

BAB II

TINJAUAN PUSTAKA

2.1 Definisi Multimedia

Multimedia berasal dari kata 'multi' dan 'media'. Multi berarti banyak, dan media berarti tempat, sarana atau alat yang digunakan untuk menyimpan informasi. Jadi berdasarkan kata, 'multimedia' dapat diasumsikan sebagai wadah atau penyatuan beberapa media yang kemudian didefinisikan sebagai elemen-elemen pembentukan multimedia. Elemen-elemen tersebut berupa : teks, gambar, suara, animasi, dan video. Multimedia merupakan suatu konsep dan teknologi baru bidang teknologi informasi, dimana informasi dalam bentuk teks, gambar, suara, animasi, dan video disatukan dalam komputer untuk disimpan, diproses, dan disajikan baik secara linier maupun interaktif[3].

Penyajian dengan menggabungkan seluruh elemen multimedia tersebut menjadikan informasi dalam bentuk multimedia yang dapat diterima oleh indera penglihatan dan pendengaran, lebih mendekati bentuk aslinya dalam dunia sebenarnya.

Multimedia interaktif adalah bila suatu aplikasi terdapat seluruh elemen multimedia yang ada dan pemakai (user) diberi kebebasan / kemampuan untuk mengontrol dan menghidupkan elemen-elemen tersebut.

2.2 Elemen – elemen Multimedia

1. Objek-objek

1. Media Diskrit : elemen tunggal
 - a. *Icon* : gambar semantik (seperti simbol STOP). Pemakai harus terlebih dahulu mempunyai pengetahuan mengenai icon.
 - b. Grafik : menjadi tujuan.
 - c. Citra : yang dihasilkan dari komputer, bisa berupa grafik 2D/3D tergantung sumbernya (seperti foto).
 - d. Teks : ukuran, tipe huruf, warna.
2. Media Kontinu : elemen tunggal yang disusun berdasarkan waktu
 - a. Gambar bergerak (audio + video).

2. Layout Spasial

- a. Absolut

Koordinat relatif dengan aslinya / umumnya (pojok kiri atas), contohnya seperti aplikasi Windows.
- b. Relasi berarah

Menentukan susunan dalam ruang, contohnya seperti peta *subway* (petunjuk arah).
- c. Relasi topologi

Posisi elemen terhadap elemen lain, contohnya seperti *contains, inside of, equals, cover, overlap, disjoint, covered by*.
- d. Alur teks

Alur berdimensi satu, ditunjukkan dengan area berdimensi dua.

3. Dimensi Temporer

1. Model temporer :
 - a. Terbatas : misalnya 6 detik.
 - b. Tidak terbatas : misalnya pengguna mengklik *button*.
 - c. Relasi paralel dan sekuensial, misalnya 2 video dimulai bersamaan atau 1 video dimulai setelah yang pertama selesai.
2. Animasi
 - a. Gabungan dimensi temporer dan layout spasial (posisi suatu objek berubah sesuai dengan waktunya).

4. Interaksi Pengguna

1. Level Interaksi Pengguna :
 - a. Pasif : hanya visualisasi.
 - b. Reaktif : interaksi terbatas, contohnya seperti *Scroll panel*.
 - c. Proaktif : memilih jalur atau penyeleksian, contohnya seperti *button*.
 - d. Reciprocal : berhubungan dengan informasi pembuatan pada pengguna.
2. Model Interaksi :
 - a. Navigasi : memilih jalur yang diinginkan.
 - b. Perancangan : pengguna memodifikasi gaya visual dari presentasi, contohnya seperti warna, *volume* audio.
 - c. Bioskop : pengguna dapat mengontrol waktu keseluruhan (pada VCR, contohnya seperti *play, stop*)

5. Logika Aplikasi

1. Presentasi multimedia tradisional, tidak perlu logika :
 - a. Kunjungan virtual ke museum, menu DVD.
2. Sistem interaktif *real-time* :
 - a. Dunia *virtual reality*, permainan.
3. Logika aplikasi membutuhkan bahasa pemrograman (*if case, goto ...*)
 - a. Bahasa terkompilasi : C, C++.
 - b. *Virtual machine* : Java.
 - c. *World Wide Web*, MPEG-4, Director : *scripting*.
 - d. *Action Script*.

