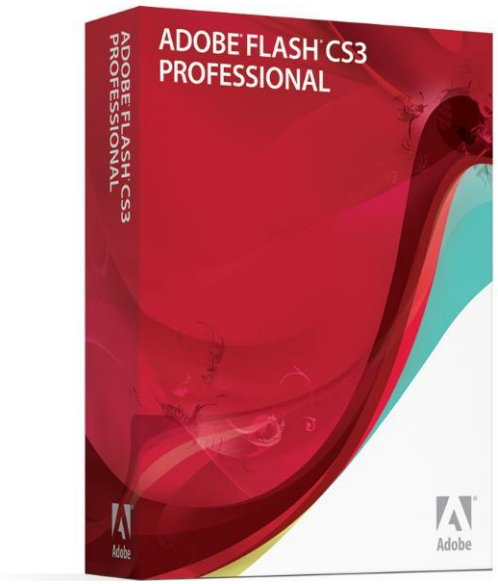
TEKNIK MULTIMEDIA

Dosen Pembina :

Bella Hardiyana, S. Kom, M. Kom



Chapter III

SEKILAS TENTANG ACTIONSCRIPT 2.0 (PART 2)

FUNCTION, METHOD DAN CLASS

Function dan Method

Function dan method adalah kumpulan dari baris kode yang dapat digunakan berkali-kali dengan teknik pemanggilan yang sederhana. Sebuah function dikatakan sebagai method jika fungsi tersebut berada di dalam sebuah class tertentu.
Berikut ini adalah contoh sederhana penggunaan function:

```
function cetakPesan() {
 trace("Pesan ini dipanggil menggunakan function");
}
cetakPesan();
```

```
function cetakPesan(pesan:String) {
 trace(pesan);
}
cetakPesan("Pesan ini dipanggil menggunakan function");
```

Function dan Method

Contoh function untuk banyak parameter adalah, sebagai berikut:

```
var nim:String="10506357";
var nama:String="Edward Kenway";
var prodi:String="Sistem Informasi";
function biodata(nim:String, nama:String, prodi:String) {
 trace("NIM "+nim+",nama saya "+nama+" prodi "+prodi+".");
}
biodata(nim,nama,prodi);
```

TAMPILAN OUTPUT

NIM 10506357,nama saya Edward Kenway prodi Sistem Informasi.

Function dan Method

Analisis Sintaks berikut ini

```
function pertambahan(a:Number , b:Number, c:Number):Number {
 return (a + b + c);
}
trace(pertambahan(1, 4, 6));
trace(pertambahan(1, 4));
trace(pertambahan(1, 4, 6, 8));
```

TAMPILAN OUTPUT

11
NaN
11

Apa maksud tampilan output diatas??

Tipe-tipe Function dan Method

Function yang terdapat di dalam class disebut sebagai method dari class tersebut. Terdapat beberapa jenis fungsi yang dapat digunakan dalam membangun suatu aplikasi menggunakan actioscript, yaitu:

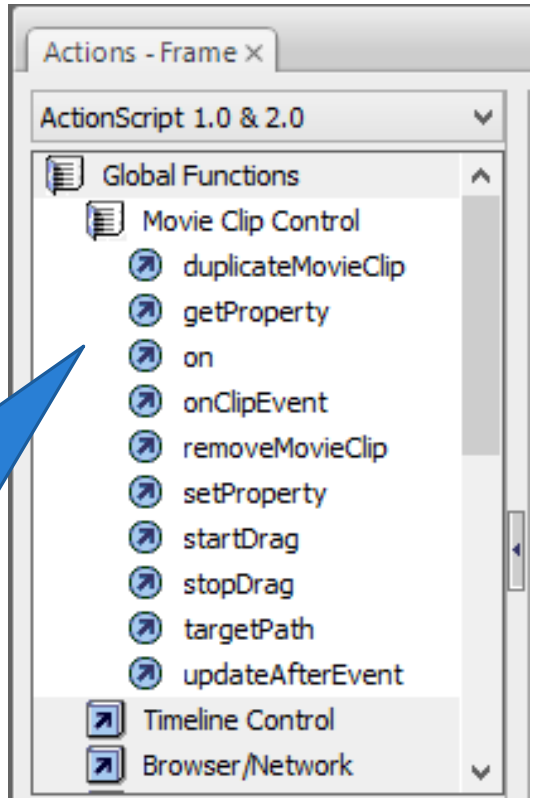
- built-in functions
- named and user-defined functions
- anonymous functions
- callback functions
- constructor functions
- function literal

