# RANCANG BANGUN APLIKASI *FISHING MAP* BERBASIS ANDROID

## SKRIPSI


Disusun Oleh:

**ARIF ALFIYANTO**

**08.18.086**





**PROGRAM STUDI TEKNIK INFORMATIKA S-1**

**FAKULTAS TEKNOLOGI INDUSTRI**

**INSTITUT TEKNOLOGI NASIONAL MALANG**


**2014**

# LEMBAR PERSETUJUAN DAN PENGESAHAN

# RANCANG BANGUN APLIKASI *FISHING MAP* BERBASIS ANDROID

## SKRIPSI

***Disusun dan Diajukan untuk melengkapi dan memenuhi persyaratan guna mencapai Gelar Sarjana Teknik Komputer Strata Satu (S-1)***


**Disusun Oleh :**
**ARIF ALFIYANTO**
**08.18.086**


Diperiksa dan Disetujui,

| Dosen Pembimbing I | Dosen Pembimbing II |
|---|---|
| **Dr. Ir. Dhayal Gustopo, MT** | **Suryo Adi Wibowo, ST., MT** |
| **NIP.Y 1039400264** | **NIP.P 103000438** |

Ketua Program Studi Teknik Informatika S-1

**Joseph Dedy Irawan, ST., MT.**
**NIP. 197404162005031002**

**PROGRAM STUDI TEKNIK INFORMATIKA S-1**

**FAKULTAS TEKNOLOGI INDUSTRI**

**INSTITUT TEKNOLOGI NASIONAL MALANG**

**2014**

**PROGRAM STUDI TEKNIK INFORMATIKA S-1**
**FAKULTAS TEKNOLOGI INDUSTRI**
**INSTITUT TEKNOLOGI NASIONAL**
**MALANG**

---

## PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan dibawah ini :

Nama : Arif Alfiyanto

NIM : 08.18.086

Jurusan : TeknikInformatika S-1

Fakultas : Teknologi Industri

Institut Teknologi Nasional Malang

Menyatakan dengan sesungguhnya bahwa skripsi saya yang berjudul :

**"RANCANG BANGUN APLIKASI *FISHING MAP* BERBASIS ANDROID"**

Adalah skripsi saya sendiri bukan duplikat serta mengutip atau menyadur seluruhnya karya orang lain kecuali dari sumber aslinya.

Malang, Februari 2014

Yang membuat pernyataan


**Arif Alfiyanto**

*Teriring Ucapan Terima Kasih kepada*
*Ayah dan Ibu tercinta yang selalu memotivasi*

# RANCANG BANGUN APLIKASI *FISHING MAP* BERBASIS ANDROID

**Arif Alfiyanto**
**(08.18.086)**
**Jurusan Teknik Informatika S-1**
**Fakultas Teknologi Industri, Institut Teknologi Nasional Malang**
**Kampus II ITN Jalan Raya Karanglo KM 2 Malang**
**Email: arif.alviyanto@gmail.com**

**Dosen Pembimbing :I. Dr. Ir. Dhayal Gustopo, MT**
**II. Suryo Adi Wibowo, ST., MT**

***Abstrak***

*Memancing adalah aktifitas yang menyenangkan di semua kalangan dan usia dari dulu hingga sekarang. Memancing dulu hanya dikatakan sebagai hobi dan penghilang stres. Tapi, untuk saat ini memancing tidak hanya dikatakan sekedar hobi, namun bisa dikategorikan sebagai olahraga. Hingga pemancing mempunyai beberapa level atau tingkatan tersendiri, pemancing professional yang mempunyai peralatan yang sangat mahal dan sangat kuat untuk menaikkan ikan yang beratnya hingga berkilo-kilo beratnya sekalipun dan pemancing tradisional yang memiliki peralatan biasa, hingga seadanya dari alam berupa joran dari bambu serta umpan masih menggunakan cacing ataupun ikan hidup.*

*Aplikasi Fishing Map berbasis android adalah suatu aplikasi yang menyediakan dan menambah lokasi, serta Category jenis ikan. Asumsi yang mendasari terealisasinya Aplikasi Fishing Map berbasis android adalah adanya informasi yang baik sehingga memungkinkan terjadinya memberikan informasi yang tepat. Dalam tujuan ini peneliti membuat aplikasi Fishing Map untuk memberikan informasi kordinat untuk memancing sesuai dengan ikan yang di inginkan dan memberikan informasi yang tepat untuk memancing serta umpan apa yang cocok untuk menunjang si pemancing mendapatkan target ikan yang di inginkan pada tempat yang belum pernah di kunjungi sebelumnya.*

*Hasil dari pembuatan Aplikasi Mobile Fishing Map berbasis Android ini berjalan pada android versi Jellybean, Ginggerbread, dan Ice Cream Sandwitch (ICS). Semua fungsi dan tampilan sistem dapat berjalan dengan baik pada ukuran pixel 1024x768. Oleh karena itu Aplikasi Mobile Fishing Map berbasis android ini diharapkan dapat membantu pemancing dalam mencari target ikan yang di inginkan dan menggunakan umpan dengan tepat untuk mendapatkan target ikan tersebut.*

***Kata kunci** : Fishing map, (pemograman android), Android, Umpan Ikan, Fishing, Open Street Map.*

# KATA PENGANTAR

Dengan memanjatkan puji syukur kehadirat Tuhan Yang Maha Esa, karena atas rahmat dan hidayahnya sehingga penulis dapat menyelesaikan Laporan Skripsi yang berjudul "RANCANG BANGUN APLIKASI *FISHING MAP* BERBASIS ANDROID" ini dengan baik.

Laporan Skripsi ini merupakan salah satu persyaratan akademik yang harus dipenuhi oleh mahasiswa Institut Teknologi Nasional Malang untuk memperoleh gelar Strata 1 Program Studi Teknik Informatika, Institut Teknologi Nasional Malang.

Oleh karena itu, pada kesempatan ini dengan segala kerendahan hati perkenankanlah penulis mengucapkan terimah kasih kepada :

1. Bapak Ir. Soeparno Djiwo, MT.,selaku Rektor Institut Teknologi Nasional Malang.
2. Bapak Ir. H. Anang Subardi, MT., selaku Dekan Fakultas Teknologi Industri Institut Teknologi Nasional Malang.
3. Bapak Yoseph Dedy Irawan, ST, MT.,selaku Ketua Jurusan Teknik Informatika S-1 Institut Teknologi Nasional Malang.
4. Bapak Dr. Ir. Dhayal Gustopo, MT., selaku Dosen Pembimbing I Program Studi Teknik Informatika S-1 Institut Teknologi Nasional Malang.
5. Bapak Suryo Adi Wibowo, ST, MT., selaku Dosen Pembimbing II Program Studi Teknik Informatika S-1 Institut Teknologi Nasional Malang.
6. Rekan-rekan Teknik Informatika Institut Teknologi Nasional Malang serta berbagai pihak yang turut membantu dalam penyelesaian laporan ini.
7. Bapak Joko Purdiyanto dan Ibu Syarifah Mariamah, yang merupakan kedua orang tua saya sebagai pendukung utama dari segi moril maupun materiil.

Semoga apa yang telah disajikan dapat memberikan manfaat dan pengetahuan bagi para pembaca. Segala keritik dan saran yang membangun, diterima dengan senang hati sebagai tambahan ilmu dan pengetahuan.

Malang, Februari 2014

Penulis

# DAFTAR ISI

# DAFTAR TABEL

## DAFTAR GAMBAR

# BAB I
# PENDAHULUAN

## 1.1 Latar Belakang

Dalam olahraga memancing, mencari tempat dan target ikan yang di inginkan adalah hal utama. Terlebih, kebanyakan memancing tidak mendapatkan ikan sama sekali sering terjadi bukan disebabkan karena faktor keberuntungan si pemancing. Tetapi factor waktu yang tidak tepat, dimana ada saatnya ikan kawin, ikan mencari makan, dan ikan tidak mau makan karena berbagai hal ( cuaca , keadaan alam , dan tempat yang tidak di senangi ikan ).

Pada jaman yang maju ini sudah banyak menggunakan *smartphone* dengan telknologi canggih dan di dukung dengan aplikasi – aplikasi mobile yang baik dan bagus pula. Beberapa aplikasi mobile tersebut dapat menunjang pekerjaan si pemilik telepon pintar agar pekerjaan cepat selesai dengan mudah maupun mendapatkan informasi lebih cepat di saat dimanapun dibutuhkan.

*Fishing Map,* atau yang dimaksud peta memancing. Sebagai contoh beberapa pemancing hanya sekedar memancing di tempat yang acak. Dan tidak tahu ikan jenis apa yang nanti nya di dapat dan apakah tepat memancing di tempat tersebut. Tetapi dengan menggunakan peta memancing, bisa tahu daerah mana yang terdapat jenis ikan yang ingin di dapat dan apakah sudah tepat menggunakan umpan untuk memancing ikan yang di inginkan. Sehingga tidak perlu membuang – buang waktu untuk seharian menunggu hingga ikan mau memakan dan mencari tempat kesana kemari.

Maka dari itu di perlukan adanya aplikasi yang dapat menunjang di saat ingin melakukan olahraga memancing, yang dapat memberitahukan kapan jadwal memancing yang tepat dan dimana tempat memancing yang tepat dengan target yang tepat pula sesuai yang di inginkan oleh si pemancing

## 1.2 Rumusan Masalah

Berdasarkan pada latar belakang masalah diatas, maka didapatkan beberapa rumusan masalah yang di ambil untuk menunjang pembuatan aplikasi lebih spesifik yaitu :

1. Bagaimana cara membuat aplikasi *Fishing Map* berbasis android yang berguna untuk memberikan informasi.
2. Bagaimana menandai tempat memancing sesuai target jenis ikan yang di inginkan dan memberi informasi penggunaan umpan yang tepat.
3. Bagaimana menggunakan *OpenStreetMap* pada perangkat mobile berbasis android.

## 1.3 Batasan Masalah

Agar pembahasan dalam skripsi ini tidak meluas, maka ditentukan beberapa batasan masalah diantaranya, yaitu :

1. Aplikasi *Fishing Map* Berbasis Android ini dibuat hanya untuk berjalan pada sistem operasi Android versi api 11 hingga api 16 .
2. Pembuatan Aplikasi *Fishing Map* Berbasis Android ini dibangun mengunakan *Java Programming Language*.
3. Pada aplikasi *Mobile Fishing Map* hanya menampilkan posisi kordinat tempat memancing, jenis ikan yang ada pada kordinat di pilih ataupun yang sudah ada dan informasi umpan memancing sesuai jenis ikan yang ingin di dapat.
4. Adapun perangkat lunak yang digunakan untuk membangun aplikasi pada android adalah Eclipse.
5. Aplikasi menggunakan *database* SQLite
6. Aplikasi di buat hanya pada sisi pengguna (*user*) saja.

## 1.4 Tujuan

Tujuan dari penelitian untuk pembuatan Aplikasi yaitu, bertujuan Memberi informasi yang lengkap, tentang ikan apa di lokasi yang sudah di tandai dan umpan apa yang tepat untuk memancing ikan yang di inginkan atau dicari. Dengan tahap *update map* dan data informasi terlebih dahulu saat

memiliki akses *internet* yang di implementasikan pada aplikasi mobile berbasis android.

## 1.5 Metodologi Penelitian

Dalam menyusun penelitian skripsi, penulis membutuhkan data-data yang berhubungan dengan tema yang akan dikupas oleh penulis, yaitu mengenai konsep dan teori dasar rancang bangun aplikasi *fishing map* berbasis android. Dalam hal ini tentunya penulis membutuhkan data-data mengenai keuntungan serta kekurangan yang mempengaruhi sistem kerja di lapangan.

### 1.5.1 Pengambilan Sampel Data

1. Data Primer

Data primer merupakan data yang dikumpulkan oleh orang atau perseorangan secara langsung dari sumbernya. Contoh pengambilan data berupa teks hasil wawancara yang diperoleh melalui wawancara dengan narasumber yang dijadikan sampel penelitian. Data dapat direkam atau dicatat oleh peneliti.

2. Data sekunder

Data sekunder merupakan data yang tidak diperoleh secara langsung dari objek peneltian. Contoh pengambilan data berupa data-data yang sudah tersedia dan dapat diperoleh peneliti dengan cara membaca, melihat atau mendengarkan. Termasuk dalam kategori data ini ialah: data bentuk teks (dokumen, surat-surat), bentuk gambar (foto,animasi), bentuk suara (hasil rekaman kaset) dan kombinasi teks, gambar dan suara (film, video).

### 1.5.2 Desain Aplikasi

Bertujuan untuk mendesain aplikasi yang akan dirancang dan agar dapat mengidentifikasi komponen-komponen aplikasi yang akan di desain secara rinci. Tahap ini dilakukan setelah tahap analisis aplikasi selesai dilaksanakan, karena hasil desain terinci akan percuma dengan sia-sia bila tidak seseuai dengan apa yang diinginkan. Desain aplikasi ini meliputi flowchart atau teknik lain seperti sketsa dan lain-lain dalam mendukung desain sistem.

### 1.5.3 Implementasi

Setelah sistem dianalisis dan didesain secara rinci dan teknologi telah diseleksi dan dipilih, tahap implementasi sistem merupakan tahap meletakkan sistem supaya siap untuk dioperasikan. Tahap ini termasuk juga kegiatan pembuatan dan penerapan aplikasi program.

### 1.5.4 Uji Coba

Uji coba dilakukan untuk mengetahui apakah aplikasi yang dibuat sudah baik atau masih ada kesalahan untuk diperbaiki lagi,untuk mengetahui hasil uji coba yang baik maka harus dilakukan uji coba di lapangan.

## 1.6 Sistematika Penulisan

Sistematika peyusunan proposal ditunjukan untuk memberikan gambaran dan uraian dari proposal skripsi secara garis besar yang meliputi bab-bab sebagai berikut :

**BAB I. PENDAHULUAN**

Pada bab ini membahas tentang berbagai konsep dasar dan teori-teori yang berkaitan dengan topik dalam penelitian yang dilakukan dan hal-hal yang berguna dalam proses analisis permasalahan serta tinjauan terhadap penelitian-penelitian serupa yang telah pernah dilakukan sebelumnya.

**BAB II. LANDASAN TEORI**

Pada Bab ini membahas tentang Landasan Teori yang merupakan tinjauan pustaka, menguraikan teori-teori yang mendukung judul, dan pembahasan secara detail. Landasan teori dapat berupa definisi-definisi atau model yang langsung berkaitan dengan ilmu atau masalah yang diteliti. Pada bab ini juga dituliskan tentang software (komponen) yang digunakan dalam pembuatan Program atau keperluan saat penelitian.

**BAB III. ANALISIS DAN PERANCANGAN SISTEM**

Pada bab ini menganalisis tentang masalah yang dihadapi dalam pembuatan aplikasi *fishing map* berbasis android dan perancangan dalam pembuatan sistem tersebut.

**BAB IV. PENGUJIAN DAN ANALISA**

Bab ini menjelaskan tentang pengujian aplikasi secara umum maupun terperinci. Pengujian aplikasi secara umum akan membahas mengenai lingkungan uji coba untuk mengunakan aplikasi ini. Selanjutnya secara lebih terperinci dijelaskan dalam pengujian user, beserta langkah-langkah dalam uji coba tersebut, kemudian dianalisa kembali apakah telah sesuai dengan tujuan pembuatan pada bab I.

**BAB V. PENUTUP**

Pada Bab ini berisi kesimpulan dan saran. Kesimpulan didapat dari ulasan data-data penelitian, menyimpulkan bukti-bukti yang diperoleh dan akhirnya menarik intisari apakah hasil dapat dikerjakan, layak untuk untuk digunakan (diimplementasikan).

# BAB II
# LANDASAN TEORI

## 2.1 *Fishing Map*

Peta memancing ( *Fishing Map* ), yang pada umumnya memberi sejumlah informasi tempat dimana lokasi yang berpotensi memiliki ikan besar dan banyak serta kedalaman air, hingga umpan yang tepat untuk memancing ikan pada lokasi tersebut. Beberapa peta juga memberikan informasi beberapa tempat, mempunyai ikan jenis apa, menggunakan umpan apa, dan jarak kedalaman pada tempat ikan tersebut.

## 2.2 *Global Postioning System* (GPS)

GPS adalah singkatan dari *Global Positioning System*, sistem satelit yang dapat memberikan posisi di mana pun di dunia ini. Satelit GPS tidak mentransmisikan informasi posisi, yang ditransmisikan satelit adalah posisi satelit dan jarak penerima GPS dari satelit. Informasi ini diolah alat penerima GPS dan hasilnya ditampilkan. Penerima GPS memperoleh sinyal dari beberapa satelit yang mengorbit bumi. Satelit yang mengitari bumi pada orbit pendek ini terdiri dari 24 susunan satelit, dengan 21 satelit aktif dan 3 buah satelit sebagai cadangan. Dengan susunan orbit tertentu, maka satelit GPS bisa diterima diseluruh permukaan bumi dengan penampakan antara 4 sampai 8 buah satelit. GPS dapat memberikan informasi posisi dan waktu dengan ketelitian sangat tinggi.

Tiap satelit mengitari bumi kira-kira sekali dalam 12 jam dengan kecepatan sekitar 11.000 kilometer per jam. Satelit GPS mempunyai panel-panel pengumpul tenaga Matahari untuk membangkitkan energi listrik yang diperlukannya. Selain itu juga ada baterai yang menyimpan tenaga listrik dan mempergunakannya saat satelit tidak memperoleh sinar Matahari. Untuk mengetahui posisi dari GPS, diperlukan minimal 3 satelit. Pengukuran posisi GPS didasarkan oleh sistem pengukuran matematika yang disebut dengan Triliterasi. Yaitu pengukuran suatu titik dengan bantuan 3 titik acu.

## 2.3 Android

Beberapa tahun belakangan ini istilah Android sering sekali kita dengar, baca maupun kita lihat. Pada umumnya istilah Android sering dikaitkan dengan Ponsel, smartphone dan tablet. Penemu Android adalah Andy Rubin yang lahir pada tanggal 22 Juni 1946 di New Bedford, Amerika Serikat. Andy Rubin bersama-sama dengan Rich Miner, Nick Sears, dan Chris White mendirikan Android.inc dan apada Juli 2005 dibeli oleh Google.

Pengertian Android adalah sistem operasi berbasis Linux yang dipergunakan sebagai pengelola sumber daya perangkat keras, baik untuk ponsel, smartphone dan juga PC tablet. Secara umum Android adalah platform yang terbuka (Open Source) bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh berbagai piranti bergerak. Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android. Semenjak kehadirannya pada 9 Maret 2009, Android telah hadir dengan versi 1.1, yaitu sistem operasi yang sudah dilengkapi dengan pembaruan estetis pada apalikasinya, seperti jam alrm, voice search, pengiriman pesan dengan Gmail, dan pemberitahuan email. Hingga tahun 2012, Android telah berkembang dengan pesat. Dalam kurun 3 tahun Android telah diproduksi dalam versi, dan versi terakhir yang diproduksi disebut sebagai Android versi 4.1 atau Android Jelly Bean[3]. Logo Android dapat ditunjukan pada gambar 2.1 sebagai berikut :


Gambar 2.1 Logo Android

### 2.5.1 Karakteristik Android Berdasarkan Versi

Pada penghujung tahun 2010 diperkirakan hampir semua *vendor* seluler di dunia menggunakan Android sebagai *operating system*. Adapun versi-versi Android yang pernah dirilis adalah sebagai berikut:

1. Android versi 1.1 Android versi 1.1 ini merupakan generasi pertama android, dan diliris pada 9 Maret 2009 oleh google. Yang menjadi keunggulan dari Android ini adalah dilengkapinya dengan pembaruan estetis pada aplikasi, jam alarm, pencarian suara, pengiriman pesan dengan Gmail, dan pemberitahuan email.

2. Android versi 1.5 cupcake

   Untuk Android generasi ke dua ini memilki beberapa perubahan, adapun perubahannya antara lain adalah kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

3. Android versi 1.6 Donut

   Di Android versi 1.6 ini yang terkenal dengan nama Android Donut terdapat update untuk menampilkan proses pencarian yang lebih baik dibandingkan 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; pengadaan resolusi VWGA.