2.3 *Flash (Macromedia Flash)*

Flash adalah alat yang digunakan desainer dan pengembang untuk membuat presentasi, aplikasi, dan konten lainnya yang memungkinkan membuat interaksi dengan pengguna. Proyek *flash* dapat termasuk animasi sederhana, konten video, presentasi kompleks, aplikasi, dan segala sesuatu di antaranya. Secara umum, individu potongan konten yang dibuat dengan *Flash* disebut aplikasi, meskipun mereka mungkin hanya animasi dasar. Kita dapat membuat media yang kaya aplikasi *Flash* dengan memasukkan gambar, suara, video, dan efek khusus.

Flash sangat cocok untuk membuat konten untuk pengiriman melalui internet karena file yang sangat kecil. *Flash* dapat mencapai hal tersebut melalui penggunaan luas vektor grafis. Secara signifikan Vector grafis memerlukan sedikit memory dan ruang penyimpanan daripada bitmap grafis karena mereka diwakili

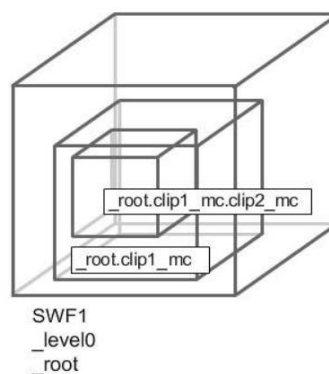
oleh rumus matematika dari kumpulan data besar. Grafik bitmap lebih besar karena masing-masing pixel dalam gambar memerlukan sepotong data terpisah untuk mewakilinya.

Untuk membangun aplikasi dalam *Flash*, kita dapat membuat gambar dengan alat gambar yang ada pada *Flash* dan memasukkan elemen media tambahan ke dalam dokumen *Flash*. Selanjutnya, kita menentukan bagaimana dan kapan kita ingin menggunakan masing-masing elemen untuk menciptakan aplikasi yang ada dalam pikiran kita.

2.3.1 Objek – objek dalam *Flash* 8

1. *Movie clip*

Movie clip dalam *Flash* pada dasarnya adalah sebuah kontainer, yang dapat diwakili secara visual sebagai sebuah kotak. Dengan *Flash* kita dapat meletakkan kotak ini ke dalam kotak lainnya. Gambaran semacam ini (ber sarang) disebut “*parent-child relationship*” atau hubungan orang tua dan anak.



Gambar 2.1 Representasi *Movie clip*

dalam diagram di atas, kita dapat melihat tiga *Movie clip* direpresentasikan, ada dua *Movie clip* yang bersarang di dalam *Movie clip* lain.

2. *Button*

Button adalah objek berupa tombol dan memiliki fungsi sebagai tombol, dalam *Flash button* sama seperti *movie clip* dapat menjadi kontainer, dan dapat menampung objek lain seperti *movie clip* dan *graphic*, perbedaannya adalah pada *time line* di *button* hanya memiliki 4 *frame* untuk kondisi *up*, *over*, *down*, *hit*.

3. *Graphic*

Graphic adalah sebuah objek yang hanya memiliki fungsi sebagai kontainer tetapi tidak dapat menampilkan animasi yang berjalan pada *movie clip* yang dibawa di dalamnya.

2.4 Interaksi manusia dan komputer

Interaksi Manusia dan komputer merupakan komunikasi dua arah antara pengguna (user) dengan sistem komputer yang saling mendukung untuk mencapai suatu tujuan tertentu.

2.4.1 Ruang Lingkup IMK

Adapun ruang lingkup IMK yaitu :

- a. Manusia dan komputer** berinteraksi lewat masukan & keluaran melalui antarmuka
- b. Fokus:** perancangan dan evaluasi antarmuka pemakai (*user interface*).
- c. Antarmuka pemakai** adalah bagian sistem komputer yang memungkinkan manusia berinteraksi dengan komputer.

2.5 Animasi

Animasi adalah sebuah proses merekam dan memainkan kembali serangkaian gambar statis untuk mendapatkan sebuah ilusi pergerakan. Berdasarkan arti harfiah, Animasi adalah menghidupkan. Yaitu usaha untuk menggerakkan sesuatu yang tidak bisa bergerak sendiri.

