

BAB 2

DASAR-DASAR PROGRAM ACTIONSCRIPT 3.0

Dalam bab ini akan dibahas mengenai

Bab ini dimulai dengan penjelasan mengenai konsep-konsep dasar pemrograman beserta dengan contoh pengkodeannya dalam ActionScript. Supaya langsung bisa melihat hasil pengkodean anda, buat project baru dalam FlashDevelop anda dengan memilih menu Project → New Project, pilih template **AS3 Project**, beri nama project anda DasarAS3. Masuk pada direktori src, double-click file Main.as, yang akan memunculkan package berikut.

```
package
{
    import flash.display.Sprite;
    import flash.events.Event;
    /**
     * ...
     * @author Nugroho Agus H
     */
    public class Main extends Sprite
    {
        public function Main():void
        {
            if (stage) init();
            else addEventListener(Event.ADDED_TO_STAGE, init);
        }
        private function init(e:Event = null):void
        {
            removeEventListener(Event.ADDED_TO_STAGE, init);
            // entry point
        }
    }
}
```

1. Pemberian komentar. Komentar adalah potongan-potongan teks yang tidak ikut dijalankan//dieksekusi oleh program. Komentar hanya digunakan untuk dokumentasi program sehingga bisa digunakan dalam pendeteksian kesalahan maupun pengembangan program. Dalam ActionScript terdapat dua tipe komentar, yaitu: komentar dalam satu baris dan komentar dalam satu blok baris.

1.1. Untuk memberikan komentar hanya dalam satu baris digunakan tanda dua forward slash (//), misalnya:
`Trace("Hello World") //komentarnya adalah untuk menampilkan tulisan Hello World.`

1.2. Untuk memberikan komentar dalam beberapa baris diawali dengan tanda /* dan diakhiri dengan tanda */ , misalnya:

```
/*      copy right by nugroho
      F. Teknologi Informasi
      UKDW
*/
```

2. Penggunaan perintah Trace

Perintah **Trace** digunakan untuk menampilkan informasi pada panel Output. Panel Output hanya digunakan untuk menguji jalannya program saja, bukan sebagai tampilan utama program. Yang menjadi tampilan utama program adalah Stage kita.

Misalnya:

```
public function Main():void {
    if (stage) init();
    else addEventListener(Event.ADDED_TO_STAGE, init);
    trace (2 + 5) // akan menampilkan bilangan 7
}
```

3. Pendeklarasian Variabel

Sebuah variabel (misalnya variabel x dalam pelajaran matematika yang menyajikan sebuah bilangan yang bisa diberi dengan suatu nilai) adalah sebuah penampung informasi yang bisa diambil atau pun diganti. Variabel-variabel yang akan digunakan dalam ActionScript harus dideklarasikan terlebih dahulu. Beberapa tipe data/tipe variabel yang sering digunakan antara lain: TextField, Number, String, Array, Sprite, Shape, dan MovieClip. Kita juga dapat membuat tipe variabel sendiri berupa objek Class untuk aplikasi kita. Perintah deklarasi variabel mempunyai bentuk umum:

```
var namaVariabel:TipeVariabel;
```

atau cara kedua berikut juga memberikan hasil yang sama:

```
var namaVariabel:TipeVariabel=new TipeVariabel();
```

Anda bisa menambahkan program berikut ke dalam kode program anda di dalam lingkup fungsi Main() untuk memunculkan tulisan "Hello World" ke dalam Stage Anda.

```
public function Main():void {
    var textSaya:TextField=new TextField();
    textSaya.text="Hello World";
    stage.addChild(textSaya);
}
```

Lanjutkan dengan menguji hasil program anda dengan menekan F5 atau Ctrl+Enter, maka anda akan melihat tulisan "Hello World" pada ujung kiri atas Stage anda.

Lakukan hal serupa untuk contoh-contoh selanjutnya, yaitu tambahkan kode yang ingin anda jalankan di dalam lingkup fungsi public Main(). Lanjutkan dengan menguji program anda dengan menekan F5.