4. Android versi 2.0/2.1 Eclair

   Android versi ini yang merupakan keluaran atau generasi yang ke empat ini yang dirilis pada 3 Desember 2009, dan terdapat beberapa perubahan yang ditujukan untuk pengoptimalan pada hardware. Perubahan yang dilakukan

adalah 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.

5. Android versi 2.2 Froyo

Android versi 2.2 froyo ini sempat mendapatkan perhatian dari pengguna karena kelebihannya dan telah banyak perubahan dari generasi yang pertama, adapun android ini dikeluarkan pada 20 Mei 2010 dengan terdapat perubahan pada Android versi 2.2, antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi dua sampai lima kali lebih cepat, integrasi V8 JavaScript engine yang dipakai Google Chrome yang dapat mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.

6. Android versi 2.3 Gingerbread

Untuk Android versi ini dirilis sekitar 6 bulan setelah android froyo tepatnya pada 6 Desember 2010, sedangkan perubahan yang terdapat pada android versi 2.3 gingerbread ini adalah dengan adanya 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), dan dukungan jumlah kamera yang lebih dari satu.

7. Android versi 3.0/3.1 Honeycomb

Android versi 3.0/3,1 ini terdapat perlakuan khusus, karena android yang satu ini memang lebih diperuntukan untuk tablet yang menggunakan layar yang lebih lebar. Selain itu User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras (hardware) untuk grafis. Dan yang menjadi

tablet yang pertama kali didukung oleh android honeycomb ini adalah Motorola Xoom.

8. Android versi 4.0 Ice Cream

Android versi 4.0 Ice cream ini dirilis pada pada akhir tahun 2011 oleh Google, dan yang menjadi perhatian adalah android yang satu ini hanya android berbasis open source jadi siapapun bisa mengembangkannya dan jika anda suatu saat nanti bisa mengembangkan android maka anda bisa juga membuat simbol-simbolnya.

9. Android versi 4.1 JELLY BEAN

Android JELLY BEAN dikembangkan melalui hasil survey ICS (Ice Cream Sandwich) yang lebih dulu dirilis pada banyak merk tablet. Dan Android yang paling muda ini memiliki beberapa keunggulan yaitu:

a. Untuk kecepatan dalam android ini memang sudah mempunyai performa yang lebih baik, karena prosesor yang sudah disematkan kedalam tablet sebenarnya sehingga bisa dipompa dengan maksimal. serta ditambah dengan tripple buffering sehingga menghasilkan hasil grafis yang maksimal dan halus.

b. Di Android ini untuk menyimpan foto kontak yang lebih besar 720 x 720, sehingga membuat kontak lebih kaya dan lebih pribadi Android jelly bean ini dapat membantu mengelola aplikasi penggunaan data secara lebih tepat bila perangkat terhubung ke jaringan sehingga mengetahui kecepatan akses, termasuk penarikan ke mobile hotspot.

c. Android jelly bean ini menggunakan teknologi berbasis NFC populer yang memungkinkan pengguna langsung terkoneksi, hanya dengan menyentuh dua NFC-enabled telepon bersama. Sehingga lebih mudah untuk berbagi foto, video, atau muatan lain dengan memanfaatkan Bluetooth untuk transfer data.

d. Terdapat Customize Widgets, artinya diberikan kemampuan untuk merubah ukuran widget yang biasanya standart, pengguna dapat mengatur ukuran widgets yang ditampilkan di halaman perangkat sesuai dengan keinginannya.

## 2.4 Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*). Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++,Cobol, Python, Perl, PHP, dan lain sebagainya. Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya. Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*[4]. Logo Eclipse dapat ditunjukan pada gambar 2.3 sebagai berikut :


Gambar 2.2 Logo Eclipse

Elipse+AVR plugin, dengan tambahan plugin tersebut kita dapat memprogram mikrokontroler AVR menggunakan IDE ini, selain itu keuntungan menggunakan

eclipse ialah dapat bekerja di berbagai sistem operasi seperti Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X[4].

Berikut ini adalah sifat dari Eclipse:

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

### 2.6.1 *Android Virtual Devices* (AVD)

*Android Virtual Devices* (AVD) adalah konfigurasi dari emulator sehingga kita dapat menjalankan perangkat Android sesuai model yang dipilih, misal Android 1.5 atau 2.2. Untuk dapat menjalankan emulator, Anda harus terlebih dahulu memiliki Android *Software Development Kit* (SDK)[4].

Setiap AVD terdiri dari:

1. Sebuah profil perangkat keras. Anda dapat mengatur opsi untuk menentukan fitur hardware emulator. Misalnya, Anda dapat menentukan apakah menggunakan perangkat kamera, apakah menggunakan keyboard QWERTY fisik atau tidak, berapa banyak memori internal, dan lain-lain.
2. Sebuah pemetaan versi Android. Anda dapat menentukan versi dari platform Android akan berjalan pada emulator.
3. Pilihan lainnya. Anda dapat menentukan skin yang ingin Anda gunakan pada emulator, yang memungkinkan Anda menentukan dimensi layar, tampilan, dan sebagainya. Anda juga dapat menentukan SD Card virtual untuk digunakan dengan di emulator.

Anda dapat membuat AVD sebanyak apapun dibutuhkan, berdasarkan jenis konfigurasi model dan jenis Android yang dipilih.

## 2.5 SQLite

SQLite merupakan sebuah sistem manajemen basisdata relasional yang bersifat *ACID-compliant* dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. SQLite merupakan proyek yang bersifat public domain yang dikerjakan oleh D. Richard Hipp. Tidak seperti pada paradigma client-server umumnya, Inti SQLite bukanlah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian integral dari sebuah program secara keseluruhan. Sehingga protokol komunikasi utama yang digunakan adalah melalui pemanggilan API secara langsung melalui bahasa pemrograman. Mekanisme seperti ini tentunya membawa keuntungan karena dapat mereduksi *overhead*, *latency times*, dan secara keseluruhan lebih sederhana. Seluruh elemen basisdata (definisi data, tabel, indeks, dan data) disimpan sebagai sebuah file. Kesederhanaan dari sisi disain tersebut bisa diraih dengan cara mengunci keseluruhan file basis data pada saat sebuah transaksi dimulai [4].

### 2.7.1 Fitur-fitur [4]

Pustaka SQLite mengimplementasikan hampir seluruh elemen-elemen standar yang berlaku pada SQL-92, termasuk transaksi yang bersifat *atomic*, konsistensi basisdata, isolasi, dan durabilitas (dalam bahasa inggris lebih sering disebut ACID), trigger, dan kueri-kueri yang kompleks. Tidak ada pengecekan tipe sehingga data bisa dientrikan dalam bentuk string untuk sebuah kolom bertipe integer. Beberapa kalangan melihat hal ini sebagai sebuah inovasi yang menambah nilai guna dari sebuah basisdata, utamanya ketika digunakan dalam bahasa pemrograman berbasis script (PHP, Perl), sementara kalangan lain melihat hal tersebut sebagai sebuah kekurangan.

Beberapa proses ataupun thread dapat berjalan secara bersamaan dan mengakses basisdata yang sama tanpa mengalami masalah. Hal ini disebabkan karena akes baca data dilakukan secara paralel. Sementara itu akses tulis data

hanya bisa dilakukan jika tidak ada proses tulis lain yang sedang dilakukan; jika tidak, proses tulis tersebut akan gagal dan mengembalikan kode kesalahan (atau bisa juga secara otomatis akan mencobanya kembali sampai sejumlah nilai waktu yang ditentukan habis). Hanya saja ketika sebuah tabel temporer dibuat, mekanisme penguncian pada proses multithread akan menyebabkan masalah. Update yang terkini (versi 3.3.4) dikatakan telah memperbaiki masalah ini.

Sebuah program yang mandiri dinamakan sqlite disediakan dan bisa digunakan untuk mengeksekusi kueri dan memanajemen file-file basisdata SQLite. Program tersebut juga merupakan contoh implementasi penulisan aplikasi yang menggunakan pustaka SQLite.

### 2.7.2 Integrasi Bahasa Lain [4]

1. SQLite termasuk dalam framework REALbasic, yang memungkinkan aplikasi yang dikembangkan dengan menggunakan REALbasic dapat memanfaatkan basisdata SQLite.
2. Pustaka SQLite bisa digunakan secara langsung pada bahasa C/C++, namun untuk Tcl dan beberapa bahasa pemrograman berbasis script juga tersedia.
3. Modul DBI/DBD untuk Perl juga tersedia pada CPAN, *DBD:SQLite*, namun modul ini bukanlah antarmuka dengan SQLite melainkan memasukkan SQLite secara keseluruhan dalam modul tersebut.
4. Modul Python juga tersedia (PySQLite) yang diimplementasikan pada DB API Python versi 2.0 (PEP 249)
5. PHP dimulai dengan PHP5 telah memasukkan SQLite, versi PHP4 sebelumnya bisa juga digunakan untuk mengakses SQLite, namun modul SQLite tidak dimasukan secara standar bawaan.
6. Dimulai Lazarus versi 0.9.8 dan Free Pascal 2.0.0, SQLite didukung untuk digunakan oleh para programmer. Tutorial tersebut bisa didapatkan disini.
7. Meskipun borland tidak mempaketkan SQLite secara standar bawaan, Delphi sudah mendukung SQLite juga menggunakan pustaka

yang dibuat oleh pihak ketiga (Aducom dan Zeos). Banyak program administrasi basisdata SQLite baik komersil ataupun bebas yang menawarkan kemudahan-kemudahan dalam memanajemen basisdata SQLite terdapat di internet dibuat dengan menggunakan bahasa ini.

8. SQLite juga termasuk dalam paket yang dibundle secara standar bawaan pada Mac OS X, dan digunakan sebagai salah satu pilihan mekanisme penyimpanan data pada API Apple.

## 2.8 Java

Java adalah bahasa pemrograman tingkat tinggi yang berorientasi objek dan program java tersusun dari bagian yang disebut kelas. Kelas terdiri atas metode-metode yang melakukan pekerjaan dan mengembalikan informasi setelah melakukan tugasnya. Para pemrogram Java banyak mengambil keuntungan dari kumpulan kelas di pustaka kelas Java, yang disebut dengan *Java Application Programming Interface* (API). Kelas-kelas ini diorganisasikan menjadi sekelompok yang disebut paket *(package).* Java API telah menyediakan fungsionalitas yang memadai untuk menciptakan *applet* dan aplikasi canggih. Jadi ada dua hal yang harus dipelajari dalam Java, yaitu mempelajari bahasa Java dan bagaimana mempergunakan kelas pada Java API. Kelas merupakan satu-satunya cara menyatakan bagian eksekusi program, tidak ada cara lain. Pada Java program javac untuk mengkompilasi file kode sumber Java menjadi kelas-kelas *bytecode.* File kode sumber mempunyai ekstensi *.java. Kompilator javac menghasilkan file *bytecode* kelas dengan ekstensi *.class. Interpreter merupakan modul utama sistem Java yang digunakan aplikasi Java dan menjalankan program *bytecode* Java.

Beberapa keunggulan java yaitu java merupakan bahasa yang sederhana. Java dirancang agar mudah dipelajari dan digunakan secara efektif. Java tidak menyediakan fitur-fitur rumit bahasa pemrograman tingkat tinggi, serta banyak pekerjaan pemrograman yang mulanya harus dilakukan manual, sekarang digantikan dikerjakan Java secara otomatis seperti dealokasi memori. Bagi

pemrogram yang sudah mengenal bahasa C++ akan cepat belajar susunan bahasa Java namun harus waspada karena mungkin Java mengambil arah (semantiks) yang berbeda dibanding C++. Java merupakan bahasa berorientasi objek *(OOP)* yaitu cara ampuh dalam pengorganisasian dan pengembangan perangkat lunak. Pada OOP, program komputer sebagai kelompok objek yang saling berinteraksi. Deskripsi ringkas OOP adalah mengorganisasikan program sebagai kumpulan komponen, disebut objek. Objek-objek ini ada secara independen, mempunyai aturan-aturan berkomunikasi dengan objek lain dan untuk memerintahkan objek lain guna meminta informasi tertentu atau meminta objek lain mengerjakan sesuatu. Kelas bertindak sebagai modul sekaligus tipe. Sebagai tipe maka pada saat jalan, program menciptakan objek-objek yang merupakan instan-instan kelas. Kelas dapat mewarisi kelas lain. Java tidak mengijinkan pewarisan jamak namun menyelesaikan kebutuhan pewarisan jamak dengan fasilitas antarmuka yang lebih elegan.

Seluruh objek diprogram harus dideklarasikan lebih dulu sebelum digunakan. Ini merupakan keunggulan Java yaitu *Statically Typed.* Pemaksaan ini memungkinkan kompilator Java menentukan dan melaporkan terjadinya pertentangan (ketidakkompatibelan) tipe yang merupakan barikade awal untuk mencegah kesalahan yang tidak perlu (seperti mengurangkan variabel bertipe integer dengan variabel bertipe string). Pencegahan sedini mungkin diharapkan menghasilkan program yang bersih. Kebaikan lain fitur ini adalah kode program lebih dapat dioptimasi untuk menghasilkan program berkinerja tinggi. Java menggunakan model pengamanan tiga lapis *(three-layer security model)* untuk melindungi sistem dari *untrusted Java code.* Pertama, *bytecode verifier* membaca *bytecode* sebelum dijalankan dan menjamin *bytecode* memenuhi aturan-aturan dasar bahasa Java. Kedua, *class loader* menangani pemuatan kelas Java ke *runtime interpreter.* Ketiga, manajer keamanan menangani keamanan tingkat aplikasi dengan mengendalikan apakah program berhak mengakses sumber daya seperti sistem file, port jaringan, proses eksternal dan sistem *window. Platform independence* adalah kemampuan program bekerja di sistem operasi 'yang berbeda. Bahasa Java merupakan bahasa yang secara sempurna tidak bergantung platform. Tipe variabel Java mempunyai ukuran sama di semua platform sehingga

variabel bertipe integer berukuran sama tidak peduli dimana program java dikompilasi. Begitu telah tercipta file .class dengan menggunakan kompilator Java di platform manapun, maka file .class tersebut dapat dijalankan di platform manapun. Jadi "dimanapun dibuat, dimanapun dapat dijalankan". Slogan ini biasa diringkas sebagai *Write Once, Run Anywhere (WORA).*

Java termasuk bahasa *Multithreading*. *Thread* adalah untuk menyatakan program komputer melakukan lebih dari satu tugas di satu waktu yang sama. Java menyediakan kelas untuk menulis program *multithreaded*, program mempunyai lebih dari satu *thread* eksekusi pada saat yang sama sehingga memungkinkan program menangani beberapa tugas secara konkuren. Program Java melakukan *garbage collection* yang berarti program tidak perlu menghapus sendiri objek-objek yang tidak digunakan lagi. Fasilitas ini mengurangi beban pengelolaan memori oleh pemrogram dan mengurangi atau mengeliminasi sumber kesalahan terbesar yang terdapat di bahasa yang memungkinkan alokasi dinamis. Java mempunyai mekanisme *exception-handling* yang ampuh. *Exception-handling* menyediakan cara untuk memisahkan antara bagian penanganan kesalahan dengan bagian kode normal sehingga menuntun ke struktur kode program yang lebih bersih dan menjadikan aplikasi lebih tegar. Ketika kesalahan yang serius ditemukan, program Java menciptakan *exception*. *Exception* dapat ditangkap dan dikelola program tanpa resiko membuat sistem menjadi turun. Program Java mendukung *native method* yaitu fungsi ditulis di bahasa lain, biasanya C/C++. Dukungan *native method* memungkinkan pemrogram menulis fungsi yang dapat dieksekusi lebih cepat dibanding fungsi ekivalen di java. *Native method* secara dinamis akan di-*link* ke program java, yaitu diasosiasikan dengan program saat berjalan. Selain itu keuntungan menggunakan bahasa pemrograman Java antara lain memori pada Java secara otomatis dilengkapi *garbage collector* yang berfungsi mendealokasi memori yang tidak diperlukan. Tidak ada lagi upaya yang dilakukan pemrogram untuk melakukan *dispose*(). Kita tidak lagi dibebani urusan korupsi memori. Java menerapkan array sebenarnya, menghilangkan keperluan aritmatika pointer yang berbahaya dan mudah menjadi salah. Menghilangkan pewarisan jamak *(multiple inheritance)* diganti fasilitas antarmuka. Dan mudah dijalankan diberbagai platform.

*Grafical User Interface* (GUI) adalah salah satu kemampuan Java dalam mendukung dan manajemen antarmuka berbasis grafis. Tampilan grafis yang akan ditampilkan terhubung dengan program serta tempat penyimpanan data. Elemen dasar di Java untuk penciptan tampilan berbasis grafis adalah dua paket yaitu AWT dan Swing. *Abstract Windowing Toolkit* (AWT), atau disebut juga *"Another Windowing Toolkit"*, adalah pustaka *windowing* bertujuan umum dan *multiplatform* serta menyediakan sejumlah kelas untuk membuat GUI di Java. Dengan AWT, dapat membuat window, menggambar, manipulasi gambar, dan komponen seperti *Button, Scrollbar, Checkbox, TextField,* dan menu *pull-down.* Penggunaan komponen AWT ditandai dengan adanya instruksi : *import java.awt.*;* Swing merupakan perbaikan kelemahan di AWT. Banyak kelas swing menyediakan komponen alternatif terhadap AWT. Contohnya kelas JButton swing menyediakan fungsionalitas lebih banyak dibanding kelas *Button.* Selain itu komponen swing umumnya diawali dengan huruf "J", misalnya JButton, JTextField, JFrame, JLabel, JTextArea, JPanel, dan sebagainya. Teknologi swing menggunakan dan memperluas gagasan-gagasan AWT. Sementara, penggunaan komponen Swing ditandai dengan adanya instruksi : *import javax.swing.*;*. Beberapa perbedaan AWT dan Swing, AWT merupakan komponen *heavyweight* (kelas berat) sedangkan Swing *lightweight* (kelas ringan). Swing memiliki lebih banyak komponen. Fasilitas Swing *Look and Feel* : Metal, Windows, Motif. Komponen Swing berdasar *model-view*, yaitu suatu cara pengembangan komponen dengan pemisahan penyimpanan dan penanganan data dari representasi visual data. Bahasa pemrograman Java merupakan salah satu bahasa pemrograman yang umum digunakan untuk mengembangkan aplikasi basis data yang dibuat menggunakan MySQL.

### 2.8.1 Kelebihan Java

1. *Multiplatform.* Kelebihan utama dari Java ialah dapat dijalankan di beberapa *platform* / sistem operasi komputer, sesuai dengan prinsip *tulis sekali, jalankan di mana saja.* Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / *bytecode*) sekali lalu hasilnya

dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis java dikerjakan diatas operating system Linux tetapi dijalankan dengan baik di atas Microsoft Windows. Platform yang didukung sampai saat ini adalah Microsoft Windows, Linux, Mac OS dan Sun Solaris. Penyebabnya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs Java) untuk meninterpretasikan *bytecode* tersebut.

2. OOP (*Object Oriented Programming* - Pemrogram Berorientasi Objek)
3. Perpustakaan Kelas Yang Lengkap, Java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas Java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.
4. Bergaya C++, memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke Java. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke Java. Universitas-universitas di Amerika Serikat juga mulai berpindah dengan mengajarkan Java kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer.
5. Pengumpulan sampah otomatis, memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

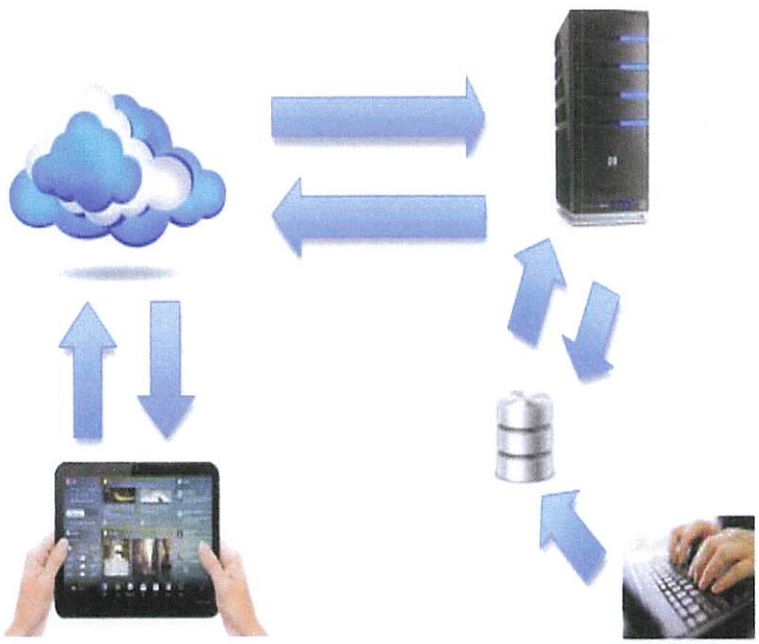
# BAB III
# ANALISA DAN PERANCANGAN APLIKASI

Pada bab ini akan dibahas mengenai perancangan aplikasi yang digunakan dan langkah-langkah yang diterapkan dalam rancang bangun aplikasi *fishing map* berbasis android.