Built-in Functions

Function yang tidak tergabung di dalam class disebut sebagai top-level functions atau ada juga yang menyebutnya dengan predefined atau built-in functions. Built-in functions dapat dipanggil langsung tanpa menggunakan constructor, artinya fungsi ini dapat dipanggil hanya dengan satu baris script saja. Beberapa contoh built-in function adalah:

- trace();
- setInterval();
- getTimer();
- parseInt();
- Dan lain-lain

Contoh-contoh built-in function pada actionscript 2 dapat dilihatpada bagian referensi script yang terdapat di panel action. Perhatikan bagian global function, di bagian ini built-in function dikategorikan berdasarkan kegunaannya.



Named and User-defined Functions

Function dibuat buatan sendiri, script yang terdapat di dalam blok function dapat dituliskan sendiri tergantung kebutuhan. Numun function ini relatif terhadap timeline. Misalkan sebuah function terletak pada frame 1, maka script tersebut tidak dapat diakses pada frame 2, demikian pula sebaliknya.
Berikut ini adalah struktur dari named and user-defined functions:

```
function namaFungsi(parameters) {
 // blok fungsi
}
```

Contoh named and user-defined functions seperti yang terdapat pada bagian pengenalan function:

```
function cetakPesan() {
 trace("Pesan ini dipanggil menggunakan function");
}
cetakPesan();
```

Anonymous Functions

Anonymous functions, adalah fungsi yang tidak disebutkan namanya atau fungsi yang tidak memiliki nama yang spesifik. Anonymous functions mempunyai variable yang bebas relatif terhadap lingkungan yang mengikat variable tersebut, sehingga fungsi ini sering digunakan dalam event handlers (event handlers akan dibahas pada pembahasan selanjutnya).
Perhatikan 2 script berikut ini:

```
pesan();
function pesan() {
 trace("ini adalah user-defined function");
}
```

User-defined function dapat dipanggil sebelum atau sesudah function dideklarasikan karena pada saat pemanggilan function, script akan secara langsung mereferensi ke tempat function itu berada.

```
pesan();
var pesan:Function = function() {
 trace("ini adalah anonymous function");
};
```

Script ini tidak dapat dijalankan karena anonymous function tidak dapat dipanggil sebelum function tersebut dideklarasikan. Coba pindahkan script **pesan();** ke bagian akhir script. Perhatikan juga bagaimana deklarasi function (script dengan underline biru), yang merupakan alasan dinamakan anonymous, sedangkan underline merah adalah variabel bebas dari anonymous function.

Callback Functions

Callback function adalah function yang dipanggil setelah mengerjakan suatu event tertentu, misalnya setelah melakukan pemanggilan data xml. Perhatikan script berikut ini:

```
my_mc.onPress = function() {
 startDrag(this);
};
my_mc.onRelease = function() {
 stopDrag();
};
```

Script ini digunakan sebagai event handler. function yang berisi script startDrag(this) akan dijalankan jika movieclip my_mc dalam keadaan ditekan atau onPress. Selanjutnya, function yang berisi script stopDrag(this) akan dijalankan setelah movieclip my_mc dilepaskan.

Constructor Functions

Constructor adalah function yang digunakan untuk inisialisasi properties dan methods dari suatu class. Constructors functions terdapat di dalam class definition dan memiliki nama yang sama dengan nama class.
Lebih lanjut tentang constructor akan dipaparkan pada bagian object oriented programming actionscript 2 dan actionscript 3.
Untuk sedikit memahami constructor functions pada actionscript 2, coba analisis script berikut ini:

```
class ClassA {
 function ClassA() {
  trace("ClassA constructor");
 }
}
```

function ClassA() adalah constructor function dari class A.

