

**RANCANG BANGUN SISTEM INFORMASI MONITORING BUS
MENGUNAKAN GPS MOBILE ANDROID
(STUDI KASUS : CV. SEMPATI STAR BANDA ACEH)**

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat-syarat
guna memperoleh gelar Sarjana Komputer
Universitas U'Budiyah Indonesia



Diajukan Oleh :

MUSHALLI
NIM : 09111020

**FAKULTAS ILMU KOMPUTER
UNIVERSITAS U'BUDIYAH INDONESIA
BANDA ACEH
2014**

LEMBAR PENGESAHAN

RANCANG BANGUN SISTEM INFORMASI MONITORING BUS MENGUNAKAN GPS MOBILE ANDROID (STUDI KASUS : CV. SEMPATI STAR BANDA ACEH)

Tugas Akhir oleh Mushalli ini telah dipertahankan di depan dewan penguji pada hari Sabtu tanggal 07 Juni 2014.

Dewan Penguji :



1. Ketua

(Fathiah, ST., M.Eng)



2. Anggota

(Fesrianevalda, ST., M.Cs)



3. Anggota

(Zuhar Musliyana, S.ST)

**RANCANG BANGUN SISTEM INFORMASI MONITORING BUS
MENGUNAKAN GPS MOBILE ANDROID
(STUDI KASUS : CV. SEMPATI STAR BANDA ACEH)**

SKRIPSI

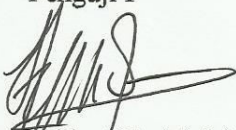
Disusun Untuk Menempuh Ujian Sarjana
Pada Fakultas Ilmu Komputer
Universitas U'Budiyah Indonesia

Diajukan Oleh :

MUSHALLI
NIM : 09111020

Disetujui,

Penguji I


(Fesrianevalda, ST., M.Cs)

Penguji II

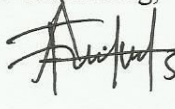

(Zuhar Musliyana, S.ST)



Ka. Prodi Teknik Informatika,


(Fathiah, ST., M.Eng)

Pembimbing,


(Fathiah, ST., M.Eng)



Mengetahui,

Dekan Fakultas Ilmu Komputer

(Jumalis J. Hius, ST., MBA)

LEMBAR PERNYATAAN

Saya menyatakan bahwa skripsi yang saya susun, sebagai syarat memperoleh gelar sarjana merupakan hasil karya tulis saya sendiri. Adapun bagian-bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan ilmiah. Saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi-sanksi lainnya sesuai dengan peraturan yang berlaku, apabila dikemudian hari ditemukan adanya plagiat dalam skripsi ini.

Banda Aceh, 14 Juni 2014



KATA PENGANTAR

Puji syukur Alhamdulillah penulis panjatkan kehadirat Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul **“Rancang Bangun Sistem Informasi Monitoring Bus Menggunakan GPS Mobile Android. (Studi Kasus : CV. Sempati Star Banda Aceh)”**. Tidak lupa pula shalawat beriring salam penulis persembahkan kepada Nabi Besar Muhammad SAW, yang dengan gigihnya membimbing manusia menuju alam yang penuh ilmu pengetahuan.

Penyusunan skripsi ini bertujuan untuk melengkapi salah satu syarat dalam rangka memperoleh gelar sarjana pada pendidikan program sarjana Teknik Informatika Fakultas Ilmu Komputer Universitas U'Budiyah Indonesia. Selesaiannya penulisan skripsi ini berkat ketekunan hati dan motivasi dari diri sendiri, keluarga, teman, serta bimbingan dari dosen pembimbing.

Dalam kesempatan ini sudah selayaknya penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Bapak Jurnalis J. Hius, S.T., MBA, selaku Dekan Fakultas Ilmu Komputer.
2. Ibu Fathiah, S.T., M.Eng, selaku ketua Program Studi Teknik Informatika dan juga sebagai dosen pembimbing yang telah bersedia meluangkan waktu, tenaga dan pikiran sehingga berkat bimbingan dan masukan-masukan yang diberikan, penulis dapat menyelesaikan skripsi ini.
3. Bapak Fesrianevalda, S.T., M,Cs dan Bapak Zuhar Musliyana, S.ST, selaku penguji sidang ujian Skripsi ini.
4. Seluruh Dosen dan Staf Universitas U'Budiyah Indonesia yang telah banyak membimbing dan mendidik penulis selama menjalani masa-masa studi.
5. Teristimewa Ayahanda dan Ibunda tercinta, terima kasih atas kasih sayang, perhatian, bantuan moral dan material serta semangat yang diberikan hingga akhirnya penulis dapat menyelesaikan Skripsi ini. Juga kepada keluarga besar serta saudara-saudara tersayang, terima kasih atas semangat dan bantuan yang telah diberikan.

6. Terima kasih juga penulis ucapkan kepada pihak CV. Sempati Star yang telah memberikan dukungan sarana dan informasi kepada penulis saat melakukan penelitian.
7. Para sahabat tercinta dan rekan-rekan mahasiswa/i pada prodi jenjang S1 Teknik Informatika serta semua pihak yang telah membantu penulis dalam menyelesaikan penulisan skripsi ini.

Segala usaha telah dilakukan untuk menyempurnakan skripsi ini. Namun, penulis menyadari bahwa dalam keseluruhan skripsi ini masih banyak kekurangan dan sangat jauh dari taraf kesempurnaan. Oleh karena itu penulis mengajak para pembaca untuk memberikan saran dan kritikan yang bersifat membangun demi mencapai kesempurnaan dalam penulisan skripsi ini, besar harapan penulis semoga skripsi ini dapat memberikan manfaat kepada para pembaca sekalian.

Banda Aceh, 14 Juni 2014

Penulis

ABSTRAK

Dengan memanfaatkan teknologi GPS dan GSM pada *mobile smartphone* berbasis Android sebagai perangkat *GPS Tracking*, sistem ini memungkinkan pengguna untuk mengetahui posisi kendaraan beserta jarak dan kecepatan secara *real time*. Modul *GPS Tracking* pada *mobile smartphone* berbasis Android tersebut terhubung dengan satelit dan mengirimkan data koordinat beserta jarak dan kecepatan ke server secara periodik melalui SMS. Posisi kendaraan yang di monitor dapat dilihat pada sebuah peta digital milik Google atau Google Maps yang ditampilkan dilayar komputer. Pengguna juga dapat mengetahui perkiraan waktu tibanya kendaraan ke terminal yang ditampilkan pada sistem *monitoring* ini.

Kata Kunci : *GPS, GPS Tracking, Android, Google Maps*

ABSTRACT

With using GPS and GSM technology in mobile smartphone based on Android as equipment of GPS Tracking. This system possibly the user to know where vehicle position and also the distance and velocity in realtime. These GPS tracking module are connected to satellite and able to send coordinate data, the distance, velocity into server periodicly via SMS. The vehicle into position are able to seing in digital Google Maps that shown in computer. User also able to predict arrived time that shown in these monitor system.

Keywords : *GPS, GPS Tracking, Android, Google Maps*

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
HALAMAN PERSETUJUAN	iii
LEMBAR PERNYATAAN	iv
KATA PENGANTAR.....	v
ABSTRAK	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL	xi
BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian.....	3
1.5 Sistematika Penulisan.....	3
BAB II TINJAUAN PUSTAKA	
2.1 Sistem Informasi.....	5
2.2 Flowchart	6
2.3 XAMPP	7
2.4 PHP Hypertext Preprocessor (PHP)	7
2.5 Macromedia Dreamweaver.....	9
2.6 MySQL	9
2.7 phpMyAdmin	10
2.8 HTML.....	11
2.9 SMS Gateway	12
2.10 NowSMS	13
2.11 Global Positioning System (GPS).....	13
2.12 GPS Tracker	18
2.13 Google Maps	19
2.14 Android.....	21
2.15 Android SDK.....	21
2.16 Android Development Tools (ADT)	22
2.17 Eclipse	23
BAB III METODELOGI PENELITIAN	
3.1 Lokasi Penelitian	24
3.2 Alat dan Bahan Penelitian	24
3.2.1 Kebutuhan Perangkat Keras dan Perangkat Lunak Server	24
3.2.2 Kebutuhan Perangkat Keras dan Perangkat Lunak Untuk GPS di Android.....	25
3.3 Jenis Penelitian	26
3.4 Jadwal Penelitian	26

BAB IV HASIL DAN PEMBAHASAN

4.1 Perancangan Sistem	28
4.1.1 Gambaran Umum Sistem	28
4.1.2 Flowchart Login	30
4.1.3 Flowchart Sistem Monitoring	31
4.1.4 Flowchart GPS Android	32
4.1.5 Perancangan Struktur Database	33
4.2 Perancangan Antarmuka	34
4.2.1 Perancangan Antarmuka Menu Login	34
4.2.2 Perancangan Antarmuka Menu Utama	35
4.2.3 Perancangan Antarmuka GPS Android	36
4.3 Implementasi Program	37
4.4 Penjelasan, Penulisan Script Program Dan Hasilnya	38
4.4.1 Koneksi	38
4.4.2 Login	38
4.4.3 Posisi Bus	40
4.4.4 Tampilan Antarmuka Sistem Monitoring Bus	42
4.4.5 Tampilan Ketika Bus Sudah Tiba Di Terminal Banda Aceh	43
4.4.6 Tampilan Ketika Bus Sudah Tiba Di Terminal Medan	43
4.4.7 Tampilan Ketika Bus Sedang Dalam Perjalanan	44
4.4.8 Prediksi Waktu Tiba Bus Di Terminal	45
4.4.9 Tampilan Struktur Tabel Dan Isi Database	46
4.5 Aplikasi GPS Android	46
4.6 Tampilan Aplikasi GPS Android	47
4.6.1 Tampilan Ketika Bus Sudah Tiba Di Terminal Medan	47
4.6.2 Tampilan Ketika Bus Sudah Tiba Di Terminal Banda Aceh	48
4.6.3 Tampilan Ketika Bus Sedang Dalam Perjalanan	48

BAB V PENUTUP

5.1 Kesimpulan	49
5.2 Saran	49

DAFTAR PUSTAKA	50
-----------------------------	-----------

BIODATA PENULIS	51
------------------------------	-----------

LAMPIRAN	52
-----------------------	-----------

DAFTAR GAMBAR

Gambar 2.1 Contoh Key Google Maps API	20
Gambar 4.1 Gambaran Umum Sistem	28
Gambar 4.2 Flowchart Login	30
Gambar 4.3 Flowchart Sistem Monitoring.....	31
Gambar 4.4 Flowchart GPS Android	32
Gambar 4.5 Perancangan Antarmuka Menu Login.....	34
Gambar 4.6 Perancangan Antarmuka Menu Utama	35
Gambar 4.7 Perancangan Antarmuka GPS Android.....	36
Gambar 4.8 Tampilan Login	40
Gambar 4.9 Tampilan Peta Dan Posisi Bus	42
Gambar 4.10 Tampilan Antarmuka Sistem Monitoring Bus	42
Gambar 4.11 Tampilan Ketika Bus Sudah Tiba Di Terminal Banda Aceh	43
Gambar 4.12 Tampilan Ketika Bus Sudah Tiba Di Terminal Medan.....	43
Gambar 4.13 Tampilan Ketika Bus Sedang Dalam Perjalanan Dan Berada dikawasan Pegunungan Seulawah	44
Gambar 4.14 Tampilan Ketika Bus Sedang Dalam Perjalanan Dan Berada dikawasan Pegunungan Seulawah	44
Gambar 4.15 Tampilan Prediksi Waktu Tiba Bus Di Terminal	46
Gambar 4.16 Tampilan Struktur Tabel Dan Isi Database	46
Gambar 4.17 Tampilan GPS Android Ketika Bus Sudah Tiba Di Terminal Medan.....	47
Gambar 4.18 Tampilan GPS Android Ketika Bus Sudah Tiba Di Terminal Banda Aceh	48
Gambar 4.19 Tampilan GPS Android Ketika Bus Dalam Perjalanan	48

DAFTAR TABEL

Tabel 2.1 Simbol flowchart.....	7
Tabel 3.1 Jadwal Penelitian.....	27
Tabel 4.1 Tabel Database Login	33
Tabel 4.2 Tabel Database SMS	33

BAB I

PENDAHULUAN

1.1 Latar Belakang

Suatu sistem pemantauan posisi kendaraan mulai dibuat seiring banyaknya permintaan terhadap sistem tersebut. Sistem pemantauan tersebut diharapkan dapat melacak keberadaan suatu kendaraan oleh pemilik kendaraan ataupun pihak ketiga lainnya. Sistem ini sering disebut sistem pelacakan kendaraan (*Vehicle Tracking System*). Sistem pelacakan modern umumnya menggunakan GPS (*Global Positioning System*) untuk menentukan lokasi kendaraan. Sistem ini juga biasanya menggunakan komponen komunikasi seperti jaringan selular atau satelit untuk mengirimkan posisi kendaraan kepada pengguna di tempat lain. Berbagai manfaat dapat diperoleh dari penerapan sistem pemantauan posisi kendaraan tersebut. Diantaranya adalah pengguna dapat mengetahui dimana posisi keberadaan kendaraan miliknya.

