BAB II

LANDASAN TEORI

2.1 Perancangan

Perancangan suatu aplikasi termasuk dalam kegiatan rekayasa perangkat lunak. Proses rekayasa perangkat lunak dimulai jauh sebelum *coding* dilakukan dan berlanjut sampai tercapainya sebuah aplikasi yang diinginkan (Pohan, 1997). Pada dasarnya Rekayasa Perangkat Lunak dilakukan untuk merancang suatu aplikasi atau *software* dengan mengurutkan transformasi masalah menjadi solusi perangkat lunak yang dapat bekerja dengan baik.

Proses perancangan perangkat lunak merupakan serangkaian kegiatan dan hasil yang berhubungan dengan perangkat lunak, yang bertujuan untuk dihasilkannya suatu produk perangkat lunak. Walaupun ada banyak proses dalam perancangan suatu perangkat lunak, ada kegiatan-kegiatan mendasar yang umum bagi semua proses perancangan perangkat lunak (Sommerville,2003), antara lain:

1. Penspesifikasian Perangkat Lunak

   Fungsionalitas Perangkat Lunak dan batasan operasinya harus didefenisikan.

2. Perancangan dan Implementasi Perangkat Lunak

   Perangkat Lunak yang memenuhi persyaratan harus dibuat.

3. Validasi Perangkat Lunak

   Perangkat lunak tersebut harus divalidasi untuk menjamin bahwa perangkat lunak bekerja sesuai dengan apa yang diinginkan.

4. Pengevolusian Perangkat Lunak

   Perangkat Lunak harus dapat berkembang untuk menghadapi kebutuhan yang dapat berubah sewaktu-waktu.

2.2 Aplikasi

Aplikasi adalah penggunaan atau penerapan suatu konsep yang menjadi pokok pembahasan. Aplikasi dapat diartikan juga sebagai program komputer yang dibuat untuk menolong manusia dalam melaksanakan tugas atau pekerjaan tertentu. http://www.google.com/*Pengertian Aplikasi*/07 Januari 2012

Kamus Lengkap Dunia Komputer (2002:12) Aplikasi adalah suatu program computer yang dibuat untuk mengerjakan atau menyelesaikan masalah-masalah tertentu.

Menurut Edi Purnomo (2003:2) Aplikasi adalah sebuah kegiatan pengolahan data suatu urusan tertentu dari sebuah perusahaan.

Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak computer 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.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (*application suite*). Contohnya adalah *Microsoft Office* dan *OpenOffice.org*, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan setiap aplikasi. Sering kali, aplikasi ini memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

2.3 Perancangan Aplikasi

Perancangan aplikasi yaitu proses merancang piranti lunak sebelum melakukan pengkodean (*coding*). Model adalah deskripsi dari suatu permasalahan atau topic dari aplikasi yanga akan dibuat. Dengan menggunakan model ini akan membantu perancangan dalam memahami lingkup permasalahan yang akan dipecahkan. Model itu sendiri adalah *visualisasi* dari aplikasi yang akan dibangun. Model piranti lunak dapat dianalogikan seperti cetak biru pada suatu perancangan. Membuat model dari sebuah system yang kompleks sangatlah penting karena kita dapat memahami sistem tersebut secara menyeluruh.

2.4 Data dan Informasi

Untuk menuju pada pengertian Sistem Informasi secara utuh, diperlukan pemahaman yang tepat tentang konsep data dan informasi. Keterkaitan data dan informasi sangatlah erat sebagaimana antara sebab dan akibat. Bahwa data merupakan dasar dari sebuah informasi, sedangkan informasi merupakan elemen yang dihasilkan dari suatu bentuk pengolahan data.

2.4.1 Data

Banyak pengertian data yang dirangkum dari berbagai sumber. Teguh Wahyono dalam bukunya Sistem Informasi: Kosep Dasar, Analisis Desain, dan Implementasi menjelaskan beberapa pengertian data sebagai berikut :

1. Menurut berbagai kamus bahasa Inggris-Indonesia, data diterjemahkan sebagai istilah yang berasal dari kata *"datum"* yang berarti fakta atau bahan-bahan keterangan.
2. Dari sudut pandang bisnis, terdapat pengertian data bisnis sebagai berikut: *"Business data is an organization's description of thing (resources) and events (transactions) that it face"*. Jadi data, dalam hal ini disebut sebagai data bisnis, merupakan deskripsi organisasi tentang sesuatu (*resources*) dan kejadian (*transactions*) yang terjadi.
3. Pengertian yang lain mengatakan bahwa *"data is the description of things and events that we face"*. Jadi data merupakan deskripsi dari sesuatu dan kejadian yang dihadapi.

