

**BAB III
PERANCANGAN DAN IMPLEMENTASI**

Pada bab ini akan dibahas tentang perancangan sistem yang bertujuan untuk mencari bentuk yang optimal dari aplikasi yang akan dibangun dengan mempertimbangkan berbagai faktor permasalahan serta kebutuhan yang ada pada sistem. Keoptimalan akses juga merupakan pertimbangan bagaimana sistem ini bisa berjalan dengan cepat dan juga tepat. Upaya yang dilakukan adalah dengan berusaha mencari kombinasi penggunaan teknologi dan perangkat lunak yang tepat sehingga diperoleh hasil yang optimal dan mudah untuk diimplementasikan.

3.1. DESKRIPSI UMUM SISTEM

Sistem adalah aplikasi *mobile* berbasis sistem operasi Android yang berisikan panduan dalam berwisata di Wisata Bahari Lamongan. Aplikasi ini dibuat sedemikian rupa agar *user* dapat mendapatkan informasi lengkap seputar tempat wisata Wisata Bahari Lamongan. Informasi yang tersedia dalam aplikasi ini antara lain navigasi menuju Wisata Bahari Lamongan, peta wisata Wisata Bahari Lamongan beserta navigasi di dalamnya, informasi daftar wahana, notifikasi wahana terdekat, dan daftar harga tiket paling *update*.

Navigasi untuk menuju Wisata Bahari Lamongan dibagi menjadi dua jenis, yaitu navigasi apabila *user* menggunakan kendaraan pribadi dan navigasi apabila *user* menggunakan kendaraan umum seperti angkutan umum ataupun bis kota. Dengan adanya navigasi ini, *user* dapat terbantu dalam perjalanan menuju Wisata Bahari Lamongan.

Peta wisata Wisata Bahari Lamongan menggunakan peta dari *vendor* OpenStreetMap karena bersifat *open source*, sehingga mudah dikembangkan. Dengan adanya navigasi dalam *indoor map*, *user* dapat mendapatkan navigasi rute jalan tercepat menuju wahana yang dipilih.

Informasi daftar wahana dan harga tiket dapat membantu *user* untuk mengetahui perkembangan informasi mengenai Wisata Bahari Lamongan.

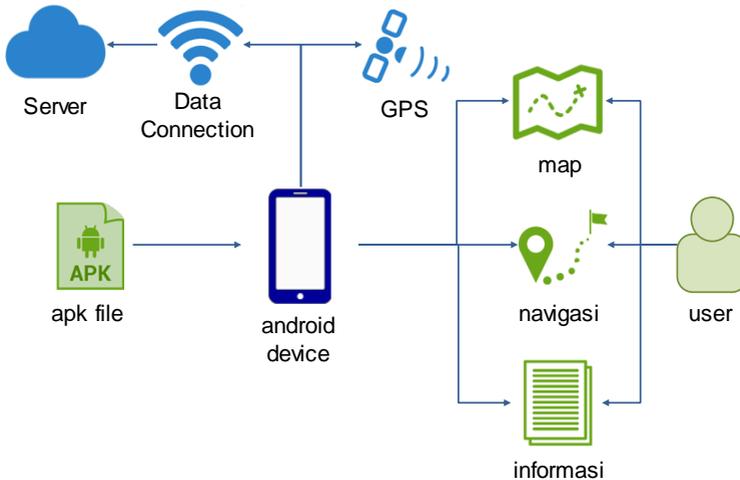
Aplikasi WBL Guide juga dilengkapi dengan sistem notifikasi mengenai wahana terdekat dari posisi *user*, sehingga *user* dapat mengetahui wahana yang akan dilalui selanjutnya

3.2. PERANCANGAN SISTEM

Pada tahap ini akan dibahas mengenai perancangan sistem aplikasi yang berisi rancangan alur sistem, rancangan aplikasi, rancangan menu aplikasi, dan rancangan fitur aplikasi.

3.2.1. Rancangan Alur Sistem

Cara kerja dari aplikasi WBL Guide ditunjukkan pada **Gambar 3.1**.



Gambar 3.1 Cara kerja aplikasi

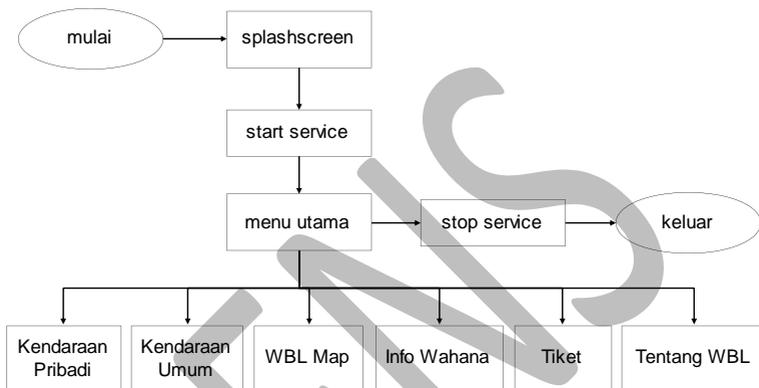
Berikut adalah penjelasan dari cara kerja aplikasi pada **Gambar 3.1**:

- User* mendownload WBLGuide.apk terlebih dahulu pada Android Store, kemudian menginstalnya pada *gadget* Android *user*.
- Setiap penggunaan aplikasi WBL Guide diperlukan koneksi GPS dan koneksi data dari *gadget* Android tersebut.
- Koneksi data digunakan untuk mengambil data dari server. Data yang diambil dari server, yaitu tampilan map dari server OpenStreetMap dan data untuk harga tiket terbaru.
- Koneksi GPS digunakan untuk mengetahui koordinat posisi dari *user*.

- e. Dari aplikasi WBL Guide akan didapatkan peta wisata dari Wisata Bahari Lamongan, navigasi *user*, dan informasi seputar Wisata Bahari Lamongan yang akan membantu *user* dalam berwisata di Wisata Bahari Lamongan.

3.2.2. Rancangan Aplikasi

Diagram rancang dari aplikasi ditunjukkan pada **Gambar 3.2**.



Gambar 3.2 Rancangan Aplikasi

Berikut adalah penjelasan dari rancangan aplikasi pada **Gambar 3.2**:

- a. *Splashscreen*
Splashscreen merupakan page yang menampilkan logo publisher, sponsor, ataupun logo aplikasi itu sendiri. *Splashscreen* adalah page yang pertama kali ditampilkan saat aplikasi dibuka.
- b. *Start Service*
Setelah *Splashscreen* muncul, maka page menu utama akan muncul. Saat menu utama telah tampil maka aplikasi memulai untuk memulai *service* dari aplikasi WBL Guide, yaitu *proximity service* notifikasi untuk menunjukkan wahana terdekat.
- c. *Menu Utama*
Menu utama merupakan menu pertama yang muncul ketika kita menjalankan aplikasi. Menu utama merupakan menu untuk menuju menu-menu yang lain, seperti kendaraan pribadi,

kendaraan umum, WBL map, info wahana, tiket, dan menu tentang WBL, maupun menuju keluar aplikasi.

d. *Stop Service*

Setelah *user* menutup aplikasi, maka seluruh proses dalam aplikasi harus dihentikan, termasuk dengan menghentikan *service* notifikasi untuk menunjukkan wahana terdekat.

3.2.3. Rancangan Fitur Aplikasi

Terdapat beberapa fitur pada aplikasi WBL Guide, yaitu:

a. *Indoor map* Wisata Bahari Lamongan

Fitur ini berfungsi untuk menampilkan denah *indoor* dari Wisata Bahari Lamongan yang berdasar dari kondisi sebenarnya. Sehingga dengan *Indoor Map* dari aplikasi sudah menginterpretasikan struktur bangunan dari Wisata Bahari Lamongan, terlihat pada **Gambar 3.3**.



Gambar 3.3 Map Indoor Wisata Bahari Lamongan

Fitur ini juga menampilkan posisi-posisi wahana dengan menggunakan *marker* sebagai penanda wahana. *Marker* yang digunakan merupakan *icon* dari kategori jenis wahana Wisata Bahari Lamongan. Tampilan *Indoor Map* dengan *marker* terlihat pada **Gambar 3.4**.



Gambar 3.4 *Indor Map* Wisata Bahari Lamongan dengan *marker*

Dengan menekan *marker* dengan *single tap* (menyentuh sekali), maka akan menampilkan nama dari wahana yang dipilih. Dengan menekan *marker* dengan *long press* (menyentuh lama), maka akan menghubungkan menuju halaman informasi wahana dengan menampilkan dari wahana yang dipilih.

b. Informasi Wahana

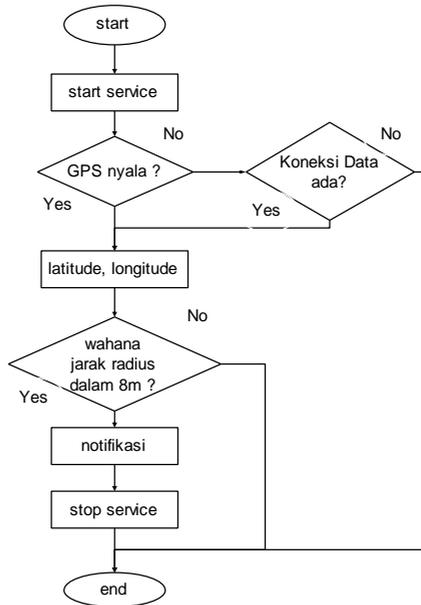
Fitur ini berfungsi untuk menampilkan informasi seputar wahana Wisata Bahari Lamongan yang dikategorikan berdasarkan jenis wahana, yaitu Wahana Adrenalin, Wahana Air, Wahana Anak, Wahana Atraktif, Resto - Store, Wahana Edukasi, Wahana Entertaining, dan Wahana Riding.

