

BAB III

ANALISIS DAN PERANCANGAN

Dalam pembuatan aplikasi ini menerapkan konsep SDLC (*Systems Development Life Cycle* (Siklus Hidup Pengembangan Sistem) yang berfungsi untuk menggambarkan tahapan-tahapan utama dan langkah-langkah dari setiap tahapan. Langkah-langkah yang akan dilakukan dalam pembuatan Aplikasi Monitoring Armada Bus Menggunakan GPS *Tracking* pada *Smartphone* Android adalah sebagai berikut:

3.1 Analisis Permasalahan

3.1.1 Identifikasi Masalah

Bus ekonomi antar kota antar propinsi adalah salah satu transportasi umum yang ada di Jawa Timur. Setiap harinya bus beroperasi sesuai dengan trayek tetap masing-masing yang sebelumnya sudah disetujui melalui surat keputusan oleh Dirjen Perhubungan Darat. Pada trayek antar kota antar propinsi jurusan Surabaya – Semarang dilayani oleh tiga perusahaan otobus.

Pemberangkatan bus dari terminal awal yaitu terminal Purabaya Surabaya sebelumnya sudah diatur secara resmi oleh Dinas Perhubungan, setiap bus sudah mempunyai jadwal tetap untuk jam keberangkatan baik dari terminal awal atau sebaliknya dari terminal Terboyo Semarang. Di lapangan para awak bus juga mempunyai kesepakatan agar jarak antara bus yang satu dengan yang lain tidak terlalu dekat, sehingga ada jeda waktu kedatangan pada saat mengangkut penumpang di perjalanan. Dalam hal ini jarak terdekat antar bus yang telah

disepakati adalah 5 kilometer. Pada saat bus depan sedang berhenti untuk ngetem atau istirahat makan, bus belakang boleh mendahului terlebih dahulu dalam istilah memakai jam bus depannya.

Berdasarkan hasil pengamatan di lapangan dan wawancara, pihak manajemen perusahaan otobus mempunyai kendala dalam mengetahui posisi armada masing-masing yang sedang beroperasi. Dari para awak bus ekonomi antar kota antar propinsi jurusan Surabaya – Semarang, di lapangan mereka mengalami kesulitan untuk mengetahui posisi bus lain dalam rute searah agar jarak antara bus yang satu dengan yang lain tidak terlalu dekat dalam hal ini kesepakatan dari para awak bus jarak terdekat antar bus adalah 5 kilometer. Sehingga ada jeda waktu kedatangan tiap bus pada saat mengangkut penumpang di perjalanan agar tidak menimbulkan masalah berebut penumpang.

Sehingga untuk dapat melakukan pemantauan armada bus yang sedang beroperasi, perlu adanya sistem yang dapat membantu pihak manajemen perusahaan otobus maupun para awak bus untuk mengetahui posisi keberadaan bus lain dalam rute searah agar jarak antara bus yang satu dengan yang lain tidak terlalu dekat. Perangkat yang memungkinkan untuk membangun sistem tersebut adalah perangkat *mobile* karena perangkat komunikasi ini dapat dengan mudah ditempatkan di dalam kendaraan.

Untuk melakukan penentuan lokasi suatu perangkat *mobile* terdapat banyak cara, namun yang kerap digunakan adalah *Cell Identification* (Cell-ID) dan *Global Positioning System* (GPS). Pemanfaatan Cell-ID memiliki kelebihan yaitu tidak membutuhkan perangkat tambahan karena memanfaatkan fasilitas penyedia jaringan yang memungkinkan seorang pelanggan seluler untuk

mengetahui posisinya terhadap *Base Transceiver Station* (BTS) terdekat. Namun akurasi dengan teknik Cell-ID ini sangat rendah yaitu berkisar 1-3 kilometer. GPS memiliki akurasi yang jauh lebih tinggi sehingga lebih cocok untuk diimplementasikan dalam sistem pelacakan meskipun teknik ini mengharuskan dibutuhkannya perangkat tambahan.

Dari uraian di atas, maka dapat ditarik kesimpulan bahwa untuk mempermudah melakukan pemantauan posisi armada bus, dapat dikembangkan sebuah aplikasi *GPS Tracking System* yang dapat memantau posisi armada bus yang sedang beroperasi untuk membantu manajemen perusahaan dan para awak bus yang sedang bekerja.

3.1.2 Hasil Analisis Sistem

Perusahaan otobus yang melayani trayek antar kota antar propinsi jurusan Surabaya – Semarang mempunyai kendala dalam mengetahui posisi armada masing-masing yang sedang beroperasi. Dari para awak bus mereka juga mengalami kesulitan untuk mengetahui posisi bus lain dalam rute searah agar jarak antara bus yang satu dengan yang lain tidak terlalu dekat. Sehingga ada jeda waktu kedatangan tiap bus pada saat mengangkut penumpang di perjalanan agar tidak menimbulkan masalah berebut penumpang.

Untuk mengatasi sulitnya melakukan pemantauan dan memperkirakan lokasi keberadaan posisi armada bus, maka dapat memanfaatkan fitur GPS pada *smartphone* Android dengan mengembangkan sebuah aplikasi *GPS Tracking System*. Sistem tersebut diharapkan dapat memantau posisi armada bus yang

sedang beroperasi dalam satu trayek untuk memudahkan pihak manajemen perusahaan dan para awak bus yang sedang bekerja.

Pencarian data dan pengolahan data yang dilakukan dengan cara merancang *database* dan membuat sistem. Data-data tersebut nantinya akan ditampung dan diolah oleh aplikasi sehingga dapat memberikan informasi dengan lebih tersruktur sehingga dapat bermanfaat bagi para *user*. Sistem ini nantinya akan dibagi menjadi dua bagian yaitu aplikasi *website* sebagai *server side* dan aplikasi *mobile* sebagai *client side*. Untuk itu diperlukan spesifikasi minimum perangkat keras dan perangkat lunak apa saja yang dibutuhkan:

1. Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang diperlukan untuk membangun Aplikasi Monitoring Armada Bus Menggunakan GPS *Tracking* pada *Smartphone* Android ditunjukkan pada tabel 3.1 di bawah ini:

Tabel 3.1 Kebutuhan Perangkat Keras Perancangan

No	Perangkat Keras	Spesifikasi
1	Processor	Intel Pentium Core i3, 2.1GHz
2	LCD	14.0" dengan resolusi 1366 x 768
3	VGA	Intel HD Graphic 3000
4	Ram	2 GB
5	Harddisk	320 GB

Sedangkan kebutuhan perangkat lunak untuk implementasi yaitu dengan menggunakan android emulator adalah sebagai berikut:

2. Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan untuk merancang Aplikasi Monitoring Armada Bus Menggunakan GPS *Tracking* pada *Smartphone* Android ditunjukkan pada tabel 3.2 di bawah ini:

Tabel 3.2 Kebutuhan Perangkat Lunak Perancangan

No	Perangkat Lunak
1	Java Development Kit (JDK)
2	IDE Eclipse
3	Android SDK
4	Android Development Tools
5	Macromedia Dreamweaver 8
6	XAMPP 1.7.2
7	WireframeSketcher

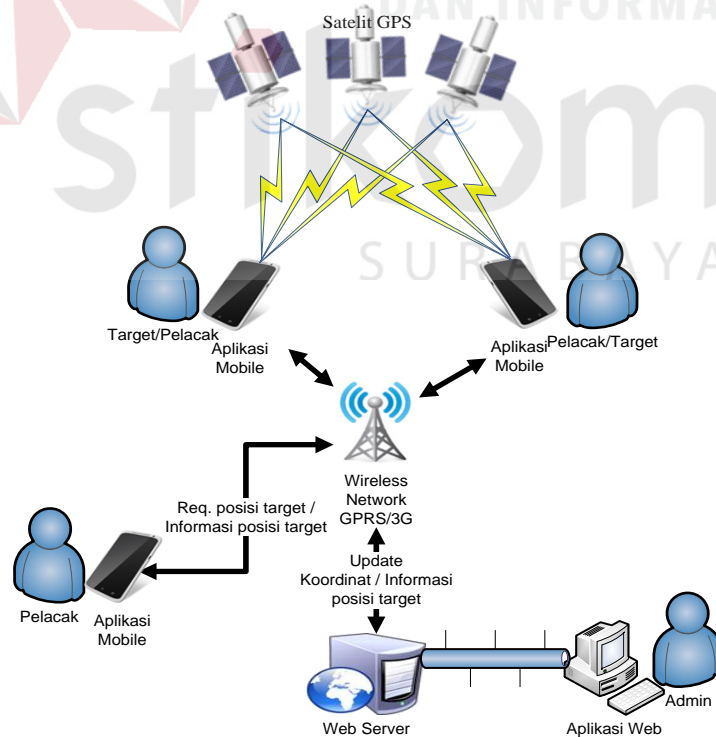
3.1.3 Arsitektur Aplikasi Monitoring Armada Bus Menggunakan GPS *Tracking* pada *Smartphone* Android

Arsitektur sistem yang ada dalam aplikasi ini secara garis besar adalah sebagai berikut:

1. GPS *module* yang terdapat pada *smartphone* Android akan memetakan posisi terhadap satelit GPS yang ada. Untuk dapat memetakan posisi perangkat *mobile* pada kendaraan maka dibutuhkan tiga atau lebih sinyal satelit GPS.
2. Setelah posisi perangkat *mobile* diketahui, lalu data posisi perangkat *mobile* akan dikirim ke *database server*. Data ini dikirim menggunakan media internet sehingga perangkat *mobile* diharuskan memiliki koneksi internet. Data yang dikirim ini adalah berupa data lokasi (*latitude* dan *longitude*), data waktu, dan data tanggal.

