

| | |
|-------------------------------------|----|
| 4.2.5 Rancangan Layar Aplikasi..... | 63 |
| 4.2.6 <i>Sequence Diagram</i> | 70 |
| 4.3 Implementasi..... | 71 |
| 4.3.1 Menginstal Aplikasi..... | 71 |
| 4.3.2 Demo Aplikasi..... | 72 |
| 4.3.3 Pengujian..... | 81 |

BAB V PENUTUP

| | |
|---------------------|----|
| 5.1 Kesimpulan..... | 86 |
| 5.2 Saran..... | 86 |

| | |
|-----------------------------|----|
| DAFTAR PUSTAKA | 87 |
|-----------------------------|----|

LAMPIRAN

DAFTAR GAMBAR

| | Halaman |
|---|---------|
| Gambar 2.1 Arsitektur Android | 12 |
| Gambar 2.2 LBS Sebagai Simpang Tiga Teknologi | 14 |
| Gambar 2.3 Komponen Dasar LBS | 16 |
| Gambar 2.4 Ilustrasi Peta Pada <i>Google Map</i> | 19 |
| Gambar 2.5 Komponen GIS..... | 22 |
| Gambar 2.6 Arsitektur <i>Web Based GIS</i> | 23 |
| Gambar 3.1 Struktur Organisasi Perusahaan | 41 |
| Gambar 3.2 <i>Work Breakdown Structure</i> | 44 |
| Gambar 3.3 <i>Milestone</i> | 45 |
| Gambar 3.4 <i>Gantt Chart</i> Jadwal Proyek | 46 |
| Gambar 3.5 Struktur Organisasi Tim Proyek..... | 48 |
| Gambar 4.1 <i>Activity Diagram</i> Proses Cek Bahan | 52 |
| Gambar 4.2 <i>Activity Diagram</i> Proses Ambil Bahan | 53 |
| Gambar 4.3 <i>Activity Diagram</i> Proses Angkut Bahan..... | 54 |
| Gambar 4.4 <i>Activity Diagram</i> Proses Pemasukan Bahan | 55 |
| Gambar 4.5 <i>Use Case Diagram</i> yang Diusulkan..... | 60 |
| Gambar 4.6 Menu Awal..... | 63 |
| Gambar 4.7 Menu Pilih..... | 64 |
| Gambar 4.8 <i>Layout</i> Lokasi Quarry Pasir Kwarsa | 65 |
| Gambar 4.9 <i>Layout</i> Lokasi Quarry Batu Granit..... | 66 |
| Gambar 4.10 <i>Layout</i> Lokasi Quarry Tanah Urukan..... | 67 |
| Gambar 4.11 <i>Layout</i> Lokasi Quarry Tanah Merah | 68 |
| Gambar 4.12 <i>Layout</i> Cari Rute | 69 |
| Gambar 4.13 <i>Layout</i> Keluar | 69 |
| Gambar 4.14 <i>Sequence Diagram</i> Aplikasi..... | 70 |
| Gambar 4.15 Instalasi Aplikasi | 71 |
| Gambar 4.16 Tampilan <i>Splashscreen</i> | 72 |
| Gambar 4.17 Tampilan Menu Awal..... | 73 |


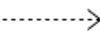







| | |
|---|----|
| Gambar 4.18 Tampilan Menu Pilih..... | 74 |
| Gambar 4.19 Tampilan Lokasi Quarry Pasir Kwarsa | 75 |
| Gambar 4.20 Tampilan Lokasi Quarry Batu Granit..... | 76 |
| Gambar 4.21 Tampilan Lokasi Quarry Tanah Uruk | 77 |
| Gambar 4.22 Tampilan Lokasi Quarry Tanah Merah..... | 78 |
| Gambar 4.23 Tampilan Rute | 79 |
| Gambar 4.24 Tampilan Kotak Dialog Keluar | 80 |


DAFTAR TABEL

| | |
|--|----|
| Tabel 3.1 <i>Stakeholder</i> | 42 |
| Tabel 3.2 Rencana Anggaran Biaya..... | 47 |
| Tabel 4.1 <i>Use Case</i> Pilih Jenis Quarry..... | 61 |
| Tabel 4.2 <i>Use Case</i> Lokasi Quarry | 61 |
| Tabel 4.3 <i>Use Case</i> Pilih <i>Site</i> Quarry..... | 61 |
| Tabel 4.4 <i>Use Case</i> Cari Titik Koordinat Pengguna..... | 62 |
| Tabel 4.5 <i>Use Case</i> Cari Rute..... | 62 |
| Tabel 4.6 Pengujian Menu Awal..... | 81 |
| Tabel 4.7 Pengujian <i>Button</i> Pilih Jenis Quarry | 81 |
| Tabel 4.8 Pengujian <i>Button</i> Lokasi Quarry Pasir Kwarsa | 81 |
| Tabel 4.9 Pengujian Peta Quarry Pasir Kwarsa | 82 |
| Tabel 4.10 Pengujian <i>Button</i> Lokasi Quarry Batu Granit..... | 82 |
| Tabel 4.11 Pengujian Peta Quarry Batu Granit..... | 82 |
| Tabel 4.12 Pengujian <i>Button</i> Lokasi Quarry Tanah Uruk..... | 83 |
| Tabel 4.13 Pengujian Peta Quarry Tanah Uruk | 83 |
| Tabel 4.14 Pengujian <i>Button</i> Lokasi Quarry Tanah Merah | 84 |
| Tabel 4.15 Pengujian Peta Quarry Tanah Merah..... | 84 |
| Tabel 4.16 Pengujian <i>Button</i> Back..... | 84 |
| Tabel 4.17 Pengujian <i>Button</i> Cari Rute | 85 |
| Tabel 4.18 Pengujian Peta Rute | 85 |
| Tabel 4.19 Pengujian <i>Button</i> Exit | 85 |






DAFTAR SIMBOL

1. Use Case Diagram

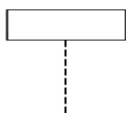
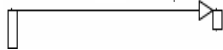

| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|-----------------------|--|
| 1 |  | <i>Actor</i> | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> . |
| 2 |  | <i>Dependency</i> | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>). |
| 3 |  | <i>Generalization</i> | Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>). |
| 4 |  | <i>Include</i> | Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> . |
| 5 |  | <i>Extend</i> | Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan. |
| 6 |  | <i>Association</i> | Apa yang menghubungkan antara objek satu dengan objek lainnya. |
| 7 |  | <i>System</i> | Menspesifikasikan paket yang menampilkan sistem secara terbatas. |
| 8 |  | <i>Use Case</i> | Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |
| 9 |  | <i>Collaboration</i> | Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi). |

| | | | |
|----|---|-------------|---|
| 10 |  | <i>Note</i> | Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi |
|----|---|-------------|---|

2. Activity Diagram

| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|----------------------------|---|
| 1 |  | <i>Activity</i> | Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain |
| 2 |  | <i>Action</i> | State dari sistem yang mencerminkan eksekusi dari suatu aksi |
| 3 |  | <i>Initial Node</i> | Bagaimana objek dibentuk atau diawali. |
| 4 |  | <i>Activity Final Node</i> | Bagaimana objek dibentuk dan dihancurkan |
| 5 |  | <i>Fork Node</i> | Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran |

3. Sequence Diagram

| NO | GAMBAR | NAMA | KETERANGAN |
|----|---|-----------------|--|
| 1 |  | <i>LifeLine</i> | Objek <i>entity</i> , antarmuka yang saling berinteraksi. |
| 2 |  | <i>Message</i> | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |
| 3 |  | <i>Message</i> | Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi |

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pada saat sekarang ini sudah banyak dijumpai pengembang/*developer* yang berkecimpung di dunia teknologi informasi seperti teknologi yang ditanamkan pada *smartphone* yaitu sistem operasi android yang memberikan kebebasan kepada para pengembang/*developers* untuk mengembangkan aplikasinya sendiri atau mengembangkan sistem operasi itu sesuai dengan kebutuhan pengguna sekarang ini.

khususnya untuk sebuah instansi perusahaan sudah banyak perusahaan sekarang yang telah menjadikan teknologi sebagai salah satu faktor penunjang dalam memajukan perusahaannya atau membantu pekerjaan para pekerjanya, khususnya dalam perusahaan yang penulis pilih sebagai penelitian skripsi penulis seperti pemetaan lokasi quarry yang dimana dapat membantu dengan memberikan titik lokasi quarry sehingga pengguna dapat memilih lokasi quarry mana yang terdekat supaya dapat menghemat waktu pekerjanya.

Perusahaan PT.INSUTAMA yang menjadi tempat penelitian penulis merupakan sebuah perusahaan yang bergerak dalam bidang konstruksi yang bertempat di Pangkal Pinang kabupaten Bangka. Pada saat ini perusahaan tersebut belum menerapkan teknologi sebagai penunjang untuk memajukan atau memudahkan pekerjaan perusahaan tersebut, seperti saat supir kendaraan yang ingin menuju ke lokasi quarry tapi belum mengetahui secara jelas dimana letak quarry tersebut dikarenakan supir tersebut berasal dari luar kota yang belum tahu seluk-beluk kawasan Kabupaten Bangka, dan hal ini dapat menurunkan kinerja proses pekerjaan sehingga pekerjaan dapat terhambat.

Aplikasi pemetaan ini merupakan suatu aplikasi yang digunakan untuk menunjukkan suatu lokasi yang bersangkutan. Oleh karena itu, penulis bermaksud membuat aplikasi Pemetaan yang mengacu pada penggunaan *Google Maps V2* untuk menentukan dengan tepat geografis posisi lokasi. Dengan aplikasi ini pengguna diharapkan dapat dengan mudah mencari lokasi dengan hanya

menggunakan fasilitas internet dan GPS melalui android dimanapun dan kapanpun.

Dalam Skripsi ini penulis akan membuat aplikasi yang berjudul “APLIKASI PEMETAAN LOKASI QUARRY MENGGUNAKAN GOOGLE MAPS V2 UNTUK SMARTPHONE ANDROID”. Diharapkan dengan dibuatnya Skripsi ini dapat memberikan kemudahan bagi para pengguna dalam pekerjaannya.

1.2 Rumusan Masalah

- a. Bagaimana membangun aplikasi pemetaan lokasi quarry menggunakan google maps v2 untuk smartphone android?
- b. Apa saja yang aplikasi dapat tampilkan saat dijalankan?
- c. Kurangnya perusahaan dalam menerapkan teknologi modern untuk meningkatkan efektifitas pekerjaan.
- d. Supir pengangkut bahan konstruksi yang masih keliru dimana lokasi quarry yang ditujui berada.

1.3 Batasan Masalah

- a. Aplikasi ini dibuat untuk *platform* android sehingga hanya dapat dijalankan pada *mobile device* yang menggunakan platform android.
- b. Aplikasi dibuat menggunakan Java Eclipse LUNA dengan JDK versi 7.
- c. Penelitian ini tidak membahas keamanan database dan keamanan jaringan yang digunakan untuk kebutuhan aplikasi.
- d. Sistem mampu menampilkan UI (*User Interface*), peta *Google Maps V2*, dan titik koordinat lokasi yang diinginkan beserta rute yang ingin ditampilkan.
- e. Aplikasi ini memanfaatkan *Google Maps API V2* sebagai penyedia peta virtual.
- f. Aplikasi ini dibuat untuk pekerja yang bertugas dalam pengangkutan bahan konstruksi dan pekerja lainnya yang termasuk didalam lingkungan perusahaan terkait.

1.4 Tujuan Penelitian

- a. Aplikasi ini dibuat untuk memberikan kemudahan pekerjaan dalam melakukan pencarian titik lokasi quarry yang berada di dalam kawasan Kabupaten Bangka.
- b. Diharapkan dapat menghemat waktu pekerjaan dalam pencarian lokasi quarry yang terdekat.
- c. Merancang aplikasi yang *user friendly* sehingga pengguna dapat mengerti cara menggunakan aplikasi yang dibuat dengan mudah.
- d. Memberikan pengenalan kepada perusahaan terhadap teknologi yang dapat digunakan untuk mempermudah suatu pekerjaan.

1.5 Metode Penelitian

Berikut metode yang digunakan dalam penelitian ini:

Penelitian yang dibuat adalah tentang “APLIKASI PEMETAAN LOKASI QUARRY MENGGUNAKAN GOOGLE MAPS V2 UNTUK SMARTPHONE ANDROID”. Penelitian dilakukan dengan menggunakan metode penelitian Waterfall. Tahapan penelitian ini dilakukan dalam 4 tahap seperti berikut yaitu Perencanaan atau *Planning*, Analisa Sistem, Perancangan aplikasi, Implementasi atau penerapan. Tahapan prosedur penelitian seperti berikut:

1.5.1 Perencanaan

Berikut merupakan langkah-langkah terhadap penelitian yang dibuat:

a. Pengumpulan Data

dalam tahap ini dilakukan sebuah studi literatur. Tahapan ini akan menghasilkan informasi yang dapat dikatakan sebagai data yang dapat dijadikan referensi, bisa berasal dari buku maupun internet.

b. Studi Kelayakan

Dalam tahap ini dilakukan penelitian terhadap ide yang didapat apakah layak dilaksanakan atau tidak. Tahap ini bertujuan untuk menilai kelayakan pembuatan proyek.

1.5.2 Analisa Sistem

Menentukan kebutuhan sistem berdasarkan data-data yang sudah diperoleh, tujuan, dan kemampuan yang dimiliki. Berikut metode yang digunakan dalam analisa sistem:

a. *Activity Diagram*

Activity Diagram merupakan pemodelan alur kerja sebuah proses sistem dan urutan aktifitas.

b. Analisa Masukan

Analisa masukan untuk memberikan informasi sebuah data untuk kebutuhan aplikasi yang akan dibangun.

c. Analisa Keluaran

Analisa keluaran adalah untuk melihat hasil dari proses data-data informasi.

d. *Use Case Diagram*

Use Case Diagram ialah untuk menjelaskan manfaat sistem yang berjalan terhadap pandangan orang yang tidak berkaitan dengan sistem.

e. Deskripsi *Use Case Diagram*

Deskripsi *Use Case Diagram* adalah gambaran fungsi masing-masing *use case* yang berada dalam sistem.

1.5.3 Perancangan Aplikasi

Pada tahap ini dilakukan perancangan terhadap sistem yang akan dibangun seperti perancangan menu utama, perancangan menu lokasi, detail atau fitur pada peta yang digunakan *,splash screen*. Karena tahap ini merupakan tahapan perancangan sistem secara keseluruhan, maka tahapan ini merupakan tahapan terpenting dalam rangkaian pembuatan aplikasi dan mempengaruhi tahapan implementasi. Berikut merupakan tahap dalam perancangan aplikasi:

a. Rancangan Dokumen Masukan

b. Rancangan Dokumen Keluaran

c. Rancangan layar program

d. *Sequence Diagram*

1.5.4 Implementasi

Penerapan aplikasi atau program yang sudah dibuat, ditujukan untuk menyelesaikan suatu pekerjaan. Dalam tahap ini dilakukan *testing* dan di uji coba atau melakukan demo aplikasi yang selanjutnya akan dipresentasikan.

1.6 Sistematika Penulisan

Secara garis besar penulisan laporan tugas akhir ini terdiri dari lima bab. Adapun sistematika penulisan laporan tugas akhir ini adalah sebagai berikut :

BAB I Pendahuluan

Dalam Bab ini dibahas mengenai latar belakang pembuatan skripsi, permasalahan yang dihadapi, tujuan yang diharapkan untuk mengatasi permasalahan, batasan-batasan dari masalah yang dibahas, metode perancangan dan sistematika penulisan skripsi.

BAB II Landasan Teori

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan yang dipakai untuk menyelesaikan permasalahan dimana berkaitan dengan aplikasi pemetaan menggunakan *google maps api v2* untuk *smartphone* android.

BAB III Pemodelan Proyek

Bab ini berisi mengenai isi dari PEP (*Project Execution Plan*), seperti *objective* proyek, identifikasi *stakeholder*, identifikasi *deliverables*, penjadwalan proyek, RAB, struktur tim proyek (jika benar-benar dibutuhkan), analisa resiko (jika diperlukan).

BAB IV Analisa dan Perancangan sistem

Bab ini membahas tentang analisis sistem dan perancangan dari pembuatan aplikasi, seperti analisa sistem dengan menggunakan *activity diagram* dan perancangan layar.

BAB V Penutup

Bab ini membahas tentang kesimpulan dan saran. Kesimpulan akan menjelaskan hasil dari evaluasi aplikasi, sedangkan saran akan menjelaskan tentang pengembangan aplikasi secara lanjut. Penulis tidak menyimpulkan masalah jika pembuktian tidak terdapat dalam hasil penelitian.

BAB II

LANDASAN TEORI

Pada bab ini penulis akan menjelaskan teori-teori yang digunakan dalam mendukung judul dan permasalahan yang dibahas sebagai dasar pemikiran.

2.1 Perangkat lunak Aplikasi

Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media.

Berikut ini adalah pengertian aplikasi menurut (Sutabri, 2012:147): “Aplikasi adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya”. Pengertian tentang Aplikasi berasal dari bahasa Inggris, yaitu “*To Apply*” yang artinya menerapkan atau terapan. Namun pengertian mengenai aplikasi secara umum adalah suatu paket program yang sudah jadi dan dapat digunakan.

2.1.1 Klasifikasi Aplikasi

Berikut merupakan klasifikasi aplikasi yang tergolong dalam beberapa kelas yaitu:

- a. Perangkat Lunak perusahaan (*enterprise*).
- b. Perangkat Lunak Infrastruktur perusahaan.
- c. Perangkat Lunak informasi kerja.
- d. Perangkat Lunak media dan hiburan.
- e. Perangkat Lunak pendidikan.
- f. Perangkat Lunak pengembangan media.
- g. Perangkat Lunak rekayasa produk.

2.1.2 Aplikasi *Mobile*

Kata *mobile* mempunyai arti bergerak atau berpindah. Sehingga diperoleh pengertian bahwa aplikasi bergerak merupakan aplikasi yang dapat dijalankan walaupun pengguna berpindah atau karena pengguna berpindah.

Pemrograman aplikasi bergerak tidak banyak berbeda dengan pemrograman aplikasi bergerak tidak banyak berbeda dengan pemrograman konvensional pada PC. Aspek karakteristik dari perangkat bergerak sering mempengaruhi arsitektur dan implementasi dari aplikasi tersebut. Dalam pemrograman aplikasi bergerak berbagai aspek teknis perangkat lebih menonjol karena memiliki banyak keterbatasan dibandingkan komputer konvensional atau PC. (Budi Darytamo dkk, 2007:1)

Perangkat bergerak (*mobile*) memiliki banyak jenis dalam hal ukuran, desain dan layout, tetapi memiliki kesamaan karakteristik yang sangat berbeda dari sistem desktop.

a. Ukuran yang kecil

Perangkat *mobile* memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.

b. Memory yang kecil

Perangkat *mobile* juga memiliki memory yang kecil, yaitu *primary* (RAM) dan *secondary* (disk). Pembatasan ini adalah salah satu faktor yang mempengaruhi penulisan program untuk berbagai jenis dari perangkat ini. Dengan pembatasan jumlah dari memory, pertimbangan-pertimbangan khusus harus diambil untuk memelihara pemakaian dari sumber daya yang mahal ini.

c. Daya proses yang terbatas

Sistem *mobile* tidaklah setangguh rekan mereka yaitu desktop. Ukuran, teknologi dan biaya adalah beberapa faktor yang mempengaruhi status dari sumber daya ini. Seperti harddisk dan RAM, anda dapat menemukan mereka dalam ukuran yang pas dengan sebuah kemasan kecil.

d. Pemakaian daya yang rendah

Perangkat mobile menghabiskan sedikit daya dibandingkan dengan mesin desktop. Perangkat ini harus menghemat daya karena mereka berjalan pada keadaan dimana daya yang disediakan dibatasi oleh baterai-baterai.

e. Kuat dan bisa diandalkan

Karena perangkat mobile selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapim benturan-benturan, gerakan, dan sesekali tetesan tetesan air.

f. Koneksi yang terbatas

Perangkat mobile memiliki bandwidth rendah, beberapa dari mereka bahkan tidak tersambung. Kebanyakan dari mereka menggunakan koneksi wireless.

g. Masa hidup yang pendek

Perangkat-perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala. Coba ambil kasus sebuah handphone, mereka booting dalam hitungan detik kebanyakan orang tidak mematikan handphone mereka bahkan ketika malam hari. PDA akan menyala jika anda menekan tombol power mereka.

2.1.3 *Application Programming Interface (API)*

Application Programming Interface (API) adalah sekumpulan commands, fungsi, dan protokol yang digunakan oleh aplikasi untuk berkomunikasi dengan sistem operasi maupun program lain seperti Database Management System(DBMS) maupun protokol komunikasi (PcMag,2013). Pembangunan aplikasi untuk Android menggunakan 3 buah API, yakni Google Maps API v2, Google Location APIs dan Google Places API. Google Maps API v2 memungkinkan pengembang untuk menampilkan peta interaktif dan kaya fitur kepada pengguna di dalam aplikasi Android. Google Location APIs memudahkan pengembang untuk membangun aplikasi yang peka terhadap lokasi pengguna, tanpa memerlukan fokus pada detail teknologi yang ada dibaliknya, sementara Google Places Api adalah sebuah layanan yang mengembalikan informasi

mengenai lokasi menggunakan HTTP request dan mengembalikan hasil dalam format JSON atau XML (Google Developer, 2013).

2.2 Android

2.2.1 Pengertian Android

Secara umum pengertian Android adalah suatu *software* (perangkat lunak) yang berbasis Linux untuk telepon seluler dan komputer tablet yang meliputi sistem operasi, *middleware* dan aplikasi inti. Android SDK menyediakan alat dan API yang diperlukan untuk memulai pengembangan aplikasi pada platform. Android menggunakan bahasa pemrograman Java, yaitu kode Java yang terkompilasi dengan data dan file resources yang dibutuhkan aplikasi dan digabungkan oleh *aapt tools* menjadi paket Android. File tersebut ditandai dengan ekstensi *.apk*, File inilah yang didistribusikan sebagai aplikasi dan dipasang pada perangkat mobile.

2.2.2 Sejarah Singkat Android

Sekitar September 2007 Google mengenalkan Nexus One, salah satu jenis telepon pintar GSM yang menggunakan Android pada sistem operasinya (Nazruddin, Safaat, 2012: hal:2). Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010).

Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, Android, perangkat bergerak (mobile) yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.

