Untuk lebih jelasnya dapat dilihat pada gambar 5.1 dibawah .



Gambar 5.1. Form mendaftarkan imei user.

Form pendaftaran user ini dibangun menggunakan bahasa pemrograman Java, yang terkoneksi dengan bahasa pemrograman PHP untuk melakukan proses penyimpanan ke *server* MySQL. Gambar 5.2 di bawah ini merupakan *source code* yang berfungsi untuk melakukan penyimpanan data *user A*

```
function simpan_user(){
    $json = $_SERVER['HTTP_JSON'];
    $data = json_decode($json);
    $imei = $data->imei_user;
    $user = $data->username;

    $hasil = database("INSERT INTO user (imei_user,username)VALUES('$imei','$user')");
    if ($hasil==1) echo "Pengguna berhasil di daftarkan!";
}
```

Gambar 5.2. function simpan_user pada index.php

B. Instalasi Aplikasi User B

Implementasi untuk user B berjalan dengan baik pada perangkat Android, implementasi dilakukan pada sistem operasi Android Gingerbeard (2.3.6). Hasil implementasi pada menu utama menampilkan menu aplikasi yang terdapat pada aplikasi *mobile position user B*



Gambar 5.2. Hasil implementasi menu utama.

5.2. Pengujian Aplikasi

5.2.1. Pengujian *Blackbox* Aplikasi *mobile position (user A)*

Tahapan pengujian dilakukan untuk mengetahui apakah sistem yang dibangun telah sesuai dengan yang di harapkan. Pengujian yang akan dilakukan adalah pengujian fungsionalitas sistem menggunakan metode *blackbox*, pada tahap pengujian sistem ini perangkat keras yang digunakan yaitu *smartphone* Samsung Galaxy y.

Table 5.1. Pengujian *aplikasi mobile position user A* dengan metode *Blackbox*

No.	Nama Pengujian	Deskripsi	Perangkat Android	Prekondisi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Hasil yang Didapat	Kesimpulan
1.	Menampilkan menu user A	Pengujian untuk menampilkan menu utama <i>mobile position</i> pada user A	Samsung galaxy y	-	Klik logo aplikasi pada perangkat Android	-	Tampil daftar menu pada aplikasi user A	Tampil daftar menu utama	Benar
2.	Menu form daftar user A	Pengujian dilakukan untuk mendaftarkan imei user A pada database MySQL server	Samsung galaxy y	Menu <i>user A</i> tampil	Klik menu daftar user pada aplikasi user A	Pada isian nama di inputkan "jajat"	Imei berserta nama inputan tersimpan pada database MySQL server	Data tersimpan pada database MySQL server	Benar
3	Menjalankan service posisi.	Pengujian dilakukan untuk menjalankan service posisi.	Samsung galaxy y	Menu <i>user A</i> tampil	Klik menu <i>Start service</i> pada daftar menu aplikasi <i>user A</i>	-	Service berjalan secara background untuk mengirimkan koordinat posisi secara berkala	Service berjalan pada background dan nilai koordinat user A dikirimkan ke database MySQL server	Benar

5.2.2. Pengujian *Blackbox* Aplikasi *mobile position* (user B)

Pengujian menggunakan metode *blackbox* juga dilakukan pada aplikasi user B, hasil pengujian ditampilkan pada tabel 5.2 sebagai berikut:

