

BAB II

TINJAUAN PUSTAKA

2.1 Definisi Simulasi

Definisi beberapa ahli tentang simulasi :

- Menurut Banks, simulasi adalah tiruan dari proses dunia nyata atau sistem. Simulasi menyangkut pembangkitan proses serta pengamatan dari proses untuk menarik kesimpulan dari sistem yang diwakili.
- Menurut Nailor dalam Rubinstein & McIamed, simulasi adalah teknik numerik untuk melakukan eksperimen pada komputer, yang melibatkan jenis matematika dan model tertentu yang menjelaskan bisnis atau ekonomi pada suatu periode waktu tertentu.
- Menurut Borowski & Borwein, simulasi adalah teknik untuk membuat konstruksi model matematika untuk suatu proses atau situasi dalam rangka menduga secara karakteristik atau menyelesaikan masalah berkaitan dengannya dengan menggunakan model yang diajukan.

Jadi, simulasi mempelajari atau memprediksi sesuatu yang belum terjadi dengan cara membuat model sistem yang dipelajari dan selanjutnya mengadakan eksperimen secara numerik dengan menggunakan komputer.¹ Simulasi yang saya rancang adalah tentang simulasi monitoring ruangan di.

2.2 Definisi Monitoring

Monitor adalah suatu tipe data abstrak yang dapat mengatur aktivitas serta penggunaan resource oleh beberapa thread. Ide monitor pertama kali diperkenalkan oleh C.A.R Hoare dan Per Brinch-Hansen pada awal 1970-an.

Monitor terdiri atas data-data private dengan fungsi-fungsi public yang dapat mengakses data-data tersebut. Method-method dalam suatu monitor sudah dirancang sedemikian rupa agar hanya ada satu buah method yang dapat bekerja pada suatu saat. Hal ini bertujuan untuk menjaga agar semua operasi dalam monitor bersifat mutual exclusion.

Monitoring / pengawasan dapat dilakukan dari jauh maupun dari dekat. Pengawasan dari jauh disebut pemantauwan dan pemantauwan ini dapat di lakukan menggunakan sarana komputer, telepon, fax, atau radio. Wujud pengawasan cara ini adalah permintaan laporan kepada bawahan dan jawaban dari bawahan atas permintaan tersebut. Jika monitoring dari jauh tidak efektif dapat dilakukan monitoring langsung ke objeknya. Dalam hal ini pengawasan yang dilakukan tersebut sebagai pemeriksaan yang berarti pemeriksa berhadapan langsung dengan objek yang di perlukan.yang menjadi acuan kegiatan pengawasan adalah rencana, program kerja.

2.2.1 Fungsi Monitoring

- **Eksplanasi**, pengawasan menghimpun informasi yang dapat menjelaskan mengapa hasil-hasil kebijakan publik dan program yang di canangkan berbeda.
- **Akuntansi**, pengawasan menghasilkan informasi yang bermanfaat untuk melakukan akuntansi atas perubahan sosial ekonomi yang setelah dilaksanakan sejumlah kebijakan publik dari waktu ke waktu
- **Pemeriksaan**, pengawasan membantu menentukan apakah sumberdaya dan pelayanan yang di maksudkan untuk kelompok sasaran maupun konsumen tertentu memang telah sampai kepada mereka.

2.3 Interaksi Manusia dan Komputer

2.3.1 Pengertian Interaksi Manusia dan Komputer

Istilah interaksi manusia dan komputer muncul pada pertengahan tahun 1980-an sebagai suatu istilah yang mengacu pada suatu bidang ilmu baru dalam dunia ilmu pengetahuan, khususnya komputer. Istilah interaksi manusia dan komputer mempunyai cakupan yang lebih luas dari pada hanya sekedar merancang *interface*.

Walaupun belum ada definisi yang pasti tentang interaksi manusia dan komputer, tapi ada suatu definisi menurut *Rogers, Sharp, Benyon* yang menyatakan bahwa interaksi manusia dan komputer adalah suatu

disiplin ilmu yang mencakup masalah perancangan, evaluasi, serta implementasi dari suatu sistem komputer yang interaktif yang digunakan oleh manusia.