2.2.3 Mengenal Android

Android merupakan sebuah sistem operasi perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Beberapa pengertian lain dari android, yaitu:

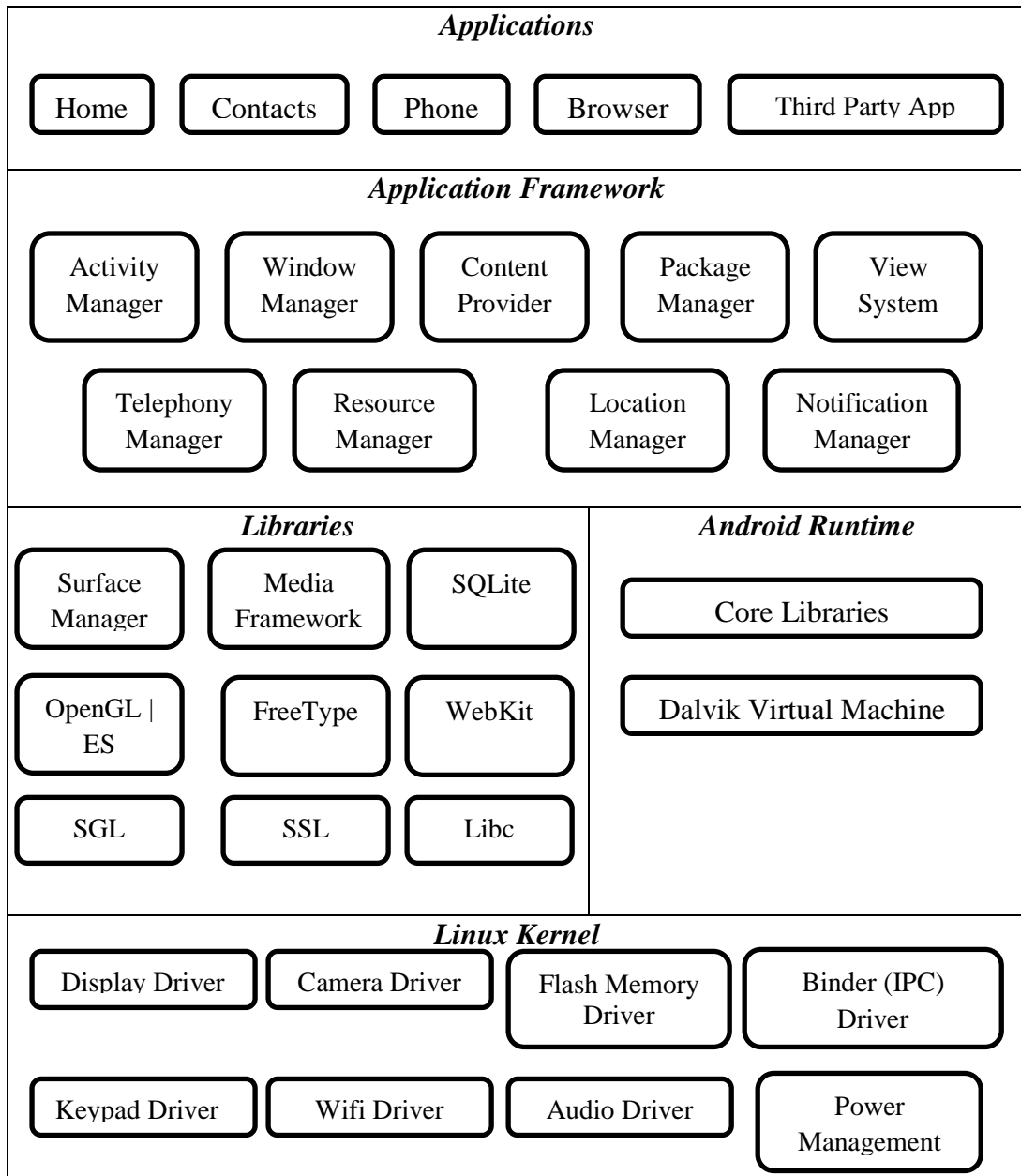
- a. Merupakan platform terbuka (*Open Source*) bagi para pengembang (*Programmer*) untuk membuat aplikasi.
- b. Merupakan sistem operasi yang dibeli **Google Inc.** dari **Android Inc.**
- c. Bukan bahasa pemrograman, tetapi hanya menyediakan lingkungan hidup atau *runtime environment* yang disebut **DVM** (*Dalvik Virtual Machine*) yang telah dioptimasi untuk alat/*device* dengan sistem memori yang kecil.

Untuk mengembangkan Android, dibentuk **OHA** (*Open Handset Aliance*), konsorsium dari 34 perusahaan peranti keras (*Hardware*), peranti lunak (*Software*), dan telekomunikasi, termasuk **Google**, **HTC**, **Intel**, **Motorola**, **Qualcomm**, **T-Mobile**, dan **Nvidia**. Pada tanggal 5 November 2007, Android dirilis pertama kali. Android bersama OHA menyatakan mendukung pengembangan *open source* pada perangkat *mobile*.

Pada saat ini juga Android bersaing dengan Apple dalam sistem operasi untuk PC Tablet. Terdapat dua jenis distributor sistem operasi android. Pertama yang mendapat dukungan penuh dari Goole atau **GMS** (*Google Mail Services*), dan kedua adalah yang benar-benar bebas distribusinya, tanpa dukungan langsung Google, atau dikenal sebagai **OHD** (*Open Handset Distribution*)

2.2.4 Arsitektur Android

Supriyanto (2012; hal 11) arsitektur android dapat dijelaskan dan digambarkan pada gambar 2.14berikut.



Gambar 2.1 Arsitektur Android

Penjelasan mengenai gambar di atas sistem operasi android terbagi menjadi:

a. *Applications*

Layer dimana akan berhubungan dengan aplikasi saja. Biasanya aplikasi yang didownload dan diinstal dan semua aplikasi yang dibuat terletak pada tingkat *applications* seperti ditemui : *phone, contact, browser*.

b. *Applications Framework*

Semacam *built-in* yang tertanam dalam sistem operasi android sehingga pengembang dapat memanfaatkannya untuk aplikasi yang sedang dibangun.

c. *Libraries*

Semua kode program yang menyediakan layanan-layanan utama sistem operasi android. Contoh library SQLite yang menyediakan dukungan database sehingga aplikasi android dapat untuk menyimpan data.

d. *Android Runtime*

Kumpulan pustaka inti yang dapat diaktifkan oleh pengembang untuk menulis kode aplikasi android. *Layer* yang membuat aplikasi android dapat dijalankan prosesnya menggunakan implementasi *linux*. *Dalvik Virtual Machine (DVM)* merupakan mesin yang membentuk dasar kerangka aplikasi android. *Runtime* android terbagi menjadi dua yaitu :

a. *Core Libraries* : aplikasi android dibangun dalam bahasa java, sementara dalvik sebagai virtual mesinnya bukan virtual mesin java, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahasa java yang ditangani oleh *core libraries*.

b. *Dalvik Virtual Machine* : virtual mesin berbasis *register* yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien dimana merupakan pengembangannya yang mampu membuat *linux kernel* untuk melakukan *threading* dan manajemen tingkat rendah.

e. *Linux Kernel*

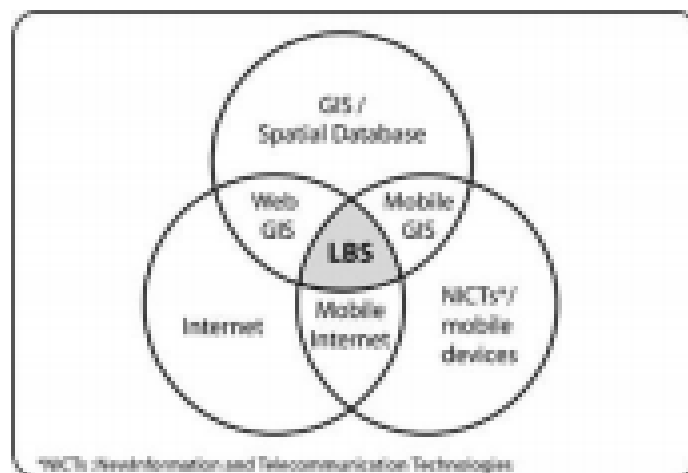
Layer inti *operating system* android itu berada. Berisi file-file sistem yang mengatur sistem processing, memory, resource, drivers dan sistem-sistem android lainnya. *Linux kernel* yang dipakai android itu *linux kernel* release 2.6.

2.2.5 Versi Android

1. Android Versi 1.1
2. Android Versi 1.5 (Cupcake)
3. Android Versi 1.6 (Donut)
4. Android Versi 2.0/2.1 (Eclair)
5. Android Versi 2.2 (Froyo)
6. Android Versi 2.3 (Gingerbread)
7. Android Versi 3.0/3.1 (Honeycomb)
8. Android Versi 4.0 (ICS : Ice Cream Sandwich)
9. Android Versi 4.1 (Jelly Bean)

2.3 Location Based Service (LBS)

Layanan Berbasis lokasi adalah layanan informasi yang dapat diakses melalui *mobile device* dengan menggunakan *mobile network*, yang dilengkapi kemampuan untuk memanfaatkan lokasi dari *mobile device* tersebut. LBS memberikan kemungkinan komunikasi dan interaksi dua arah. Oleh karena itu pengguna memberitahu penyedia layanan untuk mendapatkan informasi yang dia butuhkan, dengan referensi posisi pengguna tersebut. Layanan berbasis lokasi dapat digambarkan sebagai suatu layanan yang berada pada pertemuan tiga teknologi yaitu : *Geographic Information System*, *Internet Service*, dan *Mobile Devices*, hal ini dapat dilihat pada gambar LBS adalah pertemuan dari tiga teknologi.



Gambar 2.2 LBS sebagai simpang tiga teknologi

LBS memiliki 2 unsur utama, yaitu:

a. *Location Manager (API Maps)*

Menyediakan *tools/source* untuk LBS, API (*Application Programming Interface*) Maps. Menyediakan fasilitas untuk menampilkan, memanipulasi peta, beserta fitur-fitur lainnya, seperti tampilan satelit, jalan (*street*), maupun gabungannya.

b. *Location Providers (API Location)*

Menyediakan teknologi pencarian lokasi yang digunakan oleh perangkat (*device*). *API Location* berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi *real time*. Dengan *Location Manager*, anda dapat menemukan lokasi anda saat ini, *track* gerakan/perpindahan, serta kedekatan dengan lokasi tertentu dengan mendeteksi perpindahan.

Komponen-komponen Layanan Berbasis Lokasi juga dapat dibagi menjadi lima, yaitu:

1. *Mobile Devices*

Suatu alat yang digunakan oleh pengguna untuk meminta informasi yang dibutuhkan. Informasi dapat diberikan dalam bentuk suara, gambar, dan text.

2. *Communication Network*

Komponen kedua adalah jaringan komunikasi yang mengirim data pengguna dan informasi yang diminta dari *mobile terminal* ke *Service Provider* kemudian mengirimkan kembali informasi yang diminta ke pengguna. *Communication network* dapat berupa jaringan seluler (GSM, CDMA), *Wireless Local Area Network* (WLAN), atau *Wireless Wide Area Network* (WWAN).

3. *Positioning Component*

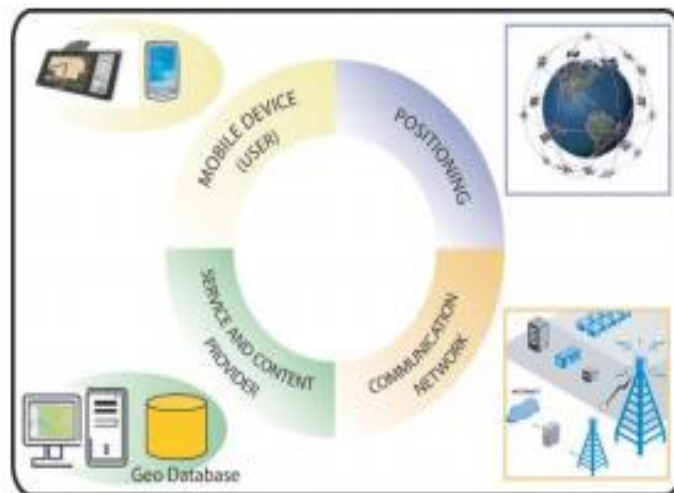
Untuk memproses suatu layanan maka posisi pengguna harus diketahui.

4. *Service and Application Provider*

Penyedia layanan menawarkan berbagai macam layanan kepada pengguna dan bertanggung jawab untuk memproses informasi yang diminta oleh pengguna.

5. *Data and Content Provider*

Penyedia layanan tidak selalu menyimpan semua data yang dibutuhkan yang bisa diakses oleh pengguna. Untuk itu, data dapat diminta dari *data and content provider*.



Gambar 2.3 Komponen dasar LBS

Selanjutnya Service and Application Provider mengirim informasi yang telah diolah melalui jaringan internet dan jaringan komunikasi. Pada akhirnya pengguna dapat menerima informasi yang diinginkan.

2.4 *Google Maps API*

API atau Application Programming Interface bukan hanya satu set class dan method atau fungsi dan signature yang sederhana. Akan tetapi API, yang bertujuan utama untuk mengatasi “*clueless*” dalam membangun software yang berukuran besar, berawal dari sesuatu yang sederhana sampai ke yang kompleks dan merupakan perilaku komponen yang sulit dipahami. Secara sederhana dapat dipahami dengan membayangkan kekacauan yang akan timbul bila mengubah database atau skema XML. Perubahan ini dapat dipermudah dengan bantuan API (Tulach, 2008).

Dari beberapa sumber yang didapat, dapat disimpulkan bahwa API adalah sekumpulan perintah, fungsi, *class* dan protokol yang memungkinkan suatu *software* berhubungan dengan *software* lainnya. Tujuan dari API adalah untuk

menghilangkan “*clueless*” dari sistem dengan cara membuat blok besar yang terdiri dari *software* di seluruh dunia dan menggunakan kembali perintah, fungsi, *class*, atau *protocol* yang mereka atau API miliki. Dengan cara ini, programmer tidak perlu lagi membuang waktu untuk membuat dan menulis infrastruktur sehingga akan menghemat waktu kerja dan lebih efisien.

Seperti yang tercatat oleh Svennerberg (2010), *Google Maps* API yang paling populer di internet. Pencatatan yang dilakukan pada bulan Mei 2010 ini menyatakan bahwa 43% mashup (aplikasi dan situs web yang menggabungkan dua atau lebih sumber data) menggunakan *Google Maps* API . Beberapa tujuan dari penggunaan *Google Maps* API adalah untuk melihat lokasi, mencari alamat, mendapatkan petunjuk mengemudi dan lain sebagainya. Hampir semua hal yang berhubungan dengan peta dapat memanfaatkan *Google Maps*. *Google Maps* (tanpa API) diperkenalkan pada Februari 2005 dan merupakan revolusi bagaimana peta di dalam *web*, yaitu dengan membiarkan *user* untuk menarik peta sehingga dapat menavigasinya. Solusi peta ini pada saat itu masih baru dan membutuhkan server khusus. Beberapa saat setelahnya, ada yang berhasil melakukan *hacking Google Maps* untuk digunakan di dalam webnya sendiri. Hal ini membuat *Google Maps* mengambil kesimpulan bahwa mereka membutuhkan API dan pada Juni 2005, *Google Maps* API dirilis secara publik.

2.4.1 MapView

MapView merupakan kelas berupa tampilan yang digunakan untuk menampilkan peta dunia digital, yang mana data-datanya didapatkan dari layanan *Google Maps*. Tampilan ini dapat menangkap *touch gesture* sehingga dapat dilakukan penggeseran dan perbesaran pada peta tersebut. Tampilan peta dapat diubah-ubah ke dalam tiga mode tampilan, yaitu tampilan *Street View*, *Satellite View*, dan *Traffic View*.

2.4.2 MapActivity

MapActivity merupakan kelas abstrak yang digunakan untuk menampilkan sebuah *MapView*. Dalam pemrograman Android, untuk bisa membuat aplikasi yang mengakses layanan *Google Maps*, harus dilakukan pembuatan kelas yang merupakan kelas anak dari *MapActivity*. Di dalam kelas itu nantinya baru dilakukan pengkodean untuk menampilkan *MapView* dan melakukan pengaturan-pengaturan yang berkaitan.

2.4.3 ItemizedOverlay<Item> extends OverlayItem

Kelas ini merupakan kelas abstrak yang digunakan sebagai kelas dasar bagi sebuah *Overlay* yang terdiri atas daftar-daftar *OverlayItems*. Sesuai dengan namanya, kelas ini memungkinkan untuk mendaftarkan point of interest yang akan ditampilkan di peta. Kelas ini menangani beberapa hal seperti proses penggambaran penanda (*marker*) untuk setiap titik dan menangani pula proses untuk menindaklanjuti adanya tap yang dilakukan pengguna terhadap item.

2.4.4 GeoPoint

Kelas *GeoPoint* merupakan kelas yang merepresentasikan sepasang titik lokasi geografis yaitu lintang dan bujur. Titik tersebut disimpan sebagai angka integer dalam satuan mikroderajat. Konstruktor kelas ini adalah *GeoPoint(int latitudeE6, int longitudeE6)*.

2.4.5 Peta

Peta adalah gambar atau lukisan pada kertas dan sebagainya yang menunjukkan letak tanah. Peta bisa disajikan dalam berbagai cara yang berbeda, mulai dari peta konvensional yang tercetak hingga peta digital yang tampil di layar komputer. Istilah peta berasal dari bahasa Yunani *mappa* yang berarti taplak atau kain penutup meja. Namun secara umum pengertian peta adalah lembaran seluruh atau sebagian permukaan bumi pada bidang datar yang diperkecil dengan menggunakan skala tertentu. Sebuah peta adalah representasi dua dimensi dari

suatu ruang tiga dimensi. Banyak peta mempunyai skala, yang menentukan seberapa besar objek pada peta dalam keadaan yang sebenarnya.



Gambar 2.4 Ilustrasi peta pada Google Map

2.4.6 Peta Digital

Menurut definisi, peta digital adalah representasi fenomena geografik yang disimpan untuk ditampilkan dan dianalisis oleh komputer. Setiap objek pada peta digital disimpan sebagai sebuah atau sekumpulan koordinat. Sebagai contoh, 10 objek berupa lokasi sebuah titik akan disimpan sebagai sebuah koordinat, sedangkan objek berupa wilayah akan disimpan sebagai sekumpulan koordinat. Beberapa kelebihan penggunaan peta digital dibandingkan dengan peta analog (yang disimpan dalam bentuk kertas atau media cetakan lain), antara lain dalam hal :

- a. Peta digital kualitasnya tetap. Tidak seperti kertas yang dapat terlipat, memuai atau sobek ketika disimpan, peta digital dapat dikembalikan ke bentuk asalnya kapanpun tanpa ada penurunan kualitas.
- b. Peta digital mudah disimpan dan dipindahkan dari satu media penyimpanan yang satu ke media penyimpanan yang lain. Peta analog yang disimpan dalam bentuk gulungan-gulungan kertas misalnya, memerlukan ruangan yang lebih besar dibanding dengan jika peta tersebut disimpan sebagai peta digital dalam sebuah CD-ROM atau DVD-ROM.

Peta digital lebih mudah diperbaharui. Penyuntingan untuk keperluan perubahan data atau perubahan sistem koordinat misalnya, dapat lebih mudah dilakukan menggunakan perangkat lunak tertentu.

Adapun karakteristik peta digital adalah sebagai berikut:

a. Skala

Pada peta digital, skala menggambarkan tingkat kedetilan objek ketika peta tersebut dibuat. Sebagai contoh, pada peta skala 1:1.000 (1 cm di peta mewakili 1.000 cm atau 10 meter di permukaan bumi), maka objek gedung atau bangunan akan terlihat dengan jelas, sedangkan pada peta skala 1:100.000 (1 cm di peta mewakili 100.000 cm atau 1 km di permukaan bumi), sebuah bangunan hanya akan terlihat sebagai sebuah titik.

b. Referensi Geografik

Referensi geografik berupa parameter-parameter *elipsoida* referensi dan *datum*. Salah satu referensi yang umum digunakan (termasuk dalam 11 penentuan posisi menggunakan satelit GPS) adalah WGS 84 (*World Geodetic System*), yang direvisi pada tahun 1984 dan akan berlaku sampai tahun 2010.

c. Sistem Proyeksi Peta

Sistem proyeksi peta menentukan bagaimana objek-objek di permukaan bumi (yang sebenarnya tidak datar) dipindahkan atau diproyeksikan pada permukaan peta yang berupa bidang datar. Penggunaan sistem proyeksi peta yang berbeda untuk sebuah daerah yang sama, akan memberikan kenampakan yang berbeda.

d. Proyeksi Peta

Pada dasarnya bentuk bumi tidak datar tapi mendekati bulat maka untuk menggambarkan sebagian muka bumi untuk kepentingan pembuatan peta, perlu dilakukan langkah-langkah agar bentuk yang mendekati bulat tersebut dapat didatarkan dan distorsinya dapat terkontrol, untuk itu dilakukan proyeksi ke bidang datar. Penggunaan sistem proyeksi peta yang berbeda untuk sebuah daerah yang sama akan memberikan kenampakan yang berbeda.

2.4.7 Pemetaan

Pemetaan merupakan suatu proses pengukuran, perhitungan dan penggambaran dengan menggunakan cara atau metode tertentu sehingga didapatkan hasil berupa *softcopy* maupun *hardcopy* peta yang berbentuk data spasial vector maupun raster. Pemetaan juga dapat diartikan sebagai proses pembuatan peta. Tujuan utama pemetaan adalah untuk menyediakan deskripsi dari suatu fenomena geografis, informasi spasial dan non-spasial, informasi tentang jenis fitur, (titik, garis dan polygon) (Indarto, 2010).

2.5 Android Location API

2.5.1 Geocoder

Kelas *Geocoder* merupakan kelas yang dipergunakan untuk menangani *geocoding* dan *reverse geocoding*. *Geocoding* sendiri merupakan proses konversi dari data geografis (misalnya alamat atau kode pos) menjadi koordinat geografis dalam bentuk lintang dan bujur (misalnya -7.057006, 110.432798). Sementara *Reverse Geocoding* sendiri merupakan kebalikan dari *Geocoding*.

2.5.2 Location

Kelas *Location* merupakan kelas yang merepresentasikan lokasi geografis. Dalam kelas ini terdapat *method-method* yang berguna yang berhubungan dengan lokasi. Untuk mencari jarak dapat digunakan *method distanceTo (Location dest)* dengan memberikan parameter lokasi tujuan.

2.5.3 LocationManager

Kelas ini memberikan akses pada layanan lokasi sistem. Hal ini memungkinkan aplikasi untuk mendapatkan pembaharuan dari lokasi geografis perangkat Android. Pada kelas ini, terdapat atribut-atribut yang berguna dalam proses pembaharuan lokasi geografis perangkat. Dua di antaranya adalah *GPS_PROVIDER* dan *NETWORK_PROVIDER*. Kedua atribut tersebut bertipe *String*.

2.5.4 *LocationListener*

LocationListener merupakan antarmuka yang digunakan untuk menerima pemberitahuan dari *LocationManager* ketika lokasi perangkat Android Berubah *LocationListener* dipanggil dari method *requestLocationUpdates (String provider, long minTime, float minDistance, LocationListener listener)* pada *LocationManager*.

2.6 GIS

GIS (Geographical Information System) adalah sistem berbasis komputer yang digunakan untuk memasukan, menyimpan, mengelola, menganalisis dan mengaktifkan kembali data yang mempunyai referensi keruangan untuk berbagai tujuan yang berkaitan dengan pemetaan dan perencanaan. Dalam suatu sistem informasi geografis, terdapat beberapa komponen utama yang saling berintegrasi dan saling terkait, yaitu :

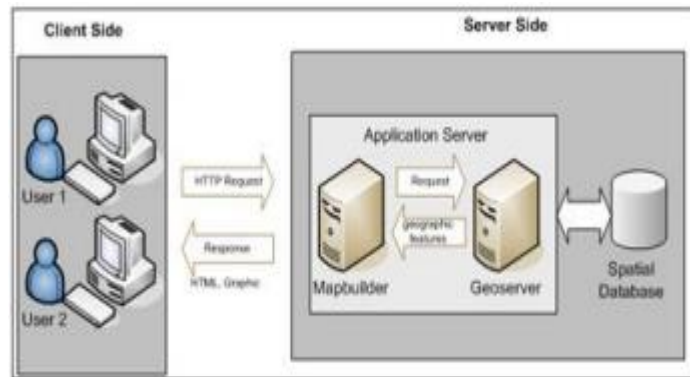
- a. Sistem komputer (*Hardware* dan *Software*)
- b. Data Geospasial.
- c. *User* atau pengguna.



Gambar 2.5 Komponen GIS

Sistem Informasi Geografis sendiri telah dikembangkan ke arah aplikasi berbasis web, atau dikenal dengan WebGIS. Tujuannya untuk memudahkan user

atau pengguna dalam mendapatkan informasi geografis terkini pada suatu kota atau wilayah tertentu, dan tidak terbatas pada lokasi dari user tersebut.



Gambar 2.6 Arsitektur *Web Based GIS*

2.7 *Global Positioning System (GPS)*

Gps atau disebut juga *Global Positioning System* adalah sistem navigasi berbasis satelit yang dikembangkan oleh departemen pertahanan amerika. GPS terdiri dari 3 segmen :

- Segmen Angkasa
- Segmen *Control/Pengendali*
- Segmen Pengguna

Dimana segmen angkasa terdiri dari 24 satelit yang beroperasi dalam 6 orbit pada ketinggian 20.200 km dan inklinasi 55 derajat dengan periode 12 jam (satelit akan kembali ke titik yang sama dalam 12 jam). Satelit tersebut memutari orbitnya sehingga minimal ada 6 satelit yang dapat dipantau pada titik manapun di bumi ini. Satelit tersebut mengirimkan posisi dan waktu kepada pengguna seluruh dunia.