## 3.1. Perancangan Sistem Perangkat Lunak

Aplikasi yang akan dibuat adalah sebuah aplikasi yang dapat update map secara langsung dan *realtime* ketika mempunyai koneksi internet. Sebelum melakukan proses update data dan peta, aplikasi mengkoneksikan jaringan ke internet terlebih dahulu.

Setelah itu server mengkomfirmasi memberitahukan map dan informasi terbaru pada aplikasi. Data yang ada pada server, didapat dari survey dan di input oleh admin dan share otomatis dari setiap aplikasi yang di tanam pada smartphone si user, dari penandaan lokasi dan informasi jenis ikan yang ada pada lokasi yang di tandai. Setelah itu diolah kembali oleh admin menjadi data yang rapi, dan lengkap.



Gambar 3.1.

Proses Aplikasi saat melakukan pertukaran data dan update map menggunakan koneksi Internet

## 3.2 Kebutuhan Sistem

Kebutuhan sistem yang diperlukan dalam pembuatan aplikasi *fishing map* berbasis android ini terbagi menjadi dua macam yaitu *software* dan *hardware*, dimana keduanya saling mendukung satu sama lain.

### 3.2.1 Software

Adapun perangkat lunak yang dibutuhkan dalam pembuatan aplikasi *fishing map* berbasis android meliputi:

i. Sistem Operasi Mac Os X 10.8.2 Mountain Lion
ii. Bahasa pemrogramman Java
iii. OmniGraffle Professional
iv. Android Virtual Devices (AVD) sebagai emulator sehingga kita dapat menjalankan perangkat Android sesuai model yang dipilih.

### 3.2.2 Hardware

Sedangkan perangkat keras yang digunakan dalam pembuatan aplikasi *fishing map* berbasis android meliputi:

1. Prosessor Intel(R) Core(TM) i5-2435M CPU @ 2.40GHz
2. RAM 4 GB 1333 MHz DDR3
3. Harddisk 500 GB

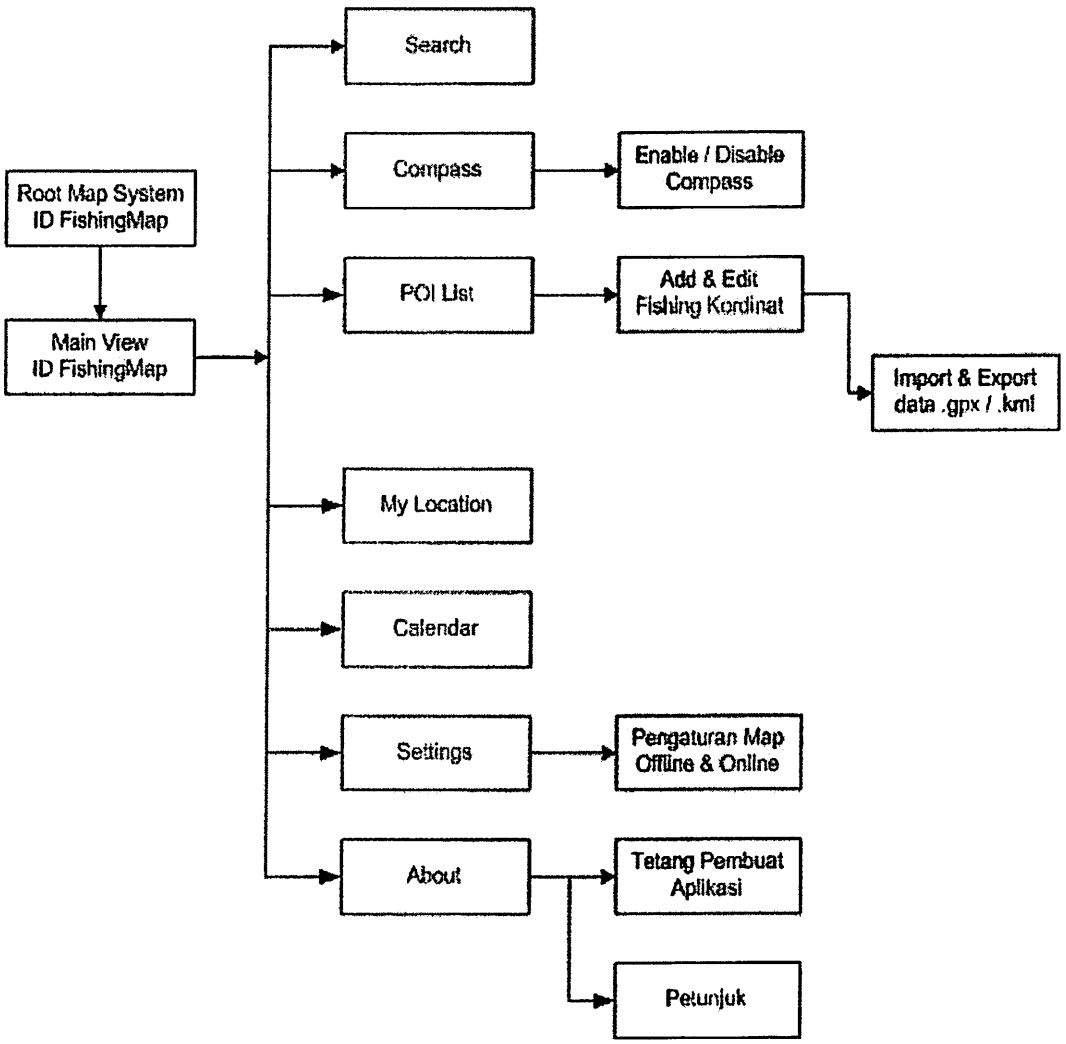
## 3.3 Arsitektur Sistem

Di dalam tahap rancangan ini semua permasalahan yang saling berelasi atau berhubungan akan diformulasikan sesuai dengan software / bahasa pemrograman yang akan digunakan untuk memaparkan hubungan relasional tersebut sesuai dengan bentuk format yang digunakan oleh sistem analisa. Dalam tahap ini sering disebut juga basis pengetahuan.

### 3.3.1 *Root Map Application*

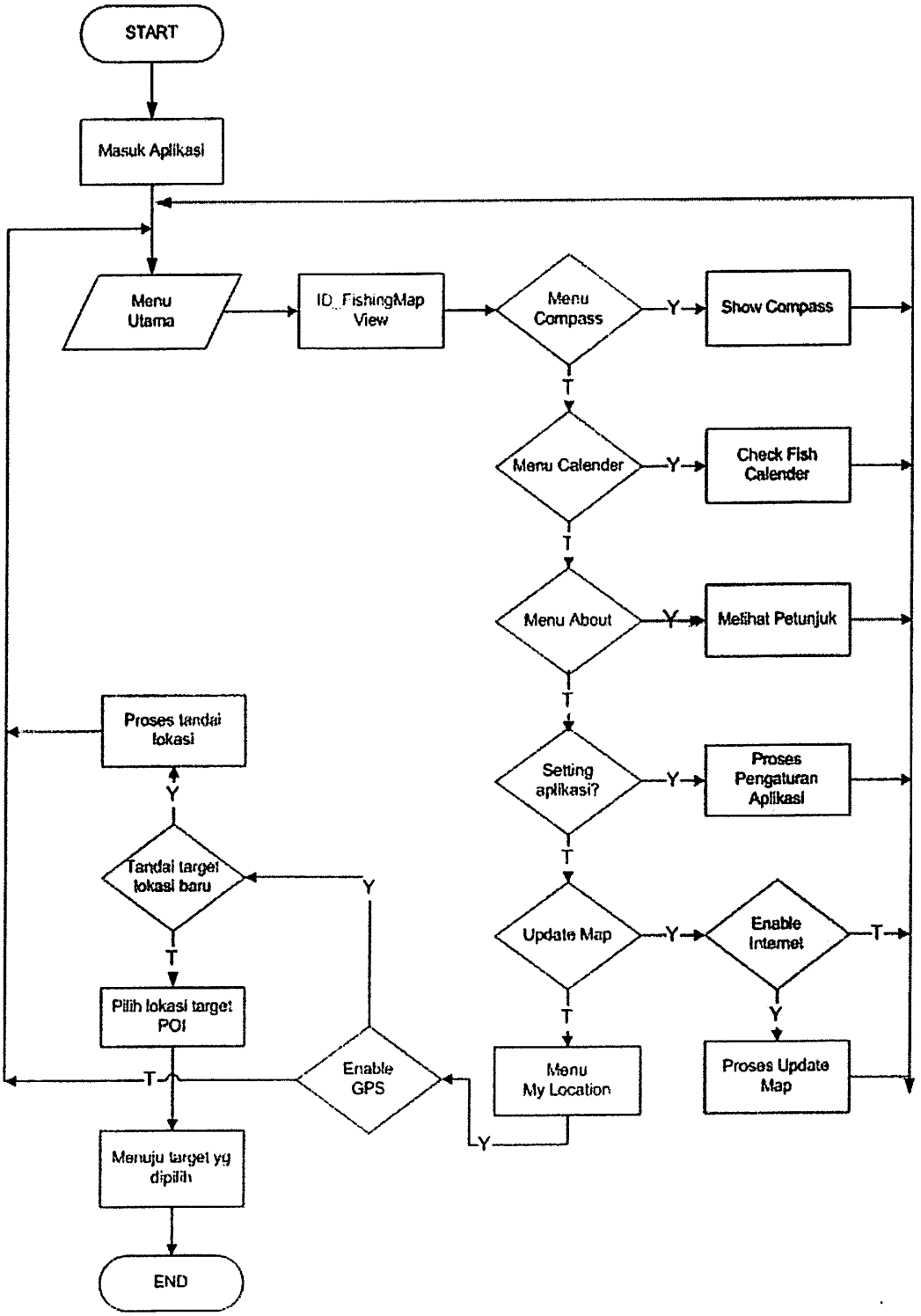
Dalam perancangan struktur menu diawali dengan masuk ke menu utama dimana terdapat 3 sub menu yang terdiri dari *Search, Compass, POI List, My Location, Calendar, Settings,* dan *About.* berdasarkan kriteria yang diinginkan

oleh pengguna, dan tutorial aplikasi. Struktur menu aplikasi di tunjukkan pada Gambar 3.2 berikut :




Gambar 3.2 *Root Map System* Aplikasi

### 3.3.2 Flowchart User



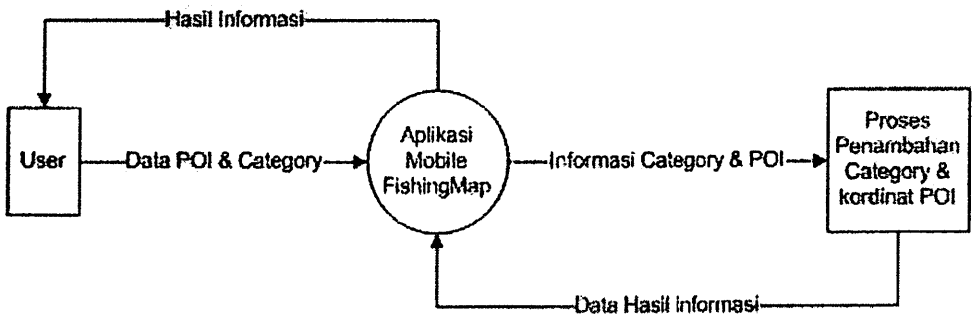

Gambar 3.3 Flowchart User

Penjelasan flowchart user yang di tunjukkan pada Gambar 3.3 di atas adalah sebagai berikut :

User membuka aplikasi terlebih dahulu sebelum menuju ke menu utama, setelah berhasil, terdapat pilihan menu slide show button yaitu menu *search*, menu *compass*, menu *POI*, menu *my location,* menu *calendar*, menu *settings,* menu *about.* Pada menu *POI* aplikasi menampilkan daftar kordinat target yang telah di tandai. User dapat mengedit, menambah *category* jenis ikan, menambah kordinat target yang baru.

### 3.3.3 Data Flow Diagram (DFD) Level 0

DFD level 0 pada Gambar 3.4. di bawah merupakan data flow diagram dari rancang bangun aplikasi *fishing map* berbasis android.




Gambar 3.4. DFD Level 0

Dari diagram konteks dalam gambar 3. dapat dijabarkan penjelasan lebih detail mengenai entitas, proses dan data yang digunakan sebagai berikut:

1. Entitas

   Merupakan pelaku-pelaku yang berperan dalam kegiatan atau aktifitas yang dilakukan oleh sistem atau aplikasi. Dalam aplikasi ini terdapat 1 entitas, yaitu:

   *a. User*

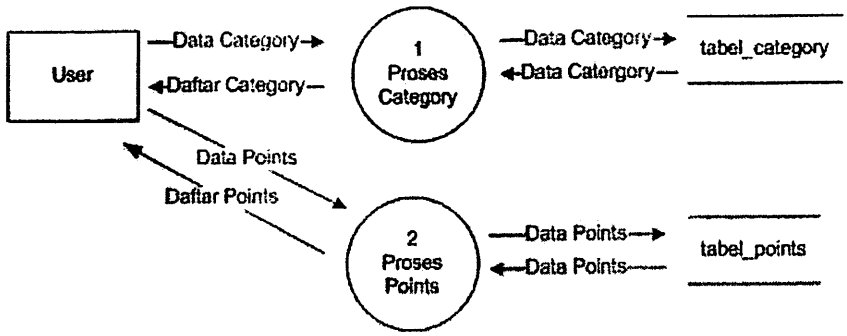
2. Proses

   Menampilkan kegiatan-kegiatan yang dilakukan dalam sebuah sistem. Namun pada diagram konteks biasanya satu proses utama dinamakan dengan nama aplikasi atau sistem tersebut. Proses utama dalam aplikasi ini adalah Aplikasi mobile Fishing Map.

3. Data

Merupakan sebuah informasi yang belum diolah atau informasi mentah. Dalam aplikasi ini terdapat beberapa data yang digunakan, yaitu Category, POI , dan kordinat.

### 3.3.4 Data Flow Diagram (DFD) Level 1




Gambar 3.5. DFD level 1

Pada gambar 3.4. dapat dijabarkan input dan output data dari masing-masing entitas dan proses sebagai berikut:

a. *User*

Input: Data Category dan Data Points

b. Proses Category

Input: Data Category

Output: Data Category, Daftar Category

c. Proses Points

Input: Data Points

Output: Data Points, Daftar Points

### 3.3.5 Desain Database

Dalam perancangan tabel dijelaskan tabel-tabel yang terdapat dalam database. Adapun contoh perancangan tabel seperti dibawah ini:

a. Tabel Data

Tabel ini berfungsi untuk menampilkan data kapal feri dari tabel data. Struktur tabel data dapat ditunjukan dalam tabel 3.1 sebagai berikut :

Tabel 3.1 Tabel Category

| No | Field | Tipe Data |
|---|---|---|
| 1 | categoryid | Null |
| 2 | name | String ( Null ) |
| 3 | hidden | False |
| 4 | iconid | Null |
| 5 | minzoom | Null |

b. Tabel Kriteria

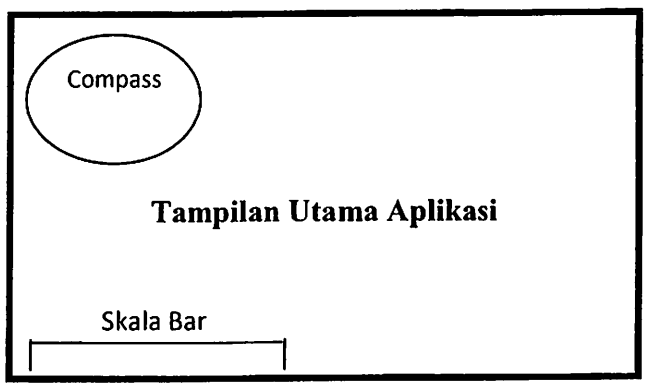
Tabel ini berfungsi untuk menampilkan data kriteria dari tabel kriteria. Struktur tabel kriteria dapat ditunjukan dalam tabel 3.2 sebagai berikut :

Tabel 3.2 Tabel Points

| No | Field | Tipe Data |
|---|---|---|
| 1 | pointid | Null |
| 2 | name | String ( Null ) |
| 3 | descr | String ( Null ) |
| 4 | lat | Null |
| 5 | lon | Null |

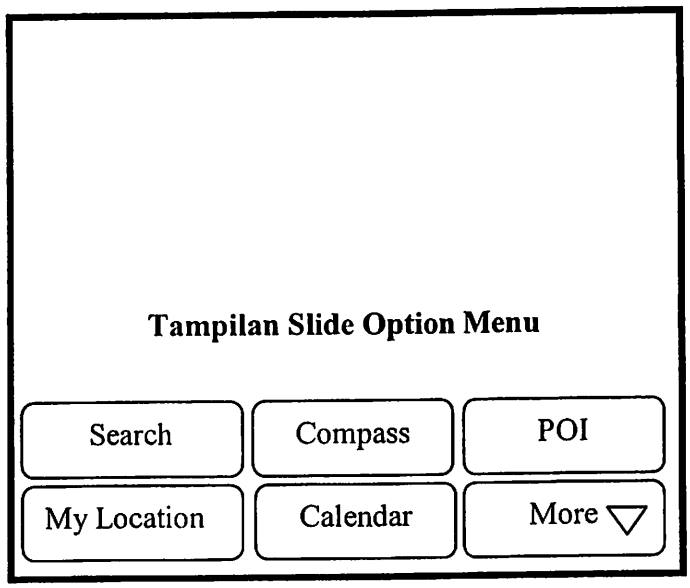
### 3.3.6 Desain Interface

Desain antarmuka pemakai (*Desain User Interface*) adalah rancangan tampilan program yang dapat dilihat dan bisa dijadikan pandangan awal bagi pengguna. Rancangan awal tampilan aplikasi *Fishing Map* berbasis android dapat dilihat sebagai berikut:



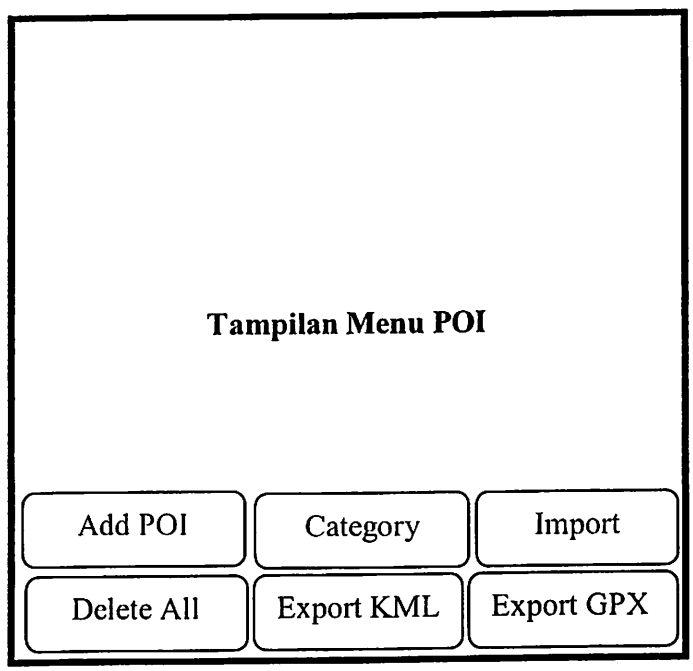

Gambar 3.6 Tampilan utama aplikasi

Pada gambar 3.6 merupakan tampilan awal aplikasi ketika aplikasi dijalankan.