Dalam tiap informasi wahana berisikan nama wahana, gambar wahana, tarif wahana untuk wahana yang berbayar, dan deskripsi singkat wahana.

c. Notifikasi wahana terdekat

Fitur ini berfungsi untuk memberikan notifikasi atau pemberitahuan kepada *user* setiap mendekati suatu wahana. Sehingga *user* dapat mengetahui wahana apa yang akan dimasuki. Fitur notifikasi ini berbentuk sebuah *service* pada

Android, sehingga apabila aplikasi di *minimize* tetap akan menjalankan *service* tersebut. Algoritma terlihat pada **Gambar 3.5**.

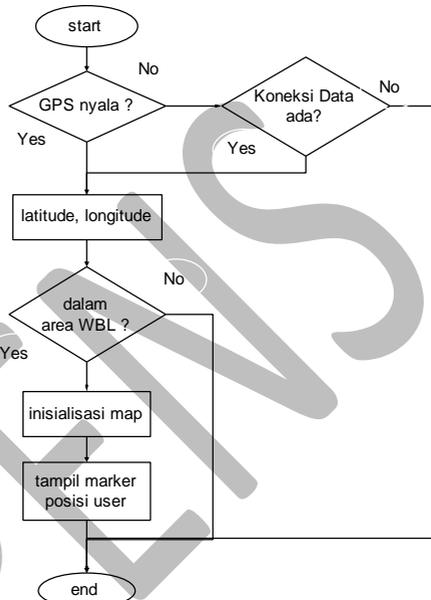


Gambar 3.5 Algoritma *Service* Notifikasi Wahana Terdekat

Fitur ini membutuhkan GPS dan koneksi data untuk mendapatkan koordinat dari posisi *user* sekarang. Fitur dimulai dengan penyalaan *service* notifikasi. *Service* akan dimulai saat aplikasi telah memasuki halaman menu utama. *Service* ini kemudian akan melakukan pengecekan apakah *gadget user* telah mengaktifkan GPS dan paket data. Apabila telah diaktifkan, maka didapatkan koordinat *longitude* dan *latitude* dari *user*. Kemudian dihitung jarak radius antara *user* dengan wahana terdekat. Apabila jarak antara *user* dengan wahana terdekat dalam radius 8 meter, maka akan menampilkan notifikasi berupa *toast* yang memberitahukan bahwa *user* telah memasuki wahana yang termasuk dalam radius 8 meter.

d. Menampilkan posisi *user* terkini

Fitur ini berfungsi untuk menampilkan posisi *user* terkini saat berada dalam area Wisata Bahari Lamongan. Sehingga *user* tidak perlu takut akan kemungkinan tersesat. Algoritma terlihat pada **Gambar 3.6**.

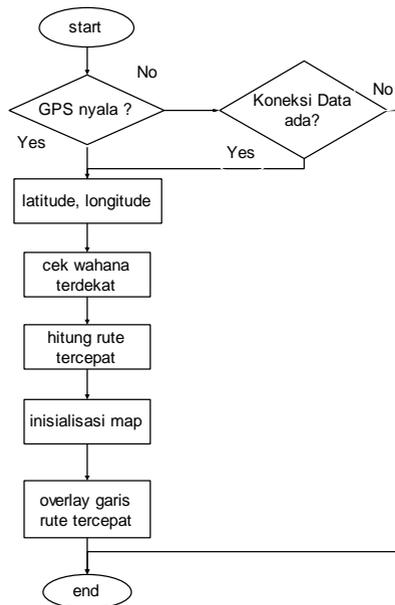


Gambar 3.6 Algoritma Fitur Posisi *User* Terkini

Pada Gambar berisikan mengenai alur dari sistem dalam proses pencarian posisi *user* terkini. Diawali dengan pengecekan apabila GPS dan koneksi data dari *gadget* Android apakah sudah berfungsi atau belum. Apabila sudah berfungsi, maka akan dilanjut dengan penginisialisasian peta dengan titik koordinat *longitude* dan *latitude* *user* sebagai *center* dari peta, dan kemudian pemberian *marker* sebagai penunjuk *user*. Apabila tidak, maka fitur tidak dijalankan, dan hanya mencetak pemberitahuan mengenai GPS dan koneksi data dari *gadget*.

e. Navigasi rute menuju wahana dari posisi *user*

Fitur ini berfungsi untuk menampilkan navigasi rute tercepat yang dapat diambil oleh *user* saat ingin menuju ke wahana yang diinginkan saat posisi awal *user* sedang berada dalam area Wisata Bahari Lamongan. Sehingga *user* dapat menghemat waktu dalam perjalanan menuju wahana tersebut. Algoritma terlihat pada **Gambar 3.7**.



