

**RANCANG BANGUN PROTOTIPE TRACKING MOBIL
PELAYANAN TEKNIK PLN MENGGUNAKAN PERANGKAT
BERGERAK BERBASIS GPS**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Radyan Ilham Cahyadi
NIM: 115060807111126



PROGRAM STUDI INFORMATIKA
PROGRAM TEKNOLOGI INFORMASI DAN ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2015

PENGESAHAN

RANCANG BANGUN PROTOTIPE TRACKING MOBIL PELAYANAN TEKNIK PLN
MENGUNAKAN PERANGKAT BERGERAK BERBASIS GPS

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Radyan Ilham Cahyadi
NIM: 115060807111126

Skripsi ini telah diuji dan dinyatakan lulus pada
18 Januari 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Dr.Eng. Herman Tolle, ST., M.T.
NIK: 19740823 200012 1 001

Agi Putra Kharisma, S.T., M.T.
NIK. 201304 860430 1 001

Mengetahui
Ketua Program Studi Informatika/Illmu Komputer

Drs. Marji, M.T.
NIP: 19670801 199203 1 001

PERNYATAAN ORISINALITAS

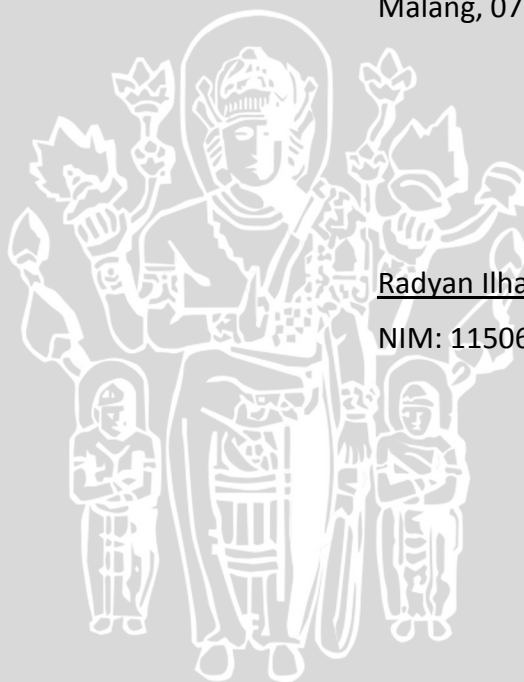
Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 07 Januari 2016

Radyan Ilham Cahyadi

NIM: 115060807111126



KATA PENGANTAR

Dengan nama Allah SWT Yang Maha Pengasih Lagi Maha Penyayang. Segala puji dan syukur penulis panjatkan atas kehadiran-Nya, karena hanya atas rahmat, pertolongan dan kasih sayang-Nya penulis dapat menyelesaikan skripsi dengan judul “Rancang Bangun Prototipe Tracking Mobil Pelayanan Teknik PLN Menggunakan Perangkat Bergerak Berbasis GPS” dengan baik.

Shalawat serta salam atas junjungan nabi besar kita Nabi Muhammad SAW beserta keluarga dan para sahabat sekalian. Melalui kesempatan ini, penulis menyampaikan rasa hormat dan terimakasih kepada pihak-pihak yang telah memberikan bantuan kepada penulis dalam penyelesaian skripsi ini. Pihak-pihak tersebut antara lain:

1. Bapak Dr.Eng. Herman Tolle, ST., M.T. dan Bapak Agi Putra Kharisma, S.T., M.T. Selaku dosen pembimbing I dan dosen pembimbing II yang telah banyak memberikan bimbingan, ilmu dan saran dalam penyusunan skripsi ini.
2. Kedua orang tua serta kedua kakak yang telah memberi motivasi, kasih sayang serta dukungan moril dan materil kepada penulis. Para saudara dari penulis yang telah menjaga kedua orang tua selama penulis berada di Malang.
3. Seluruh dosen Program Studi Informatika/Illmu Komputer atas kesediaannya membagi ilmunya kepada penulis.
4. Seluruh staf dan karyawan di FILKOM Universitas Brawijaya yang telah membantu kelancaran pengerjaan skripsi.
5. Teman-teman kuliah diantaranya seluruh anggota kelas TIF-H serta semua teman-teman angkatan 2011, terimakasih atas segala bantuannya selama menjadi mahasiswa serta dukungannya dalam menyelesaikan skripsi ini.

Semoga jasa dan amal baik kita semua mendapatkan balasan dari Allah SWT. Penulis menyadari bahwa skripsi ini masih banyak kekurangan dan masih jauh dari sempurna. Oleh karena itu, kritik dan saran untuk kesempurnaan skripsi ini, senantiasa penulis harapkan dari berbagai pihak.

Malang, 07 Januari 2016

Penulis

radyanic@gmail.com

ABSTRAK

PLN adalah perusahaan milik Negara (BUMN) yang bergerak di bidang jasa penyediaan listrik bagi masyarakat. Sebagai perusahaan listrik terbesar di Indonesia, PLN memiliki pelanggan yang tersebar di seluruh pelosok tanah air. Dengan meningkatnya jumlah pelanggan dan komitmen untuk menjaga kualitas pelayanan, banyak permasalahan yang muncul dan beberapa hal yang perlu di perbaiki. Ada beberapa petugas yang lamban atau bahkan berbohong tentang keberadaan mereka yang sebenarnya. Tidak jarang para petugas mengelak bahwasannya masih sedang melayani pelanggan di tempat lain. Manajemen PLN kesulitan dalam mencari tahu kebenaran yang terjadi.

Sejak diciptakannya *smartphone*, fungsi perangkat mobile saat ini tidak hanya dapat digunakan sebagai alat komunikasi saja, tetapi juga digunakan untuk fungsi lain untuk membantu manusia dalam melakukan tugas sehari-hari. Hampir semua *smartphone* saat ini dilengkapi dengan teknologi Global Positioning System (GPS). Salah satunya aplikasi yang menggunakan fungsi dari *GPS* adalah *map* (peta). Aplikasi *map* dapat membantu manusia dalam melihat keadaan suatu daerah atau wilayah. Aplikasi dapat menampilkan gambar secara visual berupa tata letak seperti jalan, sungai, bangunan dan lain sebagainya. Ide dari penyelesaian masalah yang terjadi adalah dengan menciptakan aplikasi yang dapat menyediakan sisi pemantau dan sisi yang dipantau untuk menjalankan fungsi *tracking* serta mengetahui kapan petugas melaksanakan perbaikan terhadap kerusakan secara langsung sehingga manager dapat menilai kinerja dari para petugas pelayanan teknik. Sistem dirancang menggunakan bahasa pemrograman java dengan memanfaatkan API Google Map. Sebagai media penyimpanan data menggunakan web hosting dan untuk transfer data menggunakan koneksi internet yang berada pada *smartphone*.

Pengujian dilakukan dengan dua cara yaitu pengujian validasi menggunakan *black box testing* dan pengujian akurasi dengan cara membandingkan tingkat akurasi melalui perhitungan selisih jarak antara aplikasi dengan Google Map. Dari Hasil pengujian fungsionalitas menunjukkan bahwa sistem telah memenuhi kriteria spesifikasi kebutuhan fungsional pada tahap perancangan. Selain itu hasil pengujian akurasi dengan cara membandingkan tingkat keakuratan antara aplikasi dengan Google Map menunjukkan selisih jarak rata-rata yaitu 5,1554 meter.

Kata kunci: GPS, Google Map, *Tracking*.

ABSTRACT

PLN is a state-owned enterprise (SOE) which is engaged in the provision of electricity to the community. As the largest power company in Indonesia, PLN has customers spread throughout the country. With the increasing number of customers and the commitment to maintain quality of service, many problems arise and some things that need to be repaired. There are officers who are slow or even lie about their actual existence. Not infrequently the officers are still being evasive bahwasannya serve customers elsewhere. PLN management difficulties in finding out the truth of what happened.

Since the creation of smartphones, mobile devices function at this time can not only be used as a communication tool, but also used for other functions to assist humans in performing everyday tasks. Almost all smartphones today are equipped with Global Positioning System (GPS). One of these applications using GPS is a function of the map (map). Applications folder can assist people in seeing the state of an area or region. Applications can display images in a visual form of the layout, such as roads, rivers, buildings and so forth. The idea of solving problems occur is to create applications that can provide the monitoring and sides are monitored to perform the function of tracking and knowing when officers carry out repairs to the damage directly so that the manager can assess the performance of the technical service personnel. The system was designed using the Java programming language by utilizing the Google Maps API. As data storage media using a web hosting and data transfer using an internet connection that is on the smartphone.

Testing is done in two ways: validation testing using black box testing and testing accuracy by comparing the degree of accuracy by calculating the difference in distance between the applications with Google Map. From the test results show that the functionality of the system has met the criteria of the specification of functional requirements at the design stage. In addition the test results accuracy by comparing the level of accuracy of the application with the Google Map shows the difference in average distance is 5.1554 meters.

Keyword: GPS, Google Map, Tracking.

DAFTAR ISI

RANCANG BANGUN PROTOTIPE TRACKING MOBIL PELAYANAN TEKNIK PLN MENGUNAKAN PERANGKAT BERGERAK BERBASIS GPS	i
PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Unit Pelayanan Teknik PLN	4
2.2 Prototipe	4
2.2.1 Jenis-Jenis Prototipe.....	4
2.3 <i>Global Positioning System</i>	5
2.3.1 Jenis-Jenis GPS	7
2.4 <i>Tracking</i>	8
2.5 <i>Database</i>	9
2.5.1 <i>Database Management System</i>	10
BAB 3 METODOLOGI	12
3.1 Studi Literatur	12
3.2 Analisis dan Perancangan	13

3.2.1 Analisis Kebutuhan	13
3.2.2 Perancangan Sistem	13
3.3 Implementasi	14
3.4 Pengujian dan Analisis	15
3.5 Pengambilan Kesimpulan dan Saran	16
BAB 4 ANALISIS DAN PERANCANGAN	17
4.1 Analisis Kebutuhan	17
4.1.1 Gambaran Aplikasi	17
4.1.2 Perubahan Proses Bisnis	18
4.1.3 Identifikasi Aktor	19
4.1.4 Identifikasi Data	19
4.1.5 Daftar Kebutuhan	20
4.1.6 Diagram <i>Use Case</i>	21
4.1.7 Skenario <i>Use Case</i>	23
4.2 Perancangan Sistem	27
4.2.1 Perancangan Arsitektur Sistem	27
4.2.2 Perancangan Aktivitas	28
4.2.3 Perancangan <i>Sequence Diagram</i>	32
4.2.4 Perancangan <i>Class Diagram</i>	36
4.2.5 Perancangan Basis Data	37
4.2.6 Perancangan <i>User Interface</i>	37
BAB 5 IMPLEMENTASI	42
5.1 Implementasi	42
5.1.1 Spesifikasi Perangkat Lunak Sistem	42
5.1.2 Batasan Implementasi	42
5.1.3 Implementasi <i>Class</i> dan <i>Asset</i>	42
5.1.4 Implementasi Kode Program	43
5.1.5 Implementasi Antarmuka Aplikasi	49
BAB 6 Pengujian	53
6.1 Pengujian Validasi	53
6.1.1 Kasus Uji <i>Login</i>	53
6.1.2 Kasus Uji <i>Tracking</i> Kendaraan	53

6.1.3 Kasus Uji Tambah Petugas	54
6.1.4 Kasus Uji Status Petugas	54
6.1.5 Kasus Uji Melaporkan Kondisi Siap Berangkat (Start).....	55
6.1.6 Kasus Uji Melaporkan Tugas Telah Selesai (Stop).....	55
6.1.7 Kasus Uji <i>Logout</i>	55
6.1.8 Hasil Pengujian Validasi.....	56
6.2 Pengujian Akurasi	57
6.3 Analisis	59
6.3.1 Hasil Pengujian Validasi.....	59
6.3.2 Hasil Pengujian Akurasi	59
BAB 7 Penutup	60
7.1 Kesimpulan.....	60
7.2 Saran	60
DAFTAR PUSTAKA.....	61



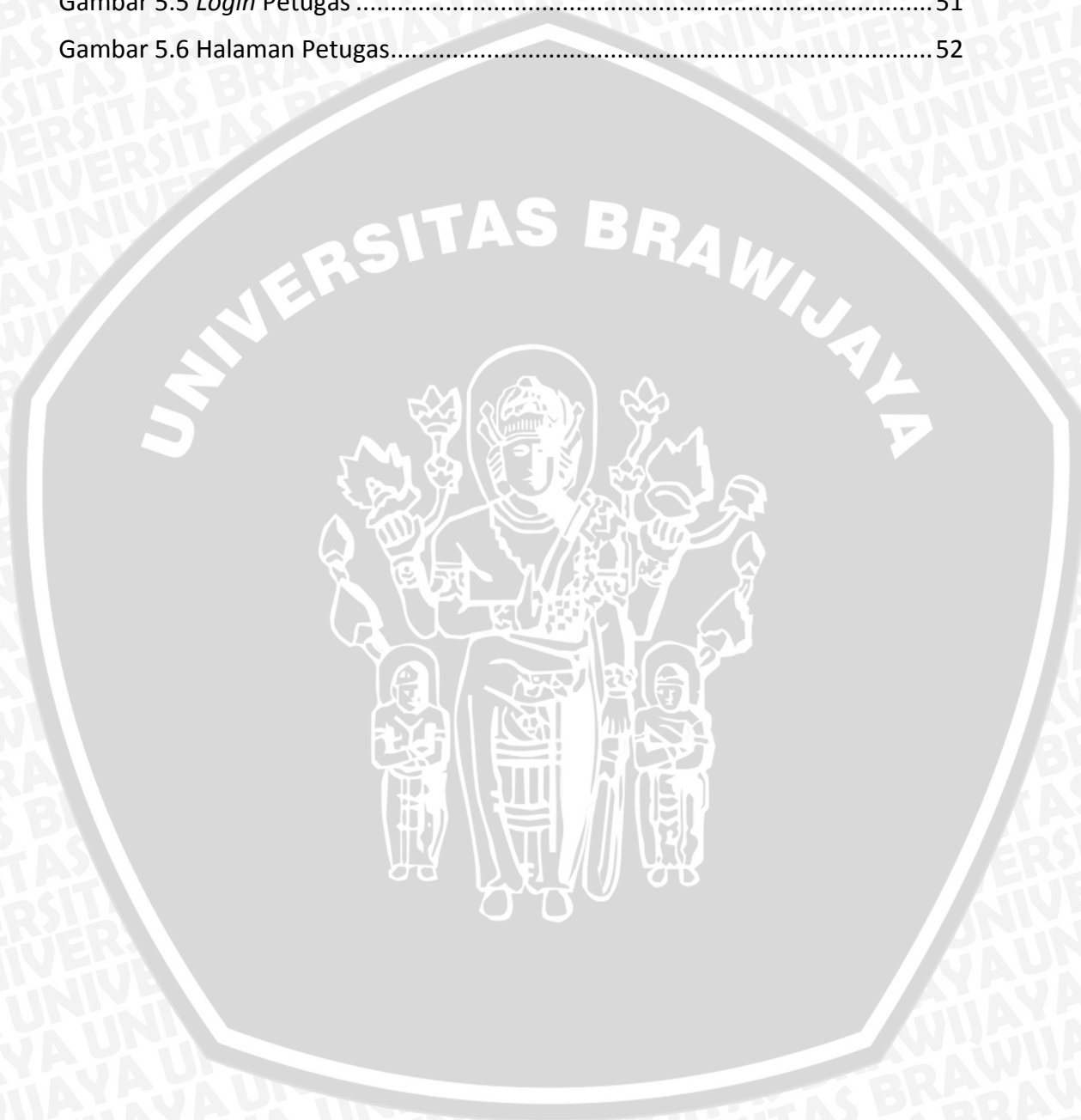
DAFTAR TABEL

Tabel 3.1 Spesifikasi komputer	15
Tabel 3.2 Spesifikasi <i>smartphone</i> 1	15
Tabel 3.3 Spesifikasi <i>smartphone</i> 2	16
Tabel 4.1 Kebutuhan Fungsional	20
Tabel 4.2 Tabel kebutuhan non-fungsional	21
Tabel 4.3 Skenario Use Case <i>Login</i>	23
Tabel 4.4 Skenario Use Case Memantau Posisi Kendaraan	24
Tabel 4.5 Skenario Use Case Tambah Petugas	24
Tabel 4.6 Skenario Use Case Status Petugas	25
Tabel 4.7 Skenario Use Case Mulai <i>Tracking</i> Posisi	25
Tabel 4.8 Skenario Use Skenario Use Case <i>Tracking</i> Selesai	26
Tabel 4.9 Skenario Use Case <i>Logout</i>	27
Tabel 5.1 Spesifikasi perangkat lunak komputer	42
Tabel 5.2 Implementasi <i>class</i> dan <i>assets</i> dalam bentuk <i>file</i>	42
Tabel 6.1 Kasus Uji <i>Login</i>	53
Tabel 6.2 Kasus Uji <i>Tracking</i> Kendaraan	53
Tabel 6.3 Kasus Uji Tambah Petugas	54
Tabel 6.4 Kasus Uji Status Petugas	54
Tabel 6.5 Kasus Uji Melaporkan Kondisi Siap Berangkat	55
Tabel 6.6 Kasus Uji Melaporkan Tugas Telah Selesai	55
Tabel 6.7 Kasus Uji <i>Logout</i>	56
Tabel 6.8 Hasil Pengujian Validasi	56
Tabel 6.9 Hasil Pengujian Akurasi	58
Tabel 6.10 Analisis Pengujian Akurasi	59

DAFTAR GAMBAR

Gambar 3.1 Tahapan Penelitian.....	12
Gambar 4.1 Gambaran Umum Aplikasi	18
Gambar 4.2 <i>Entity Relationship Diagram</i>	20
Gambar 4.3 Use Case Aplikasi Pada Sisi <i>Guest</i>	22
Gambar 4.4 Use Case Aplikasi Pada Sisi Manager	22
Gambar 4.5 Use Case Aplikasi Pada Sisi Petugas	23
Gambar 4.6 Perancangan Arsitektur Sistem.....	28
Gambar 4.7 Activity Diagram <i>Login</i>	29
Gambar 4.8 Activity Diagram Memantau Posisi Kendaraan (<i>Tracking</i>)	29
Gambar 4.9 Activity Diagram Tambah Petugas	30
Gambar 4.10 Activity Status Petugas.....	30
Gambar 4.11 Diagram Activity Mulai <i>Tracking</i> Posisi	31
Gambar 4.12 Diagram Activity <i>Tracking</i> Selesai	31
Gambar 4.13 Diagram Activity <i>Logout</i>	32
Gambar 4.14 <i>Sequence Diagram Login</i>	32
Gambar 4.15 <i>Sequence Diagram</i> Memantau Posisi Kendaraan	33
Gambar 4.16 <i>Sequence Diagram</i> Tambah Petugas.....	33
Gambar 4.17 <i>Sequence Diagram</i> Menampilkan Status	34
Gambar 4.18 <i>Sequence Diagram</i> Melaporkan Kondisi Siap Berangkat	34
Gambar 4.19 <i>Sequence Diagram</i> Melaporkan Tugas Telah Selesai	35
Gambar 4.20 <i>Sequence Diagram Logout</i>	35
Gambar 4.21 <i>Class Diagram Manager</i>	36
Gambar 4.22 <i>Class Diagram Petugas</i>	36
Gambar 4.23 Diagram Logikal Basis Data	37
Gambar 4.24 Halaman Pilih <i>User</i>	38
Gambar 4.25 Halaman <i>Login Manager</i>	38
Gambar 4.26 Halaman Manager.....	39
Gambar 4.27 Halaman Halaman <i>Register</i>	40
Gambar 4.28 Halaman <i>Login Petugas</i>	40
Gambar 4.29 Halaman Petugas.....	41

Gambar 5.1 Pilih User	49
Gambar 5.2 <i>Login</i> Manager	50
Gambar 5.3 Halaman Manager	50
Gambar 5.4 Daftarkan Petugas Baru	51
Gambar 5.5 <i>Login</i> Petugas	51
Gambar 5.6 Halaman Petugas.....	52



DAFTAR KODE

Kode 5.1 Implementasi Alur <i>Source Code</i> Proses <i>Login</i>	43
Kode 5.2 Implementasi Alur <i>Source Code</i> Proses <i>Tracking</i> Kendaraan	44
Kode 5.3 Implementasi Alur <i>Source Code</i> Proses Tambah Petugas Baru	44
Kode 5.4 Implementasi Alur <i>Source Code</i> Proses Menampilkan Status	46
Kode 5.5 Implementasi Alur <i>Source Code</i> Proses <i>Start</i>	47
Kode 5.6 Implementasi Alur <i>Source Code</i> Proses <i>Stop</i>	48
Kode 5.7 Implementasi Alur <i>Source Code</i> Proses <i>Logout</i>	48



BAB 1 PENDAHULUAN

1.1 Latar belakang

PLN adalah perusahaan milik Negara (BUMN) yang bergerak di bidang jasa penyediaan listrik bagi masyarakat. Sebagai perusahaan listrik terbesar di Indonesia, PLN memiliki pelanggan yang tersebar di seluruh pelosok tanah air. Seiring berkembangnya zaman, jumlah pelanggan PLN semakin bertambah. Hal tersebut harus diimbangi dengan peningkatan pelayanan guna menjaga mutu perusahaan. PLN melakukan kerjasama dengan banyak perusahaan jasa layanan yang berupa CV di setiap daerah guna memenuhi layanan pelanggan.