Table 5.2. Pengujian aplikasi *mobile position user B* dengan metode *Blackbox*

No.	Nama Pengujian	Deskripsi	Perangkat Android	Prekondisi	Prosedur Pengujian	Masukan	Keluaran yang diharapkan	Hasil yang Didapat	Kesimpulan
1.	Menampilkan daftar menu utama <i>mobile position</i> user B	Pengujian untuk menampilkan menu utama <i>mobile position</i> pada user B	Samsung galaxy y	-	Klik logo aplikasi pada perangkat Android	-	Tampil daftar menu pada aplikasi user B	Tampil daftar menu utama	Benar
2.	Menampilkan detail posisi <i>user A</i> pada aplikasi <i>user B</i>	Pengujian untuk menampilkan detail posisi <i>user A</i> terhadap <i>user B</i>	Samsung galaxy y	Daftar pengguna yang ingin dilihat detail posisi tampil	Pilih salah satu anggota user A yang tersedia pada <i>listview</i> anggota user A	-	Tampil detail posisi user A terhadap user B beserta jarak antara kedua posisi.	Tampil detail posisi user A terhadap user B	Benar
3.	Menampilkan posisi <i>user A</i> terhadap <i>user B</i> pada <i>map Google</i>	Pengujian dilakukan untuk menampilkan posisi <i>user A</i> terhadap user B pada <i>map Google</i>	Samsung galaxy y	Detail posisi user A terhadap user B tampil beserta menu untuk lihat peta.	Klik tombol lihat peta yang tersedia pada detail posisi user A	-	Tampil posisi user A terhadap user B pada <i>map Google</i>	Tampil user A pada <i>map Google</i>	Benar
4.	Menampilkan <i>driving direction</i>	Pengujian untuk menampilkan <i>driving direction</i> antara <i>user A</i> terhadap <i>user B</i>	Samsung galaxy y	Tampil <i>user A</i> dan <i>user B</i> pada <i>map Google</i> beserta menu <i>driving direction</i>	Klik tombol <i>driving direction</i> yang tersedia pada <i>map Google user B</i>	-	Tampil <i>driving direction</i> antara <i>user A</i> terhadap <i>user B</i>	Tampil <i>driving direction</i> antara <i>user A</i> ke arah <i>user B</i>	Benar
5.	<i>Track history</i>	Pengujian untuk menampilkan	Samsung galaxy y	Daftar <i>listview</i>	Pilih salah satu anggota	-	Tampil <i>track history</i>	Tampil <i>track history</i>	Benar

		<i>history</i> posisi user A		anggota user A tampil setelah memilih menu <i>track history</i>	pada <i>listview</i>				
6.	Menampilkan semua posisi di google map	Pengujian digunan untuk menampilkan semua posisi track anggota di <i>google map</i>	Samsung galaxy y	Daftar <i>listview</i> posisi user A tampil setelah memilih salah satu anggota <i>user A</i>	Klik tombol tampil peta.		Tampil semua posisi anggota yang dipilih pada <i>google map</i>	Tampil semua posisi pada <i>google map</i>	Benar
7.	About aplikasi	Pengujian untuk menampilkan keterangan aplikasi	Samsung galaxy y	Daftar menu utama tampil	Klik menu about pada daftar menu	-	Tampil halaman about	Tampil halaman about	Benar
8.	Exit	Pengujian digunakan untuk menampilkan pesan keluar.	Samsung galaxy y	Daftar menu utama tampil	Pilih menu exit pada daftar menu	-	Tampil pesan iformasi exit	Tampil pesan informasi dialog exit	Benar

5.2.3. Pengujian Akses Aplikasi *Mobile Position*

Pengujian yang dilakukan bertujuan untuk mengetahui proses hasil dari sistem, yaitu memperlihatkan aplikasi *mobile position* yang telah dibangun bisa diakses melalui berbagai perangkat Android dan berbagai versi Android. Hasil dari pengujian dapat dilihat halaman menu utama telah tampil, dan semua menu serta fitur dapat digunakan. Kemudian pembuktian dilakukan untuk membuktikan bahwa konten yang ditampilkan di sistem sesuai dengan *database server*. Untuk alamat *website* yang digunakan *mobile position* untuk Android adalah *http://kuliah62.pusku.com*.

Pada tabel 5.3 di bawah ini menjelaskan pengujian akses ke aplikasi *mobile position* menggunakan beberapa perangkat bersistem operasi Android.

Tabel 5.3 Pengujian akses ke aplikasi *mobile position* dari *device* Android aplikasi *user B*.

