

BAB 2

TINJAUAN PUSTAKA

1.1 Penelitian Terkait

Penelitian yang serupa juga pernah dilakukan oleh Dodo Suratmo Institut Sains dan Teknologi Akprind Yogyakarta yang membahas tentang ‘Pembuatan Aplikasi Informasi Lokasi Perguruan Tinggi Di Yogyakarta berbasis GIS’ (2012), Menurut Dodo Suratmo pada penelitian yang dilakukan menyebutkan bahwa suatu aplikasi yang dapat di laksanakan dalam rangka memenuhi tuntutan dalam pencarian lokasi perguruan tinggi adalah dengan menerapkan sistem informasi geografis berbasis android, namun pada penelitian tersebut hanya sebatas pada pencarian lokasi tanpa terintegrasi langsung dengan peta dan hanya menampilkan lokasinya saja, sehingga pengguna harus kembali membuka peta untuk menemukan lokasi tersebut. [1] Serta penelitian yang dilakukan Fatimah Zahrah Universitas Gunadarma pada penelitiannya yaitu ‘Aplikasi Location Based Service Tentang Taman Kota Di DKI Jakarta Menggunakan Eclipse’ (2014), menyebutkan masalah penentuan lokasi taman kota di DKI Jakarta yang begitu banyak dan sangat luas, sehingga harus adanya aplikasi untuk membantu menemukan lokasi taman kota tersebut. Namun pada penelitian tersebut menurut saya masih kurang akurat karena sebagian taman kota di Jakarta sudah berubah fungsi dan tidak terawat lagi fungsinya sebagai paru – paru kota melainkan sudah menjadi lapak berjualan para pedagang. Diharapkan aplikasi ini bisa mempermudah dinas tata kota untuk membantu menemukan taman kota yang sudah beralih fungsi agar dapat diperbaiki lagi sebagaimana fungsinya. [2] Penelitian yang serupa juga dilakukan oleh I Wayan Eka Swastikayana dari Universitas Pembangunan Nasional ‘Veteran’ Yogyakarta yang membahas tentang ‘Sistem Informasi Geografis Berbasis Web Untuk Pemetaan Pariwisata Kabupaten Gianyar’ (2011), Menurut I Wayan Eka Swastikayana pada penelitian yang dilakukan menyebutkan bahwa suatu aplikasi berbasis web yang dapat di jalankan dalam memenuhi tuntutan dalam pemetaan pariwisata di Kabupaten Gianyar adalah dengan

menerapkan sistem informasi geografis berbasis website, namun pada penelitian tersebut hanya sebatas pada pemetaan lokasi pariwisata tanpa terintegrasi langsung dengan sebuah aplikasi. Namun diharapkan pembuatan SIG pariwisata berbasis web akan memudahkan masyarakat untuk mengaksesnya. [3]

Dari penelitian terkait tersebut dirangkumkan pada tabel dibawah ini :

Tabel 0.1 Penelitian Terkait

No	Nama Peneliti dan Tahun	Masalah	Metode	Hasil
1.	Dodo Suratmo, 2012	Tidak semua orang mengerti tentang perguruan tinggi di DIY terutama letak perguruan tinggi tersebut sehingga masyarakat masih kesulitan dalam melakukan pencarian tempat – tempat perguruan tinggi.	Metode Waterfall	Dengan adanya aplikasi tersebut diharapkan pengguna dapat lebih mudah dalam pencarian lokasi perguruan tinggi yang ada di Yogyakarta.
2	Fatimah Zahrah, 2014	Sangat terbatasnya taman kota atau ruang terbuka hijau (RTH) di DKI Jakarta menyulitkan masyarakat yang	Location Based Service (LBS) menggunakan Eclipse	Dapat mempermudah masyarakat dan dinas ruang dan tata kota untuk membantu menemukan

No	Nama Peneliti dan Tahun	Masalah	Metode	Hasil
		akan melakukakn aktivitas seperti berolahraga ataupun hanya bersantai menikmati weekend.		taman kota di DKI Jakarta agar bersama-sama merawat dan melestarikan taman kota yang sudah ada.
3	I Wayan Eka Swastikayana, 2011	Para wisatawan masih mengalami kesulitan untuk menentukan perencanaan perjalanan wisata karena gambaran daerah wisata tersebut tidak tersedia seperti visualisasi tempat, jarak antar daerah wisata serta jalan yang akan dilalui.	Metode Waterfall	Diharapkan pembuatan SIG pariwisata berbasis web akan memudahkan masyarakat tepatnya para wisatawan dalam menentukan perencanaan perjalanan wisata.

