# BAB II

# LANDASAN TEORI

## 2.1. Aplikasi

Menurut Daryanto (2004:347), aplikasi adalah software atau perangkat lunak yang dibuat untuk mengerjakan menyelesaikan masalah-masalah khusus.

Sedangkan menurut Jogiyanto (2004:4), aplikasi merupakan program yang berisikan perintah-perintah untuk melakukan pengolahan data. Jadi aplikasi secara umum adalah suatu proses dari cara manual yang ditransformasikan ke komputer dengan membuat sistem atau program agar data diolah lebih berdaya guna secara optimal.

Dari uraian diatas dapat disimpulkan bahwa aplikasi adalah sebuah perangkat lunak yang berisi perintah untuk menyelesaikan masalah dan pengolahan data.

## 2.2. Pengertian Pemesanan

Pemesanan adalah suatu aktifitas yang dilakukan oleh konsumen sebelum membeli. Untuk mewujudkan kepuasan konsumen maka perusahaan harus mempunyai sebuah sistem pemesanan yang baik.

## 2.3. Mobile

Mobile adalah perangkat elektronik portabel yang berfungsi sebagaimana pesawat telepon normal, yang dapat bergerak pada suatu area yang luas. (bandingkan dengan cordless phone).

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.

### 2.3.1. Aplikasi Mobile

1. Messaging

   Contoh: SMS, MMS, Instant Messaging

2. Mobile transaction

   Contoh: SMS alert, MMS alert, report analisys

3. Mobile workplace

   Contoh: Email, calender, CRM, Instant Messaging

4. Mobile music and videos

   Contoh: Monotone, RTTTL, Midi, mp3, wav, mp4, screensaver, picture message, A2DP

5. Mobile games

   Contoh: Monochrome games, java games, symbian

### 2.3.2. Area Aplikasi Mobile

1. *Business-to-Customer*

   a. Personalisasi aplikasi *e-commerce*. Misal: membandingkan harga barang

   b. Bisnis barang digital. Misal: menjual aplikasi Java MIDP, Video, MP3, ringtone

c. Improvisasi layanan yang sudah ada. Misal: marketing sesuai dengan profile pemakai

2. *Business-to-Business*
   a. Mobile *supply chain management*
   b. Mobile *commerce*
3. *Business-to-Employee*
   a. Mobile Sales Marketing
4. *Government and Public services*
   a. Polisi mengecek data SIM pemilik mobil
   b. perawat mengecek data medis pasien

## 2.4. Android

Pembahasan mengenai android dan *tools* yang digunakan dalam pembuatan aplikasi, akan dijelaskan pada sub bab berikut.

### 2.4.1. Sejarah Android

Android adalah sebuah sistem operasi untuk perangkat lunak *mobile* berbasis linux yang mencakup sistem operasi, middleware dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 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 *open source* pada perangkat *mobile*. Di lain pihak, Google merilis kode-kode android dibawah lisensi *Apache*, sebuah lisensi perangkat lunak dan open *platform* perangkat seluler.

Didunia ini terdapat dua jenis distributor sistem operasi android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail *Service*s (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD).

Sekitar September 2007 Google mengenalkan Nexus One, salah satu jenis *smartphone* yang menggunakan android sebagai sistem operasinya. Telpon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android AEM Holdings, Atheros Communication, diproduksi oleh Asustek Komputer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, android perangkat *mobile* yang merupakan modifikasi kernel Linux 2.6. sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.

Pada masa saat ini kebanyakan vendor-vendor *smartphone* sudah memproduksi *smartphone* berbasis android, vendor-vendor itu antara lain HTC, Motorola, Samsung, LG, HKC, Huawei, Archos, Webstation Camangi, Dell, Nexus, SciPhone, WayteQ, Sony Ericsson, LG, Acer, Philips, T-*Mobile*, Nexian,

IMO, Asus dan masih banyak lagi vendeor *smartphone* didunia yang memproduksi android. Hal ini karena android adalah sistem oerasi yang *open source* sehingga bebas didistribusikan dan dipakai oleh vendor manapun.

