

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Rancang Bangun**

Rancang merupakan serangkaian prosedur untuk menerjemahkan hasil analisa dari sebuah sistem ke dalam bahasa pemrograman untuk mendeskripsikan dengan detail bagaimana komponen – komponen sistem diimplementasikan (Pressman, 2002). Perancangan adalah kegiatan yang memiliki tujuan untuk mendesain sistem baru yang dapat menyelesaikan masalah – masalah yang dihadapi perusahaan yang diperoleh dari pemilihan alternatif sistem yang terbaik (Ladjamudin, 2005). Sedangkan pengertian bangun atau pembangunan sistem adalah kegiatan menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun sebagian (Pressman, 2002). Bangun sistem adalah membangun sistem informasi dan komponen yang didasarkan pada spesifikasi desain (Whitten et al, 2004).

#### **2.2 Media Pembelajaran**

Arsyad (2014, p.3) mendefinisikan kata media berasal dari bahasa latin *medius* yang secara harfiah berarti ‘tengah’, perantara ‘atau’ pengantar. Media adalah perantara atau pengantar pesan dari pengantar atau pengirim ke penerima pesan menurut Gerlach & Ely (1971, p.3) mengatakan bahwa media apabila dipahami secara garis besar adalah manusia, materi, atau kejadian yang membangun kondisi kondisi yang membuat siswa mampu memperoleh pengetahuan, keterampilan atau sikap.

Pembelajaran adalah usaha-usaha yang terencana dalam memanipulasi sumber-sumber belajar agar terjadi proses belajar. Media pembelajaran dapat diartikan sebagai segala sesuatu yang dapat digunakan untuk menyalurkan pesan dari pengirim ke penerima sehingga merangsang pikiran, perasaan, perhatian, dan minat siswa sedemikian rupa sehingga proses belajar terjadi. Manfaat media pembelajaran secara umum memiliki kegunaan-kegunaan sebagai berikut :

- a. Memperjelas penyajian materi pembelajaran agar tidak terlalu verbalistik (dalam bentuk kata-kata tertulis atau lisan).
- b. Media pembelajaran dapat digunakan secara tepat, bervariasi, dan lebih interaktif mampu mengatasi sikap pasif anak didik.

### **2.3 DSLR Canon 60D**

*Digital SLR* merupakan singkatan dari *digital single lens reflect*. Secara sederhana, ia adalah proses kamera dalam menangkap gambar dan mengkonversinya menjadi sebuah data digital. *Image* yang di tangkap masuk melalui lensa lalu di konversi menjadi data digital. Kelebihan yang paling utama dari kamera ini adalah sensornya yang lebih besar dari pada *compact* dan *bridge camera*. Bahkan kamera DSLR dengan kualitas paling rendah sekalipun masih lebih baik dari pada *camera compact* yang harganya mahal. Secara fitur, kamera ini bisa di bilang lengkap. Ia memiliki semua fitur yang di miliki oleh *compact* dan juga *bridge camera* (Aditiawan, Rangga, 2015) Sedangkan DSLR Canon 60D merupakan kamera yang di produksi dari perusahaan Canon, dan 60D merupakan tipe dari kamera Canon.

### **2.4 Photographer**

Fotografer atau juru foto (Bahasa Inggris: *photographer*) adalah orang-orang yang membuat gambar dengan cara menangkap cahaya dari subyek gambar dengan kamera maupun peralatan fotografi lainnya, dan umumnya memikirkan seni dan teknik untuk menghasilkan foto yang lebih bagus serta berusaha mengembangkan ilmunya. Banyak fotografer yang menggunakan kamera dan alatnya sebagai pekerjaan untuk mencari penghasilan. (Mustikaning D.A. 2012)

