

**APLIKASI LBS (*LOCATION BASE SERVICES*) BTS (*BASE  
TOWER STATION*) CV.2K DENGAN MENERAPKAN  
ALGORITMA *FLOYD-WARSHALL***

**TUGAS AKHIR**

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

oleh :

**MUHAMMAD FADHLI**

**10751000221**



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

# **APLIKASI LBS (LOCATION BASE SERVICES) BTS (*BASE TOWER STATION*) CV.2K DENGAN MENERAPKAN ALGORITMA *FLOYD-WARSHALL***

**MUHAMMAD FADHLI**

**10751000221**

Jurusan Teknik Informatika

Fakultas Sains dan Teknologi

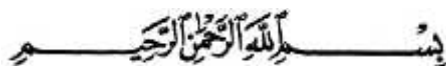
Universitas Islam Negeri Sultan Syarif Kasim Riau

## **ABSTRAK**

Layanan informasi berbasis lokasi memiliki peran penting untuk membantu manusia menyelesaikan suatu masalah. Pencarian suatu lokasi dengan rute terpendek merupakan suatu cara untuk membantu pengguna dalam mendapatkan jalur terdekat. Salah satu algoritma yang sering digunakan dalam pencarian rute terpendek adalah algoritma *Floyd-Warshall*. Algoritma ini melakukan pemecahan masalah dengan melihat solusi akhir yang akan diperoleh sebagai suatu keputusan saling terkait. Selain itu *Location Based Service* (LBS) merupakan gagasan yang timbul dari perkembangan teknologi *mobile*, termasuk didalamnya teknologi *Global Positioning System* (GPS), Sistem informasi Geografis (SIG) serta perkembangan sistem operasi *mobile*. *Location Based Service* (LBS) *Base Tower Station* (BTS) adalah aplikasi yang menerapkan algoritma pencarian rute terpendek yang digabungkan dengan layanan berbasis lokasi untuk membantu teknisi CV.2K dalam menentukan rute terpendek menuju lokasi BTS CV.2K khususnya di kota Pekanbaru. Aktifitas tersebut antara lain menampilkan lokasi-lokasi di peta, serta melihat rute perjalanan terpendek dan informasi mengenai jarak menuju lokasi tujuan. Aplikasi LBS BTS dibangun dengan menerapkan algoritma *Floyd-Warshall* untuk mencari rute terpendek perjalanan, yang berjalan pada *platform Android* menggunakan bahasa pemrograman *Java*. Aplikasi ini adalah aplikasi *client-server* dan membutuhkan koneksi internet serta sistem GPS untuk memaksimalkan penggunaan fitur yang ada. Pengujian aplikasi LBS BTS menggunakan model *Black Box*. Hasil pengujian memberikan kesimpulan bahwa seluruh fitur aplikasi dan algoritma pencarian rute terpendek (*Floyd-Warshall*) dapat berjalan dengan baik, meskipun memiliki kekurangan, seperti lamanya waktu untuk melakukan pencarian rute terpendek.

**Kata Kunci:** Android, *Base Tower Station* (BTS), *Floyd-Warshall*, *Location Based Service* (LBS),

## KATA PENGANTAR



*Alhamdulillah Robbil' alamin*, penulis bersyukur ke-hadirat Allah SWT, karena atas segala limpahan rahmat dan karunia-Nya yang diberikan sehingga penulis dapat menyelesaikan penelitian dan penulisan laporan tugas akhir ini. *Allahumma sholli'ala Muhammad wa'ala ali sayyidina Muhammad*, yang tidak lupa penulis haturkan juga untuk Rosul Allah, Muhammad SAW.

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

1. Prof. Dr. H. M. Nazir, selaku Rektor Universitas Islam Negeri Sultan Syarif Kasim Riau.
2. Dra. Yenita Morena, M.Si. selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau.
3. Elin Haerani, M.Kom, selaku Ketua Jurusan Teknik Informatika, Fakultas Sains dan Teknologi.
4. Nazruddin Safaat H, MT, selaku dosen pembimbing tugas akhir. Terimakasih pak untuk waktu yang selalu Bapak luangkan untuk penulis, ilmu, semangat, dan motivasinya yang luar biasa. Terimakasih banyak Pak.
5. Jasril, S.Si, M.Sc, selaku dosen penguji 1 yang banyak membantu dan memberi masukan penulis dalam penyempurnaan Laporan Tugas Akhir ini, untuk ilmu-ilmunya saya ucapkan terima kasih.

6. Iwan Iskandar, MT, selaku dosen penguji 2, terimakasih pak untuk ilmu-ilmunya, saran-sarannya, perbaikan-perbaikannya, dan masukannya.
7. Muhammad Affandes, ST, MT, sebagai koordinator tugas akhir yang telah memberi masukan-masukan untuk penyelesaian tugas akhir ini, dan sangat sabar membantu penulis dalam mempersiapkan semua kebutuhan penulis dalam penyelesaian Tugas Akhir ini.
8. Ibu beserta kakak dan adik penulis yang menjadi sumber semangat penulis, atas segenap do'a yang tiada hentinya dan dukungan mereka selama masa Tugas Akhir ini.
9. Sahabat penulis semua teman-teman lokal D angkatan 2007, terima kasih atas saran dan bantuannya serta semangat yang diberikan selama ini, terimakasih banyak.
10. Teman-teman karib satu angkatan 2007 Teknik Informatika, serta pihak-pihak lain yang tidak dapat penulis sebutkan satu persatu.

Akhirnya, penulis menyadari dalam penulisan laporan ini masih terdapat kekurangan. Oleh karena itu, saran dan kritik sangat penulis harapkan untuk kemajuan penulis secara pribadi. Terimakasih.

Pekanbaru, September 2013

Penulis

## DAFTAR ISI

LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN .....	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL.....	iv
LEMBAR PERNYATAAN .....	v
LEMBAR PERSEMBAHAN .....	vi
<i>ABSTRACT</i> .....	vii
ABSTRAK .....	viii
KATA PENGANTAR .....	ix
DAFTAR ISI .....	xi
DAFTAR GAMBAR .....	xv
DAFTAR TABEL.....	xviii
DAFTAR LAMPIRAN.....	xx
DAFTAR SIMBOL.....	xxi
DAFTAR ISTILAH .....	xxii
BAB I PENDAHULUAN .....	I-1
1.1. Latar Belakang .....	I-1
1.2. Rumusan Masalah .....	I-3
1.3. Batasan Masalah.....	I-3
1.4. Tujuan .....	I-3
1.5. Sistematika Penulisan .....	I-3
BAB II LANDASAN TEORI .....	II-1
2.1. <i>Mobile Computing</i> .....	II-1
2.1.1. <i>Wireless Mobile Network</i> .....	II-2
2.2. Sistem Informasi Geografis.....	II-2
2.3. Android.....	II-3
2.3.1. Sejarah Android .....	II-4

2.3.2. Arsitektur Android.....	II-6
2.4. <i>Location Based Service</i> (LBS) .....	II-7
2.4.1. Komponen Pendukung Utama LBS .....	II-8
2.5. Teori Graf.....	II-10
2.5.1. Jenis-Jenis Graf.....	II-11
2.5.2. Algoritma <i>Floyd-Warshall</i> .....	II-12
2.6. <i>Unified Modeling Language</i> (UML) .....	II-14
2.6.1. <i>Use Case Diagram</i> .....	II-14
2.6.2. <i>Class Diagram</i> .....	II-15
2.6.3. <i>Squence Diagram</i> .....	II-15
2.7. <i>Rational Unified Process</i> (RUP) .....	II-15
2.7.1. Fase pada <i>Rational Unified Process</i> (RUP) .....	II-16
<b>BAB III METODOLOGI PENELITIAN.....</b>	<b>III-1</b>
3.1. Tahapan Penelitian .....	III-1
3.2. Fase <i>Inception</i> .....	III-2
3.3. Fase <i>Elaboration</i> .....	III-2
3.4. Fase <i>Construction</i> .....	III-3
3.5. Fase <i>Transation</i> .....	III-3
<b>BAB IV ANALISA DAN PERANCANGAN.....</b>	<b>IV-1</b>
4.1. Analisis Permasalahan.....	IV-1
4.1.1. Defenisi Kebutuhan Sistem .....	IV-2
4.1.1.1. Project Plan.....	IV-2
A. Deskripsi Umum Sistem .....	IV-3
B. Deskripsi Kebutuhan Sistem.....	IV-4
C. Fungsi Sistem.....	IV-5
4.1.2. Algoritma <i>Floyd Warshall</i> .....	IV-8
4.1.2.1. Langkah Kerja Algoritma <i>Floyd Warshall</i> .....	IV-8
4.1.2.2. Perhitungan Manual Algoritma <i>Floyd Warshall</i> .....	IV-8
4.1.2.3. <i>Flowchart</i> Algoritma <i>Floyd-Warshall</i> .....	IV-9

4.1.2.4. Analisa Algoritma <i>Floyd-Warshall</i> .....	IV-11
4.1.3. Pembuatan Model .....	IV-15
A. Model Sistem .....	IV-15
B. Deskripsi Pengguna .....	IV-15
C. Pemodelan UML ( <i>Unified Modeling Language</i> ) .....	IV-16
4.2. Perancangan .....	IV-22
4.2.1. Perancangan <i>Database</i> .....	IV-22
4.2.2. Perancangan Struktur Menu Sistem .....	IV-24
4.2.3. Perancangan <i>interface</i> Pengguna Sistem .....	IV-24
4.2.3.1. Perancangan <i>interface Home</i> di Android .....	IV-25
<b>BAB V IMPLEMENTASI DAN PENGUJIAN</b> .....	<b>V-1</b>
5.1. Implementasi .....	V-1
5.1.1. Pembuatan Aplikasi .....	V-1
5.1.1.1. Pengkodean .....	V-1
A. Pembuatan Aplikasi .....	V-1
B. Batasan Dalam Pembuatan Aplikasi .....	V-3
5.1.1.2. Implementasi .....	V-4
A. Tahap-tahap Implementasi .....	V-4
B. Implementasi Aplikasi Perangkat Android .....	V-5
C. Implementasi Jalur Terpendek Sistem dan Penghitungan Secara Manual .....	V-7
D. Hasil Implementasi Antar Muka Database Aplikasi .....	V-12
5.2. Pengujian .....	V-14
5.2.1. Pengujian Aplikasi .....	V-14
A. Pengujian <i>Blackbox</i> Aplikasi LBS BTS .....	V-14
B. Pengujian Akses Aplikasi LBS BTS .....	V-16
C. Pengujian Jalur Terpendek .....	V-20
5.2.2. Kesimpulan Pengujian .....	V-24

BAB VI PENUTUP .....	VI-1
6.1. Kesimpulan.....	VI-1
6.2. Saran.....	VI-1
DAFTAR PUSTAKA	
LAMPIRAN	



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

CV.2K merupakan sebuah perusahaan yang bergerak dibidang penyedia layanan Internet atau *Internet Service Provider* (ISP). Di kota Pekanbaru terdapat beberapa *Base Tower Station* (BTS) CV.2K yang tersebar di beberapa kecamatan yang saling terhubung satu dengan yang lain. Hal ini menyebabkan penting bagi CV.2K untuk selalu mengontrol dan merawat (*maintenance*) BTS milik CV.2K agar kualitas sinyal jaringan mereka tetap terjaga dengan baik. Untuk ini dibutuhkan sebuah aplikasi *Location Base Services* (LBS) yang berjalan pada *platform* Android yang dapat memetakan BTS CV.2K tersebut. Aplikasi ini akan digunakan oleh teknisi CV.2K yang dapat membantu teknisi menemukan lokasi BTS dan mendapatkan jalur terpendek yang akan dilalui apabila terjadi kerusakan dan melakukan perawatan pada BTS CV.2K.

Berbagai macam teknologi yang disediakan oleh Android adalah *Location Base Services* (LBS), beberapa penelitian yang pernah dilakukan oleh Akbar Nuzul Putra dan kawan-kawan pada tahun 2011 yang membahas “Aplikasi Wisata Kota Bandung Menggunakan Metode *Location-Based Services* (LBS) pada Android” dan penelitian yang dilakukan oleh Parno dan kawan-kawan pada tahun 2012 “Aplikasi Android penyedia informasi alat transportasi umum (studi kasus : angkutan umum di terminal kampong rambutan)” kedua penelitian ini menggunakan teknologi LBS dalam melakukan pencarian lokasi geografis dari perangkat *mobile* untuk memunculkan berbagai informasi dari lokasi yang diperolehnya. LBS menghasilkan informasi yang dapat dicocokkan dengan keberadaan pengguna informasi sehingga menghasilkan informasi yang bernilai lebih bagi pengguna.

Seiring meningkatnya kemajuan teknologi dan tingginya mobilitas serta kebutuhan akan informasi berupa peta *Geographic Information System* (GIS), dan dengan adanya teknologi LBS maka sebuah inovasi baru perlu dilakukan. Seperti aplikasi yang dapat berjalan pada perangkat bergerak dengan sistem operasi terbaru pada saat ini salah satunya Android.

Android adalah sistem operasi yang berbasis *Linux* untuk perangkat *mobile* seperti *smartphone* dan tablet PC. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi dan merubah system ponsel mereka sendiri. Android mulanya proyek sekelompok programer di Palo Alto, California. Perusahaan kecil yang didirikan oleh Andy Rubin dan Nick Sears dibeli Google pada 2005. Sempat disempurnakan berkali-kali, Android baru benar-benar diluncurkan empat tahun kemudian ketika Google menarik ponsel bermerek T-Mobile untuk bergabung bersama mereka.

Android sendiri merupakan sistem operasi *mobile* yang perkembangannya begitu pesat. Pengguna android dari waktu ke waktu semakin meningkat sesuai penelitian dari *IDC Worldwide Mobile Phone Tracker* yang dikeluarkan pada Rabu 08 Agustus 2012, Android menduduki peringkat pertama di dunia dengan persentase pengguna 68,1 %. Di Indonesia sendiri perkembangan android juga sangat pesat. Menurut hasil penelitian dari *International Data Corporation (IDC)*, dalam artikel yang baru dirilisnya yaitu “*Android Overtakes Blackberry as the Top Smartphone Operating System in Indonesia*” pada rabu, 12 September 2012, mengatakan bahwa BlackBerry yang menguasai pasar perangkat *mobile* di Indonesia telah diambil alih posisinya oleh android dengan pemakai mencapai 52% di kuartal kedua tahun 2012. Android beranjak naik signifikan melalui vendor Samsung, HTC dan Sony. Menurut Lie Darwin seorang *market analyst for client device research at International Data Corporation (IDC)*, mengatakan peningkatan ponsel berbasis android di Indonesia didorong tidak hanya karena harga yang terjangkau tetapi juga aplikasi yang disediakan oleh android itu sendiri sangat banyak dan semakin populernya teknologi layar sentuh pada ponsel berbasis android.

Beberapa penelitian yang pernah dilakukan sebelumnya yaitu dilakukan oleh Apri Kamayudi yang membahas “*Studi dan Implementasi Algoritma Dijkstra, Bellman-Ford dan Floyd-Warshall Dalam Menangani Masalah Lintasan Terpendek Dalam Graf*” penelitian ini menjelaskan Algoritma *Dijkstra* dan Algoritma *Bellman-Ford* dapat digunakan untuk menyelesaikan masalah pada Single-source Shortest Path dan Algoritma *Floyd-Warshall* untuk menyelesaikan masalah pada Allpairs Shortest Path. Raden Aprian Diaz Novandi pada tahun 2007 yang membahas “*Perbandingan Algoritma Dijkstra dan Algoritma Floyd-Warshall Dalam Penentuan Lintasan Terpendek (Single Pair Shortest Path)*” penelitian ini menjelaskan Algoritma *Dijkstra* yang menerapkan prinsip *greedy* tidak selalu berhasil memberikan solusi

optimum untuk kasus penentuan lintasan terpendek (*single pair shortest path*) dan Algoritma *Floyd-Warshall* yang menerapkan pemrograman dinamis lebih menjamin keberhasilan penemuan solusi jalur terpendek untuk kasus penentuan lintasan terpendek (*single pair shortest path*). Kedua penelitian ini membandingkan beberapa algoritma dalam pencarian rute terpendek, penulis memilih algoritma *Floyd-Warshall* dalam pencarian jalur terpendek yang akan digunakan dalam Aplikasi *Location Base Services (LBS) Base Tower Station (BTS) CV.2K* karena algoritma ini dapat menemukan jalur terpendek.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang masalah tersebut dapat suatu rumusan masalah yaitu “Bagaimana membuat Aplikasi *Location Base Services (LBS) Base Tower Station (BTS) CV.2K* Dengan Menerapkan Algoritma *Floyd-Warshall*”.

## 1.3 Batasan Masalah

Untuk mendapatkan hasil yang optimal dalam penelitian tugas akhir ini, maka diberikan batasan-batasan penelitian sebagai berikut:

1. BTS yang dipetakan pada aplikasi ini terletak di kota Pekanbaru.
2. Penelitian ini tidak membahas tentang keamanan data pada aplikasi yang dibuat.
3. Aplikasi berjalan pada *platform* Android.

## 1.4 Tujuan

Adapun tujuan yang ingin dicapai penulis dari penelitian Tugas Akhir ini adalah membangun sebuah aplikasi *Location Base Services (LBS) Base Tower Station (BTS) CV.2K* yang berjalan di *platform* Android.

## 1.5 Sistematika Penulisan

Sistematika penulisan tugas akhir ini terdiri dari enam bagian, penjelasan dari keenam bagian ini yaitu:

**BAB I           PENDAHULUAN**

Membahas mengenai latar belakang permasalahan, rumusan masalah, batasan masalah, tujuan pembahasan, metodologi penelitian dan sistematika penulisan.

**BAB II           LANDASAN TEORI**

Membahas teori-teori pendukung. Teori yang diangkat yaitu mengenai *Mobile Computing*, *Wireless Mobile Network*, Sistem Informasi Geografis, Sistem Operasi Android pada telepon selular, *Location Based Service*, Google Map, Graf dan algoritma *Floyd-Warshall*.

**BAB III          Metodologi Penelitian**

Membahas langkah-langkah yang akan dilaksanakan dalam proses penelitian, yaitu pengamatan pendahuluan, tahapan identifikasi masalah, perumusan masalah, analisa aplikasi, perancangan aplikasi dan implementasi serta pengujian aplikasi yang telah dibuat.

**BAB IV          Analisa dan Perancangan**

Membahas mengenai kebutuhan sistem yang terdiri dari : *Flowchart* sistem, *Unified Modeling Language (UML)*, *user interface*, perancangan menggunakan pendekatan berorientasi objek.

**BAB V           Implementasi dan Pengujian**

Membahas tentang implementasi penelitian dan eksperimen yang dilakukan dengan melakukan berbagai evaluasi dan perbaikan yang dirasa perlu berdasarkan hasil penelitian yang dilakukan.

**BAB VI          Kesimpulan dan Saran**

Membahas kesimpulan yang dapat diambil dari pembahasan dan pengembangan dari aplikasi *Location Base Services (LBS) Base Tower Station (BTS)* untuk wilayah Kota Pekanbaru.

## BAB II

### LANDASAN TEORI

#### 2.1. *Mobile Computing*

*Mobile computing* merupakan paradigma baru dari teknologi yang mampu melakukan komunikasi walaupun user melakukan pergerakan, sering di sebut sebagai *mobile computer* (portable computer) yang dapat berkomunikasi dengan jaringan tanpa kabel (*nirkabel*).

Berdasarkan defenisi di atas kita dapat memahami mengapa kita membutuhkan *mobile computing*. Kata kuncinya adalah kita manusia dinamis yang selalu bergerak dan berkembang dari keadaan ke keadaan yang lain. Sehingga membutuhkan sesuatu device yang mampu mengikuti pergerakan manusia. Bergerak dapat dilihat dari dua sisi yaitu orang dan *device*.

1. Orang bergerak
  - a. Perpindahan posisi geografis
  - b. Perpindahan jaringan komunikasi
  - c. Perpindahan peralatan komunikasi
2. Device bergerak
  - a. Perpindahan posisi geografis
  - b. Perpindahan jaringan komunikasi

Dengan perkembangan *mobile computing* yang sangat pesat telah memberikan berbagai keuntungan dan kerugian dari pemakaian *mobile computing* tersebut.

