

## BAB II

### LANDASAN TEORI

#### 1.1 *Knowledge Base System*

*Knowledge Based System* (KBS) atau sistem berbasis pengetahuan merupakan bagian dari Kecerdasan buatan / *Artificial Intelligence* (AI). KBS memiliki kemampuan untuk melakukan komputasi, penyimpanan, proses berfikir, dan penyimpanan pengetahuan. KBS dapat diimplementasikan untuk membantu pakar (*expert*) menjawab pertanyaan – pertanyaan tanpa menghabiskan waktu, dapat dilakukan dimanapun, dan kapanpun. Hal ini karena pengetahuan yang mereka miliki disimpan terlebih dahulu ke dalam *Knowledge Based* (Basis Pengetahuan).

KBS sendiri terdiri dari *Knowledge Based* (KB) dan mesin inferensi yang berfungsi sebagai mesin pencari pengetahuan (Sajja & Akerkar, 2010). KB sendiri dapat berupa repository pengetahuan dengan berbagai bentuk. KBS dapat berupa sistem yang pengetahuannya diupdate secara otomatis (*machine learning*) atau diupdate secara manual (*manual update*). *User interface* berguna sebagai penghubung antara sistem dan pengguna. menunjukkan arsitektur dasar dari *Knowledge Based System*.

Kelebihan dari KBS memiliki kelebihan dibandingkan dengan sistem komputer sederhana. KBS sangat bermanfaat pada situasi sebagai berikut :

1. Saat tidak ada pakar yang tersedia di suatu lokasi.

2. Ketika pengetahuan akan disimpan untuk kepentingan dimasa yang akan datang atau ketika pengetahuan akan dibagikan atau digandakan.
3. Ketika sistem penunjang keputusan cerdas dibutuhkan dalam pengambilan keputusan suatu permasalahan yang rumit dan kompleks.

KBS sendiri dapat diklasifikasikan ke dalam 5 tipe yaitu sistem pakar (*expert system*), *hypertext manipulation system*, *CASE Based reasoning*, *Database in conjunction with an intelligent User Interface*, dan *Intelligent Tutoring System (ITS)*.

## 1.2 Java

Definisi java menurut Sun Microsystem (perusahaan pemilik hak cipta java) adalah nama untuk sekumpulan teknologi yang membuat perangkat lunak dan menjalankan perangkat lunak pada komputer standalone ataupun pada lingkungan jaringan/internet. java berdiri di atas sebuah mesin interpreter yang diberi nama java Virtual Machine (JVM). JVM inilah yang akan membaca bytecode dalam file.class dari suatu program sebagai representasi langsung program yang berisi bahasa mesin. Oleh karena itu bahasa java disebut sebagai bahasa pemrograman yang portable, karena dapat dijalankan pada berbagai sistem operasi, asalkan pada sistem operasi tersebut terdapat JVM.

Pada tahun 2009 Oracle membeli Sun Microsystem yang berakibat secara tidak langsung java menjadi milik oracle secara penuh . java resmi diakuisi pada 27 Januari 2010.

Karakteristik java :

### 1. Sederhana

Sintaks java seperti sintaks pada C++ tetapi sintaks java tidak memerlukan header file, pointer arithmetic, struktur union, operator overloading, class virtual base, dan yang lainnya. Jika mengenal C++ dengan baik, maka pengguna dapat berpindah ke sintaks java dengan mudah.

## **2. Berorientasi Objek (Object Oriented)**

Berorientasi objek merupakan suatu teknik yang memusatkan rancangan pada data (objek) dan interface. Fasilitas pemrograman berorientasi objek pada java pada dasarnya adalah sama dengan C++. perbedaan utama antara java dengan C++ terletak pada penurunan berganda (multiple inheritance). java tidak mengenal multiple inheritance seperti pada C/C++. Multiple Inheritance membingungkan dan berakibat pada sulitnya pembuatan aplikasi. Sebagai gantinya java menggunakan interface.