Secara garis besar, animasi computer dibagi menjadi dua kategori, yaitu:

- a. *Computer Assisted Animation*, animasi pada kategori ini biasanya menunjuk pada sistem animasi 2 dimensi, yaitu mengkomputerisasi proses animasi tradisional yang menggunakan gambaran tangan. Komputer digunakan untuk pewarnaan, penerapan virtual kamera dan penataan data yang digunakan dalam sebuah animasi.

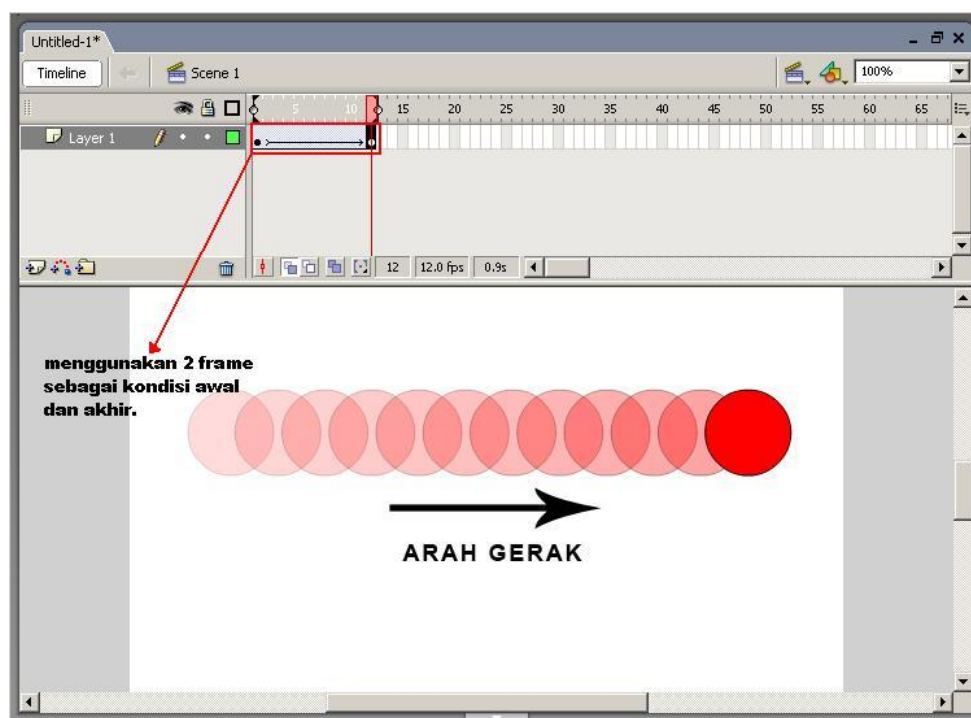
- b. *Computer Generated Animation*, pada kategori ini biasanya digunakan untuk animasi 3 dimensi dengan program 3D seperti 3D Studio Max, Maya, Autocad dan lain sebagainya.