1. Keuntungan *mobile computing* diantaranya adalah:
  - a. Dapat mengakses informasi setiap saat dan dimanapun
  - b. Mobilitas tinggi tanpa kerumitan kabel
  - c. Kompatible yang tinggi dengan teknologi lain
2. Kekurangan *mobie computing* diantaranya adalah:
  - a. Kemanan atau *security* data dan jaringan
  - b. Tergantung terhadap cuaca
  - c. *Bandwidth* yang belum memadai

### 2.1.1. *Wireless Mobile Network*

*Wireless* adalah teknologi yang digunakan pada *mobile computing* untuk melakukan proses transfer data dengan menggunakan jaringan tanpa kabel. Perkembangan *wireless mobile network* dapat diklasifikasikan menjadi beberapa generasi:

#### 1. Generasi 1G

Generasi 1G merupakan sinyal radio yang masih bersifat analog yang terdapat pada frekuensi 800 Mhz dan 400 Mhz. Perkembangan teknologi ini dimulai dari Chicago, USA dan mulai dikomersilkan pada tahun 1983

#### 2. Generasi 2G

Karakteristik dari teknologi 2G ini adalah sistem komunikasi *wireless* yang sudah bersifat digital. Dengan teknologi digital berbasis TDD/TDMA, dapat menghasilkan kualitas suara yang lebih baik dibandingkan dengan sistem analog sebelumnya. Teknologi ini beroperasi pada frekuensi 800 Mhz – 1900 Mhz

#### 3. Generasi 3G

Teknologi 3G dibagi menjadi GSM dan CDMA. Teknologi ini sering juga disebut dengan sebutan *mobile broadband* karena keunggulannya sebagai modem untuk internet yang dapat dibawa kemana saja. Perkembangan 3G secara komersial dimulai pada Oktober 2001. Sistem 3G dimaksudkan untuk menyediakan *global mobility* dengan cakupan layanan yang lebih luas, seperti telephony, paging, messaging, internet dan broadband data.

#### 4. Generasi 4G

Belakangan ini industri nirkabel mulai mengembangkan teknologi 4G, meskipun sebenarnya teknologi 4G ini seperti *Long Term Evolution (LTE)* hanya merupakan evolusi dari teknologi 3GPP dan *Ultra Mobile Broadband (UMB)* berasal dari 3GPP2, sehingga sulit untuk membedakan dengan jelas teknologi 3G dan 4G. Salah satu teknologi 4G yaitu *WiMax Mobile* telah diterima oleh ITU untuk ditambahkan pada IMT-2000, sehingga teknologi baru ini masih digolongkan kedalam keluarga 3G.

## 2.2. **Sistem Informasi Geografis**

Sistem Informasi Geografis (SIG) merupakan suatu kesatuan formal yang terdiri dari berbagai sumber daya fisik dan logika yang berkenaan dengan objek-objek yang terdapat di permukaan bumi. SIG merupakan sejenis perangkat lunak atau aplikasi yang dapat digunakan

untuk pemasukan data, penyimpanan data, manipulasi data, menampilkan data, dan keluaran informasi geografis (Adeline dan Eri, 2007).

Teknologi GIS mengintegrasikan operasi pengolahan data berbasis *database* yang biasa digunakan saat ini, seperti pengambilan data berdasarkan kebutuhan, serta analisis *statistic* dengan menggunakan visualisasi yang khas serta berbagai keuntungan yang mampu ditawarkan melalui analisis geografis melalui gambar-gambar petanya.

Sistem Informasi Geografis terdiri dari 4 komponen :

a. Perangkat keras

Seperti komputer untuk penginputan dan pengolahan data, dan perangkat keras untuk keluaran data seperti *printer*, *plotter* dan lain sebagainya.

b. Perangkat Lunak

Merupakan sistem aplikasi yang tersusun secara modular dengan basis data sebagai pemegang peranan penting dalam sistem perangkat lunak. Lebih tepatnya berupa tools-tools untuk membuat dan mengolah system informasi geografis. Seperti *Arc View*, *Idrisi*, *ARC/INFO*, *ILWIS*, *MapInfo* dan lain lain.

c. Data dan Informasi Geografis

SIG dapat melakukan pengumpulan data, penyimpanan data, dan pemrosesan data dan informasi yang diperlukan baik secara tidak langsung dengan cara mengimpornya dari perangkat lunak SIG yang lain, maupun secara langsung dengan cara mendijitasi data spasial dari peta.

d. Manajemen

Keberhasilan suatu sistem informasi geografis akan berhasil jika diatur sedemikian rupa dan dikerjakan oleh orang-orang yang memiliki keahlian pada bidangnya (Adeline dan Eri, 2007).

### 2.3. *Android*

*Android* adalah sistem operasi untuk perangkat mobile yang berbasiskan *Linux*. Sistem operasi ini menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh berbagai macam perangkat *mobile*. *Android* memiliki berbagai keunggulan sebagai *software* yang didistribusikan secara terbuka (*open source*), sehingga pengguna dapat membuat aplikasi baru yang diinginkan.

*Android* merupakan subset perangkat lunak untuk perangkat *mobile* meliputi sistem operasi, middleware, dan aplikasi inti yang dirilis oleh Google. Sedangkan *Android SDK*

(*Software Development Kit*) menyediakan *tools* dan API yang diperlukan untuk mengembangkan aplikasi pada platform *android* dengan menggunakan bahasa pemrograman Java, yaitu kode Java yang terkompilasi dengan data dan *file resources* yang dibutuhkan aplikasi dan digabungkan oleh *apt tools* menjadi paket *android*. File tersebut ditandai dengan ekstensi *.apk* dan file ini yang akan didistribusikan sebagai aplikasi dan diinstal pada perangkat *mobile*.

### 2.3.1. Sejarah *Android*

Pada tanggal 22 oktober 2008, HTC *Dream* adalah *smartphone* pertama yang menggunakan sistem operasi *android*. Dan hingga sekarang ini, telah banyak vendor-vendor *smartphone* yang menggunakan sistem operasi ini.

Adapun versi-versi *android* yang pernah dirilis oleh google.inc adalah sebagai berikut:

a. *Android* versi 1.1

Rilis pada maret 2009. *Android* versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam *alarm*, *voice search* (pencarian suara), pengiriman pesan dengan *Gmail*, dan pemberitahuan *email*.

b. *Android* versi 1.5 (*Cupcake*)

Rilis pada pertengahan Mei 2009. Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke *Youtube* dan gambar ke *Picasa* langsung dari telepon, dukungan *Bluetooth A2DP*, kemampuan terhubung secara otomatis ke *headset Bluetooth*, animasi layar, dan *keyboard* pada layar yang dapat disesuaikan dengan sistem.

c. *Android* versi 1.6 (*Donut*)

Rilis pada pertengahan September 2009. menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol *applet* VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, *camcorder* dan galeri yang dintegrasikan; CDMA / EVDO, 802.1x, VPN, *Gestures*, dan *Text-to-speech engine*; kemampuan dial kontak; teknologi *text to change speech* (tidak tersedia pada semua ponsel) dan pengadaan resolusi VWGA.



d. *Android* versi 2.0/2.1 (*eclair*)

Rilis pada 3 desember 2009. pengoptimalan *hardware*, peningkatan *Google Maps* 3.1.2, perubahan UI dengan *browser* baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, *digital Zoom*, dan *Bluetooth 2.1*.

e. *Android* versi 2.2 (*Frozen Yoghurt / Froyo*)

Dukungan *Adobe Flash* 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi *V8 JavaScript engine* yang dipakai *Google Chrome* yang mempercepat kemampuan *rendering* pada *browser*, pemasangan aplikasi dalam *SD Card*, kemampuan *WiFi Hotspot portabel*, dan kemampuan *auto update* dalam aplikasi *Android Market*.

f. *Android* versi 2.3 (*Gingerbread*)

Peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*, layar antar muka (*User Interface*) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (*reverb*, *equalization*, *headphone virtualization*, dan *bass boost*), dukungan kemampuan *Near Field Communication (NFC)* (Syafaat, 2011) .

g. *Android* versi 3.0/3.1 (*Honeycomb*)

*Android Honeycomb* dirancang khusus untuk *tablet*. *Android* versi ini mendukung ukuran layar yang lebih besar. *User Interface* pada *Honeycomb* juga berbeda karena sudah didesain untuk *tablet*. *Honeycomb* juga mendukung *multi prosesor* dan juga akselerasi perangkat keras (*hardware*) untuk grafis.

h. *Android* versi 4.0 (*ICS :Ice Cream Sandwich*)

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur *Honeycomb* untuk *smartphone* dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari *email* secara *offline*, dan berbagi informasi dengan menggunakan *NFC* (<http://www.jaringankomputer.org>).

i. *Android* 4.1 - 4.2 *Jelly Bean*

*Android* 4.1 *Jelly Bean* diumumkan pada 27 Juni 2012 pada konferensi *Google I/O*. Versi ini adalah yang tercepat dan terhalus dari semua versi *Android*. *Jelly Bean* 4.1 meningkatkan kemudahan dan keindahan tampilan dari *Ice Cream Sandwich* dan memperkenalkan pengalaman pencarian *Google* yang baru di *Android*. *Android* 4.2 *Jelly Bean* diumumkan pada 29 October 2012, versi ini menawarkan peningkatan kecepatan dan kemudahan *Android* 4.1 serta mencakup semua fitur baru

seperti Photo Sphere dan desain baru aplikasi kamera, *keyboard* Gesture Typing, Google Now dan lainnya. Jelly Bean adalah sejenis permen yang juga populer disebut kacang jeli (<http://teknologi.inilah.com/read/detail/1992003/inilah-sejarah-penamaan-dari-tiap-versi-android#.Uc7j8zsXGgs>).

### 2.3.2. Arsitektur *Android*

Ada beberapa arsitektur dari sistem operasi *mobile android*, diantaranya dijelaskan sebagai berikut:

#### a. *Applications and Widgets*

Layer ini merupakan daerah dimana si pengguna berhubungan dengan aplikasi yang digunakan, dari mengunduh (*download*), mengunggah (*install*) hingga menjalankan aplikasi tersebut. Pada layer ini, semua aplikasi yang digunakan menggunakan bahasa pemrograman *Java*.

#### b. *Applications Frameworks*

Adalah layer dimana para pembuat aplikasi melakukan pengembangan ataupun pembuatan aplikasi baru. karena pada layer ini, diberikan akses penuh terhadap sistem operasi *android*.

#### c. *Libraries*

Merupakan tempat dimana fitur-fitur *android* berada. Beberapa fitur tersebut adalah:

1. *Libraries* media, berhubungan dengan video dan audio
2. *Libraries* untuk mengatur tampilan layar
3. *Libraries graphics*
4. *Libraries SQLite* untuk database
5. *Libraries SSL* dan *WebKit* yang terintegrasi dengan *browser* dan keamanan
6. *Libraries LiveWebcore*
7. *Libraries 3D*

#### d. *Android Run Time*

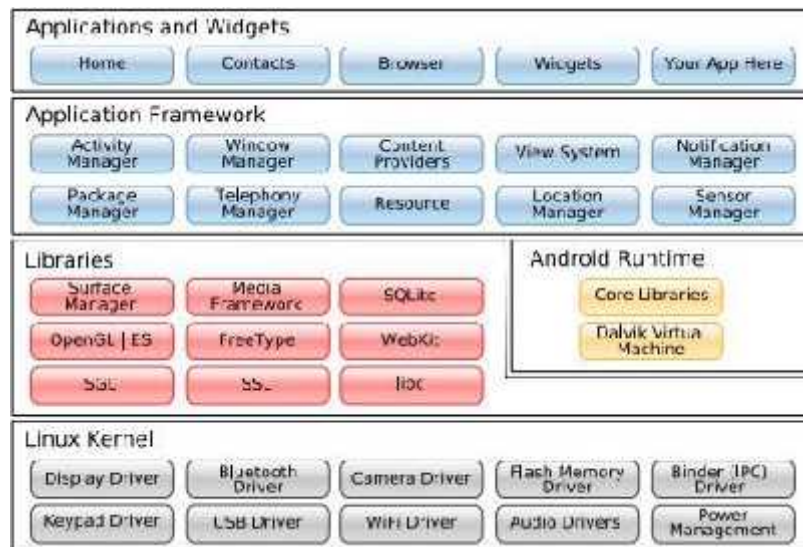
Layer yang mampu menjalankan aplikasi *android* yang telah diinstal. Didalam layer ini terbagi 2 yaitu:

1. *Core Libraries*
2. *Davik Virtual Machine (DVM)*

#### e. *Linux kernel*

Layer ini merupakan inti dari sistem operasi *android*. Berisi file-file sistem yang mampu mengatur sistem *processing*, *memory*, *drivers*, dan sistem yang lainnya pada

*android*. Kernel yang digunakan pada sistem operasi *android* ini adalah *linux kernel release 2.6* (Syafaat, 2011).



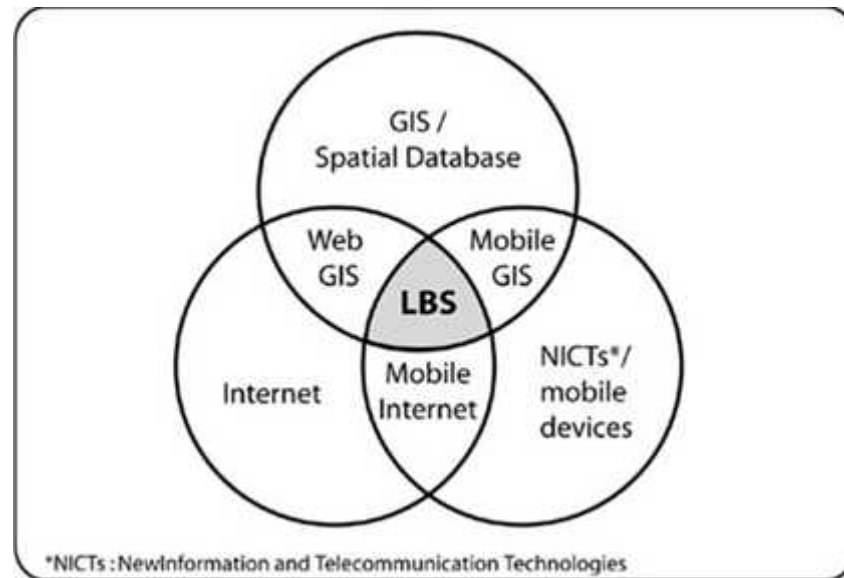
Gambar 2.1 Arsitektur *Android*

(sumber : <http://www.android-indonesia.com/home/15-developers/12156-arsitektur-android>)

#### 2.4. *Location Based Service (LBS)*

LBS atau Layanan Berbasis Lokasi adalah istilah umum yang digunakan untuk menggambarkan teknologi yang dapat menemukan lokasi perangkat *mobile* yang sedang digunakan. Ada dua definisi yang menjelaskan tentang *Location Based Services (LBS)*. Definisi pertama LBS adalah layanan informasi yang dapat diakses menggunakan perangkat *mobile* melalui jaringan internet dan seluler serta memanfaatkan kemampuan penunjuk lokasi pada perangkat *mobile* dalam (Steiniger,2012) definisi yang sama juga dikemukakan oleh *International OpenGeospatial Comsorium (OGC, 2005)*. Definisi kedua adalah layanan IP nirkabel yang menggunakan informasi geografis untuk memberikan layanan informasi lokasi kepada pengguna. Beberapa layanan aplikasi yang memberikan petunjuk posisi/lokasi perangkat *mobile* berada.

Dari definisi yang ada *Location Based Services (LBS)* dapat dinyatakan sebagai suatu layanan yang berada pada pertemuan tiga teknologi yaitu, *New Infromation and Communication Technologies (NICTS)* seperti sistem *mobile* telekomunikasi, Internet dan GIS dengan database spasial dalam (Steniger, 2012)



Gambar 2.2 LBS merupakan pertemuan tiga teknologi (Steiniger, 2012)

Sekarang ini, pengembangan LBS lebih banyak ke ponsel-ponsel cerdas. Ini dapat dilihat dengan berkembangnya aplikasi-aplikasi yang berbasis LBS pada perangkat *mobile* smartphone (Syafaat, 2011). LBS memungkinkan pengguna untuk melakukan komunikasi dua arah antara pengguna dan penyedia layanan untuk mendapatkan informasi yang dibutuhkan dengan referensi posisi pengguna tersebut.

Layanan berbasis lokasi dapat digambarkan sebagai suatu layanan yang berada pada pertemuan tiga teknologi yaitu : *Geographic Information System*, *Internet Service*, dan *Mobile Devices*, hal ini dapat dilihat pada gambar LBS adalah pertemuan dari tiga teknologi.

#### 2.4.1. Komponen Pendukung Utama LBS

Terdapat lima komponen pendukung utama teknologi Layanan Berbasis Lokasi antara lain:

##### 1. Perangkat *Mobile*

perangkat *mobile* adalah salah satu komponen penting dalam LBS. Perangkat ini berfungsi sebagai alat bantu (*tool*) bagi pengguna untuk meminta informasi. Hasil dari informasi yang diminta dapat berupa teks, suara, gambar dan lain sebagainya. Perangkat *mobile* yang dapat digunakan bisa berupa PDA, *smartphone*, *laptop*. Selain itu, perangkat *mobile* dapat juga berfungsi sebagai alat navigasi di kendaraan seperti halnya alat navigasi berbasis GPS.

## 2. Jaringan Komunikasi

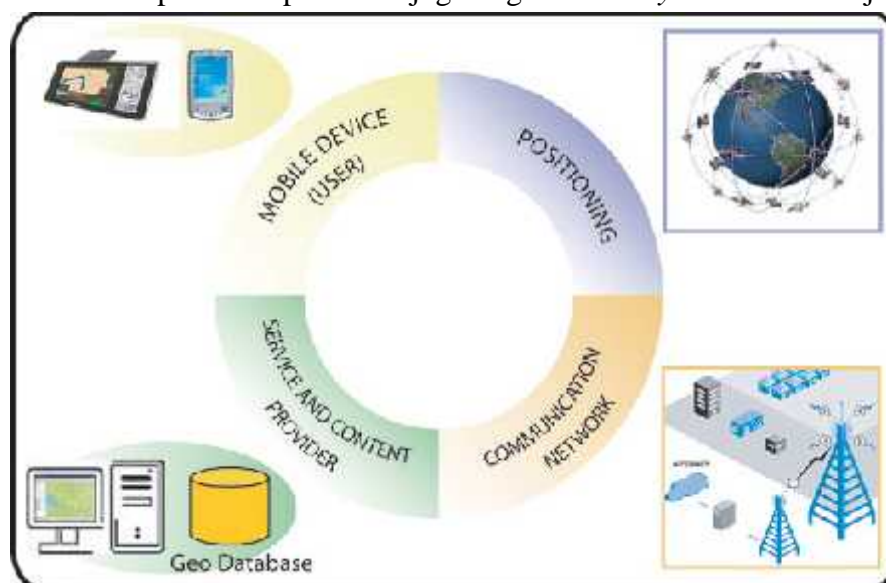
Komponen ini berfungsi sebagai jalur penghubung yang dapat mengirimkan data-data yang dikirim oleh pengguna dari perangkat *mobile*-nya untuk kemudian dikirimkan ke penyedia layanan dan kemudia hasil permintaan tersebut dikirimkan kembali oleh penyedia layanan kepada pengguna.

## 3. Komponen *Positioning* atau petunjuk lokasi

Setiap layanan yang diberikan oleh penyedia layanan biasanya akan berdasarkan pada posisi pengguna yang meminta layanan tersebut. Oleh karena itu diperlukan komponen yang berfungsi sebagai pegolah atau pemroses yang akan menentukan posisi pengguna layanan pada saat itu. Posisi pengguna tersebut bisa didapatkan melalui jaingan komunikasi *mobile* atau juga menggunakan *Global Positioning System* (GPS).

## 4. Penyedia data dan konten

Penyedia layanan tidak selalu menyimpan seluruh data dan informasi yang diolahnya. Karena bisa jadi berbagai macam data dan informasi yang diolah tersebut berasal dari pengembang/pihak ketiga yang memang memiliki otoritas untuk menyimpannya. Sebagai contoh basis data geografis dan lokasi bisa saja berasal dari badan-badan milik pemerintah atau juga data-data perusahaan/bisnis/industri bisa saja berasal dari *Yellow Pages*, maupun perusahaan penyedia data lainnya. Pada komponen ini terdapat database lokasi BTS CV.2K dan pada komponen ini juga Algoritma *Floyd-Warshall* berjalan.