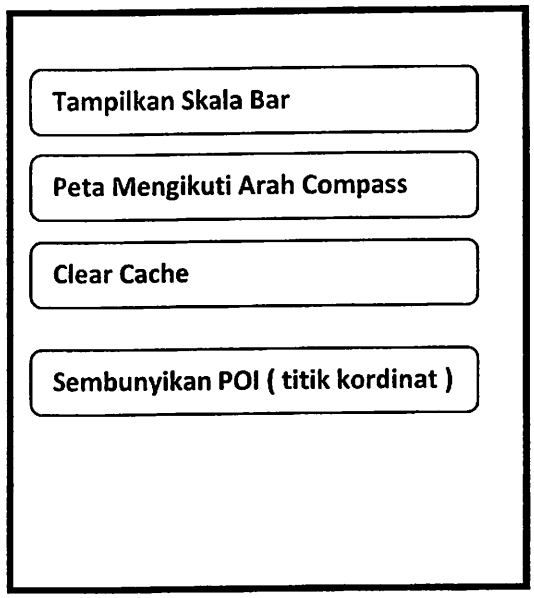

Gambar 3.7 Tampilan Slide Option Menu

Pada gambar 3.7 menampilkan menu-menu yang ada dalam aplikasi yang nantinya akan dipilih oleh pengguna aplikasi.



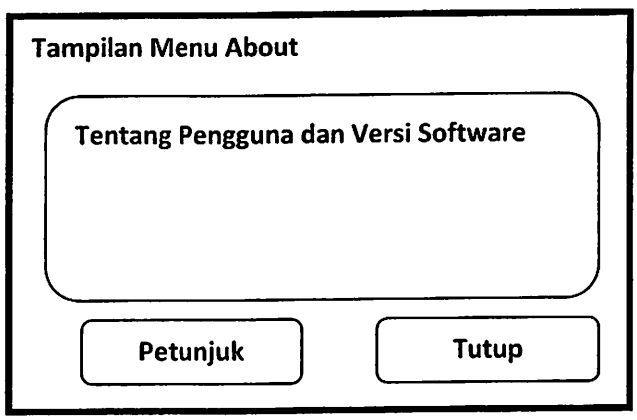

Gambar 3.8 Tampilan Menu POI

Pada gambar 3.8 menampilkan menu-menu yang ada dalam POI yang nantinya akan dipilih oleh pengguna aplikasi. Untuk menambahkan titik kordinat, menambah *category* jenis ikan, menghapus titik kordinat, import dan export data kordinat.




Gambar 3.9 Tampilan *Settings* aplikasi

Pada gambar 3.9 menampilkan list beberapa pengaturan aplikasi yang nantinya akan dipilih oleh pengguna. Untuk menampilkan skala bar, pengaturan peta, menghapus cache update peta yang lama dan menyembunyikan titik kordinat POI.




Gambar 3.10 Tampilan *About* aplikasi

Pada gambar 3.10 menampilkan tentang pembuat aplikasi dan petunjuk penggunaan aplikasi.

# BAB IV

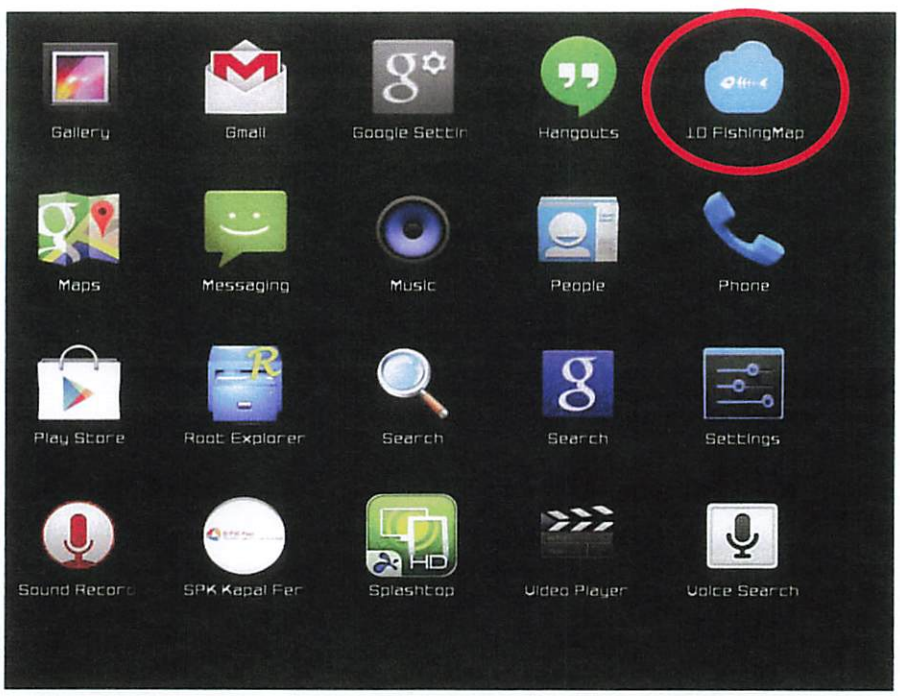
# IMPLEMENTASI DAN PENGUJIAN

## 4.1 Hasil Implementasi

Implementasi sistem adalah proses penerapan racangan sistem yang telah dibuat menjadi suatu aplikasi yang bisa dijalankan pada kenyataannya. Implementasi sistem berfungsi untuk menerapkan sistem sesuai dengan tujuan sistem. Disamping implementasi berfungsi untuk menerapkan sistem, fungsi lainya adalah untuk mengetahui sejauh mana keberhasilan dari rancangan yang telah dibuat. Rancang bangun aplikasi mobile fishing map memiliki beberapa kriteria yang harus mendukung terwujudnya implementasi sistem.

### 4.1.1 Tampilan Logo Aplikasi

Saat di jalankan pada sebuah *device* tablet maka akan muncul tampilan seperti pada gambar 4.1. yaitu icon atau logo aplikasi dimana merupakan hasil program yang telah terinstall di tablet, disini aplikasi bisa diketahui berjalan dengan baik atau tidak dari hasil yang ditampilkan melalui *device* tablet tersebut.




Gambar 4.1 Logo aplikasi pada *device* tablet

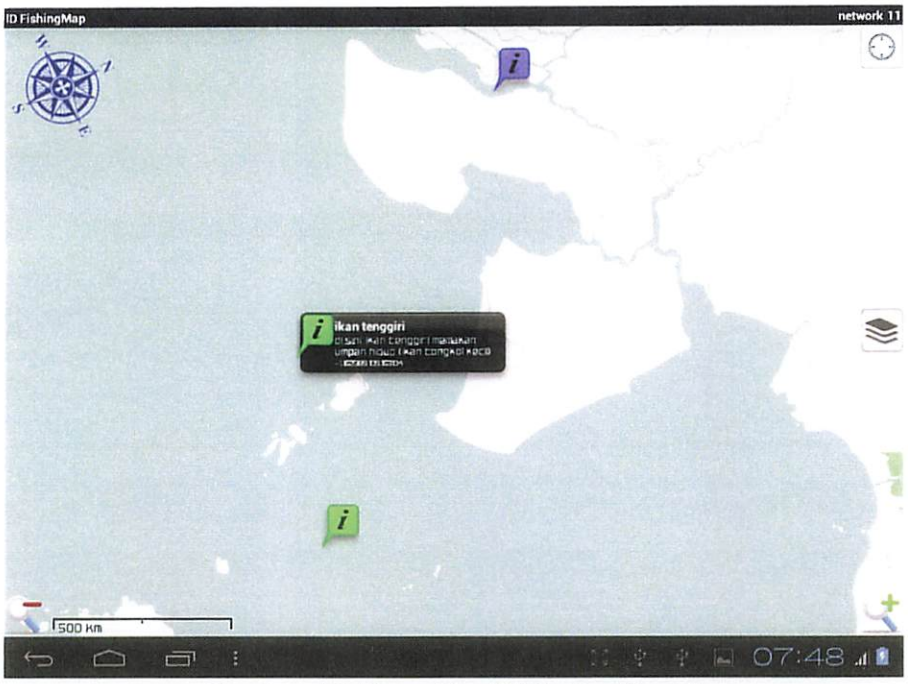
### 4.1.2 Tampilan Utama Aplikasi

Selanjutnya masuk ke tahap pada tampilan utama aplikasi *Fishing Map*, pada tahap ini diharapkan mempunyai koneksi internet. Untuk mendapatkan update map supaya bisa digunakan secara offline nantinya. Lihat gambar 4.2


Gambar 4.2 Main Map aplikasi

### 4.1.3 Tampilan Titik Kordinat Ikan

Setelah main aplikasi maka akan dilanjutkan ke tahap melihat beberapa titik yang sudah tertandai, dimana kordinat letak jenis ikan yang diketahui sering berkumpul. Untuk mengetahui lokasi saat ini, GPS harus di aktifkan. Agar cepat mencari titik lokasi. Lihat gambar 4.3.




Gambar 4.3 Tampilan Titik Kordinat Ikan

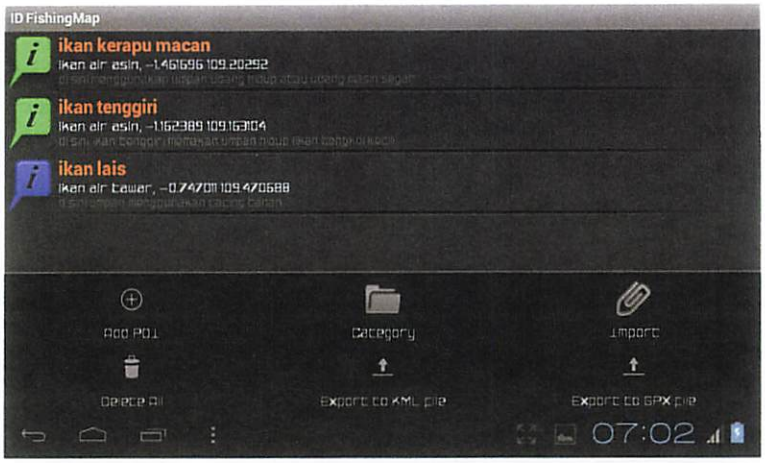
### 4.1.4 Tampilan *Option Menu Slide* Utama

Dilanjutkan ke tahap *Option Menu Slide*. *Option Menu Slide* pada aplikasi dibuat untuk menunjukan bahwa ada beberapa fitur di aplikasi ini yaitu, *Search, Compass, POI, My Location, Calendar, Settings, dan About*. Lihat gambar 4.4.


Gambar 4.4 Tampilan *Option Menu Slide* Aplikasi

### 4.1.5 Tampilan *Option Menu Slide* POI

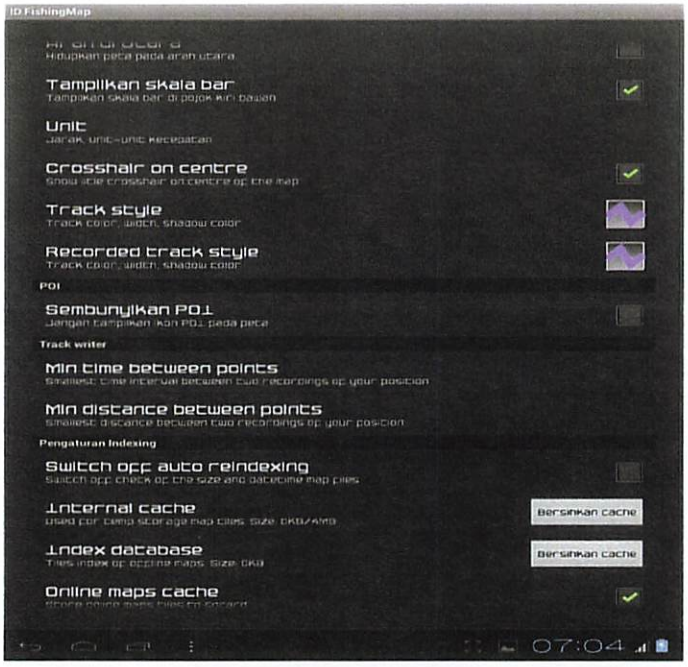
Tombol POI berfungsi untuk melihat beberapa list tempat ikan yg di tandai, menambah kategori jenis ikan, menghapus, sunting, *export* dan *Import* dalam ekstensi .kml atau .gpx. Lihat gambar 4.5.




Gambar 4.5 Tampilan *Option Menu Slide* pada POI

### 4.1.6 Tampilan *Settings* Aplikasi

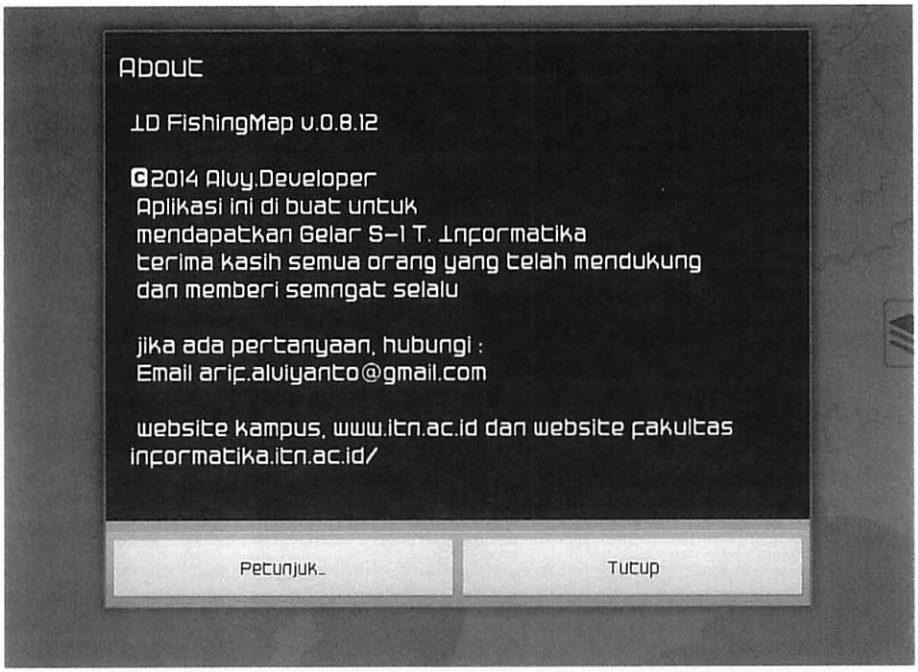
Tombol *Settings* berfungsi untuk mengatur tampilan pada peta, posisi peta, membersihkan *Cache,* mengatur *Compass,* posisi-posisi seperti sekala bar dan tempat penyimpanan *memory.* Lihat gambar 4.6.




Gambar 4.6 Tampilan Menu *Settings* Aplikasi

### 4.1.7 Tampilan Menu *About*

Sedangkan Tombol *About* berfungsi untuk melihat nama pengembang atau *developer* aplikasi dan memberi petunjuk penggunaan aplikasi. Lihat gambar 4.7




Gambar 4.7 Tampilan Menu *About*

## 4.2 Pengujian

Tahap pengujian adalah satu proses untuk menguji aplikasi yang telah selesai dibuat. Hal ini bertujuan untuk menemukan kesalahan dan kemudian memperbaikinya. Berikut merupakan bentuk pengujian yang dilakukan.

### 4.2.1 Pengujian Aplikasi Pada *Device* Android

ditunjukan Pada tahapan ini pengujian aplikasi dilakukan pada beberapa telepon genggam dengan berbagai device, versi system operasi dan ukuran *pixel* layar. Dimana pengujian ini dilakukan untuk mengetahui tingkat keberhasilan dalam pembuatan aplikasi *mobile*. Berikut adalah tabel hasil pengujian dari rancang bangun aplikasi mobile fishing map berbasis android pada beberapa *device* android yang berbeda pada tabel 4.1 sebagai berikut :

Tabel 4.1 Tabel Hasil pengujian versi (Operating System)

| Tipe Device | Sistem Operasi | Hasil |
|---|---|---|
| Samsung Galaxy Y S5360 | Gingerbread | √ |
| Sony Experia J | Ice Cream Sandwich | √ |
| SpeedUp pad 8 | Ice Cream Sandwich | √ |
| Samsung galaxy note 2 | JellyBean | √ |

*Keterangan :*

√ *: berhasil*

× *: gagal*

Kesimpulan dari tabel 4.1, aplikasi mobile *Fishing Map* berbasis android dapat berjalan dengan baik dan lancar dari beberapa device android yang memiliki system operasi yang berbeda.

Tabel 4.2 Tabel Hasil pengujian versi *Potrait*

| Tipe Device | Ukuran Pixel | Hasil |
|---|---|---|
| Samsung Galaxy Y S5360 | 240 x 320 | √ |
| Sony Experia J | 480 x 854 | √ |
| SpeedUp pad 8 | 768 x 1024 | √ |
| Samsung galaxy note 2 | 720 x 1280 | √ |

*Keterangan*

√ *: Sesuai*

× *: Tidak Sesuai*

Kesimpulan dari tabel 4.2, aplikasi mobile *Fishing Map* berbasis android dapat berjalan dengan baik dan lancar dari beberapa *device* android pada tabel 4.2 dalam posisi *Potrait.*

Tabel 4.3 Tabel Hasil pengujian versi *Landscape*

| Tipe Device | Ukuran Pixel | Hasil |
|---|---|---|
| Samsung Galaxy Y S5360 | 320 x 240 | √ |
| Sony Experia J | 854 x 480 | √ |
| SpeedUp pad 8 | 1024 x 768 | √ |
| Samsung galaxy note 2 | 1280 x 720 | √ |

*Keterangan*

√ *: Sesuai*

× *: Tidak Sesuai*

Kesimpulan dari tabel 4.3, aplikasi mobile *Fishing Map* berbasis android dapat berjalan dengan baik dan lancar dari beberapa *device* android pada tabel 4.3 dalam posisi *Landscape*

### 4.2.2 Pengujian Fungsional Sistem

Pada bagian pengujian fungsional rancang bangun aplikasi *mobile fishing map* berbasis android yang dilakukan adalah, pengujian terhadap fungsi dari tombol, proses dan tampilan pada aplikasi *Mobile Fishing Map* berbasis android yang berupa menampilkan menu-menu aplikasi yang ada.

Berikut adalah data-data yang diperoleh dari hasil pengujian sistem yang dilakukan secara satu persatu pada beberapa device. yang dapat ditunjukan pada tabel 4 sebagai berikut

Tabel 4.4. Pengujian Fungsional Sistem

| Proses | Hasil |
|---|---|
| Proses tampilan awal | √ |
| Tombol dan tampilan menu Option Slide | √ |
| Tombol dan proses Search | √ |
| Tombol dan proses Compass | √ |
| Tombol dan tampilan menu POI | √ |
| Tombol dan proses My location | √ |
| Tombol dan tampilan menu Calendar | X |
| Tombol dan tampilan menu Settings | √ |
| Tombol dan tampilan menu About | √ |
| Tombol dan tampilan menu Petunjuk | √ |

*Keterangan*

√ *: Berhasil*

× *: Gagal*

Pada sistem aplikasi *mobile Fishing Map* berbasis android proses dan menampilkan menu aplikasi yang ada, dapat berjalan dengan baik dari segi fungsi dan proses sistem. Terkecuali 1 yaitu pada tombol dan tampilan menu *Calendar.*

### 4.2.3 Pengujian *User*

Pengujian aplikasi pada pengguna bertujuan untuk memperoleh hasil responden dimana pengguna bisa memberikan penilaian saat aplikasi dicoba. Berikut adalah rekapitulasi hasil proses pengujian yang terdiri dari 10 orang meliputi :

a. Penilaian terhadap tampilan aplikasi
b. Seberapa mudah pengguna menjalankan aplikasi
c. Keakuratan kordinat dan informasi yg dihasilkan dari aplikasi

Tabel hasil pengujian aplikasi yang diambil dari beberapa pengguna seperti pada tabel 4.5 dibawah ini.

