

BAB 2

LANDASAN TEORI

2.1 Multimedia

2.1.1 Definisi Multimedia

Multimedia adalah sebuah fungsi dari komputer yang mempresentasikan dan menggabungkan teks, grafik, suara, video dan animasi sehingga *user* dapat berinteraksi, menciptakan, berkomunikasi dan mengendalikan elemen-elemen tersebut (Hofstetter, 2001, p2). Definisi tersebut mendefinisikan empat komponen yang harus ada pada multimedia. Pertama, harus ada sebuah komputer untuk mengkoordinasi apa yang kita lihat, dengar, dan berinteraksi dengan kita. Kedua, harus ada sebuah hubungan ke informasi. Ketiga, harus ada alat navigasi yang memungkinkan kita mengakses informasi tersebut. Dan keempat, karena multimedia bukan hanya menyaksikan, maka harus ada cara untuk memperoleh, memproses, dan berkomunikasi dengan informasi dan ide kita.

Multimedia muncul sebagai sebuah kemampuan dasar yang sangat penting pada abad ke-21, bahkan multimedia merubah cara membaca itu sendiri. Multimedia mendobrak batasan dari teks dan memberikan dimensi baru dari membaca dengan menambahkan dan menyajikannya lengkap dengan suara, musik, gambar, dan video. Multimedia terbukti efektif berdasarkan penelitian oleh *Computer Technology Research (CTR)*. CTR

menyatakan bahwa 20% manusia menyerap apa yang mereka lihat, 30% apa yang mereka dengar, 50% apa yang mereka lihat dan dengar, dan 80% apa yang mereka lihat, dengar, dan lakukan saat itu. Maka dari itu multimedia menjadi sangat efektif dalam pembelajaran. Multimedia juga akan membantu menyebarkan informasi kepada jutaan orang yang bahkan tidak memiliki komputer dan tidak mengerti komputer.

2.1.2 Elemen Multimedia

Ada 5 elemen multimedia yang dijelaskan Hofstetter (2001, p16) pada bukunya *Multimedia Literacy*, yaitu :

1. Teks

Walaupun tidak mustahil untuk menciptakan sebuah multimedia tanpa teks, tetapi kebanyakan sistem multimedia menggunakan teks karena teks adalah cara efektif untuk mengkomunikasikan ide-ide dan menyediakan instruksi bagi *user*. Adapun teks digolongkan menjadi :

a. *Printed text*

Teks biasa yang muncul dikertas biasanya digunakan untuk dokumentasi dari multimedia. Agar multimedia dapat membaca *printed text*, maka harus dirubah dulu menjadi teks komputer dengan mengetik dengan aplikasi *Word Processor* atau meng-*scan* teks tersebut.

b. *Scanned text*

Sebuah *scanner* merubah *printed text* menjadi sebuah bentuk yang bisa dibaca mesin dan merubahnya kembali menjadi *scanned text*. Penggunaan *scanner* biasanya dilakukan untuk menghemat waktu dan tenaga dibanding menetik langsung ke dalam komputer.

c. *Electronic text*

Electronic text adalah bentuk yang dapat dibaca oleh mesin, yang biasa diinput menggunakan aplikasi *Word Processor*. *Electronic text* sekarang digunakan secara luas baik untuk pembuatan buku, majalah, dan koran.

d. *Hypertext*

Hypertext adalah salah satu komponen dari multimedia yang membuat multimedia menjadi interaktif. Kata *hypertext* pertama kali dikemukakan oleh Ted Nelson (1965) yang mengandung arti teks yang berhubungan. Jika sebuah *hypertext* diklik maka aplikasi akan menampilkan informasi. Maka dari itu, sebuah *hypertext* menjadi bagian dari multimedia karena *hypertext* dapat menjadi objek yang berupa hubungan ke objek lain.

2. Grafik

Multimedia dapat merubah gambar menjadi objek atau *link*. Terkadang grafik juga muncul sebagai latar belakang dari teks. Selain itu gambar juga bisa berupa *icon* yang digabung dengan teks, menampilkan pilihan, atau gambar bisa ditampilkan secara *full-screen* sebagai ganti

dari teks, dengan bagian dari gambar sebagai objek atau *link* untuk menampilkan *event-event* atau objek-objek lain. Ada beberapa bentuk dari grafik yaitu :

a. *Bitmap*

Bitmap adalah gambar yang disimpan sebagai sebuah set dari *pixel-pixel* yang berhubungan dengan layar komputer. Untuk menampilkan gambar, komputer menempatkan setiap titik pada layar dan warna sesuai pada *bitmap*.

b. *Vector images*

Vector images disimpan sebagai sebuah set dari operasi matematika atau algoritma yang mendefinisikan kurva, garis, dan bentuk dalam sebuah gambar. Untuk gambar yang tidak memiliki banyak perubahan warna, *vector images* lebih efektif dibandingkan *bitmap*. *Vector images* memiliki dua kelebihan dibandingkan *bitmap*. Pertama *vector image* bisa diperkecil atau diperbesar lebih sempurna tanpa mengurangi kualitas gambar. Kedua, karena *vector images* memiliki ukuran *file* yang lebih kecil, maka lebih mudah *download* menggunakan *internet*.

c. *Clip art*

Untuk menghemat waktu dalam pembuatan aplikasi multimedia, kita dapat menggunakan sebuah *library* yang berisi *clip art*. Ada beberapa kategori dari *clip art* termasuk foto, *icon*, animasi, latar belakang, dan *button*.

d. *Digitized pictures*

Digitized picture adalah gambar yang didapatkan dari sebuah *frame* dari rekaman kamera, VCR, VCD atau *live video* lain yang di-*capture* dan dapat digunakan pada aplikasi multimedia.

e. *Hyperpictures*

Hyperpictures adalah sebuah gambar dimana bagian-bagiannya dapat digunakan sebagai objek sebagai pemicu objek lain atau *event-event* pada aplikasi multimedia.

