

## BAB II

### LANDASAN TEORI

#### 2.1 *E-Learning (Electronic Learning)*

Definisi *Electronic learning (E-Learning)* mengandung istilah yang begitu luas, sehingga banyak pakar yang menguraikan arti *e-Learning* tersebut. Beberapa definisi dari *e-Learning* diantaranya adalah:

Menurut Darin E. Hartley (Wahono, 2003):

*e-Learning merupakan suatu jenis belajar mengajar yang memungkinkan tersampainya bahan ajar ke siswa dengan menggunakan media internet, intranet atau media jaringan komputer lain.*

Menurut *LearnFrame.Com* dalam *Glossary of e-Learning Terms* (Wahono, 2003):

*e-Learning adalah sistem pendidikan yang menggunakan aplikasi elektronik untuk mendukung belajar mengajar dengan media Internet, jaringan komputer, maupun komputer standalone.*

Secara lebih rinci Rosenberg (2001) mengategorikan tiga kriteria dasar yang ada dalam e-learning (Meredith dan Newton, 2009), yaitu:

- a. *E-learning* bersifat jaringan, yang membuatnya mampu memperbaiki secara cepat, menyimpan atau memunculkan kembali, mendistribusikan, dan sharing pembelajaran dan informasi. Persyaratan ini sangatlah penting dalam *e-learning*, sehingga Rosenberg menyebutnya sebagai persyaratan absolut.
- b. *E-learning* dikirimkan kepada pengguna melalui komputer dengan menggunakan standar teknologi internet. *CD ROM, Web TV, Web Cell Phones, pagers*, dan alat bantu digital personal lainnya walaupun bisa menyiapkan pesan pembelajaran tetapi tidak bisa dikolongkan sebagai *e-learning*.

- c. *E-learning* terfokus pada pandangan pembelajaran yang paling luas, solusi pembelajaran yang mengguguli paradigma tradisional dalam pelatihan. Uraian di atas menunjukkan bahwa sebagai dasar dari *e-learning* adalah pemanfaatan teknologi internet. Jadi *e-learning* merupakan bentuk pembelajaran konvensional yang dituangkan dalam format digital melalui teknologi internet. Oleh karena itu *e-learning* dapat digunakan dalam sistem pendidikan jarak jauh dan juga sistem pendidikan konvensional.

### **2.1.1 Manfaat *E-Learning* dalam Pembelajaran**

Menurut A. W. Bates (Bates, 1995) dan K. Wulf (Wulf, 1996) manfaat Pembelajaran elektronik Learning (*e-Learning*) itu terdiri atas 4 hal (Istikharoh dan Saefurrohman, 2008), yaitu:

1. Meningkatkan kadar interaksi pembelajaran antara peserta didik dengan guru atau instruktur (*enhance interactivity*). Apabila dirancang secara cermat, pembelajaran elektronik dapat meningkatkan kadar interaksi pembelajaran, baik antara peserta didik dengan guru/instruktur, antara sesama peserta didik, maupun antara peserta didik dengan bahan belajar (*enhance interactivity*). Berbeda halnya dengan pembelajaran yang bersifat konvensional. Tidak semua peserta didik dalam kegiatan pembelajaran konvensional dapat, berani atau mempunyai kesempatan untuk mengajukan pertanyaan ataupun menyampaikan pendapatnya di dalam diskusi. Hal ini disebabkan karena pada pembelajaran yang bersifat konvensional, kesempatan yang ada atau yang disediakan dosen/guru/instruktur untuk berdiskusi atau bertanya jawab sangat terbatas.
2. Memungkinkan terjadinya interaksi pembelajaran dari mana dan kapan saja (*time and place flexibility*). Mengingat sumber belajar yang sudah dikemas secara elektronik dan tersedia untuk diakses oleh peserta didik melalui internet, maka peserta didik dapat melakukan interaksi dengan sumber belajar ini kapan saja dan dari mana saja (Dowling, 2002). Demikian juga dengan tugas-tugas kegiatan

pembelajaran, dapat diserahkan kepada guru/dosen/instruktur begitu selesai dikerjakan. Tidak perlu menunggu sampai ada janji untuk bertemu dengan dosen/instruktur.