Function Literal

Function literal adalah sebuah unnamed function yang dideklarasikan di dalam expression sebuah statement. Fungsi ini digunakan sebagai temporarily function.
Struktur dari function literal adalah, sebagai berikut:

```
function (param1, param2, etc) {
 // statements
};
```

Perhatikan baris 1, function literal tidak memiliki nama atau disebut juga dengan unnamed function.

Untuk lebih memahami function literal, perhatikan contoh script berikut ini:

```
var nama:String = "Edward Kenway";
setInterval(function() {
              trace(nama);
            }, 1000);
```

Expresi dari statement setInterval adalah literal function yang menghasilkan output value dari variabel nama.

Return Value pada Functions

Untuk memberikan return value dari suatu function, gunakan return statement. Untuk lebih memahami return value pada suatu function, coba amati script berikut ini dan perhatikan outputnya.

```
function getArea(width:Number , height:Number):Number {
 return width * height;
}
var area:Number = getArea(10, 12);
trace(area);
```

Function getArea() akan melakukan return value berupa width*height dan memasukkannya ke dalam variabel area, sehingga variabel area akan bernilai width*height, yaitu 10 * 12 = 120.

Latihan

1. Apa tampilan output berdasarkan sintaks dibawah ini

```
function myFunction(obj:Object):Void {
 trace("Nama   : "+obj.nama+"\n" +
       "Aktif  : "+obj.aktifKuliah+"\n"+
       "Nim    : "+obj.nim+"\n"+
       "Nilai  : "+obj.nilai);
}
var dataMahasiswa:Object=new Object ;
dataMahasiswa.nama="Edward Kenway";
dataMahasiswa.aktifKuliah=true;
dataMahasiswa.nim="10506357";
dataMahasiswa.nilai=98;
myFunction(dataMahasiswa);
```

Apa penjelasan dari output yang anda lihat?

2. Buatlah function yang menghasilkan output operasi pertambahan, pengurangan, perkalian dan pembagian dari 2 variabel a dan b.

CLASS

Class pada AS2

"Class adalah tulang punggung AS 2.0" (Learning Actionscript 2.0 in adobe Flash, help file Flash). Maksudnya, pada dasarnya actionscript dibangun dari class-class yang telah dibuat oleh developer flash. Programmer dapat menggunakan class tersebut dengan script-script yang sederhana dan mudah untuk diingat.
Berikut adalah struktur dasar penulisan class. Nama file sama dengan **NamaClass.as**

```
class NamaClass {
 //Class body
}
```

Pembahasan tentang class sangat erat kaitannya dengan OOP. Untuk itu, sebelum membahas tentang teknik-teknik penggunaan class pada actionscript terlebih dahulu akan dibahas beberapa hal berikut ini:

- Apa saja yang perlu diketahui tentang objek oriented dalam Flash.
- Penjelasan seputar Package
- Object-oriented programming fundamentals

Object-Oriented Programming (OOP) dan Flash

Seperti yang sudah dijelaskan sebelumnya, ActionScript2.0 adalah bahasa berorientasi objek. Pada actionscript yang berbasis OOP akan ditemukan konsep-konsep class dan instance. Sedikit review tentang OOP, proses pembuatan obyek dari sebuah class disebut dengan instantiation. Jadi obyek merupakan hasil instansiasi dari class. Obyek disebut juga dengan instance.
Object-oriented programming telah diyakini lebih baik dibandingkan dengan pemrograman berbasis prosedural. Pemrograman berbasis OOP memungkinkan programmer untuk membuat struktur program yang bersifat reusable, scalable, robust, dan maintainable. Fitur inilah yang membuat OOP lebih unggul dari pemrograman prosedural.

Pada actionscript 2, class didefinisikan pada file actionscript external berekstensi ***.as***. Untuk membuat sebuah file actionscript external, pilih **file > new > actionscript file**. Actionscript external (file berekstensi *.as) juga dapat dibangun menggunakan text editor seperti notepad, wordpad, dan lain-lain.

Ingat!!
Nama file external actionscript yang memuat class HARUS sama dengan nama class tersebut.