3. Suara

Ada 4 tipe dari suara yang dapat dipakai pada aplikasi multimedia yaitu :

a. *Waveform audio*

Waveform audio didapat dari hasil perekaman dari apa yang kita dengar. Setiap suara memiliki *waveform* yang menggambarkan frekuensi, amplitudo, dan harmoni. Aplikasi *waveform audio digitizers* menangkap suara dengan mengambil contoh *waveform* ribuan kali perdetik dan menyimpannya ke dalam *harddisk* berupa *file* ber-*extension wav* (.wav).

b. MIDI

MIDI adalah singkatan dari *Musical Instrument Digital Interface* yang menyediakan sebuah cara yang efisien untuk merekam musik. MIDI tidak merekam *waveform* dari suara yang mana membutuhkan ruang *harddisk* yang besar, tetapi merekam

performance information yang dibutuhkan oleh komputer untuk memainkan musik. *File* MIDI disimpan pada *harddisk* dengan ekstension mid (.mid).

c. Audio CD

Audio CD dapat menampung sampai 75 menit suara berkualitas tinggi. Rata-rata samplangnya bisa mencapai 44.100 sampel perdetik yang cukup cepat untuk merekam semua suara yang terdengar oleh manusia. Suara yang direkam bisa berupa bisikan lembut sampai teriakan keras. *Addressing* yang digunakan CD-ROM mendukung komputer multimedia untuk mengakses secara acak sebuah lagu dari CD dengan akurasi sampai 1/75 detik.

d. MP3

MP3 singkatan dari MPEG Audio Layer 3. MP3 adalah format *file* audio yang menggunakan *codec audio* MPEG untuk mengkompres dan mendekompres suara yang direkam. MP3 dapat mengkompres sebuah *track* CD menjadi *file* yang lebih kecil dan membutuhkan *bandwidth* yang lebih kecil jika ditransfer melalui internet tanpa perlu mengkhawatirkan kualitas hasilnya.

4. Video

Video menyediakan sumber daya yang besar bagi aplikasi multimedia. Ada beberapa macam video yang dapat digunakan sebagai objek dalam aplikasi multimedia yaitu :

a. *Live Video Feeds*

Live video feeds menyediakan objek multimedia yang *real-time*.

Live video feeds bisa didapat dari siaran televisi atau dari satelit secara langsung.

b. *Videotape*

Videotape bisa menjadi objek multimedia tetapi dibatasi oleh dua faktor. Pertama, informasi yang disimpan di dalam *videotape* bersifat linear dan jika ingin mengakses informasi yang kita inginkan, kita harus mempercepat atau mengulang kembali rekaman yang kita tonton sehingga jelas membuang waktu. Kedua, *videotape* tidak bisa dikontrol komputer, sehingga kita harus menekan tombol-tombol video sendiri pada saat presentasi multimedia.

c. *Video Disc*

Ada dua format *video disc* yaitu CLV dan CAV. *CAV disc* dapat menyimpan sampai 54.000 *frame* atau sekitar 30 menit video dan suara stereo. *CAV* juga bisa menampilkan *frame* diam. *CLV disc* bisa menyimpan sampai satu jam video pada tiap sisinya. Yang berarti dua kali lipat *CAV disc*. Tetapi *CLV disc* membutuhkan *player* yang lebih mahal dibanding *CAV*. *Video disc* sangat populer pada abad 20, tetapi sekarang keberadaannya tergeser oleh digital video dan DVD.

d. *Digital Video*

Digital video adalah media paling menjanjikan dan sebuah media *storage* yang baik. Sama seperti *waveform audio*, *digital video* dapat disimpan pada *harddisk*, CD-ROM, atau DVD. Karena video tersebut *digital*, maka bisa digunakan dalam jaringan tanpa perlu *video tapes* atau *videodisc player*. *Digital video* juga dapat diakses secara acak untuk memilih klip mana yang hendak kita putar.

5. Animasi

Animasi adalah penggunaan komputer untuk menciptakan pergerakan di layar. Ada empat macam animasi, yaitu :

a. *Frame Animation*

Frame animation membuat pergerakan objek dengan menampilkan gambar-gambar (*frame*) secara berurutan dan tampil pada lokasi berbeda di layar.

b. *Vector Animation*

Vector animation membuat pergerakan objek dengan merubah permulaan, arah dan ukuran dari objek dan disesuaikan dengan segmen objek.

c. *Computational Animation*

Pada *computational animation*, untuk membuat objek bergerak adalah dengan merubah koordinat x dan y pada layar.

d. *Morphing*

Morphing berarti transisi dari satu bentuk ke bentuk lain dengan menampilkan *serial frame* dan menciptakan gerakan halus antara perpindahan satu bentuk ke bentuk lain.