2.3.2 Tujuan Interaksi Manusia dan Komputer

Menurut *Galitz*, tujuan dari ilmu interaksi manusia dan komputer ini adalah untuk membangun atau merancang suatu sistem yang aman, sesuai dengan kebutuhan, efektif, efisien, dan tentu saja dapat digunakan. Konsep utama dari interaksi manusia dan komputer ini sebenarnya adalah bagaimana membuat sistem yang mudah dipelajari dan mudah digunakan.

2.3.3 Perancangan Antarmuka

Menurut *Shneiderman* adalah delapan aturan yang harus diperhatikan dalam merancang user interface yang baik (eight golden rules)

1. Mempertahankan konsistensi

Mempertahankan konsistensi, misalnya dalam menetapkan posisi tampilan, penggunaan kalimat untuk pesan kesalahan, dan sebagainya.

2. Penggunaan *shortcut*

Memudahkan pengguna dengan memakai *shortcut* (tombol khusus) untuk melakukan suatu perintah.

3. Umpan balik yang informatif

Adanya umpan balik yang informatif dari setiap aksi yang dilakukan, seperti layar pesan atau tanda-tanda khusus.

4. Rancangan dialog

Urutan aksi harus diatur dalam grup dimana ada awal, tengah dan akhir. Dengan adanya umpan balik dapat memberikan pilihan untuk menyiapkan grup aksi berikutnya.

5. *Error handling*

Sistem harus dirancang agar pengguna tidak membuat kesalahan yang fatal. Apabila pengguna melakukan kesalahan, sistem harus dapat mendeteksi kesalahan dan memberikan instruksi yang sederhana untuk melakukan perbaikan.

6. Aksi

Mengijinkan pengguna untuk memperbaiki suatu aksi dengan mudah.

7. *Internal locus of control*

Menyediakan kendali internal bagi pengguna, agar mereka dapat menggunakan sistem tersebut.

8. Beban ingatan

Mengurangi beban ingatan jangka pendek agar pengguna tidak disulitkan oleh tuntutan penghafalan.

2.3.4 Faktor Manusia

Menurut *Pressman* dalam merancang sistem yang interaktif harus memperhatikan sistem manusia dan teknologi tentang sistem interaktif.

Faktor manusia mencakup :

1. Pengamatan manusia

Hasil rancangan suatu sistem interaktif dapat menimbulkan dua pendapat untuk dua individu yang berbeda seperti, untuk individu yang pertama menganggap bahwa sistem interaktif tersebut “*user friendly*”, sedangkan individu yang lain tidak menganggap seperti individu pertama.

2. Tugas dan faktor manusia

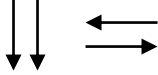
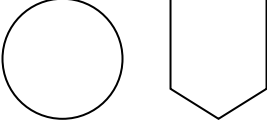
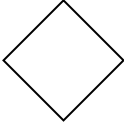
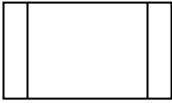
Interaksi manusia dan komputer harus melengkapi pemakai dengan kemudahan serta lingkungan yang alami dalam menyelesaikan tugasnya. Tampilan layar pun berpengaruh dalam merancang sistem interaktif agar terlihat menarik dan mudah dimengerti.

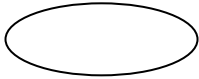
2.4 Alat Bantu yang Digunakan

2.4.1 Bagan Alir Program

Bagan alir program (program *flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program. Bagan alir program dibuat dengan menggunakan simbol-simbol seperti yang ditunjukkan pada tabel dibawah ini :