3. Selanjutnya data koordinat tersebut akan diproses oleh sistem dan kemudian disimpan dalam *database* yang terdapat pada *server*. Hal ini bertujuan agar nantinya dapat dilakukan pelacakan *histori* posisi perangkat *mobile* dikarenakan data koordinat yang disimpan di *server* akan terus di *update* dalam selang waktu tertentu sesuai dengan pengaturan *interval*.
4. Dari data yang ada maka dapat dilakukan pemantauan terhadap posisi *smartphone*. Yang dimaksud pemantauan disini adalah menampilkan posisi perangkat *mobile* pada Google Maps API berdasarkan data posisi *smartphone* yang didapatkan dari GPS dan telah disimpan ke dalam *database*. Pemantauan dapat dilakukan melalui aplikasi pada *smartphone* dan melalui *website*.

Arsitektur Aplikasi Monitoring Armada Bus Menggunakan GPS Tracking pada *Smartphone* Android dapat dilihat pada Gambar 3.1 di bawah ini:

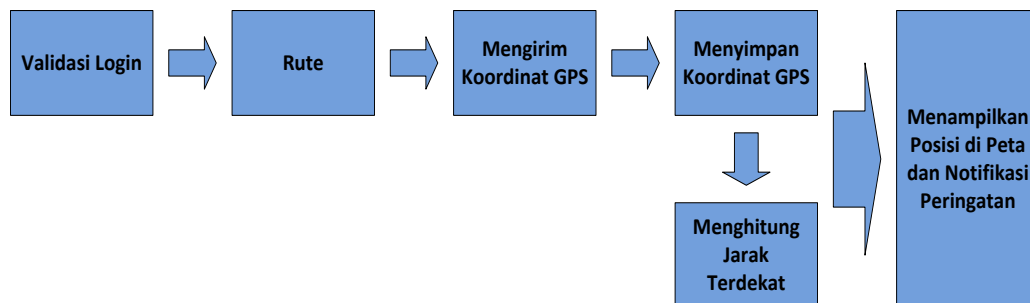


Gambar 3.1 Arsitektur Sistem Aplikasi Monitoring Armada Bus Menggunakan GPS Tracking pada *Smartphone* Android

3.2 Perancangan Sistem

Atas dasar hasil analisis yang telah dilakukan, maka selanjutnya dapat diusulkan suatu rancangan sistem berbasis web (*online*) dan *mobile* dengan dukungan fasilitas GPS yang ada pada perangkat *smartphone* Android yang mampu mengetahui koordinat posisi dari perangkat tersebut secara cepat, tepat, dan akurat, sehingga dapat dimanfaatkan untuk melakukan pemantauan keberadaan posisi armada bus pada saat beroperasi. Dengan demikian, diharapkan permasalahan yang dihadapi oleh manajemen perusahaan otobus dan para awak bus dalam melakukan pemantauan posisi armada bus dapat diatasi. Dalam merancang sistem ini ada beberapa tahap yang akan dilakukan, yaitu pembuatan blok diagram, IPO diagram, *flowchart*, *data flow diagram* (DFD), *entity relational diagram* (ERD), struktur *database* dan rancangan *Input/Output*.

Langkah pertama dilakukan pembuatan rancangan dalam bentuk blok diagram sebagai gambaran umum sistem menggunakan Microsoft Visio 2007. Desain blok diagram gambaran umum sistem dapat dilihat pada gambar 3.2 di bawah ini:



Gambar 3.2 Blok Diagram Gambaran Umum Sistem

Berikut adalah penjelasan mengenai blok diagram umum sistem di atas:

Sistem akan memvalidasi *login* dari *username* dan *password* yang dimasukkan *user* dalam hal ini adalah sopir sebagai pengguna aplikasi *mobile*. Kemudian sistem menyimpan rute pilihan dari sopir untuk membatasi apakah dari Surabaya–Semarang atau Semarang–Surabaya. Selanjutnya sistem mengirimkan koordinat perangkat GPS melalui internet untuk disimpan di *database server*. Bersamaan dengan itu sistem akan menghitung jarak terdekat antar bus. Jika jarak antar bus adalah 5 kilometer maka sistem akan memberikan notifikasi peringatan berupa bunyi sirine selama beberapa detik.

Berikut adalah rumus dasar dari Formula Haversine yang dimasukkan ke dalam sistem untuk menghitung jarak antar dua koordinat dalam GPS:

```
double EARTH_RADIUS = 6371;
double latitude;
double longitude;

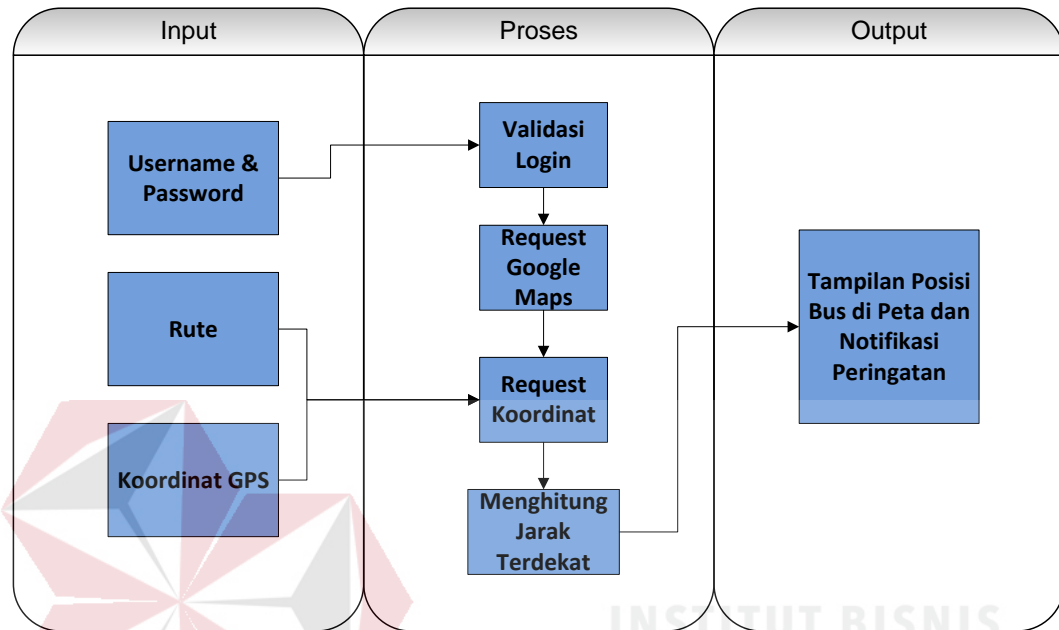
double lat2;
double lon2;

double dlat = lat2 - latitude;
double dlon = lon2 - longitude;

double a = Math.sin(dlat / 2) * Math.sin(dlat / 2) +
Math.cos(latitude) * Math.cos(lat2) * Math.sin(dlon / 2) *
Math.sin(dlon / 2);
double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1-a));
double distance = EARTH_RADIUS * c;
```


3.2.1 IPO Diagram Aplikasi Monitoring Armada Bus

Kebutuhan *input*, *proses*, dan *output* dari aplikasi yang akan dibuat digambarkan pada diagram IPO di bawah ini:



Gambar 3.3 IPO Diagram Aplikasi Monitoring Armada Bus

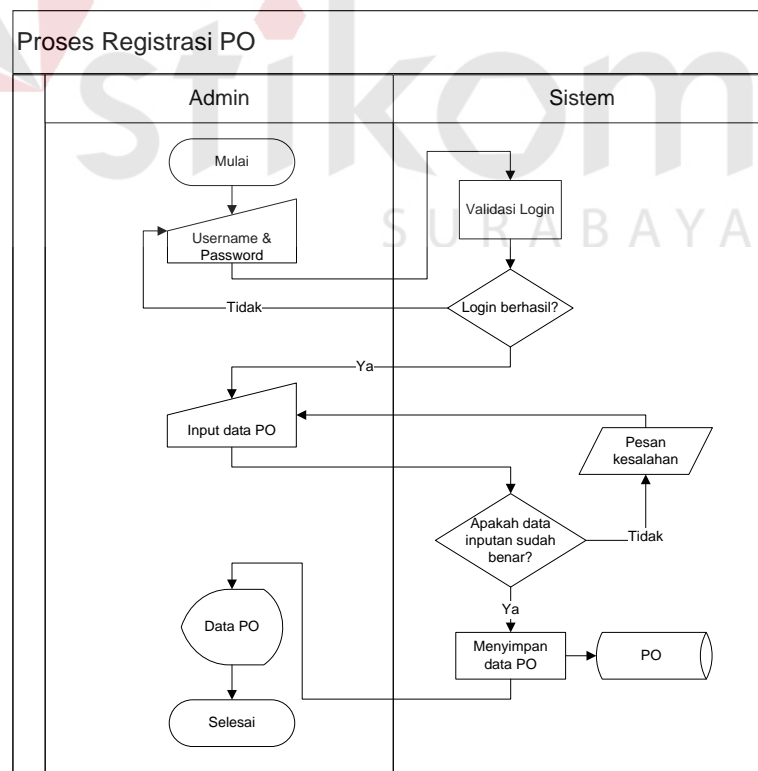
Pada aplikasi monitoring armada bus ini memiliki 3 inputan, satu inputan dari *user* yaitu *username* dan *password*, dua data yang di *load* dari *database* yaitu data rute pilihan dan data koordinat GPS. Proses di dalam sistem sendiri memiliki empat tahapan. Tahap pertama yaitu memvalidasi *login* dari *username* dan *password* yang dimasukkan oleh *user*, tahap kedua yaitu *request* google maps untuk menampilkan peta. Tahap terakhir yaitu *request* data *tracking* dari *database* dan secara bersamaan dilakukan penghitungan jarak antar dua titik koordinat oleh sistem. Output dari proses ini yaitu sistem akan menampilkan posisi bus berupa *marker* pada peta digital dan notifikasi peringatan berupa bunyi sirine selama beberapa detik jika jarak antar bus adalah 5 kilometer.