Tabel 4.5. Tabel Hasil Pengujian Aplikasi

| No. | Uraian | Jumlah Penilaian Responden | | | |
|---|---|---|---|---|---|
| | | SB | B | C | K |
| 1 | Tampilan aplikasi mobile *Fishing Map* | 80 % | 10 % | 10 % | - |
| 2 | Seberapa mudah pengguna menjalankan aplikasi | 90 % | 10 % | - | - |
| 3 | Seberapa akurat kordinat dan informasi yang dihasilkan dari aplikasi *Fishing Map* | 70 % | 10 % | 20 % | - |
| | | | | | |

*Keterangan :*

*SB* : *Sangat Baik*

*B* : *Baik*

*C* : *Cukup*

*K* : *Kurang*

# BAB V
# PENUTUP

## 5.1 Kesimpulan

Dari pembuatan aplikasi *Fishing Map* berbasis android ini, maka kesimpulan yang dapat diambil adalah :

1. Hasil pengujian tingkat kemudahan penggunaan aplikasi diperoleh nilai persentase sebesar 90% dari kuisioner yang diberikan kepada responden.
2. Aplikasi *Mobile Fishing Map* berbasis android ini bisa digunakan mulai dari versi GinggerBread , (ICS) Ice Cream Sandwich, dan JellyBean.
3. Semua fungsi dari sistem dan tampilan sistem dapat berjalan dengan baik pada ukuran pixel 1024x768.

Dan dari data kuisoner pada tabel 4.5 dapat di simpulkan bahwa :

1. 80,00 % Responden mengatakan bahwa tampilan aplikasi **Sangat Baik,** 10,00 % mengatakan **Baik,** dan 10,00 % mengatakan **Cukup,** 0 % mengatakan **Kurang.**
2. 90,00 % Responden mengatakan bahwa kemudahan menjalankan aplikasi **Sangat Mudah,** 10,00 % mengatakan **mudah,** 0 % mengatakan **Kurang.**
3. 70,00 % Responden mengatakan bahwa ke akuratan kordinat dan informasi yang dihasilkan aplikasi **Sangat Akurat,** 10,00 % mengatakan **Akurat,** 20,00 % mengatakan **Cukup,** 0 % mengatakan **Kurang.**

## 5.2 Saran

Dari beberapa kesimpulan yang telah diambil, maka dapat dikemukakan saran-saran yang akan sangat membantu untuk pengembangan perangkat lunak ini selanjutnya.

1. Dalam pengembangan sistem berikutnya supaya dapat melakukan penambahan lebih banyak informasi yang lebih lengkap lagi.
2. Agar di beri petunjuk arah jalan, supaya pengguna *user* lebih mudah dalam pencarian lokasi dan menuju lokasi yang di tuju tanpa harus melihat di list POI.
3. Menu *Calendar* diharapkan pada pengembangan agar di perbaiki lagi dan diberi informasi waktu yang tepat untuk memancing.
4. Ditambahkan sisi admin, agar kordinat dan informasi yang di ditambahkan pengguna / *user* aplikasi dapat di filter untuk mencari titik kordinat yang terbaik dan dapat di perlengkap lagi informasi yang ingin diketahui.
5. Ditambahkan sisi group kusus untuk pengembang, agar mempermudah berbagi data informasi secara detail dan update peta yang baru.

## DAFTAR PUSTAKA


[1] Anonymous. Sqlite Tutorial. http://zetcode.com/databases/sqlitetutorial/. 2007-2013.

[2] Arif Akbarul Huda. 2012. *24 Jam!! Pintar Pemmrograman Android.* Yogyakarta: Andi.

[3] Nazruddin Safaat H. 2013. *Android pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android.* Bandung: Informatika.

[4] Ostrander, jason. 2012.*Android UI Fundamentals : Develop And Design.* E-Book

[5] Reto Meier. Wrox Press. *Professional Android Application Development.* http://www.wrox.com/ diakses 29 September 2013

[6] Supriyanto, Dodit & Agustina, Rini, S.Kom, M.Pd. *Pemograman Aplikasi Android.*

[7] Wahana Komputer. 2013. *Android Programming with Eclipse.* Yogyakarta: Andi.

[8] Zigurd Mednieks. 2012. *Proramming Android 2nd Edition : Java Programming For The New Generation Of Mobile devices.* E-Book

# LAMPIRAN

**INSTITUT TEKNOLOGI NASIONAL MALANG**
**Fakultas Teknologi Nasional Malang**
**Program Studi Teknik Informatika S1**

# FORMULIR BIMBINGAN SKRIPSI

Nama : Arif Alfiyanto

NIM : 0818086

Masa Bimbingan : 12 November 2013 s/d 12 Mei 2014

Judul Skripsi : Rancang Bangun Aplikasi Fishing Map Berbasis Android

| No. | TANGGAL | URAIAN | PARAF PEMBIMBING |
|---|---|---|---|
| 1 | 02-08-2013 | Bimbingan laporan Bab I – III + Program | |
| 2 | 07-08-2013 | Bimbingan laporan Bab I – III | |
| 3 | 20-01-2014 | Bimbingan Makalah + Program | |
| 4 | 10-02-2014 | Bimbingan Makalah Seminar + Laporan I – V | |
| 5 | 13-02-2014 | Bimbingan Keseluruhan Laporan | |

Malang, Februari 2014
Dosen Pembimbing

Dr. Ir. Dhayal Gustopo, MT.
NIP.Y. 1039400264

**INSTITUT TEKNOLOGI NASIONAL MALANG**
**Fakultas Teknologi Nasional Malang**
**Program Studi Teknik Informatika S1**

# FORMULIR BIMBINGAN SKRIPSI

Nama : Arif Alfiyanto

NIM : 0818086

Masa Bimbingan : 12 November 2013 s/d 12 Mei 2014

Judul Skripsi : Rancang Bangun Aplikasi Fishing Map Berbasis Android

| No. | TANGGAL | URAIAN | PARAF PEMBIMBING |
|---|---|---|---|
| 1 | 07-01-2014 | Bimbingan laporan Bab I – III | |
| 2 | 10-01-2014 | Bimbingan Program | |
| 3 | 17-02-2014 | Bimbingan laporan Bab I – V | |
| 4 | 14-01-2014 | Bimbingan Makalah Seminar | |
| 5 | 18-02-2014 | Bimbingan Keseluruhan Laporan | |

Malang, Februari 2014
Dosen Pembimbing

Suryo Adi Wibowo, ST., MT.
NIP.P 103000438

**INSTITUT TEKNOLOGI NASIONAL MALANG**
**FAKULTAS TEKNOLOGI INDUSTRI**
**PROGRAM STUDI TEKNIK INFORMATIKA S-1**
**Jl. Karanglo, KM 2 Malang**

# BERITA ACARA UJIAN SKRIPSI
# FAKULTAS TEKNOLOGI INDUSTRI

NAMA : Arif Alfiyanto
NIM : 0818086
JURUSAN : Teknik Informatika S-1
JUDUL : RANCANG BANGUN APLIKASI FISHING MAP BERBASIS ANDROID

Dipertahankan dihadapan Majelis Penguji Skripsi Jenjang Strata Satu (S-1) pada :
Hari : Kamis
Tanggal : 20 Februari 2014
Nilai : 81.85

Panitia Ujian Skripsi :

**Ketua Majelis Penguji**

**Joseph Dedy Irawan, ST, MT**
**NIP. 19740416 2005011002**

Anggota Penguji :

**Dosen Penguji I**

**Sonny Prasetio, ST., MT**
**NIP.P 1031000433**

**Dosen Penguji II**

**Joseph Dedy Irawan, ST., MT**
**NIP. 197404162005011002**

**INSTITUT TEKNOLOGI NASIONAL MALANG**
**FAKULTAS TEKNOLOGI INDUSTRI**
**PROGRAM STUDI TEKNIK INFORMATIKA S-1**
**Jl. Karanglo, KM 2 Malang**

# FORMULIR PERBAIKAN SKRIPSI

Dalam pelaksanaan ujian skripsi jenjang Strata 1 Program Studi Teknik Informatika, maka perlu adanya perbaikan skripsi untuk mahasiswa :

NAMA : ARIF ALFIYANTO
NIM : 0818086
JURUSAN : Teknik Informatika S-1
JUDUL : RANCANG BANGUN APLIKASI FISHING MAP BERBASIS ANDROID

| No | Penguji | Tanggal | Uraian | Paraf |
|---|---|---|---|---|
| 1. | Penguji I | 20 Februari 2014 | 1. Abstrak diperbaiki<br>2. Perbaiki Penulisan makalah skripsi | |
| 2. | Penguji II | 20 Februari 2014 | 1. Tambahkan pada saran untuk pembuatan Group<br>2. Informasi tentang Fishing Map muncul jika program dibuka | |

**Dosen Penguji I**

**Sonny Prasetio, ST., MT**
**NIP.P 1031000433**

**Dosen Penguji II**

**Joseph Dedy Irawan, ST., MT**
**NIP. 197404162005011002**

**Dosen Pembimbing I**

**Dr. Ir. Dhayal Gustopo, MT**
**NIP.Y 1039400264**

**Dosen Pembimbing II**

**Suryo Adi Wibowo, ST., MT**
**NIP.P 103000438**

# LEMBAR KUESIONER

Nama : Sandy

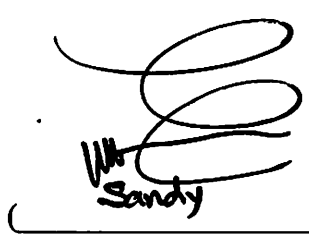
Pekerjaan : Swasta

1. Menurut anda setelah melihat Aplikasi *Fishing Map* seberapa bagus kah Tampilan aplikasi tersebut.?
   a. Sangat Bagus [✓] c. Cukup [...]
   b. Bagus [...] d. Kurang [...]

2. Dan setelah anda menggunakan aplikasi tersebut, apakah anda merasa sulit untuk menggunakannya atau menjalankannya.?
   a. Sangat Mudah [✓] c. Cukup [...]
   b. Mudah [...] d. Kurang [...]

3. Menurut anda seberapa akurat informasi dari beberapa titik kordinat pada aplikasi *Fishing Map*?
   a. Sangat Akurat [✓] c. Cukup [...]
   b. Akurat [...] d. Kurang [...]

ttd




( )

# LEMBAR KUESIONER

Nama : Alwi

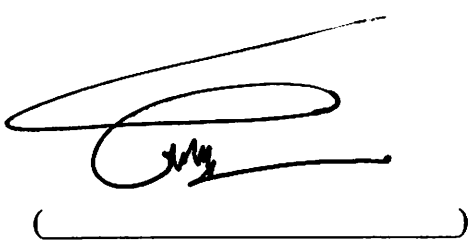
Pekerjaan : Swasta

1. Menurut anda setelah melihat Aplikasi *Fishing Map* seberapa bagus kah Tampilan aplikasi tersebut.?
   a. Sangat Bagus [✓] c. Cukup [...]
   b. Bagus [...] d. Kurang [...]

2. Dan setelah anda menggunakan aplikasi tersebut, apakah anda merasa sulit untuk menggunakannya atau menjalankannya.?
   a. Sangat Mudah [✓] c. Cukup [...]
   b. Mudah [...] d. Kurang [...]

3. Menurut anda seberapa akurat informasi dari beberapa titik kordinat pada aplikasi *Fishing Map*?
   a. Sangat Akurat [✓] c. Cukup [...]
   b. Akurat [...] d. Kurang [...]

ttd



(_____)

# LEMBAR KUESIONER

Nama : Mislan

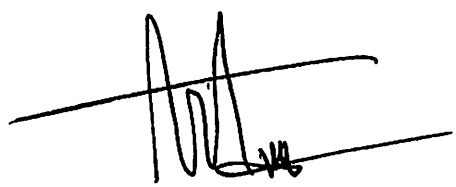
Pekerjaan : PNS

1. Menurut anda setelah melihat Aplikasi *Fishing Map* seberapa bagus kah Tampilan aplikasi tersebut.?
   a. Sangat Bagus [√] c. Cukup [...]
   b. Bagus [...] d. Kurang [...]

2. Dan setelah anda menggunakan aplikasi tersebut, apakah anda merasa sulit untuk menggunakannya atau menjalankannya.?
   a. Sangat Mudah [√] c. Cukup [...]
   b. Mudah [...] d. Kurang [...]

3. Menurut anda seberapa akurat informasi dari beberapa titik kordinat pada aplikasi *Fishing Map*?
   a. Sangat Akurat [...] c. Cukup [...]
   b. Akurat [√] d. Kurang [...]

ttd



(_____)

# LEMBAR KUESIONER

Nama : Dodik purwanto

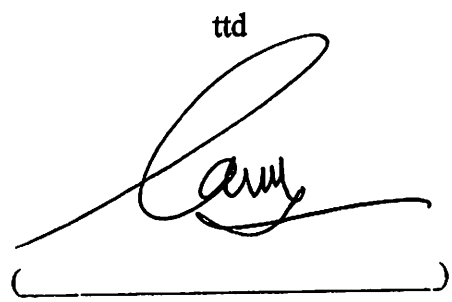
Pekerjaan : PHS

1. Menurut anda setelah melihat Aplikasi *Fishing Map* seberapa bagus kah Tampilan aplikasi tersebut.?
   a. Sangat Bagus [✓] c. Cukup [...]
   b. Bagus [...] d. Kurang [...]

2. Dan setelah anda menggunakan aplikasi tersebut, apakah anda merasa sulit untuk mengguuakannya atau menjalankannya.?
   a. Sangat Mudah [✓] c. Cukup [...]
   b. Mudah [...] d. Kurang [...]

3. Menurut anda seberapa akurat informasi dari beberapa titik kordinat pada aplikasi *Fishing Map*?
   a. Sangat Akurat [✓] c. Cukup [...]
   b. Akurat [...] d. Kurang [...]

ttd



(_____)

# LEMBAR KUESIONER

Nama : Joko

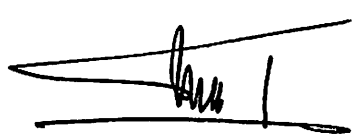
Pekerjaan : Swasta

1. Menurut anda setelah melihat Aplikasi *Fishing Map* seberapa bagus kah Tampilan aplikasi tersebut.?
   a. Sangat Bagus [√] c. Cukup [...]
   b. Bagus [...] d. Kurang [...]

2. Dan setelah anda menggunakan aplikasi tersebut, apakah anda merasa sulit untuk menggunakannya atau menjalankannya.?
   a. Sangat Mudah [√] c. Cukup [...]
   b. Mudah [...] d. Kurang [...]

3. Menurut anda seberapa akurat informasi dari beberapa titik kordinat pada aplikasi *Fishing Map*?
   a. Sangat Akurat [√] c. Cukup [...]
   b. Akurat [...] d. Kurang [...]

ttd



(_____)

# LEMBAR KUESIONER

Nama : Dona Herniawan

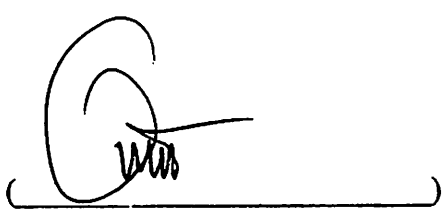
Pekerjaan : Swasta

1. Menurut anda setelah melihat Aplikasi *Fishing Map* seberapa bagus kah Tampilan aplikasi tersebut.?
   a. Sangat Bagus [✓] c. Cukup [...]
   b. Bagus [...] d. Kurang [...]

2. Dan setelah anda menggunakan aplikasi tersebut, apakah anda merasa sulit untuk menggunakannya atau menjalankannya.?
   a. Sangat Mudah [✓] c. Cukup [...]
   b. Mudah [...] d. Kurang [...]

3. Menurut anda seberapa akurat informasi dari beberapa titik kordinat pada aplikasi *Fishing Map*?
   a. Sangat Akurat [✓] c. Cukup [...]
   b. Akurat [...] d. Kurang [...]

ttd



# LEMBAR KUESIONER

Nama : Dimas Jiko

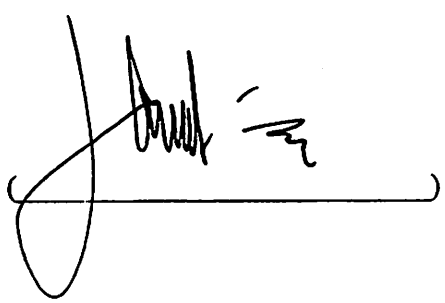
Pekerjaan : Swasta

1. Menurut anda setelah melihat Aplikasi *Fishing Map* seberapa bagus kah Tampilan aplikasi tersebut.?
   a. Sangat Bagus [✓] c. Cukup [...]
   b. Bagus [...] d. Kurang [...]

2. Dan setelah anda menggunakan aplikasi tersebut, apakah anda merasa sulit untuk menggunakannya atau menjalankannya.?
   a. Sangat Mudah [✓] c. Cukup [...]
   b. Mudah [...] d. Kurang [...]

3. Menurut anda seberapa akurat informasi dari beberapa titik kordinat pada aplikasi *Fishing Map*?
   a. Sangat Akurat [✓] c. Cukup [...]
   b. Akurat [...] d. Kurang [...]

ttd



# LEMBAR KUESIONER

Nama : Hardiyanto

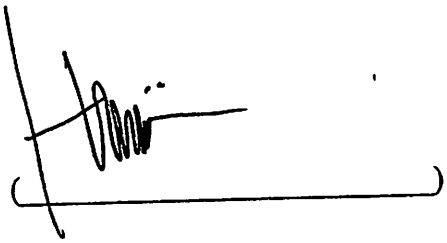
Pekerjaan : Swasta

1. Menurut anda setelah melihat Aplikasi *Fishing Map* seberapa bagus kah Tampilan aplikasi tersebut.?
   a. Sangat Bagus [✓] c. Cukup [...]
   b. Bagus [...] d. Kurang [...]

2. Dan setelah anda menggunakan aplikasi tersebut, apakah anda merasa sulit untuk menggunakannya atau menjalankannya.?
   a. Sangat Mudah [...] c. Cukup [...]
   b. Mudah [✓] d. Kurang [...]

3. Menurut anda seberapa akurat informasi dari beberapa titik kordinat pada aplikasi *Fishing Map*?
   a. Sangat Akurat [✓] c. Cukup [...]
   b. Akurat [...] d. Kurang [...]

ttd



# LEMBAR KUESIONER

Nama : Fadli

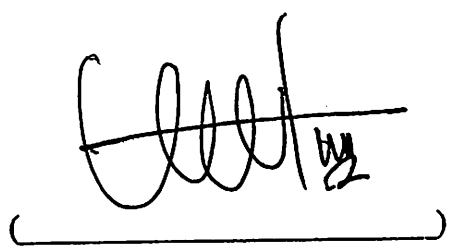
Pekerjaan : swasta

1. Menurut anda setelah melihat Aplikasi *Fishing Map* seberapa bagus kah Tampilan aplikasi tersebut.?
   a. Sangat Bagus [...] c. Cukup [√]
   b. Bagus [...] d. Kurang [...]

2. Dan setelah anda menggunakan aplikasi tersebut, apakah anda merasa sulit untuk menggunakannya atau menjalankannya.?
   a. Sangat Mudah [√] c. Cukup [...]
   b. Mudah [...] d. Kurang [...]

3. Menurut anda seberapa akurat informasi dari beberapa titik kordinat pada aplikasi *Fishing Map*?
   a. Sangat Akurat [...] c. Cukup [√]
   b. Akurat [...] d. Kurang [...]

ttd



# LEMBAR KUESIONER

Nama : Yusuf Sofian

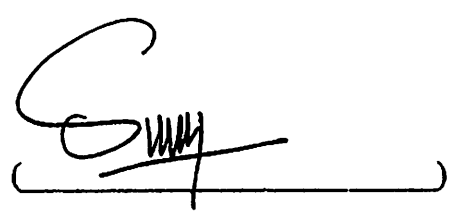
Pekerjaan : Mahasiswa

1. Menurut anda setelah melihat Aplikasi *Fishing Map* seberapa bagus kah Tampilan aplikasi tersebut.?
   a. Sangat Bagus [...] c. Cukup [...]
   b. Bagus [✓] d. Kurang [...]

2. Dan setelah anda menggunakan aplikasi tersebut, apakah anda merasa sulit untuk menggunakannya atau menjalankannya.?
   a. Sangat Mudah [✓] c. Cukup [...]
   b. Mudah [...] d. Kurang [...]