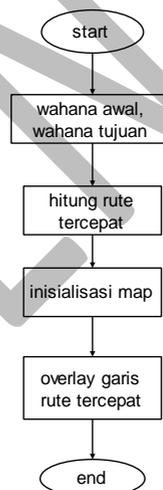
Gambar 3.7 Algoritma navigasi rute dari posisi *user* menuju wahana

Fitur ini membutuhkan koneksi data dan GPS untuk mendapatkan koordinat dari *longitude* dan *latitude* dari posisi *user* sekarang. Fitur ini merupakan keberlanjutan dari fitur menampilkan posisi *user* sekarang. Dari koordinat posisi *user* yang telah didapatkan maka akan dicari node dari jalur yang merupakan node terdekat dari posisi *user* tersebut. Kemudian menghitung rute terdekat yang dapat dilalui dari node terdekat dari *user* hingga ke wahana yang dituju. Urutan node rute

terdekat tersebut akan disimpan dalam sebuah array. Inisialisasi map dilakukan untuk menampilkan *Indoor Map* Wisata Bahari Lamongan sebagai dasar dengan koordinat posisi *user* sekarang menjadi pusat koordinat peta. Seluruh urutan node yang tersimpan dalam array akan dicetak berupa *overlay* garis, sehingga terlihat jalur rute terdekat yang dapat diambil *user* dari wahana awal hingga sampai ke wahana tujuan.

f. Navigasi rute antar wahana

Fitur ini berfungsi untuk menampilkan navigasi rute tercepat yang dapat diambil oleh *user* saat ingin menuju ke wahana yang diinginkan saat posisi awal *user* sedang berada disuatu wahana lain. Sehingga *user* dapat menghemat waktu dalam perjalanan menuju wahana tersebut. Algoritma terlihat pada **Gambar 3.8**.



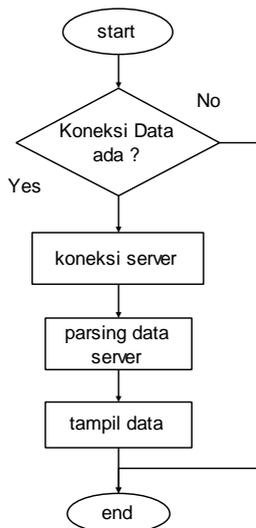
Gambar 3.8 Algoritma navigasi rute antar wahana

Fitur ini membutuhkan inputan wahana awal dan wahana tujuan yang dipilih oleh *user*. Dengan didapat node dari wahana awal dan wahana akhir, maka dilanjutkan dengan menghitung rute terdekat dari node wahana awal hingga node wahana tujuan. Setelah didapatkan urutan node rute terdekat dalam array, maka dilakukan inisialisasi peta untuk menampilkan

Indoor Map indoor Wisata Bahari Lamongan sebagai dasar dengan koordinat node wahana awal menjadi pusat koordinat peta. Seluruh urutan node yang tersimpan dalam array akan dicetak berupa *overlay* garis, sehingga terlihat jalur rute terdekat yang dapat diambil *user* dari wahana awal hingga sampai ke wahana tujuan.

g. Informasi harga tiket *terupdate*

Fitur ini berfungsi untuk menampilkan harga tiket masuk Wisata Bahari Lamongan ataupun harga tiket terusan Wisata Bahari Lamongan dan Maharani Zoo and Goa, baik pada hari kerja maupun hari *weekend* dan hari libur. Sehingga dengan menggunakan aplikasi WBL Guide, *user* sudah dapat mengetahui harga tiket tanpa harus mengecek dalam website resmi. Algoritma terlihat pada **Gambar 3.9**.



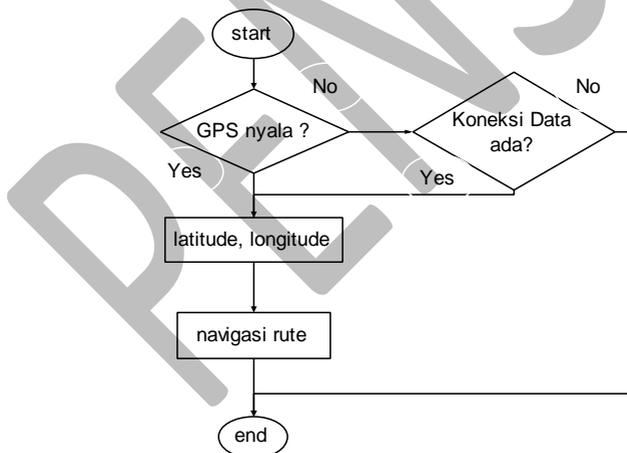
Gambar 3.9 Algoritma informasi harga tiket

Fitur ini membutuhkan koneksi data untuk menghubungkan aplikasi dengan server yang berisikan informasi *update* dari harga tiket masuk Wisata Bahari Lamongan. Pertama, terjadi pengecekan apakah *gadget* Android *user* telah mengaktifkan

koneksi data. Jika aktif, maka aplikasi mulai membuat koneksi dengan server. Dilanjut dengan parsing data dari server menjadi data pada aplikasi. Untuk parsing data server, penulis menggunakan *json (JavaScript Object Notation)* untuk parsing data yang didapat dari server web menjadi list pada aplikasi. Setelah berhasil memarsing data server, maka menampilkan list data pada aplikasi.

