

HADI SUTOPO

PANDUAN  
PRAKTIKUM

1 2 3

PEMROGRAMAN  
MULTIMEDIA



The image shows a screenshot of a multimedia software interface. It features a grid background with a ruler at the bottom. A control panel on the right side contains four buttons: 'Hitung diameter', 'Visualisasi', 'Reset', and 'Kembali'. The word 'INTERAKTIF' is visible in the bottom right corner of the interface.

PANDUAN PRAKTIKUM KOMPUTER

MATA KULIAH

**PEMROGRAMAN  
MULTIMEDIA**

HADI SUTOPO

# PENGANTAR

Pemrograman dengan ActionScript dapat digunakan untuk pembuatan visualisasi di segala bidang terutama aplikasi interaktif berbasis web. ActionScript adalah pemrograman visual berorientasi object. Seperti JavaScript, ActionScript mempunyai sintaks, tata bahasa dan struktur yang mirip dengan bahasa pemrograman C++.

ActionScript merupakan bahasa pemrograman (scripting) yang digunakan dalam Macromedia Flash. Penggunaan ActionScript mempunyai tujuan:

- memberikan kebebasan berkreasi bagi desainer
- membuat animasi interaktif
- dapat menampilkan animasi tertentu (non linier)

Diktat ini disiapkan untuk pegangan penduan praktek komputer pada mata kuliah “Pemrograman Multimedia” pada Program Studi Teknik Informatika. Dengan adanya modul untuk praktikum ini diharapkan dapat melengkapi bahan pembelajaran yang diperlukan. Harapan penulis, semoga diktat ini dapat bermanfaat bagi dunia pendidikan.

Penulis mengucapkan terima kasih kepada rekan-rekan lain yang telah memberikan bantuan dalam pelaksanaan penulisan buku ini. Kami menyadari banyak kekurangan pada diktat ini. Sangat kami harapkan kritik, saran dan gagasan dari rekan-rekan pemerhati serta pelaku multimedia khususnya dan teknologi informasi pada umumnya.

Jakarta, Maret 2007

Penulis

# DAFTAR ISI

## **PENGANTAR      iii**

## **Bab 1 ActionScript      1**

### **1.1 Berorientasi Object      1**

- 1.1.1 Karakteristik dari object      2
- 1.1.2 Karakteristik pemrograman berorientasi object      3
- 1.1.3 Tema berorientasi object      3

### **1.2. Pengenalan ActionScript      4**

### **1.3 Sintaks      5**

- 1.3.1 Dot syntax      6
- 1.3.2 Slash syntax      6
- 1.3.3 Kurung Kurawal      7
- 1.3.4 Semicolons      7
- 1.3.5 Kurung      7
- 1.3.6 Huruf besar dan kecil      8
- 1.3.7 Komentar      8
- 1.3.8 Keywords      9
- 1.3.9 Konstanta      9

### **1.4 Tipe Data      9**

- 1.4.1 String      9
- 1.4.2 Number      10
- 1.4.3 Boolean      11
- 1.4.4 Object      11
- 1.4.5 Movie clip      11

### **1.5 Variabel      12**

- 1.5.1 Nama variabel      12
- 1.5.2 Menuliskan variabel      12
- 1.5.3 Jangkauan variabel      13
- 1.5.4 Deklarasi variabel      14
- 1.5.5 Menggunakan variabel dalam suatu script      14

## **1.6 Operator 16**

- 1.6.1 Operator precedence 16
- 1.6.2 Operator asosiatif 18
- 1.6.3 Operator numerik 18
- 1.6.4 Operator perbandingan 18
- 1.6.5 Operator string 19
- 1.6.6 Operator logika 19
- 1.6.7 Operator bitwise 20
- 1.6.8 Operator kesamaan dan assignment 20
- 1.6.9 Dot dan array akses operator 21

## **1.7 Penggunaan Action 22**

- 1.7.1 Menuliskan target path 23
- 1.7.2 Memberi nama suatu instance dari movie clip 23
- 1.7.3 Mengenali movie yang diambil 24
- 1.7.4 Memasukkan target path dari suatu movie 24

## **1.8 Mengontrol Aliran Script 24**

- 1.8.1 Penggunaan pernyataan “if” 24
- 1.8.2 Action perulangan 25

## **1.9 Predefined function dan object 27**

- 1.9.1 Memanggil fungsi 27
- 1.9.2 Memanggil fungsi 27
- 1.9.3 Memanggil fungsi dalam Normal Mode 28

## **1.10 Fungsi Custom 28**

- 1.10.1 Mendefinisikan fungsi 28
- 1.10.2 Melewatkan argumen pada fungsi 28
- 1.10.3 Menggunakan variabel lokal dalam fungsi 29
- 1.10.4 Mengembalikan nilai dari fungsi 29
- 1.10.5 Memanggil fungsi 30

## **1.11 Menggunakan Predefined Object 30**

- 1.11.1 Membuat object 30
- 1.11.2 Mengakses property dari object 31

## **1.12 Menggunakan Custom Object 32**

- 1.12.1 Membuat object 32
- 1.12.2 Membuat inheritance 33

## **1.13 Latihan 33**

## **Bab 2 Operasi Matematika dan Logika 35**

### **2.1 Tipe Data Primitif 35**

- 2.1.1 Operator tambah (+) 35
- 2.1.2 Operasi matematik 40
- 2.1.3 Kode ASCII 44

### **2.2 Latihan 45**

## **Bab 3. Visualisasi Algoritma 47**

### **3.1 Visualisasi Algoritma DDA 47**

- 3.1.1 Algoritma 48

### **3.2 Visualisasi Konversi Bilangan Biner 55**

- 3.2.1 Sistem bilangan 55
- 3.2.2 Konversi bilangan desimal ke biner 56
- 3.2.3 Visualisasi konversi bilangan desimal ke biner 57
- 3.2.4 Flash Movie 57

### **3.3 Visualisasi Sorting 63**

- 3.3.1 Selection Sort 63
- 3.3.2 Bubble Sort 64
- 3.3.3. Sorting dengan manual 66
- 3.3.3. Sorting dengan pemrograman 69

### **3.4 Visualisasi Menara Hanoi 75**

- 3.4.1 Rumusan Masalah 76
- 3.4.2 Tujuan 76
- 3.4.3 Landasan Teori 76
- 3.4.4 Flash Movie 82

### **3.5 Latihan 86**

## **Daftar Pustaka 87**



# 1

## ActionScript

Pemrograman dengan ActionScript dapat digunakan untuk pembuatan visualisasi di segala bidang terutama aplikasi interaktif berbasis web. ActionScript adalah pemrograman visual berorientasi object. Seperti JavaScript, ActionScript mempunyai sintaks, tata bahasa dan struktur yang mirip dengan bahasa pemrograman C++.

ActionScript mempunyai beberapa komponen, seperti predefined object dan fungsi. ActionScript mengikuti sintaks (syntax), keywords, operator dan variabel untuk menyimpan dan mengambil informasi. European Computers Manufacturer Association (ECMA) membuat dokumen ECMA-262 yang diambil dari JavaScript, dan ECMA-262 menjadi spesifikasi ActionScript.

ActionScript merupakan bahasa pemrograman (scripting) yang digunakan dalam Macromedia Flash. Penggunaan ActionScript mempunyai tujuan:

- memberikan kebebasan berkreasi bagi desainer
- membuat animasi interaktif
- dapat menampilkan animasi tertentu (non linier)

### 1.1 Berorientasi Object

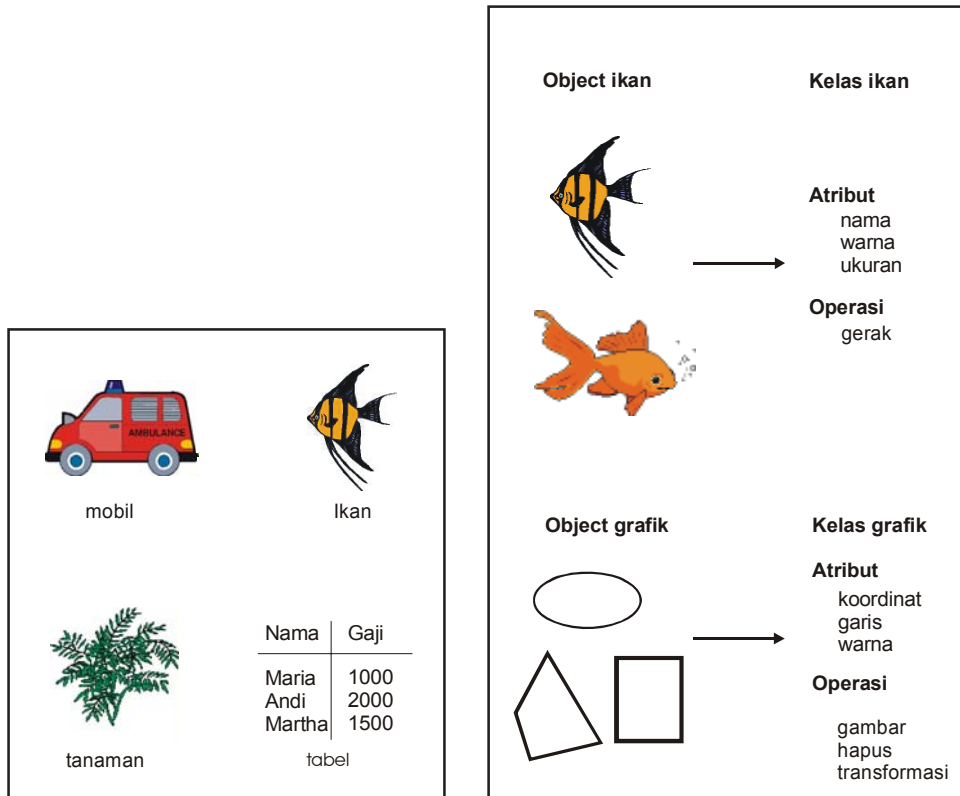
“Berorientasi object” mempunyai pengertian bahwa masalah diorganisasi sebagai kumpulan dari object tertentu yang memiliki struktur data dan perilakunya. Hal ini yang membedakan dengan pemrograman konvensional dimana struktur data dan perilaku hanya berhubungan secara terpisah. Secara umum mencakup empat hal untuk menentukan karakteristik berorientasi object, yaitu

- identifikasi
- klasifikasi
- pewarisan (inheritance)
- polimorfisme (polymorphism)

### 1.1.1 Karakteristik dari object

Identitas berarti bahwa data yang diukur mempunyai deskriptor yang membedakan entitas, dan disebut object. Object dapat kongkrit, seperti halnya arsip dalam sistem, atau konseptual seperti prosedur kerja. Setiap object mempunyai sifat yang melekat pada identitasnya. Dua object dapat berbeda walaupun semua nilai atributnya identik. Pada Gambar 1 dapat dilihat beberapa macam object.

Klasifikasi berarti mengumpulkan data (property) dan perilaku (operasi) yang mempunyai struktur data sama ke dalam satu grup yang disebut class. Class merupakan abstraksi yang menjelaskan sifat penting pada suatu aplikasi dan mengabaikan yang lain. Setiap class menunjukkan suatu kumpulan infinite yang mungkin dari object. Suatu object dapat dikatakan sebagai instance dari class. Setiap instance dari class mempunyai nilai individu untuk setiap nama property dan operasi, tetapi memiliki bersama property dan operasi dengan instance lain dalam class. Gambar 2 menunjukkan dua class dengan beberapa instance yang berhubungan dengannya.



Gambar 1.1 (kiri) Macam-macam object dan Gambar 1.2 (kanan) Class dan object

Dalam dunia nyata, suatu operasi adalah abstraksi dari analogi perilaku terhadap object yang berbeda. Setiap object mengetahui bagaimana melakukan operasinya. Dalam bahasa pemrograman berorientasi object, secara otomatis bahasa akan memilih method yang tepat untuk menjalankan operasinya berdasar nama dimana dilakukan

operasi terhadapnya. Pengguna dari operasi tidak perlu khawatir berapa banyak method yang terdapat dalam implementasi. Class baru dapat ditambahkan tanpa mengubah code (script) yang sudah ada.

### **1.1.2 Karakteristik pemrograman berorientasi object**

Pemrograman berorientasi object mempunyai tiga karakteristik utama, yaitu encapsulation, inheritance dan polymorphism.

#### **Encapsulation**

Encapsulation (pengkapsulan) merupakan dasar untuk pembatasan ruang lingkup program terhadap data yang diproses. Data dan prosedur atau fungsi dikemas dalam bersama-sama dalam suatu object, sehingga prosedur atau fungsi lain dari luar tidak dapat mengaksesnya. Data terlindung dari prosedur atau object lain kecuali prosedur yang berada dalam object itu sendiri.

#### **Inheritance**

Inheritance (pewarisan) adalah teknik yang menyatakan bahwa anak dari object akan mewarisi data/properties dan method dari induknya langsung. Property dan method dari object induk diturunkan kepada anak object, demikian seterusnya. Pendefinisian object dipergunakan untuk membangun suatu hirarki dari object turunannya, sehingga tidak perlu membuat property dan method lagi pada anaknya, karena telah mewarisi sifat induknya.

Suatu class dapat ditentukan secara umum, kemudian ditentukan secara spesifik menjadi subclass. Setiap subclass mempunyai hubungan atau mewarisi semua sifat yang dimiliki oleh class induknya, dan ditambah dengan sifat unik yang dimilikinya. Sifat yang dimiliki oleh class induknya tidak perlu diulang dalam setiap subclass.

Sebagai contoh, Kucing dan Ikan adalah subclass dari Binatang. Kedua subclass mewarisi sifat yang dimiliki oleh Binatang, yaitu:

- mempunyai tubuh
- dapat bergerak

#### **Polymorphism**

Polymorphism (polimorfisme) yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyai bentuk dan perilaku berbeda. Polimorfisme mempunyai arti bahwa operasi yang sama dapat mempunyai perbedaan dalam class yang berbeda. Operasi move mempunyai perbedaan dalam class windows atau Binatang.

### **1.1.3 Tema berorientasi object**

Teknologi berorientasi object mempunyai beberapa tema sebagai dasar. Walaupun tema ini tidak unik untuk sistem berorientasi object, namun sangat menunjang sistem berorientasi object.

## **Abstraksi**

Abstraksi berarti memfokuskan pada aspek yang melekat dari entitas dan mengabaikan sifat yang sementara. Dalam pengembangan sistem, hal ini berarti memfokuskan pada apakah suatu objek dan apa yang dikerjakan oleh object tersebut, sebelum menentukan implementasinya.

## **Pengkapsulan**

Pengkapsulan (juga information hiding) terdiri dari pemisahan aspek eksternal dari suatu object, dimana dapat diakses oleh object lain. Pengkapsulan melindungi program dari saling keterkaitan sehingga kesempatan kecil mempunyai akibat penyimpangan. Implementasi dari suatu object dapat berubah tanpa mempengaruhi aplikasi yang digunakan. Pengkapsulan mempunyai kemampuan untuk mengkombinasikan struktur data dengan perilaku dalam satu entitas.

## **Menggabungkan data dengan perilaku**

Code yang bukan berorientasi menampilkan isi dari window harus membedakan tipe dari setiap bentuk, seperti poligon, lingkaran atau teks dan memanggil prosedur yang tepat untuk menampilkannya. Program berorientasi object lebih sederhana meminta operasi draw pada setiap bentuk, penentuan dimana prosedur digunakan dibuat implisit oleh setiap object berdasarkan classnya. Tidak perlu selalu mengulang pilihan prosedur setiap saat operasi dilakukan dalam program aplikasi.

## **Penggunaan Bersama**

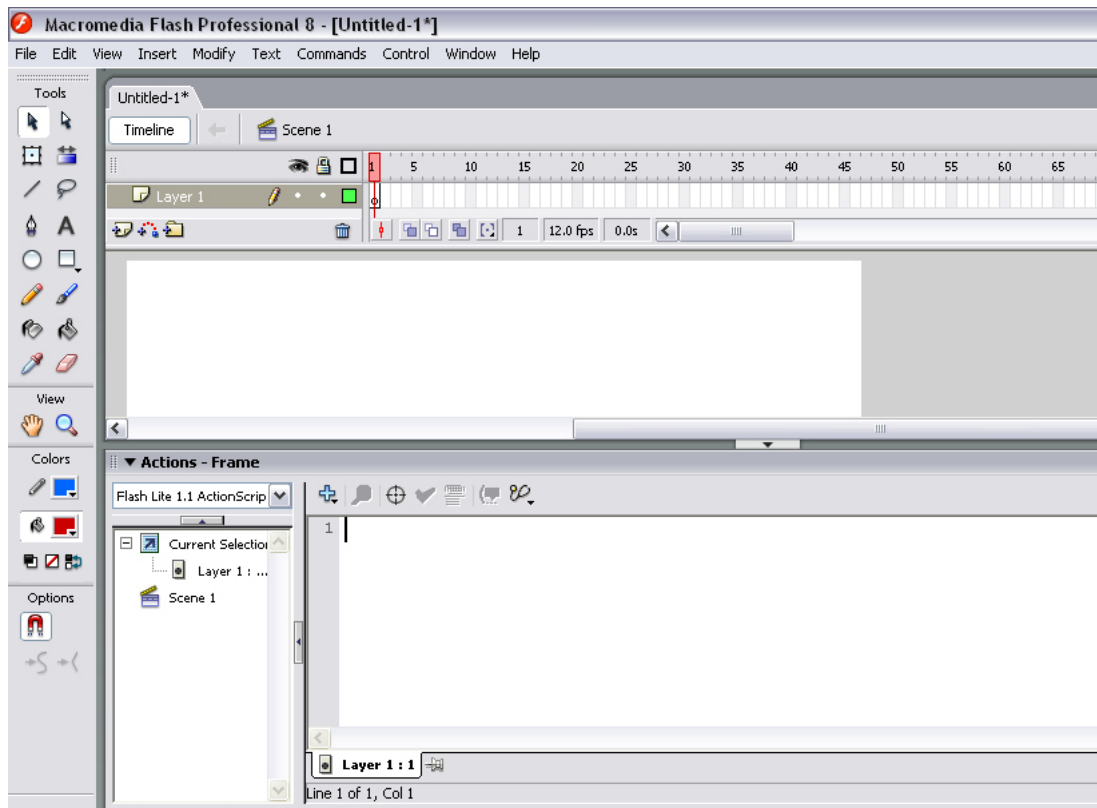
Teknik berorientasi object menawarkan reusability yaitu penggunaan bersama untuk beberapa tingkat yang berbeda. Pewarisan struktur data dan perilaku memungkinkan penggunaan bersama antara beberapa subclass yang sama tanpa redundancy. Pemakaian code bersama adalah keuntungan utama dari bahasa berorientasi objek.

## **1.2. Pengenalan ActionScript**

Gagasan dasar pemrograman berorientasi object adalah menggabungkan data dengan fungsi. Fungsi object dapat mengakses suatu object, mengakses data dan mengembalikan nilai. Data dan fungsi terbungkus pada suatu entry tunggal, data terlindung dari akses yang dilakukan oleh fungsi lain yang tidak dikehendaki.

Program berorientasi object pada umumnya terdiri dari beberapa object yang berkomunikasi dengan object lain dengan memanggil fungsi anggotanya. Beberapa bahasa pemrograman berorientasi object menggunakan istilah fungsi anggota atau method, data juga sama dengan atribut atau instance variable. Memanggil fungsi anggota dari suatu object juga berarti mengirim message ke object tersebut.

Menuliskan program ActionScript dilakukan pada panel Actions, yaitu tempat untuk membuat perintah yang ditujukan pada object atau frame.



Gambar 1.3 Panel Action

Untuk menggunakan panel action:

1. Pilih Window → Actions. Standar panel action adalah Normal Mode. Untuk mengubah ke Expert Mode, klik tanda panah di atas panel Actions. Setelah menu yang muncul, dapat dipilih Expert Mode.
2. Untuk membuat membuat action dapat dilakukan dengan mengklik tanda +, maka akan muncul kotak dengan berbagai pilihan, seperti Basic Actions, Actions, Operators, Functions, Properties dan Objects.

Bila dipilih Basic Actions, maka akan muncul tampilan seperti Gambar 8.4. Cara lain adalah dengan mengklik tanda panah pada pull-down menu Basic Actions. Action belum aktif karena belum ada frame, object atau instance yang dipilih.

### 1.3 Sintaks

ActionScript mempunyai aturan tata bahasa dan tanda baca tersendiri sehingga mengenali awal dan akhir sebuah pernyataan. Suatu pernyataan diakhiri dengan tanda semicolon (;). Sintaks (syntax) salam ActionScript ada dua macam, yaitu dot syntax dan slash syntax.

### 1.3.1 Dot syntax

Dalam ActionScript, suatu dot (.) digunakan untuk:

- menunjukkan property dari method dari object atau movie clip
- mengenali target path ke suatu movie clip atau variable.

Dot syntax dimulai dengan nama dari object atau movie clip, diikuti dengan dot, dan diakhiri dengan property, method, atau variable.

Contohnya, movie clip mempunyai property:

`_x` menunjukkan posisi ordinat x pada stage  
`lingkaranMc._x` adalah ordinat x dari movie clip `lingkaranMc`

Selain property, movie clip juga mempunyai method `play()`. Ekspresi suatu method dari object atau movie clip mengikuti pola yang sama. Dot syntax juga digunakan untuk dua alias khusus yaitu `_parent` dan `_root`. Contohnya, method `play` dari instances `lingkaranMc` menggerakkan `playhead` pada `Timeline` dari `lingkaranMc`, seperti pernyataan berikut:

```
lingkaranMc.play();
```

- Alias `_root` berhubungan dengan `Timeline` utama.
- Alias `_root` dapat digunakan untuk membuat target path yang absolut.

Contohnya, pernyataan berikut memanggil fungsi hitung pada movie clip dari `Timeline` utama.

```
_root.functions.hitung();
```

- Alias `_parent` mengacu pada movie clip bila mempunyai hirarki.
- Alias `_parent` dapat digunakan untuk membuat target path yang relatif.

Contohnya, bila movie clip `burung` terdapat dalam movie clip `binatang`, pernyataan `instance movie clip burung` memerintahkan `binatang` untuk berhenti:

```
_parent.stop();
```

### 1.3.2 Slash syntax

Slash syntax digunakan dalam Flash 3 and 4 untuk menunjukkan target path dari movie clip atau variabel. Sintaks ini masih dapat digunakan pada Flash 5, tetapi tidak disarankan. Dalam slash syntax, slash digunakan untuk menunjukkan target path atau variabel di samping dot syntax. Untuk menunjukkan suatu variabel, nama variabel dituliskan sesudah colon (:).

```
lingkaranMc/childMovieClip:myVariable
```

Target path dapat dituliskan dengan dot syntax, sebagai berikut:

```
lingkaranMc.childMovieClip.myVariable
```

Slash syntax biasanya digunakan untuk action *tellTarget*, yang juga tidak disarankan.