Dengan meningkatnya jumlah pelanggan dan komitmen untuk menjaga kualitas pelayanan banyak permasalahan yang muncul dan beberapa hal yang perlu di perbaiki. Unit pelayanan teknik merupakan titik potensial dalam memberikan jasa pelayanan. Unit pelayanan teknik ini terdiri dari karyawan PLN bersama dengan beberapa karyawan CV terkait. Unit pelayanan teknik bertugas untuk memperbaiki kerusakan listrik yang terjadi di lapangan yaitu rumah-rumah atau perusahaan yang menjadi pelanggan PLN. Unit ini memiliki program-program pelayanan yang harus di capai seperti *response time*. *Response time* yaitu rentang waktu maksimal yang diperlukan oleh petugas pelayanan teknik untuk mencapai lokasi yang terjadi gangguan listrik dalam waktu kurang dari 45 menit. Namun pada kenyataannya banyak petugas layanan yang gagal dalam mencapai *response time* ini. Ada beberapa petugas yang lamban atau bahkan mangkal di suatu tempat. Tidak jarang mereka menyangkal bahwasannya masih sedang melayani pelanggan di tempat lain. Manajemen PLN kesulitan dalam mencari tahu kebenaran yang terjadi.

Di sisi lain, perkembangan teknologi yang kian canggih telah menjadi penopang hidup manusia dalam menjalani aktivitasnya setiap hari. Ketika pertama kali ditemukan telepon seluler masih menggunakan kabel yang berfungsi untuk mentransmisikan sinyal elektrik dari *sender* kemudian merubah sinyal tersebut agar dapat didengar oleh *receiver*. Namun sejak ditemukannya teknologi nirkabel, maka terciptalah telepon seluler nirkabel yang dapat dibawa bepergian sehingga menambah mobilitas pengguna yang dikenal dengan nama *Hand Phone* (telepon genggam).

Telepon genggam yang sudah masuk kedalam jenis teknologi *mobile phone* sendiri telah mengalami pengembangan hingga dapat digunakan dengan berbagai macam fungsi selain fungsi utamanya yaitu untuk berkomunikasi. Fungsi tersebut seperti penambahan aplikasi kalkulator, *stopwatch*, multimedia dan lain sebagainya. Namun terobosan terbaru dari mobile phone saat ini adalah hadirnya *smartphone*. *Smartphone* ini merupakan telepon pintar yang memiliki banyak fungsi. Telah banyak diciptakan aplikasi-aplikasi yang berbasis *smartphone* yang bertujuan untuk memberikan dukungan terhadap aktivitas manusia hingga lebih spesifik. Dukungan tersebut berupa aplikasi bantuan yang menunjang manusia dalam melakukan pekerjaannya. Hampir semua *smartphone* saat ini dilengkapi

dengan *Global Positioning System* (GPS). Salah satunya aplikasi yang menggunakan fungsi dari GPS adalah *map* (peta). Aplikasi *map* dapat membantu manusia dalam melihat keadaan suatu daerah atau wilayah. Aplikasi dapat menampilkan gambar secara visual berupa tata letak seperti jalan, sungai, bangunan dan lain sebagainya. Kemudian dari pengembangan aplikasi *map* ini dapat melahirkan fungsi baru yaitu *tracking*. *Tracking* berguna untuk melakukan pelacakan terhadap suatu objek guna memonitoring keberadaan objek tersebut secara lebih akurat.

PLN yang bergerak di bidang jasa memerlukan sebuah sistem yang dapat menampilkan secara visual melalui *map* keberadaan mobil pelayanan teknik PLN yang bekerja di lapangan. Sistem tersebut bertujuan untuk membantu manajemen PLN dalam mengawasi guna menghindari sifat tidak profesionalisme para petugas pelayanan teknik. Dengan adanya alat ini, diharapkan PLN dapat melakukan *monitoring* terhadap kinerja para petugas pelayanan teknik yang berada di lapangan guna meningkatkan kualitas layanan dan membantu dalam memenuhi target-target kerja yang telah di program. Hal tersebut juga dapat meningkatkan citra baik di mata masyarakat.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, dapat dirumuskan masalah-masalah yang akan dibahas sebagai berikut :

1. Bagaimana rancangan sistem *tracking* posisi mobil pelayanan teknik PLN dengan menggunakan perangkat bergerak yang dapat divisualisasi pada aplikasi server ?
2. Bagaimana cara melakukan *tracking* atau *monitoring* terhadap keberadaan kendaraan pelayanan teknik PLN menggunakan perangkat bergerak berbasis GPS?
3. Bagaimana kinerja dari sistem *tracking* mobil pelayanan teknik PLN ini ?

1.3 Tujuan

Tujuan dari penelitian ini adalah untuk membuat prototipe sistem *tracking* guna mengetahui keberadaan kendaraan pelayanan teknik PLN secara tepat. Selain itu dari hasil visualisasi yang tampil dapat diketahui untuk memantau kinerja para petugas pelayanan teknik PLN, misalnya dapat diperkirakan berapa lama waktu yang diperlukan untuk mengatasi gangguan pada trafo atau kabel yang terkena benang layang-layang apakah memerlukan waktu yang lama. Dengan adanya aplikasi ini maka dapat meningkatkan pelayanan agar lebih efisien.

1.4 Manfaat

Adapun manfaat yang diharapkan dari hasil penelitian rancang bangun prototipe *tracking* mobil pelayanan teknik PLN menggunakan perangkat bergerak berbasis GPS ini adalah sebagai berikut :

1. Mempermudah manajemen PLN dalam memantau kinerja para petugas pelayanan teknik

2. Mempercepat waktu agar petugas segera menuju ke lokasi area kerusakan
3. Mendorong agar petugas pelayanan teknik lebih giat dalam bekerja
4. Memenuhi target kinerja PLN dalam pelayanan yang prima
5. Memastikan masyarakat dapat dilayani dengan baik

1.5 Batasan Masalah

Berdasarkan latar belakang dan rumusan masalah yang telah diuraikan, agar permasalahan yang dirumuskan dapat lebih terfokus, maka penelitian ini dibatasi dalam hal :

1. Perangkat bergerak yang digunakan dalam pembuatan aplikasi ini adalah *smartphone*.
2. Aplikasi dirancang dengan dukungan dari Google Map.
3. Jalur komunikasi antar server dengan client melalui jaringan internet.

1.6 Sistematika Pembahasan

Untuk mencapai tujuan yang diharapkan, maka sistematika penulisan yang disusun dalam laporan tugas akhir ini adalah sebagai berikut :

BAB I Pendahuluan

Berisi tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat dari penelitian serta sistematika penulisan.

BAB II Landasan Teori

Menguraikan tentang dasar teori dan referensi secara luas serta informasi yang diperlukan dalam pengembangan, perancangan, dan implementasi dari permasalahan yang akan dibahas.

BAB III Metode Penelitian

Membahas tentang metode yang digunakan dalam penelitian dan langkah-langkah yang akan dilakukan dalam penulisan yang terdiri dari studi literatur, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian dan analisis, serta pengambilan kesimpulan dan saran.

BAB IV Analisa dan Perancangan

Membahas analisis kebutuhan dan perancangan sistem sesuai dengan teori yang ada.

BAB V Implementasi

Membahas tentang hasil implementasi dari sistem berdasarkan metodologi yang sudah ditentukan sebelumnya, serta berdasarkan perancangan yang sudah dibuat sebelumnya.

BAB VI Pengujian

Membahas tentang proses dan hasil pengujian terhadap sistem yang sudah diimplementasi. Sistem diuji dengan berdasarkan metode-metode yang sudah ditentukan pada tahap metodologi sebelumnya.

BAB VI PENUTUP

Memuat kesimpulan yang diperoleh dari pembuatan dan pengujian perangkat lunak yang dikembangkan serta saran-saran untuk pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Unit Pelayanan Teknik PLN

Unit pelayanan teknik PLN adalah unit yang ditugaskan untuk melakukan pemeliharaan, perbaikan termasuk mengatasi keluhan pelanggan. Unit ini memiliki proses bisnis atau mekanisme kerja dalam menjalankan tugasnya. Ketika ada pelanggan yang lapor kepada PLN melalui telepon bahwa terjadi gangguan listrik di daerah mereka, maka operator akan melaporkannya kepada pihak unit pelayanan teknik melalui *Handy Talky* (HT). Informasi tersebut berupa nama pelapor, alamat pelapor, dan nomor telepon pelapor. Setelah mendapat laporan tersebut unit pelayanan teknik akan segera mengirim petugas untuk meluncur ke lokasi gangguan untuk melaksanakan tugasnya.

2.2 Prototipe

Menurut Zulkifli A. M dalam bukunya yang berjudul “Manajemen Sistem Informasi” yang diterbitkan pada tahun 1977 yang disebut dengan prototipe adalah pemakaian aplikasi khusus perangkat lunak untuk membuat versi skala kecil atau perkiraan pertama program yang direncanakan. Versi skala kecil juga disebut prototipe, dan dapat dibuat relatif cepat, serta kemudian dapat dilihat oleh pemakai dengan mencoba terlebih dahulu. Pemakai dapat mengetes, misalnya, prosedur pemrosesan transaksi, membuat laporan, dan memperbarui (update) contoh-contoh *database*.

Prototipe dibuat untuk gambaran atau sebagai demonstrasi dari suatu pembuatan dan pengembangan *software*. Prototipe juga dapat dikatakan bentuk awal sebagai contoh atau standar dari sebuah entitas. Prototipe digunakan sebagai percobaan untuk pembuatan awal atau untuk tujuan pengembangan sebelum suatu *software* dijual ke pasar. Dengan adanya prototipe ini maka masyarakat luas dapat menuangkan ide mereka untuk mengembangkan suatu *software* prototipe agar dapat dimanfaatkan secara luas. Kebanyakan dari *software* kompleks yang beredar sekarang berasal dari prototipe yang dikembangkan secara maksimal.

2.2.1 Jenis-Jenis Prototipe

Prototipe terbagi menjadi 4 jenis berdasarkan kegunaan dari sistem yang sedang dirancang. Berikut adalah jenis-jenis prototipe :

1. *Feasibility prototyping*. Untuk menguji kelayakan dari teknologi yang akan digunakan untuk sistem informasi yang akan disusun.
2. *Requirement prototyping*. Prototipe jenis ini biasanya digunakan oleh perusahaan atau untuk berbisnis. Menciptakan sistem untuk mengetahui kebutuhan aktivitas bisnis user. Misalnya dalam sebuah perusahaan terdapat

- user direktur, manajer, dan karyawan. Maka penggunaan sistem dapat dibedakan berdasarkan user tersebut sesuai dengan kebutuhannya.
3. Desain *Prototyping*. Digunakan untuk menyusun perancangan sistem informasi yang akan digunakan.
 4. *Implementation prototyping*. Merupakan lanjutan dari rancangan protipe. Prototipe ini langsung disusun sebagai suatu system informasi yang akan digunakan.

Dalam penelitian ini prototipe termasuk kedalam jenis *Implementation Prototyping* karena sistem langsung disusun atau dikerjakan. Setelah pengerjaan selesai kemudian sistem akan segera digunakan atau di implementasikan oleh pihak PLN setelah melalui uji kelayakan.

2.3 Global Positioning System

Global Positioning System (GPS) adalah sistem satelit navigasi dan penentu posisi yang memberikan posisi dan kecepatan tiga dimensi dan informasi waktu, secara kontinyu di seluruh dunia tanpa tergantung kepada waktu dan cuaca (Abidin et al., 2009).

GPS merupakan alat yang dapat memberikan informasi mengenai posisi dimana pengguna berada di permukaan bumi yang berbasiskan satelit. Data dikirim dari satelit berupa sinyal radio dengan data digital. GPS (Global Positioning System) adalah sistem navigasi yang berbasiskan satelit yang saling berhubungan yang berada di orbitnya. Satelit-satelit itu milik Departemen Pertahanan Amerika Serikat (*Departemen of Defense*) yang pertama kali diperkenalkan mulai tahun 1978 dan pada tahun 1994 sudah memakai 24 satelit.

Differensial GPS (DGPS) adalah metode lain yang dapat digunakan untuk mereduksi pengaruh ionosfer. Walaupun DGPS dapat menghasilkan akurasi level sentimeter, tetapi metode ini hanya efektif untuk skala lokal dalam jangkauan sekitar 50 km. Untuk skala regional DGPS tidak dapat digunakan. Jika harus digunakan maka dibutuhkan banyak GPS yang harus dioperasikan pada jarak sekitar 50 km, yang tentunya koreksi ionosfer skala regional menggunakan DGPS menjadi tidak efektif. Dengan alasan tersebut *Wide Area Differensial GPS (WADGPS)* dan *Regional Area Differensial GPS* menjadi semakin banyak digunakan untuk mengatasi keterbatasan metode konvensional DGPS (Buldan et al., 2006).

GPS terdiri dari 24 satelit yang mengorbit bumi pada radius rata-rata 26,560 km dengan fungsi kapasitas masing-masing (Misra and Enge Dec 2001). Setiap lokasi pada koordinat tertentu setidaknya memiliki 4 buah satelit yang berkelompok yang saling berdekatan, jumlah tersebut adalah nominal minimum yang harus dipenuhi sesuai dengan standar. Pada prinsipnya, *GPS receiver* memerlukan hanya memerlukan tiga satelit digunakan untuk menghitung posisi dimensi menggunakan prinsip *Trilateration*, satelit keempat diperlukan untuk memperkirakan offset jam *receiver* dari jam sistem.

Sejumlah sistem telah menggunakan sistem global mobile / *code division multiple access* (GMS / CDMA) jaringan bergerak seluler untuk memperkirakan

lokasi user di luar ruangan. Dua kategori utama dari system lokasi adalah *network-based* dan *handset-based*. *Network-based systems* membangun teknologi untuk menentukan posisi handset dari pengukuran yang dilakukan oleh stasiun seluler. Dalam *handset-based system*, penentuan lokasi dibuat di handset dan perangkat lunak khusus dan / atau perangkat keras yang diperlukan. Sistem *hybrid* juga dapat dimanfaatkan, dimana kedua jaringan dan handset dimodifikasi untuk melakukan fungsi pencarian posisi. Operator telepon seluler menggunakan sejumlah sistem untuk menyediakan layanan menggunakan teknik tertentu diantaranya :

1. *Cell identification* (Cell-ID).
2. *Time of arrival* (TOA).
3. *Time difference of arrival* (TDOA) with modes for enhanced observed time difference (E-OTD), *observed time difference of arrival* (OTDOA), and *uplink time difference of arrival* (U-TDOA).
4. *Angle of arrival* (AOA).
5. *Assisted-GPS* (A-GPS).

Empat teknik pertama digunakan untuk menentukan posisi *mobile user* hanya dengan menggunakan jaringan beberapa operator telepon seluler. Secara keseluruhan hal tersebut tidak dapat memperoleh akurasi posisi yang lebih baik daripada GPS. Teknik tersebut dapat mencapai akurasi dari 50-200m, tapi hampir tidak pernah kurang dari 10m (Hui et al. Nov, 2007). Posisi yang di kalkulasi menggunakan sistem tersebut dapat digunakan untuk meningkatkan GPS. Salah satu keterbatasan system terpadu tersebut adalah posisi yang menggunakan jaringan seluler tidak dapat mencapai akurasi yang lebih baik dari GPS bahkan jauh dari kemampuan GPS. Namun untuk akurasi posisi yang berbasis seluler memiliki kemampuan lebih tinggi di daerah padat tertutup dibandingkan dengan daerah kurang padat. Kelemahan lain dari sistem integrasi ini adalah adanya biaya yang dikenakan untuk mengakses jaringan seluler.

Sebuah GPS menggunakan infrastruktur GPS untuk menentukan lokasi *user*. Sinyal membantu GPS untuk mentransmisikan data melalui jaringan telepon seluler ke *receiver* yang memiliki handset. Pada umumnya, perbedaan koreksi tidak langsung dikirimkan ke *user*, informasi yang dikirimkan dimaksudkan untuk membantu telepon dalam memperoleh sinyal satelit dengan cepat. Bantuan dari jaringan seluler juga dapat diperpanjang dengan melakukan beberapa perhitungan jarak jauh dan mengirim hasilnya ke ponsel (Jacobson 2007). GPS tidak berdiri sendiri, namun saling terkait dengan sinyal dari para pengguna jaringan telepon seluler dengan membatasi wilayah operasinya. Teknik ini tidak memiliki kemampuan *positioning* dalam lingkungan penolakan GPS. Keuntungannya adalah sinyal tersebut dapat membantu GPS untuk mengirim data dalam jaringan telepon seluler (Aboelmagd Noureldin et al., 2012).

Pada penelitian ini, akan dibuat sebuah rancang bangun prototipe *tracking* yang menggunakan GPS sebagai medianya. System GPS akan di tanam pada perangkat mobile android untuk memancarkan sinyal *continue* sehingga sisi server dapat menangkap sinyal tersebut.

2.3.1 Jenis-Jenis GPS

A. Jenis Receiver Berdasarkan Metode Operasi

1. Code Phase Receivers

Sebuah kode *receiver* juga dikenal dengan *code correlating receiver* karena hal tersebut memerlukan akses ke satelit navigasi dari sinyal P- atau C/A-code untuk fungsi. Tipe ini bergantung pada pesan satelit navigasi yang menyediakan *almanac* untuk operasi dan sinyal proses. Untuk tipe ini mampu menghasilkan data navigasi secara *real-time*. Kode *receiver* memiliki kapabilitas *anywhere-fix* dan *consequently*, yang menambah kecepatan waktu *start-up* pada permulaan survei. Kapabilitas *anywhere-fix* dapat melakukan sinkronisasi sendiri dengan waktu GPS pada titik koordinat yang tidak diketahui setelah mengunci pada sinyal dari empat satelit yang telah diperoleh. *Receiver* ini merupakan *receiver* yang murah. *Code phase receiver* mampu memberikan akurasi hingga 25m (tanpa keadaan selektif).

2. Carrier Phase Receiver

Receiver ini menentukan posisi dengan pengukuran proses *carrier phase* dari sinyal satelit pada tiap periode waktu. Tipe ini tidak perlu men-decode informasi yang ditransmisikan kecuali untuk mencari satelit. Beberapa *receiver* tidak memiliki *reception capability* itu artinya *receiver* harus dimuat dengan data dari sumber lain. Keuntungan dari metode ini memiliki akurasi yang tinggi. Tipe *receiver* ini memiliki tingkat akurasi dalam sentimeter dalam *real time* ketika digunakan dalam koreksi yang berbeda

Sebuah tipe *carrier phase receiver* memanfaatkan sinyal GPS secara aktual untuk mengkalkulasi suatu lokasi. Terdapat dua tipe dari *carrier phase receiver* diantaranya :

- (i) *Single-Frequency Receiver*. Sebuah *single-frequency receiver* melacak sinyal L1. Pada umumnya model ini memiliki harga yang lebih murah daripada *dual-frequency receiver* karena model ini memiliki komponen yang lebih sedikit dan memiliki permintaan yang lebih besar. *Single frequency receiver* dapat digunakan secara efektif untuk mengembangkan posisi relative yang akurat atas baseline kurang dari 50 km atau di mana efek ionosfer diabaikan.
- (ii) *Dual-Frequency Receiver*. Sebuah *dual-frequency receiver* melacak sinyal frekuensi L1 dan L2 dan pada umumnya memiliki harga yang lebih mahal daripada *single-frequency receiver*. Sebuah *dual-frequency receiver* akan lebih efektif dalam menyelesaikan *baselines* yang lebih panjang lebih dari 59km dimana efek ionosphere memiliki dampak yang lebih besar dalam perhitungannya. *Dual-frequency receiver* menghilangkan sebagian besar efek dari ionosfer dengan mengkombinasikan observasi L1 dan L2. Kebanyakan produsen *dual-frequency receiver* memanfaatkan teknik *codeless* yang memungkinkan penggunaan L2 selama *anti-spoofing*.

3. Military Grade GPS Receiver

GPS military receiver saat ini adalah *precise lightweight GPS receiver* (PLGR), AN / PSN-11, yang menggunakan akuisisi (C/A), *precise* (P), atau dienkripsi P(Y)-kode. PLGR dirancang untuk beroperasi sebagai unit yang berdiri sendiri dan menyediakan informasi navigasi seperti posisi, kecepatan, dan waktu. PLGR membutuhkan kunci *crypto* untuk beroperasi sebagai PPS (Precise Positioning Service) *receiver*. Sebuah penerima PPS mengoreksi kesalahan diperkenalkan oleh ketersediaan selektif (S/A) dan tidak dapat palsu, ditiru atau dipancarkan kembali. PLGR atau PPS *receiver* adalah satu-satunya *receiver* yang berwenang untuk digunakan di daerah konflik.