### **2.5 Aplikasi Mobile**

Aplikasi adalah program yang digunakan orang untuk melakukan sesuatu pada sistem komputer. *Mobile* dapat diartikan sebagai perpindahan yang mudah dari satu tempat ke tempat yang lain, misalnya telepon *mobile* berarti bahwa terminal telepon yang dapat berpindah dengan mudah dari satu tempat ke tempat lain tanpa

terjadi pemutusan atau terputusnya komunikasi. Sistem aplikasi *mobile* merupakan aplikasi yang dapat digunakan walaupun pengguna berpindah dengan mudah dari satu tempat ketempat lain – lain tanpa terjadi pemutusan atau terputusnya komunikasi. Aplikasi ini dapat diakses melalui perangkat nirkabel seperti pager, seperti telepon seluler dan PDA. Adapun karakteristik perangkat *mobile* yaitu:

1. Ukuran yang kecil : Perangkat *mobile* memiliki ukuran yang kecil. Konsumen menginginkan perangkat yang terkecil untuk kenyamanan dan mobilitas mereka.
2. Memory yang terbatas : Perangkat *mobile* juga memiliki memory yang kecil, yaitu *primary (RAM)* dan *secondary (disk)*.
3. Daya proses yang terbatas : Sistem *mobile* tidaklah setangguh rekan mereka yaitu *desktop*.
4. Mengonsumsi daya yang rendah : Perangkat *mobile* menghabiskan sedikit daya dibandingkan dengan mesin desktop.
5. Kuat dan dapat diandalkan : Karena perangkat *mobile* selalu dibawa kemana saja, mereka harus cukup kuat untuk menghadapi benturan-benturan, gerakan, dan sesekali tetesan-tetesan air.
6. Konektivitas yang terbatas : Perangkat *mobile* memiliki bandwidth rendah, beberapa dari mereka bahkan tidak tersambung.
7. Masa hidup yang pendek : Perangkat – perangkat konsumen ini menyala dalam hitungan detik kebanyakan dari mereka selalu menyala.

## **2.6 Android**

Android adalah istilah dalam bahasa Inggris yang berarti “Robot yang menyerupai manusia”. Logo android sendiri dicerminkan seperti sebuah robot berwarna hijau, yang mengacu kepada arti kata Android. Android adalah sebuah sistem operasi untuk *smartphone* dan tablet. Sistem operasi dapat diilustrasikan sebagai ‘jembatan’ antara piranti (*device*) dan penggunaannya, sehingga pengguna bias berinteraksi dengan *device*-nya dan menjalankan aplikasi – aplikasi yang tersedia

pada *device*. Di dunia personal computer, sistem operasi yang banyak dipakai adalah *Windows*, *Mac*, dan *Linux*.

Android dikembangkan bersama oleh perusahaan-perusahaan yang tergabung dalam sebuah konsorsium bernama *Open Handset Alliance (OHA)*. *OHA* dipimpin oleh *Google* dan didirikan bersama dengan 34 perusahaan lainnya, dengan tujuan untuk mengembangkan teknologi *mobile device*, semikonduktor, pembuatan aplikasi, komersialisasi, dan *mobile operator*.

Android adalah sistem operasi yang bersifat *open source* (sumber terbuka). Disebut *open source* karena *source code* (kode sumber) dari sistem operasi android dapat dilihat didownload, dan dimodifikasi secara bebas. Paradigma *open source* ini memudahkan pengembangan teknologi android, karena semua pihak yang tertarik dapat memberikan kontribusi, baik pada pengembangan sistem operasi maupun aplikasi (Alfa Satyaputra, Eva Maulina Aritonang, 2014).

## **2.7 Framework**

Huda (2013) Sistem operasi Android didukung oleh banyak *framework* untuk mengembangkan aplikasi. Aplikasi android secara *native* dapat dikembangkan dengan bahasa *java* atau C, sedangkan *framework*-nya menggunakan editor *Eclipse* dan SDK Android. Selain itu, android bisa juga dikembangkan menggunakan *framework* lain seperti :

1. *Basic4android*: Bahasa pemrograman Basic
2. Adobe AIR: Bahasa pemrograman *ActionScript*.
3. *phoneGap*: Bahasa Pemrograman HTML5 dan *JavaScript*.
4. PHP for Android: Bahasa pemrograman PHP.
5. *AndEngine: Framework* untuk mengembangkan game Android.