Dengan demikian penulis mendapatkan ide untuk membuat suatu aplikasi di CV. Sempati Star Banda Aceh, dengan memanfaatkan teknologi GPS yang terdapat pada *smartphone* berbasis Android sebagai *GPS Tracker* untuk memberikan informasi lokasi tentang keberadaan bus, sehingga dengan adanya aplikasi *monitoring* ini dapat berguna bagi pihak terkait untuk melakukan pelacakan terhadap kendaraan bus miliknya.

Untuk kepentingan tersebut, maka penulis mencoba membuat suatu aplikasi dengan judul **“Rancang Bangun Sistem Informasi Monitoring Bus**

Menggunakan GPS Mobile Android. (Studi Kasus : CV. Sempati Star Banda Aceh)”.

1.2 Rumusan Masalah

Permasalahan yang dibahas didalam penelitian tugas akhir ini adalah :

1. Bagaimana memberitahukan posisi bus pada saat melakukan perjalanan dalam bentuk peta digital (*Google Maps*).
2. Bagaimana memberitahukan perkiraan waktu tibanya bus di terminal Banda Aceh yang berangkat dari Medan.
3. Bagaimana memanfaatkan teknologi GPS yang ada pada perangkat *mobile Android* sebagai alat untuk melakukan *monitoring* posisi bus.
4. Bagaimana menampilkan induk sistem dalam bentuk sistem informasi dengan memanfaatkan *Google Maps API*.

1.3 Batasan Masalah

Adapun batasan-batasan masalah didalam penelitian tugas akhir ini adalah :

1. Sistem informasi yang dibangun berbasiskan *web*, dan peta yang digunakan adalah peta digital (*Google Maps*).
2. *GPS Tracker* yang digunakan adalah perangkat *mobile* yang memiliki *Platform Android*.
3. Sistem informasi yang dibangun membutuhkan koneksi internet dan diujicobakan secara *online*.
4. Rute yang dibahas hanya Banda Aceh - Medan.
5. Perkiraan waktu tibanya bus hanya yang berangkat dari Medan menuju Banda Aceh.

1.4 Tujuan Penelitian

Dari hasil uraian diatas maka tujuan penelitian tugas akhir ini adalah untuk :

1. Mempermudah mendapatkan informasi akan posisi keberadaan bus dengan melakukan *monitoring* yang memanfaatkan teknologi GPS.
2. Memberikan informasi posisi bus pada saat melakukan perjalanan dalam bentuk peta digital (*Google Maps*).
3. Memberikan informasi akan perkiraan waktu tibanya bus di terminal.
4. Memberikan hasil bagaimana dengan memanfaatkan teknologi GPS yang ada pada perangkat *mobile Android* sebagai alat untuk melakukan *monitoring* posisi bus.

1.5 Sistematika Penulisan

Dalam penulisannya, penelitian ini dibagi menjadi 5 bab. Sistematika penulisan bab dan gambaran masing-masingnya adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini membahas latar belakang penelitian yang akan dilakukan, rumusan masalah, sampai dengan batasan masalah, serta tujuan penelitian dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini membahas dasar teori perangkat yaitu mengenai pengertian sistem informasi, PHP, *Macromedia Dreamweaver*, XAMPP, MySQL, PhpMyAdmin, GPS (*Global Positioning System*) dan GPS Tracker, SMS Gateway, NowSMS, *Google Maps*, Android dan Android SDK,

ADT, Eclipse, serta semua yang berhubungan dengan penelitian pada tugas akhir ini.

BAB III METODE PENELITIAN

Bab ini membahas tentang jadwal penelitian, tempat penelitian, analisa kebutuhan, dan proses perancangan dan proses pembuatan tampilan sistem informasi pada penelitian tugas akhir ini.

BAB IV HASIL DAN PEMBAHASAN

Bab ini membahas tentang implementasi dan pengujian beserta hasil dari penelitian tugas akhir ini.

BAB V KESIMPULAN DAN SARAN

Bab ini merupakan bab terakhir yang berisi kesimpulan dan saran dari hasil penelitian tugas akhir ini.

BAB II

TINJAUAN PUSTAKA

2.1 Sistem Informasi

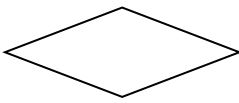
Menurut Herlambang (2005), data adalah fakta-fakta atau kejadian-kejadian yang dapat berupa angka-angka atau kode-kode tertentu. Data masih belum mempunyai arti bagi penggunanya. Untuk dapat mempunyai arti data diolah sedemikian rupa sehingga dapat digunakan oleh penggunanya. Hasil pengolahan data inilah yang disebut sebagai informasi. Secara ringkas, informasi adalah data yang telah diolah dan mempunyai arti bagi penggunanya.

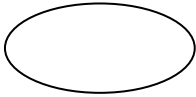
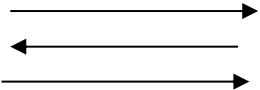
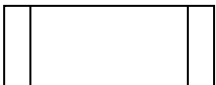
Menurut Jogiyanto (1995), sistem informasi adalah sekumpulan komponen pembentuk sistem yang mempunyai keterkaitan antara satu komponen dengan komponen lainnya yang bertujuan menghasilkan suatu informasi dalam suatu bidang tertentu. Dalam sistem informasi diperlukannya klasifikasi alur informasi, hal ini disebabkan keanekaragaman kebutuhan akan suatu informasi oleh pengguna informasi. Kriteria dari sistem informasi antara lain, fleksibel, efektif dan efisien.

Sistem informasi dapat didefinisikan sebagai prosedur-prosedur yang digunakan untuk mengolah data sehingga dapat digunakan oleh penggunanya.

2.2 Flowchart

Flowchart adalah penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. *Flowchart* menolong analisis dan *programmer* untuk memecahkan masalah kedalam *segmen-segmen* yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. Adapun simbol-simbol dalam *flowchart* antara lain:

No	Simbol	Fungsi
1		Terminal, untuk memulai atau mengakhiri Suatu program.
2		Proses, suatu simbol yang menunjukkan setiap pengolahan yang dilakukan oleh komputer
3		<i>Input, Output</i> untuk memasukkan data ataupun menunjukkan hasil dari suatu proses.
4		<i>Decision</i> , suatu kondisi yang akan menghasilkan beberapa kemungkinan jawaban atau pilihan.
5		<i>Properties</i> , suatu prosedur yang akan menyediakan tempat-tempat pengolahan data (<i>storage</i>).
6		<i>Conector</i> , suatu prosedur yang akan masuk Atau keluar melalui simbol ini dalam lembar yang berbeda

7		<i>Offline Conector</i> , merupakan simbol masuk Atau keluarnya suatu prosedur pada kertas yang sama.
8		<i>Arus/Flow</i> dari prosedur yang dapat dilakukan dari atas ke bawah.
9		<i>Document</i> , merupakan simbol atau data yang berbentuk informasi.
10		Untuk menyatakan sekumpulan kertas yang ditulis sebagai prosedur

Tabel 2.1 Simbol *flowchart*

2.3 XAMPP

XAMPP adalah perangkat lunak bebas (*open source*), yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, *MySQL database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), *Apache*, *MySQL*, *PHP* dan *Perl*. Program ini tersedia dalam *GNU General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis.

2.4 PHP Hypertext Preprocessor (PHP)

PHP adalah *server side scripting environment* yang dapat digunakan untuk membuat dan menjalankan aplikasi-aplikasi di *web server* agar lebih interaktif dan *programmable*. Dengan PHP aplikasi-aplikasi yang ada di *web server* benar-benar dijalankan di *web server* tanpa mengharuskan adanya tambahan atau syarat

tertentu untuk sisi *client* (*web browser*). PHP biasanya dijadikan sebagai *module* dalam suatu *web* agar bisa mengeksekusi *file-file* PHP yang tersedia di *web server*. PHP dapat berjalan di hampir seluruh *platform*, *open source* dan berlisensi GNU *Public License* (GPL). (Welling, 2003).

PHP pada mulanya di tulis sebagai sebuah kumpulan dari CGI dengan menggunakan bahasa pemograman C oleh *programmer* bernama Rasmus Lerdorf. *Programmer* asal Greenland ini membuat PHP pada tahun 1994 untuk menggantikan sebagian kecil kumpulan *script* dengan Perl yang digunakan untuk *maintenance* halaman web miliknya. Lerdorf mengawali menciptakan PHP untuk menampilkan *resume* miliknya dan mengumpulkan beberapa data, seperti berapa banyak lalu lintas data yang diterima dalam halaman *web* miliknya. (Welling, 2003).

Setelah mengalami perkembangan oleh suatu kelompok *open source* (termasuk Rasmus) maka mulai versi 3 PHP menampakkan keunggulan sebagai salah satu bahasa *server* yang handal. Melalui perkembangan yang pesat ini banyak fasilitas yang ditambahkan oleh kelompok ini, maka jadilah PHP disebut sebagai *Hypertext Preprocessor*. Sintak yang digunakan berasal dari bahasa C, Java maupun Perl.

2.5 Macromedia Dreamweaver

Macromedia Dreamweaver adalah perangkat lunak profesional perancang halaman *web* dengan tingkat fleksibilitas yang sangat tinggi. Tingkat fleksibilitas dimaksud memungkinkan pengguna dengan keahlian tingkat lanjut (*advance*) untuk membuat situs *web* dengan cara menulis langsung kode HTML pada *tools* pemrograman yang terintegrasi pada perangkat lunak ini atau memungkinkan pengguna tingkat awal (*beginner*) membuat situs *web* dengan menggunakan lingkungan *visual editing* pada menu-menu yang sudah disediakan seperti *drag and drop*, semua elemen atau *asset* pada dokumen situs *web*.

Macromedia Dreamweaver memiliki kompatibilitas dan dukungannya terhadap berbagai bahasa pemrograman *web*, antara lain : ASP, JSP, CFM, ASP.NET, PHP, *JavaScript*, CSS dan XML. Disamping keunggulan-keunggulan lainnya dibandingkan dengan *software web design* yang lain.

Macromedia Dreamweaver menjadi *software* utama yang digunakan oleh *web designer* dan *web programmer* guna mengembangkan situs *web*. Ruang kerja, fasilitas dan kemampuan *Dreamweaver* mampu meningkatkan produktivitas dan efektivitas dalam desain maupun pembangunan situs *web*. *Dreamweaver* juga dilengkapi dengan fasilitas untuk manajemen situs yang cukup lengkap.

2.6 MySQL

MySQL dalam perkembangannya disebut SQL yang merupakan kepanjangan dari *Structured Query Language*. SQL merupakan bahasa terstruktur yang khusus digunakan untuk mengolah *database*. SQL pertama kali didefinisikan

oleh American National Standards Institute (ANSI) pada tahun 1986. MySQL adalah sebuah sistem manajemen *database* yang bersifat *open source*.

MySQL merupakan pasangan serasi dari PHP. MySQL dibuat dan dikembangkan oleh MySQL AB yang berada di Swedia. MySQL dapat digunakan untuk membuat dan mengola *database* beserta isinya. Kita dapat memanfaatkan MySQL untuk menambahkan, mengubah dan menghapus data yang berada dalam *database*.