3. Menjangkau peserta didik dalam cakupan yang luas (*potential to reach aglobal audience*). Dengan fleksibilitas waktu dan tempat, maka jumlah peserta didik yang dapat dijangkau melalui kegiatan pembelajaran elektronik semakin lebih banyak atau meluas. Ruang dan tempat serta waktu tidak lagi menjadi hambatan. Siapa saja, dimana saja, dan kapan saja, seseorang dapat belajar. Interaksi dengan sumber belajar dilakukan melalui internet. Kesempatan belajar benar-benar terbuka lebar bagi siapa saja yang membutuhkan.
4. Mempermudah penyempurnaan dan penyimpanan materi pembelajaran (*easy updating of content as well as archivable capabilities*). Fasilitas yang tersedia dalam teknologi internet dan berbagai perangkat lunak (*software*) yang terus berkembang turut membantu mempermudah pengembangan bahan belajar elektronik. Demikian juga dengan penyempurnaan atau pemutakhiran bahan belajar sesuai dengan tuntutan perkembangan materi keilmuannya dapat dilakukan secara periodik dan mudah. Di samping itu, penyempurnaan metode penyajian materi pembelajaran dapat pula dilakukan, baik yang didasarkan atas umpan balik dari peserta didik maupun atas hasil penilaian guru/dosen/instruktur selaku penanggungjawab atau pembina materi pembelajaran itu sendiri.

## **2.2 M-Learning (Mobile Learning)**

*Mobile learning* adalah cara pembelajaran alternatif yang melibatkan perangkat bergerak sebagai sarana pembelajaran.

### **2.2.1 Pengertian M-Learning**

Clark Quinn (Quinn, 2010), berdasarkan riset *The eLearning Guild's 2007 Mobile Learning 360° Research Report* mendefinisikan *mobile learning* sebagai :

*“Any activity that allows individuals to be more productive when consuming, interacting with, or creating information, mediated through a compact digital portable device that the individual carries on a regular basis, has reliable connectivity, and fits in a pocket or purse”.*

Dari pendefinisian diatas dijelaskan bahwa *mobile learning* adalah segala aktifitas yang memungkinkan individu untuk menjadi lebih produktif dengan cara memperoleh atau memberi suatu informasi melalui media perangkat bergerak yang dapat dibawa kemanapun.

Melalui penggunaan teknologi perangkat bergerak, *mobile learning* memungkinkan setiap orang untuk mengakses informasi dan materi pembelajaran setiap saat. Peserta didik dapat mengontrol kapan dan dimana mereka ingin belajar sehingga tidak perlu mencari waktu tertentu untuk belajar atau pergi ke suatu tempat untuk belajar. Pendidik dan pelatih diberdayakan karena mereka dapat menggunakan teknologi ponsel untuk berkomunikasi dengan peserta didik dari mana saja dan kapan saja. Pada saat yang sama, pendidik dan pelatih dapat mengakses sumber belajar dari kapan saja dan di mana saja untuk merencanakan dan menyampaikan pelajaran mereka, (Ally, 2009).

### **2.2.2 Keuntungan *Mobile Learning***

Menurut Khalatbari dan Akhshabi bersaudara, ada beberapa keuntungan dari penggunaan *mobile learning* (Khalatbari dan Akhshabi, 2011), diantaranya sebagai berikut:

1. Dapat mengakses informasi tanpa terbatas ruang dan waktu
2. Kenikmatan dari metode fleksibel studi yang sesuai dengan kebutuhan pengguna
3. Tingkat kemajuan dalam studi tergantung pada pelajar. Dalam metode ini seperti kelas biasa ada kurikulum pembelajaran, panduan pelajaran dan referensi
4. Tidak ada beban untuk belajar
5. Akses mudah dari masyarakat terhadap pengetahuan dan informasi

6. Informasi lebih lanjut melekat dalam pikiran
7. Penekanan pada pemahaman pembelajaran lebih detail.
8. Peningkatan kerja sama kelompok antara peserta didik

## **2.3 Android**

Android adalah sebuah sistem sistem operasi untuk perangkat *mobile* berbasis *linux* yang meliputi sistem operasi, *middleware*, dan aplikasi yang dirilis oleh Google. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka (Safaat, 2011).