2.1.3 Aplikasi Multimedia

Dengan semakin berkembangnya teknologi multimedia pada awal tahun 1990, maka multimedia mulai merambah dan berpengaruh pada beberapa bidang dan sekarang multimedia menjadi sangat penting dalam komunikasi. Adapun bidang-bidang yang dipengaruhi oleh teknologi multimedia menurut Mohammad Dastbaz dalam bukunya *Designing Interactive Multimedia System* (2003, pp9-16) :

1. Pendidikan

Sudah tidak diragukan lagi bahwa pendidikan adalah salah satu bidang yang amat dipengaruhi oleh multimedia. Selama beberapa dekade pengembangan *Computer Aided Learning* (CAL) dengan bidang pendidikan dihambat oleh terbatasnya objek yang dipelajari karena adanya batasan dari *text-based system*. Perkembangan multimedia dan pengintegrasian suara, video, dan animasi memberikan sebuah media baru sehingga para *CAL designer* dapat menciptakan sebuah lingkungan baru yang lebih luas dalam pembuatannya. Hal ini juga memberikan tingkat interaksi yang lebih besar pada CAL. Hutchings (1992) berkata bahwa CAL yang digabung dengan multimedia bukan

hanya meliputi semua pembelajaran yang diberikan oleh CAL yang lama, tapi juga memberikan kontrol yang lebih besar pada pembelajar dalam hal akses ke materi pembelajaran dan memberikan interaksi yang lebih dengan materi pembelajaran. Seiring berkembangnya teknologi *world wide web* dan *web based multimedia*, maka berkembang pula *E-Learning*. Beberapa institusi pendidikan menawarkan kuliah melalui *web*. Dan dengan perkembangan teknologi *web* yang semakin maju, maka *E-Learning* akan semakin berkembang dimasa depan.

2. Pelatihan

Sebuah penelitian oleh departemen pertahanan Amerika Serikat menyatakan bahwa pelatihan menggunakan sistem multimedia memberikan peningkatan sebesar 40% dibandingkan pelatihan biasa, dengan tingkat ingatan 30% lebih besar dan waktu pembelajaran yang 30% lebih sedikit. Pelatihan dengan multimedia sistem menunjukkan peningkatan ingatan, pengurangan biaya dan waktu. Sekarang banyak perusahaan besar yang menggunakan struktur jaringan atau *intranets* (*internal internets*) mendesain dan membuat paket-paket latihan untuk staf mereka (dengan pendekatan *E-Learning*). *E-Training* memiliki keuntungan yaitu dengan membiarkan karyawan untuk memilih materi pelatihan sesuai dengan kehendak mereka. Lebih lanjut, pelatihan multimedia bisa menggunakan video, audio, dan animasi untuk memperkaya ruang lingkup pembelajaran. Pelatihan menggunakan

alat-alat rumit dapat disimulasikan dan *user* dapat mudah menguasai penggunaan alat hanya dengan simulasi.

3. Informasi penjualan

Dalam beberapa area, kita dapat menemukan kios multimedia yang dilengkapi dengan *hardware* yang mengintegrasikan video, audio, dan grafik dengan sebuah *touch screen* yang biasa digunakan oleh pendatang yang menginginkan informasi. Di Amerika Serikat, informasi mengenai pelayanan negara ditampilkan dalam sebuah multimedia sistem dengan menggunakan *touch screen* yang interaktif. Contoh lain dari kios multimedia adalah kios informasi yang terdapat pada museum-museum, tempat wisata, bandara, atau bahkan pusat perbelanjaan.

4. Penyampaian berita, penyiaran dan periklanan

Penyiaran dan periklanan adalah salah satu bidang multimedia interaktif. Pada awal 1992, Liebman menjelaskan bahwa peningkatan untuk agensi penyiaran dan periklanan mulai menuju pada media interaktif. Peluncuran televisi interaktif dan bahkan peningkatan kemampuan dari *web TV* dan *web casting* untuk menyiarkan informasi telah menjadi salah satu peningkatan besar pada bidang penyiaran. Sekarang, jika kita *browsing* menggunakan internet, kita dapat menemukan ribuan koran dalam ratusan bahasa. Multimedia interaktif juga dapat memperkayanya dengan menambahkan laporan langsung dan video klip, dan menawarkan *user* pada sebuah aplikasi pencarian

agar *user* dapat lebih mudah mencari berita yang diinginkan. Bahkan kantor-kantor berita sudah mulai mengeluarkan biaya untuk membiayai pembuatan *website* pemberi informasi yang bisa menampilkan informasi kapan saja.

5. Aplikasi bisnis dan komersial

Dengan digunakannya aplikasi multimedia interaktif, pasar dunia berubah dan memanfaatkan teknologi yang ada untuk menawarkan bisnis. Bahkan industri perbankan yang merupakan bidang bisnis paling konservatif sudah mulai menggunakan teknologi multimedia sebagai suatu alat yang potensial untuk mencari pasar baru. Imbas utama dari multimedia adalah pecahnya ikatan ruang dan waktu dari pasar. Setiap perusahaan dan pembeli bisa kapan saja bertemu dan berkomunikasi satu sama lain. Teknologi merubah paradigma bisnis dan pemasaran. Paradigma *one to many* dimana perusahaan menawarkan produk pada banyak konsumen berubah menjadi sebuah model paradigma *many to many* dengan memberi kebebasan pada konsumen untuk memilih dan berkomunikasi.

2.1.4 Siklus Hidup Pengembangan Sistem *Interactive Multimedia System Of Design and Development (IMSDD)*

Sistem multimedia interaktif membutuhkan perencanaan yang sangat hati-hati pada struktur navigasi dan pendekatan dalam pembuatan

interaktifitas menggunakan metafora desain yang tepat. Tingkatan dari IMSDD (Dastbaz, 2003, pp130-132) adalah :

1. Analisis Kebutuhan Sistem

Tahap ini sebanding dengan tahapan spesifikasi kebutuhan pada model *waterfall* dan mengandung elemen-elemen *feasibility* (kelayakan) dan pemilihan *hardware* seperti pada RMM model. Beberapa fungsi tahapan ini adalah :

- a. Untuk menyediakan definisi sistem seperti pembuatan *outline* mengenai tujuan dan sasaran dari sistem yang akan dibuat.
- b. Untuk memastikan siapakah *user* dari sistem yang akan dibuat dan jika ada kebutuhan spesifik lain yang perlu dipertimbangkan. Sebagai contoh jika kita merancang sebuah paket pembelajaran untuk mengajarkan kepada *user* yang sulit mendengar maka menggunakan audio sebagai media pemberi informasi adalah sangat tidak cocok.
- c. Evaluasi *hardware*, *software*, dan *authoring tools* yang dibutuhkan lalu pilih secara tepat.
- d. Pertimbangkan secara tepat *delivery platform* yang dibutuhkan oleh sistem. Jika sistem multimedia interaktif berjalan pada sebuah jaringan (WAN, LAN) maka kita membutuhkan pendekatan yang berbeda dalam mendesain dan membangun sistem dibandingkan jika kita menggunakan sistem yang bertipe CD-ROM.