Setiap satelit mentransmisikan dua sinyal yaitu L1 (1575.42 MHz) dan L2 (1227.60 MHz). Sinyal L1 dimodulasikan dengan dua sinyal pseudo-random yaitu kode P (*Protected*) dan kode C/A (*coarse/aquisition*). Sinyal L2 hanya membawa kode P. Setiap satelit mentransmisikan kode yang unik sehingga penerima (perangkat GPS) dapat mengidentifikasi sinyal dari setiap satelit. Pada saat fitur "*Anti-Spoofing*" diaktifkan, maka kode P akan dienkripsi dan selanjutnya dikenal sebagai kode P(Y) atau kode Y.

Perangkat GPS yang dikhususkan buat sipil hanya menerima kode C/A pada sinyal L1 (meskipun pada perangkat GPS yang canggih dapat memanfaatkan sinyal L2 untuk memperoleh pengukuran yang lebih teliti). Perangkat GPS menerima sinyal yang ditransmisikan oleh satelit GPS. Dalam menentukan posisi, kita membutuhkan paling sedikit 3 satelit untuk penentuan posisi 2 dimensi (lintang dan bujur) dan 4 satelit untuk penentuan posisi 3 dimensi (lintang, bujur, dan ketinggian). Semakin banyak satelit yang diperoleh maka akurasi posisi kita akan semakin tinggi. Untuk mendapatkan sinyal tersebut, perangkat GPS harus berada di ruang terbuka. Apabila perangkat GPS kita berada dalam ruangan atau kanopi yang lebat dan daerah kita dikelilingi oleh gedung tinggi maka sinyal yang diperoleh akan semakin berkurang sehingga akan sukar untuk menentukan posisi dengan tepat atau bahkan tidak dapat menentukan posisi. Melalui GPS kita dapat mengetahui keberadaan suatu objek di mana pun objek itu berada di seluruh muka bumi baik di darat, laut maupun udara.

2.8 Quarry

Quarry adalah system tambang terbuka yang diterapkan untuk menambang endapan-endapan bahan galian industri atau mineral industri, antara lain: penambangan batu gamping, marmer, granit, andesit dan sebagainya. Quarry dapat menghasilkan material atau hasil tambang dalam bentuk loose/broken materials ataupun dalam bentuk dimensional stones.

Dimensi batuan yang diproduksi pada sistem penambangan quarry, pada umumnya adalah mineral yang berbentuk prismatic pendek atau balok-balok yang memiliki ukuran dan bentuk yang kasar. Quarry pada dasarnya sama dengan open pits, namun yang membedakannya adalah material yang ditambang. Open pit pada dasarnya merupakan tambang terbuka yang menambang mineral logam. Sedangkan quarry pada dasarnya merupakan sistem penambangan terbuka yang menambang mineral non logam atau batuan, contoh material yang biasanya ditambang pada quarry yaitu : marmer, batu granit, dan masih banyak lagi yang lainnya.

Produk yang dihasilkan pada system quarry pada umumnya merupakan dimensi batuan nonlogam (Barton, 1968). Pada umumnya, dimensi batuan granit, marmer, batu gamping, batu pasir, batu ubin besar, dan slate yang diperkirakan semakin lama semakin turun atau semakin susah untuk dipotong. Karena kesulitan atau kendala dan biaya yang berasosiasi dengan proses pemotongan batuan, quarry pada umumnya lebih mahal dibandingkan dengan metode lain di tambang terbuka, dengan square set stoping, merupakan biaya terbesar dalam penambangan. Quarry juga memiliki selektifitas yang tinggi, metode dalam skala kecil, dengan produktifitas yang rendah.

2.9 Definisi Material Konstruksi

Material Konstruksi meliputi seluruh bahan yang dibutuhkan untuk menyelesaikan bagian pekerjaan dalam satu kesatuan pekerjaan pada suatu proses konstruksi. Sistem Pengelolaan Material Konstruksi adalah suatu sistem yang merencanakan dan mengendalikan seluruh kegiatan untuk menjamin agar material dan peralatan konstruksi dapat diperoleh dalam jumlah yang tepat, sesuai dengan spesifikasi, dengan harga yang pantas dan tersedia pada saat dibutuhkan.

2.10 Definisi RAB

Rencana Anggaran Biaya adalah suatu bangunan atau proyek adalah perhitungan banyaknya biaya yang diperlukan untuk bahan dan upah, serta biaya-biaya lain yang berhubungan dengan pelaksanaan bangunan atau proyek. Anggaran biaya merupakan harga dari bahan bangunan yang dihitung dengan teliti, cermat dan memenuhi syarat. Anggaran biaya pada bangunan yang sama akan berbeda- beda di masing- masing daerah, disebabkan karena perbedaan harga bahan dan upah tenaga kerja.

2.11 Software yang Digunakan

2.11.1 Java

Java menurut definisi dari Sun adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer standalone

ataupun pada lingkungan jaringan. Java2 adalah generasi kedua dari Java platform (generasi pertamanya adalah Java Development Kit). Platform Java memiliki tiga buah edisi yang berbeda, yaitu J2EE (*Java2 Enterprise Edition*), J2ME (*Java2 Micro Edition*), dan J2SE (*Java2 Standart Edition*).

a. *Java2 Standart Edition (J2SE)*

J2SE adalah inti dari bahasa pemrograman Java. JDK (*Java Development Kit*) adalah salah satu tool dari J2SE untuk mengompilasi dan menjalankan program Java.

b. *Java2 Micro Edition (J2ME)*

J2ME adalah lingkungan pengembangan yang didesain untuk meletakkan perangkat lunak Java pada barang elektronik beserta perangkat pendukungnya. J2ME biasa digunakan pada telepon seluler, pager, *personal digital assistants* (PDA's) dan sejenisnya.

c. *Java2 Enterprise Edition (J2EE)*

J2EE adalah kelompok dari beberapa API dari Java dan teknologi selain Java. J2EE dibuat untuk membuat aplikasi yang kompleks. J2EE sering dianggap sebagai *middleware* atau teknologi yang berjalan di server, namun sebenarnya J2EE tidak hanya sebatas untuk itu.

2.11.2 Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform (*platform-independent*). Berikut ini adalah sifat dari Eclipse:

- a. *Multi-Platform* : Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
- b. *Multi-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.

- c. *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.

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.

2.11.3 Sejarah Eclipse

Eclipse awalnya dikembangkan oleh IBM untuk menggantikan perangkat lunak IBM Visual Age for Java 4.0. Produk ini diluncurkan oleh IBM pada tanggal 5 November 2001, yang menginvestasikan sebanyak US\$ 40 juta untuk pengembangannya. Semenjak itu konsorsium Eclipse Foundation mengambil alih untuk pengembangan Eclipse lebih lanjut dan pengaturan organisasinya.

2.11.4 Arsitektur Eclipse

Sejak versi 3.0, Eclipse pada dasarnya merupakan sebuah kernel, yang mengangkat plug-in. Apa yang dapat digunakan di dalam Eclipse sebenarnya adalah fungsi dari plug-in yang sudah diinstal. Ini merupakan basis dari Eclipse yang dinamakan Rich Client Platform (RCP). Berikut ini adalah komponen yang membentuk RCP: Core Platform, OSGi, SWT (Standard Widget Toolkit), JFace, Eclipse Workbench.

Secara standar Eclipse selalu dilengkapi dengan JDT (Java Development Tools), plug-in yang membuat Eclipse kompatibel untuk mengembangkan program Java, dan PDE (Plug-in Development Environment) untuk mengembangkan plug-in baru. Eclipse beserta plug-in-nya diimplementasikan dalam bahasa pemrograman Java.

Konsep Eclipse adalah IDE yang terbuka (*open*), mudah diperluas (*extensible*) untuk apa saja, dan tidak untuk sesuatu yang spesifik. Jadi, Eclipse tidak saja

untuk mengembangkan program Java, akan tetapi dapat digunakan untuk berbagai macam keperluan, cukup dengan menginstal plug-in yang dibutuhkan. Apabila ingin mengembangkan program C/C++ terdapat plug-in CDT (C/C++ Development Tools). Selain itu, pengembangan secara visual bukan hal yang tidak mungkin oleh Eclipse, plug-in UML2 tersedia untuk membuat diagram UML. Dengan menggunakan PDE setiap orang bisa membuat plug-in sesuai dengan keinginannya. Salah satu situs yang menawarkan plug-in secara gratis seperti Eclipse downloads by project.

2.12 Adobe Photoshop CS3

Adobe Photoshop CS3 merupakan salah satu *software* pengolah gambar untuk gambar *bitmap* keluaran dari Adobe System Inc. Sebagai salah satu *software* yang sangat handal, *photoshop* telah banyak digunakan oleh para *professional* untuk membuat dan memanipulasi gambar, baik untuk keperluan media percetakan, internet, maupun elektronik.

2.13 The Unified Modeling Language

The Unified Modeling Language merupakan seperangkat aturan dan notasi untuk spesifikasi sistem perangkat lunak, dikelola dan dibuat oleh *Object Management Group* yang digunakan untuk memodelkan sebuah perangkat lunak.

UML digunakan untuk memvisualisasikan, membangun, dan mendokumentasikan rancangan sistem dari sebuah perangkat lunak. Diagram UML antara lain terdiri dari *Use Case Diagram*, *Activity Diagram*, *Class Diagram*, dan *Sequence Diagram*.

2.14 Stakeholder

Istilah *stakeholder* telah dipakai oleh banyak pihak dalam hubungannya dengan berbagai ilmu atau konteks, misalnya manajemen bisnis, ilmu komunikasi, pengelolaan sumberdaya alam, sosiologi, dan lain-lain. Lembaga-lembaga publik telah menggunakan secara luas istilah *stakeholder* ini ke dalam proses-proses pengambilan dan implementasi keputusan.

Menurut Freeman (Bryson, 2001), *stakeholder* dalam lingkup bisnis merupakan kelompok atau individu yang dipengaruhi dan mempengaruhi masa depan perusahaan yaitu pelanggan, pekerja, pemilik, pemerintah, lembaga keuangan dan kritikus. Sedangkan dalam konteks organisasi, baik di pemerintahan maupun swasta. Bryson (2001), mendefinisikan *stakeholder* sebagai individu, kelompok atau organisasi apapun yang dapat melakukan klaim atau perhatian terhadap sumber daya atau hasil organisasi atau dipengaruhi oleh hasil itu.

a. Identifikasi *Stakeholder*

Tujuan kegiatan ini adalah mengidentifikasi dan mengkategorisasi stakeholder baik yang berkaitan dengan dampak, manfaat dan kepentingan. Identifikasi dilakukan dengan memetakan masalah dan lokasi sehingga ditemukan masyarakat sebagai stakeholder utama, kemudian mengaitkan masalah baik dari segi kepentingan, pemihakan dan kewenangan yang dimiliki oleh aktor-aktor baik pemerintah maupun non pemerintah.

b. Pemetaan Masalah dan *Stakeholder*

Pemetaan ini merupakan pendalaman dari kegiatan identifikasi *stakeholder*, yang dilakukan melalui wawancara dan diskusi terfokus. Informasi yang dikumpulkan menyangkut isu yang relevan dengan kebijakan, program, dan proyek, pemahaman *stakeholder*, sikap, alasan dan kepentingan mereka, jaringan mereka, posisi dan kekuatan pengaruh serta usulan-usulan mereka.

2.15 WBS (*Work Breakdown Structure*)

WBS adalah suatu metode pengorganisaian proyek menjadi struktur pelaporan hierarakis. WBS digunakan untuk melakukan *Breakdown* atau memecahkan tiap proses pekerjaan menjadi lebih detail. hal ini dimaksudkan agar proses perencanaan proyek memiliki tingkat yang lebih baik.

WBS disusun berdasarkan dasar pembelajaran seluruh dokumen proyek yang meliputi kontrak, gambar-gambar, dan spesifikasi. Proyek kemudian diuraikan menjadi bagian-bagian dengan mengikuti pola struktur dan hirarki tertentu menjadi item-item pekerjaan yang cukup terperinci, yang disebut sebagai *Work Breakdown Structure*.

Pada prinsipnya *Work Breakdown Structure* (WBS) adalah pemecahan atau pembagian pekerjaan ke dalam bagian yang lebih kecil (sub-kegiatan), alasan perlunya WBS adalah :

- a. Pengembangan WBS di awal *Project Life Cycle* memungkinkan diperolehnya pengertian cakupan proyek dengan jelas, dan proses pengembangan WBS ini membantu semua anggota untuk lebih mengerti tentang proyek selama tahap awal.
- b. WBS membantu dalam pengawasan dan peramalan biaya, jadwal, dan informasi mengenai produktifitas yang meyakinkan anggota manajemen proyek sebagai dasar untuk membuat perundingan.

2.16 MS Project 2007

Pengertian Microsoft Project 2007 (atau MSP atau WinProj) adalah suatu manajemen proyek perangkat lunak program yang dikembangkan dan dijual oleh Microsoft yang dirancang untuk membantu manajer proyek dalam mengembangkan rencana, menetapkan sumber daya untuk tugas-tugas, pelacakan kemajuan, mengelola anggaran dan menganalisis beban kerja.

2.17 Rational Rose

Rational Rose adalah software yang memiliki perangkat-perangkat pemodelan secara visual untuk membangun suatu solusi dalam rekayasa *software* dan pemodelan bisnis. *Rational Rose* dikeluarkan oleh perusahaan *software* bernama *Rational Software*, perusahaan yang mencetuskan ide pembentukan konsorsium bagi perusahaan-perusahaan yang memakai standar UML sebagai bahasa pemodelan di perusahaannya. *Rational Rose* memakai UML sebagai bahasa pemodelannya, ditambah beberapa fitur lain yang membuat *Rational Rose* menjadi software pemodelan visual yang terkemuka.

Beberapa fitur terkemuka diantaranya *Rational Rose* memiliki *Rational Unified Process* (RUP) yaitu proses yang paling terperinci yang ada saat ini dan akan memberikan pedoman secara umum dalam pembuatan *software* dan pemodelan bisnis. Selain itu, *Rational Rose* memiliki kemampuan membuat solusi

client/server, yang kemudian dapat diterapkan dan didistribusikan dalam lingkungan perusahaan. Keunggulan *Rational Rose* diantaranya:

- a. Bahasa yang digunakan adalah bahasa pemodelan standar yaitu UML, akan meningkatkan komunikasi intra tim.
- b. *Rational Rose* mendukung *round-trip engineering* sehingga kita dapat *generate* model kedalam kode (Java, C++, Visual Basic, dan sebagainya) dan melakukan *reverse engineering* untuk menampilkan arsitektur *software* dari kode yang ada. Hal ini dapat dilakukan secara bolak-balik sebagai proses *iterative* selama proses rekayasa *software*.
- c. Model dan kode senantiasa sinkron selama dalam *development cycle*.
- d. Membangun *software* menggunakan *Rational Rose* memudahkan dalam memperbaiki *software* tersebut karena apabila suatu saat ditemukan *requirement* baru, kita dapat menggambarkan lagi *software* tersebut dalam UML.
- e. Para *user Rational Rose* dapat berkomunikasi walaupun bekerja dalam sistem operasi yang berbeda (Windows atau UNIX).
- f. Dengan menggunakan *Rose Web Publisher* suatu tim dapat mengkomunikasikan model dan spesifikasinya dalam *web browser*.
- g. Mendukung rekayasa *software* untuk sistem *client/server* sehingga *Rational Rose* merupakan *software* pemodelan visual yang tangguh dalam lingkungan *client/server*, e-business, dan lingkungan perusahaan terdistribusi (kantor-kantor terletak dalam tempat yang berbeda-beda).

2.18 Analisa dan Perancangan Sistem

Dalam perancangan sistem berorientasi objek, diagram-diagram UML yang digunakan adalah mengikuti standar UML dan untuk perancangan basis datanya dapat menggunakan basis data relational dengan pendekatan model data ERD. Diagram-diagram tersebut dijelaskan dibawah ini.

a. *Use Case Diagram*

Use Case menurut Martin Fowler (2005 : 141) adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use Case* mendeskripsikan interaksi

tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan. *Use Case Diagram* menampilkan aktor mana yang menggunakan *use case* mana, *uses case* mana yang memasukkan *use case* lain dan hubungan antara aktor dan *use case*. Berikut merupakan komponen-komponen dari *Use Case Diagram*:

a. *Actor*

Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.



Simbol *Actor*

b. *Dependency*

Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).



Simbol *Dependency*

c. *Generalization*

Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).



Simbol *Generalization*

d. *Include*

Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.



Simbol *Include*

e. *Extend*

Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.



Simbol *Extend*

f. *Association*

Apa yang menghubungkan antara objek satu dengan objek lainnya.



Simbol *Association*

g. *System*

Menspesifikasikan paket yang menampilkan sistem secara terbatas.



Simbol *System*

h. *Use Case*

Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.



Simbol *Use Case*

i. *Collaboration*

Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).



Simbol *Collaboration*

j. *Note*

Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.



Simbol *Note*

b. Activity Diagram

Activity diagram menurut Martin Fowler (2005 : 163) adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. Dalam beberapa hal, *activity diagram* memainkan peran mirip diagram alir, tetapi perbedaan prinsip antara notasi diagram alir adalah *activity diagram* mendukung *behavior paralel*. Node pada sebuah *activity diagram* disebut sebagai *action*, sehingga diagram tersebut menampilkan sebuah *activity* yang tersusun dari *action*. Berikut merupakan komponen-komponen dari *Activity Diagram*:

a. *Initial*

Titik awal, untuk memulai suatu aktivitas.



Simbol *Initial*

b. *Final*

Titik akhir, untuk mengakhiri aktivitas.



Simbol *Final*

c. *Activity*

Menandakan sebuah aktivitas.



Simbol *Activity*

d. *Decision*

Pilihan untuk mengambil keputusan.



Simbol *Decision*

e. *Fork/Join*

Digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel.



Simbol *Fork/Join*

f. *Rake*

Menunjukkan adanya dekomposisi.



Simbol *Rake*

g. *Receive*

Tanda penerimaan.



Simbol *Receive*

c. *Sequence Diagram*

Diagram interaksi yang menekankan pada waktu pengiriman *message*. *Sequence diagram* menunjukkan sekumpulan objek dan pengiriman serta penerimaan *message* antar objek. Objek yang umumnya memiliki nama atau instansiasi dari *class*, tapi dapat pula merupakan turunan dari *things* lain, seperti *collaboration*, *component* dan *node*. Diagram ini digunakan untuk mengilustrasikan *dynamic view* dari system. Berikut merupakan komponen-komponen dari *Sequence Diagram*:

a. *Actor*

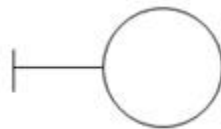
Menggambarkan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem.



Simbol *Actor*

b. *Boundary*

Mengambarkan interaksi antara satu atau lebih *actor* dengan sistem, memodelkan bagian dari sistem yang bergantung pada pihak lain disekitarnya dan merupakan pembatas sistem dengan dunia luar.



Simbol *Boundary*

c. *Control*

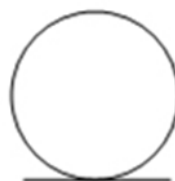
Mengambarkan “perilaku mengatur”, mengkoordinasikan perilaku sistem dan dinamika dari suatu sistem, menangani tugas utama dan mengontrol alur kerja suatu sistem.



Simbol *Control*

d. *Entity*

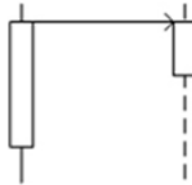
Mengambarkan informasi yang harus disimpan oleh sistem (struktur data dari sebuah sistem).



Simbol *Entity*

e. *Object Message*

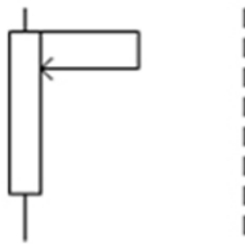
Menggambarkan pesan/hubungan antar obyek yang menunjukkan urutan kejadian yang terjadi.



Simbol *object message*

f. *Message to self*

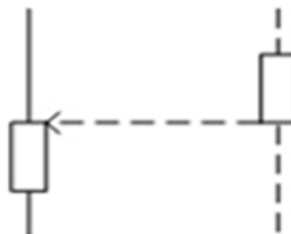
Menggambarkan pesan/hubungan obyek itu sendiri, yang menunjukkan urutan kejadian yang terjadi.



Simbol *Message to Self*

g. *Return Message*

Menggambarkan pesan/hubungan antar obyek, yang menunjukkan urutan kejadian yang terjadi.



Simbol *Return Message*

h. *Lifeline*

Eksekusi obyek selama sequence (message dikirim atau diterima dan aktifasinya).



Simbol *Lifeline*

d. Analisa Masukan Sistem Berjalan

Analisa dokumen keluaran merupakan analisa dokumen yang dihasilkan oleh sistem yang berjalan.

e. Analisa Keluaran Sistem Berjalan

Analisa dokumen masukan merupakan analisa dokumen yang dipakai sebagai sumber data ataupun dokumen yang digunakan sebagai masukan dalam sistem.

f. Rancangan Dokumen Masukan

Rancangan Dokumen Masukan merupakan informasi yang akan dimasukkan dari sistem yang akan dirancang.

g. Rancangan Dokumen Keluaran

Rancangan Dokumen Keluaran merupakan informasi yang akan dihasilkan dari masukan sistem yang akan dirancang.

h. Rancangan Layar

Rancangan layar aplikasi merupakan sebuah desain tampilan yang berfungsi sebagai antar muka antara si pengguna dengan aplikasi, atau bisa disebut dengan *user interface*. Rancangan layar dibuat berdasarkan dengan mempertimbangkan dari sudut pandang *user* atau kebutuhan *user*.

2.19 Implementasi

Implementasi diartikan sebagai pelaksanaan atau penerapan. Artinya yang dilaksanakan dan diterapkan adalah aplikasi yang telah dirancang atau didisain untuk kemudian dijalankan sepenuhnya. Maka, implementasi aplikasi juga dituntut untuk melaksanakan sepenuhnya apa yang telah direncanakan, permasalahan besar yang akan terjadi apabila yang dilaksanakan bertolak belakang atau menyimpang dari yang telah dirancang maka terjadilah kesia-siaan antara rancangan dengan implementasi. Implementasi adalah suatu tindakan atau pelaksanaan dari sebuah rencana yang sudah disusun secara matang dan terperinci. Implementasi biasanya dilakukan setelah perencanaan sudah dianggap sempurna.

BAB III

PEMODELAN PROYEK

3.1 Objective Project

Objektifitas proyek yaitu hal-hal yang berisi tentang tujuan pelaksanaan proyek, hal-hal yang melandasi terciptanya sebuah proyek serta manfaat dan tujuan proyek secara rinci. Berikut ini merupakan *Objective Project* aplikasi yang dibuat sebagai berikut:

- a. Membuat Aplikasi Pemetaan lokasi *quarry* agar mempermudah pencarian lokasi *quarry*.
- b. Memberi pengenalan kepada perusahaan terhadap teknologi yang dapat membantu pekerjaan perusahaan.
- c. Aplikasi yang dibuat dapat digunakan kapanpun dan dimanapun jika ada internet.
- d. Merancang aplikasi sehingga aplikasi dapat dipahami dan digunakan dengan mudah tanpa harus mempelajarinya terlebih dahulu.
- e. Implementasi aplikasi yang sudah dirancang dan dibuat.