### **2.3.1 Sejarah Android**

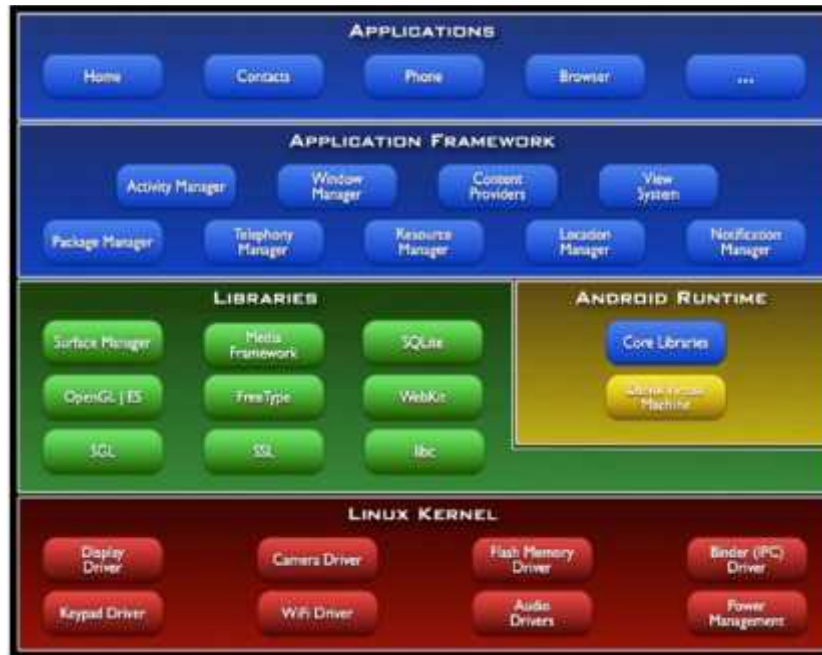
Pada tahun 2005 Google mengakuisisi Android Inc yang pada saat itu dimotori oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White. Kemudian untuk mengembangkan Android dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilis perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile* dan *smartphone* pertama yang menggunakan sistem operasi Android dikeluarkan oleh T-Mobile dengan sebutan G1 pada bulan September 2008.. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler.

Hingga saat ini Android telah merilis beberapa versi Android untuk menyempurnakan versi sebelumnya. Selain berdasarkan penomoran, pada setiap versi Android terdapat kode nama berdasarkan nama-nama makanan penutup. Hingga saat ini sudah terdapat beberapa versi yang telah diluncurkan, diantaranya: versi 1.5 dirilis pada 30 April 2009 diberi nama Cupcake, versi 1.6 dirilis pada 15 September 2009 diberi nama Donut, versi 2.0/2.1 dirilis pada 26 Oktober 2009 diberi nama Éclair, versi 2.2 dirilis pada bulan Mei 2010 diberi nama Froyo dan versi 2.3 dirilis pada Desember 2010 yang diberi nama Gingerbread.

### 2.3.2 Arsitektur Android

Dalam paket sistem operasi Android terdiri dari beberapa unsur seperti tampak pada gambar 2.1. Secara sederhana arsitektur Android merupakan sebuah kernel Linux dan sekumpulan pustaka C / C++ dalam suatu *framework* yang menyediakan dan mengatur alur proses aplikasi.



Gambar 2.1 Arsitektur Android (sumber: <http://developer.Android.com>)

#### 2.3.2.1 Applications dan Widgets

*Applications* dan *Widgets* ini adalah layer yang berhubungan dengan aplikasi saja. Biasanya aplikasi di *download* kemudian dilakukan instalasi dan jalankan aplikasi tersebut. Di layer ini terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, *browser*, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

#### 2.3.2.2 Applications Frameworks

Android merupakan “*Open Development Platform*” dimana Android menawarkan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *service background*, mengatur alarm, menambah status

pemberitahuan dan lainnya. Pengembang memiliki akses penuh menuju API *framework* seperti yang dilakukan oleh aplikasi yang kategori inti.

*Applications Frameworks* merupakan layer dimana para pembuat aplikasi melakukan pengembangan aplikasi yang akan dijalankan di sistem operasi Android, karena pada *layer* inilah aplikasi dapat dirancang dan dibuat.

Komponen-komponen yang termasuk dalam *Applications Frameworks* adalah sebagai berikut (developer.Android.com):

1. *Views*
2. *Content Provider*
3. *Resource Manager*
4. *Notification Manager*
5. *Activity Manager*

### **2.3.2.3 Libraries**

*Libraries* adalah layer di mana fitur-fitur Android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan di atas kernel, *layer* ini meliputi berbagai *library* C/C++ intin seperti Libc dan SSL, serta:

1. *Libraries* media untuk pemutaran media audio dan video
2. *Libraries* untuk manajemen tampilan
3. *Libraries Graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D
4. *Libraries* SSL dan *Webkit* terintegrasi dengan web browser dan *security*
5. *Libraries* LiveWebcore mencakup modern *web browser* dengan *security*
6. *Libraries* 3D yang mencakup implementasi OpenGL ES 1.0 API's

### **2.3.2.4 Android Runtime**

Pada *Android* tertanam paket pustaka inti yang menyediakan sebagian besar fungsi *Android*. *Android Runtime* merupakan mesin virtual yang membuat aplikasi *Android* menjadi lebih tangguh dengan paket pustaka yang telah ada.