Gambar 2.3 Komponen pendukung utama teknologi LBS (Steiniger, 2012)

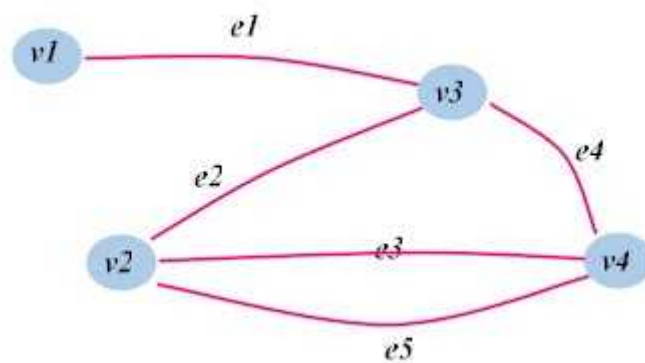
## 2.5. Teori Graf

Graf adalah kumpulan simpul (*nodes*) yang dihubungkan satu sama lain melalui sisi/busur (*edges* atau *arcs*). Graf digunakan untuk merepresentasikan objek-objek diskrit dan hubungan antar objek tersebut. Representasi visual dari graf adalah suatu objek dinyatakan sebagai noktah, bulatan atau titik, dan hubungan antar objek dinyatakan dengan garis. (Rinaldi Munir, 2005)

Suatu Graf  $G$  terdiri dari dua himpunan Graf yaitu himpunan  $V$  dan himpunan  $E$ .

- Verteks* (simpul) dilambangkan dengan  $V$  = himpunan simpul yang terbatas dan tidak kosong.
- Edge* (sisi/busur) dilambangkan dengan  $E$  = himpunan busur yang menghubungkan sepasang simpul.

Dapat dikatakan graf adalah kumpulan dari simpul-simpul yang dihubungkan oleh sisi-sisi.



Gambar 2.4 Graf  $G$

<http://home.unpar.ac.id/~gatut/kuliah/PPSIK/Jaringan-Komputer/>

Pada  $G$  diatas, graf terdiri dari himpunan  $V$  dan  $E$  yaitu:

$$V = \{v1, v2, v3, v4\}$$

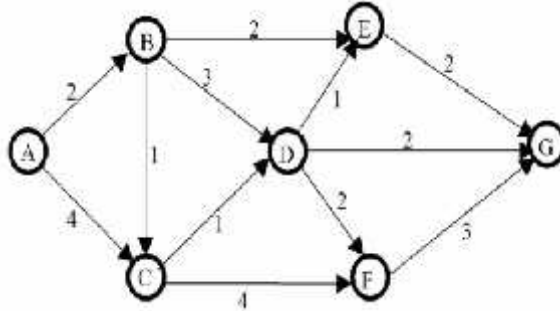
$$E = \{e1, e2, e3, e4, e5\}$$

Dari contoh di atas, maka  $e1=(v1,v3)$ ,  $e2=(v2,v3)$ ,  $e3=(v2,v4)$ ,  $e4=(v3,v4)$ ,  $e5=(v4,v2)$

**2.5.1. Jenis-Jenis Graf :**

- a. Graf berarah dan berbobot

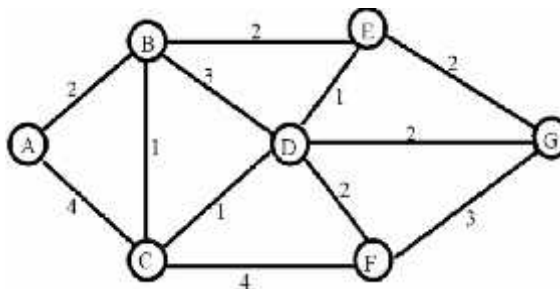
Adalah graf yang mempunyai bobot nilai yang menentukan nilai dari setiap edge disertai arahnya dari noktah satu menuju ke noktah yang lainnya.



Gambar 2.5 Graf berarah dan berbobot  
(sumber : Ifatul Faizah, 2010)

- b. Graf tidak berarah dan berbobot

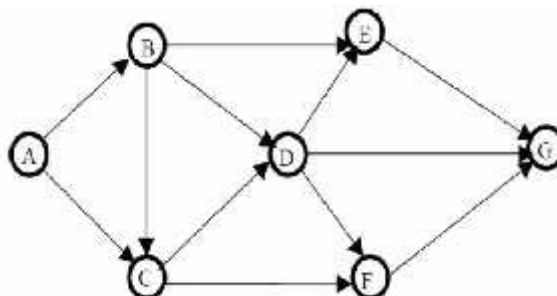
Graf ini hampir sama dengan graf sebelumnya, yang membedakannya adalah pada tiap *edge*, tidak memiliki arah baik dari noktah awal ke tujuan ataupun sebaliknya.



Gambar 2.6 Graf tidak berarah dan berbobot  
(sumber : Ifatul Faizah, 2010)

- c. Graf berarah dan tidak berbobot

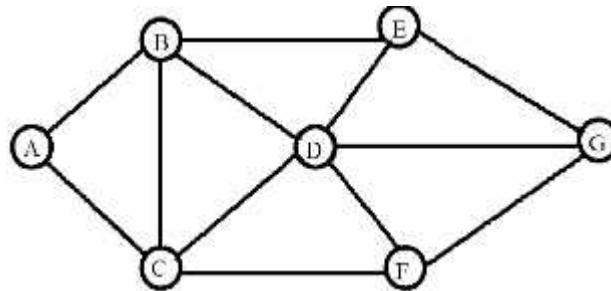
Tiap busur mempunyai anak panah yang saling menghubungkan noktah dan tidak memiliki berbobot nilai.



Gambar 2.7 Graf berarah dan tidak berbobot  
(sumber : Ifatul Faizah, 2010)

- d. Graf tidak berarah dan tidak berbobot

Tiap busur tidak mempunyai anak panah dan tidak mempunyai bobot nilai.



Gambar 2.8. Graf tidak berarah dan tidak berbobot  
(sumber : Ifatul Faizah, 2010)

### 2.5.2. Algoritma *Floyd-Warshall*

Algoritma *Floyd-Warshall* adalah salah satu varian dari pemrograman dinamis, yaitu suatu metode yang melakukan pemecahan masalah dengan memandang solusi yang akan diperoleh sebagai suatu keputusan yang saling terkait. Artinya solusi-solusi tersebut dibentuk dari solusi yang berasal dari tahap sebelumnya dan ada kemungkinan solusi lebih dari satu.

Hal yang membedakan pencarian solusi menggunakan pemrograman dinamis dengan algoritma greedy adalah bahwa keputusan yang diambil pada tiap tahap pada algoritma greedy hanya berdasarkan pada informasi yang terbatas sehingga nilai terkecil yang diperoleh pada saat itu. Jadi pada algoritma greedy, kita tidak memikirkan konsekuensi yang akan terjadi seandainya kita memilih suatu keputusan pada suatu tahap.

Dalam beberapa kasus, algoritma greedy gagal memberikan solusi terbaik karena kelemahan yang dimilikinya tadi. Di sinilah peran pemrograman dinamis yang mencoba untuk memberikan solusi yang memiliki pemikiran terhadap konsekuensi yang ditimbulkan dari pengambilan keputusan pada suatu tahap. Pemrograman dinamis mampu mengurangi pengenumerasian keputusan yang tidak mengarah ke solusi. Prinsip yang dipegang oleh pemrograman dinamis adalah prinsip optimalitas, yaitu jika solusi total optimal, maka bagian solusi sampai suatu tahap (misalnya tahap ke- $i$ ) juga optimal.

Algoritma *Floyd-Warshall* membandingkan semua kemungkinan lintasan pada graf untuk setiap sisi dari semua simpul. Menariknya, algoritma ini mampu mengerjakan proses perbandingan ini sebanyak  $V^3$  kali (bandingkan dengan kemungkinan jumlah sisi sebanyak  $V^2$  (kuadrat jumlah simpul) pada graf, dan setiap kombinasi sisi diujikan). Hal tersebut bisa terjadi karena adanya perkiraan pengambilan keputusan (pemilihan jalur terpendek) pada



setiap tahap antara dua simpul, hingga perkiraan tersebut diketahui sebagai nilai optimal, misalkan terdapat suatu graf  $G$  dengan simpul-simpul  $V$  yang masing-masing bernomor 1 s.d.  $N$  (sebanyak  $N$  buah). Misalkan pula terdapat suatu fungsi  $\text{shortestPath}(i, j, k)$  yang mengembalikan kemungkinan jalur terpendek dari  $i$  ke  $j$  dengan hanya memanfaatkan simpul 1 s.d.  $k$  sebagai titik perantara. Tujuan akhir penggunaan fungsi ini adalah untuk mencari jalur terpendek dari setiap simpul  $i$  ke simpul  $j$  dengan perantara simpul 1 s.d.  $k+1$ .

Ada dua kemungkinan yang terjadi:

1. Jalur terpendek yang sebenarnya hanya berasal dari simpul-simpul yang berada antara 1 hingga  $k$ .
2. Ada sebagian jalur yang berasal dari simpul-simpul  $i$  s.d.  $k+1$ , dan juga dari  $k+1$  hingga  $j$

Perlu diketahui bahwa jalur terpendek dari  $i$  ke  $j$  yang hanya melewati simpul 1 s.d.  $k$  telah didefinisikan pada fungsi  $\text{shortestPath}(i, j, k)$  dan telah jelas bahwa jika ada solusi dari  $i$  s.d.  $k+1$  hingga  $j$ , maka panjang dari solusi tadi adalah jumlah (konkatenasi) dari jalur terpendek dari  $i$  s.d.  $k+1$  (yang melewati simpul-simpul 1 s.d.  $k$ ), dan jalur terpendek dari  $k+1$  s.d.  $j$  (juga menggunakan simpul-simpul dari 1 s.d.  $k$ ). Maka dari itu, rumus untuk fungsi  $\text{shortestPath}(i, j, k)$  bisa ditulis sebagai suatu notasi rekursif sbb :

```

Basis-0
shortestPath(i, j, 0) = edgeCost(i,
j);
Rekurens
shortestPath(i, j, k) =
min(shortestPath(i, j, k-1),
shortestPath(i, k, k-1) +
shortestPath(k, j, k-1));

```

Rumus ini adalah inti dari algoritma *Floyd-Warshall*. Algoritma ini bekerja dengan menghitung  $\text{shortestPath}(i,j,1)$  untuk semua pasangan  $(i,j)$ , kemudian hasil tersebut akan digunakan untuk menghitung  $\text{shortestPath}(i,j,2)$  untuk semua pasangan  $(i,j)$ , dst. Proses ini akan terus berlangsung hingga  $k = n$  dan kita telah menemukan jalur terpendek untuk semua pasangan  $(i,j)$  menggunakan simpul-simpul perantara (Raden Aprian Diaz Novandi, 2007).

Algoritma *Floyd-Warshall*:

```
//Asumsikan bahwa terdapat fungsi edgeCost(i,j) yang mengembalikan biaya
(cost) di ujung dari i ke j (tak hingga jika tidak ada)
//Juga asumsikan bahwa n adalah jumlah simpul dan edgeCost(i,i)=0
```

```
path = array of integer;
```

```
//Matriks dua dimensi. Pada setiap langkah di algoritma, path[i][j] adalah jalur
terpendek dari i ke j memanfaatkan nilai perantara pada (1..k-1). Setiap path[i][j]
diinisialisasi ke edgeCost(i,j);
```

```
procedure FloydWarshall()
```

```
k traversal [1..n]
```

```
    foreach (i,j) pada [1..n]
```

```
        path[i][j] = min(path[i][j],
                           path[i][k]+path[k][j]);
```

```
    {end foreach}
```

```
{end traversal}
```

## 2.6. *Unified Modelling Language (UML)*

*Unified Modelling Language (UML)* adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem (Dharwiyanti dan Wahono, 2006).

Untuk merancang sebuah model, UML memiliki beberapa diagram antara lain : *use case diagram*, *class diagram*, *statechart diagram*, *activity diagram*, *sequence diagram*, *collaboration diagram*, *component diagram*, *deployment diagram*.

### 2.6.1. *Use Case Diagram*

Dharwiyanti (2006) menyebutkan, *use case diagram* merupakan sebuah gambaran fungsionalitas sebuah sistem. Sebuah *use case* merepresentasikan interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, *create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

Dalam sebuah sistem *use case diagram* akan sangat membantu dalam hal menyusun *requirement*, mengkomunikasikan rancangan dengan klien dan merancang *test case* untuk semua fitur yang ada pada sistem.

### 2.6.2. *Class Diagram*

Dharwiyanti (2006) menyebutkan, *Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

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

### 2.6.3. *Sequence Diagram*

Dharwiyanti (2006), menyebutkan *Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

## 2.7. *Rational Unified Process (RUP)*

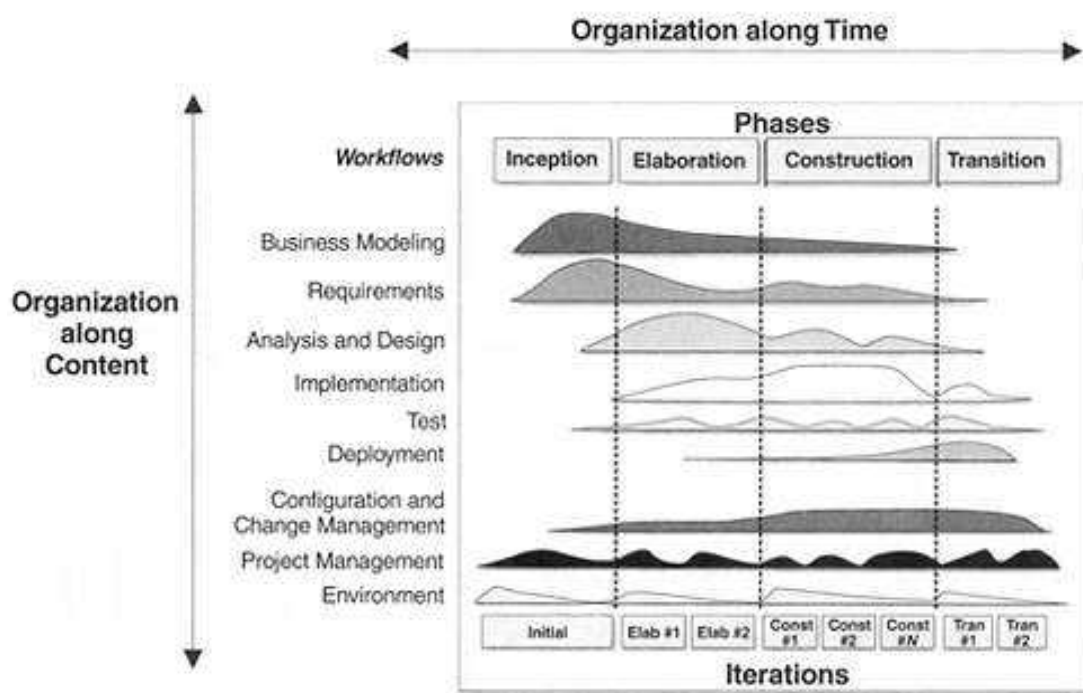
*Rational Unified Process (RUP)* merupakan suatu metode rekayasa perangkat lunak yang dikembangkan dengan mengumpulkan berbagai *best practises* yang terdapat dalam industri pengembangan perangkat lunak. Ciri utama metode ini adalah menggunakan *use-case driven* dan pendekatan iteratif untuk siklus pengembangan perangkat lunak (Kruchten, 2003).

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

1. Dimensi pertama digambarkan secara horizontal. Dimensi ini mewakili aspek-aspek

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

2. Dimensi kedua digambarkan secara vertikal. Dimensi ini mewakili aspek-aspek statis dari proses pengembangan perangkat lunak yang dikelompokkan ke dalam beberapa disiplin. Proses pengembangan perangkat lunak yang dijelaskan kedalam beberapa disiplin terdiri dari empat elemen penting, yakni *who is doing*, *what*, *how* dan *when*. Dimensi ini terdiri atas, *Business Modeling*, *Requirement, Analysis and Design*, *Implementation*, *Test*, *Deployment*, *Configuration* dan *Change Manegement*, *Project Management*, *Environment* (Kruchten, 2003)



Gambar 2.9 Arsitektur RUP (Kruchten, 2003)

### 2.7.1. Fase pada *Rational Unified Process* (RUP)

Fase-fase pada RUP dapat dibagi menjadi 4 fase, yaitu *Inception*, *Elaboration*, *Construction*, dan *Transition*.

#### 1. Fase *Inception*

Pada tahap ini dilakukan proses identifikasi masalah dan batasan kegiatan, melakukan analisa kebutuhan, manajemen resiko, prakiraan biaya, merencanakan penjadwalan,

dan perancangan awal perangkat lunak (perancangan arsitektural dan *use case*) pada akhir fase ini prototipe perangkat lunak versi *Alpha* sudah dirilis. Tujuan yang diharapkan dalam tahapan fase *inception* adalah :

- a. Tujuan dari kebutuhan proyek, kata kunci, dan batasan utama dalam proyek.
- b. Inisialisasi dalam pemodelan *use case* 10%-20% selesai.
- c. Daftar kata.
- d. *Business Case*.
- e. Penilaian resiko.
- f. Rencana proyek.
- g. Model bisnis jika diperlukan.

Kriteria evaluasi pada fase *Inception* adalah:

- a. Menyesuaikan *stackholder* dengan ruang lingkup definisi (*scope definition*) dan perencanaan biaya atau prakiraan jadwal.
- b. Pemahaman terhadap *main use case*.
- c. Kredibilitas dari perencanaan biaya, jadwal, prioritas, resiko dan proses pengembangan.
- d. Pemahaman terhadap *prototype* yang dibangun.

## 2. Fase *Elaboration*

Tujuan dari fase ini adalah fase pengembangan yaitu melakukan perancangan perangkat lunak mulai dari menspesifikasi fitur perangkat lunak hingga perilsan prototipe versi *betha* dari perangkat lunak. Tujuan yang diharapkan dari fase ini adalah :

- a. *Use case* model, seluruh *use case* dan aktor telah teridentifikasi.
- b. *Requirement* tambahan yang mungkin tidak bersifat fungsionalitas bagi proyek yang dibangun.
- c. Deskripsi arsitektur perangkat lunak.
- d. Prototipe dari arsitektur yang dapat dieksekusi.
- e. Revisi daftar tingkat resiko yang ada.
- f. Rencana pengembangan.
- g. Persiapan dokumen panduan bagi pengguna

Kriteria utama dalam fase ini adalah :

- a. Apakah produk yang dibangun sudah bersifat stabil ?
- b. Apakah rancangan arsitekturalnya sudah stabil?

- c. Apakah unsur yang berpotensi memiliki resiko telah dapat diatasi?
- d. Apakah rencana konturksi telah detail dan akurat?
- e. Pakah stackholder telah bersedia untuk menyepakati pengembangan dalam proyek tersebut?
- f. Apakah rencana penanggaran sumber daya (*actual resource*) dapat diterima?

### 3. Fase *Construction*

Pada tahap ini dilakukan pengimplementasian rancangan perangkat lunak yang telah dibuat, semua komponen dan fitur yang telah dikembangkan pada tahap sebelumnya telah terintegrasi kedalam produk dan secara menyeluruh. Proses konstruksi adalah sebuah proses *manufacturing*, dimana terdapat penekanan dalam mengelola *resource* dan mengatur operasi untuk mengoptimalkan jadwal dan kualitas. kriteria evaluasi pada fase ini adalah :

- a. Apakah produk dapat diterima dilingkungan pengguna?
- b. Apakah semua satekholder siap untuk beralih?
- c. Apakah pembelanjaan *actual-resource* terhadap rencana pembelanjaan masih tetap diterima?