### 1.3.3 Kurung Kurawal

Pernyataan pada ActionScript dikelompokkan bersama ke dalam satu blok dengan kurung kurawal ({}), seperti script berikut:

```
on(release) {  
    hari = new Date();  
    bulan = myDate.getMonth();  
}
```

### 1.3.4 Semicolons

Suatu pernyataan dalam ActionScript diakhiri dengan semicolon, tetapi bila tanda tersebut dihilangkan, kompilasi oleh Flash masih dapat dilakukan dengan sukses. Contohnya, pernyataan-pernyataan ini diakhiri dengan semicolcon (;).

```
hasil = "Benar";  
nilai = 1;
```

Pernyataan yang sama dapat ditulis tanpa menggunakan semicolon.

```
hasil = "Benar"  
nilai = 1
```

### 1.3.5 Kurung

Untuk mendefinisikan suatu fungsi, argumen harus ditempatkan di antara tanda kurung.

```
function daftar_karyawan (nama, umur, alamat, pendidikan) {  
    ...  
}
```

Untuk memanggil suatu fungsi, setiap argumen (parameter) dilewatkan dalam fungsi di antara tanda kurung, seperti berikut.

```
data_karyawan ("Andi", 25, "Jakarta", "Akademi");
```

Tanda kurung juga digunakan untuk mengevaluasi ekspresi di sebelah kiri dari dot (.) pada dot syntax. Contohnya seperti pernyataan berikut, tanda kurung mengakibatkan `new Color(this)` dievaluasi dan membuat suatu objek warna baru.

```
(new Color(this)).setRGB(0x00CC33));
```

Bila tidak menggunakan tanda kurung, perlu ditambahkan pernyataan pada code untuk mengevaluasinya.

```
warna = new Color(this);  
warna.setRGB(0x00CC33);
```

### **1.3.6 Huruf besar dan kecil**

Huruf besar dan kecil dapat digunakan sesuai keinginan, kecuali keywords. Penggunaan huruf besar ekuivalen dengan huruf kecil seperti contoh berikut.

```
warna.setRGB(0x00CC33);  
WARNA.setRGB(0x00CC33);
```

Namun, kebiasaan menuliskan secara konsisten dapat memudahkan ActionScript untuk mengenali nama fungsi dan variabel dengan baik.

### **1.3.7 Komentar**

Penggunaan komentar (comment) bermanfaat dalam pengembangan sistem dan pihak lain yang memerlukannya. Dalam ActionScript, komentar diawali dengan tanda `//`. Komentar digunakan untuk memberikan catatan yang diperlukan pada suatu pernyataan. Bahkan pernyataan yang sangat sederhana memerlukan komentar, supaya mudah dipahami, karena tertulis dalam catatan.

```
on(release) {  
    // hitung x pangkat y  
    hasil = Math.pow(x,y);  
}
```

Dalam window script (Action list), komentar ditampilkan berwarna pink, dengan panjang yang tidak terbatas dan tidak mengikuti aturan sintaks atau keywords.

### 1.3.8 Keywords

Keywords atau kata kunci digunakan untuk keperluan tertentu, sehingga tidak dapat digunakan sebagai variabel, fungsi atau nama label. Keywords dalam ActionScript seperti dalam tabel berikut:

break	for	new	var
continue	function	return	void
delete	if	this	while
else	in	typeof	with

### 1.3.9 Konstanta

Konstanta adalah nilai tetap yang sudah disediakan oleh Flash. Contoh konstanta adalah BACKSPACE, ENTER, QUOTE, RETURN, SPACE, sedangkan TAB adalah property dari objek Key yang berhubungan dengan tombol pada keyboard. Untuk memeriksa apakah pengguna menekan tombol Enter, digunakan pernyataan berikut:

```
if(keycode() == Key.ENTER) {
    hasil= "Game selesai"
}
```

## 1.4 Tipe Data

Tipe data menggambarkan macam elemen informasi yang dapat ditangani oleh variabel dalam ActionScript. Terdapat dua macam tipe data yaitu:

Tipe data primitif (string, number, dan Boolean)

- mempunyai nilai konstanta, sehingga dapat menangani nilai aktual dari elemen yang ada.

Tipe data referensi (movie clip dan object)

- mempunyai nilai yang dapat berubah, sehingga berisi referensi dari nilai aktual yang dimiliki oleh elemen.

### 1.4.1 String

Sebuah string adalah deretan dari karakter seperti huruf, angka dan tanda baca. Dalam ActionScript, string dituliskan di antara tanda petik double.

```
alamat= "Salemba9";
```

Operator tambah (+) dapat digunakan untuk menggabungkan atau menyambung dua buah string. Dalam ActionScript spasi pada awal atau akhir string diperlakukan sebagai bagian dari string tersebut. Ekspresi berikut termasuk spasi setelah tanda koma:

```
ucapan = "Selamat Tahun Baru, " + nama;
```

Penulisan string literal sangat sensitif, walaupun ActionScript tidak membedakan antara huruf besar dan kecil untuk nama variabel, instans, label. Contohnya, dua pernyataan berikut menempatkan teks yang berbeda pada variabel field, karena keduanya adalah literal string.

```
ucapan = "Selamat Tahun Baru";  
ucapan = "SELAMAT TAHUN BARU";
```

Untuk memasukkan tanda baca dalam string, dapat dimulai dengan karakter backslash (\) yang disebut dengan "escaping" karakter. Terdapat beberapa karakter yang ditampilkan pada ActionScript kecuali menggunakan karakter escape. Daftar karakter escape seperti tercantum sebagai berikut.

**Karakter escape**

\b	Backspace character (ASCII 8)
\f	Form-feed character (ASCII 12)
\n	Line-feed character (ASCII 10)
\r	Carriage return character (ASCII 13)
\t	Tab character (ASCII 9)
\"	Tanda petik double
\'	Tanda Petik tunggal
\\	Backslash
\000 - \377	Byte yang dinyatakan dalam bilangan oktal
\x00 - \xFF	Byte yang dinyatakan dalam bilangan heksadesimal
\u0000 - \uFFFF	Karakter yang dinyatakan dalam heksadesimal

### 1.4.2 Number

Tipe data number dapat berupa integer (bilangan bulat) atau float (bilangan pecahan). Manipulasi number dilakukan menggunakan operator aritmatik seperti:

+	tambah
-	kurang
*	perkalian

/	pembagian
%	modulo
++	increment
--	decrement

Method dari predefined object Math juga dapat digunakan untuk memanipulasi bilangan. Contoh berikut menggunakan metoda pow untuk menghitung pangkat dari bilangan 4 dipangkatkan 1.3.

```
Math.pow(4, 1.3);
```

### 1.4.3 Boolean

Boolean adalah tipe data yang mempunyai nilai true atau false. ActionScript megkonversi nilai true and false menjadi 1 and 0 bila sesuai. Boolean sering digunakan dengan operator logika untuk mengontrol aliran dari script. Contohnya, pada script berikut menggambarkan hasil yang ditampilkan pada layar adalah string literal “Anda sukses”, bila pemeriksaan kondisi menghasilkan nilai true.

Operator && (and) menghasilkan nilai true bila kedua operan adalah true, tetapi menghasilkan nilai false bila salah satu operan mempunyai nilai false. Tabel kebenaran menunjukkan hasil operasi boolean yang dapat menjadi pedoman untuk setiap pemrograman.

```
on(release) {  
    if ((ikan_mujair >=10) && (ikan_lele >=25)) {  
        hasil = “Anda sukses.”;  
    }  
}
```

### 1.4.4 Object

Suatu object adalah koleksi dari property, dimana setiap property mempunyai nama dan nilai. Nilai dari property dapat berupa tipe data primitif atau tipe data referensi. Hal ini dapat digunakan untuk menempatkan objek di dalam satu sama lain, atau membuatnya ‘nest’. Untuk membuat spesifikasi object dengan property yang dimilikinya dapat digunakan dot (.), seperti contoh code berikut.

```
mahasiswa.namaSemester.nilaiPrestasi
```

### 1.4.5 Movie clip

Movie clip adalah symbol yang dapat menjalankan animasi pada Flash movie, berisi tipe data yang berhubungan dengan elemen grafik. Tipe data movie clip dapat mengontrol symbol movie clip dengan menggunakan method dari object movie clip.

Method dapat dipanggil dengan dot (.) operator, seperti contoh berikut.

```
ikanMc.play();
```

Method dari movie clip `ikanMc` adalah `play()`. Pernyataan tersebut digunakan untuk menjalankan movie clip `ikanMc`. Bila dilakukan perubahan (editing) pada symbol movie clip, instance yang terdapat pada stage dengan sendirinya akan berubah juga.

## 1.5 Variabel

Variabel (variable) merupakan komponen penting dalam pemrograman. Variabel digunakan untuk menyimpan nilai, dan nilai yang ada padanya dapat diubah selama eksekusi program berlangsung. Variabel dapat menangani semua tipe data number, string, Boolean, object, atau movie clip. Penulisan variabel mempengaruhi perubahan nilai variabel pada saat dinyatakan dalam script.

Informasi yang umum dapat disimpan dalam variabel termasuk URL, nama pengguna, hasil dari operasi matematik, jumlah event yang terjadi, atau melakukan klik pada sebuah button. Setiap movie dan instance movie clip mempunyai satu set variabel sendiri dimana setiap variabel mempunyai nilai independen terhadap variabel dari movie atau movie clip lain.

### 1.5.1 Nama variabel

Nama variabel dibuat mengikuti aturan pemrograman pada umumnya, sebagai berikut:

- harus sebuah identifier
- tidak boleh menggunakan keyword atau Boolean literal (true atau false).
- harus unik

### 1.5.2 Menuliskan variabel

Penulisan script memerlukan pernyataan tipe data tertentu yang eksplisit dari suatu variabel, seperti number, string, dan lain-lain. Flash mengenali tipe data suatu variabel pada saat variabel tersebut dinyatakan.

```
x = 3;
```

Pada ekspresi `x = 3`, Flash mengevaluasi elemen di sebelah kanan operator dan mengenali bahwa elemen tersebut mempunyai tipe data number. Pernyataan berikutnya mungkin mengubah tipe data dari `x`; contohnya, `x = "hello"` menyebabkan tipe data dari `x` menjadi string. Variabel yang belum ditentukan nilainya dituliskan dengan `undefined`.

ActionScript mengkonversi tipe data secara otomatis bila ekspresi memerlukannya. Contohnya, pada saat suatu nilai diberikan pada action trace, secara otomatis trace mengkonversi nilai ke dalam string dan menampilkannya ke window

Output. Dalam ekspresi yang menggunakan operator, tipe data dapat dikonversi bila diperlukan. Contohnya, pada saat menggunakan string, operator `+` menghendaki peran yang lain juga string.

```
"Salemba nomor " + 99
```

ActionScript mengkonversi bilangan 12 ke dalam string "12" dan menambahkan pada akhir string sehingga hasilnya seperti berikut:

```
"Salemba nomor 99"
```

Pada saat melakukan debug scripts, tipe data akan dikenali dari ekspresi atau variabelnya. Hal ini dapat dilakukan dengan menggunakan operator `typeof`, seperti contoh:

```
trace(typeof(namaKaryawan));
```

Konversi tipe data string ke dalam nilai numerik, dapat menggunakan fungsi `Number`. Sebaliknya, nilai numerik dapat dikonversi ke dalam string dengan fungsi `String`.

### 1.5.3 Jangkauan variabel

Variabel dalam ActionScript dapat bersifat global atau lokal. Variabel global digunakan bersama seluruh Timelines, sedangkan variabel local (lokal) hanya disediakan untuk satu blok dari code (antara kurung kurawal).

Pernyataan `var` dapat digunakan untuk mendeklarasikan variabel lokal dalam suatu script. Contohnya, variabel `i` dan `j` sering digunakan dalam perulangan. Pada contoh berikut `i` hanya digunakan sebagai variabel lokal; variabel tersebut hanya digunakan dalam fungsi `hitungKoordinat`.

```
function hitungKoordinat() {
    var i
    for ( i=0; i<steps; i++) {
        x = 1.0 * x + x_increment;
        y = 1.0 * y + y_increment;
        x_pik = Math.round(x);
        y_pik = Math.round(y);
        trace ( x_pik + " " + y_pik);
    }
}
```

Variabel lokal juga dapat digunakan untuk menghindari tabrakan (collision) karena kesalahan dalam movie. Dalam suatu masalah, suatu nama untuk variabel lokal dapat digunakan untuk menyimpan nama pengguna dan nama instance movie

clip lain. Hal ini disebabkan oleh perbedaan jangkauan variabel, sehingga tidak akan terjadi tabrakan.

Suatu fungsi sebaiknya menggunakan variabel lokal, sehingga fungsi tersebut dapat dijalankan sebagai code yang independen. Variabel lokal hanya berubah oleh code dalam blok dimana variabel tersebut berada. Ekspresi dari fungsi yang menggunakan variabel global, menyebabkan suatu nilai di luar fungsi dapat berubah.

#### **1.5.4 Deklarasi variabel**

Untuk mendeklarasikan variabel global, digunakan action setVariables atau operator assignment (=). Untuk mendeklarasikan variabel lokal dapat digunakan pernyataan var dalam blok dari fungsi. Variabel lokal berlaku dalam blok fungsi dan tidak berlaku setelah akhir dari blok tersebut.

Action call juga menciptakan variabel lokal yang mempunyai jangkauan dimana script dipanggil. Bila terdapat script lain yang kemudian dipanggil, jangkauan variabel tersebut hilang. Namun hal ini tidak disarankan, karena action call dihapus oleh action with yang lebih kompatibel dengan dot syntax.

Nilai variabel dapat diperiksa menggunakan action trace untuk mengirimkan nilai pada window Output. Dalam mode test-movie, trace ( x\_pik + y\_pik); mengirimkan nilai variabel x\_pik dan y\_pik ke window Output. Nilai variabel juga dapat diperiksa dan diatur pada Debugger dalam mode test-movie.

#### **1.5.5 Menggunakan variabel dalam suatu script**

Variabel harus dideklarasikan lebih dulu dalam script sebelum digunakan dalam ekspresi. Dalam contoh berikut, nilai variabel yang tidak dikenali dalam script akan menghasilkan suatu kesalahan.

```
getURL(myWebSite);
namaWebSite = "http://www.shrimpmeat.net";
```

Variabel namaWebSite harus dideklarasikan lebih dulu sehingga variabel dalam action getUrl dapat diberikan nilai. Nilai variabel dapat diubah berkali-kali dalam script. Pengaruh perubahan sangat dipengaruhi oleh tipe data yang disimpan oleh variabel. Tipe data primitif seperti string dan number dapat menerima setiap nilai. Misalnya, x diberi nilai 15, dan nilainya disalin ke dalam y. Bila x berubah menjadi 30, nilai y tetap 15.

```
var x = 15;
var y = x;
var x = 30;
```

Contoh lain, variabel in berisi nilai primitif 4, sehingga nilai tersebut dapat dilewatkan oleh fungsi sqrt akan mengembalikan nilai 2.

```
function sqrt(x) {  
    return x * x;  
}  
  
var bilangan = 4;  
var hasil = sqrt(bilangan);
```

Tipe data object dapat berisi informasi yang luas dan kompleks, karena variabel dengan tipe data ini tidak mempunyai nilai aktual. Variabel ini mempunyai hubungan dengan suatu nilai, seperti alias yang menunjuk isi dari suatu variabel. Contoh berikut memperlihatkan nilai dengan melewati referensi.

```
var arrayNama = ["Andi", "Susi", "Tomas"];  
var arrayKaryawan = arrayNama;  
arrayNama[1] = "Maria";  
trace(arrayKaryawan);
```

Code di atas menyatakan bahwa objek array arrayNama mempunyai tiga elemen. Variabel arrayKaryawan diciptakan dan dilewatkan secara referensi dengan arrayNama. Pada saat elemen kedua dari arrayNama berubah, mengakibatkan perubahan pada semua variabel yang mempunyai referensi terhadapnya. Action trace akan menampilkan ["Andi", "Maria", "Tomas"] pada window Output.

Pada contoh berikutnya, arrayBilangan berisi sebuah objek array, kemudian dilewatkan sebagai referensi dengan fungsi zeroArray, dimana fungsi tersebut mengubah isi dari array dalam arrayBilangan.

Fungsi zeroArray menerima objek array sebagai argumen dan memberi nilai semua elemen 0. Hal ini dapat mengubah array karena array dilewatkan sebagai referensi.

```
function zeroArray (array) {  
    var i;  
    for (i=0; i < array.length; i++) {  
        array[i] = 0;  
    }  
}  
  
var arrayBilangan = new Array();  
arrayBilangan[0] = 1;  
arrayBilangan[1] = 2;  
arrayBilangan[2] = 3;  
  
var out = zeroArray(arrayBilangan)
```

Referensi pada semua objek selain movie clip disebut hard reference karena bila suatu objek dihubungkan dengan referensi, objek tersebut tidak dapat dihilangkan. Referensi pada movie clip adalah referensi khusus yang disebut soft reference. Soft reference tidak memaksakan object untuk ada. Bila movie clip dihilangkan dengan suatu action seperti removeMovieClip, setiap referensi kepadanya tidak akan bekerja.

## 1.6 Operator

Operator merupakan simbol yang biasa dilibatkan dalam program untuk untuk melakukan operasi atau manipulasi. Operator dapat menggabungkan, membandingkan atau mengubah nilai suatu ekspresi. Elemen dimana operator melakukan performansi disebut operan. Operator terdiri dari beberapa macam dan mempunyai karakteristik serta penggunaan yang berbeda-beda. Contohnya, operator + menambahkan nilai literal numerik pada nilai variable bilangan; sedangkan bilangan dan 5 adalah operan.

bilangan + 5

### 1.6.1 Operator precedence

Bila dua atau lebih operator digunakan pada suatu pernyataan, beberapa operator mendapatkan prioritas lebih tinggi dari lainnya. ActionScript mengikuti hirarki yang pasti untuk menjelaskan operator yang mana lebih mendapatkan prioritas pertama di banding operator lain.

Contohnya, perkalian selalu dilakukan lebih dulu dari penjumlahan, namun tanda kurung menyebabkan prioritas diberikan kepada operator yang berada di dalam kurung.

jumlah= 5 + 4 \* 2;

Hasil perhitungan 13.

jumlah = (5 + 4) \* 2;

Hasil perhitungan 18.

Untuk mengetahui urutan prioritas operator, dapat dilihat pada daftar sebagai berikut:

<b>Operator</b>	<b>Deskripsi</b>	<b>Asosiatif</b>
+	Unary plus	Kanan ke kiri
-	Unary minus	Kanan ke kiri
~	Bitwise one's complement	Kanan ke kiri
!	Logical NOT	Kanan ke kiri
not	Logical NOT (Flash 4 style)	Kanan ke kiri
++	Post-increment	Kiri ke kanan
--	Post-decrement	Kiri ke kanan

( )	Function call	Kiri ke kanan
[ ]	Elemen array	Kiri ke kanan
.	Structure member	Kiri ke kanan
++	Pre-increment	Kanan ke kiri
—	Pre-decrement	Kanan ke kiri
new	Mengalokasikan object	Kanan ke kiri
delete	Menghapus object	Kanan ke kiri
typeof	Type of object	Kanan ke kiri
void	Returns undefined value	Kanan ke kiri
*	Kali	Kiri ke kanan
/	Bagi	Kiri ke kanan
%	Modulo	Kiri ke kanan
+	Tambah	Kiri ke kanan
add	String concatenation ( & )	Kiri ke kanan
-	Subtract	Kiri ke kanan
<<	Bitwise Left Shift	Kiri ke kanan
>>	Bitwise Right Shift	Kiri ke kanan
>>>	Bitwise Right Shift (Unsigned)	Kiri ke kanan
<	Lebih kecil	Kiri ke kanan
<=	Lebih kecil atau sama dengan	Kiri ke kanan
>	Lebih besar	Kiri ke kanan
>=	Lebih besar atau sama dengan	Kiri ke kanan
lt	Lebih kecil (versi string)	Kiri ke kanan
le	Lebih kecil atau sama (versi string )	Kiri ke kanan
gt	Lebih besar (versi string)	Kiri ke kanan
ge	Lebih besar atau sama (versi string)	Kiri ke kanan
==	Sama dengan	Kiri ke kanan
!=	Tidak sama dengan	Kiri ke kanan
eq	Sama dengan (versi string)	Kiri ke kanan
ne	Tidak sama dengan (versi string)	Kiri ke kanan
&	Bitwise AND	Kiri ke kanan
^	Bitwise XOR	Kiri ke kanan
	Bitwise OR	Kiri ke kanan
&&	Logical AND	Kiri ke kanan
and	Logical AND (Flash 4)	Kiri ke kanan
	Logical OR	Kiri ke kanan
or	Logical OR (Flash 4)	Kiri ke kanan
?:	Conditional	Kanan ke kiri
=	Assignment	Kanan ke kiri
"*=", "/=", "%=", "+=", "-=", "&=", " =", "^=", "<<=", ">>=", ">>>="	Compound assignment	Kanan ke kiri
,	Evaluasi multiple	Kiri ke kanan

### 1.6.2 Operator asosiatif

Bila dua atau lebih operator bersama-sama mempunyai prioritas sama, asosiatif menentukan bahwa urutan dilakukan menurut performasinya. Asosiatif dapat terjadi dari kiri ke kanan atau dari kanan ke kiri.