Nama variabel harus memenuhi aturan-aturan berikut:

- Hanya terdiri dari huruf, angka, karkater garis bawah, dan simbol \$.
- Tidak diawali dengan angka
- Tidak sama dengan kata khusus yang sudah dipakai oleh ActionScript (*reserved word*), seperti **var**, **Sprite**, dan lain-lain.
- Harus unik

Untuk penulisan dalam penamaan variabel, anda dapat menggunakan mode KataKapital (CamelCase) yang merupakan proses penulisan beberapa kata tanpa tanda spasi namun dengan mengkapitalisasi (menulis dengan huruf besar) setiap kata pembentuknya, misalnya textSaya, radiusBola, dan tinggiAwalStage. Untuk tipe data kompleks (movie Clip, button, textField) yang ada dalam stage yang dibuat dengan Adobe Flash CS3 Profesional, anda dapat mendeklarasikannya dengan menuliskannya pada nama instan (*instance name*) pada bagian properti objeknya. Flash CS3 akan membuat variabel dengan nama sama dengan nama instan tersebut.

4. Tipe-tipe Data

ActionScript 3.0 mempunyai beberapa tipe data primitif seperti yang diberikan dalam Tabel 2.1. Selain itu ActionScript juga mempunyai beberapa tipe data kompleks yang dibangun dari tipe data primitif seperti yang diberikan dalam Tabel 2.2. Bahkan kita bisa membangun tipe data sendiri, dalam bahasa pemrograman objek sering dikenal dengan nama class objek.

Tabel 2.1. Tipe data primitif ActionScript 3.0

Tipe Data	Pengertian	Contoh
String	Sebuah karakter atau sederet karakter	var textSaya:String = "Hello World", bisa juga ditulis dengan: var textSaya: String textSaya="Hello World"
Boolean	Hanya bernilai benar (true) atau salah (false)	var sudahSelesai:Boolean = true
Number	Bilangan riil positif dan negatif	var percepatan:Number = 0.01
Int	Bilangan bulat positif dan negatif (-2.147.483.648 s.d. 2.147.483.647)	var radiusBola:int = 10
Uint	Bilangan bulat positif (0 s.d. 4.294.967.295)	var warnaBola: uint = 0xFF00FF
*	Tipe data sembarang (biasanya digunakan untuk menampung data yang belum diketahui tipenya)	var dataBaru: * = dataSembarang

Tabel 2.2. Beberapa Tipe data kompleks ActionScript 3.0

Tipe Data	Pengertian	Contoh
MovieClip	Sebuah simbol movie clip	var tokohSaya:MovieClip = new MovieClip()
Sprite	Sebuah Sprite	var kotak: Sprite = new Sprite()
TextField	Sebuah field teks yang bertipe dynamic atau field teks yang bertipe	var tulisan:TextField =new TextField()

	input (secara default akan bertipe dynamic)	tulisan.text="Tulisan Dinamis"
SimpleButton	Sebuah simbol button (tombol)	var start:SimpleButton=new SimpleButton()
Date	Data dalam format tanggal	var ultah:Date= new Date()
Array	Larik atau array yang dapat memuat lebih dari satu nilai	var arrayBilangan: Array = new Array() arrayBilangan = [2, 3, 15]
Objek	Struktur dasar dari setiap entitas ActionScript. Objek juga merupakan suatu bentuk variabel yang bisa diatur sendiri untuk menyimoan berbagai nilai sebagai alternatif dari Array	var gambar:Objek: new Object gambar.nama ="foto action" gambar.panjang=600

Untuk tipe data kompleks akan lebih mudah dipahami pada saat nanti kita memakainya.

Apabila anda mulai bekerja dengan konsep pemrograman berorientasi objek yang menggunakan beberapa Class, anda perlu menggunakan beberapa kata tambahan di depan variabel yang menunjukkan ruang lingkup pengaksesan variabel, misalnya:

```
public var radiusBola: int
```

Penjelasan mengenai Class akan dibahas dalam bab 3, namun demikian ruang lingkup variabel dalam ActionScript 3.0 akan diberikan di awal karena sudah kita butuhkan dalam pemrograman dengan FlashDevelop. ActionScript 3.0 mendefinisikan empat kata untuk menunjukkan ruang lingkup variabel (atau pun fungsi/*method*), yaitu:

- public: dapat diakses oleh semua bagian kode di dalam keseluruhan aplikasi/project tempat variabel didefinisikan
- private: hanya dapat diakses oleh method/fungsi tempat variabel didefinisikan
- internal: dapat diakses oleh sebarang class yang didefinisikan dalam package yang sama.
- protected: hanya dapat diakses oleh method/fungsi tempat variabel didefinisikan dan class yang extend class tempat variabel didefinisikan