4. Gordon B. Davis dalam bukunya *Management Informations System: Conceptual Foundations, Structures, And Development* menyebutkan data sebagai bahan mentah dari informasi, yang dirumuskan sebagai sekelompok lambang-lambang tidak acak yang menunjukkan jumlah atau tindakan atau hal-hal lain.

Dapat diambil kesimpulan bahwa data adalah suatu bahan baku informasi, didefinisikan sebagai kelompok teratur simbol-simbol yang mewakili kuantitas, tindakan, benda, dan sebagainya. Data terbentuk dari karakter-karakter dapat berupa alfabet, angka, maupun symbol khusus seperti *, #, dan /. Data disusun untuk diolah dalam bentuk struktur data, struktur file, dan basis data.

2.4.2 Informasi

Menurut Gordon B. Davis dalam bukunya *Management Informations System: Conceptual Foundations, Structures, and Deveopment* menyebutkan informasi sebagai data yang telah diolah menjadi bentuk yang berguna bagi penerimanya dan nyata, berupa nilai yang dapat dipahami di dalam keputusan sekarang maupun masa yang akan datang.

Menurut Barry E. Cushing dalam buku *Accounting Information System And Business Organization*, dikatakan bahwa informasi merupakan sesuatu yang menunjukan hasil pengolahan data yang diorganisasikan dan berguna kepada orang yang menerimanya.

Menurut Robert N. Anthony dan John Dearden dalam buku *Management Control System*, menyebutkan informasi sebagai suatu kenyataan, data, item yang menambah pengetahuan bagi penggunanya.

Dapat disimpulkan bahwa informasi merupakan hasil dari pengolahan data menjadi bentuk yang lebih berguna bagi yang menerimanya yang menggambarkan suatu kejadian-kejadian nyata dan dapat digunakan sebagai alat Bantu untuk pengambilan suatu keputusan.

2.5 Lokasi Based Service (LBS)

Location Based Service (LBS) atau dalam bahasa indonesia diartikan sebagai Layanan Berbasis Lokasi adalah layanan informasi yang dapat diakses menggunakan piranti *mobile* melalui jaringan Internet dan seluler serta memanfaatkan kemampuan penunjuk lokasi pada piranti *mobile* (Virrantasu, et al, 2001). *Definisi Kedua*: Layanan IP nirkabel yang menggunakan informasi geografis untuk memberikan layanan informasi lokasi kepada pengguna. Beberapa layanan aplikasi yang memberikan petunjuk posisi/lokasi piranti *mobile* berada.

Terdapat empat komponen pendukung utama dalam teknologi Layanan Berbasis Lokasi, antara lain (Steiniger, et al 20):

1. Piranti Mobile

   Piranti Mobile adalah salah satu komponen penting dalam LBS. Piranti ini berfungsi sebagai alat bantu (*tool*) bagi pengguna untuk meminta

informasi. Hasil dari informasi yang diminta dapat berupa teks, suara, gambar dan lain sebagainya. Piranti *mobile* yang dapat digunakan bisa berupa PDA, *smartphone, laptop*. Selain itu, piranti *mobile* dapat juga berfungsi sebagai alat navigasi di kendaraan seperti halnya alat navigasi berbasis GPS.

