



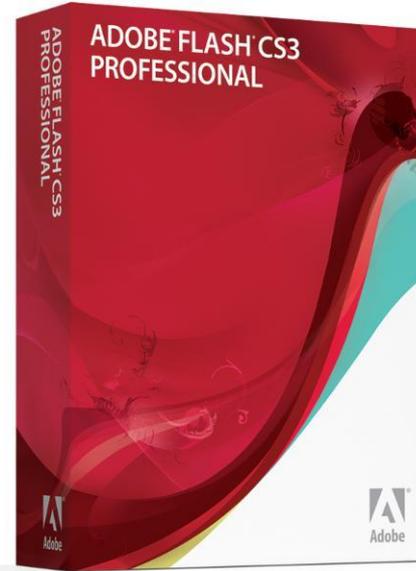
# TEKNIK MULTIMEDIA



**Dosen Pembina :**

Bella Hardiyana, S. Kom, M. Kom



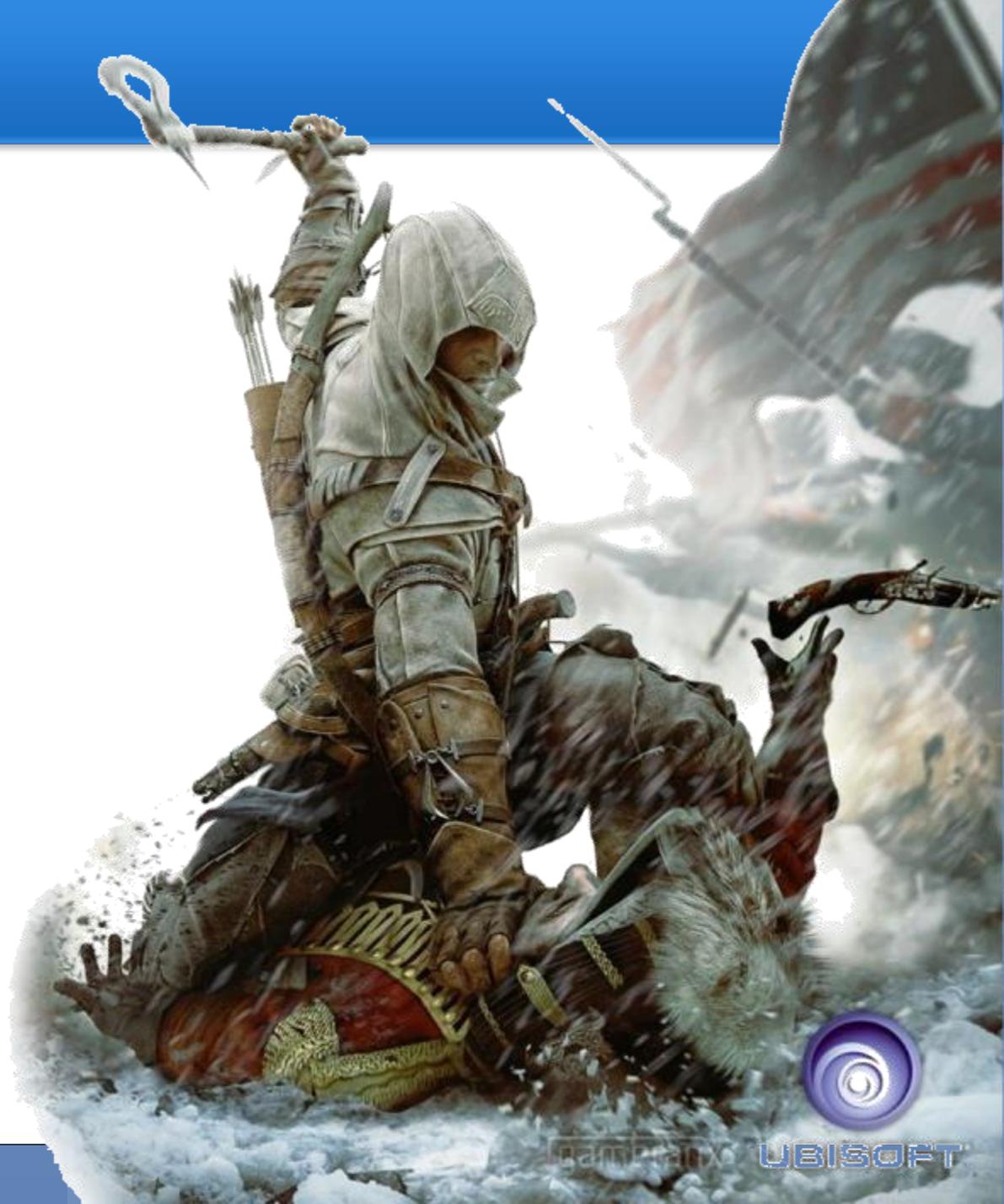


# Chapter IV

## ACTIONSCRIPT 3.0



# OVERVIEW





# Overview

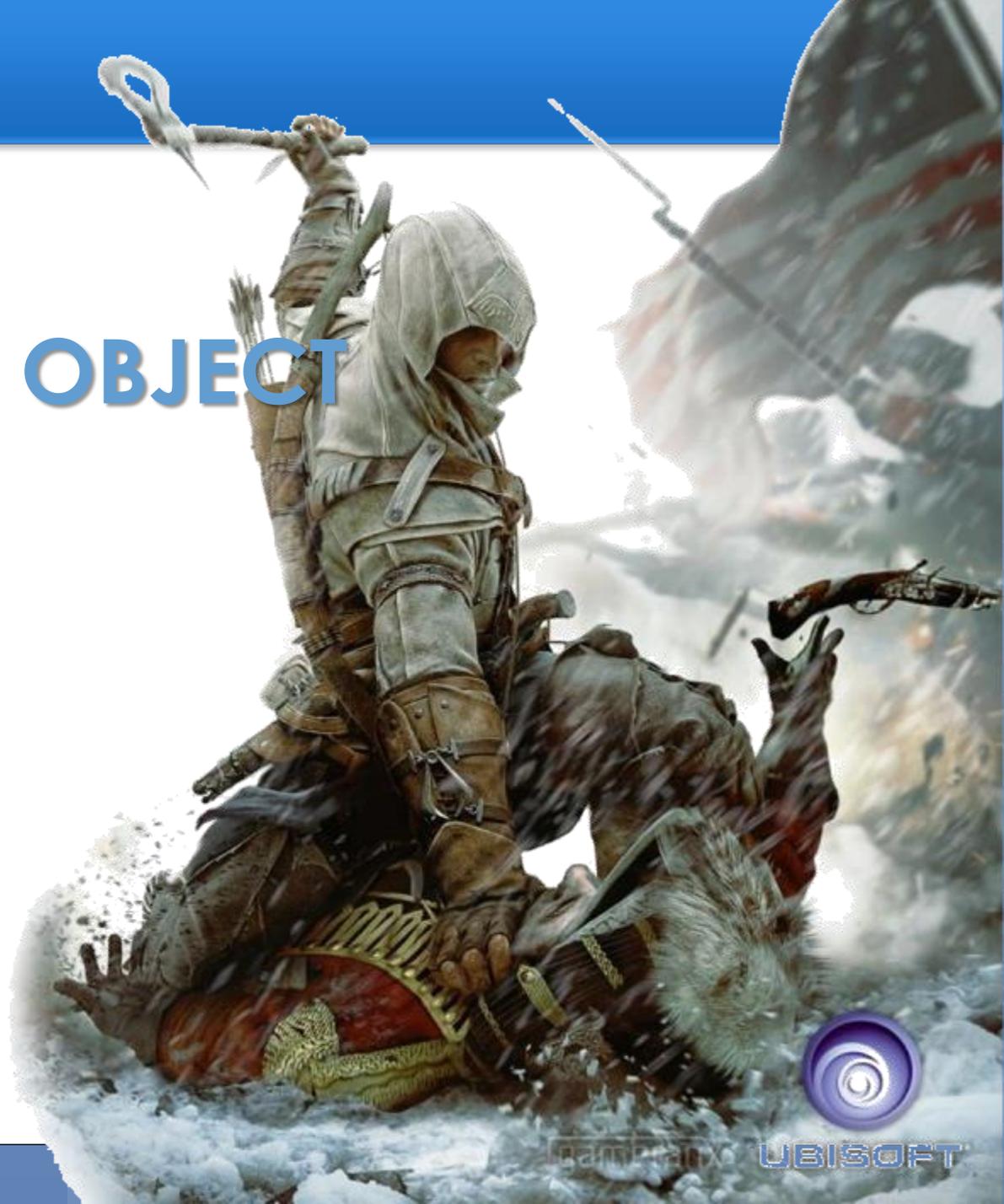
Sebelum bermigrasi ke AS 3.0, pertemuan ini akan melakukan review kembali tentang konsep OOP. Sebelumnya, konsep OOP pada Flash telah dibahas pada pertemuan sebelumnya, yaitu "SEKILAS TENTANG ACTIONSCRIPT 2.0". Review konsep OOP ini dimaksudkan agar programmer lebih memahami lebih dalam tentang konsep OOP yang diterapkan pada AS 3.0.