- h. Navigasi rute pengguna kendaraan pribadi menuju Wisata Bahari Lamongan

Fitur ini berfungsi untuk menampilkan rute dari *user* berada menuju lokasi Wisata Bahari Lamongan dengan menggunakan *library google maps*. Sehingga *user* dapat terbantu dalam perjalanan menuju Wisata Bahari Lamongan dan mengurangi kejadian tersesat saat dalam tengah perjalanan. Algoritma ditunjukkan pada **Gambar 3.10**.



Gambar 3.10 Algoritma navigasi *user* kendaraan pribadi

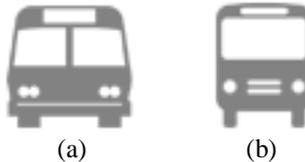
Fitur ini membutuhkan koneksi data untuk menghubungkan dengan Google Map dan GPS untuk mendapatkan titik koordinat dari posisi *user* sekarang. Pertama, dilakukan pengecekan apakah telah diaktifkan koneksi data dan GPS pada *gadget*. Apabila aktif, maka akan didapatkan koordinat *longitude* dan *latititude* dari posisi *user* yang akan ditampilkan

dalam Google Maps untuk mendapatkan navigasi rute menuju Wisata Bahari Lamongan.

- i. Navigasi *user* pengguna kendaraan umum menuju Wisata Bahari Lamongan

Fitur ini berfungsi untuk menampilkan panduan menuju Wisata Bahari Lamongan menggunakan kendaraan umum yang dimana *user* berada di Bandara terdekat, Terminal terdekat, dan Stasiun terdekat. Bandara terdekat yaitu Bandara Juanda, Terminal terdekat yaitu Terminal Lamongan, dan Stasiun terdekat yaitu Stasiun Kota Lamongan. Sehingga *user* dapat terbantu untuk bepergian menuju Wisata Bahari Lamongan.

Pada aplikasi WBL Guide, transportasi umum dibagi menjadi 2 jenis, yaitu bemo atau angkot dan bis. *Icon* terlihat pada **Gambar 3.11**.



Gambar 3.11 *Icon* panduan kendaraan umum
(a) *icon* bemo (b) *icon* bis

- j. Informasi umum seputar Wisata Bahari Lamongan

Fitur ini berfungsi untuk menampilkan informasi umum seputar objek wisata Wisata Bahari Lamongan, lengkap dengan deskripsi umum dan nomor yang dapat dihubungi dari Wisata Bahari Lamongan.

3.3. IMPLEMENTASI SISTEM

Pada tahap ini mengenai pengimplementasian sistem pada aplikasi WBL Guide yang berisi *tool* yang digunakan dalam pembuatan sistem dan alur pembuatan aplikasi.

3.3.1. *Tool* yang digunakan

Berikut adalah *tool* yang digunakan selama pembuatan sistem dari aplikasi WBL Guide:

a. MyTrack

MyTrack adalah *software* berbasis Android yang berfungsi dalam *tracking* rute dalam penyimpanan jalur melalui titik koordinatnya. MyTrack langsung terintegrasi dengan akun google *user*, sehingga semua hasil dari *tracking* yang telah dilakukan akan langsung dapat tersambung pada google drive *user*. Aplikasi ini digunakan oleh penulis untuk survey rute dalam area Wisata Bahari Lamongan, sehingga didapatkan jalur yang diinginkan.

b. Java OpenStreetMap

Java OpenStreetMap atau JOSM adalah *software* yang digunakan oleh penulis untuk membuat peta dari denah Wisata Bahari Lamongan. Java OpenStreetMap adalah *editor* untuk OpenStreetMap, berupa aplikasi *desktop* yang dimana kita dapat mengedit peta dari database server OpenStreetMap kemudian mengupload peta editan kita ke dalam database server OpenStreetMap.

c. Eclipse

Eclipse adalah *compiler* yang digunakan penulis dalam pengodingan aplikasi. Eclipse adalah sebuah IDE (*Integrated Development Enviroment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (*platform-independet*). Eclipse merupakan IDE yang memiliki sifat multi. Dimulai dari *multi platform*, *multi language*, dan *multi role*. Eclipse merupakan IDE favorit dikarenakan gratis dan mendukung penggunaan *open source*, yang berarti setiap developer dapat melihat kode pemrograman perangkat lunak ini.

Secara standar bahasa pemrograman yang digunakan oleh eclipse adalah Java yang dilengkapi dengan JDT (*Java Development Tools*) memudahkan Eclipse untuk dikembangkan, plug in yang membuat Eclipse menjadi kompatibel, dan PDE (*Plug-in Development Enviroment*) yang diperuntukan dalam pengembangan *plug-in* baru. *Plug-in* dalam Eclipse dikembangkan dalam pemrograman Java.

d. Adobe Dreamweaver

Adobe Dreamweaver adalah *software* yang digunakan oleh penulis sebagai *compiler* untuk bahasa pemrograman php dalam pengerjaan json yang akan digunakan dalam aplikasi WBL Guide.