### 4. Fase *Transition*

Pada fase ini dilakukanya proses transisi dari produk perangkat lunak ke pengguna akhir. Apabila produk telah digunakan oleh pengguna, maka akan ada isu-isu dari pengguna yang nantinya isu-isu ini akan digunakan sebagai perbaikan terhadap produk. Kriteria evalusai untuk tahap *Transition* adalah :

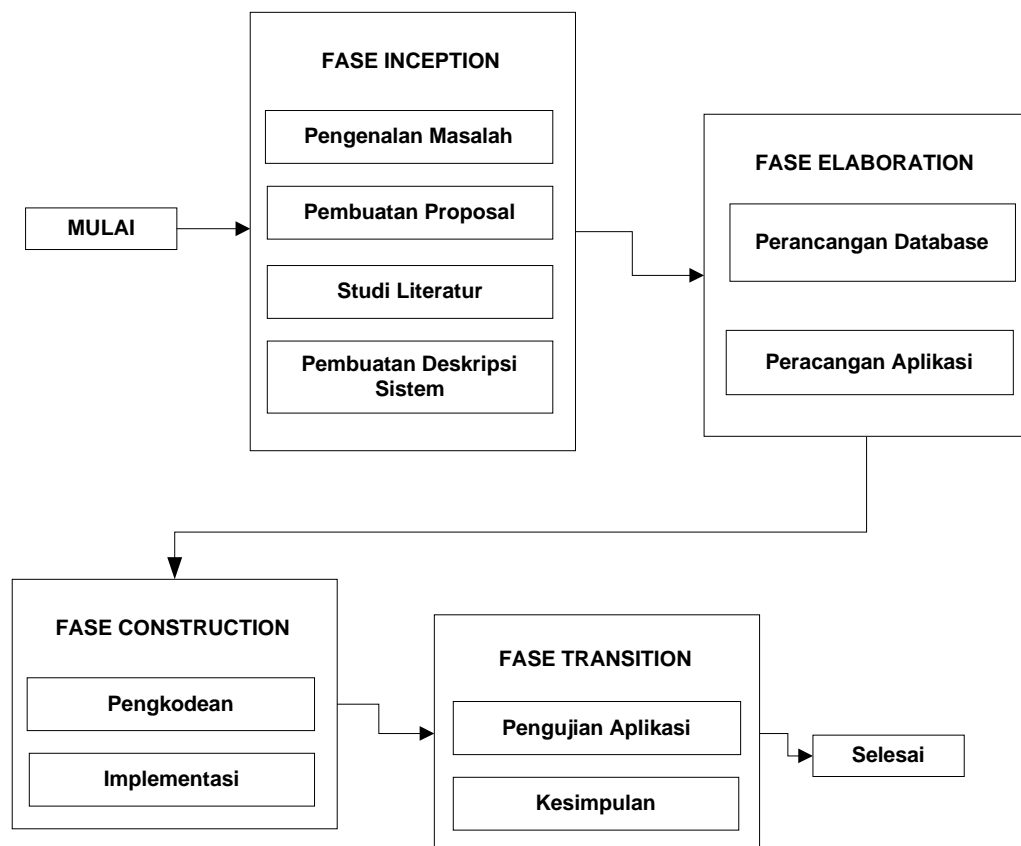
- a. Apakah pengguna merasa puas?
- b. Apakah pembelanjaan *actual resource* terhadap rencana pembelanjaan masih tetap diterima?

## BAB III

### METODOLOGI PENELITIAN

#### 3.1. Tahapan Penelitian

Pada bab ini akan dijelaskan tahapan-tahapan dalam pengerjaan Tugas Akhir ini yaitu menggunakan metode *Rational Unified Process* (RUP). Gambar 3.1 di bawah akan menjelaskan tahapan penelitian dalam membangun *Aplikasi Location Base Service (LBS) Base Tower Station (BTS) CV.2K dengan Menerapkan Algoritma Floyd-Warshall* dengan metode RUP.



Gambar 3.1 Tahapan Penelitian

### 3.2. Fase *Inception*

Pada fase ini akan dilakukan pengidentifikasian masalah yang ada pada *Aplikasi LBS BTS CV.2K Pekanbaru* yang akan dibangun. Proses yang akan dilakukan adalah sebagai berikut :

- a. Pengenalan masalah, memahami permasalahan yang terjadi, mengapa diperlukan Aplikasi LBS BTS CV.2K menggunakan algoritma *Floyd-Warshall* di kota Pekanbaru pada perangkat smartphone dengan sistem operasi Android, yaitu :
  - i. Teknologi yang berkembang pesat sehingga dibutuhkan suatu sarana informasi (database) yang dapat diakses kapan dan di mana saja.
  - ii. Dibutuhkan informasi berupa lokasi dan status BTS CV.2K untuk selalu mengontrol dan merawat (*maintenance*) BTS.
  - iii. Belum adanya penelitian yang membahas mengenai pemetaan BTS CV.2K yang bersifat arahan serta petunjuk untuk dilalui, terutama kota Pekanbaru.
- b. Pembuatan proposal Aplikasi LBS BTS CV.2K, meliputi latar belakang permasalahan, tujuan, batasan permasalahan, rumusan masalah, tujuan, dan sistematika penulisan.
- c. Studi literatur yang mencakup teori-teori yang berhubungan dengan permasalahan pembangunan aplikasi LBS BTS CV.2K, melakukan perhitungan manual Algoritma *Floyd-Warshall* yang bersumber dari buku, jurnal artikel internet dan penelitian-penelitian lainnya.
- d. Pembuatan deskripsi sistem dengan melakukan perancangan *use case*, *class diagram*, *activity diagram*, *sequence diagram*, dan *deployment diagram*.

### 3.3. Fase *Elaboration*

Pada fase ini dilakukan tahap pengembangan dan analisa terhadap aplikasi LBS BTS CV.2K yang akan dibangun. Tujuan dari fase ini adalah untuk



meminimalisir kesalahan yang akan terjadi, tugas-tugas yang akan dilakukan pada fase ini adalah :

- a. Perancangan Database LBS BTS.
- b. Perancangan Aplikasi LBS BTS.

### **3.4. Fase *Construction***

Fase *Construction* merupakan tahapan dimana aplikasi LBS BTS CV.2K dapat dibangun sesuai dengan rancangan yang ada pada tahap *elaboration*. Pada fase ini akan dilakukan tugas-tugas :

- a. Membangun (pengkodean) aplikasi LBS BTS CV.2K berbasis *android* berdasarkan pada model *use case* yang telah dirancang sebelumnya menggunakan bahasa pemrograman Java.
- b. Melakukan Implementasi pada aplikasi LBS BTS CV.2K.

### **3.5. Fase *Transition***

Pada fase ini dilakukan tahap evaluasi dari aplikasi yang telah dibangun berdasarkan rancangan dan rencana yang ada. Pada fase ini dapat ditemukan kesalahan dan kekurangan yang ada pada sistem yang telah dibangun. Pada fase ini akan dilakukan kegiatan :

- a. Melakukan pengujian terhadap aplikasi LBS BTS CV.2K.
- b. Mengamati kekurangan dan kesalahan yang terdapat pada aplikasi LBS BTS CV.2K yang telah dilakukan pengujian sebelumnya.
- c. Menyimpulkan fitur-fitur tambahan guna pengembangan dari aplikasi LBS BTS CV.2K untuk kedepannya.

## BAB IV

### ANALISA DAN PERANCANGAN

Bab ini merupakan bagian dari fase *inception* dan fase *elaboration*, pada Bab ini akan dilakukan analisa dan pembuatan deskripsi arsitektur yang dibutuhkan aplikasi sebagai bagian dari fase *inception*, dan kemudian dilakukan pembuatan *Unified Modelling Language* (UML), struktur menu aplikasi yang akan dibangun, dan perancangan *prototype* antarmuka aplikasi yang akan dibangun sebagai bagian dari fase *elaboration*.

#### 4.1 Analisis Permasalahan

*Base Tower Station*(BTS) merupakan sarana yang penting dalam bidang telekomunikasi, terutama dalam mengantarkan sinyal ke tempat tujuan yang diinginkan. Seperti CV.2K yang memiliki beberapa BTS yang tersebar di kota Pekanbaru. BTS CV.2K berguna sebagai *repeater* untuk mengantarkan sinyal ke seluruh client mereka yang tersebar di kota Pekanbaru.

Hal ini menyebabkan penting bagi CV.2K untuk selalu mengontrol dan merawat (*maintenance*) BTS milik CV.2K agar kualitas sinyal jaringan mereka tetap terjaga dengan baik. Untuk ini dibutuhkan sebuah aplikasi *Location Base Services* (LBS) yang berjalan pada *platform* Android yang dapat memetakan BTS CV.2K tersebut. Aplikasi ini akan digunakan oleh teknisi CV.2K yang dapat membantu teknisi menemukan lokasi BTS dan mendapatkan jalur terpendek yang akan dilalui apabila terjadi kerusakan dan melakukan perawatan pada BTS CV.2K.

Pada dasarnya teknologi *mobile* atau komputasi bergerak memberikan berbagai dampak keuntungan dan kerugian dari pemakaian *mobile computing* tersebut.

1. Keuntungan *mobile computing* diantaranya adalah :
  - a. *Extreme personalitation*
  - b. Pengaksesan informasi setiap saat dan dimanapun
  - c. Mobilitas tinggi tanpa kerumitan kabel

- d. Kompatible yang tinggi dengan teknologi lain
- 2. Kekurangan *mobile computing* diantaranya adalah :
  - a. Kemanan atau *security* terhadap perangkat mobile
  - b. Sensitifitas terhadap cuaca
  - c. *Interferences*

Aplikasi yang akan dibangun menggunakan sebuah peta digital yang bersumber dari *googlemap* yang terintegrasi dan dikoneksikan dengan bahasa pemrograman PHP. Keterbatasan yang dimiliki dalam penggunaan peta digital ini diantaranya belum terdatanya BTS CV.2K yang ada di kota Pekanbaru, dan keakuratan titik koordinat yang diberikan. Oleh karena itu dalam mendapatkan nilai valid koordinat yang diinginkan dilakukan secara manual dengan menggunakan perangkat android.

Berdasarkan latar belakang yang ada, dalam melakukan proses penentuan jalur atau *direction* pada aplikasi yang akan dibangun menggunakan algoritma *Floyd-Warshall*. Seiring kebutuhan teknisi CV.2K akan informasi Status BTS, maka sebuah pemecahan masalah yang ada dapat dikurangi dengan menggunakan bantuan sebuah aplikasi yang berjalan dalam sebuah perangkat *mobile* atau *gadget* yang dimiliki dan secara garis besar mudah dibawa oleh teknisi saat melakukan aktifitas *troubleshooting*.

#### **4.1.1. Definisi Kebutuhan Sistem**

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

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

##### **4.1.1.1. Project Plan**

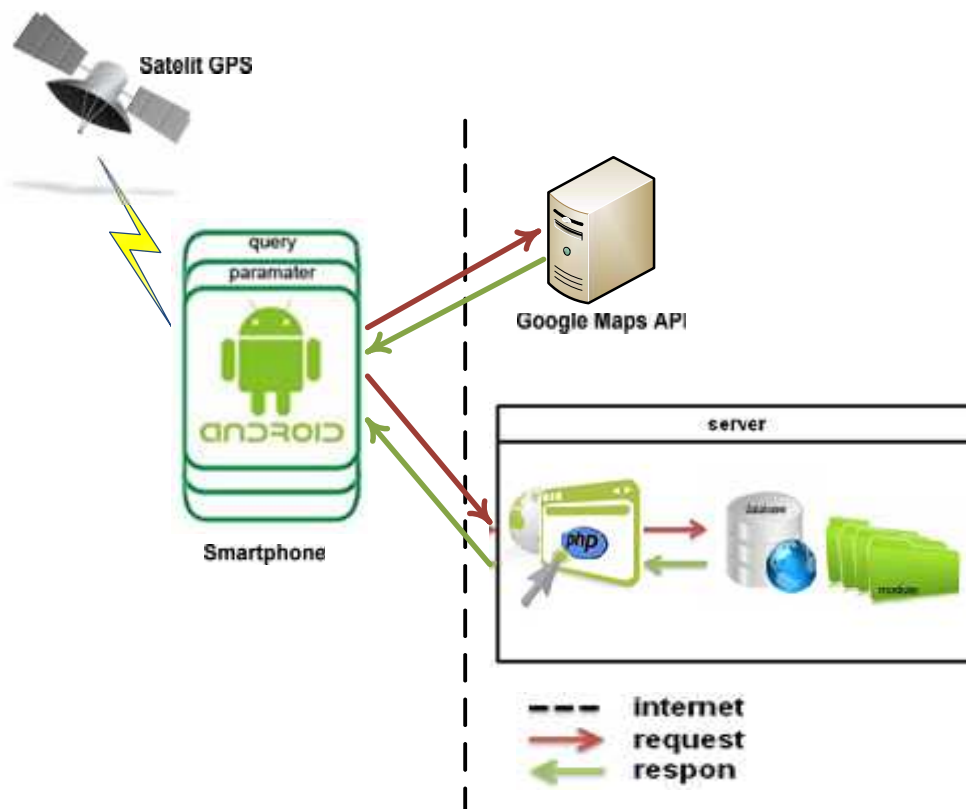
Pada *project plan* ini, akan dijelaskan mengenai gambaran umum sistem, deskripsi kebutuhan sistem, dan fungsi sistem yang nantinya akan dibuat.

### A. Deskripsi Umum Sistem

Aplikasi LBS BTS CV. 2K dengan pencarian rute terpendek menggunakan Algoritma *Floyd-Warshall* yang dibahas dalam penelitian ini adalah aplikasi yang berbasis *client-server*, dan akan dijalankan pada perangkat *smartphone* dengan sistem operasi Android. Aplikasi ini menggunakan bahasa pemrograman PHP sebagai penghubung dalam pengiriman *request* dan penerimaan respon terhadap *server* yang menggunakan *database* MySQL.

Aplikasi *client* yang akan dibangun merupakan aplikasi yang mampu *request* untuk menampilkan lokasi fasilitas umum dari *server* yang meliputi *database* melalui bahasa pemrograman berbasis *web*, kemudian *server* akan memberikan respon kepada *client* melalui jalur pengiriman *request* sebelumnya.

Gambaran umum pencarian rute terpendek menggunakan Algoritma *Floyd-Warshall* ini bertujuan memberikan penjelasan mengenai struktur menu dan konsep dasar aplikasi. Deskripsi arsitektur sistem ini dapat di lihat pada Gambar 4.1. dibawah ini.



Gambar 4.1 Gambaran Umum Sistem

Dari Gambar 4.1 diatas dapat dilihat proses kerja aplikasi yang akan dibuat, ada tiga bagian penting yang saling terhubung dalam kerja sistemnya, diantaranya:

1. *Mobile Device* (Android) merupakan perangkat tempat berjalannya aplikasi LBS BTS CV.2K. Dari perangkat inilah pengguna berinteraksi dengan sistem dengan memanfaatkan jaringan internet *mobile* untuk mengakses informasi yang diinginkan oleh pengguna (teknisi CV.2K).
2. *Database Server*, terdiri dari dua bagian, yaitu:
  - a. Penghubung (PHP), berfungsi sebagai koneksi antara sistem yang berjalan pada perangkat Android (*client*) dan *database*. Peran penghubung sangat penting, karena sisi *client* tidak bisa langsung mengakses *database* tanpa perantara. Penghubung ini yang bertugas mengirimkan *request* dan respon antara *client* dan *server*.
  - b. *Database*, merupakan bagian yang berfungsi sebagai *database* dari aplikasi LBS BTS CV.2K. *Database* ini yang bertanggung jawab memberikan respon sesuai *request* dari *client*. *Database* yang digunakan adalah MySQL.
3. Penentu lokasi, yang terdiri dari 2 bagian, yaitu:
  - a. GPS, merupakan perangkat yang ditanamkan ke *smartphone android* untuk bisa mengirimkan *request* data berupa lokasi dan posisi pengguna berdasarkan koordinat *lattitude* dan *longitude*.
  - b. Satelit, merupakan perangkat yang memetakan serta memberikan respon berdasarkan permintaan GPS pada perangkat *smartphone android*.

*User* langsung dapat mendapatkan hak akses aplikasi jika aplikasi ini telah dipasangkan pada perangkat Android. Aplikasi ini tidak membatasi hak akses.

## **B. Deskripsi Kebutuhan Sistem**

Untuk membangun sebuah system yang efisien, kita harus memperhatikan kebutuhan system. Dengan mengetahui kebutuhan system akan membantu dalam pembangunan system tersebut.

## 1. Sistem Yang Akan Dibangun

Untuk kebutuhan sistem yang akan dibangun terdapat dua bagian yakni dari kebutuhan sistem dari perangkat *Android* dan kebutuhan sistem dari sisi *server database*.

### a. Sistem Pada Perangkat Android

1. Bahasa pemrograman yang digunakan adalah *Java*.
2. Saat aplikasi dibuka, halaman yang pertama muncul adalah *Home*, menu *Home* berfungsi untuk menampilkan lokasi-lokasi BTS CV.2K yang ada di kota Pekanbaru. Selain menu *Home*, ada 3 menu lain yang terdapat pada aplikasi ini, yaitu *BTS*, *Maintenance*, dan *About*.
3. Pengguna dapat memilih BTS mana yang akan dicari rute terpendeknya menggunakan menu *BTS*, pada menu *BTS* terdapat list BTS yang dimiliki oleh CV.2K.
4. Untuk melihat status BTS, pengguna bisa melihatnya pada menu *Maintenance*, selain itu menu ini juga ada tombol map yang berfungsi untuk mencari rute terpendek untuk menuju lokasi BTS yang terdapat dalam menu *Maintenance*.
5. Untuk mengetahui informasi mengenai aplikasi LBS BTS dengan menerapkan Algoritma *Floyd-Warshall*, pengguna dapat melihat menu *About*.

### b. Connector

*Connector* ini berfungsi sebagai penghubung antara system yang berjalan di perangkat *Android* dengan *database*. *Connector* dibangun menggunakan bahasa pemrograman *PHP*. Berikut beberapa *connector* yang akan di bangun:

1. *Connector floydw*, menangani *request* untuk pencarian rute terpendek yang dikirimkan oleh pengguna, kemudian melakukan pencarian rute terpendek berdasarkan data yang ada di *server database*, kemudian mengirimkan respon dari *database* ke aplikasi.
2. *Connector cek\_bts*, menangani *request* untuk menampilkan list BTS tujuan yang terdapat di *server database*.

3. *Connector* koordinat\_lokasi, bagian ini menangani *request* dari pengguna untuk menampilkan seluruh lokasi BTS yang ada di *server database* ke menu *Home*, sehingga pengguna bisa melihat seluruh *marker* lokasi BTS yang ada di *server database* di peta pada perangkat android pengguna.
4. *Connector* maintenance, bagian ini menangani *request* untuk menampilkan status BTS yang terdapat di *server database*.
5. *Connector* status\_bts, bagian ini menangani *update* Status Bts dari aplikasi ke *server database*.

## 2. Performansi Aplikasi

Aplikasi LBS BTS ini merupakan aplikasi yang beroperasi di lingkungan perangkat pintar bersistem operasi Android. Terdapat beberapa keterbatasan yang ditemui pada perangkat ini, sehingga perlu diperhatikan untuk menjadi acuan dalam pengembangan aplikasi ini, yaitu di antaranya:

1. Sumber daya yang terbatas, hingga saat ini perangkat Android yang banyak beredar memiliki kapasitas memori terbatas. Adapun yang tertinggi saat ini adalah 512Mb.
2. Tampilan antar muka dengan pengguna sangat berpengaruh terhadap waktu tunggu aplikasi hingga aplikasi benar-benar siap digunakan, semakin banyak komponen yang digunakan akan semakin lama pula waktu tunggu yang dibutuhkan.

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

1. Merancang aplikasi yang menggunakan memori seefektif mungkin, sehingga tidak mengganggu siklus operasi Android dan aplikasi lain.
2. Merancang aplikasi dengan antarmuka yang sederhana namun tetap menarik dan ramah bagi pengguna.
3. Merancang aplikasi yang memiliki fitur menu yang hanya berkaitan dengan tujuan utama dibuatnya aplikasi LBS BTS.

### **C. Fungsi Sistem**

Secara umum fungsi system ada dua bagian yaitu system yang akan dibangun dari sisi perangkat Android dan media penghubung.

#### **a. Fungsi Sistem dari Sisi Perangkat Android**

Sistem yang akan dibangun dari sisi perangkat Android memiliki fungsi-fungsi sebagai berikut:

1. Menampilkan lokasi BTS yang telah ditandai di peta.
2. Menampilkan lokasi pengguna (*realtime*) yang di tampilkan secara bersamaan dengan lokasi-lokasi BTS yang ada.
3. Mampu melakukan pencarian lokasi tujuan yang terdapat di *server database*.
4. Mampu menampilkan Status BTS dan meng-*update* status BTS dari perangkat *android*.
5. Mampu menampilkan rute terpendek yang telah ditandai di peta.

#### **b. Fungsi Media Penghubung**

Mesin penghubung berfungsi untuk menerima *request* dari perangkat *android* pengguna dan meneruskannya ke *server database* dan kemudian mengolahnya serta mengirimkan lagi hasilnya ke perangkat *android*.