Tanggal – Pukul	Perangkat Android	Versi Android	Provider	Status GPS	Pengujian	Hasil	Ket
8 April 2013 – 7.17 PM	Samsung Galaxy y	Gingerbread .DXLB1 (2.3.6)	Exis	Aktif	Koneksi ke <i>server</i>	Berhasil	-
					Daftar <i>user</i>	Berhasil	Data tersimpan
					Login aplikasi	Berhasil	Sesuai <i>database</i>
					Menu utama	Berhasil	Halaman utama tampil
					List anggota	Berhasil	Sesuai <i>database</i>
					Posisi terakhir	Berhasil	Sesuai <i>database</i>
					Tampil peta	Berhasil	Posisi pada peta sesuai <i>database</i>
					List Tracking posisi anggota	Berhasil	Sesuai <i>database</i>
					Tracking posisi pada peta	Berhasil	Sesuai <i>database</i>
7 April 2013 – 11.00 AM	Samsung Galaxy Mini	Froyo (2.2)	Tri	Aktif	Koneksi ke <i>server</i>	Berhasil	-
					Daftar <i>user</i>	Berhasil	Data tersimpan
					Login aplikasi	Berhasil	Sesuai <i>database</i>
					Menu utama	Berhasil	Halaman utama tampil
					List anggota	Berhasil	Sesuai <i>database</i>
					Posisi terakhir	Berhasil	Sesuai <i>database</i>
					Tampil peta	Berhasil	Posisi pada peta sesuai <i>database</i>
					List Tracking posisi anggota	Berhasil	Sesuai <i>database</i>
					Tracking posisi pada peta	Berhasil	Sesuai <i>database</i>

10 April 2013 – 7.17 PM	smart friend andro	Gingerbread .DXLB1 (2.3.6)	<i>Smartfri end</i>	Aktif	Koneksi ke <i>server</i>	Berhasil	-
					Daftar <i>user</i>	Berhasil	Data tersimpan
					<i>Login</i> aplikasi	Berhasil	Sesuai <i>database</i>
					Menu utama	Berhasil	Halaman utama tampil
					List anggota	Berhasil	Sesuai <i>database</i>
					Posisi terakhir	Berhasil	Sesuai <i>database</i>
					Tampil peta	Berhasil	Posisi pada peta sesuai <i>database</i>
					<i>List Tracking</i> posisi anggota	Berhasil	Sesuai <i>database</i>
<i>Tracking</i> posisi pada peta	Berhasil	Sesuai <i>database</i>					

Pada tabel 5.4 di bawah ini menjelaskan pengujian akses ke aplikasi *mobile position* pada *user A* menggunakan beberapa perangkat bersistem operasi Android.

Tabel 5.4 Pengujian akses ke aplikasi *position* dari *device* Android pada aplikasi *user A*.

Tanggal – Pukul	Perangkat Android	Versi Android	<i>Provider</i>	Status GPS	Pengujian	Hasil	Ket
8 April 2013 – 7.17 PM	Samsung Galaxy y	Gingerbread .DXLB1 (2.3.6)	Exis	Aktif	Koneksi ke <i>server</i>	Berhasil	-
					Daftar <i>user</i>	Berhasil	Data tersimpan
					Menu utama Jalankan <i>service</i> lokasi	Berhasil	Data posisi terkirim Service berjalan pada <i>background</i> aplikasi
7 April 2013 – 11.00 AM	Samsung Galaxy Mini	Froyo (2.2)	Tri	Aktif	Koneksi ke <i>server</i>	Berhasil	-
					Daftar <i>user</i>	Berhasil	Data tersimpan
					Menu utama	Berhasil	Halaman utama tampil
					Jalankan <i>service</i> lokasi	Data posisi terkirim	Service berjalan pada <i>background</i> aplikasi
10 April 2013 – 7.17 PM	smart friend andro	Gingerbread .DXLB1 (2.3.6)	<i>smartfri end</i>	Aktif	Koneksi ke <i>server</i>	Berhasil	-
					Daftar <i>user</i>	Berhasil	Data tersimpan
					Menu utama	Berhasil	Halaman utama tampil
					Jalankan <i>service</i> lokasi	Data posisi terkirim	Service berjalan pada <i>background</i> aplikasi