Contohnya, perkalian mempunyai asosiatif kiri ke kanan, sehingga kedua pernyataan berikut ekuivalen. Pernyataan pertama dan kedua mempunyai hasil perhitungan yang sama, yaitu 24.

```
hasil = 2 * 3 * 4;
hasil = (2 * 3) * 4;
```

### 1.6.3 Operator numerik

Operator numerik (numeric operator) seperti tambah, kurang, kali, bagi, menjalankan operasi aritmatika. Tanda kurung dan minus juga termasuk dalam operator aritmatika. Operator numerik seperti pada daftar berikut:

<b>Operator</b>	<b>Keterangan</b>
+	Tambah
*	Perkalian
/	Pembagian
%	Modulo
-	Kurang
++	Penambahan
—	Pengurangan

### 1.6.4 Operator pembandingan

Operator pembandingan (comparison operator) membandingkan nilai pada suatu ekspresi dan mengembalikan nilai Boolean. Operator ini biasa digunakan pada perulangan dan pernyataan kondisi. Pada contoh berikut, bila variabel score adalah 100, maka movie tertentu dijalankan. Namun, Bila kondisi yang diperiksa ternyata tidak memenuhi syarat, maka movie lain yang dijalankan.

```
if nilai == 100) {
    loadMovie("maya.swf", 2);
}
else {
    loadMovie("bali.swf", 2);
}
```

Operator pembanding seperti pada daftar berikut:

<b>Operator</b>	<b>Keterangan</b>
<	Lebih kecil
>	Lebih besar d
<=	Lebih kecil atau sama dengan
>=	Lebih besar atau sama dengan

### **1.6.5 Operator string**

Operator + mempunyai akibat khusus bila digunakan untuk operasi pada string, yaitu menggabungkan dua operan string, seperti contoh berikut.

```

"Selamat Ulang Tahun," to "Maria!":
"Selamat Ulang Tahun, " + "Maria!"
    
```

Hasil dari operasi tersebut adalah "Selamat Ulang Tahun, Maria!" Bila hanya ada satu operator string +, Flash mengkonversi operan menjadi string.

Operator pembanding >, >=, <, and <= juga mempunyai akibat khusus bila digunakan untuk operasi pada string. Operator ini membandingkan yang mana mendahului dalam urutan alfabet. Operator pembanding hanya membandingkan string bila kedua operan adalah string. Bila hanya satu operan string, ActionScript mengkonversi kedua operan menjadi number dan membandingkan sebagai pembanding numerik.

### **1.6.6 Operator logika**

Operator logika membandingkan nilai Boolean (true and false) dan mengembalikan nilai Boolean ketiga. Contohnya, bila kedua operan mempunyai nilai true maka operator logika AND (&&) mengembalikan nilai true. Bila satu atau kedua operan mempunyai nilai true, operator logika OR (||) mengembalikan nilai true. Operator logika sering digunakan pada konjungsi dengan pembanding operator untuk mengetahui kondisi dari action IF. Pada contoh script berikut, bila kedua ekspresi true action IF akan mengeksekusi pernyataan yang ada:

```

if ((ikan_mujair >=10) && (ikan_lele >=25)) {
    hasil = "Anda sukses.";
}
    
```

Daftar operator logika, sebagai berikut:

<b>Operator</b>	<b>Keterangan</b>
&&	Logika AND
	Logika OR
!	Logika NOT

### 1.6.7 Operator bitwise

Operator bitwise memanipulasi angka floating point dan berubah menjadi integer 32 bit yang lebih mudah dikerjakan. Operator bitwise mengevaluasi 8 digit angka floating point terpisah untuk menghitung nilai baru.

Tabel berikut berisi daftar operator bitwise.

<b>Operator</b>	<b>Keterangan</b>
&	Bitwise And
	Bitwise Or
^	Bitwise Xor
~	Bitwise Not
<<	Shift left
>>	Shift right
>>>	Shift right zero fill

### 1.6.8 Operator kesamaan dan assignment

Operator kesamaan (==) dapat digunakan untuk mengenali apakah nilai dari dua operan identik. Perbandingan ini mengembalikan suatu nilai Boolean (true atau false). Fungsi operator berbeda untuk beberapa hal dapat dilihat dari operan yang bersangkutan sebagai berikut:

- Bila operan adalah string, number, atau nilai Boolean, operator membandingkan nilainya.
- Bila operan adalah objek atau array, operator membandingkan menurut referensi.

Penggunaan operator assignment:

- memberikan nilai pada suatu varabel  
password = "Sk8tEr";
- memberikan nilai pada banyak variabel dalam satu ekspresi  
a = b = c = d;
- membuat kombinasi dari operasi

Contohnya, pernyataan berikut adalah ekuivalen.

```
x += 15;
x = x + 15;

x *= 3;
x = x * 3;
```

Tabel berikut berisi operator kesamaan dan assignment:

<b>Operator</b>	<b>Keterangan</b>
==	Equality
!=	Inequality
=	Assignment
+=	Addition and assignment
-=	Subtraction and assignment
*=	Multiplication and assignment
%=	Modulo and assignment
/=	Division and assignment
<<=	Bitwise shift left and assignment
>>=	Bitwise shift right and assignment
>>>=	Shift right zero fill and assignment
^=	Bitwise Xor and assignment
=	Bitwise Or and assignment
&=	Bitwise And and assignment

### **1.6.9 Dot dan array akses operator**

Dot operator (.) dan array access operator ([ ]) dapat digunakan untuk mengakses property dari predefined atau custom object, termasuk movie clip. Pada dot operator nama object di sebelah kiri dan nama property atau variabel di sebelah kanan. Nama property atau variabel harus merupakan nama identifier, dan tidak boleh menggunakan string atau variabel yang dievaluasi ke dalam string.

```
tahun.bulan = "Februari";
tahun.bulan.minggu = "Senin";
```

Dot operator dan array operator mempunyai peranan yang sama, tetapi mempunyai perbedaan sebagai berikut:

- dot operator menggunakan identifier sebagai property
- array access operator untuk mengisi dan mengakses isi dari array

Contohnya, kedua pernyataan mengakses variabel kecepatan pada movie clip mobil.

```
mobil.kecepatan;
mobil["kecepatan"];
```

Array access operator dapat digunakan:

- mengambil nama instance dan variabel secara dinamis. Ekspresi di dalam operator [ ] dievaluasi dan hasil evaluasi digunakan sebagai nama variabel yang diambil dari nama movie clip.

```
nama["lingkaranMc" + i ]
```

- pada sebelah kiri pernyataan assignment. Hal ini mengikuti inisialisasi nama instance, variabel, and object secara dinamis.

```
nama[index] = "Andi";
```

- menyatakan array multidimensi

```
daftar[baris][kolom]
```

## 1.7 Penggunaan Action

Action adalah pernyataan atau perintah ActionScript. Beberapa action ditempatkan pada satu frame atau objek untuk menciptakan script. Action dapat independen satu dengan lainnya, seperti pernyataan berikut:

```
gotoAndPlay(2);
stop();
loadMove("logo.swf", 1);
```

Action bersarang (nested actions) dapat dibuat dengan menggunakan satu action dalam action lain, sehingga action akan saling mempengaruhi. Dalam contoh berikut, action `if` memerintahkan `gotoAndPlay` untuk dieksekusi.

```
if (nilai < 95.0) {
    gotoAndPlay(1);
    for (i=1; i<5; i++) {
        total = total + nilai;
    }
}
```

Action dapat menggerakkan playhead pada Timeline (`gotoAndPlay`), kontrol aliran script dengan perulangan (`do while`) atau kondisi logika (`if`), atau membuat fungsi dan variabel baru (`function`, `setVariable`). Tabel berikut berisi daftar semua action.

### **Action**

<code>break</code>	<code>evaluate</code>	<code>include print</code>	<code>stopDrag</code>
<code>call for</code>	<code>loadMovie</code>	<code>printAsBitmap</code>	<code>swapDepths</code>
<code>comment</code>	<code>for...in loadVariables</code>	<code>removeMovieClip</code>	<code>tellTarget</code>
<code>continue</code>	<code>fsCommand</code>	<code>nextFrame</code>	<code>return</code>
<code>toggleHighQuality</code>	<code>nextScene</code>	<code>delete</code>	<code>function on</code>
<code>setVariable</code>	<code>stopDrag</code>	<code>do...while</code>	<code>getURL</code>
<code>onClipEvent</code>	<code>setProperty</code>	<code>trace</code>	<code>duplicate</code>
<code>gotoAndPlay</code>	<code>play</code>	<code>startDrag</code>	<code>unloadMovie</code>
<code>gotoAndStop</code>	<code>else</code>	<code>if</code>	<code>prevFrame</code>
<code>stop</code>	<code>var</code>	<code>else if</code>	<code>ifFrameLoaded</code>
<code>prevScene</code>	<code>stopAllSounds</code>	<code>while</code>	

### 1.7.1 Menuliskan target path

Penggunaan action untuk mengontrol atau load, maka harus menuliskan spesifikasi nama dan alamat yang disebut target path. Action menggunakan satu atau lebih target path sebagai argumen.

loadMovie	loadVariables
unloadMovie	setProperty
startDrag	duplicateMovieClip
removeMovieClip	print
printAsBitmap	tellTarget

Contohnya, action loadMovie menggunakan argumen URL, Location, dan Variables. URL adalah lokasi dari movie yang akan di-load. Location adalah target path yang akan dimasukkan ke dalam movie yang dikehendaki.

```
loadMovie(URL, Lokasi, Variabel);
```

Pernyataan berikut mengambil URL `http://www.diamond.com/kuis.swf` ke dalam instance bar pada Timeline utama, `_root`, dimana `_root.bar` adalah target path;

```
loadMovie("http://www.diamond.com/kuis.swf", _root.bar);
```

Dalam ActionScript suatu movie clip dapat dikenali dari nama instance. Contohnya, pada pernyataan berikut, property `_alpha` dari movie clip elips diberi nilai dengan 20% visibility:

```
elipsMc._alpha = 20;
```



Gambar 1.4 Elips tanpa alpha dan dengan alpha 20%

### 1.7.2 Memberi nama suatu instance dari movie clip

1. Pilih movie clip pada stage.
2. Pilih Window —> Properties —> Properties
3. Masukkan nama instance pada field < Instance Name >.

### 1.7.3 Mengenali movie yang diambil

Dengan menggunakan `_levelX` dimana `X` adalah nomor level yang menjadi spesifikasi dalam action `loadMovie`.

Contohnya, suatu movie diambil ke level 5 mempunyai nama instance `_level5`. Dalam contoh berikut, suatu movie diambil ke level 5 dengan `visibility` diberikan nilai `false`.

```
onClipEvent(load) {
    loadMovie("maya.swf", 2);
}

onClipEvent(enterFrame) {
    _level2._visible = false;
}
```

### 1.7.4 Memasukkan target path dari suatu movie

Klik button `Insert Target Path` dalam panel `Action`, dan pilih movie clip dari daftar yang muncul. Script yang digunakan adalah sebagai berikut:

```
on (release) {
    tellTarget ("bee_instance") {
        gotoAndPlay (1);
    }
}
```

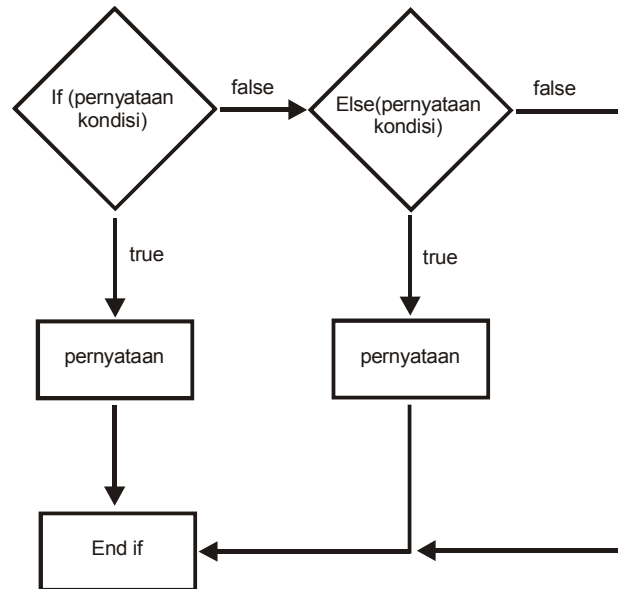
## 1.8 Mengontrol Aliran Script

ActionScript menggunakan action `if`, `for`, `while`, `do...while`, dan `for...in` untuk memperoleh performansi suatu action. Hasil dari evaluasi tersebut tergantung pada kondisi yang ada.

### 1.8.1 Penggunaan pernyataan "if"

Pernyataan yang memeriksa suatu keadaan mempunyai nilai `true` atau `false` dimulai dengan bentuk `if`. Bila kondisi dipenuhi, ActionScript melakukan eksekusi pernyataan yang mengikutinya. Bila kondisi tidak dipenuhi ActionScript melewatinya dan berpindah ke pernyataan berikut di luar blok dari code.

```
if (pernyataan kondisi) {
    pernyataan1;
}
else (pernyataan kondisi) {
    pernyataan2;
}
```



Gambar 1.5 Perulangan if dan if-else

Untuk menghasilkan performansi optimal dari code, periksa kondisi yang menyerupainya. Pernyataan berikut memeriksa beberapa kondisi dengan menggunakan bentuk `else if` yang memberikan hasil kemudian menjalankan pernyataan berikutnya. Bila kondisi dipenuhi, maka dilakukan action `loadMovie` dengan target “maya.swf”, yaitu suatu animasi dengan nama file “maya.swf”. Namun bila kondisi tidak dipenuhi, maka yang dijalankan adalah movie “europe.swf”.

```

if (nilai > 90.0) {
    loadMovie (“maya.swf”,2);
}
else {
    loadMovie (“europe.swf”,2)
}
  
```

### 1.8.2 Action perulangan

ActionScript dapat melakukan perulangan sebanyak tertentu atau selama kondisi tertentu berlaku. Penggunaan `while`, `do...while`, `for`, dan `for...in` pada action untuk menciptakan perulangan.

#### Pernyataan while

Perulangan `while` mengevaluasi suatu ekspresi dan mengeksekusi code dalam blok bila ekspresi tersebut `true`. Setelah semua pernyataan dalam blok tersebut dieksekusi, ekspresi dievaluasi lagi. Pada contoh berikut, perulangan dilakukan empat kali.

```

i = 0
while (i < 5) {
    x = x + x_increment;
    y = y + y_increment;
    i ++;
}

```

Pernyataan do...while dapat digunakan untuk perulangan yang sama. Pada perulangan do...while, ekspresi dievaluasi di bawah dari blok code, sehingga perulangan selalu dilakukan paling sedikit sekali, seperti contoh berikut.

```

i = 0
do {
    x = x + x_increment;
    y = y + y_increment;
    i++;
} while (i < 5);

```

### **Pernyataan for**

Hampir semua perulangan menggunakan counter yang digunakan untuk mengontrol berapa banyak perulangan dilakukan. Suatu variabel dapat dideklarasikan dan menuliskan pernyataan yang menambah atau mengurangi setiap kali perulangan dieksekusi. Contoh berikut menggambarkan penggunaan pernyataan for.

```

for ( i=0; i<5; i++) {
    x = x + x_increment;
    y = y + y_increment;
}

```

### **Pernyataan for..in**

Turunan termasuk movie clip lainnya, fungsi, object, dan variabel. Contoh berikut menggunakan trace untuk mencetak hasil pada window Output:

```

karyawan = { nama:'Andi', umur:25, alamat:'Jakarta' };
for (namaProperti in karyawan) {
    trace("Karyawan mempunyai data: " + namaProperti + ",
        nilai: " + karyawan[namaProperti]);
}

```

Bila dieksekusi maka hasilnya pada window Output:

```

Karyawan mempunyai data: nama, nilai: Andi
Karyawan mempunyai data: umur, nilai: 25
Karyawan mempunyai data: alamat, nilai: Jakarta

```

Kadang-kadang script diperlukan untuk membuat perulangan pada tipe tertentu dari turunan. Perulangan untuk turunan hanya terdapat pada movie clip.

Pernyataan `for...in` digunakan dalam konjungsi dengan operator `typeof`.

```
for (nama in karyawanMc) {
    if (typeof (karyawanMc[nama]) == "Andi") {
        trace("Nama karyawan adalah: " + nama);}
    }
}
```

## 1.9 Predefined function dan object

Fungsi adalah satu blok dari code yang bisa digunakan kembali di manapun dalam movie. Bila nilai spesifik atau argumen dilewatkan dalam suatu fungsi, fungsi akan dioperasikan dengan nilai tersebut. Fungsi juga dapat mengembalikan suatu nilai. Flash mempunyai predefined function yang dapat digunakan untuk mengakses dan menampilkan informasi tertentu. Contohnya, seperti pendeteksian kolisi (`hitTest`), mendapatkan nilai tombol yang ditekan terakhir (`keycode`), dan mendapatkan nomor versi dari Flash Player yang digunakan untuk menjalankan movie (`getVersion`).

### 1.9.1 Memanggil fungsi

Fungsi dapat dipanggil pada setiap Timeline, termasuk mengambil movie. Setiap fungsi mempunyai karakteristik dan keharusan tertentu untuk melewati suatu nilai. Bila argumen yang dilewatkan lebih banyak dari pada persyaratan yang ada pada fungsi, nilai lainnya akan diabaikan. Bila tidak dilewatkan persyaratan argumen, argumen yang kosong dinyatakan sebagai tipe data `undefined`, yang dapat menyebabkan kesalahan pada saat script diekspor. Untuk memanggil fungsi harus berada dalam frame yang terjangkau oleh `playhead`.

Daftar berikut berisi macam-macam predefined function:

Boolean	<code>getTimer</code>	<code>isFinite</code>	<code>newline</code>	<code>scroll</code>
<code>escape</code>	<code>getVersion</code>	<code>isNaN</code>	<code>number</code>	<code>String</code>
<code>eval</code>	<code>globalToLocal</code>	<code>keycode</code>	<code>parseFloat</code>	<code>targetPath</code>
<code>false</code>	<code>hitTest</code>	<code>localToGlobal</code>	<code>parseInt</code>	<code>true</code>
<code>getProperty</code>	<code>int</code>	<code>maxscroll</code>	<code>random</code>	<code>unescape</code>

### 1.9.2 Memanggil fungsi

Untuk memanggil fungsi dalam Expert Mode, dapat dilakukan menggunakan nama fungsi serta memberikan argumen yang diperlukan. Contoh berikut adalah

memanggil suatu fungsi tanpa argumen, yaitu fungsi `initialize` yang tidak mempunyai argumen.

```
initialize();
```

### **1.9.3 Memanggil fungsi dalam Normal Mode**

Untuk memanggil fungsi dalam Normal Mode, dapat dilakukan menggunakan nama fungsi serta memberikan argumen yang diperlukan dalam field Expression. Untuk memanggil fungsi pada Timeline lain dapat digunakan target path.

## **1.10 Fungsi Custom**

Fungsi custom (custom function) adalah fungsi yang dibuat untuk keperluan tertentu. Fungsi dapat ditentukan untuk mengeksekusi serangkaian pernyataan dan melewati nilai. Fungsi juga dapat mengembalikan nilai. Sekali fungsi ditentukan, fungsi tersebut dapat dipanggil dari setiap Timeline.

Suatu fungsi dapat merupakan “black box” pada saat fungsi dipanggil, dilengkapi dengan input (argumen). Hal ini menampilkan beberapa operasi dan menghasilkan nilai output (nilai yang dikembalikan). Fungsi yang ditulis dengan baik dan sangat teliti menempatkan komentar tentang input, output dan kegunaan dari pernyataan.

### **1.10.1 Mendefinisikan fungsi**

Seperti halnya variabel, fungsi dihubungkan dengan movie clip yang ditentukan dalam fungsi tersebut. Pada saat suatu fungsi didefinisikan kembali, fungsi baru menggantikan definisi fungsi yang lama.

Untuk mendefinisikan suatu fungsi, gunakan action function yang diikuti dengan nama dari fungsi, argumen yang digunakan untuk melewati fungsi, dan pernyataan yang mengindikasikan apa yang dikerjakan fungsi tersebut. Fungsi berikut disebut Lingkaran dengan argumen radius:

```
function Lingkaran(radius) {
    this.radius = radius;
    this.luas = Math.PI * radius * radius;
}
```

Keyword `this` digunakan pada blok fungsi yang memiliki referensi ke movie clip.

### **1.10.2 Melewatkan argumen pada fungsi**

Argumen adalah elemen dimana fungsi mengeksekusi code. (Dalam buku ini pengertian argumen dan parameter adalah sama). Contohnya, fungsi berikut mengambil argumen kode dan nilai Akhir:

```
function tampilkanHasil(kode, nilaiAkhir) {
    hasil.tampilan = kode;
    hasil.nilai = nilaiAkhir;
}
```

Bila satu fungsi dipanggil, argumen yang diperlukan harus dilewatkan pada fungsi tersebut. Contohnya, scorecard adalah nama instans dari movie clip; mempunyai nilai yaitu field input dari instance. Fungsi berikut memanggil variabel dan menampilkan nilai kode “ABC” dan variabel nilai diberi nilai 4000.

```
tampilkanHasil(“ABC”, 4000)
```

Argumen kode pada fungsi tampilkanHasil serupa dengan variabel lokal yang keberadaannya ditentukan oleh pemanggilan fungsi. Bila argumen itu dihilangkan selama fungsi dipanggil, argumen dinyatakan sebagai undefined.

### 1.10.3 Menggunakan variabel lokal dalam fungsi

Variabel lokal bermanfaat untuk mengorganisasi dan membuat lebih mudah memahami code. Bila fungsi menggunakan variabel lokal, variabel tersebut terlindungi dari script di luar fungsi. Variabel lokal akan musnah bila fungsi yang berkaitan tidak ada. Setiap argumen yang dilewatkan pada fungsi juga diperlakukan seperti variabel lokal.

### 1.10.4 Mengembalikan nilai dari fungsi

Action return digunakan untuk mengembalikan nilai dari action. Action return menghentikan fungsi dan menggantikan dengan nilai yang diperoleh dari action return. Bila tidak ada action return sebelum akhir suatu fungsi, string kosong menjadi nilai kembali yang diperolehnya. Contohnya, fungsi berikut mengembalikan kuadrat dari argumen x.

```
function sqr(x) {
    return x * x;
}
```

Beberapa fungsi menampilkan sejumlah permintaan, tanpa mengembalikan nilai. Contohnya, fungsi berikut menginisialisasi beberapa variabel global.

```
function inialisasi() {
    lingkaran_x = _root.lingkaran._x;
    lingkaran_y = _root.lingkaran._y;
    poligon_x = _root.poligon._x;
    poligon_y = _root.poligon._y;
}
```

### 1.10.5 Memanggil fungsi

Untuk meminta suatu fungsi dapat dilakukan dengan menggunakan Action Panel pada Normal Mode. Fungsi dapat dipanggil dari setiap Timeline, termasuk mengambil movie. Contohnya, pernyataan ini meminta fungsi sqrt pada movieClip MathLib.

```
var temp = _root.MathLib.sqrt(3);
```

## 1.11 Menggunakan Predefined Object

Predefined object digunakan untuk mengakses informasi tertentu. Pada umumnya predefined object mempunyai method (fungsi pada suatu object) yang dapat mengembalikan suatu nilai atau menampilkan action. Contohnya object date memberikan informasi dari sistem clock dan object Sound dapat digunakan untuk mengontrol elemen suara dalam movie.