2.4 Tracking

Menurut Sudha Challa dalam bukunya yang berjudul "*Fundamentals of Object Tracking*" yang diterbitkan pada tahun 2011 yang dimaksud dengan *object tracking* adalah pelacakan yang mengacu pada suatu objek menggunakan pengukuran sensor untuk menentukan lokasi, jalan, dan karakteristik dari objek yang di ukur. Sebuah sensor dapat menjadi alat ukur, contohnya radar, sonar, lidar, camera, infrared sensor, microphone, ultrasound atau sensor lain yang dapat digunakan untuk mengumpulkan informasi tentang objek dalam suatu daerah. Terdapat beberapa tipe dari tujuan *object tracking* adalah menentukan jumlah objek, identitas dan keadaan, posisi, kecepatan dan beberapa fitur lain yang diperlukan. Salah satu tipe dari contoh *object tracking* adalah *radar tracking* dari pesawat terbang. Permasalahan *object tracking* dalam konteks ini adalah upaya untuk menentukan jumlah dari pesawat terbang yang berada di satu wilayah pengawasan yang sama, tipe dari pesawat, seperti pesawat militer, pesawat komersial atau rekreasi, identitas dari pesawat tersebut, kecepatan dan posisi dari pesawat tersebut, semua didasarkan pada pengukuran yang diperoleh dari radar.

Tracking adalah estimasi suatu keadaan objek yang bergerak yang di ukur berdasarkan jarak perpindahan objek tersebut. Hal ini dilakukan dengan menggunakan satu atau lebih sensor di platforms yang tetap atau bergerak. Pada pandangan pertama tracking mungkin tampaknya tertuju pada kasus estimasi. Namun, pada kenyataannya hal tersebut memiliki lingkup yang lebih luas. Bukan hanya menggunakan tools yang ada pada estimasi, tetapi juga memerlukan ekstensif menggunakan teori keputusan statistic pada beberapa permasalahan perhitungan lain (data association - "which is my measurement?" - lihat, e.g., [Bar-Shalom 95]. (Yaakov Bar-Shalom et al, 2004:2). Hasil rancang bangun prototipe yang akan dibuat akan menghasilkan output pada tampilan layar berupa letak atau posisi dari client. Apabila objek melakukan perpindahan tempat, maka system dapat mengetahui secara tepat kemana objek tersebut berpindah. Hal ini yang merupakan fungsi dari tracking tersebut.

Mobile phone tracking berarti mengetahui letak dari posisi dari ponsel baik ketika diam maupun bergerak. Pencarian dapat terjadi baik melalui multiteral sinyal radio antar beberapa menara radio dari jaringan telepon atau hanya melalui

GPS. Untuk pencarian telepon menggunakan multiteral dari sinyal radio, hal tersebut setidaknya harus memancarkan *roaming signal* untuk mengontak menara antena terdekat berikutnya, namun proses ini tidak memerlukan panggilan aktif. GSM didasarkan pada kekuatan sinyal dari tiang antena terdekat.

Mobile positioning mencakup lokasi berdasarkan layanan yang menyingkap koordinat dari ponsel secara actual. Teknologi tersebut digunakan oleh perusahaan telekomunikasi untuk memperkirakan lokasi dimana ponsel juga termasuk pembawanya berada. Penerapan yang lebih tepat untuk istilah *locating* mengacu pada tujuan daripada proses *positioning*. Layanan tersebut ditawarkan sebagai pilihan kelas dari *location based service* (LBS) (Kevin Roebuck., 2012).

Penelitian ini menggunakan sistem tracking guna untuk mendapatkan posisi objek dalam kasus ini adalah mobil pelayanan teknik PLN secara visual dalam perangkat smartphone. Sistem tracking ini akan mengirim sinyal secara periodik dari perangkat smartphone yang dipantau menuju database kemudian dilanjutkan menuju perangkat smartphone yang menjadi sisi pemantau untuk melakukan monitoring. Monitoring dalam penelitian ini bersifat realtime sehingga data yang ditampilkan pada layar adalah letak dari mobil yang di track. Namun pengiriman sinyal dalam *range* yang tertentu akan dapat mempengaruhi akurasi dari tampilan.

2.5 Database

Database adalah kumpulan *file-file* yang mempunyai kaitan antara satu *file* dengan *file* yang lain sehingga membentuk satu bangunan data untuk menginformasikan satu perusahaan, instansi dalam batasan tertentu. Bila terdapat *file* yang tidak dapat dipadukan atau dihubungkan dengan *file* yang lainnya berarti *file* tersebut bukanlah kelompok dari satu database, ia akan dapat membentuk satu database sendiri (Ir. Harianto Kristanto., 1993).

Menurut Ema Utami dan Sukrisno dalam bukunya yang berjudul “Konsep Dasar Pengelolaan dan Pemrograman *Database* dengan SQL Server, Ms. Access, dan Ms. Visual Basic” yang diterbitkan tahun 2005 Sistem penyimpanan data dalam komputer sering disebut sebagai Sistem Database. Database sendiri merupakan kumpulan data yang pada umumnya menggambarkan aktivitas-aktivitas dan pelakunya dalam suatu organisasi, misalkan database universitas yang berisi mahasiswa, dosen, kuliah, dan lain-lain. Sistem database memiliki empat komponen penting, yakni:

1. Data, merupakan informasi yang disimpan dalam suatu struktur tertentu yang terintegrasi.
2. *Hardware*, merupakan perangkat keras berupa komputer dengan media penyimpanan sekunder yang digunakan untuk menyimpan data karena pada umumnya database memiliki ukuran yang besar.
3. *Software*, merupakan perangkat lunak yang digunakan untuk melakukan pengolahan data. Perangkat lunak ini sering disebut sebagai *Database Management System* (DBMS).

User, merupakan orang yang menggunakan data yang tersimpan dan terkelola. User dapat berupa seorang yang mengelola database tersebut, yang disebut dengan database administrator (dba), bisa juga end user yang mengambil hasil dari pengelolaan database melalui bahasa query. User juga dapat seorang programmer yang membangun aplikasi yang terhubung ke database dengan menggunakan bahasa pemrograman seperti C, Cobol, dan lain-lain.

2.5.1 Database Management System

DBMS merupakan perangkat lunak yang dirancang untuk dapat melakukan utilisasi dan mengelola koleksi data dalam jumlah yang besar. DBMS juga dirancang untuk dapat melakukan manipulasi data secara lebih mudah. Sebelum adanya DBMS maka data pada umumnya disimpan dalam bentuk flat *file*, yaitu *file* teks yang ada pada system operasi. Sampai sekarangpun masih ada aplikasi yang menyimpan dalam bentuk flat *file* secara langsung.

Menyimpan data dalam flat *file* mempunyai kelebihan dan kekurangan. Penyimpanan data dalam bentuk ini akan mempunyai manfaat yang optimal jika ukuran *file* relatif kecil, seperti *file* passwd. *file* passwd pada umumnya digunakan untuk menyimpan nama yang jumlahnya tidak lebih dari 1000 orang.

Selain dalam bentuk flat *file*, penyimpanan data juga dapat dilakukan dengan menggunakan program bantu seperti spreadsheet. Penggunaan Perangkat lunak ini memperbaiki beberapa kelemahan dari flat *file*, seperti bertambahnya kecepatan dalam pengolahan data. Namun demikian metode ini masih memiliki banyak kelemahan, diantaranya adalah masalah manajemen dan keamanan data yang masih berkurang.

Penyimpanan data dalam bentuk DBMS mempunyai banyak manfaat dan kelebihan dibandingkan dengan penyimpanan data dalam bentuk *file* atau spreadsheet, di antaranya:

1. Performance yang didapat dengan penyimpanan dalam bentuk DBMS cukup besar, sangat jauh berbeda dengan performance data yang disimpan dalam flat *file*. Di samping memiliki unjuk kerja yang lebih baik, juga akan didapatkan efisiensi penggunaan media penyimpan dan memori.
2. Integritas data lebih terjamin dengan penggunaan DBMS. Masalah redundansi sering terjadi dalam data flat *file* tidak akan terjadi dalam DBMS. Redundansi adalah kejadian berulangnya data atau kumpulan data yang sama dalam sebuah *database* yang mengakibatkan pemborosan media penyimpanan.
3. Independensi. Perubahan struktur database dimungkinkan terjadi tanpa harus mengubah aplikasi yang mengaksesnya sehingga pembuatan antarmuka ke dalam data akan lebih mudah dengan penggunaan DBMS.
4. Sentralisasi. Data yang terpusat akan mempermudah pengelolaan *database*. Kemudahan di dalam melakukan bagi pakai dengan DBMS dan juga kekonsistenan data yang diakses secara bersama-sama akan lebih

- terjamin daripada data disimpan dalam bentuk *file* ataupun worksheet yang tersebar.
5. Sekuritas. DBMS memiliki system keamanan yang lebih fleksibel daripada pengamanan pada *file* sistem operasi. Keamanan dalam DBMS akan memberikan keluwesan dalam pemberian hak akses kepada pengguna.

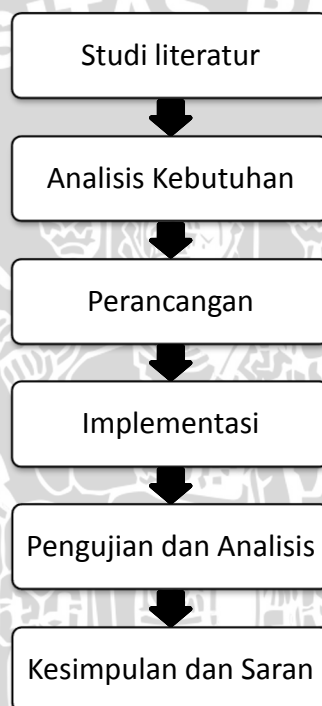
Dengan menggunakan DBMS maka pengguna dapat mendefinisikan data yang akan disimpan dalam suatu bentuk yang dinamakan model data (data model). Model data merupakan koleksi dari konstruksi diskripsi data tingkat tinggi yang menyembunyikan detail tingkat rendahnya. Secara umum terdapat tiga model data yang digunakan untuk memodelkan penyimpanan data, yaitu model hierarki, jaringan, dan relasional. Model hierarki merupakan model data yang dibangun secara hierarki di mana terdapat root sebagai titik puncaknya. Pada model data ini setiap data hanya akan memiliki sebuah penghubung untuk data di bawahnya.

Model data jaringan menggunakan pointer untuk menghubungkan data yang satu dengan data yang lain. Jadi dalam model data jaringan, satu data dapat memiliki banyak penghubung dengan data yang lain. Sedangkan model data relasional menggunakan pendekatan dimana setiap data memiliki relasi dengan data yang lain (Ema Utami, Sukrisno., 2005).

Penggunaan DBMS akan sangat bermanfaat untuk membantu proses pertukaran data untuk alat komunikasi karena berperan sebagai jembatan penghubung antara manager atau operator yang bertugas untuk memonitoring dengan mobil pelayanan teknik PLN. Hal tersebut dapat mempercepat proses pengiriman sinyal periodik yang dikirim oleh client menuju sisi monitor.

BAB 3 METODOLOGI

Pada bab ini dijelaskan langkah-langkah yang akan dilakukan dalam penelitian ini. Langkah pertama yang dilakukan yaitu mengumpulkan teori-teori pendukung dan mengemasnya ke dalam studi literatur. Kemudian dilanjutkan dengan proses analisis kebutuhan. Setelah analisis kebutuhan dilakukan, proses selanjutnya yaitu melakukan perancangan dan melakukan implementasi sesuai dengan perancangan. Kemudian dilakukan pengujian dan analisis pada sistem yang telah dibuat. Kesimpulan dan saran disertakan sebagai catatan atas sistem dan kemungkinan arah pengembangan sistem selanjutnya. Gambar 3.1 menunjukkan tahapan penelitian secara umum.



Gambar 3.1 Tahapan Penelitian

3.1 Studi Literatur

Studi literatur menjelaskan dasar teori yang digunakan untuk menunjang penulisan skripsi. Teori-teori pendukung tersebut meliputi :

- Prototipe*
- Global Positioning System (GPS)*
- Tracking*
- Database*
- Data Base Management System (DBMS)*

3.2 Analisis dan Perancangan

Analisa dan perancangan merupakan sebuah tahapan yang berfungsi untuk merumuskan kebutuhan-kebutuhan yang diperlukan dalam pengimplementasian aplikasi ini. Tahapan-tahapan yang ada dalam analisis dari perancangan antara lain:

3.2.1 Analisis Kebutuhan

Dalam tahapan ini diperlukan gambaran tentang bagaimana jalannya aplikasi dan analisa tentang kebutuhan yaitu menjelaskan menu-menu apa saja yang akan dibutuhkan untuk aplikasi *tracking* mobil PLN ini.

Selain itu, pada tahapan ini diperlukan analisa kebutuhan sistem yang dibutuhkan, yaitu menjelaskan aktor yang terlibat dan interaksinya terhadap aplikasi. Misal manager dan beberapa *smartphone* yang diletakkan di dalam mobil pelayanan teknik yang berlaku sebagai client.

Untuk memperoleh data yang diperlukan dalam penelitian maka dilakukan beberapa metode dalam pengumpulan data diantaranya :

1. Wawancara (*Interview*)

Merupakan metode pengumpulan data dengan melakukan tanya jawab kepada narasumber yang dinilai mengetahui tentang permasalahan yang terdapat pada PLN khususnya pada bidang pelayanan teknik.

2. Pengamatan (*Observasi*)

Pengamatan yaitu teknik pengumpulan data melalui pengamatan langsung terhadap kinerja petugas pelayanan teknik. Berdasarkan pengalaman ketika memanggil petugas pelayanan teknik kerumah ketika terjadi gangguan.

3.2.2 Perancangan Sistem

Dalam perancangan sistem digunakan metode *object-oriented* (OO) dan *structure* yang divisualisasikan dengan UML (Unified Modeling Language), Pendekatan *object oriented* ini bekerja dengan melihat objek-objek yang ada didalam sistem. Pendekatan *structure* digunakan untuk melaksanakan fungsi-fungsi sehingga mampu membentuk suatu perintah tertentu. Kemudian sebagai alat bantu menggunakan *Unified Modeling Language* (UML). UML merupakan Bahasa pemodelan standar pada rekayasa perangkat lunak. UML akan mampu meningkatkan produktifitas dan kualitas serta pengurangan biaya dan waktu.

Pada tahapan ini nantinya akan dibuat rancangan cara kerja sistem secara keseluruhan dan ditentukan fitur-fitur yang diperlukan dalam sistem. Akan dijelaskan pula desain basis data dan relasi antar entitas serta bagaimana desain antarmuka yang akan diimplementasikan. Berikut komponen yang digunakan untuk mendukung UML :

1. Use Case

Use case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* digunakan untuk mendeskripsikan tipe interaksi antara user sebuah sistem dengan sistemnya melalui suatu penjelasan bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara user dengan sistem disebut *scenario*. Setiap *scenario* mendeskripsikan urutan kejadian, dan setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu. Didalam *use case* user disebut aktor yang berperan untuk melakukan interaksi dengan sistem.

2. Sequence Diagram

Sebuah *scenario* memerlukan gambaran untuk menunjukkan aktivitas yang ada pada *seknario*. Oleh karena itu *Sequence diagram* digunakan untuk menggambarkan perilaku yang ada pada sebuah *scenario*. Diagram ini menunjukkan sejumlah contoh objek dan pesan atau *message* yang diletakkan diantara objek-objek di dalam *use case*. Komponen utama *sequence diagram* terdiri dari objek yang dituliskan pada segiempat bernama. Setiap pesan diwakili dengan tanda panah. *Sequence diagram* digunakan untuk mengetahui perilaku beberapa objek pada *use case*.

3. Class Diagram

Class Diagram digunakan untuk menampilkan beberapa kelas serta paket-paket yang ada dalam sistem/perangkat lunak. *Class diagram* memberi gambaran (diagram statis) tentang sistem/perangkat lunak dan relas-relasi yang ada didalamnya. Class juga menggambarkan atribut atau properti suatu sistem sekaligus menawarkan layanan untuk memanipulasi keadaan suatu metoda/fungsi.

4. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan suatu model diagram untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. ERD dibagi menjadi 2 yaitu ERD Logical Data Model dan ERD Physical Data Model. ERD *Logical Data Model* adalah konsep *entity relationship diagram* yang mana data dapat merepresentasikan sebuah kenyataan kemudian di proses secara logika untuk menghasilkan informasi, sedangkan *Entity Relationship Diagram Physical Data Model* adalah konsep ERD yang mana data disimpan pada media penyimpanan dalam suatu susunan fisik.

3.3 Implementasi

Implementasi merupakan tahap pembangunan sistem. Setelah melakukan desain sistem, pada tahapan ini barulah dibuat prototipe aplikasi ini. Berikut adalah beberapa hal yang harus diterapkan diantaranya:

- Implementasi basis data, dengan menggunakan sebuah phpMyAdmin yang untuk menyimpan data petugas.
- Implementasi pengkodean sistem, dengan menggunakan bahasa pemrograman java menggunakan Android Studio.

3.4 Pengujian dan Analisis

Pengujian perangkat lunak merupakan tahapan untuk mengetahui terpenuhinya kinerja dan performa sistem sesuai dengan kebutuhan dari user (PLN). Hal tersebut dilakukan agar pemanfaatan dari hasil penelitian ini dapat mencapai target sehingga pelayanan dapat dimaksimalkan. Berikut beberapa poin pengujian yang dilakukan kepada sistem :

1. Pengujian Validasi

Pengujian validasi dengan menggunakan metode *black-box testing* untuk memeriksa hasil eksekusi aplikasi dan fungsionalitasnya. Kesesuaian fitur-fitur aplikasi dengan kebutuhan sistem dilakukan untuk mengetahui validitas aplikasi.

2. Pengujian Akurasi

Pengujian dilakukan untuk membandingkan jarak antara posisi *icon* kendaraan petugas pada map dalam aplikasi dengan posisi kendaraan petugas yang sebenarnya.

Sebagai tambahan, Tabel 3.1 merupakan spesifikasi komputer yang akan digunakan untuk melakukan proses penyusunan *source code* dan pengembangan aplikasi dan Tabel 3.2 merupakan spesifikasi perangkat lunak yang digunakan selama proses pengembangan aplikasi.

Tabel 3.1 Spesifikasi Komputer

Sistem Operasi	Windows 7
System Model	Inspiron N4010
CPU	Intel(R) Core(TM) i3 CPU M 330 @2.13Ghz
Memory	6144 MB RAM
GPU	AMD Mobility Radeon HD 5000 Series

Tabel 3.2 Spesifikasi Smartphone 1

Sistem Operasi	Android 5.0
System Model	ASUS Zenfone 2
CPU	Quad-core 1.8 GHz
Memory	2 GB RAM
GPU	PowerVR G6430

Tabel 3.3 Spesifikasi Smartphone 2

Sistem Operasi	Android 5.0
System Model	Sony Xperia M4 Aqua
CPU	Octa-core 1.5 GHz Cortex-A53
Memory	2 GB RAM
GPU	Adreno 405

3.5 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah semua tahapan perancangan sistem, implementasi sistem, dan pengujian sistem telah selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis rancang bangun prototipe *tracking* mobil pelayanan teknik PLN menggunakan perangkat bergerak berbasis GPS. Penulisan saran berguna untuk memberikan pertimbangan atas hasil yang telah dilakukan.



BAB 4 ANALISIS DAN PERANCANGAN

Pada bab ini menjelaskan tentang analisis sistem dan desain dari aplikasi tracking mobil pelayanan teknik PLN pada perangkat bergerak yaitu *smartphone*. Pada tahap analisis terdiri dari gambaran dan alur dari aplikasi, identifikasi data, daftar kebutuhan, memodelkan ke dalam use case dan diagram *activity*. Untuk tahap perancangan terdiri dari arsitektur, perancangan basis data, pemodelan diagram sequence untuk penggambaran interaksi antar objek pada aplikasi. Membuat diagram *class* yang dipergunakan untuk memberikan gambaran mengenai struktur *class* yang ada di dalam aplikasi tracking mobil pelayanan teknik PLN. Kemudian perancangan antarmuka pengguna baik pada sisi manager maupun sisi mobil.

4.1 Analisis Kebutuhan

Tahap analisis kebutuhan merupakan tahap mengidentifikasi semua kebutuhan dari rancang bangun prototipe *tracking* mobil pelayanan teknik PLN menggunakan perangkat bergerak berbasis GPS. Terdapat beberapa tahapan dalam analisis kebutuhan diantaranya adalah gambaran aplikasi, proses bisnis baru, identifikasi aktor, daftar kebutuhan fungsional, dan non-fungsional dengan menggunakan pemodelan diagram *use case*. Analisis kebutuhan ini bertujuan untuk memberikan analisa secara jelas tentang apa-apa saja kebutuhan yang disediakan sistem demi memenuhi kebutuhan pengguna.