2. Pertimbangan Desain

Tujuan dari langkah ini adalah untuk menggambarkan secara jelas panduan tentang detail desain. Langkah ini mencakup:

a. Metafora Desain

Memilih sebuah model nyata untuk digunakan sebagai solusi kunci desain *interface* bagi sistem contohnya film, buku, *game*, dll.

b. Format dan tipe informasi

Definisikan tipe-tipe informasi yang perlu diintegrasikan ke dalam sistem (teks, grafik, animasi, suara, video). Sebagai contoh jika kita membuat sistem multimedia interaktif tentang film atau bioskop maka format informasi video akan berperan paling banyak.

c. Struktur navigasi

Menerangkan sebuah strategi navigasi yang jelas termasuk struktur hubungan dan fitur-fitur yang mungkin akan menghindari masalah yang berhubungan dengan sistem *hypermedia* seperti disorientasi.

d. Sistem kontrol

Menjelaskan tipe dan fitur-fitur dari kontrol dan peralatan yang dibutuhkan oleh sistem.

3. Implementasi

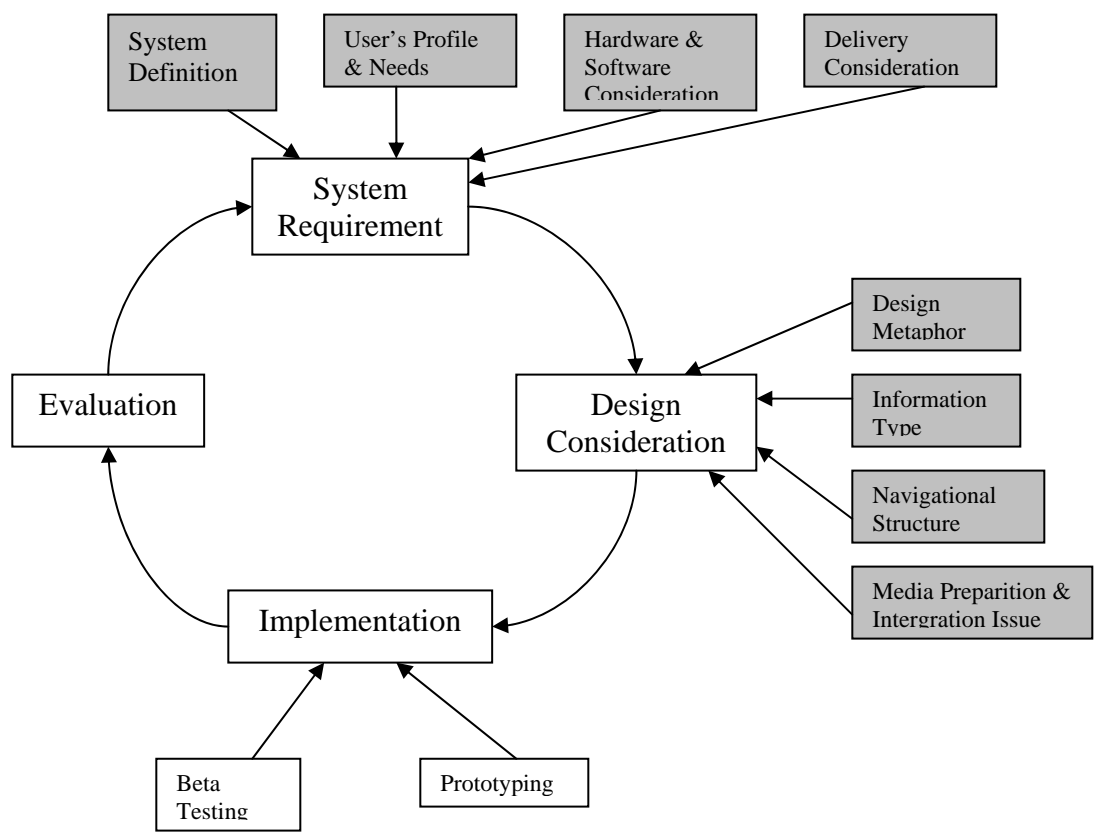
Saat fitur desain telah ditentukan maka tahap implementasi dari sistem dimulai menggunakan multimedia *authoring tools*. Tahap implementasi terdiri atas :

a. Membuat prototipe dari sistem.

- b. Melakukan beta test terhadap prototipe untuk mencari masalah yang mungkin dari kontrol atau desain.

4. Evaluasi

Pada tahapan ini sistem dievaluasi terhadap tujuan sebelumnya. Tipe-tipe evaluasi yang digunakan bisa berupa evaluasi formatif atau sumatif.



Gambar 2.1 Siklus IMSDD

2.2 Kios Informasi

Suatu kios informasi memberikan informasi dalam bentuk peta, pamflet, dan litelatur yang lain, yang ditawarkan oleh seorang penjaga.

(http://en.wikipedia.org/wiki/Information_kiosks.htm).

