

PELANTIHAN

AGUSTUS 2008



MULTIMEDIA  
INTERAKTIF  
DENGAN FLASH

[topazart.info](http://topazart.info)



# 1

# DASAR ANIMASI FLASH

**Setelah mempelajari bagian ini, diharapkan dapat:**

1. Dasar animasi Flash
2. Membuat animasi logo
3. Membuat animasi title

## 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. Macromedia Flash mampu melengkapi situs web dengan beberapa macam animasi, suara, animasi interaktif dan lain-lain. Gambar hasil dari Macromedia Flash dapat diubah ke dalam format lain untuk digunakan pada pembuatan desain web yang tidak langsung mengadaptasi Flash.

Multimedia mempunyai arti tidak hanya integrasi antara teks dan grafik sederhana saja, tetapi dilengkapi dengan suara dan animasi. Sambil mendengarkan penjelasan, dapat melihat gambar, animasi maupun membaca penjelasan dalam bentuk teks.

Dengan pemrograman ActionScript dapat dibuat 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 mempunyai struktur, sintaks dan tata bahasa mirip dengan bahasa pemrograman C++.

### Topik : **Dasar Animasi Flash**

- Konsep dasar animasi
- Tools pada Macromedia Flash
- Dasar animasi Flash (frame-by-frame, tweening)
- Animasi multi layer (animasi logo)
- Animasi dengan mask (animasi judul)
- Symbol
- Button
- Rollover image

## A. Konsep Dasar Animasi

Animasi menggambarkan object yang bergerak agar kelihatan hidup. Membuat animasi berarti menggerakkan gambar seperti kartun, lukisan, tulisan dan lain-lain. Animasi mulai dikenal sejak media televisi mulai menyajikan gambar-gambar bergerak yang berasal dari rekaman kamera maupun hasil karya seorang animator. Animasi sangat baik untuk presentasi,

pemodelan, dokumentasi dan lain-lain. Film-film animasi kartun dari Walt Disney yang telah terkenal di dunia beberapa tahun yang lalu, masih tetap disukai orang.

## 1. Teknik pembuatan animasi

Teknik pengerjaan animasi pun telah berubah seiring dengan perkembangan teknologi komputer. Dahulu pembuatan animasi dilakukan dengan membuat gambar-gambar yang digabungkan sehingga merupakan gambar yang bergerak. Untuk membuat satu durasi animasi memerlukan jumlah gambar (frame) yang cukup banyak. Jumlah frame tiap detik (frame per second /fps) merupakan satuan yang akan menghasilkan kualitas animasi. Makin banyak frame per detik, makin baik kualitas animasi yang dihasilkannya.

Efisiensi pengerjaan pembuatan animasi telah dilakukan sejak diperkenalkannya teknik animasi **cell** (celluloid sheet, semacam kertas transparan). Gagasan dasar dari cell adalah satu gambar dibuat untuk satu frame. Perubahan kecil dibuat dalam frame-frame berikutnya, sampai perubahan yang berarti merupakan **keyframe**. Animasi yang dibentuk oleh frame-frame antara dua keyframe disebut **in-between animation**, biasanya dibuat oleh seorang animator tersendiri.

Teknik animasi cell menjadi dasar pembuatan semua animasi. Dengan komputer, yang perlu ditentukan adalah keyframe, sedangkan frame-frame di antaranya akan diselesaikan oleh komputer.

## 2. Animasi dalam Macromedia Flash

Multimedia terdiri dari dua kategori, yaitu **linier** dan **non linier** (interaktif). Movie non linier dapat berinteraksi dengan aplikasi web yang lain melalui penekanan sebuah tombol navigasi, pengisian form dan lain-lain. Desainer Web membuat movie non linier dengan membuat tombol navigasi, animasi logo, animasi form dengan sinkronisasi suara. **Flash movie** adalah grafik dan animasi untuk situs web yang merupakan grafik vektor dengan ukuran file kecil, sehingga dapat di-load dalam waktu singkat. Pada dasarnya animasi terdiri dari grafik vektor, tetapi dapat juga dilengkapi dengan bitmap dan suara.

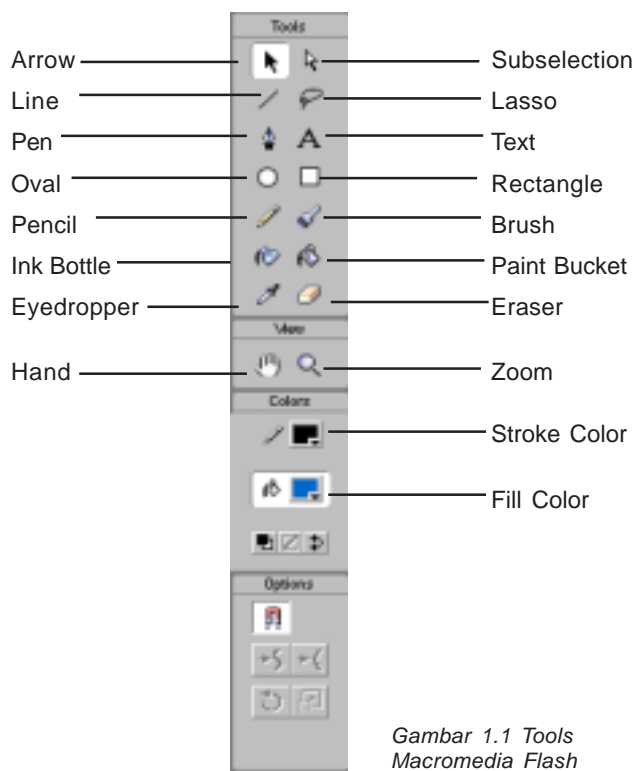
Flash movie dapat dijalankan dengan Flash Player melalui browser atau pada aplikasi stand alone. Macromedia Flash dapat digunakan untuk membuat animasi interaktif, di mana pengunjung dapat memasukkan data, kemudian Flash mengevaluasi dan menampilkan hasilnya.

## B. Tools Pada Macromedia Flash

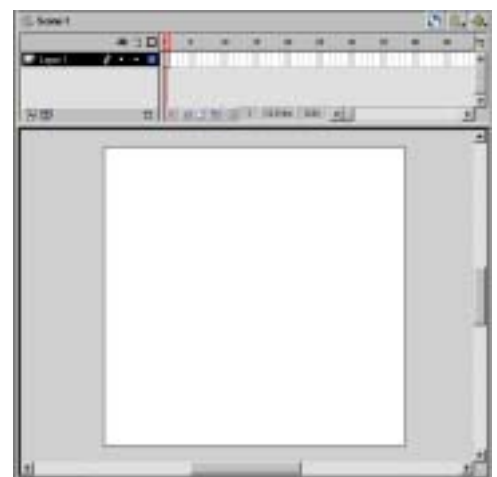
Sebelum mulai mempelajari Flash, perlu diketahui beberapa istilah yang digunakan dalam Flash, serta bagaimana aturan menggunakannya. Toolbox berisi alat-alat kerja dengan icon untuk masing-masing fungsi. Toolbox terdiri dari empat bagian, yaitu tools, view, color dan options.

**TOOLS** berisi fungsi-fungsi untuk menggambar, memilih, membuat teks, mewarnai, menghapus, dan membuat path. Tools terdiri dari beberapa icon untuk menggambar, yaitu:

- *Pencil Tool* - untuk menggambar garis seperti menggunakan pensil
- *Pen Tool* - untuk menggambar path seperti garis lurus dan garis lengkung
- *Line Tool* - untuk menggambar garis lurus
- *Oval Tool* - untuk menggambar lingkaran dan elips
- *Rectangle Tool* - untuk menggambar kotak
- *Brush Tool* - untuk menggambar menggunakan brush



Gambar 1.1 Tools Macromedia Flash



Gambar 1.2 Stage

**VIEW** berisi fungsi-fungsi untuk mewarnai.

- *Zoom Tool* - untuk memperbesar dan memperkecil gambar
- *Hand Tool* - untuk menggeser gambar

**COLORS** berisi fungsi-fungsi untuk menampilkan gambar.

- *Stroke Color* - untuk menentukan warna outline pada gambar
- *Fill Color* - untuk mewarnai gambar

**OPTIONS** berisi pilihan lain untuk membuat variasi pada fungsi-fungsi tools dan view. Bila Eraser tool diklik, akan muncul Eraser mode, Faucet dan Eraser Shape, dengan banyak pilihan.



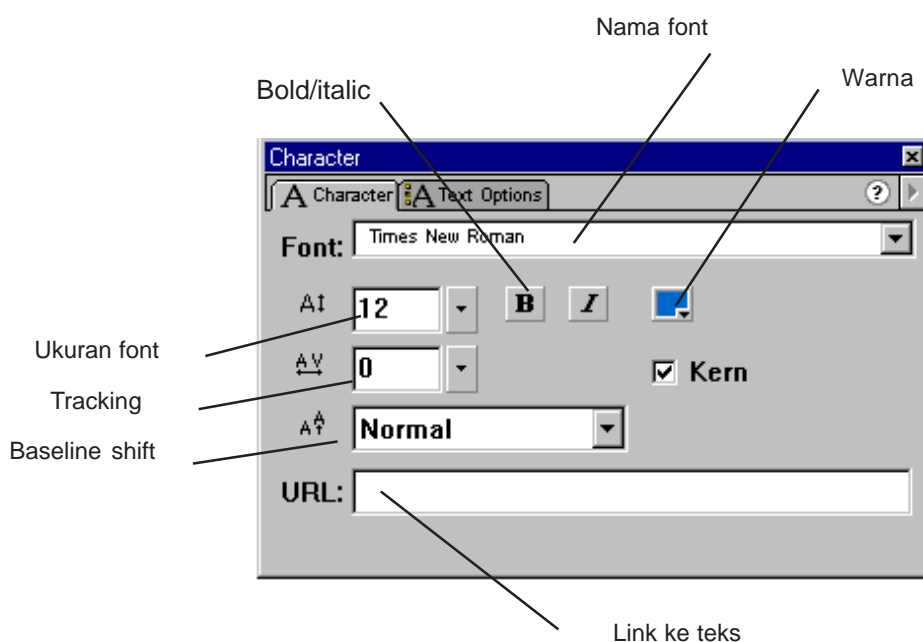
Gambar 1.3 Options pada Toolbox

## Stage

Seperti film, Flash movie mempunyai panjang yang terdiri dari sejumlah frame. Stage merupakan bidang yang berwarna putih, dimana semua object seperti gambar, teks dan foto ditempatkan dan diatur di dalamnya.

## Panel

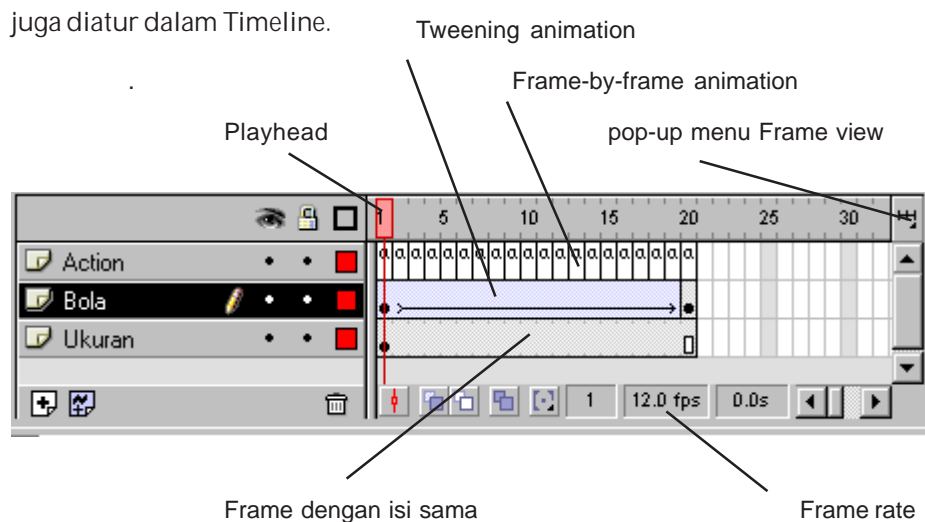
Panel berfungsi untuk menampilkan serta mengubah informasi object yang berada di stage. Terdapat beberapa macam panel dalam Flash, dan setiap panel menampilkan informasi dari suatu object yang kita kehendaki. Panel juga dapat ditemukan pada Window Properties.



Gambar 1.4 Panel Character digunakan untuk mengatur font dengan atributnya.

## Timeline

Timeline digunakan untuk mengatur semua jalan cerita, di mana actor ditampilkan dan menghilang. Suatu object yang muncul diatur dalam Timeline tersebut. Komponen utama dalam Timeline adalah *layer*, *frame* dan *playhead*. Timeline berfungsi untuk mengatur waktu suatu movie dan memunculkan objek tertentu. Penempatan efek suara dan musik latar belakang



Gambar 1.5 Timeline, frame, dan playhead

## Frame

Frame adalah gambar yang membentuk suatu gerakan bila frame tersebut ditampilkan satu demi satu berurutan, seperti Gambar 1.6. Frame merupakan konsep animasi yang dibuat secara manual maupun dengan alat bantu komputer.



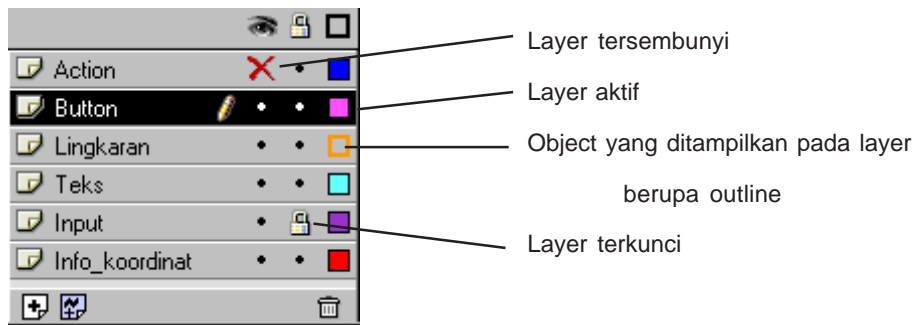
Gambar 1.6 Gambaran frame pada suatu animasi

## Playhead

Playhead dengan garis merah vertikal, menunjukkan posisi frame berada pada suatu saat. Bila posisi playhead tersebut berubah, maka gambar yang ada di stage juga berubah. Pada Gambar 2.5 dapat dilihat posisi playhead yang menunjukkan posisi frame.

## Layer

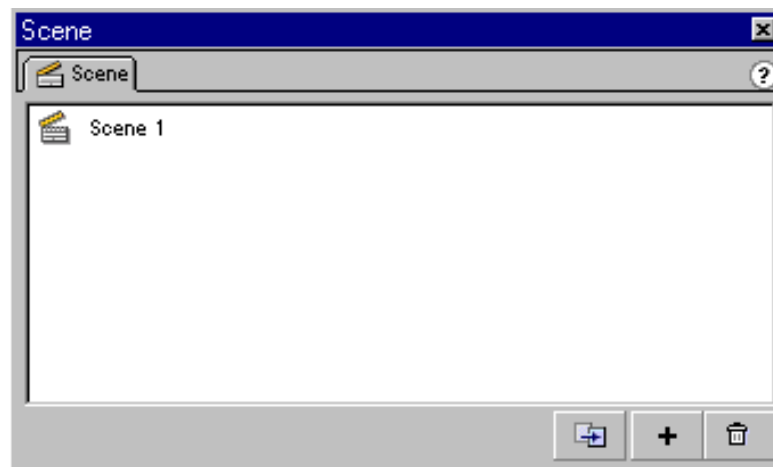
Layer digunakan untuk menempatkan object yang berbeda-beda seperti kertas transparan, dimana beberapa layer bersama-sama merupakan suatu gambar yang lengkap. Object tidak hanya gambar animasi saja, melainkan dapat berupa gambar latar belakang, teks, movie dan suara. Setiap object berada pada layer tersendiri yang independen. Macam-macam layer dapat dilihat pada gambar berikut.



Gambar 1.7 Layer

## Scene

Pada pembuatan film yang mempunyai jalan cerita cukup panjang, untuk memudahkan pengembuatannya, maka dibagi-bagi menjadi beberapa tema yang dinyatakan dalam scene. Seperti halnya pembuatan film yang terdiri dari banyak scene, animasi juga dibuat dengan konsep yang sama, untuk memudahkan dalam mengatur movie. Nama scene dapat diubah dengan mengklik dua kali pada nama scene tersebut, kemudian diketik nama yang dikehendaki. Untuk menampilkan panel Scene seperti Gambar 1.8, pilih menu **Window** -----> **Panels** -----> **Scene**.



Gambar 1.8 Scene

## C. Dasar Animasi

Dalam Flash terdapat dua macam cara membuat animasi, yaitu *frame-by-frame animation* dan *tweened animation*. *Frame-by-frame animation* dilakukan dengan membuat gambar yang berbeda-beda, kemudian ditempatkan pada frame-frame secara berurutan. *Tweened animation* dibuat dengan menentukan posisi frame awal dan akhir. Pengerjaan frame di antara kedua posisi frame tersebut (*in between frames*) dilakukan oleh Flash.

## 1. Keyframe

Keyframe adalah sebuah frame yang didefinisikan terlebih dulu untuk pembuatan animasi. Frame tersebut merupakan frame kunci perubahan pada animasi. Perbedaan keyframe pada frame-by-frame animation dan tweened animation:

- **frame-by-frame animation**  
Pada frame-by-frame animation (animasi frame-by-frame), setiap frame adalah keyframe sehingga setiap frame sangat berarti dalam animasi.
- **tweened animation**  
Pada tweened animation, keyframe merupakan kunci yaitu salah satu frame yang penting dalam animasi. Dua keyframe membentuk animasi dengan adanya frame-frame di antaranya.

## 2. Gambaran animasi pada Timeline

Gambaran keyframe pada area Timeline dapat dilihat dalam bentuk titik hitam. Macam gambaran pada area Timeline saat pembuatan animasi:



Gambaran motion-tweened animation. Frame-frame yang membentuk tweened animation digambarkan dengan garis berikut tanda panah hitam dengan latar belakang berwarna biru terang.



Shape-tweened animation. Frame-frame yang membentuk tweened animation digambarkan sebagai garis berikut tanda panah hitam dengan latar belakang berwarna hijau terang.



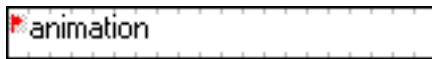
Garis putus menunjukkan bahwa keyframe akhir tidak ada.



Gambaran menunjukkan bahwa frame-frame yang ada sama dengan frame awal.



Huruf "a" kecil menunjukkan bahwa frame berhubungan dengan action yang dapat dilihat pada panel Action.



