



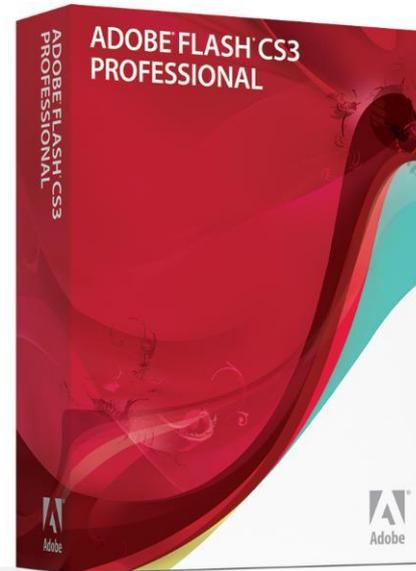
## **TEKNIK MULTIMEDIA**



**Dosen Pembina :**

**Bella Hardiyana, S. Kom, M. Kom**





## Chapter II

# SEKILAS TENTANG ACTIONSCRIPT 2.0 (PART I)



# OVERVIEW



# Overview

Kebanyakan pemula yang masih belum terbiasa dengan konsep *Object Oriented Programming* (OOP) akan mengalami kesulitan dalam mempelajari Actionscript 3.0 (AS 3). Sehingga pada chapter kali ini kita akan membahas tentang AS 2, agar ketika sudah memiliki gambaran ketika mempelajari AS3.

Berikut konsep-konsep apa saja yang akan dibahas, yaitu :

- Mengetahui Tipe Data
- Teknik Seleksi dan Perulangan
- Function, Method dan Class
- Menggambar dengan Script
- Event Handler
- Component
- Opensource AS2



# TIPE DATA



# Mengenal Tipe Data

“Data merepresentasikan huruf, angka, gambar, dan semua informasi yang bisa kita manipulasi menggunakan Flash” (Learning Actionscript 2.0 in Adobe Flash). Data didefinisikan ke dalam variabel. Data juga dapat di *load* dari *file* eksternal (.txt, .xml, dll), webservice (via http, dll) dan database.

Berikut beberapa tipe data yang sering digunakan adalah sebagai berikut :

- **String** (kumpulan karakter)
- **Number** (*default float*, tapi bisa juga kita gunakan **integer**).
- **Boolean** (terdiri dari **true** dan **false**).
- **Object** (kumpulan dari **properties**, atribut yang mendeskripsikan objek).
- **MovieClip** (symbol yang memainkan animasi).
- **Void** (mengidentifikasi function yang tidak memberikan return value).
- **null** (segala sesuatu yang bersifat null).
- **undefined** (tipe data yang belum didefinisikan).



# Variabel

Seperti yang telah dibahas sebelumnya, data didefinisikan ke dalam variabel. Variabel berfungsi untuk menyimpan informasi.

Beberapa aturan dalam penulisan variabel adalah, sebagai berikut:

- Variabel dapat berupa sebuah atau sekumpulan karakter yang terdiri dari huruf dan angka, hindari penggunaan **punctuation** karakter (!@#\$% dst.)
- Variabel harus diawali dengan huruf, disarankan huruf kecil.
- Variabel bersifat **CASE-SENSITIVE**, yang berarti membedakan antara huruf besar dan kecil. Jadi variabel **A** dengan variabel **a** adalah 2 variabel yang berbeda.
- Yang paling penting, hindari penamaan variabel sama dengan **keyword Actionscript** (contoh: **trace**, **var**, **String**, dll.). **Keyword Actionscript** adalah syntax yang berwarna **biru** jika dituliskan pada flash.

Untuk melihat value dari suatu variabel, dapat menggunakan perintah **trace(nama\_variabel)**. Value variabel tersebut akan muncul di **Output panel**.



# Deklarasi Variabel

Contoh deklarasi variabel namaBarang dengan tipe data String.

```
var namaBarang: String;
```

Nama variabel diikuti oleh karakter titik dua (:) tanpa menggunakan space.

var menyatakan deklarasi variabel, teks var akan berwarna biru.

Tipe data untuk variabel **namaBarang** adalah String.