3.2.2 Flowchart

Aplikasi Monitoring Armada Bus Menggunakan GPS *Tracking* pada *Smartphone* Android dibagi menjadi lima proses utama yaitu:

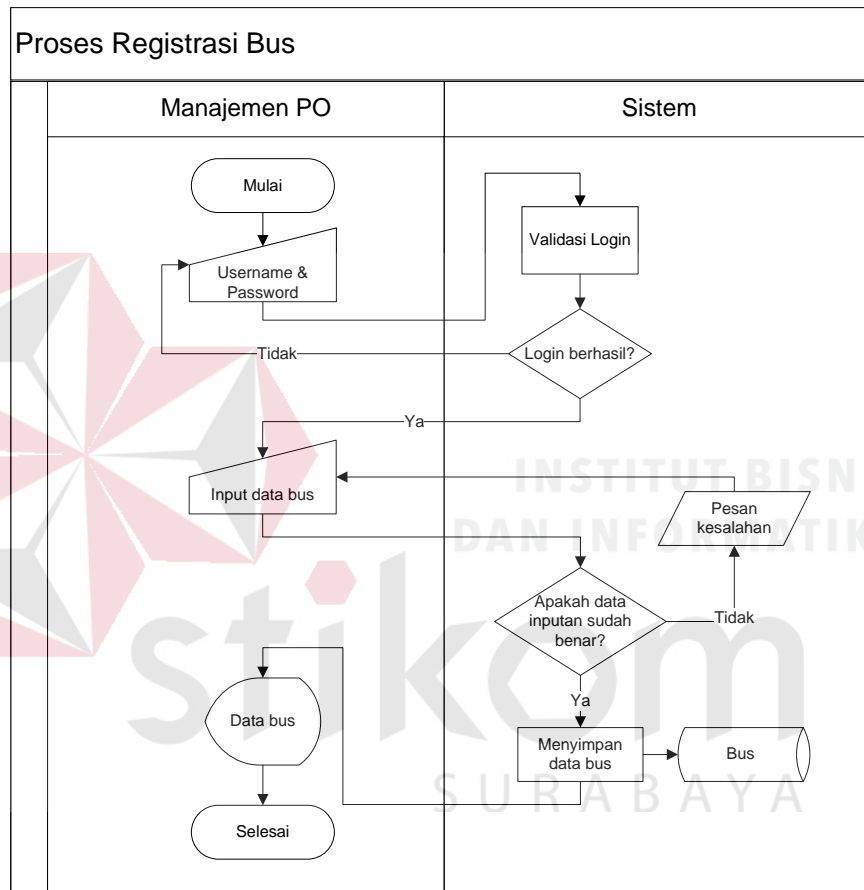
1. Proses registrasi PO
2. Proses registrasi bus
3. Proses registrasi sopir
4. Proses *tracking*
5. Proses GPS *Logger*

Proses registrasi PO digunakan untuk mencatat jumlah dan data Perusahaan Otobus yang melayani trayek Surabaya-Semarang. Proses registrasi ini dilakukan oleh admin yaitu Dinas Perhubungan propinsi Jawa Timur. Proses registrasi PO dapat dilihat pada Gambar 3.4 dibawah ini:



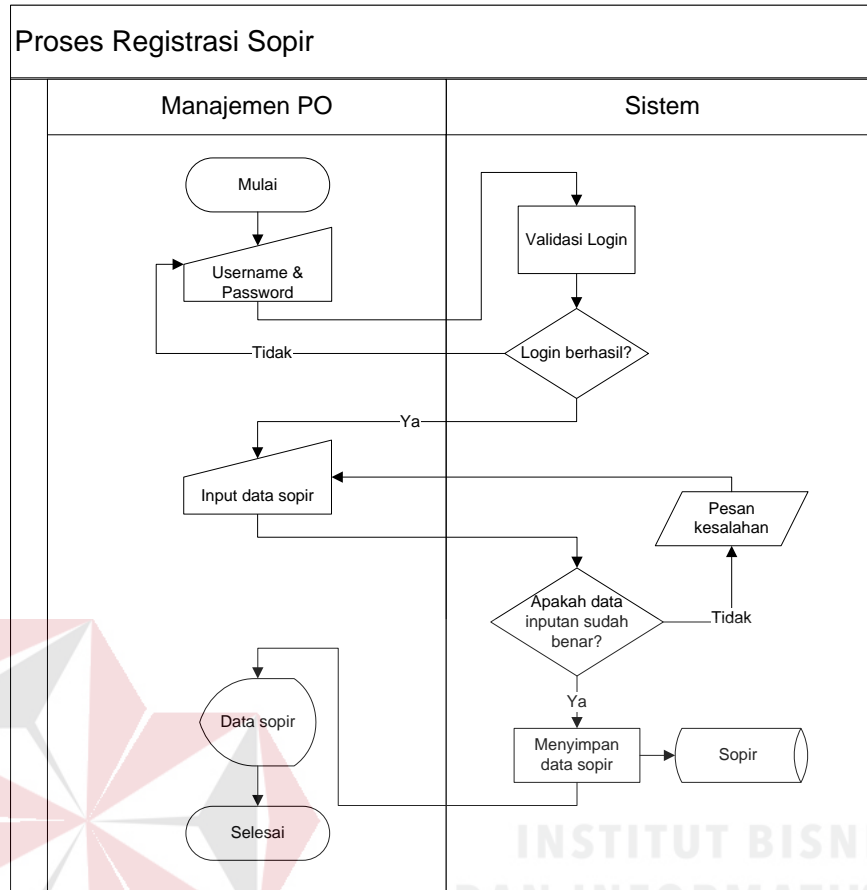
Gambar 3.4 *Flowchart* Proses Registrasi PO

Pada proses registrasi bus, hanya *user* dengan hak akses sebagai pengguna yang bisa mengakses proses pada sistem ini yaitu masing-masing manajemen PO. Setelah melakukan *login* maka pengguna dapat melakukan registrasi data bus masing-masing perusahaan. Proses registrasi data bus dapat dilihat pada Gambar 3.5 di bawah ini:



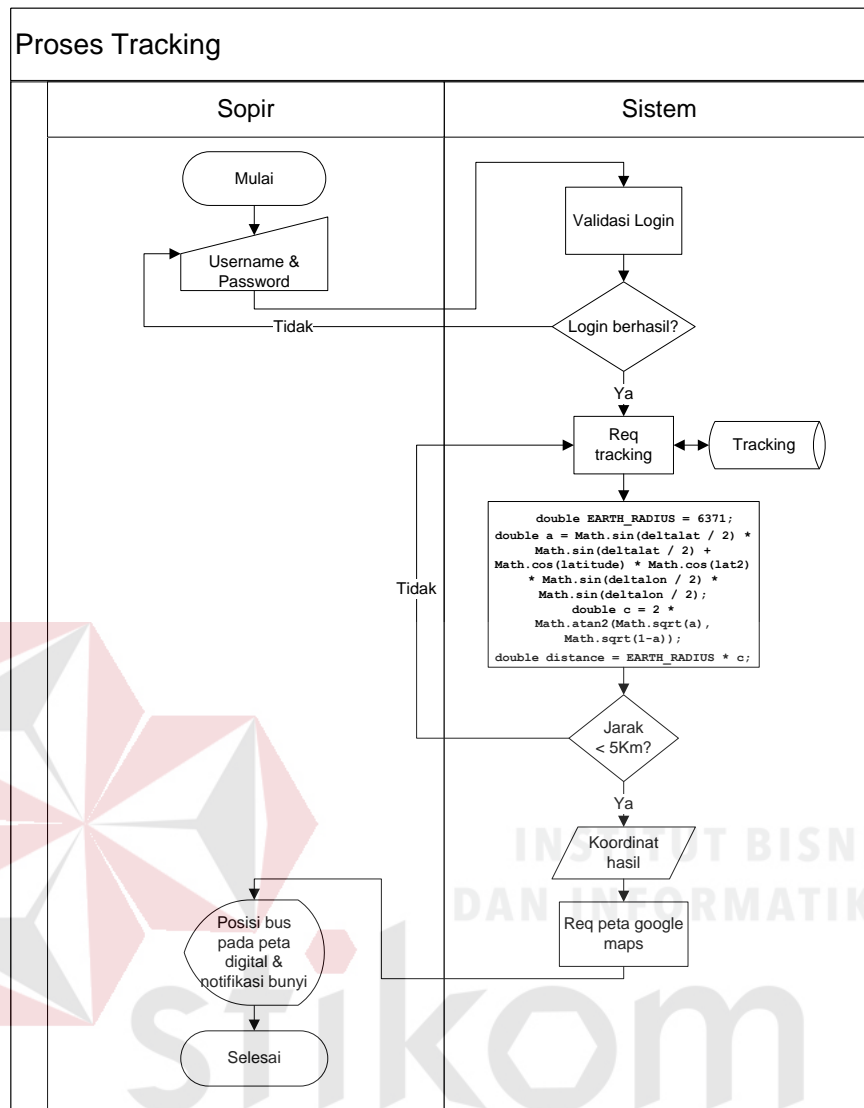
Gambar 3.5 *Flowchart* Proses Registrasi Bus