Pertemuan ini juga akan membahas konsep programming dari AS 3.0. Pada bagian ini akan dipaparkan perbedaan penulisan script antara AS 2.0 dan AS 3.0.





# CLASS DAN OBJECT





# Perbedaan Antara Class dan Object (1)



1. Mari membangun rumah



2. Meminta arsitek untuk mendesain rumah.



3. Proses Membangun rumah.



4. Rumah Selesai dibangun

*Class* dan *object* memang sedikit mirip, tapi sebenarnya kedua konsep ini sangatlah berbeda. Perbedaan antara *class* dan *object* sebenarnya sederhana. *Class* adalah *blueprint* (cetak biru)/ rancangan/ desain sebelum dibuatkannya objek/benda. Sedangkan objek adalah instansiasi/ perwujudan berdasarkan dari *blueprint* (cetak biru)/ rancangan/ desain tersebut. Untuk lebih memahami perbedaan antara keduanya, perhatikanlah ilustrasi disamping





# Perbedaan Antara *Class* dan *Object* (2)

Untuk lebih memahami perbedaan kelas dan objek, perhatikan contoh yang sederhana dibawah ini.



**desain rumah**

**rumah jadi**

Sehingga, desain rumah disebut kelas (*class*), sedangkan rumah yang sudah jadi disebut objek (*object*). Rumah yang dibangun pasti memiliki tipe, bentuk, dsb yang sama dengan apa yang dirancang. Sedangkan proses pembangunan rumah disebut proses instansiasi/penyediaan kelas menjadi objek.





# Mengenal Kelas, Objek, Atribut dan Metode

**Kelas** adalah desain/ rancangan/ *blue print* dengan menggambarkan *attribute* dan *metode* sebelum nantinya diinstansiasi menjadi objek.

**Objek** adalah hasil perwujudan/instansiasi dari kelas, yang tentu saja memiliki atribut dan metode yang sama dengan kelas-nya. **Atribut** adalah ciri yang melekat pada suatu objek. **Metode** adalah kemampuan yang dimiliki oleh suatu objek.

Semua yang ada didunia ini bisa disebut objek, dari makhluk hidup sampai makhluk mati bahkan objek sendiri tidak selalu berwujud benda ataupun makhluk. Selama memiliki atribut dan metode.

Cara mudah untuk membedakan Kelas, Atribut dan Metode dari segi penamaan.

**Kelas** : selalu diawali dengan huruf besar serta memiliki nama yang sama dengan nama file-nya.

**Atribut** : variabel yang dideklarasikan dibawah sintaks **class** dan selalu diawali huruf kecil.

**Metode** : penamaannya diawali dengan huruf kecil dan memiliki tanda kurung **()**.





# Contoh objek

**Objek :**

Kucing

**Atribut/Properti :**

Warna bulu, umur kucing, warna mata, dsb.