## **2.8 Java**

*Java* dalam ilmu komputer, merupakan bahasa pemrograman berorientasi objek yang diperkenalkan pada tahun 1995 oleh Sun Microsystems Inc., yang saat *Java*

diciptakan, dipimpin oleh James Gosling (Nugroho, 2008). Bahasa pemrograman *java* tercipta berawal dari sebuah perusahaan *Sun Microsystem* yang ingin membuat sebuah bahasa pemrograman yang dapat berjalan di semua *device* tanpa harus terikat oleh *device* tersebut, terlaksanalah sebuah proyek yang dipelopori oleh Patrick Naughton, James Gosling, Mike Sheriden dan Bill Joy pada tahun 1991, maka terciptalah bahasa pemrograman *java* yang awalnya bernama “Oak”.

*Java* adalah sebuah teknologi dimana teknologi tersebut mencakup *java* sebagai bahasa pemrograman yang memiliki sintaks dan aturan pemrograman tersendiri, juga mencakup *java* sebagai *platform* dimana teknologi ini memiliki *virtual machine* dan *library* yang diperlukan untuk menulis dan menjalankan program *java* adalah keinginan akan terbentuknya suatu bahasa pemrograman *java* adalah keinginan akan terbentuknya suatu bahasa pemrograman yang bisa berjalan di berbagai perangkat tanpa harus terikat oleh *platform*, sehingga *java* ini bersifat portable dan *platform independent* (tidak tergantung mesin atau sistem operasi).

## **2.9 Aspek Rekayasa Perangkat Lunak dalam media pembelajaran**

Kriteria penilaian dalam aspek rekayasa perangkat lunak yang akhirnya disetujui dalam diskusi di tim penyusun (LIPI, Pustekkom, Ilmu Komputer.com) menurut wahono (2006) adalah :

1. Efektif dan efisien dalam pengembangan maupun penggunaan media pembelajaran
2. *Reliabilitas* (Kehandalan)
3. *Maintainabilitas* (Dipertahankan/Dapat dipelihara/dikelola dengan mudah)
4. *Usabilitas* (Mudah digunakan dan sederhana dalam pengoperasiannya)
5. Ketepatan pemilihan jenis aplikasi/*software/tool* untuk pengembangan
6. Kompatibilitas (Media pembelajaran dapat diinstalasi/dijalankan diberbagai *hardware* dan *software* yang ada)
7. Pemaketan program media pembelajaran terpadu dan mudah dalam eksekusi
8. Dokumen program media pembelajaran yang lengkap

9. *Reusable* (Sebagian atau seluruh program media pembelajaran dapat dimanfaatkan kembali untuk mengembangkan media pembelajaran)

Wahono (2010) menjelaskan bahwa terdapat 6 (enam) langkah mudah mengembangkan multimedia pembelajaran itu, yaitu :

1. Tentukan jenis multimedia pembelajaran

Jenis multimedia pembelajaran menurut kegunaannya ada 2 macam :

- a. Multimedia persenasi pembelajaran

Alat bantu guru dalam proses pembelajaran di kelas dan tidak menggantikan guru secara keseluruhan. Berupa *point-point* materi yang disajikan (*explicit knowledge*) dan bisa saja ditambahi dengan multimedia linier berupa film dan video untuk memperkuat pemahaman siswa. Dapat dikembangkan dengan *software* persentasi seperti : *Open Office Impress, Microsoft Power Point* dsb.