Pada proses registrasi sopir, hanya *user* dengan hak akses sebagai pengguna yang bisa mengakses proses pada sistem ini yaitu masing-masing manajemen PO. Setelah melakukan *login* maka pengguna dapat melakukan registrasi data sopir masing-masing perusahaan. Proses registrasi data sopir dapat dilihat pada Gambar 3.6 di bawah ini:



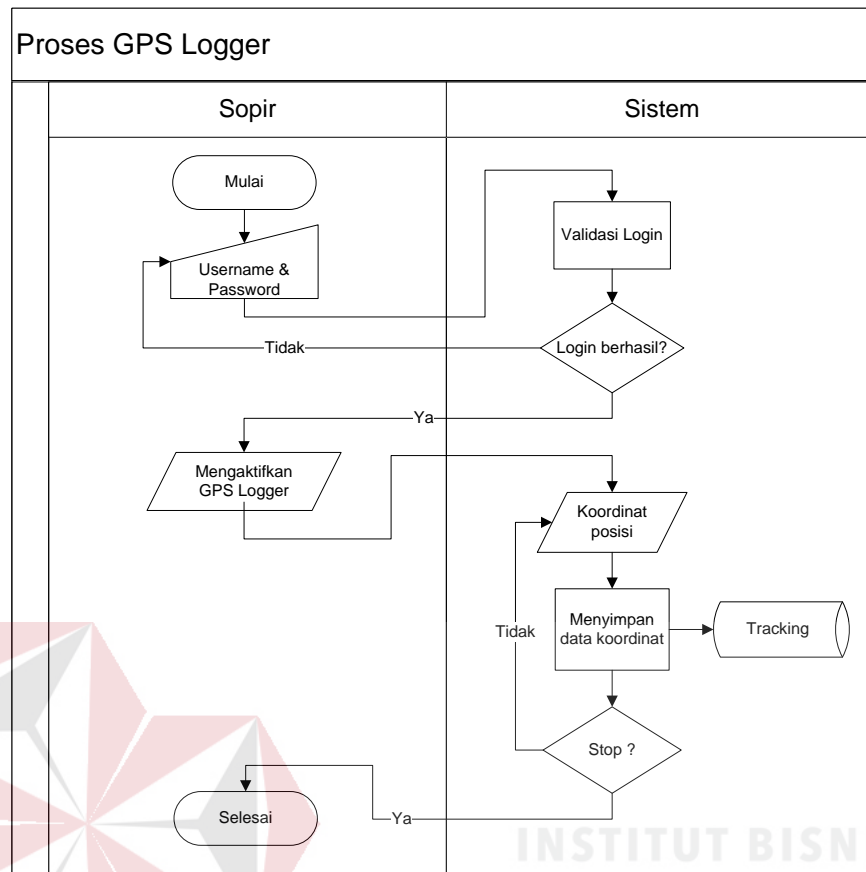
Gambar 3.6 *Flowchart* Proses Registrasi Sopir

Proses *tracking* bus merupakan proses yang berfungsi untuk memantau keberadaan posisi bus baik pada aplikasi web atau mobile. Pada proses *tracking*, posisi koordinat yang ditampilkan pada peta adalah posisi terakhir dari kendaraan yang berhasil disimpan dalam *database*. Kemudian sistem juga menghitung apakah jarak bus satu dengan yang lain kurang dari 5 kilometer. Jika ya maka sistem pada perangkat mobile akan mengeluarkan notifikasi berupa bunyi. Proses *tracking* dapat dilihat pada Gambar 3.7 di bawah ini:



Gambar 3.7 Flowchart Proses Tracking

Proses GPS *Logger* merupakan proses yang berfungsi untuk melakukan penyimpanan koordinat posisi perangkat mobile pada kendaraan ke dalam *database*. Proses ini merupakan *background service*, dimana proses tetap berjalan walaupun program utama tidak dalam kondisi aktif. GPS *Logger* akan berulang terus menerus secara otomatis sesuai dengan pengaturan interval. Proses GPS *Logger* dapat dilihat pada Gambar 3.8 di bawah ini:



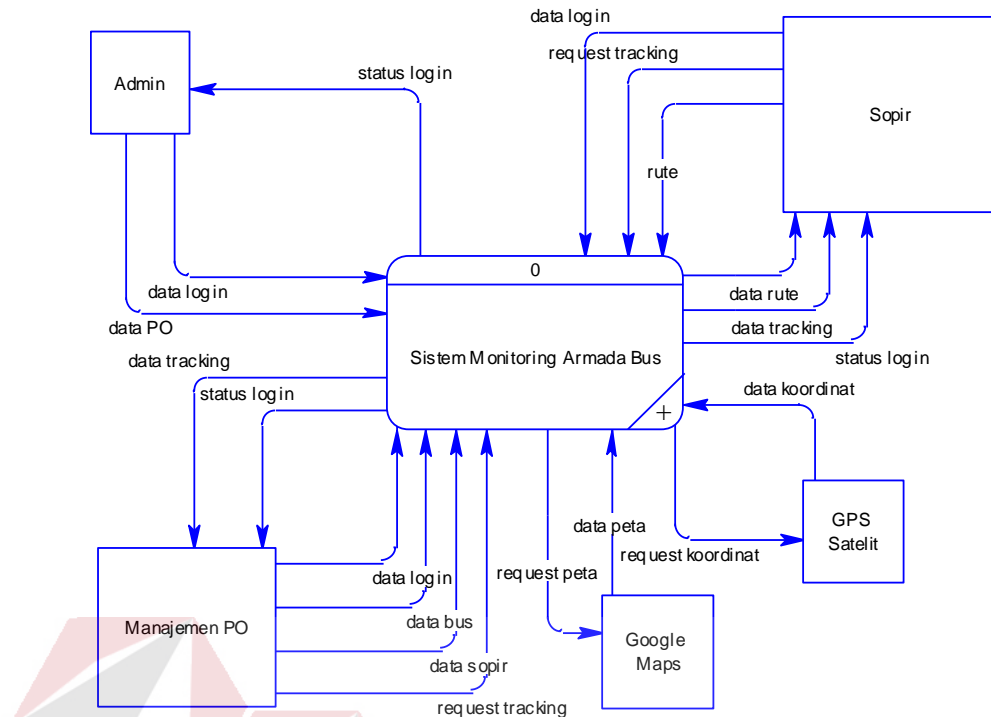
Gambar 3.8 Flowchart Proses GPS Logger

3.2.3 Data Flow Diagram (DFD)

Data Flow Diagram berfungsi untuk menggambarkan proses aliran data yang terjadi pada sistem dari tingkat tertinggi sampai tingkat terendah:

A. Context Diagram

Context Diagram ditunjukkan pada gambar 3.9 dibawah ini, dimana terdapat lima *entity* yang berhubungan dengan sistem dan mempunyai input sekaligus output masing-masing, yaitu: admin, manajemen PO, sopir, GPS Satelit, dan Google Maps.



Gambar 3.9 Context Diagram Sistem Monitoring Armada Bus

B. DFD Level 0

DFD level 0 dapat dilihat pada gambar 3.10 di bawah ini, pada DFD level 0 ini terdapat empat proses yaitu:

1. Proses *Mastering*

Proses *mastering* merupakan proses yang digunakan untuk menyimpan data PO, bus dan sopir.

2. Proses *Login*

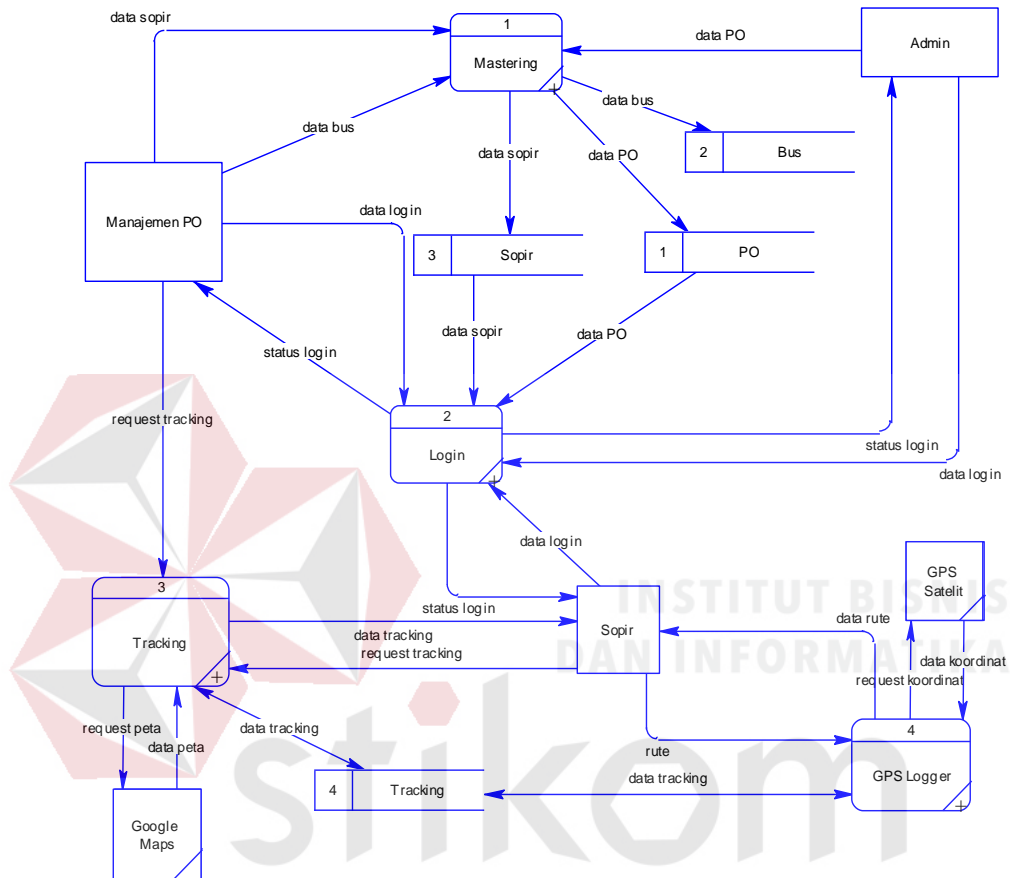
Proses *login* adalah sebagai validasi dari pengguna yang dapat melakukan proses pemantauan yaitu: Admin, Manajemen PO, dan Sopir.