Adobe Dreamweaver adalah aplikasi pemrograman untuk desain web yang menyediakan *editor* WYSIWYG visual (*Design View*) dan *code editor* dengan fitur standar seperti *syntax highlighting*, *code completion*, dan *code collapsing* serta fitur yang lebih canggih seperti *real-time syntax checking* dan *code introspection* untuk membantu *user* menulis kode. *Design view layout* memfasilitasi desain cepat dan pembuatan kode yang memungkinkan pengguna untuk dengan cepat membuat tata letak dan manipulasi elemen HTML.

e. Corel Draw

Corel Draw adalah *software* yang digunakan penulis untuk membuat *image vector* sehingga tidak bisa pecah apabila diperbesar. Corel Draw adalah *editor* grafis *vector* yang dikembangkan oleh Corel Corporation Ottawa, Kanada. Corel Draw juga merupakan nama Corel Graphics Suite, yang bundle Corel Draw dengan *editor* gambar *bitmap*, Corel Photo Paint, dan program grafis lain yang berhubungan.

f. Adobe Photoshop

Adobe Photoshop adalah *software image editor* yang digunakan penulis untuk membuat *image bitmap* dengan menggabungkan kumpulan *image vector* menjadi sebuah *image* hasil. Adobe Photoshop adalah program editing grafis dikembangkan dan diterbitkan oleh Adobe Systems. Photoshop diluncurkan mempunyai dua edisi, yaitu Adobe Photoshop dan Adobe Photoshop Extended. Adobe Photoshop Extended mempunyai tambahan penciptaan gambar 3D, mengedit gerakan grafik, dan fitur analisis citra yang rumit.

g. Android

Android adalah perangkat atau *device* yang digunakan untuk mengimplementasikan aplikasi ini. Android adalah sistem operasi *mobile* yang berbasis Linux.

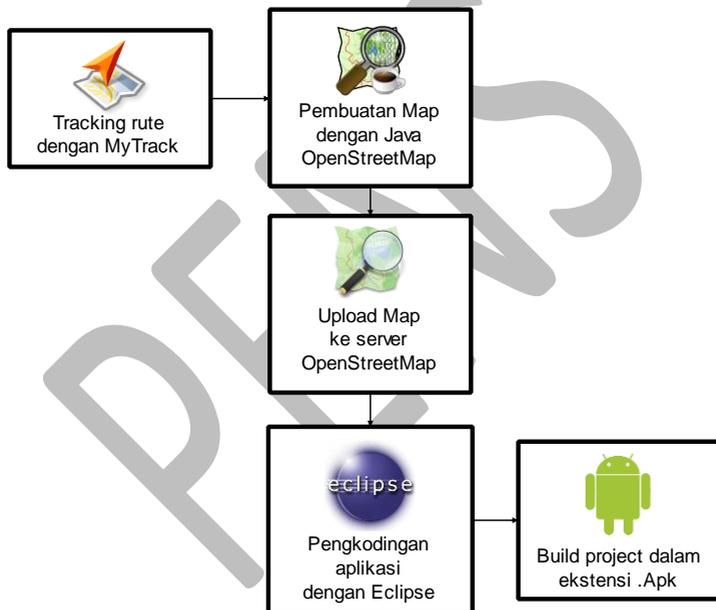
Fitur yang tersedia pada Android adalah:

- Dalvik mesin virtual: mesin virtual dioptimalkan untuk perangkat *mobile*.
- Grafik: grafik di 2D dan grafis 3D berdasarkan pustaka OpenGL.

- SQLite: untuk penyimpanan data.
- Mendukung media: audio, video, dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF).
- GSM, Bluetooth, EDGE, 3G, dan WiFi (*hardware dependent*)
- Kamera, *Global Positioning System* (GPS), kompas, dan accelerometer (tergantung *hardware*).

3.3.2. Pembuatan Aplikasi

Pada tahap ini akan dijelaskan mengenai alur pembuatan aplikasi WBL Guide.



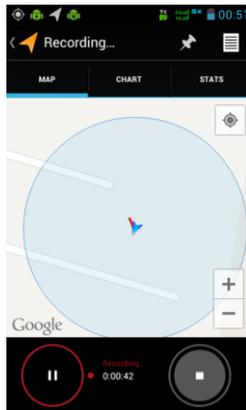
Gambar 3.12 Diagram Pembuatan Aplikasi

Pada **Gambar 3.12** berisikan diagram pembuatan aplikasi yang menjelaskan alur dalam proses pembuatan aplikasi.

a. *Tracking* Rute dengan MyTrack

Tahapan awal dalam penelitian ini adalah mencari titik-titik koordinat daerah Wisata Bahari Lamongan. Pada tahapan ini,

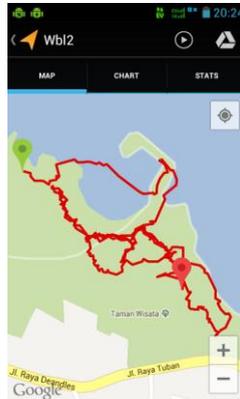
penulis menggunakan bantuan aplikasi MyTrack yang tersedia pada *device* Android yang berguna untuk mengambil titik koordinat pada jalur yang telah dilewati oleh penulis.