MySQL merupakan sistem manajemen *database* yang bersifat *relational*. Artinya data-data yang dikelola dalam *database* akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan menjadi jauh lebih cepat. MySQL dapat digunakan untuk mengelola *database* mulai dari yang kecil sampai dengan yang sangat besar.

MySQL juga dapat menjalankan perintah-perintah *Structured Query Language* (SQL) untuk mengelola *database-database* yang ada di dalamnya. Hingga kini, MySQL sudah berkembang hingga versi 7. MySQL 7 sudah mendukung *trigger* untuk memudahkan pengelolaan tabel dalam *database*.

2.7 phpMyAdmin

phpMyAdmin adalah sebuah perangkat lunak bebas (*open source*) yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi *database* MySQL melalui jaringan lokal maupun internet. phpMyAdmin mendukung berbagai operasi MySQL, diantaranya untuk mengelola *database*, tabel, *fields*, *relations*, *indeks*, *users*, *permissions*, dan lain-lain.

Pada dasarnya, mengelola *database* dengan MySQL harus dilakukan dengan cara mengetikkan baris-baris perintah yang sesuai (*command line*) untuk setiap maksud tertentu. Jika seseorang ingin membuat *database*, maka harus menuliskan baris perintah yang sesuai untuk membuat *database*. Jika seseorang menghapus tabel, maka harus menuliskan baris perintah yang sesuai untuk menghapus tabel. Hal tersebut tentu saja sangat menyulitkan karena seseorang harus hafal dan mengetikkan perintahnya satu per satu.

Saat ini banyak sekali perangkat lunak yang dapat dimanfaatkan untuk mengelola *database* dalam MySQL, salah satunya adalah phpMyAdmin. Dengan phpMyAdmin, seseorang dapat membuat *database*, membuat tabel, mengisi data, dan lain-lain dengan mudah, tanpa harus menghafal baris perintahnya.

phpMyAdmin merupakan bagian untuk mengelola *database* MySQL yang ada di komputer. Untuk membukanya, buka *browser* misal *Mozilla Firefox*, lalu tuliskan alamat *http://localhost/phpMyAdmin*, maka akan muncul halaman phpMyAdmin. Di situ nantinya seseorang bisa membuat *database* baru, dan mengelolanya.

2.8 HTML

HTML (*Hyper Text Markup Language*) adalah sebuah *bahasa markup* yang digunakan untuk membuat sebuah halaman *web* halaman atau dokumen pada sebuah *web* yang kemudian dapat diakses dan dibaca layaknya sebuah artikel dan menampilkan berbagai informasi di dalam sebuah *browser* internet. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (*Standard Generalized*

Markup Language), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman *web*. HTML saat ini merupakan standar internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium* (W3C).

HTML berupa kode-kode *tag* yang menginstruksikan *browser* untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sebuah *file* yang merupakan *file* HTML dapat dibuka dengan menggunakan *browser web* seperti *Mozilla Firefox*, *Google Chrome*, *Opera Browser*, *Safari* atau *Microsoft Internet Explorer*. HTML juga dapat digunakan sebagai *link-link* antara *file-file* dalam situs atau dalam komputer dengan menggunakan *localhost* atau *link* yang menghubungkan antar situs dalam dunia internet.

2.9 SMS Gateway

SMS *Gateway* adalah aplikasi SMS dimana pesan yang di terima dan dikirimkan menggunakan bantuan *Gateway Device* terintegrasi dengan *database server* yang dapat mendistribusikan pesan SMS secara otomatis. Kelebihan SMS *Gateway* adalah SMS *Gateway* merupakan aplikasi berbasis komputer, sehingga dapat di otomatisasi. Dapat menyimpan data dalam jumlah yang banyak karena disimpan di sebuah hardisk *server*.

2.10 NowSMS

NowSMS merupakan sebuah perangkat lunak untuk SMS *Gateway* yang dikembangkan oleh *Now Wireless Limited* sejak tahun 2002 yang digunakan untuk membuat sistem SMS *Gateway*. NowSMS merupakan perangkat lunak

yang bersifat *shareware*. Dengan menggunakan NowSMS telah terdapat fitur yang mendukung SMS *Gateway* tersebut antara lain:

1. Pembacaan *modem* secara otomatis.
2. Instalasi yang mudah dan cepat.
3. Tersedia fitur pengiriman pesan ke email.
4. Mampu mengirimkan *ringtone* dan pesan multimedia.
5. Adanya tampilan *web* interaktif dan fitur-fitur lainnya.

2.11 Global Positioning System (GPS)

GPS atau *Global Positioning System*, merupakan sebuah alat atau sistem yang dapat digunakan untuk menginformasikan penggunanya dimana dia berada (secara *global*) di permukaan bumi yang berbasiskan satelit. Data dikirim dari satelit berupa sinyal radio dengan data digital. Dimanapun anda berada, maka GPS bisa membantu menunjukkan arah, selama anda melihat langit. Layanan GPS ini tersedia gratis, bahkan tidak perlu mengeluarkan biaya apapun kecuali membeli GPS *receiver*nya.

GPS adalah sistem navigasi yang berbasiskan satelit yang saling berhubungan yang berada di orbitnya. Satelit-satelit itu milik Departemen Pertahanan (*Department of Defense*) Amerika Serikat yang pertama kali diperkenalkan mulai tahun 1978 dan pada tahun 1994 sudah memakai 24 satelit. Untuk dapat mengetahui posisi seseorang maka diperlukan alat yang diberi nama GPS *receiver* yang berfungsi menerima sinyal yang dikirim dari satelit GPS. Posisi di ubah menjadi titik yang dikenal dengan nama *way-point* nantinya akan

berupa titik-titik koordinat lintang dan bujur dari posisi seseorang atau suatu lokasi kemudian di layar pada peta elektronik.

Untuk menginformasikan posisi *user*, 24 satelit GPS yang ada di orbit sekitar 12.000 mil di atas kita. Bergerak konstan bergerak mengelilingi bumi 12 jam dengan kecepatan 7.000 mil per jam. Satelit GPS berkekuatan energi sinar matahari, mempunyai baterai cadangan untuk menjaga agar tetap berjalan pada saat gerhana matahari atau pada saat tidak ada energi matahari. Roket penguat kecil pada masing-masing satelit agar dapat mengorbit tepat pada tempatnya.

Satelit GPS adalah milik Departemen Pertahanan (*Department of Defense*) Amerika Serikat, adapun hal-hal lainnya adalah :

1. Nama satelit adalah NAVSTAR.
2. GPS satelit pertama kali adalah tahun 1978.
3. Mulai ada 24 satelit dari tahun 1994.
4. Satelit di ganti setiap 10 tahun sekali.
5. GPS satelit beratnya kira-kira 2.000 pounds.
6. Kekuatan *transmitter* hanya 50 *watts* atau kurang.

Satelit-satelit GPS harus selalu berada pada posisi orbit yang tepat untuk menjaga akurasi data yang dikirim ke GPS *receiver*, sehingga harus selalu dipelihara agar posisinya tepat. Stasiun-stasiun pengendali di bumi ada di Hawaii, Ascension Island, Diego Garcia, Kwajalein dan Colorado Spring. Stasiun bumi tersebut selalu memonitor posisi orbit jam-jam satelit dan dipastikan selalu tepat.

Setiap daerah di atas permukaan bumi ini minimal terjangkau oleh 3-4 satelit. Pada prakteknya setiap GPS terbaru bisa menerima sampai dengan 12

channel satelit sekaligus. Kondisi langit yang cerah dan bebas dari halangan membuat GPS dapat dengan mudah menangkap sinyal yang dikirimkan oleh satelit. Semakin banyak satelit yang diterima oleh GPS, maka akurasi yang diberikan juga akan semakin tinggi.

Cara kerja GPS secara *logic* ada 5 langkah:

1. Memakai perhitungan "*triangulation*" dari satelit.
2. Untuk perhitungan "*triangulation*", GPS mengukur jarak menggunakan *travel time* sinyal radio.
3. Untuk mengatur *travel time*, GPS memerlukan akurasi waktu yang tinggi.
4. Untuk perhitungan jarak, kita harus tahu dengan posisi satelit dan ketinggian pada orbitnya.
5. Terakhir harus mengoreksi *delay* sinyal waktu perjalanan di atmosfer sampai diterima *receiver*.

Satelit GPS berputar mengelilingi bumi selama 12 jam di dalam orbit yang akurat dan mengirimkan sinyal informasi ke bumi. GPS *receiver* mengambil informasi itu dengan menggunakan perhitungan "*triangulation*" menghitung lokasi *user* dengan tepat. GPS *receiver* membandingkan waktu sinyal di kirim dengan waktu sinyal tersebut di terima. Dari informasi itu dapat diketahui berapa jarak satelit. Dengan perhitungan jarak GPS *receiver* dapat melakukan perhitungan dan menentukan posisi *user* dan menampilkan dalam peta elektronik.

Sebuah GPS *receiver* harus mengunci sinyal minimal tiga satelit untuk menghitung posisi 2D (*latitude, longitude dan altitude*) dan *track* pergerakan. Jika

GPS *receiver* dapat menerima empat atau lebih satelit, maka dapat menghitung posisi 3D (*latitude, longitude dan altitude*). Jika sudah dapat menentukan posisi *user*, selanjutnya GPS dapat menghitung informasi lain, seperti kecepatan, arah yang dituju, jalur, tujuan perjalanan, jarak tujuan, matahari terbit dan matahari terbenam dan masih banyak lagi.

Satelit GPS dalam mengirim informasi waktu sangat presisi karena satelit tersebut memakai jam atom. Jam atom yang ada pada satelit adalah jam dengan partikel atom yang di isolasi, sehingga dapat menghasilkan jam yang akurat dibandingkan dengan jam biasa.

Perhitungan waktu yang akurat sangat menentukan akurasi perhitungan untuk menentukan informasi lokasi kita. Selain itu semakin banyak sinyal satelit yang dapat diterima maka akan semakin baik presisi data yang diterima karena ketiga satelit mengirim *pseudo-random code* dan waktu yang sama. Ketinggian itu menimbulkan keuntungan dalam mendukung proses kerja GPS, bagi kita karena semakin tinggi maka semakin bersih atmosfer sehingga gangguan semakin sedikit dan orbit yang cocok dan perhitungan matematika yang cocok. Satelit harus tetap pada posisi yang tepat sehingga stasiun di bumi harus memonitor setiap pergerakan satelit dengan bantuan radar yang presisi selalu di cek tentang *altitude*, posisi dan kecepatannya.

Dengan teknologi GPS dapat digunakan untuk beberapa keperluan sesuai dengan tujuannya. GPS dapat digunakan oleh peneliti, olahragawan, petani, tentara, pilot, petualang, pendaki, pengantar barang, pelaut, kurir, penebang

pohon, pemadam kebakaran, pengemudi, dan orang dengan berbagai kepentingan untuk meningkatkan produktivitas, keamanan, dan untuk kemudahan.

Dari beberapa pemakaian di atas dikategorikan menjadi:

1. Lokasi. Digunakan untuk menentukan dimana lokasi suatu titik dipermukaan bumi berada.
2. Navigasi. Membantu mencari lokasi suatu titik di bumi.
3. *Tracking*. Membantu untuk memonitoring pergerakan obyek.
4. Membantu memetakan posisi tertentu, dan perhitungan jaringan terdekat.
5. *Timing*. Dapat dijadikan dasar penentuan jam seluruh dunia, karena memakai jam atom yang jauh lebih presisi di banding dengan jam biasa.

Beberapa istilah yang penting untuk diketahui yang berhubungan dengan GPS adalah :

Waypoint: Istilah yang digunakan oleh GPS untuk suatu lokasi yang telah ditandai. *Waypoint* terdiri dari koordinat lintang (*latitude*) dan bujur (*longitude*). Sebuah *waypoint* biasa digambarkan dalam bentuk titik dan simbol sesuai dengan jenis lokasi.

Mark: Menandai suatu posisi tertentu pada GPS. Jika Anda menandai lokasi menjadi *waypoint*, maka dikatakan Anda melakukan *marking*.

Route: Kumpulan *waypoint* yang ingin Anda tempuh secara berurutan dan dimasukkan kedalam GPS.