Diakhiri dengan tanda semicolon (;)



# Contoh Deklarasi Variabel

Amati contoh deklarasi dan pemanggilan *value* variabel pada *script* berikut :

```
var namaBarang:String;  
namaBarang = "Komputer";  
trace (namaBarang) ;
```

```
var namaBarang:String="Komputer";  
trace (namaBarang) ;
```

```
namaBarang="Komputer";  
trace (namaBarang) ;
```

Pada saat dijalankan, ketiga *script* di atas akan memberikan *value* yang sama pada *output panel*, yaitu **Komputer**

Untuk menjalankan *script*, gunakan shortcut key **Ctrl + Enter**



# Tipe Data String

**String** adalah tipe data yang dapat menampung kumpulan karakter. **Karakter** adalah semua anggota yang ada di *keyboard*, meliputi huruf, angka, tombol fungsi lainnya. Value variabel bertipe string harus diapit oleh **single quotation marks ('')** maupun **double quotation marks ("")**.

Berikut beberapa contoh value untuk tipe data string:

```
var namaProdi:String="Sistem Informasi";
var namaUniversitas:String='UNIKOM Bandung';
namaKota="Bandung";
var namaDepan:String="Edward";
var namaBelakang:String="Kenway";
trace(namaProdi);
trace(namaUniversitas);
trace(namaKota);
trace(namaDepan+" "+namaBelakang);
```



# Escape sequence

**Escape sequence** adalah karakter khusus yang hanya bisa digunakan dengan mengawali penulisan menggunakan *single backslash* (\). Berikut tabel escape sequence

Escape sequence	Character
\b	Backspace
\f	Form feed
\n	Baris baru
\r	Carriage return
\t	Tab
\'	Single quotation (')
\"	Double quotation (")
\\	Single backslash (\)

## Contoh Penggunaan

```
var namaSaya:String="Nama saya \"Edward Kenway\"";
```



# Tipe Data Number

Secara **default**, tipe data Number bersifat **Float (bilangan real/pecahan)**, dengan value  $-5e-324$  s/d  $+1.79E+308$ . Tipe data Number dapat dimanipulasi dengan menggunakan operator aritmatik tambah (+), kurang (-), kali (\*), bagi/div (/), mod (%), increment (++) dan decrement (--).

Contoh script :

```
var a:Number=100;  
var b:Number=5;  
trace(a+b);  
trace(a-b);  
trace(a*b);  
trace(a/b);  
trace(a%b);  
a++;  
b--;  
trace(a);  
trace(b);
```

-> menghasilkan output 105  
-> menghasilkan output 95  
-> menghasilkan output 500  
-> menghasilkan output 20  
-> menghasilkan output 0  
-> sama dengan  $a=a+1$   
-> sama dengan  $b=b-1$   
-> menghasilkan output 101  
-> menghasilkan output 4



# Tipe Data Boolean

Value untuk tipe data boolean yaitu true (1) atau false (0). Boolean digunakan untuk logical operator. Berikut ini adalah logical operator yang terdapat dalam Actionscript:

- Logical && (AND)
- Logical || (OR)
- Logical ! (NOT)

```
var c:Boolean;  
var d:Boolean;  
var e:Boolean;  
a=1;  
b=0;  
c=a&&b;  
d=a||b;  
e=!a;  
trace(c);  
trace(d);  
trace(e);
```

-> deklarasi variabel c sebagai boolean

-> deklarasi variabel d sebagai boolean

-> deklarasi variabel e sebagai boolean

-> a bernilai true

-> b bernilai true

-> c = a AND b

-> d = a OR b

-> e = NOT a

-> true AND false akan bernilai false (0)

-> true OR false akan bernilai true (1)

-> NOT true bernilai false



# Tipe Data Object

**Tipe Data Objek** adalah kumpulan properties. Properti adalah atribut yang mendeskripsikan suatu objek. Untuk lebih memahami tipe data objek, perhatikan script berikut ini:

```
var dataMahasiswa:Object = new Object();  
dataMahasiswa.nim="10506357";  
dataMahasiswa.nama="Edward Kenway";  
dataMahasiswa.prodi="Sistem Informasi";  
dataMahasiswa.nilai=97;  
trace("NIM      : "+dataMahasiswa.nim);  
trace("Nama     : "+dataMahasiswa.nama);  
trace("Prodi    : "+dataMahasiswa.prodi);  
trace("Nilai    : "+dataMahasiswa.nilai);
```