**Metode :**

Mencakar, berlari, bermain, makan, minum, memanjat pohon, berburu, dsb.

**Objek :**

Handphone

**Atribut/Properti :**

Merk, Tipe, Harga, VGA, dsb.

**Metode :**

Telpon, SMS, MMS, pemutar musik, pemutar video, memotret, dsb.

**Objek :**

Yahoo Messenger

**Atribut/Properti :**

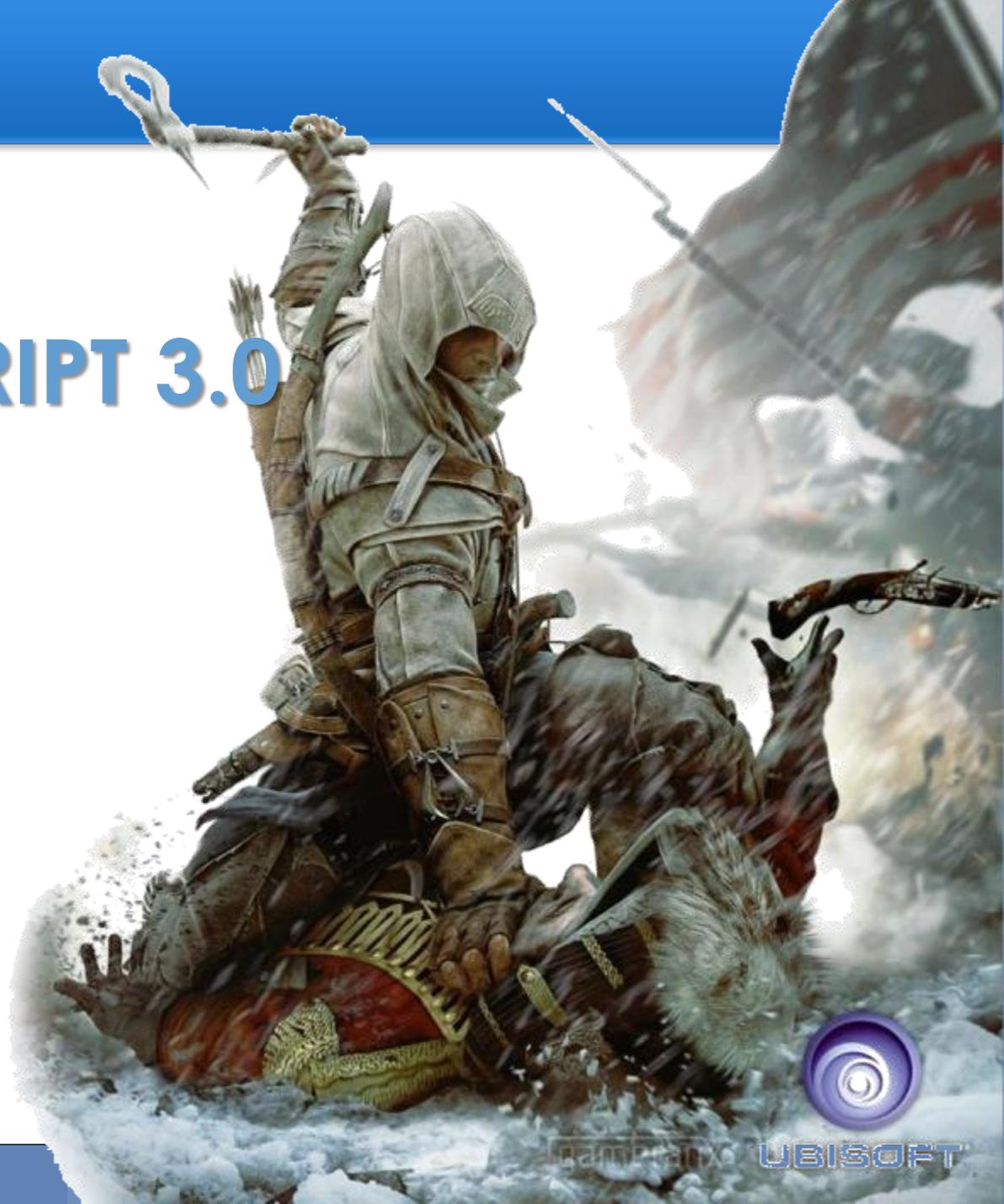
Versi, pengaturan koneksi, dsb.