Track: Arah perjalanan yang sedang Anda tempuh dengan menggunakan GPS. Biasanya digambarkan berupa garis pada display GPS.

Elevation: Istilah pada GPS untuk menentukan ketinggian. Ada dua jenis pengukur ketinggian pada GPS, yaitu menggunakan alat klasik “*barometer*” atau menggunakan perhitungan satelit. Pengukuran ketinggian menggunakan barometer jauh lebih akurat di udara bebas, namun tidak bisa bekerja dalam pesawat atau ruang vakum dengan tekanan udara di luar. Pengukuran ketinggian menggunakan satelit akan lebih akurat pada tempat seperti itu.

Bearing: arah/posisi yang ingin Anda tuju. Contohnya, Anda ingin menuju ke suatu lokasi di posisi A yang letaknya di Utara, maka *bearing* Anda dikatakan telat diset ke Utara.

Heading: Arah *actual* yang sedang dijalankan. Contohnya, pada saat menuju ke posisi A tadi, Anda menemui halangan sehingga harus memutar ke Selatan terlebih dahulu, maka *heading* Anda pada saat itu adalah Selatan. (Andi, 2005).

2.12 GPS Tracker

GPS *Tracker* atau sering disebut dengan GPS *Tracking* adalah teknologi AVL (*Automated Vehicle Locater*) atau suatu sistem pemantauan jarak jauh yang menggunakan satelit GPS yang memungkinkan pengguna untuk melacak posisi kendaraan dalam keadaan *real-time*. GPS *Tracking* memanfaatkan kombinasi teknologi GSM dan GPS sebagai penentu lokasi kendaraan/*asset* bergerak dengan tepat dan akurat dalam bentuk titik koordinat yang kemudian diimplementasikan ke dalam bentuk peta digital.

Cara kerja GPS *Tracker* adalah sebagai berikut:

1. GPS *Tracking Device* menerima sinyal GPS dari beberapa satelit GPS.
2. Berdasarkan sinyal-sinyal tersebut, GPS *Tracking* menghitung posisinya.

3. GPS Tracking mengirim data posisinya tersebut serta secara *online* ke *Tracking Server System* melalui jaringan GSM.
4. Pengguna dapat menggunakan *web browser* untuk melakukan pelacakan dan pemantauan kendaraan yang akan ditampilkan pada peta digital melalui jaringan internet.

2.13 Google Maps

Google Maps adalah layanan gratis yang diberikan oleh Google dan sangat populer. *Google Maps* adalah suatu peta dunia yang dapat kita gunakan untuk melihat suatu daerah. Dengan kata lain, *Google Maps* merupakan suatu peta yang dapat dilihat dengan menggunakan suatu *browser*. Kita dapat menambahkan fitur *Google Maps* dalam *website* yang telah kita buat atau pada blog kita yang berbayar maupun gratis sekalipun dengan *Google Maps API*. (Pandu, 2010).

Google Maps API adalah suatu *library* yang berbentuk *JavaScript*. Cara membuat *Google Maps* untuk ditampilkan pada suatu *website* atau *blog* sangat mudah hanya dengan membutuhkan pengetahuan mengenai HTML serta *JavaScript*, serta koneksi internet yang sangat stabil. Dengan menggunakan *Google Maps API*, kita dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga kita dapat fokus hanya pada data-data yang akan ditampilkan. Dengan kata lain, kita hanya membuat suatu data sedangkan peta yang akan ditampilkan adalah milik Google sehingga kita tidak dipusingkan dengan membuat peta suatu lokasi bahkan peta dunia.

Untuk menggunakan *Google Maps API*, maka pengguna diwajibkan untuk mendaftarkan diri untuk mendapatkan *API key* yang nantinya akan digunakan

untuk mengakses Google Maps API tersebut. Sebuah *key* dapat digunakan pada suatu *domain*, *subdomain*, dan semua URL pada *domain* tersebut.



Gambar 2.1 Contoh Key Google Maps API

2.14 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan perangkat lunak sumber terbuka (*open source*) bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam perangkat bergerak. Awalnya, Google Inc. membeli Android Inc., perusahaan yang membuat perangkat lunak untuk telepon seluler. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. (Elcom, 2010).

Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan perangkat lunak sumber terbuka (*open source*) pada perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD).

2.15 Android SDK

Android SDK adalah *Software Development Kit* yang digunakan untuk membuat aplikasi untuk *platform* Android. Di dalam Android SDK sudah termasuk *project* contoh dengan *source code*, *tools development*, *emulator* dan *library* yang dibutuhkan untuk membangun aplikasi Android.

Untuk mendapatkan Android SDK, pengguna cukup mendownloadnya secara gratis di alamat <http://developer.android.com/sdk/index.html>. Seluruh *resource* yang diperlukan telah disediakan pada *website* tersebut, mulai dari *Developer Guide*, *Reference* sampai *Resource* yang dibutuhkan untuk membangun aplikasi Android.

2.16 Android Development Tools (ADT)

Android Development Tools (ADT) adalah *plugin* yang didesain untuk IDE Eclipse yang memberikan kita kemudahan dalam mengembangkan aplikasi Android dengan menggunakan IDE Eclipse. Dengan menggunakan ADT untuk Eclipse akan memudahkan kita dalam membuat aplikasi *project* Android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya,

begitu juga kita dapat melakukan *running* aplikasi menggunakan Android SDK melalui Eclipse. Dengan ADT juga kita dapat melakukan pembuatan *package* Android (*.apk*) yang digunakan untuk distribusi aplikasi Android yang kita rancang. Mengembangkan aplikasi Android dengan menggunakan ADT di eclipse sangat dianjurkan dan sangat mudah untuk memulai mengembangkan aplikasi Android.

2.17 Eclipse

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

Berikut ini adalah sifat dari Eclipse:

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

Pada saat ini, Eclipse merupakan salah satu IDE favorit karena gratis dan *open source*. *open source* berarti setiap orang boleh melihat kode pemrograman

perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan membuat komponen yang disebut *plugin*.

BAB III

METODE PENELITIAN

3.1 Lokasi Penelitian

Dalam membuat semua penelitian ini penulis melakukan penelitian di perusahaan bus CV. Sempati Star dimana aplikasi yang dibangun diterapkan untuk melakukan *monitoring* terhadap bus pada perusahaan tersebut.

3.2 Alat dan Bahan Penelitian

Agar sistem perancangan yang telah dibangun dapat berjalan dengan baik, maka perlu kiranya dilakukan pengujian sistem yang telah dibuat. Untuk itu dibutuhkan beberapa komponen utama mencakup perangkat keras (*hardware*) dan perangkat lunak (*software*).

3.2.1 Kebutuhan Perangkat Keras dan Perangkat Lunak Server

A. Perangkat Keras (*Hardware*)

Adapun kebutuhan minimum perangkat keras (*hardware*) adalah:

1. *Processor* : *Dual Core* 1 GHz setara atau yang lebih tinggi.
2. *Harddisk* : 40 GB atau yang lebih tinggi.
3. *Memory* : 1 GB atau yang lebih tinggi.
4. *Monitor* : *Monitor* dengan resolusi 1024x768 atau yang lebih tinggi.

Adapun perangkat keras lainnya yang dibutuhkan adalah satu unit *modem* GSM sebagai pelaku SMS *gateway* untuk menerima sms data koordinat yang dikirimkan dari perangkat GPS *Tracking* yaitu GPS *mobile* Android. Komputer *server* juga harus mempunyai LAN *Card* atau *wireless* LAN sebagai alat untuk koneksi internet namun ini adalah *optional* karena *modem* GSM telah bisa untuk membuat koneksi internet.

B. Perangkat Lunak (*Software*)

Adapun kebutuhan minimum perangkat lunak (*software*) adalah :

1. Sistem operasi *Windows* XP atau yang lebih tinggi.
2. *Mozilla Firefox* versi 3.6 atau yang lebih tinggi.
3. *Apache web server* versi 2.2.1 atau yang lebih tinggi.
4. *MySQL database server* versi 5.0.51 atau yang lebih tinggi.
5. *NowSMS Gateway* versi 2008.02.09 atau yang lebih tinggi.

3.2.2 Kebutuhan Perangkat Keras dan Perangkat Lunak Untuk GPS di Android

A. Perangkat Keras (*Hardware*)

1. Processor : 600 MHz ARMv6 atau yang lebih tinggi.
2. System Storage : 512 MB atau yang lebih tinggi.
3. Memory : 256 MB atau yang lebih tinggi.
4. Display : 240x320 *pixels*, 3,2 *inches* atau yang lebih tinggi.
5. Type GPS : A-GPS

B. Perangkat Lunak (*Software*)

Adapun kebutuhan minimum perangkat lunak pada GPS *mobile* Android adalah Sistem Operasi Android 2.3.6 atau yang lebih tinggi.

3.3 Jenis Penelitian

a. Pengamatan (*Observation*)

Dalam hal ini penulis melakukan pengamatan untuk mendapatkan data secara umum dengan melihat secara langsung, dan mengamati obyek pengamatan yaitu berupa titik koordinat lokasi terminal.

b. Wawancara (*Interview*)

Dalam hal ini penulis melakukan wawancara untuk melengkapi bahan yang sudah ada selama pengamatan (*observasi*). Penulis melakukan tanya jawab kepada pihak terkait tentang penelitian yang sedang diteliti.

c. Penelitian Perpustakaan (*Library Research*)

Penelitian perpustakaan adalah penelitian dengan sumber-sumber yang didapatkan pada perpustakaan. Baik berupa sumber dari buku-buku pustaka maupun sumber-sumber yang ada di internet.

3.4 Jadwal Penelitian

Penelitian ini dilaksanakan pada minggu ketiga bulan Februari 2014 sampai dengan minggu kedua bulan Juni 2014, mulai dari tahap pengumpulan data, menganalisa data, merancang sistem, implementasi program, melakukan pengujian terhadap program yang telah dibuat, mengevaluasi program sampai tahap akhir yaitu melakukan dokumentasi pada setiap jadwal kegiatan yang telah dilakukan, agar pembuatan sistem nantinya dapat terstruktur dengan baik. Selengkapnya dapat dilihat seperti pada tabel jadwal penelitian berikut ini:

No	KEGIATAN	Bulan																	
		Februari				Maret				April				Mei				Juni	
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2
1	Pengumpulan Data																		
2	Analisa Data																		
3	Perancangan Sistem																		
4	Implementasi																		
5	Pengujian Program																		
6	Evaluasi Program																		
7	Dokumentasi																		

Tabel 3.1 Jadwal Penelitian

BAB IV

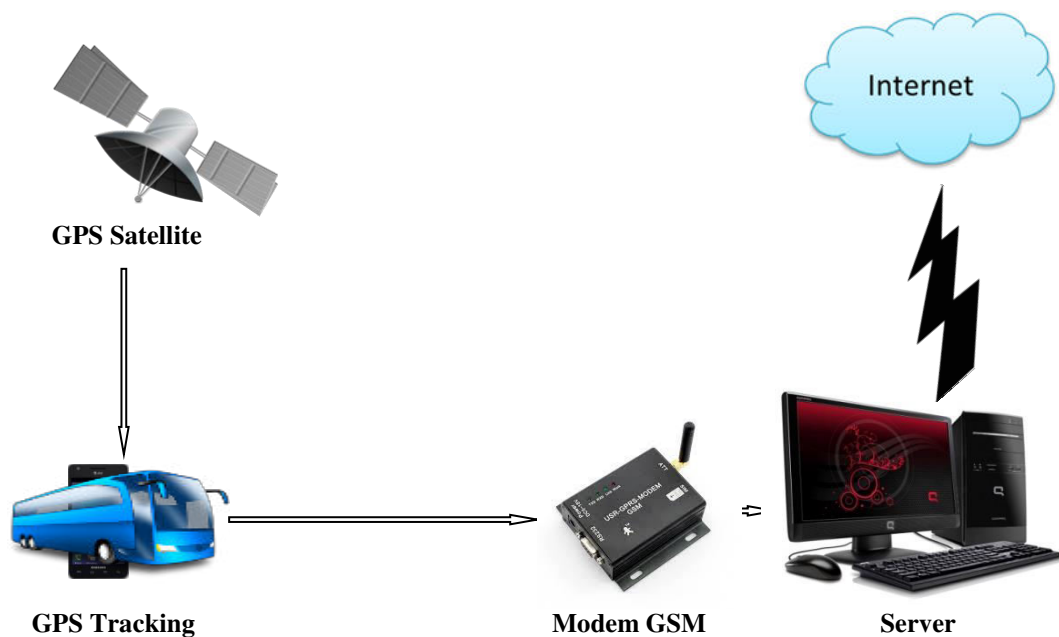
HASIL DAN PEMBAHASAN

4.1 Perancangan Sistem

4.1.1 Gambaran Umum Sistem

Sistem informasi yang dibuat dalam penelitian ini adalah sistem yang berbasis *web* dan membutuhkan koneksi internet agar sistem dapat diakses oleh pengguna yaitu *admin* untuk melakukan *monitoring*.