1.2 Sistem Informasi

1.2.1 Pengertian Sistem Informasi

Sistem yang menyediakan informasi untuk manajemen dalam mengambil keputusan, dimana pengertian dari sistem tersebut merupakan kombinasi dari

orang-orang, teknologi informasi dan prosedur-prosedur yang terorganisasi. Biasanya suatu perusahaan atau badan usaha menyediakan semacam informasi yang berguna bagi manajemen. [5] Dengan demikian, sistem informasi berhubungan dengan sistem data di satu sisi dan sistem aktivitas di sisi lain. Sistem informasi adalah suatu bentuk komunikasi di mana data yang mewakili dan diproses sebagai bentuk dari memori. Sistem informasi juga dapat disebut sebagai bahasa semi formal yang mendukung manusia dalam pengambilan tindakan dan keputusan.

1.2.2 Sistem Informasi Menurut Para Ahli

1. Menurut Mc Leod

Sistem informasi adalah sebuah sistem yang memiliki kemampuan untuk mengumpulkan informasi dari semua sumber dan menggunakan berbagai media untuk menampilkan informasi. [6]

2. Menurut O'Brien

Sistem informasi adalah kombinasi dari setiap unit dikelola orang, *hardware* (perangkat keras), *software* (perangkat lunak), jaringan komputer dan jaringan komunikasi data (komunikasi), serta *database* (basis data) yang mengumpulkan, mengubah, dan menyebarkan informasi tentang bentuk organisasi. [6]

3. Menurut Robert A. Leitch dan K. Roscoe Davis

Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi, mendukung operasi, kegiatan manajerial dan strategis dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan – laporan yang diperlukan. [6]

1.3 Java

1.3.1 Pengertian dan Sejarah Java

Java adalah bahasa pemrograman yang dapat dijalankan di beberapa perangkat elektronik seperti komputer, telepon genggam dan perangkat lain yang mendukung java. Java tidak tergantung dengan platform dimana ia dijalankan, dengan kata lain java bisa dijalankan di beragam sistem operasi seperti Windows, Linux, Macintosh dan lainnya, java dikenal pula dengan slogannya, "*Tulis sekali, jalankan di mana pun*". Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems dan saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa pemrograman java banyak mengadopsi sintaks yang terdapat pada C dan C++ namun dengan sintaks model objek yang lebih sederhana. Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web. Pengertian lainnya tentang java, Java adalah bahasa pemrograman tingkat tinggi yang berorientasi objek dan program java tersusun dari bagian yang disebut kelas. Kelas terdiri atas metode-metode yang melakukan pekerjaan dan mengembalikan informasi setelah melakukan tugasnya. Para pemrogram Java banyak mengambil keuntungan dari kumpulan kelas di pustaka kelas Java, yang disebut dengan *Java Application Programming Interface* (API). Kelas-kelas ini diorganisasikan menjadi sekelompok yang disebut paket (*package*). Java API telah menyediakan fungsionalitas yang memadai untuk menciptakan *applet* dan aplikasi canggih. Jadi ada dua hal yang harus dipelajari dalam Java, yaitu mempelajari bahasa Java dan bagaimana mempergunakan kelas pada Java API. Kelas merupakan satu-satunya cara menyatakan bagian eksekusi program, tidak ada cara lain. Pada Java program `javac` untuk mengkompilasi file kode sumber Java menjadi kelas-kelas *bytecode*. File kode sumber mempunyai ekstensi `*.java`. Kompilator `javac` menghasilkan file *bytecode* kelas dengan ekstensi `*.class`. Interpreter merupakan modul utama sistem Java yang digunakan aplikasi Java dan menjalankan program *bytecode* Java. [7]

Bahasa pemrograman Java terlahir dari The Green Project, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan Oak. Proyek ini dimotori oleh Patrick Naughton, Mike Sheridan, dan James Gosling, beserta sembilan pemrogram lainnya dari Sun Microsystems. Salah satu hasil proyek ini adalah maskot *Duke* yang dibuat oleh Joe Palrang. Pertemuan proyek berlangsung di sebuah gedung perkantoran *Sand Hill Road* di Menlo Park. Setelah era *Star Seven* selesai, sebuah anak perusahaan Tv kabel tertarik ditambah beberapa orang dari proyek The Green Project. Mereka memusatkan kegiatannya pada sebuah ruangan kantor di 100 Hamilton Avenue, Palo Alto. Perusahaan baru ini bertambah maju jumlah karyawan meningkat dalam waktu singkat dari 13 menjadi 70 orang. Pada rentang waktu ini juga ditetapkan pemakaian Internet sebagai medium yang menjembatani kerja dan ide di antara mereka. Pada awal tahun 1990-an, Internet masih merupakan rintisan, yang dipakai hanya di kalangan akademisi dan militer. Mereka menjadikan perambah (*browser*) Mosaic sebagai landasan awal untuk membuat perambah Java pertama yang dinamai Web Runner, terinspirasi dari film 1980-an, Blade Runner. Pada perkembangan rilis pertama, Web Runner berganti nama menjadi Hot Java. Pada sekitar bulan Maret 1995, untuk pertama kali kode sumber Java versi 1.0.2 dibuka. Kesuksesan mereka diikuti dengan untuk pemberitaan pertama kali pada surat kabar *San Jose Mercury News* pada tanggal 23 Mei 1995. Sayangnya terjadi perpecahan di antara mereka suatu hari pada pukul 04.00 di sebuah ruangan hotel Sheraton Palace. Tiga dari pimpinan utama proyek, Eric Schmidt dan George Paolini dari Sun Microsystems bersama Marc Andreessen, membentuk Netscape. Nama Oak, diambil dari pohon oak yang tumbuh di depan jendela ruangan kerja "Bapak Java", James Gosling. Nama Oak ini tidak dipakai untuk versi release Java karena sebuah perangkat lunak lain sudah terdaftar dengan merek dagang tersebut, sehingga diambil nama penggantinya menjadi "Java". Nama ini diambil dari kopi murni yang digiling langsung dari biji (kopi tubruk) kesukaan Gosling. Konon kopi ini berasal dari Pulau Jawa. Jadi nama bahasa pemrograman Java tidak lain berasal dari kata Jawa (bahasa Inggris untuk Jawa adalah Java). [7]