- b. Multimedia Pembelajaran Mandiri

*Software* pembelajaran yang dapat dimanfaatkan oleh siswa secara mandiri alias tanpa bantuan guru. Multimedia pembelajaran mandiri harus dapat memadukan *explicit knowledge* (pengetahuan tertulis yang ada di buku, artikel dan lain – lain) dan *tactic knowledge* (*know how, rule of thumb*, pengalaman guru).

2. Tentukan materi ajar
3. Susun *Storyboard*
4. Mulai membuat
5. Gunakan teknik amati, tiru dan modifikasi
6. Tetapkan Target

## **2.10 Adobe Integrated Runtime (Adobe AIR)**

Adobe AIR (*Adobe Integrated Runtime*) adalah sebuah *cross operating system runtime* yang memungkinkan pengembang membangun RIA (*Rich Internet Application*) dan contentnya ke dalam *platform* baru dengan menggunakan *feature adobe flash, adobe flax, HTML* atau *Ajax*. Adobe AIR dikembangkan oleh adobe

dan dirilis pertama kali pada *adobe professional CS5* yang kemudian mengalami pengembangan pada versi selanjutnya. Tim pengembang adobe telah mengumumkan kompatibilitas adobe AIR dengan *platform* android, sehingga para pengembang aplikasi sudah dapat menggunakan *flash builder* atau *flash professional CS5/CS6* untuk membuat aplikasi android yang menarik.

### 2.11 Android Package (APK)

APK adalah paket aplikasi Android (*Android PacKage*). APK umumnya digunakan untuk menyimpan sebuah aplikasi atau program yang akan dijalankan pada perangkat Android. APK pada dasarnya seperti zip file, karena berisi dari kumpulan file. APK dapat diperoleh melalui berbagai metode, seperti menginstal sebuah aplikasi melalui #Market, *download* dari sebuah situs web, atau membuat sendiri dengan bahasa *Java*. Jika Anda memiliki file APK pada komputer Anda, dan ingin menginstalnya pada telepon Anda, Anda dapat menjalankan perintah ‘adb install apkname.apk’ untuk install aplikasi melalui USB ke telepon Anda.

File APK berada di perangkat Android di direktori `/data/app/filename.apk`. Direktori ini biasanya tidak dapat diakses, untuk melindungi aplikasi berbayar, kecuali *rooted* ROM digunakan di telepon. Pada *rooted* ROM, sangat mungkin untuk menyalin aplikasi ke folder ini secara manual dengan menggunakan perintah ‘adb push’ atau semacamnya. Setelah sebuah file berada di direktori `/data/app`, maka aplikasi sudah terpasang pada telepon.

Selain itu dimungkinkan juga untuk menginstal aplikasi ke partisi #sistem dari telepon yang sudah di-root. Cara ini memiliki keuntungan yaitu aplikasi tersebut tidak dapat dihapus dari perangkat, sehingga berguna untuk keamanan/anti-pencurian aplikasi seperti *Wave Secure* dan *Mobile Defense*. Sistem aplikasi ini akan disimpan dalam `/system/app`. Aplikasi dalam folder ini biasanya bagian dari sistem operasi Android, dan sebaiknya jangan diubah atau dihapus kecuali Anda mengetahui apa yang Anda lakukan.

File APK dapat dihapus dengan banyak cara, antara lain: menggunakan aplikasi #Market (jika aplikasi diinstal dari Market), perintah ‘uninstall adb apkname.apk’, melalui menu *Settings, Applications, Manage Applications*, atau dengan menghapus secara manual file-file dari direktori /data/app. Namun biasanya metode yang sering digunakan adalah melalui market atau menu aplikasi.

## **2.12 Adobe Flash**

Adobe Flash adalah salah satu perangkat lunak komputer yang merupakan produk unggulan Adobe Systems. Adobe Flash sebelumnya bernama Macromedia Flash. Adobe Flash digunakan untuk membuat gambar vektor maupun animasi gambar tersebut. Berkas yang dihasilkan dari perangkat lunak ini mempunyai *file extension* swf dan dapat diputar di penjelajah web yang telah dipasang Adobe Flash Player. Flash menggunakan bahasa pemrograman bernama *Action Script* yang muncul pertama kalinya pada Flash 5.