3. Proses *Tracking*

Proses *tracking* armada bus merupakan proses yang berfungsi untuk memantau keberadaan posisi armada bus.

4. Proses *Logger*

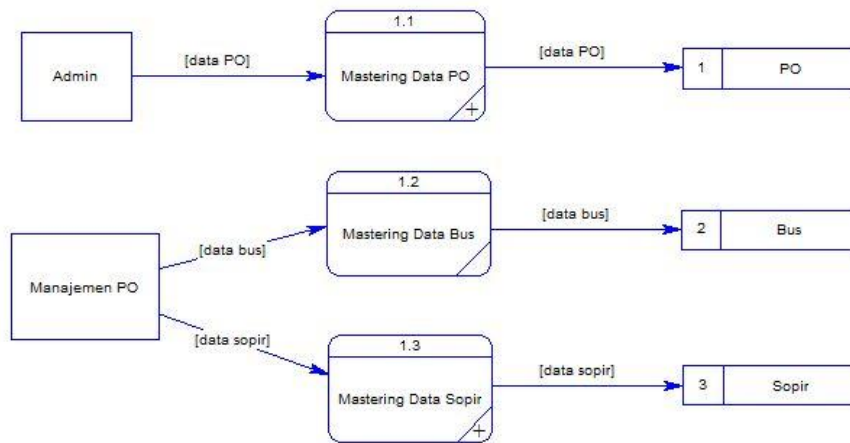
Proses *logger* merupakan proses yang berfungsi untuk melakukan penyimpanan koordinat kendaraan ke dalam *database*.



Gambar 3.10 DFD level 0 Sistem Monitoring Armada Bus

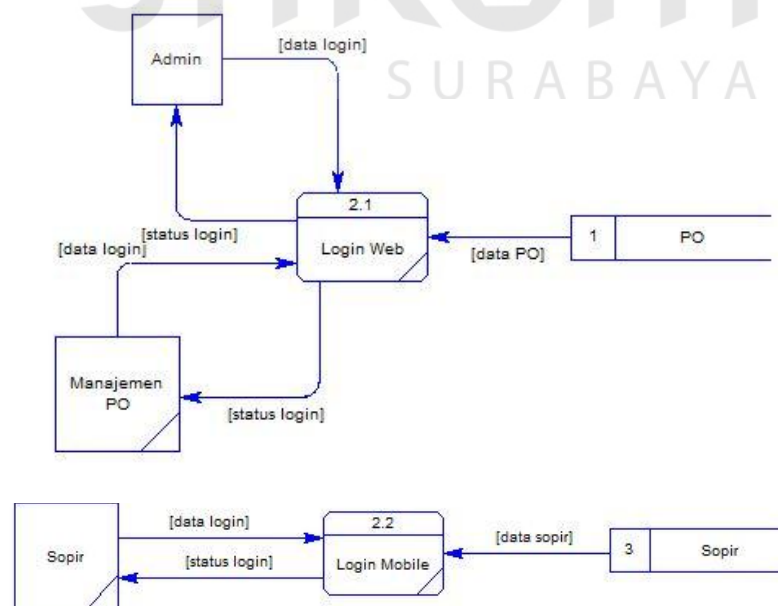
C. DFD Level 1 Proses *Mastering*

Gambar 3.11 di bawah ini merupakan penjabaran lebih detail lagi tentang DFD level 1 Proses *Mastering*. DFD level 1 Proses *Mastering* ini dijabarkan menjadi tiga proses yaitu: *mastering* data PO, *mastering* data bus, dan *mastering* data sopir.

Gambar 3.11 DFD Level 1 Proses *Mastering*

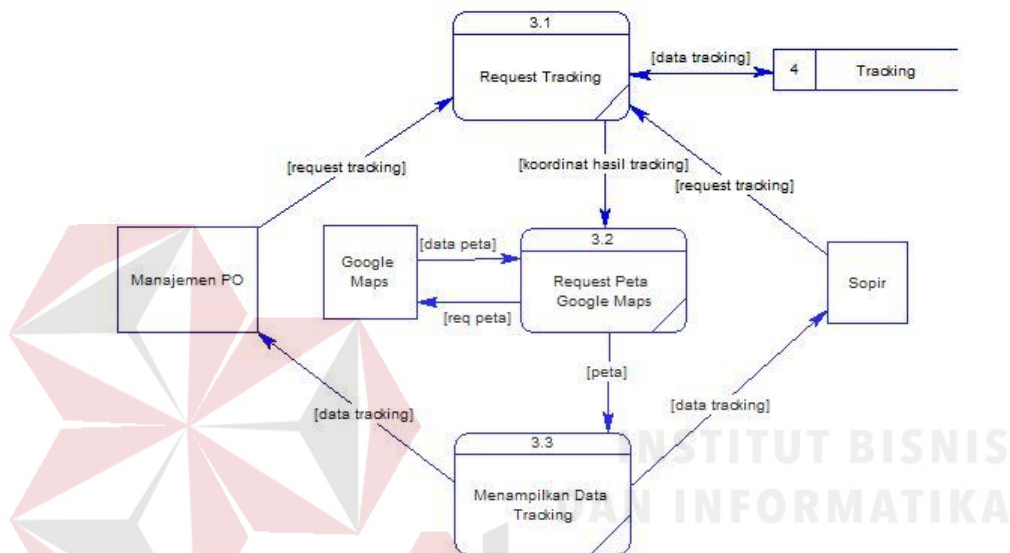
D. DFD Level 1 Proses *Login*

Gambar 3.12 di bawah ini merupakan penjabaran lebih detail lagi tentang DFD level 1 Proses *Login*. DFD level 1 Proses *Login* ini dijabarkan menjadi dua proses yaitu: Proses *Login Web* untuk Admin dan Manajemen PO dan Proses *Login Mobile* untuk Sopir.

Gambar 3.12 DFD Level 1 Proses *Login*

E. DFD Level 1 Proses *Tracking*

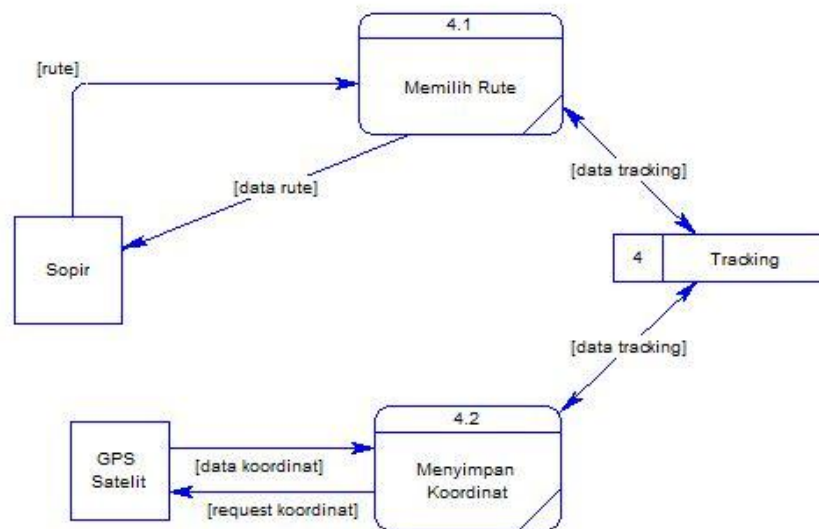
Gambar 3.13 di bawah ini merupakan penjabaran lebih detail lagi tentang DFD level 1 Proses *Tracking*. DFD level 1 Proses *Tracking* ini dijabarkan menjadi tiga proses yaitu: *request tracking*, *request peta google maps*, dan menampilkan data *tracking*.



Gambar 3.13 DFD Level 1 Proses *Tracking*

F. DFD Level 1 Proses *Logger*

Gambar 3.14 di bawah ini merupakan penjabaran lebih detail lagi tentang DFD level 1 Proses *Logger*. DFD level 1 Proses *Tracking* ini dijabarkan menjadi dua proses yaitu: memilih rute dan penyimpanan posisi secara berkala.