Daftar berikut adalah predefined object pada Flash:

Array	Boolean
Color	Date
Key	Math
MovieClip	Number
Object	Selection
Sound	String
XML	XMLSocket

Beberapa predefined object mempunyai property yang nilainya dapat dibaca. Contohnya object Key mempunyai nilai konstanta yang mewakili tombol pada keyboard. Setiap object mempunyai karakteristik dan kemampuan yang dapat digunakan pada movie.

Instance movie clip adalah object dalam Flash. Method dari predefined movie clip dapat dipanggil hanya bila akan memanggil method dari object lain.

### 1.11.1 Membuat object

Object dapat dibuat dengan dua macam cara, yaitu menggunakan:

- operator new - untuk membuat object predefined object class atau custom predefined object class
- inisialisasi object ( { } ) - untuk membuat object dari tipe generik object

Untuk menggunakan operator new dalam membuat object, diperlukan fungsi sederhana, yaitu fungsi constructor. Object yang baru memiliki semua property dan method yang sama dengan object tersebut. Hal ini seperti melakukan drag suatu object

dari Library ke dalam stage. Contohnya, pernyataan berikut menggambarkan inisialisasi object Date. Beberapa predefined object dapat diakses tanpa menginisialisasi terlebih dahulu. Contohnya, pernyataan yang memanggil method abs dari object Math.

```
musik = new Sound();  
Math.abs();
```

Setiap object memerlukan fungsi constructor yang berhubungan elemen dalam kotak panel ActionScript. Contohnya, new Sound, new Color, new Date, new String dan lain-lain.

### **1.11.2 Mengakses property dari object**

Dot (.) operator dapat digunakan untuk mengakses nilai property dari suatu object. Nama object diletakkan di sebelah kiri dari dot, dan nama property diletakkan di sebelah kanan. Contohnya, pada pernyataan berikut, nama object adalah karyawanObject dan nama adalah property.

```
karyawanObject.nama
```

Untuk memberikan nilai property pada Normal Mode menggunakan action setVariable. Nilai property dapat berubah bila diberikan nilai baru.

```
karyawanObjewct.nama = "Andi";  
karyawanObjewct.nama = "Maria";
```

Memanggil method dari object

Method object dapat dipanggil menggunakan dot operator yang diikuti dengan method. Contohnya, pernyataan berikut memanggil method setVolume dari object Sound.

```
musik = new Sound(this);  
musik.setVolume(50);
```

Menggunakan object movie clip

Method dari predefined object MovieClip dapat digunakan untuk mengontrol instance movie clip pada stage. Contoh berikut menggambarkan perintah kepada instance untuk menjalankan hitungMc.

```
hitungMc.play();
```

Menggunakan object Array

Object Array dapat digunakan untuk mengakses dan memanipulasi data array. Array adalah object yang mempunyai properti menjelaskan posisi dalam array. Semua array berbasis 0, yang berarti bahwa elemen pertama adalah [0], elemen kedua [1] dan selanjutnya. Contoh berikut myArray berisi nama bulan dalam setahun. Untuk mengakses elemen ke-2 dari array, digunakan ekspresi arrayBulan[1].

```
arrayBulan[0] = "Januari"
arrayBulan[1] = "Februari"
arrayBulan[2] = "Maret"
arrayBulan[3] = "April"
```

Object array mempunyai predefined property length, yaitu nilai dari jumlah elemen dalam array. Pada saat elemen object array diberikan nilai integer positif seperti  $index \geq length$ , length secara otomatis diubah menjadi  $index + 1$ .

## 1.12 Menggunakan Custom Object

Custom object dapat dibuat untuk mengorganisasi informasi dalam script sehingga lebih modau untuk disimpan dan diakses oleh property dan method dari object. Setelah membuat master object atau "class", maka dapat secara langsung dibuat object dalam movie.

Suatu object adalah tipe data yang kompleks berisi beberapa atau tidak satupun properti. Setiap property, seperti variabel, mempunyai nama dan nilai. Nilai dapat berupa string, number, Boolean, object, movie clip atau undefined. Berikut adalah contoh berbagai tipe data dari property.

```
pelanggan.nama = "Linda";
pelanggan.umur = 30;
```

### 1.12.1 Membuat object

Operator new dapat digunakan untuk membuat object dari fungsi constructor. Fungsi constructor memberikan nama yang sama pada tipe object yang dibuatnya. Constructor yang membuat object hitung akan bernama Hitung. Contoh berikut menggambarkan pembuatan object dari fungsi constructorFunction.

```
new constructorFunction (argument1, argument2, ... argumentN);
```

Pada saat constructorFunction dipanggil, Flash melewati hidden argumen this, yaitu argumen yang menghubungkan dengan object constructorFunction. Pada saat constructor dibuat, maka hidden argumen this akan menghubungkan object yang dibuat oleh constructor tersebut. Contoh berikut memperlihatkan fungsi constructor membuat object lingkaran.

```
function Lingkaran(radius) {
    this.radius = radius;
    this.luas = Math.PI * radius * radius;
}
```

Constructor biasanya digunakan untuk mengisi method dari suatu object.

```
function Luas() {
    this.luasLingkaran = Math.PI * radius * radius;
}
```

Untuk menggunakan object dalam script, harus dinyatakan dulu ke dalam variabel. Object lingkaran dengan radius 5 dibuat dengan operator new untuk membuat object dan memasukkannya ke dalam variabel Lingkaran.

```
var Lingkaran = new Circle(5);
```

### 1.12.2 Membuat inheritance

Bila menggunakan fungsi constructor untuk membuat object, semua property dan method yang dimiliki oleh prototype dari constructor diturunkan kepada object sebagai `_proto_property`. Langkah berikut menggambarkan cara menentukan method yang sederhana.

1. Membuat fungsi constructor Lingkaran:

```
function Lingkaran(radius) {
    this.radius = radius;
}
```

2. Membuat method luas dari object Lingkaran:

```
Lingkaran.prototype.luas = function () {
    return Math.PI * this.radius * this.radius;
}
```

3. Membuat instance dari object Lingkaran:

```
var Ling = new Circle(4);
```

4. Memanggil method area dari object baru myCircle:

```
var lingLuas = Ling.luas()
```

Suatu method untuk suatu object dibuat dengan memasukkannya pada instance masing-masing.

```
function Lingkaran(radius) {
    this.radius = radius;
    this.luas= function() {
        return Math.PI * this.radius * this.radius;
    }
}
```

### 1.13 Latihan

1. Tuliskan sebuah program yang dapat menghasilkan output sebagai berikut:

```
Masukkan angka : 300
Angka tersebut lebih besar dari 100
Masukkan angka : 3
Angka tersebut tidak lebih besar dari pada 100
```

2. Buatlah program untuk menjumlahkan waktu1 dan waktu2 di mana nilai variabel tersebut dimasukkan melalui keyboard, terdiri dari jam, menit dan detik seperti contoh:

3        7        8

Bila angka detik menunjukkan 60, maka akan menambahkan nilai menit, demikian juga dengan nilai menit 60 akan menambahkan nilai jam. Bila program dijalankan menghasilkan output sebagai berikut:

Masukkan waktu 1	:	1	6	56
Masukkan waktu 2	:	2	17	30
Jumlah	:	3 jam, 24 menit, 26 detik		

3. Tuliskan sebuah program yang dapat mengurutkan bilangan 30, 78, 3, 98 dan 56! Buat dengan predefined object yang dimiliki oleh Flash dan algoritma pengurutan seperti selection sort, insertion sort, bubble sort dan quick sort.
4. Bila diketahui nilai1 = 100 dan nilai2 = 56, berapakah hasil operasi berikut:

- a.  $(\text{nilai1} \&\& \text{nilai2}) \wedge (\text{!(nilai1} \ll 1))$
- b.  $((\text{!(nilai1} \ll \text{nilai1}) \& (\text{nilai1} \wedge \text{nilai2})) \gg 1$

5. ISBN (International Standard Book Number) adalah penomoran buku dengan standard internasional. Contohnya, buku Komputer dengan nomor

ISBN. 979-20-1345-8  
 979 = Kode Negara  
 20 = Kode Penerbit  
 1345 = Nomor Buku  
 8 = Check Digit

Check digit diperoleh menggunakan perhitungan sebagai berikut:  
 $(9*10 + 7*9 + 9*8 + 2*7 + 0*6 + 1*5 + 3*4 + 4*3 + 5*2 + \text{Check Digit})$   
 habis dibagi 11.

Buatlah program yang dapat digunakan oleh semua penerbit di Indonesia untuk menghitung nomor ISBN buku yang diterbitkannya. Untuk menghitung nomor

# 2

## Operasi Matematika dan Logika

Aplikasi Flash diperlukan untuk penyajian informasi interaktif, misalnya pengunjung dapat memasukkan input dan dan mendapatkan hasil setelah input tersebut dievaluasi. Hasil yang diperoleh dapat merupakan numerik, string atau movie. Kebutuhan situs web untuk menyajikan informasi dapat dibuat lebih menarik dengan menggunakan unsur-unsur multimedia.

Pemahaman ActionScript dengan menggunakan Normal Mode maupun Expert Mode diperlukan untuk pembuatan aplikasi lebih lanjut. Dalam bab ini dibahas pembuatan beberapa macam aplikasi interaktif, seperti mengubah warna suatu object, membuat scrolling teks, mendeteksi tabrakan, membuat puzzle, dan lain-lain.

### 2.1 Tipe Data Primitif

Tipe data primitif yaitu string, number, dan Boolean. Teks dan karakter termasuk dalam tipe data string, sedangkan tipe data float dan integer termasuk dalam tipe data number. Boolean adalah tipe data yang mempunyai nilai true (1) atau false (0).

#### 2.1.1 Operator tambah (+)

Operator tambah (+) dapat digunakan untuk operan yang memiliki tipe data numerik maupun string.

Sintaks:  
ekspresi1 + ekspresi2

Argumen ekspresi1 dan ekspresi2 dapat berupa integer, number, floating-point number atau string. Bila salah satu ekspresi string, semua ekspresi dikonversi menjadi

string dan digabungkan. Bila kedua ekspresi integer, hasilnya merupakan integer, tetapi bila satu atau keduanya adalah float, maka hasilnya float.

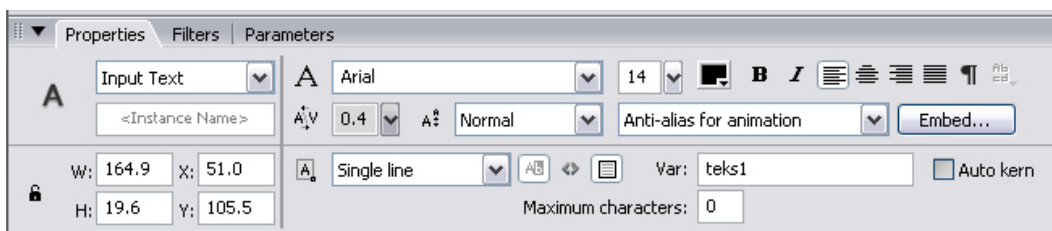
Pernyataan berikut merupakan contoh menggabungkan dua buah string dan menampilkan hasilnya “hari ini ulang tahun saya”.

“hari ini ” + “ulang tahun saya”

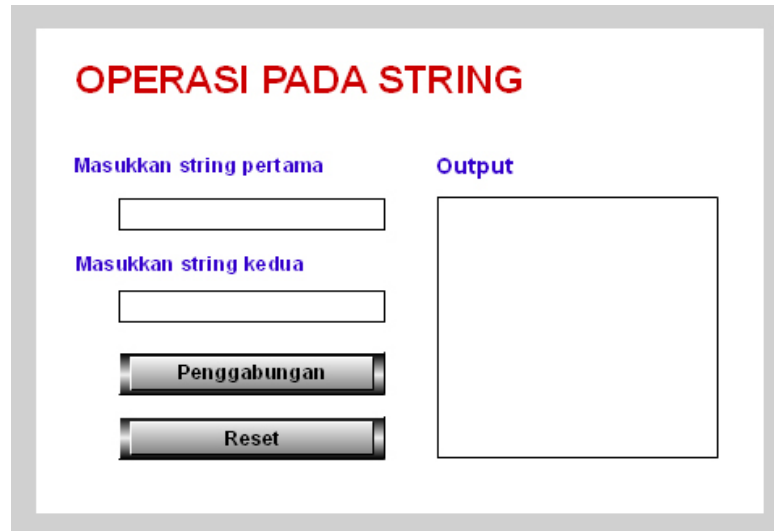
## String

Untuk menggabungkan string:

1. Pilih menu File —> New untuk membuat file baru.
2. Buat kotak dengan Text Tool pada stage, dilakukan dengan cara memilih Window —> Properties —> Properties, maka muncul kotak Text Options pada window Properties.
3. Pilih Input Text untuk jenis teks, dan Single Line untuk jumlah baris.
4. Untuk field Variable masukkan “teks1”.
5. Berikan tanda pilihan pada kotak Border/Bg.
6. Buat kotak dengan Text Tool untuk variabel “teks2”, seperti dilakukan pada pembuatan variabel “teks1”.
7. Berikan keterangan di atas kedua kotak tersebut dengan membuat static text “Masukkan string pertama” dan “Masukkan string kedua”.
8. Buat kotak dengan Text Tool untuk menempatkan output. Pada Text Options, pilih Dynamic Text dengan memasukkan nama variabel “hasil”. Berikan keterangan “output” untuk kotak tersebut.
9. Buat judul dari movie “OPERASI PADA STRING”.
10. Buat dua buah button dengan mengambil dari Common Library. Tuliskan teks “Penggabungan” dan “Reset” untuk button tersebut.



Gambar 2.1 Teks options pada window Properties



Gambar 2.2 Tampilan operasi string pada stage

11. Klik pada button “Penggabungan”, dan pilih Window → Actions. Kemudian pilih Expert Mode, dan buat script sebagai berikut:

```
on(release) {
    hasil = teks1 + teks2;
}
```

12. Klik pada button “Reset”, dan pilih Window → Actions. Kemudian tuliskan script sebagai berikut:

```
on(release) {
    teks1 = “ ”;
    teks2 = “ ”;
    hasil = “ ”;
}
```

Pada script tertulis teks1 = “ ”, berarti nilai variabel teks1 adalah kosong, demikian juga yang lain.

13. Pilih Control → Test Movie untuk melihat hasilnya.
14. Simpan file tersebut dengan nama lat1001\_string fla.

## Numerik

Aplikasi penggunaan operator + pada numerik dapat dilakukan dengan cara yang sama seperti pada pembuatan operasi string. Pada aplikasi ini terdapat perbedaan pada script, yaitu menggunakan fungsi number. Pernyataan berikut menjumlahkan integer 2 dan 3, serta menghasilkan integer 5.

Sintaks:

```
Number(x) + Number(y)
number(2) + number(3)
```

Untuk menambahkan numerik:

1. Button “Penggabungan” pada operasi string diganti dengan “Tambah”
2. Klik pada button “Tambah”, dan pilih Window → Actions. Kemudian buat script pada editor Action sebagai berikut:

```
on(release) {
    hasil = number(teks2) + number(teks2);
}
```

3. Pilih Control → Test Movie untuk melihat hasilnya.
4. Simpan file tersebut dengan nama lat1002\_number fla.

## Konversi string ke float dan integer

Teks yang dimasukkan melalui keyboard ke dalam movie merupakan string. Fungsi parseFloat mengkonversi string ke dalam tipe data float, sedangkan parseInt mengkonversi string ke integer. Bila data bukan merupakan angka, parseFloat atau parseInt akan memberikan nilai NAN atau 0. Integer yang diawali dengan angka 0 dibaca sebagai bilangan oktal, sedangkan bila diawali dengan 0x dibaca sebagai bilangan heksadesimal. Spasi yang terdapat di tengah integer dianggap sebagai karakter.

Sintaks:

```
parseFloat(ekspresi);
parseInt(ekspresi);
```

Contoh berikut penggunaan parseFloat dan parseInt untuk mengevaluasi beberapa macam angka:

```
parseFloat("-2") hasilnya -2
parseFloat("2.5") hasilnya 2.5
parseFloat("3.5e6") hasilnya 3.5e6, or 3500000
parseFloat("mangga") hasilnya NAN
```

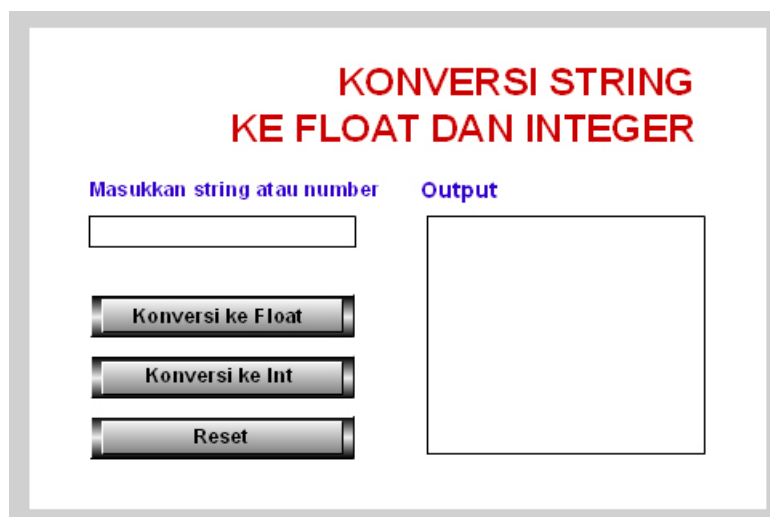
parseInt("3.5") hasilnya 3.5  
parseInt("bar") hasilnya NaN  
parseInt("4mangga") hasilnya 4

Fungsi newline digunakan untuk menyisipkan baris kosong (spasi) pada code ActionScript, sehingga informasi yang ditampilkan pada monitor kelihatan lebih baik dan mudah dibaca, seperti pernyataan berikut:

```
hasil = "Data yang dimasukkan:" + newline + data_masuk + newline + newline +  
"Konfersi ke float" + parseFloat(data_masuk);
```

Untuk menggunakan parseFloat, parseInt dan newline:

1. Pilih menu File → New untuk membuat file baru.
2. Buat kotak dengan Text Tool pada stage. Kemudian pilih Window → Properties → Properties, maka muncul kotak pilihan Text Options.
3. Pilih Input Text untuk jenis teks, dan Single Line untuk jumlah baris.
4. Untuk field Variable masukkan "data\_masuk".
5. Berikan tanda pilihan pada kotak Border/Bg.
6. Berikan keterangan di atas kotak tersebut dengan membuat static text "Masukkan string atau number".
7. Buat kotak dengan Text Tool untuk menempatkan output. Pada Text Options, pilih Dynsmic Text dengan memasukkan nama variabel "hasil". Berikan keterangan "output" untuk kotak tersebut.



Gambar 2.3 Konversi string ke float dan integer

8. Buat judul “KONVERSI STRING KE FLOAT DAN INTEGER”.
9. Buat tiga buah button dengan mengambil dari Common Library. Tuliskan teks “Konversi ke Float”, “Konversi ke Int” dan “Reset” sebagai keterangan button tersebut.
10. Klik pada button “Konversi ke Float”, dan pilih Window → Actions. Kemudian pilih Expert Mode, dan buat script sebagai berikut:
 

```
on(release) {
    hasil = "Data yang dimasukkan:" + newline + data_masuk + newline +
           newline + "Konversi ke float:" + parseFloat(data_masuk);
}
```
11. Klik pada button “Konversi ke Int”, dan pilih Window → Actions. Kemudian buat script pada Action list sebagai berikut:
 

```
on(release) {
    hasil = "Data yang dimasukkan:" + newline + data_masuk + newline +
           newline + "Konversi ke integer:" + parseInt(data_masuk);
}
```
12. Klik pada button “Reset”, dan pilih Window → Actions. Kemudian buat script pada Action list sebagai berikut:
 

```
on(release) {
    data_masuk = " ";
    hasil = " ";
}
```
13. Pilih Control → Test Movie untuk melihat hasilnya.
14. Simpan file tersebut dengan nama lat1003\_konversi string number fla.

### **2.1.2 Operasi matematik**

Method dari predefined object Math digunakan untuk memanipulasi bilangan. Flash mempunyai banyak predefined object Math, namun hanya beberapa yang dapat diberikan contoh pada buku ini, yaitu Math.pow, Math.abs dan Math.round.

#### **Pangkat**

Sintaks:  
Math.pow(x , y);

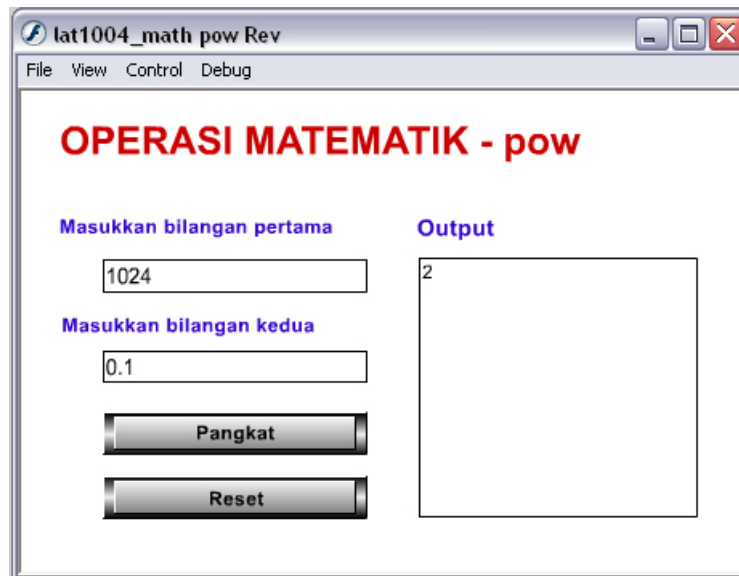
Pada sintaks di atas, x adalah angka yang dipangkatkan, sedangkan y adalah angka yang digunakan untuk memangkatkan x. Tipe data dari x dan y dapat berupa

float atau integer. Untuk dapat membuat perhitungan dengan predefined object Math.pow, dapat dilakukan langkah-langkah berikut.