Tabel 2.1 Simbol Bagan Alir

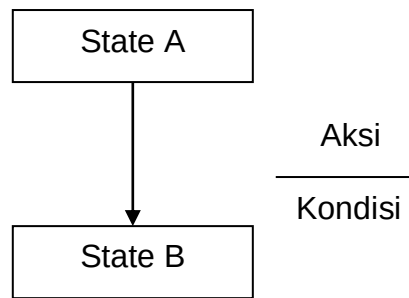
No.	Gambar	Nama	Keterangan
1.		Simbol input/output	Digunakan untuk mewakili data
2.		Simbol proses	Digunakan untuk mewakili suatu proses
3.		Simbol garis alir	Digunakan untuk menunjukkan arus dari proses
4.		Simbol penghubung	Digunakan untuk menunjukkan sambungan dari bagan alir yang terputus di halaman yang masih sama atau halaman lainnya
5.		Simbol keputusan	Digunakan untuk suatu penyeleksian kondisi didalam program
6.		Simbol proses terdefinisi	Digunakan untuk menunjukkan suatu operasi yang rinciannya ditunjukkan ditempat

			lain
7.		Simbol persiapan	Digunakan untuk memberi nilai awal suatu besaran
8.		Simbol terminal	Digunakan untuk menunjukkan awal dan akhir dari suatu proses

2.4.2 State Transition Diagram (STD)

STD merupakan diagram yang menggambarkan bagaimana *state* yang satu dihubungkan dengan *state* yang lain pada satu waktu. STD menggambarkan suatu *state* yang mempunyai kondisi dimana dapat menyebabkan perubahan satu *state* ke *state* yang lain.

STD pada dasarnya merupakan sebuah diagram yang terdiri dari *state* dan *transisi state*. *Transisi state* terdiri dari kondisi dan aksi. Transisi diantara kedua keadaan pada umumnya disebabkan oleh suatu kondisi. Kondisi adalah suatu kejadian yang dapat diketahui oleh sistem. Sedangkan aksi adalah tindakan yang dilakukan oleh sistem apabila terjadi perubahan *state* atau reaksi dari sistem.

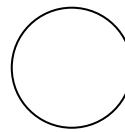


Gambar 2.2 Contoh perubahan state

Adapun komponen atau simbol yang digunakan dalam diagram STD adalah sebagai berikut :

a. Modul

Menggunakan simbol lingkaran kecil (gambar 2.3) yang mewakili modul yang dipanggil apabila terjadi suatu tindakan.



Gambar 2.3 Notasi modul

b. Tampilan kondisi (state)

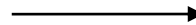
Merupakan layer yang ditampilkan menurut keadaan atau atribut, untuk memenuhi suatu tindakan pada waktu tertentu yang mewakili suatu bentuk keberadaan atau kondisi tertentu. Disimbolkan dengan gambar kotak seperti yang terlihat pada gambar 2.4.



Gambar 2.4 *Notasi tampilan*

c. Tindakan (*State Transition*)

Menggunakan simbol anak panah (gambar 2.5 disertai keterangan tindakan yang dilakukan).



Gambar 2.5 *Notasi tindakan*

2.5 Spesifikasi Proses

Spesifikasi proses merupakan penjelasan dari proses-proses yang terjadi di dalam sistem. Spesifikasi proses harus dipahami dengan baik oleh pemakai maupun membuat sistem. Spesifikasi proses akan menjadi pedoman untuk mendekati spesifikasi suatu proses dengan menggunakan salah satu metode, yaitu :

1. Tabel keputusan

Tabel ini digunakan sebagai alat bantu untuk menyelesaikan logika di dalam *flowchart*. Tabel keputusan digunakan bila kondisi yang akan diseleksi di dalam program yang jumlahnya banyak dan rumit.

2. Bahasa terstruktur

Seperti bahasa inggris terstruktur dan bahasa indonesia terstruktur.

3. Kondisi awal dan akhir

Kondisi ini menjelaskan pada awalnya dan tujuan yang akan dicapai.

4. Bagan alir (*flowchart*)

Bagan alir (*flowchart*) adalah suatu bagan yang menggambarkan arus logika dari data yang akan diproses dalam suatu program dari awal sampai akhir.