Dalam teknologi informasi sebuah kios adalah sebuah bangunan kecil (biasanya berisi sebuah komputer dan sebuah layar) yang menampilkan informasi bagi para pengunjung. Kios biasanya terletak dekat dengan pintu masuk dari pusat perbelanjaan yang berfungsi untuk menunjukkan arah bagi para pengunjung. (http://whatis.techtarget.com/definition/0,289893,sid9_gci212445,00.html)

2.3 Rekayasa Piranti Lunak

2.3.1 Definisi Piranti Lunak

Menurut Roger S. Pressman (2001, p10), piranti lunak adalah (1) perintah (program komputer) yang bila dieksekusi memberikan fungsi dan unjuk kerja seperti yang diinginkan. (2) struktur data yang memungkinkan program memanipulasi informasi secara proporsional, dan (3) dokumen yang menggambarkan operasi dan kegunaan program.

2.3.2 Karakteristik Piranti Lunak

Untuk lebih mendalami arti dari piranti lunak, ada baiknya kita mengamati karakteristik dari piranti lunak itu sendiri dan apa saja yang membedakannya dari kreasi manusia lainnya. Piranti lunak lebih bersifat logikal dibanding fisikal. Menurut Roger S. Pressman (2001, p10) karakteristik dari piranti lunak adalah :

1. Piranti lunak dibuat atau disusun tidak dibangun (dalam arti klasik)

Pada *hardware*, kualitas dari suatu produk diukur dari keunggulan desainnya. Berbeda dengan piranti lunak, kualitas piranti lunak yang memiliki pendekatan berbeda mengenai kualitas. Piranti lunak lebih berkonsentrasi pada proses penyusunan, bukan pada proses perakitan.

2. Piranti lunak tidak akan habis dipakai

Tingkat kegagalan piranti lunak akan meningkat jika ditemukan kesalahan. Tetapi saat kesalahan itu dikoreksi, maka kualitas piranti akan meningkat kembali. Saat kejadian ini berulang terus, maka semakin lama piranti lunak mulai tergantikan dengan yang lain. Pada *hardware*, jika komponen rusak maka dapat diganti. Sedang pada piranti lunak, tidak ada istilah komponen. Pada dasarnya, perawatan *hardware* lebih mudah dibandingkan dengan perawatan piranti lunak.

3. Piranti lunak dibuat menurut pesanan

Komponen piranti lunak dibuat agar bisa digunakan kembali. Banyak piranti lunak sekarang yang berasal dari komponen piranti lunak dahulu. Kita dapat menambah komponen pada piranti lunak yang sudah ada dan merubahnya menjadi piranti lunak yang baru dan sesuai dengan pesanan.

2.3.3 Aplikasi Piranti Lunak

Piranti lunak biasa diimplementasikan berdasarkan determinasi dan isi dari informasi. Adapun pembagian aplikasi piranti lunak adalah sebagai berikut (Roger S. Pressman (2001, pp16-18)) :

1. *System Software*

System software adalah kumpulan program yang dibuat untuk melayani program lain. Beberapa *system software* memproses struktur informasi yang kompleks, tetapi telah ditentukan. Sedangkan sistem aplikasi lain memproses data besar yang tidak ditentukan. Pada kasus lain area *system software* dikarakteristikan oleh interaksi dengan *hardware* komputer penggunaan oleh *multiple user* operasi yang bersamaan, membutuhkan penjadwalan dan pembagian sumber daya proses manajemen proses yang rumit struktur data yang kompleks dan *multiple interface* eksternal.

2. *Real-time Software*

Software yang memonitor, menganalisa, mengontrol *real world events* disebut dengan *real time*. Elemen dari *real time software* berupa sebuah komponen pengumpulan data yang mengumpulkan dan memformat informasi dari lingkungan luar, sebuah komponen analisis yang merubah informasi seperti apa yang dibutuhkan aplikasi, sebuah komponen kontrol atau output yang merespon lingkungan luar, sebuah komponen pengamatan yang mengkoordinasi semua komponen sehingga *real time* respon bisa diatur.

3. *Business Software*

Proses informasi bisnis adalah area terbesar dari sebuah area aplikasi *software*. Sistem bisnis telah berubah menjadi sebuah *software* manajemen sistem informasi yang bisa mengakses lebih dari satu database besar yang mengandung informasi bisnis.

4. *Engineering and Scientific Software*

Engineering and Scientific Software dikarakteristikan oleh algoritma-algoritma yang banyak memproses angka. Penggunaan aplikasi dari *Engineering and Scientific Software* sangat luas tetapi aplikasi dengan area *Engineering and Scientific* mulai bergeser dari algoritma *numeric* ke *real time simulation*.

5. *Embedded Software*

Software yang diletakan pada sebuah produk sehingga produk tersebut menjadi lebih pintar. *Embedded software* terdapat pada sebuah *read-only memory* dan digunakan untuk mengontrol produk bagi konsumen dan pasar industri.

6. *Personal Computer Software*

Dapat berupa *word processing*, grafik komputer, multimedia, *entertainment*, manajemen database, akses database dan masih banyak lainnya.

7. *Web-based Software*

Halaman *web* yang ditampilkan *browser* adalah *software* yang beroperasi dengan *executable instruction* dan data. Sekarang *network*

menjadi program komputer besar yang mendukung banyak sekali sumber *software* yang bisa diakses oleh siapa saja melalui *modem*.

8. *Artificial Intelligence Software*

AI *software* menggunakan algoritma *non numeric* untuk menyelesaikan masalah *complex* yang tidak sesuai dengan perhitungan atau analisis langsung, contohnya sistem pakar, pengenalan *image* atau suara, jaringan syaraf buatan, dan *game*.