Package

Package adalah folder yang berisi kumpulan class. Pembuatan dan penentuan package bertujuan untuk megorganisir class-class yang telah dibuat. Bayangkan saja jika terdapat 1000 class dengan fungsi yang bervariasi ditempatkan dalam sebuah folder saja.
Paket biasanya digunakan untuk mengatur class yang saling berhubungan. Sebagai contoh, class SegiEmpat, Lingkaran, dan SegiTiga, didefinisikan dalam file SegiEmpat.as, Lingkaran.as, dan SegiTiga.as. Asumsikan ketiga file actionscript ini disimpan dalam folder BangunDatar . Maka penulisan class harus didahului dengan nama folder dan dipisahkan dengan tanda dot (.), seperti contoh berikut ini:

```
class BangunDatar.SegiEmpat {
 //isi class SegiEmpat
}

class BangunDatar.Lingkaran {
 //isi class Lingkaran
}

class BangunDatar.SegiTiga {
 //isi class SegiTiga
}
```

Jika file eksternal tersebut tidak dimasukkan ke dalam folder, maka class dapat dituliskan seperti biasa.

```
class NamaClass {
 //Class body
}
```

Namun jika file class terletak pada subfolder di dalam folder, maka penamaan class menjadi:

```
class Folder.SubFolder.NamaClass {
 //Class body
}
```

Konsep OOP

Terdapat 3 (tiga) konsep inti dari pemrograman berorientasi objek (OOP), yaitu diantaranya :

1. Pewarisan (*Inheritance*)

Pewarisan adalah mewariskan ciri dan tingkah laku kepada keturunannya. Maksud dari keturunan disini adalah semua yang berada pada hirarki dibawah sang pemberi waris, tidak hanya anak, tetapi juga cucu, dsb. Apa yang diwariskan? Yaitu, ciri dan tingkah laku dari sang pemberi waris kepada sang pewaris.

2. Kebanyakrupaan (*Polymorphism*)

Kebanyakrupaan (*Polymorphism*) merupakan salah satu konsep OOP di Java. Dimana Kebanyakrupaan (*Polymorphism*) adalah kemampuan sebuah variabel *reference* untuk merubah *behavior* sesuai dengan apa yang dipunyai *object* (Berganti-ganti peran). Kita bisa menggunakan variabel dengan tipe data kelas super untuk memegang referensi dari objek kelas sub-nya.

3. Pembungkusan/Pengkapsulan (*Encapsulation*)

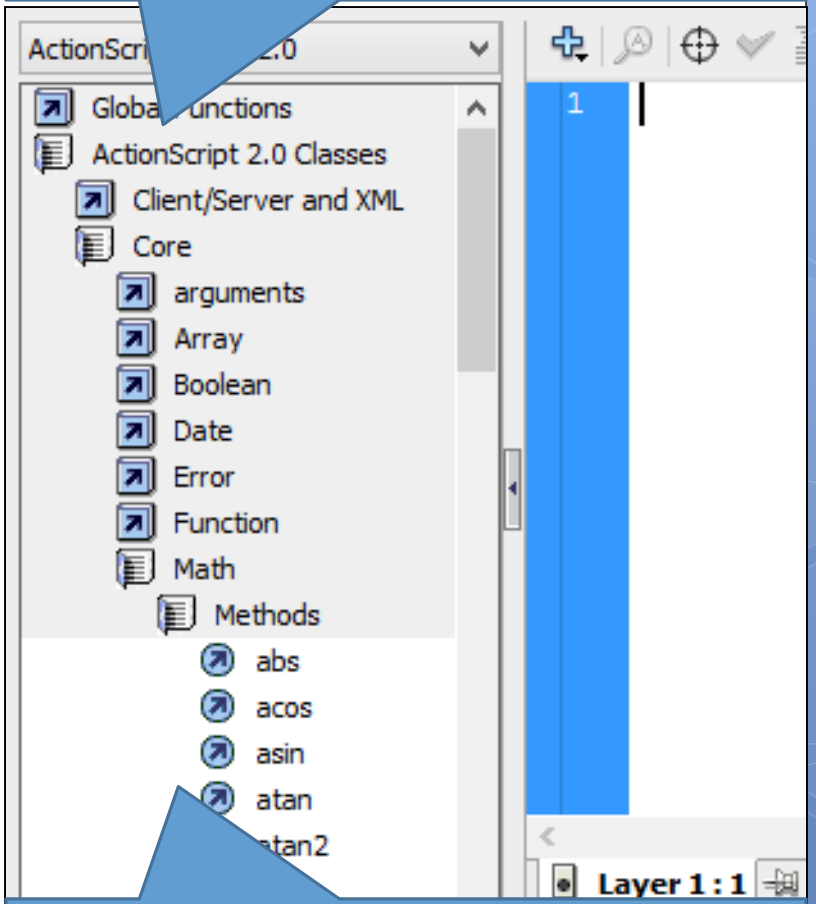
Pembungkusan/Pengkapsulan (*Encapsulation*) adalah teknik penyembunyian atribut dan metode, dimana keterintegrasian antara class, atribut dan metode **tidak boleh** semua deklarasian sebagai public. Java mengenal istilah *visibility/access modifier* yang akan membedakan hak pengaksesan atribut dan metode. Teknik penyembunyian atribut dan metode inilah yang disebut dengan konsep Pembungkusan/Pengkapsulan (*Encapsulation*).