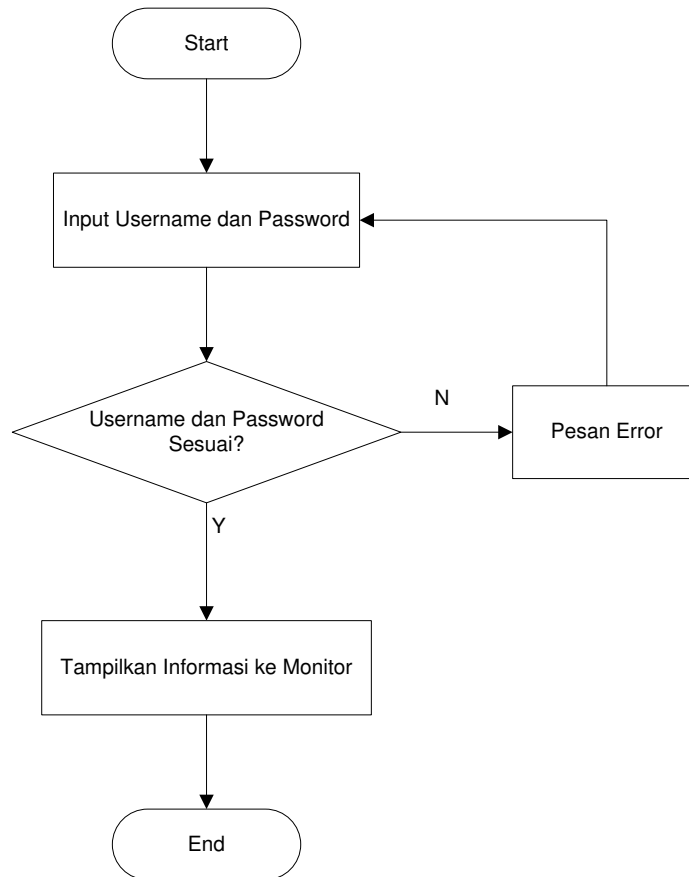
Adapun gambaran umum sistem informasi *monitoring* yang akan dibangun adalah seperti terlihat pada gambar dibawah ini :



Gambar 4.1 Gambaran Umum Sistem

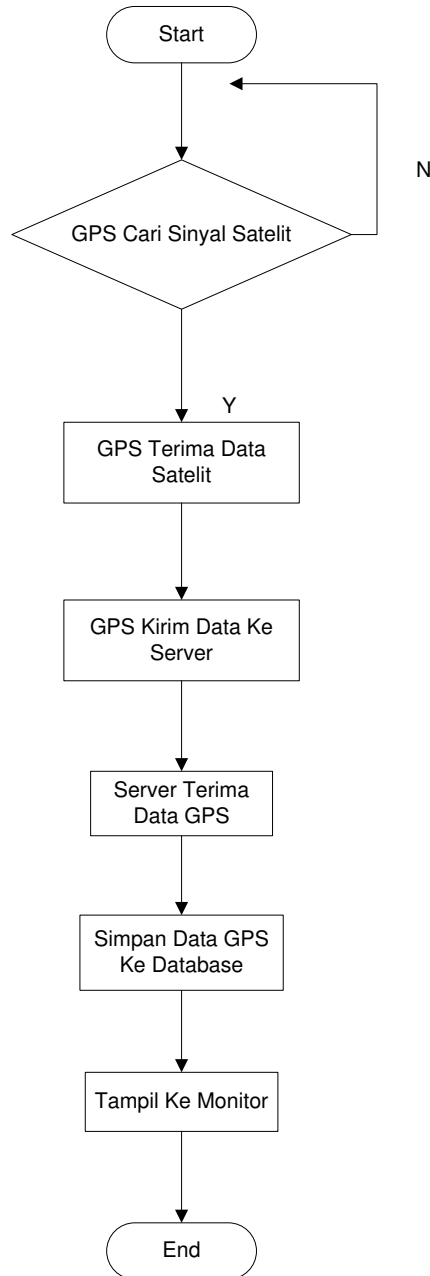
Pada gambaran umum sistem menunjukkan ada beberapa perangkat yang terhubung, yaitu perangkat *mobile* Android sebagai perangkat yang dipasang aplikasi *GPS Tracking* yang ditempatkan didalam bus, *modem* GSM sebagai pelaku *SMS gateway* dan komputer sebagai *server*. *GPS Tracking* yang terpasang pada perangkat *mobile* Android menerima sinyal GPS berupa koordinat lokasi dan selanjutnya dikirim ke *SMS Gateway (modem GSM)* yang terpasang dikomputer *server* berupa SMS. Agar aplikasi *GPS Tracking* tersebut dapat mengirimkan koordinat lokasi ke server maka dibutuhkan minimal 3 satelit untuk menangkap sinyal yang dikirimkan. Data yang diterima berupa data lokasi yaitu *longitude* dan *latitude* , kecepatan dan jarak serta informasi pendukung seperti data tanggal dan data waktu. Data koordinat yang dikirim dari *GPS Tracking mobile* Android tersebut dikirim ke *server* dan diterima oleh *SMS gateway (modem GSM)* tersebut kemudian disimpan kedalam *database* dan kemudian ditampilkan pada layar *monitor* sistem *monitoring* dengan tampilan peta digital atau *google maps*.

4.1.2 Flowchart Login



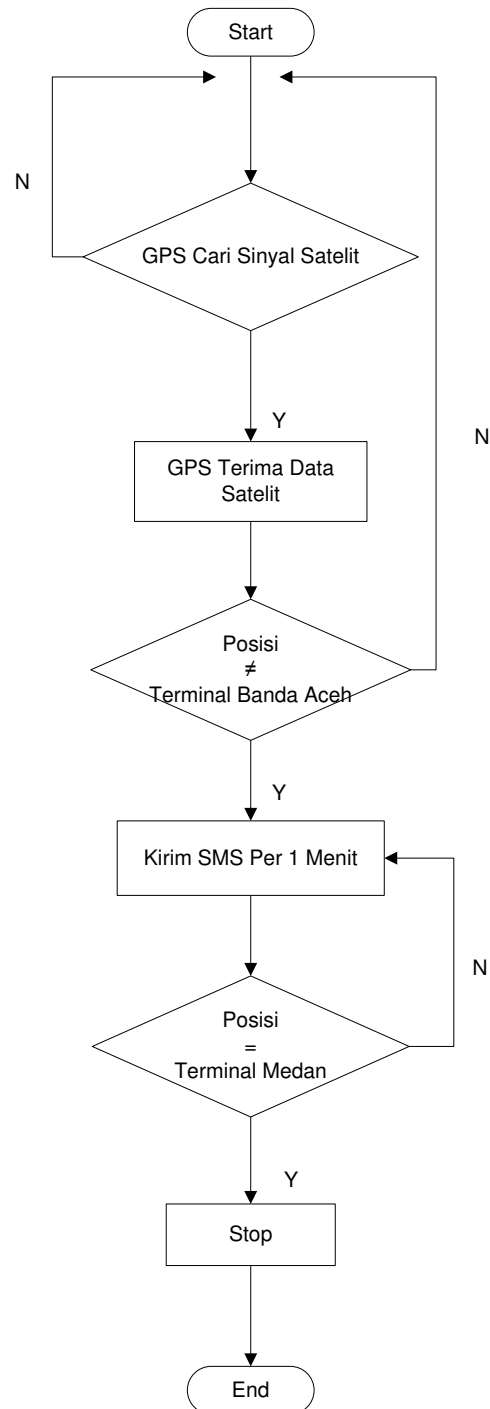
Gambar 4.2 Flowchart Login

4.1.3 Flowchart Sistem Monitoring



Gambar 4.3 Flowchart Sistem Monitoring

4.1.4 Flowchart GPS Android



Gambar 4.4 Flowchart GPS Android

4.1.5 Perancangan Struktur *Database*

Dalam hal ini penulis memberikan struktur tabel *database* yang penulis gunakan untuk merancang sistem informasi *monitoring* bus, antara lain sebagai berikut :

1. Tabel *database login*

<i>Field name</i>	<i>Data Type</i>	<i>size</i>	Description
username	varchar	20	username
password	varchar	30	password

Tabel 4.1 Tabel Database Login

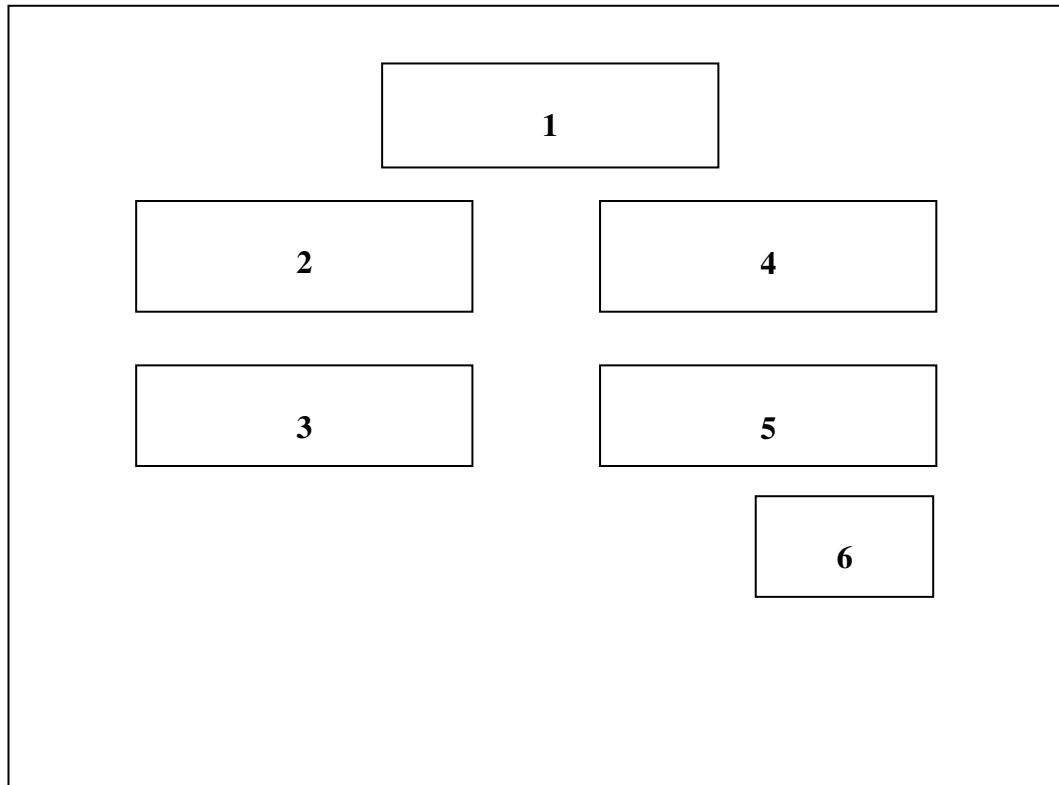
2. Tabel *database SMS*

<i>Field name</i>	<i>Data Type</i>	<i>size</i>	Description
id	integer	15	id
sender	varchar	15	sender
isi	varchar	100	isi
time	timestamp	-	time

Tabel 4.2 Tabel Database SMS

4.2 Perancangan Antarmuka

4.2.1 Perancangan Antarmuka Menu *Login*



Gambar 4.5 Perancangan Antarmuka Menu *Login*

Keterangan :

1. Pada label ini akan diisikan dengan *text* judul (*login*).
2. Pada label ini akan diisikan dengan *text* *username*.
3. Pada label ini akan diisikan dengan *text* *password*.
4. Pada *textbox* ini akan diisikan dengan nama pengguna (*username*) untuk bisa *login*.
5. Pada *textbox* ini akan diisikan dengan kata sandi (*password*) untuk bisa *login*.
6. Tombol *login* untuk menampilkan menu utama, jika *username* dan *password* di dalam *textbox* tersebut benar.