4.1.1 Gambaran Aplikasi

Prototipe *tracking* kendaraan pelayanan teknik PLN ini merupakan aplikasi yang dirancang untuk mempermudah manager pelayanan teknik PLN dalam memantau keberadaan mobil pelayanan teknik atau lebih tepatnya keberadaan para petugas lapangan ketika dalam jam kerja. Sulitnya manager dalam mengetahui kebenaran dari lokasi keberadaan mobil pelayanan teknik menjadi alasan dibuatnya prototipe ini. Prototipe ini dioperasikan dengan memanfaatkan koneksi internet (GPRS). Sistem di tanamkan pada perangkat mobile berupa *smartphone* yang diletakkan pada mobil pelayanan teknik PLN dengan memanfaatkan sumber daya listrik dari pemantik rokok untuk senantiasa mengisi daya agar baterai *smartphone* tidak habis. Sisi client akan mengirimkan koordinat lokasi secara periodik pada web server. Kemudian web server akan mengirimnya menuju *smartphone* yang di pegang oleh manager yang menjadi sisi pemantau.

Pada sisi yang di *track* (kendaraan PLN) tampilan pada layar akan menampilkan tombol *start* dan *stop*, karena sistem berjalan melalui *background* sehingga tidak terlihat pada tampilan di layar kecuali tombol *start* dan *stop*. Hal tersebut untuk menghindari perubahan settingan yang dilakukan oleh pihak yang tidak memiliki hak untuk merubah operasi dari sistem. Pada sisi manager, setelah di klik aplikasi maka pada tampilan akan muncul *map*. *Map* tersebut seperti *map* pada umumnya. Untuk gambaran aplikasi dapat dilihat dalam Gambar 4.1.



Gambar 4.1 Gambaran Umum Aplikasi

4.1.2 Perubahan Proses Bisnis

Dengan adanya sistem ini proses bisnis PLN mengalami sedikit perubahan untuk menyesuaikan dengan adanya sistem yang baru. Pada proses bisnis baru, dalam satu kendaraan terdapat pegawai PLN yang diberi tugas sebagai kepala tim. Kepala tim mendapat username dan password agar dapat masuk kedalam sistem. Username dan nama kepala tim tersebut yang nantinya sebagai identitas dari satu tim dan yang membedakan dengan tim yang lainnya. Pada proses bisnis yang lama Setelah mendapat laporan baik dari pelanggan maupun informasi lainnya maka operator akan segera menginformasikan kepada seluruh petugas pelayanan teknik melalui *Handy Talky* (HT) bahwasanya sedang terjadi gangguan di suatu tempat. Operator juga memberitahukan mengenai informasi pelapor seperti nama pelanggan, alamat dan nomor telepon. Kemudian petugas pelayanan teknik dapat langsung berangkat menuju lokasi kerusakan.

Pada proses bisnis yang lama setelah mendapat laporan, petugas akan di instruksikan melalui HT agar menuju ke lokasi kerusakan. Pada proses bisnis baru, sesaat sebelum berangkat menuju lokasi kerusakan petugas diharuskan menekan tombol *start* pada aplikasi sehingga sistem segera melakukan pengiriman data menuju server. Setelah menekan tombol *start* petugas baru berangkat menuju lokasi kerusakan. Manager dapat langsung memantau keberadaan petugas pelayanan teknik melalui smartphone yang telah di install aplikasi tracking. Selama petugas sedang bertugas keberadaan mereka dapat diketahui. Manager dapat

melihat kehadiran petugas lapangan dengan melihat status petugas lapangan sedang online atau offline dan juga melihat jenis kendaraan apa yang digunakan oleh sebuah tim. Manager juga dapat menilai kinerja dengan melihat laporan jumlah tugas yang telah diselesaikan oleh suatu tim.

4.1.3 Identifikasi Aktor

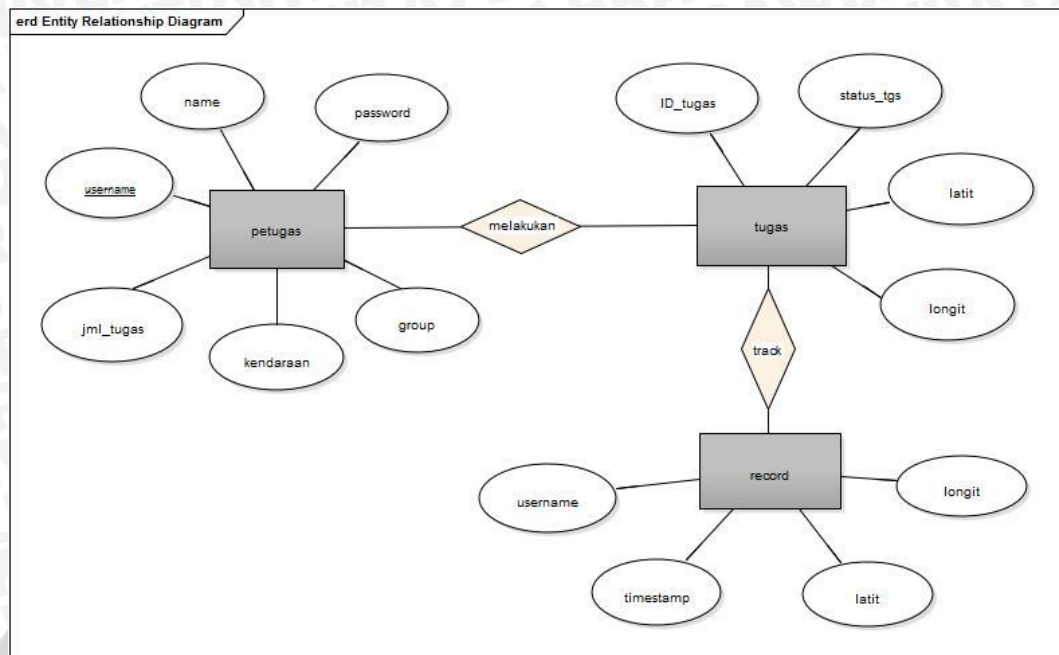
Tahap ini bertujuan untuk melakukan identifikasi terhadap aktor yang terlibat dengan sistem tracking mobil pelayanan teknik PLN ini. Aktor sebagai orang yang berinteraksi langsung dengan sistem memiliki peran untuk mengoperasikannya. Aktor disini terdiri dari 2 jenis yaitu yang pertama manager dan yang kedua adalah petugas pelayanan teknik. Manager sebagai pemantau dan petugas merupakan yang di pantau keberadaannya melalui kendaraan yang dikendarai.

4.1.4 Identifikasi Data

Kendaraan pelayanan teknik yang ada di setiap kantor PLN masing-masing memiliki *username* yaitu sebagai identitas unik yang membedakan antara kendaraan satu dengan kendaraan yang lainnya. Untuk setiap mobil telah diberikan username dengan kode berupa nama dari wilayah atau kantor dimana mobil tersebut ditugaskan kemudian diikuti oleh kode berupa angka. Contoh : Pasuruan 1, Sukorejo 2. Pasuruan adalah nama kota dan Sukorejo adalah nama sebuah wilayah kecamatan di Kabupaten Pasuruan. Angka 1 menunjukkan mobil pelayanan pertama, angka 2 menunjukkan mobil pelayanan kedua. Setiap mobil pelayanan teknik terdapat beberapa petugas yang bekerja sebagai satu tim.

Selain mobil terkadang pada beberapa kantor unit pelayanan jaringan (UPJ) terdapat kendaraan pelayanan teknik berupa motor. Motor ditugaskan apabila gangguan yang terjadi pada suatu tempat tidak terlalu berat sehingga dapat ditangani oleh seorang atau dua orang petugas teknik. Armada motor juga diperlukan apabila gangguan terjadi di daerah yang hanya bisa di lewati oleh motor dikarenakan kondisi jalan atau medan yang tidak memungkinkan untuk di lewati mobil.

Untuk membedakan antara mobil dengan motor pada map, maka untuk icon mobil digambarkan dengan warna biru. Sedangkan untuk icon motor digambarkan dengan warna hijau. Untuk user yang sedang tidak bertugas ditunjukkan dengan *icon* berwarna merah yang artinya sedang offline. Identifikasi data dapat dilihat pada *entity relationship diagram* dalam Gambar 4.2.



Gambar 4.2 Entity Relationship Diagram

4.1.5 Daftar Kebutuhan

Daftar kebutuhan terdiri dari kebutuhan fungsional dan non-fungsional. Dalam Tabel 4.1 kebutuhan fungsional ditunjukkan dengan kode penomoran KFX, sedangkan kebutuhan non-fungsional ditunjukkan dengan kode penomoran KNFX dalam Tabel 4.1.

Tabel 4.1 Kebutuhan Fungsional

Kode	Kebutuhan	Use Case	Aktor
KF01	Aplikasi harus menyediakan fasilitas login atau autentifikasi bagi manager atau client	Login	Guest
KF02	Aplikasi harus bisa menampilkan lokasi keberadaan mobil pelayanan teknik PLN	Memantau posisi kendaraan (tracking)	Manager
KF03	Aplikasi harus menyediakan fasilitas untuk menambahkan petugas baru kedalam sistem	Tambah Petugas	Manager
KF04	Aplikasi harus dapat menampilkan status, data petugas dan jumlah tugas yang telah diselesaikan.	Status Petugas	Manager

KF05	Aplikasi harus menyediakan fasilitas start untuk memulai sistem agar kendaraan yang bertugas dapat di tracking sehingga tampil pada map manager	Melaporkan kondisi siap berangkat (start)	Petugas
KF06	Aplikasi harus menyediakan fasilitas stop untuk menunjukkan kepada manager bahwa tugas telah selesai dilaksanakan.	Melaporkan tugas telah selesai (stop)	Petugas
KF07	Aplikasi harus bisa mereset perhitungan total tugas	Logout	Petugas

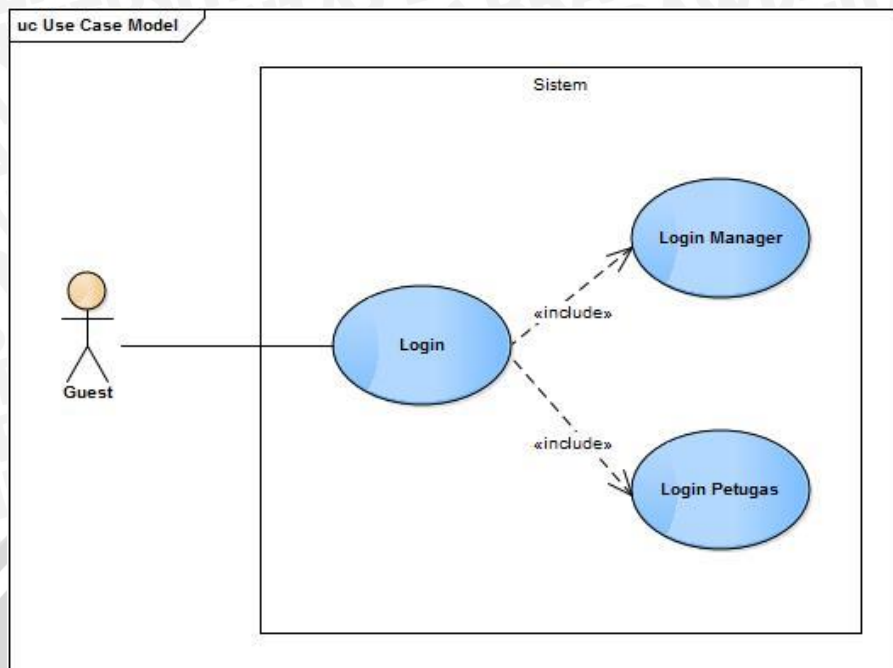
Tabel 4.1 merupakan table kebutuhan fungsional yang terdiri dari dua kebutuhan sistem yang harus dapat dipenuhi sistem. Setiap kebutuhan fungsional sistem dimodelkan dalam bentuk use case, dan setiap kebutuhan fungsional harus dilakukan pengujian terlebih dahulu sebelum sistem di publikasikan.

Tabel 4.2 Kebutuhan non-fungsional

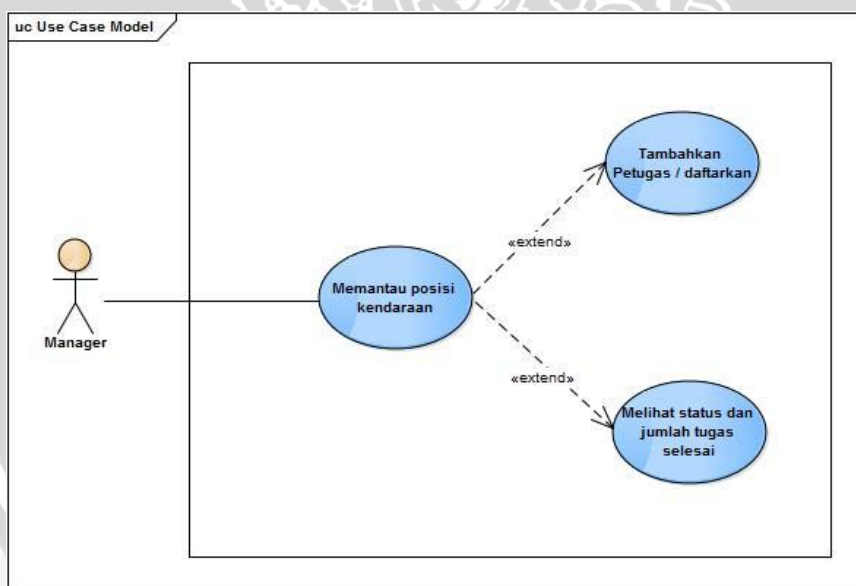
Parameter	Kode	Deskripsi Kebutuhan
<i>Compatibility</i>	KNF01	Aplikasi harus diakses pada perangkat bergerak dengan sistem operasi Android pada beberapa versi
<i>Usability</i>	KNF02	Sistem harus dapat digunakan dengan mudah oleh pengguna baik pada sisi manager maupun sisi client pada semua jenis <i>smartphone</i> android dengan target <i>usability</i> lebih dari 80%

4.1.6 Diagram Use Case

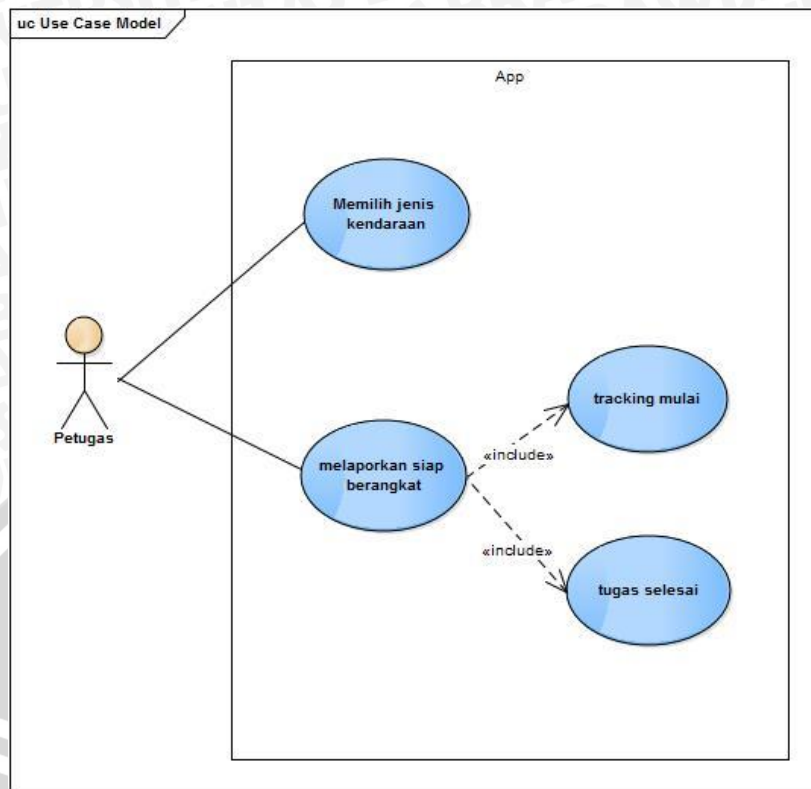
Pemodelan *use case* sistem dirancang berdasarkan kebutuhan fungsional yang digunakan untuk menggambarkan interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Dalam Gambar 4.3 merupakan diagram use case dari sistem tracking mobil pelayanan teknik PLN.



Gambar 4.3 Use Case Aplikasi Pada Sisi Guest



Gambar 4.4 Use Case Aplikasi Pada Sisi Manager



Gambar 4.5 Use Case Aplikasi Pada Sisi Petugas

4.1.7 Skenario Use Case

Skenario *use case* berisi tentang penjelasan dari tiap-tiap use case yang ada dalam Tabel 4.3. Pada setiap scenario diberikan uraian *use case*, aktor yang terlibat, tujuan, deskripsi, alur dari *use case*.

Tabel 4.3 Skenario Use Case Login

Kode Use Case	KF01
Nama <i>use case</i>	<i>Login</i>
Tujuan	Tujuan dari <i>use case</i> ini adalah untuk memilih user untuk login.
Aktor	<i>Guest</i>
Deskripsi	<i>Guest</i> adalah user yang masih belum teridentifikasi jabatannya di dalam kantor. Oleh sebab itu guest wajibkan memilih user yang tersedia antara manager atau petugas.
Prakondisi	-
Aliran Utama	

1. Guest mengaktifkan system 2. Sistem menampilkan halaman pilihan user 3. Guest memilih user sesuai dengan jabatannya 4. Sistem menampilkan halaman autentifikasi sesuai user yang telah dipilih 5. Guest memasukkan password sesuai dengan user pilihannya	
Kondisi Akhir Sukses	Muncul halaman autentifikasi sesuai dengan apa yang telah dipilih
Kondisi Akhir Gagal	-

Tabel 4.4 Skenario Use Case Memantau Posisi Kendaraan

Kode Use Case	KF02
Nama use case	Memantau Posisi Kendaraan
Tujuan	Melakukan monitoring
Aktor	Manager
Deskripsi	Use case ini menampilkan map dan menunjukkan lokasi dimana kendaraan yang sedang di monitoring berada pada perangkat smarphone manager.
Prakondisi	-
Aliran Utama	
1. Manager melakukan login 2. Sistem menampilkan map beserta semua kendaraan yang ada.	
Kondisi Akhir Sukses	Aplikasi menampilkan map beserta seluruh kendaraan yang di pantau
Kondisi Akhir Gagal	-

Tabel 4.5 Skenario Use Case Tambah Petugas

Kode Use Case	KF03
Nama use case	Tambah Petugas
Aktor	Manager
Tujuan	Menambahkan petugas baru
Deskripsi	Use case ini menjelaskan bagaimana manager dapat menambahkan atau mendaftarkan petugas baru.

Prakondisi	Aplikasi menampilkan map
Aliran Utama	
1. Manager login sistem 2. Sistem menampilkan map 3. Manager memilih menu daftarkan 4. Sistem menampilkan halaman isi data petugas 5. Manager mengisi data petugas	
Kondisi Akhir Sukses	Muncul notifikasi Registrasi petugas telah berhasil
Kondisi Akhir Gagal	-

Tabel 4.6 Skenario Use Case Status Petugas

Kode Use Case	KF04
Nama use case	Status Petugas
Aktor	Manager
Tujuan	Memperlihatkan status petugas
Deskripsi	Use case ini menjelaskan bagaimana manager melihat status dan jumlah tugas yang telah diselesaikan oleh petugas.
Prakondisi	User Offline
Aliran Utama	
1. Manager login 2. Sistem menampilkan map dan icon kendaraan beserta status petugas	
Kondisi Akhir Sukses	Aplikasi menampilkan nama username atau ketua grup, nama grup, jumlah tugas yang telah diselesaikan dan icon kendaraan sesuai dengan warna jenis kenaraan
Kondisi Akhir Gagal	Akan muncul notifikasi yang menjelaskan bahwa lokasi kendaraan tidak di temukan atau kosong

Tabel 4.7 Skenario Use Case Mulai Tracking Posisi

Kode Use Case	KF05
Nama use case	Mulai Tracking Posisi
Aktor	Petugas

Tujuan	Melaporkan kepada manager atau admin bahwa petugas siap berangkat.
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana petugas melaporkan kondisi siap berangkat untuk bertugas menuju tempat kerusakan melalui perangkat <i>mobile</i> mereka.
Prakondisi	Petugas telah <i>login</i>
Aliran Utama	
1. Sistem menampilkan pilihan <i>user login</i> 2. Petugas memilih login client atau petugas 3. Sistem menampilkan halaman autentifikasi 4. Petugas mengetikkan <i>username</i>, <i>password</i> beserta jenis kendaraan yang akan di kendarainya 5. Sistem menampilkan halaman <i>tracking</i> 6. Petugas memilih <i>start tracking</i>	
Kondisi Akhir Sukses	Muncul pilihan <i>stop tracking</i>
Kondisi Akhir Gagal	-

Tabel 4.8 Skenario *Use Case Tracking* Selesai

Kode Use Case	KF06
Nama use case	<i>Tracking Selesai</i>
Aktor	Petugas
Tujuan	Melaporkan kepada manager atau admin bahwa tugas telah selesai dilaksanakan.
Deskripsi	<i>Use case</i> ini menjelaskan bagaimana petugas melaporkan bahwa tugas telah selesai dilaksanakan melalui perangkat <i>mobile</i> mereka.
Prakondisi	Sistem sedang dalam kondisi <i>tracking</i> dengan indikasi menampilkan pilihan <i>stop tracking</i>
Aliran Utama	
1. Petugas memilih <i>stop tracking</i>	
Kondisi Akhir Sukses	Sistem menampilkan pilihan <i>start tracking</i>

Kondisi Akhir Gagal	-
---------------------	---

Tabel 4.9 Skenario Use Case Logout

Kode Use Case	KF07
Nama use case	<i>Logout</i>
Aktor	Petugas
Tujuan	Menunjukkan petugas keluar dari system, mereset jumlah tugas yang telah selesai
Deskripsi	Use case ini menjelaskan bagaimana petugas keluar dari sistem. Use case juga menandakan pergantian jam kerja/shift pada petugas.
Prakondisi	Jumlah tugas yang telah di selesaikan masih belum menunjukkan angka nol
Aliran Utama	
1. Petugas melakukan <i>logout</i>	
Kondisi Akhir Sukses	Sistem menampilkan halaman pilihan user login
Kondisi Akhir Gagal	-

4.2 Perancangan Sistem

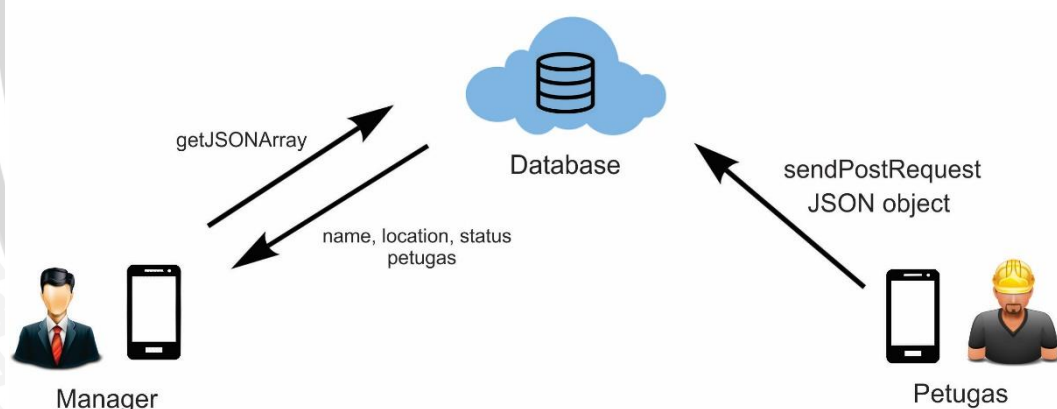
Perancangan sistem menjelaskan mengenai gambaran dari sistem yang akan dibuat. Perancangan sistem dilakukan setelah kebutuhan-kebutuhan dari sistem telah terkumpul dari proses analisis kebutuhan. Untuk perancangan sistem pada rancang bangun prototipe *tracking* mobil pelayanan teknik PLN menggunakan perangkat bergerak berbasis GPS meliputi 6 tahap yaitu perancangan arsitektur sistem, perancangan aktivitas, perancangan *sequence diagram*, perancangan *class diagram*, perancangan basis data, dan perancangan *interface*. Masing-masing perancangan menjelaskan mengenai langkah dari setiap poin untuk melakukan fungsinya.