Tidak hanya menjadi sistem operasi di *smartphone*, saat ini android menjadi pesaing utama dari Apple pada sistem operasi Table PC. Pesatnya pertumbuhan Android selain faktor yang disebutkan diatas adalah karena android itu sendiri adalah *platform* sangat lengkap baik itu sistem oeprasinya, aplikasi dan Tool Developmen, Market aplikasi android serta dukungan yang sangat tinggi dari komunitas *Open source* didunia, sheingga android terus berkembang pesat dari segi teknologi maupun dari segi jumlah device yang ada didunia.

### 2.4.2. *The Dalvik Virtual Machine* (DVM)

Salah satu elemen kunci dari android adalah *Dalvik Virtual Machine* (DVM). Android berjalan di *Dalvik Virtual Machine* (DVM) bukan di Java *Virtual Machine* (JVM), sebernarnya banyak persamaan dengan Java *Virtual Machine* (JVM) seperti Java ME (*Java Mobile Edition*), tetapi android menggunakan *Virtual Machine* sendiri yang diskostumisasi dan dirancang untuk memastikan bahwa beberapa fitur-fitur berjalan lebih efisien pada perangkat *mobile*.

*Dalvik Virtual Machine* (DVM) adalah "*register bases*" sementara java *Virtual Machine* (JVM) bersifat "*stack based*", DVM didesain dan ditulis oleh Dan Bornsten dan beberapa enginers Google lainnya. Jadi bisa dikatakan "*Dalvik* equal(Java) == False" *Dalvik Virtual Machine* menggunakan kernel linux untuk menangani fungsionalitas tingkat rendah termasuk keamanan, *threading*, dan proses serta manajemen memori. Ini memungkinkan kita untuk menulis aplikasi

C/C++ sama halnya seperti pada OS Linux kebanyakan. Meskipun dalam kenyataannya kita harus banyak memahami Arsitektur dan proses sistem dari kernel linux yang digunakan dalam Android tersebut.

Semua *hardware* yang berbasis android dijalankan dengan menggunakan *Virtual Machine* untuk eksekusi aplikasi, pengembang tidak perlu khawatir tentang implementasi pernagkat keras tertentu. *Dalvik Virtual Machine* mengeksekusi *Executable* file, sebuah format yang dioptimalkan untuk memastikan memori yang digunakan sangat kecil. *The Executable* file diciptakan dengan mengubah kelas bahasa java dan dikompilasi menggunakan *tools* yang disediakan dalam SDK Android.

### 2.4.3. Arsitektur Android

Secara garis besar arsitektur android dapat dijelas dan digambarkan sebagai berikut :

1. *Application and Widgets*

*Application and widgets* adalah layer dimana berhubungan dengan aplikasi dan biasanya download aplikasi kemudian lakukan instalasi dan jalankan aplikasi tersebut, delayer inilah terdapat seperti aplikasi inti termasuk klien email, program SMS, kalender, peta, *browser*, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

*2. Application Frameworks*

Android adalah "*Open Development Platform*" yaitu android menawarkan kepada pengembang atau member kemampuan kepada pengembangan untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk

mengakses perangkat keras, akses informasi *resources*, menjalankan serive *background*, mengatur alarm, dan menambahkan tambahan seperti status*notification*s dan masih banyak lagi. Pengembang memiliki akses penuh menuju API *Framework* seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang supaya dengan mudah dapat menggunakan komponen yang sudah digunakan (*reuse*).

Sehingga bisa disimpulkan Application Framework adalah layer dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem android, karena pada layer inilah aplikasi dapat dirancang, seperti *content-providers* yang berupa sms dan lain sebagainya.

Komponen-komponen yang termasuk didalam *application Framework* adalah sebagai berikut :

*1) Views*

*2) Content Provider*

*3) Resource Manager*

*4) Notification Manager*

*5) Activity Manager*

*3. Libraries*

*Libraries* adalah layer dimana fitur-fitur android berada biasanya para pembuat aplikasi kebanyakan mengakses *libraries* untuk menjalankan aplikasinya. Berjalan diatas kernel, layer ini meliputi berbagai library C/C++ inti seperti Libc dan SSL, serta :

1) *Libraries* media untuk pemutar media audio dan video.

2) *Libraries* untuk manajemen tampilan.

3) *Libraries Graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D.

4) *Libraries SQLite* untuk dukungan database.