## **3. Portabel**

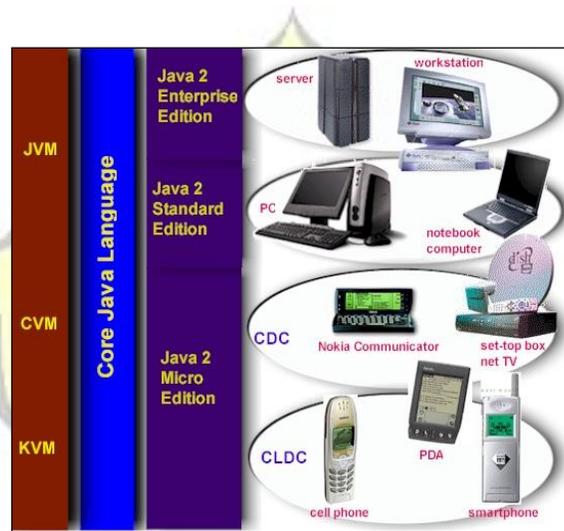
Java dapat digunakan pada segala macam arsitektur komputer dan perangkat karena sifatnya yang portable. Dapat dieksekusi di beragam platform tanpa harus melakukan perubahan kode secara menyeluruh. Sebagai contoh, aplikasi java yang dapat berjalan di windows maka juga dapat berjalan dengan baik di sistem operasi linux dan mac. Dengan hanya membutuhkan java virtual machine yang serupa tanpa melakukan perubahan pada kode aplikasi tersebut.

Platform java terdiri dari sekumpulan library, compiler, debugger dan alat lain yang dipaket dalam java development kit (JDK). Agar sebuah program java dapat dijalankan, maka file dengan ekstensi .java harus dikompilasi menjadi file bytecode. Untuk menjalankan file

bytecode tersebut dibutuhkan JRE (java runtime environment) yang memungkinkan pengguna untuk menjalankan program java. JRE terdiri dari JVM dan pustaka java yang digunakan.

Platform java memiliki tiga buah edisi yang berbeda :

1. J2EE (java 2 Enterprise Edition)
2. J2SE (java 2 Standard Edition)
3. J2ME (java 2 Micro Edition)



Gambar 2.1 Edisi java dan sasaran targetnya pada perangkat

KVM (K Virtual Machine) adalah : sebuah subset dari java virtual machine yang dioptimalisasikan untuk perangkat bermemori kecil dan perangkat berukuran kecil atau lebih dikenal dengan perangkat CLDC (Connected Limited Device Configuration). KVM support penggunaan memori hingga 512KB

CVM (C Virtual Machine) adalah : sebuah subset dari virtual machine yang dioptimalisasikan untuk perangkat-perangkat berukuran kecil yang penggunaan memorinya diatas +2MB atau lebih dikenal dengan perangkat CDC (Connected Device Configuration).

JVM (Java Virtual Machine) adalah: Java virtual machine yang merupakan pusat dari CVM dan KVM dan meliputi J2ME, J2SE dan J2EE dengan penerapan pada banyak perangkat dan cakupan support memori yang luas

### **1.3 Android**

Pengertian Android adalah sistem operasi berbasis Linux yang dipergunakan sebagai pengelola sumber daya perangkat keras, baik untuk ponsel, smartphone dan juga PC tablet. Secara umum Android adalah platform yang terbuka (*Open Source*) bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang digunakan oleh berbagai piranti bergerak. Telepon pertama yang memakai sistem operasi Android adalah HTC Dream, yang dirilis pada 22 Oktober 2008. Pada penghujung tahun 2009 diperkirakan di dunia ini paling sedikit terdapat 18 jenis telepon seluler yang menggunakan Android. Semenjak kehadirannya pada 9 Maret 2009, Android telah hadir dengan versi 1.1, yaitu sistem operasi yang sudah dilengkapi dengan pembaruan estetis pada aplikasinya, seperti jam alm, *voice search*, pengiriman pesan dengan Gmail, dan pemberitahuan email.

Pada perkembangannya, sistem operasi Android telah mengalami beberapa perubahan dan perbaikan. Dan yang paling menarik adalah versi keluaran Android yang diberi nama seperti nama-nama makanan.

