

# OOP 2 - Java

# TOKEN

- Token adalah elemen terkecil di program yang masih memiliki arti.
- Ada 5 token dalam bahasa java yaitu
  - identifier,
  - keyword,
  - literal dan tipe data,
  - operator,
  - serta separator.

# Identifier

- Identifier adalah token yang merepresentasikan nama sesuatu.
- Sesuatu tersebut adalah variabel, atau konstanta, atau attribute, atau method, atau kelas, atau package, atau interface, atau nama file, dan lain-lain.
- Identifier bisa diawali dengan sembarang huruf, underscore (\_) atau dollar (\$).

```
1  -  /**
2      * @author xxx
3      */
4  public class Tes {
5  -      public static void main (String[] args) {
6              int Bilangan=0;
7              String $aku="namaku";
8              double __angka=3934.38;
9              short bil=123;
10         }
11     }
```

# Keyword (ada 51 keyword dalam JAVA)

abstract	continue	for	new	switch
boolean	default	goto	null	synchronized
break	do	if	package	this
byte	double	implements	private	threadsafe
byvalue	else	import	protected	throw
case	extends	instanceof	public	throws
catch	false	int	return	transient
car	final	interface	short	true
class	finally	long	static	try
const	float	native	super	void
				while

# Literal dan Tipe Data Primitif

- Literal adalah nilai variabel/attribute atau nilai konstanta atau nilai objek data.
- Ada tiga besaran literal dalam java yaitu angka, karakter, dan string.
- Angka terdiri dari byte, short, int, long, float, double, dan boolean (dianggap angka true = 1 atau false = 0)
- Semua variabel dan konstanta yang akan digunakan harus dipesan terlebih dahulu dalam deklarasi.

# Literal dan Tipe Data Primitif

- Bentuk umum :

*TipeData namaVar = ungkapan\_atau\_nilai;*

*TipeData namaVar1, namaVar2, ...;*

*[modifier] static final TipeData NAMA KONSTANTA = nilai;*

Contoh deklarasi :

*double a=3, b=4;*

*double c = Math.sqrt(a\*a+b\*b);*

*static final PHI=3.14;*

*static final double CM\_PER\_INC = 2.54;*

# Tabel Tipe Data Primitif

Tipe Data Primitif	Jangkauan	Ukuran (bit)
byte	-128 s/d 127	8
short	-32767 s/d 32767	16
int	-2147483648 s/d 2147483647	32
long	-9223372036854775808 s/d 9223372036854775807	64
char	sebuah Unicode	16
float	3.4e-038 s/d 3.4e+038	32
double	1.7e-308 s/d 1.7e+308	54
boolean	false = 0 atau true = 1	8



# Operator

- Operator melakukan komputasi terhadap satu/dua objek data.
- Operan yang dioperasikan dapat berupa literal, variabel, atau nilai yang dikirim method.

# Operator

Prioritas	Kelompok Operator	Keterangan
1	. [] ()	Sekaligus
2	++var, --var, ~, instanceof	preincrement, predecrement, unary, instance dari kelas ...
3	(type) (casting)	
4	!	Not
5	*, /, %	perkalian, pembagian, modulus
6	+, -	penjumlahan, pengurangan
7	<<, >>, >>>	geser untuk bil biner
8	<, >, <=, >=	pembandingan
9	==, !=	kesamaan, ketidaksamaan
10	&	and
11	^	exclusive or
12		unconditional or
13	&&	conditional and
14		conditional or
15	? :	shorthand untuk if..then...else...
16	=, +=, -=, *=, /=, %=, ^=	operator penugasan
17	&=,  =, <<=, >>=, >>>=	operator penugasan
18	var++, var--	postincrement, postdecrement