3.2 Identifikasi Stakeholder

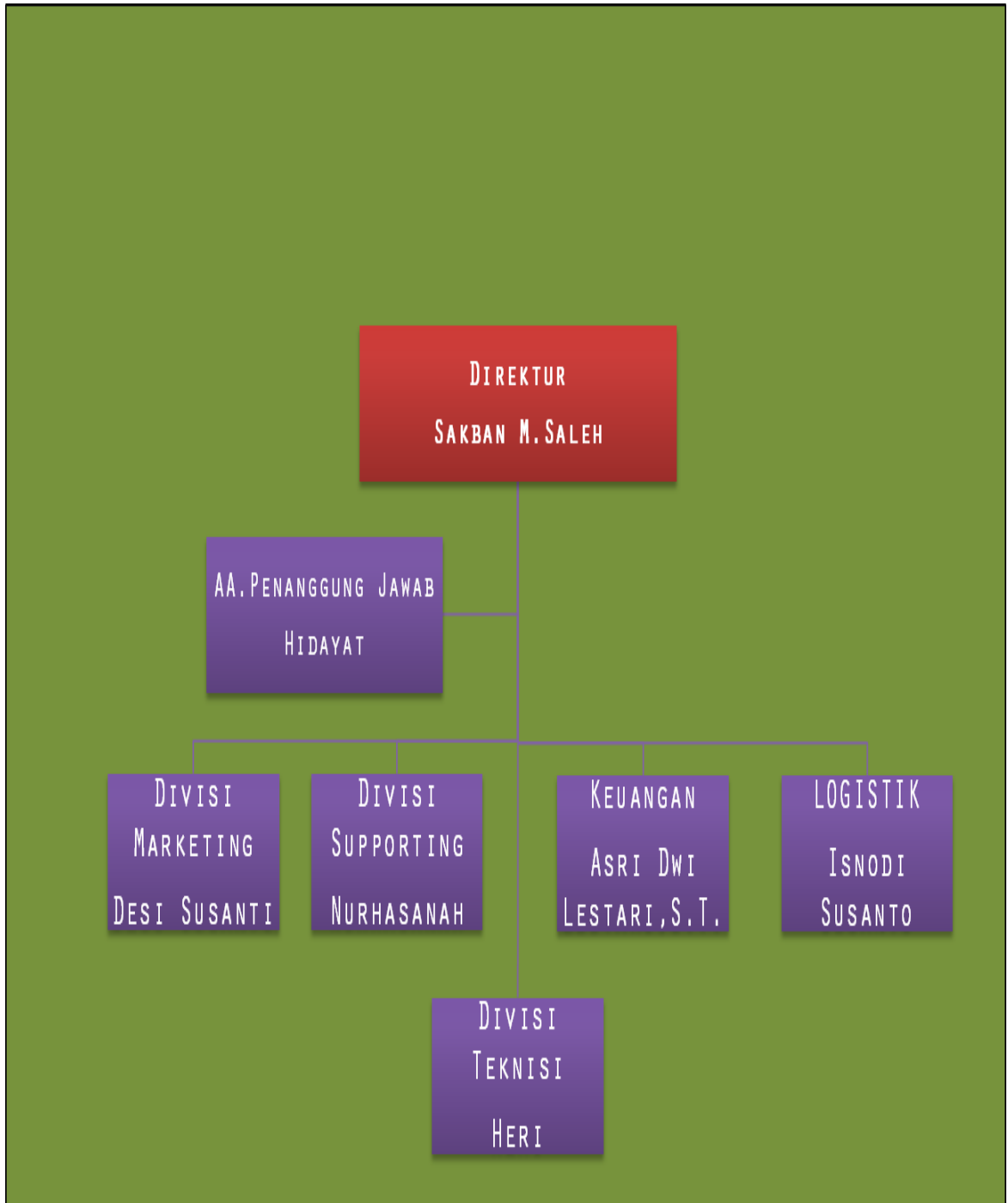
Kondisi instansi tempat penelitian dilakukan yaitu pada perusahaan PT.INTI SAKTI UTAMA yang terletak di Pangkalpinang adalah sebagai berikut:

3.2.1 Profil Perusahaan

Perusahaan PT.INTI SAKTI UTAMA yang menjadi tempat penelitian beralamat di Jl. Sumedang Kec.Taman Sari Pangkalpinang. Perusahaan ini berdiri sejak tahun 2014 dengan bergerak di bidang Perdagangan dan Jasa Konstruksi.

3.2.2 Struktur Organisasi

Berikut ini adalah Struktur Organisasi Perusahaan PT.INSUTAMA:



Gambar 3.1 Struktur Organisasi Perusahaan

3.2.3 Stakeholder

Tabel 3.1 Stakeholder

| DAFTAR STAKEHOLDER | | | | |
|--------------------|---------------------|-----------------------|--|--------------|
| No | Nama/Instansi | Posisi | Tugas/Penanggung Jawab | Kontak |
| 1 | PT.INTI SAKTI UTAMA | Leader Proyek | -Memfasilitasi proyek agar berjalan sebagaimana mestinya | - |
| 1. | Izet Aulia Rahmat | Manajer Proyek | -Menjadwalkan Proyek -Mengelola Sumber Daya -Mengkomunikasikan Proyek -Mengontrol jalannya proyek | 087802979619 |
| 2. | Izet Aulia Rahmat | <i>System Analist</i> | Mengumpulkan, Menyusun, Merancang, Menganalisis dan Mengawasi semua kegiatan dalam sistem yang baru | 087802979619 |
| 3. | Izet Aulia Rahmat | Admin Proyek | Bertanggung jawab mempersiapkan semua dokumentasi proyek | 087802979619 |
| 4. | Izet Aulia Rahmat | <i>Programmer</i> | Membuat, menuliskan kedalam | 087802979619 |

| | | | | |
|----|-------------------|-----------------|--|--------------|
| | | | bahasa pemrograman, yang meliputi tahap pengkodean program | |
| 5. | Izet Aulia Rahmat | <i>Designer</i> | Merekayasa atau membuat desain untuk tampilan aplikasi | 087802979619 |

3.3 Identifikasi *Deliverables*

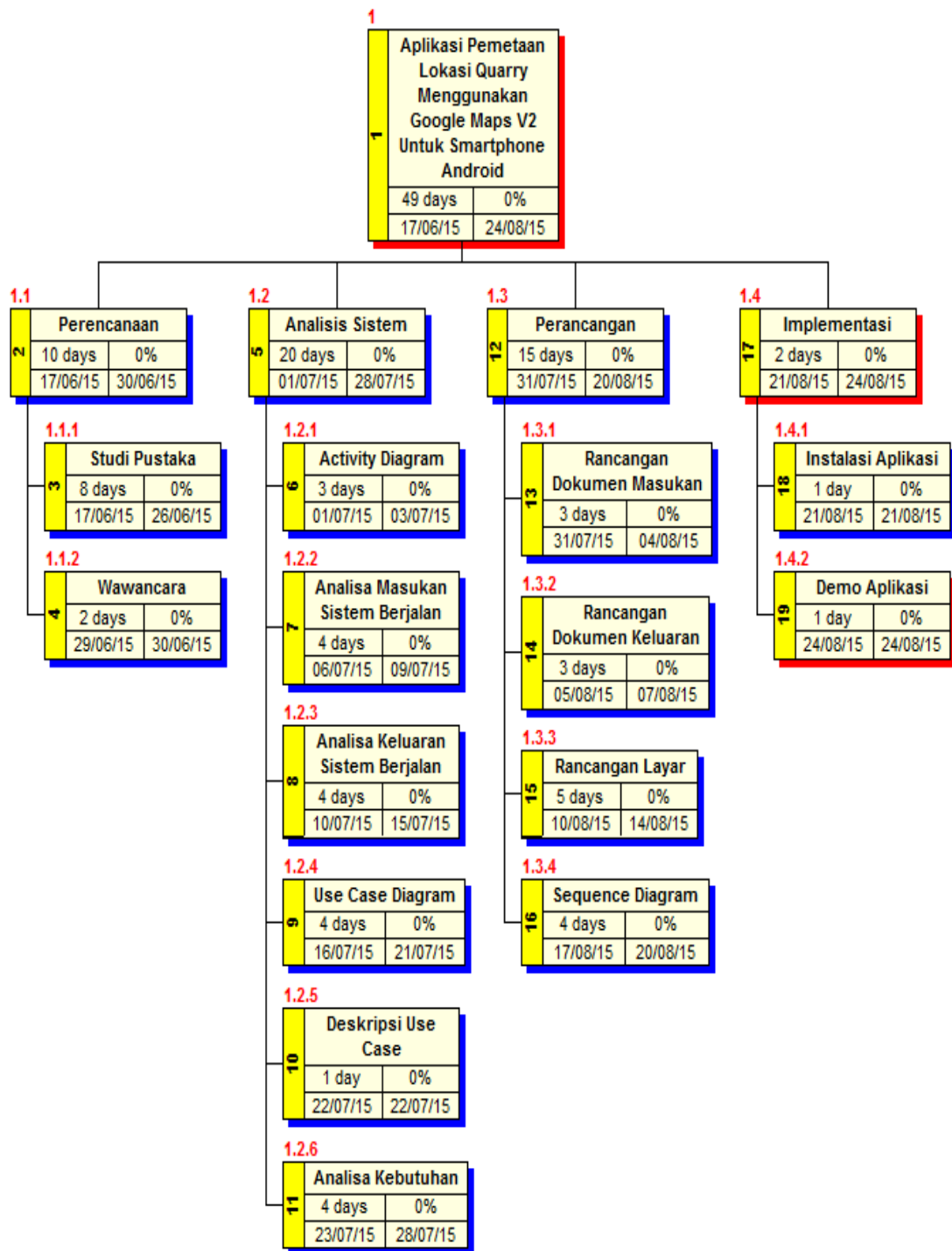
Deliverables diartikan sebagai hasil kerja. Dalam hal ini dapat berarti Identifikasi /perkiraan dari hasil pekerjaan atau hasil proyek bersangkutan seperti produk / barang dan jasa yang dihasilkan dari proyek. Hasil pekerjaan proyek dapat berupa materi yang dapat dihitung (*tangible*) ataupun berupa hasil yang tidak dapat dihitung (*Intangible*) seperti Kemasyhuran, kebanggaan dan lain sebagainya. Berikut merupakan *deliverables* pada penelitian yang dilakukan:

1. Tersedianya Aplikasi Pemetaan Lokasi Quarry Menggunakan Google Maps Api V2 berbasis Android.
2. Aplikasi ditujukan untuk pihak yang ingin mencari lokasi quarry.
3. Aplikasi yang dibuat dapat mempermudah pekerjaan penggunanya dalam melakukan pencarian maupun rute ke suatu lokasi.
4. Aplikasi yang dibuat dapat dikembangkan menjadi lebih baik dan lengkap oleh pengembang aplikasi kedepannya.
5. Aplikasi yang diimplementasikan pada perusahaan terkait adalah aplikasi android yang pastinya bisa dipakai oleh perusahaan maupun perorangan lainnya.

3.4 Penjadwalan Proyek

Penjadwalan pada proyek ini meliputi: *Work Breakdown Structure* (WBS), *Milestone* proyek, dan Jadwal Proyek.

3.4.1 Work Breakdown Structure



Gambar 3.2 Work Breakdown Structure

3.4.2 Milestone

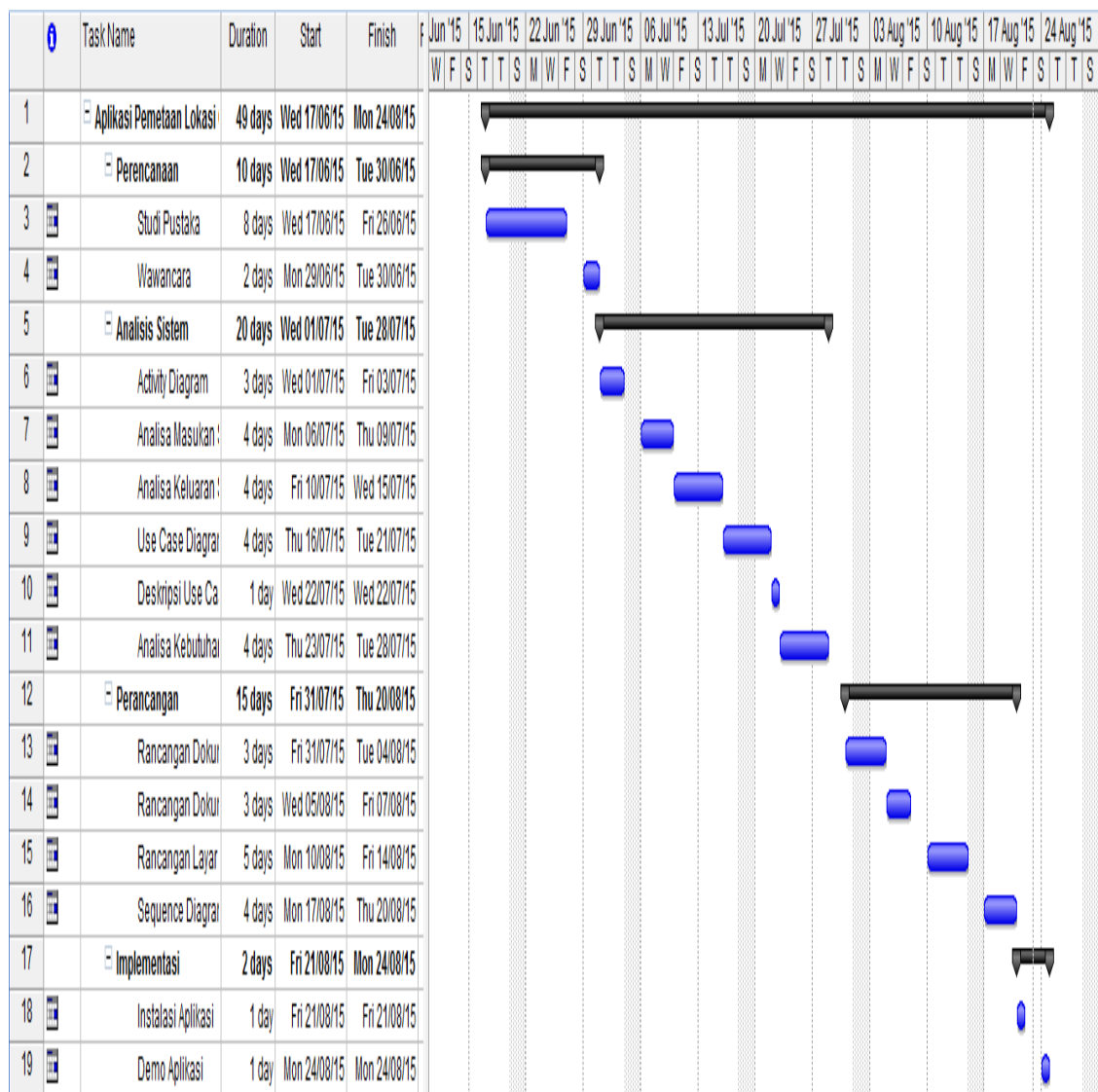
Milestone digunakan untuk membagi pekerjaan menjadi lebih kecil sehingga mudah untuk dimonitoring dan dievaluasi, *milestone* bisa dikatakan sebagai penjabaran dari WBS. Berikut *milestone* dari proyek Aplikasi Pemetaan ini:

| | | | |
|---|---------|--------------|--------------|
| <input type="checkbox"/> Aplikasi Pemetaan Lokasi | 49 days | Wed 17/06/15 | Mon 24/08/15 |
| <input type="checkbox"/> Perencanaan | 10 days | Wed 17/06/15 | Tue 30/06/15 |
| Studi Pustaka | 8 days | Wed 17/06/15 | Fri 26/06/15 |
| Wawancara | 2 days | Mon 29/06/15 | Tue 30/06/15 |
| <input type="checkbox"/> Analisis Sistem | 20 days | Wed 01/07/15 | Tue 28/07/15 |
| Activity Diagram | 3 days | Wed 01/07/15 | Fri 03/07/15 |
| Analisa Masukan : | 4 days | Mon 06/07/15 | Thu 09/07/15 |
| Analisa Keluaran : | 4 days | Fri 10/07/15 | Wed 15/07/15 |
| Use Case Diagram | 4 days | Thu 16/07/15 | Tue 21/07/15 |
| Deskripsi Use Case | 1 day | Wed 22/07/15 | Wed 22/07/15 |
| Analisa Kebutuhan | 4 days | Thu 23/07/15 | Tue 28/07/15 |
| <input type="checkbox"/> Perancangan | 15 days | Fri 31/07/15 | Thu 20/08/15 |
| Rancangan Doku | 3 days | Fri 31/07/15 | Tue 04/08/15 |
| Rancangan Doku | 3 days | Wed 05/08/15 | Fri 07/08/15 |
| Rancangan Layar | 5 days | Mon 10/08/15 | Fri 14/08/15 |
| Sequence Diagram | 4 days | Mon 17/08/15 | Thu 20/08/15 |
| <input type="checkbox"/> Implementasi | 2 days | Fri 21/08/15 | Mon 24/08/15 |
| Instalasi Aplikasi | 1 day | Fri 21/08/15 | Fri 21/08/15 |
| Demo Aplikasi | 1 day | Mon 24/08/15 | Mon 24/08/15 |

Gambar 3.3 *Milestone*

3.4.3 Jadwal Proyek

Proyek dilaksanakan mulai dari 17 Juni 2015 sampai dengan 24 Agustus 2015. Berikut jadwal proyek yang disusun dalam bentuk *Gantt Chart*:



Gambar 3.4 *Gantt Chart* Jadwal Proyek

3.5 RAB (Rencana Anggaran Biaya)

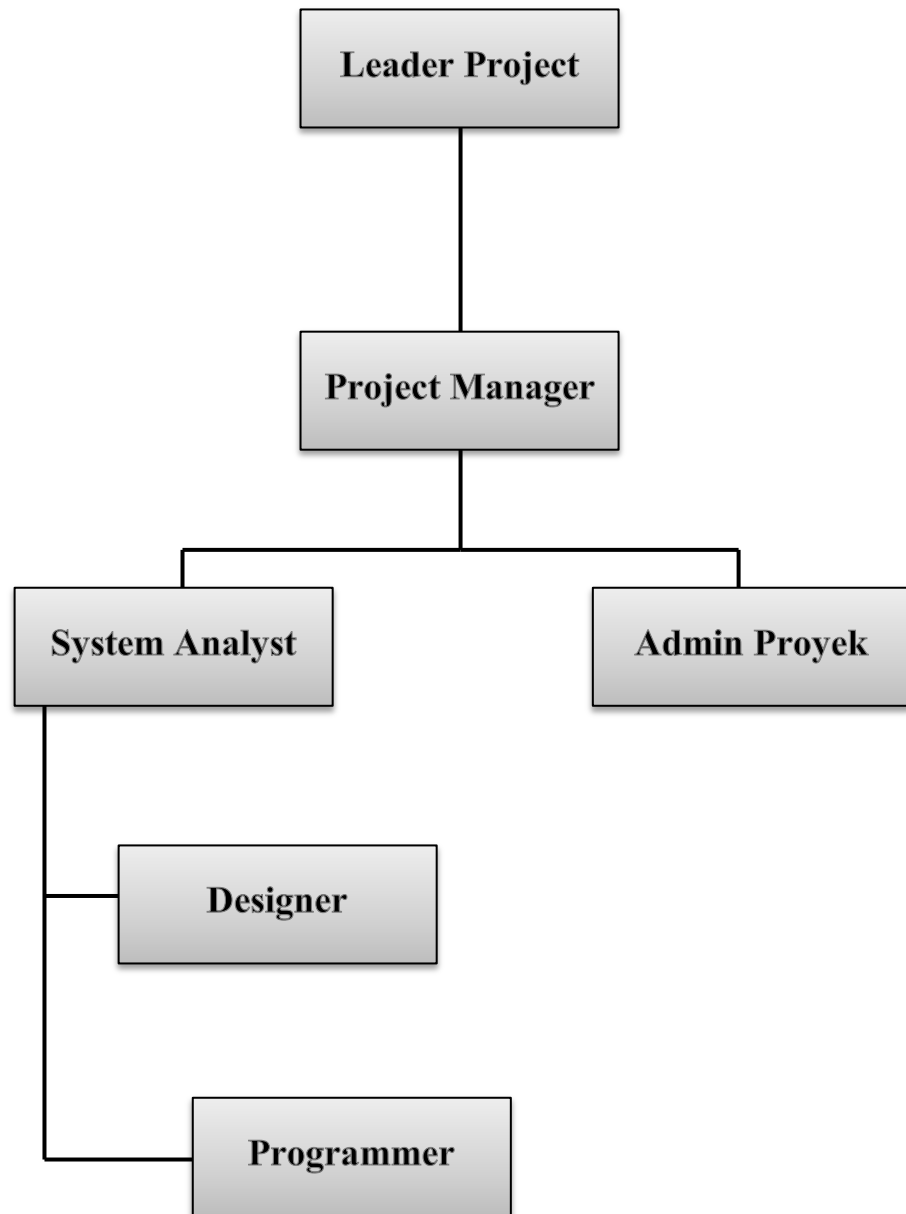
Berikut ini adalah RAB (Rencana Anggaran Biaya) pelaksanaan proyek secara garis besar:

Tabel 3.2 Rencana Anggaran Biaya

1. Biaya Pelaksanaan Proyek:

| Nama | Biaya |
|--------------------------|------------------|
| Manajer Proyek | Rp. 2.000.000,- |
| <i>System Analist</i> | Rp. 3.000.000,- |
| Admin Proyek | Rp. 500.000,- |
| <i>Programmer</i> | Rp. 2.500.000,- |
| <i>Designer</i> | Rp. 750.000,- |
| Pengumpulan Data | Rp. 350.000,- |
| Pengujian Aplikasi | Rp. 500.000,- |
| Pembuatan Laporan | Rp. 750.000,- |
| Biaya Keseluruhan | Rp. 10.350.000,- |

3.6 Struktur Tim Proyek



Gambar 3.5 Struktur Organisasi Tim Proyek

Pada struktur tim proyek diatas terdiri dari beberapa bagian, berikut penjelasan dari tugas masing-masing bagian tersebut:

a. *Project Leader*

Project Leader adalah orang yang memimpin proyek sekaligus sebagai penanggung jawab proyek.

b. *Project Manager*

Project manager adalah orang yang bertugas membantu *project leader* dalam proyek yang sedang berjalan.

c. *System Analyst*

Orang yang menganalisa kebutuhan aplikasi sekaligus menentukan *software* yang akan digunakan dalam pembuatan aplikasi.

d. Admin Proyek

Orang yang membuat laporan proyek.

e. *Designer*

Orang yang bertugas dan bertanggung jawab merancang tampilan aplikasi.

f. *Programmer*

Setelah aplikasi selesai dirancang oleh *designer*, proyek pembuatan aplikasi akan diambil alih oleh *programmer* untuk merancang *coding* yang kemudian diaplikasikan ke aplikasi yang telah dirancang.

BAB IV

PERANCANGAN DAN ANALISIS

4.1 Analisa Masalah

Pada zaman sekarang yang teknologinya sudah semakin maju, telah membuat banyak orang untuk berpikir supaya bagaimana cara memanfaatkan teknologi khususnya teknologi *mobile* seperti *smartphone* untuk bermanfaat untuk kehidupan atau perusahaan mereka. Namun masih ada suatu perusahaan yang tidak memanfaatkan teknologi itu sama sekali karena belum tahu atau belum mengalami bagaimana rasanya untuk dapat memanfaatkan suatu dengan teknologi tersebut. Dalam perusahaan yang berkelut dibidang konstruksi pasti sangat dibutuhkan sekali efisiensi waktu dan ketepatan dalam berkerja, sehingga dapat memenuhi keinginan klien, contohnya saat dalam pengambilan bahan-bahan konstruksi pasti sangat dibutuhkan sekali ketepatan waktu sehingga dapat memperlancar pekerjaan konstruksi, namun karena perusahaan tidak menerapkan teknologi pada perusahaannya banyak para pekerjanya yang bertugas untuk mengambil bahan konstruksi keliru terhadap letak-letak *site quarry* yang ingin dituju, apalagi untuk pekerja yang datang dari luar kota yang masih belum mengetahui seluk beluk jalan di Bangka sehingga itu dapat menghambat pekerjaan konstruksi, dan dapat membuat pekerjaan proyek tidak sesuai jadwal sehingga dapat menimbulkan kerugian waktu.

4.1.1 Analisa Sistem Berjalan

Analisis Masukan Sistem Berjalan bertujuan untuk menjelaskan proses sistem yang sedang berjalan pada perusahaan PT.INTI SAKTI UTAMA. Pekerjaan biasanya yang dilakukan pada perusahaan ini minim sekali dalam penggunaan teknologi *mobile*, yang artinya belum menerapkan teknologi *mobile* dalam hal mempermudah pekerjaan, sehingga pekerjaan sering mengalami keterlambatan.