Gambar 3.14 DFD Level 1 Proses *Logger*

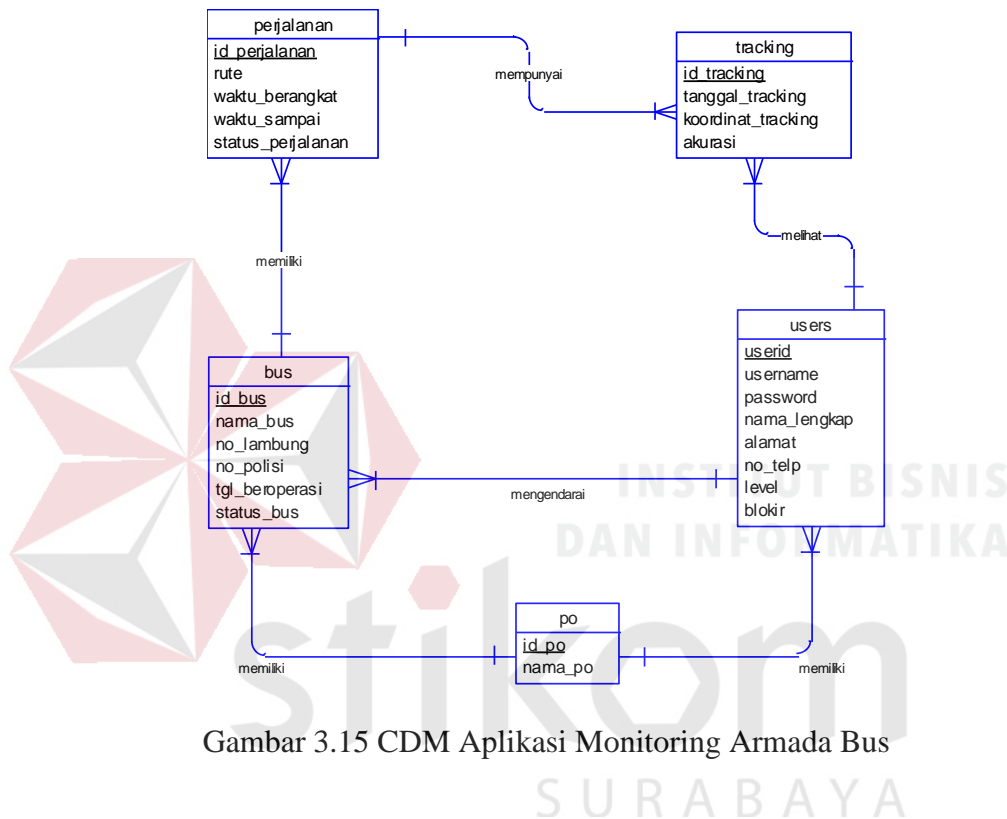
3.2.4 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah suatu desain sistem yang digunakan untuk mempresentasikan, menentukan dan mendokumentasikan kebutuhan-kebutuhan untuk sistem pemrosesan *database*. ERD juga menyediakan bentuk untuk menunjukkan struktur keseluruhan dari pemakai. Dalam perancangan sistem ini telah terbentuk ERD yang merupakan lanjutan dari pembuatan desain dengan menggunakan DFD. Pembuatan ERD bertujuan untuk mengetahui kebutuhan sistem berkaitan dengan basis data yang dibutuhkan oleh sistem.

A. Conceptual Data Model (CDM)

Sebuah CDM menggambarkan secara keseluruhan konsep struktur basis data yang dirancang untuk suatu program atau aplikasi. Pada CDM belum tergambar jelas untuk tabel-tabel penyusun basis data beserta field-field yang

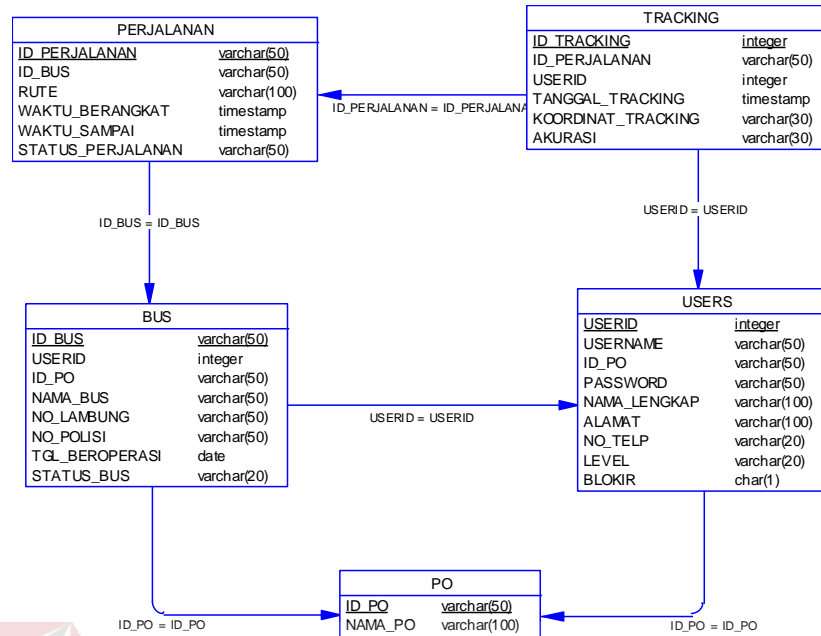
terdapat pada setiap tabel. Tabel-tabel tersebut sudah mengalami *relationship* tetapi tidak terlihat pada kolom yang mana hubungan antar tabel tersebut. Pada CDM telah didefinisikan kolom mana yang menjadi *primary key*. CDM Aplikasi Monitoring Armada Bus Menggunakan GPS *Tracking* pada *Smartphone* Android dapat dilihat pada Gambar 3.15 di bawah ini:



Gambar 3.15 CDM Aplikasi Monitoring Armada Bus

B. Physical Data Model (PDM)

PDM Aplikasi Monitoring Armada Bus Menggunakan GPS *Tracking* pada *Smartphone* Android dapat dilihat pada Gambar 3.16 di bawah ini:



Gambar 3.16 PDM Aplikasi Monitoring Armada Bus

3.2.5 Struktur Tabel

Struktur tabel merupakan penjabaran dan penjelasan dari suatu *database*. Dalam struktur tabel dijelaskan fungsi dari masing-masing tabel hingga fungsi masing-masing *field* yang ada di dalam tabel. Selain itu juga terdapat tipe data dari masing-masing *field* beserta konstrainnya.

A. Tabel Users

Primary Key : Userid

Foreign Key : Id_PO

Fungsi : Untuk menyimpan data user

Tabel 3.3 Users

No	Field	Tipe Data	Keterangan
1	<u>Userid</u>	Integer	Pengisian id user secara autonumber
2	Username	Varchar(50)	Username bersifat unik
3	Password	Varchar(50)	Dienkripsi dalam bentuk md5

4	Id_PO	Varchar(50)	Id PO user
5	Nama_lengkap	Varchar(100)	Nama user
6	Alamat	Varchar(100)	Alamat user
7	No_telp	Varchar(20)	No telepon user
8	Level	Varchar(20)	Level hak akses user
9	Blokir	Char(1)	Status pemblokiran user

B. Tabel PO

Primary Key : Id_PO

Foreign Key : Nama_PO

Fungsi : Untuk menyimpan data PO

Tabel 3.4 PO

No	Field	Tipe Data	Keterangan
1	<u>ID_PO</u>	Varchar(50)	Id PO
2	Nama_PO	Varchar(100)	Nama Perusahaan Otobus

C. Tabel Bus

Primary Key : Id_Bus

Foreign Key : Id_PO

Fungsi : Untuk menyimpan data bus

Tabel 3.5 Bus

No	Field	Tipe Data	Keterangan
1	<u>Id_Bus</u>	Varchar(50)	Id Bus
2	Userid	Integer	Pengisian id user secara autonumber
3	Id_PO	Varchar(50)	Id PO
4	Nama_bus	Varchar(50)	Nama Bus
5	No_lambung	Varchar(50)	Nomor lambung bus
6	No_polisi	Varchar(50)	Nomor Polisi bus
7	Tgl_beroperasi	date	Tanggal mulai beroperasi
8	Status_bus	Varchar(20)	Status aktif bus

D. Tabel Perjalanan

Primary Key : Id_Perjalanan

Foreign Key : Rute

Fungsi : Untuk menyimpan data perjalanan

Tabel 3.6 Perjalanan

No	Field	Tipe Data	Keterangan
1	<u>Id Perjalanan</u>	Varchar(50)	Pengisian id user secara autonumber
2	Id_bus	Varchar(50)	Id bus
3	Rute	Varchar(100)	Rute perjalanan
4	Waktu_berangkat	Timestamp	Tanggal catatan berangkat
5	Waktu_sampai	Timestamp	Tanggal catatan sampai
6	Status_perjalanan	Varchar(50)	Status perjalanan

E. Tabel Tracking

Primary Key : Id_Tracking

Foreign Key : Id_Perjalanan

Fungsi : Untuk menyimpan data *tracking*

Tabel 3.7 Tracking

No	Field	Tipe Data	Keterangan
1	<u>Id Tracking</u>	Integer	Pengisian id user secara autonumber
2	Id_perjalanan	Varchar(50)	Id perjalanan
3	Userid	Integer	Pengisian id user secara autonumber
4	Tanggal_tracking	Timestamp	Tanggal catatan tracking
5	Koordinat_tracking	Varchar(30)	Koordinat posisi terakhir
6	Akurasi	Varchar(30)	Akurasi posisi

3.3 Perancangan Desain *Input / Output*

Desain *Input/Output* memvisualisasikan tampilan dari aplikasi. Desain *Input/Output* Aplikasi Monitoring Armada Bus Menggunakan *GPS Tracking* pada *Smartphone* Android dibagi menjadi beberapa bagian sebagai berikut:

3.3.1 Rancangan Form *Login (Mobile)*

Rancangan form *login* merupakan rancangan tampilan aplikasi saat user melakukan *login* untuk masuk ke dalam menu aplikasi *mobile*. Form *login* terdiri dari *username* dan *password* yang harus di isi terlebih dahulu, selanjutnya user menekan *button login* untuk masuk ke dalam menu aplikasi. Rancangan form *login* dapat dilihat pada gambar 3.17 di bawah ini:



Gambar 3.17 Rancangan Form *Login (Mobile)*

3.3.2 Rancangan Menu Utama

Rancangan menu utama merupakan rancangan tampilan awal aplikasi *mobile* saat user berhasil melakukan *login*. Halaman menu utama terdiri dari 6 *button* menu yaitu: mulai perjalanan, selesai perjalanan, *start logger*, *stop logger*, peta, dan *setting*. Rancangan menu utama dapat dilihat pada gambar 3.18 di bawah ini:

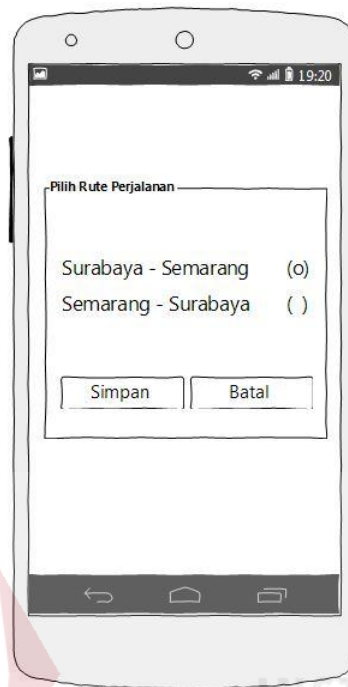


Gambar 3.18 Rancangan Menu Utama

3.3.3 Rancangan Menu Mulai Perjalanan

Rancangan menu mulai perjalanan merupakan rancangan tampilan aplikasi saat user menekan *button* mulai perjalanan pada menu utama untuk memilih rute perjalanan. Disediakan dua pilihan rute dalam aplikasi ini yaitu Surabaya-Semarang atau Semarang-Surabaya. Setelah memilih user harus menekan *button*

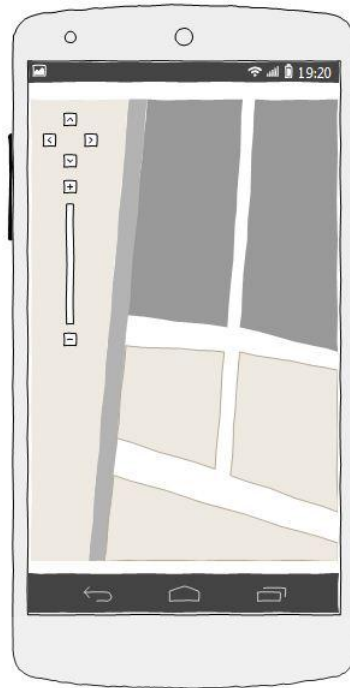
simpan. Rancangan menu mulai perjalanan dapat dilihat pada gambar 3.19 di bawah ini:



Gambar 3.19 Rancangan Menu Mulai Perjalanan

3.3.4 Rancangan Menu Peta

Rancangan menu peta merupakan rancangan untuk menampilkan peta digital pada aplikasi *mobile*. Pada peta ini *user* dapat melihat posisi diri sendiri dan posisi armada bus lain yang sedang beroperasi dalam rute searah. Rancangan menu peta dapat dilihat pada gambar 3.20 di bawah ini:



Gambar 3.20 Rancangan Menu Peta

3.3.5 Rancangan Form Login (Web)

Rancangan form *login* merupakan rancangan tampilan aplikasi saat user melakukan *login* untuk masuk ke dalam menu aplikasi web. Form *login* terdiri dari *username* dan *password* yang harus di isi terlebih dahulu, selanjutnya user menekan *button login* untuk masuk ke dalam menu aplikasi. Rancangan form *login* dapat dilihat pada gambar 3.21 di bawah ini:

Gambar 3.21 Rancangan Form *Login* (Web)

3.3.6 Rancangan Form Master PO

Rancangan form master PO merupakan rancangan tampilan saat admin melakukan registrasi data Perusahaan Otobus agar dapat melakukan *login* pada saat mengakses aplikasi *web*. Form ini hanya dapat diakses oleh admin. Form ini diisi oleh admin berdasarkan data dari manajemen masing-masing perusahaan. Rancangan form master PO dapat dilihat pada Gambar 3.22 di bawah ini:

No	ID PO	Nama PO	Username	Alamat	No Telp	Blokir	Aksi
1	P001	INDONESIA	petugas1	Jl. Asd	123456	N	edit delete
2	P002	SINAR	petugas2	Jl. Asd	123456	N	edit delete
3	P003	WIDJI	petugas3	Jl. Asd	123456	N	edit delete

Gambar 3.22 Rancangan Form Master PO

3.3.7 Rancangan Form Master Bus

Rancangan form master bus merupakan rancangan tampilan saat manajemen PO melakukan registrasi data bus. Form ini hanya dapat diakses oleh manajemen PO. Form ini diisi oleh manajemen PO berdasarkan data dari masing-masing bus. Rancangan form master bus dapat dilihat pada Gambar 3.23 di bawah ini:

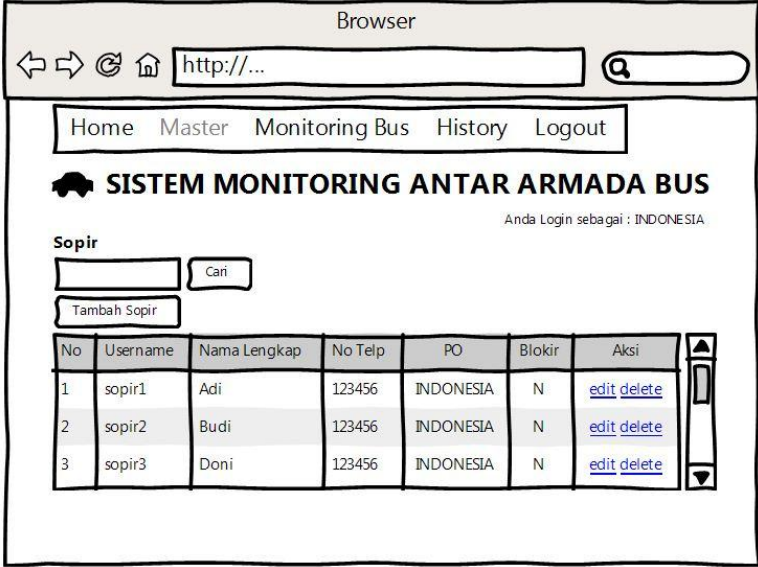
The screenshot shows a web browser window with the URL 'http://...'. The page title is 'SISTEM MONITORING ANTAR ARMADA BUS' and the user is logged in as 'INDONESIA'. The interface includes a navigation menu with 'Home', 'Master', 'Monitoring Bus', 'History', and 'Logout'. Below the menu, there is a search bar with a 'Cari' button and a 'Tambah Bus Baru' button. A table displays the following data:

No	ID Bus	Nama Bus	PO	Sopir	No Polisi	Tgl beroperasi	Aksi
1	B001	Indo1	INDONESIA	Adi	L7072UV	27/03/2014	edit delete
2	B002	Indo2	INDONESIA	Budi	L7073UV	27/03/2014	edit delete
3	B003	Indo3	INDONESIA	Doni	L7074UV	27/03/2014	edit delete

Gambar 3.23 Rancangan Form Master Bus

3.3.8 Rancangan Form Master Sopir

Rancangan form master sopir merupakan rancangan tampilan saat manajemen PO melakukan registrasi data sopir. Form ini hanya dapat diakses oleh manajemen PO. Form ini diisi oleh manajemen PO berdasarkan data dari masing-masing sopir. Rancangan form master sopir dapat dilihat pada Gambar 3.24 di bawah ini:



Browser

Home Master Monitoring Bus History Logout

SISTEM MONITORING ANTAR ARMADA BUS

Anda Login sebagai : INDONESIA

Sopir

Cari

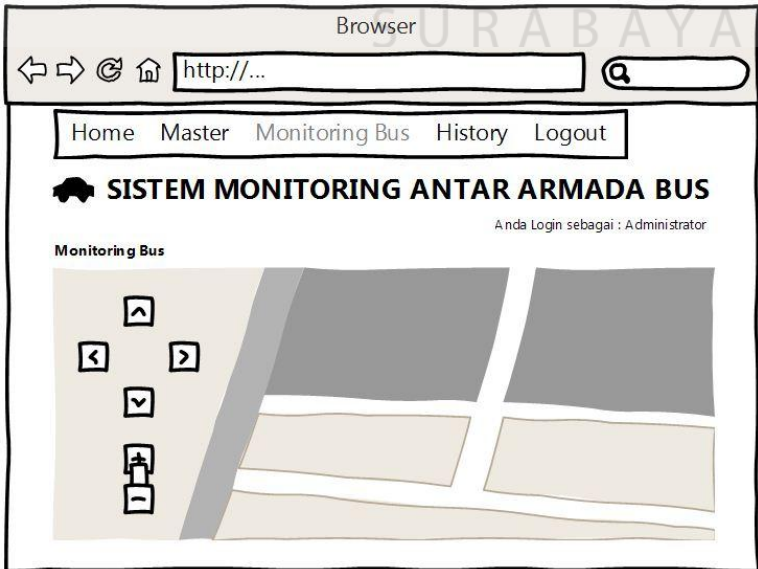
Tambah Sopir

No	Username	Nama Lengkap	No Telp	PO	Blokir	Aksi
1	sopir1	Adi	123456	INDONESIA	N	edit delete
2	sopir2	Budi	123456	INDONESIA	N	edit delete
3	sopir3	Doni	123456	INDONESIA	N	edit delete

Gambar 3.24 Rancangan Form Master Sopir

3.3.9 Rancangan Menu Monitoring

Rancangan menu monitoring merupakan rancangan untuk menampilkan peta digital pada aplikasi web. Pada peta ini *user* dapat melihat posisi armada bus yang sedang beroperasi berupa marker. Rancangan menu monitoring dapat dilihat pada gambar 3.25 di bawah ini:



Browser

Home Master Monitoring Bus History Logout

SISTEM MONITORING ANTAR ARMADA BUS

Anda Login sebagai : Administrator

Monitoring Bus

Map navigation controls: up, down, left, right arrows, zoom in, zoom out, bus icon.