3. Menurut anda seberapa akurat informasi dari beberapa titik kordinat pada aplikasi *Fishing Map*?
   a. Sangat Akurat [...] c. Cukup [✓]
   b. Akurat [...] d. Kurang [...]

ttd



## Lampiran Script Program

### 1. Script Main Activity

```
package com.robert.maps;

import java.io.BufferedInputStream;
import java.io.BufferedOutputStream;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.InputStream;
import java.io.OutputStream;
import java.net.URL;
import java.net.URLEncoder;
import java.util.ArrayList;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;

import org.andnav.osm.util.GeoPoint;
import org.andnav.osm.util.TypeConverter;
import org.andnav.osm.views.util.StreamUtils;
import org.andnav.osm.views.util.constants.OpenStreetMapViewConstants;
import org.json.JSONArray;
import org.json.JSONObject;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.Dialog;
import android.app.SearchManager;
import android.content.ActivityNotFoundException;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.database.Cursor;
import android.database.sqlite.SQLiteException;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.net.Uri;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.Message;
import android.os.PowerManager;
import android.preference.PreferenceManager;
import android.provider.Browser;
import android.provider.SearchRecentSuggestions;
import android.util.DisplayMetrics;
import android.view.ContextMenu;
import android.view.ContextMenu.ContextMenuInfo;
```

```
import android.view.LayoutInflater;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.view.ViewGroup.LayoutParams;
import android.view.Window;
import android.view.WindowManager;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.Toast;

import com.google.android.apps.analytics.GoogleAnalyticsTracker;
import com.robert.maps.downloader.AreaSelectorActivity;
import com.robert.maps.kml.PoiActivity;
import com.robert.maps.kml.PoiListActivity;
import com.robert.maps.kml.PoiManager;
import com.robert.maps.kml.PoiPoint;
import com.robert.maps.kml.Track;
import com.robert.maps.kml.TrackListActivity;
import com.robert.maps.kml.XMLparser.PredefMapsParser;
import com.robert.maps.overlays.CurrentTrackOverlay;
import com.robert.maps.overlays.MyLocationOverlay;
import com.robert.maps.overlays.PoiOverlay;
import com.robert.maps.overlays.SearchResultOverlay;
import com.robert.maps.overlays.TrackOverlay;
import com.robert.maps.overlays.YandexTrafficOverlay;
import com.robert.maps.tileprovider.TileSource;
import com.robert.maps.tileprovider.TileSourceBase;
import com.robert.maps.utils.CompassView;
import com.robert.maps.utils.CrashReportHandler;
import com.robert.maps.utils.RException;
import com.robert.maps.utils.SearchSuggestionsProvider;
import com.robert.maps.utils.Ut;
import com.robert.maps.view.IMoveListener;
import com.robert.maps.view.MapView;
import com.robert.maps.view.TileView;
import com.robert.maps.view.TileViewOverlay;

public class MainActivity extends Activity {
        private static final String MAPNAME = "MapName";
        private static final String ACTION_SHOW_POINTS =
"com.robert.maps.action.SHOW_POINTS";

        private MapView mMap;
        private ImageView ivAutoFollow;
        private CompassView mCompassView;

        private TileSource mTileSource;
        private PoiManager mPoiManager;
        private Handler mCallbackHandler = new MainActivityCallbackHandler();
        private MoveListener mMoveListener = new MoveListener();
        private SensorManager mOrientationSensorManager;
        private PowerManager.WakeLock myWakeLock;
```

```
        // Overlays
        private YandexTrafficOverlay mYandexTrafficOverlay = null;
        private boolean mShowOverlay = false;
        private String mOverlayId = "";
        private MyLocationOverlay mMyLocationOverlay;
        private PoiOverlay mPoiOverlay;
        private TrackOverlay mTrackOverlay;
        private CurrentTrackOverlay mCurrentTrackOverlay;
        private SearchResultOverlay mSearchResultOverlay;

        private int mMarkerIndex;
        private boolean mAutoFollow = true;
        private String mGpsStatusName = "";
        private int mGpsStatusSatCnt = 0;
        private int mGpsStatusState = 0;
        private float mLastSpeed, mLastBearing;
        private boolean mCompassEnabled;
        private boolean mDrivingDirectionUp;
        private boolean mNorthDirectionUp;
        private int mPrefOverlayButtonBehavior;
        private int mPrefOverlayButtonVisibility;

        private GoogleAnalyticsTracker mTracker;
        private ImageView mOverlayView;

        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);

    if(!OpenStreetMapViewConstants.DEBUGMODE)
        CrashReportHandler.attach(this);

                mTracker = GoogleAnalyticsTracker.getInstance();
                mTracker.startNewSession("UA-10715419-3", 20, this);

                requestWindowFeature(Window.FEATURE_NO_TITLE);
                CreateContentView();

                mPoiManager = new PoiManager(this);
                mLocationListener = new SampleLocationListener();
                mMap.setMoveListener(mMoveListener);
                if(!OpenStreetMapViewConstants.DEBUGMODE) //
                        mOrientationSensorManager =
(SensorManager)getSystemService(Context.SENSOR_SERVICE);

                final SharedPreferences pref =
PreferenceManager.getDefaultSharedPreferences(this);
                SharedPreferences uiState = getPreferences(Activity.MODE_PRIVATE);

                // Init
                mPrefOverlayButtonBehavior =
Integer.parseInt(pref.getString("pref_overlay_button_behavior", "0"));
                mPrefOverlayButtonVisibility =
Integer.parseInt(pref.getString("pref_overlay_button_visibility", "0"));
                if(mPrefOverlayButtonVisibility == 1) // Always hide
                        mOverlayView.setVisibility(View.GONE);
                mCompassEnabled = uiState.getBoolean("CompassEnabled", false);
```

```
                mCompassView.setVisibility(mCompassEnabled ? View.VISIBLE :
View.INVISIBLE);
                mAutoFollow = uiState.getBoolean("AutoFollow", true);

                mMap.getController().setCenter(new GeoPoint(uiState.getInt("Latitude",
0), uiState.getInt("Longitude", 0)));
                mGPSFastUpdate = pref.getBoolean("pref_gpsfastupdate", true);
                mAutoFollow = uiState.getBoolean("AutoFollow", true);
                setAutoFollow(mAutoFollow, true);

                this.mTrackOverlay = new TrackOverlay(this, mPoiManager,
mCallbackHandler);
                this.mCurrentTrackOverlay = new CurrentTrackOverlay(this,
mPoiManager);
                this.mPoiOverlay = new PoiOverlay(this, mPoiManager, null,
pref.getBoolean("pref_hidepoi", false));
                mPoiOverlay.setTapIndex(uiState.getInt("curShowPoiId", -1));
        this.mMyLocationOverlay = new MyLocationOverlay(this);
        this.mSearchResultOverlay = new SearchResultOverlay(this);
        mSearchResultOverlay.fromPref(uiState);
        FillOverlays();

                mDrivingDirectionUp = pref.getBoolean("pref_drivingdirectionup", true);
                mNorthDirectionUp = pref.getBoolean("pref_northdirectionup", true);

                final int screenOrientation =
Integer.parseInt(pref.getString("pref_screen_orientation", "-1"));
                setRequestedOrientation(screenOrientation);

            final boolean fullScreen = pref.getBoolean("pref_showstatusbar", true);
                if (fullScreen)

            getWindow().setFlags(WindowManager.LayoutParams.FLAG_FORCE_NOT_FU
LLSCREEN,

            WindowManager.LayoutParams.FLAG_FORCE_NOT_FULLSCREEN);
                else
                        getWindow()

            .setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);


                if(uiState.getString("error", "").length() > 0){
                        showDialog(R.id.error);
                }


                if (!uiState.getString("app_version",
"").equalsIgnoreCase(Ut.getAppVersion(this))) {
                        DisplayMetrics metrics = new DisplayMetrics();
                        getWindowManager().getDefaultDisplay().getMetrics(metrics);

                        mTracker.setCustomVar(1, "Build", Ut.getAppVersion(this), 1);
                        mTracker.setCustomVar(2, "Ver", "Free", 1);
                        mTracker.setCustomVar(3, "DisplaySize",
""+Math.min(metrics.widthPixels,
metrics.heightPixels)+"x"+Math.max(metrics.widthPixels, metrics.heightPixels), 1);
```

```
                        mTracker.setCustomVar(4, "DisplayDensity",
""+(int)(160*metrics.density), 1);
                        mTracker.setCustomVar(5, "APILevel", Build.VERSION.SDK, 1);
                        mTracker.trackPageView("/InstallApp");

                        showDialog(R.id.whatsnew);
                }

                final Intent queryIntent = getIntent();
                final String queryAction = queryIntent.getAction();

                if (Intent.ACTION_SEARCH.equals(queryAction)) {
                        doSearchQuery(queryIntent);
                } else if (ACTION_SHOW_POINTS.equalsIgnoreCase(queryAction)) {
                        ActionShowPoints(queryIntent);
                } else if (Intent.ACTION_VIEW.equalsIgnoreCase(queryAction)) {
                        Uri uri = queryIntent.getData();
                        if(uri.getScheme().equalsIgnoreCase("geo")) {
                                final String latlon =
uri.getEncodedSchemeSpecificPart().replace("?"+uri.getEncodedQuery(), "");
                                if(latlon.equals("0,0")) {
                                        final String query =
uri.getEncodedQuery().replace("q=", "");
                                        queryIntent.putExtra(SearchManager.QUERY,
query);
                                        doSearchQuery(queryIntent);

                                } else {
                                        GeoPoint point =
GeoPoint.fromDoubleString(latlon);
                                        mPoiOverlay.setGpsStatusGeoPoint(point,
"GEO", "");
                                        setAutoFollow(false);
                                        mMap.getController().setCenter(point);
                                }
                        }
                }
        }

        @Override
        protected void onNewIntent(Intent intent) {
                super.onNewIntent(intent);

    final String queryAction = intent.getAction();
    if (Intent.ACTION_SEARCH.equals(queryAction)) {
       doSearchQuery(intent);
    } else if (ACTION_SHOW_POINTS.equalsIgnoreCase(queryAction))
                        ActionShowPoints(intent);

        }

        private void doSearchQuery(Intent queryIntent) {
                mSearchResultOverlay.Clear();
                this.mMap.invalidate();

                final String queryString =
queryIntent.getStringExtra(SearchManager.QUERY);
```

```
    // Record the query string in the recent queries suggestions provider.
    SearchRecentSuggestions suggestions = new SearchRecentSuggestions(this,
SearchSuggestionsProvider.AUTHORITY, SearchSuggestionsProvider.MODE);
    suggestions.saveRecentQuery(queryString, null);

            InputStream in = null;
            OutputStream out = null;

            try {
                final SharedPreferences pref =
PreferenceManager.getDefaultSharedPreferences(this);
                URL url = new URL(

      "http://ajax.googleapis.com/ajax/services/search/local?v=1.0&sll="
                                                    +
this.mMap.getMapCenter().toDoubleString()
                                                    + "&q=" +
URLEncoder.encode(queryString, "UTF-8")
                                                    + "");
                Ut.dd(url.toString());
                in = new BufferedInputStream(url.openStream(),
StreamUtils.IO_BUFFER_SIZE);

                final ByteArrayOutputStream dataStream = new
ByteArrayOutputStream();
                out = new BufferedOutputStream(dataStream,
StreamUtils.IO_BUFFER_SIZE);
                StreamUtils.copy(in, out);
                out.flush();

                String str = dataStream.toString();
                JSONObject json = new JSONObject(str);
                //Ut.dd(json.toString(4)); //
                JSONArray results = (JSONArray) ((JSONObject)
json.get("responseData")).get("results");
                //Ut.dd("results.length="+results.length());
                if(results.length() == 0){
                    Toast.makeText(this, R.string.no_items,
Toast.LENGTH_SHORT).show();
                    return;
                }
                JSONObject res = results.getJSONObject(0);
                //Ut.dd(res.toString(4));
                //Toast.makeText(this, res.getString("titleNoFormatting"),
Toast.LENGTH_LONG).show();
                final String address = res.getString("addressLines").replace("\"",
"").replace("[", "").replace("]", "").replace(",", ", ").replace("  ", " ");
                //Toast.makeText(this, address, Toast.LENGTH_LONG).show();
                //Toast.makeText(this, ((JSONObject)
json.get("addressLines")).toString(), Toast.LENGTH_LONG).show();

                setAutoFollow(false, true);
                final GeoPoint point = new GeoPoint((int)(res.getDouble("lat")*
1E6), (int)(res.getDouble("lng")* 1E6));
                this.mSearchResultOverlay.setLocation(point, address);
                this.mMap.getController().setZoom((int) (2 *
res.getInt("accuracy")));
                mMap.getController().setCenter(point);
```

```
                        //this.mOsmv.getController().animateTo(new
GeoPoint((int)(res.getDouble("lat")* 1E6), (int)(res.getDouble("lng")* 1E6)),
OpenStreetMapViewController.AnimationType.MIDDLEPEAKSPEED,
OpenStreetMapViewController.ANIMATION_SMOOTHNESS_HIGH,
OpenStreetMapViewController.ANIMATION_DURATION_DEFAULT);

                        setTitle();

                } catch (Exception e) {
                        e.printStackTrace();
                        Toast.makeText(this, R.string.no_inet_conn,
Toast.LENGTH_LONG).show();
                } finally {
                        StreamUtils.closeStream(in);
                        StreamUtils.closeStream(out);
                }
        }

        private View CreateContentView() {
                setContentView(R.layout.main);

                final SharedPreferences pref =
PreferenceManager.getDefaultSharedPreferences(this);

                final RelativeLayout rl = (RelativeLayout) findViewById(R.id.map_area);
                final int sideBottom = Integer.parseInt(pref.getString("pref_zoomctrl",
"1"));
                final boolean showTitle = pref.getBoolean("pref_showtitle", true);

                if(!showTitle)
                        findViewById(R.id.screen).setVisibility(View.GONE);

                mMap = new MapView(this,
Integer.parseInt(pref.getString("pref_zoomctrl", "1")),
pref.getBoolean("pref_showscalebar", true) ? 1 : 0);
                mMap.setId(R.id.main);
                final RelativeLayout.LayoutParams pMap = new
RelativeLayout.LayoutParams(LayoutParams.FILL_PARENT,
LayoutParams.FILL_PARENT);
        rl.addView(mMap, pMap);

        mCompassView = new CompassView(this, sideBottom == 2 ? false : true);
        mCompassView.setVisibility(mCompassEnabled ? View.VISIBLE :
View.INVISIBLE);

        final RelativeLayout.LayoutParams compassParams = new
RelativeLayout.LayoutParams(RelativeLayout.LayoutParams.WRAP_CONTENT,
RelativeLayout.LayoutParams.WRAP_CONTENT);
        compassParams.addRule(RelativeLayout.ALIGN_PARENT_LEFT);
        compassParams.addRule(!(sideBottom == 2 ? false : true) ?
RelativeLayout.ALIGN_BOTTOM : RelativeLayout.ALIGN_TOP, R.id.main);
        rl.addView(mCompassView, compassParams);

        ivAutoFollow = new ImageView(this);
        ivAutoFollow.setImageResource(R.drawable.autofollow);
        ivAutoFollow.setVisibility(ImageView.INVISIBLE);

        final RelativeLayout.LayoutParams followParams = new
```

```
RelativeLayout.LayoutParams(RelativeLayout.LayoutParams.WRAP_CONTENT,
RelativeLayout.LayoutParams.WRAP_CONTENT);
    followParams.addRule(RelativeLayout.ALIGN_PARENT_RIGHT);
    if(!(sideBottom == 2 ? false : true))
        followParams.addRule(RelativeLayout.ALIGN_PARENT_BOTTOM);
    else
        followParams.addRule(RelativeLayout.ALIGN_TOP, R.id.main);
    rl.addView(ivAutoFollow, followParams);

    ivAutoFollow.setOnClickListener(new OnClickListener(){
                    public void onClick(View v) {
                            setAutoFollow(true);
                            mSearchResultOverlay.Clear();
                            setLastKnownLocation();
                    }
    });

    mOverlayView = new ImageView(this);
    mOverlayView.setImageResource(R.drawable.r_overlays);
    final RelativeLayout.LayoutParams layerParams = new
RelativeLayout.LayoutParams(RelativeLayout.LayoutParams.WRAP_CONTENT,
RelativeLayout.LayoutParams.WRAP_CONTENT);
    layerParams.addRule(RelativeLayout.CENTER_VERTICAL);
        layerParams.addRule(RelativeLayout.ALIGN_PARENT_RIGHT);
    rl.addView(mOverlayView, layerParams);

    mOverlayView.setOnClickListener(new View.OnClickListener() {

                    public void onClick(View v) {
                            if(mTileSource.YANDEX_TRAFFIC_ON == 1) {
                                    mShowOverlay = !mShowOverlay;
                                    FillOverlays();
                            } else {
                                    if(mPrefOverlayButtonBehavior == 1) {
                                            v.showContextMenu();
                                    } else if(mPrefOverlayButtonBehavior == 2) {
                                            setTileSource(mTileSource.ID,
mOverlayId, !mShowOverlay);
                                    } else if(mOverlayId.equalsIgnoreCase("") &&
mTileSource.MAP_TYPE != TileSourceBase.MIXMAP_PAIR) {
                                            v.showContextMenu();
                                    } else {
                                            setTileSource(mTileSource.ID,
mOverlayId, !mShowOverlay);
                                    }
                            }

                            mMap.postInvalidate();
                    }
            });
    mOverlayView.setOnLongClickListener(new View.OnLongClickListener() {

                    public boolean onLongClick(View v) {
                            if(mTileSource.YANDEX_TRAFFIC_ON != 1 &&
mPrefOverlayButtonBehavior == 0)
                                    v.showContextMenu();
                            return true;
                    }
```

```
            });
        mOverlayView.setOnCreateContextMenuListener(new
View.OnCreateContextMenuListener() {

                        public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenuInfo menuInfo) {

                                SharedPreferences pref =
PreferenceManager.getDefaultSharedPreferences(MainActivity.this);

                                File folder = Ut.getRMapsMapsDir(MainActivity.this);
                                if (folder.exists()) {
                                        File[] files = folder.listFiles();
                                        if (files != null)
                                                for (int i = 0; i < files.length; i++) {
                                                        if
(files[i].getName().toLowerCase().endsWith(".mnm")
                                                                        ||
files[i].getName().toLowerCase().endsWith(".tar")
                                                                        ||
files[i].getName().toLowerCase().endsWith(".sqlitedb")) {
                                                                String name =
Ut.FileName2ID(files[i].getName());
                                                                if
(pref.getBoolean("pref_usermaps_" + name + "_enabled", false)
                                                                                &&
(mTileSource.PROJECTION == 0 || mTileSource.PROJECTION ==
Integer.parseInt(pref.getString("pref_usermaps_" + name + "_projection", "1")))
                                                                                &&
pref.getBoolean("pref_usermaps_" + name + "_isoverlay", false)) {
                                                                        MenuItem item
= menu.add(R.id.isoverlay, Menu.NONE, Menu.NONE, pref.getString("pref_usermaps_"
+ name + "_name", files[i].getName()));

        item.setTitleCondensed("usermap_" + name);
                                                                }
                                                        }
                                                }
                                }

                                Cursor c =
mPoiManager.getGeoDatabase().getMixedMaps();
                                if(c != null) {
                                        if(c.moveToFirst()) {
                                                do {
                                                        if
(pref.getBoolean("PREF_MIXMAPS_" + c.getInt(0) + "_enabled", false) && c.getInt(2) ==
3) {
                                                                final JSONObject json =
MixedMapsPreference.getMapCustomParams(c.getString(3));

        if(mTileSource.PROJECTION == 0 || mTileSource.PROJECTION ==
json.optInt(MixedMapsPreference.MAPPROJECTION)) {
                                                                        MenuItem item
= menu.add(R.id.isoverlay, Menu.NONE, Menu.NONE, c.getString(1));

        item.setTitleCondensed("mixmap_" + c.getInt(0));
                                                                }
```

```
                                                }
                                        } while(c.moveToNext());
                                }
                                c.close();
                        }

                        final SAXParserFactory fac =
SAXParserFactory.newInstance();
                        SAXParser parser = null;
                        try {
                                parser = fac.newSAXParser();
                                if(parser != null){
                                        final InputStream in =
getResources().openRawResource(R.raw.predefmaps);
                                        parser.parse(in, new
PredefMapsParser(menu, pref, true, mTileSource.PROJECTION));
                                }
                        } catch (Exception e) {
                                e.printStackTrace();
                        }



                }
        });

    registerForContextMenu(mMap);

                return rl;
        }

        private void FillOverlays() {
                this.mMap.getOverlays().clear();

                if(mTileSource == null) {
                } else if(mTileSource.YANDEX_TRAFFIC_ON == 1 && mShowOverlay
&& mYandexTrafficOverlay == null) {
                        mYandexTrafficOverlay = new YandexTrafficOverlay(this,
mMap.getTileView());
                } else if((mTileSource.YANDEX_TRAFFIC_ON != 1 || !mShowOverlay)
&& mYandexTrafficOverlay != null) {
                        mYandexTrafficOverlay.Free();
                        mYandexTrafficOverlay = null;
                }

                if(mYandexTrafficOverlay != null)
                        this.mMap.getOverlays().add(mYandexTrafficOverlay);
    if(mTrackOverlay != null)
        this.mMap.getOverlays().add(mTrackOverlay);
    if(mCurrentTrackOverlay != null)
        this.mMap.getOverlays().add(mCurrentTrackOverlay);
    if(mPoiOverlay != null)
        this.mMap.getOverlays().add(mPoiOverlay);
    this.mMap.getOverlays().add(mMyLocationOverlay);
    this.mMap.getOverlays().add(mSearchResultOverlay);
        }

        private void setAutoFollow(boolean autoFollow) {
```

```
                setAutoFollow(autoFollow, false);
        }

        private void setAutoFollow(boolean autoFollow, final boolean supressToast) {
                mAutoFollow = autoFollow;

                if (autoFollow) {
                        ivAutoFollow.setVisibility(ImageView.INVISIBLE);
                        if(!supressToast)
                                Toast.makeText(this, R.string.auto_follow_enabled,
Toast.LENGTH_SHORT).show();
                } else {
                        ivAutoFollow.setVisibility(ImageView.VISIBLE);
                        if(!supressToast)
                                Toast.makeText(this, R.string.auto_follow_disabled,
Toast.LENGTH_SHORT).show();
                }
        }

        private void setLastKnownLocation() {
                final GeoPoint p = mMyLocationOverlay.getLastGeoPoint();
                if(p != null)
                        mMap.getController().setCenter(p);
                else {
                        final LocationManager lm = (LocationManager)
getSystemService(Context.LOCATION_SERVICE);
                        final Location loc1 =
lm.getLastKnownLocation(SampleLocationListener.GPS);
                        final Location loc2 =
lm.getLastKnownLocation(SampleLocationListener.NETWORK);

                        boolean boolGpsEnabled =
lm.isProviderEnabled(SampleLocationListener.GPS);
                        boolean boolNetworkEnabled =
lm.isProviderEnabled(SampleLocationListener.NETWORK);
                        String str = "";
                        Location loc = null;

                        if(loc1 == null && loc2 != null)
                                loc = loc2;
                        else if (loc1 != null && loc2 == null)
                                loc = loc1;
                        else if (loc1 == null && loc2 == null)
                                loc = null;
                        else
                                loc = loc1.getTime() > loc2.getTime() ? loc1 : loc2;

                        if(boolGpsEnabled){}
                        else if(boolNetworkEnabled)
                                str = getString(R.string.message_gpsdisabled);
                        else if(loc == null)
                                str = getString(R.string.message_locationunavailable);
                        else
                                str = getString(R.string.message_lastknownlocation);

                        if(str.length() > 0)
                                Toast.makeText(this, str,
Toast.LENGTH_LONG).show();
```

```
                        if(loc != null)

        mMap.getController().setCenter(TypeConverter.locationToGeoPoint(loc));
                }
        }

        private void setTitle(){
                try {
                        final TextView leftText = (TextView) findViewById(R.id.left_text);
                        if(leftText != null)
                                leftText.setText(mMap.getTileSource().NAME);

                        final TextView gpsText = (TextView)
findViewById(R.id.gps_text);
                        if(gpsText != null){
                                gpsText.setText(mGpsStatusName);
                        }

                        final TextView rightText = (TextView)
findViewById(R.id.right_text);
                        if(rightText != null){
                                final double zoom = mMap.getZoomLevelScaled();
                                if(zoom > mMap.getTileSource().ZOOM_MAXLEVEL)

        rightText.setText(""+(mMap.getTileSource().ZOOM_MAXLEVEL+1)+"+");
                                else
                                        rightText.setText(""+(1 + Math.round(zoom)));
                        }
                } catch (Exception e) {
                }
        }

        @Override
        protected void onResume() {
                final SharedPreferences pref =
getPreferences(Activity.MODE_PRIVATE);

                final String mapId = pref.getString(MAPNAME, TileSource.MAPNIK);
                mOverlayId = pref.getString("OverlayID", "");
                mShowOverlay = pref.getBoolean("ShowOverlay", true);

                setTileSource(mapId, mOverlayId, mShowOverlay);

                mMap.getController().setZoom(pref.getInt("ZoomLevel", 0));
                setTitle();

                FillOverlays();

                if(mCompassEnabled)
                        mOrientationSensorManager.registerListener(mListener,
mOrientationSensorManager
                                .getDefaultSensor(Sensor.TYPE_ORIENTATION),
SensorManager.SENSOR_DELAY_UI);

                if(mTrackOverlay != null)
                        mTrackOverlay.setStopDraw(false);
                if(mCurrentTrackOverlay != null)
```

```
                mCurrentTrackOverlay.onResume();

                mLocationListener.getBestProvider();

                if (pref.getBoolean("pref_keepscreenon", true)) {
                myWakeLock = ((PowerManager)
getSystemService(POWER_SERVICE)).newWakeLock(
                                PowerManager.SCREEN_BRIGHT_WAKE_LOCK |
PowerManager.ON_AFTER_RELEASE, "RMaps");
                myWakeLock.acquire();
        } else {
                myWakeLock = null;
        }

                super.onResume();
        }

        @Override
        protected void onRestart() {
                if(mTrackOverlay != null)
                        mTrackOverlay.clearTrack();

                super.onRestart();
        }

        @Override
        protected void onPause() {
                final GeoPoint point = mMap.getMapCenter();

                SharedPreferences uiState = getPreferences(Activity.MODE_PRIVATE);
                SharedPreferences.Editor editor = uiState.edit();
                editor.putString("MapName", mTileSource.ID);
                editor.putString("OverlayID", mTileSource.getOverlayName());
                editor.putBoolean("ShowOverlay", mShowOverlay);
                editor.putInt("Latitude", point.getLatitudeE6());
                editor.putInt("Longitude", point.getLongitudeE6());
                editor.putInt("ZoomLevel", mMap.getZoomLevel());
                editor.putBoolean("CompassEnabled", mCompassEnabled);
                editor.putBoolean("AutoFollow", mAutoFollow);
                editor.putString("app_version", Ut.getAppVersion(this));
                if(mPoiOverlay != null)
                        editor.putInt("curShowPoiId", mPoiOverlay.getTapIndex());
                mSearchResultOverlay.toPref(editor);
                editor.commit();

                uiState = getSharedPreferences("MapName", Activity.MODE_PRIVATE);
                editor = uiState.edit();
                editor.putString("MapName", mTileSource.ID);