5.2.4. Kesimpulan Pengujian

Setelah dilakukan beberapa pengujian terhadap aplikasi yang telah dibangun, maka dapat ditarik kesimpulan dari hasil pengujian tersebut. Berikut kesimpulannya:

1. Aplikasi *mobile position* pada aplikasi *user A* telah mampu mengirimkan nilai posisi GPS ke server dari *background service* secara berkala.
2. Beberapa pengujian pada aplikasi *user A* mengalami kegagalan disebabkan beberapa kemungkinan. Diantaranya, konektivitas jaringan *internet* yang tidak stabil, respon dari *server* yang terlalu lama dan status GPS yang tidak aktif.
3. Aplikasi *mobile position* pada *user B* telah mampu membaca posisi lokasi *mobile* pada *user A* di *server online*.
4. Aplikasi *mobile position* pada *user B* telah mampu menampilkan posisi keberadaan *user A* pada *Google maps* dan melakukan *driving direction* terhadap *user A*.
5. Aplikasi *mobile position* pada *user B* telah mampu melakukan *track history* posisi *user A* pada setiap anggota *user B* yang terdaftar di *server*.

BAB VI

PENUTUP

Pada bab ini akan di paparkan kesimpulan yang didapat dari penjelasan pada bab-bab sebelumnya. Kemudian juga disampaikan beberapa saran yang berguna untuk kelanjutan pengembangan topik penelitian ini.

6.1. Kesimpulan

Setelah menyelesaikan serangkaian tahapan analisis, perancangan dan implementasi pada aplikasi *mobile phone position* pada *platform* Android, maka dapat diambil beberapa kesimpulan diantaranya adalah sebagai berikut:

1. Berhasil mengimplementasikan aplikasi *mobile position* menggunakan *location based service* pada *platform* Android.
2. Pengujian pada beberapa perangkat Android yang berbeda Samsung Galaxy y, Galaxy mini dan Smartfren dengan versi sistem operasi 2.2 dan 2.3.6 memperlihatkan aplikasi berjalan dengan baik.
3. Aplikasi *user A* berhasil mengirimkan informasi *latitude* dan *longitude* ke *server online* secara *background service*.
4. Aplikasi berhasil menjalankan fitur *tracking* posisi dari waktu ke waktu pada setiap anggota yang terdaftar pada aplikasi *user B*.
5. Aplikasi ini memiliki kekurangan yang disebabkan pada pembatasan masalah tugas akhir. Kekurangan seperti pengguna hanya terbatas pada pengguna *mobile* dengan sistem operasi Android, selain itu juga penggunaan aplikasi hanya pada user yang aktif sehingga data posisi hanya dapat dikirimkan apabila *user A* mengaktifkan GPS di perangkat *mobile*.

6. Aplikasi ini memiliki kekurangan dari layanan Google maps yang digunakan, yaitu terdapat beberapa lokasi posisi yang tidak tergambar oleh *google maps*.

6.2. Saran

Beberapa hal yang disarankan dalam pengembangan aplikasi *mobile phone position* pada *platform Android* ini adalah sebagai berikut:

1. Pada pengembangan aplikasi *mobile phone position* berikutnya, aplikasi ini hendaklah ditambahkan fitur keanggotaan tersendiri pada masing-masing pengguna *mobile phone* user B.
2. Pada pengembangan aplikasi *mobile phone position* berikutnya dapat di buat dengan berbasis *web apps*.
3. Pada aplikasi pengirim posisi dapat dibuat durasi pengiriman posisi secara dinamis.
4. Kekurangan aplikasi pada user yang pasif dapat ditambahkan fitur pengaktifan GPS secara otomatis ketika *user* B ingin mengetahui informasi posisi terbaru dari user A.
5. Untuk mengoptimalkan aplikasi juga dapat dilakukan penambahan keamanan data seperti enkripsi nilai *imei* id pada *server* online.
6. Untuk mengoptimalkan aplikasi ini dapat menggunakan layanan Google maps dengan versi pro/berbayar sehingga gambaran peta lebih lengkap.