**Metode :**

Chatting lewat teks, kirim gambar, web chat, dsb.



# ACTIONSCRIPT 3.0





# Deklarasi Variabel

Berikut ini cara mendeklarasikan variabel versi AS 3.0

```
var nama = "Edward Kenway";  
var aktif:* = true;  
var nilai:Number = 19;  
var prodi:Array = new Array("SI", "IF", "TK", "KA");
```

Coba amati sintaks diatas dan cari apa perbedaannya





# Kelas dan Paket

Sama halnya dengan AS 2.0, kelas pada AS 3.0 ditempatkan pada file AS eksternal. AS 3.0 juga mengenal konsep package atau struktur folder tempat menyimpan file AS eksternal. Penamaan file AS-pun harus sama dengan nama kelas.

Misalkan, terdapat kelas **Bola** yang disimpan dalam file **Bola.as**. File ini disimpan dalam folder **sport**. Maka teknik pemanggilan yang paling mudah dengan melakukan import semua kelas yang terdapat pada package sport.

```
import sport.*;
```

Atau dengan memanggil kelas yang spesifik.

```
import sport.Bola;
```





# Latihan (1)

1. Buat sebuah folder bernama **sport**
2. Buat sebuah file bertipe **AS File** dengan nama **Bola.as**
3. Berikut ini sintaks yang ada pada **Bola.as**

```
package sport{  
  public class Bola{  
    function Bola(nama, warna, bentuk, berat){  
      trace("Pembuatan bola "+nama+" sukses");  
      trace("Warna   : "+warna);  
      trace("Bentuk  : "+bentuk);  
      trace("Berat   : "+berat);  
    }  
  }  
}
```

4. Pastikan **Bola.as** dalam kondisi telah tersimpan





## Latihan (2)

5. Buat sebuah file bertipe **Flash File (AS 3.0)** dengan nama **Terserah anda** dan disimpan dalam folder yang sama dengan folder **sport**.
6. Berikut ini sintaks yang ada pada **Flash File** tersebut

```
import sport.Bola;

var basket:Bola = new Bola(
    "Bola Basket", "orange", "bundar", "ringan");
var bowling:Bola = new Bola(
    "Bola Bowling", "hitam", "bundar", "berat");
var rugby:Bola = new Bola(
    "Bola Rugby", "coklat", "lonjong", "ringan");
```