Dalam *Android Runtime* terdapat 2 bagian utama, diantaranya (developer.Android.com):

1. Pustaka Inti, *Android* dikembangkan melalui bahasa pemrograman Java, tapi *Android Runtime* bukanlah mesin *virtual Java*. Pustaka inti *Android* menyediakan hampir semua fungsi yang terdapat pada pustaka Java serta beberapa pustaka khusus *Android*.
2. Mesin Virtual Dalvik, Dalvik merupakan sebuah mesin virtual yang dikembangkan oleh Dan Bornstein yang terinspirasi dari nama sebuah perkampungan yang berada di Iceland. Dalvik hanyalah interpreter mesin virtual yang mengeksekusi file dalam format *Dalvik Executable* (\*.dex). Dengan format ini Dalvik akan mengoptimalkan efisiensi penyimpanan dan pengalamanan memori pada file yang dieksekusi. Dalvik berjalan diatas kernel Linux 2.6, dengan fungsi dasar seperti *threading* dan manajemen memori yang terbatas.

### **2.3.2.5 Linux Kernel**

Linux *kernel* adalah *layer* di mana inti dari *operating* sistem dari Android iutu berada. Berisi *file-file system* yang mengatur sistem *processing*, *memory*, *resource*, *drivers*, dan sistem-sistem operasi android lainnya. Linux *kernel* yang digunakan android adalah linux *kernel* release 2.6.

### **2.3.3 Android SDK (Software Development Kit)**

Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk memulai pengembangan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang dirilis oleh Google. Untuk sumber SDK Android dapat dilihat dan diunduh langsung kesitus resmi Android di <http://developer.Android.com>.

### **2.3.4 Komponen Aplikasi**

Aplikasi Android ditulis dalam bahasa pemograman Java. Kode Java dikompilasi bersama dengan data file resource yang dibutuhkan oleh aplikasi, prosesnya di paket oleh *tools* yang dinamakan *atp tools* kedalam paket Android,



sehingga menghasilkan file dengan ekstensi *apk*. Ada 4 jenis komponen pada aplikasi Android, yaitu (<http://developer.Android.com>):

1. *Activities*

Suatu *Activity* akan menyajikan *user interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Pada umumnya sebuah aplikasi Android memiliki banyak *activity*. 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, kita dapat melakukan dengan satu even, misalnya klik tombol, memilih opsi atau menggunakan triggers tertentu. Secara hirarki sebuah windows *activity* dinyatakan dengan *method* *Activity setContentView()*. *ContentView* adalah objek yang berada pada *root hirarki*.

2. *Service*

*Service* tidak memiliki *Graphic User Interface* (GUI), tetapi *service* berjalan secara *background*. Komponen *service* diproses tidak terlihat, memperbaharui sumber data dan menampilkan notifikasi. *Service* digunakan untuk melakukan pengolahan data yang terus diproses, bahkan ketika *activity* tidak aktif.

3. *Broadcast Receiver*

*Broadcast receiver* berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. *Broadcast receiver* tidak memiliki *user interface*, tetapi memiliki sebuah *activity* untuk merespon informasi yang diterima atau menggunakan *notification manager* untuk memberi tahu pengguna, seperti lampu latar atau getaran.

4. *Content Provider*

*Content provider* membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam file seperti *database SQLite*. *Content provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika kita menggunakan aplikasi yang membutuhkan peta atau aplikasi yang

membutuhkan untuk mengakses data kontak dan navigasi, maka disinilah fungsi dari *content provider*.