Gambar 3.13 Tampilan aplikasi MyTrack

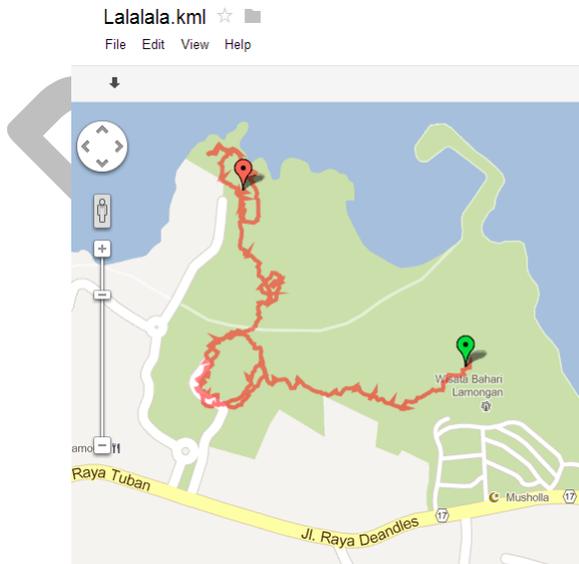
Pada **Gambar 3.13** merupakan tampilan dari aplikasi MyTrack di Android. Pada saat melakukan recording pada MyTrack, *gadget* penulis harus memastikan fitur GPS dan fitur Paket Data untuk memaksimalkan akurasi dari pengambilan posisi *current location* dari *user* dan penampilan pada map posisi penulis.

Pada saat penyimpanan beberapa hasil dari *tracking* terlihat kurang rapi dikarenakan faktor kekuatan GPS dari *gadget* Android tersebut saat melakukan *tracking*, ataupun dikarenakan faktor sinyal dari *provider* yang digunakan dalam *device* Android tersebut, seperti pada **Gambar 3.14**.



Gambar 3.14 Tampilan hasil *tracking* pada MyTrack

Setelah hasil *tracking* didapat, penulis dapat mengambil hasil *tracking* di akun google drive penulis berupa *file* kml, seperti pada **Gambar 3.15**.

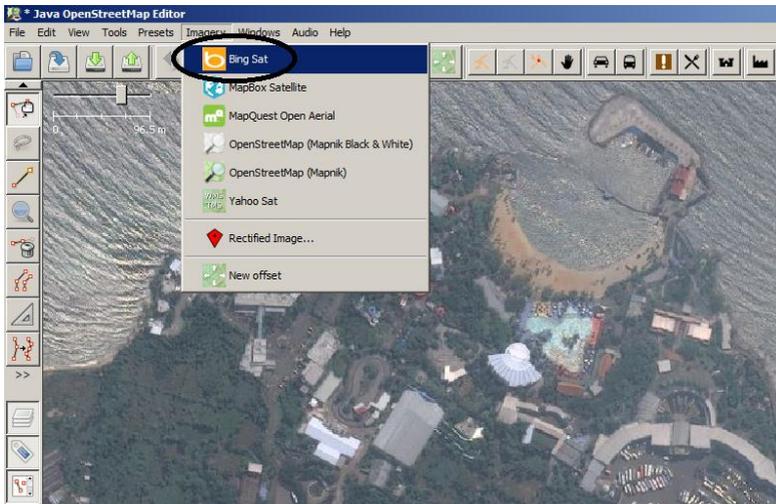


Gambar 3.15 Tampilan hasil *tracking* di Google Drive

b. Pembuatan Map dengan Java OpenStreetMap

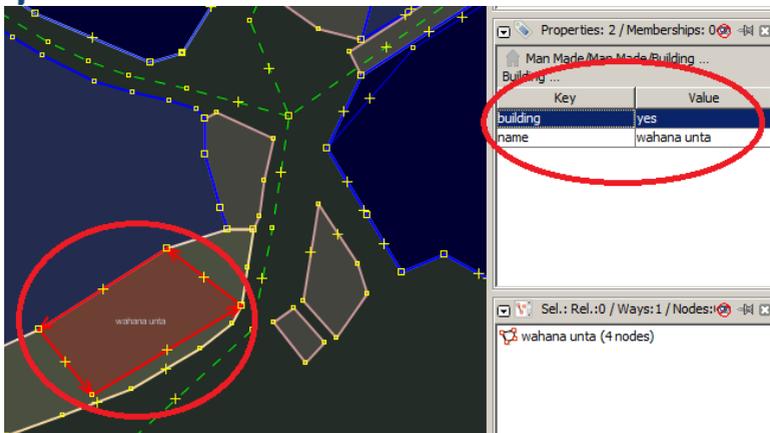
Setelah semua tempat wahana dan fasilitas didapatkan jalur tracknya, maka berlanjut pada tahap penggabungan dan penghalusan hasil *tracking* yang digabung dengan pembuatan peta, sehingga terbentuk satu kesatuan peta. Untuk penggabungan penulis menggunakan aplikasi Java OpenStreetMap.