2. Jaringan Komunikasi

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

3. Komponen *Positioning* (Penunjuk Posisi/Lokasi)

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

4. Penyedia layanan dan aplikasi

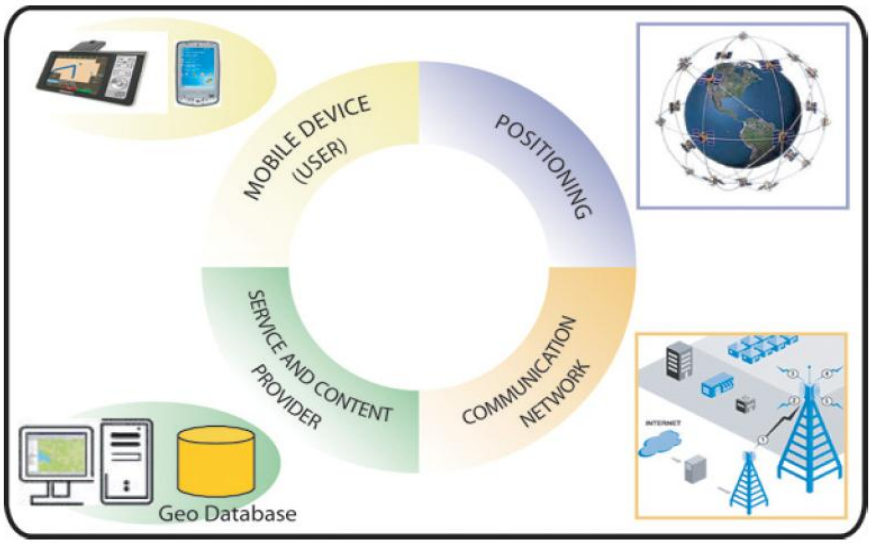
   Merupakan komponen LBS yang memberikan berbagai macam layanan yang bisa digunakan oleh pengguna. Sebagai contoh ketika pengguna meminta layanan agar bisa tahu posisinya saat itu, maka aplikasi dan

penyedia layanan langsung memproses permintaan tersebut, mulai dari menghitung dan menentukan posisi pengguna, menemukan rute jalan, mencari data di Yellow Pages sesuai dengan permintaan.

5. Penyedia data dan konten

   Penyedia layanan tidak selalu menyimpan seluruh data dan informasi yang diolahnya. Karena bisa jadi berbagai macam data dan informasi yang diolah tersebut berasal dari pengembang/pihak ketiga yang memang memiliki otoritas untuk menyimpannya. Sebagai contoh basis data geografis dan lokasi bisa saja berasal dari badan-badan milik pemerintah atau juga data-data perusahaan/bisnis/industri bisa saja berasal dari Yellow Pages, maupun perusahaan penyedia data lainnya.

Secara lengkap keempat komponen pendukung LBS tersebut dapat dilihat pada gambar berikut.



Gambar 2.1 Komponen pendukung utama teknologi LBS (Steiniger, et al 20)

2.6 Fasilitas Umum

Fasilitas umum adalah fasilitas yang disediakan untuk kepentingan umum, seperti jalan dan alat penerangan umum. Dalam hukum tata usaha negara, pengertian fasilitas umum adalah "barang yang dikuasai negara, dibiayai sebagian atau seluruhnya oleh anggaran dan belanja negara yang pemakaiannya atau peruntukkannya oleh pemerintah atau negara (bestemming atau bestimmung) bagi umum

2.7 Mobile

Mobile adalah perangkat elektronik yang berfungsi sebagaimana pesawat telepon normal, yang dapat bergerak pada suatu area yang luas.

Aplikasi bergerak merupakan istilah yang digunakan untuk suatu aplikasi yang dijalankan pada perangkat mobile yang cenderung memiliki ukuran kecil, portable dan dapat terkoneksi melalui jaringan nirkabel. Perangkat mobile memiliki banyak jenis dalam hal ukuran, desain dan layout, tetapi mereka memiliki kesamaan karakteristik yang sangat berbeda dari sistem desktop. http://www.google.com/*Pengertian Mobile/07 Maret 2012*

2.7.1 Aplikasi Mobile

Beberapa aplikasi yanga ada pada mobile :

1. Messaging : SMS, MMS, Instant Messaging
2. Mobile Transaction : SMS alert, MMS alert, report analisys
3. Mobile Workplace : Email, CRM, Instant Messaging

4. Mobile Music and Videos : Monotone, RTTTL, mp3, wav, mp4, screensaver, picture message, A2DP.
5. Mobile games : monochrome games, java games, symbian

2.8 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis *linux*. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. http://www.google.com/*Pengertian Android/07 April 2012*

Android merupakan subset perangkat lunak untuk perangkat *mobile* yang meliputi sistem operasi, *middleware* dan aplikasi inti yang di release oleh Google. Sedangkan Android SDK (*Software Development Kit*) menyediakan *tools* dan API yang diperlukan untuk mengembangkan aplikasi pada *platform* android dengan menggunakan bahasa pemrograman Java. (Mulyadi 2010)

Android adalah sistem operasi untuk telepon seluler yang berbasis *Linux*. Android menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola,Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode–kode Android di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan kedua adalah yang benar–benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD).