# Tipe Data Movieclip

**Movieclip** adalah symbol yang dapat memainkan animasi. Movieclip adalah tipe data yang berhubungan dengan graphic element. Movieclip memiliki fungsi untuk mengontrol grafis dengan menggunakan event handler .

```
this.createEmptyMovieClip("segitiga_mc", 1);  
segitiga_mc.lineStyle(2, 0x00FF00, 100);  
segitiga_mc.beginFill(0xFF0000, 100);  
segitiga_mc.moveTo(100,300);  
segitiga_mc.lineTo(300,300);  
segitiga_mc.lineTo(200,100);  
segitiga_mc.lineTo(100,300);  
segitiga_mc.endFill();  
trace(segitiga_mc);
```

-> segitiga\_mc  
adalah movieclip

Arti dari script ini akan dijelaskan pada pembahasan berikutnya



# Tipe Data Void

**Void** mendefinisikan suatu function yang tidak mengembalikan value suatu variabel. Penjelasan function akan dibahas pada bagian selanjutnya. Untuk lebih jelasnya, perhatikan script berikut ini:

```
function zeroArray (theArr:Array):Void {
  var i:Number;
  for (i = 0; i < theArr.length; i++) {
    theArr[i] = 0;
  }
}
var myArr:Array = new Array();
myArr[0] = 1;
myArr[1] = 2;
myArr[2] = 3;
trace(myArr);
zeroArray(myArr);
trace(myArr);
```

Fungsi zeroArray ini bertipe Void, yang artinya fungsi ini tidak melakukan return value.

Variabel myArr adalah array yang berisi data {1,2,3}

- > myArr sebelum dioperasikan
- > jalankan fungsi zeroArray
- > variabel myArr akan ditimpa

Output :

1,2,3

0,0,0



# Tipe Data null

Tipe data **null** hanya memiliki sebuah value yaitu null yang artinya variabel ini tidak menyimpan data. Tipe data null digunakan untuk beberapa situasi dimana membutuhkan identifikasi terhadap isi suatu variabel. Untuk lebih jelasnya, amati script berikut ini:

```
if (Selection.getFocus() == null) {  
    trace("no selection");  
}
```

Pernyataan di atas berarti jika tidak terseleksi, `Selection.getFocus() == null`, maka cetak tulisan no selection pada output panel.



# Tipe Data undefined

**Undefined** adalah tipe data default untuk setiap variabel yang belum memiliki value atau yang sama sekali belum dideklarasikan. Value dari tipe data undefined ini adalah undefined. Perhatikan script berikut ini:

```
var c:String;
trace(c);
if (c == undefined) {
  trace("c belum memiliki value");
}
trace(a);
var b:String;
b = "Halo";
trace(b);
```

## TAMPILAN OUTPUT

```
undefined
c belum memiliki value
undefined
Halo
```

Variabel **c** adalah bertipe string, namun sebelum diberikan value, tipe data untuk **c** adalah undefined. Maka hasil keluaran dari **trace(c)** adalah **undefined**. Sedangkan untuk variabel **a**, saat dilakukan **trace(a)**, output panel akan menampilkan **undefined**. Di sini **a** adalah variabel yang belum didefinisikan.



# Konversi Tipe Data

Beberapa bentuk konversi dari tipe data dapat dilihat pada script berikut ini:

```
var a:String="100";  
var b:String="22.5";  
var d:Number=55.5;  
c=a+b;  
trace(c);  
c=parseInt(a)+parseFloat(b);  
trace(c);  
c=d.toString()+a+b;  
trace(c);
```

- Script `parseInt( )` digunakan untuk melakukan konversi data ke tipe Number integer.
- Script `parseFloat` digunakan untuk konversi data ke tipe float.
- Script `toString` digunakan untuk konversi data ke tipe data string.