### 2.3.5 Kelebihan Platform Android

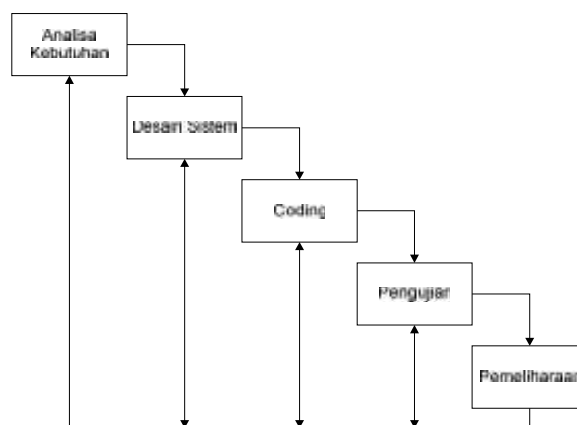
Persaingan *platform* atau sistem operasi semakin ketat, ini dapat dilihat dari banyaknya sistem operasi yang ada seperti, Symbian, iPhone, Windows Mobile, BlackBerry, Java Mobile Edition, Linux Mobile (LiM0), dan banyak lagi. Namun ada beberapa hal yang menjadi kelebihan Android menurut Burnette (Amiral, 2010). Walaupun beberapa fitur-fitur yang ada telah muncul sebelumnya pada platform lain, Android adalah yang pertama menggabungkan hal seperti berikut:

1. Keterbukaan, Bebas pengembangan tanpa dikenakan biaya terhadap sistem karena berbasis Linux dan *open source*. Pembuat perangkat menyukai hal ini karena dapat membangun *platform* yang sesuai yang diinginkan tanpa harus membayar *royalty*. Sementara pengembang *software* menyukai karena Android dapat digunakan diperangkat manapun dantapa terikat oleh vendor manapun.
2. Arsitektur komponen dasar Android terinspirasi dari teknologi internet *Mashup*. Bagian dalam sebuah aplikasi dapat digunakan oleh aplikasi lainnya, bahkan dapat diganti dengan komponen lain yang sesuai dengan aplikasi yang dikembangkan.
3. Banyak dukungan *service*, kemudahan dalam menggunakan berbagai macam layanan pada aplikasi seperti penggunaan layanan pencarian lokasi, *database SQL*, *browser* dan penggunaan peta. Semua itu sudah tertanam pada Android sehingga memudahkan dalam pengembangan aplikasi.
4. Siklus hidup aplikasi diatur secara otomatis, setiap program terjaga antara satu sama lain oleh berbagai lapisan keamanan, sehingga kerja sistem menjadi lebih stabil. Pengguna tak perlu khawatir dalam menggunakan aplikasi pada perangkat yang memorinya terbatas.

5. Dukungan grafis dan suarat terbaik, dengan adanya dukungan 2D grafis dan animasi yang diilhami oleh *Flash* menyatu dalam 3D menggunakan *OpenGL* memungkinkan membuat aplikasi maupun *game* yang berbeda.
6. Portabilitas aplikasi, aplikasi dapat digunakan pada perangkat yang ada saat ini maupun yang akan datang. Semua program ditulis dengan menggunakan bahas pemrograman Java dan dieksekusi oleh mesin virtual Dalvik, sehingga kode program portabel antara ARM, X86, dan arsitektur lainnya. Sama halnya dengan dukungan masukan seperti penggunaan *Keyboard*, layar sentuh, *trackball* dan resolusi layar semua dapat disesuaikan dengan program.

## 2.4 SDLC (*System Development Life Cycle*)

Model sistem yang dikembangkan dalam menganalisa perangkat lunak menggunakan metode konvensional dengan memanfaatkan model Siklus Hidup Pengembangan Sistem (*System Development Life Cycle*) atau yang lebih sering disebut *Waterfall Model*. *Waterfall* adalah sebuah model perkembangan perangkat lunak dilakukan secara sekuensial, dimana satu tahap dilakukan setelah tahap sebelumnya selesai dilaksanakan.



Gambar 2.2 Tahapan Metode *Waterfall* Menurut Sommerville (Wijaya, 2011)

### 2.4.1 Analisa Kebutuhan

Langkah ini merupakan analisa terhadap kebutuhan sistem meliputi identifikasi masalah, perumusan masalah, tujuan penelitian dan pengumpulan

data. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau studi literatur. Sistem analis akan menggali informasi sebanyak-banyaknya dari *user* sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh *user* tersebut. Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan sistem. Dokumen inilah yang akan menjadi acuan sistem analis untuk menerjemahkan ke dalam bahasa pemrograman.

#### **2.4.2 Desain Sistem**

Proses desain sistem menentukan arsitektur secara keseluruhan. Desain perangkat lunak mewakili fungsi sistem perangkat lunak dalam suatu bentuk yang dapat ditransformasikan ke dalam satu atau lebih program yang dapat dieksekusi

#### **2.4.3 Coding**

*Coding* merupakan penulisan kode program yaitu penerjemahan *design* sistem kedalam bahasa yang bisa dikenali oleh komputer. *Coding* dilakukan oleh *programmer* yang akan menerjemahkan transaksi yang diminta oleh *user*. Tahapan ini lah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini.