2.8.1 Perkembangan Android

http://www.google.com/*Perkembangan Android/07 Maret 2012*

1. **Android versi 1.1**

   Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, *voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

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

   Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (*Software Development Kit*) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video

ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

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

   Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indicator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus pada kamera, camcorder dan galeri yang dintegrasikan pada CDMA / EVDO, 802.1x, VPN, Gestures, dan *Textto- speech engine*. Kemampuan dial kontak teknologi *text to change speech* (tidak tersedia pada semua ponsel.

4. **Android versi 2.0/2.1 (Eclair)**

   Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan *Google Maps* 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, *digital Zoom*, dan Bluetooth 2.1. Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi *Mobile* terbaik (*killer apps* - aplikasi unggulan). Kompetisi ini berhadiah $25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang

tiap tahapnya dipilih 50 aplikasi terbaik. Dengan semakin berkembangnya dan semakin bertambahnya jumlah handset Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi Android adalah Shazam, Backgrounds, dan WeatherBug. Sistem operasi Android dalam situs Internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya oleh MySpace dan Facebook.

5. **Android versi 2.2 (Froyo: Frozen Yoghurt)**

   Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahanperubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, intergrasi V8 JavaScript *engine* yang dipakai Google Chrome yang mempercepat kemampuan *rendering* pada browser, pemasangan aplikasi dalam SD Card, kemampuan *WiFi Hotspot portabel*, dan kemampuan auto update dalam aplikasi Android Market.

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

   Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi *copy paste*, layar antar muka (*User Interface*) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru *(reverb, equalization,*

headphone virtualization, dan bass boost), dukungan kemampuan *Near Field Communication (NFC)*, dan dukungan jumlah kamera yang lebih dari satu.

7. Android versi 3.0 (Honeycomb)

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

2.8.2 Fitur Android

Fitur-fitur yang tersedia pada platform android adalah sebagai berikut:

1. Framework aplikasi

   Fitur ini mendukung penggantian komponen dan penggunaan kembali komponen yang sudah dibuat (reusable). Seperti pada umumnya, framework memiliki keuntungan dalam proses pengkodingan karena kita tidak perlu membuat kodingan untuk hal-hal yang pasti dilakukan seperti kodingan menampilkan gambar, kodingan konek database, dll.

2. Mesin Virtual Dalvik

   Sudah pernah dibhasa pada tulisan sebelumnya. Intinya dia adalah lingkungan dimana aplikasi android akan bekerja.

3. Integrated Browser

   berdasarkan open source engine WebKit.

4. Grafis

   Dengan ada fitur ini, kita bisa membuat aplikasi grafis 2D dan 3D karena android memiliki library opengGL ES 1,0

5. SQLite

   Nah, tugas dari fitur ini adalah berperan dalam penyimpanan data. Bahasa gampangnya sih dia itu sistem databasenya android

6. Media Support

   Fitur yang mendukung audio, video, dan gambar.

7. GSM Telephony

   Gak semua android punya fitur ini karena fitur ini tergantung dari smartphone yang dimiliki.

8. Bluetooth, EDGE, 3G, WiFi

   Fitur ini tidak selalu tersedia pada android karena tergantung hardware atau smartphone.

9. Dukungan Perangkat Tambahan

   Android dapat memanfaatkan kamera, layar sentuh, accelerometer, magnetometers, GPS, akselerasi 2D, dan Akselerasi 3D.

10. Multi- touch

    Kemampuan layaknya handset modern yang dapat menggunakan dua jari atau lebih untuk berinteraksi dengan perangkat.

11. Lingkungan Development

    Memiliki fitur emulator, tools untu debugging, profil dan kinerja memori. dan plugin untuk IDE Eclipse.