# Passing an Object by Reference

Berikut ini adalah contoh script pass object by reference.

```
var dataMahasiswa:Object=new Object();  
dataMahasiswa.nama="Lina Fadilah";  
dataMahasiswa.nim="10506350";  
var biodata:Object=dataMahasiswa;  
trace(dataMahasiswa.nama);  
trace(biodata.nama);  
trace(dataMahasiswa.nim);  
trace(biodata.nim);
```

```
dataMahasiswa.nama="Bella Hardiyana";  
biodata.nim="10506351";  
trace(dataMahasiswa.nama);  
trace(biodata.nama);  
trace(dataMahasiswa.nim);  
trace(biodata.nim);
```

Perhatikan output dari script pada baris ini, data pada variabel `dataMahasiswa` akan dicopy atau dicetak juga pada variabel `biodata`.

Perhatikan juga script pada baris ini dimana jika data pada variabel `dataMahasiswa` berubah, maka data pada variabel `biodata` pun ikut berubah. Demikian pula sebaliknya, jika data pada variabel `biodata` berubah maka data pada variabel `dataMahasiswa` pun ikut berubah. Kondisi ini disebut dengan pass object by reference.



# TEKNIK SELEKSI DAN PERULANGAN



# Teknik Seleksi Dan Perulangan

Seleksi dan Perulangan merupakan salah satu inti pemrograman. **Seleksi** adalah melakukan aksi/pernyataan tergantung kondisi; dimana hanya ada 2 kemungkinan pengerjaan aksi tersebut, yaitu **true** atau **false**. Sedangkan **Perulangan** adalah melakukan aksi/pernyataan secara berulang tergantung kondisi bernilai **true** dan perulangan akan berakhir jika kondisi bernilai **false**.

Teknik Seleksi	Teknik Perulangan
if... if...else... if...elseif... switch...	for... for...in... while... do...while...



# Struktur Seleksi

Teknik decision pada actionscript menggunakan perpaduan statement if dan else. Perhatikan beberapa kasus decision pada script berikut ini:

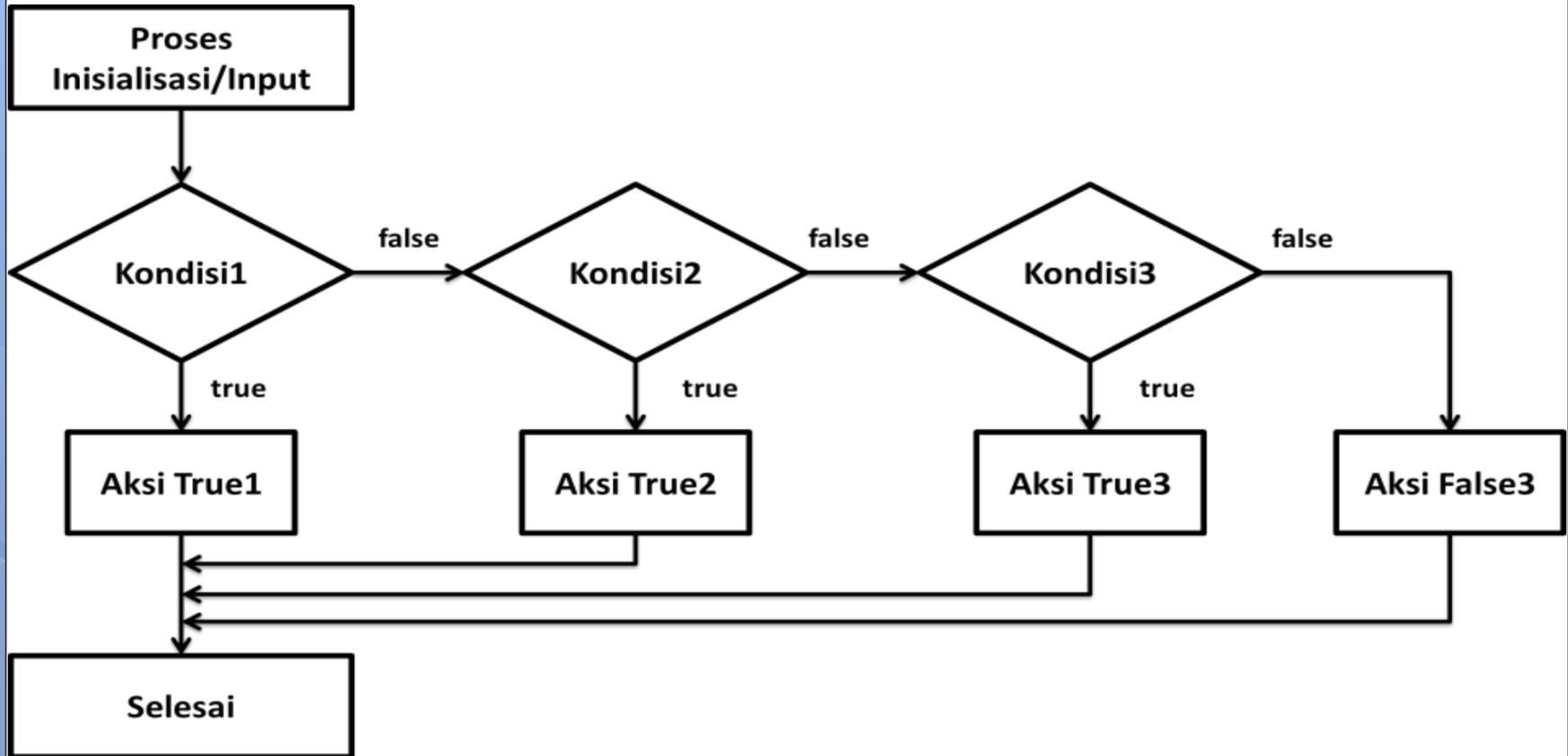
```
if(kondisi1) {  
    AksiTrue1;  
    ...  
}else if(kondisi2) {  
    AksiTrue2;  
    ...  
}else if(kondisi3) {  
    AksiTrue3;  
    ...  
}else {  
    AksiFalse3;  
    ...  
}
```