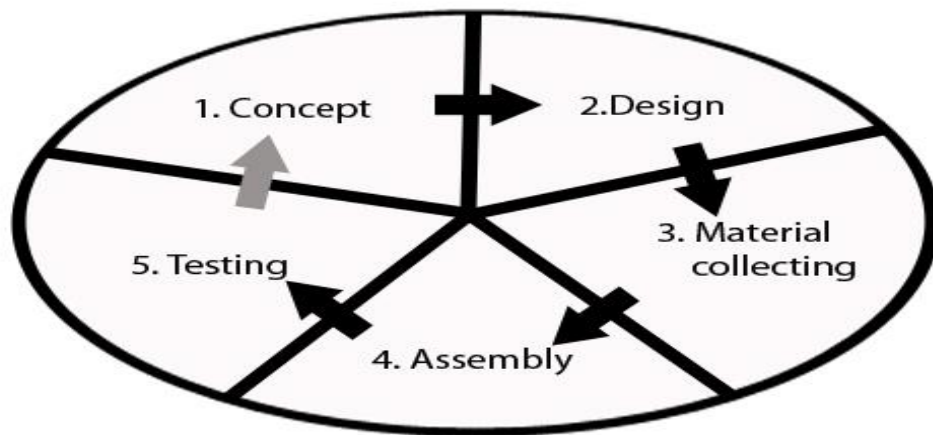
5. Bentuk narasi atau cerita

Berisi keteranga-keterangan tertulis mengenai program, dapat meliputi tujuan program, struktur program, data yang dibutuhkan, prosedur, bentuk input/output, tanggal dibuat, dan sebagainya.

2.6 ***System Development Life Cycle (SDLC)***

2.6.1 Metodologi pengembangan Multimedia

Pada tahap ini penulis mengambil teori-teori yang berkaitan dengan penelitian seperti: monitoring ruangan serta program pendukung Macromedia Flash. Berikut adalah 6 tahapan dari metodologi pengembangan multimedia, yaitu *concept*, *design*, *material collecting*, *assembly*, *testing* dan *distribution* seperti gambar di bawah ini:



Gambar 2.5 Metodologi pengembangan Multimedia

1. *Concept*

Tahap *concept* (konsep) adalah tahap untuk menentukan tujuan dan siapa pengguna program (*identifikasi audience*). Selain itu menentukan macam aplikasi (presentasi, interaktif, dll) dan tujuan aplikasi (hiburan, pelatihan, pembelajaran, dll).

2. *Design*

Design (perancangan) adalah tahap membuat spesifikasi mengenai arsitektur program, gaya, tampilan dan kebutuhan material/bahan untuk program.

3. *Material Collecting*

Material Collecting adalah tahap dimana pengumpulan bahan yang sesuai dengan kebutuhan dilakukan. Tahap ini dapat dikerjakan paralel dengan tahap *assembly*. Pada beberapa kasus, tahap *Material Collecting* dan tahap *Assembly* akan dikerjakan secara linear tidak paralel.

4. Assembly

Tahap *assembly* (pembuatan) adalah tahap dimana semua objek atau bahan multimedia dibuat. Pembuatan aplikasi didasarkan pada tahap *design*.

5. Testing

Dilakukan setelah selesai tahap pembuatan (*assembly*) dengan menjalankan aplikasi/program dan dilihat apakah ada kesalahan atau tidak. Tahap ini disebut juga sebagai tahap pengujian *alpha (alpha test)* dimana pengujian dilakukan oleh pembuat atau lingkungan pembuatnya sendiri.

2.7 Pengenalan Flash

Flash merupakan salah satu program animasi yang sangat populer saat ini, semua pekerjaan yang berhubungan dengan animasi 2D dapat dilakukan dengan mudah menggunakan program ini.

Dalam pembuatan suatu aplikasi seperti gambar bergerak atau animasi, Flash memiliki kelebihan yaitu bahwa objek-objek yang digunakan untuk pembuatan animasi tersebut dapat digunakan secara berulang-ulang (*reuseable*), sehingga dapat memperkecil ukuran file Flash yang dihasilkan. Selain itu objek yang dapat digunakan secara berulang-ulang juga akan lebih memudahkan dalam *editing* pada *movie*.