4.2.1 Perancangan Arsitektur Sistem

Perancangan arsitektur sistem dibagi kedalam dua arsitektur. Arsitektur yang pertama ditujukan untuk manager dan arsitektur yang kedua ditujukan untuk petugas. Masing-masing arsitektur memiliki fungsi yang berbeda sesuai dengan

aktor yang mengoperasikannya. Untuk arsitektur pada sisi manager, manager diharuskan melakukan autentifikasi (*login*) setelah manager login maka tampilan awal adalah map termasuk *icon-icon* kendaraan yang sedang di *track*. Manager juga dapat menambah atau mendaftarkan petugas baru. Hal ini dilakukan apabila terjadi penambahan armada pada bagian pelayanan teknik PLN.

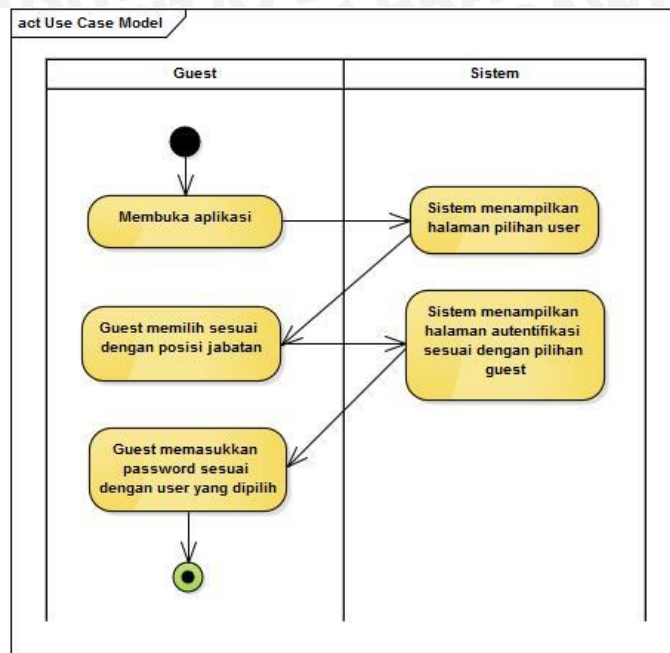
Arsitektur pada sisi petugas ketika membuka aplikasi, petugas harus melakukan *login* dengan cara memasukkan *username* dan *password* kemudian memilih jenis kendaraan yang akan di kendarai. Setelah login sistem akan menampilkan halaman start tracking. Satu armada kendaraan diisi dengan satu tim dan masing-masing tim memiliki ID tim berupa nama grup. Langkah terakhir sebelum petugas berangkat ke lokasi kerusakan yaitu menekan tombol start sehingga sistem akan mulai merekam jejak dari perjalanan mobil. Setelah selesai melaksanakan tugas maka petugas harus menekan tombol stop untuk melaporkan bahwasannya tugas telah dilaksanakan. Hal tersebut menunjukkan tanda selesainya suatu tugas dengan memberhentikan rekam jejak *tracking* mobil, menambah jumlah tugas yang telah selesai dilaksanakan. Hasil perhitungan jumlah tugas ditampilkan diatas *icon* kendaraan yang bersangkutan. Petugas dapat melakukan *logout* apabila waktu jam kerja telah berakhir atau terjadi pergantian petugas dikarenakan jam piket (pergantian shift kerja) telah berakhir.



Gambar 4.6 Perancangan Arsitektur Sistem

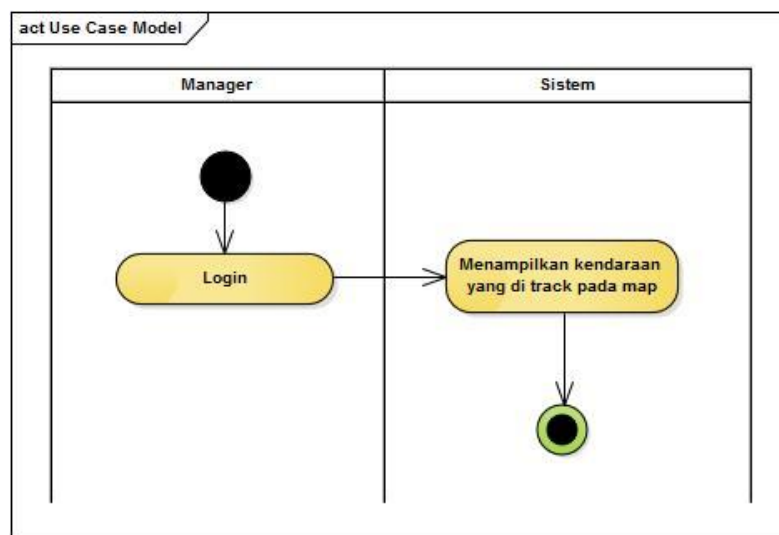
4.2.2 Perancangan Aktivitas

Perancangan aktivitas memodelkan antara pengguna dengan sistem yang berjalan pada skenario *use case*. Aktivitas digambarkan menggunakan Diagram *Activity* yang bisa dilihat dalam Gambar 4.6 sampai 4.13.



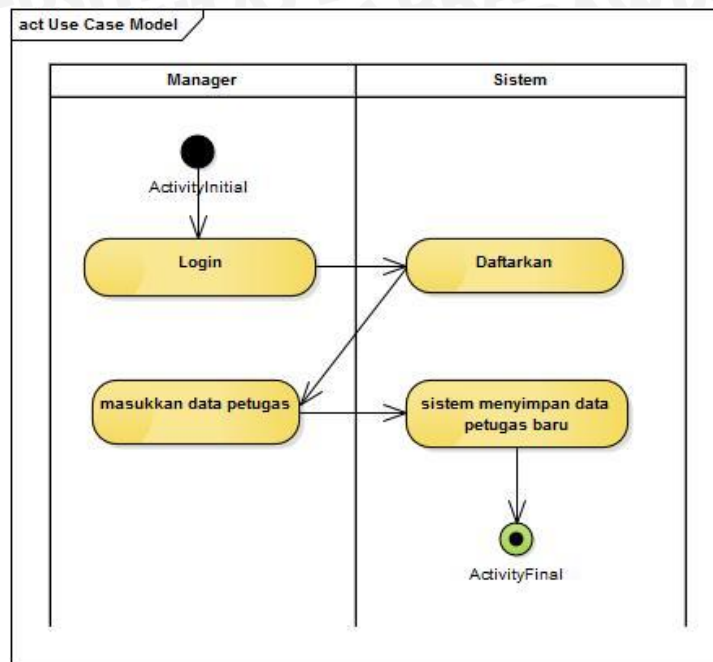
Gambar 4.7 Activity Diagram Login

Dalam Gambar 4.7 menunjukkan aktivitas apabila *guest* sedang menggunakan aplikasi. Sistem belum mengetahui identitas dari *guest* oleh sebab itu *guest* dapat berupa manager atau petugas sehingga *guest* diwajibkan untuk memilih pilihan *login* sesuai dengan posisi jabatannya.



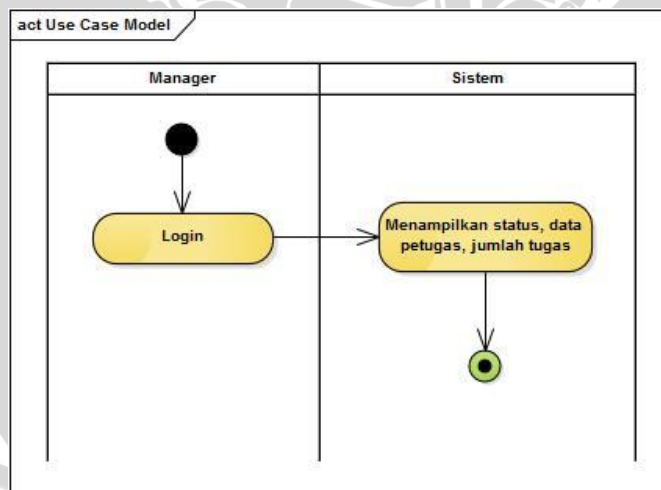
Gambar 4.8 Activity Diagram Memantau Posisi Kendaraan (Tracking)

Dalam Gambar 4.8 menunjukkan aktivitas menampilkan lokasi kendaraan pada *map*. Ketika manager membuka aplikasi, maka sistem akan langsung menampilkan *map* beserta kendaraan yang di *track* pada layar. Sistem menerima data koordinat dari sisi petugas.



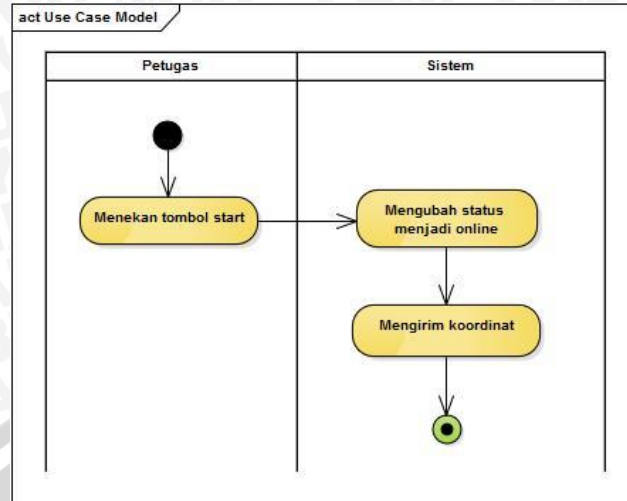
Gambar 4.9 Activity Diagram Tambah Petugas

Dalam Gambar 4.9 menunjukkan aktivitas menambah petugas yang dilakukan oleh manager. Deskripsi *activity* diagram sesuai dengan skenario yang ada pada use case tambah petugas. Manager harus login terlebih dahulu agar bisa masuk kedalam sistem lalu pilih menu daftarkan. Manager memasukkan data petugas seperti username, password, nama tim. Data petugas yang telah diinputkan tadi kemudian di simpan kedalam database.



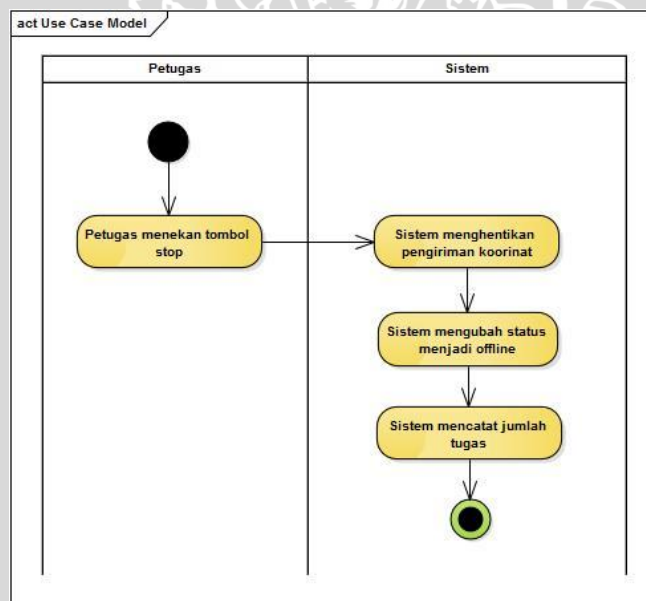
Gambar 4.10 Activity Diagram Status Petugas

Dalam Gambar 4.10 menunjukkan aktivitas status petugas dengan manager sebagai aktor. Manager diharuskan masuk kedalam sistem melalui *login*. Sistem akan menampilkan posisi petugas pada map beserta status, data petugas, dan jumlah tugas yang telah di selesaikan.



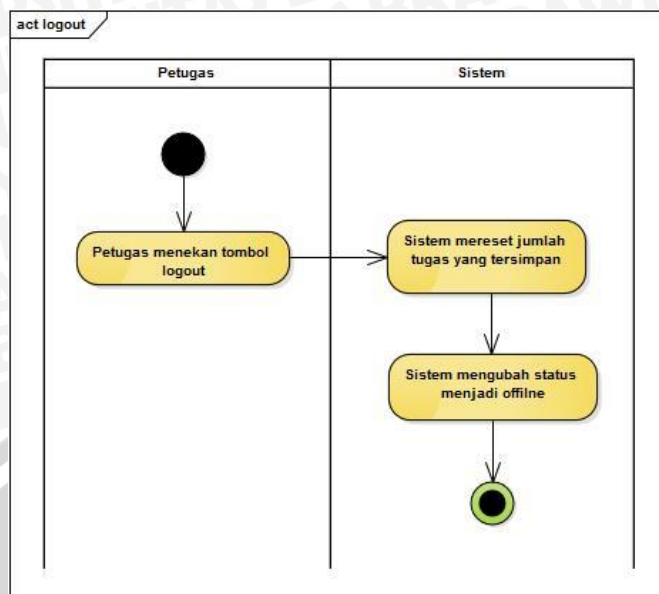
Gambar 4.11 Activity Diagram Mulai Tracking Posisi

Dalam Gambar 4.11 menunjukkan aktivitas mulai *tracking* posisi atau melaporkan kondisi siap berangkat oleh petugas lapangan. Setelah melakukan *login* sistem akan menampilkan halaman *start*. Pada halaman *start* terdapat tombol *start*. Setelah petugas menekan tombol *start* sistem akan mengirim koordinat lokasi secara *continous* menuju *database* server.



Gambar 4.12 Activity Diagram Tracking Selesai

Dalam Gambar 4.12 menunjukkan aktivitas *tracking* selesai atau *stop tracking*. Setelah petugas menekan tombol *stop*, sistem akan menghentikan pengiriman koordinat. Sistem mengubah status dari *online* menjadi *offline* dan juga menambah jumlah tugas yang telah diselesaikan oleh suatu tim.



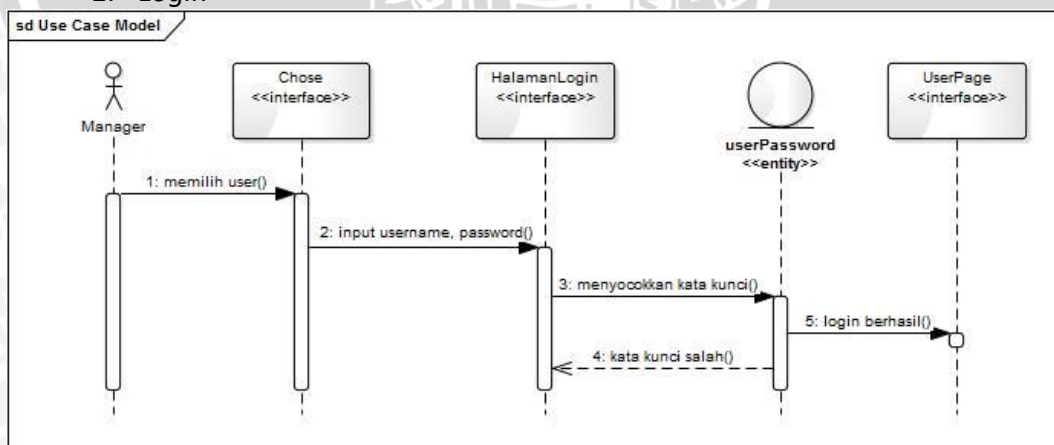
Gambar 4.13 Activity Diagram Logout

Dalam Gambar 4.13 menunjukkan aktivitas *logout*. Setelah petugas melakukan logout sistem akan mereset ulang waktu yang telah tercatat kembali ke 0. Sistem juga melakukan reset pada perhitungan jumlah tugas yang telah di selesaikan oleh tim yang menggunakan kendaraan selama bertugas kembali ke angka 0.

4.2.3 Perancangan Sequence Diagram

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi di antara obyek-obyek tersebut. Berikut adalah *sequence diagram* untuk rancang bangun prototipe *tracking* mobil pelayanan teknik PLN menggunakan perangkat bergerak berbasis GPS :

1. Login

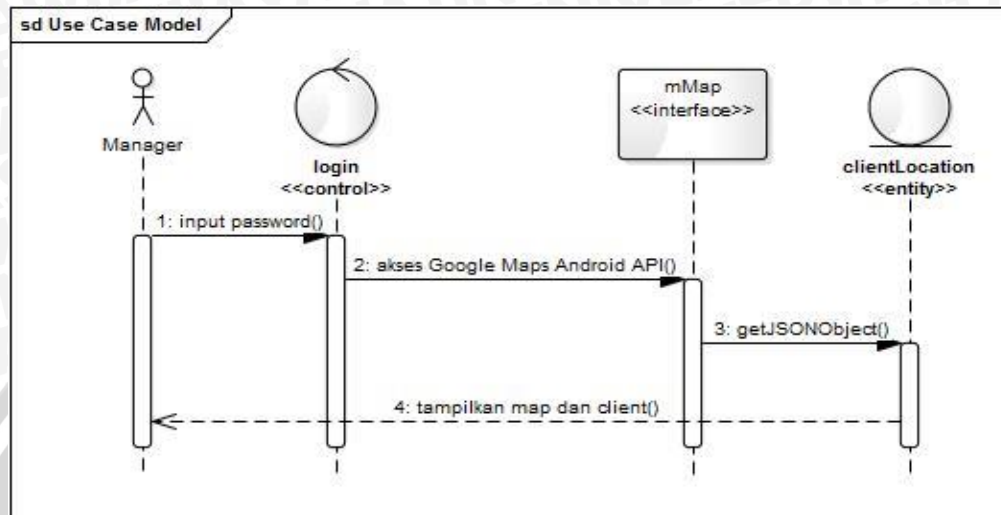


Gambar 4.14 Sequence Diagram Login

Dalam Gambar 4.14 merupakan diagram *sequence login user*. Ketika diaktifkan sistem belum mengetahui identitas dari user sehingga user diwajibkan memilih salah satu dari dua pilihan *user* yang tersedia yaitu *manager* atau

client/petugas. Setelah memilih *user* langkah selanjutnya adalah memasukkan *username* dan *password* pada interfece halaman *login*. Setelah dilakukan penyocokkan *username* dan *password*, sistem akan melanjutkan ke *interface* *UserPage* sesuai dengan identitas yang telah dimasukkan.