Berikut ini merupakan penjabaran proses sistem yang sedang berjalan atau terjadi diperusahaan tersebut:

a. Proses Cek bahan

Mandor melakukan survei pekerjaan, membuat daftar bahan yang dibutuhkan, menyerahkan daftar bahan ke petugas stockpile, petugas stockpile melakukan cek ketersediaan bahan.

b. Proses ambil bahan

Mandor memberi perintah ke supir pengangkut untuk mengambil bahan, supir pengangkut mengambil bahan ke lokasi quarry.

c. Proses angkut bahan

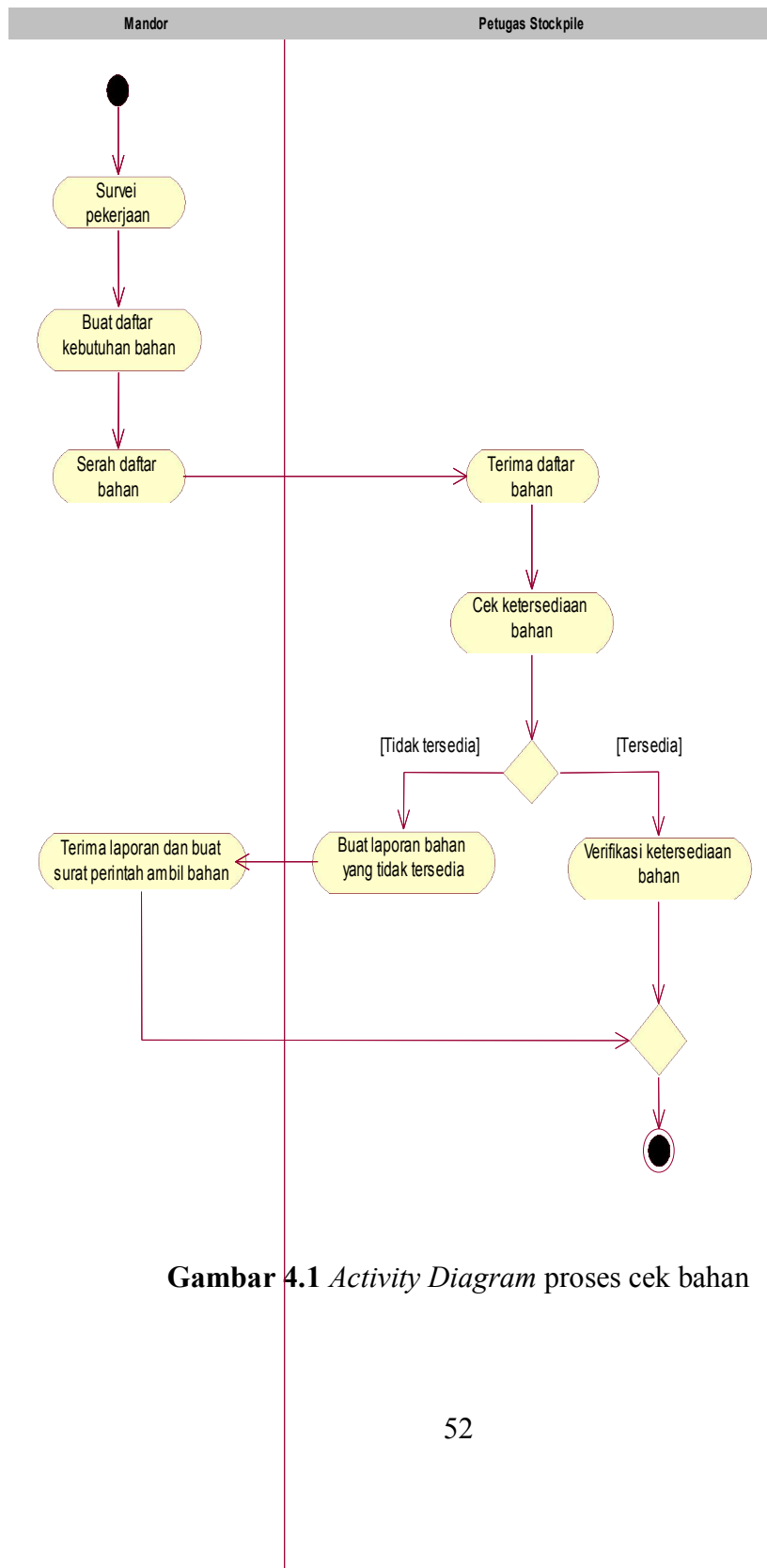
Supir pengangkut bahan melapor untuk mengangkut bahan, supplier mendapat laporan, supplier mengecek kontrak kerjasama, supplier menghitung kuantitas bahan yang akan diangkut, beri izin pengangkutan, supir pengangkut mendapat izin, memulai pengangkutan.

d. Proses pemasukan bahan

Supir pengangkut mengantar bahan ke stockpile, melapor ke petugas stockpile, petugas stockpile mengecek bahan yang masuk, buat laporan bahan yang masuk, menyerahkan laporan ke mandor, mandor menerima laporan, mandor memulai pekerjaan proyek.

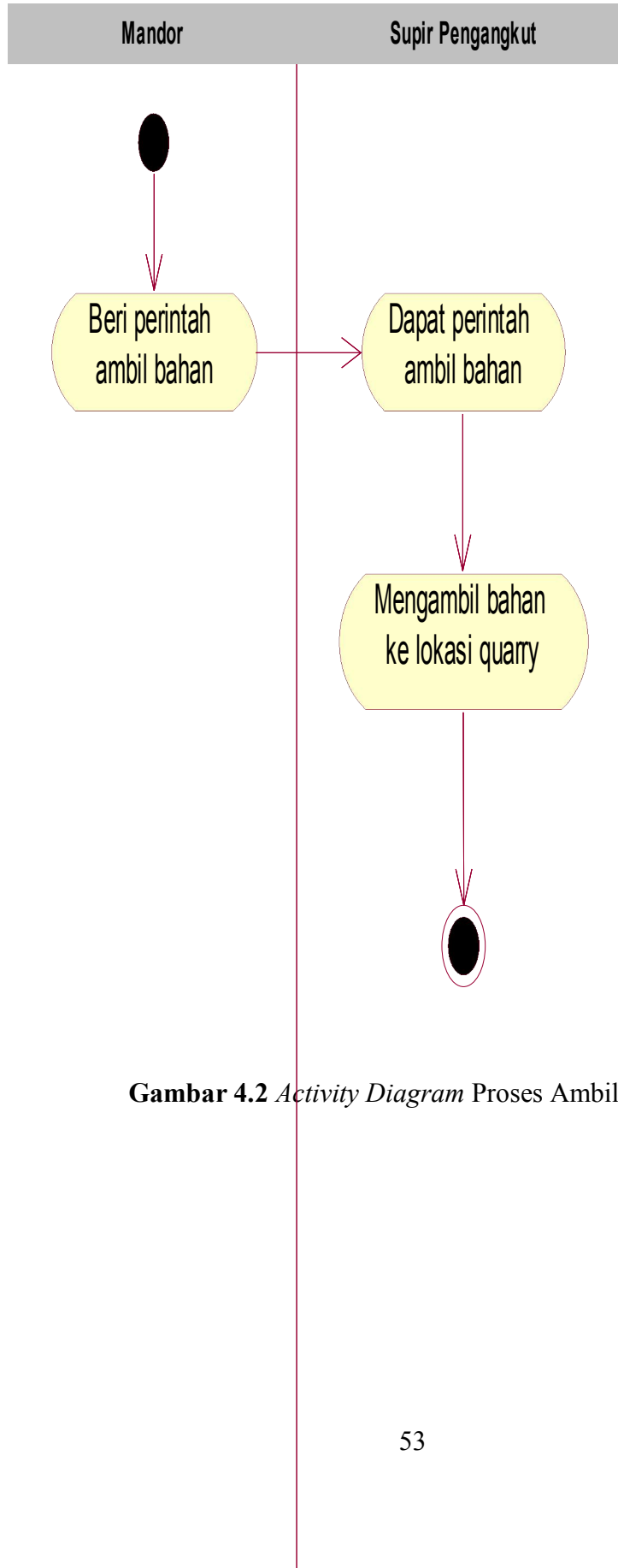
1. Activity Diagram Proses Cek Bahan

Activity diagram ini menggambarkan alir aktivitas dalam sistem yang sedang berjalan, mulai dari proses awal hingga proses akhir pada perusahaan. Berikut merupakan Activity Diagram yang mengacu pada sistem yang sedang berjalan:



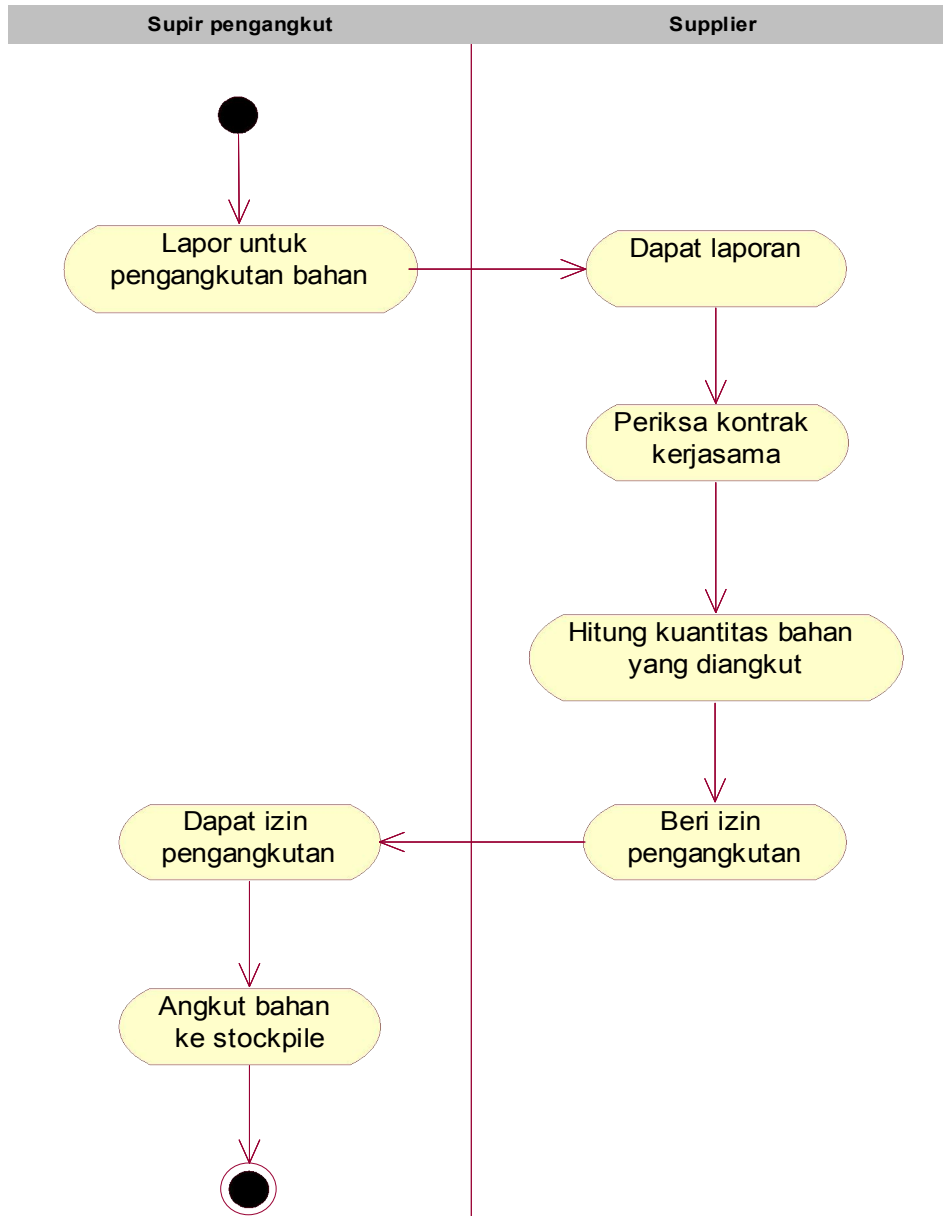
Gambar 4.1 Activity Diagram proses cek bahan

2. Activity Diagram Proses Ambil Bahan



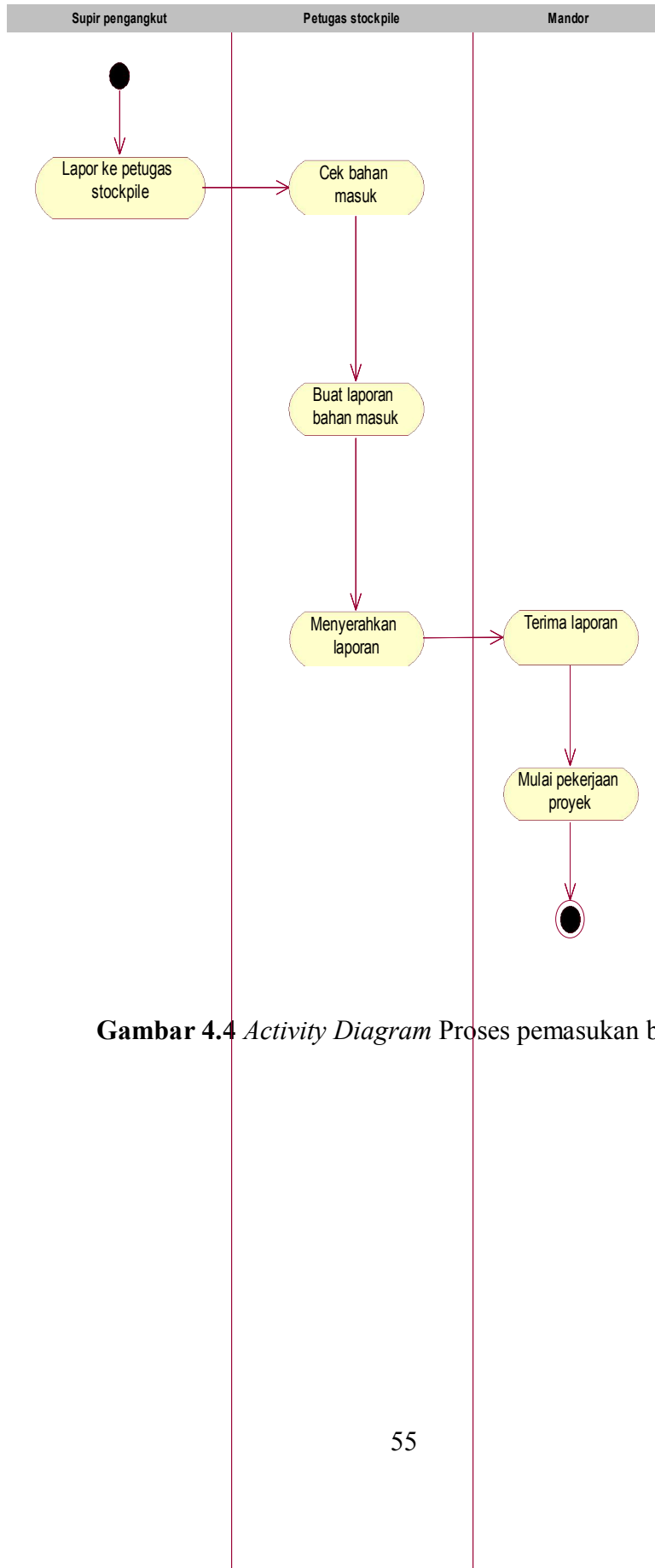
Gambar 4.2 Activity Diagram Proses Ambil Bahan

3. Activity Diagram Proses Angkut Bahan



Gambar 4.3 Activity Diagram proses angkut bahan

4. Activity Diagram Proses Pemasukan Bahan



Gambar 4.4 Activity Diagram Proses pemasukan bahan

4.1.2 Analisa Masukan Sistem Berjalan

Pada analisa masukan berisi tentang semua data yang merupakan masukan (*input*) dimana proses sedang terjadi yaitu daftar bahan yang dibutuhkan, ketersediaan bahan.

- a. Nama Masukan : Daftar bahan yang dibutuhkan
Fungsi : Untuk mengadakan bahan konstruksi yang dibutuhkan
Media : Kertas
Sumber : Mandor
Rangkap : 1 (Satu)
Frekuensi : 1 Pekerjaan
Keterangan : -

- b. Nama Masukan : Perintah pengambilan bahan
Fungsi : Untuk mengadakan bahan yang tidak tersedia
Media : Kertas
Sumber : Mandor
Rangkap : 1 (Satu)
Frekuensi : 1 Pekerjaan
Keterangan : -

4.1.3 Analisa Keluaran Sistem Berjalan

Pada analisa keluaran berisi tentang semua data yang merupakan keluaran (*output*) sistem yang sedang berjalan. Yaitu rekapitulasi bahan konstruksi dalam 1 pekerjaan.

- a. Nama Masukan : Rekapitulasi bahan konstruksi dalam 1 pekerjaan
Fungsi : Melaporkan semua bahan yang tersedia dalam 1 pekerjaan
Media : Kertas
Sumber : Mandor
Rangkap : 1 (Satu)
Frekuensi : 1 Pekerjaan
Keterangan : -
- b. Nama Masukan : Laporan bahan masuk
Fungsi : Bukti pengangkutan bahan
Media : Kertas
Sumber : Petugas *Stockpile*
Rangkap : 1 (Satu)
Frekuensi : 1 Pekerjaan
Keterangan : -

4.1.5 Analisa Kebutuhan

Dalam pembuatan aplikasi ini dibutuhkan perangkat keras (*Hardware*) atau PC (*Personal Computer*) yang dapat memperlancar pembuatan aplikasi.

a. Perangkat keras yang dibutuhkan adalah sebagai berikut:

1. Processor Intel Core i3-3120M CPU
2. RAM 4GB
3. VGA Intel HD Graphic 4000

b. Perangkat lunak yang dibutuhkan:

1. Eclipse versi Luna
2. Java JDK 7
3. Android SDK

c. Spesifikasi *Smartphone* yang dibutuhkan:

1. *Processor Dual Core* 1 Ghz
2. RAM 512 MB
3. *Android Jelly Bean* versi 4.2.2 keatas
4. Kapasitas memori internal minimal 100 MB
5. VGA 4" HVGA LCD

4.2 Perancangan

4.2.1 Rancangan Dokumen Masukan

Berikut masukan-masukan yang diusulkan:

a. Angka Lattitude dan Longitude

| | |
|--------------|---|
| Nama Masukan | : Angka Lattitude dan Longitude |
| Fungsi | : Sebagai masukan untuk menginput titik koordinat |
| Media | : <i>Softcopy</i> |
| Rangkap | : 1 (Satu) |
| Frekuensi | : 1 Kali |
| Format | : <i>Number</i> |
| Keterangan | : - |

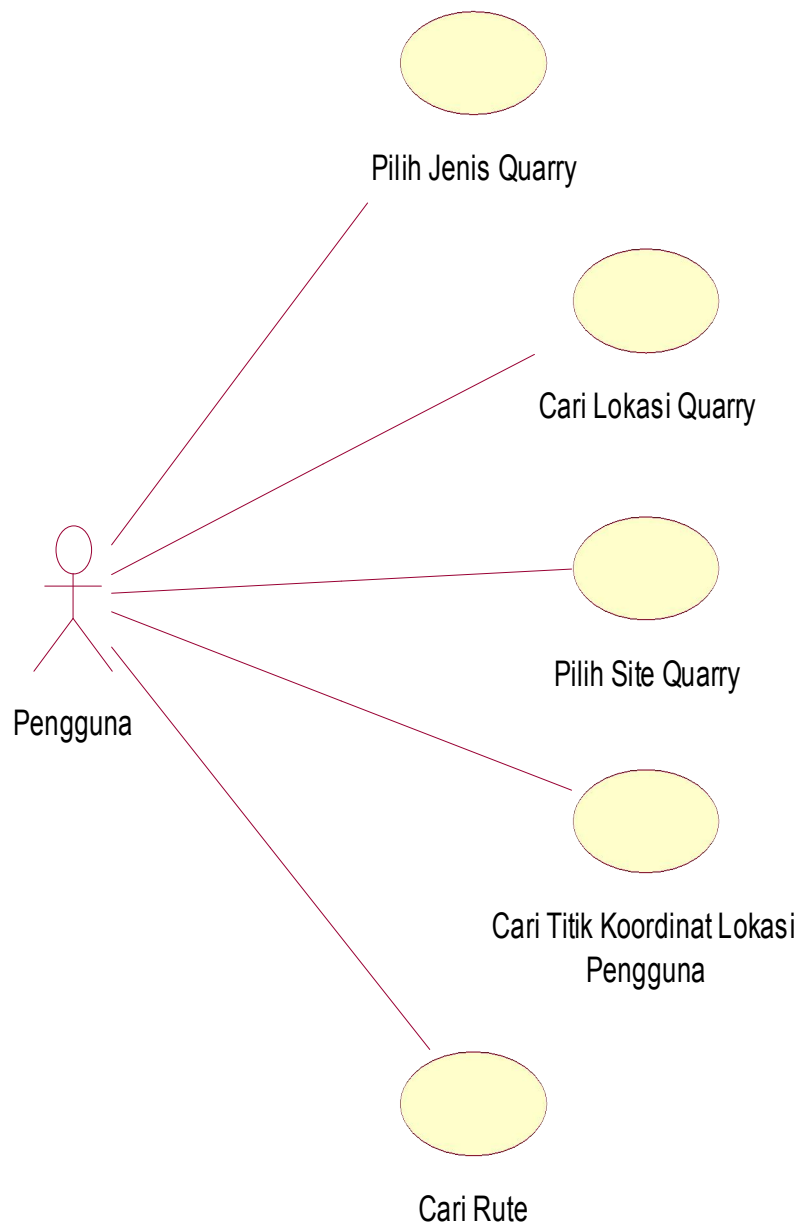
4.2.2 Rancangan Dokumen Keluaran

Berikut merupakan keluaran yang diusulkan:

a. Tampilan Lokasi Quarry

| | |
|------------|---|
| Nama | : Lokasi Quarry |
| Fungsi | : Menampilkan Titik Koordinat pada peta |
| Media | : Aplikasi Android |
| Rangkap | : 1 (Satu) |
| Frekuensi | : 1 Kali |
| Format | : Lihat pada peta aplikasi |
| Keterangan | : - |

4.2.3 Use Case Diagram yang Diusulkan



Gambar 4.5 Use Case Diagram yang diusulkan

4.2.4 Deskripsi Use Case

Use case dapat digambarkan sebagai suatu cara tertentu untuk menggunakan sistem dari sudut pandang satu pengguna (*an actor*). Berikut adalah penjelasan dari beberapa *use case* yaitu:

a. Pilih jenis *quarry*

Tabel 4.1 *Use Case* pilih jenis *quarry*

| | |
|----------------------|---|
| <i>Use Case</i> | pilih jenis <i>quarry</i> |
| <i>Actor</i> | Pengguna |
| <i>Description</i> | Proses pilih jenis <i>quarry</i> |
| <i>Precondition</i> | Pengguna berada pada tampilan Menu Awal |
| <i>Flow of Event</i> | Sistem akan menampilkan <i>layout</i> MenuPilih |
| <i>postcondition</i> | Pengguna masuk ke <i>layout</i> MenuPilih |

b. Cari Lokasi Quarry

Tabel 4.2 *Use Case* Lokasi Quarry

| | |
|----------------------|--|
| <i>Use Case</i> | Cari Lokasi <i>Quarry</i> |
| <i>Actor</i> | Pengguna |
| <i>Description</i> | Proses pilih lokasi <i>quarry</i> sesuai material tambang |
| <i>Precondition</i> | Pengguna berada pada tampilan <i>layout</i> MenuPilih |
| <i>Flow of Event</i> | Sistem akan menampilkan <i>layout</i> lokasi sesuai material tambang |
| <i>postcondition</i> | Map akan muncul pada <i>layout</i> lokasi sesuai material tambang |

c. Pilih Site Quarry

Tabel 4.3 *Use Case* Pilih Site *Quarry*

| | |
|---------------------|---|
| <i>Use Case</i> | Pilih Site <i>Quarry</i> |
| <i>Actor</i> | Pengguna |
| <i>Description</i> | Proses untuk menampilkan titik koordinat lokasi dalam peta |
| <i>Precondition</i> | Pengguna berada dalam <i>layout</i> lokasi serta berinteraksi dengan peta yang tersedia |

| | |
|----------------------|--|
| <i>Flow of Event</i> | Sistem akan menampilkan titik koordinat pada peta dan menampilkan informasi lainnya. |
| <i>postcondition</i> | Pengguna dapat melihat titik koordinat lokasi <i>quarry</i> |

d. Cari Titik Koordinat Lokasi Pengguna

Tabel 4.4 *Use Case* Cari titik Koordinat Pengguna

| | |
|----------------------|---|
| <i>Use Case</i> | Cari titik koordinat lokasi pengguna |
| <i>Actor</i> | Pengguna |
| <i>Description</i> | Proses untuk menampilkan titik koordinat lokasi pengguna sekarang berada |
| <i>Precondition</i> | Pengguna berada dalam <i>interface</i> peta yang sama dengan pencarian lokasi <i>quarry</i> |
| <i>Flow of Event</i> | Sistem akan mengunduh <i>magnitude</i> dan <i>longitude</i> GPS <i>smartphone</i> pengguna dari satelit yang kemudian akan ditampilkan kedalam peta |
| <i>postcondition</i> | Pengguna dapat mengetahui dimana lokasi tempat dia berada |

e. Cari Rute

Tabel 4.5 *Use Case* Cari Rute

| | |
|----------------------|---|
| <i>Use Case</i> | Cari Rute |
| <i>Actor</i> | Pengguna |
| <i>Description</i> | Proses untuk menampilkan rute dari lokasi pengguna ke lokasi yang ingin dituju untuk ditampilkan dipeta |
| <i>Precondition</i> | Pengguna berada dalam layout cari rute dengan peta yang telah ditampilkan |
| <i>Flow of Event</i> | Pada saat peta disentuh maka akan muncul marker baru dimana marker tersebut merupakan lokasi tujuan, setelah itu sistem akan mengunduh data dari google untuk menampilkan rute dipeta, setelah data selesai diunduh akan muncul rute dari lokasi pengguna ke lokasi tujuan dipeta |