2.3.4 Daur Hidup Pengembangan Piranti Lunak

Penulis menggunakan metode *The Classic Life Cycle* atau metode yang biasanya disebut *Waterfall Model* (Model Air Terjun) dalam perancangan aplikasi ini. Menurut Pressman (2001, pp37-38), tahapan-tahapan yang terdapat dalam metode ini adalah, yaitu:

- Analisis dan rekayasa sistem (*system engineering and modeling*)

Dalam tahap ini, pekerjaan dimulai dengan menetapkan kebutuhan-kebutuhan piranti lunak untuk semua elemen sistem, karena piranti lunak adalah bagian dari yang penting dari sebuah sistem. Peninjauan dalam tahap ini penting karena piranti lunak harus dapat berinteraksi dengan elemen-elemen lainnya seperti piranti keras, manusia, dan basis data.

- Analisis kebutuhan piranti lunak (*software requirements analysis*)

Dalam tahap ini, seorang analis piranti lunak harus menganalisis kebutuhan agar dapat mengetahui seluruh informasi, fungsi, performa,

dan antarmuka yang dibutuhkan agar dapat digunakan sebagai dasar sistem yang akan dibangun.

- Perancangan (*design*)

Dalam tahap ini, perancangan terhadap piranti lunak dilakukan. Perancangan piranti lunak merupakan sebuah proses yang terdiri atas beberapa langkah yang difokuskan pada empat perlengkapan program yang berbeda yaitu struktur data, arsitektur piranti lunak, representasi antar muka, dan prosedur yang detail.

- Pembuatan program (*coding*)

Dalam tahap ini, hasil perancangan diterjemahkan ke dalam bentuk yang dapat dimengerti oleh mesin/komputer.

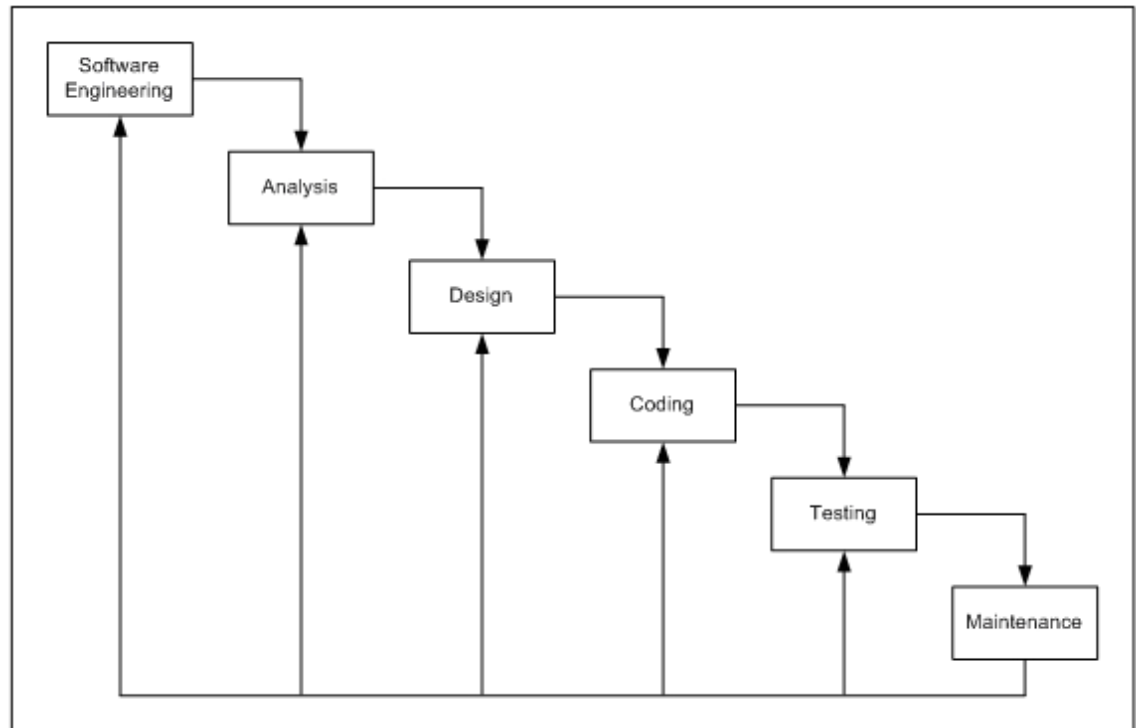
- Pengujian (*testing*)

Dalam tahap ini, pengujian terhadap program yang telah selesai dilakukan. Proses pengujian difokuskan pada logika dalam program dari piranti lunak untuk memastikan bahwa semua kode program telah diuji. Tujuan dilaksanakannya pengujian ini untuk menemukan kesalahan dan juga untuk memastikan bahwa piranti lunak yang telah dibuat telah sesuai dengan yang diinginkan.

- Pemeliharaan (*maintenance*)

Dalam tahap ini dilakukan pemeliharaan terhadap piranti lunak yang telah dibuat. Pemeliharaan dilakukan dengan mengecek apakah ada kesalahan-kesalahan setelah sistem dijalankan, dan setiap kesalahan harus diperbaiki. Selain itu juga dilakukan pengecekan apakah piranti

lunak ini dapat beradaptasi dengan segala perubahan yang ada, baik dari segi kebutuhan sistem maupun pengaruh dari luar sistem.



Gambar 2.2 Diagram *Classic Life Cycle of Software Engineering*

2.3.5 Definisi Rekayasa Piranti Lunak

Menurut Pressman (2001, p28) rekayasa piranti lunak adalah pengembangan dan penggunaan prinsip pengembangan suara untuk memperoleh perangkat lunak secara ekonomis yang reliabel dan bekerja secara efisien pada mesin nyata.