5) *Libraries* SSL dan *WebKit* terintegrasi dengan web *browser* dan security

6) *Libraries Live Webcore* mencakup modern web *browser* dengan engine embedded web view

4. *Android Run Time*

Layer yang membuat aplikasi android dapat dijalankan dimana dalam prosesnya menggunakan implementasi Linux. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Didalam *Android Run Time* dibagi menjadi dua bagian yaitu:

1) *Core Libraries* : aplikasi android dibangun dalam bahasa java, sementara *Dalvik* sebagai virtual mesin bukan *Java Virtual Machine*, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan bahasa Java/C yang dihandle oleh Core *Libraries*

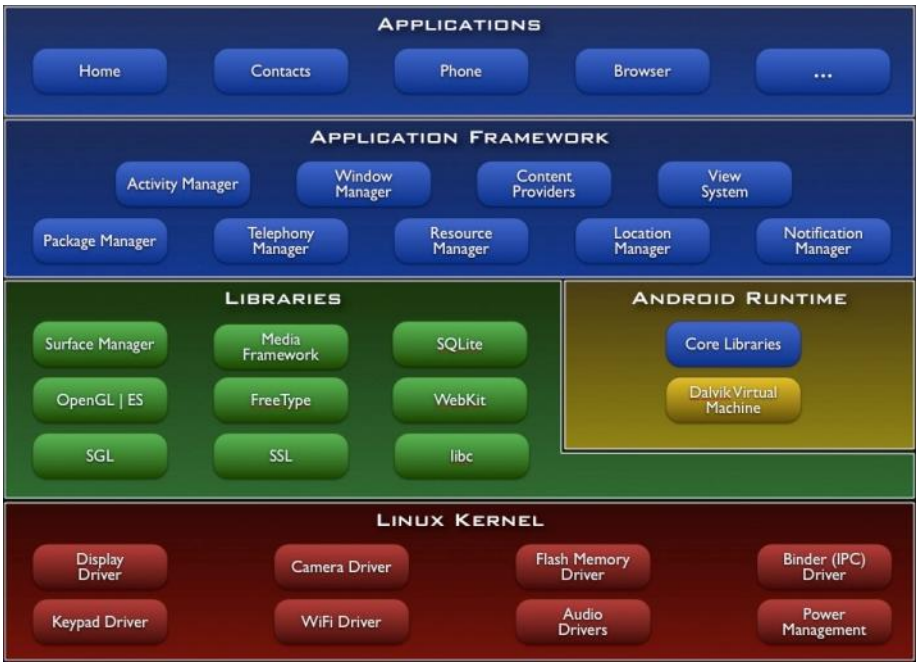
2) *Dalvik Virtual Machine*

   Virtual mesin yang berbasis *register* yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien dimana merupakan pengembangan yang mampu membuat linux kernel untuk *threading* dan manajemen tingkat rendah.

3) Linux Kernel

   Linux kernel adalah layer dimana inti dari operating sistem dari android itu sendiri, berisi file-file sistem yang mengatur sistem *processing*, *memory*,

*resources*, *drivers*, dan sistem-sistem operanting andoroid lainnya. Linux kernel yang digunakan android adalah linux kernel *release* 2.6.




Gambar 2.1 Arsitektur Andoroid

### 2.4.4. Fundamental Aplikasi

Aplikasi android ditulis dalam bahasa pemrograman java, kode java dikompilasi bersama data file *resource* yang dibutuhkan oleh aplikasi dimana prosesnya dipaket oleh *tools* yang dinamakan "apt *tools*" ke dalam paket android sehingga menghasilkan file dengan ekstensi apk. File apk itulah yang sebenernya disebut dengan aplikasi yang dapat diinstal di perangkat *mobile* nantinya.

Ada enam jenis komponen pada aplikasi android yaitu:

*a. Activities*

Suatu *activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi android bisa jadi hanya memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity*

tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut. Satu *activity* biasanya akan dipakai untuk menampilkan aplikasi atau yang bertindak sebagai *user interface* (UI) saat aplikasi diperlihatkan kepada *user*. Untuk pindah dari satu *activity* ke *activity* lain dapat dilakukan dengan satu even misalnya click tombol, memilih opesi atau menggunakan triggers tertentu. Secara hirarki sebuah window *activity* dinyatakan dengan method *Activity*.setContentView(). ContentView adalah objek yang berada pada root hirarki.