Tipe-Tipe Class

Terdapat 2 tipe class pada Actionscript 2, yaitu:

1. **Built-in Class,** yaitu class yang sudah disediakan oleh Flash. Programer dapat memanggilnya dalam bentuk statement yang sederhana. Seperti yang telah dijelaskan sebelumnya, **"Classes are the backbone of ActionScript 2.0"**, beberapa syntax Actionscript yang tergolong built-in class adalah: Array, Button, Color, Date, Math, dan lain-lain.
2. **Custom Class,** adalah class yang dibuat sendiri dengan maksud dan tujuan yang sesuai dengan kebutuhan programmer dalam membangun suatu aplikasi.

Built in class pada Flash, dapat dilihat pada kategori AS 2.0 Classes



Daftar method yang terdapat di dalam built-in class Math

Built-in Class

Built-in class yang tersedia pada AS 2.0 terdiri dari:

- Top-level classes
- The flash.display package
- The flash.external package
- The flash.filters package
- The flash.geom package
- The flash.net package
- The flash.text package
- The mx.lang package
- The System and TextField packages

Top-level classes adalah class yang dapat dipanggil langsung dengan statement/syntax yang sederhana. Untuk class selain top-level class atau class yang terdapat dalam packages, maka pemangilan class harus dilakukan dengan menambahkan script:

```
import NamaPackage.NamaClass;
```

Berikut ini adalah pemaparan dari sebagian build-in class yang sering digunakan.

Built-in Class – Array (1)

Array adalah top-level class yang menangani data array. Data array adalah kumpulan dari berbagai jenis tipe data. Berikut ini adalah contoh penggunaan class array:

```
trace("new Array : Hardiyana,Fadilah,Bolo-Bolo");
var namaMahasiswa:Array=new Array("Hardiyana","Fadilah","Bolo-Bolo");
trace("menampilkan Array menggunakan perulangan for");
for(i=0;i<namaMahasiswa.length;i++){
 trace("Mahasiswa "+i+": "+namaMahasiswa[i]);
}
trace("\n-unshift('Udin Sigarantang')");
namaMahasiswa.unshift("Udin Sigarantang");
trace(namaMahasiswa);

trace("\n-shift()");
namaMahasiswa.shift();
trace(namaMahasiswa);

trace("\n-push('Ujang Sigaucing')");
namaMahasiswa.push("Ujang Sigaucing");
trace(namaMahasiswa);

trace("\n-pop()");
namaMahasiswa.pop();
trace(namaMahasiswa);

trace("\n-reverse()");
namaMahasiswa.reverse();
trace(namaMahasiswa);
```

Built-in Class – Array (2)