## CONTOH :

```
var nilai:Number=95;  
if (nilai > 90) {  
    trace("A");  
}else if (nilai > 75) {  
    trace("B");  
}else if (nilai > 60) {  
    trace("C");  
}else {  
    trace("D");  
}
```



# Struktur Seleksi Dalam Bentuk Flowchart





# Struktur Switch

Teknik decision switch ini berfungsi untuk pilihan yang banyak. Berikut format penulisannya :

```
switch (ekspresi){ //variabel yang akan dicocokkan
  case nilai1: aksiCase1 //aksi case 1
                break;
  case nilai2: akseCase2 //aksi case 2
                break;
  ...
  case nilaiN: aksiCaseN //aksi case N
                break;
  default: aksiDefault //bersifat opsional
           //dikerjakan jika tidak ada case yang cocok
}
```



# Contoh Penerapan Switch

Berikut contoh penerapannya:

```
var obj:Object = new Object();
obj.onKeyDown = function(){
  switch(String.fromCharCode(Key.getAscii())) {
    case "A": trace("Kamu menekan tombol A"); break;
    case "a": trace("Kamu menekan tombol a"); break;
    case "E": case "e": trace("Kamu menekan tombol E/e");
               break;
    case "I": case "i": trace("Kamu menekan tombol I/i");
               break;
    default: trace("Kamu menekan tombol yang lain");
  }
}
Key.addListener(obj);
```



# Struktur Perulangan For

Salah satu teknik pengulangan yaitu dengan menggunakan statement for. Berikut ini adalah contoh penggunaan statement for pada actionscript:

```
for(inisialisasi; kondisi; counter){  
  Aksi1; // -----  
  Aksi2; // Ini adalah blok Aksi yang akan diulang  
  AksiN; // -----  
}
```

## CONTOH :

```
var total:Number = 0;  
for (i= 1; i <= 100; i++){  
  total += i;  
}  
trace(total);
```

Script di atas adalah operasi penjumlahan deret aritmatika:

1+2+3+...+99+100



# Struktur Perulangan For-In

Statement lain yang dikombinasikan dengan statement for adalah statement in, dikenal dengan statement for-in. Script ini digunakan untuk melakukan pemanggilan properties pada object (masih ingat tentang tipe data object?).

Berikut contoh kasus pengulangan statement for-in.

```
var dataMhs:Object = new Object();  
dataMhs.nim="10506357";  
dataMhs.nama="Edward Kenway";  
dataMhs.prodi="Sistem Informasi";  
dataMhs.nilai=92;  
for (var mahasiswa in dataMhs) {  
    trace("Data Mahasiswa :"+mahasiswa+" = "+dataMhs[mahasiswa]);  
}
```

Hasil dari pengulangan for in akan disajikan terbalik. Perhatikan output dari script di samping.