Pada dasarnya tiap simbol memiliki perilaku atau *behavior* tersendiri, yaitu bagaimana simbol tersebut akan berperilaku dalam *movie*. Untuk itu ada tiga macam perilaku simbol, yaitu:

1. *Movie Clip* : Simbol ini terdiri dari rangkaian gambar dan berperilaku seperti film, yang secara default akan dimainkan secara berulang-ulang.
2. *Button* : Simbol ini berperilaku / berfungsi sebagai tombol yang dapat diklik.
3. *Graphic* : Simbol ini berperilaku sebagai gambar.

2.7.1 Library dan Shared Library

Setiap kali simbol dibuat, simbol tersebut akan berada di dalam *library* panel. Karena dalam *movie* yang cukup besar, akan menyebabkan jumlah simbol yang digunakan juga banyak maka dibutuhkan adanya suatu *library*. *Library* ini berguna untuk mengorganisasikan simbol-simbol dalam susunan yang memudahkan penggunaannya. Hal tersebut dilakukan dengan cara mengelompokkan simbol-simbol dalam folder-folder.

Adapun penggunaan *shared library* misalnya pada saat :

1. Menggunakan satu *file sound* untuk situs yang berbeda.
2. Menggunakan satu sumber elemen animasi untuk berbagai *movie*.
3. Menggunakan satu *library* yang memudahkan revisi sebuah situs secara keseluruhan.

File *movie* yang didalamnya terdapat *shared library* harus dipublish sebagai swf agar *shared library* dapat dipanggil file *movie* yang lain, karena *shared library* merupakan *external reference*.

2.7.2 Layer

Suatu *layer* dapat dianggap sebagai suatu lapisan transparan yang melekat satu sama lain. Hal ini dapat terjadi apabila tidak terdapat satu pun elemen di suatu *layer* sehingga *layer* tersebut akan menjadi tembus pandang. Karena pada dasarnya *layer* membantu mengorganisasikan elemen-elemen yang ada di dalam *movie*.

Selain itu *layer* juga dapat digunakan untuk membatasi apa yang kelihatan dari objek di dalam *layer* lain sebatas bentuk objek yang ada *mask layer*nya, *layer* yang demikian disebut *mask layer* sedangkan *layer* yang dibatasi disebut *masked layer*. Prinsip dari *mask layer* sering digunakan seperti dalam pembuatan animasi kaca pembesar, animasi *spotlight*, animasi sumbu bom dan lain-lain.

2.7.3 Animasi Dalam Flash

Animasi dibuat dengan membentuk serangkaian *frame* yang berisi grafik di dalam *timeline*. *Keyframe* adalah *frame* di mana terdapat perubahan yang spesifik di dalam animasi. Sebuah *movie Flash* dapat dibagi dalam berbagai *scene*, dan biasanya satu *scene* menampilkan satu adegan. Tujuan pembagian *movie* ke dalam *scene* adalah untuk memudahkan dalam mengorganisasikan *movie*. Ada beberapa cara dalam pembuatan suatu animasi dalam Flash yaitu:

1. Animasi *Keyframe to Keyframe*

Biasanya cara ini digunakan untuk membuat animasi yang membutuhkan penggambaran *frame* demi *frame*, contohnya adalah seperti pada film kartun. Adapun kerugian menggunakan cara ini adalah ukuran file akan menjadi sangat besar, karena setiap *frame* membutuhkan satu penggambaran yang jelas dan berbeda lagi dengan penggambaran pada *frame* berikutnya.

2. Animasi *Motion Tweening*

Cara ini digunakan khusus untuk menganimasikan *instance*, sehingga apabila objek yang akan diberi *motion tweening* bukan *instance* maka *Flash* akan membuat simbol objek tersebut di *library* dengan nama *Tween 1*, *Tween 2* dan seterusnya.

Dengan menggunakan cara ini ukuran *file* pada *movie* akan lebih kecil bila dibandingkan dengan menggunakan *keyframe to keyframe*, karena pada dasarnya cara ini hanya membutuhkan satu penggambaran pada satu *frame* saja dan untuk *frame-frame* berikutnya hanyalah merupakan duplikat dari *frame* yang pertama.