*b. Service*

*Service* tidak memiliki visual *user interface* (UI), tetapi *service* berjalan secara *background*, sebagai contoh dalam memainkan music, *service* mungkin memainkan music atau mengambil data dari jaringan, tetapi setiap *service* haruslah berada dalam kelas induknya. Misalnya media player sedang memutar lagu dari *list* yang ada. Aplikasi ini akan memiliki dua atau lebih *activity* yang memungkinkan *user* untuk memilih lagu atau menulis sms sambil *player* sedang jalan untuk menjaga *music* tetapi dijalankan, *activity player* dapat menjalankan *service* untuk membuat aplikasi tetap berjalan. *Service* dijalankan pada *thread* utama dari proses aplikasi.

*c. Broadcast Receiver*

*Broadcast Receiver* berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Contoh *broadcast* seperti notifikasi zona waktu berubah, baterai *low*, gambar telah selesai diambil oleh kamera, atau pengubah referensi bahasa yang digunakan. Aplikasi juga dapat menginisiasi *broadcast* misalnya memberikan informasi pada aplikasi lain bahwa ada data yang telah didownload ke perangkat

dan siap untuk digunakan *Broadcast Receiver* tidak memiliki *user interface* (UI) tetapi memiliki sebuah *activity* untuk merespon informasil yang mereka terima atau mungkin menggunakan *notification* manager untuk memberitahu kepada pengguna seperti lampu latar atau *viberating* (getaran) perangkat dan lain sebagainya.

*d. Content Provider*

*Content Provider* membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam file sistem seperti database SQLite.*Content Provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika menggunakan aplikasi yang membutuhkan peta (MAP) atau aplikasi yang membutuhkan untuk mengkases data kontak dan navigasi, maka disinilah fungsi *content provider*.

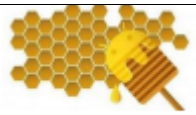
### 2.4.5. Versi Android

Sejak dirilis pertama kali dan diperkenalkan, android mengalami perkembangan untuk versi dari versi pertama yang disesuaikan dengan kebutuhan aplikasi, baik dari segi perbaikan bug ataupun penambahan fitur baru. Perkembangan versi android dapat dilihat pada tabel di bawah ini.

Tabel 2.1. Versi Android

| Versi | Logo | Penjelasan |
|---|---|---|
| Versi 1.1 |  | Rilis 9 Maret 2009, dilengkapi pembaruan estetis pada aplikasi, jam alarm, voice search, pengiriman pesan dengan Gmail, dan pemberitahuan email. |

| | | |
|---|---|---|
| Versi 1.5 (Cupcake) |  | Rilis pertengahan Mei 2009, dilengkapi kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke YouTube dan gambar ke Picasa, dukungan bluetooth A2DP, animasi layar, serta keyboard pada layar yang dapat disesuaikan dengan sistem. |
| Versi 1.6 (Donut) |  | Rilis September 2009, dilengkapi proses pencarian lebih baik, baterai indikator dan kontrol applet VPN, galeri foto, Text-to-speech engine, kemampuan dial kontak, teknologi text to change speech, VPN, gesture, CDMA / EVDO, 802.1x, dan integrasi kamera, camcorder, dan galeri. |
| Versi 2.0/2.1 (Eclair) |  | Rilis 3 Desember 2009, pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, dukungan flash untuk kamera 3,2MP, digital zoom, Bluetooth 2.1. |
| Versi 2.2 (Froyo: Frozen Yoghurt) |  | Rilis 20 Mei 2010, dilengkapi dukungan Adobe Flash 10.1, kecepatan kinerja, integrasi V8 javascript engineI yang dipakai Google Chrome, pemasangan aplikasi dalam SD card, kemampuan WiFi hotspot portable, dan auto update dalam aplikasi Android Market. |

| | | |
|---|---|---|
| Versi 2.3 (Gingerbread) |  | Rilis 6 Desember 2010, dilengkapi peningkatan kemampuan gaming, fungsi copy-paste, efek audiobaru(reverbm equalization, headphone virtualization, dan bass boost), kemampuan Near Field Communication (NFC), dan jumlah kamera lebih dari satu. |
| Versi 3.0 (Honeycomb) |  | Versi yang dirancang untuk tablet, mendukung layar yang lebih besar dan User Interface yang didesain untuk tablet, multi prosesor, dan akselerasi hardware untuk grafis. |
| Versi 4.0 (ICS :Ice Cream Sandwich) |  | Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC. |

## 2.5. Pemrograman Berorientasi Objek

Pemrograman berorientasi objek (Object-Oriented Programming) merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek.

Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya.

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

### 2.5.1. *Unified Modelling Language* (UML)

UML (*Unified Modelling Language*) adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi objek. Hal ini disebabkan UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Object Modeling Technique* (OMT) dan *Object Oriented Software Engineering* (OOSE). Metode Booch dari Grady Booch sangat terkenal dengan nama metode *Design Object Oriented*. Metode ini menjadikan proses analisis dan *design* ke dalam empat tahapan iterative, yaitu: identifikasi kelas-kelas dan obyek-obyek, identifikasi semantic dari hubungan obyek dan kelas tersebut, perincian *interface* dan implementasi. Keunggulan metode Booch adalah pada detil dan kayanya dengan notasi dan elemen. Pemodelan OMT yang dikembangkan oleh

Rumbaugh didasarkan pada analisis terstruktur pemodelan entity-relationship. Tahapan utama dalam metodologi ini adalah analisis, *design* sistem, *design* obyek dan implementasi. Keunggulan metode ini adalah dalam penotasian yang mendukung semua konsep OO. Metode OOSE dari Jacobson lebih member penekanan pada use case. OOSE memiliki tiga tahapan yaitu membuat model requirement dan analisis, *design* dan implementasi, dan model pengujian (*test model*). Keunggulan metode ini adalah mudah dipelajari karena memiliki notasi yang sederhana namun mencakup seluruh tahapan dalam rekayasa perangkat lunak.

*Design* UML, metode Booch, OMT dan OOSE digabungkan dengan membuang elemen-elemen yang tidak praktis ditambah dengan elemen-elemen dari metode lain yang lebih efektif dan elemen-elemen baru yang belum ada pada metode terdahulu sehingga UML lebih ekspresif dan seragam dari pada metode lainnya.

#### 2.5.1.1. Use Case Diagram

*Use Case* adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use Case* bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut *scenario*. Setiap scenario mendeskripsikan urutan kejadian. Setiap urutan diinisialisasi oleh orang, sistem yang lain, perangkat keras atau urutan waktu. Dengan demikian secara singkat bisa

dikatakan *use case* adalah serangkaian scenario yang digabungkan bersama-sama oleh tujuan umum pengguna.

Dalam pembicaraan tentang *use case*, pengguna biasanya disebut dengan aktor. Aktor adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksinya dengan sistem.

Model *use case* adalah bagian dari model *requirement* (Jacob et all, 1992). Termasuk disini adalah problem domain *object* model dan penjelasan tentang *user interface*. *Use case* memberikan spesifikasi fungsi-fungsi yang ditawarkan oleh sistem dari perspektif *user*.

**2.5.1.2. Sequence Diagram**

*Sequence* diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek dan *message* yang diletakkan diantara objek-objek ini di dalam *use case*. *Sequence* diagram menambahkan dimensi waktu pada interaksi diantara objek. Pada diagram ini *participant* diletakkan di atas dan waktu ditunjukkan dari atas ke bawah. *Life line participant* diurutkan dari setiap *participant*. Kotak kecil pada *lifelin*e menyatakan *activation*, yaitu menjalankan salah satu operation dari *participant*. State bisa ditambahkan dengan menambahkannya sepanjang life line.

*Message* (sederhana, *synchronou*s atau *asynchronous*) adalah tanda panah yang menghubungkan suatu *life line* ke *life line* yang lain. Lokasi *life line* dalam dimensi vertikal mewakili urutan waktu dalam *sequence* diagram. *Message* yang pertama terjadi adalah yang paling dekat dengan bagian atas diagram dan yang terjadi belakangan adalah yang dekat dengan bagian bawah.

Pada beberapa sistem, operasi bisa dilakukan kepada dirinya sendiri. Hal ini disebut dengan rekursif. Untuk melukiskannya digunakan anak panah dari *activation* kembali ke dirinya sendiri, dan sebuah kotak kecil diletakkan pada bagian atas dari *activation*.