1.4 Android

Android adalah sistem operasi bergerak (*mobile operating system*) yang telah dimodifikasi mengadopsi sistem operasi linux, namun telah dimodifikasi. Google mengambil alih seluruh hasil kerja android termasuk tim yang mengembangkan sistem android pada tahun 2005 dari Android, Inc sebagai sebuah bagian strategi untuk mengisi pasar sistem operasi bergerak. [8]

Pengembang hanya berkonsentrasi pada aplikasinya saja, aplikasi tersebut bisa berjalan pada beberapa perangkat berbeda selama masih ditenagai oleh Android (pengembang tidak perlu mempertimbangkan kebutuhan jenis perangkatnya). Keuntungan utama dari Android adalah adanya pendekatan aplikasi secara terpadu.

1.4.1 Sejarah Android

Android awalnya dikembangkan oleh android inc, sebuah perusahaan yang membuat software untuk ponsel yang berada di Palo Alto, California Amerika Serikat. Di tahun 2005 Google membeli dan mengambil alih perusahaan yang waktu itu bernama android sebuah perusahaan yang berkecimpung di dunia mobile. Setelah diambil alih oleh Google, pada bulan November 2007 Google mengumumkan bahwa mereka sedang mengembangkan ponsel Google dengan OS (Operating System) mobile terbaru yang mereka beri nama Android. OS Android ini sebenarnya didasarkan pada kernel Linux dan dirancang digunakan oleh Open Handset Alliance yang terdiri dari sekelompok puluhan pembuat hardware, carrier dan perusahaan terkait perangkat mobile lainnya, diantaranya HTC, Google, Motorola, Intel, Nvidia, T-Mobile, Qualcomm, dan lain - lain. Open Handset Alliance dibentuk untuk mengembangkan Android yang bersifat open source yang sourcenya di berikan secara gratis bagi bagi para pengembang (developer). [8]

Meskipun para pengembang Android tersebut merupakan pakar – pakar teknologi yang sangat berpengalaman, Android Inc. dijalankan secara diam – diam dan hanya diungkapkan bahwa para pengembang tersebut sedang

berusaha menciptakan sebuah perangkat lunak yang dapat diperuntukkan untuk telepon seluler. Masih pada tahun yang sama, Andy Rubin kehabisan uang. Steve Perlman adalah seorang teman dekat Andy Rubin dan meminjaminya \$10.000 tunai serta menolak tawaran saham di perusahaan. Tujuan awal dari perkembangan tersebut pada mulanya diperuntukkan bagi kamera digital, namun disadari bahwa pasar dari kamera digital tidak terlalu besar potensinya dan pengembangan android lalu dialihkan pada pasar telepon pintar atau smartphone untuk menyaingi Symbian serta Windows Mobile (iPhone Apple pada saat itu belum dirilis).[8]

Google mengambil alih perusahaan Android Inc. pada tanggal 17 Agustus 2005 dan menjadikannya sebagai anak perusahaan yang dimiliki oleh Google. Pendiri Android Inc. yaitu Rubin, Miner, serta White tetap bekerja pada perusahaan tersebut setelah diakuisisi oleh Google. Di Google, tim yang dipimpin oleh Andy Rubin mulai mengembangkan sebuah platform perangkat seluler dengan kernel linux.