1. Pilih menu File → New untuk membuat file baru.
2. Buat kotak dengan Text Tool pada stage. Kemudian pilih Window → Properties → Properties, maka muncul kotak dialog Text Options.
3. Pilih Input Text untuk jenis teks, dan Single Line untuk jumlah baris.
4. Untuk field Variable masukkan “x”.
5. Berikan tanda pilihan pada kotak Border/Bg.
6. Buat kotak dengan Text Tool untuk variabel “y”, seperti dilakukan pada pembuatan variabel “x”.
7. Berikan keterangan di atas kedua kotak tersebut dengan membuat static text “Masukkan bilangan pertama” dan “Masukkan bilangan kedua”.
8. Buat kotak dengan Text Tool untuk menempatkan output. Pada Text Options, pilih Dynamic Text dengan memasukkan nama variabel “hasil”. Berikan keterangan “output” untuk kotak tersebut.
9. Buat judul dari movie “OPERASI MATEMATIK - pow”.
10. Buat dua buah button dengan mengambil dari Common Library. Tuliskan teks “Pangkat” dan “Reset” untuk button tersebut.
11. Klik pada button “Pangkat”, dan pilih Window -----> Actions. Kemudian pilih Expert Mode, dan buat script pada Action list sebagai berikut:
 

```
on(release) {
    hasil = Math.pow(x,y);
}
```
12. Klik pada button “Reset”, dan pilih Window -----> Actions. Kemudian tuliskan script sebagai berikut:
 

```
on(release) {
    x = “ “;
    y = “ “;
    hasil = “ “;
}
```
13. Pilih Control → Test Movie untuk melihat hasilnya.
14. Simpan file tersebut dengan nama lat1004\_math pow fla.



Gambar 2.4 Penggunaan predifined object Math.pow, bila movie dijalankan

## Absolut dan pembulatan

Sintaks:

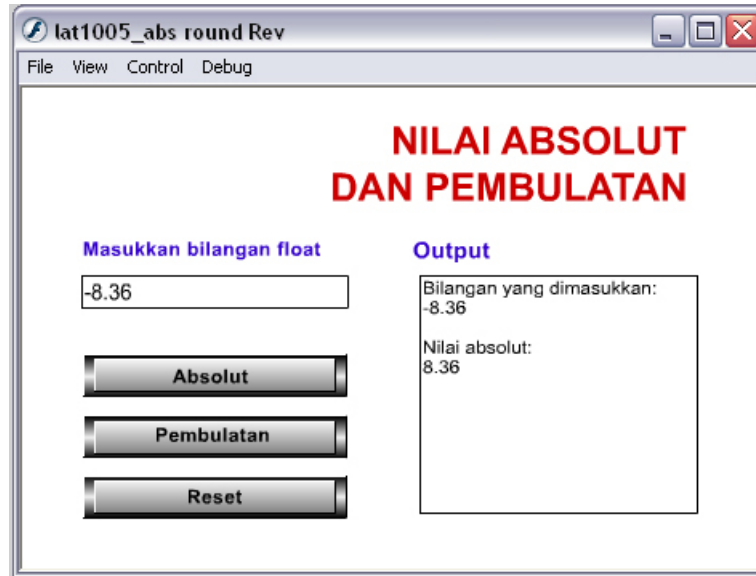
Math.abs(x);

Math.round(x);

Predefined object Math.abs. digunakan untuk mendapatkan nilai absolut suatu bilangan. Sedangkan untuk mendapatkan pembulatan ke atas atau ke bawah tergantung lebih dekatnya pada nilai integer, digunakan Math.round.

Untuk menggunakan Math.abs dan Math.round:

1. Pilih menu File → New untuk membuat file baru.
2. Buat kotak dengan Text Tool pada stage.. Kemudian pilih Window → Properties → Properties, maka muncul kotak dialog Text Options.
3. Pilih Input Text untuk jenis teks, dan Single Line untuk jumlah baris.
4. Untuk field Variable masukkan “x”.
5. Berikan tanda pilihan pada kotak Border/Bg.
6. Berikan keterangan di atas kotak tersebut dengan membuat static text “Masukkan bilangan float” .
7. Buat kotak dengan Text Tool untuk menempatkan output. Pada Text Options, pilih Dynamic Text dengan memasukkan nama variabel “hasil”. Berikan keterangan “output” untuk kotak tersebut.



Gambar 2.5 Nilai absolut dan pembulatan, bila movie dijalankan

8. Buat judul dari movie “NILAI ABSOLUT DAN PEMBULATAN”.
9. Buat tiga buah button dengan mengambil dari Common Library. Tuliskan teks “Absolut”, “Pembulatan” dan “Reset” untuk button tersebut.
10. Klik pada button “Absolut”, dan pilih Window → Actions. Kemudian buat script pada editor Action sebagai berikut:

```
on(release) {
    hasil = "Bilangan yang dimasukkan:" + newline + data_masuk + newline +
        newline + "Nilai absolut:" + Math.abs(x);
}
```

11. Klik pada button “Pembulatan”, pilih Window → Actions. Kemudian buat script pada editor Action sebagai berikut:

```
on(release) {
    hasil = "Bilangan yang dimasukkan:" + newline + data_masuk + newline +
        newline + "Nilai pembulatan:" + Math.round(x);
}
```

12. Klik pada button “Reset”, pilih Window → Actions. Kemudian tuliskan script sebagai berikut:

```
on(release) {
    x = " ";
    hasil = " ";
}
```

13. Pilih Control —> Test Movie untuk melihat hasilnya.
14. Simpan file tersebut dengan nama lat1005\_abs round fla.

### 2.1.3 Kode ASCII

Nilai ASCII (American Standard Code for Information Interchange) ditentukan untuk satu set yang terdiri dari 127 karakter. Fungsi `ord(character)` mengkonversi karakter menjadi nomor kode, sedangkan `chr(number)` mengkonversi nilai ASCII ke karakter yang bersangkutan.

Sintaks:

```
ord(character);
chr(number);
```

Untuk membuat konversi dari karakter - ASCII dan sebaliknya:

1. Pilih menu File —> New untuk membuat file baru.
2. Buat kotak dengan Text Tool pada stage. Kemudian pilih Window —> Properties —> Properties, maka muncul kotak dialog Text Options.
3. Pilih Input Text untuk jenis teks, dan Single Line untuk jumlah baris.
4. Untuk field Variable masukkan “karakter”.
5. Berikan tanda pilihan pada kotak Border/Bg.
6. Buat kotak untuk memasukkan teks lagi, yaitu untuk variabel “kode”.
7. Buat static text “Masukkan data”, “Karakter” dan “Kode” sebagai keterangan dari kedua kotak input teks tersebut.
8. Buat kotak dengan Text Tool untuk menempatkan output. Pada Text Options, pilih Dynamic Text dengan memasukkan nama variabel “hasil”. Berikan keterangan “output” untuk kotak tersebut.
9. Buat judul dari movie “KONVERSI KARAKTER - ASCII”.
10. Buat dua buah button dengan mengambil dari Common Library. Tuliskan teks “Konversi ke ASCII” dan “Konversi ke Karakter” untuk button tersebut.
11. Klik pada button “Konversi ke ASCII”, kemudian pilih Window —> Actions. Kemudian buat script pada editor Action sebagai berikut:

```
on(release) {
    hasil = "Karakter yang dimasukkan:" + newline + karakter + newline +
        newline + "Kode ASCII:" + ord(karakter);
}
```



Gambar 10.6. Konversi karakter - ASCII, bila movie dijalankan

12. Klik pada button “Konversi ke Karakter”, dan pilih Window -----> Actions. Kemudian pilih Expert Mode, dan buat script pada Action list sebagai berikut:

```
on(release) {
    hasil = “Bilangan yang dimasukkan:” + newline + kode + newline + newline +
        “Karakter:” + chr(kode);
}
```

13. Pilih Control —> Test Movie untuk melihat hasilnya.
14. Simpan file tersebut dengan nama lat1006\_ascii fla.

## 2.2 Latihan

1. Buatlah movie dengan button sebagai alat interaktifnya! Movie dengan animasi fish dibuat dengan mengambil Fish Movie Clip dari Common Libraries. Pada movie yang dibuat terdapat Variable “Harga”, dimana user dapat memasukkan input harga satuan, dan “Jumlah” untuk memasukkan jumlah ikan yang dibeli. Bila movie dijalankan:
  - Bila Total Harga lebih besar dari 200 atau Jumlah Ikan lebih besar dari 50, maka animasi ikan dapat dihentikan (stop) dan posisi dimanapun berada. Pada saat button Play ditekan, animasi ikan akan berjalan (play) kembali dimulai pada tempat awal dari animasi ikan tersebut.

- Bila Total Harga paling besar 25 atau Jumlah Ikan tidak lebih besar dari 10, maka animasi ikan dapat dihentikan (stop) dan posisi kembali pada tempat awal animasi. Pada saat button Play ditekan, animasi ikan tidak dapat akan berjalan lagi.
  - Bila kedua kondisi tersebut tidak dipenuhi, maka penekanan pada button tidak berakibat apapun terhadap movie clip.
2. Pada aplikasi real estat, pengguna dapat memilih tipe rumah yang dapat dilihat pada animasi! Setelah memilih salah satu tipe tersebut, kemudian memasukkan jumlah uang muka dan lama angsuran yang disanggupi. Dengan menggunakan button, hasil angsuran tiap bulan menurut perhitungan termasuk bunga bank, dapat diketahui. Buatlah aplikasi tersebut!

## 3

# Visualisasi Algoritma

Ilmuwan, ahli kedokteran, analis bisnis, dan lain-lain sering menggunakan banyak informasi untuk menganalisis suatu masalah dalam mempelajari perilaku proses tertentu. Simulasi numerik menggunakan file yang berisi ratusan bahkan ribuan data untuk memberikan gambaran hasil suatu evaluasi. Data tersebut diproses sehingga mendapatkan hasil dalam bentuk visual. Visualisasi grafik dan proses untuk sekumpulan data dari ilmuwan, engineering, dan data kedokteran biasanya disebut scientific visualisasi.

Visualisasi yang efektif tergantung dari jenis kumpulan data yang diolah. Koleksi data terdiri dari nilai skalar, vektor, tekanan, atau kombinasi dari beberapa jenis yang berbeda. Kumpulan data dapat merupakan data dua dimensi atau tiga dimensi. Teknik pembuatan visualisasi termasuk graph, chart, surface rendering, dan visualisasi isi dari interior. Demikian juga, teknik image processing, animasi, dan multimedia dapat digunakan untuk pembuatan visualisasi.

## 3.1 Visualisasi Algoritma DDA

Pemrograman grafik dilengkapi dengan fungsi untuk menyatakan tampilan dalam bentuk struktur dasar geometri yang disebut output primitif dengan memasukkan output primitif tersebut sebagai struktur yang lebih kompleks. Setiap output primitif mempunyai data koordinat dan informasi lain tentang bagaimana cara objek ditampilkan pada layar. Titik dan garis lurus adalah bentuk geometri paling sederhana dari komponen gambar. Output primitif dapat digunakan untuk membentuk gambar termasuk garis, lingkaran, kurva spline, conic, dan lain-lain. Salah satu algoritma pembentukan garis adalah Digital Differential Analyzer yang dikenal dengan Algoritma DDA. Pada umumnya, implementasi algoritma DDA menggunakan bahasa pemrograman Pascal, C++, Visual Basic, dan lainnya. Aplikasi juga dapat dibuat untuk dijalankan pada browser menggunakan applet Java atau JavaScript. Supaya tampilan menarik dan

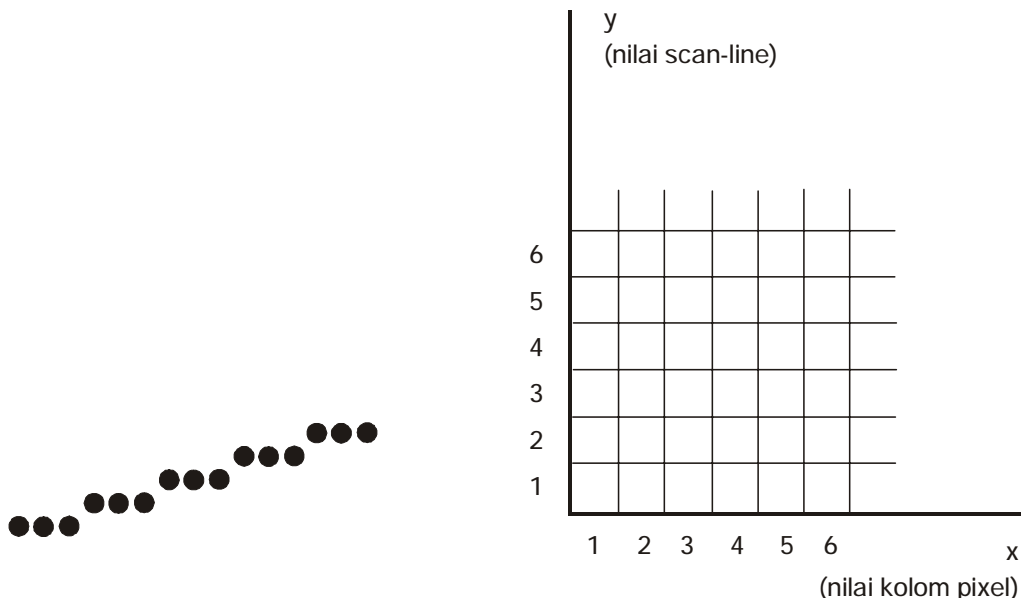
merupakan aplikasi multimedia, dapat digunakan Macromedia Flash. Macromedia Flash dilengkapi dengan pemrograman ActionScript yang mampu membuat animasi dan visualisasi dari perhitungan matematika dan logika. Hasil pemrograman dengan ActionScript dapat menghasilkan visualisasi sehingga lebih mudah dipahami oleh yang melihatnya.

### 3.1.1 Algoritma

Pembentukan titik dilakukan dengan mengkonversi suatu posisi koordinat dengan program aplikasi ke dalam suatu operasi tertentu menggunakan peralatan output. Dengan monitor CRT, pancaran elektron timbul dan hilang untuk menyalakan fosfor. Bagaimana pancaran elektron menentukan posisi titik tersebut tergantung pada teknologi display yang digunakan.

Garis dibuat dengan menentukan posisi titik di antara titik awal dan akhir dari suatu garis. Kemudian, peralatan output membuat garis sesuai posisi titik-titik tersebut. Untuk peralatan analog, seperti plotter dan random-scan display, garis lurus dapat dihasilkan dengan halus. Sedangkan pada peralatan digital, garis lurus dihasilkan dengan menetapkan titik diskrit antara titik awal dan akhir. Posisi titik diskrit sepanjang garis lurus dapat diperhitungkan dari persamaan garis tersebut. Perhitungan titik yang menghasilkan nilai pecahan, seperti (5.28, 5.83) dikonversi menjadi posisi pixel (5, 6). Pembulatan nilai koordinat ke integer menghasilkan garis yang ditampilkan pada layar menyerupai gambar tangga, seperti pada Gambar 1.

Posisi pixel dapat digambarkan sesuai nilai scan-line dan nilai kolom (posisi pixel yang dibuat tegak lurus dengan scan-line). Nilai scan-line dimulai dengan 0 pada bagian bawah layar, dan kolom pixel dimulai dengan 0 dari bagian kiri layar.



Gambar 3.1 (kiri) Garis yang dihasilkan dengan pembentukan suatu deretan titik  
 Gambar 3.2 (kanan) Posisi pixel ditentukan oleh scan-line (y) dan kolom (x)

Digital Differential Analyzer (DDA) adalah algoritma pembentukan garis berdasarkan perhitungan dx maupun dy, menggunakan rumus  $dy = m \cdot dx$ . Garis dibuat dengan menentukan dua endpoint, yaitu titik awal dan titik akhir. Setiap koordinat titik yang membentuk garis diperoleh dari perhitungan, kemudian dikonversikakan menjadi nilai integer.

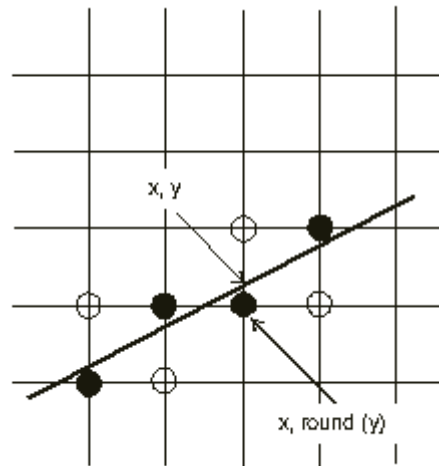
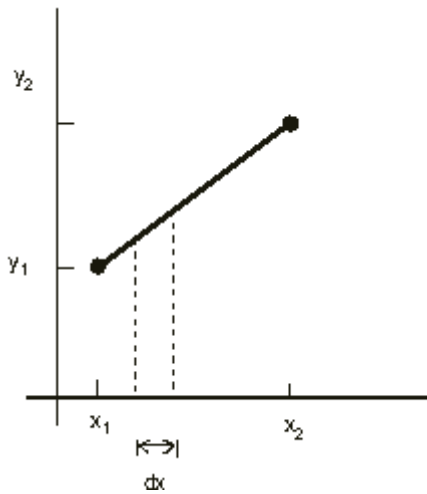
Persamaan garis menurut koordinat Cartesian adalah

$$y = m \cdot x + b$$

di mana m adalah slope (kemiringan) dari garis yang dibentuk dari dua titik, yaitu  $(x_1, y_1)$  dan  $(x_2, y_2)$  seperti dapat dilihat pada Gambar 3. Untuk penambahan x sepanjang garis yaitu dx akan mendapatkan penambahan y sebesar

$$dy = m \cdot dx$$

Pada raster system, garis dibentuk berdasarkan pixel, dengan ukuran step pada arah horisontal dan vertikal. Dengan demikian, posisi tiap pixel harus ditentukan dengan nilai diskrit posisi atau yang mendekatinya.



Gambar 3.3 (kiri) Garis lurus dengan penambahan nilai koordinat dx pada sumbu x  
 Gambar 3.4 (kanan) Pembulatan nilai posisi pixel

Pembentukan garis menurut algoritma DDA mempunyai urutan langkah sebagai berikut:

1. Tentukan dua titik yang akan dihubungkan dalam pembentukan garis.
2. Tentukan salah satu titik sebagai titik awal  $(x_1, y_1)$  dan titik akhir  $(x_2, y_2)$ .
3. Hitung  $dx = x_2 - x_1$ , dan  $dy = y_2 - y_1$ .

4. Tentukan step, yaitu jarak maksimum jumlah penambahan nilai x atau nilai y, dengan ketentuan sebagai berikut:
  - Bila nilai absolut dx lebih besar dari absolut dy, maka step = absolut dari dx
  - Bila tidak, maka step = absolut dari dy
5. Hitung penambahan koordinat pixel, yaitu  $x\_increment = dx/step$ , dan  $y\_increment = dy/step$ .
6. Koordinat selanjutnya  $(x + x\_increment, y + y\_increment)$
7. Posisi pixel pada layar ditentukan dengan pembulatan nilai koordinat tersebut.
8. Ulangi nomor 6 dan 7 untuk menentukan posisi pixel selanjutnya, sampai  $x = x_2$  dan  $y = y_2$ .

### Contoh algoritma DDA

Untuk menggambarkan algoritma DDA dalam pembentukan suatu garis yang menghubungkan titik (10,10) dan (15,12), pertama-tama ditentukan dx dan dy, kemudian dicari step untuk mendapatkan  $x\_increment$  dan  $y\_increment$ .

$$dx = x_2 - x_1 = 15 - 10 = 5, \text{ sehingga } \text{abs}(dx) = 5$$

$$dy = y_2 - y_1 = 12 - 10 = 2, \text{ sehingga } \text{abs}(dy) = 2$$

Karena  $\text{abs}(dx) > \text{abs}(dy)$ , maka  $steps = \text{abs}(dx) = 5$ , maka diperoleh

$$x\_increment = 5/5 = 1$$

$$y\_increment = 2/5 = 0.4$$

*Tabel 3.1 Koordinat pixel yang menyala*

k	x	y	round(x), round(y)
			(10,10)
0	11	10.4	(11,10)
1	12	10.8	(12,11)
3	13	11.2	(13,11)
4	14	11.6	(14,12)
5	15	12	(15,12)

## Macromedia Flash

Pada saat ini, grafik komputer telah menunjukkan kemajuan yang pesat dengan kemampuannya menghasilkan animasi menjadi lebih komunikatif. Macromedia Flash adalah perangkat lunak aplikasi untuk pembuatan animasi yang digunakan pada web, dan termasuk dalam kategori timeline-based authoring system. Macromedia Flash mampu melengkapi situs web dengan beberapa macam animasi, suara, animasi interaktif dan lain-lain. Macromedia Flash dilengkapi dengan pemrograman ActionScript yang dapat digunakan untuk membuat animasi dan visualisasi yang berhubungan dengan penyajian informasi, seperti kuis, puzzle dan aplikasi interaktif lain yang memerlukan pemrograman dengan baik. ActionScript adalah bahasa pemrograman visual berorientasi objek yang memiliki struktur, sintaks dan tata bahasa mirip dengan C++.

### Komponen yang diperlukan dalam Flash

Tujuan visualisasi algoritma DDA adalah memperlihatkan koordinat titik yang membentuk garis. Implementasi dengan Macromedia Flash dapat dilakukan dengan membuat beberapa komponen sebagai berikut:

- Input Text - Terdapat input text dengan nama variabel  $x_1$ ,  $y_1$ ,  $x_2$ , dan  $y_2$ . Koordinat titik awal dan titik akhir ditentukan dengan memasukkan data numerik melalui keyboard. Nilai variabel tersebut akan digunakan dalam perhitungan dengan algoritma DDA.
- Dynamic Text - Dynamic text adalah teks yang diperoleh dari suatu perhitungan atau input dari variabel eksternal yang berasal dari file text eksternal atau database. Hasil perhitungan algoritma DDA menghasilkan sekumpulan nilai dengan variabel  $output_x$  dan  $output_y$ . Koordinat pixel yang menyala mempunyai nilai ( $output_x$ ,  $output_y$ ).
- Scroll bar - Scroll bar digunakan untuk menggeser tampilan nilai koordinat pixel sehingga dapat dilihat semua.
- Button Hitung - Dalam Flash, diperlukan button sebagai handler untuk menghitung koordinat pixel yang menyala. Program atau script pada Flash dapat dibuat pada frame, button, dan movie clip.
- Button Visualisasi - Button yang digunakan sebagai handler untuk menampilkan garis yang dibentuk oleh koordinat pixel yang menyala.
- Garis koordinat x dan y - garis koordinat Cartesian x dan y digunakan untuk melengkapi informasi pada tampilan garis yang terbentuk oleh algoritma DDA.