4.2.2 Perancangan Antarmuka Menu Utama

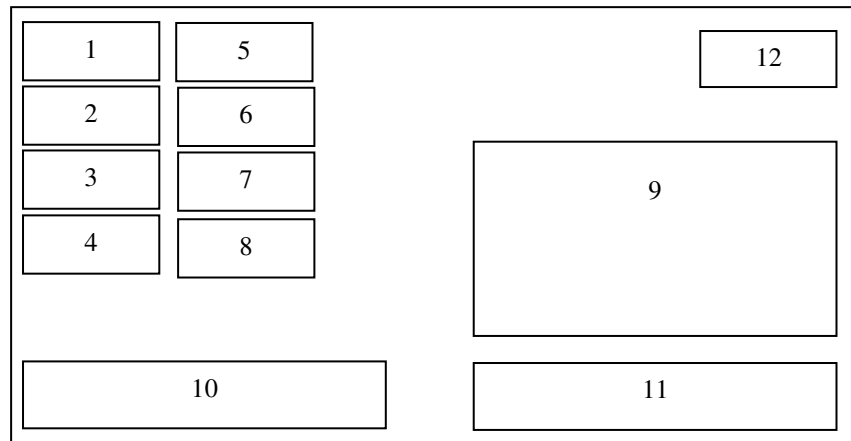
1	2	3
4		
5		

Gambar 4.6 Perancangan Antarmuka Menu Utama

Keterangan :

1. Bagian pada nomor 1 berisikan dengan jam.
2. Bagian pada nomor 2 berisikan dengan label sistem informasi *monitoring* bus.
3. Bagian pada nomor 3 berisikan dengan hari dan tanggal.
4. Bagian pada nomor 4 berisikan dengan peta dan posisi bus beserta kecepatan dan jarak.
5. Bagian pada nomor 5 berisikan dengan prediksi tibanya bus ke Terminal Banda Aceh yang berangkat dari Medan.

4.2.3 Perancangan Antarmuka GPS Android



Gambar 4.7 Perancangan Antarmuka GPS Android

Keterangan :

1. Pada *textbox* 1 merupakan *text latitude*.
2. Pada *textbox* 2 merupakan *text longitude*.
3. Pada *textbox* 3 merupakan *text kecepatan (speed)*.
4. Pada *textbox* 4 merupakan *text jarak (distance)*.
5. Pada *textbox* 5 merupakan data *latitude* dari GPS.
6. Pada *textbox* 6 merupakan data *longitude* dari GPS.
7. Pada *textbox* 7 merupakan data kecepatan dari GPS.
8. Pada *textbox* 8 merupakan data jarak dari GPS.
9. Pada *textbox* 9 merupakan gabungan dari semua data yang ada pada nomor 5-8 untuk dikirim ke *server*.
10. Pada *textbox* 10 merupakan keterangan posisi bus.
11. Pada *textbox* 11 merupakan nomor GSM tujuan untuk dikirim data dari GPS.
12. Pada *textbox* 12 merupakan tombol *switch on/off*.

4.3 Implementasi Program

Implementasi merupakan langkah-langkah dalam menerapkan hasil dari rancangan ke dalam bahasa pemrograman. Dalam membangun dan melakukan pengujian tersebut, maka akan membutuhkan perangkat keras maupun perangkat lunak. Adapun perangkat keras dan perangkat lunak yang digunakan penulis dalam membangun dan melakukan pengujian program penelitian ini yaitu :

A. Perangkat Keras

1. *Notebook* Acer Aspire E1-472G
2. *Smartphone* Samsung Galaxy SII
3. Modem GSM Huawei E1823

B. Perangkat Lunak

1. Sistem Operasi *Windows* 8.1 *Enterprise* 64bit
2. XAMPP versi 1.8.3
3. *Mozilla Firefox* versi 23
4. NowSMS versi 2013.11.05
5. *Macromedia Dreamweaver* MX 2004
6. *Eclipse* IDE versi 20140321

4.4 Penjelasan, Penulisan *Script* Program Dan Hasilnya

4.4.1 Koneksi

Agar dapat mengambil data GPS dari *database*, maka *database* harus saling terkoneksi. Berikut adalah *script* yang digunakan untuk membuat agar *database* saling terkoneksi :

```
<?php
ini_set('display_errors',FALSE);
$host="localhost";
$user="root";
$pass="";
$db="terminal";
$koneksi=mysql_connect($host,$user,$pass);
mysql_select_db($db,$koneksi);
if ($koneksi){
}
else{
?>
<script language="javascript">alert("Gagal Koneksi Database MySql
!!")</script>
<?php
} ?>
```

4.4.2 Login

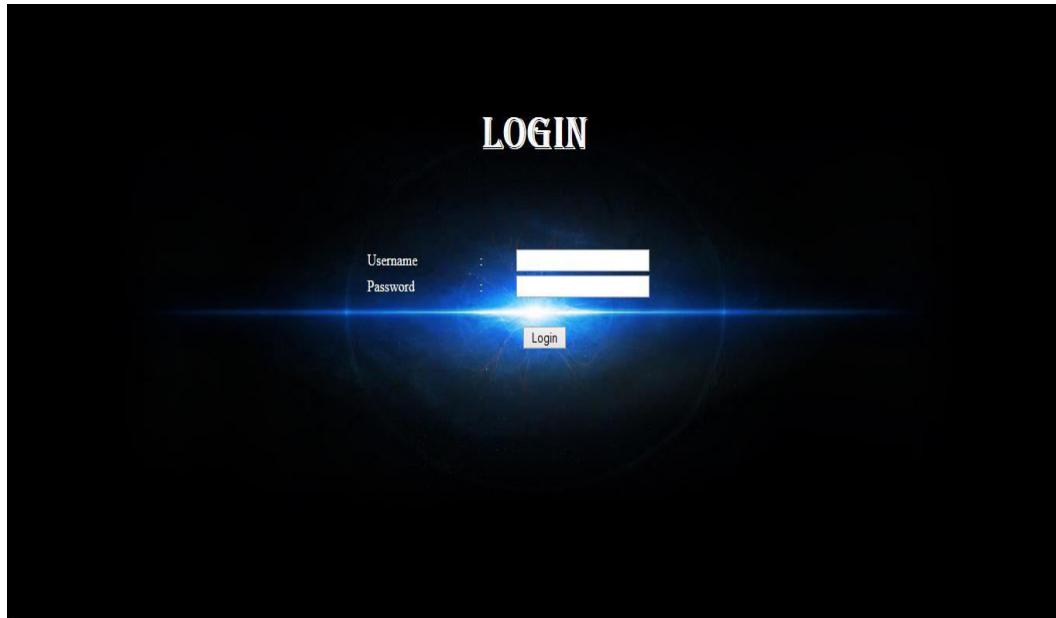
Untuk melakukan *monitoring* terhadap bus, *admin* harus melakukan *login* terlebih dahulu agar bisa masuk kedalam sistem *monitoring* tersebut guna bisa melihat posisi keberadaan bus beserta kecepatan dan jaraknya dan juga untuk

mengetahui perkiraan waktu tibanya bus ke terminal Banda Aceh yang berangkat dari Medan.

Berikut adalah *script* program untuk *login* :

```
<?php
include "conf.php";
if (isset($_POST['password'])) {
    $username = mysql_real_escape_string($_POST['username']);
    $password = mysql_real_escape_string($_POST['password']);
    $squery = "SELECT * from login where username ='$username' and password
    ='$password'";
    $hasil = mysql_query($squery);
    if (mysql_num_rows($hasil)) {
        ?>
        <script language="javascript">
        document.location='index.php';
        </script>
        <?php
        }
    else
    echo"<center><h1>Login Gagal<h1><br><a href=javascript:history.back()>
    <h6>Coba Lagi<h6> </a></center>";
    mysql_close();
} ?>
```

Berikut ini adalah tampilan untuk melakukan *login* :



Gambar 4.8 Tampilan Login

4.4.3 Posisi Bus

Pada saat *database* menerima data dari GPS pada *mobile* Android yang berisi koodinat lokasi yaitu *latitude* dan *longitude* beserta kecepatan dan jarak, maka posisi bus akan selalu berpindah-pindah karena pengiriman data dari GPS dikirim secara *realtime* (terus-menerus) setiap satu menit sekali. Pengiriman data dari GPS disimpan kedalam *database*. Posisi bus beserta kecepatan dan jarak akan ditampilkan pada *google maps* dalam bentuk *icon* bus. Halaman sistem *monitoring* akan selalu memuat ulang (*refresh*) secara otomatis setiap 60 detik (satu menit) sekali ketika data posisi bus dari GPS masuk kedalam *database*. Ketika bus tersebut memasuki area terminal maka dihalaman sistem monitoring akan tampil teks "Bus sudah tiba diterminal Banda Aceh" maupun "Bus sudah

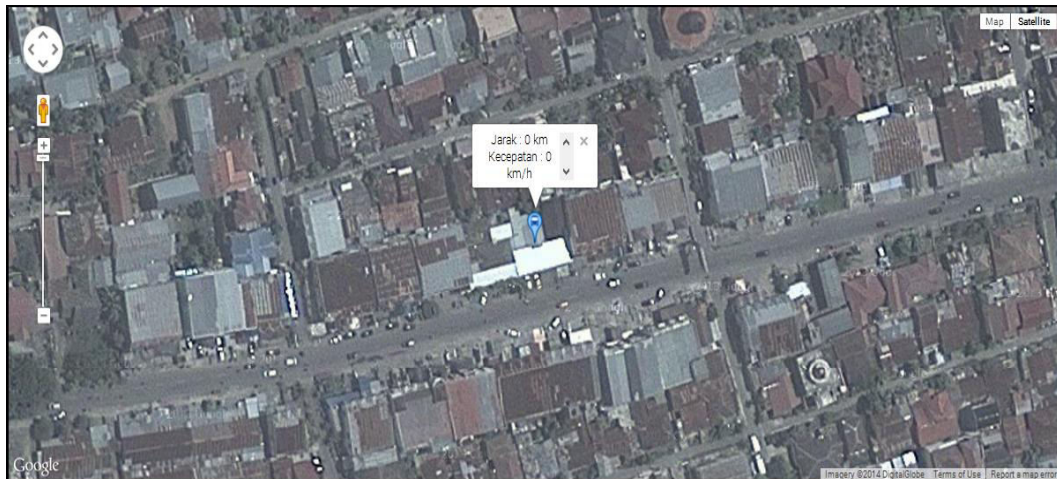
tiba di terminal Medan” yang selalu berkedip. Sedangkan ketika bus keluar dari area terminal maka teks tersebut akan hilang.

Berikut adalah *script* program peta dan posisi bus :

```
<script>
type="text/javascript"src="http://maps.google.com/maps/api/js?sensor=false"><
/script>
<script type="text/javascript">
function initialize() {
    var position = new google.maps.LatLng(<?php echo $lat; ?>, <?php echo
$long; ?>);
    var myOptions = {
        zoom: 15,
        center: position,
        mapTypeId: google.maps.MapTypeId.SATELLITE
    };
    var map = new google.maps.Map(
        document.getElementById("map_canvas"),
        myOptions);
    var marker = new google.maps.Marker({
        position: position,
        map: map,
        icon:"busicon.png"
    });
    var contentString = '<b> Posisi Bus <br>Jarak : <?php echo $jarak1; ?> </br>
<br>Kecepatan : <?php echo $speed; ?></br>';
    var infowindow = new google.maps.InfoWindow({
        content: contentString
    });
    google.maps.event.addListener(marker, 'click', function(){
        infowindow.open(map,marker);
    });
}
```

```
});
}
</script>
```

Berikut adalah tampilan peta dan posisi bus :



Gambar 4.9 Tampilan peta dan posisi bus

4.4.4 Tampilan Antarmuka Sistem Monitoring Bus



Gambar 4.10 Tampilan Antarmuka Sistem Monitoring Bus

4.4.5 Tampilan Ketika Bus Sudah Tiba Di Terminal Banda Aceh



Gambar 4.11 Tampilan Ketika Bus Sudah Tiba Di Terminal Banda Aceh

4.4.6 Tampilan Ketika Bus Sudah Tiba Di Terminal Medan



Gambar 4.12 Tampilan Ketika Bus Sudah Tiba Di Terminal Medan

4.4.7 Tampilan Ketika Bus Sedang Dalam Perjalanan



Gambar 4.13 Tampilan Ketika Bus Sedang Dalam Perjalanan dan berada di kawasan pegunungan Seulawah.