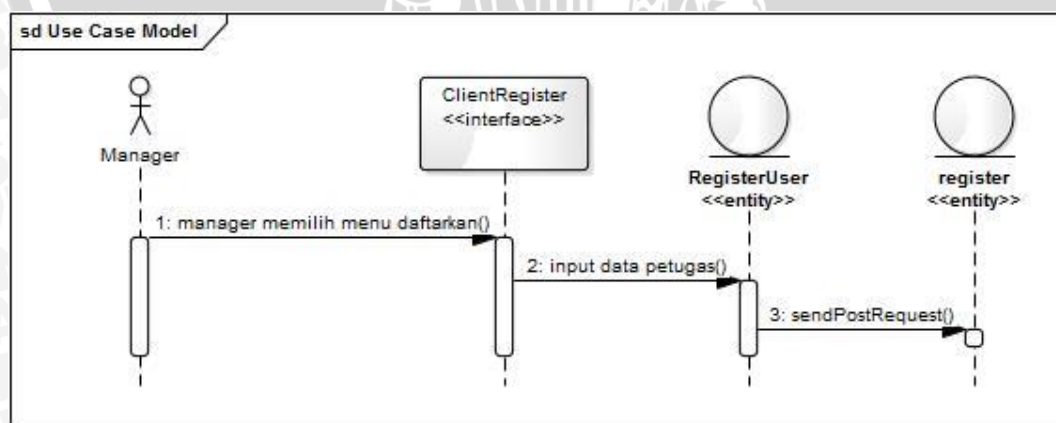
2. Memantau Posisi Kendaraan



Gambar 4.15 Sequence Diagram Memantau Posisi Kendaraan

Dalam Gambar 4.15 merupakan diagram *sequence* untuk melakukan tracking yaitu menampilkan map beserta kendaraan yang di tracking. Pertama manager harus login. Setelah itu sistem akan mengakses google map android API yang akan menampilkan *interface* *mMap* atau Google Map. Kemudian mengambil data petugas menggunakan method *getJSONObject* untuk mengakses *entity* *clientLocation*. Hasil *tracking* akan ditampilkan pada layar manager berupa *map* beserta kendaraan yang di *track*.

3. Tambah Petugas

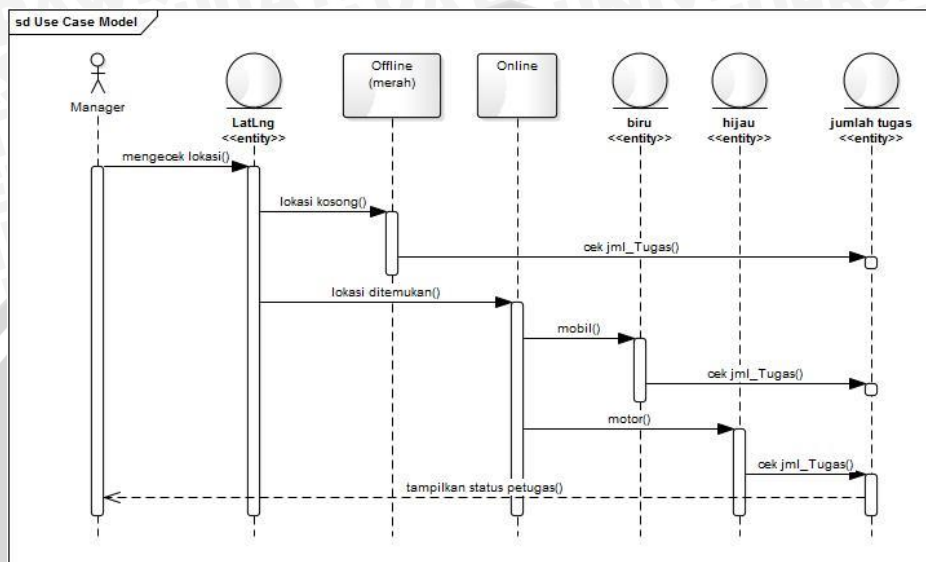


Gambar 4.16 Sequence Diagram Tambah Petugas

Dalam Gambar 4.15 merupakan diagram *sequence* menambah atau mendaftarkan petugas baru yang dilakukan oleh aktor manager. Pertama manager menekan tombol daftarkan pada halaman map yang mengarahkan pada *interface*

ClientRegister. Kemudian masuk kedalam halaman *input* data petugas baru yaitu mengisi data petugas seperti *username*, *password*, nama grup kedalam *entity RegisterUser*. Langkah selanjutnya yaitu mengirim dan menyimpan data tersebut menggunakan method *sendPostRequest* kedalam *entity register* pada database.

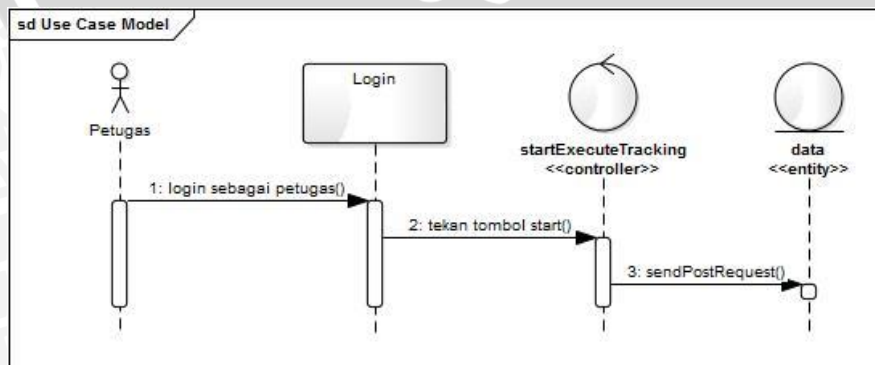
4. Menampilkan Status Petugas



Gambar 4.17 Sequence Diagram Menampilkan Status

Dalam Gambar 4.17 merupakan diagram *sequence* untuk menampilkan status petugas. Jika entity *LatLng* atau koordinat lokasi tidak ditemukan maka sistem akan menampilkan status offline yang ditandai dengan icon berwarna merah kemudian sistem juga menampilkan jumlah tugas terakhir yang telah diselesaikan. Apabila entity *LatLng* ditemukan maka status kendaraan menjadi online warna icon akan menyesuaikan dengan jenis kendaraan yang sedang online yaitu berwarna biru apabila entity kendaraan berjenis mobil dan warna hijau apabila entity jenis kendaraan adalah motor. Kemudian sistem juga akan menampilkan jumlah tugas yang telah diselesaikan.

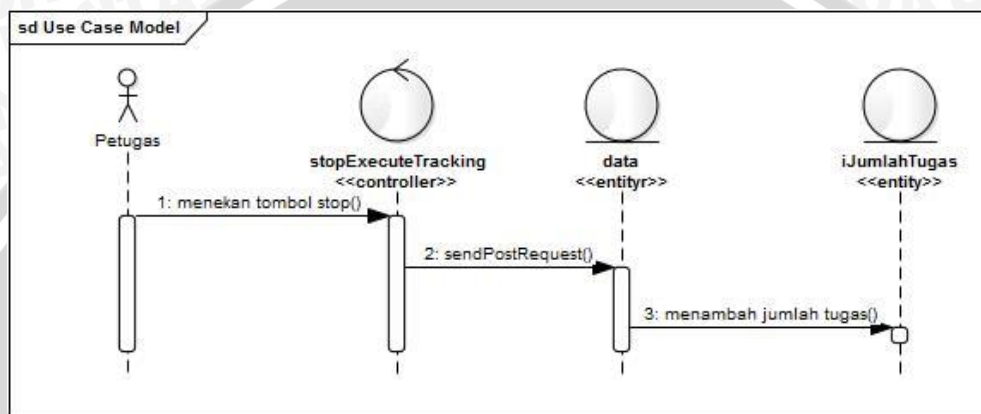
5. Melaporkan Kondisi Siap Berangkat (Start)



Gambar 4.18 Sequence Diagram Melaporkan Kondisi Siap Berangkat (Start)

Dalam Gambar 4.16 merupakan diagram *sequence* melaporkan kondisi siap berangkat (*start*). Pertama petugas harus login sebagai client dengan memasukkan *username*, *password* dan jenis kendaraan. Setelah berhasil *login* sistem akan menampilkan *ClientPage*. Pada halaman *ClientPage* terdapat *controller startExecuteTracking* yang berupa tombol *start*. Setelah tombol *start* di tekan maka sistem akan mengirim data petugas berupa *username*, grup, jumlah tugas, status, dan koordinat menggunakan *method* *sendPostRequest* menuju *entity URL_StartTracking* yaitu alamat database server.

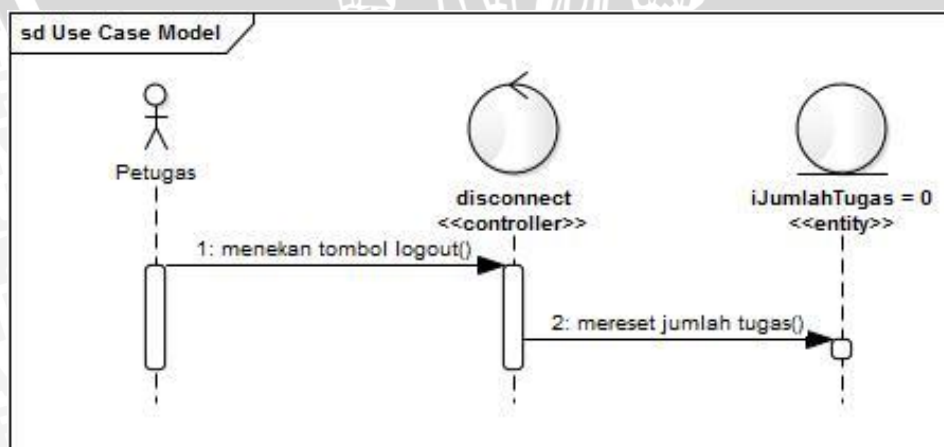
6. Melaporkan Tugas Telah Selesai (*Stop*)



Gambar 4.19 Sequence Diagram Melaporkan Tugas Telah Selesai (*Stop*)

Pada Gambar 4.17 merupakan diagram *sequence* melaporkan tugas telah selesai (*stop*). Sebelumnya sistem masih dalam kondisi sedang melakukan *tracking* atau *start tracking* masih berjalan. Untuk menghentikan *tracking* petugas harus menekan tombol *stop tracking* yang berwarna merah pada perangkat mobile nya sehingga menjalankan *controller stopExecuteTracking*. Hal tersebut menjalankan *method* *sendPostRequest* untuk penghentian pengiriman *entity* data petugas. Kemudian sistem akan menambahkan jumlah tugas pada *entity* *iJumlahTugas*.

7. Logout

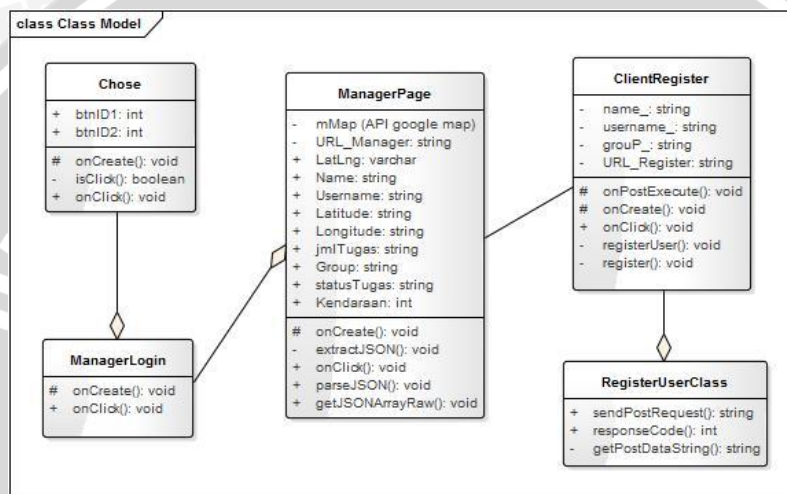


Gambar 4.20 Sequence Diagram Logout

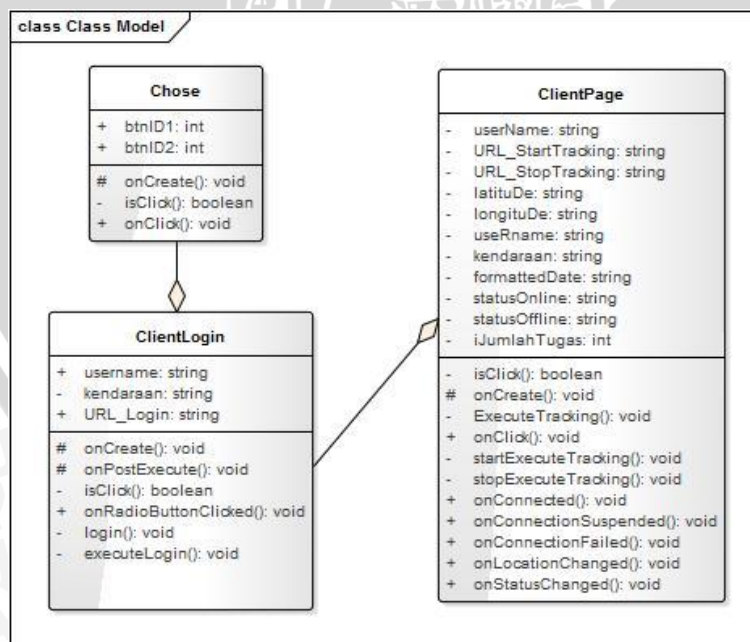
Dalam Gambar 4.18 merupakan diagram *sequence logout*. Setelah petugas menekan tombol *logout* maka *controller disconnect* akan berjalan. Kemudian jumlah tugas yang telah diselesaikan akan di *reset* menjadi nol pada *entity iJumlahTugas = 0*.

4.2.4 Perancangan Class Diagram

Diagram *class* adalah diagram statis yang menampilkan struktur dan deskripsi dari class serta hubungannya antara *class*. Diagram ini terdiri dari nama *class*, atribut, dan operasi/*method*. Berikut ini digambarkan *class diagram* dari rancang bangun prototipe *tracking* mobil pelayanan teknik PLN menggunakan perangkat bergerak berbasis GPS yang bisa dilihat dalam Gambar 4.20 dan 4.21 di bawah ini.



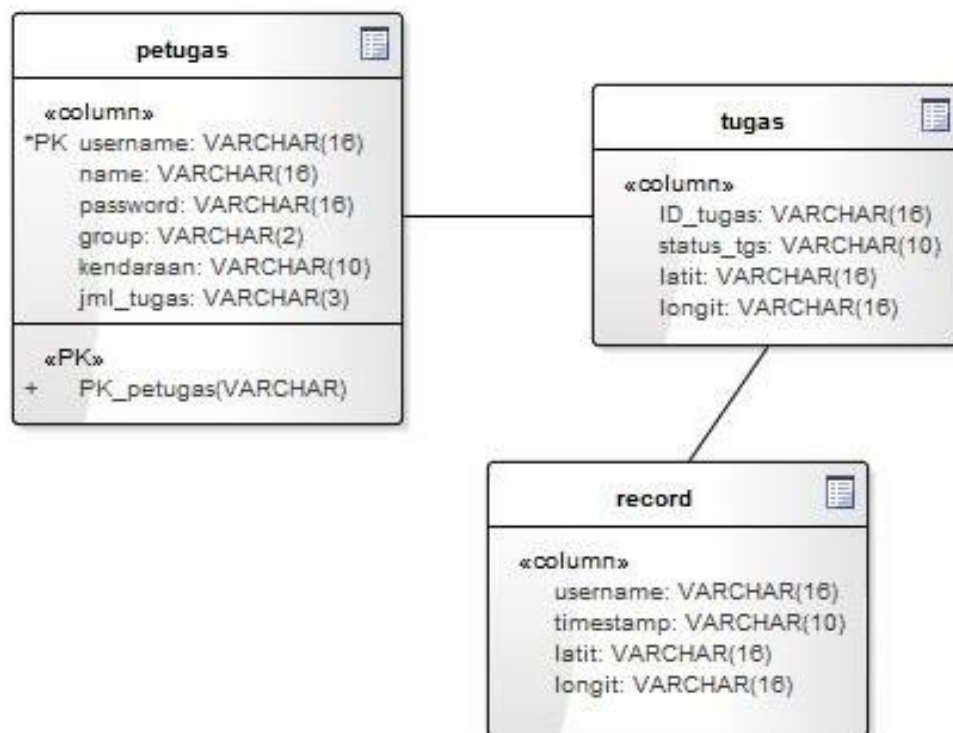
Gambar 4.21 Class Diagram Manager



Gambar 4.22 Class Diagram Petugas

4.2.5 Perancangan Basis Data

Perancangan basis data digunakan untuk menyimpan data-data petugas pada aplikasi *tracking* mobil pelayanan teknik PLN. Perancangan basis data ini direpresentasikan dalam bentuk diagram logikal. Diagram logikal menggambarkan entitas dan relasi dalam *database* yang ditampilkan kedalam bentuk grafis. Berikut skema basis data dari rancang bangun *prototipe tracking* mobil pelayanan teknik PLN menggunakan perangkat bergerak berbasis GPS yang bisa dilihat dalam Gambar 4.22 di bawah ini.



Gambar 4.23 Diagram Logikal Basis Data

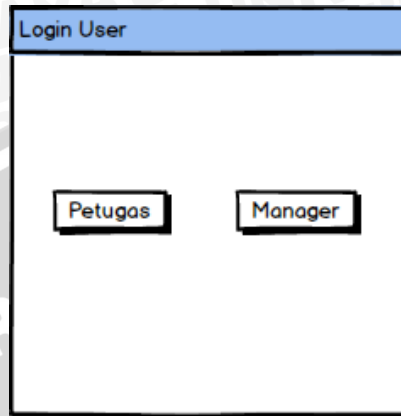
Diagram logikal basis data yang ditunjukkan dalam Gambar 4.22 menunjukkan terjadi komunikasi data antar tabel dalam proses *tracking* kendaraan. Terdapat data petugas yang berfungsi untuk menyimpan data petugas berupa username sebagai *primary key*, name, password, group, kendaraan, jml_tugas. Pada tabel basis data tugas terdapat data tugas berupa ID_tugas, status_tgs, latit, longit. Pada tabel basis data record terdapat data record antara lain berupa username sebagai *foreign key*, timestamp, latit, longit. Semua tipe data berupa VARCHAR.

4.2.6 Perancangan User Interface

Pada tahap ini menjelaskan tentang perancangan *interface* pada rancang bangun *prototipe tracking* mobil pelayanan teknik PLN menggunakan perangkat bergerak berbasis GPS. Berikut gambaran dari masing-masing *interface*

a. Halaman Pilih User

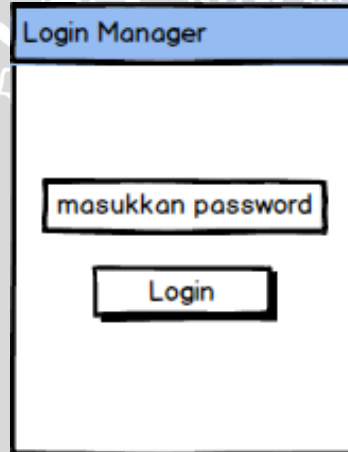
Halaman pilih user digunakan untuk memilih antara Manager dan Petugas. User harus memilih sesuai dengan bidang mereka agar dapat menampilkan halaman login yang sesuai dengan posisi mereka di kantor. Perancangan *interface* halaman pilih user dapat dilihat dalam Gambar 4.23.



Gambar 4.24 Halaman Pilih User

b. Login Manager

Halaman *login* manager digunakan oleh manager untuk melakukan autentifikasi agar dapat masuk kedalam sistem. Perancangan *interface* login manager dapat dilihat dalam Gambar 4.24.



Gambar 4.25 Halaman Login Manager

c. Halaman Manager

Halaman Manager adalah halaman bagi manager untuk melakukan tugasnya sebagai pemantau. Pada halaman ini terdapat map beserta *icon* para petugas pelayanan teknik sesuai dengan posisi dimana mereka berada. Pada halaman ini, terdapat menu bar. Tombol *refresh* digunakan untuk melakukan *refresh* pada map apabila tidak ada perubahan secara otomatis. Tombol next dan prev digunakan untuk melihat satu persatu kendaraan, untuk melihat kendaraan

selanjutnya dapat menekan tombol next dan untuk melihat kendaraan sebelumnya dapat menekan tombol prev. Tombol all berfungsi untuk menampilkan seluruh kendaraan beserta data dari masing-masing kendaraan pada map seperti seperti nama, group, jumlah tugas, status yang telah diselesaikan. Untuk petugas yang menggunakan mobil ditandai dengan icon berwarna biru. Untuk petugas yang menggunakan motor ditandai dengan icon berwarna hijau. Sedangkan petugas yang sedang tidak bertugas atau offline maka ditandai dengan *icon* berwarna merah. Tombol daftar digunakan oleh manager untuk mendaftarkan petugas baru. Perancangan *interface* halaman manager dapat dilihat dalam Gambar 4.25.



Gambar 4.26 Halaman Manager

d. Halaman *Register*

Halaman register digunakan oleh aktor manager untuk mendaftarkan petugas baru. Manager memasukkan nama, group, username, password petugas baru. Setelah menekan tombol *register* maka data petugas baru akan dikirim menuju *database*. Manager menginformasikan *username*, *password*, dan group kepada petugas untuk melakukan *login* agar petugas dapat masuk kedalam sistem. Perancangan *interface* halaman *register* dapat dilihat dalam Gambar 4.26.