Berikut merupakan beberapa versi dari Android:

- a. Versi 1.5 bernama Cupcake yang dirilis pada April 30, 2009

- b. Versi 1.6 bernama Donut yang dirilis pada September 15, 2009
- c. Versi 2.0–2.1 bernama Eclair yang dirilis pada October 26, 2009
- d. Versi 2.2 bernama Froyo yang dirilis pada May 20, 2010
- e. Versi 2.3–2.3.2 bernama Gingerbread yang dirilis pada December 6, 2010
- f. Versi 2.3.3–2.3.7 bernama Gingerbread yang dirilis pada February 9, 2011
- g. Versi 3.1 bernama Honeycomb yang dirilis pada May 10, 2011
- h. Versi 3.2 bernama Honeycomb yang dirilis pada July 15, 2011
- i. Versi 4.0.x bernama Ice Cream Sandwich yang dirilis pada December 16, 2011
- j. Versi 4.1.x bernama Jelly Bean yang dirilis pada July 9, 2012
- k. Versi 4.4 bernama Kitkat yang dirilis pada October 31, 2013
- l. Versi 5.0 bernama Lolipop yang dirilis pada tahun 2015
- m. Versi 6.0 bernama Marshmallow yang dirilis pada tahun 2015

Berikut ini adalah tabel perbandingan fitur-fitur android sesuai dengan versinya yaitu :

Tabel 2.1 Versi dan fitur Android

NO.	Versi dan Nama Anroid	Fitur-Fitur
1.	Android versi 1.1 1.1	<ul style="list-style-type: none"> <li>- Pembaharuan estetis pada aplikasi jam, alm, <i>voice search</i> (pencarian suara)</li> <li>- pengiriman pesan dengan Gmail, dan pemberitahuan email.</li> </ul>
2.	Android versi 1.5	<ul style="list-style-type: none"> <li>- Kemampuan merekam dan menonton</li> </ul>

	<p>Cupcake</p> 	<p>video dengan modus kamera</p> <ul style="list-style-type: none"> <li>- Meng-<i>upload video</i> ke Youtube</li> <li>- Gambar ke Picasa langsung dari telepon</li> <li>- Dukungan Bluetooth A2DP</li> <li>- Kemampuan terhubung secara otomatis ke <i>headset Bluetooth</i></li> <li>- Animasi layar, dan keyboard pada layar yang disesuaikan dengan system.</li> </ul>
3.	<p>Android versi 1.6</p> <p>Donut</p> 	<ul style="list-style-type: none"> <li>- Menampilkan proses pencarian yang lebih baik dibandingkan sebaliknya.</li> <li>- Penggunaan baterai indikator dan control applet VPN.</li> <li>- Galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus ; kamera, <i>camcorder</i> dan galeri yang diintegrasikan, CDMA/EVDO, 802.1x, VPN, Gestures, dan <i>Text-to-speech engine</i>, kemampuan <i>dial</i> kontak, teknologi <i>text to change speech</i> (tidak tersedia pada semua ponsel, pengadaan resolusi VWGA.)</li> </ul>
4.	<p>Android versi 2.0/2.1</p> <p>Eclair</p>	<ul style="list-style-type: none"> <li>- Pengoptimalan <i>Hardware</i></li> <li>- Peningkatan Google Maps 3.1.2</li> </ul>

		<ul style="list-style-type: none"> <li>- Perubahan UI dengan <i>browser</i> baru dan dukungan HTML5</li> <li>- Daftar kontak yang baru</li> <li>- Dukungan <i>Flash</i> untuk kamera 3,2 MP, <i>Digital Zoom</i>, dan <i>Bluetooth 2.1</i></li> </ul>
5.	<p>Android versi 2.2 Froyo (Frozen Youghurt)</p> 	<ul style="list-style-type: none"> <li>- Kerangka aplikasi memungkinkan penggunaan dan penghapusan komponen yang tersedia</li> <li>- <i>Dalvik Virtual Machine</i> dioptimalkan untuk perangkat <i>mobile</i></li> <li>- Grafik : grafik 2D dan grafis 3D berdasarkan <i>libraries</i> OpenGL</li> <li>- SQLite : untuk penyimpanan data</li> <li>- Mendukung media : <i>audio</i>, <i>video</i>, dan berbagai format gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)</li> <li>- GSM, <i>Bluetooth</i>, EDGE, 3G, dan WiFi (<i>hardware independent</i>) <i>accelerometer</i> (tergantung <i>hardware</i>)</li> </ul>
6.	<p>Android versi 2.3</p>	<ul style="list-style-type: none"> <li>- Hal-hal yang direvisi dari versi sebelumnya adalah kemampuan sebagai berikut:</li> </ul>