Pada awal pembuatan peta, penulis menggunakan bantuan *Imagery* dari Bing Sat agar menampilkan tampilan satelit dari Bing, sehingga penulis dapat memperkirakan ketepatan dari tiap node dengan kondisi sebenarnya, pada **Gambar 3.16**. Sehingga dapat memaksimalkan tingkat presisi dari peta dengan posisi sebenarnya.



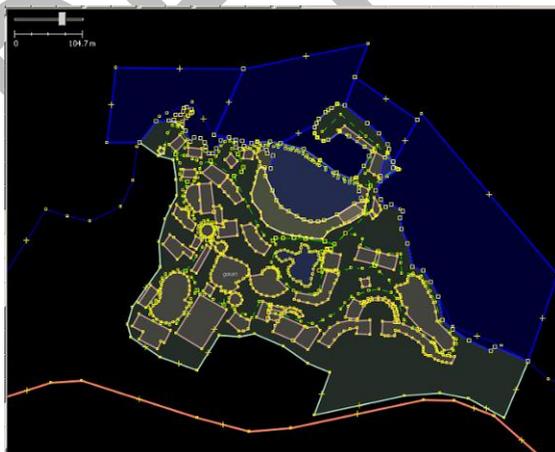
Gambar 3.16 Penggunaan *Imagery* Bing Sat pada JOSM

Tiap daerah yang menandakan suatu tempat akan di kelilingi oleh beberapa node. Sekumpulan node inilah yang akan membentuk suatu bangunan pada Java OpenStreetMap, seperti pada **Gambar 3.17**.



Gambar 3.17 Properties pada node Java OpenStreetMap

Sebuah bangunan yang telah dikelilingi node tersebut diharuskan memberikan properties, seperti *building*, *tower*, *waterwork*, dan lain-lain. Properties pada tiap node digunakan untuk memberikan ciri khas dalam penampilan warna dalam hasil pada map dalam OpenStreetMap seperti pada **Gambar 3.18**.



Gambar 3.18 Tampilan *Indoor map* Java OpenStreetMap

Pada **Gambar 3.18** terlihat hasil dari penggabungan seluruh hasil *tracking* yang disatukan menjadi satu peta untuk tempat Wisata Bahari Lamongan. Setelah penggabungan seluruh hasil *tracking* selesai, tahap selanjutnya adalah mengupload peta ke server OpenStreetMap.

c. Upload Map ke server OpenStreetMap

Setelah tahapan pembuatan *Indoor map* untuk Wisata Bahari Lamongan selesai, maka tahapan selanjutnya adalah mengupload hasil *Indoor Map* pada server OpenStreetMap dengan fitur upload pada *Java OpenStreetMap* dengan menggunakan akun OpenStreetMap penulis.

Dikarenakan *vendor* OpenStreetMap bersifat *open source*, sehingga proses upload dapat cepat terupload tanpa harus menunggu konfirmasi. Hanya dengan mendaftar sebagai member dari OpenStreetMap sudah dapat ikut mengupload peta pada server OpenStreetMap.

Untuk upload map ke server OpenStreetMap dari Java OpenStreetMap, pilih menu *File* kemudian memilih *Upload Data* atau dengan *shortcut Ctrl + Shift + Up*.



Gambar 3.19 Hasil tampilan pada server OpenStreetMap

Pada **Gambar 3.19** adalah tampilan dari hasil upload projek pada Java OpenStreetMap. Tampilan yang beragam pada map menandakan banyaknya *type properties* untuk tiap node yang digunakan oleh penulis.

d. Pengkodean aplikasi dengan Eclipse

Tahapan selanjutnya adalah scripting aplikasi Proyek Akhir dengan *software* Eclipse dengan menggunakan bahasa pemrograman java native. Untuk fitur penampilan Map OpenStreetMap, penulis menggunakan *library osmdroid* dan *slf4j*, yaitu *library* khusus yang digunakan untuk penggunaan elemen OpenStreetMap pada pemrograman java dalam Android developing.

Penulis juga menggunakan pemrograman web seperti html, css, php, dan javascript untuk membuat server data dari data harga tiket masuk Wisata Bahari Lamongan.

e. *Build project* dalam ekstensi .apk untuk dijalankan pada Android

Setelah semua tahapan selesai, maka *project* dapat dibangun dalam ekstensi file apk untuk dijalankan dalam *device* android.



Halaman ini sengaja dikosongkan