DAFTAR PUSTAKA

- Amiral, Muhammad. 2010. *Aplikasi Pengingat Shalat dan Arah Kiblat Menggunakan Global Positioning System (GPS) Berbasis Android 1.6*. [Online] available. [http://kupluk.googlecode.com/files/Muhammad Amiral - Tugas Akhir.pdf](http://kupluk.googlecode.com/files/Muhammad%20Amiral%20-%20Tugas%20Akhir.pdf), 16 September 2011.
- Amri, Syaiful. 2012. *Membangun Sistem Navigasi di Surabaya Menggunakan Google Maps API*. Teknik Informatika, Politeknik Elektronika Negeri Surabaya. 2012
- Booch, Grady. 1993. *Object-Oriented Analysis and Design*. [Online] available. [http://bib.tiera.ru/dvd37/Booch G.-Object oriented analysis and design with applications\(1993\)\(Second Edition\)\(608\).pdf](http://bib.tiera.ru/dvd37/Booch%20G.-Object%20oriented%20analysis%20and%20design%20with%20applications(1993)(Second%20Edition)(608).pdf), 24 September 2012.
- Dario Ortuno. *Android application locates Facebook friends on a google map*. Dari : www.uv.es/~montanan/redes/trabajos/Facebook%20Friends.pdf, 24 Juni 2011.
- Dharwiyanti, Sri. 2003. Pengantar *Unified Modeling Language (UML)*. Dari : <http://www.ilmukomputer.org/wp-content/uploads/2006/08/yanti-uml.zip>, 24 September 2012.
- Ed Burnette, Hello Android. Dari: <http://pragprog.com/book/eband/hello-android>. Desember 2008.
- Hashimi, Sayed Y., Dkk. 2009. *Pro Android*. Dari : <http://ncc.dlut.edu.cn/course/android/reference/Pro.Android.pdf>, 16 September 2011.
- Kadir, Abdul. *Pengenalan Sistem Informasi*. Yogyakarta : Andi Yogya, 2003.
- Kruchten, Philippe. *The Rational Unified Process An Introduction, Second Edition*. Addison Wesley, 2000.

- Manav Singhal, Anupam Shukla. *Implementation of Location based Services in Android using GPS and Web Services*. International Journal of Computer Science Issues (Vol.9-2012/Iss.1), January 2012.
- Martin Sauter, *Communication System for the Mobile Information Society*. Germany 2006
- Nugroho, Adi. *Rekayasa Perangkat Lunak Menggunakan UML (Unified Modeling Language) dan Java*. Penerbit ANDI Yogyakarta, Yogyakarta. 2009.
- Pender, Thomas A. 2002. *UML Weekend Crash Course*. [Online] available <http://www.crnarupa.singidunum.ac.rs/Valjevo/2008-ProjektovanjeInformacionihSistema/Predavanja/e-knjige/UMLWeekendCrashCourse.pdf>, 24 September 2012.
- Ramadhani. *Rancang Bangun Aplikasi Mobile Kuliner (eMKUL) menggunakan Location Based Service (LBS) Berbasis Android*. Laporan Tugas Akhir Sarjana, Jurusan Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim Riau. 2012.
- Saidil, M. *Aplikasi Android Lbs (Location Based Service) PON Riau 2012*. Laporan Tugas Akhir Sarjana, Jurusan Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim Riau. 2012.
- <http://noerone4all.blogspot.com/2009/10/evolusi-1g-4g.html> [online] di akses 7.09AM tanggal 27 oktober 2012.
- <https://play.google.com/store/apps/details?id=com.cr.akm> [online] di akses 7.09 AM tanggal 29 oktober 2012.
- <http://www.equator-news.com/kubu-raya/20120702/semakin-mudah-burukan> [online] di akses 8.25 AM tanggal 3 Desember 2012.
- <http://www.idc.com/getdoc.jsp?containerId=prUS23638712> [online] di akses 6.46 AM tanggal 31 oktober 2012.
- <http://www.codeproject.com/Articles/470107/ASP-NET-MVC-4-Introduction> [online] di akses 7.03 AM tanggal 3 Desember 2012.