	<p>Gingerbread</p> 	<ul style="list-style-type: none"> <li>- <i>SIP-based VoIP</i></li> <li>- <i>Near Field Communication (NFC)</i></li> <li>- <i>Gyroscope</i> dan sensor</li> <li>- <i>Multiple cameras support</i></li> <li>- <i>Mixable audio effects</i></li> <li>- <i>Download Manager</i></li> </ul>
7.	<p>Android versi 3.0/3.1 Honeycomb</p> 	<ul style="list-style-type: none"> <li>- Android Honeycomb dirancang untuk tablet</li> <li>- Mendukung multiprocessor dan akselerasi untuk pengolahan grafis yang lebih baik.</li> </ul>
8.	<p>Android versi 4.0 Ice Cream Sandwich</p> 	<ul style="list-style-type: none"> <li>- <i>Lockscreen Dengan Face Detection</i></li> <li>- <i>Speak Your Words</i></li> <li>- Merekam <i>video</i> sekaligus Foto</li> <li>- <i>Advanced Call Option</i></li> <li>- Mengambil <i>Screenshot</i></li> <li>- Melakukan <i>Multitasking</i></li> <li>- Menghubungi rekan hanya dengan menggeser layar</li> <li>- <i>Private Browsing</i></li> <li>- Membuka <i>Gagged</i> dengan <i>account</i> Google</li> </ul>

		<ul style="list-style-type: none"> <li>- Melacak pengguna data</li> <li>- Transfer data dengan <i>Android Beam</i></li> <li>- <i>Widgets</i> yang mudah diatur</li> <li>- Cuaca hari ini pada layar</li> <li>- Hapus Notifikasi dengan geser ke kanan</li> <li>- <i>Keyboard Layouts</i> dalam berbagai bahasa</li> </ul>
9.	<p>Android versi 4.1-4.3</p> <p>Jelly Bean</p> 	<ul style="list-style-type: none"> <li>- Photo Sphere : memungkinkan pengguna mengambil foto dari setiap arah</li> <li>- Wireless Display : dapat menghubungkan perangkat Android dengan monitor lain atau TV tanpa menggunakan kabel</li> <li>- Notifikasi : memungkinkan pengguna untuk menindak lanjuti pemberitahuan atau notifikasi yang muncul hanya dengan sekali sentuh</li> </ul>
10.	<p>Android versi 4.4</p> <p><a href="#">KitKat</a></p>	<ul style="list-style-type: none"> <li>• Pembaruan antarmuka dengan bar status dan navigasi transparan pada layar depan.<sup>[114]</sup></li> <li>• Optimasi kinerja pada perangkat dengan spesifikasi yang lebih rendah</li> <li>• Kerangka kerja pencetakan</li> <li>• NFC Host Card Emulation sebagai</li> </ul>



		<p>emulator <a href="#">kartu pintar</a></p> <ul style="list-style-type: none"> <li>• WebViews berbasis Chromium</li> <li>• Perluasan fungsionalitas bagi layanan pendengar notifikasi</li> <li>• API umum untuk mengembangkan dan mengelola klien <a href="#">pesan teks</a>, kemampuan untuk menentukan aplikasi SMS standar.</li> <li>• Peningkatan tampilan mode layar penuh, tombol perangkat lunak dan status bar bisa diakses dari tepi dengan cara menggesek</li> <li>• Penyeimbang audio, pemantauan audio, dan peningkatan suara audio</li> <li>• Pererekam aktivitas layar yang terintegrasi</li> <li>• Peningkatan aksesibilitas API</li> <li>• Mesin virtual eksperimental baru, ART</li> <li>• Dukungan <a href="#">Bluetooth Message</a> <a href="#">Access Profile (MAP)</a></li> </ul>
11.	Android Versi 5.0 Lolipop	<ul style="list-style-type: none"> <li>• Desain antarmuka (tampilan) yang dinamakan "Material Design".</li> </ul>



		<ul style="list-style-type: none"> <li>• 64-bit ART compiler</li> <li>• Project volta, yang berguna untuk meningkatkan daya hidup baterai 30% lebih tahan lama.</li> <li>• ‘<i>factory reset protection</i>’. Fitur ini berguna ketika smartphone hilang, ia tidak bisa direset ulang tanpa memasukkan id google dan kata sandi (password).<sup>[118]</sup></li> </ul>	<p>Android dipuji sebagai “<i>Platform mobile</i>” karena :</p> <p>a. Lengkap (</p>
12	<p>Android 6.0 Marshmallow</p> 	<ul style="list-style-type: none"> <li>• Android Pay</li> <li>• Android Marshmallow fingerprint support</li> <li>• Voice controls</li> </ul>	<p><i>Complete Platform</i> ) :</p> <p>Para desainer dapat</p>