Gambar 4.27 Halaman Register

e. *Login Petugas*

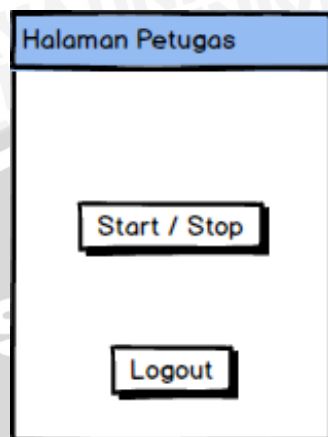
Halaman login manager digunakan oleh petugas untuk melakukan autentifikasi agar dapat masuk kedalam sistem. Perancangan *interface login* petugas dapat dilihat dalam Gambar 4.27.

Gambar 4.28 Login Petugas

f. *Halaman Petugas*

Halaman petugas adalah tempat bagi petugas untuk memberikan informasi kepada manager mengenai kondisi siap berangkat atau menginformasikan apabila tugas telah selesai dikerjakan. Terdapat dua tombol yaitu tombol *start/stop* dan tombol *logout*. Apabila tombol start ditekan maka sistem akan mengirim data berupa data petugas termasuk koordinat kepada server kemudian diarahkan

menuju smartphone manager sehingga manager dapat mengetahui informasi keberadaan petugas (*tracking*). Tombol start yang ditekan akan berganti menjadi tombol stop. Ketika tombol stop ditekan maka sistem akan berhenti mengirim data petugas sehingga menjadi pertanda bahwa tugas telah selesai dilaksanakan. Perancangan interface halaman petugas dapat dilihat dalam Gambar 4.28.



Gambar 4.29 Halaman Petugas

BAB 5 IMPLEMENTASI

Pada bab ini akan membahas tentang implementasi dari rancang bangun prototipe *tracking* mobil pelayanan teknik PLN menggunakan perangkat bergerak berbasis GPS. Implementasi dibagi kedalam beberapa tahap yang masing-masing akan menjelaskan mengenai detail dari kebutuhan implementasi.

5.1 Implementasi

Implementasi dibagi kedalam beberapa tahap antara lain yaitu tersusun dari spesifikasi perangkat lunak sistem, batasan-batasan dalam implementasi, implementasi *class* dan *assets* pada *file* program, implementasi kode program, serta implementasi antarmuka aplikasi.

5.1.1 Spesifikasi Perangkat Lunak Sistem

Dalam proses pengembangan aplikasi, terapat beberapa perangkat lunak yang digunakan. Spesifikasi perangkat lunak sistem yang digunakan dapat dilihat dalam Tabel 5.1

Tabel 5.1 Spesifikasi perangkat lunak komputer

Sistem Operasi	Windows 7
<i>Programming Language</i>	Java
<i>Software Development Kit</i>	Java Development Kit 8
<i>Programming Environment</i>	Java Runtime Environment 8
<i>Integrated Development Environment</i>	Android Studio

5.1.2 Batasan Implementasi

Dalam implementasi rancang bangun prototipe *tracking* mobil pelayanan teknik PLN menggunakan perangkat bergerak berbasis GPS terdapat batasan-batasan yang diberikan yaitu sebagai berikut :

1. Aplikasi dirancang dengan dukungan dari Google Map.
2. Jalur koneksi antara manager dengan petugas melalui jaringan internet.
3. Pembuatan antarmuka aplikasi menggunakan bahasa pemrograman XML.
4. Penyimpanan data petugas berada pada web hosting

5.1.3 Implementasi *Class* dan *Asset*

Tabel 5.2 Implementasi *class* dan *assets* dalam bentuk *file*

Folder	Nama <i>file</i>
\\src\\main\\java\\com\\my_project\\vfinderv1	Chose.java
\\src\\main\\java\\com\\my_project\\vfinderv1	ClientLogin.java
\\src\\main\\java\\com\\my_project\\vfinderv1	ClientPage.java
\\src\\main\\java\\com\\my_project\\vfinderv1	ClientRegister.java

\\src\\main\\java\\com\\my_project\\vfinderv1	ManagerLogin.java
\\src\\main\\java\\com\\my_project\\vfinderv1	ManagerPage.java
\\src\\main\\java\\com\\my_project\\vfinderv1	RegisterUserClass.java

5.1.4 Implementasi Kode Program

Implementasi kode program berikut akan menjelaskan mengenai alur utama aplikasi, yaitu proses *tracking*. Perlu diketahui beberapa method yang dicantumkan hanya untuk method yang dirasa cukup penting dan paling berpengaruh dalam sistem.

5.1.4.1 Implementasi Alur Proses Login

Proses *login* dilakukan untuk melakukan autentifikasi dan mengidentifikasi user agar user dapat masuk kedalam sistem sesuai dengan jabatannya. Potongan *source code* dari proses ini ditunjukkan dalam Kode 5.1.

```

1      public void onClick(View v) {
2          switch(v.getId()){
3              case (R.id.btnClient):
4                  isClick = !isClick;
5                  int btnID1 = isClick ? R.drawable.normbutton :
6                  R.drawable.clickbutton;
7                  btnClient.setBackgroundResource(btnID1);
8                  isClick = false;
9                  startActivity(new Intent(this,
10                  ClientLogin.class));
11
12                  break;
13
14              case (R.id.btnManager):
15                  isClick = !isClick;
16                  int btnID2 = isClick ? R.drawable.normbutton :
17                  R.drawable.clickbutton;
18                  btnManager.setBackgroundResource(btnID2);
19                  isClick = false;
20                  startActivity(new Intent(this,
21                  ManagerLogin.class));
22
23                  break;
24          }
25      }
26  }
```

Kode 5.1 Implementasi Source Code Alur Proses Login

Berikut ini merupakan penjelasan dari source code yang terdapat pada Kode 5.1:

1. Baris 1-5 inisiasi tombol.
2. Baris 7-11 merupakan *method* yang dijalankan apabila *guest* memilih *user* petugas.
3. Baris 14-21 merupakan *method* yang dijalankan apabila *guest* memilih *user* manager.

5.1.4.2 Implementasi Alur Proses Tracking Kendaraan

Proses *tracking* kendaraan ini dilakukan oleh manager. Pengiriman data oleh sisi petugas kemudian sisi manager menerima data dan menampilkannya pada

layar *smartphone* sehingga petugas dapat dipantau oleh manager. Hal ini dimaksudkan agar data petugas dapat dilihat oleh manager. Potongan *source code* dari proses ini dapat ditunjukkan dalam Kode 5.2.

```

1 public void parseJSON(int index){
2     try {
3         JSONObject jsonObject = resultArray.getJSONObject(index);
4         String Name = jsonObject.getString(TAG_Name);
5         String Username = jsonObject.getString(TAG_Username);
6         String Latitude = jsonObject.getString(TAG_Latitude);
7         String Longitude = jsonObject.getString(TAG_Longitude);
8         String jmlTugas = jsonObject.getString(TAG_Jml);
9         String Group = jsonObject.getString(TAG_Group);
10        String statusTugas = jsonObject.getString(TAG_Status);
11        String Kendaraan = jsonObject.getString(TAG_Kendaraan);
12
13        if (Latitude.isEmpty() && Longitude.isEmpty()) {
14            Toast.makeText(ManagerPage.this, Name+"\n"+"Lokasi
15Kosong", Toast.LENGTH_SHORT).show();
16        }else{
17            Double Lat = Double.parseDouble(Latitude);
18            Double Long = Double.parseDouble(Longitude);
19
20            Marker marker;

```

Kode 5.2 Implementasi Alur Source Code Proses Tracking Kendaraan

Berikut ini adalah penjelasan dari source code yang terdapat pada Kode 5.2 :

1. Baris 1-11 adalah inisiasi *JSON object*.
2. Baris 13-15 adalah method yang dikerjakan ketika data koordinat tidak ditemukan.
3. Baris 16-20 adalah *method* yang dikerjakan apabila koordinat ditemukan.

5.1.4.3 Implementasi Alur Proses Tambah Petugas Baru

Penambahan *user* baru yang dilakukan oleh manager bertujuan apabila dalam suatu kantor kekurangan armada pelayanan teknik. Hal tersebut dapat ditanggulangi dengan menambah *account* baru untuk digunakan oleh petugas baru agar dapat masuk kedalam sistem.

```

1 protected void onCreate(Bundle savedInstanceState) {
2     super.onCreate(savedInstanceState);
3     setContentView(R.layout.activity_client_register);
4
5     Button btnRegister = (Button) findViewById(R.id.btnRegister);
6     btnRegister.setOnClickListener(this);
7
8     Button btnCancel = (Button) findViewById(R.id.btnCancel);
9     btnCancel.setOnClickListener(this);
10
11     etName = (EditText) findViewById(R.id.etName);
12     etGroup = (EditText) findViewById(R.id.etGroup);
13     etUsername = (EditText) findViewById(R.id.etUsername);
14     etPassword = (EditText) findViewById(R.id.etPassword);
15
16 }
17
18 public void onClick(View v) {
19     switch (v.getId()){
20         case (R.id.btnRegister):
21             registerUser();

```

```

22         break;
23
24         case (R.id.btnCancel):
25             startActivity(new Intent(this, ManagerPage.class));
26             break;
27     }
28 }
29
30 private void registerUser(){
31     String name = etName.getText().toString().trim();
32     String group =
33     etGroup.getText().toString().trim().toUpperCase();
34     String username = etUsername.getText().toString().trim();
35     String password = etPassword.getText().toString().trim();
36
37     register(name, group, username, password);
38 }
39
40 private void register(String name, String group, String username,
41 String password){
42     class RegisterUser extends AsyncTask<String, Void, String>{
43         ProgressDialog loading;
44         RegisterUserClass registerUserClass = new
45 RegisterUserClass();
46
47         @Override
48         protected void onPreExecute() {
49             super.onPreExecute();
50             loading =
51 ProgressDialog.show(ClientRegister.this, "Please wait!!!", null,
52 true, true);
53         }
54
55         @Override
56         protected void onPostExecute(String s) {
57             super.onPostExecute(s);
58             loading.dismiss();
59
60             if(s.equalsIgnoreCase("Registrasi petugas telah
61 berhasil")){
62
63 Toast.makeText(getApplicationContext(), s, Toast.LENGTH_LONG).show()
64 ;
65                 Intent intent = new Intent(ClientRegister.this,
66 ManagerPage.class);
67                 intent.putExtra(name_,
68 etName.getText().toString());
69                 intent.putExtra(group_,
70 etGroup.getText().toString());
71                 intent.putExtra(username_,
72 etUsername.getText().toString());
73
74                 finish();
75                 startActivity(intent);
76             }else {
77
78 Toast.makeText(getApplicationContext(), s, Toast.LENGTH_LONG).show()
79 ;
80                 startActivity(new Intent(ClientRegister.this,
81 ManagerPage.class));
82             }
83         }

```

Kode 5.3 Implementasi Alur Source Code Proses Tambah Petugas Baru

Berikut ini adalah penjelasan dari source code yang terdapat dalam Kode 5.3 :

1. Baris 1-10 adalah inisiasi tombol.
2. Baris 12-15 adalah proses memasukkan data petugas baru.
3. Baris 31-46 adalah proses menyimpan data petugas baru.
4. Baris 48-62 adalah *method* yang dikerjakan untuk mengirim data menuju *database*.

5.1.4.4 Implementasi Alur Proses Menampilkan Status Petugas

Penampilan status petugas dimaksudkan untuk melihat status petugas apakah petugas sedang online atau offline dan juga menampilkan keterangan seperti nama, group, status, jumlah tugas sehingga manager mengetahui data dan dapat menilai kinerja dari petugas.

```

1  clientLocation = new LatLng(Lat,Long) ;
2
3  if (statusTugas.equalsIgnoreCase("userOnline")
4  &&Kendaraan.equalsIgnoreCase("Mobil")){
5      marker = mMap.addMarker(new MarkerOptions()
6          .position(clientLocation)
7          .title(Username)
8          .snippet(Name + ", " + Group + ", " + jmlTugas + ", "
9      )
10
11  .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactor
12  y.HUE_BLUE));
13      CameraPosition cameraPosition = new CameraPosition.Builder()
14          .target(clientLocation)
15          .zoom(17)
16          .bearing(90)
17          .tilt(45)
18          .build();
19
20  mMap.animateCamera(CameraUpdateFactory.newCameraPosition(cameraPos
21  ition));
22      marker.showInfoWindow();
23  }
24  else if(statusTugas.equalsIgnoreCase("userOnline")
25  &&Kendaraan.equalsIgnoreCase("Motor")){
26      marker = mMap.addMarker(new MarkerOptions()
27          .position(clientLocation)
28          .title(Username)
29          .snippet(Name + ", " + Group + ", " + jmlTugas + ", "
30      )
31
32  .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactor
33  y.HUE_GREEN));
34      CameraPosition cameraPosition = new CameraPosition.Builder()
35          .target(clientLocation)
36          .zoom(17)
37          .bearing(90)
38          .tilt(45)
39          .build();
40
41  mMap.animateCamera(CameraUpdateFactory.newCameraPosition(cameraPos
42  ition));
43      marker.showInfoWindow();
44  }
45  else{
46      marker = mMap.addMarker(new MarkerOptions()
47          .position(clientLocation)
48          .title(Username)
49          .snippet("User Offline")
50  }

```

```

51 .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactor
52 y.HUE_RED));
53     CameraPosition cameraPosition = new CameraPosition.Builder()
54         .target(clientLocation)
55         .zoom(17)
56         .bearing(90)
57         .tilt(45)
58         .build();
59
60     mMap.animateCamera(CameraUpdateFactory.newCameraPosition(cameraPos
61 ition));
62     marker.showInfoWindow();
63 }

```

Kode 5.4 Implementasi Alur Source Code Proses Menampilkan Status Petugas

Berikut ini adalah penjelasan dari *source code* yang terdapat pada Kode 5.4 :

1. Baris 1-23 adalah *method* yang dikerjakan apabila petugas menggunakan jenis kendaraan mobil sehingga *icon* kendaraan *online* akan berwarna biru.
2. Baris 24-44 adalah *method* yang dikerjakan apabila petugas menggunakan jenis kendaraan motor sehingga *icon* kendaraan *online* akan berwarna hijau.
3. Baris 45-63 adalah *method* yang dikerjakan apabila petugas sedang *offline* sehingga *icon* kendaraan berwarna merah.

5.1.4.5 Implementasi Alur Proses Melaporkan Kondisi Siap Berangkat (Start)

Melaporkan kondisi siap berangkat adalah keadaan dimana petugas hendak berangkat bertugas jika terjadi kerusakan. Hal ini menandakan dimulainya tugas sehingga petugas menekan tombol *start*.

```

1 private void ExecuteTracking(final String username, final String
2 latit, final String longit, final String status_tgs){    class
3 ExecuteTrackingClass extends AsyncTask<String, Void, String>{
4
5     @Override
6     protected String doInBackground(String... params) {
7         HashMap<String, String> data = new HashMap<>();
8         data.put("username",params[0]);
9         data.put("latit",params[1]);
10        data.put("longit",params[2]);
11        data.put("status_tgs",params[3]);
12
13        RegisterUserClass registerUserClass = new
14 RegisterUserClass();
15        String result =
16 registerUserClass.sendPostRequest(URL_StartTracking, data);
17
18        return result;
19    }
20 }
21 ExecuteTrackingClass executeTrackingClass = new
22 ExecuteTrackingClass();
23 executeTrackingClass.execute(username, latit, longit,
24 status_tgs);
25 }

```

Kode 5.5 Implementasi Alur Source Code Proses Start

Berikut ini adalah penjelasan dari *source code* yang terdapat pada Kode 5.4 :

1. Baris 1-11 adalah inisiasi data petugas atau parsing JSON object.

- Baris 13-25 adalah *method* yang dikerjakan untuk mengirim data petugas menuju server.

5.1.4.6 Implementasi Alur Proses Melaporkan Tugas Telah Selesai (Stop)

Melaporkan tugas selesai dilakukan untuk melaporkan bahwa tugas telah selesai dilaksanakan sehingga menambah jumlah tugas.

```

1 private void ExecuteStop(final String username, final String
2 jml_tugas, final String status_tgs){
3     class ExecuteStopClass extends AsyncTask<String, Void,
4 String>{
5         @Override
6         protected String doInBackground(String... params) {
7             HashMap<String, String> data = new HashMap<>();
8             data.put("username",params[0]);
9             data.put("jml_tugas",params[1]);
10            data.put("status_tgs",params[2]);
11
12            RegisterUserClass registerUserClass = new
13 RegisterUserClass();
14            String result =
15 registerUserClass.sendPostRequest(URL_StopTracking, data);
16
17            return result;
18        }
19    }
20    ExecuteStopClass executeStopClass = new ExecuteStopClass();
21    executeStopClass.execute(username, jml_tugas, status_tgs);
22 }

```

Kode 5.6 Implementasi Alur Source Code Proses Stop

Berikut ini adalah penjelasan dari source code yang terdapat pada Kode 5.4 :

- Baris 1-10 adalah inisiasi data petugas atau parsing JSON object.
- Baris 13-25 adalah *method* yang dikerjakan untuk menghentikan pengiriman data petugas menuju server yaitu URL StopTracking.
- Baris 20-22 adalah fungsi penambahan jumlah tugas.

5.1.4.7 Implementasi Alur Proses Logout

Logout dilakukan ketika terjadi pergantian petugas karena berakhirnya jam kerja atau pergantian piket sehingga petugas keluar dari sistem.

```

1 private void stopExecuteTracking(){
2     googleApiClient.disconnect();
3     iJumlahTugas = 0;
4     String jumlahTugas = String.valueOf(iJumlahTugas).trim();
5     ExecuteStop(userName, jumlahTugas, statusOffline);
6
7     startActivity(new Intent(ClientPage.this, Chose.class));

```

Kode 5.7 Implementasi Alur Source Code Proses Logout

Berikut ini adalah penjelasan dari source code yang terdapat pada Kode 5.4 :

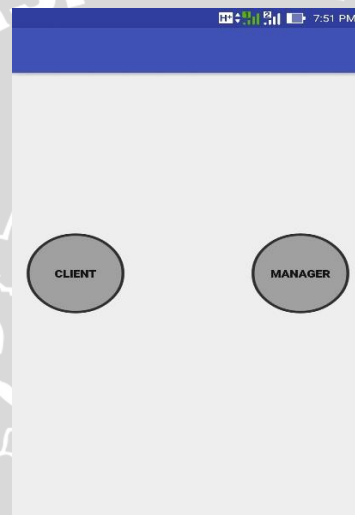
- Baris 1-2 adalah proses *disconnect* dari server.
- Baris 3-4 adalah proses mereset jumlah tugas.
- Baris 5-8 adalah proses menghentikan pengiriman data menuju server.

5.1.5 Implementasi Antarmuka Aplikasi

Pada bagian ini akan menunjukkan hasil implementasi antarmuka pada aplikasi yang terdiri dari halaman login, halaman manager, halaman petugas, dan halaman daftarkan petugas seperti yang terlihat dalam Gambar 5.1 sampai Gambar 5.6.

5.1.5.1 Halaman Pilih User

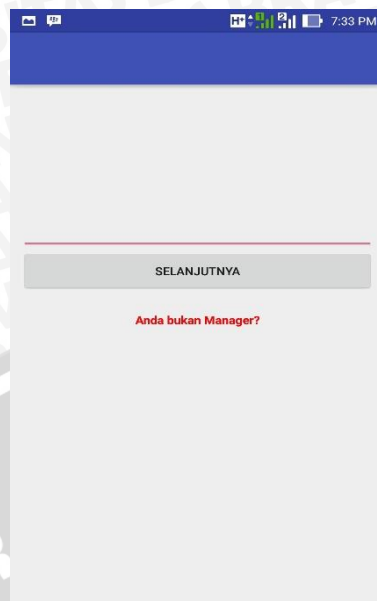
Halaman pilih *user* adalah halaman pertama yang ditampilkan ketika seorang *guest* atau *user* membuka aplikasi. *User* atau *guest* tersebut masih belum diketahui identitasnya oleh sistem sehingga *user* harus memilih salah satu *user* yang tersedia sesuai dengan posisi atau jabatan mereka di kantor. Disini terdapat dua *user* yaitu petugas dan manager. Tampilan halaman pilih *user* dapat dilihat dalam Gambar 5.1.



Gambar 5.1 Pilih User

5.1.5.2 Halaman Login Manager

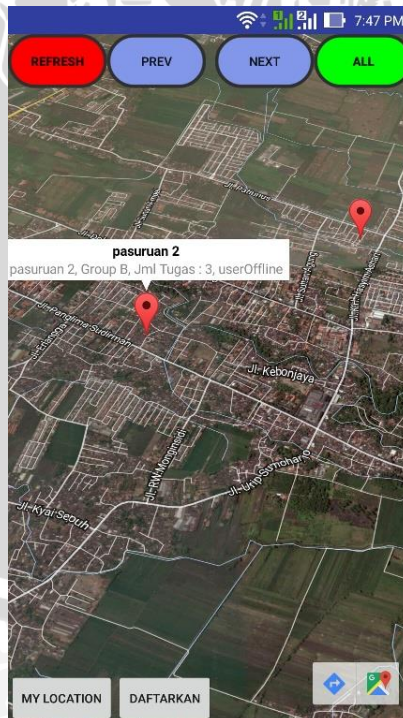
Halaman *login* manager adalah halaman untuk melakukan autentifikasi bagi manager agar dapat masuk dan mengoperasikan sistem. Manager hanya perlu memasukkan *password* karena dalam satu kantor terdapat satu manager teknik. Halaman *login* manager dapat dilihat dalam Gambar 5.2.