## **4.1.2 Algoritma *Floyd Warshall***

### **4.1.2.1 Langkah Kerja Algoritma *Floyd Warshall***

Algoritma ini mencari rute terpendek dari node asal ke node tujuan dalam sebuah graf dengan cara membandingkan semua kemungkinan lintasan (rute) pada graf untuk setiap sisi dari semua simpul. Langkah-langkah dalam menentukan rute terpendek pada algoritma *floyd warshall* yaitu :

1. Pertama persoalan dibagi atas beberapa tahap dan buat *flowchart* untuk memudahkan pencarian rute terpendek.
2. Ketika masuk ke suatu tahap, hasil pada tahap tersebut akan menjadi simpul baru untuk tahap selanjutnya.
3. Tentukan 1 titik sebagai titik awal agar pencarian algoritma dapat dilakukan.
4. Cari node yang bertetangga langsung dengan titik simpul(titik awal).
5. Bandingkan rute tiap tahap yang bobotnya sudah dijumlahkan dengan bobot pada tahap sebelumnya, cari rute dengan bobot yang terkecil sampai proses pencarian berakhir.
6. Bobot yang dimiliki oleh suatu tahap akan dijumlahkan dengan bobot yang ada pada tahap-tahap sebelumnya seiring dengan bertambahnya jumlah tahapan.
7. Pencarian berhenti apabila node tujuan telah ditemukan.
8. Setelah proses selesai, lihat ada berapa rute yang diperoleh untuk ke suatu tujuan tertentu dan pilih rute dengan jumlah bobot yang paling kecil untuk menjadi rute terpendek dari algoritma *floyd-warshall*.

### **4.1.2.2 Perhitungan Manual Algoritma *Floyd Warshall***

Sebagai contoh, terdapat suatu graf berbobot yang merepresentasikan kondisi keterhubungan antar BTS di Pekanbaru, dalam kasus ini kita ambil contoh seorang teknisi akan melakukan perjalanan dari BTS Hotel Ibis ke BTS Xnet (Titik U).



Gambar 4.2 Contoh *Node* Jalan Berbobot

#### 4.1.2.3 Flowchart Algoritma *Floyd-Warshall*

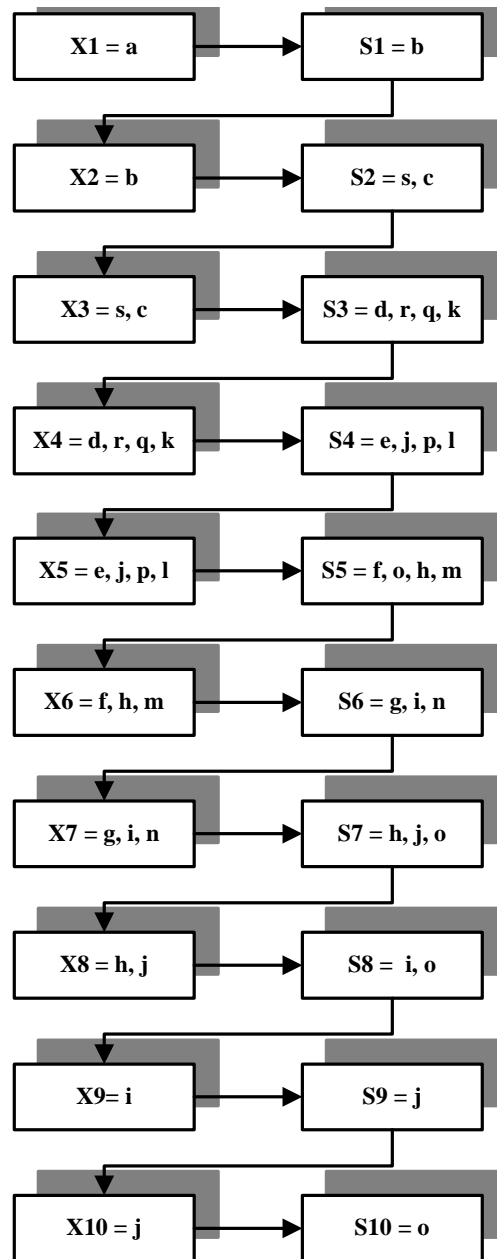
Langkah pertama adalah mengelompokkan proses pencarian setiap tahap dan mencari node yang terhubung langsung dengan titik simpul yang sedang ditinjau, berikut adalah prosesnya:

- Tahap 1: Pada tahap ini titik simpul yang sedang di tinjau adalah (a). Kemudian titik (a) memiliki 1 kandidat solusi yaitu (b).
- Tahap 2: Setelah tahap 1 selesai ditinjau, maka sekarang proses yang dilakukan ada pada tahap 2, dimana titik kandidat solusi yang ada pada tahap 1 yaitu (b) dijadikan titik simpul pada tahap 2, jadi titik simpul pada tahap 2 ini adalah (b). Titik (b) ini memiliki kandidat solusi (s) dan (c).
- Tahap 3: Setelah tahap 2 selesai ditinjau, maka sekarang proses yang dilakukan ada pada tahap 3, dimana titik kandidat solusi yang ada pada tahap 2 yaitu (s) dan (c) dijadikan titik simpul pada tahap 3. Sehingga titik simpul tahap 3 adalah (s) dan (c) memiliki kandidat solusi yaitu (d), (r), (q), dan (k).
- Tahap 4: Setelah tahap 3 selesai ditinjau, maka sekarang proses yang dilakukan ada pada tahap 4, dimana titik kandidat solusi yang ada pada tahap 3 yaitu (d), (r), (q), dan (k) dijadikan titik simpul pada tahap 4. Sehingga titik simpul tahap 4 adalah (d), (r), (q), dan (k) memiliki kandidat solusi yaitu (e), (j), (p), dan (l).
- Tahap 5: Setelah tahap 4 selesai ditinjau, maka sekarang proses yang dilakukan ada pada tahap 5, dimana titik kandidat solusi yang ada pada tahap 4 yaitu (e), (j), (p), dan (l) dijadikan titik simpul pada tahap 5. Sehingga titik simpul tahap 5 adalah (e), (j), (p), dan (l) yang memiliki kandidat solusinya adalah (f), (o), (h), dan (m)
- Tahap 6: Untuk tahap 6 sampai dengan tahap akhir, proses yang dilakukan adalah sama, untuk lebih mengetahui lebih jelas proses yang dilakukan dibawah ini adalah *flowchart* dari seluruh proses pencarian Algoritma *Floyd-Warshall*.

Keterangan gambar:

$X_i$  : Titik Simpul

$S_i$  : Kandidat Solusi



Gambar 4.3 Flowchart Perhitungan Algoritma Floyd-Warshall

#### 4.1.2.4 Analisa Algoritma Floyd-Warshall

Dimana:

$f$  : nilai jarak antar titik per-tahap

$k$  : tahap ke- $n$

$s$  : titik simpul yang sedang ditinjau

**Tahap 1:**

Titik Tujuan	Titik Asal	
s1	Solusi Terpendek	
	$fI(s)$	$xI$
b	900	a

**Penjelasan:**

Pada tahap pertama ini, algoritma melakukan proses dari titik A ke titik-titik yang saling berhubungan, dimana pada tahap ini titik **a** terhubung dengan titik **b** mempunyai bobot **900 m**. Berikut tampilan pencarian rute terpendek pada tahap 1:



Gambar 4.4 Tahap 1

**Tahap 2:**

Titik Tujuan	Titik Asal		
	<b>s2</b>	$f1(s)$	Solusi Terpendek
	<b>b</b>	$f2(s)$	$x2$
s	2600	2600	s
c	1200	1200	c

**Penjelasan:**

Pada tahap 2 ini, titik tujuan yang ada pada tahap 1, berubah menjadi titik simpul pada tahap ini. Sehingga untuk mendapatkan jumlah jarak pada tahap ini maka:

1. Jarak pada tahap 1 yaitu **a => b = 900 m;**
2. Jarak pada tahap 2 ini yaitu **b => c = 1200 m;**
3. Maka jumlah jarak yang didapatkan pada tahap ini adalah:

$$(a \Rightarrow b) + (b \Rightarrow c) = 900 \text{ m} + 1200 \text{ m} = 2100 \text{ m.}$$

Kemudian untuk jumlah jarak pada titik yang kedua, yaitu:

1. Jarak pada tahap 1 yaitu **a => b = 900 m;**
2. Jarak pada tahap 2 ini yaitu **b => s = 3500 m.**
3. Maka jumlah jarak yang didapatkan pada tahap ini adalah:

$$(a \Rightarrow b) + (b \Rightarrow s) = 900 \text{ m} + 2600 \text{ m} = 3500 \text{ m.}$$

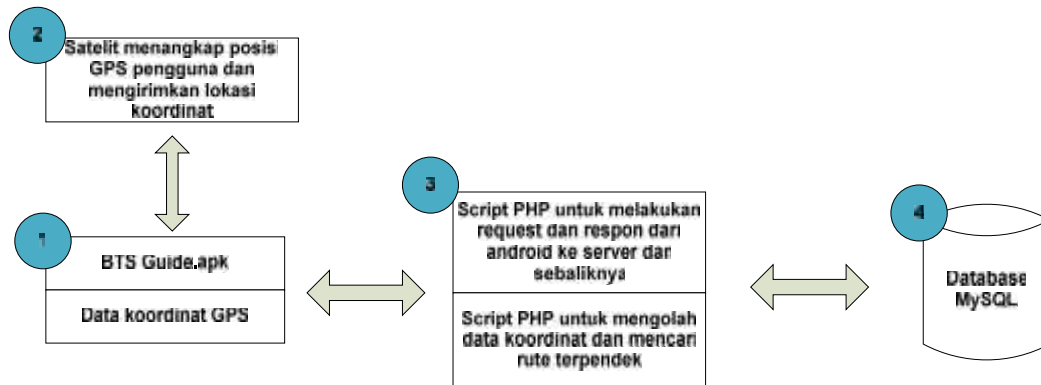
Berikut tampilan pencarian rute terpendek pada tahap 2, untuk proses selanjutnya dapat dilihat pada lampiran C:



### 4.1.3 Pembuatan Model

Pada bagian pembuatan model ini, terbagi 3 tahapan pengerjaan. Yaitu pembuatan model sistem, penjelasan deskripsi pengguna dan pemodelan UML.

#### A. Model Sistem



Gambar 4.6 Model Sistem

Objek-objek yang berperan pada model sistem diatas yaitu:

- Perangkat *android* sebagai media untuk mengakses *server* melalui aplikasi LBS BTS yang telah terinstall.
- GPS sebagai penentu posisi pengguna aplikasi yang terhubung ke saelit untuk meminta request posisi pengguna berdasarkan *longitude* dan *lattitude*, yang nantinya akan digunakan untuk diproses sebagai titik awal dalam memulai navigasi.
- Script* PHP berperan sebagai media untuk *request* dan respon dari perangkat *android* ke *database* dan sebaliknya serta untuk memproses pencarian rute terpendek dengan menggunakan data yang ada di *database*.
- Database* MySQL berperan sebagai tempat penyimpanan data yang dibutuhkan oleh aplikasi serta oleh script php untuk memproses pencarian rute terpendeknya.

#### B. Deskripsi Pengguna

Pengguna dari sistem ini adalah pengguna yang dalam hal ini sebagian besar adalah teknisi CV.2K, pengguna diberikan akses penuh terhadap semua



menu dan fungsi yang ada pada aplikasi ini. Untuk lebih jelasnya dapat di lihat di tabel 4.1. di bawah ini.

Tabel 4.1 Deskripsi pengguna

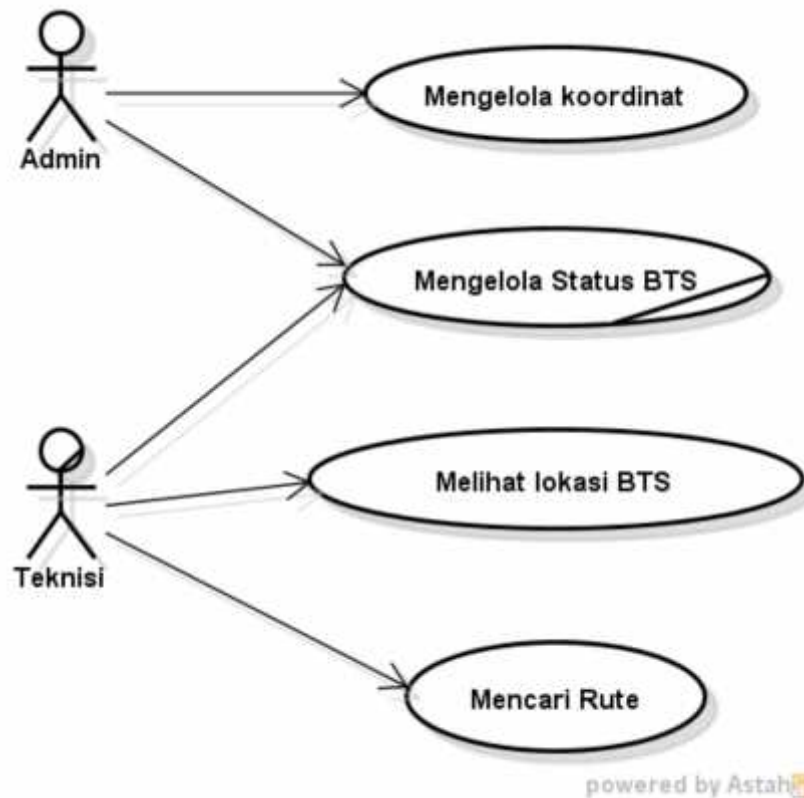
No	Kategori Pengguna	Hak Akses	Keterangan
1	Teknisi (Perangkat <i>Android</i> )	a. Melakukan navigasi ke BTS tujuan yang disertai dengan informasi jarak yang diawali dari posisi teknisi saat menggunakan aplikasi b. Melakukan pencarian lokasi BTS tujuan c. Melihat lokasi-lokasi BTS tujuan yang tertera di peta d. Mampu untuk men- <i>update</i> Status BTS dari aplikasi LBS BTS.	Hak akses penuh
2	Admin	a. Melakukan update <i>database</i> , baik <i>database</i> koordinat jalan maupun koordinat lokasi dan Status BTS terkini	Hak akses penuh

### C. Pemodelan UML (*Unified Modeling Language*)

Analisa perancangan pada sistem ini menggunakan UML, yaitu terdiri dari *usecase diagram*, *class diagram*, *activity diagram*, *sequence diagram* dan *deployment diagram*. Untuk lebih jelasnya dapat dilihat pada lampiran A.

#### 1. *Usecase Diagram*

*Usecase diagram* merupakan suatu aktivitas yang menggambarkan urutan interaksi antar satu atau lebih aktor dan sistem. *Usecase* yang akan dirancang yaitu *usecase diagram* untuk pengaksesan melalui perangkat *Android*. Gambar 4.7 dibawah ini menjelaskan aliran *usecase diagram* pengaksesan melalui perangkat *Android*.



Gambar 4.7. Usecase Diagram

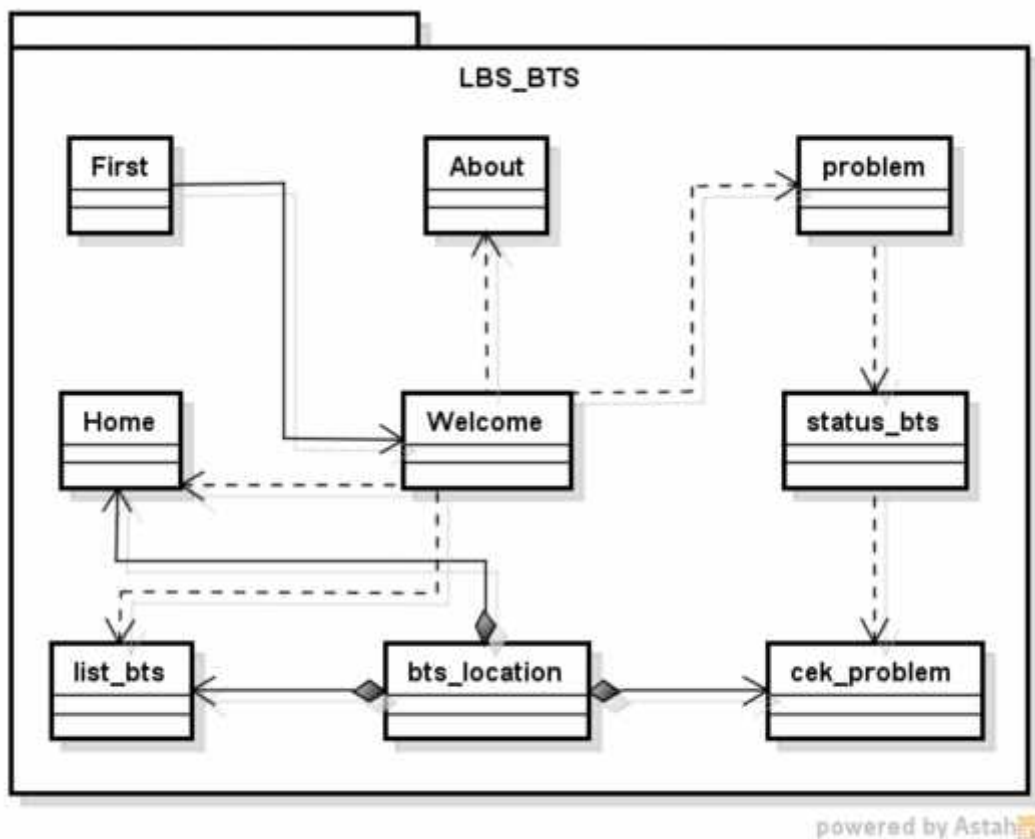
Dari gambar 4.7 dapat dilihat bahwa sistem ini terdiri dari 2 aktor dan 4 *usecase*. Untuk lebih jelasnya, spesifikasi dari *usecase diagram* (pengaksesan melalui perangkat Android) dapat dilihat pada tabel berikut ini.

Tabel 4.2. Spesifikasi *usecase diagram*

No.	Aktor	Nama Usecase	Deskripsi
1.	Admin <i>Database</i>	Login	Proses autentifikasi untuk masuk ke <i>database server</i>
		Melakukan <i>update database</i>	Menambah koordinat jalan dan <i>update database</i> Status BTS.
2.	User (teknisi)	Mengelola Status BTS	Proses melakukan <i>update database</i> Status BTS.
		Pengecekan Status BTS	Proses untuk mengecek Status BTS.
		Melihat Lokasi BTS	Proses untuk menavigasi peta dan melihat lokasi-lokasi BTS yang ditandai di peta
		Mencari Route	Proses menampilkan jalur terpendek ke BTS tujuan.

## 2. Class Diagram

*Class Diagram* menggambarkan struktur dan deskripsi *class*, *package*, dan objek yang saling terhubung. *Class diagram* yang dijelaskan pada analisa ini adalah *class diagram* sistem yang terpasang pada perangkat *android*. Gambar 4.8 di bawah ini menjelaskan *class diagram* sistem yang terpasang pada perangkat Android.



Gambar 4.8. *Class Diagram* LBS BTS

Tabel 4.8 berikut merupakan penjelasan mengenai atribut dan *method* dari masing-masing *class* yang ada.