Berikut hasil output dari sintaks sebelumnya

```
new Array : Hardiyana,Fadilah,Bolo-Bolo
menampilkan Array menggunakan perulangan for
Mahasiswa 0: Hardiyana
Mahasiswa 1: Fadilah
Mahasiswa 2: Bolo-Bolo

-unshift('Udin Sigarantang')
Udin Sigarantang,Hardiyana,Fadilah,Bolo-Bolo

-shift()
Hardiyana,Fadilah,Bolo-Bolo

-push('Ujang Sigaucing')
Hardiyana,Fadilah,Bolo-Bolo,Ujang Sigaucing

-pop()
Hardiyana,Fadilah,Bolo-Bolo

-reverse()
Bolo-Bolo,Fadilah,Hardiyana
```

Built-in Class – String

String adalah class yang menangani tipe data String. Berikut adalah beberapa contoh method dari class String:

```
var nama:String=new String("Universitas Komputer Indonesia");
trace("String = Universitas Komputer Indonesia");
trace("substr(0,11)         : " + nama.substr(0,11));
trace("substring(0,11)      : " + nama.substring(0,11));
trace("concat(' - UNIKOM') : " + nama.concat(" - UNIKOM"));
trace("toLowerCase()        : " + nama.toLowerCase());
trace("toUpperCase()        : " + nama.toUpperCase());
```

Berikut hasil output dari program diatas

```
String = Universitas Komputer Indonesia
substr(0,11)         : Universitas
substring(0,11)      : Universitas
concat(' - UNIKOM') : Universitas Komputer Indonesia - UNIKOM
toLowerCase()        : universitas komputer indonesia
toUpperCase()        : UNIVERSITAS KOMPUTER INDONESIA
```

LATIHAN

1. Apa kegunaan dari substr(). Substring(), concat(), toLowerCase() dan toUpperCase().
2. Apa koding yang digunakan untuk menampilkan Kata Komputer berdasarkan contoh koding diatas.

Built-in Class – Date

Berikut adalah contoh penggunaan class Date:

```
var waktu:Object=new Date();
var dataHari:Array=new Array("Minggu","Senin","Selasa","Rabu","Kamis",
                    "Jum\'at","Sabtu");
var dataBulan:Array=new Array("Januari","Februari","Maret","April",
                     "Mei","Juni","Juli","Agustus",
                     "September","Oktober","November","Desember");
tanggal=waktu.getDate();
hari=dataHari[waktu.getDay()];
bulan=dataBulan[waktu.getMonth()];
tahun=waktu.getFullYear();
trace("Hari ini : "+hari+", "+tanggal+" "+bulan+" "+tahun);
```

Berikut hasil output dari program diatas

```
Hari ini : Selasa, 18 Maret 2014
```

LATIHAN

Ubahlah koding baris ke-2 menjadi **("Senin", "Selasa", "Rabu", "Kamis", "Jum\'at", "Sabtu", "Minggu")**.

Lalu bagaimana caranya agar tetap menghasilkan output benar??

Built-in Class dalam Package

Penggunaan built-in class yang terdapat di dalam package akan terlihat lebih rumit dibandingkan dengan top-level class. Berikut adalah salah satu contoh penggunaan build-in class yang terdapat di dalam package :

```
import flash.display.BitmapData;
import flash.geom.Rectangle;

var myBitmapData:BitmapData = new BitmapData(100, 80, false, 0x00FFCCAA);
var mc:MovieClip = this.createEmptyMovieClip("mc",
this.getNextHighestDepth());
mc.attachBitmap(myBitmapData, this.getNextHighestDepth());
trace(myBitmapData.rectangle);
myBitmapData.rectangle = new Rectangle(1, 2, 4, 8);
trace(myBitmapData.rectangle);
```

CUSTOM CLASS

Membuat Custom Class (1)

Untuk membuat dan menuliskan suatu custom class, ikuti langkah berikut:

1. Pilih **File > New > ActionScript File**.
2. Save dengan nama **Mahasiswa.as**. Ingat, nama file harus sama dengan nama class, untuk itu perhatikan penggunaan huruf kapital.
3. Ketik koding berikut ini

```
class Mahasiswa {
 private var __nim:String;
 private var __nama:String;
 public function Mahasiswa(p_nim:String, p_nama:String) {
  this.__nim = p_nim;
  this.__nama = p_nama;
 }
 public function get nim():String {
  return this.__nim;
 }
 public function set nim(value:String):Void {
  this.__nim = value;
 }
 public function get nama():String {
  return this.__nama;
 }
 public function set nama(value:String):Void {
  this.__nama = value;
 }
}
```

Membuat Custom Class (2)

4. Berikutnya, buat sebuah file flash baru, File > New > Flash File (AS 2)
5. Tuliskan script berikut ini pada panel action.

```
import Mahasiswa;
var dataMahasiswa:Mahasiswa = new Mahasiswa("001", "Bella Hardiyana");
trace("Sebelum :");
trace("\t Nim = " + dataMahasiswa.nim);
trace("\t Nama = " + dataMahasiswa.nama);
dataMahasiswa.nim = "002";
dataMahasiswa.nama = "Lina Fadilah";
trace("Sesudah :");
trace("\t Nim = " + dataMahasiswa.nim);
trace("\t Nama = " + dataMahasiswa.nama);
```

6. Simpan proyek anda didalam folder yang sama dengan **Mahasiswa.as** dan tes program anda. Sehingga menghasilkan tampilan output seperti dibawah ini

```
Sebelum :
          Nim = 001
          Nama = Bella Hardiyana
Sesudah :
          Nim = 002
          Nama = Lina Fadilah
```

EVENT HANDLER

Event Handler

Events adalah actions yang terjadi saat file SWF dimainkan. Event terdisi atas 2 golongan, yaitu:

1. **User event**, event yang berupa mouse click atau keypress. User event adalah event yang membutuhkan interaksi dengan user untuk menjalankannya.
2. **System event**, event yang dijalankan otomatis oleh Flash tanpa memerlukan interaksi dengan user.

Pada Flash terdapat 2 teknik penerapan Event, yaitu:

1. **Event pada symbol.** Teknik ini adalah teknik menempatkan actionscript pada symbol. Cara menambahkan script pada symbol adalah dengan menseleksi (mengklik dengan mouse) symbol yang diinginkan, kemudian ketikkan script pada panel action.
2. **Event pada frame.** Teknik ini ditandai dengan adanya simbol penanda pada frame. Teknik ini dikenal dengan **dynamic event handler**.

Yang akan kita gunakan pada pertemuan kali ini adalah teknik **dynamic event handler.**

Contoh Penggunaan Event Handler

Ketik koding berikut ini

```
var button:MovieClip=this.createEmptyMovieClip
          ("tombol_mc",this.getNextHighestDepth());
tombol_mc.lineStyle(1,0x0000ff,100);
tombol_mc.beginFill(0x0000ff,100);
tombol_mc.moveTo(0,0);
tombol_mc.lineTo(30,0);
tombol_mc.lineTo(30,30);
tombol_mc.lineTo(0,30);
tombol_mc.lineTo(0,0);
tombol_mc.endFill();

this.onLoad=function(){
 tombol_mc._alpha=50;
}

tombol_mc.onRollOver=function(){
 tombol_mc._alpha=100;
}
tombol_mc.onRollOut=function(){
 tombol_mc._alpha=50;
}
tombol_mc.onPress=function(){
 tombol_mc.startDrag(false);
}
tombol_mc.onRelease=function(){
 tombol_mc.stopDrag();
}
tombol_mc.onReleaseOutside=function(){
 tombol_mc.stopDrag();
}
```

Script untuk membuat movieclipt dan menggambar di dalam movieclip

onLoad adalah system event handler. tombol_mc akan secara otomatis memiliki _alpha atau transparansi 50%.

User event handler terdiri dari:

- onRollOver
- onRollOut
- onPress
- onRelease
- onReleaseOutside
- Dll..

Contoh Penggunaan Event Handler

Perhatikan juga output dari script dengan system event handler berikut ini:

```
var X:Number=0;
var Y :Number=0;
_root.lineStyle(3,0xff0000,100);
_root.moveTo(X,Y);

this.onEnterFrame=function(){
 _root.lineTo(X++,Y++);
}
```

Event handler onEnterFrame adalah system event handler sehingga script akan dijalankan tanpa interaksi dengan user.