### Pembuatan Flash Movie

Pembuatan Flash movie dapat dilakukan sebagai berikut:

1. Buat file baru.

2. Buat layer “Data” untuk menempatkan input dan output. Pada layer tersebut buat Input Text dengan nama variabel x1, x2, y1, dan y2 serta Dynamic Text dengan nama variabel output dan nama instance output1. Kemudian buat teks dengan Static Text, yaitu “titik awal”, “titik akhir”, “x”, “y”, “Koordinat pixel yang menyala” sebagai label input dan output tersebut.
3. Buat layer “Button”, kemudian buat button Hitung dan Visualisasi pada stage dengan memilih button dari Common Libraries.
4. Buat script pada button Hitung untuk menghitung pixel yang menyala berdasarkan algoritma DDA adalah sebagai berikut:

```

on(release) {
    dx=number((x2)-(x1));
    dy=number((y2)-(y1));

    if(Math.abs(dx)>Math.abs(dy)){
        steps=Math.abs(dx);}
    else if(Math.abs(dx)<Math.abs(dy)){
        steps=Math.abs(dy);}
    else {
        steps=Math.abs(dx);}

    x_increment =dx/(steps);
    y_increment=dy/(steps);
    x=x1;
    y=y1;
    myArray=new Array();
    for(var i=0;i<steps;i++){
        xa=1.0*x+x_increment;
        x=1.0*x+x_increment;
        myArray[i]=Math.round(xa);
    }

    outputx=myArray;
    myArray=new Array();
    for (var j=0;j<steps;j++){
        ya=1.0*y+y_increment;
        y=1.0*y+y_increment;
        myArray[j]=Math.round(ya);
    }
    outputy=myArray;
    output="x "+x1+" "+outputx.join(" ")+"\n"+y "+y1+" "+outputy.join(" ");
}

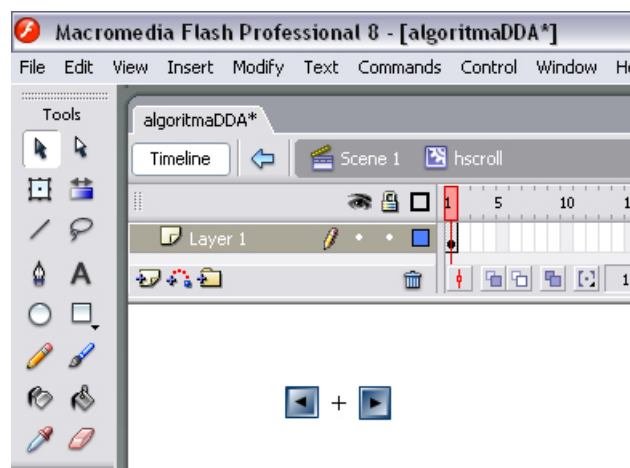
```

5. Buat script pada button Visualisasi untuk untuk menampilkan garis yang dibentuk oleh koordinat pixel yang menyala sebagai berikut:

```
on (release) {
    a=x1*10+100;
    b=y1*(-10)+400;
    c=x2*10+100;
    d=y2*(-10)+400;
    _root.createEmptyMovieClip("garis",1);
    _root.garis.lineStyle(2);
    _root.garis.moveTo(a,b);
    _root.garis.lineTo(c,d);
}
```

Di mana (a,b) merepresentasikan koordinat titik awal, dan (c,d) merepresentasikan koordinat titik akhir yang membentuk garis.

6. Buat movie clip "hscroll". Pada editing symbol buat dua buah button dengan mengambil dari Common Libraries.



Gambar 3.5. Editing simbol movie clip hscroll

Klik pada button <--, dan buat script sebagai berikut:

```
on (press) {
    pressing=true; incdec= -1;
}
on (release) {
    pressing=false;
}
```

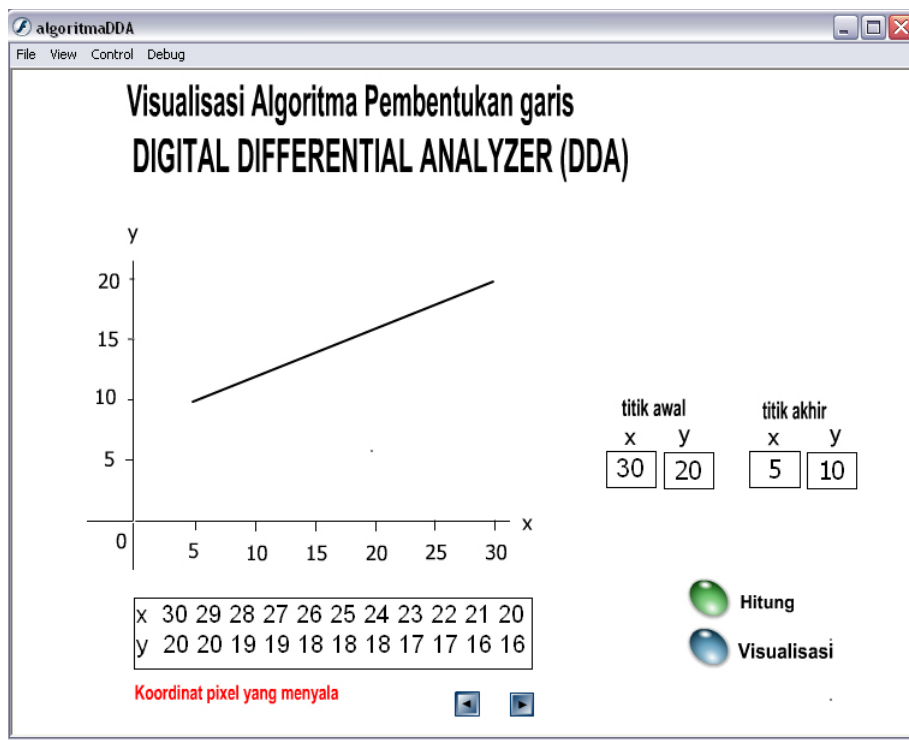
Klik pada button -->, dan buat script sebagai berikut:

```
on (press) {
    pressing=true; incdec= +1;
}
on (release) {
    pressing=false;
}
```

7. Buat layer “Scroll” kemudian drag simbol hscroll dari Library. Pada instance tersebut buat script sebagai berikut:

```
onClipEvent (enterFrame) {
    if (pressing==true){
        _root.output1.hscroll=_root.output1.hscroll+incdec;
    }
}
```

8. Pilih Control —> Test Movie untuk melihat hasilnya.



Gambar 3.6 Tampilan movie bila aplikasi dijalankan

## 3.2 Visualisasi Konversi Bilangan Biner

Bilangan biner adalah sistem bilangan dengan basis 2 digunakan untuk pengolahan data pada komputer, yaitu bilangan yang dinyatakan dengan 0 atau 1. Sedangkan bilangan desimal adalah bilangan basis 10 di mana menggunakan digit 0, 1, .. 9. Visualisasi konversi bilangan desimal ke biner merupakan bagian dari pembelajaran dengan bantuan komputer (Computer-Aided Learning). Visualisasi interaktif dapat dibuat menggunakan Macromedia Flash di mana user memasukkan bilangan desimal, kemudian komputer mengkonversinya ke dalam bilangan biner. Visualisasi dilengkapi dengan animasi digit 0 dan 1 yang membentuk bilangan biner tersebut. Macromedia Flash tidak hanya sebagai perangkat lunak dengan standar profesional untuk membuat animasi, tetapi dilengkapi dengan pemrograman ActionScript yang mampu membuat animasi dan visualisasi dari perhitungan matematika dan logika.

Pembelajaran dengan bantuan komputer sudah banyak dilakukan. Banyak perangkat lunak untuk membuat alat peraga menampilkan suatu materi pelajaran dengan mudah digunakan. Presentasi tersebut termasuk presentasi linier, yaitu siswa sebagai pendengar atau pemirsa pasif. Di samping itu terdapat presentasi non-linier atau interaktif di mana audiens dapat ikut mengambil bagian dalam pembelajaran dengan memasukkan data, kemudian komputer melakukan proses dan menampilkan hasilnya pada layar. Presentasi non-linier dapat dilengkapi dengan visualisasi supaya audiens lebih mudah memahaminya. (Anleigh, 1996)

Simulasi numerik menggunakan ratusan bahkan ribuan data untuk memberikan gambaran hasil suatu evaluasi. Data tersebut diproses sehingga mendapatkan hasil dalam bentuk visual. Visualisasi grafik dan proses untuk sekumpulan data dari ilmuwan, engineering, dan data kedokteran biasanya disebut scientific visualization. Teknik pembuatan visualisasi termasuk pembuatan graph, chart, surface rendering, dan animasi dapat dibuat menggunakan perangkat lunak komputer. Teknik image processing, animasi, dan multimedia dapat digunakan untuk pembuatan visualisasi.

Bilangan biner adalah sistem bilangan yang digunakan untuk pengolahan data pada komputer, yaitu bilangan yang dinyatakan dengan 0 atau 1, disebut bilangan basis 2. Sedangkan bilangan desimal adalah bilangan basis 10 di mana menggunakan digit 0, 1, .. 9. Konversi bilangan desimal ke biner telah banyak dibuat menggunakan bahasa pemrograman seperti Pascal, C, Visual Basic, Delphi, dan lainnya.

### 3.2.1 Sistem bilangan

Komputer tidak bekerja dengan bilangan desimal, tetapi bilangan biner. Sistem bilangan desimal adalah bilangan dengan basis 10, dan setiap posisi digit mempunyai bobot tersendiri. Bobot dalam bilangan desimal terdiri dari bobot satuan, puluhan, ratusan, dan seterusnya. Jumlah dari semua digit yang dikalikan dengan bobot masing-masing merupakan harga dari bilangan desimal tersebut. Sebagai contoh, digit paling kanan mempunyai bobot 100 (satuan), digit kedua dari kanan mempunyai bobot 101 (puluhan), digit ketiga mempunyai bobot 102 (ratusan), dan seterusnya. (Malvino, 1991)

Contoh bilangan desimal:

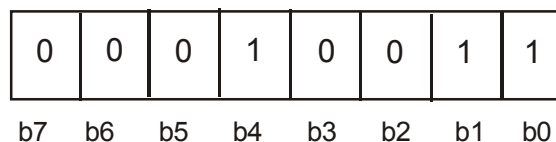
$$167 = 1 \times 10^2 + 6 \times 10^1 + 7 \times 10^0 = 1 \times 100 + 6 \times 10 + 7 \times 1 \\ = 100 + 60 + 7$$

Bilangan biner disebut juga bilangan basis 2, mempunyai bobot sebesar pangkat dari bilangan 2. Seperti dapat dilihat pada contoh berikut, bobot tersebut adalah 2<sup>0</sup>, 2<sup>1</sup>, 2<sup>2</sup>, dan seterusnya. Kesetaraan desimal dari bilangan biner sama dengan jumlah dari semua bilangan biner yang dikalikan dengan bobot masing-masing.

Contoh bilangan biner

$$10011 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ = 16 + 0 + 0 + 2 + 1 \\ = 19$$

Dengan demikian bilangan biner 10011 ekuivalen dengan desimal 19. Bilangan biner 10011 dapat dituliskan dalam 1 byte (8 digit) dengan 00010011 seperti pada Gambar 1. Bit ke-0 mempunyai nilai 1, bit ke-1 mempunyai nilai 1, bit ke-2 mempunyai nilai 0, dan seterusnya. Penambahan tiga buah nilai 0 pada bit ke-5, bit ke-6 dan bit ke-7 tidak mengubah arti bilangan biner tersebut.



*Gambar 3.7 Susunan digit biner dalam satu byte yang terdiri dari 8 digit*

### 3.2.2 Konversi bilangan desimal ke biner

Untuk mengubah bilangan desimal ke biner dapat digunakan cara double dabble. Cara ini menggunakan pembagian secara berulang-ulang dari bilangan desimal yang dikonversi, menuliskan hasil bagi serta sisanya setiap kali pembagian. Contoh di bawah adalah bagaimana mengkonversi bilangan desimal 14 menjadi bilangan biner.

$$\begin{array}{r}
 14 \\
 2 \overline{) 14} \quad 0 \text{ sisa pembagian} \\
 \underline{7} \\
 7 \overline{) 7} \quad 1 \\
 \underline{7} \\
 3 \overline{) 3} \quad 1 \\
 \underline{3} \\
 1
 \end{array}$$

*Gambar 3.8 Konversi bilangan desimal ke biner dengan cara double dabble*

Setelah selesai, hasilnya diperoleh dengan membacanya dari bawah ke atas. Dengan demikian bilangan desimal 14 ekuivalen dengan biner 1110. Algoritma untuk mengkonversi bilangan desimal menjadi biner dapat dituliskan sebagai berikut:

1. Tentukan bilangan desimal.
2. Tentukan sebuah array untuk menyimpan sisa pembagian.
3. Bilangan desimal dibagi dengan 2, sisa pembagian disimpan ke dalam array sebagai elemen ke-0.
4. Hasil bagi tersebut dibagi dengan 2, dan sisa pembagian disimpan ke dalam array sebagai elemen ke-1.
5. Ulangi nomor 4 selama hasil bagi lebih besar dari 1.
6. Hasil konversi diperoleh dengan membaca sisa pembagian dari urutan paling akhir ke urutan yang lebih awal.

### **3.2.3 Visualisasi konversi bilangan desimal ke biner**

Visualisasi dibuat dengan cara menggerakkan digit 0 atau 1 sesuai dengan sisa pembagian yang disimpan ke dalam array. Karena bilangan biner hasil konversi dibatasi hanya menampung 8 digit, maka jumlah movie digit 0 dan digit 1 masing-masing 8, sesuai ukuran array yang disediakan. Untuk membuat visualisasi dapat dituliskan algoritma dari fungsi visualisasi sebagai berikut:

1. Mulai dengan elemen array ke-0.
2. Baca elemen array. Jika nilai elemen ke-0 dari array adalah 0, maka movie digit 0 dimainkan. Jika tidak, maka movie digit 1 dimainkan.
3. Ulangi nomor sampai elemen array ke-7.

### **3.2.4 Flash Movie**

Flash movie adalah movie yang dibuat dengan Macromedia Flash. Untuk membuat visualisasi, sebelumnya perlu diketahui kemampuan Flash mengenai scripting atau pemrograman. Macromedia Flash dilengkapi dengan pemrograman ActionScript, yaitu event driven programming berorientasi objek yang memiliki sintaks seperti C++. ActionScript mempunyai kemampuan untuk melakukan pemrosesan matematika dan logika dengan baik.

#### **Kemampuan Flash**

Beberapa kemampuan dan keterbatasan yang dimiliki oleh Flash (Turner, 2001) adalah sebagai berikut:

1. Flash dapat digunakan untuk hampir semua macam visualisasi.

2. Visualisasi yang dibuat dengan Flash dapat digunakan pada halaman web karena ukuran file yang kecil. Pada umumnya, orang mengakses Internet menggunakan modem, sehingga ukuran file untuk visualisasi pada halaman web harus dibuat sekecil mungkin.
3. Pada saat ini, Flash tidak mempunyai kemampuan yang optimal untuk pembuatan animasi 3D.

### **Scripting**

Sebelum melakukan implementasi dengan Macromedia Flash, perlu direncanakan langkah-langkah untuk membuat visualisasi, sebagai berikut:

1. Tentukan pembuatan movie berdasarkan frame, scene, atau multiple movie.
2. Buat script yang digunakan untuk pemrosesan matematika dan logika.
3. Buat movie clip yang digunakan untuk animasi.
4. Sisipkan script yang digunakan untuk membuat visualisasi.

### **Konversi Bilangan Desimal ke Biner**

Dalam konversi bilangan menggunakan bahasa pemrograman, biasanya seseorang memasukkan bilangan desimal, kemudian melihat hasilnya pada layar komputer. Visualisasi digunakan untuk memperjelas pemahaman dengan animasi yang menarik, sehingga lebih mudah dipahami. Animasi menampilkan gerakan digit 0 dan 1 pada layar yang membentuk bilangan biner. Visualisasi yang digunakan dalam penelitian ini merupakan salah satu visualisasi algoritma konversi bilangan desimal ke biner. Visualisasi ini sangat sederhana, yaitu memberikan kesempatan user untuk memasukkan bilangan desimal pada kotak masukan. Dengan menekan button, komputer akan menampilkan animasi digit 0 dan 1 yang membentuk bilangan biner. Jumlah digit adalah 8, sehingga bilangan biner hasil konversi merupakan bilangan biner dengan panjang 1 byte (8 bit). Sebagai alternatif lain, juga dituliskan bilangan biner dengan jumlah digit secukupnya. Misalnya bilangan desimal 14 dikonversi ke dalam biner menjadi 00001110 atau 1110.

### **Komponen yang diperlukan dalam Flash**

Pengembangan visualisasi dengan Macromedia Flash dapat dilakukan dengan membuat beberapa komponen sebagai berikut:

1. Movie clip digit 0 - Animasi digit 0 sebanyak 8 buah sesuai dengan ukuran array yang menyimpan digit bilangan biner.
2. Movie clip digit 1 - Animasi digit 1 sebanyak 8 buah sesuai dengan ukuran array yang menyimpan digit bilangan biner.

3. Kotak Masukan - Kotak masukan merupakan Input Text, yang digunakan oleh user untuk memasukkan bilangan desimal.
4. Kotak Keluaran - Kotak keluaran merupakan DynamicText, yang digunakan untuk menampilkan hasil konversi dengan jumlah bit secukupnya.
5. Button Konversi - Button yang digunakan untuk memberikan perintah kepada komputer supaya melakukan konversi dan visualisasi.
4. Button Reset - Button yang digunakan untuk menghilangkan hasil konversi dari layar ke keadaan awal di mana user dapat memasukkan data lagi.

### **Pembuatan Flash Movie**

Pembuatan Flash movie dengan Macromedia Flash dapat dilakukan dengan mengikuti langkah-langkah sebagai berikut:

1. Buat ukuran dokumen 800 x 600 piksel.
2. Gunakan Scene 1 sebagai tampilan awal. Pada scene ini buat animasi judul “Konversi Bilangan”. Pada frame akhir untuk langsung berpindah ke Scene 2 dengan script sebagai berikut:

```
gotoAndPlay (“Scene 2”, 1);
```

Tambahkan frame pada frame akhir untuk langsung berpindah ke Scene 2 dengan cara memilih Insert—> Frame, kemudian juga buat script yang digunakan untuk berpindah langsung ke Scene 2, yaitu:

```
gotoAndPlay (“Scene 2”, 1);
```

3. Buat simbol movie clip “no10Mc”, “no11Mc”, .. “no17Mc”. Pada editing simbol buat animasi digit 0.
4. Buat simbol movie clip “satu0Mc”, “satu1Mc”, .. “satu7Mc”. Pada editing simbol buat animasi digit 1.
5. Buat scene baru dengan memilih Insert —> Scene.
6. Pada layer pertama Scene 2 buat kotak masukan dengan Input Text dan nama variabel bilangan. Kemudian buat kotak keluaran dengan Dynamic Text dan nama variabel hasil.
7. Buat teks untuk keterangan dengan Static Text seperti pada Gambar 3.
8. Buat layer “Bit 0”. Drag simbol movie clip “no10Mc” dari Library ke stage dan beri nama instance “no10”. Demikian juga “no11Mc”, “no12Mc”, .. “no17Mc”, dan tempatkan dengan urutan dari kanan ke kiri.



Gambar 3.9 Desain stage dan timeline pada Scene 2

9. Buat layer “Bit 1”. Drag simbol movie clip “satu0Mc” dariLibrary ke stage dan beri nama instance “satu0”. Demikian juga “satu1Mc”, “satu2Mc”, .. “satu7Mc”, dan tempatkan dengan urutan dari kanan ke kiri.
10. Buat layer “Button”. Pada layer tersebut buat button “Konversi” dengan script sebagai berikut:

```

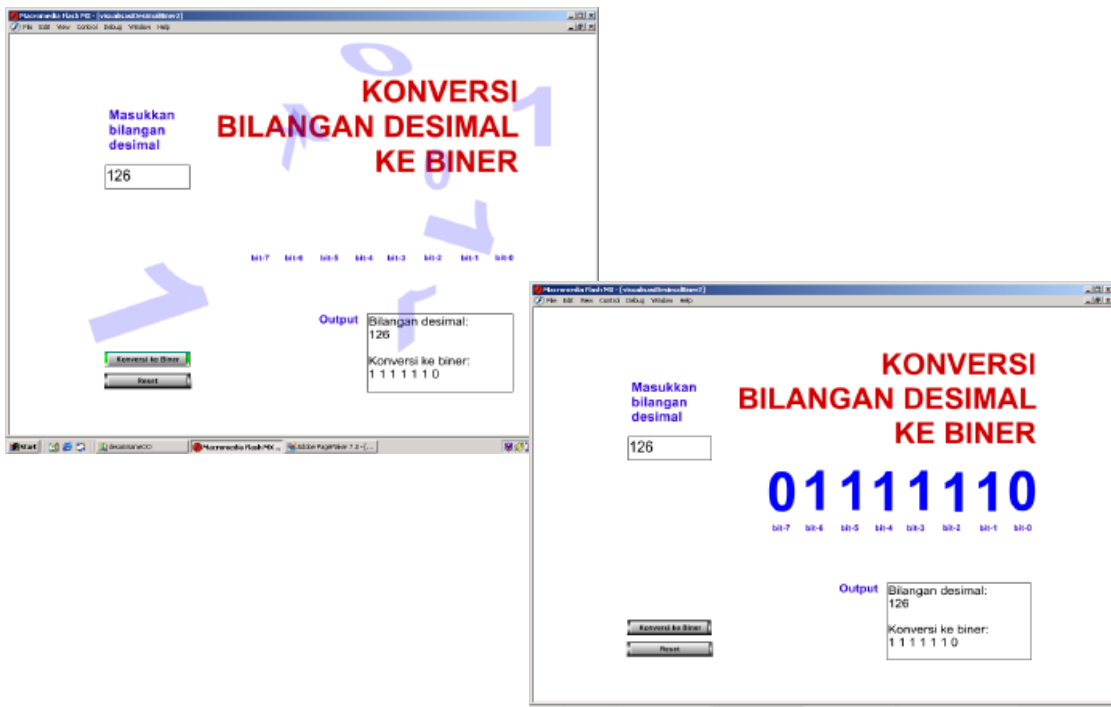
on (release) {
    bil_desimal = bilangan;
    myArray = new Array;
    i = 0;
    while(bil_desimal >= 1){
        sisa_bagi = ((parseInt(bil_desimal)) % 2);
        bil_desimal = (parseInt(bil_desimal)) / 2;
        myArray[i] = sisa_bagi;
        i ++;
    }
}
    
```

```
switch (myArray[0]) {
    case 1:
        satu0.gotoAndPlay(2);
        break;
    default:
        nol0.gotoAndPlay(2)
}
switch (myArray[1]) {
    case 1:
        satu1.gotoAndPlay(2);
        break;
    default:
        nol1.gotoAndPlay(2)
}
switch (myArray[2]) {
    case 1:
        satu2.gotoAndPlay(2);
        break;
    default:
        nol2.gotoAndPlay(2)
}
switch (myArray[3]) {
    case 1:
        satu3.gotoAndPlay(2);
        break;
    default:
        nol3.gotoAndPlay(2)
}
switch (myArray[4]) {
    case 1:
        satu4.gotoAndPlay(2);
        break;
    default:
        nol4.gotoAndPlay(2);
}
switch (myArray[5]) {
    case 1:
        satu5.gotoAndPlay(2);
        break;
    default:
        nol5.gotoAndPlay(2);
}
```

```

switch (myArray[6]) {
    case 1:
        satu6.gotoAndPlay(2);
        break;
    default:
        nol6.gotoAndPlay(2)
}
switch (myArray[7]) {
    case 1:
        satu7.gotoAndPlay(2);
        break;
    default:
        nol7.gotoAndPlay(2)
}
// hasil = hasil_temp.join(" ");
// reverse() untuk membalik urutan array
// join(" ") untuk menghilangkan tanda koma pada array
hasil_temp = myArray.reverse();
hasil = "Bilangan desimal: " + newline + bilangan + newline + newline
+ "Konversi ke biner: " + newline + hasil_temp.join(" ");
}

```



Gambar 3.10 Flash movie bila dijalankan

11. Pada layer “Button” buat button “Reset” dengan script sebagai berikut:

```
on (release) {
    bilangan = " "
    hasil = " "
    gotoAndPlay("Scene 1", 61);
}
```

12. Buat layer “Action”. Pada frame 1 buat script sebagai berikut:

```
stop();
```

13. Pilih Control → Test Movie untuk memeriksa hasilnya.

14. Simpan file.

### 3.3 Visualisasi Sorting<sup>1</sup>

Sorting adalah salah satu operasi yang cukup penting dalam bidang ilmu komputer. Sorting mengurutkan data secara ascending (urut naik) atau descending (urut turun) sehingga menghasilkan data-data yang terurut yang dapat memudahkan pencarian (searching).