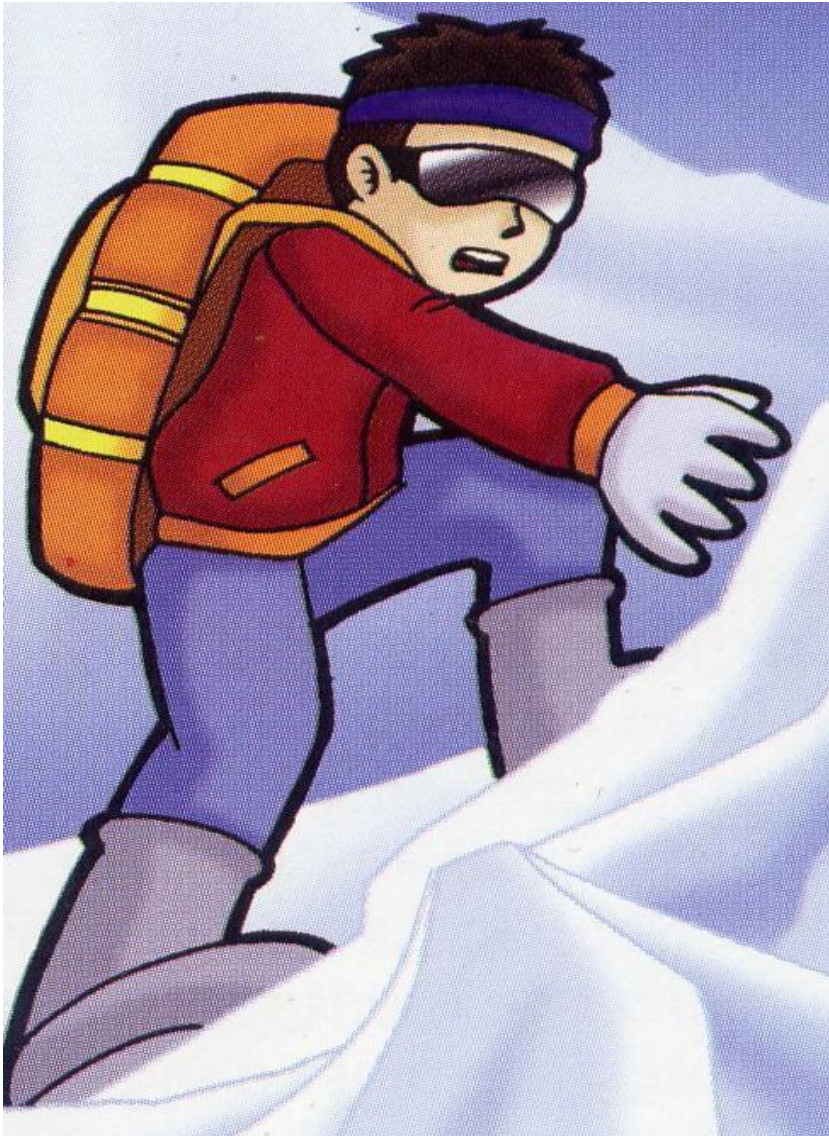
# Seperator

- Separator menginformasikan ke compiler java mengenai adanya kelompok kode program.

Notasi	Nama	Deskripsi
(...)	kurung	mengelompokkan parameter method.
{...}	kurung kurawal	mengelompokkan nilai-nilai suatu array, mendefinisikan blok kode kelas ataupun kode method.
[...]	kurung siku	mendeklarasikan tipe array
:	titik koma	mengakhiri pernyataan, merangkai pernyataan-pernyataan di dalam for.
,	koma	memisahkan identifier-identifier di bagian deklarsi variable.
.	titik	memisahkan nama-nama package, memisahkan kelas dari objek, dan objek dari method.