Sejak tahun 2008, Android mulai secara bertahap melakukan sejumlah pembaharuan atau *update* untuk meningkatkan kinerja dari sistem operasi tersebut dengan menambahkan fitur – fitur baru, memperbaiki bug pada versi android yang sebelumnya. Setiap versi yang dirilis dinamakan secara alfabetis dengan berdasarkan nama makanan ringan yang biasa kita jumpai sehari – hari. [8]

1.4.2 Perkembangan Android

Penamaan versi android selalu menggunakan nama makanan ringan dan diawali dengan abjad yang berurutan seperti berikut :

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 (Honeycomb)
8. Android versi 4.0 (Ice Cream Sandwich)
9. Android versi 4.1 (Jelly Bean)
10. Android versi 4.4 (Kitkat)
11. Android versi 5.0 (Lolipop)

1.4.3 Fitur Yang Tersedia Di Android

Android mendukung fitur – fitur berikut ini :

1. Penyimpanan (*Storage*) menggunakan program SQLite yang merupakan database relasional yang ringan untuk menyimpan data.
2. Koneksi (*Connectivity*) mendukung jaringan GSM/EDGE, CDMA, IDEN, UMTS, EV-DO, *Bluetooth* (termasuk AVRCP dan A2DP), LTE, WiFi, dan WiMAX.
3. Pesan (*Messaging*) mendukung MMS dan SMS.
4. Web browser menggunakan *open-source* WebKit termasuk di dalamnya *engine* Chrome V8 JavaScript.
5. Media-media pendukung antara lain : H.263, H.264 (3GP atau MP4 container), AMR-WB (3GP container), AMR, MPEG-4 SP, AAC, HE-AAC (MP4 atau 3GP container), MIDI, MP3, Ogg Vorbis, JPEG, WAV, GIF, PNG, dan Bitmap.
6. *Hardware* terdapat *Accelerometer Sensor*, *Digital Compas*, *Camera*, *Proximity Sensor* dan GPS.
7. *Multi – touch* mendukung layar *multi – touch*.
8. *Multi – tasking* mendukung aplikasi *multi – tasking*.
9. Dukungan flash android 2.3 mendukung flash android 10.1

1.4.4 Arsitektur Android

Dalam sistem operasi Android terdiri dari beberapa unsur. Secara sederhana arsitektur android merupakan sebuah kernel linux dan sekumpulan pustaka C / C++ dalam suatu framework yang menyediakan dan mengatur alur proses aplikasi (Google IO, Android *Anatomy and Physiology*).

1.4.4.1 Linux Kernel

Android bukan merupakan linux tetapi android dibangun diatas linux kernel. Tingkat ini berisi semua *driver* perangkat rendah untuk komponen – komponen *hardware* perangkat android. Kernel juga bertugas sebagai lapisan abstrak antara software dan hardware stucknya.

1.4.4.2 Libraries

Android menyertakan libraries C / C++ yang digunakan oleh beberapa komponen dari sistem android serta berisi semua kode program yang menyediakan layanan – layanan utama sistem operasi android. Beberapa inti libraries seperti tercantum di bawah ini:

1. Libraries media untuk pemutaran media music audio dan video.
2. Libraries untuk mamajemen tampilan.
3. Libraries Graphic mencakup SGL dan OpenGL untuk grafis 2D dan 3D.
4. Libraries SQLite untuk dukungan database.
5. Libraries SSL dan WebKit terintegrasi dengan Web browsir dan security.
6. Libraries live WebCore mencakup modern Web browser dengan engine embeded web view.
7. Libraries 3D yang mencakup penerapan OpenGL ES 1.0 API's.

1.4.4.3 Android Runtime

Android terdiri dari satu set core libraries yang menyediakan sebagian besar fungsi yang sama yang kedudukannya setingkat dengan bahasa pemrograman java. *Android runtime* menyediakan kumpulan pustaka inti yang dapat diaktifkan oleh pengembang untuk menulis kode aplikasi android dengan bahasa pemrograman java. *Dalvik Virtual Machine* aktif setiap kali aplikasi Android berproses (aplikasi Android sikompilasi menjadi Dalvik Executable). Dalvik adalah mesin semu yang dirancang khusus untuk android yang dapat mengoptimalkan daya battery perangkat bergerak dengan memori dan CPU terbatas. [8]

1. *Application Framework* adalah semacam kumpulan *class built-in* yang tertanam dalam sebuah sistem operasi android sehingga pengembang dapat memanfaatkannya untuk aplikasi lain yang sedang dibangun.
2. *Applications* pada tingkat inilah kita akan bekerja, contoh aplikasi ini banyak ditemui, seperti : *Phone, Contact, Browse*, dan lain-lain. Seperti aplikasi android pada umumnya yang bisa di *download* dan di *install* dari market android. Semua aplikasi yang dibuat terletak pada tingkat *Applications*.

1.4.5 Komponen Aplikasi

Komponen aplikasi merupakan bagian penting dari sebuah aplikasi Android. Setiap komponen mempunyai fungsi yang berbeda, dan antara komponen satu dan lainnya bersifat saling melengkapi. Berikut ini empat komponen yang harus kita ketahui, yaitu :

1. *Activities*

Activity merupakan satu halaman antarmuka yang bisa digunakan oleh user untuk berinteraksi dengan aplikasi. Biasanya dalam satu activity terdapat Button, Spinner, ListView, EditText, dan sebagainya. Dalam satu aplikasi bisa terdiri atas lebih dari satu *activity*.