### **2.12.1 Adobe Flash Professional CS6**

Andi (2013, p.1) mendefinisikan, *Adobe Flash Professional CS6* merupakan perangkat lunak multiguna yang dapat dimanfaatkan untuk berbagai macam animasi dengan berbagai fitur canggih yang ada didalamnya dan juga dapat menggambar, membuat animasi, hingga digunakan untuk membuat game yang menarik dan berkualitas. Area kerja *Adobe Flash Professional CS6* dirancang secara khusus agar ruang kerja yang digunakan dapat diatur dan lebih mudah dipahami oleh pemakai pemula maupun para desainer *flash* yang telah berpengalaman. *Adobe Flash Professional CS6* adalah program yang cukup kompleks sehingga tidak mungkin untuk menjabarkannya secara lengkap. Disini hanya akan diberikan sedikit pengertian dan fungsi dasar dari *Adobe Flash Professional CS6*.

#### **2.12.1.1 User Interface**

Tampilan awal dalam membuka program *Adobe Flash Professional CS6* yang merupakan scene dimana kita akan memulai membuat sebuah flash. Kemudian



tampilan bagian kanan adalah panel yang berisi *tool*, *color*, dan semacamnya yang berfungsi untuk memberi variasi pada tampilan. Sedangkan tampilan bagian bawah merupakan *timeline*.

#### **2.12.1.2 Toolbox**

*Toolbox* adalah sebuah panel yang menampung tombol – tombol yang berguna untuk membuat suatu desain animasi mulai dari tombol seleksi, *pen*, *pencil*, *3D rotation*, dan lain – lain.

#### **2.12.1.3 Timeline**

*Timeline* berguna untuk menentukan durasi animasi, jumlah *layer*, *frame*, menempatkan *script* dan beberapa keperluan animasi lainnya. Semua bentuk animasi yang anda buat akan diatur dan ditempatkan pada *layer* dalam *timeline*.

#### **2.12.1.4 Stage**

*Stage* adalah lembar kerja yang digunakan untuk membuat atau mendesain objek yang akan dianimasikan. Objek yang dibuat dalam lembar kerja dapat berupa objek *Vektor*, *Movie clip*, *Text*, *Button*, dan lain – lain.

### **2.12.2 Sejarah Action Script**

*Action Script* pertama kali dikenalkan pada tahun 1996 pada *flash* versi 4. Saat itu namanya belum dikenal sebagai *Action Script* karena kita tidak menuliskan kode-kode program, tetapi hanya memberikan *action* dengan perintah-perintah yang terdapat pada menu *drop down*.

Pada tahun 2000, *flash* 5 hadir dengan performa yang lebih baik dari versi sebelumnya dan secara resmi meresmikan *Action Script* 1.0 yang dikembangkan pada bahasa yang berbasis web seperti *Macromedia Direction* dan *Sun Java*. *Action* 1.0 ini sampai digunakan sampai *Flash MX* atau *Flash6*.

*Flash MX 2004* atau *Flash 7* hadir dengan membawa *Action Script 2.0* sebagai generasi penerus *Action Script 1.0*. AS2 merupakan Bahasa pemrograman berbasis objek yang lebih dekat dengan standar *ECMAScript* sehingga versi kedua ini jauh lebih kuat dan mudah untuk membuat program.

*Action Script 3.0* merupakan perkembangan terakhir dari *Action Script* saat ini. Kehadirannya membawa perubahan yang signifikan, menjadikan AS3 sebagai bahasa perograman yang lebih kompleks, cepat, dan powerful untuk membuat *game* (Irman Maulana, 2014).