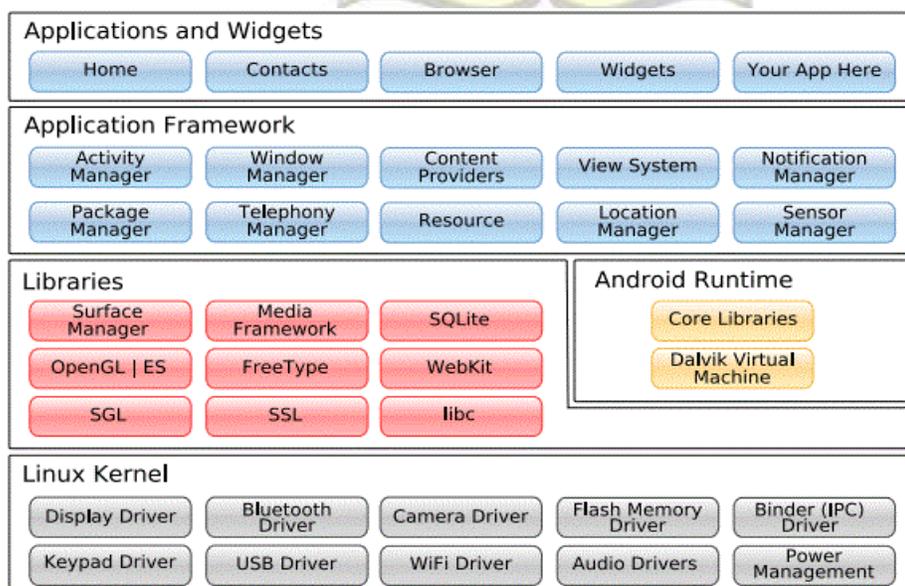
melakukan pendekatan yang komprehensif ketika mereka sedang mengembangkan *platform* Android. Android merupakan sistem operasi yang aman dan banyak menyediakan *tools* dalam membangun *software* dan memungkinkan untuk peluang pengembangan aplikasi.

- b. Terbuka ( *Open Source Platform* ) : *Platform* Android disediakan melalui lisensi *open source*. Pengembang dapat dengan bebas untuk mengembangkan aplikasi. Android sendiri menggunakan Linux Kernel 2.6.
- c. *Free* ( *Free Platform* ) : Android adalah *platform*/Aplikasi yang bebas untuk develop. Tidak ada lisensi atau biaya royalty untuk dikembangkan pada *platform* Android. Tidak

ada biaya keanggotaan diperlukan. Tidak diperlukan biaya pengujian. Tidak ada kontrak yang diperlukan. Aplikasi untuk android dapat didistribusikan dan diperdagangkan dalam bentuk apa pun.

### 2.3.1 Arsitektur System Android

Google sebagai pencipta Android yang kemudian diasuh oleh Handset Alliance mengibaratkan Android sebagai sebuah tumpukan software. Setiap lapisan pada tumpukan ini menghimpun beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Tumpukan paling bawah adalah kernel, Google menggunakan kernel Linux Versi 2.6 untuk membangun Android, yang mencakup memory management, security setting, power management, dan beberapa driver hardware. Bertempat di level yang sama dengan library adalah lapisan runtime yang mencakup serangkaian inti library Java. Lapisan selanjutnya adalah aplikasi framework yang mencakup program untuk mengatur fungsi-fungsi dasar smartphone.



Gambar 2.2 Arsitektur System Layer Android (Wei Meng Lee, 2011)

### 1. **Linux Kernel**

Linux *Kernel* adalah *layer* di mana inti dari *operating system* dari Android itu berada. Berisi *file-file* sistem yang mengatur *system processing*, *memory*, *resource*, *drivers*, dan sistem-sistem operasi Android lainnya.