5. Operator

- Aritmetika (+, -, *, /). ActionScript mengakomodasi semua operator aritmetika sesuai dengan aturan-aturan perhitungannya, dengan perkalian dituliskan dengan * sedangkan pembagian dituliskan dengan /.
- Modulo (%). ActionScript juga mengakomodasi operator modulo dengan simbol %, misal: 5 % 3 //menghasilkan 2.
- Increment (++) dan decrement (--). ActionScript dapat menambah atau mengurangi 1 dari suatu bilangan dengan operator tunggal (++ atau --), misalnya:
var bilangan:int = 10
bilangan ++ //menghasilkan 11
bilangan -- //menghasilkan 10 lagi
- Operator majemuk (+=, -=, *=, /=, dan %=). ActionScript menyediakan operator majemuk untuk mempersingkat penulisan, misalnya:
var bilangan:int = 10
bilangan +=2 //artinya sama dengan bilangan = bilangan + 2, menghasilkan 12
bilangan /=3 //artinya sama dengan bilangan = 12/3, menghasilkan 4

6. Penggunaan pernyataan kondisional

6.1. Pernyataan If digunakan untuk mengijinkan pemrosesan bagian dari blok ActionScript hanya jika kondisi yang diminta benar. Pernyataan If mempunyai bentuk penyajian sebagai berikut:

```

If (kondisi){
    Proses yang akan dikerjakan jika kondisi benar
}

```

Jika kondisi benar, maka proses akan dijalankan. Silahkan mencoba potongan program berikut dalam FlashDevelop file DasarAS3 anda

```

public function Main():void {
    var x: Number = 2
    if (x < 3){
        trace ("x lebih kecil dari 3")
    }
    Else {
        trace ("x lebih besar atau sama dengan xtidak akan ditulis apapun")
    }
}

```

Anda dapat menggunakan operator logika atau pun perbandingan seperti dalam tabel 2.3

Tabel 2.3. Simbol-simbol yang digunakan untuk membentuk Ekspresi logika

Simbol	Peterangan	Contoh
==	Sama dengan	If (a==1){trace ("a sama dengan 1")}
!=	Tidak sama dengan	If (a != 1){trace("a tidak sama dengan 1")}
<, <=, >, >=	Perbandingan	If (a >= 1){trace("a lebih besar atau sama dengan 1")}
And(&&), atau ()	Operator majemuk	If (a = 1 && b=2){trace("a=1 dan b=2")}

6.2. If ... Else. Penambahan kata Else sebagai tambah blok pernyataan If berarti program akan mengerjakan blok program di dalam blok Else jika kondisi di dalam pengujian If tidak benar.

```

If (a==1){
    trace ("a sama dengan 1"
}
Else {
    trace ("a tidak sama dengan 1")
}

```

6.3. Switch- case. Pernyataan Switch-Case digunakan untuk menguji beberapa kemungkinan kondisi. Pernyataan Switch memuat variabel yang akan diuji, sedangkan setiap pernyataan Case menyatakan kemungkinan nilai untuk menguji. Diberikan juga pernyataan Break untuk mencegah pengerjaan instruksi-intruksi selanjutnya. Perhatikan contoh berikut ini.

```

Switch (a){
    case 1 : trace ("satu")
    case 2: trace ("dua")
    break}

```

7. Penggunaan kalang (Loop)

7.1. For ... Pernyataan For digunakan untuk mengerjakan proses iterasi dalam jumlah tertentu.

```
for (var i:Number = 0; i < 3; i++) {  
    trace(i)  
} // akan menampilkan 012
```

7.2. while ... Proses di dalam blok While akan diuji selama kondisi di dalam pernyataan While benar.

```
var i:int = 0  
while (i < 5){  
    trace(i)  
    i ++  
}  
// akan menampilkan 01234
```

7.3. do ... While. Proses di dalam blok do-while minimal akan dikerjakan sekali, kemudian akan dilanjutkan pengujian kondisi di dalam pernyataan While. Proses akan dijalankan berulang kali selama kondisi masih benar.

```
var i:int = 0  
do {  
    trace(i)  
    i ++  
} while (i < 5) // akan menampilkan 01234
```

8. Bekerja dengan Class

Salah satu kelebihan ActionScript 3.0 adalah dukungannya terhadap pemrograman menggunakan class-class. Dengan penggunaan class, anda dapat menerapkan konsep pemrograman berorientasi objek dalam pengkodean program anda. Sebuah class adalah cetak biru dari sebuah objek. Class mendefinisikan objek secara penuh, meliputi data objek dan operasi-operasinya. Untuk membuat class perlu diperhatikan beberapa hal berikut ini.