2. *Services*

Merupakan komponen aplikasi yang bisa berjalan secara *background* misalnya digunakan untuk memuat data dari *server database*. Selain itu, aplikasi *music player* atau radio juga memanfaatkan *services* supaya aplikasinya bisa tetap berjalan meskipun *user* melakukan aktivitas dengan aplikasi lain.

3. *Content Provider*

Komponen ini digunakan untuk mengelola data sebuah aplikasi, misalnya kontak telepon. Siapa pun bisa membuat aplikasi Android dan bisa mengakses kontak yang tersimpan pada sistem Android. Nah, agar bisa mengakses kontak, maka kita memerlukan komponen *Content Provider*.

4. *Broadcast Receiver*

Fungsi komponen ini sama seperti bahasa terjemahannya, yaitu penerima pesan. Misalnya pada kasus baterai lemah. Sistem Android dirancang

menyampaikan “pengumuman” secara otomatis jika baterai habis. Apabila aplikasi yang dibuat dilengkapi dengan komponen *Broadcast Receiver*, maka kita bisa mengambil tindakan seperti menyimpan kemudian menutup aplikasi atau tindakan yang lain. [9]

1.4.6 Software Pendukung

Dalam membangun suatu aplikasi dibutuhkan beberapa *software* pendukung yang harus ter-*install* sebelum mengembangkan aplikasi Android, antara lain : (Suprianto, Dodit, & Agustina, Rini, 2012).

1. *Android Software Development Kit (SDK)*

SDK Android berisi *debugger*, *library*, *emulator*, dokumentasi, contoh kode program dan tutorial. SDK android adalah mesin utama untuk mengembangkan aplikasi android.

2. *Android Development Tools (ADT) Plugins*

Plugins ADT berguna sebagai pengenalan Android di dalam IDE Eclipse. Dengan ADT *plugins* kita bisa membuat *project* aplikasi android baru, mengakses *tools* emulator dan perangkat Android , melakukan kompilasi dan men-*debug* aplikasi, mengeksport aplikasi ke *Android Packages (APK)*, membuat sertifikasi digital terhadap kode program APK. [9]

1.4.6.1 Instalasi Android Studio

Tempatkan android Studio setelah di ekstrak pada satu tempat di partisi yang ada, misalnya di D:\Project\android-sdk-windows\. Juga bisa diganti nama dari folder SDK dan menempatkannya di partisi lain, misal di C:\Android\android-sdk-windows\.

Langkah – langkah instalasi Android Studio :

1. Setting android Studio tempatkan pada PATH JDK, yaitu dengan langkah sebelumnya ketika konfirmasi PATH pada java.
2. Klik environment variable, sehingga akan terlihat setting PATH java. Klik dua kali pada variable PATH kemudian masukkan PATH android SDK,

dipisahkan dengan tanda titik koma. Misal : C:\Program Files\Java\jdk1.7.0\bin;
C:\Users\Edward\Document\android-sdk-windows.

Langkah – langkah instalasi Eclipse dan ADT :

1. Ekstrak paket zip eclipse.
2. Masuk dalam folder eclipse yang sudah di ekstrak sebelumnya. Lalu jalankan program Eclipse.exe (*eclipse tidak memerlukan instalasi*).
3. Ketika dijalankan eclipse akan membutuhkan ruang kerja (*workspace*) untuk menyimpan project yang akan dibuat, bisa menggunakan tempat default yang diarahkan oleh Eclipse. Atau juga bisa membuat folder sendiri, misal D:\Project\Workspace\. Pilih checklist *Use this as default and do not ask again*, agar folder tersebut menjadi default workspace dan tidak akan muncul ketika menjalankan eclipse.
4. Ketika pertama kali dijalankan eclipse akan terlihat jendela *opening welcome*.
5. Terdapat beberapa ikon yang bisa dipilih, yaitu *overview, what's new samples, dan tutorial*. Dengan begitu dapat dipelajari lebih dalam lagi tentang eclipse maupun java, sedangkan unruk memulai menggunakan elipse dapat dimulai dengan menu ikon workbench. [9]

Mengkonfigurasi Eclipse dan ADT :

1. Klik pada menu windows preference, sehingga muncul jendela preference, pilih pada android kemudian browse pada SDK location sesuai dengan folder dimana SDK ditempatkan. Klik apply, tunggu sebentar sampai muncul list SDK, klik ok.
2. Kemudian ke menu utama kemudian pilih android SDK dan AVD manager pada menu windows, klik sehingga muncul jendela android SDK, pilih pada AVD kemudian klik pada start.
3. Maka akan muncul AVD (*android virtual device*).
4. Android akan loading beberapa saat, sampai akhirnya AVD telah siap. Klik pada tombol menu untuk masuk ke dalam program explorer.

5. Apabila ingin memperbarui plugin, harus mengulangi membuat AVD dengan plugin yang baru.

Perlu diketahui bahwa fitur “android editor” ADT memerlukan beberapa komponen opsional eclipse. Jika menemukan kesalahan ketika menginstal ADT, beberapa komponen mungkin tidak ada pada eclipse sehingga salah satu solusinya adalah update ulang dengan koneksi internet.