## **OUTPUT**

Data Mahasiswa :nilai = 92

Data Mahasiswa :prodi = Sistem Informasi

Data Mahasiswa :nama = Edward Kenway

Data Mahasiswa :nim = 10506357



# Struktur Perulangan While

Konsep pengulangan while hampir sama dengan konsep pengulangan for. Dimana pengulangan ini akan mengecek kondisi terlebih dahulu. Jika kondisi bernilai true, maka melakukan aksi berupa pengulangan. Pengulangan while memiliki struktur yang berbeda dengan pengulangan for. Berikut adalah struktur penulisan pengulangan while.

```
inisialisasi;
while(kondisi) {
  Aksi1; // -----
  Aksi2; // Ini adalah blok Aksi yang akan diulang
  AksiN; // -----
  counter;
}
```

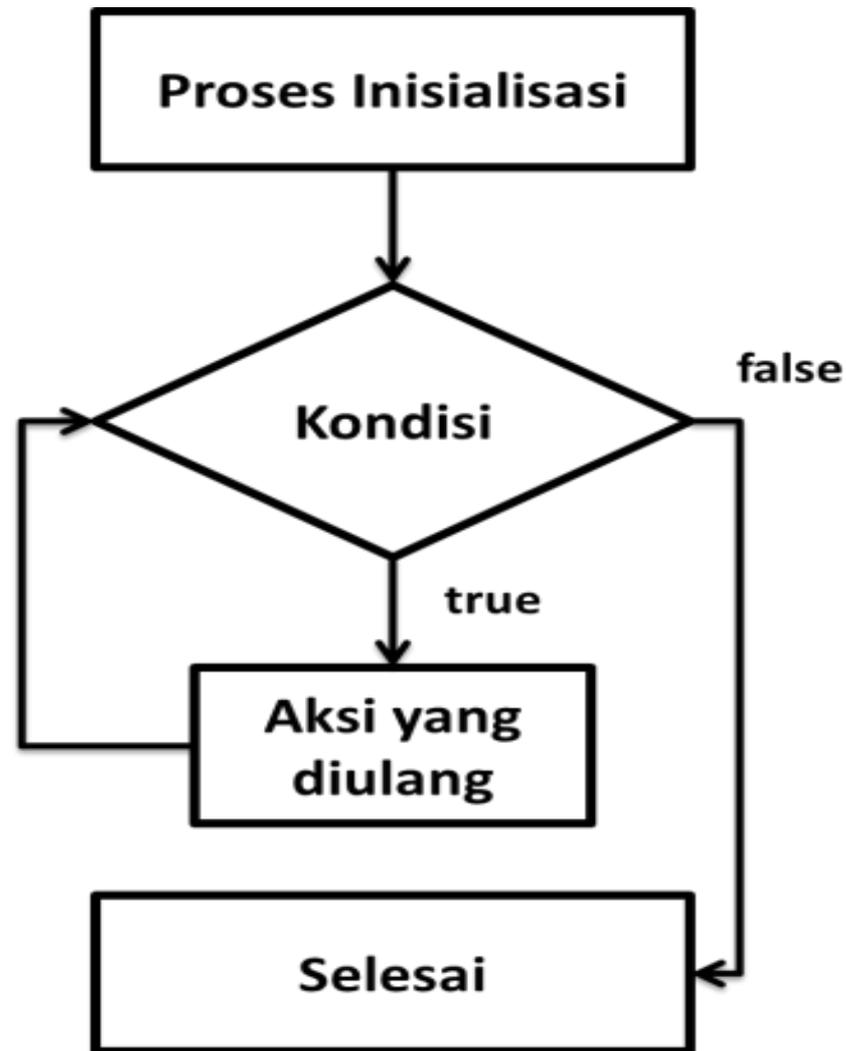
## CONTOH :

```
var i:Number = 1;
while(i <= 10) {
  trace(i);
  i++;
}
```