Gambar bendera merah menunjukkan bahwa pada frame tersebut terdapat label atau komentar.

### 3. Frame-by-frame animation

Animasi frame-by-frame seperti pekerjaan manual pada pembuatan film animasi, yaitu dengan membuat gambar satu persatu untuk setiap frame. Contoh berikut adalah animasi yang dibuat dengan menempatkan 5 buah gambar bitmap. Posisi maupun skala gambar pada setiap frame diatur, sehingga sejumlah frame yang banyak membentuk animasi.

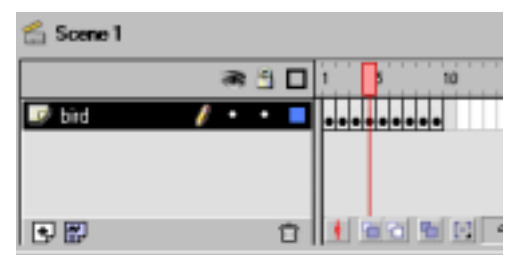
#### Untuk membuat animasi frame-by-frame:

1. Pada **File** —> **New**. Kemudian buat gambar burung pada frame 1.
2. Buat keyframe baru setelah frame tersebut dengan cara **Insert** —> **Keyframe**. Kemudian modifikasi gambar burung yang sudah ada.
3. Buat keyframe baru setelah frame tersebut dengan cara **Insert** —> **Keyframe**. Kemudian modifikasi lagi gambar burung yang sudah ada.



Gambar 1.9 (kiri)  
Gambar pada frame-by-frame animation

Gambar 1.10 (kanan)  
Timeline pada frame-by-frame animation

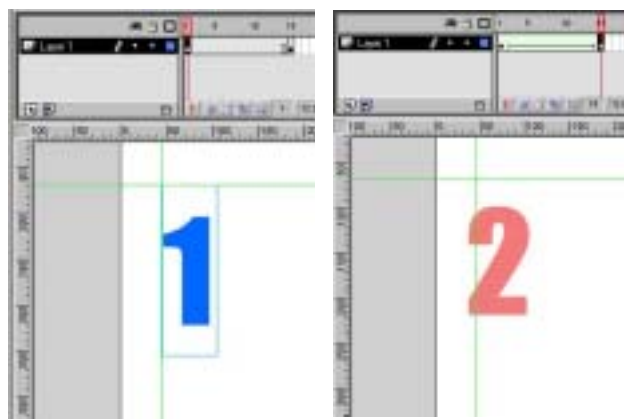


4. Ulangi nomor 4, sampai frame 8.
5. Pilih **Control** —> **Test Movie**.
6. Simpan file tersebut dengan nama **frame\_animation fla**.

## 4. Tweening animation

### Shape Tweening

Animasi shape tweening adalah animasi perubahan bentuk (shape), seperti morphing, yaitu perubahan dari satu bentuk ke bentuk lain. Selain bentuk, lokasi, ukuran dan warna juga dapat dibuat tweening. Shape tweening hanya digunakan untuk shape saja, dan tidak dapat digunakan untuk symbol, gambar bitmap atau satu blok dari teks.



Gambar 1.11 Shape tweening

**Untuk membuat shape tweening:**

1. Tentukan frame untuk memulai animasi, dan buat keyframe.
2. Gambar image awal, angka 1 dengan warna biru.
3. Tentukan keyframe kedua dan buat angka 2 dengan warna orange.
4. Klik angka "1" pada keyframe awal dan dijadikan shape dengan cara memilih **Modify** —> **Break Apart**.
5. Klik angka "2" pada keyframe akhir dan dijadikan shape. Pada angka "2" akan muncul shape hint.
6. Tempatkan ketiga shape hint pada angka "2". Select **Modify** —> **Frame**, maka keluar kotak dialog Frame Properties.

7. Pada kotak Tweening pilih **Shape**.
8. Pilih **Control** —> **Test Movie** untuk memeriksa hasilnya.
9. Simpan file tersebut dengan nama **shape\_animation fla**

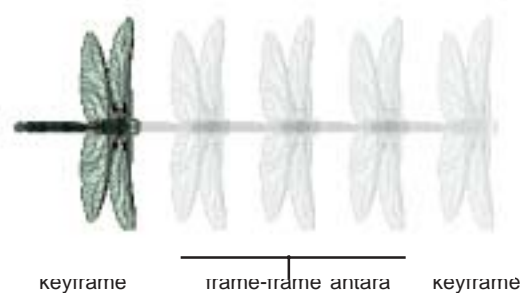
## ***Motion tweening***

Pada animtion ini, property dari instance, group dan text block dapat diubah. Flash dapat men-tween warna dari instance dengan gradasi, fade in dan fade out. Untuk men-tween warna dari group dan tulisan harus dijadikan symbol terlebih dulu.

### **Untuk membuat motion tweening:**

1. Pilih frame di mana akan dimulai animasi, dan buat keyframe.
2. Buat image awal atau drag symbol dari Library.
3. Pastikan frame awal yang aktif, pilih **Insert** —> **Create Motion Tween**
4. Tentukan frame akhir.
5. Select **Insert** —> **Keyframe**.
6. Pastikan frame akhir yang aktif.
7. Gerakkan gambar pada posisi yang dikehendaki.
8. Pilih **Control** —> **Test Movie** untuk memeriksa animasi tweening.
9. Simpan file tersebut dengan nama **tweening\_animation fla**.

Motion tweening dapat digunakan untuk membuat animasi dari image yang sudah ada. Contoh berikut adalah image dengan format file .gif.



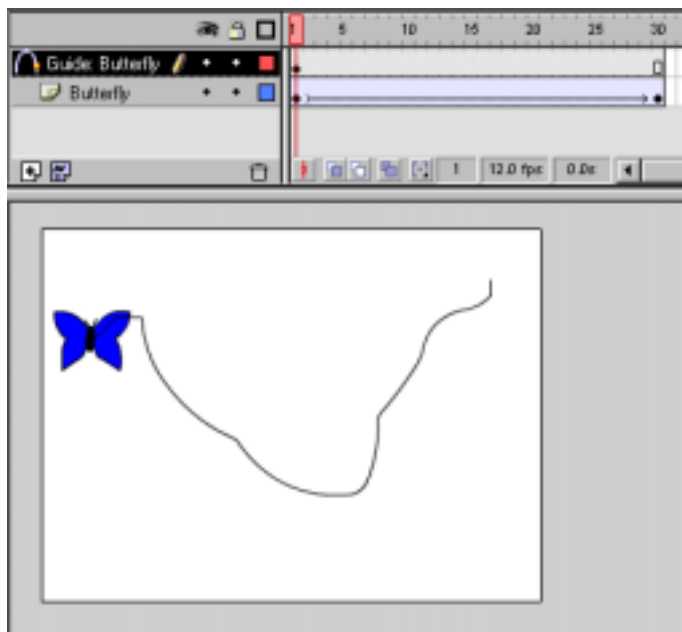
Gambar 1.12 Tweening animation

**Untuk membuat motion tweening:**

1. Pilih frame di mana akan dimulai animasi, dan buat keyframe.
2. Import file image yang ada. P
- 3 - 7. Sama dengan di atas.
8. Untuk mengatur perubahan warna pada frame akhir, pada panel Effect, pilih perubahan warna yang dikehendaki.
9. Pilih **Control** —> **Test Movie** untuk memeriksa animasi tweening.
10. Simpan file tersebut dengan nama **tweening\_insect.fla**.

***Motion tweening along path***

Gerakan object pada motion tweening dapat dibuat mengikuti path (jalur) yang ditentukan. Path berupa garis terbuka maupun tertutup, mialnya lingkaran dan elips.



Gambar 1.13 Tweening dengan path

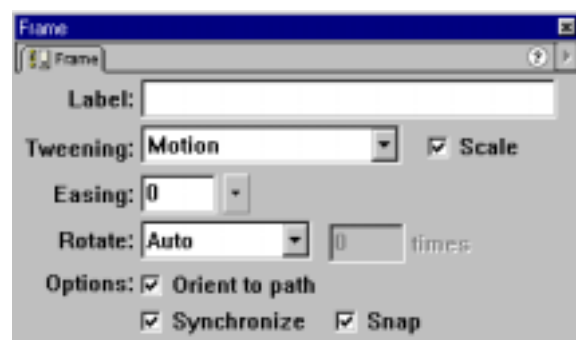
**Untuk membuat animasi mengikuti path:**

1. Pilih **File** —> **New** untuk membuat file baru.
2. Buat layer "Butterfly" untuk menempatkan gambar kupu-kupu.
3. Buat gambar kupu-kupu dengan tools yang ada dalam Toolbox, atau gambar dibuat menggunakan Macromedia Freehand.

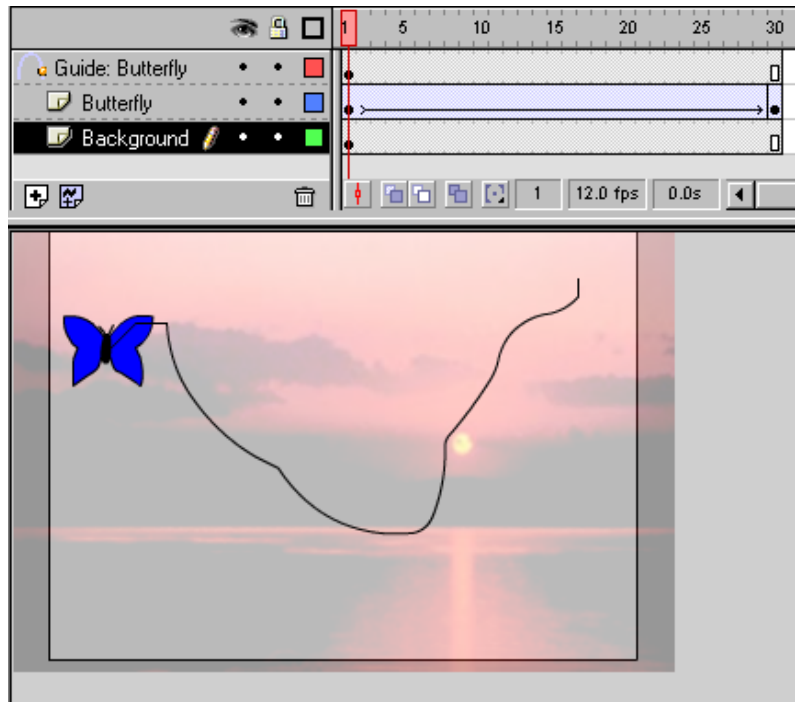
4. Buat motion tweening seperti telah dibicarakan sebelumnya.
5. Pada layer "Butterfly" klik tombol kanan mouse, kemudian pilih **Add Motion Guide**, maka muncul sebuah layer baru. Layer tersebut digunakan untuk menempatkan gambar yang digunakan sebagai path.
6. Gambar dengan Pen, Pencil, Line, Circle, Rectangle tool untuk membuat path yang diinginkan.
7. Letakkan gambar kupu-kupu tepat (snap center) pada awal path, kemudian lakukan yang sama pada akhir path.
8. Pilih **Control** —> **Test Movie** untuk memeriksa animasi tweening ditampilkan mengikuti path.
9. Simpan file tersebut dengan nama **tweening\_path.fla**.

Kupu-kupu berjalan mengikuti path dengan posisi tegak, tidak seperti layaknya kupu-kupu terbang. Bila menghendaki jalan kupu-kupu kelihatan wajar (selalu menghadap ke depan), dapat dibuat dengan cara berikut:

1. Klik frame antara awal dan akhir keyframe pada layer kupu-kupu.
2. Pilih **Modify** —> **Frame**, maka keluar kotak dialog Frame Properties.
3. Pilih **Orient to path** dan **Snap**. Pilihan **Orient to path** digunakan untuk membuat animasi berjalan menyesuaikan diri menghadap ke arah yang benar. Pilihan **Snap** digunakan untuk membuat elemen animasi tepat dengan titik registrasi.
4. Pilih **Control** —> **Test Movie**. Animasi tweening akan ditampilkan mengikuti path dengan gambar kupu-kupu selalu menghadap ke depan.
5. Simpan file tersebut dengan nama **tweening\_orient.fla**.



Gambar 1.14 Panel Frame untuk motion tweening



Gambar 1.15 Motion tweening dengan background image

Animasi dapat dibuat dengan latar belakang dari image tertentu. Pada animasi kupu-kupu dapat ditambahkan latar belakang yang ditempatkan pada layer paling bawah.

#### Untuk menambahkan latar belakang pada tweening animation:

1. Buat layer "Background" untuk menempatkan latar belakang.
2. Pilih **File** —> **Import**, kemudian impor file "sunset.jpg".
3. Image "sunset" ditempatkan pada layer "Background" yang terletak pada paling bawah. Image tersebut ditransformasi dengan skala sehingga besarnya sesuai dengan ukuran movie.
4. Pilih **Control** —> **Test Movie** untuk memeriksa hasilnya.
5. Simpan file tersebut dengan nama **tweening\_back fla**.

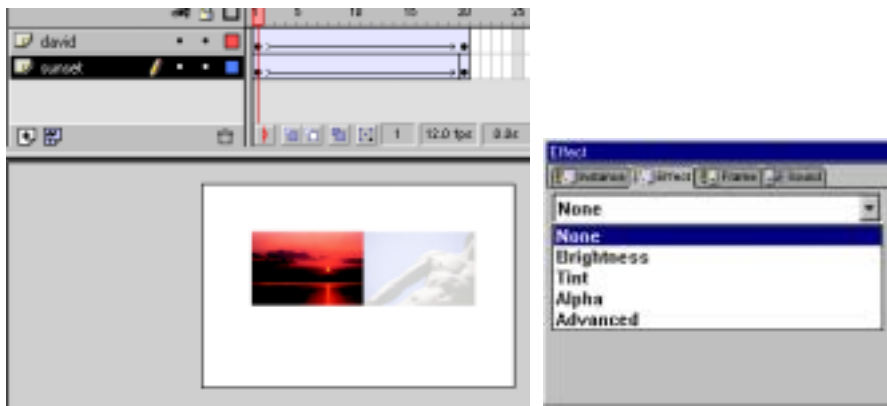
## 5. Transisi Warna

### ***Mengubah warna dan transparansi gambar***

Setiap gambar merupakan instance dari suatu simbol, dapat dilakukan efek warna terhadapnya. Untuk itu dapat digunakan panel Effect.

### Untuk membuat efek warna:

1. Pilih **Insert** —> **Keyframe** untuk membuat keyframe pada frame 1.
2. Pilih **File** —> **Import**, kemudian impor file "sunset.jpg".
3. Pilih **Insert** ---> **Convert to Symbol**, buat symbol movie clip "sunsetMc".
4. Buat keyframe kedua pada frame 20.



Gambar 1.16 Efek warna pada gambar

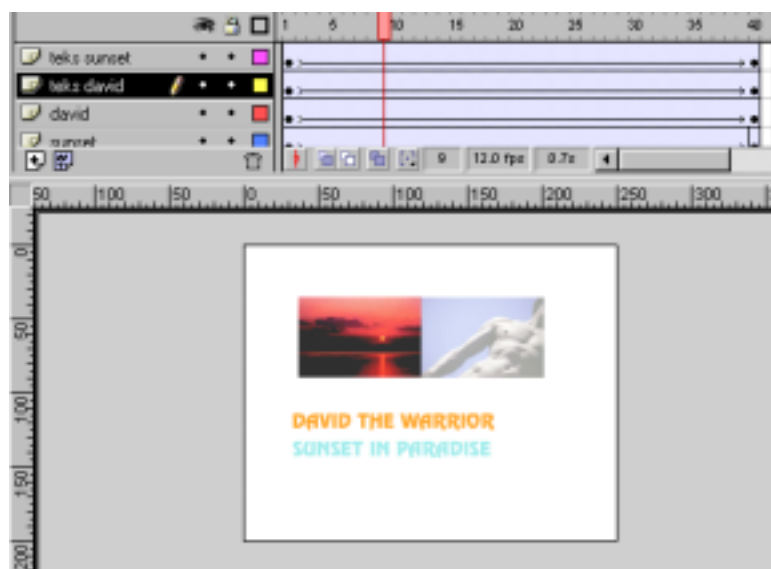
5. Pilih **Window** ---> **Panels** ---> **Effect**. Terdapat beberapa pilihan, yaitu none, brightness, alpha, tint dan advance. Tentukan nilai Alpha 20%.
6. Buat animasi motion tweening.
7. Buat layer baru untuk menempatkan gambar David.
8. Pilih **File** ---> **Import**, kemudian impor file "david.jpg".
9. Pilih **Insert** ---> **Convert to Symbol**, buat symbol movie clip "davidMc".
10. Buat keyframe kedua pada frame 20.
11. Pada frame 1 layer "David", pilih menu **Window** ---> **Panels** ---> **Effect**. Terdapat pilihan none, brightness, alpha, tint, advance. Pilih Alpha 20% untuk gambar tersebut. Pilihan effect dapat berfungsi bila gambar telah dibuat symbol movie clip.
12. Buat animasi motion tweening.
13. Pilih **Control** ---> **Test Movie**, maka animasi tweening akan ditampilkan dengan gambar yang warnanya menghilang.
14. Simpan file tersebut dengan nama **effect\_image fla**.

## Mengubah warna dan transparansi teks

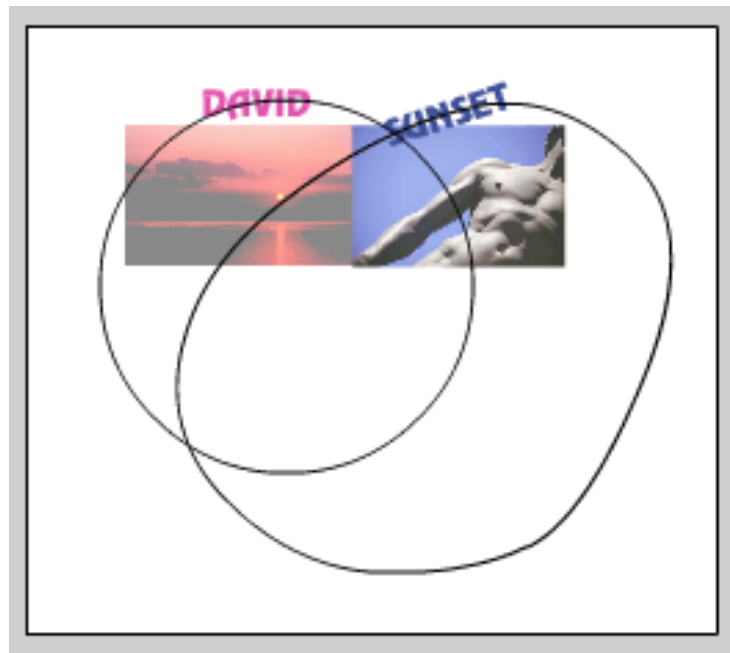
Setiap teks merupakan instance dari suatu simbol dapat dilakukan efek warna terhadapnya. Untuk itu dapat digunakan Effect panel.