# Class dan Object

# Berorientasi Object?



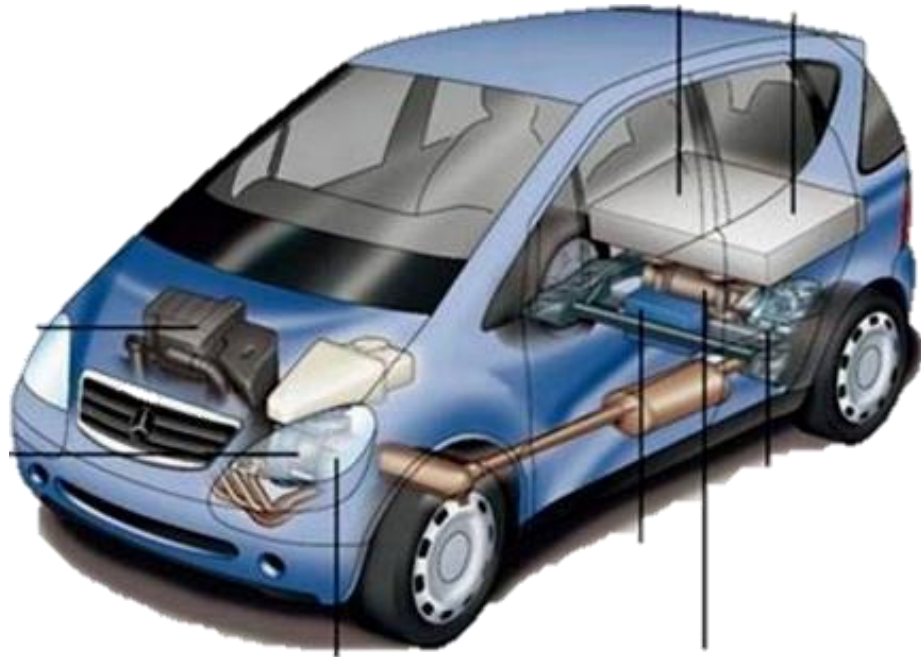
## **Attribute:**

Baju, celana, jaket,  
kacamata, tas punggung,  
sarung tangan, sepatu

## **Behavior:**

Cara Mendaki  
Cara Berlari  
Cara Berjalan  
Cara Memegang

# Berorientasi Object?



## Attribute (State):

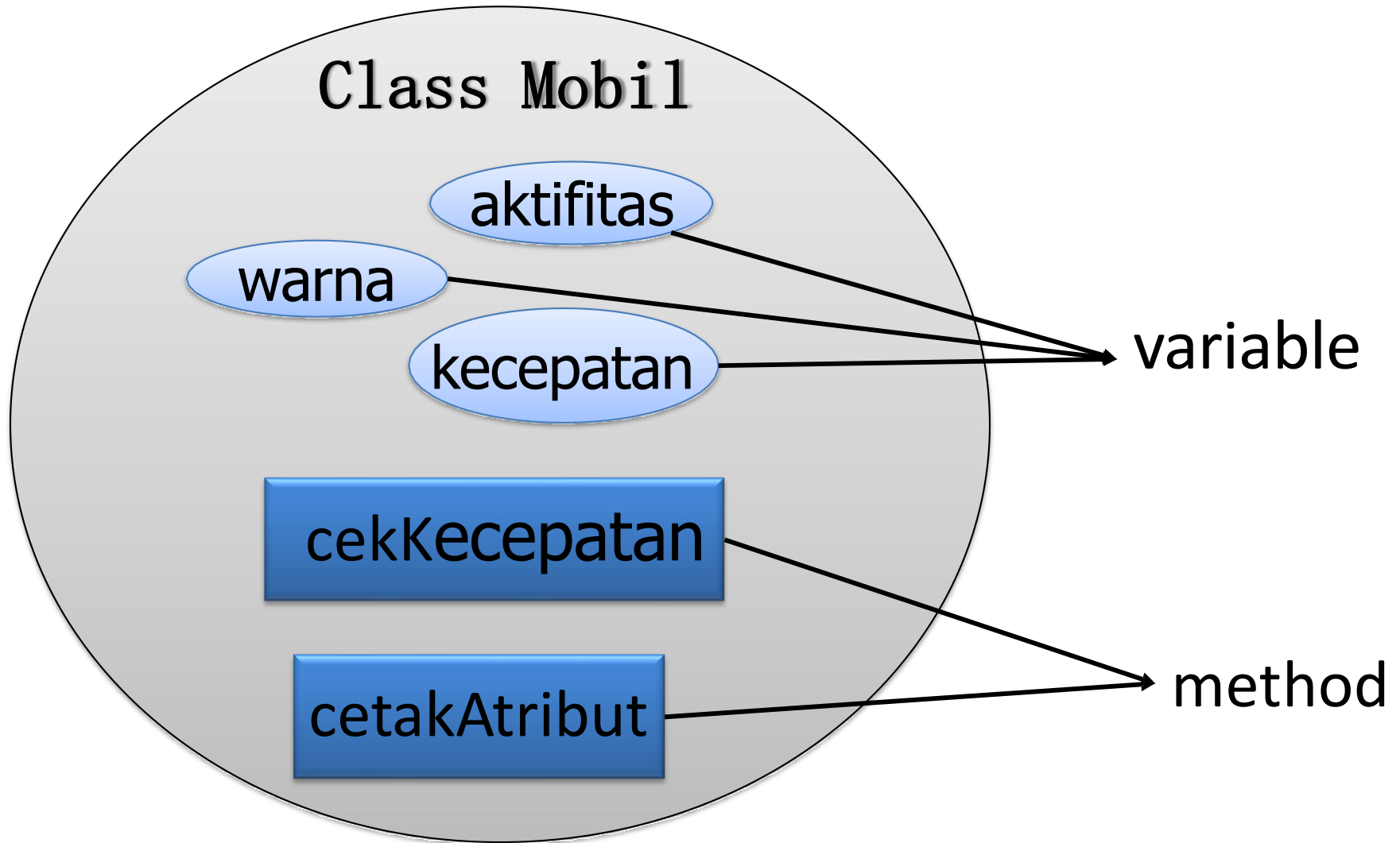
Ban, Stir, Pedal Rem, Pedal Gas,  
Warna, Tahun Produksi