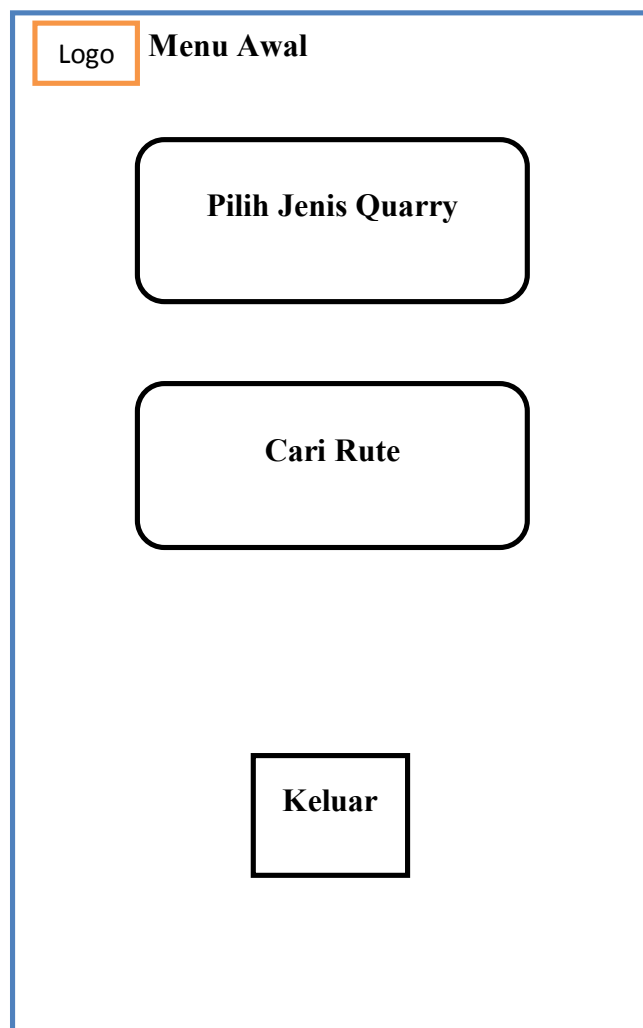
| | |
|----------------------|--|
| <i>postcondition</i> | Sistem telah menampilkan rute dipeta yang dapat dilihat pengguna |
|----------------------|--|

4.2.5 Rancangan Layar Aplikasi

Berikut merupakan rancangan layar untuk aplikasi yang akan dibuat.

a. Menu Awal

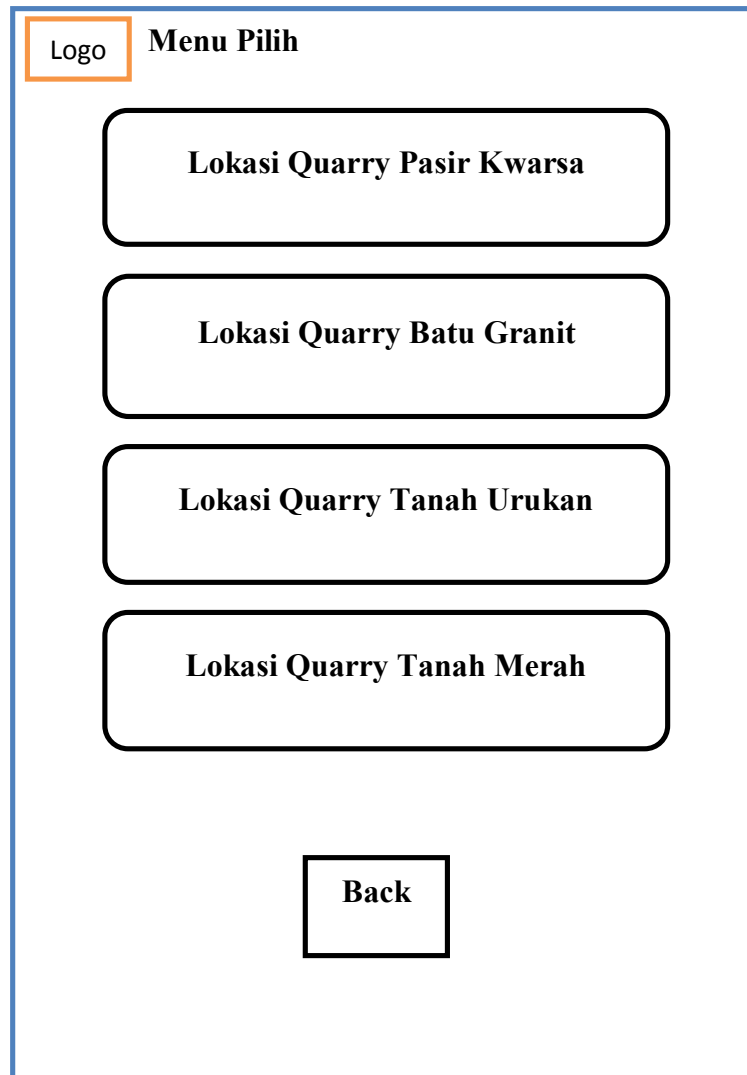
Didalam menu awal ini terdapat *button* Pilih Jenis Quarry, Cari Rute dan Keluar. Saat *Button* Pilih Jenis *Quarry* dipilih maka aplikasi akan memanggil layout Menu Pilih, jika *button* Cari Rute dipilih maka aplikasi akan memanggil *layout* rute, dan jika *button* Keluar dipilih maka aplikasi akan memunculkan kotak dialog keluar.



Gambar 4.6 Menu Awal

b. Menu Pilih

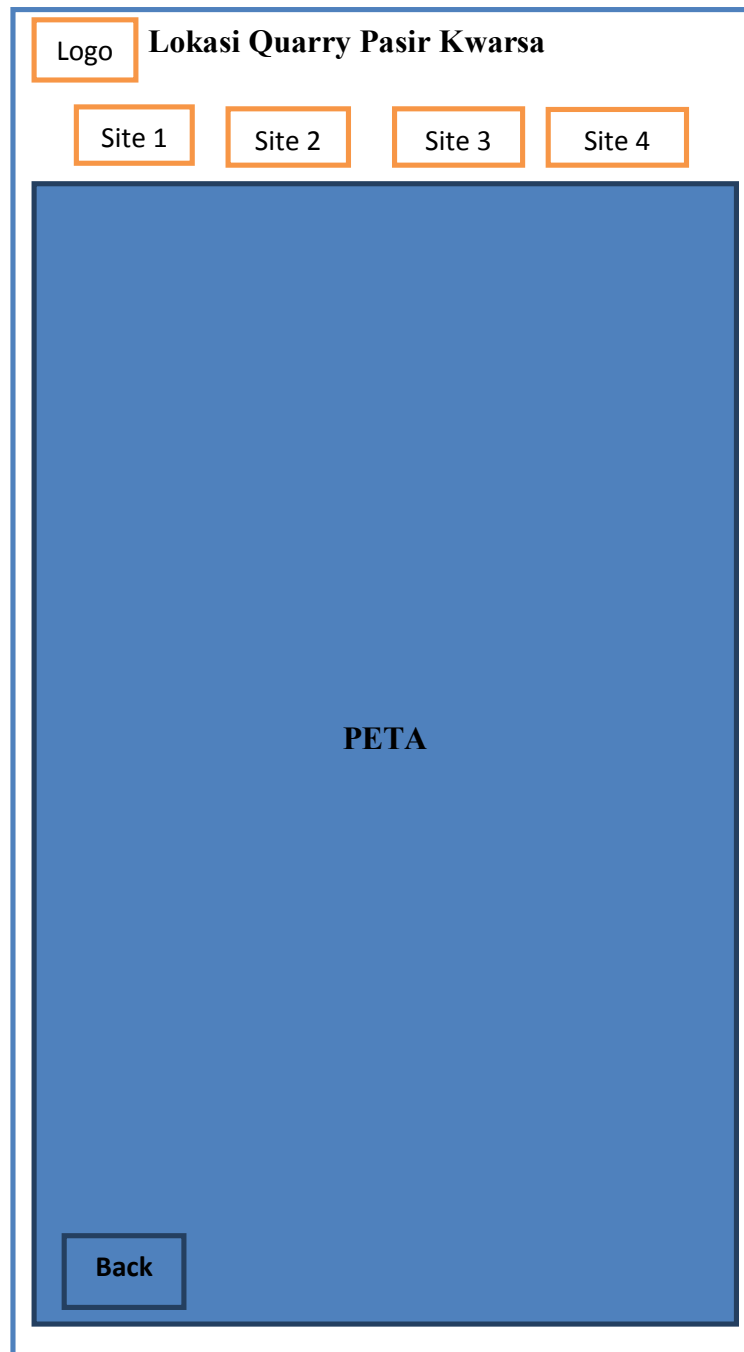
Didalam menu pilih ini terdapat *button* Lokasi Quarry Pasir Kwarsa, Lokasi Quarry Batu Granit, Lokasi Quarry Tanah Urukan, Lokasi Quarry Tanah Merah dan *button back*.



Gambar 4.7 Menu Pilih

c. *Layout* Lokasi Quarry Pasir Kwarsa

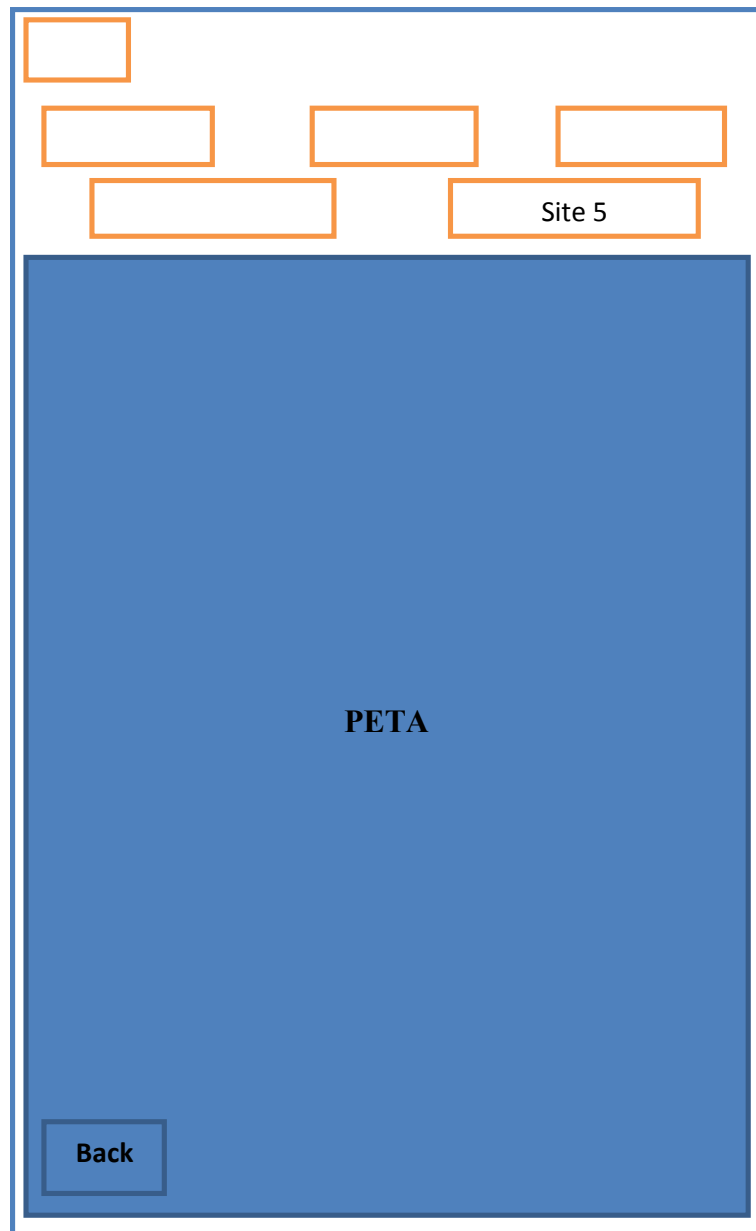
Didalam menu ini terdapat *button* site 1, site 2, site 3 dan site 4, tampilan peta *google map* dan *button back*.



Gambar 4.8 *Layout* Lokasi Quarry Pasir Kwarsa

d. *Layout* Lokasi Quarry Batu Granit

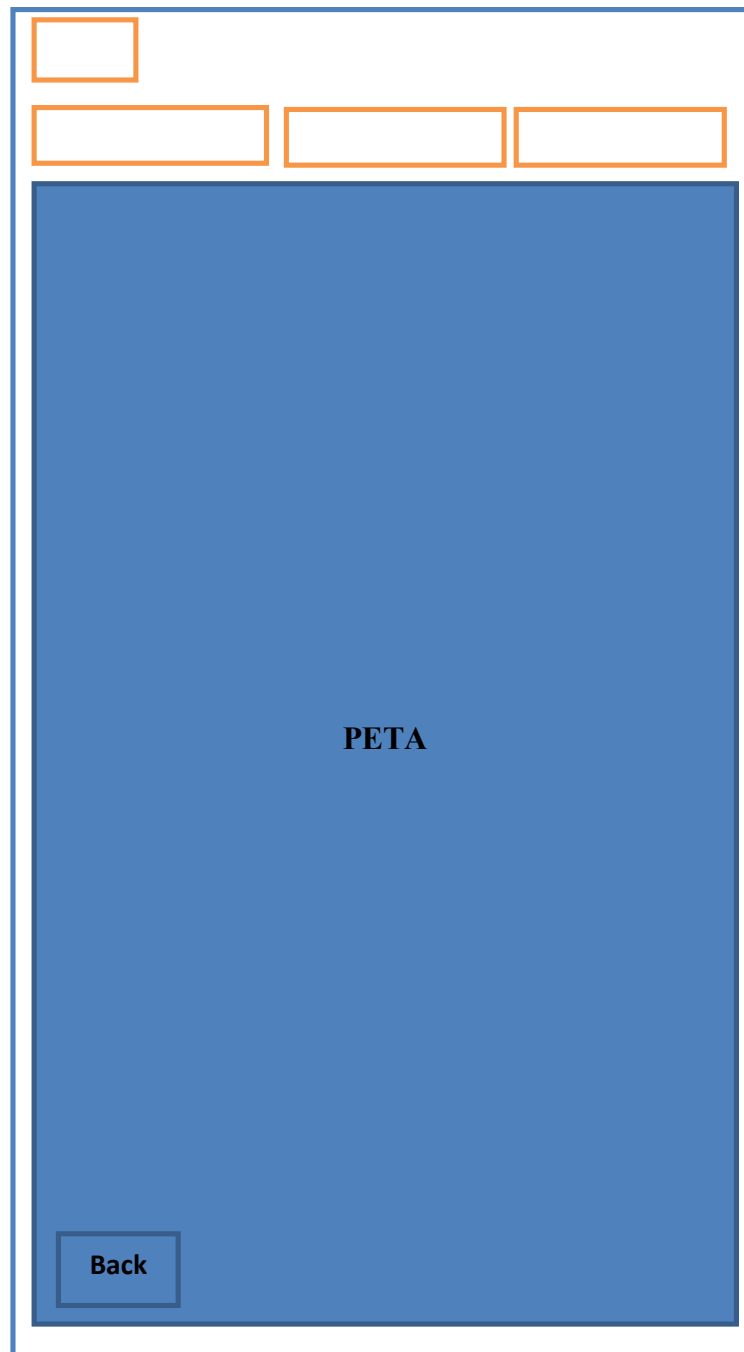
Didalam *layout* ini terdapat *button* site 1, site 2, site 3, site 4, site 5, tampilan peta google map dan *button* back.



Gambar 4.9 *Layout* Lokasi Quarry Batu Granit

e. *Layout* Lokasi Quarry Tanah Urukan

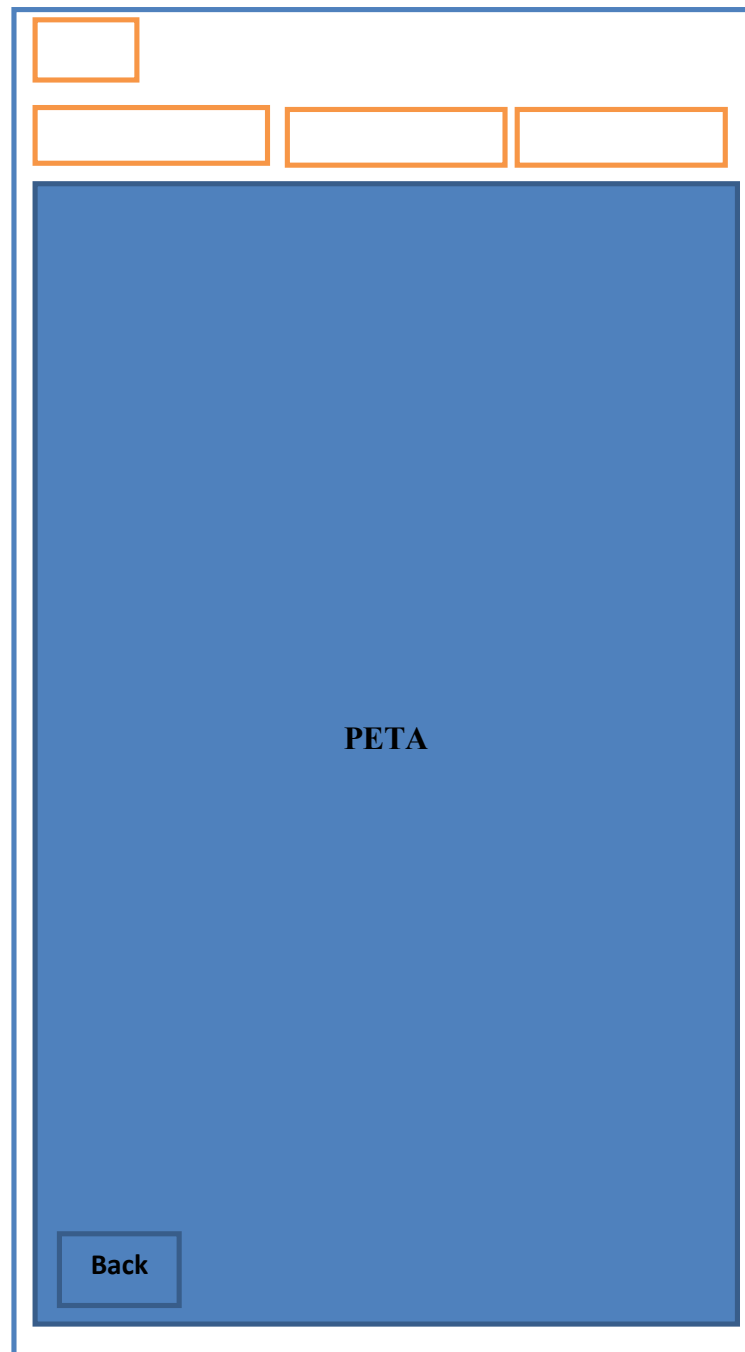
Didalam *layout* ini terdapat *button* site 1, site 2, site 3, tampilan peta *google map* dan *button back*.



Gambar 4.10 *Layout* Lokasi Quarry Tanah Urukan

f. *Layout* Lokasi Quarry Tanah Merah

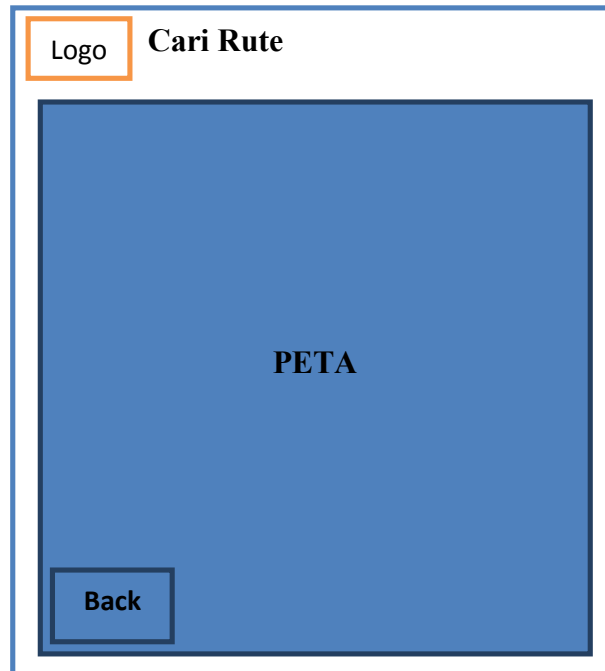
Didalam *layout* ini terdapat *button* site 1, site 2, site 3, tampilan *google map* dan *button back*.



Gambar 4.11 *Layout* Lokasi Quarry Tanah Merah

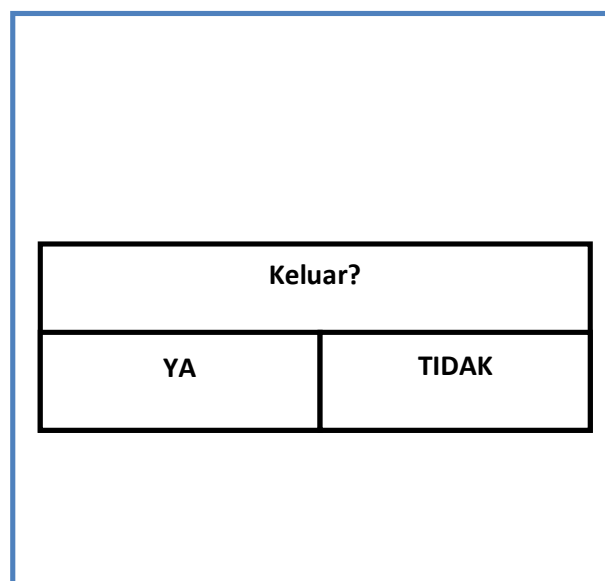
g. Layout Rute

Didalam layout Rute ini terdapat tampilan peta *google map* dan *button back*.