                editor.putInt("Latitude", point.getLatitudeE6());
                editor.putInt("Longitude", point.getLongitudeE6());
                editor.putInt("ZoomLevel", mMap.getZoomLevel());
                editor.putBoolean("CompassEnabled", mCompassEnabled);
                editor.putBoolean("AutoFollow", mAutoFollow);
                editor.commit();

                if (myWakeLock != null)
                        myWakeLock.release();
```

```
                if(mOrientationSensorManager != null)
                        mOrientationSensorManager.unregisterListener(mListener);

                if(mCurrentTrackOverlay != null)
                        mCurrentTrackOverlay.onPause();

                mPoiManager.FreeDatabases();


        mLocationListener.getLocationManager().removeUpdates(mLocationListener);
                if(mNetListener != null)

        mLocationListener.getLocationManager().removeUpdates(mNetListener);

                super.onPause();
        }

        @Override
        protected void onDestroy() {
                for (TileViewOverlay osmvo : mMap.getOverlays())
                        osmvo.Free();

                mTileSource.Free();
                mTileSource = null;
                mMap.setMoveListener(null);
                mTracker.stopSession();

                super.onDestroy();
        }

        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
                super.onCreateOptionsMenu(menu);

                MenuInflater inflater = getMenuInflater();
                inflater.inflate(R.menu.main_option_menu, menu);

                return true;
        }

        @Override
        public boolean onPrepareOptionsMenu(Menu menu) {
                Menu submenu = menu.findItem(R.id.mapselector).getSubMenu();
                submenu.clear();
                SharedPreferences pref =
PreferenceManager.getDefaultSharedPreferences(this);

                File folder = Ut.getRMapsMapsDir(this);
                if (folder.exists()) {
                        File[] files = folder.listFiles();
                        if (files != null)
                                for (int i = 0; i < files.length; i++) {
                                        if
(files[i].getName().toLowerCase().endsWith(".mnm")
                                                        ||
files[i].getName().toLowerCase().endsWith(".tar")
                                                        ||
files[i].getName().toLowerCase().endsWith(".sqlitedb")) {
```

```
                                        String name =
Ut.FileName2ID(files[i].getName());
                                        if (pref.getBoolean("pref_usermaps_" +
name + "_enabled", false)) {
                                            MenuItem item =
submenu.add(pref.getString("pref_usermaps_" + name + "_name",
        files[i].getName()));
        item.setTitleCondensed("usermap_" + name);
                                        }
                                    }
                                }
            }

            Cursor c = mPoiManager.getGeoDatabase().getMixedMaps();
            if(c != null) {
                if(c.moveToFirst()) {
                    do {
                                        if (pref.getBoolean("PREF_MIXMAPS_" +
c.getInt(0) + "_enabled", false) && c.getInt(2) < 3) {
                                            MenuItem item =
submenu.add(c.getString(1));
                                            item.setTitleCondensed("mixmap_" +
c.getInt(0));
                                        }
                    } while(c.moveToNext());
                }
                c.close();
            }

            final SAXParserFactory fac = SAXParserFactory.newInstance();
            SAXParser parser = null;
            try {
                parser = fac.newSAXParser();
                if(parser != null){
                    final InputStream in =
getResources().openRawResource(R.raw.predefmaps);
                    parser.parse(in, new PredefMapsParser(submenu,
pref));
                }
            } catch (Exception e) {
                e.printStackTrace();
            }

            return super.onPrepareOptionsMenu(menu);
        }

        @Override
        public boolean onOptionsItemSelected(MenuItem item) {
            super.onOptionsItemSelected(item);
            final GeoPoint point = mMap.getMapCenter();

            switch (item.getItemId()) {
            case (R.id.area_selector):
                startActivity(new Intent(this,
AreaSelectorActivity.class).putExtra(MAPNAME, mTileSource.ID).putExtra("Latitude",
point.getLatitudeE6()).putExtra("Longitude",
```

```
point.getLongitudeE6()).putExtra("ZoomLevel", mMap.getZoomLevel()));
                        return true;
                case (R.id.poilist):
                        startActivityForResult((new Intent(this,
PoiListActivity.class)).putExtra("lat", point.getLatitude()).putExtra("lon",
point.getLongitude()).putExtra("title", "POI"), R.id.poilist);
                        return true;
                case (R.id.tracks):
                        startActivityForResult(new Intent(this, TrackListActivity.class),
R.id.tracks);
                        return true;
                case (R.id.search):
                        onSearchRequested();
                        return true;
                case (R.id.settings):
                        startActivityForResult(new Intent(this, MainPreferences.class),
R.id.settings_activity_closed);
                        return true;
                case (R.id.about):
                        showDialog(R.id.about);
                        return true;
                case (R.id.mapselector):
                        return true;
                case (R.id.compass):
                        mCompassEnabled = !mCompassEnabled;
                        mCompassView.setVisibility(mCompassEnabled ? View.VISIBLE
: View.INVISIBLE);
                        if(mCompassEnabled)
                                mOrientationSensorManager.registerListener(mListener,
mOrientationSensorManager

        .getDefaultSensor(Sensor.TYPE_ORIENTATION),
SensorManager.SENSOR_DELAY_UI);
                        else {

        mOrientationSensorManager.unregisterListener(mListener);
                                mMap.setBearing(0);
                        };
                        return true;
                case (R.id.mylocation):
                        setAutoFollow(true);
                        setLastKnownLocation();
                        return true;
                default:

                        final String mapid = (String)item.getTitleCondensed();
                        setTileSource(mapid, "", true);

                        if(mTileSource.MAP_TYPE == TileSource.PREDEF_ONLINE) {
                                mTracker.setCustomVar(1, "MAP", mapid);
                                mTracker.trackPageView("/maps");
                        }

                        FillOverlays();

            setTitle();

                        return true;
```

```
                }

        }

        private void setTileSource(String aMapId, String aOverlayId, boolean
aShowOverlay) {
                final String mapId = aMapId == null ? (mTileSource == null ?
TileSource.MAPNIK : mTileSource.ID) : aMapId;
                final String overlayId = aOverlayId == null ? mOverlayId : aOverlayId;
                final String lastMapID = mTileSource == null ? TileSource.MAPNIK :
mTileSource.ID;

                if(mTileSource != null) mTileSource.Free();

                if(overlayId != null && !overlayId.equalsIgnoreCase("") &&
aShowOverlay) {
                        mOverlayId = overlayId;
                        mShowOverlay = true;
                        try {
                                mTileSource = new TileSource(this, mapId, overlayId);
                        } catch (SQLiteException e) {
                                mTileSource = null;
                        } catch (RException e) {
                                mTileSource = null;
                                addMessage(e);
                        }
                } else {
                        try {
                                mTileSource = new TileSource(this, mapId,
aShowOverlay);
                                mShowOverlay = aShowOverlay;
                                if(mapId != lastMapID)
                                        mOverlayId = "";
                        } catch (SQLiteException e) {
                                mTileSource = null;
                        } catch (RException e) {
                                mTileSource = null;
                                addMessage(e);
                        }
                }

                mMap.setTileSource(mTileSource);

                if(mPrefOverlayButtonVisibility == 2)
                        mOverlayView.setVisibility(mTileSource.MAP_TYPE ==
TileSourceBase.MIXMAP_PAIR || mTileSource.YANDEX_TRAFFIC_ON == 1 ?
View.VISIBLE : View.GONE);
        }

        private void addMessage(RException e) {

                LinearLayout msgbox = (LinearLayout) findViewById(e.getID());
                if(msgbox == null) {
                        msgbox = (LinearLayout)
LayoutInflater.from(this).inflate(R.layout.error_message_box, (ViewGroup)
findViewById(R.id.message_list));
                        msgbox.setId(e.getID());
                }
```

```
                msgbox.setVisibility(View.VISIBLE);
                ((TextView)
msgbox.findViewById(R.id.descr)).setText(e.getStringRes(this));
                msgbox.findViewById(R.id.message).setOnClickListener(new
OnClickListener() {

                        public void onClick(View v) {
                                if (v.findViewById(R.id.descr).getVisibility() ==
View.GONE)

        v.findViewById(R.id.descr).setVisibility(View.VISIBLE);
                                else

        v.findViewById(R.id.descr).setVisibility(View.GONE);
                                }
                });
                msgbox.findViewById(R.id.btn).setTag(Integer.valueOf(e.getID()));
                msgbox.findViewById(R.id.btn).setOnClickListener(new
OnClickListener() {

                        public void onClick(View v) {
                                final int id = (Integer)v.getTag();
                                findViewById(id).setVisibility(View.GONE);
                        }
                });
        }

        @Override
        public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenuInfo menuInfo) {
                if(menuInfo instanceof TileView.PoiMenuInfo &&
((TileView.PoiMenuInfo) menuInfo).MarkerIndex >= 0) {
                        menu.add(0, R.id.menu_editpoi, 0, getText(R.string.menu_edit));
                        menu.add(0, R.id.menu_hide, 0, getText(R.string.menu_hide));
                        menu.add(0, R.id.menu_deletepoi, 0,
getText(R.string.menu_delete));
                } else {
                        menu.add(0, R.id.menu_addpoi, 0,
getText(R.string.menu_addpoi));
                }

                super.onCreateContextMenu(menu, v, menuInfo);
        }

        @Override
        public boolean onContextItemSelected(MenuItem item) {
                if (item.getGroupId() == R.id.isoverlay) {
                        final String overlayid = (String)item.getTitleCondensed();
                        setTileSource(mTileSource.ID, overlayid, true);

                        if(mTileSource.MAP_TYPE == TileSource.PREDEF_ONLINE) {
                                mTracker.setCustomVar(1, "OVERLAY", overlayid);
                                mTracker.trackPageView("/overlays");
                        }

                        FillOverlays();

            setTitle();
```

```
                } else {
                        switch (item.getItemId()) {
                        case R.id.hide_overlay:
                                setTileSource(mTileSource.ID, mOverlayId, false);

                                FillOverlays();
                        setTitle();
                                break;
                        case R.id.menu_addpoi:
                                GeoPoint point = ((TileView.PoiMenuInfo)
item.getMenuInfo()).EventGeoPoint;
                                startActivityForResult((new Intent(this,
PoiActivity.class)).putExtra("lat", point.getLatitude()).putExtra("lon", point.getLongitude())
                                                .putExtra("title", "POI"),
R.id.menu_addpoi);
                                break;
                        case R.id.menu_editpoi:
                                startActivityForResult((new Intent(this,
PoiActivity.class)).putExtra("pointid", mPoiOverlay.getPoiPoint(mMarkerIndex).getId()),
                                                R.id.menu_editpoi);
                                mMap.postInvalidate();
                                break;
                        case R.id.menu_deletepoi:

        mPoiManager.deletePoi(mPoiOverlay.getPoiPoint(mMarkerIndex).getId());
                                mPoiOverlay.UpdateList();
                                mMap.postInvalidate();
                                break;
                        case R.id.menu_hide:
                                final PoiPoint poi =
mPoiOverlay.getPoiPoint(mMarkerIndex);
                                poi.Hidden = true;
                                mPoiManager.updatePoi(poi);
                                mPoiOverlay.UpdateList();
                                mMap.postInvalidate();
                                break;
                        case R.id.menu_toradar:
                                final PoiPoint poi1 =
mPoiOverlay.getPoiPoint(mMarkerIndex);
                                try {
                                        Intent i = new
Intent("com.google.android.radar.SHOW_RADAR");

        i.setFlags(Intent.FLAG_ACTIVITY_REORDER_TO_FRONT);
                                        i.putExtra("name", poi1.Title);
                                        i.putExtra("latitude", (float)
(poi1.GeoPoint.getLatitudeE6() / 1000000f));
                                        i.putExtra("longitude", (float)
(poi1.GeoPoint.getLongitudeE6() / 1000000f));
                                        startActivity(i);
                                } catch (Exception e) {

                                }
                                break;
                        }
                }
                return super.onContextItemSelected(item);
```

```
        }

        @Override
        protected Dialog onCreateDialog(int id) {
                switch (id) {
                case R.id.add_yandex_bookmark:
                        return new AlertDialog.Builder(this)
                                .setTitle(R.string.ya_dialog_title)
                                .setMessage(R.string.ya_dialog_message)
                                .setPositiveButton(R.string.ya_dialog_button_caption,
new DialogInterface.OnClickListener() {
                                                public void onClick(DialogInterface
dialog, int whichButton) {

        Browser.saveBookmark(MainActivity.this, "ÃÓ·ËÎ¸Ì¹È fÌ‰ÅÍÒ", "m.yandex.ru");
                                                }
                                }).create();
                case R.id.whatsnew:
                        return new AlertDialog.Builder(this) //.setIcon(
R.drawable.alert_dialog_icon)
                                        .setTitle(R.string.about_dialog_whats_new)
                                        .setMessage(R.string.whats_new_dialog_text)
                                        .setNegativeButton(R.string.about_dialog_close,
new DialogInterface.OnClickListener() {
                                                public void onClick(DialogInterface
dialog, int whichButton) {

                                                        /* User clicked Cancel so do
some stuff */
                                                }
                                        }).create();
                case R.id.about:
                        return new AlertDialog.Builder(this)
//.setIcon(R.drawable.alert_dialog_icon)
                                        .setTitle(R.string.menu_about)
                                        .setMessage(getText(R.string.app_name) + " v."
+ Ut.getAppVersion(this) + "\n\n"
                                                        +
getText(R.string.about_dialog_text))

        .setPositiveButton(R.string.about_dialog_whats_new, new
DialogInterface.OnClickListener() {
                                                public void onClick(DialogInterface
dialog, int whichButton) {

                                                        showDialog(R.id.whatsnew);
                                                }

        }).setNegativeButton(R.string.about_dialog_close, new
DialogInterface.OnClickListener() {
                                                public void onClick(DialogInterface
dialog, int whichButton) {

                                                        /* User clicked Cancel so do
some stuff */
                                                }
                                        }).create();
                case R.id.error:
```

```
                return new AlertDialog.Builder(this)
//.setIcon(R.drawable.alert_dialog_icon)
                .setTitle(R.string.error_title)
                .setMessage(getText(R.string.error_text))
                .setPositiveButton(R.string.error_send, new
DialogInterface.OnClickListener() {
                        @SuppressWarnings("static-access")
                        public void onClick(DialogInterface dialog, int
whichButton) {

                                SharedPreferences settings =
getPreferences(Activity.MODE_PRIVATE);
                                String text =  settings.getString("error", "");
                                String subj = "ID FishingMap error: ";
                                try {
                                        final String[] lines = text.split("\n", 2);
                                        final Pattern p =
Pattern.compile("[.][\\w]+[:| |\\t|\\n]");
                                        final Matcher m =
p.matcher(lines[0]+"\n");
                                        if (m.find())
                                                subj += m.group().replace(".",
"").replace(":", "").replace("\n", "")+" at ";
                                        final Pattern p2 =
Pattern.compile("[.][\\w]+[(][\\w| |\\t]*[)]");
                                        final Matcher m2 = p2.matcher(lines[1]);
                                        if (m2.find())
                                                subj += m2.group().substring(2);
                                } catch (Exception e) {
                                }

                                final Build b = new Build();
                                final Build.VERSION v = new Build.VERSION();
                                text = "Your message:"
                                        +"\n\nID FishingMap:
"+Ut.getAppVersion(MainActivity.this)
                                        +"\nAndroid: "+v.RELEASE
                                        +"\nDevice: "+b.BOARD+"
"+b.BRAND+" "+b.DEVICE+/*" "+b.MANUFACTURER+*/" "+b.MODEL+" "+b.PRODUCT
                                        +"\n\n"+text;

                                startActivity(Ut.SendMail(subj, text));

                                SharedPreferences uiState = getPreferences(0);
                                SharedPreferences.Editor editor = uiState.edit();
                                editor.putString("error", "");
                                editor.commit();

                        }
                }).setNegativeButton(R.string.about_dialog_close, new
DialogInterface.OnClickListener() {
                        public void onClick(DialogInterface dialog, int
whichButton) {

                                SharedPreferences uiState = getPreferences(0);
                                SharedPreferences.Editor editor = uiState.edit();
                                editor.putString("error", "");
                                editor.commit();
```

```
                        }
                    }).create();

            }
            return null;
        }

        @Override
        protected void onActivityResult(int requestCode, int resultCode, Intent data) {
            switch(requestCode){
            case R.id.menu_addpoi:
            case R.id.menu_editpoi:
                mPoiOverlay.UpdateList();
                mMap.postInvalidate();
                break;
            case R.id.poilist:
                if(resultCode == RESULT_OK){
                    PoiPoint point =
mPoiManager.getPoiPoint(data.getIntExtra("pointid", PoiPoint.EMPTY_ID()));
                    if(point != null){
                        setAutoFollow(false);
                        mPoiOverlay.UpdateList();
                        mMap.getController().setCenter(point.GeoPoint);
                    }
                } else {
                    mPoiOverlay.UpdateList();
                    mMap.postInvalidate();
                }
                break;
            case R.id.tracks:
                if(resultCode == RESULT_OK){
                    Track track =
mPoiManager.getTrack(data.getIntExtra("trackid", PoiPoint.EMPTY_ID()));
                    if(track != null){
                        setAutoFollow(false);

        mMap.getController().setCenter(track.getBeginGeoPoint());
                    }
                }
                break;
            case R.id.settings_activity_closed:
                finish();
                startActivity(new Intent(this, this.getClass()));
                break;
            }

            super.onActivityResult(requestCode, resultCode, data);
        }

        private class MainActivityCallbackHandler extends Handler{
            @Override
            public void handleMessage(final Message msg) {
                final int what = msg.what;
                switch(what){
                    case Ut.MAPTILEFSLOADER_SUCCESS_ID:
                        mMap.postInvalidate();
                        break;
                    case R.id.user_moved_map:
```

```
                                        //setAutoFollow(false);
                                        break;
                                case R.id.set_title:
                                        setTitle();
                                        break;
                                case R.id.add_yandex_bookmark:
                                        showDialog(R.id.add_yandex_bookmark);
                                        break;
                                case Ut.ERROR_MESSAGE:
                                        if(msg.obj != null)
                                                Toast.makeText(MainActivity.this,
msg.obj.toString(), Toast.LENGTH_LONG).show();
                                        break;
                        }
                }
        }

        private boolean mGPSFastUpdate;
        private SampleLocationListener mLocationListener, mNetListener;

        private class SampleLocationListener implements LocationListener {
                public static final String GPS = "gps";
                public static final String NETWORK = "network";
                public static final String OFF = "off";

                public void onLocationChanged(Location loc) {
                        mMyLocationOverlay.setLocation(loc);
                        Ut.d("onLocationChanged " + loc.getProvider());

                        if (loc.getProvider().equals(GPS) && mNetListener != null) {
                                getLocationManager().removeUpdates(mNetListener);
                                mNetListener = null;
                                Ut.d("NETWORK provider removed");
                        }

                        //int cnt = loc.getExtras().getInt("satellites", Integer.MIN_VALUE);
                        mGpsStatusName = loc.getProvider(); // + " 2 " + (cnt >= 0 ? cnt :
0);
                        setTitle();

                        mLastSpeed = loc.getSpeed();

                        if (mAutoFollow) {
                                if (mDrivingDirectionUp)
                                        if (loc.getSpeed() > 0.5)
                                                mMap.setBearing(loc.getBearing());


        mMap.getController().setCenter(TypeConverter.locationToGeoPoint(loc));
                        } else
                                mMap.invalidate();

                        setTitle();
                }

                public void onProviderDisabled(String provider) {
                        Ut.d("onProviderDisabled "+provider);
                        getBestProvider();
```

```
		}

		public void onProviderEnabled(String provider) {
			Ut.d("onProviderEnabled "+provider);
			getBestProvider();
		}

		public void onStatusChanged(String provider, int status, Bundle extras)
{
			Ut.d("onStatusChanged "+provider);
			mGpsStatusSatCnt = extras.getInt("satellites",
Integer.MIN_VALUE);
			mGpsStatusState = status;
			mGpsStatusName = provider;
			Ut.d(provider+" status: "+status+" cnt: "+extras.getInt("satellites",
Integer.MIN_VALUE));
			setTitle();
		}

		private LocationManager getLocationManager() {
			return (LocationManager)
getSystemService(Context.LOCATION_SERVICE);
		}

		private void getBestProvider() {
			int minTime = 0;
			int minDistance = 0;

			if (!mGPSFastUpdate) {
				minTime = 2000;
				minDistance = 20;
			}
			;

			getLocationManager().removeUpdates(mLocationListener);
			if (mNetListener != null)
				getLocationManager().removeUpdates(mNetListener);

			if (getLocationManager().isProviderEnabled(GPS)) {
				Ut.d("GPS Provider Enabled");
				getLocationManager().requestLocationUpdates(GPS,
minTime, minDistance, mLocationListener);
				mGpsStatusName = GPS;

				try {
					if
(getLocationManager().isProviderEnabled(NETWORK)) {
						Ut.d("NETWORK Provider Enabled");
						mNetListener = new
SampleLocationListener();

	getLocationManager().requestLocationUpdates(NETWORK, minTime,
minDistance, mNetListener);
						mGpsStatusName = NETWORK;
					}
				} catch (Exception e) {
					e.printStackTrace();
				}
```

```
                        } else if (getLocationManager().isProviderEnabled(NETWORK))
{
                                Ut.d("only NETWORK Provider Enabled");

        getLocationManager().requestLocationUpdates(NETWORK, minTime,
minDistance, mLocationListener);
                                mGpsStatusName = NETWORK;
                        } else {
                                Ut.d("NO Provider Enabled");
                                mGpsStatusName = OFF;
                        }

                        setTitle();
                }
        }

        private class MoveListener implements IMoveListener {

                public void onMoveDetected() {
                        if(mAutoFollow)
                                setAutoFollow(false);
                }

                public void onZoomDetected() {
                        setTitle();
                }

        }

        private final SensorEventListener mListener = new SensorEventListener() {
                private int iOrientation = -1;

                public void onAccuracyChanged(Sensor sensor, int accuracy) {

                }

                public void onSensorChanged(SensorEvent event) {
                        if (iOrientation < 0)
                                iOrientation = ((WindowManager)
getSystemService(Context.WINDOW_SERVICE))
                                                .getDefaultDisplay().getOrientation();

                        mCompassView.setAzimuth(event.values[0] + 0 * iOrientation);
                        mCompassView.invalidate();

                        if (mCompassEnabled)
                                if (mNorthDirectionUp)
                                        if (mDrivingDirectionUp == false || mLastSpeed
== 0) {

        mMap.setBearing(updateBearing(event.values[0]) + 0 * iOrientation);
                                                mMap.invalidate();
                                        }
                }

        };
```

```
        private float updateBearing(float newBearing) {
                float dif = newBearing - mLastBearing;
                // find difference between new and current position
                if (Math.abs(dif) > 180)
                        dif = 360 - dif;
                // if difference is bigger than 180 degrees,
                // it's faster to rotate in opposite direction
                if (Math.abs(dif) < 1)
                        return mLastBearing;
                // if difference is less than 1 degree, leave things as is
                if (Math.abs(dif) >= 90)
                        return mLastBearing = newBearing;
                // if difference is bigger than 90 degress, just update it
                mLastBearing += 90 * Math.signum(dif) * Math.pow(Math.abs(dif) / 90,
2);
                // bearing is updated proportionally to the square of the difference
                // value
                // sign of difference is paid into account
                // if difference is 90(max. possible) it is updated exactly by 90
                while (mLastBearing > 360)
                        mLastBearing -= 360;
                while (mLastBearing < 0)
                        mLastBearing += 360;
                // prevent bearing overrun/underrun
                return mLastBearing;
        }


        private void ActionShowPoints(Intent queryIntent) {
                final ArrayList<String> locations =
queryIntent.getStringArrayListExtra("locations");
                if(!locations.isEmpty()){
                        Ut.dd("Intent: "+ACTION_SHOW_POINTS+" locations:
"+locations.toString());
                        String [] fields = locations.get(0).split(";");
                        String locns = "", title = "", descr = "";
                        if(fields.length>0) locns = fields[0];
                        if(fields.length>1) title = fields[1];
                        if(fields.length>2) descr = fields[2];

                        GeoPoint point = GeoPoint.fromDoubleString(locns);
                        mPoiOverlay.setGpsStatusGeoPoint(point, title, descr);
                        setAutoFollow(false);
                        mMap.getController().setCenter(point);
                }
        }

}
```