7. Berikut tampilan outputnya

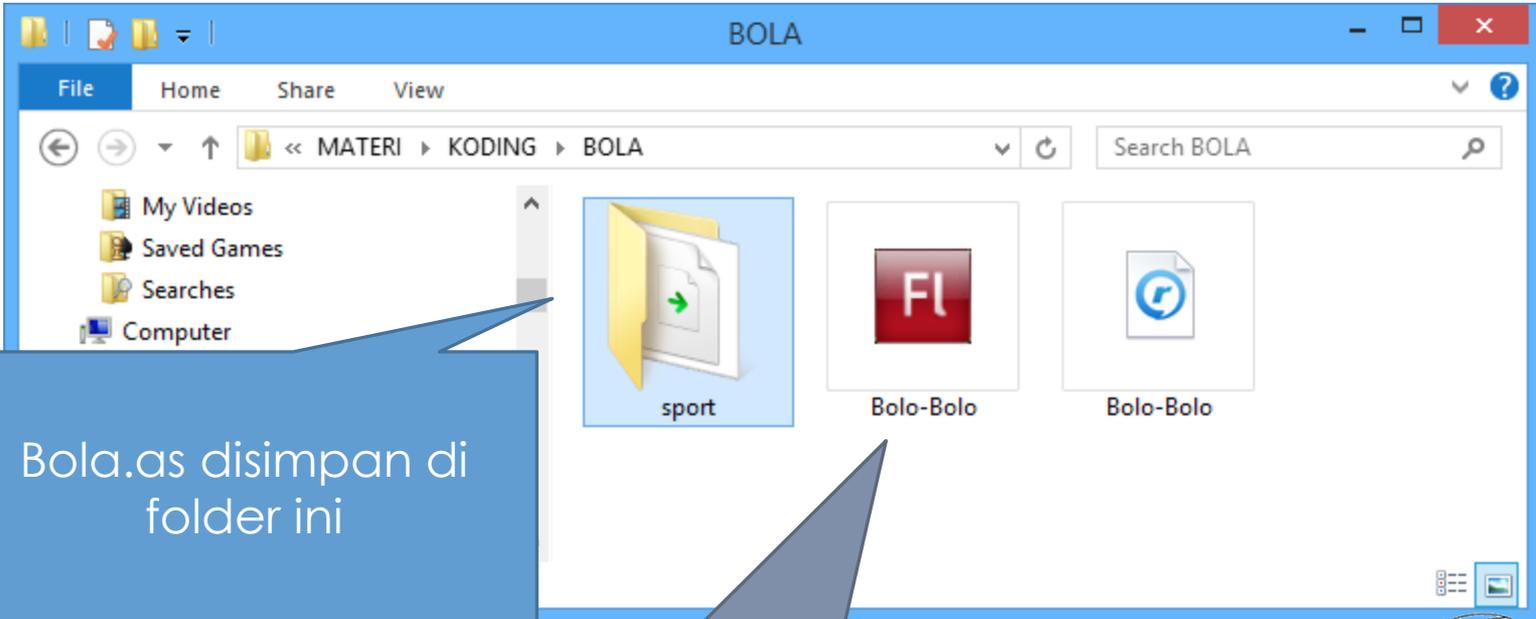
```
Pembuatan bola Bola Basket sukses
Warna : orange
Bentuk : bundar
Berat : ringan
Pembuatan bola Bola Bowling sukses
Warna : hitam
Bentuk : bundar
Berat : berat
Pembuatan bola Bola Rugby sukses
Warna : coklat
Bentuk : lonjong
Berat : ringan
```