Gambar 4.12 Layout Cari Rute

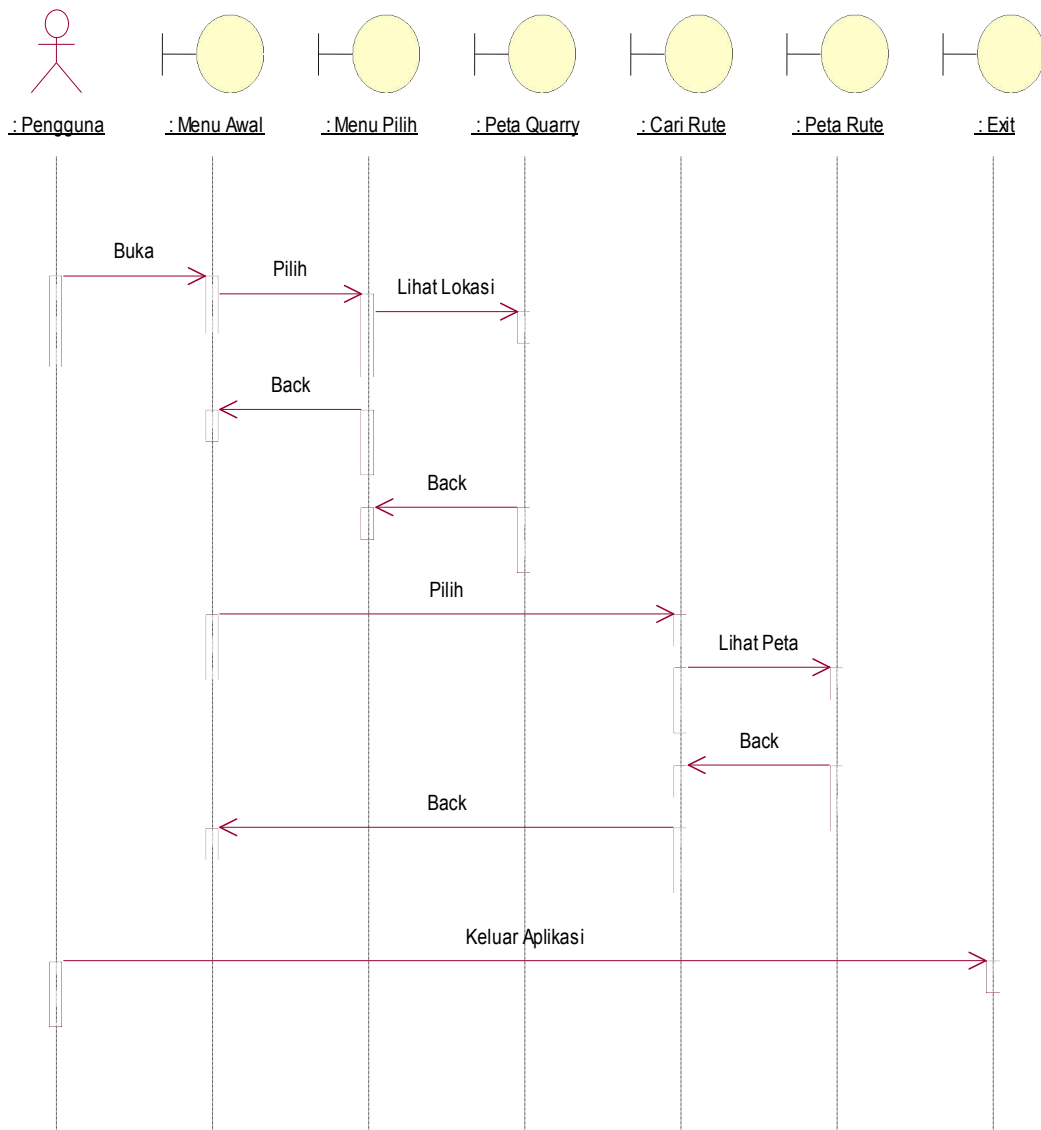
h. Layout Keluar



Gambar 4.13 Layout Keluar

4.2.6 Sequence Diagram

Sequence Diagram ini menjelaskan intraksi Objek yang disusun dalam urutan waktu. Diagram ini secara khusus berasosialisasi dengan *use case*. *Diagram sequence* memperlihatkan tahap demi tahap apa yang sebenarnya terjadi untuk menghasilkan sesuatu didalam *use case* berikut ini adalah gambar *sequence diagram* pada aplikasi yang dibuat:

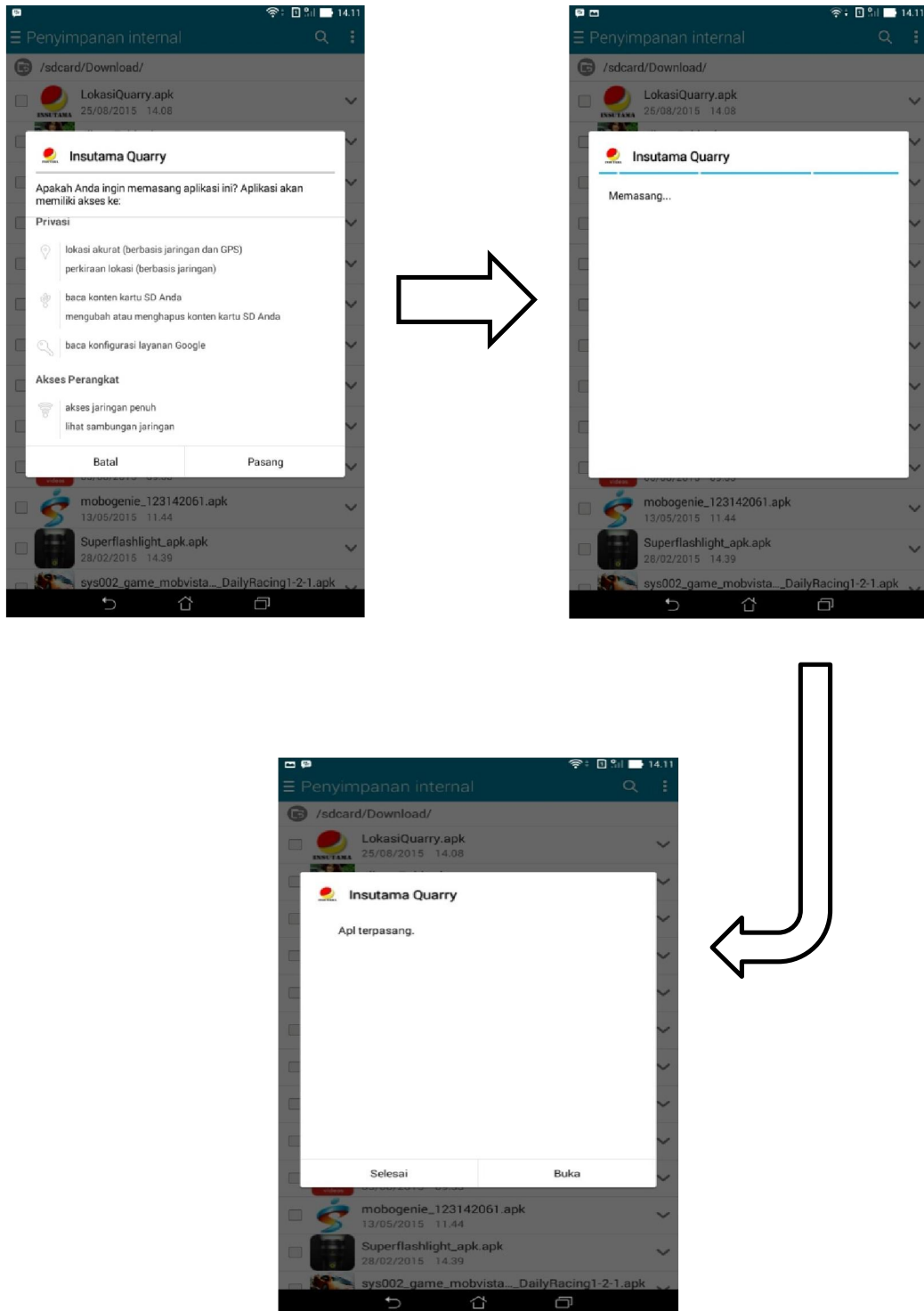


Gambar 4.14 *Sequence Diagram* Aplikasi

4.3 Implementasi

Adapun tahapan implementasi yang dilakukan pada tempat riset antara lain:

4.3.1 Menginstal Aplikasi



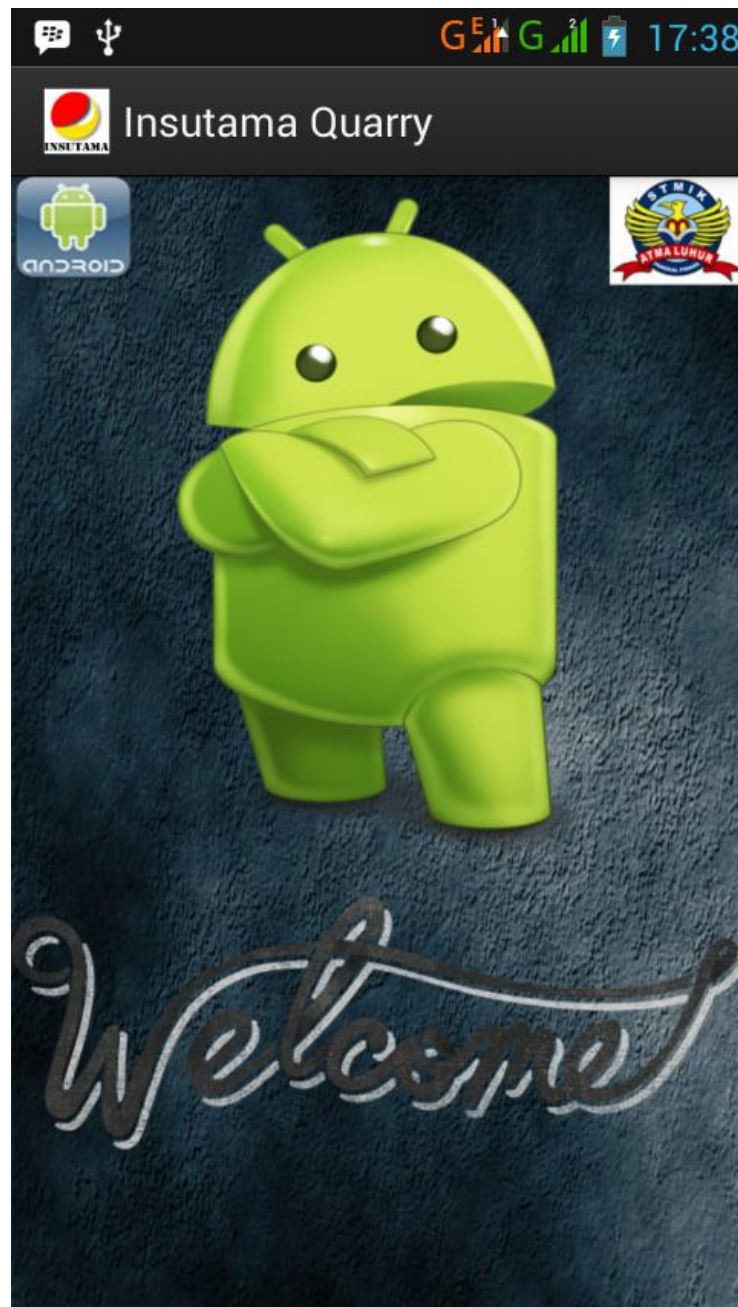
Gambar 4.15 Instalasi Aplikasi

4.3.2 Demo Aplikasi

Berikut ini merupakan gambar hasil *screenshot* dari aplikasi yang dijalankan di android.

a. Tampilan *Splashscreen*

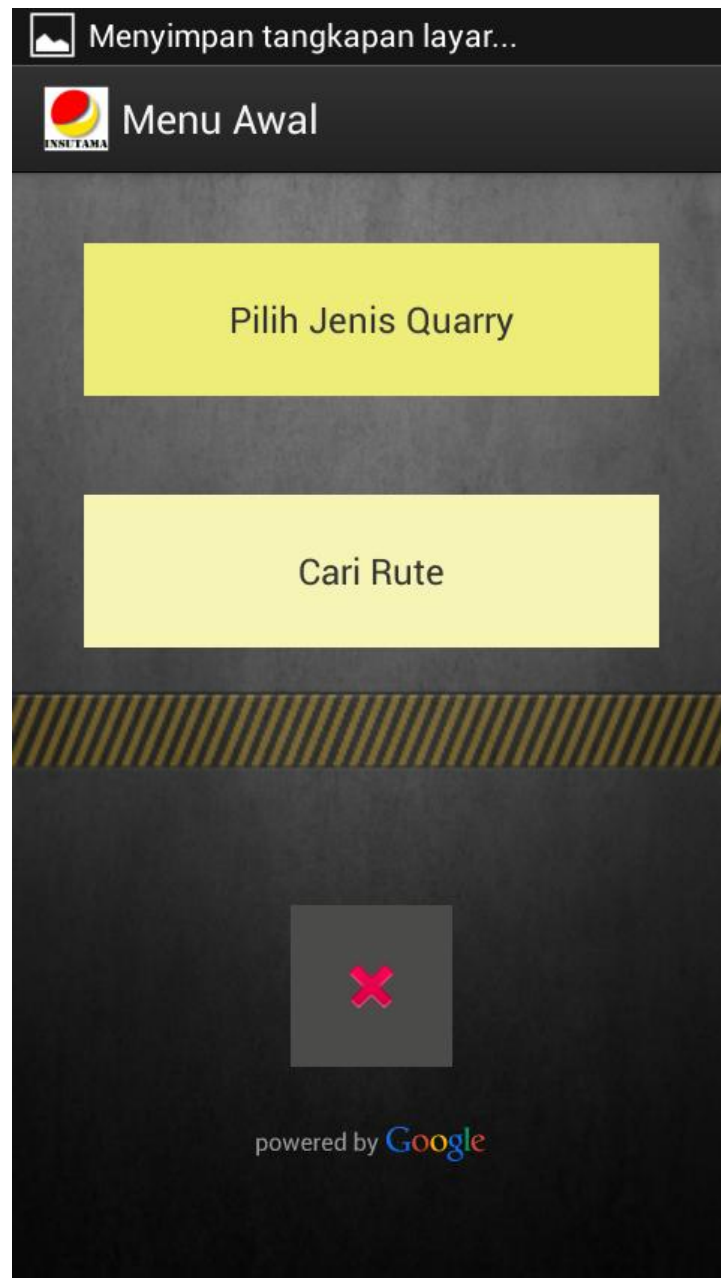
Saat aplikasi dijalankan pertama-tama akan muncul tampilan *splashscreen* seperti gambar berikut :



Gambar 4.16 Tampilan *Splashscreen*

b. Tampilan Menu Awal

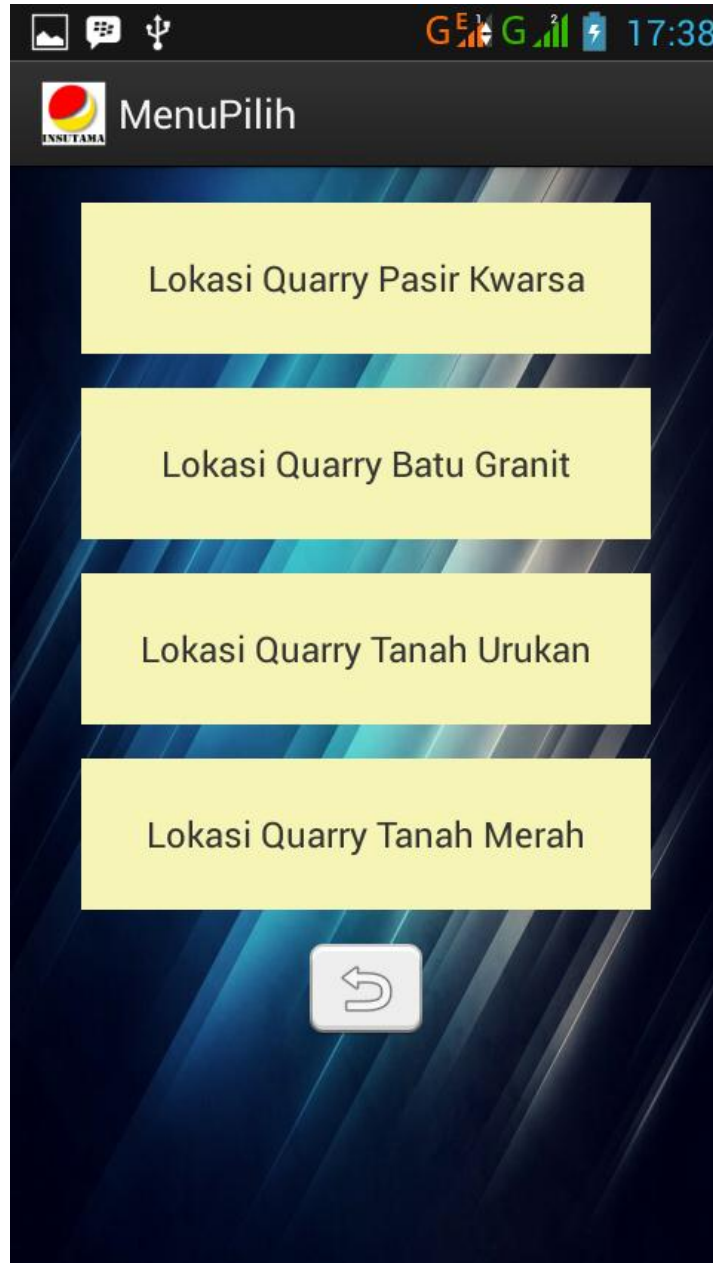
Setelah tampilan *splashscreen* selesai, maka akan muncul tampilan menu awal seperti gambar berikut ini:



Gambar 4.17 Tampilan Menu Awal

c. Menu Pilih

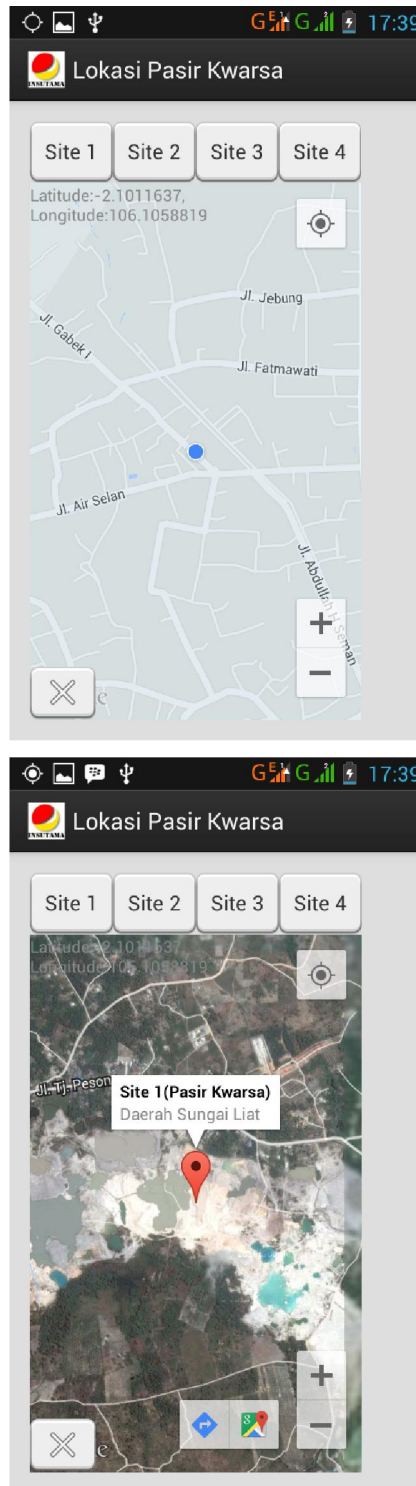
Jika *button* pilih jenis quarry yang ada pada menu awal dipilih maka selanjutnya akan muncul tampilan menu pilih seperti gambar berikut ini:



Gambar 4.18 Tampilan Menu Pilih

d. Tampilan Lokasi Quarry Pasir Kwarsa

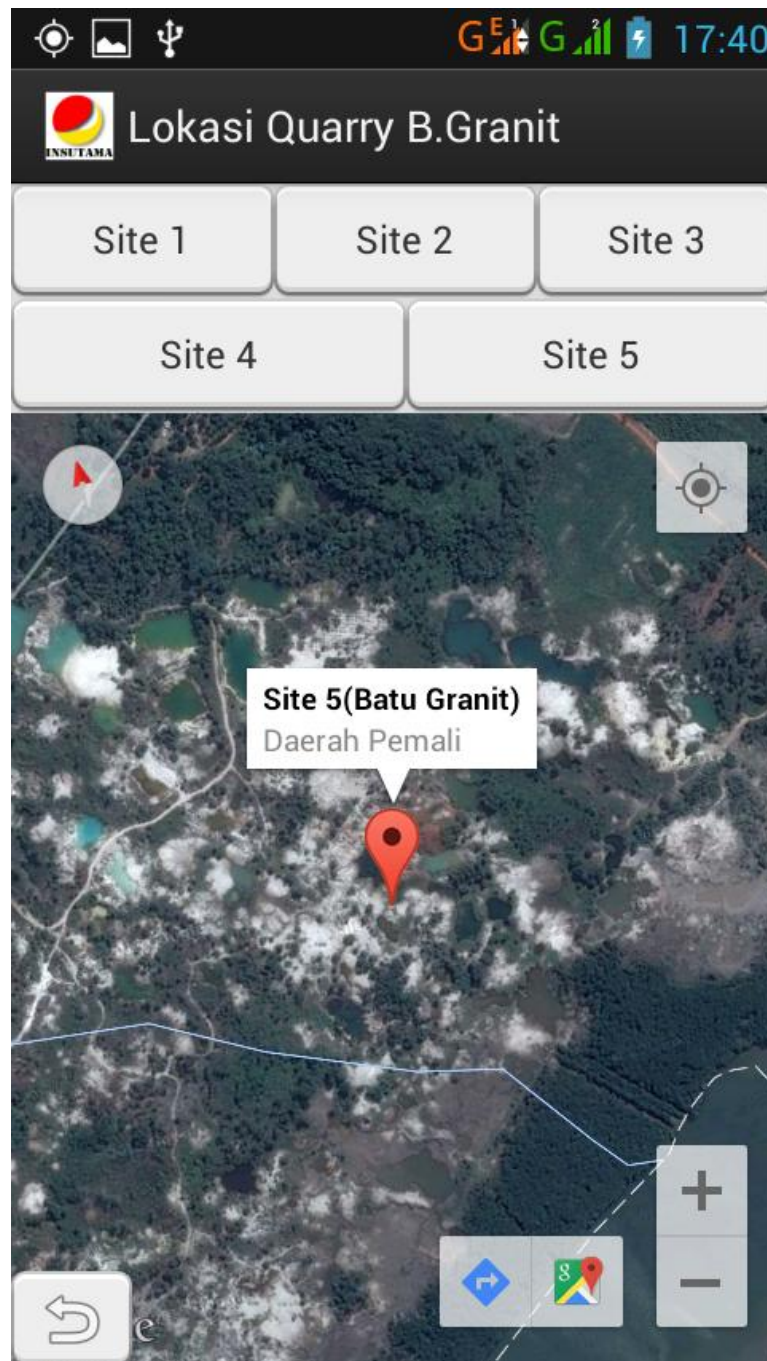
Jika button Lokasi Quarry Pasir Kwarsa yang ada pada menu pilih dipilih maka akan muncul tampilan seperti dibawah ini:



Gambar 4.19 Tampilan Lokasi Quarry Pasir Kwarsa

e. Tampilan Lokasi Quarry Batu Granit

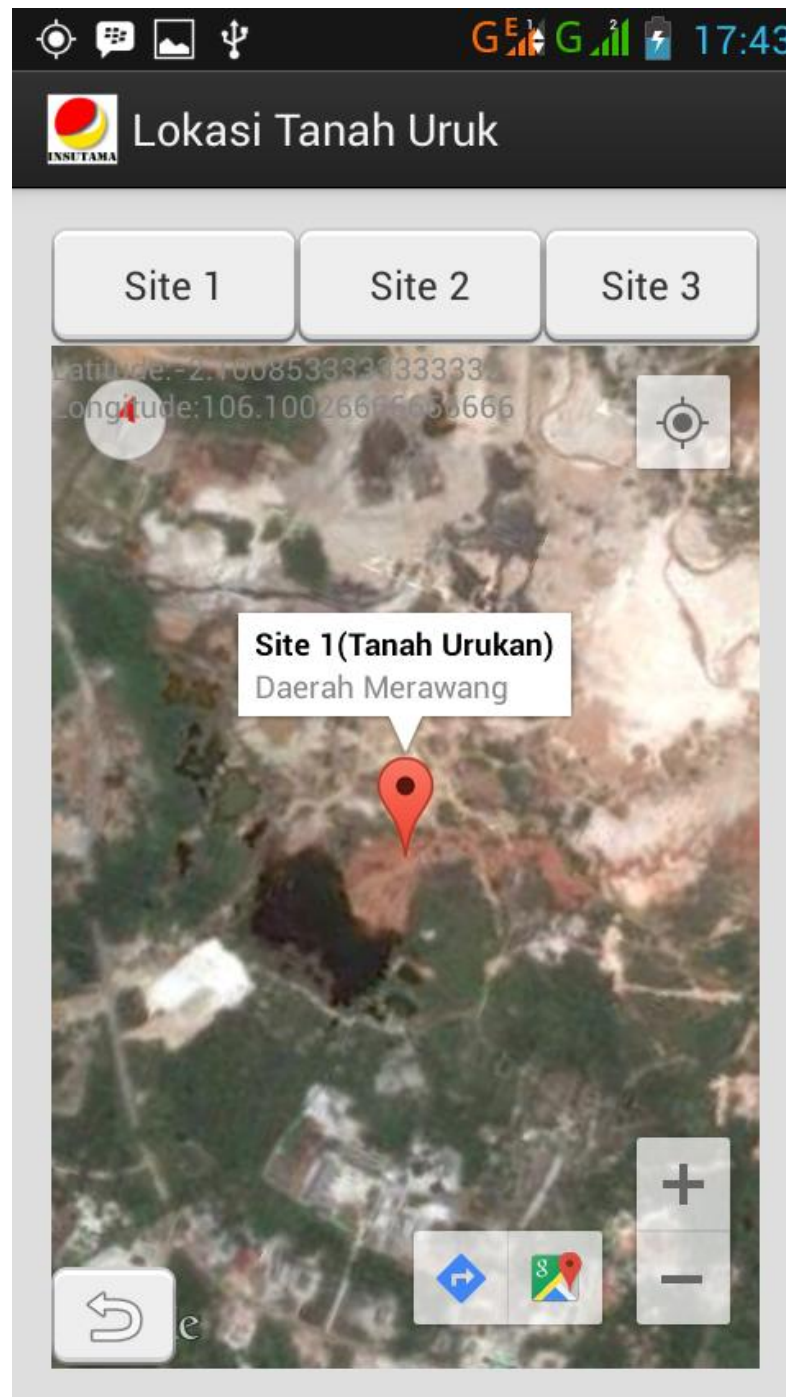
Jika button Lokasi Quarry Batu Granit yang ada pada menu pilih dipilih maka akan muncul tampilan seperti dibawah ini:



Gambar 4.20 Tampilan Lokasi Quarry Batu Granit

f. Tampilan Lokasi Quarry Tanah Uruk

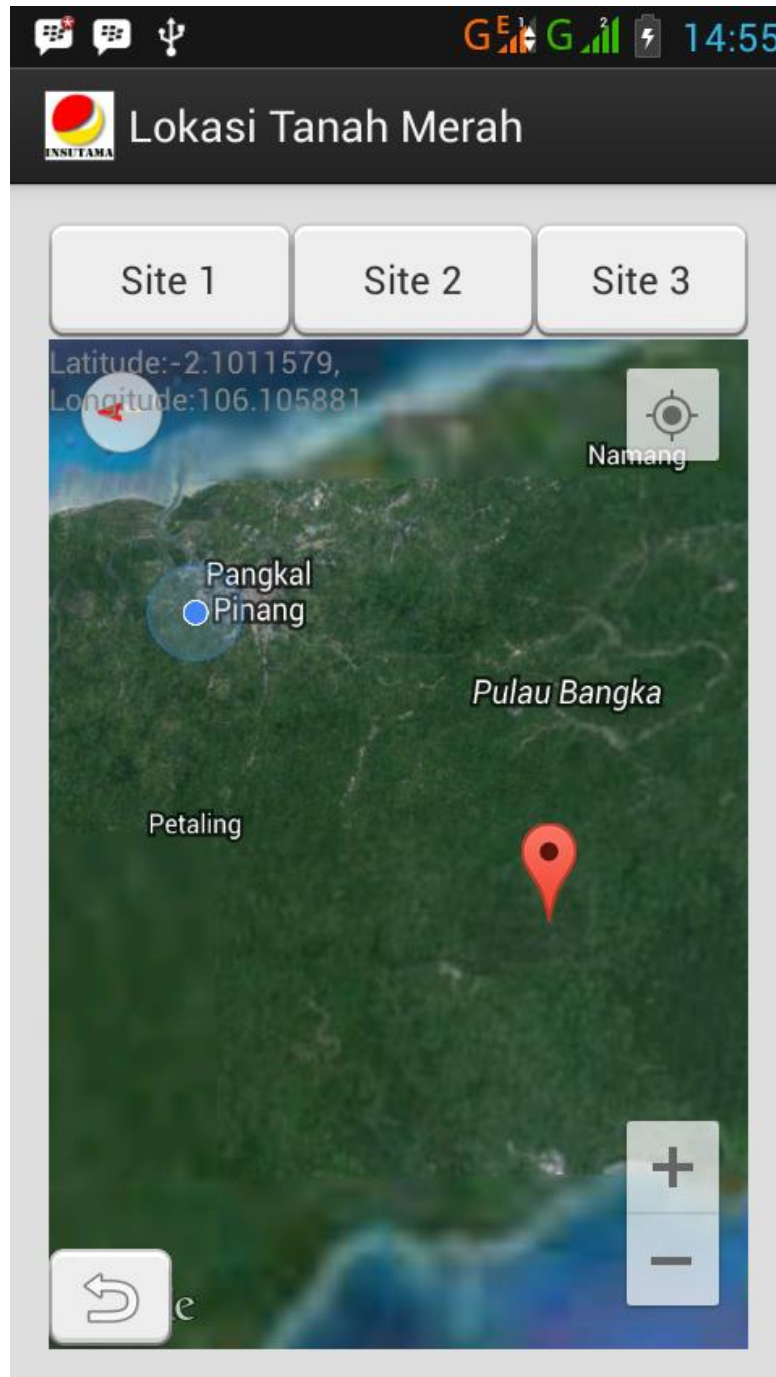
Jika button Lokasi Quarry Tanah Uruk yang ada pada menu pilih dipilih maka akan muncul tampilan seperti dibawah ini:



Gambar 4.21 Tampilan Lokasi Quarry Tanah Uruk

g. Tampilan Lokasi Quarry Tanah merah

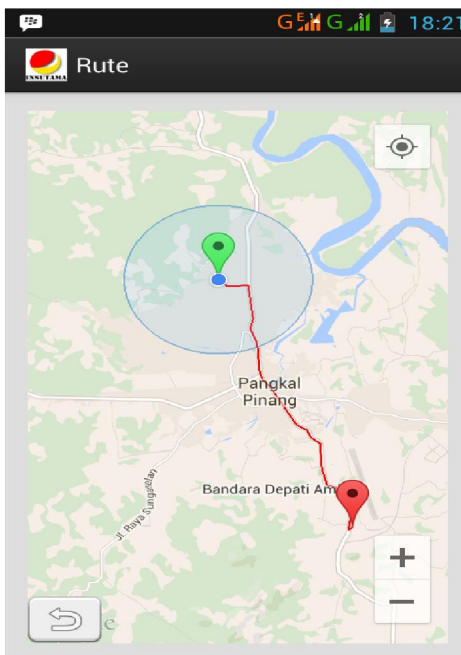
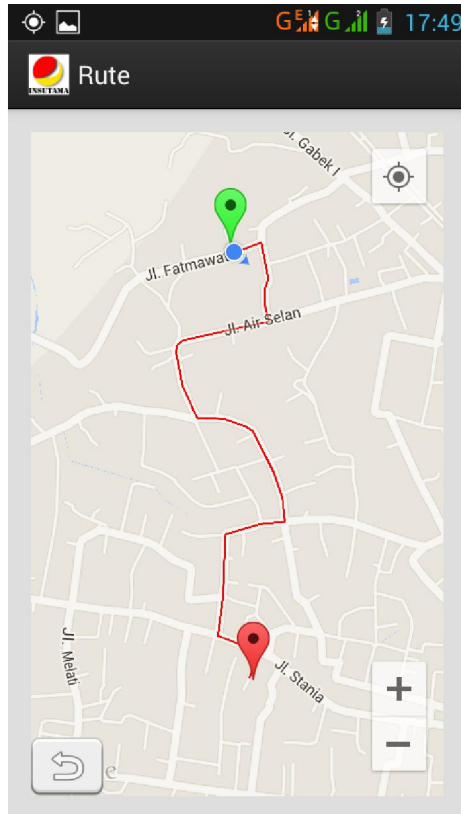
Jika button Lokasi Quarry Tanah Merah yang ada pada menu pilih dipilih maka akan muncul tampilan seperti dibawah ini:



Gambar 4.22 Tampilan Lokasi Quarry Tanah Merah

h. Tampilan Rute

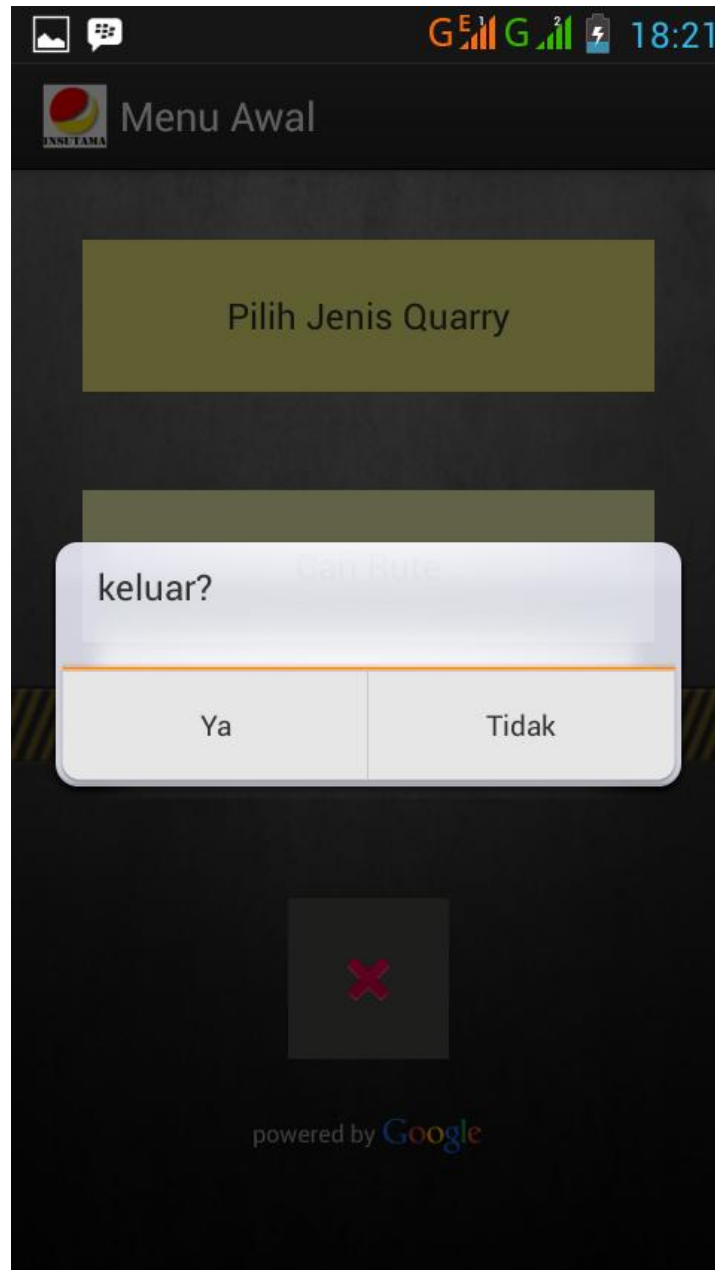
Jika button Cari Rute yang ada pada menu awal dipilih maka akan muncul tampilan seperti dibawah ini:



Gambar 4.23 Tampilan Rute

i. Tampilan Kotak Dialog Keluar

Jika button keluar yang ada pada menu awal dipilih maka akan muncul kotak dialog seperti gambar dibawah ini:



Gambar 4.24 Tampilan Kotak Dialog Keluar

4.3.3 Pengujian

Berikut merupakan Pengujian program aplikasi ini dengan menggunakan metode Pengujian *Black Box*. Pengujian ini dilakukan untuk mengetahui apakah aplikasi yang dibuat sesuai dengan yang direncanakan, dan juga untuk mengetahui kekurangan dan kesalahan pada aplikasi.

4.3.3.1 Data Hasil Pengujian

Berikut merupakan data hasil pengujian Aplikasi yang dibuat dengan menggunakan metode *Black Box*:

a. Pengujian Menu Awal

Tabel 4.6 Pengujian Menu Awal

| | |
|-----------------------|--|
| Modul Yang di Uji | Tampilan Menu Awal |
| Prosedur Pengujian | Akses ke Manu Awal <i>Smartphone android</i> , pilih <i>icon</i> Aplikasi Pemetaan Lokasi Quarry |
| Hasil Yang diharapkan | Sistem menampilkan halaman Menu Awal |
| Hasil pengujian | Sesuai dengan yang diharapkan |
| Keterangan | Valid |

b. Pengujian *Button* Pilih Jenis Quarry

Tabel 4.7 Pengujian *Button* Pilih Jenis Quarry

| | |
|-----------------------|--|
| Modul Yang di Uji | <i>Button</i> Pilih Jenis Quarry |
| Prosedur Pengujian | Akses ke Manu Awal, pilih <i>Button</i> Pilih Jenis Quarry |
| Hasil Yang diharapkan | Sistem Menampilkan Menu Pilih |
| Hasil pengujian | Sesuai dengan yang diharapkan |
| Keterangan | Valid |

c. Pengujian *Button* Lokasi Quarry Pasir Kwarsa

Tabel 4.8 Pengujian *Button* Lokasi Quarry Pasir Kwarsa

| | |
|--------------------|--|
| Modul Yang di Uji | <i>Button</i> Lokasi Quarry Pasir Kwarsa |
| Prosedur Pengujian | Pilih <i>button</i> Lokasi Quarry Pasir Kwarsa yang berada di Menu Pilih |

| | |
|-----------------------|---|
| Hasil Yang diharapkan | Sistem menampilkan <i>Layout Map Fragment</i> yang dapat menampilkan peta dan <i>button</i> Site 1, Site 2, Site 3, Site 4. |
| Hasil pengujian | Sesuai dengan yang diharapkan |
| Keterangan | Valid |

d. Pengujian Peta Quarry Pasir Kwarsa

Tabel 4.9 Pengujian Peta Quarry Pasir Kwarsa

| | |
|-----------------------|---|
| Modul Yang di Uji | Peta Quarry Pasir Kwarsa |
| Prosedur Pengujian | Pilih <i>button</i> Lokasi Quarry Pasir Kwarsa yang berada di Menu Pilih |
| Hasil Yang diharapkan | Sistem menampilkan peta dan 4 buah <i>Button</i> , dimana masing-masing <i>button</i> jika dipilih akan menampilkan <i>marker</i> titik koordinat didalam peta sesuai dengan <i>button</i> yang dipilih |
| Hasil pengujian | Sesuai dengan yang diharapkan |
| Keterangan | Valid |

e. Pengujian *Button* Lokasi Quarry Batu Granit

Tabel 4.10 Pengujian *Button* Lokasi Quarry Batu Granit

| | |
|-----------------------|---|
| Modul Yang di Uji | <i>Button</i> Lokasi Quarry Batu Granit |
| Prosedur Pengujian | Pilih <i>button</i> Lokasi Quarry Batu Granit yang berada di Menu Pilih |
| Hasil Yang diharapkan | Sistem Menampilkan peta beserta 5 buah <i>button</i> |
| Hasil pengujian | Sesuai dengan yang diharapkan |
| Keterangan | Valid |

f. Pengujian Peta Quarry Batu Granit

Tabel 4.11 Pengujian Peta Quarry Batu Granit

| | |
|-------------------|-------------------------|
| Modul Yang di Uji | Peta Quarry Batu Granit |
|-------------------|-------------------------|

| | |
|-----------------------|---|
| Prosedur Pengujian | Pilih <i>button</i> Lokasi Batu Granit yang berada di Menu Pilih |
| Hasil Yang diharapkan | Sistem menampilkan peta dan 5 buah <i>Button</i> , dimana masing-masing <i>button</i> jika dipilih akan menampilkan <i>marker</i> titik koordinat didalam peta sesuai dengan <i>button</i> yang dipilih |
| Hasil pengujian | Sesuai dengan yang diharapkan |
| Keterangan | Valid |

g. Pengujian *Button* Lokasi Quarry Tanah Uruk

Tabel 4.12 Pengujian *Button* Lokasi Quarry Tanah Uruk

| | |
|-----------------------|--|
| Modul Yang di Uji | <i>Button</i> Lokasi Quarry Tanah Uruk |
| Prosedur Pengujian | Pilih <i>Button</i> Lokasi Quarry Tanah Uruk yang berada di Menu Pilih |
| Hasil Yang diharapkan | Sistem menampilkan peta beserta 3 buah <i>button</i> |
| Hasil pengujian | Sesuai dengan yang diharapkan |
| Keterangan | Valid |

h. Pengujian peta Quarry Tanah Uruk

Tabel 4.13 Pengujian Peta Quarry Tanah Uruk

| | |
|-----------------------|---|
| Modul Yang di Uji | Peta Quarry Tanah Uruk |
| Prosedur Pengujian | Pilih <i>Button</i> Lokasi Quarry Tanah Uruk yang berada di Menu Pilih |
| Hasil Yang diharapkan | Sistem menampilkan peta dan 3 buah <i>Button</i> , dimana masing-masing <i>button</i> jika dipilih akan menampilkan <i>marker</i> titik koordinat didalam peta sesuai dengan <i>button</i> yang dipilih |
| Hasil pengujian | Sesuai dengan yang diharapkan |
| Keterangan | Valid |

i. Pengujian *button* Lokasi Quarry Tanah Merah

Tabel 4.14 Pengujian *button* Lokasi Quarry Tanah Merah

| | |
|-----------------------|---|
| Modul Yang di Uji | <i>Button</i> Lokasi Quarry Tanah Merah |
| Prosedur Pengujian | Pilih <i>Button</i> Lokasi Quarry Tanah Merah yang berada di Menu Pilih |
| Hasil Yang diharapkan | Sistem akan menampilkan peta beserta 3 buah <i>button</i> |
| Hasil pengujian | Sesuai dengan yang diharapkan |
| Keterangan | Valid |

j. Pengujian Peta Quarry Tanah Merah

Tabel 4.15 Pengujian Peta Quarry Tanah Merah

| | |
|-----------------------|---|
| Modul Yang di Uji | Peta Quarry Tanah Merah |
| Prosedur Pengujian | Pilih <i>Button</i> Lokasi Quarry Tanah Merah yang berada di Menu Pilih |
| Hasil Yang diharapkan | Sistem menampilkan peta dan 3 buah <i>Button</i> , dimana masing-masing <i>button</i> jika dipilih akan menampilkan <i>marker</i> titik koordinat didalam peta sesuai dengan <i>button</i> yang dipilih |
| Hasil pengujian | Sesuai dengan yang diharapkan |
| Keterangan | Valid |

k. Pengujian *Button* Back

Tabel 4.16 Pengujian *button* Back

| | |
|-----------------------|--|
| Modul Yang di Uji | <i>Button</i> Back |
| Prosedur Pengujian | Pilih <i>button</i> Pilih Jenis Quarry, pilih <i>button</i> back |
| Hasil Yang diharapkan | Sistem akan kembali ke menu sebelumnya |
| Hasil pengujian | Sesuai dengan yang diharapkan |
| Keterangan | Valid |

l. Pengujian *Button* Cari Rute

Tabel 4.17 Pengujian *button* Cari Rute

| | |
|-----------------------|---|
| Modul Yang di Uji | <i>Button</i> Cari Rute |
| Prosedur Pengujian | Akses ke Menu Awal, pilih <i>button</i> Cari Rute |
| Hasil Yang diharapkan | Sistem menampilkan peta |
| Hasil pengujian | Sesuai dengan yang diharapkan |
| Keterangan | Valid |

m. Pengujian Peta Rute

Tabel 4.18 Pengujian Peta Rute

| | |
|-----------------------|--|
| Modul Yang di Uji | Peta Rute |
| Prosedur Pengujian | Pilih <i>button</i> Cari Rute yang berada di Menu Awal |
| Hasil Yang diharapkan | Sistem menampilkan peta dengan titik koordinat yang ditandai dengan <i>marker</i> lokasi pengguna, jika peta di sentuh maka akan muncul <i>marker</i> baru, yang dimana akan muncul garis merah yang menghubungkan antara <i>marker</i> lokasi pengguna dengan <i>marker</i> pada peta yang muncul setelah sentuh. |
| Hasil pengujian | Sesuai dengan harapan |
| Keterangan | Valid |

n. Pengujian *Button Exit*

Tabel 4.19 Pengujian *button Exit*

| | |
|-----------------------|---|
| Modul Yang di Uji | <i>Button Exit</i> |
| Prosedur Pengujian | Akses ke Manu Awal, pilih <i>button Exit</i> |
| Hasil Yang diharapkan | Sistem akan kembali ke menu aplikasi pada android |
| Hasil pengujian | Sesuai dengan harapan |
| Keterangan | Valid |

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan penulis pada aplikasi pemetaan lokasi quarry menggunakan *google maps v2* untuk *smartphone* android, maka dapat diambil kesimpulan bahwa pada penelitian ini telah dibangun sebuah aplikasi pemetaan lokasi quarry dengan metode layanan berbasis lokasi pada *platform* berbasis android yang dapat menampilkan titik koordinat lokasi quarry yang telah ditandai dengan *marker* dari *map*, menampilkan rute menuju tempat yang diinginkan.

5.2 Saran

Penelitian yang dilakukan tidak lepas dari kekurangan dan kelemahan, oleh karena itu untuk kebaikan pengembangan aplikasi lebih lanjut, maka perlu diperhatikan beberapa hal, diantaranya:

1. Aplikasi ini perlu dikembangkan lagi dengan menggunakan *software* yang lebih baru, supaya *compatible* dengan *hardware/software* yang terbaru dan untuk mempermudah pengembangan aplikasi lebih lanjut.
2. Perlunya penambahan fitur dan fungsional pada aplikasi, sehingga informasi yang disampaikan kepada pengguna lebih lengkap.
3. Untuk pengembangan aplikasi selanjutnya diharapkan untuk memakai struktur *coding* yang lebih baru, supaya terhindar dari kesalahan atau *bug* pada aplikasi.

DAFTAR PUSTAKA

- Yuniar, S. *Semua Bisa Menjadi Programmer Android Case Study* Jakarta: PT Elex Media Komputindo, 2014.
- Muis, S., *Global Positioning System*, Graha Ilmu, Yogyakarta, 2012.
- Dokumentasi Google Maps API, From <https://developers.google.com/maps/documentation/android/>, diakses mulai tanggal 25 Juni 2015.
- Global Positioning System, http://en.wikipedia.org/wiki/Global_Positioning_System, diakses mulai tanggal 25 Juni 2015.
- National Coordination Office. 2007. GPS System, <http://www.gps.gov> diakses mulai tanggal 25 Juni 2015
- Suprianto. D, & R. Agustina. 2012. *Pemrograman Aplikasi Android*. Yogyakarta, MediaKom.
- Stefan Steiniger, Moritz Neun and Alistair Edwardes., 2005. *Foundations of Location Based Services*.
- [https://id.wikipedia.org/wiki/Eclipse_\(28perangkat_lunak\)](https://id.wikipedia.org/wiki/Eclipse_(28perangkat_lunak)), diakses tanggal 25 Juni 2015.
- Budi Darytamo dkk, 2007. <http://mtamim.files.wordpress.com/2010/05/bukupab.pdf>, diakses 25 Juni 2015
- Safaat, Nazrudin H. 2012. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung, Informatika.

Aris Pramono, Joko. 2013. RANCANG BANGUN APLIKASI TEXT TO SPEECH SEBAGAI ALAT BANTU PEMBELAJARAN BAHASA INGGRIS. YOGYAKARTA, Naskah Publikasi STMIK AMIKOM.

Lestari, Eny Wiji. 2011. Geografi 3 Untuk SMA/MA Kelas XII. CV Wilian.

Melly Ujutiurma Malau. 2011. Makalah Quarry, <http://tiurma-berbagikasihilmudanberkat.blogspot.com/2011/01/makalah-quary.html>
Diakses pada 27 Agustus 2015

Sutabri, Tata. 2012. Analisis Sistem Informasi. Yogyakarta, Andi.

Indarto. 2010. DASAR-DASAR SISTEM INFORMASI GEOGRAFIS. Jember, Jember University Pers.

Fowler, M. 2005. UML Distilled 3th ED., Panduan Singkat Bahasa Pemodelan Objek Standar. Yogyakarta, Penerbit Andi.

API. 2013, from PCMag: <http://www.pcmag.com/encyclopedia/term/37856/api>
Diakses pada 04 Agustus 2015

Svennerberg, R. 2010. Beginning Google Maps Api 3. New York, Paul Manning.