Gambar 4.14 Tampilan Ketika Bus Sedang Dalam Perjalanan dan berada di kawasan Bireun.

4.4.8 Prediksi Waktu Tiba Bus Di Terminal

Pada fitur ini akan menampilkan prediksi waktu akan tibanya bus yang berangkat dari Medan ke terminal Banda Aceh. Fitur prediksi ini dibuat dengan mengambil data jarak bus dan kecepatan laju bus tersebut. Data jarak bus ke terminal Banda Aceh dibagi dengan kecepatan laju bus maka akan menghasilkan waktu tempuh bus ke terminal Banda Aceh. Bila kecepatan laju bus menurun maka otomatis waktu tempuh akan lama dan ketika kecepatan laju bus tinggi maka waktu sampainya akan lebih cepat.

Berikut adalah *script* program untuk menampilkan prediksi tibanya bus diterminal Banda Aceh yang berangkat dari Medan :

```
<?php
    $dis = (int) $jarak/1000;
    $t = ($dis/($speed/60));

    $jam=$t/60;
    $menit= $t%60;

    $waktu = explode(".", $jam);
    $jam = $waktu[0];

    $waktu1 = explode(".", $menit);
    $menit = $waktu1[0];

    echo "Bus Diperkirakan Akan Tiba &nbsp; &plusmn; &nbsp;". $jam."
jam ".$menit." Menit lagi";
?>
```

Berikut adalah tampilan fitur prediksi tibanya bus di terminal :



Gambar 4.15 Tampilan Prediksi Waktu Tiba Bus Di Terminal

4.4.9 Tampilan Struktur Tabel Dan Isi Database

	id	sender	isi	time
☐ Edit Salin Hapus	1	+6289607277270	5.562366480007768,95.32640056684613,0,3610.144,	2014-05-15 12:34:00
☐ Edit Salin Hapus	2	+6289607277270	5.562333664856851,95.32633946277201,0,3607.003,	2014-05-15 12:35:00
☐ Edit Salin Hapus	3	+6289607277270	5.562326498329639,95.32638774253428,0,3605.833,	2014-05-15 12:35:00
☐ Edit Salin Hapus	4	+6289607277270	5.5623494647443295,95.32632957212627,0,3608.8245,	2014-05-15 12:36:00

Gambar 4.16 Tampilan Struktur Tabel Dan Isi Database

4.5 Aplikasi GPS Android

Aplikasi GPS *Tracking* di *smartphone* berbasis Android dibangun menggunakan Android ADT dan Eclipse IDE. Aplikasi ini berfungsi sebagai alat untuk *monitoring* yang ditempatkan didalam bus. Aplikasi GPS ini mengambil koordinat lokasi bus dari satelit juga kecepatan laju bus dan juga menghitung jarak antara titik lokasi posisi keberadaan bus berada ke terminal Banda Aceh yang sudah diprogramkan didalam aplikasi ini. Aplikasi GPS *Tracking* ini saya beri nama *Bus Track*. Aplikasi ini akan mengirim data GPS ke server setiap satu menit namun apabila posisi bus tidak bergerak kurang dari 10 meter maka aplikasi tidak akan memperbarui koordinat GPS dan tidak akan mengirimkan SMS ke server. Ketika aplikasi berhasil mengirim SMS maka akan keluar pesan *toast (popup)* “SMS Terkirim” dan bila SMS gagal dikirimkan yang disebabkan oleh faktor jaringan atau tidak adanya pulsa maka akan tampil pesan “SMS Gagal, Harap

Coba Lagi!”. Aplikasi Android ini akan otomatis berhenti mengirimkan sms bila bus sudah memasuki area terminal yaitu pada *range* yang sudah ditetapkan dari titik luar terminal (pintu gerbang) sampai ketitik lokasi bus parkir (titik dalam), namun ketika bus tersebut keluar dari area tersebut maka aplikasi akan kembali mengirimkan sms. Untuk mematikan atau menunda pengiriman SMS secara manual juga penulis membuat sebuah tombol *switch* on/off. Ketika pengguna hendak menutup atau keluar dari aplikasi maka akan keluar pesan “Tekan tombol sekali lagi untuk keluar”.

4.6 Tampilan Aplikasi GPS Android

4.6.1 Tampilan Ketika Bus Sudah Tiba Di Terminal Medan



Gambar 4.17 Tampilan GPS Android Ketika Bus Sudah Tiba Di Terminal Medan

4.6.2 Tampilan Ketika Bus Sudah Tiba Di Terminal Banda Aceh



Gambar 4.18 Tampilan GPS Android Ketika Bus Sudah Tiba Di Terminal Banda Aceh

4.6.3 Tampilan Ketika Bus Sedang Dalam Perjalanan



Gambar 4.19 Tampilan GPS Android Ketika Bus Dalam Perjalanan

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan dari hasil pengujian sistem yang sudah dilakukan, dapat diambil kesimpulan bahwa :

1. Setelah melalui beberapa tahapan pengujian, sistem yang dibangun berjalan dengan baik sesuai dengan keinginan yang diharapkan.
2. Dari hasil *range* (jarak) yang ditetapkan pada aplikasi GPS Android, ketika bus memasuki area terminal maka GPS otomatis akan berhenti mengirimkan SMS yang berisi data koordinat, kecepatan serta jarak dan menampilkan teks bahwa bus telah sampai ke terminal.

5.2 Saran

Berdasarkan pengujian yang sudah dilakukan, saran yang dapat disampaikan oleh penulis yaitu :

1. Untuk menjalankan program sistem *monitoring* ini disarankan menggunakan koneksi internet dengan *bandwidth* kurang lebih 1 Mbps dikarenakan untuk data *google maps* membutuhkan koneksi data yang cepat dan stabil.
2. Disarankan menggunakan *browser Mozilla Firefox* versi 23 kebawah agar fitur “*blink*” pada teks ketika bus sudah tiba di terminal berfungsi dengan baik. Pihak *Firefox* telah menghapus fitur “*blink*” pada *Mozilla Firefox* versi 23 keatas.

DAFTAR PUSTAKA

- Herlambang, Soendoro, Haryanto Tanuwijaya, 2005, *Sistem Informasi, Konsep, Teknologi & Manajemen*, Graha Ilmu, Yogyakarta
- Jogiyanto, H.M, 1995, *Analisa dan Desain Sistem Informasi*, Andi, Yogyakarta
- Luke Welling, Laura Thomson, 2003, *PHP & MySQL Web Development*, Sam Publishing
- Sutarma, 2007, *Membangun Aplikasi Web dengan PHP & MySQL*, Graha Ilmu, Yogyakarta
- Sunyoto, Andi, 2005, Jurnal: Global Positioning System (GPS) Overview", Amikom, Yogyakarta
- Abidin, Hasanuddin Z, 2005, *Penentuan Posisi Dengan GPS dan Aplikasinya*, Paramitha Padnya, Bandung.
- Elcom, 2010, *Google Android*, Andi, Yogyakarta
- Safa'at, Nazruddin, H, 2011, *Android: Pemograman Aplikasi Mobile Smartphone dan Tablet*, Informatika, Bandung
- Widiyaksono, Pandu, 2010, Pengertian Google Maps API
<http://blog.xinthinx.us/2010/06/pengertian-google-maps-api.html>. (diakses tanggal 24 Februari 2014)
- [http://id.wikipedia.org/wiki/Eclipse_\(perangkat_lunak\)](http://id.wikipedia.org/wiki/Eclipse_(perangkat_lunak)) (diakses tanggal 25 Februari 2014)
- http://id.wikipedia.org/wiki/SMS_Gateway (diakses tanggal 25 Februari 2014)

BIODATA PENULIS

Nama : Mushalli
NIM : 09111020
Jurusan : S-1 Teknik Informatika
Tempat / Tanggal Lahir : *Sigli*, 27 April 1991
Jenis Kelamin : Laki-laki
Alamat : Jln. Setia No. 18 Kelurahan Keuramat
Kuta Alam, Banda Aceh.
Agama : Islam
Email : ayierichard@gmail.com
Social Network : Twitter : @ayierichard
Path : Ayie Richard
Instagram : ayierichard

Riwayat Pendidikan

SD : SD Negeri Mesjid Lameue
MTsN : MTs Negeri 1 Sakti
SMA : SMA Negeri 1 Sakti

LAMPIRAN

1. Source Code Server

```
<?
include "conf.php";
?>

<link rel="stylesheet" href="css.css" type="text/css"
media="screen">
<html>

<meta http-equiv=refresh content=50;>

<title>Sistem Monitoring Bus</title>

<style type="text/css">
<!--
.style18 {font-size: xx-large; font-weight: bold; }
.style19 {color: #FF0000}
.style20 {font-size: xx-large}
.style21 {font-family: Algerian}
.style22 {font-size: larger}
.style23 {color: #00FF00}
.style28 {font-size: large}

-->
</style>

<table width="90" height="585" border="1"
bordercolor="#FF0000" class="bgindex">
<tr>
<td height="61" colspan="5" align="left">
<table width="1303" height="60" align="center" >
<tr>
```

```

<!-- Tabel jam !-->
    <td width="66" height="63" align="left">
        <script src="http://www.clocklink.com/embed.js">
</script>
        <script type="text/javascript"
language="JavaScript">obj=new Object;obj.clockfile="5010-
blue.swf";obj.TimeZone="JOG";obj.width=222;obj.height=66;obj
.wmode="transparent";showClock(obj);</script> </td>

<!-- Tabel Informasi Bus !-->
    <td width="988" > <span class="style19"> <span
class="style20"> <span class="style21"><span
class="style22"> <span class="style23"> <marquee
behavior="alternate"> <strong> ...: SISTEM MONITORING BUS
:... </strong> </marquee> </span> </span> </span> </span>
</span> </td>

<!-- Tabel Menampilkan hari dan tanggal !-->
    <td width="233" class="style19"><font size="+1"
color="#FFFFFF">
        <?PHP
            date_default_timezone_set('Asia/Jakarta');
            $namaHari = array("Ahad", "Senin", "Selasa", "Rabu",
"Kamis", "Jumat", "Sabtu","Minggu");
            $namaBulan = array("Januari", "Februari", "Maret",
"April", "Mei", "Juni", "Juli", "Agustus", "September",
"Oktober", "November", "Desember");
            $today = date('l, F j, Y');
            $sekarang = $namaHari[date('N')] . ", " . date('j') .
" " . $namaBulan[(date('n')-1)] . " " . date('Y');
            echo $sekarang;
        ?></font>
    </td>

```

```

        </tr>
    </table>
</td>
</tr>

<!------- Tabel Peta -----!-
->
    <tr>
        <th height="200" colspan="3" align="center">
            <table width="1340" height="246">
                <tr>

<!-- Tabel Peta      !-->
            <td width="1026" height="100" align="center">

<body>
<div>

<?php
    include "conf.php";

    $q="select * from sms order by id desc limit 1";
    $c=mysql_query($q);
    while($data = mysql_fetch_array($c))
    {
        $isi = $data['isi'];
    }
?>

<!--</div>
</body>!-->
<form>

<?php

```

```

include "conf.php";

// mengambil isi database terbaru
    $q="select * from sms order by id desc limit 1";
    $c=mysql_query($q);
    while($data = mysql_fetch_array($c))
    {

        $pecah = explode(",", $isi);
//mencari element array 0
        $lat = $pecah[0];
        $long = $pecah[1];
        $speed = $pecah[2];
        $jarak = $pecah[3];
        $j = $jarak/1000;
        $jarak1 = floor($j*100)/100;

        echo "<input type='hidden' name='latitude' id='lat'
value=".$lat.">";
        echo "<input type='hidden' name='longitude' id='long'
value=".$long.">";
        echo "<input type='hidden' name='speed' id='speed'
value=".$speed.">";
        echo "<input type='hidden' name='jarak' id='jarak'
value=".$jarak1.">";
    }
?>
</form>

<?php
include "conf.php";