## Behavior:

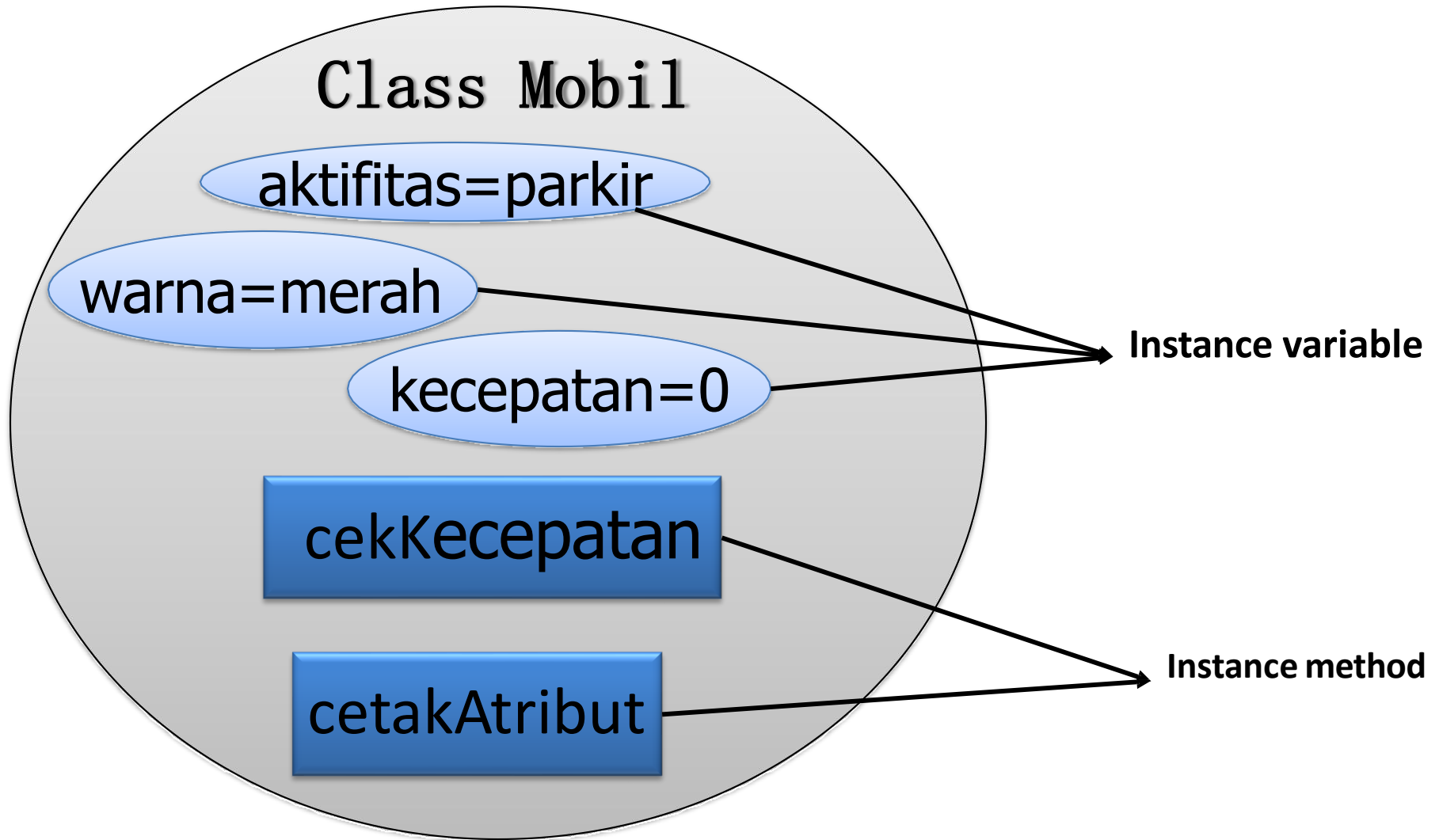
Cara Menghidupkan Mesin  
Cara Manjalankan Mobil  
Cara Memundurkan Mobil