Saat ini, ada beberapa jenis sorting yang banyak digunakan. Salah satunya, yang sederhana dan juga mudah digunakan yaitu selection sort dan bubble sort. Selection sort dan bubble sort digunakan untuk mengurutkan data secara ascending (urut naik).

Visualisasi mengenai cara sorting dengan selection dan bubble ini akan membantu memberikan penjelasan lebih rinci mengenai sorting tersebut sehingga memudahkan kita dalam memahaminya.

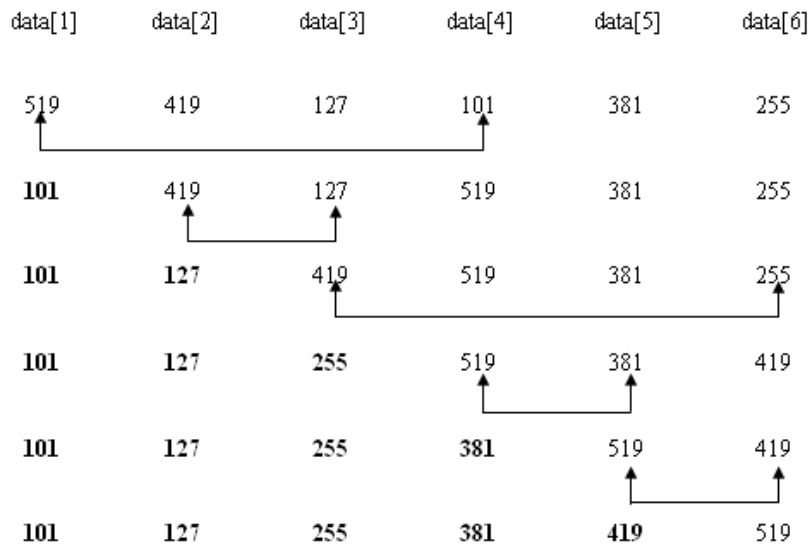
#### 3.3.1 Selection Sort

Selection sort merupakan salah satu cara yang termudah untuk mengurutkan data. Syarat dilakukannya selection sort adalah data harus berada dalam struktur array. Untuk mengurutkan data secara ascending, algoritma selection sort secara umum adalah:

1. Repeat for I ← 1 to N-1
  - Repeat for J ← I + 1 to N
  - Cari data terkecil, lalu tukar tempat dengan data pada posisi I
  - end repeat
- end repeat
2. Halt.

Contoh :

<sup>1</sup> Sundari et.al. 2007. Tugas kuliah Multimedia Universitas Tarumanagara Sem Genap 2006/2007. Jakarta.



Gambar 3.11 Proses pengurutan padaselection sort

Loop	Data terkecil	Ditukar dengan posisi, data
1	101	1, data 519
2	127	2, data 419
3	255	3, data 419
4	381	4, data 519
5	419	5, data 519

Gambar 3.12 Loop behavior table

### 3.3.2 Bubble Sort

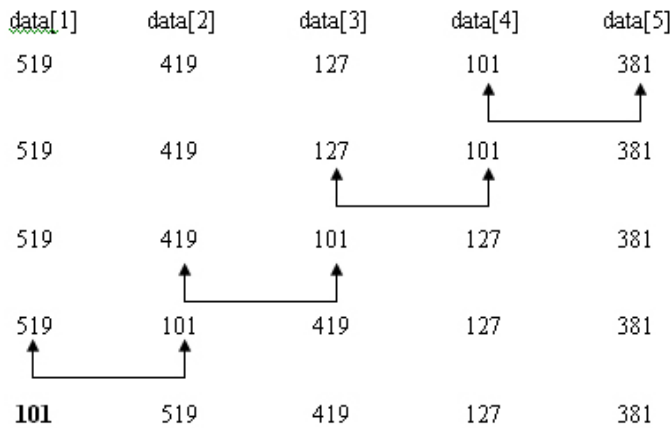
Syarat dilakukannya bubble sort adalah data harus berada dalam struktur array. Untuk mengurutkan data secara ascending, algoritma bubble sort secara umum adalah:

1. Repeat for I ← 2 to N
  - Repeat for J ← N downto I
  - Jika data[J-1] > data[J] maka tukarkan posisi data
  - end repeat
- end repeat
2. Halt.

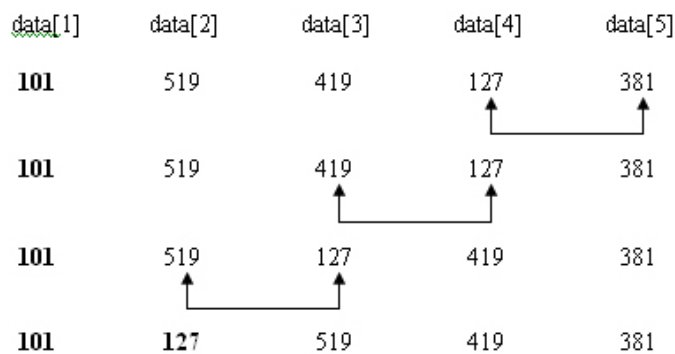
Contoh:

data[1]	data[2]	data[3]	data[4]	data[5]
519	419	127	101	381

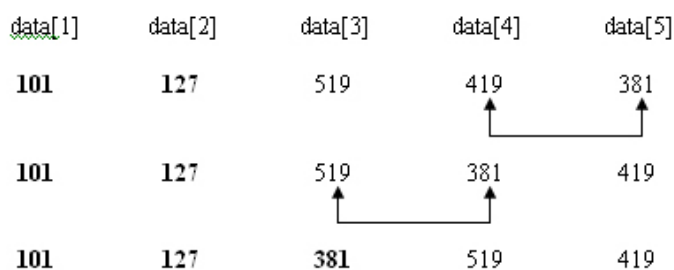
- Pola pergerakan elemen pada putaran 1 :



- Pola pergerakan elemen pada putaran 2 :



- Pola pergerakan elemen pada putaran 3 :



*Gambar 3.12 Proses pengurutan pada bubble sort*

- Pola pergerakan elemen pada putaran 4 :

data[1]	data[2]	data[3]	data[4]	data[5]
101	127	381	519	419
101	127	381	419	519

- Pola pergerakan elemen pada putaran 5 :

data[1]	data[2]	data[3]	data[4]	data[5]
101	127	381	419	519

Gambar 3.13 Proses pengurutan pada bubble sort (lanjutan)

### 3.3.3. Sorting dengan manual

Pembuatan sorting dilakukan dengan cara menukar elemen secara manual, sehingga menghasilkan tampilan sorting seperti yang diharapkan.

#### Desain Visual

Aplikasi visualisasi sorting ini terdiri dari 5 scene, sebagai berikut:

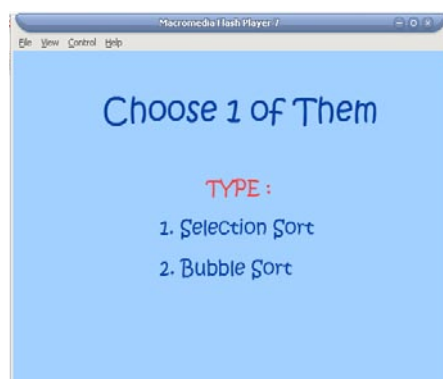
1. Scene 1 : Intro, digunakan untuk mulai masuk aplikasi
2. Scene 2 : Visualisasi Selection sort, memperlihatkan proses Selection sort dengan jumlah data yang dipilih.
3. Scene 3 : Visualisasi Bubble sort, memperlihatkan proses Bubble sort dengan jumlah data yang dipilih.
4. Scene 4 : Pilihan Selection sort, memperlihatkan 3 pilihan jumlah data dengan Selection sort
5. Scene 5 : Pilihan Bubble sort, memperlihatkan 3 pilihan jumlah data dengan Bubble sort.

#### Storyboard

##### 1. Scene 1

Link : Scene 4

Scene 5



2. Scene 4

Link : Scene 1  
Scene 2



2.1 Scene 2

Link : Scene 4



2.2 Scene 2

Link : Scene 4



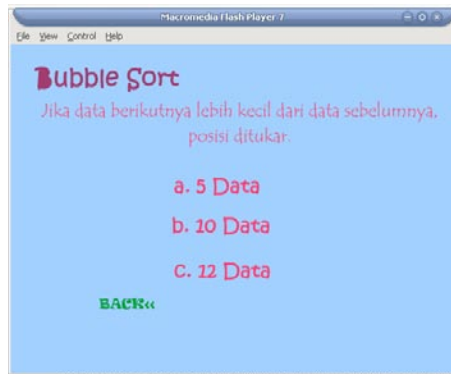
2.3 Scene 2

Link : Scene 4



3. Scene 5

Link : Scene 1  
Scene 3



3.1 Scene 3

Link : Scene 5



3.2 Scene 3

Link : Scene 5



3.3 Scene 3

Link : Scene 5



### 3.3.3. Sorting dengan pemrograman<sup>2</sup>

Pembuatan sorting dilakukan dengan cara pemrograman dengan ActionScript, sehingga menghasilkan tampilan yang diharapkan diharapkan.

#### Class Animasi

Class Animasi merupakan external ActionScript yang digunakan untuk menjalankan animasi pada stage. Script pada class Animasi adalah sebagai berikut:

```
class Animasi extends MovieClip {
    public var index = new Array(1);
    private var swap = new Array(1);
    private var temp = new Array(1);
    private var poin = new Array(1);
    private var poin2 = new Array(1);
    private var awal = new Array(1);
    private var animasi, frame:Number;
    public function banding(m:Number, n:Number):Void {
        //trace(index.length);
        var x:Number;
        x = 1;
        var y:Number;
        y = 1;
        for (var i = 1; i<m; i++) {
            index[i] = i;
            //trace(index[i]);
            //trace("-----");
        }
        for (var k = 1; k<m-1; k++) {
            for (var l = 1; l<m-1; l++) {
                if (_root["Tumpukan"+index[l]].angkabox<_root["Tumpukan"+
index[l+1]].angkabox) {
                    //trace("Tentukan Poin Pindah");
                    poin[x] = index[l].toString();
                    poin[x+1] = index[l+1].toString();
                    //trace("Poin "+poin[x]+""+" dan "+poin[x+1]+""");
                    //trace(_root["Tumpukan"+index[l]].angkabox+
Tukar dgn "+_root["Tumpukan"+index[l+1]].angkabox);
                    x = x+2;
                    swap[1] = index[l];
                    index[l] = index[l+1];
                    index[l+1] = swap[1];
                }
            }
        }
    }
}
```

2 Tubagus Saepul Anwar at.al. Tugas kuliah Interactive Multimedia Sem Genap 2005/2006 Sekolah Tinggi Teknik Multimedia Cendekia Abditama. Tangerang. 2006

```

        for (var i = 1; i<m; i++) {

//trace(" Index |"+index[i]+"---->|"+_root["Tumpukan"+index[i]].angkabox+"|");
            }
        }
    }
    var h:Number;
    h = 2;
    for (var s = 1; s<h; s++) {
        if (poin[s] != undefined) {
            //trace(poin[s]+"---->"+_root["Tumpukan"+poin[s]].angkabox);
            h++;
        }
    }
}
public function cekUrutan(m:Number):Number {
    var n:Number;
    var pos:Number;
    pos = 1;
    for (var i = 1; i<m; i++) {
        if (_root["Tumpukan"+pos].angkabox>=_root["Tumpukan"+(i+1)].
angkabox) {
            pos = i+1;
            n = 0;
            //trace(_root["Tumpukan"+pos].angkabox+"    <    "+_
root["Tumpukan"+(i+1)].angkabox);
        }
        if (_root["Tumpukan"+pos].angkabox<_root["Tumpukan"+(i+1)].
angkabox) {
            //trace(_root["Tumpukan"+pos].angkabox+"    >    "+_
root["Tumpukan"+(i+1)].angkabox);
            n = 1;
            i = m+1;
        }
    }
    //trace("TRUE SORT");
    return n;
}
public function banding2(m:Number, n:Number):Void {
    var y:Number;
    y = 1;
    //trace("//DATA INDEX AWAL //");
}

```

```

for (var i = 1; i<m; i++) {
    temp[i] = _root["Tumpukan"+i].angkabox;
    index[i] = i;
    awal[i] = i;
    //trace("index : "+index[i]);
}
//trace("-----");
var pos:Number;
var pindah:Number;
for (var i = 1; i<m-1; i++) {
    pos = i;
    pindah = 0;
    //for (var h = 1; h<m; h++) {
    //trace("langkah "+i+ " -- "+index[h]+"--"+temp[index[h]]);
    //}
    //trace("-----/ "+i+ " /--start----");
    for (var j = 1; j<m-1; j++) {
        //trace("---"+temp[index[pos]]+" dgn "+temp[index[j+i]]);
        if (temp[index[pos]]<temp[index[j+i]]) {
            //trace("TRUE      "+temp[index[pos]]+"      <
"+temp[index[j+i]]);
            pos = j+i;
            pindah = 1;
        }
    }
    if (pindah == 1) {
        //trace("Tukar pada i ke- "+i);
        swap[1] = index[pos];
        index[pos] = index[i];
        index[i] = swap[1];
        for (var h = 1; h<m; h++) {
            //trace("langkah      "+i+ "      --      "+index[h]+"--
"+temp[index[h]]);
        }
    }
    //trace("-----/ "+i+ " /-end----");
    pindah = 0;
    //swap[1] = temp[index[pos]];
    //temp[index[pos]] = temp[index[i]];
    //temp[index[i]] = swap[1];
    poin2[y] = index[i].toString();
    poin2[y+1] = index[pos].toString();
    y = y+2;
}

```

```

        //awal[i] = index[index[pos-2]];
        //trace("awal "+awal[i]);
    }
    //MENETEPAKAN JUMLAH MAX LANGKAH ADALAH 100/2 = 50
    LANGKAH.....
    for (var b = 1; b<100; b=b+2) {
        if (poin2[b] == poin2[b+1] or poin2[b] == undefined) {
            poin2[b] = null;
            poin2[b+1] = null;
        }
    }
    //trace("----/ poin2 /----");
    for (var b = 1; b<10; b++) {
        //trace("----"+poin2[b]);
    }
    //trace("----/ HASIL AKHIR /----");
    for (var h = 1; h<m; h++) {
        //trace("index "+index[h]);
    }
}
public function kirimKoordinat(t:Number):Number {
    return poin[t];
}
public function kirimKoordinat2(p:Number):Number {
    return poin2[p];
}
public function HapusArray(k:Number) {
    for (var i = 1; i<=k; i++) {
        index.pop();
        swap.pop();
        poin.pop();
        poin2.pop();
        awal.pop();
    }
    //trace("=====");
    //trace("==//semua isi Array di hapus//...");
    //trace("=====");
}
}
}

```

## Class Sortir

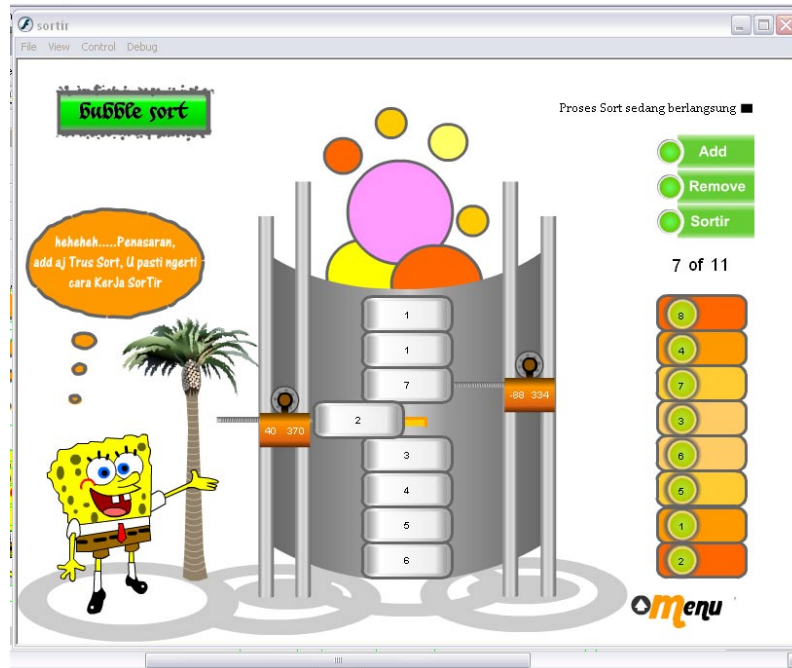
Class Sortir merupakan external ActionScript untuk menjalankan pengurutan sesuai metode yang dipilih. Script pada class Anmasi adalah sebagai berikut:

```
class Sortir {
    public var nTumpukan, besar, Max, posisiAwal_y:Number;
    public var tumpukan:Array;
    private var posisi_y = new Array(1);
    private var posisi_x = new Array(1);
    public function setMax(nomor:Number):Number {
        Max = nomor;
        return Max;
    }
    public function setPosisi():Number {
        posisiAwal_y = 0;
        return posisiAwal_y;
    }
    public function setnTumpukan(no:Number):Number {
        nTumpukan = no;
        return nTumpukan;
    }
    public function duplicate(i:Number):Void {
        if (i<9) {
            //trace(i);
            _root.attachMovie("Tumpukan", "Tumpukan"+i, 9-i);
            _root.attachMovie("Papan", "Pointer"+i, 10+(i*3));
            _root["Pointer"+i]._alpha = 50;
            _root["Pointer"+i]._visible = 1;
            _root["Pointer"+i]._y = 550-36*i;
            _root["Pointer"+i]._x = 400;
            _root["Tumpukan"+i]._y = 550-36*i;
            _root["Tumpukan"+i]._x = 400;
            //trace("---DEPTH "+_root["Tumpukan"+i].getDepth);
            _root["Pointer"+i].angkabox = i;
            _root["Tumpukan"+i].angkabox = 1+random(9);
            //trace(_root["Tumpukan"+i].angkabox);
            //trace("Tumpukan"+i);
            //trace(_root["Tumpukan"+i]._x);
            //trace(_root["Tumpukan"+i]._y);
        }
        for (var k = 1; k<=i; k++) {
            posisi_y[k] = _root["Pointer"+k]._y;
            posisi_x[k] = _root["Pointer"+k]._x;
        }
    }
}
```

```

public function ReturnTumpukan():Array {
    return tumpukan;
}
public function hapusDuplicate(i:Number):Void {
    //trace(_root["Tumpukan"+i]);
    //trace(_root["Pointer"+i]);
    removeMovieClip(_root["Tumpukan"+i]);
    removeMovieClip(_root["Pointer"+i]);
}
public function SapuBersih(i:Number):Void {
    for (var j = 0; j<=i; j++) {
        removeMovieClip(_root["Tumpukan"+j]);
        removeMovieClip(_root["Pointer"+j]);
        //trace("Tumpukan"+i);
    }
}
public function balikPosisix(m:Number, nilai:Number):Void {
    posisi_x[m] = nilai;
}
public function balikPosisiy(m:Number, nilai:Number):Void {
    posisi_y[m] = nilai;
}
public function getPosisix(m:Number):Number {
    return posisi_x[m];
}
public function getPosisiy(m:Number):Number {
    return posisi_y[m];
}
public function tambahTumpukan(nTumpukan:Number, Max:Number):Number {
    //check Tumpukan yang ada
    if (nTumpukan<=Max) {
        nTumpukan = nTumpukan+1;
    }
    return nTumpukan;
}
public function kurangTumpukan():Void {
    if (nTumpukan>=1) {
        nTumpukan -= 1;
    }
}
}

```



Gambar 3.14 Movie sorting bila dijalankan

### 3.4 Visualisasi Menara Hanoi<sup>3</sup>

Di sebuah kuil besar di Benares, dibawah kubah yang merupakan pusat dari dunia, terdapat sebuah disk kuning yang teratur dalam tiga buah jarum berlian, dimana masing-masing jarum tersebut setinggi kubik dan setebal badan lebah. Pada salah satu dari tiga jarum tersebut, pada penciptaannya, Tuhan meletakkan 64 disk emas murni, disk yang terbesar diletakkan pada bagian bawah dan yang lain semakin kecil ukurannya di letakkan di atas yang besar. Ini merupakan Menara Bramah (Tower of Bramah). Siang dan malam, para pendeta memindahkan disk dari satu jarum ke jarum lainnya berdasarkan aturan yang ditetapkan dari Bramah, dimana dalam bertugas pendeta tidak boleh memindahkan lebih dari satu disk dalam waktu yang sama dan dia harus meletakkan plat dalam jarum hingga tidak ada disk yang kecil berada di bawah. Ketika 64 disk tersebut telah berhasil dipindahkan dari jarum yang satu ke jarum yang lainnya, seluruh menara akan menjadi debu dan dengan halilintar dunia akan menghilang.