3. Animasi *Shape Tweening*

Biasanya cara ini digunakan untuk pembuatan animasi dengan teknik *morphing*. Dengan menggunakan cara ini ukuran *file* yang terbentuk tidak akan jauh berbeda dengan menggunakan *motion tweening*, karena cara ini hanyalah membutuhkan penggambaran pada *frame* awal dan *frame* akhir saja, dimana penggambaran pada *frame* awal tidak sama dengan penggambaran pada *frame* akhir.

4. Animasi *Motion Guide*

Cara ini biasanya digunakan untuk menganimasikan suatu objek yang bergerak seolah-olah berada pada suatu jalur tertentu. Seperti pergerakan roket misalnya, roket akan bergerak seolah-olah berada di dalam suatu jalur yang meliuk-liuk dan berarah. Memang pada dasarnya cara ini menggunakan satu layer penuntun (*layer guide*) yang digunakan untuk membuat jalur dari pergerakan objek yang akan dianimasikan. Cara ini hanya membutuhkan satu penggambaran objek saja dan kemudian objek yang sama

diletakkan pada *frame-frame* yang berbeda, yaitu pada setiap belokkan pada jalur yang telah dibuat dalam layer *guide* dengan menggunakan *pencil tool*.

5. Animasi Berulang (*Loop Animation*)

Cara animasi berulang ini biasa digunakan sebagai *background* atau untuk pembuatan menu yang terus berputar, orang berjalan dan rotasi planet. Prinsip utama dari cara ini adalah membuat *frame* awal dan *frame* akhir dimana objek yang ada di dalam *frame* awal dan objek yang ada di dalam *frame* akhir mempunyai bentuk yang sama atau hampir sama.

2.7.4 **Actionscript**

Actionscript adalah bahasa pemrograman yang ada di dalam Flash. *Actionscript* digunakan untuk mengontrol objek yang ada di *Flash*, untuk membuat navigasi dan elemen interaktif lainnya serta membuat *movie Flash* dan aplikasi web yang interaktif.

2.7.4.1 **Fungsi Actionscript**

Actionscript memiliki beberapa fungsi atau peran yang dapat membantu dalam merancang situs atau *movie Flash*, fungsi-fungsi tersebut adalah :

A. Membuat system navigasi

B. Hal ini berkaitan dengan bagaimana *actionscript* dapat mengantar *user* menjelajahi situs yang telah di-*publish*. Dengan demikian animasi situs tersebut tidak hanya menjadi “film iklan” belaka.

C. Menambah interaktivitas dengan user

Dengan *actionscript* user dapat berinteraksi dengan setiap elemen yang ada di dalam *movie Flash* baik itu tombol, *movie* klip maupun teks.

D. Membuat situs yang dinamis

Actionscript dapat digabungkan dengan berbagai bahasa lain, seperti *JavaScript*, ASP, PHP, CGI, MySQL ataupun XML untuk membuat berbagai aplikasi yang sulit atau bahkan yang tidak dapat dibuat hanya dengan *actionscript* saja.

E. Mempertahankan *user* untuk kembali mengunjungi situs

Actionscript dapat membuat situs *Flash* yang menarik dan interaktif, yang akan menambah kualitas *user experience* saat menjelajah situs. Misalnya adanya *game* di dalam situs *Flash* yang dibuat secara tidak langsung akan dapat mempertahankan *user* untuk kembali mengunjungi situs tersebut.

Pada prinsipnya *actionscript* hanya merupakan sebuah alat bantu yang mempermudah untuk merancang atau membuat *movie* atau pun juga situs Flash.

2.7.4.2 Kategori *Actionscript*

Pada normal *mode action* dibagi ke dalam beberapa kategori, yaitu :

1. *Basic Actions*

Kategori ini menampung *action* sederhana yang seringkali digunakan untuk *movie* Flash, misalnya untuk navigasi dan perilaku tombol.

2. *Actions*

Kategori ini meliputi *basic actions* ditambah dengan banyak *action* lain yang kompleks.