1.4.7 Struktur Proyek Android

Saat akan membuat aplikasi android, terdapat banyak folder, untuk mengenal dan mengetahui fungsi dan isi masing – masing folder butuh waktu sesaat. Hal tersebut sangatlah penting karena semua aplikasi yang dibuat nanti berhubungan dengan susunan folder – folder tersebut, antaranya adalah src, gen, android version (versi andriod yang akan digunakan), assets, bin, res.

Penjelasan mengenai folder – folder tersebut sebagai berikut :

1. Src

Folder src mengandung *source code* dan *packagenya* yang kita gunakan untuk membuat aplikasi. Secara default, saat kita membuat proyek baru aplikasi Android, kita akan disuguhkan sebuah *activity* dan kita harus memberikan nama/class untuk *activity* tersebut.

2. Gen

Folder bin berisikan hasil compile aplikasi, pada folder inilah file .apk diletakkan. File apk (*Android Package*) inilah yang nantinya kita install pada Real HP Android.

3. Assets

Assets adalah folder yang dapat dimanfaatkan oleh programmer untuk menyimpan file audio, video, file html, gambar dan file bantu lainnya. Biasanya file assets ini banyak dipakai oleh aplikasi game.

4. Res

Folder res berisikan *resource* yang digunakan untuk menyusun aplikasi.

Folder ini berisikan beberapa folder penting. Secara default folder tersebut diantaranya adalah sebagai berikut:

Sub folder tersebut sebagai berikut :

a. *Drawable-hdpi, drawable-idpi, drawable-mdpi*

Folder ini berisi ikon – ikon atau bitmap yang bisa anda buat agar nantinya Android secara otomatis menyesuaikan tampilan aplikasi anda berdasarkan *device* yang pengguna gunakan.

b. *Layout*

Folder yang berisi activity dengan ekstensi .xml. File ini merupakan interface dari aplikasi android yang dibuat.

c. *Menu*

Pada folder ini berisi file .xml yang digunakan untuk membuat *Actionbar*. Semua action button didefinisikan pada file .xml di dalam folder menu.

d. *Values*

Secara default, pada folder – folder ini terdapat berbagai file .xml yaitu :

- 1) *dimens.xml*, di dalam file *dimens.xml*, terdapat string-string untuk konfigurasi dimensi aplikasi.
- 2) *strings.xml*, file *strings.xml* mirip dengan *dimens.xml*, hanya saja pada file *strings.xml* berisi variable atau item yang memuat string.
- 3) *styles.xml*, pada file ini terdapat pada deklarasi variable untuk tema. [9]

1.5 Rekayasa Perangkat Lunak

1.5.1 Pengertian Perangkat Lunak

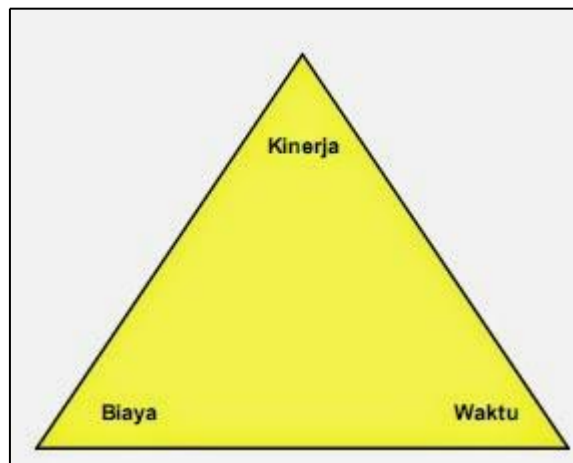
Istilah Rekayasa Perangkat Lunak secara umum disepakati sebagai terjemahan dari istilah *software engineering*. Istilah Software Engineering mulai di perkenalkan pada tahun 1968 pada *software engineering conference* yang diselenggarakan oleh NATO. Sebagian orang mengartikan RPL hanya sebatas pada bagaimana membuat suatu program komputer. Padahal ada perbedaan yang mendasar antara perangkat lunak (software) dan program komputer. [10]

Perangkat lunak adalah seluruh perintah yang digunakan untuk memproses informasi. Perangkat lunak dapat berupa program atau prosedur. Program adalah kumpulan perintah yang dimengerti oleh komputer sedangkan prosedur adalah perintah yang dibutuhkan oleh pengguna dalam memproses informasi (O'Brien, 1999). [10]

RPL sendiri adalah ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal yaitu analisa kebutuhan pengguna, menentukan spesifikasi dari kebutuhan pengguna, desain, pengkodean, pengujian sampai pemeliharaan sistem setelah digunakan. Dari pengertian ini jelaslah bahwa RPL tidak hanya berhubungan dengan bagaimana cara pembuatan program komputer. Pernyataan "semua aspek produksi" pada pengertian di atas, mempunyai arti semua hal yang berhubungan dengan proses produksi seperti manajemen proyek, penentuan personil, anggaran biaya, metode, jadwal, kualitas sampai dengan pelatihan pengguna merupakan bagian dari RPL.