### 2. **Android Runtime**

Lapisan setelah Kernel Linux adalah Android *Runtime*. Android *Runtime* ini berisi *Core Libraries* dan *Dalvik Virtual Machine*. *Core Libraries* mencakup serangkaian inti *library* Java, artinya Android menyertakan satu set *library-library* dasar yang menyediakan sebagian besar fungsi-fungsi yang ada pada *library-library* dasar bahasa pemrograman Java.

### 3. **Libraries**

Bertempat di level yang sama dengan Android *Runtime* adalah *Libraries*. Android menyertakan satu set *library-library* dalam bahasa C/C++ yang digunakan oleh berbagai komponen yang ada pada sistem Android.

### 4. **Application Framework**

Lapisan selanjutnya adalah yang mencakup program untuk mengatur fungsi-fungsi dasar *smartphone*. *Application Framework* merupakan serangkaian tool dasar seperti

alokasi *resource* smartphone, aplikasi telepon, pergantian antar – proses atau program, dan pelacakan lokasi fisik telepon.

## **5. *Application dan Widgets***

Di lapisan teratas bertempat pada aplikasi itu sendiri. Di lapisan inilah anda menemukan fungsi-fungsi dasar smartphone seperti menelepon dan mengirim pesan singkat, menjalankan web browser, mengakses daftar kontak, dan lain-lain. Bagi rata-rata pengguna, lapisan inilah yang paling sering mereka akses. Mereka mengakses fungsi-fungsi dasar tersebut melalui user interface.

## **2.4 Eclipse, IDE Standar**

Dalam membuat aplikasi android, eclipse adalah IDE standar yang digunakan untuk merancang atau membuat aplikasi android. Walaupun pada praktiknya dalam membuat aplikasi android dapat juga menggunakan IDE seperti Netbeans ataupun yang lain namun eclipse sudah menjadi standar serta paling banyak digunakan dalam mengembangkan aplikasi android. Selain opensource, serta mendapat dukungan dari google sebagai penyedia sdk android, fitur dan plugin yang berlimpah pada eclipse menjadikan IDE ini menjadi mudah digunakan serta superior.

Tabel 2.2 Versi Eclipse

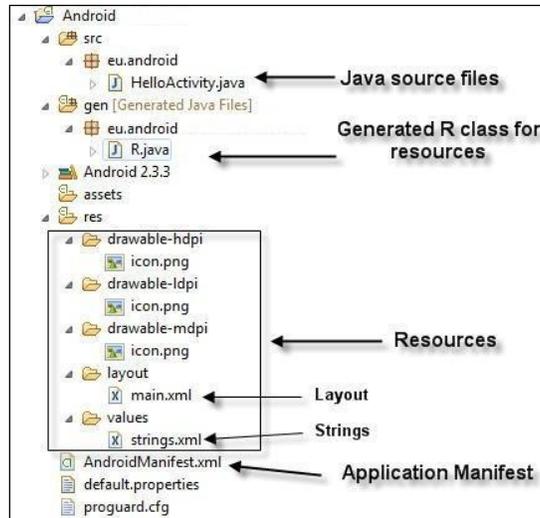
Kode Peluncuran	Tanggal Peluncuran	Platform	Nama Project
Eclipse 3.0	28 Juni 2004	3.0	
Eclipse 3.1	28 Juni 2005	3.1	
Callisto	30 Juni 2006	3.2	Callisto Projects
Europa	29 Juni 2007	3.3	Europa Projects
Ganymede	25 Juni 2008	3.4	Ganymede Project
Galileo	24 Juni 2009	3.5	Galileo Projects
Helios	23 Juni 2010	3.6	
Indigo	23 Juni 2011	3.7	
Juno	27 Juni 2012	4.2	
Kepler	26 Juni 2013	4.3	
Luna (planned)	25 Juni 2014		
Mars (planned)	24 Juni 2015		

### 2.4.1 Arsitektur Eclipse

Secara standar eclipse selalu dilengkapi dengan JDT (java development tools), plug-in yang membuat eclipse kompatibel untuk mengembangkan program java, dan PDE (plugin development environment) untuk mengembangkan plug-in baru. Eclipse beserta plug-in-nya diimplementasikan dalam bahasa pemrograman java. Konsep eclipse IDE adalah :

1. terbuka (open),
2. mudah diperluas (extensible) untuk apa saja, dan
3. Mudah digunakan (easy to use)

Dalam struktur eclipse khususnya dalam struktur project aplikasi android memiliki keterangan struktur:



Gambar 2.3 Struktur folder android di eclipse IDE

1. Folder src merupakan folder letak file sumber java (source) sebagai contoh file hello.java.
2. Folder gen merupakan folder letak file generate java (R.java).
3. Folder Android 2.x merupakan folder letak semua file libraries untuk Android 2.x. karena kita menetapkan target development untuk device dengan sistem operasi Android 2.3.3 (GingerBread), jadi yang tampil folder-nya nanti adalah Android 2.3.3
4. Folder assets merupakan folder letak semua asset raw (raw files). File-file yang dalam folder ini contohnya file XML, file-file SQLite.
5. Folder res (resources) merupakan folder letak file-file string, audio, dan gambar.
6. Folder drawable-hdpi merupakan folder untuk menyimpan file gambar yang terkompilasi yang ditujukan untuk layar beresolusi tinggi (contoh file XML dan bitmap).
7. Folder drawable-ldpi merupakan folder untuk menyimpan file gambar yang terkompilasi yang ditujukan untuk layar beresolusi rendah.

8. Folder `drawable-mdpi` merupakan folder untuk menyimpan file gambar yang terkompilasi yang ditujukan untuk layar beresolusi sedang.
9. Folder `layout` merupakan folder berisi file-file XML yang mendefinisikan tampilan antar muka/userinterface aplikasi kita.
10. Folder `values` merupakan folder berisi file XML yang berisi nilai-nilai API sederhana, seperti `colors`, `integers`, `strings`.
  - `Styles.xml` file XML yang merepresentasikan styles (seperti CSS pada HTML).
  - `Arrays.xml` file XML yang digunakan untuk mendeklarasikan sebuah array.
  - `Colors.xml` file XML yang digunakan untuk mendeklarasikan nilai color, dapat diakses melalui class `R.colors`.
  - `String.xml` file XML yang digunakan untuk menyimpan nilai string, dapat diakses melalui class `R.strings`.
  - `dimens.xml` file XML yang digunakan untuk mendefinisikan nilai `dimens/dimensi`. Sebagai contoh, nilai `10px` sama dengan `10 pixels`, dapat diakses melalui class `R.dimens`.
11. File `AndroidManifest.xml` merupakan file XML yang berisi segala informasi penting yang dibutuhkan oleh aplikasi kita untuk bekerja. File manifest ini juga berisi nama dari paket java aplikasi kita, komponen aplikasi seperti `activities` atau `background service`, informasi minimum platform yang dapat menjalankan aplikasi kita (kita yang menentukan sendiri), deklarasi izin (`permission`) dari aplikasi kita.
12. File `default.properties` merupakan file yang berisi project setting seperti target platform jangan sampai hilang setting yang ada pada file ini.

## 2.5 UML

UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. (Hermawan Julius, 8:2004)

Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem. Singkatnya *Unified Modelling Language (UML)* adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

Sampai era tahun 1990 puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Diantaranya adalah metodologi *booch*, metodologi *coad*, metodologi *OOSE*, metodologi *OMT*, metodologi *shlaer-mellor*, metodologi *wirfs-brock*, dsb. Masa itu terkenal dengan masa perang metodologi (*method war*) dalam pendesainan berorientasi objek. Masing-masing metodologi membawa notasi sendiri-sendiri, yang mengakibatkan timbul masalah baru apabila kita bekerjasama dengan kelompok/perusahaan lain yang menggunakan metodologi yang berlainan.

Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikatakan metodologinya banyak digunakan memelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 dirilis *draft* pertama dari

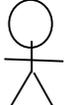
UML (versi 0.8). Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh *Object Management Group (OMG)*.

### 2.5.1 Use Case Diagram

*Use case* adalah abstraksi dari interaksi antara *system* dan *actor*. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah *system* dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah *system* dipakai. *Use case* merupakan konstruksi untuk mendeskripsikan bagaimana *system* akan terlihat di mata *user*. Sedangkan *use case diagram* memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan *client*.