3. *Operators*

Kategori ini berisi simbol yang digunakan, misalnya untuk operasi logika dan matematika seperti tambah, kurang, kali dan lain-lain.

4. *Functions*

Functions berisi *action* yang dapat menerima data tertentu yang kemudian menghasilkan informasi yang dapat digunakan.

5. *Properties*

Kategori *properties* berisi *property* objek yang dapat dimodifikasi. Sebagian besar *property* ini digunakan untuk objek *movie klip*.

6. *Objects*

Flash memiliki kelas objek yang sudah didefinisikan (*predefined class*). Kelas-kelas ini berada dalam kategori *objects* di *actionscript*.

2.7.4.3 Variabel dan Tipe Data

Variabel adalah suatu “*container*” yang mengandung suatu informasi tertentu. *Container* atau wadahnya selalu sama tetapi informasi yang terkandung di dalam suatu *variable* dapat berubah-ubah bergantung dari tipe informasi atau datanya. Ada empat macam tipe data, yaitu :

1. *Number* : Angka tertentu.
2. *String* : Kumpulan berbagai angka, huruf atau simbol.
String selalu berada di dalam tanda petik.
3. *Boolean*: Nilai yang berisi *true* atau *false*, dapat juga dipakai angka 1 untuk *true* dan 0 untuk *false*.
4. *Object* : Nama objek yang dibuat berdasar proses *instantiate*.

2.7.4.4 Operator

Suatu *variable* dapat dimodifikasi dengan menggunakan operator. *Variable* dapat diganti nilainya dengan informasi terbaru dari kondisi movie klip, mengganti suatu nilai dengan *true* atau *false*, atau dengan menambah, mengurangi dan mengalikan. Untuk memodifikasi informasi ini merupakan pekerjaan dari operator. Operator dalam Flash yaitu :

1. *Assignment operator* (=)

Operator ini adalah operator yang menetapkan suatu nilai untuk suatu variabel.

2. *Arithmetic operator* (+, -, *, /, ++, --, +=, -=, *=, /=)

Operator ini terdiri atas simbol-simbol yang biasa digunakan dalam matematika.

3. *Comparison operator* (=, <, >, <=, >=, !=)

Operator ini membandingkan kedua *expression* di antara simbol operator dan mengeluarkan suatu nilai *true* atau *false*.

2.7.4.5 Struktur Bahasa *Actionscript*

Sama seperti menggunakan bahasa pada umumnya yang urutannya adalah kata benda, kata sifat dan kata kerja dalam susunan yang dimengerti orang lain. Pada *actionscript* juga mempunyai susunan yaitu objek, *property*

dan *method* sampai membentuk *statement* yang dimengerti Flash. Dimana masing-masing kata dipisahkan oleh dot (.).

Contoh : Balerina.pakaian.warna = putih

(dibaca putih adalah pakaian dari Balerina)

Contoh di atas menunjukkan suatu *statement* dengan objek = Balerina, *property* = pakaian dan *method* = warna dan nilainya adalah putih.

2.7.4.6 Tanda Baca dalam *Actionscript*

Seperti dalam bahasa, tulisan memiliki tanda baca untuk mengakhiri kalimat atau mengelompokkan berbagai kalimat menjadi satu paragraf, Flash juga memiliki tanda baca seperti itu.

1. *Curly braces* dan *semicolon*

Curly braces atau kurung kurawal ({}) digunakan untuk melakukan pengelompokan, sedangkan *semicolon* atau titik koma (;) digunakan untuk mengakhiri suatu *statement*.

Contoh : on(release) {
 stopAllsounds();
 play();
 }

2. *Comments*

Comments adalah keterangan yang ditambahkan di dalam *script* untuk menjelaskan apa arti kode tersebut. *Comments* dapat membantu memahami *script* apabila hendak me-review kembali kode tersebut untuk perbaikan. *Comments* ditandai dengan *double slash (//)*.

Contoh : on(release) {

```
//matikan semua suara  
stopAllsounds();  
  
//mainkan  
play();  
}
```