### **2.12.3 Action Script 3.0**

*Action Script 3.0* atau disingkat AS3 merupakan bahasa pemrograman yang bekerja pada Adobe Flash, Flex, dan FlashDevelop, *Action Script 3.0* pertama kali di liris pada tahun 2006 bersama dengan diluncurkannya flash versi 9 sekaligus flash pertama yang kini telah diakusisi oleh *Adobe System Inc.* yaitu Adobe Flash CS3. *ActionScript 3.0* adalah bahasa pemrograman yang didasarkan pada *ECMAScript*, yaitu standar bahasa pemrograman yang dikembangkan oleh ECMA (*European Computer Manufacture Association*). Dengan standar ini, AS3 mampu melakukan integrasi data yang cepat dengan berbagai bahasa pemrograman lainnya seperti *JavaScript* dan XML.

Jika dibandingkan dengan versi sebelumnya, AS3 dapat menghasilkan performa yang lebih baik dalam pembuatan game, namun semakin kompleks dari segi struktur bahasa. Ada sedikit perbedaan antara AS2 dan AS3 dalam penulisan *script* yaitu, AS3 tidak bias dituliskan langsung pada symbol movie clip ataupun button seperti pada AS2 sehingga kita bisa mengontrol movie hanya dengan menuliskan script pada frame atau memulai file kelas (Irman Maulana, 2014).

Salah satu fungsi *ActionScript* adalah memberikan sebuah konektivitas terhadap sebuah objek, yaitu dengan menuliskan perintah-perintah didalamnya. Tiga hal yang harus diperhatikan dalam *ActionScript* yaitu:

## 1. Event

*Event* merupakan peristiwa atau kejadian untuk mendapatkan aksi sebuah objek. *Event* pada Adobe Flash CS 6 ada empat yaitu:

*Mouse event*, *Event* yang berkaitan dengan penggunaan *mouse*.

*Keyboard Event*, Kejadian pada saat menekan tombol *keyboard*. *Frame Event*, *Event* yang diletakkan pada *keyframe*.

*Movie Clip Event*, *Event* yang disertakan pada *movie clip*.

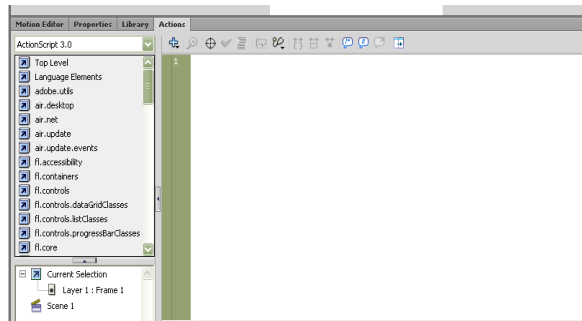
## 2. Target

Target adalah objek yang dikenai aksi atau perintah. Sebelum dikenai aksi atau perintah, sebuah objek harus dikonversi menjadi sebuah simbol dan memiliki nama instan. Penulisan nama target pada skrip harus menggunakan tanda petik ganda (" ").

## 3. Action

Pemberian *action* merupakan langkah terakhir dalam pembuatan interaksi antarobjek. *Action* dibagi menjadi dua antara lain:

1. *Action Frame*: adalah *action* yang diberikan pada *keyframe*. Sebuah *keyframe* akan ditandai dengan huruf a bila pada *keyframe* tersebut terdapat sebuah *action*.
2. *Action Objek*: adalah *action* yang diberikan pada sebuah objek, baik berupa tombol maupun *movie clip*.
3. *Action Script* diketikkan pada panel *actions* yang tersedia pada *software*. *Action Script* hanya dapat dituliskan pada objek yang bertipe *Movie Clip*, *keyframe*, *Button*, dan objek *components*. *Action Script* tidak dapat digunakan pada objek tulisan atau gambar lain yang bukan bertipe *Movie Clip*. Jadi bila ingin menggunakan *ActionScript* pada suatu objek, objek tersebut harus diubah menjadi *Movie Clip* terlebih dahulu. Untuk membuka *Panel Actions*, klik tulisan *Action* yang ada pada jendela panel. Berikut tampilan dari panel *Action Script* pada Adobe Flash Professional CS6 dapat dilihat pada gambar 2.1 berikut ini :