Key.addListener (1)

Salah satu teknik event handler yang efektif adalah dengan metode addListener. Statement addListener ini memungkinkan programmer untuk menentukan event handlernya sendiri. Perhatikan script berikut ini:

```
var customListener:Object = new Object();
customListener.onKeyDown = function() {
 trace("Keyboard ditekan");
}
customListener.onKeyUp = function() {
 trace("Keyboard dilepaskan");
}
Key.addListener(customListener);
```

onKeyDown adalah kondisi saat keyboard ditekan.
onKeyUp adalah kondisi saat keyboard dilepaskan.

Key.addListener (2)

Kode karakter keyboard dan kode ASCII memungkinkan programmer untuk mendeteksi tombol pada keyboard.

```
var keyListener:Object = new Object();
keyListener.onKeyDown = function() {
  trace("Tombol keyboard yang Anda tekan");
  trace("Kode  : "+String.fromCharCode(Key.getAscii()));
  trace("Kode  : "+Key.getCode());
  trace("ASCII : "+Key.getAscii());
};
Key.addListener(keyListener);
```

Berikut hasil output dari program diatas

```
Tombol keyboard yang Anda tekan
Kode  : g
Kode  : 71
ASCII : 103
Tombol keyboard yang Anda tekan
Kode  : 5
Kode  : 53
ASCII : 53
```

Key.addListener (3)

Script berikut akan memberikan gambaran tentang penerapan Key.addListener:

```
var DISTANCE:Number = 10;
var car_mc:MovieClip=this.createEmptyMovieClip("car_mc",this.getNextHighestDepth());
with(car_mc){
 lineStyle(3,0x000000,100);
 moveTo(100,100);
 lineTo(100,150);
 lineTo(200,150);
 lineTo(200,100);
 lineTo(100,100);
}
var keyListener_obj:Object = new Object();
keyListener_obj.onKeyDown = function() {
 switch (Key.getCode()) {
  case Key.LEFT : car_mc._x -= DISTANCE; break;
  case Key.UP : car_mc._y -= DISTANCE; break;
  case Key.RIGHT : car_mc._x += DISTANCE; break;
  case Key.DOWN : car_mc._y += DISTANCE; break;
 }
};
Key.addListener(keyListener_obj);
```

Gunakan tombol panah untuk menggerakkan mobil-mobilan

Latihan

Ketik koding berikut ini

```
var gambar_mc:MovieClip=this.createEmptyMovieClip("gambar_mc",this.getNextHighestDepth());
var gambar1_mc:MovieClip=this.createEmptyMovieClip("gambar1_mc",this.getNextHighestDepth());
var gambar2_mc:MovieClip=this.createEmptyMovieClip("gambar2_mc",this.getNextHighestDepth());
function buatGambar(mc, warna, korX, korY){
 mc.lineStyle(1, warna, 100);
 mc.beginFill(warna, 100)
 mc.moveTo(korX, korY);
 mc.lineTo(korX+100, korY);
 mc.lineTo(korX+100, korY+100);
 mc.lineTo(korX, korY+100);
 mc.lineTo(korX, korY);
 mc.endFill();
}

function klikObjek(){
 this.swapDepths(getNextHighestDepth());
 this.startDrag(false);
}

function releaseObjek(){
 this.stopDrag();
}

buatGambar(gambar_mc, 0x808040, 25, 25);
buatGambar(gambar1_mc, 0xC0C0C0, 150, 25);
buatGambar(gambar2_mc, 0x008081, 275, 25);
gambar_mc.onPress=klikObjek;
gambar1_mc.onPress=klikObjek;
gambar2_mc.onPress=klikObjek;
gambar_mc.onRelease=releaseObjek;
gambar1_mc.onRelease=releaseObjek;
gambar2_mc.onRelease=releaseObjek;
```

BERSAMBUNG

TERIMA KASIH !!!

SAMPAI JUMPA MINGGU DEPAN