Script di atas adalah menampilkan angka 1 s/d 10:  
1, 2, 3, ..., 10



# Struktur Perulangan While Dalam Bentuk Flowchart





# Struktur Perulangan Do-While

Sama seperti pengulangan sebelum for dan do-while yang akan melakukan pengulangan selama kondisi terpenuhi (true). Hanya saja pengulangan do-while sedikit berbeda dari kedua pengulangan sebelumnya. Dimana pengulangan do-while akan melakukan aksi terlebih dahulu, setelah itu baru melakukan pengecekan kondisi.

```
inisialisasi;  
do{  
  Aksi1; // -----  
  Aksi2; // Ini adalah blok Aksi yang akan diulang  
  AksiN; // -----  
  counter;  
}while(kondisi);
```

## CONTOH :

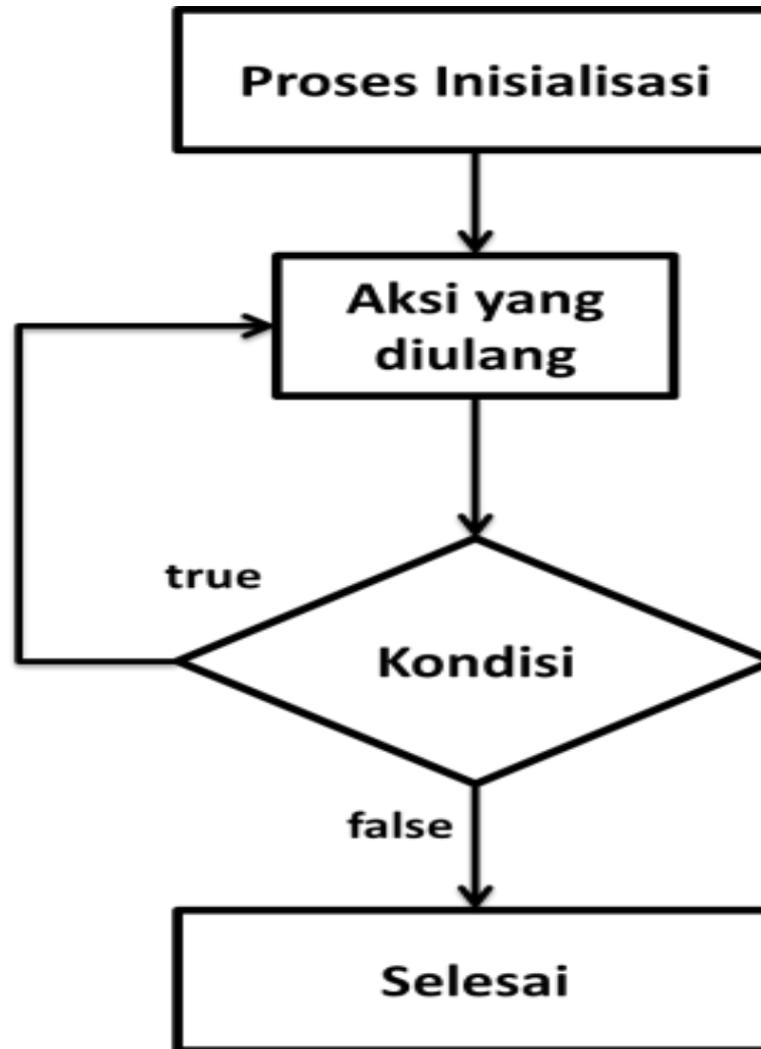
```
var i:Number = 1;  
do{  
  trace(i);  
  i++;  
}while(i <= 10);
```

Script di atas adalah menampilkan angka 1 s/d 10:

1, 2, 3, ..., 10



# Struktur Perulangan Do-While Dalam Bentuk Flowchart





# Latihan

Buatlah program berdasarkan tampilan output dibawah ini

5	4	3	2	1	*				1			*	*	*	*	*	
5	4	3	2		*	*			1	2			*		*		
5	4	3			*	*	*		1	2	3			#			
5	4				*	*			1	2			*		*		
5					*				1				*	*	*	*	*

Apa output berdasarkan sintaks dibawah ini

```
var nilaiMahasiswa:Array = [86, 79, 92, 67, 94];  
for (var i:String in nilaiMahasiswa) {  
    trace(nilaiMahasiswa[i]);  
}
```

Ayo coba kerjakan !  
Sekalian mengingat materi algoritma  
dasar 😊



# BERSAMBUNG

**TERIMA KASIH !!!**  
**SAMPAI JUMPA MINGGU DEPAN**