Rekayasa piranti lunak adalah sebuah teknologi yang terdiri dari *layer-layer* yang saling bersusun. Sebagai dasar, ada *quality focus*, lalu *process*, *methods*, dan terakhir *tools*. Manajemen kualitas harus dilakukan untuk meningkatkan kebiasaan dan saat kebiasaan meningkat, maka

pembuatan piranti lunak akan mencapai pendekatan yang lebih meningkat. *Layer proses* mendefinisikan sebuah *framework* bagi sebuah area kunci proses yang merupakan basis dari manajemen sebuah piranti lunak. *Layer method* menyediakan teknik dari pembuatan piranti lunak. Pada *layer* ini ditentukan tugas-tugas mana yang cocok dan sesuai bagi pembuatan *software*. *Layer tools* memperhitungkan *support* mana yang akan digunakan pada *process* dan *method*.

2.4 Interaksi Manusia dan Komputer

Dalam merancang suatu sistem yang interaktif, diperlukan perancangan model yang memiliki sifat *user friendly* agar dapat berinteraksi dengan baik dalam penggunaannya. Menurut Shneiderman (1998, p15) ada lima kriteria yang harus dipenuhi yaitu :

1. Waktu belajar

Berapa lama waktu yang diperlukan bagi *user* untuk dapat mempelajari serta bagaimana menggunakan perintah yang berhubungan dengan tugas.

2. Kecepatan dalam menyajikan informasi

Berapa lama waktu yang diperlukan sampai tugas diselesaikan

3. Tingkat kesalahan *user*

Berapa banyak dan apa jenis dari kesalahan yang dibuat *user* sewaktu melaksanakan tugas.

4. Daya ingat *user* setelah jangka waktu tertentu

Seberapa baik *user* memelihara pengetahuan masalah setelah satu jam, satu

hari, satu minggu.

5. Kepuasan subjektif

Seberapa banyak *user* menyukai penggunaan aspek sistem yang berbeda.

2.4.1 Antar Muka Pengguna

Semua yang ditampilkan di layar dibaca dalam dokumentasi atau manipulasi dengan *keyboard* atau *mouse* adalah bagian dari *user interface*. Sebagian besar program komputer yang dibuat oleh manusia memiliki *user interface* sebagai penghubung antara *user* program dengan program aplikasinya. Para pengembang program aplikasi harus memperhatikan dengan baik perancangan *user interface* aplikasinya.

Menurut Shneiderman (1998, pp74-75) ada delapan aturan emas dalam perancangan antarmuka pengguna (*user interface*) antara lain:

a. Berusaha untuk konsisten

Semua urutan aksi harus konsisten dalam situasi yang sama, seperti penggunaan istilah, warna, tampilan, dan jenis huruf yang sama.

b. Memungkinkan *user* menggunakan *shortcut*

Terkadang *user* menginginkan agar jumlah interaksi yang dilakukan dikurangi untuk meningkatkan kepraktisannya. Penggunaan *shortcut* dapat dilakukan untuk mengatasi hal tersebut dan meningkatkan kecepatan tampilan.

c. Memberikan umpan balik yang informatif

Respon harus berisi informasi kepada *user* sesuai dengan *action* yang dilakukannya. *User* harus mengetahui *action* apa yang telah dan akan dilakukan dengan respon balik tersebut.

d. Merancang dialog yang memberikan keadaan akhir

Urutan aksi harus diatur menjadi bagian awal, tengah, dan akhir. Hal ini dilakukan agar *user* mengetahui bahwa sekelompok tindakan telah dilakukan sehingga memberikan kepuasan pada *user* dan memberikan kesiapan untuk melakukan tindakan selanjutnya.

e. Memberikan pecegahan kesalahan dan penanganan kesalahan secara sederhana.

Sistem harus dirancang sedemikian rupa agar tidak membuat kesalahan yang serius. Jika terjadi kesalahan, sistem harus mendeteksi dan menawarkan mekanisme penanganan yang sederhana dan mudah dimengerti.

f. Memungkinkan memberikan aksi yang mudah

Aksi harus bisa dibalik untuk mengurangi kegelisahan dari *user* jika dia menjelajah ke bagian yang tidak dia kenal. Mendorong penjelajahan pilihan yang tidak biasa dipakai oleh *user*.

g. Mendukung pusat kendali internal.

User harus merasa bahwa dia menguasai sistem dan sistem bekerja sesuai dengan keinginannya. Kesulitan memperoleh informasi yang penting atau informasi yang diinginkan akan memberikan ketidakpuasan bagi *user*.

h. Mengurangi beban ingatan jangka pendek

Manusia memiliki ingatan yang terbatas oleh karena itu aplikasi harus dibuat sedemikian rupa agar *user* tidak terlalu banyak menyimpan *memory*.

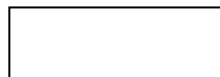
2.5 State Transition Diagram (STD)

Sebuah diagram *state* menggambarkan bagaimana sebuah kelas dari objek-objek menunjukkan reaksi di dalam sistem, dengan kata lain, bagaimana kelas bereaksi terhadap semua *use case* yang mempengaruhinya. (Britton, 2001, p107)

State Transition Diagram digunakan untuk menunjukkan suatu interval keadaan dari kelas tertentu, kejadian yang menyebabkan suatu perubahan dari suatu keadaan menjadi keadaan lain dan tindakan yang dihasilkan dari suatu perubahan keadaan. (Booch, 1994)

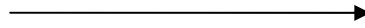
Sebagai suatu *modeling tool*, STD menggambarkan sifat ketergantungan waktu dari suatu sistem. STD digunakan untuk menggambarkan suatu sistem yang memiliki transaksi yang bersifat *real-time*. STD menggambarkan semua itu kedalam hubungan suatu *state* ke *state* lain. Notasi yang digunakan oleh STD sebagai berikut :

1. *State*, merupakan kumpulan keadaan yang memberikan perincian suatu objek pada waktu dan kondisi tertentu. State terbagi menjadi 3 bagian, yaitu : *Initial State* (*state* awal), *Normal State*, *Final State* (*state* akhir). Gambar *state* :



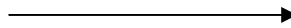
Gambar 2.3 Notasi State

2. *Transition state*, disebut juga perubahan *state*. Gambar *transition state* :



Gambar 2.4 Transition State

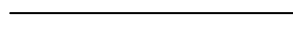
3. *Condition*, adalah suatu kejadian dari lingkungan luar yang dapat terdeteksi oleh sistem saat terjadi perubahan *state*.