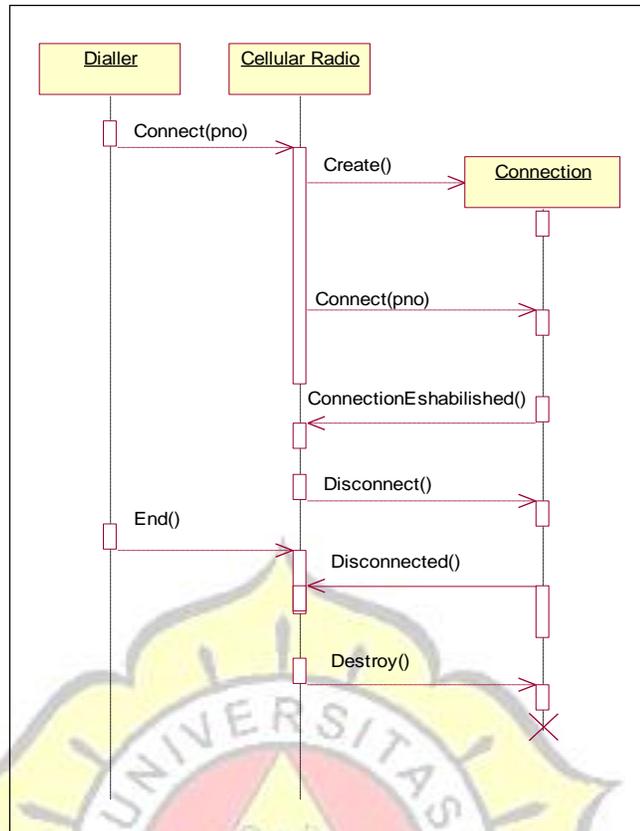
Tabel 2.3 Simbol *Use Case Diagram*.

Penjelasan	Simbol Use Case
<p>Aktor adalah sesuatu yang menggunakan sistem/ merupakan <i>interface</i> bagi <i>system</i> untuk berhubungan dengan luar sistem. <i>Aktor</i> bisa berupa orang, <i>software</i> lain, peralatan <i>Hardware</i>, penyimpanan data/ jaringan.</p>	
<p><i>Use Case</i> adalah rangkaian yang</p>	

memperlihatkan bagaimana <i>user</i> berinteraksi dengan sistem (kemampuan apa saja yang bisa diberikan kepada <i>user</i> oleh sistem tersebut	Mengisi soal
<i>Association</i> adalah abstraksi dari penghubung antara aktor dan <i>use case</i>	→
Generalisasi adalah abstraksi dari interaksi aktor dengan aktor. <i>Include</i> adalah abstraksi dari interaksi antara <i>use case</i> dengan <i>use case</i> yang melanjutkan dari <i>use case</i> sebelumnya. <i>Extend</i> adalah abstraksi dari interaksi antara <i>use case</i> dengan <i>use case</i> yang memberikan keterangan <i>use case</i> sebelumnya.	- - - - ->

### 2.5.2 Sequence Diagram

*Sequence Diagram* digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antara objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.



Gambar 2.4 Contoh *Diagram Sequence*. (Robert C. Martin, 2008)

### 2.5.3 Activity Digram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

Tabel 2.4 Simbol *Activity Diagram*.

Keterangan	Simbol
Titik Awal atau permulaan.	●

Titik Akhir atau akhir dari aktivitas.	
Activity, atau aktivitas yang dilakukan oleh aktor.	
<i>Decision</i> , atau pilihan untuk mengambil keputusan.	
Arah tanda panah alur proses.	

#### 2.5.4 Tujuan Penggunaan UML

- Memberikan bahasa pemodelan yang bebas dari berbagai bahas pemrograman dan proses rekayasa.
- Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.
- Memberikan model yang siap pakai, bahasa pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
- UML bisa juga berfungsi sebagai sebuah (*blue print*) cetak biru karena sangat lengkap dan detail. Dengan cetak biru ini maka akan bias diketahui informasi secara detail tentang *coding* program atau bahkan membaca program dan menginterpretasikan kembali ke dalam bentuk diagram (*reverse engineering*).

#### 2.6 Databases Server

*Server* basis data adalah sebuah program komputer yang menyediakan layanan pengelolaan basis data dan melayani komputer atau program aplikasi basis data yang menggunakan *client-server*.

Sistem manajemen basis data (SMBD) pada umumnya menyediakan fungsi-fungsi *server* basis data, dan beberapa SMBD (seperti halnya MySQL atau Microsoft SQL *Server*) sangat bergantung kepada model *client-server* untuk mengakses datanya. (Abdul Kadir, 2008)