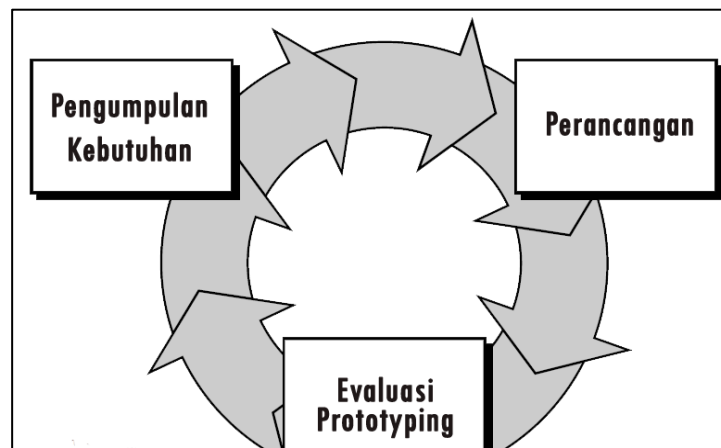


Gambar 2.1 Panel Action

## 2.13 Metode Pengembangan Sistem

### 2.13.1 Metode *Prototype*

Model prototipe (*prototyping model*) dimulai dari mengumpulkan kebutuhan pelanggan terhadap perangkat lunak yang akan dibuat. Lalu dibuatlah program prototipe agar pelanggan lebih terbayang dengan apa yang sebenarnya diinginkan. Program prototipe biasanya merupakan program yang belum jadi. Program ini biasanya menyediakan tampilan dengan simulasi alur perangkat lunak sehingga tampak seperti perangkat lunak yang sudah jadi. Program prototipe ini dievaluasi oleh pelanggan atau user sampai ditemukan spesifikasi yang sesuai dengan keinginan pelanggan atau user. Rosa A.S. dan M. Shalahuddin (2015, p.31). Berikut adalah gambar dari model prototipe :

Gambar 2.2 Model *Prototype*

Tahap–tahap rekayasa *software* dalam *prototype model* pada gambar 2.2 di atas adalah sebagai berikut:

1) Pengumpulan kebutuhan

Developer dan klien bertemu untuk menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya. Detail kebutuhan mungkin tidak dibicarakan disini, pada awal pengumpulan kebutuhan. Selanjutnya peneliti akan melakukan analisis terhadap data apa saja yang dibutuhkan, seperti analisis terhadap sistem yang berjalan, analisis kebutuhan perangkat lunak, analisis kebutuhan perangkat keras.

2) Perancangan

Perancangan dilakukan dengan cepat dan rancangan mewakili semua aspek *software* yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*. Dalam tahap ini peneliti akan membangun sebuah versi *prototype* yang dirancang kembali dimana masalah-masalah tersebut diselesaikan

3) Evaluasi *prototype*

Pada tahap ini, calon pengguna mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan *software*. *Software* yang sudah jadi dijalankan dan akan dilakukan perbaikan apabila kurang memuaskan. Perbaikan termasuk dalam memperbaiki kesalahan atau kerusakan yang tidak ditemukan pada langkah sebelumnya.

Kelebihan *Prototype Model* adalah:

- 1) End user dapat berpartisipasi aktif.
- 2) Penentuan kebutuhan lebih mudah diwujudkan.
- 3) Mempersingkat waktu pengembangan *software*.