Attribute ⑦	Variable(Member)
Behavior ⑦	Method(Fungsi)

# Class = Method + Variable



# Class = Method + Variable **Bernilai**





# Contoh Class Mobil

```
1 class Mobil {
2     // atribut
3     String aktifitas;
4     String warna;
5     int kecepatan;
6     // method untuk memeriksa kecepatan
7     void cekKecepatan() {
8         if (kecepatan==0)
9             aktifitas="parkir";
10    }
11    // method untuk mencetak atribut
12    void cetakAtribut() {
13        System.out.println("Aktifitas = "+aktifitas);
14        System.out.println("warna = "+warna);
15        System.out.println("Kecepatan = "+kecepatan);
16    }
17    public static void main(String [] args) {
18        // membuat obyek baru bernama mobilku
19        Mobil mobilku = new Mobil();
20        // memberi nilai awal pada atribut
21        mobilku.kecepatan=0;
22        mobilku.warna="merah";
23        // memeriksa kecepatan
24        mobilku.cekKecepatan();
25        // mencetak atribut
26        mobilku.cetakAtribut();
27    }
28 }
```

## Output

```
Aktifitas = parkir
warna = merah
Kecepatan = 0
```

# Method

- Method adalah implementasi dari bagaimana bekerjanya sebuah class
- Method melakukan:
  1. Manipulasi data
  2. Perhitungan matematika
  3. Memonitor kejadian dari suatu event

## Contoh methodMhs.java

```
public class methodMhs {  
    int npm;  
    String nama;  
  
    void dataMhs() {  
        System.out.println("NPM : " + npm);  
        System.out.println("Nama: " + nama);  
    }  
  
    public static void main(String[] args) {  
        // Membuat object  
        methodMhs mhs = new methodMhs();  
        // memanggil atribut dan memberi nilai */  
        mhs.npm = 1006700201;  
        mhs.nama= "Andi Gunawan";  
        mhs.dataMhs();  
    }  
}
```

# Konstruktor

- Method yang digunakan untuk memberi nilai awal pada saat object diciptakan
- Dipanggil secara otomatis ketika **new** digunakan untuk membuat instan class
- Sifat konstruktor:
  - Nama konstruktor sama dengan nama class
  - Tidak memiliki nilai balik

# Contoh konstruktorMhs.java

```
public class konstruktorMhs {  
    int npm;  
    String nama;  
  
    public konstruktorMhs(int npm, String nama) {  
        npm= npm;  
        nama = nama;  
        System.out.println("NPM   : " + npm);  
        System.out.println("Nama  : " + nama);  
    }  
  
    public static void main(String[] args){  
        konstruktorMhs mhs = new konstruktorMhs (1006700201,  
            "Andi Gunawan");  
    }  
}
```

# Contoh Kuadrat.java

```
public class Kuadrat {  
    int bilangan;  
    int hasil;  
    public Kuadrat(int nilai){  
        bilangan = nilai;  
    }  
    int getHasil(){  
        hasil = bilangan * bilangan;  
        return hasil;  
    }  
    public static void main(String[] args) {  
        Kuadrat kuadrat = new Kuadrat(8);  
        System.out.println( "8 * 8 = " + kuadrat.getHasil() );  
    }  
}
```

# Latihan

## 1. Buat class **Buku**

- Tentukan variable buku: **judul, pengarang, penerbit, tahun**
- Buat konstruktor class Buku dengan parameter: **judul, pengarang, penerbit, tahun**
- Buat method **cetakBuku**

## 2. Buat class **tesBuku** dan tampilkan 2 buku:

- Pemrograman Berbasis Objek dengan Java, Indraajani, Elexmedia Komputindo, 2007
- Dasar Pemrograman Java, Abdul Kadir, Andi Offset, 2006

# Latihan

- Buat Aplikasi untuk menghitung Volume Bangun Ruang menggunakan konstruktor dan method:
  - Volume Kubus
  - Volume Balok
  - Volume Bola