2.5.1 Jenis Animasi Dalam Menggunakan *Macromedia Flash 8*

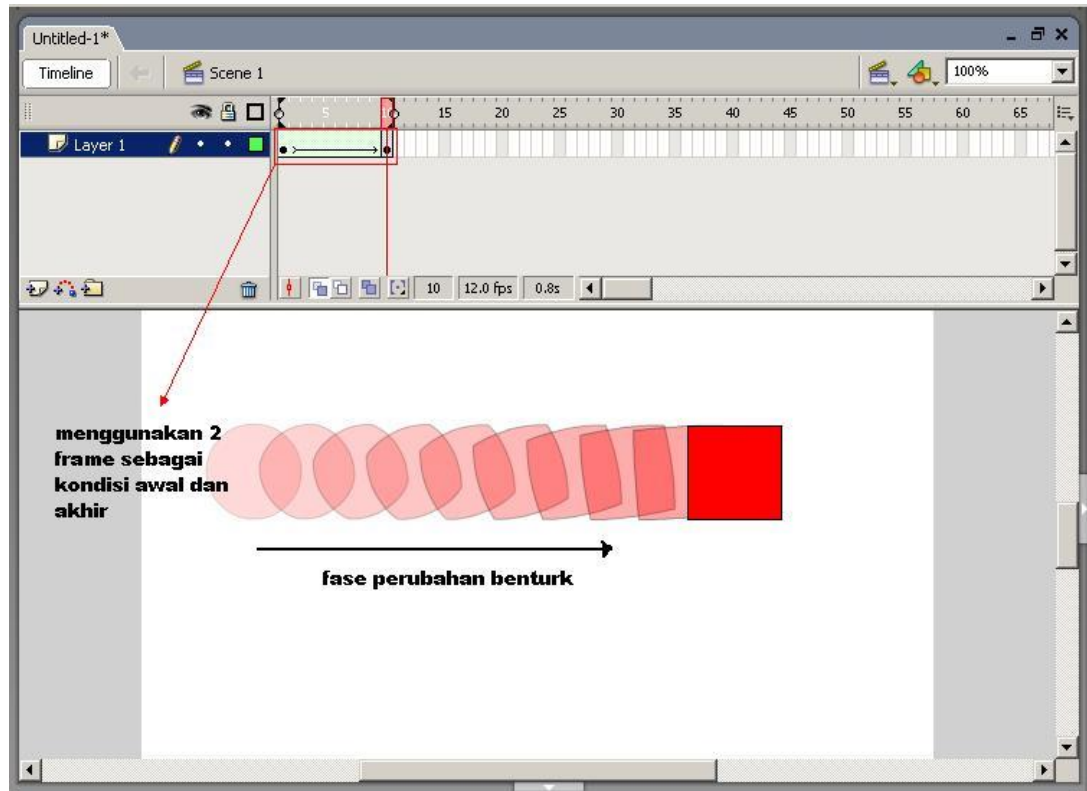
1. Animasi *Inbetween*

Animasi *inbetween* terbagi menjadi 2 yaitu *shape tween* dan *motion tween*, untuk *shape tween* adalah animasi perubahan bentuk maupun warna untuk suatu objek dengan tipe *shape*.

Sedangkan untuk *motion tween* adalah animasi perubahan posisi, rotasi dan skala dari suatu objek yang dapat berupa *movie clip*, *graphic*, ataupun *button*.



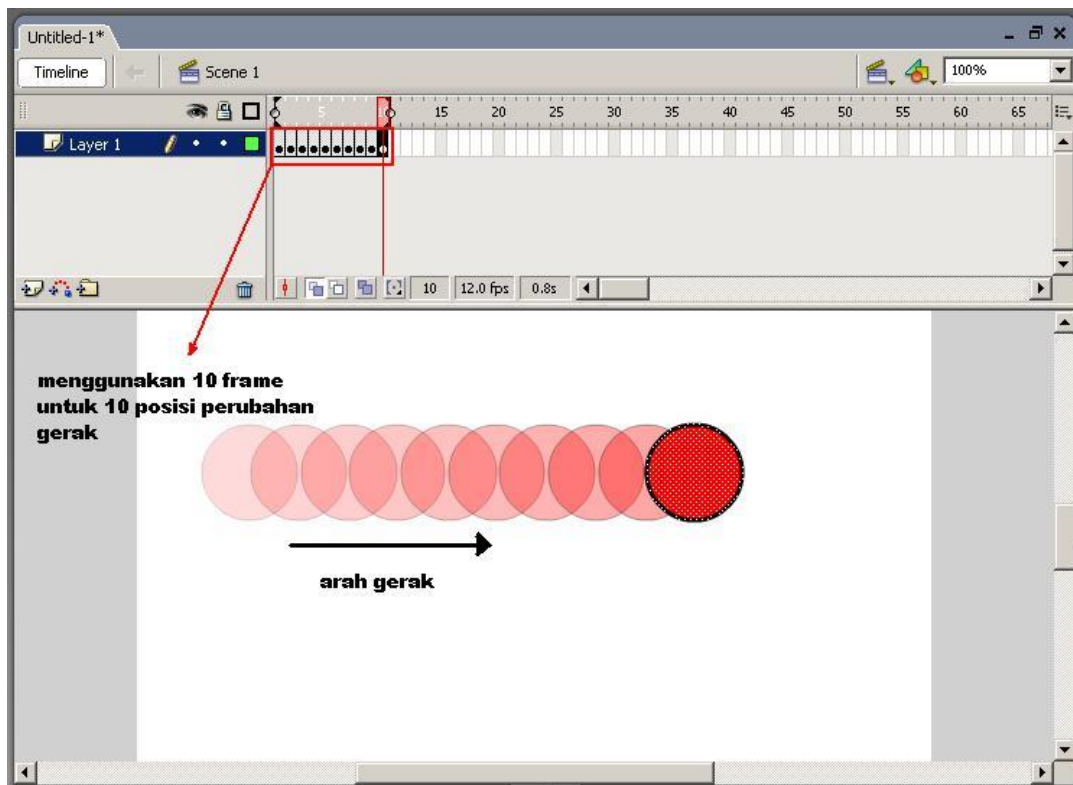
Gambar 2.2 Animasi *Motion tween*



Gambar 2.3 Animasi *Shape tween*

2. Animasi *Frame by frame*

Untuk animasi *Frame by frame* merupakan jenis animasi dasar yang prinsip kerjanya menampilkan beberapa gambar berbeda dalam satu waktu sehingga menimbulkan efek pergerakan.