12. Market

    Seperti kebanyakan handphone yang memiliki tempat penjualan aplikasi, Market pada android merupakan katalog aplikasi yang dapat di-download dan di-install pada handphone melalui internet.

2.9 Java

Bahasa Java dikembangkan oleh *Sun Microsystem* tahun 1991 sebagai bagian dari suatu proyek penelitian untuk mengembangkan software bagi konsumer barang – barang elektronik seperti televise, *VCR*, *toaster* dan mesin – mesin lainnya yang dapat dibeli di swalayan. Tujuan penciptaan Java pada waktu itu adalah menjadi suatu program yang berukuran kecil, efisien, dan *portable* di segala jenis *hardware*. Tujuan yang sama ini membuat Java menjadi satu bahasa yang ideal untuk mendistribusikan program – program yang dapat dijalankan melalui *www* dan juga suatu bahasa pemrograman untuk segala tujuan untuk mengembangkan program – program yang dapat digunakan dengan mudah dan *portable* di berbagai *platform* yang berbeda.

Sekarang, *Sun* telah mengeluarkan berbagai program Java yang dapat digunakan seperti *Java API,* atau JDK atau *JAVA Developer Kit* . Selain itu, banyak juga program-program lain yang dapat digunakan untuk membuat

program Java, seperti *Eclipse, NetBeans, JBuilder, JCreator, J++,* dan sebagainya.

Saat ini distribusi Java dan kelas pendukungnya dibagi dalam tiga bagian yang masing-masing memiliki konsentrasi tersendiri yaitu:

1. *Java 2 Standart Edition* (J2SE), untuk aplikasi *desktop*
2. *Java 2 Enterprise Edition* (J2EE), untuk aplikasi *server*
3. *Java 2 Micro Edition* (J2ME), untuk piranti dengan kemampuan terbatas

2.9.1 Eclipse

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

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

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*.

2.9.2 XML

XML (*extensible markup language*) adalah bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membut dokumen markup keperluan pertukaran data antar system yang beraneka ragam. XML merupakan kelanjutan dari HTML (*hyper text markup language*) yang merupakan bahasa standar untuk melacak internet.

XML dan HTML yang dirancang dengan tujuan yang berbeda : XML dirancang untuk transportasi dan menyimpan data, dengan focus pada apa data, sedangkan HTML dirancang untuk menampilkan data, dengan focus pada bagaimana data terlihat.

2.9.3 ADT Plugin For Eclipse

Android Development Tools (ADT) adalah plugin untuk Eclipse IDE yang dirancang khusus untuk memberikan *integrated environment* yang kuat untuk membuat aplikasi Android. ADT memberikan kemampuan kepada Eclipse untuk membuat projek baru Android secara cepat, membuat aplikasi *User Interface*, menambahkan komponen berdasarkan Android *Framework* API, melakukan

debugging aplikasi yang di buat dengan menggunakan Android SDK *tools* dan bahkan melakukan distribusi aplikasi yang dibuat.

Pembuatan aplikasi android dengan Eclipse beserta ADT sangat dianjurkan karena merupakan cara tercepat untuk memulai membuat projek Android. Dengan disediakannya *project setup,* serta *tools* yang sudah terintegrasi di dalamnya, *custom XML editor*, dan *debugging* beserta *output* dalam *emulator* yang sudah disediakan Android SDK, mempermudah para pengembang aplikasi Android dalam pembuatan aplikasinya.

2.10 PHP (Personal Home Page Tool)

Personal home page tool (PHP) merupakan sebuah bahasa scripting yang dibundel dengan HTML, yang dijalankan disisi server. Sebagian besar perintahnya berasal dari C, Java dan beberapa lainnya dengan tambahan fungsi PHP. Bahasa ini memungkinkan para pembuat aplikasi web menyajikan halaman HTML, dinamis dan interaktif dengan cepat dan mudah, yang dihasilkan oleh server. (http://white2jasmine.wordpress.com/pengertian-php /14April2012)

2.11 Perangkat Pendukung

Perangkat lunak pendukung adalah perangkat lunak yang digunakan dalam merancang sistem dari mulai coding sampai implementasinya, salah satu tool / model untuk merancang pengembangan software yang berbasis object oriented adalah UML (*Unified Modeling Language*).