### Untuk membuat efek warna pada teks:

1. Buka file "effect\_image.fla".
2. Buat layer "Teks david" untuk menempatkan teks pertama.
3. Pilih **Insert** —> **Keyframe**, keyframe pertama.
4. Buat teks dengan **Text tool** pada stage "DAVID THE WARRIOR".
5. Pilih **Insert** —> **Convert to Symbol**, buat symbol movie clip dengan nama "teksdavidMc".
6. Buat keyframe kedua. Pilih **Window** —> **Panels** —> **Effect**. Terdapat pilihan none, brightness, alpha, tint, advance. Pilih Alpha 20% untuk gambar tersebut, dan letaknya dipindah.
8. Buat animasi motion tweening.
9. Buat layer "Teks sunset" untuk menempatkan teks kedua.
10. Buat teks "SUNSET IN PARADISE", dan dibuat animasi seperti teks pertama.
11. Select **Control** —> **Test Movie**, maka animasi tweening akan ditampilkan dengan teks yang warnanya menghilang.
12. Simpan file tersebut dengan nama **effect\_text.fla**.



Gambar 1.17 Animasi teks



Gambar 1.18 Animasi teks dengan path

## ***Membuat animasi teks dengan path***

### **Untuk membuat animasi teks:**

Setiap karakter merupakan instance dari suatu symbol yang dapat diubah warnanya menggunakan panel Effect.

### **Langkah-langkah untuk membuat animasi teks:**

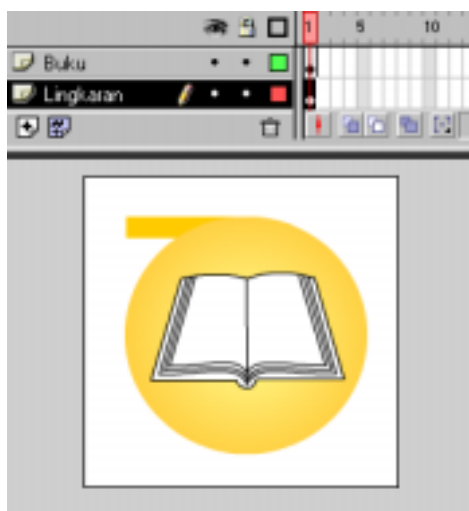
1. Buka file "effect\_image.fla".
2. Buat layer untuk menempatkan teks "DAVID".
3. Pilih **Insert** —> **Keyframe** untuk membuat keyframe pertama.
4. Buat teks "DAVID" dengan **Text tool** pada stage.
5. Buat motion tweening seperti telah dibicarakan sebelumnya.
6. Pada layer klik tombol kanan mouse, pilih **Add Motion Guide**.
7. Gambar dengan Pen, Pencil, Line, Circle, Rectangle tool path yang diinginkan.
8. Snap center dari teks pada awal path, kemudian pada akhir path.
9. Pada Frame Properties terdapat pilihan Orient to Path, Synchronize dan Snap. Bila memilih orient to path, animasi selalu menghadap ke arah path.

10. Ulangi no 2 - 8 untuk teks 'SUNSET'.
11. Pilih **Control** —> **Test Movie**, maka animasi teks akan ditampilkan mengikuti path.
12. Simpan file tersebut dengan nama **animasi\_teks\_path fla**.

## 6. Animasi logo - tweening

Animasi multilayer dapat digunakan untuk membuat animasi logo. Langkah-langkah berikut dapat dilakukan untuk membuat animasi logo.

1. Pilih **File** —> **New** untuk membuat file baru. Buat ukuran stage 150 x 150 pixel.
2. Buka file "logo\_millennium" dengan Macromedia Freehand.
3. Kembali ke Flash, buat layer pertama "Lingkaran". Copy gambar lingkaran dari Freehand dan paste ke stage pada Flash.
4. Buat layer kedua "Buku". Copy gambar buku dari Freehand dan paste ke stage pada Flash.
5. Buat motion tweening dari lingkaran dengan gerakan berputar, menggunakan **Modify** —> **Rotate**.
4. Buat motion tweening dari buku yang bergerak dari atas ke bawah.
5. Berikan effect untuk buku dan lingkaran dengan membuka Panel Effect. Pilih effect yang ada pada kotak dialog.
6. Pilih **Control** —> **Test Movie** untuk memeriksa animasi logo yang ditampilkan.
9. Simpan file tersebut dengan nama **millennium\_logo fla**.



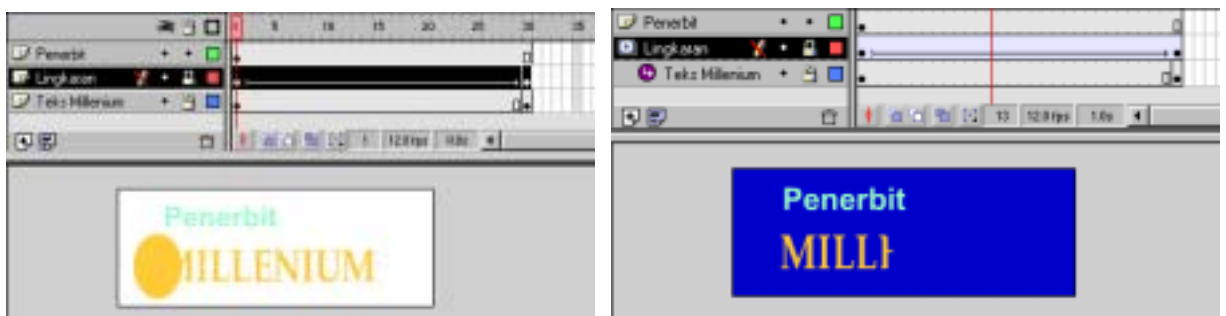
Gambar 1.19 Setting movie yang digunakan untuk animasi logo

## 7. Animasi title - mask

Untuk membuat mask pada teks:

1. Buat file baru dengan memilih **File** —> **New**. Buat ukuran stage 200 x 75 pixel.
2. Buat layer pertama "Teks Millenium" untuk menempatkan judul.
3. Tentukan keyframe awal dengan memilih **Insert** —> **Frame**.
4. Buat teks "MILLENIUM" dengan warna yellow. Buat panjang frame dari frame 1 sampai frame 30.
5. Buat layer kedua "Lingkaran" untuk menempatkan mask.
6. Buat symbol graphic "lingkaran", dengan cara memilih **Insert** —> **New Symbol**. Pada kotak dialog Symbol Properties pilih Graphic dan berikan nama "lingkaran".
7. Pada editing symbol yang muncul buat gambar lingkaran dengan Elipse Tool. Warna lingkaran dibuat sembarang.
8. Buat symbol movie clip "lingkaranMc", dengan cara memilih **Insert** —> **New Symbol**. Pada kotak dialog Symbol Properties pilih Movie Clip dan berikan nama "lingkaranMc".
9. Pada editing symbol yang muncul drag symbol graphic "lingkaran" ke dalam stage.
10. Pada Library dapat dilihat kedua symbol. Untuk membuka Library pilih **Window** —> **Library**.
11. Kembali ke Scene 1. Klik frame 1 pada layer "Lingkaran". Drag symbol "lingkaranMc" ke dalam stage.
12. Buat tweening animation dari frame 1 sampai frame 30. Klik pada frame 30, kemudian lakukan transformasi skala terhadap lingkaran sehingga lingkaran tersebut membesar dan menutupi seluruh huruf "MILLENIUM".

Gambar 1.20 Title sebelum dibuat masking (atas) dan sesudahnya (bawah)



13. Buat mask dengan cara klik pada layer "Lingkaran", kemudian pilih **Insert** —> **Mask**.
14. Buat layer ketiga "Penerbit". Pada layer tersebut tempatkan teks "Penerbit" dengan warna biru muda.
15. Buat background movie warna biru dengan memilih **Modify** —> **Movie**.
16. Pilih **Control** —> **Test Movie** untuk memeriksa hasilnya.
17. Simpan file tersebut dengan nama **millenium\_title fla**.

Effect dapat dibuat pada teks untuk mendapatkan macam-macam tampilan animasi.



2

# BUTTON

**Setelah mempelajari bagian ini, diharapkan dapat:**

1. Membuat button
2. Membuat rollover image
3. Mengubah gambar bitmap menjadi vektor

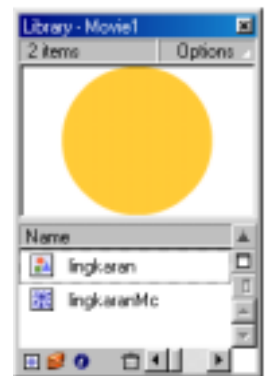
Walaupun Flash menyediakan button dalam Common Library, tetapi pembuatan button perlu diketahui untuk mendapatkan macam-macam inovasi yang menggunakan prinsip pembuatan button.

## 1. Button

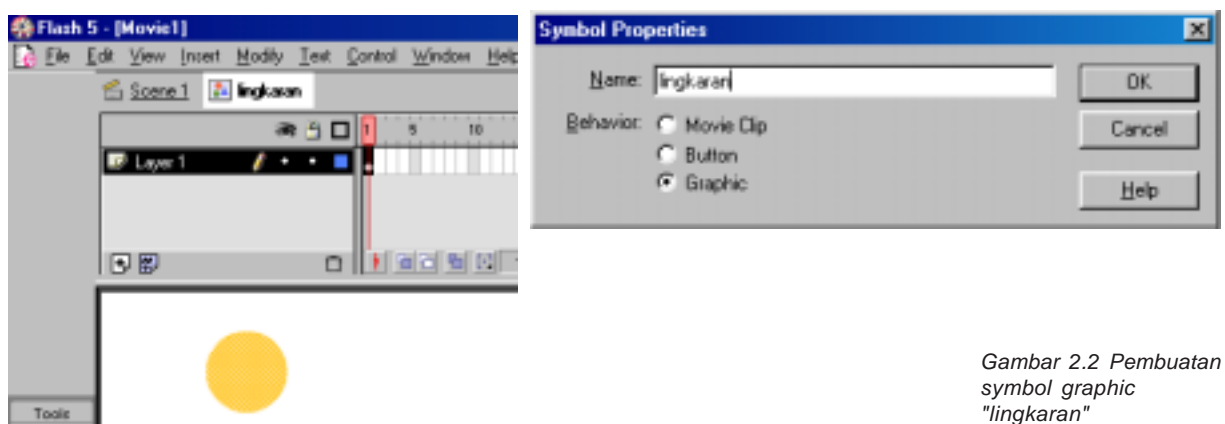
Button mempunyai 4 frame, yaitu frame 1 (up), frame 2 (over), frame 3 (down), dan frame 4 (hit).

Untuk membuat button:

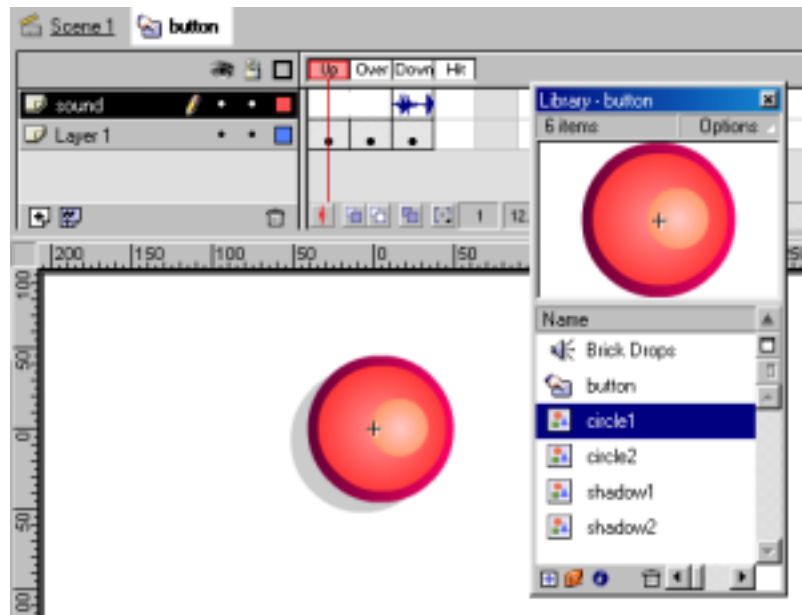
1. Buat file baru dengan memilih **File** —> **New**.
2. Buat symbol graphic "circle1", dan buat gambar lingkaran dengan warna merah.
3. Buat symbol graphic "circle2", dan buat gambar lingkaran dengan warna kuning.
4. Buat symbol graphic "shadow", dan buat gambar lingkaran dengan abu-abu.
5. Buat symbol button dengan nama "button", maka akan muncul editing symbol button.
6. Pada editing symbol, klik frame 1 dan drag symbol "circle1" dari "shadow" dari Library ke dalam stage.
7. Aktifkan frame 2 dengan memilih **Insert** —> **Keyframe**. Drag symbol "circle2" dari Library ke dalam stage. Gambar "circle1" pada frame 2 dihilangkan. Posisi "circle2" dengan shadow diatur sedemikian rupa sehingga bila kursor di atas button, button akan bergerak.



Gambar 2.1 Library



Gambar 2.2 Pembuatan symbol graphic "lingkaran"

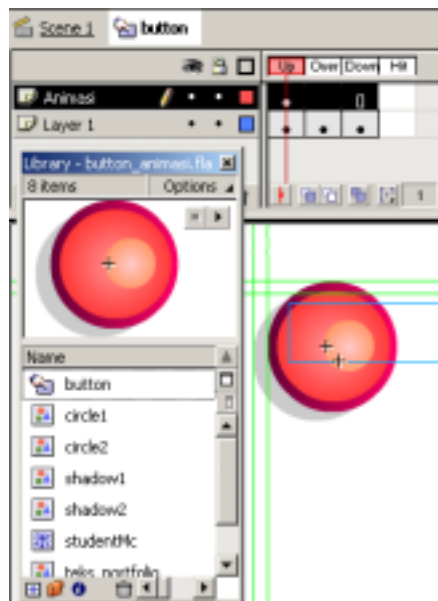


Gambar 2.3 Editing button dan Library

8. Aktifkan frame 3 dengan memilih **Insert** —> **Keyframe**. Posisi "circle2" dengan shadow diatur sedemikian rupa sehingga bila button di-klik, maka button akan bergerak.
9. Buat layer "Sound" untuk menempatkan suara.
10. Ambil file suara dari library yang ada di Flash dengan cara memilih **Window** —> **Common Library** —> **Sounds**. Drag symbol sound ke dalam stage.
11. Pilih **Control** —> **Test Movie** untuk memeriksa hasilnya.
12. Simpan file tersebut dengan nama **button.fla**.

## 2. Button dengan animasi

1. Buka file **button.fla**.
2. Buat symbol graphic "treks\_portfolio". Dalam editing symbol buat teks "portfolio".
3. Seperti nomor 2, buat symbol graphic "teks\_student" yang berisi teks "Student".
4. Buat symbol movie clip "StudentMc". Pada Editing symbol, buat animasi tweening "Student" dan "portfolio".
5. Pada editing symbol "button" buat layer untuk menempatkan animasi.



Gambar 2.4 Button dengan animasi

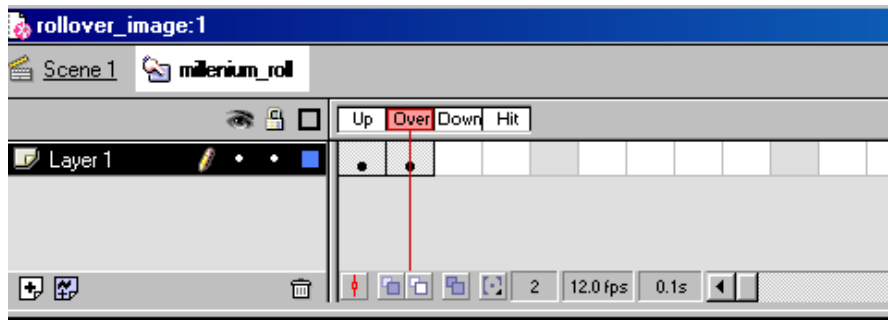
6. Pada layer "Animasi" drag symbol "studentMc" ke stage.
7. Kembali ke Scene 1, kemudian drag symbol button.
8. Simpan file dengan nama **button\_animasi. fla**.
9. Test Movie.

### 3. Rollover image

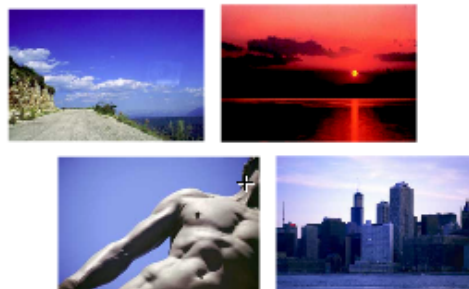
Rollover image dapat dibuat dengan memperlakukannya sebagai button.

#### Untuk membuat rollover image:

1. Buat file baru dengan memilih **File** —> **New**.
2. Buat symbol graphic "millenium", dan buat teks "MILLENIUM" dengan warna biru muda.
3. Buat symbol movie clip "milleniumMc", dan drag symbol graphic "millenium" dari Library ke dalam stage.
4. Buat symbol graphic "penerbit", dan buat teks "Penerbit" dengan warna biru muda.
5. Buat symbol movie clip "penerbitMc", dan drag symbol graphic "penerbit" dari Library ke dalam stage.
6. Buat symbol button dengan nama "millenium\_roll", maka akan muncul editing symbol button.