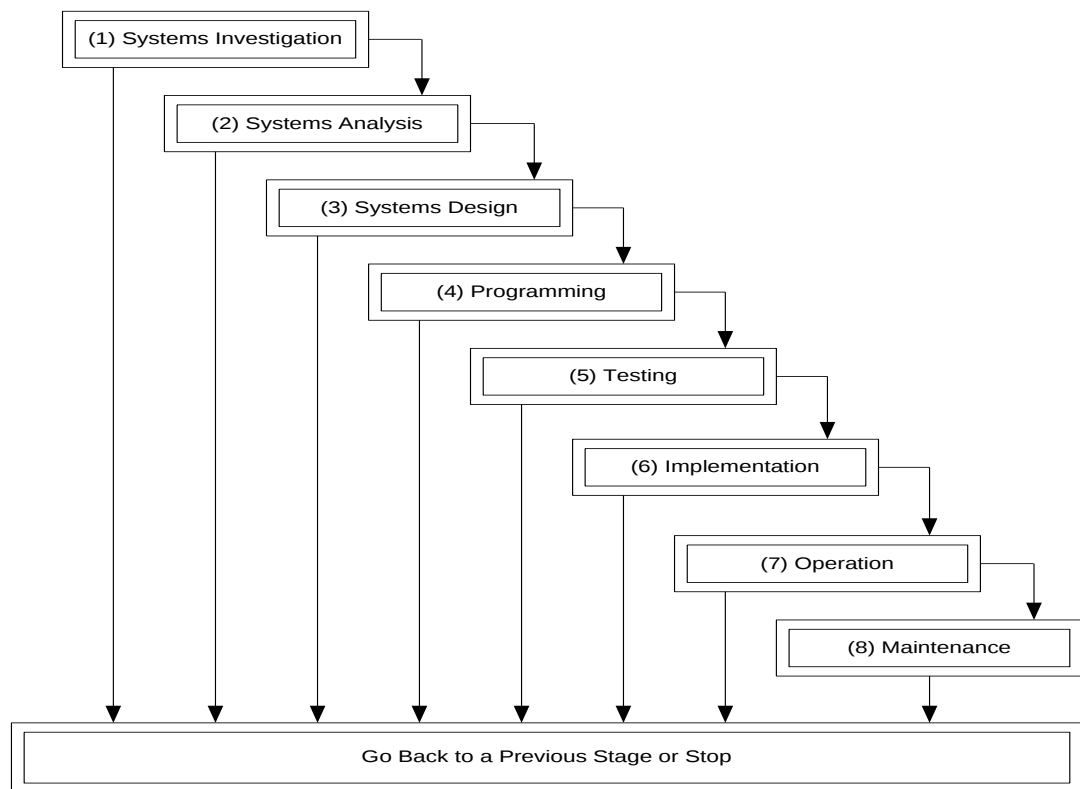
Gambar 2.4 Animasi *Frame by frame*

3. Animasi *Script*

Sedangkan untuk animasi skrip, untuk menggerakkan, mengatur dan memanipulasi suatu objek menggunakan perintah skrip, dalam *Macromedia Flash 8* digunakan *Action Script 2.0*.

2.6 Metode Perancangan

Systems development life cycle (SDLC) adalah metode pengembangan sistem tradisional yang digunakan oleh banyak organisasi sekarang ini. SDLC merupakan kerangka kerja terstruktur yang terdiri dari proses-proses sequensial yang membangun sistem informasi[4].



Gambar 2.5 Siklus Hidup Pengembangan Sistem (SDLC) 8 level

Pada metode pengembangan sistem SDLC, tugas-tugas pada satu tahapan diselesaikan sebelum pengembangan dilanjutkan ke tahapan berikutnya. Kini, apabila dibutuhkan, pengembang bisa bergerak maju mundur dari berbagai tahapan seperti ditunjukkan oleh gambar 2.5.