#### **2.4.4 Pengujian**

Tahapan akhir dimana sistem yang baru diuji kemampuan dan keefektifannya sehingga didapatkan kekurangan dan kelemahan sistem yang kemudian dilakukan pengkajian ulang dan perbaikan terhadap aplikasi menjadi lebih baik dan sempurna.

#### **2.4.5 Pemeliharaan**

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (peripheral atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.

## **2.5 Analisa dan Perancangan Berorientasi Objek**

Teknologi objek menganalogikan sistem aplikasi seperti kehidupan nyata yang didominasi oleh objek. Didalam membangun sistem berorientasi objek akan menjadi lebih baik apabila langkah awalnya didahului dengan proses analisis dan perancangan yang berorientasi objek. Tujuannya adalah untuk mempermudah *programmer* dalam mendesain program dalam bentuk objek-objek dan hubungan antar objek tersebut untuk kemudian dimodelkan dalam sistem nyata (A.Suhendar, 2002).

Perusahaan *software*, *Rational Software*, telah membentuk konsorsium dengan berbagai organisasi untuk meresmikan pemakaian *Unified Modelling Language* (UML) sebagai bahasa standar dalam *Object Oriented Analysis Design* (OOAD).

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

*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 (Dharwiyanti dan Wahono, 2006).

Untuk merancang sebuah model, UML memiliki beberapa diagram antara lain: *usecase diagram*, *class diagram*, *activity diagram* dan *sequence diagram*.

#### **2.5.1.1 Usecase Diagram**

*Usecase diagram* merupakan sebuah gambaran fungsionalitas sebuah sistem. Sebuah *usecase* merepresentasikan interaksi antara aktor dengan sistem. *Usecase* merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, *create* sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu (Dharwiyanti, 2006).

Dalam sebuah sistem *usecase diagram* akan sangat membantu dalam hal menyusun *requirement*, mengkomunikasikan rancangan dengan klien dan merancang *test case* untuk semua fitur yang ada pada sistem.

#### **2.5.1.2 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) (Dharwiyanti, 2006).

*Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. *Class* memiliki tiga area pokok yaitu, nama, *stereotype*, atribut dan metoda

### **2.5.1.3 Activity Diagram**

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi (Dharwiyanti, 2006).

*Activity diagram* merupakan *state diagram* khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-trigger oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

### **2.5.1.4 Sequence Diagram**

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang memicu aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan (Dharwiyanti, 2006).

## 2.6 Kajian Penelitian Sebelumnya

Beberapa penelitian terkait yang pernah dilakukan adalah sebagai berikut:

1. Penelitian oleh Aditia Budi Raharja yang dilakukan pada tahun 2012 berjudul “Aplikasi Kamus Bahasa Inggris – Bahasa Indonesia Dan Bahasa Indonesia – Bahasa Inggris Pada Telepon Seluler (Ponsel) Menggunakan Teknologi J2ME”. Penelitian ini membahas bagaimana menciptakan suatu aplikasi yang dapat membaca teks masukan dalam bahasa Inggris kemudian diterjemahkan dalam bahasa Indonesia begitu juga sebaliknya.
2. Penelitian oleh Joko Aris Pramono yang dilakukan pada tahun 2013 berjudul “Rancang Bangun Aplikasi *Text To Speech* Sebagai Alat Bantu Pembelajaran Bahasa Inggris”. Penelitian ini membahas bagaimana menciptakan suatu aplikasi yang digunakan untuk mengkonversikan sebuah teks kedalam bentuk ucapan dengan menggunakan pemodelan bahasa natural manusia. Dengan menggunakan aplikasi *Text To Speech* ini pengguna dapat mempelajari cara pengucapan bahasa Inggris dengan tepat, baik versi British ataupun versi American.
3. Penelitian yang dilakukan oleh Aris Setiawan pada tahun 2012 yang berjudul “Aplikasi *Mobile Learning* untuk Persiapan *Test of English for International Communication (TOEIC)* Berbasis *Android*. Penelitian ini membahas rancang dan bangun sebuah aplikasi yang dapat digunakan untuk membantu dalam persiapan diri sebelum mengikuti *TOEIC*