Penerbit  
MILLENIUM



Gambar 2.5 Editing symbol rollover image

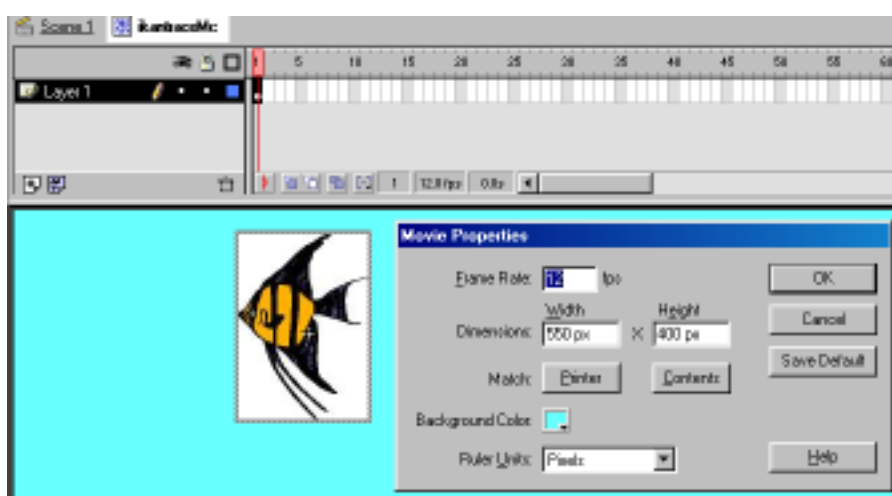
7. Pada editing symbol, klik frame 1 dan drag symbol "penerbitMc" dan milleniumMc" ke dalam stage.
8. Aktifkan frame 2 dengan memilih **Insert** —> **Keyframe**. Posisi teks dengan warna diubah sehingga memberikan efek pada saat kursor di atas teks.
9. Impor file "bluesky.jpg", "chicago.jpg", "david.jpg", dan "sunset.jpg". Gambar-gambar tersebut yang berada di stage dihilangkan, karena lebih baik menggunakan dari Library.
10. Drag gambar-gambar tersebut dari Library, dan tempatkan sedemikian rupa seperti contoh.
11. Kembali ke Scene 1. Drag symbol button "millenium\_roll" ke dalam stage.
12. Pilih **Control** —> **Test Movie** untuk memeriksa hasilnya.
13. Simpan file tersebut dengan nama **rollover\_image fla**.

## 4. Mengubah gambar bitmap menjadi vektor

Gambar bitmap dapat diubah menjadi vektor dengan perintah **Trace Bitmap**. Gambar bitmap dapat digunakan dalam Flash movie dengan menyertakannya dalam library, tetapi tidak demikian dengan gambar vektor, karena gambar vektor bukan merupakan suatu link pada gambar bitmap.

Perintah Trace Bitmap mengkonversi image bitmap menjadi grafik vektor. Bila sudah diubah menjadi vektor, gambar tersebut tidak mempunyai link dengan symbol dalam Library.

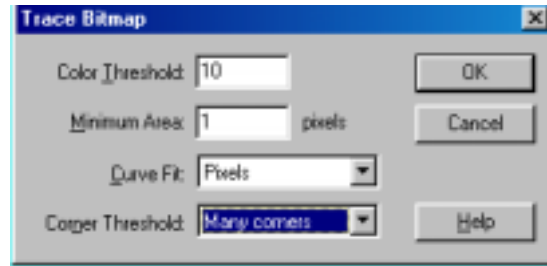
Gambar dengan format bitmap, bila dijalankan pada movie mempunyai garis batas persegi empat dengan warna putih. Warna putih tersebut dapat dihilangkan dengan mengubahnya menjadi format vektor.



Gambar 2.6 Editing symbol movie clip dan kotak dialog Movie Properties untuk mengubah warna background

### Untuk mengkonversi bitmap menjadi vektor

1. Buat file baru dengan memilih **File** —> **New**.
2. Impor file "ikan.jpg" dengan memilih **File** —> **Import**. Pada Library telah masuk symbol "ikan.jpg".
3. Buat symbol movie clip "ikantraceMc". Pada editing symbol, drag symbol "ikan.jpg" dari Library.
4. Background movie dibuat berwarna dengan memilih **Modify** —> **Movie**. Tentukan warna background pada Background Color dari kotak dialog Movie Properties.
5. Klik instance ikan pada stage, kemudian diubah menjadi vektor dengan memilih **Modify** —> **Trace Bitmap**. Masukkan nilai pada kotak dialog Trace Bitmap:



Gambar 2.7 Kotak dialog Trace Bitmap

Color Tracehold : 10  
 Minimum Area : 1 pixel  
 Curve Fit : Pixel  
 Corner Treshold : Many Corner

6. Persegi empat berwarna putih di sekeliling gambar dihilangkan dengan Eraser Tool. Dalam hal ini, ketelitian seseorang untuk mengerjakannya diperlukan.
7. Buat animasi motion tweening yang menggambarkan ikan berjalan dari kanan ke kiri dengan panjang 20 frame.
8. Buat symbol movie clip "ikanMc". Pada editing symbol, drag symbol "ikabn.jpg" ke stage. Kemudian buat animasi motion tweening yang menggambarkan ikan berjalan dari kanan ke kiri dengan panjang 20 frame.
9. Kembali ke Scene 1.
10. Drag symbol "ikanMc" dan "ikantraceMc" ke dalam stage.
11. Pilih **Control** —> **Test Movie** untuk memeriksa hasilnya. Pada movie kelihatan perbedaan antara animasi yang menggunakan gambar ikan asli dengan gambar ikan yang telah diubah menjadi vektor.
12. Simpan file tersebut dengan nama **trace\_bitmap fla**.

# 3

## ACTION DASAR

**Setelah mempelajari bagian ini, diharapkan dapat:**

1. Menggunakan script pada frame
2. Menggunakan script pada button
3. Mengontrol movie clip

Dalam animasi sederhana, Flash hanya mengatur scene dan frame dari movie secara sekuensial. Tetapi dalam movie interaktif, user dapat berinteraksi menggunakan keyboard, mouse, menggerakkan objek, memasukkan data ke dalam form input, dan lain-lain.

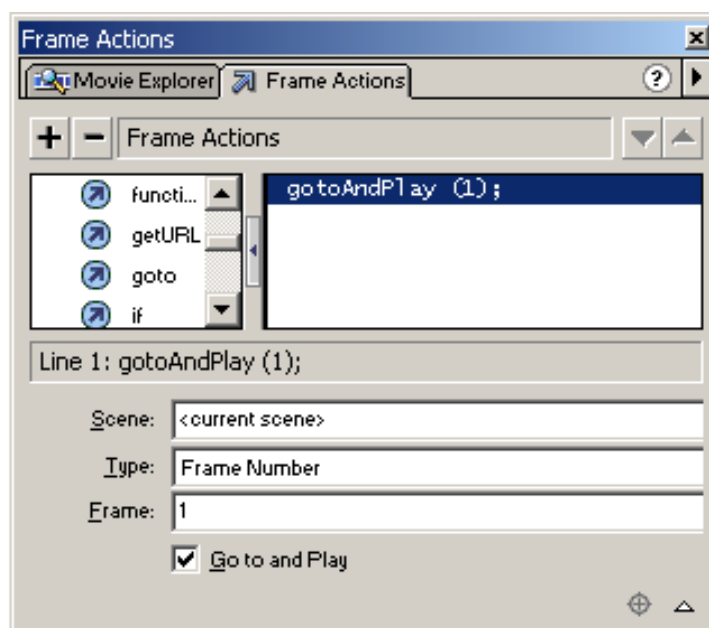
Pembuatan movie interaktif dilakukan dengan mengatur action, yaitu instruksi yang ditulis dalam bahasa pemrograman ActionScript dan dapat berjalan bila suatu event terjadi. Event yang mengendalikan event terjadi bila playhead mencapai frame tertentu, atau user menekan button atau tombol lain pada keyboard. Flash mempunyai dua panel action, yaitu **Normal Mode** dan **Expert Mode**.

## 1. Panel Action

Pada Normal Mode, action dapat dimasukkan tanpa menuliskan script. Namun bagi yang sudah berpengalaman dengan pemrograman, action dapat dituliskan langsung dalam Expert Mode.

### **Normal Mode**

Dalam Normal Mode, pembuatan action dapat dilakukan dengan memilih action pada daftar yang terdapat pada Toolbox list, yaitu di bagian kiri panel. Toolbox list berisi Basic Actions, Actions, Operators, Properties, dan kategori Objek. Kategori Basic Actions berisi action sederhana yang hanya tersedia pada Normal Mode.



Gambar 3.1. Panel Normal Mode

Untuk memilih action dapat dilakukan dengan cara sebagai berikut:

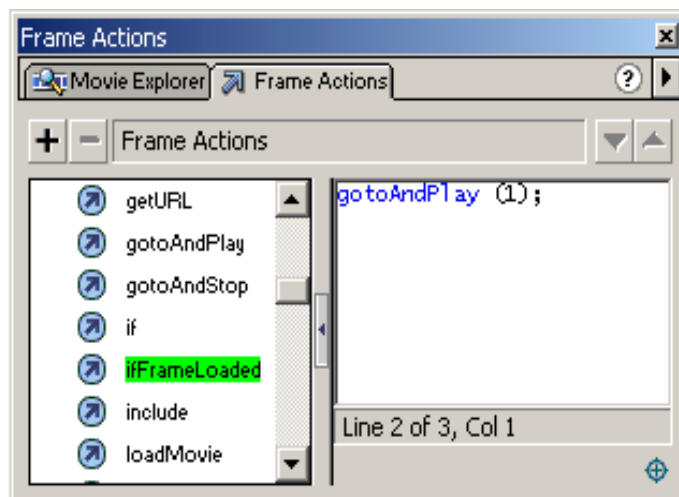
1. Klik kategori Action pada Toolbox untuk menampilkan daftar action.
2. Klik dua kali pada action yang dipilih atau drag action ke dalam Action list ( Action Editor).

Untuk menggunakan kotak parameter:

1. Klik segituga di bawah kanan dari panel Action.
2. Pilih action dan masukkan nilai baru pada kotak parameter.

## Expert Mode

Dalam Expert Mode, action dapat dibuat dengan menuliskan script pada Action list (Text Box) di bagian kanan dari panel. Script dapat diedit atau dihapus seperti menuliskan pemrograman pada Text Editor.



Gambar 3.2. Panel Expert Mode.

Perbedaan Expert Mode dibandingkan dengan Normal Mode:

- Tidak ada kotak parameter.
- Pada panel button, hanya button Add (+) yang berfungsi.
- Button Up dan Down tidak aktif.

## 2. Action Dasar

Action dasar meliputi:

- **Go To** - melompat ke frame atau scene tertentu.
- **Play** dan **Stop** - untuk menjalankan dan menghentikan movie.

- **Stop All Sounds** - untuk menghentikan audio.
- **Get URL** - berpindah ke URL lain.
- **Load Movie, Unload Movie** - menambahkan dan menghilangkan movie.
- **Tell Target** - mengontrol movie lain dan movie clip.
- **On mouse Event** - action bila menggunakan mouse.

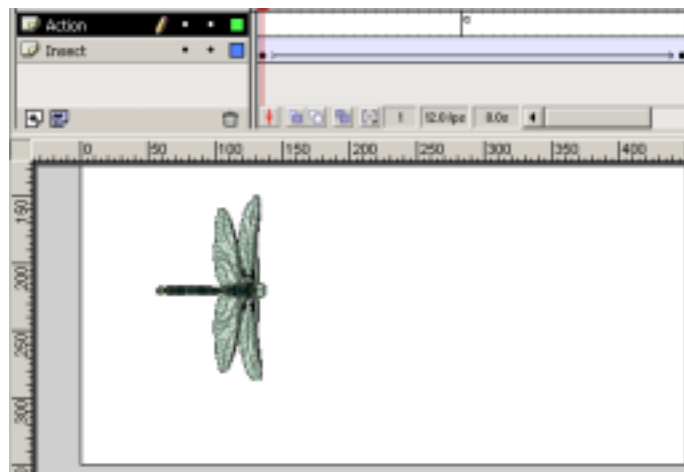
### 3. Go To, Play, dan Stop

#### **Action pada Frame - Stop**

Action Stop digunakan untuk menghentikan movie setelah melewati frame tertentu. Untuk membuat action tersebut dapat dilakukan:

1. Buka file **tweening\_insect.fla**.
2. Buat layer "Action" baru untuk menempatkan action pada frame.
3. Buat keyframe pada frame 20.
4. Pilih **Window** —> **Actions** —> **Normal Mode**.
5. Pada Toolbox list (Text Editor) buat script sebagai berikut:

```
stop();
```



Gambar 3.3. Action stop pada frame.

6. Pilih **Control** —> **Test Movie** untuk memeriksa animasi tweening.
7. Simpan file tersebut dengan nama **frame\_stop.fla**.

Action Go to and Stop digunakan untuk menghentikan movie setelah melewati frame tertentu, dan movie berhenti pada posisi frame yang ditentukan.

**Untuk membuat action pada frame:**

1. Buka file **tweening\_insect fla.**
2. Buat layer "Action" untuk menempatkan action pada frame.
3. Buat keyframe pada frame 20 dan buat script sebagai berikut:  

```
gotoAndStop(1);
```
4. Pilih **Control** —> **Test Movie** untuk memeriksa animasi tweening.
5. Simpan file tersebut dengan nama **frame\_gotoAndStop fla.**

**Action pada Frame - Play**

Action Go to and Play digunakan untuk menjalankan movie melompat ke frame tertentu setelah melewati suatu frame dimana terdapat script.

**Untuk membuat action pada frame:**

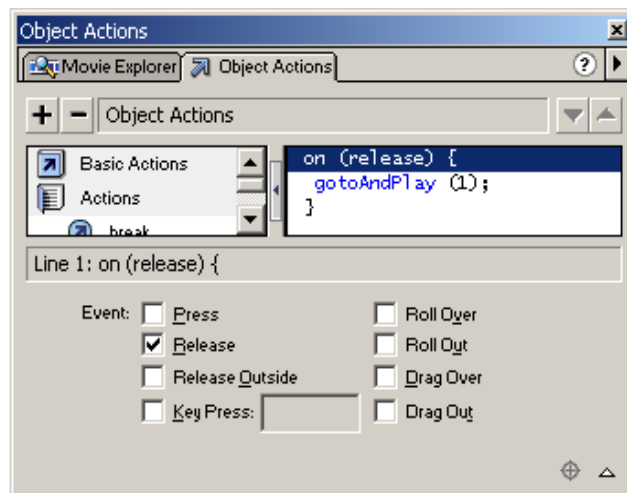
1. Buka file **frame\_gotoandStop fla.**
2. Buat layer "Action" baru untuk menempatkan action pada frame.
3. Modifikasi script pada frame 20 menjadi sebagai berikut:  

```
gotoAndPlay(1);
```
4. Pilih **Control** —> **Test Movie** untuk memeriksa animasi tweening.
5. Simpan file dengan nama **frame\_gotoandPlay fla.**

**Action pada Button - Go to dan Stop**

Action Go to dan Stop digunakan untuk menjalankan dan menghentikan movie. Button yang digunakan untuk mengendalikan movie memerlukan handler yang diawali dengan on dan diikuti dengan event seperti contoh berikut:

```
on (release)
on (press)
on (keypress "<Space>")
on (rollOver)
```



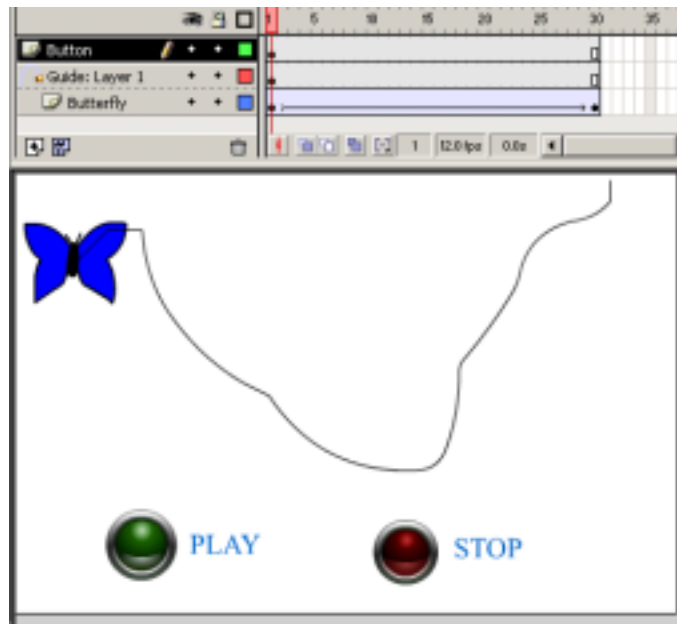
Gambar 3.4 Macam macam handler.

Terdapat beberapa pilihan untuk event handler, yaitu:

- **Press** - untuk action bila tombol mouse ditekan.
- **Release** - untuk action bila tombol mouse dilepas pada saat pointer di atas button.
- **Release Outside** - untuk action bila tombol mouse dilepas pada saat pointer di tidak atas button.
- **Key Press** - untuk action bila tombol keyboard tertentu ditekan.
- **Roll Over** - untuk action bila pointer di atas area button.
- **Roll Out** - untuk action bila pointer keluar dari atas area button.
- **Drag Over** - untuk action bila tombol mouse ditekan bersamaan pada saat pointer di atas button dengan menggeserkan pointer ke dalam area button.
- **Drag Out** - untuk action bila tombol mouse ditekan bersamaan pada saat pointer di atas button dengan menggeserkan pointer ke luar area button.

**Untuk membuat movie yang dikendalikan oleh button:**

1. Buka file **tweening\_path fla.**
2. Buat layer "Button" untuk menempatkan button.
3. Buat button Play dan Stop pada stage dengan memilih dari Common Library.
4. Pilih button Stop, kemudian pilih **Window** —> **Actions** —> **Expert Mode.**



Gambar 3.5. Action play dan stop pada button.

5. Pada Toolbox list (Text Editor) buat script sebagai berikut:
 

```
on (release) {
    stop();
}
```
6. Pilih button Play, kemudian pilih buat script sebagai berikut:
 

```
on (release) {
    gotoAndPlay(1);
}
```

Pada pernyataan gotoAndPlay (1) menunjukkan bahwa movie mulai berjalan pada frame 1.

7. Pilih **Control** —> **Test Movie** untuk memeriksa animasi tweening.
8. Simpan file tersebut dengan nama **button\_stop fla**.

## 4. Tell Target

Movie clip merupakan movie mini yang dapat diberi perintah melakukan apapun. Untuk dapat mengontrol movie clip, movie clip harus diberi nama instance terlebih dahulu .

Syarat untuk dapat mengontrol movie clip:

- Target dalam suatu perintah harus jelas.
- Movie clip yang menjadi target harus mempunyai nama instance.
- Timeline dari movie clip harus berada di stage untuk menjadi target.