Tahapan-tahapan dalam SDLC adalah sebagai berikut

1. Investigasi sistem (*systems investigation*)

Pada tahap ini, dilakukan studi kelayakan untuk menentukan kemungkinan suksesnya proyek pengembangan sistem yang diajukan dan menaksir kelayakan proyek dari segi teknis, ekonomis, dan perilaku.

2. Analisis sistem (*systems analysis*)

Tahapan ini mendefinisikan masalah bisnis yang dihadapi, mengenali penyebabnya, menentukan solusinya, dan mengenali kebutuhan informasi. Untuk memahami sifat program yang dibangun, perancang harus memahami domain informasi, tingkah laku, unjuk kerja, dan *interface* yang diperlukan.

3. Perancangan sistem (*systems design*)

Hasil dari fase perancangan sistem adalah perancangan teknis yang menentukan hal-hal berikut:

- a. Output, input, dan antar muka (*interface*) sistem
- b. Perangkat keras, piranti lunak, database, telekomunikasi, personel, dan prosedur
- c. Bagaimana setiap komponen berintegrasi

Perancangan sistem mencakup pembuatan model logikal dan fisikal dari komponen-komponen data sistem.

4. Pemrograman (*programming*)

Spesifikasi desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca (bahasa pemrograman) untuk mengurangi kompleksitas dan meningkatkan kesinambungan (*modularity*). Pengecekan atas kesalahan (*error*) yang mungkin terjadi perlu dilakukan sebagai tindak lanjut.

5. Pengujian (*testing*)

Testing dilakukan untuk memastikan apakah kode komputer yang ditulis memberikan hasil aktual yang diinginkan di bawah kondisi – kondisi yang diberikan. Proses pengujian juga berfokus pada logika internal perangkat

lunak dan eksternal fungsional, yaitu untuk mendeteksi adanya *error* (“*bugs*”) dalam kode komputer.

6. Implementasi (*implementasi*)

Implementasi merupakan proses konversi (*conversion*) sistem lama menjadi sistem yang baru. Setelah sistem lama dibongkar dan sistem baru diinstal, perubahan terjadi dan fase operasi dimulai.

7. Operasi (*operation*)

Setelah konversi, sistem baru akan beroperasi dalam periode tertentu. Saat operasi sistem baru telah stabil, audit dilakukan guna memperkirakan kapabilitas sistem dan menentukan apakah penggunaannya tepat.

8. Pemeliharaan (*maintenance*)

Perangkat lunak akan mengalami perubahan setelah berada di tangan klien. Perubahan akan terjadi karena perangkat lunak harus disesuaikan untuk mengakomodasi perubahan dalam lingkungan eksternalnya atau karena pelanggan membutuhkan perkembangan fungsional.

2.7 State Transition Diagram (STD)

STD digunakan untuk membuat model perangkaian banyak sistem *realtime*. Sistem *realtime* adalah suatu kondisi untuk mengoperasikannya secara bersama-sama dengan waktu reaksi yang teratur atau sudah diprediksi dengan keadaan yang sebenarnya [5].

STD mempresentasikan perilaku suatu sistem dengan menggambarkan keadaan-keadaannya dan kejadian kejadian yang dapat membuat sistem berubah keadaan. Susunan STD dalam aplikasi memerlukan dua langkah[6].

Langkah 1 : Mengidentifikasi semua keadaan terbatas yang mungkin ada untuk sistem.

Langkah 2 : Mengidentifikasi aktifitas yang menyebabkan transisi dari satu keadaan ke keadaan yang lain.

Notasi State Transition Diagram

Notasi yang umum digunakan pada *State Transition Diagram* antara lain adalah :

a. Keadaan Sistem

State adalah kumpulan keadaan atau atribut yang mencirikan seseorang atau benda pada waktu tertentu, bentuk keadaan tertentu maupun kondisi tertentu.

State disimbolkan dengan empat persegi panjang atau segi empat.



Gambar 2.6 Keadaan Sistem

b. Perubahan Keadaan

Transisi *state* atau perubahan *state* disimbolkan dengan anak panah.