# Latihan (3)

Pastikan posisi file dan folder seperti berikut ini



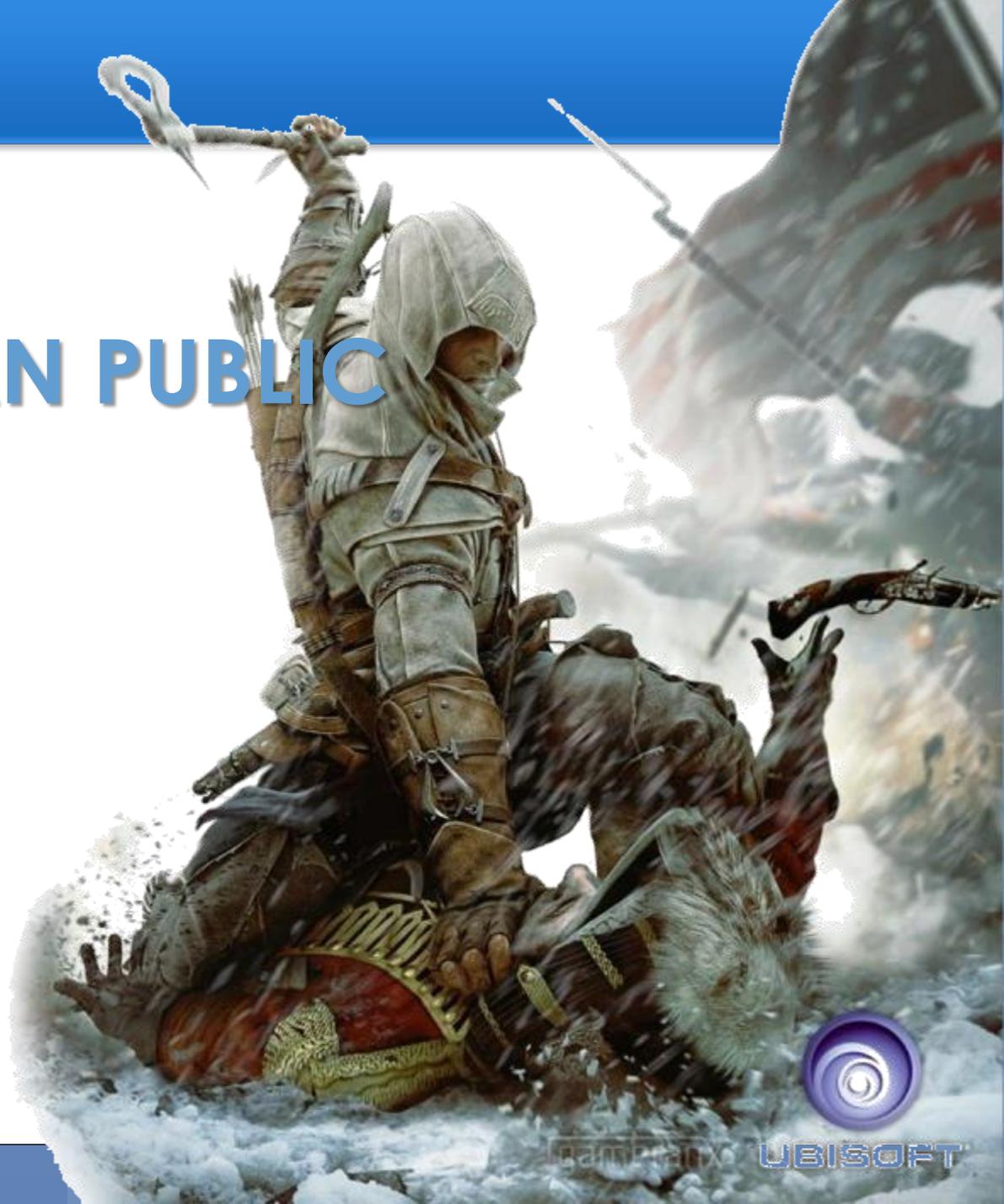
Bola.as disimpan di folder ini

Flash File (AS 3.0) disimpan dalam folder yang sama dengan folder sport





# PRIVATE DAN PUBLIC





# Latihan (1)

1. Buat sebuah **AS File** dengan nama **Domba.as**
2. Berikut ini sintaks yang ada pada **Domba.as**

```
package{
    import flash.utils.*;
    public class Domba{
        var dombaSkrng:uint;
        var maxDomba:uint;
        var hitungInterval:uint;

        function Domba(maxDomba:uint){
            this.dombaSkrng = 0;
            this.maxDomba = maxDomba;
        }

        public function mulaiHitung():void{
            hitungInterval = setInterval(dombaNambah, 1000);
        }

        private function dombaNambah():void{
            if(this.dombaSkrng < this.maxDomba){
                this.dombaSkrng++;
                trace(this.dombaSkrng+" domba terhitung");
            }else{
                trace("Domba bobo ...");
                clearInterval(this.hitungInterval);
            }
        }
    }
}
```