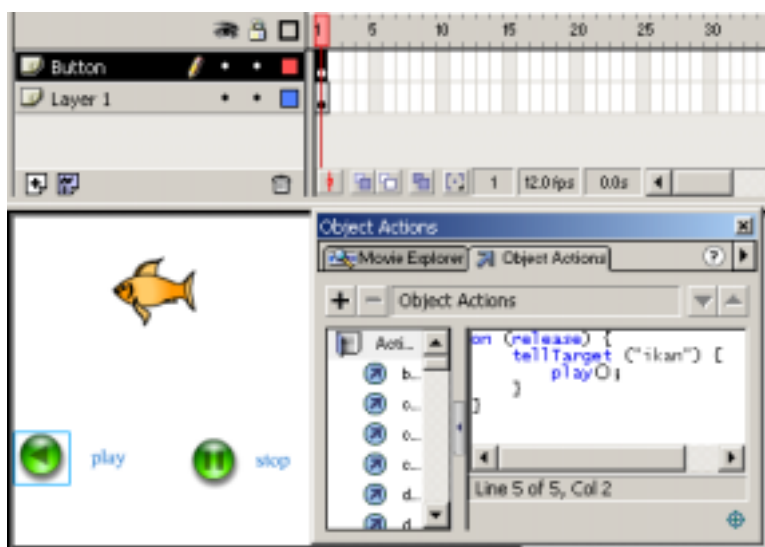
#### Untuk dapat mengontrol movie clip:

1. Buat file baru, kemudian drag simbol movie clip "Fish Movie Clip"
2. Klik pada instance di stage dan pilih **Window** —> **Panels** —> **Instance**. Kemudian berikan nama instance **ikan**.
2. Buat layer "Button" baru untuk menempatkan button.
3. Buat button Play dan Stop pada stage dengan memilih dari Common Library.
4. Pilih button Stop, kemudian pilih buat script sebagai berikut:

```
on (release) {
    tellTarget ("ikan") {
        stop();
    }
}
```

atau

```
on (release) {
    ikan.stop();
}
```



Gambar 3.6. Memberi nama instance.

- Pilih button Play, kemudian pilih buat script sebagai berikut:

```
on (release) {
    tellTarget ("ikan") {
        play():
    }
}
```

atau

```
on (release) {
    ikan.play();
}
```

- Pilih **Control** —> **Test Movie** untuk memeriksa animasi tweening.
- Simpan file tersebut dengan nama **tell\_target fla**.

## 5. Load Movie

Script pada frame maupun button dapat digunakan untuk me-load movie lain (SWF). Movie yang di-load harus berada dalam satu folder dengan Flash movie utama. Untuk dapat me-load movie lain, dapat dilakukan sebagai berikut:

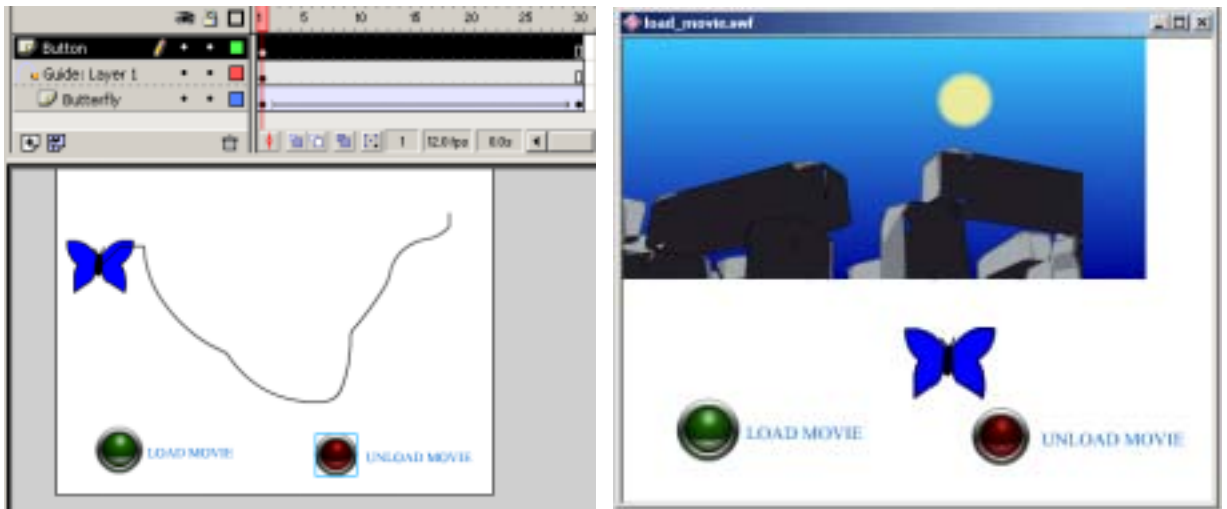
- Buka file **tweening\_path fla**.
- Buat layer "Button" untuk menempatkan button.
- Buat button Load Movie pada stage dengan memilih dari Common Library.
- Buat script pada button sebagai berikut:

```
on (release) {
    loadMovie ("maya.swf", 1);
}
```

- Buat button UnLoad Movie dengan script sebagai berikut:

```
on (release) {
    unloadMovie (1);
}
```

Pada pernyataan loadMovie ("maya.swf", 1) menunjukkan bahwa movie maya swf ditampilkan sebagai tambahan pada movie yang ada, yaitu dengan memberikan level 1. Bila movie yang di-load menggantikan movie yang ada, maka level yang diberikan adalah 0. Pada pernyataan unloadMOvie (1), menunjukkan bahwa movie yang dihilangkan dari tampilan mempunyai level 1.



7. Pilih **Control** —> **Test Movie** untuk memeriksa animasi tweening.
8. Simpan file tersebut dengan nama **button\_stop fla**.

Gambar 3.7 (kiri)  
Layout pada stage.  
Gambar 3.8 (kanan)  
Movie bila dijalankan.

### **Level dan target**

Flash player dapat tersedia pada Web browser karena plug-in. Jika suatu halaman HTML menyertakan plug-in, biasanya membuat referensi ke file SWF yang di-load ke Level 0 dari Flash player. Flash player plug-in dapat menangani 16.000 file SWF dan Flash movie pada suatu saat dengan satu file SWF untuk satu level. Jumlah level terdiri dari Level 0 sampai dengan Level 15999.

## **6. Action pada Movie Clip**

Selain pada objek button, action juga dapat dilakukan pada objek movie clip. Action yang diberikan berada pada handler onClipEvent. Jika action diberikan pada movie clip.

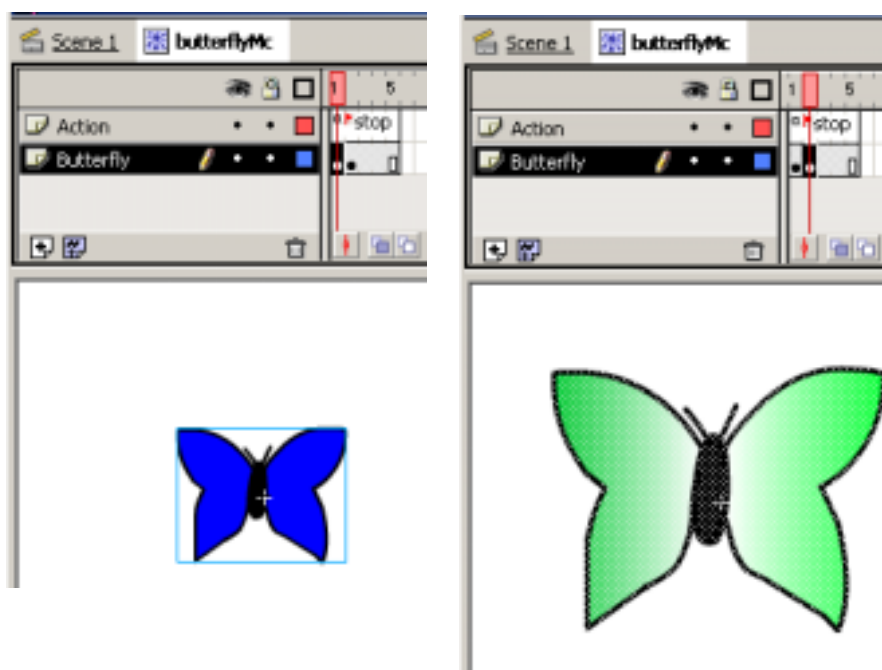
Macam handler yang dapat dibuat adalah sebagai berikut:

- **Unload** - saat movie clip hilang
- **MouseMove** - saat mouse bergerak
- **MouseDown** - saat mouse ditekan
- **MouseUp** - saat mouse dilepas
- **KeyDown** - saat tombol keyboard ditekan
- **KeyUp** - saat tombol keyboard dilepas

Untuk membuat action pada movie clip, pertama tama harus dibuat movie clip terlebih dahulu.

1. Buat file baru.
2. Buat simbol movie clip "butterflyMc".
3. Pada editing simbol "butterflyMc" import file "butterfly.fh8".
4. **Insert** —> **Keyframe** untuk membuat frame 2
5. Pada frame 2, warna diubah, demikian pula ukurannya.
6. Perpanjang frame 2 tersebut menjadi beberapa frame.
7. Buat layer Action, dan buat script pada frame 1:]

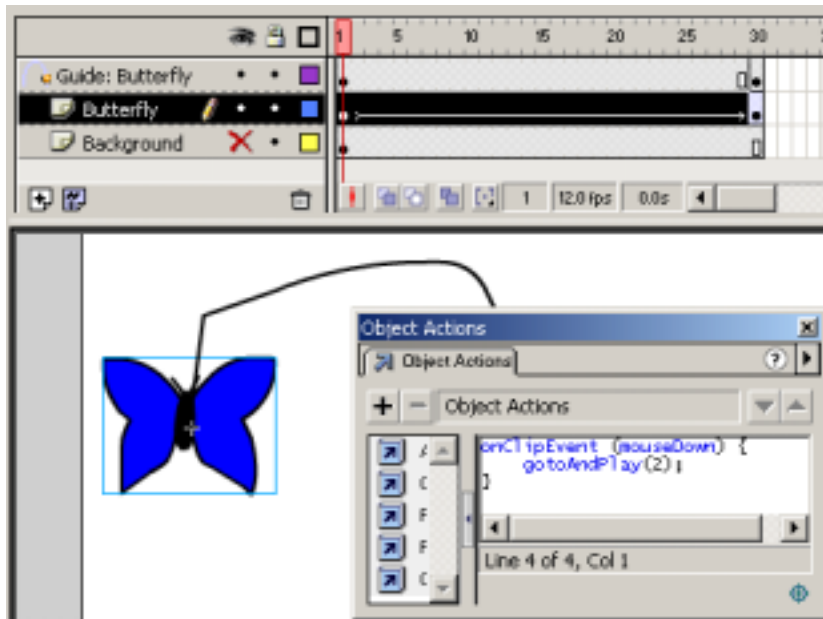
```
stop();
```



Gambar 3.9 Editing simbol "butterflyMc" pada frame 1 (kiri) dan frame 2 (kanan).

8. Kembali ke Scene 1, kemudian drag simbol "butterflyMc" dari Library ke dalam stage.
9. Buat animasi motion tweening dengan path.
10. Klik pada kupu-kupu di stage dan buat scrip sebagai berikut:

```
onClipEvent (mouseDown) {
    gotoAndPlay(2);
}
```



Gambar 3.10. Layout pada stage dengan script pada movie clip

11. Pilih **Control** —> **Test Movie** untuk memeriksa animasi tweening. Bila mouse ditekan, maka movie clip yang dijalankan adalah pada frame 2, sehingga warna dan ukuran gambar kupu-kupu berubah.
12. Simpan file dengan nama **movie\_clip**.

4

# PEMROGRAMAN ACTIONSCRIPT

**Setelah mempelajari bagian ini, diharapkan dapat:**

1. Menggunakan input dan output teks
2. Menggunakan script untuk operasi matematika dan logika
3. Membuat visualisasi grafik dari perhitungan matematika

Aplikasi Flash digunakan untuk menyajikan informasi interaktif, misalnya user dapat memasukkan input data dan memperoleh hasil setelah dilakukan operasi matematika atau logika. Hasil dapat berupa string, numerik, atau movie.

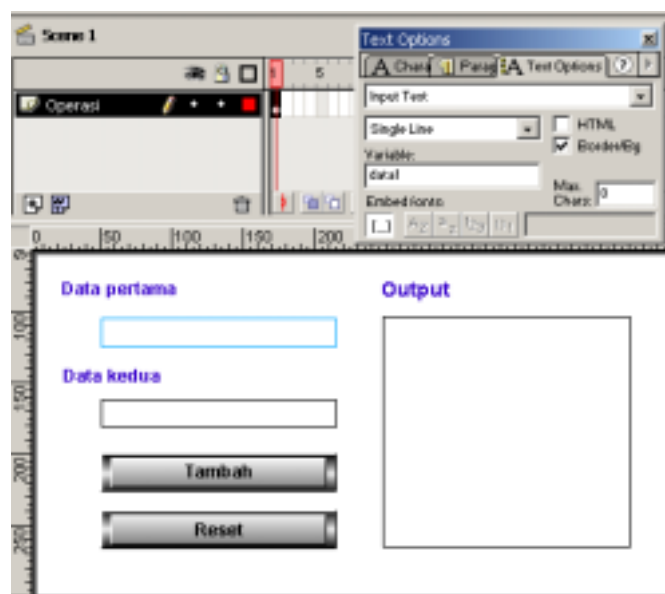
## 1. Operasi Matematika

### Operator Matematika

Operator tambah (+) dapat digunakan untuk tipe data string maupun numerik. Sedangkan operator lain (\*, /, -, dan %) digunakan untuk tipe data numerik.

Untuk membuat movie dengan input data:

1. Buat file baru.
2. Buat kotak dengan Teks Tool pada stage.
3. Buka panel Text Options dengan cara memilih **Window** —> **Panels** —> **Text Options**. Pada kotak dialog tentukan **Input Text**, **Single Line**, **Border/Bg**. Buat nama variabel **data1**.
4. Buat nama label "Data pertama" dengan Static Text.
5. Dengan cara yang sama buat input text dengan nama variabel **data2** dan label "Data kedua".
6. Buat kotak untuk output, dan pada kotak dialog IText Options tentukan **Dynamic Text**, **Single Line**, **Border/Bg**. Buat nama variabel **hasil**.



Gambar 4.1. Layout pada stage dan menentukan nama variabel pada Text Options..

7. Buat button "Tambah" dan "Reset" dengan memilih button dari Common Library.
8. Pilih button "Tambah", kemudian buat script sebagai berikut:

```
on (release) {
    hasil = data1 + data2;
}
```

9. Pilih button "Reset", kemudian buat script sebagai berikut:

```
on (release) {
    data1=" ";
    data2=" ";
    hasil=" ";
}
```

10. Pilih **Control** —> **Test Movie** untuk hasil operasi tambah pada tipe data string.
11. Simpan file tersebut dengan nama **tambah\_string fla.**

Bila operasi tambah digunakan pada tipe data numerik, maka script pada button Tambah diubah menjadi sebagai berikut:

```
on (release) {
    hasil = number(data1) + number(data2);
}
```

12. Pilih **Control** —> **Test Movie** untuk hasil operasi tambah pada tipe data numerik.
13. Simpan file tersebut dengan nama **tambah\_numerik fla.**

## Object Math

Method dari predefined object math digunakan untuk memanipulasi bilangan, seperti `Math.pow`, `Math.abs`, `Math.round`, dan lain-lain.

Sintaks:

```
Math.pow (x, y);
Math.abs(x);
Math.round(x);
```

### Untuk membuat movie dengan menggunakan object Math

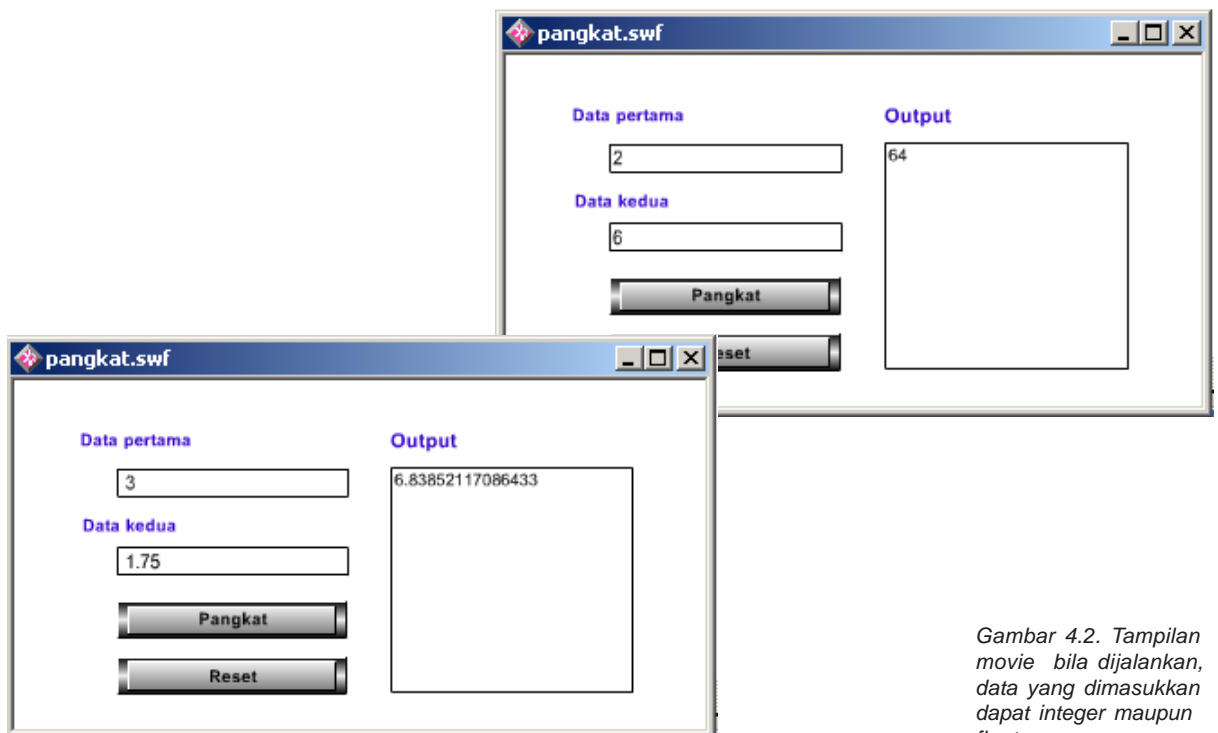
1. Buka file **tambah\_string fla.**
2. Button tambah diganti untuk menghitung operasi dengan object `Math`.

Bila operasi pangkat digunakan pada perhitungan, maka script pada button diubah menjadi sebagai berikut:

```
on (release) {
    hasil = Math.pow(data1, data2);
}
```

3. Pilih **Control** —> **Test Movie** untuk hasil operasi pangkat. Data dapat berupa bilangan integer maupun float.
4. Simpan file tersebut dengan nama **pangkat.fla**.

Hal yang sama dilakukan bila menggunakan object `Math.abs` dan `Math.round`.