```

```

        if (abs(5.546622 - $lat) <= 0.002 &&
abs(95.330231 - $long) <= 0.0003)
        {
            echo "<blink> <b> Bus sudah
tiba di Terminal Banda Aceh </b> </blink>";
            $count = ("results.txt");
            $clicks = file($count);
            $clicks[0] = 0;

            $fp = fopen($count , "w");
            fputs($fp , "$clicks[0]");
            fclose($fp);
        }

        else if (abs(5.514646 - $lat) <=
0.00001 && abs(95.330115 - $long) <= 0.0001)
        {
            echo " <blink> <b> <font
size='5' color='red'> Bus sudah tiba di Terminal Medan
</font> </b> </blink> ";
            $count = ("results.txt");
            $clicks = file($count);
            $clicks[0] = 0;

            $fp = fopen($count , "w");
            fputs($fp , "$clicks[0]");
            fclose($fp);
        }

if (p == p && d == d && $pecah[2]<=2)
{
    $count = ("results.txt");
    $clicks = file($count);
    $clicks[0]++;
}

```

```

        $fp = fopen($count , "w");
        fputs($fp , "$clicks[0]");
        fclose($fp);
    }

    if ($pecah[2]>=3)
    {
        $count = ("results.txt");
        $clicks = file($count);
        $clicks[0] = 0;

        $fp = fopen($count , "w");
        fputs($fp , "$clicks[0]");
        fclose($fp);
    }

    if($clicks[0]==4)
    {

        echo "<script>location.href =

; </script>";

        $count = ("results.txt");
        $clicks = file($count);
        $clicks[0] = 0;

        $fp = fopen($count , "w");
        fputs($fp , "$clicks[0]");
        fclose($fp);
    }

?>

</form>

```

```

<!--<head>!-->
<script
type="text/javascript"src="http://maps.google.com/maps/api/j
s?sensor=false"></script>
<script type="text/javascript">
    function initialize() {
        var position = new google.maps.LatLng(<?php echo $lat;
?>, <?php echo $long; ?>);
        var myOptions = {
            zoom: 15,
            center: position,
            mapTypeId: google.maps.MapTypeId.SATELLITE
        };
        var map = new google.maps.Map(
            document.getElementById("map_canvas"),
            myOptions);
        var marker = new google.maps.Marker({
            position: position,
            map: map,
            icon:"34.png"
        });

        var contentString = '<b>Posisi Bus BL 7755 AA
</b><br>Jarak : <?php echo $jarak1; ?> km </br>Kecepatan :
<?php echo $speed; ?> km/h </br>';
        var infowindow = new google.maps.InfoWindow({
            content: contentString
        });
        google.maps.event.addListener(marker, 'click',
function() {
            infowindow.open(map,marker);

        });
    }
}

```

```

</script>
<!--</head>!-->
<body onload="initialize()">
<div id="map_canvas" style="width:100%; height:500px"></div>
</body>
</td>

        </tr>
    </table>
</th>
</tr>

<!------- TABLE KETERANGAN -----
----!-->
    <tr height="5">
        <th font color="white" align="center"> <span
style="color:#FFFFFF;" class="style44"> Prediksi Tibanya Bus
</span></th>
    </tr>

<!------- TABLE MENGHITUNG PREDIKSI TIBANYA BUS
-----!-->
    <td width="452" ><span style="color:#FFFFFF;"
class="style18"> <marquee behavior="scroll">
<?php
    $dis = (int) $jarak/1000;    //jarak dalam meter / 1000
= km
    $t = ($dis/($speed/60));    //dalam menit

    $jam=$t/60;                //menghitung jam
    $menit= $t%60;             //menghitung menit

    $waktu = explode(".", $jam);
    $jam = $waktu[0];

```

```

        $waktu1 = explode(".", $menit);
        $menit = $waktu1[0];

        echo "Bus Diperkirakan Akan Tiba &nbsp; &plusmn;
&nbsp;". $jam." jam ". $menit." Menit lagi";
?>
</marquee> </span>
    </td>

</table>
</html>

```

2. Source Code Android

```

package com.bus.track;

import java.util.TimerTask;

import com.google.android.maps.GeoPoint;
import com.google.android.maps.MapActivity;
import com.google.android.maps.MapView;

import android.app.ProgressDialog;
import android.content.Intent;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.CountDownTimer;
import android.os.Handler;
import android.telephony.SmsManager;
import android.widget.CompoundButton;
import android.widget.EditText;
import android.widget.Switch;
import android.widget.TextView;
import android.widget.Toast;

public class Maps extends MapActivity implements
LocationListener {

    int running ;
    // ProgressDialog ringProgressDialog =
ProgressDialog.show(Maps.this, "Please wait ...", "Loading
Location ...", true);

```

```

        LocationManager locmanager ;
        LocationListener locListener = new LocationListener() {

            @Override
            public void onLocationChanged(Location location)
            {

                // TODO Auto-generated method stub

                double latitude = location.getLatitude();
                double longitude = location.getLongitude();
                int speed1 = (int) location.getSpeed();
                int speedkm = (speed1*60*60)/1000;
                //double accuracy = location.getAccuracy();

                // 5.548298,95.316183 bj depan
                // 5.548306,95.316157 bj depan 1
                // 5.548405,95.31584 bj belakang
                // 5.548426,95.315762 bj belakang
                double a1 = Math.abs(5.548298-latitude);
                // latitude terminal banda aceh
                double a2 = Math.abs(5.548405-latitude);
                // latitude terminal medan
                double a3 = Math.abs(95.316183-longitude);
                //lonitude terminal banda aceh
                double a4 = Math.abs(95.31584-longitude);
                // logitude terminal medan

                //    ringProgressDialog.dismiss();

                lat = (TextView)
                findViewById(R.id.latitude);
                longt = (TextView)
                findViewById(R.id.longitude);
                speed = (TextView) findViewById(R.id.speed);
                // accuracy = (TextView)
                findViewById(R.id.accur);

                lat.setText(+latitude+",");
                longt.setText(+longitude+",");
                speed.setText(+speedkm+",");
                //
                accuracy.setText("Acurracy      : "+" "
                +accuracy+" m");

                String a=speed.getText().toString();

```

```

        double distance;

        Location locationA = new Location("point
A");

        double lat = 5.529690;
        double log = 95.328293;

        locationA.setLatitude(lat);
        locationA.setLongitude(log);

        Location locationB = new Location("point
B");

        locationB.setLatitude(latitude);
        locationB.setLongitude(longitude);

        //locationB.setLatitude(latitude);
        //locationB.setLongitude(longitude);

        distance = locationA.distanceTo(locationB);

        float c = (float) distance;

        dist = (TextView) findViewById(R.id.dist);
        dist.setText(+c+",");

        String b = dist.getText().toString();

        estimate = (TextView)
findViewById(R.id.estimate);
        int speed = speed1 ;
        int estimated = (int) (distance / speed);

        estimate.setText(""+estimated+" minute");

        viewtext.setText(lat.getText().toString()+
longt.getText().toString()+a+b) ;

        start =true;

        if((a1 <= 0.00001) && (a3 <= 0.00001)
){

            running=0;
            note.setText("Bus Sudah Tiba Di
Terminal Banda Aceh");

        }

```

```

//      if((a2 <= 0.00001) && (a4 <= 0.0001)
){
      if((a2 <= 3.5927609) && (a4 <=
98.6269659) ){

      running=0;
      note.setText("Bus Sudah Tiba Di
Terminal Medan");

      }

      else{

      running=1;
      note.setText("bus dalam perjalanan");

      }

}

@Override
public void onProviderDisabled(String provider) {
    // TODO Auto-generated method stub

}

@Override
public void onProviderEnabled(String provider) {
    // TODO Auto-generated method stub

}

@Override
public void onStatusChanged(String provider, int
status, Bundle extras) {
    // TODO Auto-generated method stub

}

};

TextView lat;
TextView longt;
TextView speed;
TextView accuracy;

```

```

        TextView viewtext;
        TextView dist;
        TextView estimate;

        MapView map;
        GeoPoint p;

        boolean start=true ;
        TextView timer;
        CountdownTimer a=null;
        private EditText numb;
        private TextView note;
        TimerTask timertask;

        private Switch switch1;
        private int backButtonCount;

        ProgressDialog dialog ;

        int count=1;
        private Handler handler = new Handler();

        private Runnable runnable = new Runnable() {
            private SmsManager smsManager;

            @Override
            public void run() {
                final String phoneNo =
numb.getText().toString(); //String phoneNo =
"+6285311292934";

                String sms = (String)
viewtext.getText();

                try {

                    smsManager =
SmsManager.getDefault();

                    smsManager.sendTextMessage(phoneNo, null, sms, null,
null); // kirim sms

                    Toast.makeText(getApplicationContext(), "SMS
Terkirim!",

                    Toast.LENGTH_LONG).show();

```

```

        } catch (Exception e) {

            Toast.makeText(getApplicationContext(),
                                                                    "SMS Gagal,
Harap Coba Lagi!",
            Toast.LENGTH_LONG).show();
                                                                    e.printStackTrace();

        }handler.postDelayed(this, 15000);//50
seconds is 50000 milliseconds
        }
    };

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);

        note = (TextView) findViewById(R.id.note);
        lat = (TextView) findViewById(R.id.latitude);
        numb = (EditText) findViewById(R.id.numb);
        longt = (TextView) findViewById(R.id.longitude);
        // map = (MapView) findViewById(R.id.mapview);

        locmanager =
        (LocationManager) getSystemService(LOCATION_SERVICE);

        //locmanager.requestLocationUpdates(LocationManager.GPS_PROVID
        IDER, 1L, 1.0f, locListener);

        locmanager.requestLocationUpdates(LocationManager.GPS_PROVID
        ER, 20000, 10, locListener); // update setiap 10 meter dan
        15 detik

        viewtext = (TextView)
        findViewById(R.id.sendmessage);

        //ringProgressDialog.setCancelable(false);

```

```

        switch1= (Switch) findViewById(R.id.switch1);
        switch1.setChecked(false);

        switch1.setOnCheckedChangeListener(new
CompoundButton.OnCheckedChangeListener() {

            @Override
            public void
onCheckedChanged(CompoundButton buttonView, boolean
isChecked) {
                // TODO Auto-generated method stub
                //dialog =
ProgressDialog.show(Maps.this, "", "Loading location...
Please wait...", true);

                if
(buttonView.isChecked()==true){

if(running==1){handler.postDelayed(runnable, 5000);} //5000=
5 detik

                else
if(running==0){handler.removeCallbacks(runnable);}

                }

                else if(!buttonView.isChecked()){

                    handler.removeCallbacks(runnable);

                }
            }

        });

    }

    @Override
    public void onStop() {
        super.onStop();
        handler.removeCallbacks(runnable);
    }
}

```

```

protected boolean isRouteDisplayed() {
    return false;
}

@SuppressWarnings("deprecation")
public void onBackPressed()
{
    //dialog.dismiss()
    if(backButtonCount >= 1)
    {
        Intent intent = new Intent(Intent.ACTION_MAIN);
        intent.addCategory(Intent.CATEGORY_HOME);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
        System.runFinalizersOnExit(true);

        android.os.Process.killProcess(android.os.Process.myPid());

        Maps.this.finish();

        return;
    }
    else
    {
        Toast.makeText(this, "Tekan tombol kembali
sekali lagi untuk keluar !", Toast.LENGTH_SHORT).show();
        backButtonCount++;
    }
}

@Override
public void onLocationChanged(Location location) {
    // TODO Auto-generated method stub

}

@Override

```

```
    public void onProviderDisabled(String provider) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onProviderEnabled(String provider) {
        // TODO Auto-generated method stub

    }

    @Override
    public void onStatusChanged(String provider, int
status, Bundle extras) {
        // TODO Auto-generated method stub

    }

}
```