- Pembuatan Class dalam ActionScript harus diawali dengan sebuah package. Sebuah package adalah grup dari class (bisa hanya satu class atau beberapa class). Anda bisa juga menambahkan namaFolder di belakang kata package yang menunjukkan direktori tempat anda menyimpan class anda. Misalnya anda bisa membuat class Bola yang akan anda taruh dalam folder lib dengan menambahkan kata lib di belakang kata package
- Dokumen dari class harus disimpan dengan nama yang sama dengan nama dari class, misalnya namaClass.as. Misalnya class Bola yang anda buat harus anda simpan dengan nama Bola.as
- Sebaiknya class yang anda buat merujuk ke class yang sudah dimiliki Flash, misalnya Sprite atau Movie Clip. Gunakan tambahan kata extends Sprite pada lanjutan namaClass anda, misalnya `public class Bola extends Sprite.`
- Anda bisa juga menambahkan variabel dengan ruang lingkup public, internal atau pun yang lainnya sesuai kebutuhan. Misalnya anda bisa menambahkan variabel-variabel yang bersifat public, seperti: radius yang menunjukkan jari-jari bola, warna yang menunjukkan warna bola, vx yang menunjukkan kecepatan bola searah sumbu-x, dan vy yang menunjukkan bola searah sumbu-y.
- Anda bisa juga menambahkan method berupa fungsi dengan ruang lingkup yang anda inginkan. Di dalam fungsi. Salah satu fungsi yang sebaiknya ada adalah fungsi dengan nama fungsi sama dengan nama class anda. Jadi buatlah fungsi Bola dengan ruang lingkup public dengan model pengiriman parameter variabel radius dan warna.

Misalnya anda dapat membuat class Bola yang anda taruh di dalam Folder lib dari project DasarAS3.as3proj dengan langkah berikut ini. Buka project DasarAS3 yang sudah anda buat sebelumnya. Lanjutkan dengan membuat file class AS3 melalui menu File → New → AS3 Document. Lanjutkan dengan memberi nama dokumen anda melalui menu File → Save As (Ctrl+Shift+S), pilih subfolder lib dari folder DasarAS3, beri nama Bola.as.

Misalkan objek bola anda mempunyai beberapa variabel yang bersifat public, yaitu: radius, vx, dan vy

```
package lib { //tambahkan folder tempat anda menyimpan file Bola.as Anda jika menginginkan,
    // misal package sport { ....isi class ..... } artinya anda menyimpan file Bola.as pada folder sport

import flash.display.Sprite;

public class Bola extends Sprite {
    //deklarasi variabel private yang hanya berlaku dan digunakan di dalam class Bola saja
    private var radius:Number; // jari-jari bola yang akan dibuat
    private var color:uint; // warna bola yang akan dibuat

    //deklarasi variabel public yang berlaku dan digunakan oleh file yang lain.
    public var vx:Number = 0; //kecepatan bola searah sumbu x dengan rumus  $vx=v*\cos q$ 
    public var vy:Number = 0; //kecepatan bola searah sumbu y dengan rumus  $vy=v*\sin q$ 
    public var v:Number = 0; //kecepatan bola
    public var massa:Number=0;
    public var indeks:Number=0;

    public function Bola(radius:Number=40, color:uint=0xff0000) { //pemberian nilai default
        properti bola
            this.radius = radius; //menyamakan properti bola dengan parameter input dari
        fungsi bola
            this.color = color; //menyamakan properti bola dengan parameter input dari
        fungsi bola
            graphics.beginFill(color); //mulai mewarnai bola
            graphics.drawCircle(0, 0, radius); //mulai menggambar bola dengan pusat (0,0)
        dan jari-jari=radius
            graphics.endFill(); //mengakhiri mewarnai bola
    }
}

package
{
    public class
    {
```

```

    }

}

f. package namaFolder{
g.     public class namaClass{
h.     }
i.     }
j.

```

9. Fungsi

10. Event, Listener, dan Event Handler

Semua interaksi user di dalam Flash dikelola oleh “event-event” beberapa contoh event adalah: mouse click, mouse move, input keyboard, loading file, dan lain-lain. Dalam buku ini akan banyak menggunakan pengaturan event untuk mendisain interaksi user.

Dalam ActionScript, suatu objek bisa mendengarkan (*listen*) event yang terjadi pada dirinya dengan menggunakan method **addEventListener** pada objek tersebut. Method ini membutuhkan dua argumen, yaitu:

- Event apa yang akan didengarkan, misalnya MouseEvent atau pun KeyboardEvent
- Fungsi apa yang akan saya kerjakan apabila saya mendengarkan event.