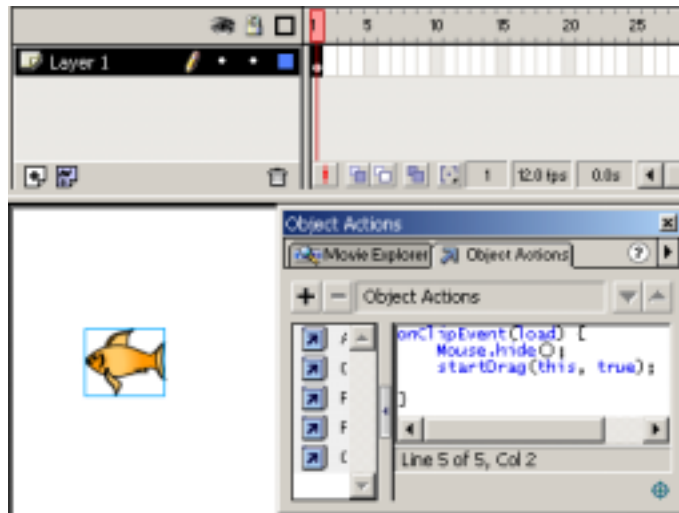
Gambar 4.2. Tampilan movie bila dijalankan, data yang dimasukkan dapat integer maupun float.

## 2. Mouse dan Keyboard Interaktif

### *Mouse untuk mengendalikan movie*

Movie yang sedang berjalan dapat dikendalikan dengan mouse. Method dari object `MovieClip` adalah `show` dan `hide`.

- `show` - menampilkan movie dalam movie
- `hide` - menyembunyikan movie dalam movie



Gambar 4.3. Script pada instance dari movie clip

Untuk dapat menggunakan mouse sebagai pengendali movie dapat dilakukan sebagai berikut:

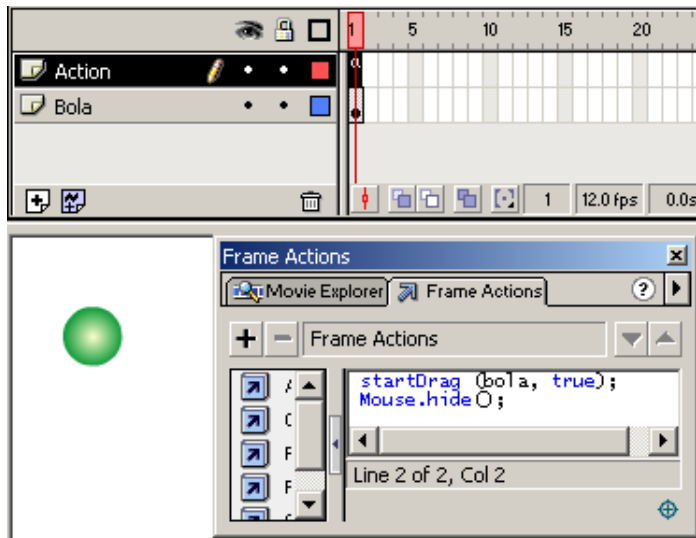
1. Buat file baru.
2. Buka Common Library, kemudian drag salah satu movie clip yang ada di dalamnya ke stage.
3. Klik instance movie clip yang berada di stage, kemudian buat script sebagai berikut:

```
onClipEvent(load) {
    Mouse.hide();
    startDrag(this, true);
}
```

4. Pilih **Control** —> **Test Movie** untuk melihat hasilnya.
5. Simpan file tersebut dengan nama **custom\_mouse1 fla**.

Terdapat cara lain untuk menempatkan script, yaitu pada frame.

1. Buat file baru.
2. Buat simbol graphic dengan nama circle.
3. Buat simbol movie clip dengan nama "circleMc". Pada editing simbol movie clip tersebut buat animasi tweening menggunakan simbol; graphic circle.
4. Drag simbol movie clip "circleMc" ke dalam stage dan beri nama instance **bola**.



Gambar 4.4. Script pada frame untuk mengendalikan movie clip dengan mouse.

5. Buat layer baru, pada frame 1 buat script sebagai berikut:
 

```
startDrag (bola, true);
Mouse.hide();
```
6. Pilih **Control** —> **Test Movie** untuk melihat hasilnya.
7. Simpan file tersebut dengan nama **custom\_mouse2 fla**.

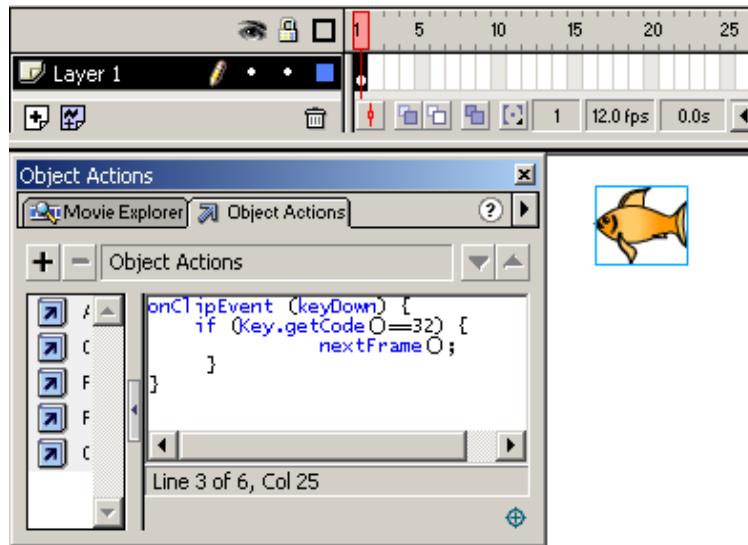
### **Keyboard untuk mengendalikan movie**

Movie yang sedang berjalan dapat dikendalikan dengan keyboard sebagai berikut:

1. Buat file baru.
2. Buka Common Library, kemudian drag salah satu movie clip yang ada di dalamnya ke stage.
3. Klik instance movie clip yang berada di stage, kemudian buat script sebagai berikut:

```
onClipEvent (keyDown) {
    if (Key.getCode()==32) {
        nextFrame();
    }
}
```

Konstanta dengan nilai key code 32 adalah Spcebar.



Gambar 4.4. Script pada instance bila menggunakan tombol keyboard untuk mengontrol movie clip.

4. Pilih **Control** —> **Test Movie** untuk melihat hasilnya.
5. Simpan file tersebut dengan nama **keypressed fla**.

### 3. Pernyataan Kondisi

#### *Pernyataan if*

Pernyataan if memeriksa suatu keadaan dengan menghasilkan nilai true atau false. Bila kondisi terpenuhi, pernyataan tertentu akan dijalankan, tetapi bila tidak, maka pernyataan lain yang dijalankan. Contoh sederhana dapat dibuat untuk menghitung nilai mahasiswa dengan ketentuan sebagai berikut:

Nilai akhir = 10% Kehadiran + 15% Tugas + 25% Midtest +  
50% FinalTest

Untuk membuat Flash movie dapat dilakukan sebagai berikut:

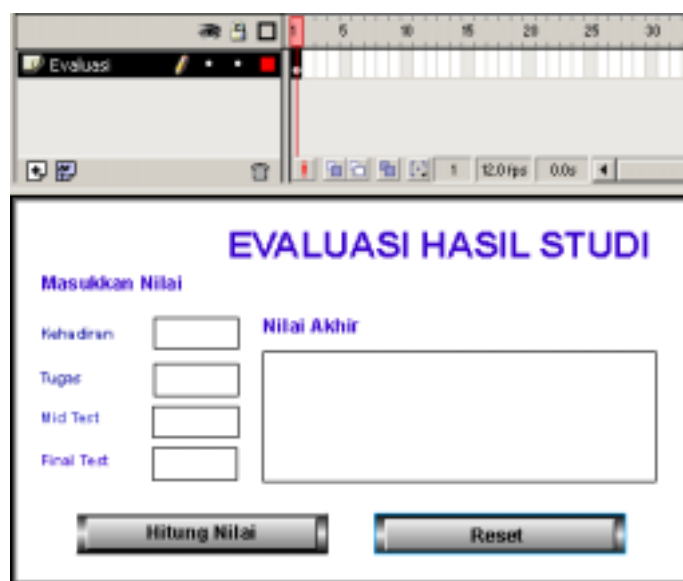
1. Buat file baru.
2. Buat kotak dengan Teks Tool pada stage.
3. Buka panel Text Options dengan cara memilih **Window** —> **Panels** —> **Text Options**. Pada kotak dialog tentukan **Input Text**, **Single Line**, **Border/Bg**. Buat nama variabel **kehadiran**.
4. Buat nama label "Kehadiran" dengan Static Text.
5. Dengan cara yang sama buat input text dengan nama variabel **tugas**, **midtest**, **finaltest** berikut label yang diperlukan".

6. Buat kotak untuk output, dan pada kotak dialog IText Options tentukan **Dynamic Text, Single Line, Border/Bg**. Buat nama variabel **hasil**.
7. Buat button "Hitung Nilai" dan "Reset".
8. Pilih button "Hitung Nilai", kemudian buat script sebagai berikut:

```
on (release) {
    nilai = 0.10 * kehadiran + 0.15 * tugas + 0.25 * midtest +
        0.50 * finaltest;
    if (nilai >= 80.0) {
        hasil = "Nilai akhir: " + nilai + newline + "Predikat A";
    }
    else if (nilai >= 69.0) {
        hasil = "Nilai akhir: " + nilai + newline + "Predikat B";
    }
    else if (nilai >= 56.0) {
        hasil = "Nilai akhir: " + nilai + newline + "Predikat C";
    }
    else if (nilai >= 41.0) {
        hasil = "Nilai akhir: " + nilai + newline + "Predikat D";
    }
    else {
        hasil = "Nilai akhir : " + nilai + newline + "Predikat E";
    }
}
```

9. Pilih button "Reset", kemudian buat script sebagai berikut:

```
on (release) {
    kehadiran = " ";
    tugas = " ";
    midtest = " ";
    finaltest = " ";
    hasil = " ";
}
```



Gambar 4.5. Layout movie menggunakan pernyataan if.



Gambar 4.6. Movie menggunakan pernyataan if bila dijalankan.

10. Pilih **Control** → **Test Movie** untuk melihat hasilnya.
11. Simpan file tersebut dengan nama **statement\_if fla**.

## 4. Operator Logika

Operator logika and, or, xor, nor dapat digunakan pada Flash. Contoh operator && (and) dapat digunakan untuk menghitung diskon dengan ketentuan:

- Untuk pembelian paling sedikit 100 buah dan waktu pesanan lebih dari 10 hari, maka diberikan diskon 10%
- Bila jumlah pembelian paling banyak 10 buah dan waktu pesan kurang dari 5 hari, maka tidak mendapatkan diskon.
- Kondisi lainnya mendapatkan hadiah hiburan.

Untuk membuat Flash movie dapat dilakukan sebagai berikut:

1. Buat file baru.
2. Buat kotak dengan Teks Tool pada stage.
3. Buka panel Text Options dengan cara memilih **Window** → **Panels** → **Text Options**. Pada kotak dialog tentukan **Input Text**, **Single Line**, **Border/Bg**. Buat nama variabel **jumlah**.
4. Buat nama label "Jumlah" dengan Static Text.
5. Dengan cara yang sama buat input text dengan nama variabel **waktu** berikut label yang diperlukan".



Gambar 4.7. Layout movie menggunakan operator and.

6. Buat kotak untuk output, dan pada kotak dialog IText Options tentukan **Dynamic Text, Single Line, Border/Bg**. Buat nama variabel **hasil**.
7. Buat button "Hitung Dsikon" dan "Reset".
8. Pilih button "Hitung Diskon", kemudian buat script sebagai berikut:
 

```
on (release) {
    if ((jumlah >=100) && (waktu >10)) {
        hasil = "Anda mendapat diskon 10%" }
    else if ((jumlah <=10) && (waktu < 5)) {
        hasil = "Anda tidak mendapat diskon"}
    else {
        hasil = "Anda mendapat hadiah hiburan.";
    }
}
```
9. Pilih button "Reset", kemudian buat script sebagai berikut:
 

```
on (release) {
    jumlah = " ";
    waktu = " ";
    hasil = " ";
}
```
10. Pilih **Control** —> **Test Movie** untuk melihat hasilnya.
11. Simpan file tersebut dengan nama **operator\_and fla**.

## 5. Visualisasi grafik dari hasil perhitungan

Hasil perhitungan dapat digunakan untuk membuat visualisasi gambar yang mencerminkan perhitungan tersebut. Contoh yang akan dibuat adalah visualisasi radius pusat gempa dari hasil pemantauan, bila terjadi gempa pertama dan kedua.

Gempa bumi merupakan gerakan bumi yang disebabkan oleh tenaga dalam yang berasal dari bumi. Sumber gempa yang terdapat di dalam bumi disebut **hipocenter**, dan tempat di permukaan bumi di atasnya disebut **epi-center**. Lokasi di permukaan bumi yang mengalami kerusakan paling hebat pada epicenter. Gempa tumbukan hanya mempunyai epicenter. Getaran yang berasal dari hipocenter merambat ke semua arah dalam bentuk gelombang primer **P** dan sekunder **S**. Getaran primer adalah getaran yang pertama kali sampai ke tempat tujuan.

Untuk mengukur kekuatan gempa digunakan seismograph, yang dapat membuat grafik getaran gempa disebut seismogram. Ukuran kekuatan gempa didasarkan pada akibat dari gempa tersebut, dengan satuan skala. Skala yang banyak digunakan pada saat ini adalah skala Richter.

Kekuatan gempa menggunakan skala Richter dengan perbedaan 1 berarti bahwa 10 kali lebih kuat dari pada gempa yang skalanya lebih kecil. Contohnya gempa dengan kekuatan 5 pada skala Richter berarti 10 kali lebih kuat dari pada gempa dengan kekuatan 4 pada skala Richter. Gempa bumi dengan kekuatan 8 pada skala Richter berarti lebih kuat 10 x 10 dari pada gempa dengan kekuatannya 6 pada skala Richter.

Untuk menentukan lokasi epicenter dapat digunakan rumus Laska sebagai berikut

$$d = ((s-p) - 1') * 1000 \text{ km}$$

dengan parameter waktu datangnya getaran sekunder **s** dan waktu datangnya getaran primer **p**. Contohnya, bila getaran pertama gempa datang pada jam 07. 50'. 31" dan getaran gempa yang kedua datang pada jam 07. 57'.42", maka jarak epicenter dapat diperoleh

$$\begin{aligned} d &= ((s-p)-1') * 1000, \\ &= ((07. 57'. 43'' - 07. 50'. 31'') - 1) * 1000 \\ &= (7' 12'' - 1') * 1000 \\ &= 6'. 12'' * 1000 = (6 + 12/60) * 1000 \\ &= 6200 \text{ km} \end{aligned}$$

Contoh penggunaan variabel dapat dijelaskan, bila datangnya getaran pertama pada jam 08. 30'. 45" maka dituliskan  $p_j = 8$ ,  $p_m = 30$ , dan  $p_d = 45$ . Bila getaran kedua datang pada jam 10. 24' 57" maka dituliskan  $s_j = 10$ ,  $s_m = 24$ , dan  $s_d = 57$ .

## Perhitungan jarak pusat gempa

1. Buat file baru.
2. Buat kotak input dengan nama variabel "pj", "pm", "pd", "sj", "sm", dan "sd".
3. Berikan keterangan di atas keenam kotak tersebut dengan membuat static text "Masukkan waktu getaran pertama", "jam", "menit", "detik" untuk input data waktu getaran pertama. Demikian juga untuk waktu getaran kedua. dan "Masukkan waktu getaran kedua".
4. Buat kotak dengan Text Tool untuk menempatkan output. Pada Text Options, pilih Dynamic Text dengan memasukkan nama variabel "d". Berikan keterangan "jarak epicenter" untuk kotak tersebut.
5. Buat judul dari movie "JARAK DARI EPICENTER".
6. Buat dua buah button dengan mengambil dari Common Libraries. Tuliskan teks "Hitung jarak", dan "Reset" untuk button tersebut.
7. Klik pada button "Hitung jarak", dan pilih **Window** -----> **Actions**. Kemudian pilih Expert Mode, dan buat script pada Action list sebagai berikut:

```
on (release) {
    d = (((sj-pj)*60 + (sm-pm) + (sd-pd)/60)-1) *
        1000;
}
```

8. Klik pada button "Reset", kemudian tuliskan script sebagai berikut:

```
on (release) {
    pj = " ";
    pm = " ";
    pd = " ";
    sj = " ";
    sm = " ";
    sd = " ";
    d = " ";
}
```

**Getaran pertama**

jam	menit	detik
<input type="text" value="7"/>	<input type="text" value="50"/>	<input type="text" value="31"/>

**Getaran kedua**

jam	menit	detik
<input type="text" value="7"/>	<input type="text" value="57"/>	<input type="text" value="43"/>

**Jarak dari epicenter**

<input type="text" value="6200"/>	km
-----------------------------------	----

**Hitung Jarak**

Gambar 4.8. Hasil perhitungan bila movie dijalankan.

## Visualisasi

Langkah selanjutnya adalah membuat visualisasi dari perhitungan yang diperoleh. Karena pengamatan hanya dilakukan pada satu titik lokasi yang mencatat waktu gempa, maka lokasi epicentrum ditemukan merupakan lingkaran. Visualisasi memperlihatkan radius di mana epicentrum berada.

Pertama-tama dibuat lingkaran kecil, yang ditempatkan pada lokasi pengamatan ditentukan pada koordinat dalam satuan pixel (400, 200). Pada titik tersebut ditempatkan lingkaran kecil sehingga tidak terlihat sebagai lingkaran. Lingkaran akan diskala sesuai dengan perhitungan jarak epicenter yang diperoleh. Variabel yang digunakan untuk skala adalah scaling factor :

$s_x$  = scaling factor menurut sumbu x

$s_y$  = scaling factor menurut sumbu y

Bila hasil jarak dengan epicenter telah ditemukan, maka  $s_x$  dan  $s_y$  diberikan nilai sama yaitu sebesar  $d$ . Untuk pembuatan visualisasi dengan ActionScript, dapat dilakukan langkah-langkah berikut.