## 2. Script Map View

```
package com.robert.maps.view;

import java.util.List;

import org.andnav.osm.util.GeoPoint;

import android.content.Context;
import android.content.SharedPreferences;
import android.content.res.TypedArray;
import android.preference.PreferenceManager;
import android.util.AttributeSet;
import android.view.ContextMenu.ContextMenuInfo;
import android.view.KeyEvent;
import android.view.View;
import android.widget.ImageView;
import android.widget.RelativeLayout;

import com.robert.maps.R;
import com.robert.maps.tileprovider.TileSource;
import com.robert.maps.utils.ScaleBarDrawable;
import com.robert.maps.utils.Ut;

public class MapView extends RelativeLayout {
        public static final int ZOOM_CONTROL_HIDE = 0;
        public static final int ZOOM_CONTROL_TOP = 1;
        public static final int ZOOM_CONTROL_BOTTOM = 2;
        public static final String MAPNAME = "MapName";

        private final TileView mTileView;
        private final MapController mController;
        private IMoveListener mMoveListener;
        private boolean mStopBearing = false;

        public MapView(Context context, int sideInOutButtons, int scaleBarVisible) {
                super(context);

                mController = new MapController();
                mTileView = new TileView(context);
                mMoveListener = null;
                addView(mTileView, new
RelativeLayout.LayoutParams(LayoutParams.FILL_PARENT,
LayoutParams.FILL_PARENT));

                displayZoomControls(sideInOutButtons);

                if (scaleBarVisible == 1) {
              final ImageView ivScaleBar = new ImageView(getContext());
                        final ScaleBarDrawable dr = new ScaleBarDrawable(context,
this, 0/*Integer.parseInt(pref.getString("pref_units",
                                        "0"))*/);
                        ivScaleBar.setImageDrawable(dr);
                        final RelativeLayout.LayoutParams scaleParams = new
RelativeLayout.LayoutParams(

        RelativeLayout.LayoutParams.WRAP_CONTENT,
RelativeLayout.LayoutParams.WRAP_CONTENT);
```

```
                        scaleParams.addRule(RelativeLayout.RIGHT_OF,
R.id.whatsnew);

        scaleParams.addRule(RelativeLayout.ALIGN_PARENT_BOTTOM);
                        addView(ivScaleBar, scaleParams);
                }

                setFocusable(true);
                setFocusableInTouchMode(true);
        }

        public MapView(Context context) {
                this(context, 1, 1);

        }

        public MapView(Context context, AttributeSet attrs) {
                super(context, attrs);

                TypedArray a = context.obtainStyledAttributes(attrs,
R.styleable.MapView);

                mController = new MapController();
                mTileView = new TileView(context);
                addView(mTileView, new
RelativeLayout.LayoutParams(LayoutParams.FILL_PARENT,
LayoutParams.FILL_PARENT));

                final int sideBottom = a.getInt(R.styleable.MapView_SideInOutButtons,
0);
                displayZoomControls(sideBottom);

                if (a.getInt(R.styleable.MapView_SideInOutButtons, 0) == 1) {
              final ImageView ivScaleBar = new ImageView(getContext());
                        final ScaleBarDrawable dr = new ScaleBarDrawable(context,
this, 0/*Integer.parseInt(pref.getString("pref_units",
                                        "0"))*/);
                        ivScaleBar.setImageDrawable(dr);
                        final RelativeLayout.LayoutParams scaleParams = new
RelativeLayout.LayoutParams(

        RelativeLayout.LayoutParams.WRAP_CONTENT,
RelativeLayout.LayoutParams.WRAP_CONTENT);
                        scaleParams.addRule(RelativeLayout.RIGHT_OF,
R.id.whatsnew);

        scaleParams.addRule(RelativeLayout.ALIGN_PARENT_BOTTOM);
                        addView(ivScaleBar, scaleParams);
                }

    a.recycle();
        }

        public TileView getTileView() {
                return mTileView;
        }

        public class MapController {
```

```
		public void setCenter(GeoPoint point) {
			mTileView.setMapCenter(point);
		}

		public void setZoom(int zoom) {
			mTileView.setZoomLevel(zoom);
		}

		public void zoomOut() {
			mTileView.setZoomLevel(mTileView.getZoomLevel() - 1);
		}

		public void zoomIn() {
			mTileView.setZoomLevel(mTileView.getZoomLevel() + 1);
		}
}

	public MapController getController() {
		return mController;
	}

	public void setTileSource(TileSource tilesource) {
		mTileView.setTileSource(tilesource);
	}

	public TileSource getTileSource() {
		return mTileView.getTileSource();
	}

	public void displayZoomControls(final boolean takeFocus) {
		displayZoomControls(1);
	}

	public void displayZoomControls(final int SideInOutButtons) {
		if(SideInOutButtons == 0) return;

    final ImageView ivZoomIn = new ImageView(getContext());
    ivZoomIn.setImageResource(R.drawable.zoom_in);
    final RelativeLayout.LayoutParams zoominParams = new
RelativeLayout.LayoutParams(RelativeLayout.LayoutParams.WRAP_CONTENT,
RelativeLayout.LayoutParams.WRAP_CONTENT);
    zoominParams.addRule(RelativeLayout.ALIGN_PARENT_RIGHT);
    zoominParams.addRule((SideInOutButtons == 2 ? false : true) ?
RelativeLayout.ALIGN_PARENT_BOTTOM : RelativeLayout.ALIGN_PARENT_TOP);
    addView(ivZoomIn, zoominParams);
    ivZoomIn.setOnClickListener(new OnClickListener(){
                        // @Override
                        public void onClick(View v) {
                                mTileView.setZoomLevel(mTileView.getZoomLevel() +
1);
                                if(mMoveListener != null)
                                        mMoveListener.onZoomDetected();
                        }
    });
    ivZoomIn.setOnLongClickListener(new OnLongClickListener(){
                        // @Override
                        public boolean onLongClick(View v) {
                                SharedPreferences pref =
```

```
PreferenceManager.getDefaultSharedPreferences(getContext());
                                final int zoom =
Integer.parseInt(pref.getString("pref_zoommaxlevel", "17"));
                                Ut.d("ZoomIn OnLongClick pref_zoomminlevel="+zoom);
                                if (zoom > 0) {
                                        mTileView.setZoomLevel(zoom - 1);
                                        if(mMoveListener != null)
                                                mMoveListener.onZoomDetected();
                                }
                                return true;
                        }
        });

        final ImageView ivZoomOut = new ImageView(getContext());
        ivZoomOut.setId(R.id.whatsnew);
        ivZoomOut.setImageResource(R.drawable.zoom_out);
        final RelativeLayout.LayoutParams zoomoutParams = new
RelativeLayout.LayoutParams(RelativeLayout.LayoutParams.WRAP_CONTENT,
RelativeLayout.LayoutParams.WRAP_CONTENT);
        zoomoutParams.addRule(RelativeLayout.ALIGN_PARENT_LEFT);
        zoomoutParams.addRule((SideInOutButtons == 2 ? false : true) ?
RelativeLayout.ALIGN_PARENT_BOTTOM : RelativeLayout.ALIGN_PARENT_TOP);
        addView(ivZoomOut, zoomoutParams);
        ivZoomOut.setOnClickListener(new OnClickListener(){
                        // @Override
                        public void onClick(View v) {
                                mTileView.setZoomLevel(mTileView.getZoomLevel() -
1);
                                if(mMoveListener != null)
                                        mMoveListener.onZoomDetected();
                        }
        });
        ivZoomOut.setOnLongClickListener(new OnLongClickListener(){
                        // @Override
                        public boolean onLongClick(View v) {
                                SharedPreferences pref =
PreferenceManager.getDefaultSharedPreferences(getContext());
                                final int zoom =
Integer.parseInt(pref.getString("pref_zoomminlevel", "10"));
                                Ut.d("ZoomOut OnLongClick
pref_zoomminlevel="+zoom);
                                if (zoom > 0) {
                                        mTileView.setZoomLevel(zoom - 1);
                                        if(mMoveListener != null)
                                                mMoveListener.onZoomDetected();
                                }
                                return true;
                        }
        });
        }

        public int getZoomLevel() {
                return mTileView.getZoomLevel();
        }

        public double getZoomLevelScaled() {
                return mTileView.getZoomLevelScaled();
        }
```

```
        public GeoPoint getMapCenter() {
                return mTileView.getMapCenter();
        }

        public List<TileViewOverlay> getOverlays() {
                return mTileView.getOverlays();
        }

        @Override
        protected ContextMenuInfo getContextMenuInfo() {
                return mTileView.mPoiMenuInfo;
        }

        public void setBearing(float bearing) {
                if(!mStopBearing)
                        mTileView.setBearing(bearing);
        }

        public void setMoveListener(IMoveListener moveListener) {
                mMoveListener = moveListener;
                mTileView.setMoveListener(moveListener);
        }

        public double getTouchScale() {
                return mTileView.mTouchScale;
        }

        @Override
        public boolean onKeyDown(int keyCode, KeyEvent event) {
                boolean stopPropagation = false;
                switch (keyCode) {
                        case KeyEvent.KEYCODE_DPAD_UP:
                                stopPropagation = true;
                                getController().zoomOut();
                                break;
                        case KeyEvent.KEYCODE_DPAD_DOWN:
                                stopPropagation = true;
                                getController().zoomIn();
                                break;
                        case KeyEvent.KEYCODE_DPAD_CENTER:
                                if(mStopBearing)
                                        mStopBearing = false;
                                else {
                                        setBearing(0);
                                        mStopBearing = true;
                                }
                                stopPropagation = true;
                                break;
                        default:
                                break;
                }

           if(stopPropagation) return true;

           return super.onKeyDown(keyCode, event);
        }
}
```