Tabel 4.3. Detil *Class Diagram*

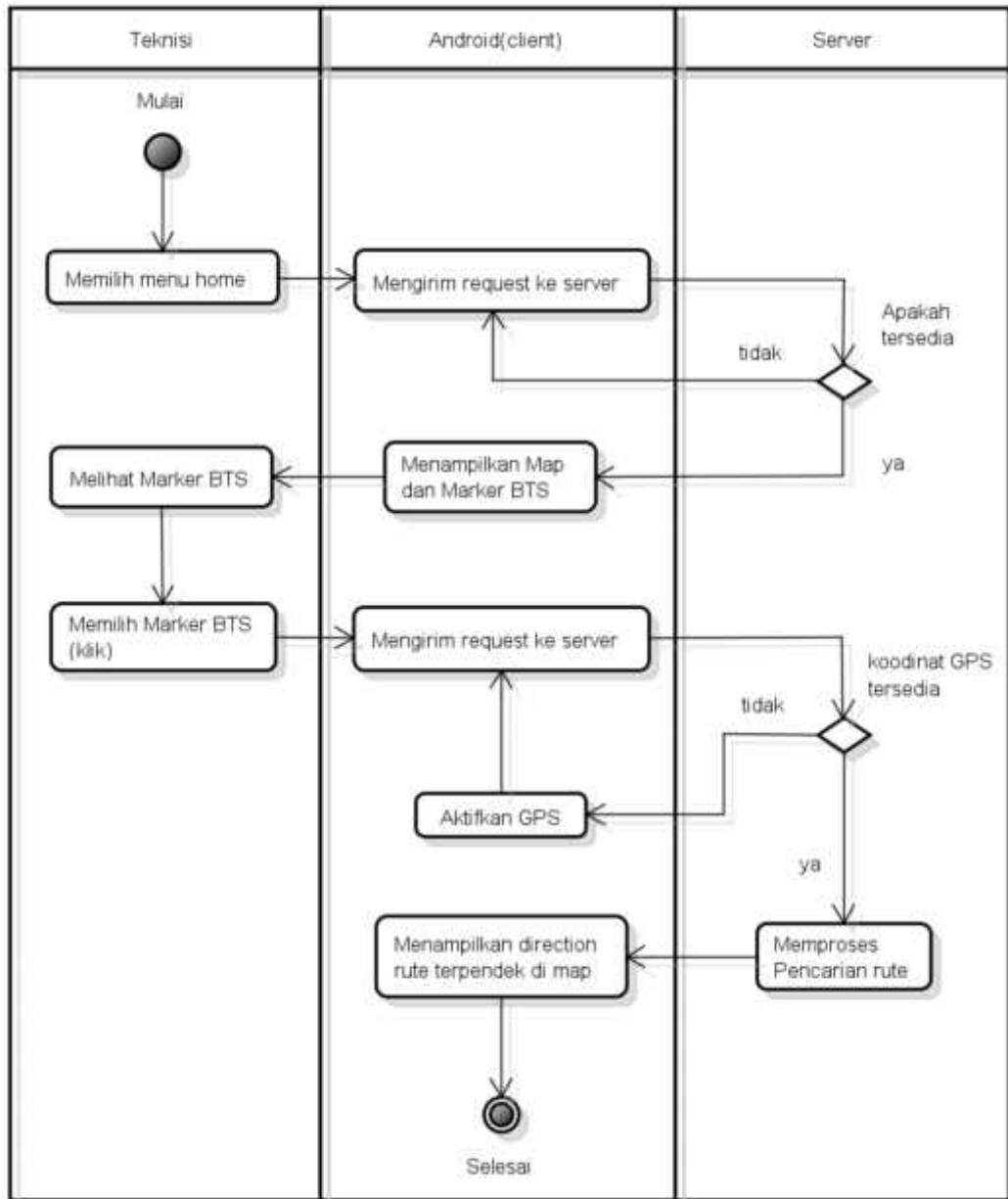
No	Nama Class	Atribut	Method
1	First		onCreate(Bundle)
2	Welcome		onCreate(Bundle)
			onPause()
			onResume()
			startAnimasi()

3	About		onCreate(Bundle)
4	Home	data : Int kategori : String keterangan : String titik : Double titik_akhir : String	Request(HttpResponse) getRequest(String) isRouteDisplayed() onCreate(Bundle) onCreateOptionsMenu(Menu)
5	List_bts	kategori2 : String lokasi : ListView nama_lokasi : String nama_pilihan : TextView titik_akhir : String	onOptionsItemSelected(MenuItem) Request(HttpResponse) getRequest(String) onCreate(Bundle)
6	Problem	bts : ListView nama_bts : String	Request(HttpResponse) onCreate(Bundle)
7	Cek_problem	gbr : ImageView ket : String keter : TextView nama : TextView	Request(HttpResponse) onCreate(Bundle)
8	Btslocation	data : int drawable : Drawable drawable2 : Drawable jarak : String titik_akhir : double titik_awal : double txtjarak2 : TextView	Request(HttpResponse) getRequest(String) isRouteDisplayed() onCreate(Bundle)
9	status_bts	btn_simpan : Button edt_statusbts : EditText extras : Bundle nama_bts : String status_bts : String txt_namabts : TextView	Request(HttpResponse) getRequest(String) onClick(View) onCreate(Bundle)

### c. Activity Diagram

*Activity diagram* merupakan alur kerja pada setiap *usecase*. *Activity diagram* pada analisa ini mencakup *activity diagram* setiap *usecase*. Untuk memudahkan dalam perancangan *activity diagram* maka dalam aplikasi LBS BTS ini akan dipecah menjadi beberapa bagian.

Gambar 4.9 berikut ini menjelaskan *activity* untuk melihat lokasi BTS pada aplikasi Android. Untuk *activity diagram* lainnya dapat dilihat pada Lampiran A.



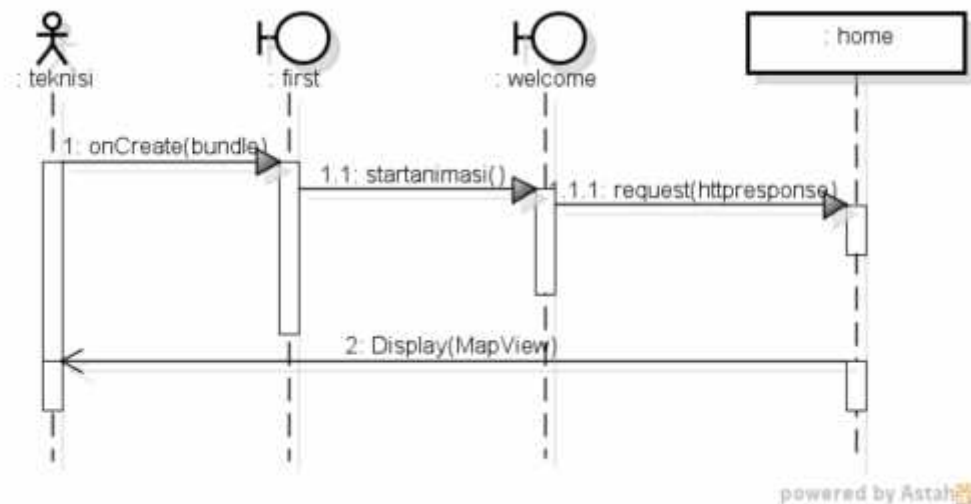
powered by Astah

Gambar 4.9 Activity Diagram melihat lokasi BTS

#### d. Sequence Diagram

*Sequence diagram* adalah representasi dari interaksi-interaksi objek yang berjalan pada system. Dengan menggunakan *sequence diagram* kita dapat melihat bagaimana objek-objek bekerja pada aplikasi. *Sequence diagram* dapat menampilkan bagaimana sistem merespon setiap kejadian atau permintaan dari user, dapat mempertahankan integritas internal, bagaimana data dipindahkan dari server ke *user interface* dan bagaimana objek-objek diciptakan dan dimanipulasi.

Gambar berikut ini akan menjelaskan mengenai *sequence diagram* untuk menampilkan home. Penjelasan mengenai *sequence diagram* yang lainnya pada aplikasi android dapat dilihat pada lampiran A.



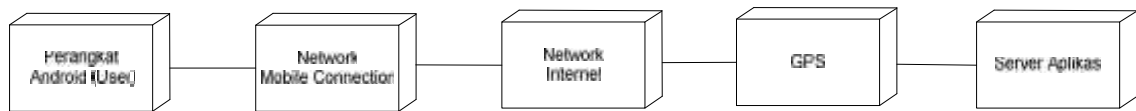
Gambar 4.10 *Sequence Diagram* untuk melihat lokasi BTS

Pada gambar 4.10 tersebut, pengguna memulai untuk membuka aplikasi LBS BTS dan akan ditampilkan halaman home, kemudian akan ditampilkan beberapa BTS yang tersedia untuk dipilih dan akan ditampilkan arah yang optimal sesuai dengan pencarian menggunakan metode *Floyd-Warshall* yang terdapat pada aplikasi LBS BTS.

#### 5. Deployment Diagram

*Deploy diagram* menggambarkan detail bagaimana komponen di *deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras apa).

Aplikasi diterapkan pada perangkat *android* yang menggunakan koneksi internet untuk bisa mengakses *server* aplikasi dan perangkat GPS untuk menentukan lokasi pengguna sehingga *deploy diagram* yang dihasilkan dapat dilihat pada Gambar 4.11 dibawah ini.



Gambar 4.11 *Deployment Diagram*

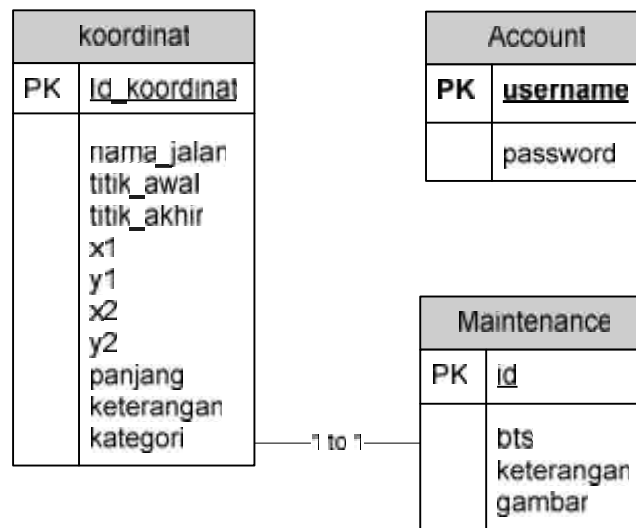
## 4.2. Perancangan

Tahap perancangan merupakan tahap dimana perancangan aplikasi mulai dilakukan. Perancangan tersebut antara lain perancangan perancangan *graphic user interface* untuk admin *database*, perancangan *database* aplikasi struktur menu, dan perancangan *graphic user interface* aplikasi di perangkat *android*.

Perancangan *interface* aplikasi di *android* ditujukan kepada pengguna aplikasi, sedangkan *interface* admin *database* ditujukan untuk admin *database* yang akan melakukan *update* status BTS, serta *update* koordinat lokasi BTS dan koordinat jalan.

### 4.2.1. Perancangan Database

Aplikasi LBS BTS merupakan aplikasi mobile berbasis *client server*, dan memerlukan sebuah database menjalankan fungsinya. Perancangan database pada aplikasi *LBS BTS* dirancang dengan satu database utama yaitu database server, pada gambar 4.12 berikut adalah model logic database server pada aplikasi *LBS BTS*.

Gambar 4.12 Model *Logic Database Server*

Tabel 4.4. Keterangan Atribut dari tabel koordinat

No	Nama Field	Tipe Data	Null	Keterangan
1	No (PK)	int(100)	Not Null	id koordinat jalan
2	nama_jalan	varchar(100)	Not Null	nama jalan
3	titik_awal	int(100)	Not Null	titik awal graf berarah
4	titik_akhir	int(100)	Not Null	titik akhir graf berarah
5	x1	varchar(100)	Not Null	Koordinat Longitude titik awal graf berarah
6	y1	varchar(100)	Not Null	Koordinat Lattitude titik awal graff berarah
7	x2	varchar(100)	Not Null	Koordinat Longitude titik akhir graf berarah
8	y2	varchar(100)	Not Null	Koordinat Lattitude titik akhir graf berarah
9	panjang	varchar(100)	Not Null	Panjang antara titik awal dan titik akhir dari graf berarah
10	keterangan	varchar(100)	Null	Keterangan dari titik akhir graf berarah yg berupa nama jalan atau nama simpang
11	kategori	varchar(100)	Null	Nama Kategori



Tabel 4.5. Keterangan Atribut dari tabel maintenance

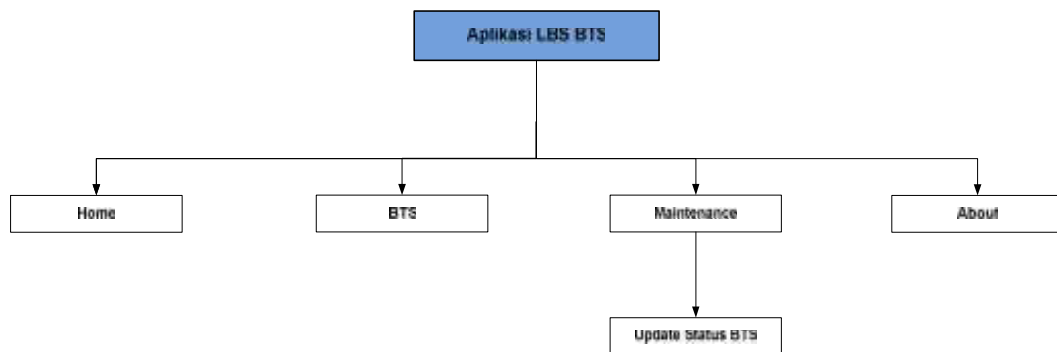
No	Nama Field	Type Data	Null	Keterangan
1	No (PK)	int(11)	Not Null	id BTS
2	bts	varchar(100)	Not Null	nama BTS
3	keterangan	varchar(100)	Not Null	status BTS
4	gambar	varchar(100)	Not Null	Nama Url gambar BTS

Tabel 4.6. Keterangan Atribut dari tabel Account

No	Nama Field	Type Data	Null	Keterangan
	username (PK)	Varchar(100)	Not Null	Merupakan username yang digunakan untuk login mengakses antarmuka <i>database</i> aplikasi
2	password	Varchar(100)	Not Null	Kata kunci untuk tiap username yang digunakan

#### 4.2.2. Perancangan Struktur Menu Sistem

Rancangan struktur menu merupakan tahapan untuk merancang bagaimana struktur menu yang akan dibangun. Berikut struktur menu dari sistem yang akan dibangun dapat dilihat pada gambar 4.13 berikut ini.



Gambar 4.13 Rancangan Struktur Menu

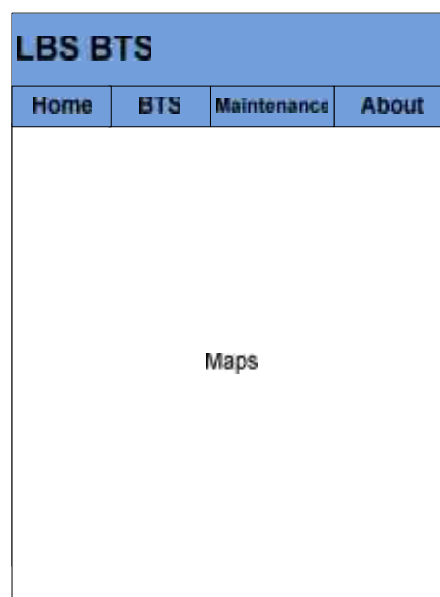
#### 4.2.3 Perancangan *Interface* Pengguna Sistem

Rancangan *interface* pengguna sistem berfungsi sebagai landasan awal dalam merancang tampilan *interface* sistem. Pada analisa dan perancangan ini, *interface* untuk pengguna sistem pada perangkat Android antara lain, *interface* Home, *interface* BTS, *interface* Maintenance, dan *interface* about. Berikut adalah

tampilan *interface* aplikasi LBS BTS yang berjalan pada sistem operasi android. Untuk perancangan antarmuka yang lebih lengkapnya dapat dilihat pada Lampiran B.

#### 4.2.3.1 Perancangan *Interface Home* di Android

Gambar 4.14 berikut ini menjelaskan perancangan *interface Home* aplikasi LBS BTS pada perangkat Android.



Gambar 4.14 Perancangan *interface Home* pada perangkat Android

Tabel 4.7. Deskripsi *interface Home* pada perangkat Android

No.	Nama Item	Deskripsi
1	LBS BTS	Merupakan sebuah <i>widget</i> berupa <i>TextView</i>
2	<i>Home</i>	Merupakan sebuah <i>widget</i> berupa <i>Button</i>
3	<i>BTS</i>	Merupakan sebuah <i>widget</i> berupa <i>Button</i>
4	<i>Maintenance</i>	Merupakan sebuah <i>widget</i> berupa <i>Button</i>
5	<i>About</i>	Merupakan sebuah <i>widget</i> berupa <i>Button</i>
6	<i>Maps</i>	Merupakan sebuah <i>widget</i> berupa <i>MapView</i>

## **BAB V**

### **IMPLEMENTASI DAN PENGUJIAN**

Bab ini merupakan bagian dari fase *construction* dan fase *transition*, dimana telah dilakukan pengkodean aplikasi, dan akan dilakukan implementasi aplikasi dan pengujian fungsi-fungsi aplikasi dengan metode *Blackbox* sebagai bagian dari fase *construction*, dan akan dilakukan pengujian aplikasi terhadap pengguna, kemudian akan dilakukan pengamatan dari hasil pengujian tersebut untuk mengetahui kekurangan aplikasi dan kemudian dilakukan pengambilan kesimpulan sebagai bagian dari fase *transition*.

#### **5.1. Implementasi**

Tahapan ini merupakan tahap dimana aplikasi akan dibuat setelah melakukan analisa dan perancangan pada bab sebelumnya. Implementasi ini mencakup pembuatan aplikasi, pembuatan *user interface* untuk admin *database*, pembuatan *database* aplikasi, serta penerapan dari algoritma *Floyd-Warshall*.

##### **5.1.1. Pembuatan Aplikasi**

###### **5.1.1.1. Pengkodean**

Pada tahap ini, akan dibagi menjadi 2 tahap, yaitu tahap pembuatan aplikasi dan tahap implementasi.

###### **A. Pembuatan Aplikasi**

Tahapan ini merupakan tahapan dimana aplikasi yang telah dirancang, dianalisa, akan dibangun, lalu diuji kelayakannya untuk selanjutnya dioperasikan sebagaimana mestinya sesuai dengan fungsi dan kelayakannya. Berikut ini akan dijelaskan beberapa hal yang berhubungan dengan pembuatan aplikasi berdasarkan analisis dan perancangan yang telah dilakukan terhadap aplikasi LBS BTS berbasis *client-server* pada sistem operasi Android ini.

## 1. Lingkungan Pengembangan

Perangkat keras dan perangkat lunak merupakan hal yang selalu dibutuhkan untuk mengimplementasikan rancangan yang telah dibuat sebelumnya. Berikut ini akan dijelaskan mengenai lingkungan implementasi yang meliputi lingkungan pada perangkat keras dan perangkat lunak.

### a. Spesifikasi Perangkat Keras

Berdasarkan rancangan yang ada, maka untuk membuat aplikasi LBS BTS dibutuhkan beberapa perangkat keras. beberapa perangkat keras pendukung tersebut antara lain adalah:

#### 1. Telepon Genggam (*Handphone*)

Perangkat ini digunakan untuk menjalankan aplikasi LBS BTS dari sisi pengguna. Perangkat yang digunakan yaitu *Sony Xperia Mini Pro* dengan spesifikasi:

- a. Sistem Operasi : Android versi 4.0.4 (*Ice Cream Sandwich*)
- b. CPU : 1 GHz *Processor*
- c. Ruang Penyimpanan *external* : 8 GB
- d. *Memory* : 400 MB *internal*, 512 MB RAM
- e. Dimensi Layar : 320 x 480 px
- f. Konektivitas : GSM 850 / 900 / 1800 / 1900 & HSDPA 900 / 2100

#### 2. 1 unit komputer sebagai *server database* dengan spesifikasi sebagai berikut:

- a. *Processor* : AMD Athlon II
- b. *Memory* : 2 GB
- c. *Hardisk* : 512 GB SATA



4. Tahap pembuatan aplikasi menggunakan *IDE Eclipse Galileo*, Android SDK version 8 dengan dukungan *Google APIs 8* dan perangkat *Sony Xperia Mini Pro*.

### 5.1.1.2. Implementasi

#### A. Tahap-tahap Implementasi

Pada tahap-tahap implementasi ini akan dijelaskan bagaimana tahapan yang penulis lakukan dalam implementasi aplikasi yang telah dibangun. Adapun tahap-tahap yang dilakukan adalah:

##### a. Instalasi Penghubung

Seperti yang telah dijelaskan pada bab analisa dan perancangan, penghubung mempunyai peran penting untuk aplikasi LBS BTS pada sistem operasi Android ini. Penghubung ini berperan sebagai media antara aplikasi di Android dengan *database server*.