Gambar 3.25 Rancangan Form Menu Monitoring

3.3.10 Rancangan Menu *History*

Rancangan menu *history* merupakan rancangan untuk menampilkan riwayat perjalanan bus. Pada menu ini admin maupun manajemen PO bisa melihat *history* masing-masing bus yang beroperasi. Rancangan menu *history* dapat dilihat pada gambar 3.26 di bawah ini:

The screenshot shows a web browser window with the following elements:

- Browser address bar: `http://...`
- Navigation menu: Home, Master, Monitoring Bus, History, Logout
- Page Title: **SISTEM MONITORING ANTAR ARMADA BUS**
- User Status: Anda Login sebagai : Administrator
- Section: **History**
- Form: PO
- Table of History Records:

No	Nama Bus	No Polisi	Sopir	PO	Tanggal	Rute	Aksi
1	Indo1	L7072UV	Adi	INDONESIA	27/03/2014	Sby-Smg	detail history
2	Indo2	L7073UV	Budi	INDONESIA	27/03/2014	Sby-Smg	detail history
3	Indo3	L7074UV	Doni	INDONESIA	27/03/2014	Sby-Smg	detail history

Gambar 3.26 Rancangan Form Menu *History*

3.4 Desain Uji Coba

Desain uji coba mempunyai tujuan untuk memastikan bahwa aplikasi telah dibuat dengan benar sesuai dengan kebutuhan atau tujuan yang diharapkan. Pada pengujian aplikasi ini akan dilakukan satu tahapan pengujian. Proses pengujian menggunakan *Black Box Testing* yaitu aplikasi akan diuji dengan melakukan berbagai percobaan untuk membuktikan bahwa aplikasi yang telah dibuat sudah sesuai dengan tujuan. Uji coba yang akan dilakukan adalah Uji coba fungsi aplikasi:

3.4.1 Uji Coba Fungsi Aplikasi

Proses uji coba fungsi aplikasi ini dilakukan untuk mengetahui apakah fungsi-fungsi dari Aplikasi Monitoring Armada Bus Menggunakan GPS *Tracking* pada *Smartphone* Android ini telah berjalan dengan benar. Setiap fitur yang disediakan akan diuji hasilnya sesuai dengan tabel *test case*. Desain uji coba fungsi aplikasi adalah sebagai berikut:

A. Desain Uji Coba *Login* pada *Mobile*

Uji coba ini nantinya dilakukan untuk verifikasi *user* sebagai pengguna aplikasi *mobile* yang sudah terdaftar di dalam *database* sehingga bisa menggunakan aplikasi. Desain pengujian *login* pada *mobile* dapat dilihat pada tabel 3.8 di bawah ini:

Tabel 3.8 Desain Uji Coba *Login* pada *Mobile*

Test Case ID	Tujuan	Input	Output yang diharapkan
1	Menampilkan splash screen	Membuka aplikasi pada perangkat mobile	Menampilkan splash screen selama beberapa detik
2	Menampilkan halaman login	Membuka aplikasi pada perangkat mobile	Menampilkan halaman login setelah splash screen
3	Menampilkan halaman utama	Memasukkan username dengan benar dan memasukkan password dengan benar	Login berhasil, menampilkan halaman utama dan status login
4	Menampilkan pesan peringatan	Memasukkan username atau password yang salah	Menampilkan pesan peringatan 'login gagal'

B. Desain Uji Coba *Login* pada *Web*

Uji coba ini nantinya dilakukan untuk verifikasi *user* sebagai pengguna aplikasi *web* yang sudah terdaftar di dalam *database* sehingga bisa menggunakan aplikasi. Desain pengujian *login* pada *web* dapat dilihat pada tabel 3.9 di bawah ini:

Tabel 3.9 Desain Uji Coba *Login* pada *Web*

Test Case ID	Tujuan	Input	Output yang diharapkan
5	Menampilkan halaman login	Membuka aplikasi pada web browser	Menampilkan halaman login
6	Menampilkan halaman home	Memasukkan username dengan benar dan memasukkan password dengan benar	Login berhasil, menampilkan halaman home dan status login
7	Menampilkan pesan peringatan	Memasukkan username atau password yang salah	Menampilkan peringatan login gagal dan link 'ulangi lagi'

C. Desain Uji Coba *GPS Logger*

GPS Logger Service berfungsi untuk mendapatkan posisi koordinat dari perangkat *mobile* pada kendaraan dan kemudian disimpan ke dalam *database server*. Fungsi ini berjalan pada *background* proses sehingga walaupun aplikasi utama tidak dibuka fitur ini tetap dapat berjalan. Desain pengujian dapat dilihat pada tabel 3.10 di bawah ini:

Tabel 3.10 Desain Uji Coba *GPS Logger*

Test Case ID	Tujuan	Input	Output yang diharapkan
8	Menampilkan sub menu pilih rute	Menekan menu mulai perjalanan	Menampilkan sub menu pilih rute

	perjalanan		perjalanan
9	Menampilkan rute perjalanan yang dipilih	Memilih salah satu rute perjalanan	Menampilkan rute perjalanan yang dipilih pada halaman utama
10	Menampilkan pesan update koordinat	Menekan menu start logger	Menampilkan pesan update koordinat sesuai pengaturan interval
11	Menampilkan pesan update koordinat telah dimatikan	Menekan menu stop logger	Menampilkan pesan update koordinat telah dimatikan
12	Menampilkan pesan status perjalanan telah dimatikan	Menekan menu selesai perjalanan	Menampilkan pesan status perjalanan telah dimatikan

D. Desain Uji Coba *Tracking* pada *Mobile*

Desain uji coba ini nantinya dilakukan untuk menampilkan posisi perangkat *mobile* yang terdapat pada kendaraan berupa *marker* ke dalam peta *digital* dari Google Maps yang bisa ditampilkan pada aplikasi *mobile*. Desain pengujian dapat dilihat pada tabel 3.11 di bawah ini:

Tabel 3.11 Desain Uji Coba *Tracking* pada *Mobile*

Test Case ID	Tujuan	Input	Output yang diharapkan
13	Menampilkan posisi pada peta digital	Menekan menu peta	Menampilkan posisi berupa marker pada peta digital
14	Menampilkan posisi bus lain berupa marker pada peta digital	Menekan menu peta	Menampilkan posisi bus lain yang sedang berjalan berupa marker berwarna biru (Sby-Smg) pada peta digital
15	Menampilkan posisi bus lain	Menekan menu peta	Menampilkan posisi bus lain yang

	berupa marker pada peta digital		sedang berjalan berupa marker berwarna merah (Smg-Sby) pada peta digital
16	Menampilkan posisi bus lain berupa marker pada peta digital	Menekan menu peta	Menampilkan posisi bus lain yang sedang berhenti berupa marker berwarna putih pada peta digital
17	Menampilkan posisi bus saat notifikasi bunyi berupa marker pada peta digital	Menekan menu peta	Menampilkan posisi bus saat notifikasi bunyi berupa marker pada peta digital (memberi garis lurus warna hijau untuk menandakan jarak antar bus lebih dari 5 kilometer dan garis merah menandakan jarak antar bus kurang dari 5 kilometer)

E. Desain Uji Coba *Tracking* pada *Web*

Desain uji coba ini nantinya dilakukan untuk menampilkan posisi perangkat *mobile* yang terdapat pada kendaraan berupa *marker* ke dalam peta *digital* dari Google Maps yang bisa ditampilkan pada aplikasi *web*. Desain pengujian dapat dilihat pada tabel 3.12 di bawah ini:

Tabel 3.12 Desain Uji Coba *Tracking* pada *Web*

Test Case ID	Tujuan	Input	Output yang diharapkan
18	Menampilkan posisi bus yang sedang beroperasi pada peta digital	Menekan menu monitoring bus	Menampilkan posisi bus yang sedang beroperasi berupa marker pada peta digital

19	Menampilkan posisi bus berjalan berupa marker pada peta digital	Menekan menu monitoring bus	Menampilkan posisi bus lain yang sedang berjalan berupa marker berwarna biru (Sby-Smg) pada peta digital
20	Menampilkan posisi bus berjalan berupa marker pada peta digital	Menekan menu monitoring bus	Menampilkan posisi bus yang sedang berjalan berupa marker berwarna merah (Smg-Sby) pada peta digital
21	Menampilkan posisi bus berhenti berupa marker pada peta digital	Menekan menu monitoring bus	Menampilkan posisi bus yang sedang berhenti berupa marker berwarna putih pada peta
22	Menampilkan daftar riwayat perjalanan bus	Menekan menu history	Menampilkan daftar riwayat perjalanan bus beserta link 'detail history'
23	Menampilkan detail riwayat perjalanan bus yang dimaksud	Menekan link 'detail history' pada kolom aksi	Menampilkan detail riwayat perjalanan bus yang dimaksud beserta marker saat update koordinat posisi pada peta digital