#### 2.5.1.3. *Deployment* Diagram

*Deployment diagram* menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian software yang berjalan pada bagian-bagian hardware yang digunakan untuk mengimplementasikan sebuah sistem dan keterhubungan antara komponen-komponen *hardware* tersebut. *Deployment* diagram dapat digunakan pada bagian-bagian awal proses perancangan sistem untuk mendokumentasikan arsitektur fisik sebuah sistem.

#### 2.5.1.4. *Activity* Diagram

*Activity* diagram seperti sebuah *flow chart*. *Activity* diagram menunjukkan tahapan, pengambilan keputusan dan percabangan. Diagram ini sangat berguna untuk menunjukkan *operation* sebuah obyek dan proses bisnis. Kelebihan *activity* diagram dibandingkan *flowchart* adalah kemampuannya dalam menampilkan aktivitas parallel.

#### 2.5.1.5. Class Diagram

Kotak adalah notasi UML untuk *class*. Nama, *attribute*, *operation* dan *responsibility* dari *class* ada pada kotak tersebut. *Stereotype* bisa dipergunakan untuk mengorganisasikan daftar *attribute* dan *operation*. Dalam beberapa kasus, kadang kala hanya perlu ditampilkan sebagian saja dari *attribute* dan *operation*. Tipe *attribute* dan nilai default bisa dimunculkan sebagaimana pada *operation*.

Untuk mengurangi ambiguitas pada pendeskripsian *class*, *constraint* bisa ditambahkan. Bahkan kalau perlu bisa ditambahkan *attached* notes ke dalam kotaktersebut.

## 2.6. Java Platform Android

Aplikasi yang dibuat dan dikembangkan dalam penulisan ini menggunakan bahasa pemrograman berbasis java untuk platform android. Platform java ini mirip dengan Java 2 Standard Edition (J2SE) dikarenakan struktur perintah untuk kondisi dan perulangan, tipe variabel yang dapat digunakan, dan operator yang digunakan. Platform ini dapat dikatakan pengembangan dari J2SE yang diterapkan untuk aplikasi mobile yang berbeda dari J2ME (Java 2 Micro Edition), karena file yang dihasilkan berekestensi .apk bukan .jar.

### 2.6.1. Gambaran Java Platform Android

Pada android, penggunaan bahasa pemrograman java tidak berbeda dengan java untuk platform lainya, yaitu membutuhkan dua buah produk yang dikeluarkan untuk membantu dalam membuat aplikasi, yaitu:

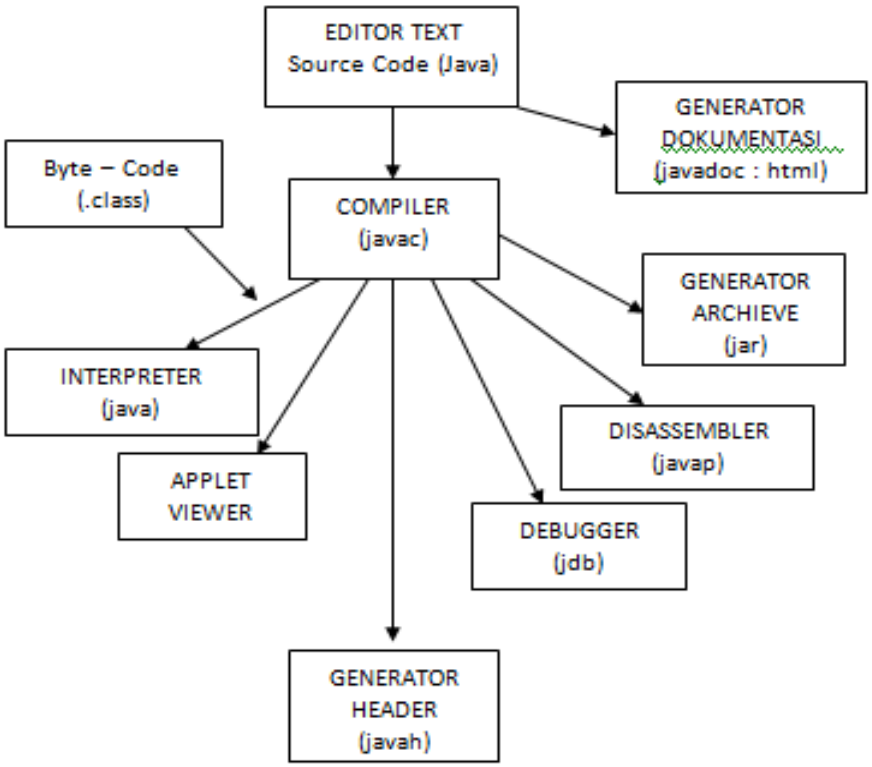
1. Java SE Runtime Environment (JRE)