Gambar 2.7 Perubahan Keadaan

Untuk melengkapi STD dibutuhkan kondisi yang dimaksud dengan kondisi adalah suatu kejadian pada lingkungan luar yang dapat dideteksi oleh sistem, sedangkan aksi adalah yang dilakukan oleh sistem apabila terjadi perubahan

state atau merupakan reaksi terhadap kondisi. Aksi akan menghasilkan keluaran, tampilan pada layar, menghasilkan kalkulasi, dan lain sebagainya.

c. Spesifikasi Proses

Spesifikasi proses digunakan untuk mendeskripsikan proses yang terjadi pada tingkat paling dasar. Model ini berfungsi untuk mendeskripsikan apa yang dilakukan ketika masukan ditransformasi menjadi keluaran. Kalimat dalam spesifikasi proses umumnya tersusun dari sejumlah komposisi seperti rumusan matematis, kata kerja dan obyek, misalnya variabel atau elemen data. Terminologi dalam komputasi dideskripsikan dengan kata kerja seperti: cari (*find, search atau locate*), jumlahkan (*add*), tulis (*display atau write*) dan lain lain[7].

Adapun bentuk dari spesifikasi proses dibedakan menjadi 4 macam:

1. Bentuk naratif, merupakan bentuk yang paling sederhana dalam spesifikasi proses karena menggunakan kalimat-kalimat yang singkat tetapi harus jelas penggunaannya.
2. Algoritma singkat, merupakan pola pikir yang terstruktur berisi tahap-tahap penyelesaian suatu masalah, yang nantinya akan diimplementasikan kedalam suatu bahasa pemrograman.
3. Penekanan pada *user interface*, spesifikasi proses ini bentuknya sangat mudah dan sering digunakan oleh pemakai karena tampilannya yang mudah dimengerti dan jelas.

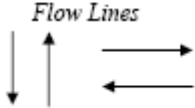
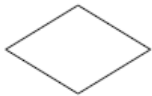
4. *Block chart*, spesifikasi proses bentuk ini berfungsi untuk memodelkan masukan, keluaran, proses maupun transaksi dengan menggunakan simbol – simbol tertentu.

2.8 Diagram Pemodelan Berorientasi Objek

Pengembangan berorientasi objek dalam proses konseptual terpisah dengan bahasa pemrograman sampai tahap akhir. Pengembangan berorientasi objek secara mendasar merupakan cara berpikir baru dan bukan suatu teknik pemrograman. Hal ini dapat berfungsi sebagai media spesifikasi, analisis, dokumentasi, dan *interface* seperti halnya pemrograman.

2.9 Flowchart

Bagan alir sistem merupakan alat yang tepat untuk menggambarkan physical system. Bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan aliran (*flow*) di dalam program atau prosedur system secara logika, digunakan terutama sebagai alat bantu komunikasi dan alat dokumentasi. Bagan alir sistem (*system flowchart*) merupakan bagan yang menunjukkan arus pekerjaan dari sistem secara keseluruhan, menjelaskan urutan dari prosedur-prosedur yang ada di dalam sistem serta menunjukkan apa yang dikerjakan di dalam sistem. Simbol-simbolnya[8]:

Simbol	Arti
<i>Flow Lines</i> 	Garis ini menunjukkan arah selanjutnya yang akan dituju
Terminal (mulai atau berhenti) 	Simbol ini digunakan untuk menunjukkan awal kegiatan atau akhir kegiatan atau berhenti suatu program.
<i>Input atau Output</i> 	Untuk mewakili data <i>input</i> untuk memuliskan <i>output</i> .
Proses (pengolahan) 	Suatu simbol yang melambangkan diprosesnya suatu alat.
<i>Predefined</i> 	Untuk program-program yang dipergunakan dalam sebuah program berulang kali, biasanya program dibuat terpisah dengan sebutan sub program (<i>subroutine</i>), untuk menghubungkan program utama dengan <i>subroutine</i> dipergunakan simbol ini.
<i>Decision</i> (keputusan) 	Menunjukkan suatu perbandingan yang harus dibuat bila hasilnya "ya", maka alir data akan menunjukkan ke suatu tempat, bila "tidak" maka akan menuju ke tempat lain.
<i>Connector</i> (penghubung) 	Simbol untuk keluar atau masuk <i>procedure</i> atau proses dalam lembar atau halaman yang sama.
<i>Off-line Connector</i> 	Simbol untuk keluar atau masuk <i>procedure</i> atau proses dalam lembar atau halaman yang sama.

Gambar 2.8 Flowchart