1.5.2 Tujuan Rekayasa Perangkat Lunak

Secara umum tujuan RPL tidak jauh berbeda dengan bidang rekayasa yang lain. Hal tersebut dapat kita lihat pada gambar di bawah ini. [10]



Gambar 2.1 Tujuan RPL

Dari gambar di atas dapat disimpulkan bahwa bidang rekayasa akan selalu menghasilkan output yang kinerjanya tinggi, biaya rendah serta waktu penyelesaian yang tepat. Secara lebih khusus dapat dinyatakan tujuan RPL adalah:

1. Memperoleh biaya produksi perangkat lunak yang rendah
2. Menghasilkan perangkat lunak yang kinerjanya tinggi, bagus dan tepat waktu
3. Menghasilkan perangkat lunak yang bisa bekerja pada berbagai jenis platform
4. Menghasilkan perangkat lunak yang biaya dalam perawatannya rendah [10]

1.6 Perancangan Sistem

1.6.1 Metode Agile

Agile methods merupakan salah satu dari beberapa metode yang digunakan dalam pengembangan software. Agile method adalah jenis pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dan pengembang terhadap perubahan dalam bentuk apapun. [10] Dalam Agile Software Development interaksi dan personel lebih penting dari pada proses dan alat, software yang berfungsi lebih penting daripada dokumentasi yang lengkap, kolaborasi dengan klien lebih penting daripada negosiasi kontrak, dan sikap tanggap terhadap perubahan lebih penting daripada mengikuti rencana. *Agile Method* juga dapat diartikan sekelompok metodologi pengembangan software yang didasarkan pada prinsip-prinsip yang sama atau pengembangan sistem jangka pendek yang memerlukan adaptasi cepat dari pengembang terhadap perubahan dalam bentuk apapun.

1.6.2 Prinsip Agile Software Development

Agile Software Development juga melihat pentingnya komunikasi antara anggota tim, antara orang-orang teknis dan *businessmen*, antara *developer* dan managernya. Ciri lain adalah klien menjadi bagian dari tim pembangun software. Ciri-ciri ini didukung oleh 12 prinsip yang ditetapkan oleh Agile Alliance. [10] Menurut Agile Alliance, 12 prinsip ini adalah bagi mereka yang ingin berhasil dalam penerapan *Agile Software Development* :

1. Prioritas tertinggi adalah memuaskan pelanggan melalui penyerahan awal dan berkelanjutan perangkat lunak yang bernilai.
2. Menerima perubahan requirements meskipun perubahan tersebut diminta pada akhir pengembangan.
3. Memberikan perangkat lunak yang sedang dikerjakan dengan sering, beberapa minggu atau beberapa bulan, dengan pilihan waktu yang paling singkat.
4. Pihak bisnis dan pengembang harus bekerja sama setiap hari selama pengembangan berjalan.
5. Bangun proyek dengan individu-individu yang bermotivasi tinggi dengan memberikan lingkungan dan dukungan yang diperlukan, dan mempercayai mereka sepenuhnya untuk menyelesaikan pekerjaannya.
6. Metode yang paling efektif dan efisien dalam menyampaikan informasi kepada tim pengembangan adalah dengan komunikasi langsung face to face.
7. Perangkat lunak yang dikerjakan merupakan pengukur utama kemajuan.
8. Proses agile memberikan proses pengembangan yang bisa ditopang. Sponsor, pengembang, dan user harus bisa menjaga ke konstanan langkah yang tidak pasti.
9. Perhatian yang terus menerus terhadap rancangan dan teknik yang baik meningkatkan agility.
10. Kesederhanaan seni untuk meminimalkan jumlah pekerjaan adalah penting.
11. Arsitektur, requirements, dan rancangan terbaik muncul dari tim yang mengatur sendiri.
12. Pada interval reguler tertentu, tim merefleksikan bagaimana menjadi lebih efektif, kemudian menyesuaikannya. [11]

1.6.3 Model Metode Agile

1. Extreme Programming (XP)
2. Adaptive Software Development (ASD)
3. Dynamic Systems Development Method (DSDM)
4. Scrum Methodology
5. Crystal
6. Feature Driven Development (FDD)
7. Agile Modeling (AM)
8. Rational Unified Process

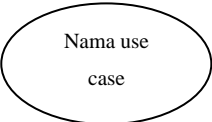
1.6.4 UML (*Unified Modeling Language*)