   Java Runtime Environment (JRE) menyediakan perpustakaan, Java Virtual Machine (JVM), dan komponen lain untuk menjalankan applet dan aplikasi yang ditulis dengan bahasa pemrograman java. Selain itu, terdapat dua buah kunci teknologi yang merupakan bagian dari JRE, yaitu: Java Plug-in, yang memungkinkan menjalankan applet di browser populer dan Java Web Start, yang menyebarkan aplikasi mandiri melalui jaringan. JRE tidak mengandung

utilitas seperti compiler atau debugger untuk mengembangkan applet dan aplikasi. JRE yang digunakan adalah JRE6.

2. Java Development Kit (JDK)

Java Development Kit (JDK) merupakan perangkat lunak yang digunakan untuk menajemen dan membangun berbagai aplikasi java. JDK merupakan superset dari JRE, berisikan segala sesuatu yang ada di JRE ditambahkan compiler dan debugger yang diperlukan untuk mengembangkan applet dan aplikasi. JDK yang mendukung aplikasi android adalah JDK1.6.0_24 (JDK 1.6 update 24). Pada JDK, terdapat 25 berbagai tools yang digunakan untuk membangun aplikasi java. Tools tersebut digambarkan dengan diagram berikut:




Gambar 2.2 Tools Dasar JDK

## 2.7. XML

XML (Extensible Markup Language) adalah bahasa markup untuk keperluan umum yang disarankan oleh W3C untuk membuat dokumen markup keperluan pertukaran data antar sistem yang beraneka ragam. XML merupakan kelanjutan dari HTML (HyperText Markup Language) yang merupakan bahasa standar untuk melacak Internet.

### 2.7.1. Mengenal XML

XML didesain untuk mempu menyimpan data secara ringkas dan mudah diatur. Kata kunci utama XML adalah data (jamak dari datum) yang jika diolah bisa memberikan informasi. XML menyediakan suatu cara terstandarisasi namun bisa dimodifikasi untuk menggambarkan isi dari dokumen. Dengan sendirinya, XML dapat digunakan untuk menggambarkan sembarang view database, tetapi dengan suatu cara yang standar.

### 2.7.2. Tipe XML

XML memiliki tiga tipe file :

1. XML, merupakan standar format dari struktur berkas (file).
2. XSL, merupakan standar untuk memodifikasi data yang diimpor atau diekspor.
3. XSD, merupakan standar yang mendefinisikan struktur database dalam XML.

### 2.7.3. Keunggulan XML

Keunggulan XML bisa diringkas sebagai berikut :

1. Pintar (Intelligence). XML dapat menangani berbagai tingkat (level) kompleksitas.

2. Dapat beradaptasi. Dapat mengadaptasi untuk membuat bahasa sendiri. Seperti Microsoft membuaat bahasa MSXML atau Macromedia mengembangkan MXML.
3. Mudah pemeliharaannya.
4. XML lebih sederhana.
5. Mudah dipindah-pindahkan (Portability). XML mempunyai kemudahan perpindahan (portabilitas) yang lebih bagus.

## 2.8. 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:

a. *Multi-platform*: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
b. *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.
c. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman

perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

## 2.9. MySQL

MySQL adalah sebuah sistem manajemen database relasi *(relational database management system)* yang bersifat "terbuka" *(open source).* Terbuka maksudnya adalah MySQL boleh di-*download* oleh siapa saja, baik versi kode program aslinya maupun versi binernya dan dapat digunakan secara (relatif) gratis baik untuk dimodifikasi sesuai dengan kebutuhan seseorang maupun sebagai suatu program aplikasi komputer.