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (*Object-Oriented*). UML sendiri juga memberikan standar penulisan sebuah system blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponen-komponen yang diperlukan dalam sistem software.

UML (*Unified Modeling Language*) tidak hanya merupakan sebuah bahasa pemograman visual saja, namun juga dapat secara langsung dihubungkan ke berbagai bahasa pemograman, seperti JAVA, C++, Visual Basic, atau bahkan dihubungkan secara langsung ke dalam sebuah object-oriented database. Begitu juga mengenai pendokumentasian dapat dilakukan seperti; *requirements*, arsitektur, *design*, *source code*, *project plan*, *tests*, dan *prototypes*. Untuk dapat memahami UML membutuhkan bentuk konsep dari sebuah bahasa model, dan mempelajari 3 (tiga) elemen utama dari UML seperti *building block*, aturan-aturan yang menyatakan bagaimana *building block* diletakkan secara bersamaan, dan beberapa mekanisme umum (common).

2.12.1 Use Case Diagram

Sebuah *use case* menggambarkan suatu urutan interaksi antara satu atau lebih aktor dan sistem. Dalam fase *requirements*, model *use case* mengambarkan sistem sebagai sebuah kotak hitam dan interaksi antara aktor dan sistem dalam suatu bentuk naratif, yang terdiri dari input user dan respon-respon sistem. Setiap

use case menggambarkan perilaku sejumlah aspek sistem, tanpa mengurangi struktur internalnya. Selama pembuatan model use case secara pararel juga harus ditetapkan obyek-obyek yang terlibat dalam setiap use case.

2.12.2 Class Diagram

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

2.12.3 State Diagram

Diagram status atau *state diagram atau statechart diagram menunjukkan* kondisi yang dapat dialami atau terjadi pada sebuah objek sehingga setiap objek memiliki sebuah diagram status.

Diagram status diadopsi dari penggambaran kondisi mesin status (*state machine) yang menggambarkan* status apa saja yang dialami oleh mesin, misalnya mesin pembelian kopi dengan uang koin.

Diagram Status mengambarkan seluruh *state/status yang memungkinkan* obyek-obyek dalam *class dapat dimiliki dan kejadian-kejadian yang* menyebabkan status berubah.

2.12.4 Activity Diagram

Diagram aktivitas atau *activity diagram menggambarkan workflow (aliran* kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu

diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan actor.

Diagram aktivitas juga banyak digunakan untuk mendefinisikan :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokan tampilan dari sistem / *user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

2.12.5 Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Banyaknya diagram sekuen yang harus digambar adalah sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak.

2.12.6 Component Diagram

Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time*, *link time*, maupun *run time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

2.12.7 Deployment Diagram

Deployment/physical diagram menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, server atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi server, dan hal-hal lain yang bersifat fisikal Sebuah *node* adalah server, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya. Hubungan antar *node* (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.

2.12 Pengujian Sistem

Pengujian sistem bertujuan untuk menguji komponen sistem yang telah dirancang sebelumnya dan untuk memastikan bahwa setiap elemen dari sistem telah berfungsi sesuai dengan yang diharapkan.

2.12.1 Pengujian Alpha

Pengujian alpha merupakan pengujian fungsional yang diadakan di lingkungan pembangun oleh sekumpulan pengguna yang akan menggunakan perangkat lunaknya. Pihak pembangun mendampingi serta mencatat kesalahan-kesalahan maupun permasalahan yang dirasakan oleh pengguna. Pengujian alpha dilakukan untuk mengetahui apakah data yang dimasukkan sudah sesuai dengan yang diharapkan.

2.12.2 Pengujian Betha

Pengujian betha dilakukan dilingkungan pengguna tanpa kehadiran pihak pembangun aplikasi. Pengujian ini merupakan pengujian yang bersifat langsung di lingkungan yang sebenarnya. Pengguna melakukan penilaian terhadap aplikasi dengan menggunakan media kuisioner. Dari hasil kuisioner tersebut maka dapat ditarik kesimpulan apakah aplikasi yang dibangun telah sesuai dengan tujuannya.