Dalam contoh kita, baris:

```

tombolMulai.addEventListener(MouseEvent.CLICK, tekanTombol);
function tekanTombol(mevt:MouseEvent):void{
    ISI FUNGSI
}

```

Mempunyai maksud sebagai berikut:

- Tombol dengan nama instance tombolMulai akan mendengarkan event Mouse Click (**MouseEvent.CLICK**)
- Apabila tombolMulai di-click dengan mouse, maka fungsi **tekanTombol** akan dipanggil dan dikerjakan. Fungsi **tekanTombol** dinamakan **Event Handler**. Sebuah event handler harus berupa fungsi dengan satu nilai input (misalnya dalam contoh kita adalah `mevt:MouseEvent`) dan tidak mengembalikan nilai apa-apa (yaitu mempunyai tipe keluaran `void`).

beri nama layer pertama sebagai utama

buat sebuah text statis dengan tulisan "berapa jumlah bola yang anda lihat?"

buat sebuah textbox lagi dengan mode inputText beri nama instace "txtJawab. nama txtJawab ini yang akan kita gunakan dalam kode program sebagai variabel yang menunjuk ke text jawaban

buat sebuah layer lagi beri nama script. disinilah tempat kita akan menempatkan kode program dalam actionscript 3.0

tambahkan sebuah objek sprite dengan nama spPapan sebagai tempat mearuh bola yang akan kita tampilkan.

buatlah sebuah objek shape berbentuk papan kotak dengan ukuran 300x300. taruhlah kotak tersebut di dalam objek Sprite, dengan menjadikan shape tersebut sebagai anak dari Sprite spPapan

dengan menuliskan `spPapan.addChild(shpKotak)`

buat fungsi untuk membuat bola beri nama fungsi `buatBola` dengan parameter `k` yang nantinya menunjukk pada bola ke-`k`

bangkitkan bilangan random `r` kemudian lakukan looping untuk membuat bola

buat Tombol OK Cek untuk melihat benar tidaknya jawaban

masuk ke menu window library common button pilih tombol button ayng anda inginkan, misalnya tumboll bulat arcade, saya pilih classic button-arcade button-red. beri nama `tmbCek` (diambil dari kata tombol Cek)

pada button `tmbCek` tambahkan `addEventListener` untuk mendengarkan event, misalnya `mouse.Click`, dihubungkan dengan fungsi `cekJawab`, artinya jika tombol ini diklik dengan mouse maka akan mengerjakan fungsi `cekJawab`

dalam fungsi `cekJawab`. lakukan pengujian apakah isi dari `txtJawab` sama dengan jumlah bola, jika ya, maka keluarkan pesan

jawaban anda benar. dengan menuliskannya di dalam text dinamik `txtPesan`.

```
var spPapan:Sprite = new Sprite();
```

```

spPapan.x=20;

spPapan.y=20;

spPapan.graphics.strokeStyle(2,0);

spPapan.graphics.beginFill(0xFFFFFF,0.9);

spPapan.graphics.drawRect(0,0,300,300);

spPapan.graphics.endFill();

stage.addChild(spPapan);

var shpBola:Shape = new Shape();

function buatBola(k:int):void{

    shpBola.graphics.strokeStyle(2,0);

    var indeksWarna:int;

    indeksWarna=k*25+255;

    var posisiBola:int;

    posisiBola=10+k*25;

    shpBola.graphics.beginFill(indeksWarna, 0.9);

    shpBola.graphics.drawCircle(posisiBola,20,10);

    shpBola.graphics.endFill();

    spPapan.addChild(shpBola);

}

var r:int;

r=1+Math.round(Math.random()*9);

for(var i:int=1; i<=r; i++){

    buatBola(i);

}

```

```
tmbCek.addEventListener(MouseEvent.CLICK, cekJawab);
```

```
function cekJawab(mevt:MouseEvent):void{
```

```
    var jawab:int = int(txtJawab.text);
```

```
    if (jawab==r) {
```

```
        txtPesan.text="Anda Benar";
```

```
    }
```

```
    else {
```

```
        txtPesan.text="Maaf, Salah";
```

```
    }
```

```
}
```

```
tmbAcak.addEventListener(MouseEvent.CLICK, acak);
```

```
function acak(mevt:MouseEvent):void{
```

```
shpBola.graphics.clear();
```

```
r=1+Math.round(Math.random()*9);
```

```
for(i=1; i<=r; i++){
```

```
    buatBola(i);
```

```
}
```

```
}
```