Condition

Gambar 2.5 Notasi Condition

4. *Action*, adalah kegiatan yang dilakukan oleh sistem saat terjadi perubahan *state*. *Action* mempunyai keluaran seperti tampilan pesan, mencetak, dan hal lain yang berhubungan dengan *output*. Gambar *action* :



Action

Gambar 2.6 Notasi Action

2.6 Sistem Basis Data

Definisi sistem menurut O'Brien (2003, pp8-9) adalah kumpulan dari komponen-komponen yang saling berhubungan untuk bekerja sama mencapai tujuan bersama dengan menerima *input* yang menghasilkan *output* melalui proses transformasi yang terorganisir.

Menurut Date (1995, p4) *database* adalah kumpulan data yang telah diorganisasikan sehingga dapat diakses, diatur, dan diperbaharui dengan mudah. *Database* mengandung kumpulan catatan data dan atau *file*.

Tujuan dari sistem basis data secara keseluruhan adalah untuk melakukan perawatan informasi dan menyajikan kapan saja dibutuhkan. *Static Query Language* (SQL) adalah bentuk biasa untuk melakukan suatu perubahan dan atau untuk membuat perintah *query* yang interaktif dari sebuah aplikasi *database* seperti Microsoft Access, Oracle, atau Microsoft SQL Server.

Sistem basis data terdiri dari :

- a. *Field*, merupakan unit terkecil dalam suatu *record* yang membentuk suatu atribut dari *record* yang merupakan suatu data.
- b. *Record*, merupakan kumpulan dari *field-field* yang saling berkaitan.
- c. *File*, merupakan kumpulan dari *record-record* yang membentuk suatu kesatuan data yang sejenis.

2.6.1 Pemodelan Data

Pemodelan data adalah teknik yang digunakan untuk mengorganisasikan data dan dokumen. Biasanya disebut dengan *database modeling*. (Whitten, Bentley & Dittman, 2004, p294)

2.6.1.1 Entity Relationship Diagram

Terdapat beberapa alat atau *tools* yang dapat digunakan untuk membuat pemodelan data. *Entity Relationship Diagram* (ERD), merupakan pemodelan data yang mengutilisasi berbagai notasi untuk

menggambarkan data dari segi entitas dan relasi yang digambarkan oleh data tersebut. Berikut ini merupakan beberapa konsep dasar yang mendasari semua pemodelan data : (Whitten, Bentley & Dittman, 2004, p295-307)

1. *Entities*, adalah suatu *class* dari orang, tempat, objek, kejadian atau konsep tentang data yang dibutuhkan dan disimpan.
2. *Attributes*, merupakan *descriptive property* atau karakteristik dari suatu entitas. *Attributes* kadang-kadang disebut juga dengan *element*, *property*, dan *field*.
3. *Compound Attribute*, adalah suatu atribut yang terdiri dari beberapa atribut lainnya. Biasanya dikenal juga sebagai *concatenated attribute*, *composite attribute*, dan *data structure*.
4. *Data Type*, adalah properti dari suatu atribut yang mengidentifikasi tipe data apa saja yang bisa disimpan dalam atribut tersebut.
5. *Domain*, merupakan properti dari suatu atribut yang mendefinisikan nilai yang memungkinkan untuk suatu atribut.
6. *Default Value*, yaitu nilai yang akan dimasukkan ke *record* jika nilai tersebut tidak dispesifikasikan oleh *user*.
7. *Key*, merupakan suatu atribut atau kumpulan atribut yang memberikan nilai yang unik untuk setiap *instance* dari entitas. *Key* disebut juga dengan *identifiers*.
8. *Relationships*, adalah hubungan antara satu atau lebih entitas.

9. *Cardinality*, merupakan jumlah maksimum dan minimum dari munculnya suatu entitas yang mungkin berelasi dengan entitas lainnya.
10. *Degree*, adalah jumlah entitas yang berpartisipasi dalam suatu relasi.
11. *Recursive Relationship*, adalah suatu relasi yang terdapat diantara *instances* dari entitas yang sama.
12. *Associative Entity*, merupakan entitas yang menurunkan *primary key* nya dari satu atau lebih entitas.
13. *Nonidentifying Relationship*, adalah relasi dimana setiap entitas yang berpartisipasi memiliki *primary key* tersendiri.
14. *Identifying Relationship*, adalah suatu relasi dimana *key* dari entitas *parent* juga merupakan bagian dari *primary key* entitas *child* nya.
15. *Nonspecific Relationship*, adalah suatu relasi dimana banyak *instances* dari suatu entitas berasosiasi dengan banyak *instances* dari entitas lain. Disebut juga dengan relasi *many to many*.
16. *Generalization*, merupakan konsep dimana atribut yang biasa untuk beberapa tipe entitas dikelompokkan kedalam kelompok entitas tersendiri.
17. *Supertype*, adalah suatu entitas yang menyimpan atribut yang biasa untuk satu atau lebih entitas.
18. *Subtype*, adalah suatu entitas yang bisa menurunkan atribut dari entitas *supertype*.