Penghubung sendiri dibangun menggunakan bahasa pemrograman PHP, secara umum penghubung ini berisi baris kode yang bekerja meneruskan *query* dan respon dari *client* dan *server*. Algoritma 5.1 di bawah ini merupakan penghubung koneksi yang bertugas sebagai penghubung antara PHP dan *Database Server*.

```

<?php
$link=mysql_connect("localhost", "root", "");
mysql_select_db("lbs_bts");

if(!$link){
die('Could not connect: ' . mysql_error());
}
?>

```

Gambar 5.1 Algoritma Koneksi.php

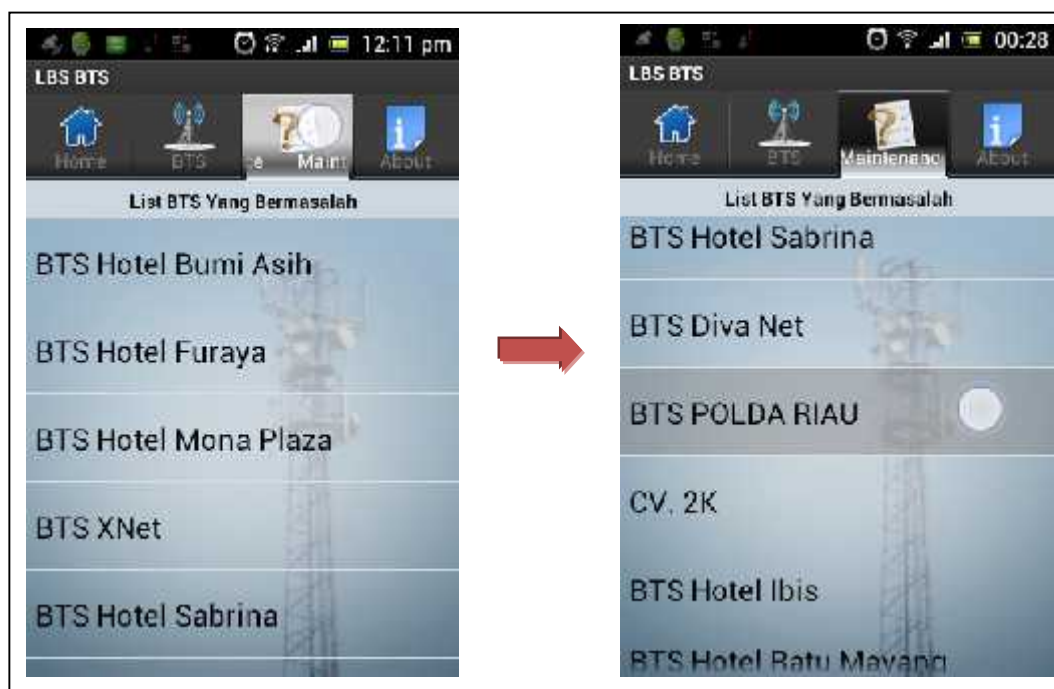
##### b. Instalasi Aplikasi *LBS BTS*

Tahap ini merupakan tahap memasang aplikasi *LBS BTS* yang telah dibangun berdasarkan analisa dan perancangan. Aplikasi dipasang pada perangkat Android. Perangkat Android yang digunakan yaitu *smartphone Sony Xperia Mini*

*Pro.* Untuk melakukan instalasi aplikasi, cukup klik aplikasi *LBS BTS* yang telah di *package* ke dalam format \*.apk (BTSGuide.apk) pada perangkat Android dan selesai. Aplikasi siap dijalankan.

## B. Implementasi Aplikasi Perangkat Android

Implementasi kali ini menggunakan perangkat dengan sistem operasi Android 4.0.4 (*Ice Cream Sandwich*). Hasil implementasi menampilkan menu *maintenance*. Dalam implementasinya, perangkat android telah mengaktifkan fitur GPS sehingga koordinat pengguna bisa didapat agar aplikasi bisa berjalan sebagaimana mestinya. Seorang teknisi ingin melakukan *maintenance*. Posisi awal teknisi berada di Jln. Yos Sudarso akan menuju BTS POLDA RIAU. Teknisi meng-klik tombol MAP, maka aplikasi akan melakukan pencarian jalur terpendek seperti gambar 5.3 Setelah *maintenance* BTS selesai, teknisi bisa langsung merubah status BTS seperti gambar 5.4 dan 5.5 di bawah ini. Hasil implementasi pada perangkat Android lainnya dapat dilihat pada lampiran D.



Gambar 5.2 Tampilan Pencarian Rute Terpendek Melalui Menu *Maintenance*



Gambar 5.3 Tampilan Hasil Pencarian Rute Terpendek Melalui Menu *Maintenance*



Gambar 5.4 Tampilan *update* Status Melalui Menu *Maintenance*





Gambar 5.5 Tampilan setelah *update* Status Melalui Menu *Maintenance*

Sistem yang dibangun berjalan dengan baik di perangkat Android, hal ini dilihat dari keberhasilan aplikasi dalam menampilkan menu *Maintenance* di perangkat Android.

### C. Implementasi Jalur Terpendek Sistem dan Penghitungan Secara Manual

Pada perbandingan jalur terpendek sistem dan penghitungan secara manual ini diambil contoh kasus seperti pada BAB IV. Terdapat suatu graf berbobot yang merepresentasikan kondisi keterhubungan antar BTS di Pekanbaru, dalam kasus ini kita ambil contoh seorang teknisi akan melakukan perjalanan dari BTS Hotel Ibis ke BTS Xnet (Titik U). Berikut tampilan pencarian rute terpendek secara manual :



Gambar 5.6 Hasil Implementasi Penghitungan Manual

Dari hasil Penghitungan diatas didapatkan 4 rute yang menuju ke BTS Xnet (o), di bawah ini akan kita bandingkan antara 4 rute tersebut yang bertujuan untuk melihat rute mana yang merupakan rute terpendek:

**Route I : a, b, s, r, j, o = 7042 m**

**Route II : a, b, c, k, l, m, n, o = 8150 m**

**Route III : a, b, s, q, p, h, i, j, o = 7482 m**

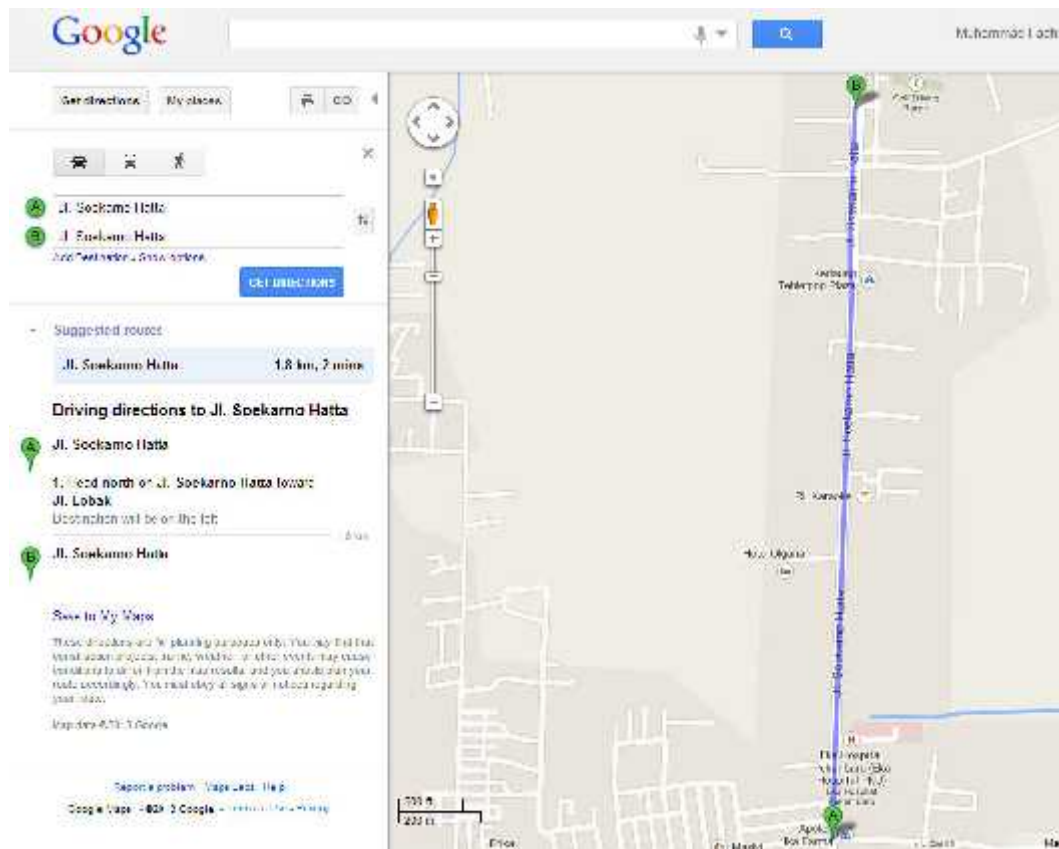
**Route IV : a, b, c, d, e, f, g, h, i, j, o = 7032 m**

Dari hasil perbandingan 4 rute diatas, maka yang merupakan rute terpendek dari BTS Hotel Ibis (a) menuju BTS Xnet (o) dengan menggunakan Algoritma *Floyd-Warshall* adalah **a, b, c, d, e, f, g, h, i, j, o** dengan total jarak **7032 Meter**. Implementasi terhadap sistem memiliki hasil yang sama dengan penghitungan manual. Berikut hasil pencarian pada Aplikasi :



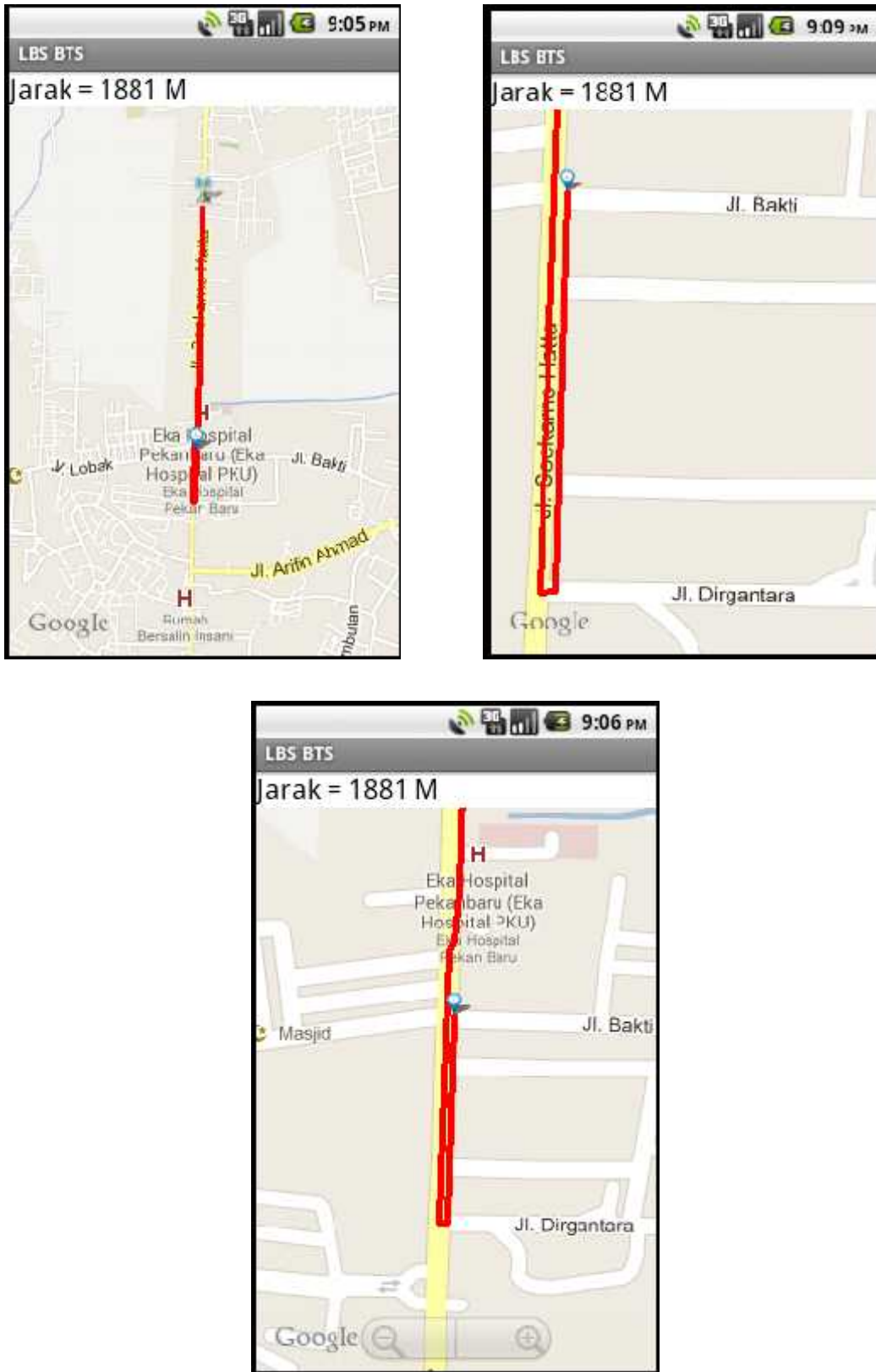
Gambar 5.7 Hasil Implementasi Rute Terpendek dari Hotel Ibis ke BTS X-net

Contoh kasus lain, dalam kasus ini kita ambil contoh seorang teknisi akan melakukan perjalanan dari jalan Bakti menuju BTS Hotel Ibis. Berikut tampilan pencarian rute terpendek menggunakan *Google Direction* :



Gambar 5.8 Hasil Implementasi *google direction*

Dari hasil pencarian rute terpendek menggunakan *google direction* didapatkan rute yang berbeda dengan rute hasil pencarian rute terpendek menggunakan aplikasi, hal ini berbeda karena database *googlemap* dan database aplikasi juga berbeda, pada *googlemap* jalan Soekarno Hatta itu masih 1 jalur, oleh karena itu dari jalan Bakti teknisi bisa langsung belok kanan menuju Hotel Ibis. Sedangkan pada database aplikasi jalan Soekarno Hatta tersebut sudah memiliki 2 jalur, dan untuk menuju ke Hotel Ibis teknisi harus belok kiri terlebih dahulu lalu memutar pada *u-turn* di jalan Dirgantara dan lurus menuju Hotel Ibis. Dari implementasi ini didapatkan hasil yang berbeda pula, jarak yang dihasilkan pada *google direction* adalah 1800 meter, sedangkan pada Aplikasi adalah 1881 meter, walaupun pada *google direction* lebih dekat jaraknya tapi pada jarak sebenarnya di lokasi jarak yang dihasilkan aplikasi adalah yang benar, karena database aplikasi sudah lebih dahulu meng-*update* data jalan Soekarno Hatta tersebut. Berikut hasil pencarian pada Aplikasi :



Gambar 5.9 Hasil Implementasi Rute Terpendek dari Jl. Bakti ke Hotel Ibis

#### D. Hasil Implementasi Antarmuka Database Aplikasi

Pada tahap implementasi ini, seluruh *connector* php telah berada di *server* dan siap untuk diakses melalui alamat url <http://192.168.175.1/bts/login.php> (login), [http://192.168.175.1/bts/olah\\_koordinat.php](http://192.168.175.1/bts/olah_koordinat.php) (tabel koordinat), dan [http://192.168.175.1/bts/olah\\_bts.php](http://192.168.175.1/bts/olah_bts.php) (tabel data BTS).

Gambar 5.9 hingga 5.11 di bawah ini merupakan tampilan dari antarmuka database aplikasi.



Gambar 5.10 Tampilan Login

No.	Nama Jalan	Titik Awal	Titik Akhir	X1	Y1	X2	Y2	Panjang	Keterangan	Kategori
1	UIN SUSKA RIAU	1	477	101.361813	0.46554	101.365349	0.46215	414	Dari UIN ke Lintas	
2	Lintas Pekanbaru Bangkinang	477	478	101.363348	0.46215	101.367318	0.464307	500	Turn right lintas p	
3	Lintas Pekanbaru Bangkinang	478	2	101.367318	0.464307	101.368285	0.464301	110	Simpang 4 paman	
4	UIN SUSKA RIAU	1	102	101.361813	0.46554	101.366611	0.468958	600	Dari UIN ke Garuda	
5	Jl. HR. Subrantas (Arah Simp. H2)	3	3	101.368285	0.464301	101.368462	0.464328	22	Point 2 Jl. HR. Sub	
6	Jl. HR. Subrantas (Arah Simp. H3)	4	4	101.368462	0.464328	101.371152	0.464307	287	U Turn 15-Jl. HR. S	
7	Jl. HR. Subrantas (Arah Simp. H4)	5	5	101.371152	0.464307	101.371152	0.464224	10	Value U Turn 15 d	
8	Jl. HR. Subrantas (Arah Simp. H5)	4	4	101.371152	0.464224	101.371152	0.464307	70	Value U Turn 15 d	
9	Jl. HR. Subrantas (Arah Simp. H4)	6	6	101.371152	0.464307	101.375192	0.464283	449	U Turn 16-Jl. HR. S	
10	Jl. HR. Subrantas (Arah Simp. H8)	7	7	101.375192	0.464283	101.375192	0.464199	70	Value U Turn 16 d	
11	Jl. HR. Subrantas (Arah Simp. H7)	6	6	101.375192	0.464199	101.375192	0.464263	10	Value U Turn 16 d	
12	Jl. HR. Subrantas (Arah Simp. H8)	8	8	101.375192	0.464283	101.378107	0.464284	325	U Turn 17 Jl. HR. S	
13	Jl. HR. Subrantas (Arah Simp. H6)	9	9	101.378107	0.464264	101.378107	0.464181	10	Value U Turn 17 d	
14	Jl. HR. Subrantas (Arah Simp. H9)	8	8	101.378107	0.464181	101.378107	0.464284	10	Value U Turn 17 d	
15	Jl. HR. Subrantas (Arah Simp. H8)	10	10	101.378107	0.464264	101.379014	0.464201	101	U Turn 18-Jl. HR. S	
16	Jl. HR. Subrantas (Arah Simp. H10)	11	11	101.379014	0.464261	101.379014	0.464175	10	Value U Turn 18 d	
17	Jl. HR. Subrantas (Arah Simp. H11)	10	10	101.379014	0.464175	101.379014	0.464201	10	Value U Turn 18 d	
18	Jl. HR. Subrantas (Arah Simp. H10)	137	137	101.379014	0.464261	101.382125	0.464254	346	Point Depan RS JIWA	
19	RS JIWA PANAM	137	138	101.382125	0.464254	101.382125	0.464677	51	Kil. JIWA PANAM	
20	RS JIWA PANAM	138	137	101.382125	0.464677	101.382125	0.464254	51	RS JIWA PANAM AF	
21	MTC GIANT	137	139	101.382125	0.464254	101.384619	0.464215	250	MTC GIANT	
22	MTC GIANT	139	12	101.384619	0.464215	101.386351	0.464181	278	U Turn 19-Jl. HR. S	
23	MTC GIANT	12	12	101.386351	0.464181	101.389351	0.464108	300	U Turn 19-Jl. HR. S	

Gambar 5.11 Tampilan Data Tabel Koordinat

No.	Bts	Keterangan	Gambar
1	BTS Hotel Bumi Asih	Status OK	HotelBumiAsih.jpg
2	BTS Hotel Furaya	Sinyal TX dan RX ke arah Perawang tidak stabil	HotelFuraya.jpg
3	BTS Hotel Mera Plaza	Status OK	HotelMera.jpg
4	BTS XNet	Status OK	x_net.jpg
5	BTS Hotel Sabrina	Status OK	hotelsabrina.jpg
6	BTS Diva Net	Status OK	divanet.jpg
7	BTS POLDA RIAU	Sinyal TX ke arah Jl. Hangtuah tidak stabil	POLDARIAU.jpg
8	CV. 2K	Status OK	ic_launcher.png
9	BTS Hotel Ibis	Rusak	hotelibis.jpg
10	BTS Hotel Ratu Mayang Garden	Paket data arah ke Simpang Tiga tidak stabil, Status sinyal OK.	hotelratumayang.jpg
11	BTS Hotel Mutiara Merdeka	Status OK	HotelMutiaraMerdeka.jpg
12	BTS Prima Net	Status OK	Primanet.jpg
13	BTS Neo Net	Status OK	neonet.jpg

Gambar 5.12 Tampilan Data Tabel Maintenance

## 5.2. Pengujian

Tahapan pengujian merupakan tahapan dimana akan dilakukan pengujian terhadap aplikasi. Pengujian dilakukan dengan metode pengujian blackbox. Selain itu, dilakukan juga pengujian terhadap akses ke aplikasi pada lokasi tertentu, jam tertentu, menggunakan provider internet tertentu, dan pengujian jalur terpendek.

### 5.2.1. Pengujian Aplikasi

Tahapan pengujian dilakukan untuk mengetahui apakah sistem yang dibangun telah sesuai dengan yang diharapkan. Tujuan utama dari pengujian sistem adalah untuk memastikan bahwa elemen-elemen atau komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan. Salah satu metode pengujian jenis ini dikenal dengan pengujian *blackbox*.