1. Buat simbol graphic "lingkaran". Pada editing simbol buat lingkaran.
2. Buat simbol movie clip "lingkaranMc".
3. Pada editing symbol buat layer pertama "Lingkaran". Drag grafik "lingkaran" ke dalam stage. Perpanjang frame pada Timeline dibuat 35.
4. Buat layer "Action", kemudian pada frame 35 berikan action gotoAndPlay(1).
5. Buat garis horisontal dan vertikal untuk koordinat pada titik pengamatan gempa. Titik pusat koordinat dipilih lokasi pixel (400, 200). Buat keterangan yang diperlukan untuk informasi koordinat, seperti "x", "1000", "2000" dan "lokasi pengamatan".
6. Buat button dengan mengambil dari Common Libraries. Tuliskan teks "Visualisasi".
7. Klik pada button "Visualisasi" dan buat script sebagai berikut:

```
on (release) {
    sx = d;
    setProperty ("target_lingkaran", _xscale, (sx/10));
    sy = d;
    setProperty ("target_lingkaran", _yscale, (sy/10));
}
```

8. Klik pada button "Reset", dan tambahkan script sehingga menjadi sebagai berikut:

```

on (release) {
    pj = " ";
    pm = " ";
    pd = " ";
    sj = " ";
    sm = " ";
    sd = " ";
    d = " ";
    setProperty ("lingkaran_gempa", _xscale, "10");
    setProperty ("lingkaran_gempa", _yscale, "10");
}

```

9. Pilih **Control** —> **Test Movie** untuk melihat hasilnya.
10. Simpan file tersebut dengan nama **gempa fla**.



Gambar 4.9.  
Visualisasi jarak pusat gempa bila movie dijalankan.

5

# FILE TEXT EKSTERNAL

**Setelah mempelajari bagian ini, diharapkan dapat:**

1. Menggunakan file text eksternal
2. Menggunakan file text eksternal dengan lebih dari satu variabel

## 1 File text eksternal

Variabel yang dimasukkan ke dalam Flash dari file text eksternal atau dihasilkan oleh script dari server merupakan cara yang baik untuk membuat Flash movie dinamis dan interaktif. Penggunaan variabel eksternal memudahkan user untuk mengubah data numerik ke dalam text dan sebaliknya dengan cepat, tanpa membuka Flash terlebih dahulu untuk menggunakan file SWF yang baru.

Format kode URL merupakan penulisan standar untuk variabel yang dikirimkan melalui internet. Berikut ini akan dibahas bagaimana memformat variabel dan memasukkan variabel dari file text eksternal, di mana meletakkan variabel, dan bagaimana memeriksa hal itu sudah terlaksana dengan baik.

### **Membuat format variabel dalam file text**

Cara paling mudah untuk memasukkan variabel ke dalam Flash, yaitu dengan mengetiknya sebagai file text menggunakan editor teks, kemudian menggunakan perintah **loadVariables** dalam Flash sesuai dengan nama file tersebut. Editor teks yang umum digunakan pada PC adalah Notepad, sedangkan pada Macintosh menggunakan SimpleText. Perintah **loadVariables** mengarahkan Flash ke file text dan mencoba menggunakan apa yang terdapat di dalamnya.

Spasi antara baris yang dibuat menggunakan enter akan menghasilkan tampilan spasi lebih lebar dari yang dikehendaki. Kelemahan tersebut dapat diatasi dengan mengaturnya pada Flash. Untuk mengatur tampilan teks pada Flash movie dapat menggunakan panel Paragraph, yang dapat digunakan untuk mengatur spesifikasi teks seperti alignment, indent, dan spasi antara baris.

File text eksternal untuk menyajikan informasi, dimana data terpisah dari file Flash. Dengan demikian editing teks dapat dilakukan tanpa pada editor text, seperti Notepad.

### **Perintah loadVariables**

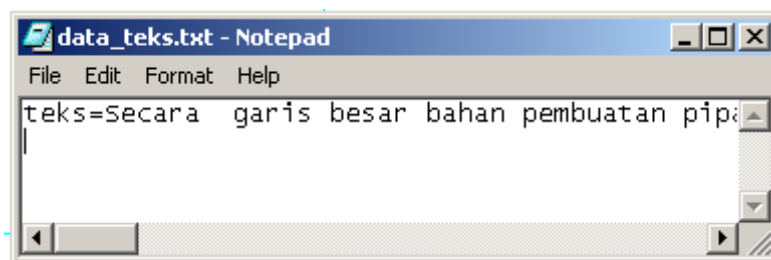
Untuk mendapatkan proses seluruhnya dari loading (dan pengiriman) variabel, perlu digunakan **loadVariables**. Parameter **level** atau **target** mengatur di mana variabel akan ditentukan dan kapan digunakan. Sintaks **loadVariablesNum** dengan parameter **level** secara otomatis mencari nomor level dalam Flash dimana variabel akan diletakkan.

Nilai **0** untuk **level** membuat Flash me-load variabel apapun dalam file text ke Level 0, yang pada dasarnya ekuivalen dengan timeline utama, kecuali perintah movie pada level lain. Loading pada Level 0 membuat variabel dapat diakses dari manapun dengan Flash menggunakan nama variabel **\_root**.

## 1. Menampilkan informasi dari text eksternal

### Membuat file text

1. Buat file text dengan editor Notepad sebagai berikut:  
 teks=Secara garis besar bahan pembuatan pipa dalam dunia perminyakan (oil industry) digunakan jenis logam dari baja, karena kuat, mudah pengolahannya, dan mudah dilas untuk menyambung satu pipa dengan pipa lainnya sehingga merupakan rangkaian/string tertentu untuk suatu tujuan.
2. Simpan file dengan nama **dataTeks.txt**.



Gambar 5.1. File text dengan nama variabel teks.

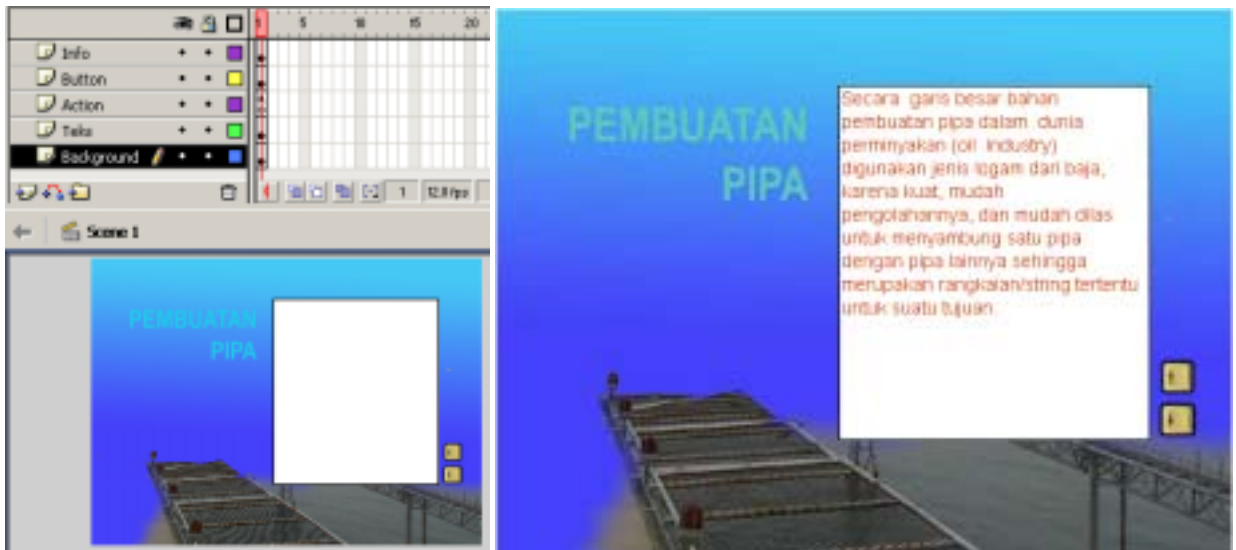
### Membuat Flash movie

1. Buat file baru.
2. Pada layer pertama "Background" import atau drag gambar untuk background.
3. Buat layer baru "Teks", kemudian buat kotak teks dengan Text Tool, pilih **Dynamic Text, Multiline Border/Bg, Word wrap**, dan beri nama variabel **teks**.
4. Buat layer ketiga "Actions", kemudian buat action pada frame sebagai berikut:

```
loadVariablesNum ("data_teks.txt", 0);
```

5. Buat layer keempat "Button", kemudian buat dua buah button untuk menscroll teks ke atas dan ke bawah. Script pada button up dibuat sebagai berikut:

```
on (release) {
    teks.scroll = teks.scroll + 1;
}
```



Pada button down buat script sebagai berikut:

```
on (release) {
    teks.scroll = teks.scroll - 1;
}
```

Gambar 5.2 (kiri) Layout pada Stage dan Timeline. Gambar 5.3 (kanan) Movie bila dijalankan.

6. Buat layer kelima "Info", kemudian buat animasi tweening "PEMBUATAN PIPA"
7. Simpan file dengan nama **scrolling\_teks fla** dalam satu direktori dengan file **data\_teks.txt**.
8. Pilih **Control** —> **Test Movie** untuk melihat hasilnya.

## 2. Text eksternal dengan lebih dari satu variabel

### Membuat file text

1. Buat file **gallerydata.txt** dengan editor teks sebagai berikut:
2. Buat nama variabel **profil**, **produk**, **spesial**, dan **kontak**. Setiap variabel mempunyai beberapa teks yang nilainya berhubungan dengan data yang akan ditampilkan pada movie. Tuliskan pada editor teks sebagai berikut:

```
&profil=TOPAZ Art Gallery berdiri sejak tahun 1990, dan bergerak dalam bidang jasa seni. Telah terdaftar 3000 anggota terdiri dari profesional di bidang semua bidang art termasuk grafik dan multimedia&
```

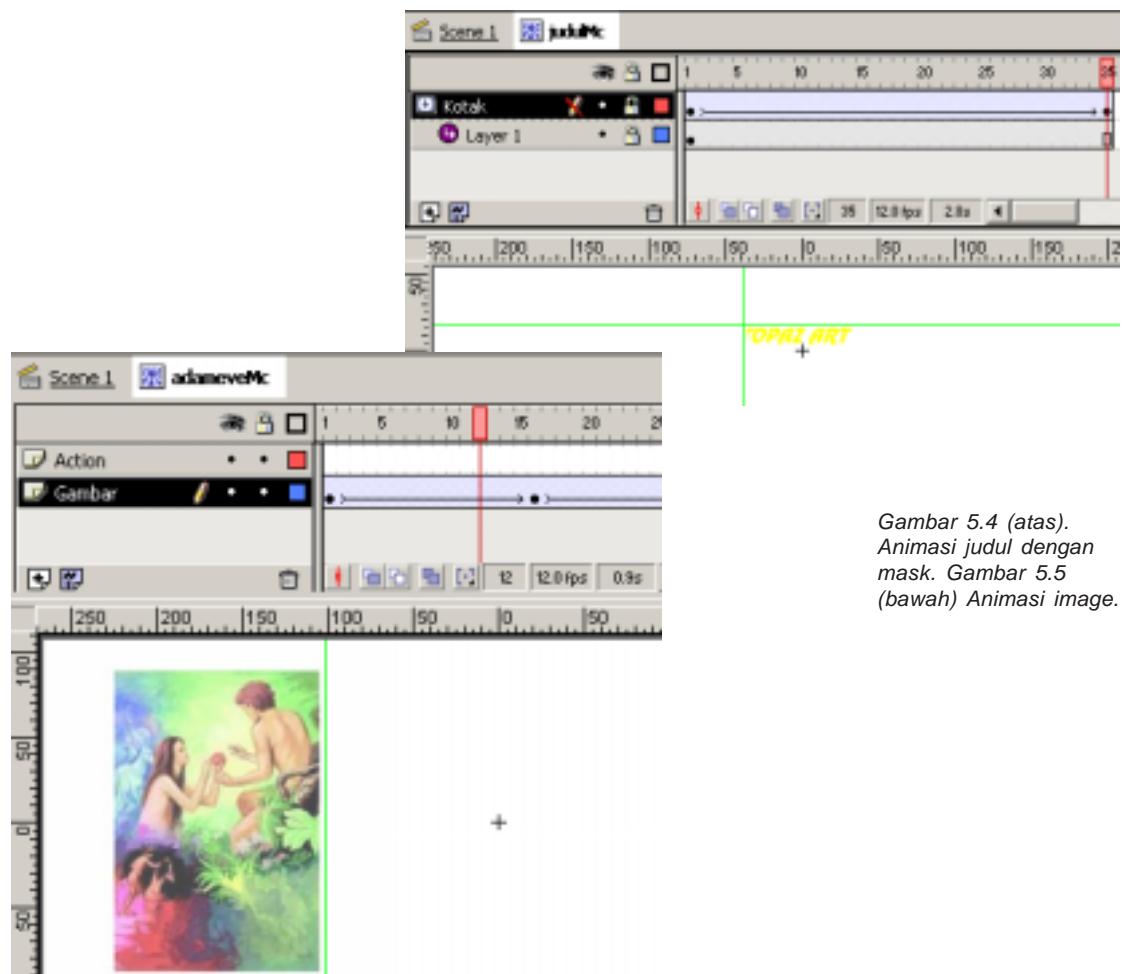
&produk=Karya seni yang disediakan oleh TOPAZ ART adalah lukisan, desain grafis, animasi, dan lain-lain&

&spesial=Karya khusus untuk bulan ini adalah digital painting. Digital painting adalah lukisan yang dibuat menggunakan alat bantu komputer&

&kontak=TOPAZ ART Gallery dibuka mulai jam 09.00 sampai jam 21.00 setiap hari. Informasi dengan email: info@monalisa.com&

## Membuat Flash movie

1. Buat file baru.
2. Buat simbol movie clip "judulMc" yang berisi animasi judul dengan mask.
3. Buat simbol movie clip "gambarMc" yang berisi animasi tweening suatu image.



Gambar 5.4 (atas). Animasi judul dengan mask. Gambar 5.5 (bawah) Animasi image.

4. Kembali ke Scene 1, buat layer "Output" untuk menempatkan field teks yang akan menampilkan data.
5. Gunakan Text Tool untuk membuat field dynamic text dengan nama variabel **displayteks**. Gunakan panel Character untuk mengatur karakter dan paragraph yang akan ditampilkan.
6. Buat layer "Title", kemudian drag simbol movie clip "JudulMc" ke dalam stage.
7. Buat layer "ActionText" untuk menempatkan script pada frame 1 sebagai berikut:

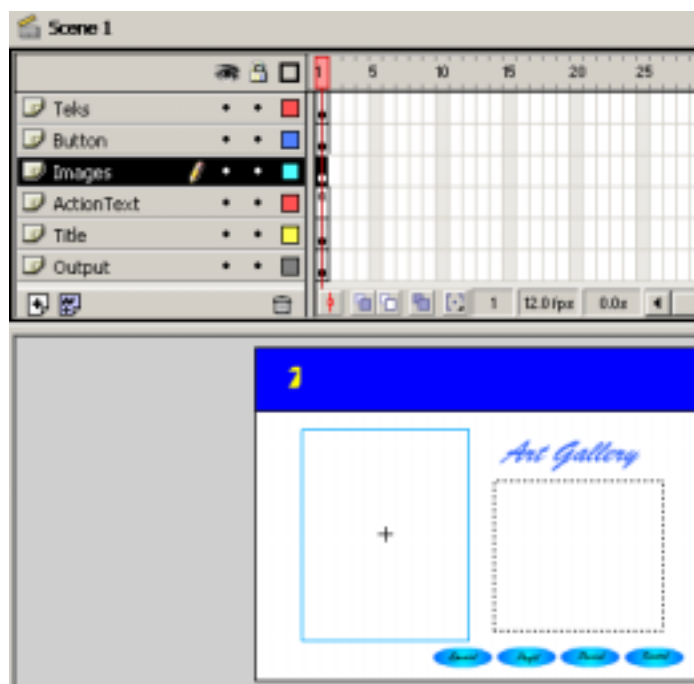
```
loadVariablesNum ("gallerydata.txt",0);
```

Script ini akan menyebabkan Flash me-load ke semua variabel yang telah ditentukan pada file text.

8. Buat layer "Image", kemudian drag simbol movie clip "gambarMc" dari Library. Buat panjang frame menjadi 2 frame.
9. Buat layer "Button", kemudian buat button "profil" dengan script seperti berikut:

```
on (press){
    displayteks=profil;
}
```

Script tersebut akan membuat field dynamic text yang telah dibuat untuk menampilkan data teks yang berada pada file text **gallerydata.txt**.



Gambar 5.6. Layout pada Stage dan Timeline.

10. Lakukan cara yang sama dengan nomor 9, kemudian buat tiga buah button, yaitu "produk", "spesial", dan "kontak".

11. Klik pada button Produk dan buat script sebagai berikut:

```
on (press){  
    displayteks=produk;  
}
```

12. Klik pada button Kontak dan buat script seperti berikut:

```
on (press){  
    displayteks=kontak;  
}
```

13. Klik pada button Spesial dan buat script seperti berikut:

```
on (press){  
    displayteks=spesial;  
}
```

14. Pilih **Control** —> **Test Movie** untuk hasil operasi tambah pada tipe data string.

15. Simpan file tersebut dengan nama **artgallery fla** satu folder dengan file text **gallerydata.txt**.



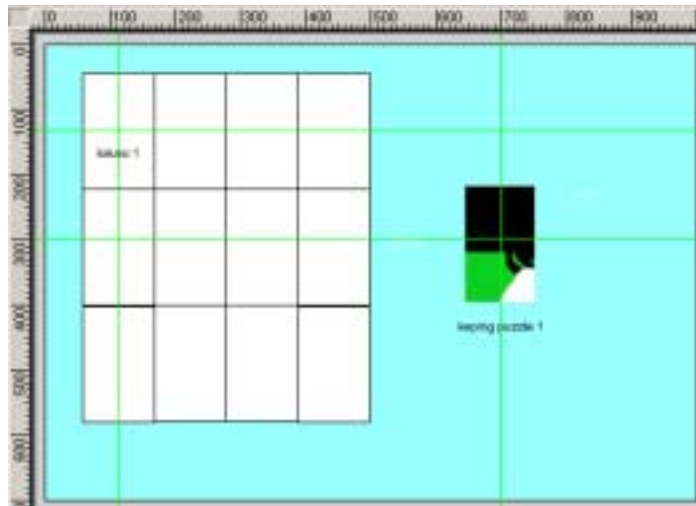
# GAME

**Setelah mempelajari bagian ini, diharapkan dapat:**

1. Memahami pembuatan game puzzle
2. Memahami pembuatan timer untuk game
3. Mendeteksi tabrakan sebagai dasar game animasi

## 1. Puzzle

Puzzle merupakan game yang sederhana, yaitu menggeser keping puzzle ke dalam suatu lokasi. Bila lokasi sesuai dengan ketentuan, maka keping puzzle tersebut dapat diletakkan, tetapi keping puzzle kembali ke posisi semula bila tidak sesuai. Dengan demikian harus ditentukan koordinat dari masing-masing lokasi. Dalam puzzle yang menggunakan banyak keping puzzle, perlu ditentukan koordinat semua lokasi tersebut.