### 2.9.1. Keistimewaan MySQL

Sebagai *database server* yang memiliki konsep *database modern*, MySQL memiliki banyak keistimewaan. Berikut ini beberapa keistimewaan yang dimiliki oleh MySQL.

1. *Portability*

MySQL dapat berjalan stabil pada berbagai system operasi seperti Windows, Linux, FreeBSD, Mac OS X Server, dan lain-lain.

*2. Open Source*

MySQL didistribusikan secara *Open Source*, sehingga dapat digunakan secara bebas.

3. *Multi-user*

MySQL dapat digunakan oleh beberapa *User* dalam waktu yang bersamaan tanpa mengalami masalah. Hal ini memungkinkan sebuah *database server* MySQL dapat diakses *client* secara bersamaan.

4. *Performance Tuning*

MySQL memiliki kecepatan yang tinggi dalam menangani *query*, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. *Column Types*

MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed/unsigned integer, float, double, char, varchar, text, blob, date, time, timestamp, year, set*.

6. *Command* dan *Function*

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah SELECT dan WHERE dalam *query*.

7. *Security*

MySQL memiliki lapisan lapisan sekuritas seperti *level subnetmask,* nama *host*, dan izin akses *User* dengan *system* perizinan yang mendetail serta *password* terenkripsi.

8. *Scalability* dan Limits

MySQL mampu menangani *database* dengan skala besar, dengan jumlah records lebih dari 50 juta dan 60 juta table serta 5 miliar baris. Selain itu, batas indeks yang dapat ditampung mencapai 32 indeks pada setiap tabelnya.

9. *Connectivity*

MySQL dapat melakukan koneksi dengan *client* menggunakan *protocol* TCP/IP, Unix Socet (UNIX), atau Named Pipes (NT).

10. *Localisation*

Deteksi pesan kesalahan pada *client* dengan menggunakan lebih dari 20 bahasa.

11. *Interface*

*Interface* terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*)

12. *Client* dan *Tools*

Dilengkapi dengan berbagai *Tool* yang dapat digunakan untuk administrasi *database*, dan pada setiap *tool* yang ada disertakan petunjuk *online*.

13. Struktur Tabel

Struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibanding database lainnya semacam PostgreSQL ataupun Oracle.

## 2.10. *Personal Home Page Tool* (PHP)

PHP adalah salah satu bahasa *Server-side* yang didesain khusus untuk aplikasi web. PHP dapat disisipkan diantara bahasa HTML dan karena bahasa *Server side*, maka bahasa PHP akan dieksekusi di server, sehingga yang dikirimkan ke browser adalah "hasil jadi" dalam bentuk HTML, dan kode PHP tidak akan terlihat.

### 2.10.1. Keistimewaan PHP

Sebagai bahasa skrip yang bersifat server side PHP memiliki beberapa keunggulan antara lain:

1. Tidak diperlukan kompabilitas browser atau harus menggunakan browser tertentu karena serverlah yang akan mengerjakan skrip PHP tersebut
2. Freeware artinya dapat didistribusikan dengan bebas

   Memiliki kemampuan koneksi keberbagai macam database seperti: MySQL, PostgreSQL, Oracle, dBase, Sybase dan banyak lagi.

## 2.11. Internet

Internet adalah suatu jaringan komputer global yang terbentuk dari jaringan - jaringan komputer lokal dan regional yang memungkinkan komunikasi data antar komputer yang terhubung ke jaringan tersebut.

Internet awalnya merupakan suatu recana dari Departemen Pertahanan Amerika Serikat *(US Departement of Defenses)* pada sekitar tahun 1960. Dimulai dari suatu proyek yang dinamakan ARPANET atau *Advanced Research Project agency Network*. Beberapa universitas di Amerika Serikat diantaranya UCLA, Stanford, UC Santa Barbara dan Univercity of Utah, diminta bantuan dalam mengerjakan proyek ini dan awalnya telah berhasil menghubungkan empat komputer di lokasi universitas yang berbeda tersebut. Perkembangan ARPANET ini cukup pesat dilihat perkembangan komputer pada saat itu.

Sebagai gambarannya pada tahun 1977 ARPANET telah menghubungkan lebih dari 100 mainframe komputer dan saat ini terdapat 4 juta *host* jaringan yang terhubung pada jaringan ini.