Visualisasi Hanoi Tower adalah salah satu contoh dari aplikasi penggunaan multimedia untuk mempresentasikan cara pemindahan tingkat dari satu tempat ke tempat lain dengan syarat penyusunan dimulai dari ukuran yang paling besar ke paling kecil. Konsep dari visualisasi ini adalah menggambarkan sulitnya menyajikan informasi dengan satu gambar saja, atau sekumpulan gambar. Demikian juga tidak dapat menggunakan teks untuk menerangkan informasi, sehingga menggunakan

3 Tek Yin at.al. 2007. Tugas kuliah Multimedia Univrsitas Tarumanagara Sem Genap 2006/2007. Jakarta.

Macromedia Flash MX, karena lebih mudah dalam pembuatan dan menghasilkan visualisasi yang menarik dan mudah dimengerti. Visualisasi ini merupakan animasi berbasis cast (cast-based animation) yang mencakup pembuatan kontrol dari masing-masing objek yang bergerak melintasi background.<sup>4</sup>

### 3.4.1 Rumusan Masalah

Visualisasi ini terdiri dari tiga tingkat, dan lima tingkat. Pemindahan tingkat-tingkat ini menggunakan algoritma Hanoi Tower, yaitu pemindahan satu demi satu dan menghasilkan tingkat yang sudah teratur yakni dari ukuran terbesar ke ukuran terkecil.

### 3.4.2. Tujuan

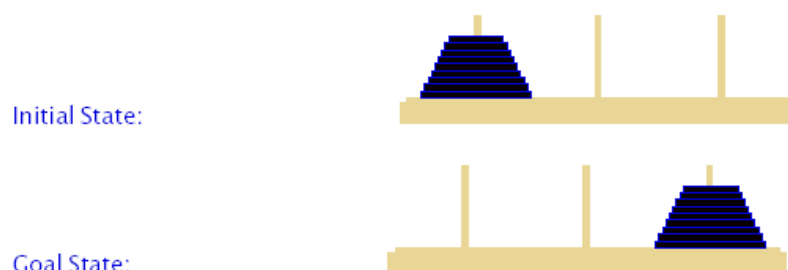
Tujuan dari laporan ini adalah untuk membuat visualisasi Hanoi Tower yang dapat mempresentasikan cara pemindahan tingkat dari satu tempat ke tempat lain dengan syarat penyusunan dimulai dari ukuran yang paling besar ke paling kecil. Aplikasi ini menghasilkan visualisasi yang menarik dan mudah dimengerti dengan menggunakan Macromedia Flash MX.

### 3.4.3 Landasan Teori

Hanoi Tower merupakan puzzle yang ditemukan oleh Edouard Lucas pada tahun 1883. Misalkan stack dari  $n$  disks diurutkan dari disk yang terbesar pada posisi atas ke disk yang terkecil diposisi bawah, bersama dengan 2 tangkai batang yang kosong, Hanoi Tower memindahkan satu stack pada satu tangkai batang ke tangkai batang yang lain. Puzzle dengan  $n = 4$  tingkat dan  $n$  disk kadang-kadang sebagai Reve's puzzle.

Masalah isomorphic untuk menemukan sebuah Hamiltonian path pada sebuah  $n$ -hypercube (Gardner 1957, 1959)

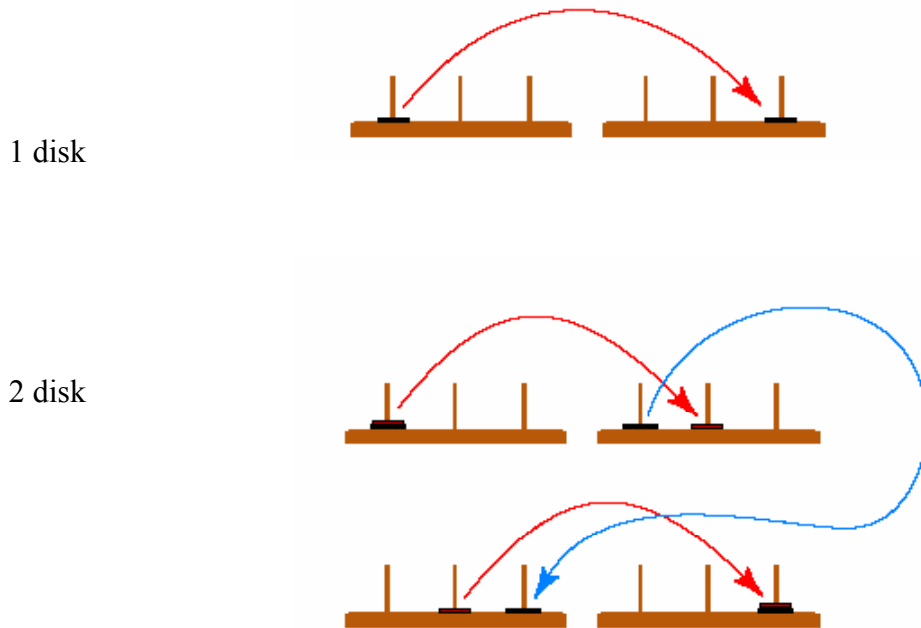
### Tower of Hanoi



Gambar 3.15 Tower of Hanoi

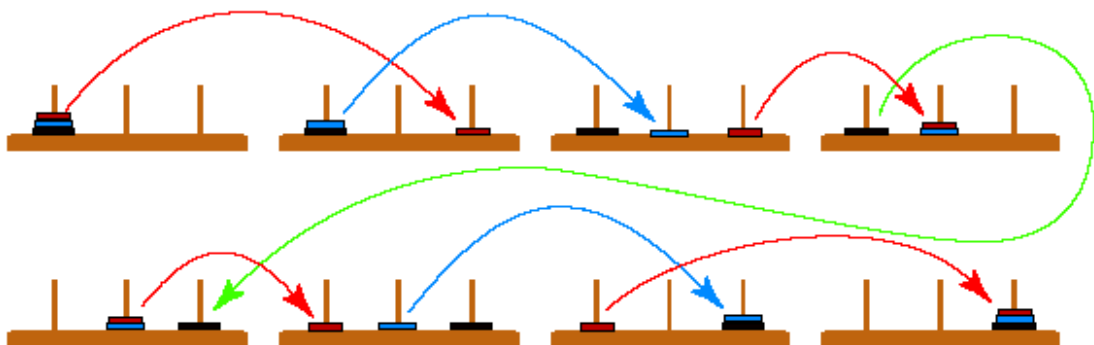
4 Seliwati et.al. 2007. Tugas kuliah Multimedia Univrsitas Tarumanagara Sem Genap 2006/2007. Jakarta.

Hanya satu disk yang bisa berpindah pada satu waktu taraf ke taraf yang lainnya, dimana yang berukuran lebih kecil selalu diatas yang besar.



Gambar 3.16 Menjalankan Tower of Hanoi dengan 1 disk (atas) dan 2 disk (bawah)

Ini merupakan solusi pemecahan dari Tower of Hanoi dengan 2 disk, dengan perhitungan  $2 \cdot 1 + 1 = 3$  gerakan.



Gambar 3.16 Menjalankan Tower of Hanoi dengan 3 disk

Ini merupakan solusi pemecahan dari Tower of Hanoi dengan 3 disk, dengan perhitungan  $2(2 \cdot 1 + 1) + 1 = 7$  gerakan.

Anggap  $M(n)$  menunjukkan jumlah minimum gerakan untuk menyelesaikan puzzle Tower of Hanoi yang memiliki  $n$  disk.

$n$	$M(n)$
1	1
2	3
3	7

Berdasarkan tabel diatas, untuk  $n = 4$  kita butuh untuk menyelesaikan dua kali puzzle dengan tiga plat ditambah satu operasi untuk memindahkan disk yang terbesar.

$$M(4) = 2 \cdot M(3) + 1 = 2 \cdot 7 + 1 = 15.$$

Sama halnya dengan  $n = 5$ , kita harus melakukan sebanyak :

$$M(5) = 2 \cdot M(4) + 1 = 2 \cdot 15 + 1 = 31.$$

### Tower of Hanoi: n Disks

$M(n)$  menunjukkan jumlah langkah minimum yang dibutuhkan untuk menyelesaikan puzzle tower of hanoi yang memiliki  $n$  disk.

- Sebelum disk terbesar (i.e., the  $n$ -th disk) dipindahkan ke taraf yang paling kanan, semua sisa yang ada ( $n - 1$ ) disk- disk tersebut harus dipindahkan ke taraf tengah. ( $n - 1$ ) disk ini harus berada pada suatu tempat dan mereka tidak bisa menghalangi perpindahan disk terbesar.) Ini membutuhkan  $M(n - 1)$ .
- Dibutuhkan lebih 1 operasi untuk memindahkan  $n$ - disk ke taraf paling kanan.
- Akhirnya, urutan langkah-langkah  $M(n - 1)$  dibutuhkan untuk memindahkan  $n - 1$  disk dari taraf tengah ke taraf paling kanan.

Kita kemudian mengambil relasi rekrusif,

$$M(n) = 2M(n - 1) + 1.$$

### Tower of Hanoi: Solusi

Dimana solusi untuk disk tunggal adalah

$$M(1) = 1$$

Dan relasi rekursifnya adalah

$$M(n) = 2M(n-1) + 1$$

Yang mendefinisikan solusinya

$$M(n) = 2^n - 1$$

### Tower of Hanoi: Practical Consequences

N.B., 1 century  $\approx 4.5 \times 10^9$  seconds. (The age of the universe is  $4 \times 10^{17}$  seconds.)

Tabel 3.2 Jumlah iterasi

$n$	$2^n - 1$	$n$	$2^n - 1$	$n$	$2^n - 1$
8	255	19	524287	30	1073741823
9	511	20	1048575	31	2147483647
10	1027	21	2097151	32	4294967295
11	2047	22	4194303	33	8589934591
12	4095	23	8388607	34	17179869183
13	8191	24	16777215	35	34359738367
14	16383	25	33554431	36	68719476735
15	32767	26	67108863	37	137438953471
16	65635	27	134217727	38	274877906943
17	131071	28	268435455	39	549755813887
18	262143	29	536870911	64	18446744073709551615

### Tower of Hanoi: Strategy

Kita telah melihat bahwa penyelesaian dari permasalahan ini adalah dengan rekursif:

- Untuk memindahkan semua  $n$  disks ke taraf paling kanan, pertama kali kita harus memindahkan puncak  $n - 1$  dari disks ke taraf tengah.
- Tetapi sebelumnya kita harus memindahkan puncak  $n - 2$  dari disks ke taraf paling kanan.
- Sebelum kita melaksanakan langkah kedua, kita harus memindahkan puncak  $n - 3$  dari disks ke taraf tengah dan begitu seterusnya.

**Tower of Hanoi: Solution Path**

- Jika jumlah disks ganjil, maka langkah pertama seharusnya mentransfer disk teratas ke taraf kanan.
- Jika jumlah disk genap, maka langkah pertama seharusnya transfer disk teratas ke taraf tengah.

Rangkaian state yang digunakan untuk menyelesaikan puzzle disebut Solution Path. Jumlah langkah terpendek solution path ke n - disk adalah  $2^n$

**Tower of Hanoi: Legal States**

Konfigurasi dalam puzzle tower hanoi dikatakan legal state jika sebuah disk berada diatas disk yang lebih kecil darinya. Artinya, plat yang lebih besar harus berada di bawah, dan sisannya berada di atasnya sesuai urutan dari diameter besar ke kecil.

Pertanyaan: Berapa banyak legal states untuk n disk diletakkan pada tiga taraf?

**Tower of Hanoi: Accessible States**

Sebuah konfigurasi disk dikatakan sebagai accessible states, jika dapat di realisasikan dari initial state setelah rangkaian langkah legal.

Pertanyaan: Berapa banyak accessible states untuk n disk diletakkan pada tiga taraf?

**Tower of Hanoi: Examples**

Tabel 3.3 Solusi path pada Tower of Hanoi

Number of Disks $n$	Length of Solution Path $2^n$	Number of Legal States $3^n$	Fraction of Visited States $(2/3)^n$
1	2	3	0.6666667
2	4	9	0.4444446
3	8	27	0.2962963
4	16	81	0.1975309
5	32	243	0.1316872
6	64	729	0.0877915
7	128	2187	0.0585277
8	256	6561	0.0390184
32	4294967296	1853020188851841	$2.32 \times 10^{-6}$

## Representasi keadaan

Sangat tidak praktis merepresentasikan sebuah state dengan ilustrasi. Maka kita akan mengadopsi sebuah list notasi.

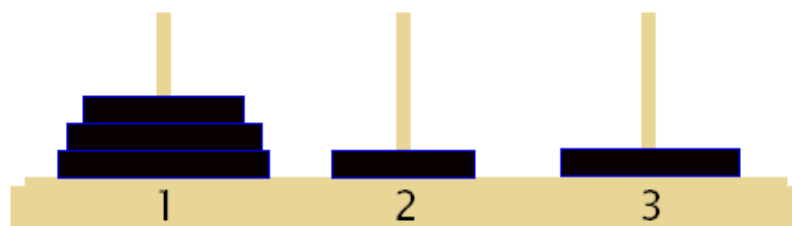
Sebagai contoh untuk lima disk puzzle, notasinya

( 2 1 3 1 1 )

Mengindikasikan

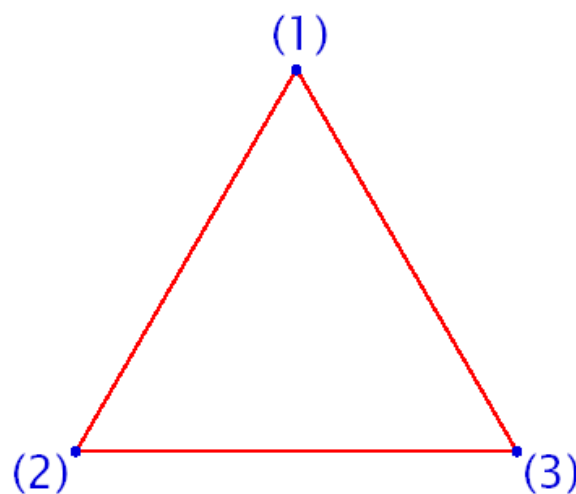
- Disk terkecil berada pada taraf 2 ( taraf tengah)
- Disk lebih besar selanjutnya pada taraf 1 ( taraf kiri )
- Disk lebih besar selanjutnya pada taraf 3 (taraf kanan)
- Disk lebih besar selanjutnya pada taraf 1
- Disk lebih besar selanjutnya (i.e., paling besar) pada taraf 1

Konfigurasi legal satu – satunya digambarkan seperti ini



Gambar 3.17 Representasi disk yang dipindahkan dari menara 1 ke menara 2 dan 3

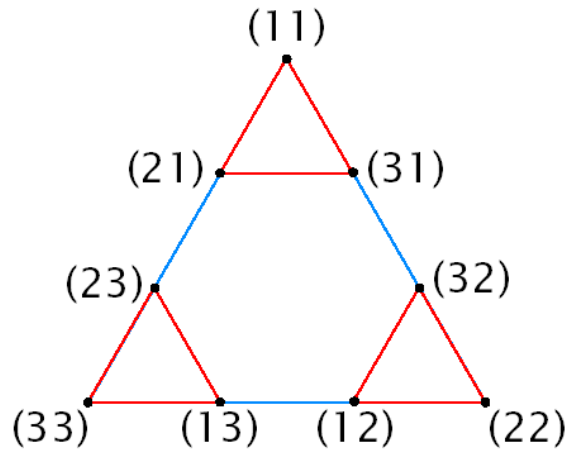
States Graph: 1 disk



Gambar 3.18 State graph 1 disk

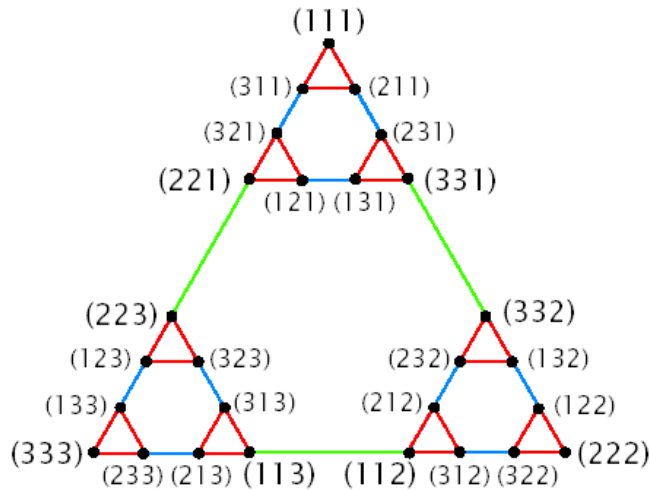
Tiap state direpresentasikan dengan labeled vertex; legal move direpresentasikan dengan edges.

States Graph: 2 disk



Gambar 3.19 State graph 2 disk

States Graph: 3 disk



Gambar 3.20 State graph 3 disk

### 3.4.4 Flash Movie

Flash movie dibuat merupakan game di mana user dapat memindahkan disk pada posisi yang dikehendaki. Bila disk yang lebih besar ternyata di atas disk yang lebih kecil, maka disk akan kembali ke tempat semula. Disk yang tersusun pada menara

merupakan array yang dideklarasikan dan ditempatkan scriptnya pada frame, sebagai berikut:

```

box1 = new Array(6);
box1[0] = "pin";
box1[1] = "pin1";
box1[2] = "pin2";
box1[3] = "pin3";
box1[4] = "pin4";
box1[5] = "pin5";
box2 = new Array(6);
box2[0] = "pin";
box2[1] = "pin";
box2[2] = "pin";
box2[3] = "pin";
box2[4] = "pin";
box2[5] = "pin";
box3 = new Array(6);
box3[0] = "pin";
box3[1] = "pin";
box3[2] = "pin";
box3[3] = "pin";
box3[4] = "pin";
box3[5] = "pin";

number_step=0;
step=number_step;

```

Sedangkan script yang digunakan untuk menggerakkan disk dari menara pertama adalah sebagai berikut:

```

on (press) {
  xxx = this._x;
  yyy = this._y;
  for (x=1; x<6; x++) {
    if (_root.box1[x] == this._name) {
      if (_root.box1[x-1] == "pin") {
        startDrag(this);
      }
    }
    if (_root.box2[x] == this._name) {
      if (_root.box2[x-1] == "pin") {
        startDrag(this);
      }
    }
  }
}

```

```

        if (_root.box3[x] == this._name) {
            if (_root.box3[x-1] == "pin") {
                startDrag(this);
            }
        }
    }
}
on (release) {
    stopDrag();
    ////////////release on tile 1
    y = 0;
    if (this._droptarget == "/til1") {
        z = 0;
        for (x=1; x<6; x++) {
            if ((_root.box1[x+1] != "pin") and (_root.box1[x+1]<this._name) and
(_root.box1[x] == "pin")) {
                y = 0;
                break;
            }
            if (_root.box1[x]<=this._name) {
                y = x;
            }
        }
    }
    ////////////release on tile 2
    if (this._droptarget == "/til2") {
        z = 1;
        for (x=1; x<6; x++) {
            if ((_root.box2[x+1] != "pin") and (_root.box2[x+1]<this._name) and
(_root.box2[x] == "pin")) {
                y = 0;
                break;
            }
            if (_root.box2[x]<=this._name) {
                y = x;
            }
        }
    }
    ////////////release on tile 3
    if (this._droptarget == "/til3") {
        z = 2;
        for (x=1; x<6; x++) {
            if ((_root.box3[x+1] != "pin") and (_root.box3[x+1]<this._name) and

```





Gambar 3.21 Movie Tower of hanoi jika dijalankan

### 3.5 Latihan

1. Buatlah aplikasi untuk mengkonversi bilangan oktal desimal menjadi oktal dan heksadesimal. Buatlah juga untuk mengkonversi bilangan oktal dan heksadesimal menjadi biner!
2. Buatlah visualisasi pendeteksian gempa bumi;
  - dari satu lokasi pendeteksian
  - dari tiga lokasi pendeteksian.
3. Buatlah visualisasi gerakan jarum jam, menit dan detik, dimana waktu yang berlaku adalah waktu default komputer yang ada pada pengguna!
4. Buatlah pengurutan data menggunakan algoritma Insertion Sort, Binary Sort dan Quick Sort!
5. Buatlah pengurutan data menggunakan algoritma Searching!

## Daftar Pustaka

- Anleigh, Prabhat K. and Kiran Thakar. 1996. *Multimedia Systems Design*. Upper Saddle River, NJ: Prentice Hall PTR.
- Budiardjo, Bagio. *Komputer dan Masyarakat*. Jakarta: Elex Media Komputindo. 1991
- Hearn, Donald and Pauline Baker. *Computer Graphics*. Upper Saddle River, NJ: Prentice-Hall International Inc. 1996
- Luther, Arch. *Authoring Interactive Multimedia*. AP Professional. New York. 1993
- Malvino, Albert Paul. 1991. *Elektronika Komputer Digital*. Jakarta: Penerbit Erlangga.
- Sanders, William B dan Winstanley, Mark. *Server-side Flash*. Hungry Minds. New York. 2001
- Santosa, Insap P, *Dasar-dasar Pemrograman Pascal, Teori dan Program Terapan*. Yogyakarta. Andi Offset. 1997
- Seliwati et.al. *Tower of Hanoi*. Tugas kuliah Multimedia Universitas Tarumanagara Sem Genap 2006/2007. Jakarta.2007
- Sundari et.al. *Selection Sort dan Bubble Sort*. Tugas kuliah Multimedia Universitas Tarumanagara Sem Genap 2006/2007. Jakarta.2007
- Sutopo. Ariesto Hadi. *Pengantar Grafika Komputer*. Yogyakarta. Gava Media. 2002  
-----, Animasi dengan Macromedia Flash berikut Actionscript. Salemba Infotek. Jakarta. 2002
- Tek Yin at.al. *Tower of Hanoi*. Tugas kuliah Multimedia Universitas Tarumanagara Sem Genap 2006/2007. Jakarta.2007
- Turner, Bill et. al. 2001. *Flash 5 Cartoons and Games*. New Delhi: Dreamtech Press.
- Tubagus Saepul Anwar at.al. *Sorting*. Tugas kuliah Interactive Multimedia Sem Genap 2005/2006 Sekolah Tinggi Teknik Multimedia Cendekia Abditama. Tangerang. 2006