Gambar 6.1 Konsep pembuatan puzzle

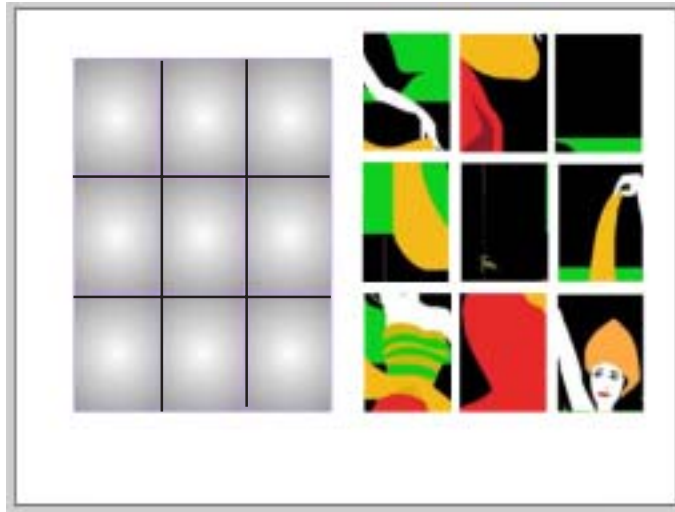
### Scene 2

Untuk membuat game puzzle, dapat dilakukan dengan mengikuti langkah-langkah berikut:

1. Buka file **source\_puzzle.fla** yang terdapat dalam CD pada direktori **game**. Dalam Library terdapat beberapa simbol image "model.jpg", "puzzle1.jpg", ..., "puzzle9.jpg", simbol graphic "tekspuzzle", "indikator", "teksgameOver", dan "display", simbol movie clip "judulMc", "game\_overMc" dan simbol audio "Giggle".

### Membuat keping puzzle

2. Buat simbol graphic "puzzleGraf\_01". Pada editing simbol graphic drag simbol image "puzzle1.jpg".
3. Buat simbol movie clip "puzzleMc\_01". Pada editing simbol movie clip drag simbol graphic "puzzleGraf\_01". Dalam panel Instance, pada field Behavior pilih "Button", dan pada field Option pilih "Track as Button". Pilihan ini akan memungkinkan user dapat menggeser keping puzzle dengan mouse.
4. Ulangi nomor 2 dan 3 untuk membuat simbol movie clip "puzzleMc\_02", ... "puzzleMc\_09".



Gambar 6.2 Layout keping dan papan puzzle

5. Kembali ke Scene 1, pada layer "Puzzle" drag semua simbol movie clip "puzzleMc\_01", ... "puzzleMc\_09" ke stage.

### Membuat papan puzzle

6. Buat simbol graphic "lokasi". Pada editing simbol graphic buat kotak yang ukurannya sama dengan ukuran keping puzzle, karena akan digunakan untuk menempatkannya.
7. Buat simbol movie clip "lokasiMc", kemudian drag simbol graphic "lokasi" ke dalam editing simbol.
8. Kembali ke Scene 1, kemudian drag simbol movie clip "lokasiMc" ke stage dan berikan nama instance "lokasi\_01".
9. Dengan cara yang sama buat instance "lokasi\_02", ... "lokasi\_09". Letakkan sedemikian semua instance lokasi sedemikian rupa sehingga membentuk papan puzzle.

### Membuat timer

10. Buat simbol movie clip "timemeterMc". Pada layer pertama "Grafik" drag simbol graphic "display" dari Library. Buat panjang gambar 35 frame.
11. Buat layer "Indikator", kemudian drag simbol graphic "Indikator" dan buat animasi tweening sehingga warna dari frame 1 ke frame 35 berubah.
12. Buat layer "Kotak", kemudian buat animasi tweening berupa gambar kotak dan pada frame terakhir kotak tersebut menutupi seluruh indikator.
13. Dengan layer "Kotak" aktif, klik kanan kemudian pilih Mask.



Gambar 6.3 Editing simbol "timemeterMc".

14. Buat layer "Action", kemudian buat action **stop()** pada setiap frame.
15. Buat layer "Suara" , kemudian drag symbol audio "Giggle" dari Library.

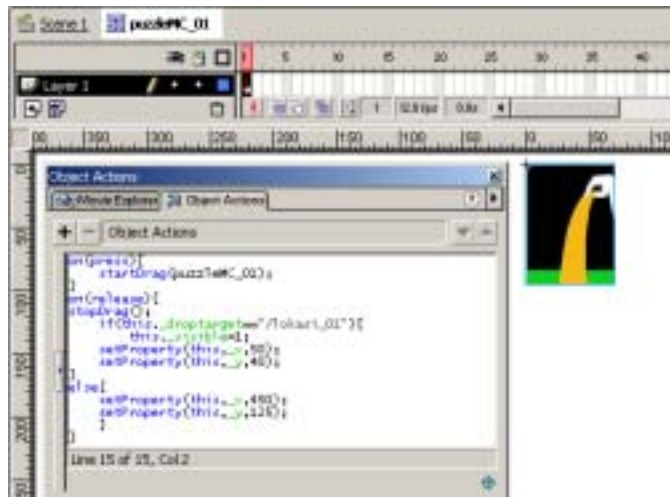
### Membuat script pada keping puzzle

16. Pada simbol movie clip "puzzleMc\_01" klik gambar keping puzzle, kemudian buat script sebagai berikut:

```

on(press){
    startDrag(puzzleMC_01);
}
on(release){
    stopDrag();
    if(this._droptarget=="lokasi_01"){
        this._visible=1;
        // bila sesuai keping puzzle berhenti
        // pada kotak lokasi dengan tepat
        setProperty(this,_x,50);
        setProperty(this,_y,40);
    }
    else{
        // bila tidak sesuai, maka keping puzzle
        // kembali ke tempat semula
        setProperty(this,_x,450);
        setProperty(this,_y,125);
    }
}

```

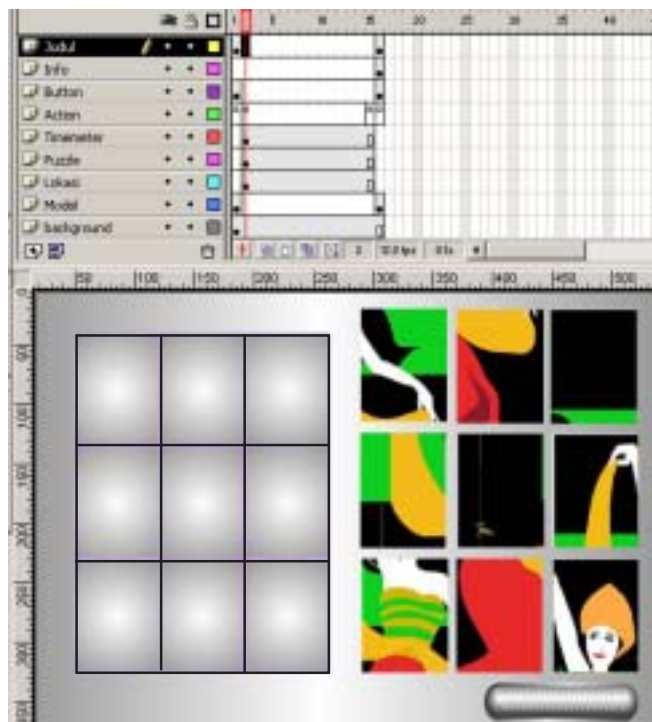


Gambar 6.4 Script pada keping puzzle

17. Dengan cara yang sama pada nomor 16, buat script untuk setiap movie clip dengan diperhatikan koordinat kotak lokasi dan posisi keping puzzle di stage.

### Mengatur timeline utama

18. Buat layer pertama "Background", kemudian buat kotak dengan warna gradasi yang panjangnya 16 frame.
19. Pada layer "Lokasi" dan "Puzzle" buat panjang dari frame 2 sampai 15.



Gambar 6.5 Layout pada timeline utama

20. Buat layer "Timemeter", kemudian drag simbol movie clip "timmerMc" ke stage, dari frame 2 sampai frame 15.
21. Buat layer "Model", kemudian drag simbol image "model.jpg" ke stage pada frame 1 dan 16.
22. Buat layer "Action", kemudian buat script pada:

frame 1:

```
timelimit = 0;
time_meter.gotoAndStop (1);
stop ();
```

frame 2:

```
timelimit = timelimit+1;
time_meter.nextFrame ();
if (timelimit>=36) {
    gotoAndPlay (16);
}
else {
    gotoAndPlay (2);
}
```

frame 15:

```
gotoAndPlay (2);
```

frame 16:

```
stop();
```

### Scene 1 dan 3

Scene pertama ini merupakan tampilan awal untuk masuk ke dalam sistem yang berisi animasi judul, model puzzle, dan button Start.

23. Buat layer "Button", kemudian buat button "Start" pada frame 1 dan 16 dengan script sebagai berikut:

```
on (release) {
    gotoAndPlay (2);
}
```

24. Buat layer "Info", kemudian pada frame 16 drag simbol movie clip "game\_overMc" ke dalam stage.
25. Buat layer "Judul", kemudiam pada frame 1 dan 16 drag simbol movie clip "judulMc".
26. Simpan file dengan nama **puzzle.fla**, kemudian **Test Movie** untuk menjalankan puzzle tersebut.

## 2. Mendeteksi Tabrakan

Method dari objek movie clip hitTest digunakan untuk mendeteksi tabrakan, dan memeriksa apakah suatu objek bertabrakan dengan movie clip.

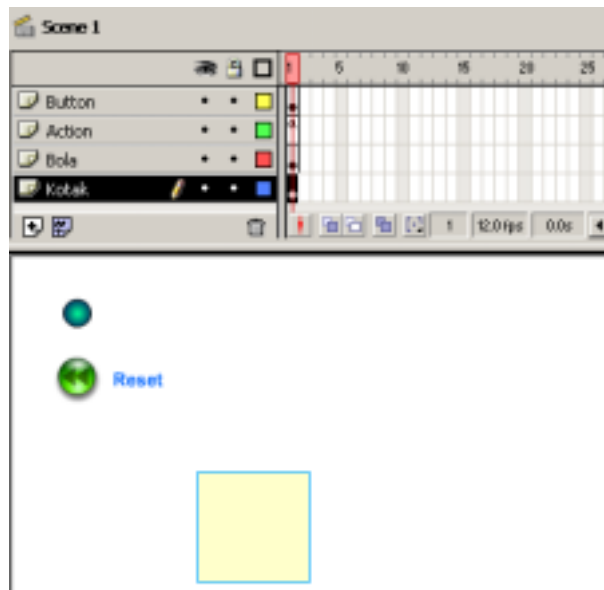
### *Membuat Flash movie*

1. Buat file baru.
2. Buat simbol movie clip **kotakMc**. Pada editing simbol buat gambar persegi empat, kemudian buat animasi tweening motion sampai frame 10. Buat layer Action, kemudian buat action stop() pada frame 1 dan 10. Animasi ini dibuat untuk mendapatkan gerakan bila terjadi tabrakan.
3. Buat simbol movie clip **bolaMc**. Pada editing simbol buat gambar lingkaran.
4. Klik Scene 1, kemudian drag simbol movil clip **kotakMc** ke stage, kemudian buat nama instance **kotak**.
5. Buat layer "Bola", kemudian drag simbol movie clip **bolaMc** ke dalam stage, kemudian buat nama instance **bola**.
6. Agar bola dapat digerakkan dengan mouse, buat layer "Action" kemudian pada frame 1 buat script sebagai berikut:

```
startDrag ("bola", true);
Mouse.hide();
```



Gambar 6.6. (kiri)  
Editing simbol kotakMc.  
Gambar 6.7. (kanan)  
Editing simbol bolaMc



Gambar 6.8 Layout pada stage dan timeline

7. Buat layer "Button", kemudian buat button "Reset" dengan script sebagai berikut:

```
on (release) {
    with (_root.kotak) {
        gotoAndStop(1);
    }
}
```

9. Untuk mendeteksi tabrakan, buat script pada instance kotak. Klik pada kotak di stage kemudian buat script sebagai berikut:

```
onClipEvent (enterFrame) {
    if (_root.kotak._currentframe == 1) {
        if (_root.bola.hitTest(_root.kotak)) {
            with (_root.kotak) {
                gotoAndPlay(2);
            }
        }
    }
}
```

10. Simpan file dengan nama **collision fla**.
11. Pilih **Control** —> **Test Movie** untuk melihat hasilnya.



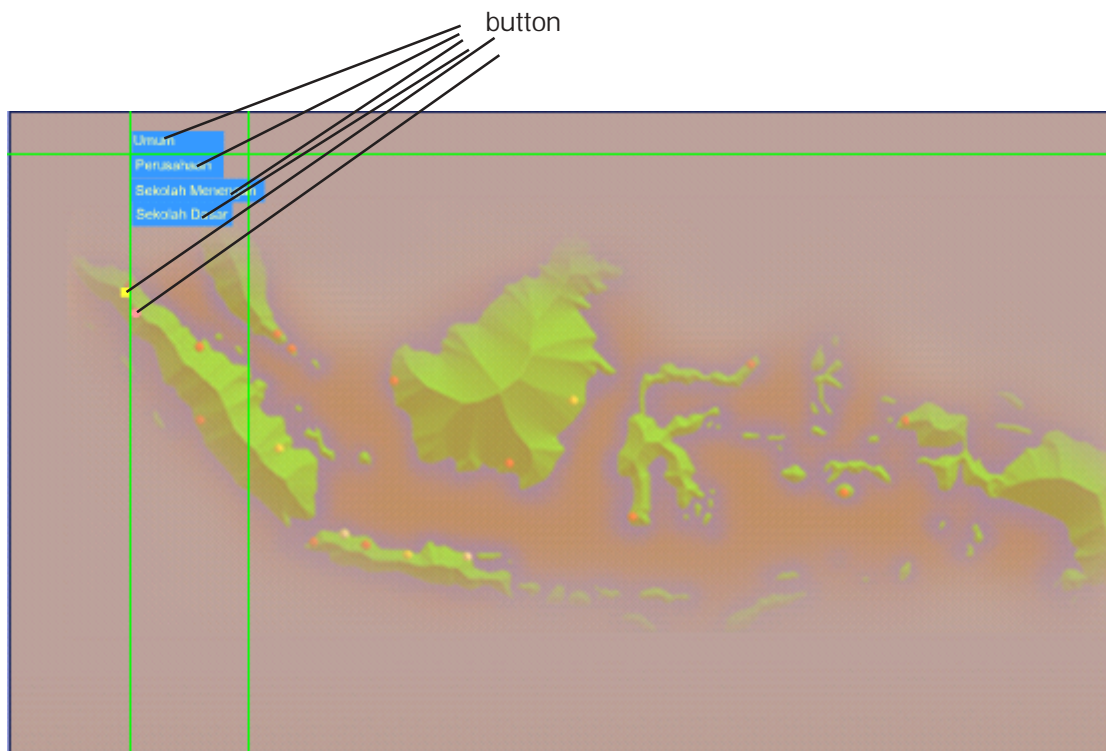
7

# PULL DOWN MENU

**Setelah mempelajari bagian ini, diharapkan dapat:**

1. Memahami pembuatan pull down menu
2. Menggunakan data dari text eksternal

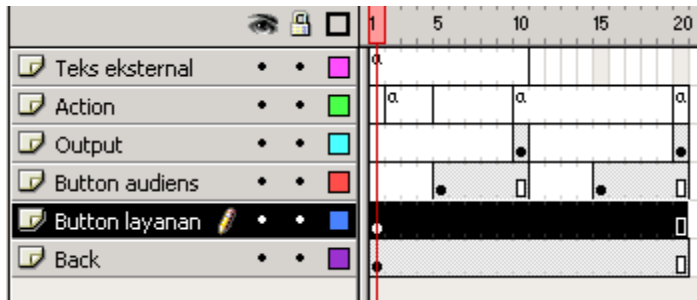
## 1. Menu dengan Button



Gambar 7.1 Enam macam button

### Button

1. Buat simbol button "training" sebagai berikut:  
Pada frame up, buat persegi empat berwarna kuning, Pada frame Over buat teks "TRAINING".
2. Buat simbol button "hardware" sebagai berikut:  
Pada frame up, buat lingkaran berwarna pinI, Pada frame Over buat teks "HARDWARE".
3. Buat simbol button "umum" sebagai berikut:  
Pada frame up, buat layer pertama persegi empat dengan warna biru sebagai warna background. Buat layer kedua dengan teks "Umum" warna kuning. Pada frame Over ganti warna teks tersebut.
4. Seperti nomor 3, buat simbol button "perusahaan", sekolahMenengah", dan "sekolahDasar".



Gambar 7.2 Layout pada timeline.

### Scene1

5. Import gambar peta "peta\_Indonesia.jpg", kemudian gunakan sebagai background pada layer pertama, dengan panjang frame sampai frame 20.
6. Buat layer "Button layanan", kemudian drag simbol button "training" dan "hardware". Tempatkan kedua button di ddekat kota Medan sesuai peta.
7. Buat layer "Button audiens", kemudian drag simbol button "umum", "perusahaan", "sekolahMenengah", dan "sekolahDasar". keempat button berturutan dari atas ke bawah di sebelah kiiri atas dari peta.
9. Buat layer "Output" untuk menempatkan output teks. Buat keyframe pada frame 10 dan 20. Pada frame 10 dan 20 buat dynamic text dengan nama variabel **output**.
9. Buat layer "Action" dan buat action stop() pada frame 2, 10 dan 20.
10. Buat layer "Teks eksternal" dan buat action sebagai berikut:

```
loadVariablesNum ("dataProyek.txt", 0);
```

11. Klik pada button "training" kemudian buat script sebagai berikut:

```
on (release) {
    gotoAndPlay(5);
    output = " ";
}
```

11. Klik pada button "hardware" kemudian buat script sebagai berikut:

```
on (release) {
    gotoAndPlay(15);
    output = " ";
}
```

**File teks eksternal**

12. Buat file text dengan nama **dataProyek.txt** dengan Notepad yang isinya sebagai berikut:

```
&futureKids=Future Kids&
&sekDasar=SD Negeri 2, SD Immanuel&
&sekMenengah=SMP Negeri 8, SMP Regina&
&perusahaan=Pertamina, RS Daerah Medan&
&umum=TB Gramedia, Mall Taman Anggrek&
```

```
&profex=Professional Executive Training&
&sekDasar=SD Negeri 5&
&sekMenengah=SMP Negeri 8, SMA Negeri 10&
&perusahaan=Pertamina, RS Daerah Medan&
&umum=TB Gramedia, Mall Citraland&
```

10. Simpan file dengan nama **InfoProyek fla**.
11. Pilih **Control** —> **Test Movie** untuk melihat hasilnya.