#### A. Pengujian *Blackbox* Aplikasi *LBS BTS*

Pada tahap pengujian sistem ini, perangkat keras yang digunakan yaitu *smartphone Sony Xperia Mini Pro*. Sedangkan material pengujian untuk sistem ini menggunakan data koordinat yang telah dimasukkan oleh Admin Database ke *database* pada *server*. Pengujian yang akan dilakukan adalah pengujian akses ke sistem dan pengujian fungsionalitas sistem menggunakan metode *blackbox*. Hasil dari pengujian ini dapat dilihat pada tabel 5.1.

Tabel 5.1. Hasil Pengujian *Blackbox*

No	Komponen Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Keterangan
1	Menu Home	Teknisi mengklik aplikasi di perangkat, akan muncul opening aplikasi kemudian menuju ke <i>home</i> aplikasi dan muncul lokasi BTS dikota Pekanbaru	<i>Opening</i> aplikasi berjalan dan lokasi BTS berhasil ditampilkan	Benar
2	Menu BTS	Teknisi memilih menu BTS kemudian akan muncul list	List BTS berhasil ditampilkan	Benar



		BTS		
3	Memilih 1 list BTS	Teknisi memilih salah satu BTS yang diinginkan, jika koordinat teknisi sudah di dapat GPS, maka akan ditampilkan rute terpendek ke lokasi BTS tujuan, jika tidak akan ada pemberitahuan bahwa koordinat lokasi belum didapat	Koordinat didapat dan muncul rute terpendek menuju lokasi BTS yang dipilih beserta informasi berupa nama BTS dan jarak menuju BTS tersebut	Benar
4	Menu Maintenance	Teknisi memilih menu Maintenance dan akan muncul list BTS yang bermasalah	List BTS yang bermasalah berhasil ditampilkan	Benar
5	Memilih 1 list BTS yang bermasalah	Teknisi memilih salah satu BTS yang bermasalah dan akan ditampilkan gambar lokasi BTS serta info BTS	Gambar lokasi BTS dan info BTS berhasil ditampilkan	Benar
6	Button MAP !!!	Teknisi mengklik tombol MAP. Jika koordinat teknisi sudah didapat oleh GPS maka akan ditampilkan rute terpendek menuju BTS	Koordinat didapat dan muncul rute terpendek menuju lokasi BTS yang dipilih beserta informasi berupa nama BTS dan jarak menuju BTS tersebut	Benar
9	Menu About	Teknisi memilih menu About dan akan menampilkan informasi mengenai aplikasi LBS BTS	Informasi berhasil ditampilkan	Benar
10	Button Update	Teknisi mengklik tombol	Status BTS yang baru	Benar



	Status BTS	Update Status BTS setelah mengisi Status BTS yang baru maka akan ditampilkan Status BTS yang baru	berhasil di Update	
--	------------	---	--------------------	--

### **B. Pengujian Akses Aplikasi *LBS BTS***



Pengujian yang dilakukan bertujuan untuk mengetahui proses hasil dari sistem, yaitu memperlihatkan aplikasi LBS BTS yang telah dibangun bisa diakses melalui berbagai perangkat Android dan berbagai versi Android. Hasil dari pengujian dapat dilihat halaman *home* telah tampil, dan semua menu serta fitur dapat digunakan.

Pada tabel 5.2 di bawah ini menjelaskan pengujian akses ke aplikasi LBS BTS di beberapa lokasi yang berbeda.

Tabel 5.2 Pengujian akses ke aplikasi LBS BTS di beberapa lokasi berbeda dan provider yang berbeda

No	Pukul / Tanggal	Lokasi Awal	Lokasi Tujuan	Provider	Pengujian	Hasil	Tampilan Hasil Pengujian
1	22.50 WIB / 10 Mei 2013	Jl. Yos Sudarso	BTS X-Net	Smartfren	Koordinat pengguna	Berhasil	
					Direction	Berhasil	
					Map	Berhasil	
					List BTS	Berhasil	
					List lokasi BTS	Berhasil	
2	10.25 WIB / 15 Mei 2013	Jl. Yos Sudarso	BTS Neo Net	Axiz	Koordinat pengguna	Berhasil	
					Direction	Berhasil	
					Map	Berhasil	
					List BTS	Berhasil	
					List lokasi BTS	Berhasil	

3	11.20 WIB / 9 Juni 2013	Jl. Yos Sudarso	BTS Hotel Sabrina	XL	Koordinat pengguna	Berhasil	
					Direction	Berhasil	
					Map	Berhasil	
					List BTS	Berhasil	
					List lokasi BTS	Berhasil	
4	11.00 WIB / 12 Juni 2013	Kampus UIN Panam	BTS Hotel Furaya	Telkomsel	Koordinat pengguna	Berhasil	
					Direction	Berhasil	
					Map	Berhasil	
					List BTS	Berhasil	
					List lokasi BTS	Berhasil	

5	11.00 WIB / 14 Juni 2013	Kampus UIN Panam	BTS Hotel Ibis	3	Koordinat pengguna	Berhasil	
					Direction	Berhasil	
					Map	Berhasil	
					List BTS	Berhasil	
					List lokasi BTS	Berhasil	
6	11.30 WIB / 15 Juni 2013	Kampus UIN Panam	BTS Hotel Ratu Mayang Garden	Im3	Koordinat pengguna	Berhasil	
					Direction	Berhasil	
					Map	Berhasil	
					List BTS	Berhasil	
					List lokasi BTS	Berhasil	

### C. Pengujian Jalur Terpendek Aplikasi

Tabel 5.3 Pengujian Jalur Terpendek Aplikasi Pada Beberapa Lokasi BTS

NO	ASAL	TUJUAN	RUTE	Penghitungan manual	Hasil Pencarian Sistem	Hasil
1	UIN SUSKA	BTS Hotel Ibis	Jl. Garuda Sakti - Jl. HR. Subrantas -	9.733m	9.733m	Sama (berhasil)
			Jl. Delima - Jl. Lobak - Jl. Soekarno Hatta			
			Jl. Garuda Sakti - Jl. HR. Subrantas -	10.200m		
			Jl. Soekarno Hatta			
			Jl. Garuda Sakti - Jl. HR. Subrantas -	9.900m		
			Jl. Rajawali Sakti - Jl. Melati - Jl. Lobak -			
			Jl. Soekarno Hatta			
2	UIN SUSKA	BTS H. Ratu Mayang Garden	Jl. Garuda Sakti - Jl. HR. Subrantas -	12.580m	12.580m	Sama (berhasil)
			Jl. Soekarno Hatta - Jl. Arifin Ahmad -			
			Jl. Jendral Sudirman			
			Jl. Garuda Sakti - Jl. HR. Subrantas -	13.800m		
			Jl. Adi Sucipto - Jl. Bandara -			
			Jl. Jendral Sudirman	17.700m		
			Jl. Garuda Sakti - Jl. HR. Subrantas -			
			Jl. Mr SM Yamin - Jl. Nangka 2 -			
Jl. Tuanku Tambusai - Jl. Jend Sudirman						

3	BTS Hotel Ibis	BTS Xnet	Jl. Soekarno Hatta - Jl. Nangka 2 -	7.032m	7.032m	Sama (berhasil)
			Jl. Srikandi - Jl. Delima - Jl. HR Subrantas			
			Jl. Soekarno Hatta - Jl. Nangka 2 -	8.200m		
			Jl. Mr SM Yamin - Jl. HR. Subrantas			
			Jl. Soekarno Hatta - Jl. Nangka 2 -	7.900m		
			Jl. Srikandi - Jl. Melati - Jl. Rajawali Sakti -			
Jl. HR. Subrantas						
4	BTS Hotel Ibis	BTS H. Ratu Mayang Garden	Jl. Soekarno Hatta - Jl. Tuanku Tambusai -	8.607m	8.607m	Sama (berhasil)
			Jl. Jendral Sudirman			
			Jl. Soekarno Hatta - Jl. Durian - Jl. Pepaya -	10.000m		
			Jl. Tuanku Tambusai - Jl. Jend Sudirman			
			Jl. Soekarno Hatta - Jl. Arifin Ahmad -	9.000m		
Jl. Jendral Sudirman						
5	CV. 2K	BTS DivaNet	Jl. Juanda - Jl. Ahmad Yani -	3.764m	3.764m	Sama (berhasil)
			Jl. KH Ahmad Dahlan - Jl. Durian			
			Jl. Juanda - Jl. Ahmad Yani -	4.000m		
			Jl. KH Ahmad Dahlan - Jl. Teratai - Jl. Durian			
			Jl. Juanda - Jl. Riau - Jl. Melur -	4.000m		
			Jl. Rajawali - Jl. Durian			

6	CV. 2K	BTS NeoNet	Jl. Juanda - Jl. Ahmad Yani -	8.566m	8.566m	Sama (berhasil)
			Jl. Jendral Sudirman - Jl. Harapan Raya			
			Jl. Juanda - Jl. Jendral Sudirman -	9.000m		
			Jl. Harapan Raya			
			Jl. Juanda - Jl. Ahmad Yani -	9.000m		
			Jl. Pepaya - Jl. Jend Sudirman -			
Jl. Harapan Raya						
7	BTS Hotel Ibis	BTS PrimaNet	Jl. Soekarno Hatta - Jl. Nangka 2 -	10.037m	10.037m	Sama (berhasil)
			Jl. Air Hitam - Jl. Melati - Jl. Garuda Sakti			
			Jl. Soekarno Hatta - Jl. Nangka 2 -	10.700m		
			Jl. Mr SM Amin - Jl. HR Subrantas -			
			Jl. Garuda Sakti	11.000m		
			Jl. Soekarno Hatta - Jl. HR Subrantas			
Jl. Garuda Sakti						
8	BTS DivaNet	BTS Hotel Sabrina	Jl. Durian - Jl. KH Ahmad Dahlan -	3.009m	3.009m	Sama (berhasil)
			Jl. Tuanku Tambusai			
			Jl. Durian - Jl. Pepaya - Jl. Tuanku Tambusai -	4.200m		
			Jl. Jendral Sudirman - Jl. Tuanku Tambusai			
Jl. Durian - Jl. Arjuna - Jl. Tuanku Tambusai	4.300m					



9	UIN SUSKA	BTS Hotel Sabrina	Jl. HR. Subrantas - Jl. Delima - Jl. Lobak -	13.000m	13.000m	Sama (berhasil)
			Jl. Soekarno Hatta - Jl. Tuanku Tambusai			
			Jl. HR. Subrantas - Jl. Mr SM Amin -	13.100m		
			Jl. Nangka 2 - Jl. Tuanku Tambusai			
			Jl. HR. Subrantas - Jl. Soekarno Hatta -	13.600m		
Jl. Arifin Ahmad - Jl. Paus - Jl. Nangka						
10	CV. 2K	BTS Xnet	Jl. Juanda - Jl. Ahmad Yani -	12.554m	12.554m	Sama (berhasil)
			Jl. KH Ahmad Dahlan - Jl. Durian -			
			Jl. Soekarno Hatta - Jl. Nangka 2 -			
			Jl. Srikandi - Jl. Delima - Jl. HR Subrantas			
			Jl. Juanda - Jl. Riau - Jl. Riau Ujung -	13.800m		
			Jl. Mr SM Amin - Jl. HR Subrantas			
			Jl. Juanda - Jl. Jendral Sudirman -	14.300m		
			Jl. Tuanku Tambusai - Jl. Nangka 2 -			
Jl. Mr SM Amin - Jl. HR Subrantas						

Tabel 5.4 Pengujian Aplikasi Pada Beberapa Perangkat dan Sistem Operasi

No	Merk Perangkat	Sistem Operasi	Hasil Pengujian Aplikasi
1	<i>Sony Experia Mini Pro</i>	<i>Android Ice Cream Sandwich 4.0</i>	Berjalan dengan lancar
2	<i>Samsung Galaxy Ace 2</i>	<i>Android Ice Cream Sandwich 4.0</i>	Berjalan dengan lancar
3	<i>Samsung Galaxy tab 2</i>	<i>Anroid Ice Cream Sandwich 4.0</i>	Berjalan dengan lancar
4	<i>Android Cross A20</i>	<i>Android Ice Cream Sandwich 4.0.4</i>	Berjalan dengan lancar

### 5.2.2. Kesimpulan Pengujian

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

1. Aplikasi *LBS BTS* yang dibangun untuk perangkat Android, dapat melakukan koneksi ke *server* dan dapat menampilkan konten sesuai dengan analisa dan perancangan.
2. Aplikasi yang dijalankan di beberapa perangkat Android yang berbeda merk dan operator mendapatkan hasil yang berbeda, seperti saat me-load foto BTS pada menu *Maintenance* sering terjadi *force close*.
3. Aplikasi yang dijalankan di beberapa perangkat Android yang berbeda merk, dan berbeda operator dapat berjalan dengan lancar.
4. *List* lokasi BTS dan rute terpendek yang ditampilkan di perangkat Android semua telah sesuai dengan *database* yang di-*server*.
5. Aplikasi yang dijalankan di perangkat android ini memiliki beberapa kendala, yaitu:
  - a. Kondisi cuaca: kondisi cuaca cerah lebih baik dan cepat GPS untuk nge-*lock* posisinya, dan sebaliknya jika kondisinya dalam keadaan mendung maka GPS akan mengalami kesulitan untuk nge-*lock* posisinya.
  - b. *Obstacle*/ hambatan, seperti: berada didalam gedung atau ruangan tertutup.
  - c. Kualitas sinyal dari masing-masing operator.

6. Dari pengujian yang dilakukan diatas maka dapat disimpulkan bahwa operator yang terbaik untuk mengakses aplikasi ini adalah Smartfren dan Telkomsel.
7. Aplikasi ini sangat berpengaruh pada koordinat pengguna. Apabila koordinat pengguna berhasil didapat, maka aplikasi bisa berjalan sebagaimana mestinya. Apabila tidak, maka aplikasi tidak dapat membantu pengguna untuk menentukan rute terpendek.
8. Waktu yang dibutuhkan dalam mengakses aplikasi jika dalam kondisi terbaik, maksudnya dengan kondisi terbaik adalah dengan cuaca yang cerah, tidak berada di dalam ruangan dan dengan sinyal operator yang baik adalah  $\pm 8$  menit.

## BAB VI

### PENUTUP

#### 6.1 Kesimpulan

Setelah menyelesaikan serangkaian tahapan dalam merancang dan membangun aplikasi *LBS BTS* berbasis *client-server* yang menerapkan teknologi *Location Based Service* dan algoritma *Floyd-Warshall* pada sistem operasi Android, hingga pada tahapan pengujian, maka dapat diambil beberapa kesimpulan diantaranya adalah sebagai berikut:

1. Aplikasi yang dibangun sudah berjalan pada perangkat Android dan bisa mengakses *database server* serta mampu menunjukkan arah rute terpendek menuju lokasi tujuan.
2. Berbagai perangkat dengan *hardware* dan *software* yang berbeda masih dapat menjalankan semua fitur aplikasi dengan baik.
3. Aplikasi *LBS BTS* sudah bisa menampilkan beberapa *list* lokasi *BTS*, Status *BTS* yang bermasalah dan rute terpendek pada perangkat android dalam bentuk *map* menggunakan algoritma *floyd-warshall*.
4. Waktu yang dibutuhkan dalam mengakses aplikasi jika dalam kondisi terbaik, maksudnya dengan kondisi terbaik adalah dengan cuaca yang cerah, tidak berada di dalam ruangan dan dengan sinyal operator yang baik adalah  $\pm 8$  menit.

#### 6.2 Saran

Beberapa hal yang disarankan dalam pengembangan aplikasi *LBS BTS* berbasis *client-server* yang menerapkan teknologi *Location Based Service* dan algoritma *Floyd-Warshall* pada sistem operasi Android ini adalah sebagai berikut:

1. Untuk pengembangan aplikasi selanjutnya, diharapkan pada data jalan yang digunakan sudah mencakup seluruh tipe jalan dan jalan yang ada di Kota Pekanbaru yang disesuaikan dengan peraturan lalu lintas yang ada.

2. Pada pengembangan aplikasi selanjutnya, diharapkan agar mampu memberikan hasil pencarian yang lebih cepat dengan menambahkan algoritma optimasi dalam melakukan pencarian rute terpendek.
3. Pada pengembangan aplikasi selanjutnya, diharapkan *server database* bisa menyimpan hasil pencarian yang pernah dilakukan sebelumnya, sehingga apabila ada pencarian lokasi yang berasal dari lokasi yang sama menuju lokasi tujuan yang sama pula, maka akan langsung menampilkan hasilnya atau *direction* tanpa perlu melakukan proses pencarian kembali.
4. Pada pengembangan aplikasi LBS BTS selanjutnya diharapkan dapat dioperasikan tidak hanya di android saja, melainkan juga bisa dioperasikan pada *platform* yang lain.

## DAFTAR PUSTAKA

- Adeline Narwastu dan Eri Prasetyo W. *Perancangan Sistem Informasi Geografis Daerah Banjir Di DKI Jakarta Dengan Menggunakan Arc View*. Proceeding PESAT (Psikologi, Ekonomi, Sastra, dan Sipil) Vol. 2, Agustus, 2007.
- Akbar Nuzul Putra, Toufan D. Tambunan, S.T., Kurniawan Nur Ramadhan, S.T. *Aplikasi Wisata Kota Bandung Menggunakan Metode Location-Based Services (LBS) pada Android*. Program Studi Manajemen Informatika Politeknik Telkom Bandung, 2011.
- Apri Kamayudi. *Studi dan Implementasi Algoritma Dijkstra, Bellman-Ford dan Floyd-Warshall Dalam Menangani Masalah Lintasan Terpendek Dalam Graf*. Program Studi Teknik Informatika, Institut Teknologi Bandung
- Darwiyanti, Sri dan Romi Satria Wahono. *Pengenalan Unified Modeling Language(UML)*. [Online] Available.
- Faizah Ifatul, *Rancang Bangun Perangkat Lunak Penentuan Rute Perjalanan Wisata Di Malang Menggunakan Algoritma Dijkstra*. Laporan Tugas Akhir Sarjana, Jurusan Teknik Informatika, UIN Maulana Malik Ibarahim. Malang, 2010.
- Kruchten, Philippe, Per Kroll. *Rational Unified Process Made Easy A Practitioner's Guide to RUP*. IBM Software Group, 2003.
- Michell Setyawati Handaka. *Perbandingan Algoritma Dijkstra (Greedy), Bellman-Ford (BFS-DFS), dan Floyd-Warshall (Dynamic Programming) dalam Pengaplikasian Lintasan Terpendek pada Link-State Routing Protocol*. Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, 2010/2011.
- Munir, Rinaldi. *Buku Teks Ilmu Komputer : Matematika Diskrit Edisi Ketiga*. Penerbit Informatika, Bandung, 2005.
- Parno, Febriani, Andhika Prakasa Kasma. *Aplikasi Android Penyedia Informasi Alat Transportasi umum (Studi Kasus : Angkutan umum di Terminal Kampung Rambutan)*. Prosiding SNTIKI 4, Fakultas Sains dan Teknologi UIN Sultan Syarif Kasim, 2012.

Putra, Thio Pratama. *Rancang Bangun Aplikasi Mobile Learning Client Server Berbasis Moodle Pada Platform Android*. Laporan Tugas Akhir Sarjana, Jurusan Teknik Informatika, Universitas Islam Negeri Sultan Syarif Kasim Riau. 2011.

Raden Aprian Diaz Novandi. *Perbandingan Algoritma Dijkstra dan Algoritma Floyd-Warshall Dalam Penentu Lintasan Terpendek (Single Pair Shortest Path)*. Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, 2007.

Rational Team. *“Rational Unified Process : Best Practices for Software Development Teams”*. 2001.

Safaat, Nazruddin. *Android: Pemrograman Aplikasi Mobile Smartphone dan Tablet PC*. Penerbit Informatika, Bandung, 2011.

Steiniger, Stefan, Moritz Neun, and Alistair Edwardes. *Foundation of Location Based Services*. University of Zurich, 2012.

<http://www.idc.com/getdoc.jsp?containerId=prUS23638712>

<http://ilmukoputer.org/2006/08/05/pengantar-uml/> 14 April 2011

<http://teknologi.inilah.com/read/detail/1992003/inilah-sejarah-penamaan-dari-tiap-versi-android#.Uc7j8zsXGgs>