Gambar 5.2 Login Manager

5.1.5.3 Halaman Manager

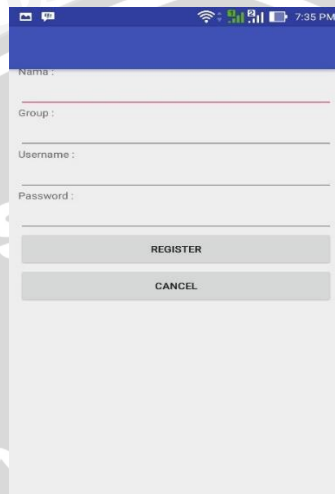
Pada halaman manager terdapat icon-icon kendaraan petugas yang di *track* pada tampilan *map*. Diatas *icon* terdapat informasi mengenai kendaraan petugas seperti nama, *group*, dan jumlah tugas. Halaman Manager dapat dilihat dalam Gambar 5.3.



Gambar 5.3 Halaman Manager

5.1.5.4 Halaman Daftarkan Petugas Baru

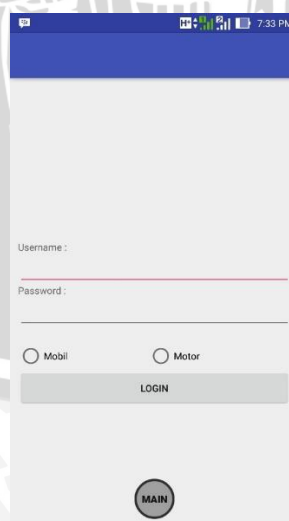
Halaman daftarkan petugas baru adalah halaman yang ditampilkan ketika manager menekan tombol daftarkan pada halaman manager. Halaman ini digunakan untuk mendaftarkan *acount* dengan memasukkan data petugas baru kedalam *database* sehingga petugas baru dapat menggunakan *acount* tersebut untuk masuk kedalam sistem. Halaman daftarkan petugas baru dapat dilihat dalam Gambar 5.4.



Gambar 5.4 Halaman Daftarkan

5.1.5.5 Halaman Login Petugas

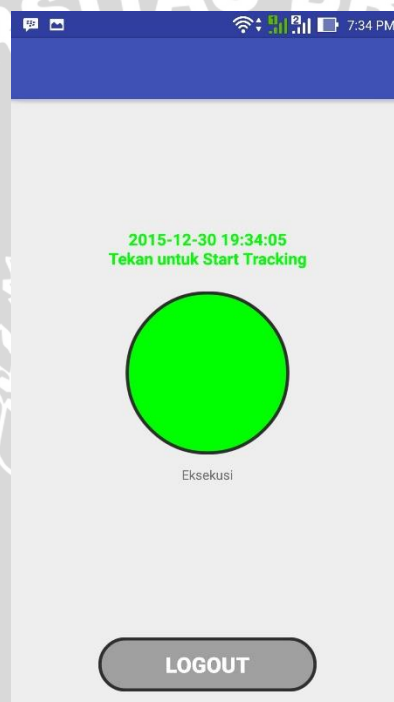
Halaman login petugas adalah halaman untuk melakukan autentifikasi bagi petugas agar dapat masuk dan mengoperasikan sistem. Petugas diwajibkan memasukkan *username* dan *password* kemudian memilih jenis kendaraan sesuai dengan yang dikendarainya. Halaman *login* petugas dapat dilihat dalam Gambar 5.5.



Gambar 5.5 Login Petugas

5.1.5.6 Halaman Petugas

Halaman petugas ditampilkan ketika petugas berhasil *login* kedalam sistem. Pada halaman petugas terdapat tombol berwarna hijau yang berfungsi untuk memulai *tracking*. Setelah tombol *start* ditekan maka sistem akan mengirimkan data berupa nama, group, koordinat secara terus menerus sampai petugas menekan tombol *stop*. Tombol *start* akan berubah menjadi tombol *stop* yang berwarna merah setelah petugas menekan tombol *start*. Pada halaman petugas terdapat tombol *logout* yang berfungsi untuk keluar dari sistem dan kemudian mereset data jumlah tugas menjadi nol. Hal ini terjadi pada saat terdapat pergantian jam kerja atau jam piket antar petugas. Halaman petugas dapat dilihat dalam Gambar 5.6.



Gambar 5.6 Halaman Petugas

BAB 6 PENGUJIAN

Pada bab ini menjelaskan tentang pengujian dan analisis yang akan dilakukan pada aplikasi yang telah dibangun. Pengujian yang dilakukan adalah pengujian unit, pengujian validasi dan pengujian akurasi serta melakukan analisis terhadap hasil dari pengujian yang telah dilakukan untuk mendapatkan kesimpulan.

6.1 Pengujian Validasi

Pengujian validasi dilakukan untuk mengetahui apakah sistem dibangun sudah berjalan seperti yang diharapkan. Pengujian dilakukan berdasarkan skenario *use case* dan menggunakan metode *black-box testing*, karena tidak diperlukan konsentrasi terhadap alur jalannya program dan lebih ditekankan untuk menemukan kesesuaian antara kinerja sistem dengan daftar kebutuhan. Untuk mengetahui kesesuaian antara daftar kebutuhan dengan kinerja sistem, maka pada kebutuhan fungsional utama sistem akan dilakukan pengujian dengan kasus uji masing-masing yang sudah ditentukan.

6.1.1 Kasus Uji Login

Tabel 6.1 merupakan kasus uji terhadap kebutuhan fungsional KF01. Kasus uji ini ditujukan untuk memastikan bahwa aplikasi dapat melakukan autentifikasi untuk mengidentifikasi user yang akan menggunakan aplikasi.

Tabel 6.1 Kasus Uji Login

Nama kasus uji	Login
Objek uji	Kebutuhan Fungsional (KF01)
Tujuan pengujian	Mengidentifikasi user.
Prosedur uji	1. Membuka aplikasi 2. Memilih salah satu user sesuai dengan jabatan
Hasil yang diharapkan	Aplikasi dapat menampilkan halaman user sesuai dengan user yang telah login

6.1.2 Kasus Uji Tracking Kendaraan

Tabel 6.2 merupakan kasus uji terhadap kebutuhan fungsional KF02. Kasus uji ini ditujukan untuk memastikan bahwa aplikasi dapat melakukan tracking sehingga menampilkan *icon* kendaraan petugas pada map.

Tabel 6.2 Kasus Uji Tracking Kendaraan

Nama kasus uji	Tracking
Objek uji	Kebutuhan Fungsional (KF02)
Tujuan pengujian	Memastikan bahwa aplikasi dapat melakukan tracking terhadap kendaraan petugas pelayanan teknik di lapangan.

Prosedur uji	1. Membuka aplikasi 2. Memilih user manager 3. Login sebagai manager
Hasil yang diharapkan	Aplikasi akan menampilkan icon kendaraan petugas pelayanan teknik pada <i>map</i>

6.1.3 Kasus Uji Tambah Petugas

Tabel 6.3 merupakan kasus pengujian yang dilakukan terhadap kebutuhan fungsional KF03. Kasus uji ini dilakukan untuk mengetahui apakah aplikasi dapat menambahkan user baru.

Tabel 6.3 Kasus Uji Tambah Petugas

Nama kasus uji	Tambah Petugas
Objek uji	Kebutuhan Fungsional (KF03)
Tujuan pengujian	Memastikan bahwa aplikasi dapat penambahan petugas baru.
Prosedur uji	1. Membuka aplikasi 2. <i>Login</i> sebagai manager 3. Memilih menu daftarkan
Hasil yang diharapkan	Aplikasi dapat menambah <i>account</i> petugas baru

6.1.4 Kasus Uji Status Petugas

Tabel 6.4 merupakan kasus pengujian yang dilakukan terhadap kebutuhan fungsional KF04. Kasus uji ini dilakukan untuk melihat perubahan pada status petugas.

Tabel 6.4 Kasus Uji Status Petugas

Nama kasus uji	Status Petugas
Objek uji	Kebutuhan Fungsional (KF04)
Tujuan pengujian	Memastikan bahwa aplikasi dapat menampilkan status petugas.
Prosedur uji	1. Membuka aplikasi 2. <i>Login</i> sebagai manager 3. Melihat status petugas pada <i>icon</i> kendaraan
Hasil yang diharapkan	Aplikasi dapat menunjukkan status petugas sesuai dengan kondisi yang terjadi (offline/online).

6.1.5 Kasus Uji Melaporkan Kondisi Siap Berangkat (*Start*)

Tabel 6.5 merupakan kasus pengujian yang dilakukan terhadap kebutuhan fungsional KF05. Kasus uji ini dilakukan untuk mengetahui apakah data petugas dapat dikirim menuju server.

Tabel 6.5 Kasus Uji Pengiriman Data Menuju Server oleh Petugas

Nama kasus uji	Melaporkan Kondisi Siap Berangkat (<i>Start</i>)
Objek uji	Kebutuhan Fungsional (KF05)
Tujuan pengujian	Memastikan bahwa aplikasi dapat mulai mengirim data petugas berupa nama, group, jenis kendaraan, koordinat, status petugas.
Prosedur uji	1. Membuka aplikasi 2. Login sebagai petugas 3. Menekan tombol start
Hasil yang diharapkan	Aplikasi dapat mengirim data petugas menuju server

6.1.6 Kasus Uji Melaporkan Tugas Telah Selesai (*Stop*)

Tabel 6.6 merupakan kasus pengujian yang dilakukan terhadap kebutuhan fungsional KF06. Kasus uji ini dilakukan untuk mengetahui apakah aplikasi dapat menghentikan pengiriman data menuju server setelah tugas telah selesai dilaksanakan.

Tabel 6.6 Kasus Uji Melaporkan Tugas Telah Selesai

Nama kasus uji	Melaporkan Tugas Telah Selesai (<i>Stop</i>)
Objek uji	Kebutuhan Fungsional (KF06)
Tujuan pengujian	Memastikan bahwa aplikasi dapat mulai menghentikan pengiriman data menuju server.
Prosedur uji	1. Membuka aplikasi 2. Login sebagai petugas 3. Menekan tombol start 4. Menekan tombol stop
Hasil yang diharapkan	Aplikasi dapat menghentikan pengiriman data menuju server.

6.1.7 Kasus Uji *Logout*

Tabel 6.7 merupakan kasus pengujian yang dilakukan terhadap kebutuhan fungsional KF07. Kasus uji ini dilakukan untuk mengetahui apakah user dapat keluar dari sistem.

Tabel 6.7 Kasus Uji Logout

Nama kasus uji	Login
Objek uji	Kebutuhan Fungsional (KF07)
Tujuan pengujian	Memastikan bahwa user dapat keluar dari sistem.
Prosedur uji	1. Membuka aplikasi 2. Login sebagai petugas 3. Menekan tombol logout
Hasil yang diharapkan	Aplikasi dapat menghentikan pengiriman data menuju server.

6.1.8 Hasil Pengujian Validasi

Seperti yang terlihat pada Tabel 6.8, hasil pengujian validasi berdasarkan skenario *use case* menunjukkan bahwa kedua kasus uji tersebut menghasilkan nilai valid.

Tabel 6.8 Hasil Pengujian Validasi

No.	Nama Kasus Uji	Hasil Yang Diharapkan	Hasil Yang Didapatkan	Validitas
1.	Login	User dapat masuk kedalam sistem	Menampilkan halaman user sesuai dengan jabatannya	Valid
2.	Tracking	Aplikasi menampilkan icon kendaraan petugas pada map	Aplikasi berhasil menampilkan icon kendaraan petugas lapangan pada map	Valid
3.	Tambah Petugas	Sistem dapat menambahkan account petugas baru	Sistem berhasil menambahkan account petugas baru	Valid
4.	Status Petugas	Sistem menampilkan status petugas pada icon yang ada di map	Sistem berhasil menampilkan status petugas. Hasil yang didapat ketika petugas yang sedang online dengan jenis kendaraan mobil ditandai dengan icon berwarna biru untuk jenis kendaraan motor ditandai dengan icon berwarna hijau dan user yang sedang	Valid

			offline ditandai dengan warna merah	
5.	Melaporkan kondisi siap berangkat (start)	Aplikasi pada sisi petugas dapat mengirim data menuju server	Aplikasi berhasil mengirim data menuju server sehingga manager dapat mengetahui nama, lokasi, group dan jumlah tugas yang telah diselesaikan oleh petugas	Valid
6.	Melaporkan tugas telah selesai (stop)	Aplikasi menghentikan pengiriman data menuju server	Aplikasi berhasil menghentikan pengiriman data. Hal ini ditunjukkan dengan bertambahnya jumlah tugas dan berubahnya status petugas menjadi offline dan	Valid
7.	Logout	Petugas dapat keluar dari sistem	Petugas berhasil keluar dari sistem. Hal ini ditandai dengan perubahan jumlah tugas menjadi 0 atau sistem mereset jumlah petugas menjadi 0.	Valid

6.2 Pengujian Akurasi

Pada tahapan ini akan dilakukan pengujian akurasi dengan cara membandingkan koordinat kendaraan pada aplikasi dengan koordinat kendaraan pada Google Map. Metode yang digunakan untuk menghitung selisih koordinat pada kedua aplikasi tersebut adalah metode *Euclidean* seperti yang dapat dilihat pada Persamaan 6.1.

$$Euclidean Distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \times 111,325 \quad (6.1)$$

x1 : kordinat latitude titik 1

x2 : kordinat latitude titik 2

y1 : kordinat longitude titik 1

y2 : kordinat longitude titik 2

Hasil dari perhitungan dikalikan dengan 111.325 untuk mendapatkan dalam satuan kilometer. Nilai 111.325 diperoleh dari konversi 1 derajat bumi kedalam kilometer. (Anthony P. Kirvan., 1997)

Tabel 6.9 Hasil Pengujian Akurasi

Lokasi	Koordinat Aplikasi		Koordinat Google Map		Hasil Selisih Jarak
	Latitude	Longitude	Latitude	Longitude	
Papa Kuning No 1	-7,9436184	112,6284853	-7,9436240	112,6284359	5,534 meter
Jl. Borobudur No. 80	-7,9387896	112,6341084	-7,9388370	112,6341219	5,486 meter
Masjid FILKOM Universitas Brawijaya	-7,954563	112,6147441	-7,9545583	112,6147580	1,633 meter
Jl. Candi Mendut (di depan indomaret)	-7,9444151	112,6344408	-7,9444509	112,6344518	4,169 meter
Jl. Kalpataru (di depan BRI)	-7,9477551	112,6250467	-7,9476907	112,6250949	8,955 meter
Nilai rata-rata					5,1554 meter

Dengan menggunakan persamaan *Euclidean* maka didapatkan perhitungan pada lokasi 1 sebagai berikut :

$$n = \sqrt{(-7,9436240 - (-7,9436184))^2 + (112,6284359 - 112,6284853)^2} \times 111.319$$

$$n = 0,005534 \text{ kilometer}$$

$$n = 5,534 \text{ meter}$$

Dengan menggunakan cara perhitungan yang sama maka didapatkan hasil selisih jarak pada 4 lokasi yang lainnya seperti yang ditampilkan dalam Tabel 6.9

6.3 Analisis

Proses analisis bertujuan untuk mendapatkan kesimpulan dari hasil pengujian yang telah dilakukan sebelumnya. Proses analisis yang dilakukan adalah analisis pada pengujian unit, validasi dan akurasi.

6.3.1 Hasil Pengujian Validasi

Proses analisis pengujian validasi mengacu pada kesesuaian kinerja sistem dengan kebutuhan fungsionalitas sistem. Berdasarkan hasil pengujian validasi dapat disimpulkan bahwa aplikasi telah memenuhi kebutuhan yang diuraikan pada tahap analisis kebutuhan, sebab hasil dari semua pengujian validasi ini menghasilkan status valid.

6.3.2 Hasil Pengujian Akurasi

Proses analisis pengujian akurasi dilakukan dengan cara membandingkan koordinat dari aplikasi dengan koordinat Google Map pada 5 lokasi yang berbeda. Perhitungan jarak antara 2 koordinat dilakukan dengan menggunakan metode Euclidean. Jarak selisih koordinat pada 5 lokasi yang berbeda dapat dilihat dalam Tabel 6.10

Tabel 6.10 Analisis Pengujian Akurasi

No	Lokasi	Hasil Selisih Jarak
1	Papa Kuning No 1	5,534 meter
2	Jl. Borobudur No. 80	5,486 meter
3	Masjid FILKOM Universitas Brawijaya	1,633 meter
4	Jl. Candi Mendut (di depan indomaret)	4,169 meter
5	Jl. Kalpataru (di depan BRI)	8,955 meter
Nilai rata-rata		5,1554 meter

Dari Tabel 6.10 didapatkan nilai hasil rata-rata jarak antara aplikasi dengan Google Map yaitu 5,1554 meter. Dari hasil pengujian dapat disimpulkan bahwa aplikasi memiliki tingkat keakuratan hingga 5,1554 meter sehingga manager dapat memastikan letak keberadaan kendaraan pelayanan teknik di lapangan dengan baik.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan hasil penelitian skripsi yang telah dilakukan maka dapat ditarik kesimpulan sebagai berikut:

1. Aplikasi disusun dengan konsep arsitektur *client-server* yang terdiri dari dua aktor yaitu manager sebagai sisi pemantau dan petugas sebagai sisi yang dipantau. Perancangan dilakukan dengan menggunakan bahasa pemrograman Java yang ditulis dengan IDE Android Studio. Aplikasi beroperasi dengan memanfaatkan API Google Map. Untuk penyimpanan dan pertukaran data menggunakan web hosting dan untuk transfer data menggunakan koneksi internet yang berada pada *smartphone*.
2. Berdasarkan hasil pengujian validasi, aplikasi berhasil melakukan *tracking* atau *monitoring* dengan cara sisi petugas mengirim data berupa nama, group, koordinat, jumlah tugas menuju server. Kemudian aplikasi pada sisi manager sebagai pemantau menerima data petugas dari server sehingga dapat menampilkan icon petugas pada *map* sesuai dengan lokasi petugas yang sebenarnya.
3. Berdasarkan hasil pengujian akurasi yang dilakukan dengan cara membandingkan selisih jarak antara koordinat yang dihasilkan oleh layanan Google Map dengan koordinat yang dihasilkan oleh aplikasi rancangan, menunjukkan bahwa aplikasi memiliki tingkat keakuratan hingga 5,1554 meter berdasarkan nilai rata-rata yang diperoleh dari hasil perhitungan jarak antar koordinat dari 5 lokasi yang berbeda.

7.2 Saran

Saran yang dapat diberikan untuk pengembangan selanjutnya dari aplikasi ini adalah:

1. Untuk pengembangan lebih lanjut, penggunaan koneksi yang baik akan membantu proses *tracking* sehingga mendapatkan hasil yang *real time*.
2. Pengembangan antarmuka aplikasi yang lebih menarik sehingga dapat menjadi daya tarik serta kenyamanan saat menggunakannya.
3. Aplikasi ini dapat diimplementasikan ke *platform* lainnya.

DAFTAR PUSTAKA

- Gopi, Satheesh. 2005. *Global Positioning System: Principles And Applications*. [online] Tersedia di: <<https://books.google.co.id>> [Diakses 30 April 2015].
- Bar-Shalom, Yaakov, X Rong Li dan Thiagalingam Kirubarajan. 2004. *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. [online] Tersedia di: <<https://books.google.co.id>> [Diakses 30 April 2015]
- Noureldin, Aboelmagd, Tashfeen B. K dan Jacques Georgy. 2012. *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration*. [online] Tersedia di: <<https://books.google.co.id>> [Diakses 4 Mei 2015].
- Challa, Subhash, Mark R. M., Darko M. dan Robin J. E. 2011. *Fundamentals of Object Tracking*. [online] tersedia di : <<https://books.google.co.id>> [Diakses tanggal 6 Mei 2015].
- Ir. Kristanto, Harianto. 1993. *Konsep & Perancangan Database*. [online] Tersedia di: <<http://developer.android.com>> [Diakses 6 Mei 2015].
- Utami, Ema dan Sukrisno. 2005. *Konsep Dasar Pengolahan Dan Pemrograman Database Dengan Sql Server, Ms.Access, Dan Ms.Visual Basic*. [online] Tersedia di: <<http://developer.android.com>> [Diakses 6 Mei 2015].
- El-Rabbany, Ahmed. 2002. *Introduction to GPS: The Global Positioning System*. [online] Tersedia di: <<http://developer.android.com>> [Diakses 21 Mei 2015]
- Kirvan, Anthony P. 1997. Unit 014 - Latitude and Longitude. [online] Tersedia di: <<http://www.ncgia.ucsb.edu/units/u014/u014.html>> [Diakses 06 Januari 2016].