## Latihan (2)

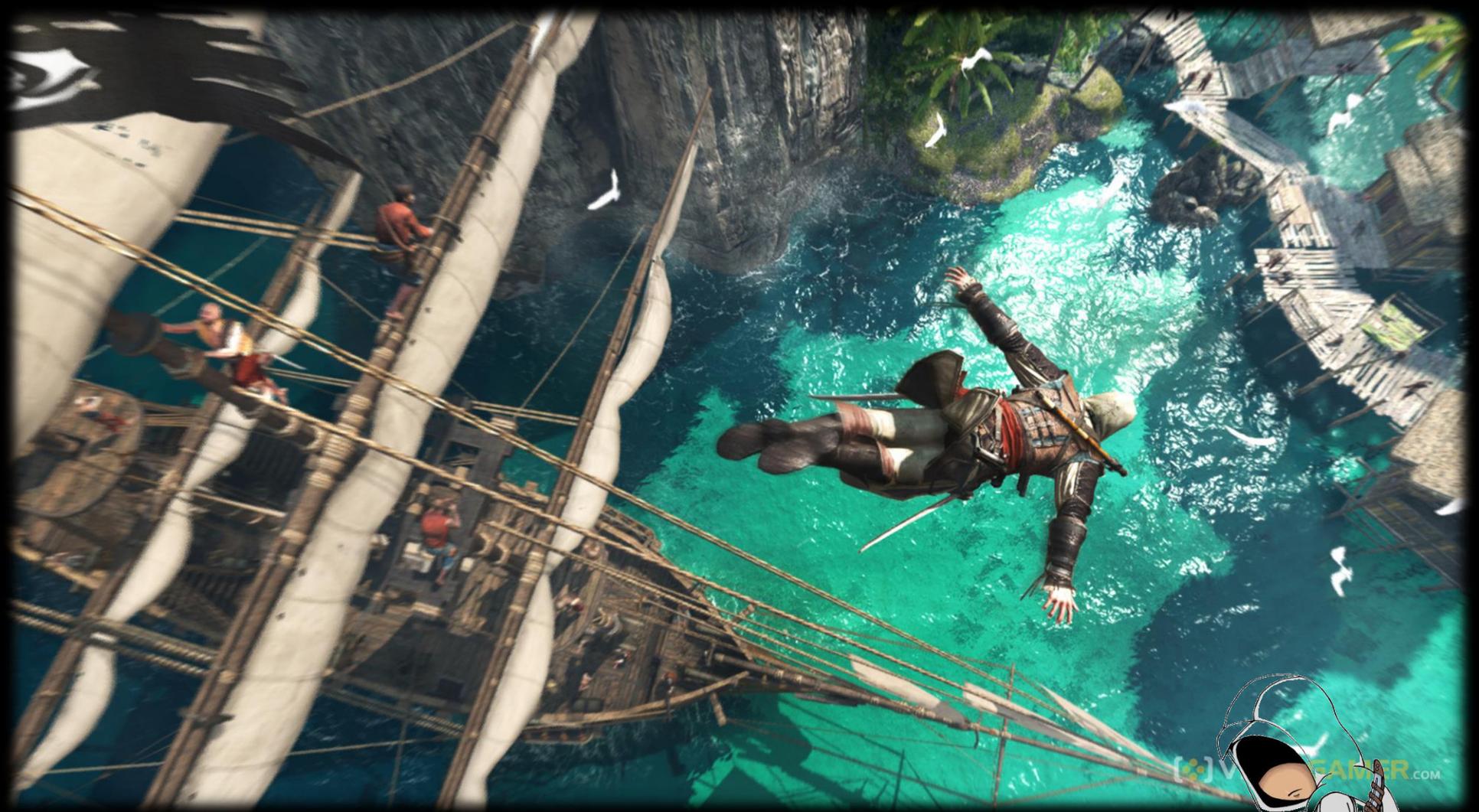
3. Buat sebuah file bertipe **Flash File (AS 3.0)** dengan nama **Terserah anda** dan disimpan dalam folder yang sama dengan file **Domba.as**
4. Berikut ini sintaks yang ada pada **Flash File** tersebut

```
var bobo = new Domba(10);  
bobo.mulaiHitung();  
//bobo.dombaNambah();
```

5. Berikut tampilan outputnya

```
1 domba terhitung  
2 domba terhitung  
3 domba terhitung  
4 domba terhitung  
5 domba terhitung  
6 domba terhitung  
7 domba terhitung  
8 domba terhitung  
9 domba terhitung  
10 domba terhitung  
Domba bobo ...
```





**PEMBAHASAN TENTANG ACTIONSCRIPT  
SELESAAAIIII !!!!**





**TERIMA KASIH !!!**

**MINGGU DEPAN KITA  
BELAJAR ANIMASI FLASH**