Kekurangan *Prototype Model* adalah:

- 1) Proses analisis dan perancangan terlalu singkat.
- 2) Mengesampingkan alternatif pemecahan masalah.
- 3) Biasanya kurang fleksibel dalam menghadapi perubahan.
- 4) *Prototype* yang dihasilkan tidak selamanya mudah dirubah.
- 5) *Prototype* terlalu cepat selesai.

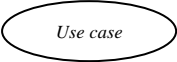
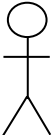
## 2.14 UML (*Unified Modelling Language*)

UML adalah keluarga notaris grafis yang didukung oleh model – model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi obyek. UML merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek (Whitten, 2004). Jeffrey L. Whitten menambah bahwa UML menawarkan diagram yang dikelompokkan menjadi beberapa perspektif berbeda untuk memodelkan suatu sistem.

### 2.14.1 *Use Case Diagram*

Diagram yang menggambarkan interaksi antara sistem dengan eksternal sistem dan pengguna. Dengan kata lain, secara grafis menggambarkan siapa yang akan menggunakan sistem dan dengan cara apa pengguna mengharapkan untuk berinteraksi dengan sistem. Adapun simbol dari *use case* diagram terdapat pada tabel 2.1 berikut ini:

Tabel 2.1 *Use Case*

Nama Komponen	Keterangan	Simbol
<i>Use Case</i>	<i>Use case</i> digambarkan sebagai lingkaran elips dengan nam <i>use case</i> dituliskan didalam elips tersebut.	
<i>Actor</i>	<i>Actor</i> adalah pengguna sistem. <i>Actor</i> tidak terbatas hanya manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan <i>input</i> atau memberikan <i>output</i> , maka aplikasi tersebut juga bias dianggap sebagai <i>actor</i> .	











Tabel 2.1 *Use Case* (Lanjut)

<i>Association</i>	Asosiasi digunakan untuk menghubungkan <i>actor</i> dengan <i>use case</i> . Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara actor dengan use case.	_____
--------------------	--	-------

### 2.14.2 Activity Diagram

Activity diagram secara garis digunakan untuk menggambarkan rangkaian aliran aktifitas baik proses bisnis atau *use case*. Adapun symbol – symbol dari activity diagram terdapat pada tabel 2.2 berikut ini :



Tabel 2.2 Simbol *Activity* Diagram

Simbol	Keterangan
	Titik awal
	Titik akhir
	<i>Activity</i>
	Pilihan untuk mengambil keputusan
	Fork; digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel menjadi satu.
	Rake; menunjukkan adanya dekomposisi.
	Tanda waktu
	Tanda pengirim
	Tanda penerima
	Aliran akhir ( <i>Flow final</i> )

### 2.14.3 Sequence Diagram

*Sequence* diagram secara grafis menggambarkan bagaimana object berinteraksi dengan satu sama lain melalui pesan pada eksekusi sebuah *use case* atau proses. Diagram ini mengilustrasikan bagaimana pesan terkirim dan diterima di antara objek dan *sequence* (Urutan). Adapun simbol-simbol dari *sequence* diagram terdapat pada tabel 2.3 berikut:

Tabel 2.4 Simbol *Sequence* Diagram

Nama Komponen	Komponen	Sombol
<i>Lifeline</i>	<i>Lifeline</i> mengindikasikan keberadaan sebuah <i>object</i> dalam basis waktu. Notasi untuk <i>Lifeline</i> adalah garis putus-putus <i>vertikal</i> yang ditarik dari sebuah <i>object</i> .	
<i>Activation</i>	<i>Activation</i> dinotasikan sebagai sebuah kontak segi empat yang gambar pada sebuah <i>lifeline</i> . Mengindikasikan sebuah <i>object</i> yang akan melakukan sebuah aksi.	
<i>Message</i>	<i>Message</i> , digambarkan dengan anak panah <i>horizontal</i> antara <i>activation</i> message mengindikasikan komunikasi antara <i>object-object</i> .	