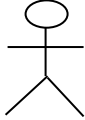
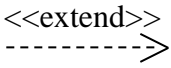
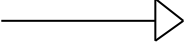
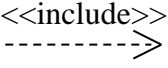
Perancangan sistem pada tugas akhir ini dilakukan dengan pembuatan perancangan dengan menggunakan UML (*Unified Modeling Language*). UML merupakan bahasa visual pemodelan untuk komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasi, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. [11] Untuk membuat suatu model, digunakan beberapa diagram dalam pembangunan aplikasi berorientasi objek, yaitu :

1. *Use Case Diagram*

Use Case Diagram merupakan pemodelan untuk kelakuan (*Behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan suatu interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Berikut adalah simbol-simbol yang ada pada diagram *use case* :

Tabel 2.2 Simbol-simbol *Use Case Diagram*

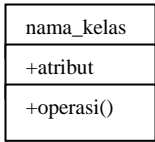


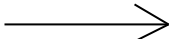
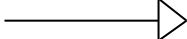
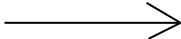
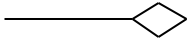
SIMBOL	NAMA	DESKRIPSI
	<i>Use case</i>	Fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit atau aktor.

SIMBOL	NAMA	DESKRIPSI
	<i>Aktor / actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistim informasi yang akan dibuat itu sendiri.
	Asosiasi / <i>Association</i>	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi pada aktor.
	Ekstensi / <i>extend</i>	Relasi <i>use case</i> tambahan sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu.
	Generalisasi / <i>generalization</i>	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
	<i>Include</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

2. Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas – kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode. Berikut adalah simbol-simbol yang ada pada *class diagram* :

Tabel 2.3 Simbol - simbol *Class Diagram*

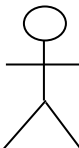

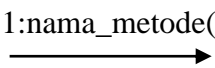
SIMBOL	NAMA	DESKRIPSI
	Kelas	Kelas pada struktur sistem.
	Antarmuka / <i>interface</i>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
	Asosiasi / <i>association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	Asosiasi berarah / <i>directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
	Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum- khusus).
	Kebergantungan / <i>dependency</i>	Relasi antar kelas dengan makna ketergantungan antar kelas.
	Agregasi / <i>aggregation</i>	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>).

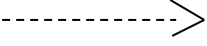
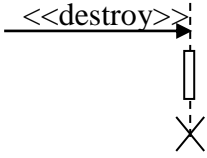



3. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antara objek. Dengan begitu menggambar diagram sekuen harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-

metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Berikut adalah simbol-simbol yang ada pada *sequence diagram* :

Tabel 2.4 Simbol - simbol *Sequence Diagram*


SIMBOL	NAMA	DESKRIPSI
	Aktor / <i>actor</i>	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri.
	Garis hidup / <i>lifeline</i>	Menyatakan kehidupan suatu objek.
	Objek	Menyatakan objek yang berinteraksi pesan.
	Waktu aktif	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini merupakan tahapan yang dilakukan didalamnya.
	Pesan tipe <i>create</i>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
	Pesan tipe <i>call</i>	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
	Pesan tipe <i>send</i>	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.

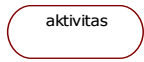



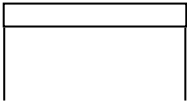
SIMBOL	NAMA	DESKRIPSI
	Pesan tipe <i>return</i>	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi / metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
	Pesan tipe <i>destroy</i>	Menyatakan suatu objek yang mengkhiri hidup objek lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .
	<i>Boundary</i>	Sistem yang merupakan alat penghubung yang saling berinteraksi dengan sistem pada suatu tingkatan layar.
	<i>Control</i>	Menggambarkan suatu pengendalian manajer atau kesatuan.
	<i>Entity</i>	Mekanisme yang menangkap pengetahuan atau informasi di dalam suatu sistem.

4. Activity Diagram

Activity diagram atau diagram aktifitas menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Berikut adalah simbol-simbol yang ada pada *activity diagram* :

Tabel 2.5 Simbol - simbol Activity Diagram

SIMBOL	NAMA	DESKRIPSI
	Status awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.

SIMBOL	NAMA	DESKRIPSI
	Aktivitas	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
	Percabangan / <i>decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
	Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
	Penggabungan / <i>join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabung menjadi satu.
	Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

1.7 Metode Pengujian Sistem

1.7.1 Black Box

Pengujian kotak hitam, juga disebut pengujian perilaku, berfokus pada persyaratan fungsional perangkat lunak. Artinya, teknik pengujian kotak hitam memungkinkan untuk membuat beberapa kumpulan kondisi masukan yang sepenuhnya akan melakukan semua kebutuhan fungsional untuk program. Pengujian kotak hitam merupakan pendekatan pelengkap yang mungkin dilakukan untuk mengungkap kelas kesalahan yang berbeda.[13] Pengujian kotak hitam berupaya untuk menemukan kesalahan dalam kategori berikut :

1. Fungsi yang salah/hilang serta kesalahan antarmuka
2. Kesalahan dalam struktur data atau akses basis data eksternal
3. Kesalahan perilaku atau kinerja
4. Kesalahan inisialisasi dan penghentian

