



PEMROGRAMAN BERORIENTASI OBJEK

Pertemuan 2 : Bahasa Java dan PBO dengan Java

Materi

- Pengenalan Bahasa Java
- Review, dan Pembuatan Program Berorientasi objek dalam java (Program Sederhana)



MATERI PERTAMA : PENGENALAN BAHASA PEMROGRAMAN JAVA

Pemrograman Berorientasi Objek

Apa yang Disebut Java ?

- Bahasa pemrograman berorientasi objek murni yang dibuat berdasarkan kemampuan-kemampuan terbaik bahasa pemrograman objek sebelumnya (C++, Ada, Simula).
- Diciptakan oleh James Gosling, developer dari Sun Microsystems pada tahun 1991.

Karakteristik Java

- Sederhana (Simple)
- Berorientasi Objek (Object Oriented)
- Terdistribusi (Distributed)
- Interpreted
- Aman (Secure)
- Architecture Neutral
- Portable
- Performance
- Multithreaded
- Dinamis

Perangkat Pemrograman Java

1. Compiler (Interpreter):

Java Standard Edition (JSE)

2. Code Editor:

1. Text Editor:

TextPad, Notepad++

2. Integrated Development Environment (IDE):

Netbeans, Eclipse, JCreator

Instalasi Java SE dan Netbeans IDE

1. Instalasi Java SE dengan mengklik:
jdk-7u21-windows-i586.exe
(download dari: <http://java.sun.com/javase/downloads>)
2. Instalasi Netbeans dengan mengklik: **netbeans-7.3-ml-windows.exe**
(download dari: <http://netbeans.org>)
3. Ikuti seluruh proses instalasi sampai selesai

Instalasi Text Editor dan Set Path

Set path dan instalasi text editor diperlukan untuk yang mengembangkan **aplikasi text-based** dengan console

1. Klik **Start** → **Control Panel** → **System** → **Advanced** → **Environment Variables** dan set system PATH:
;C:\Program Files\Java\jdk1.7.0_21\bin
2. Instal text editor untuk editing code:
textpad, notepad++, JCreator

How Java Works?

```
public class Hello
{
    public static void main(String[] args){
        System.out.println("Hello World!");
    }
}
```

Hello.java

↓ javac (java compiler)



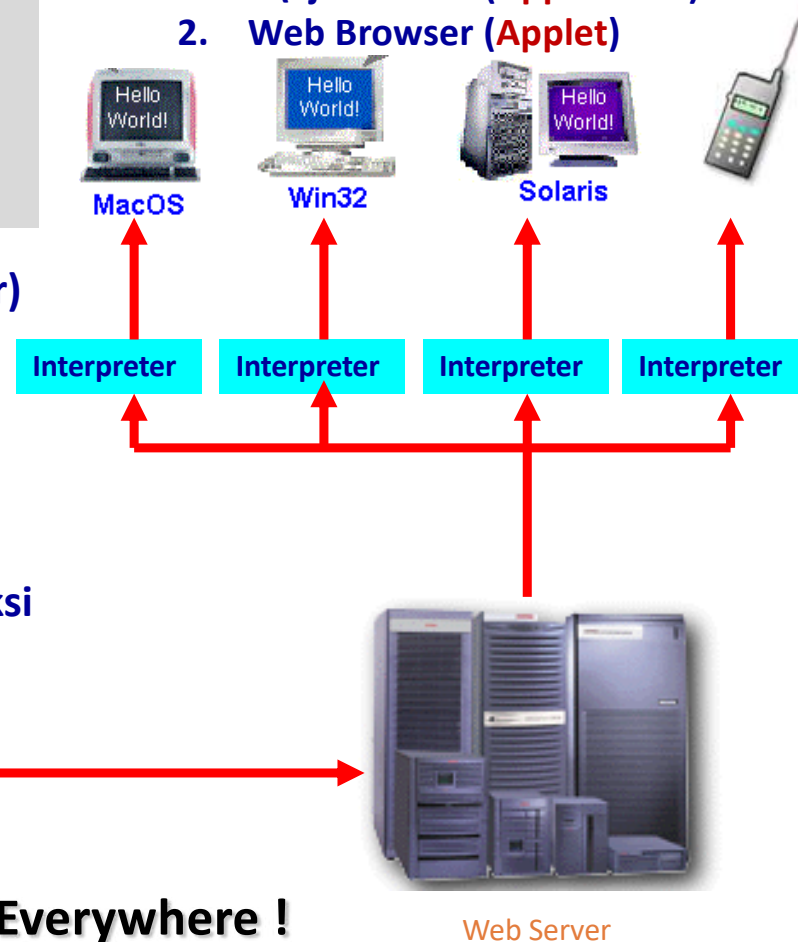
↓ Kompiler memproduksi
Bytecode (Class)

Hello.class

Write Once Run Everywhere !

Jalankan dengan:

1. C:\>java Hello (**Application**)
2. Web Browser (**Applet**)



Compile and Run Java Applet

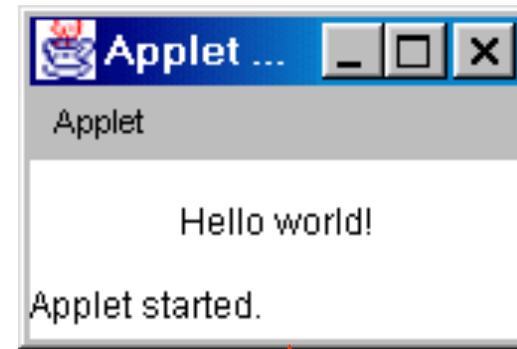
```
import java.applet.*;
import java.awt.*;

public class HelloWorld extends Applet {
    public void paint(Graphics g) {
        g.drawString("Hello world!",50,25);
    }
}
```

C:\javac HelloWorld.java

```
<HTML>
<HEAD>
<TITLE>A Simple Program</TITLE>
</HEAD>
<BODY>
Here is the output of my program:
<APPLET CODE="HelloWorld.class" WIDTH=150
HEIGHT=25>
</APPLET>
</BODY>
</HTML>
```

C:\appletviewer Hello.html



Applet on a Web Page

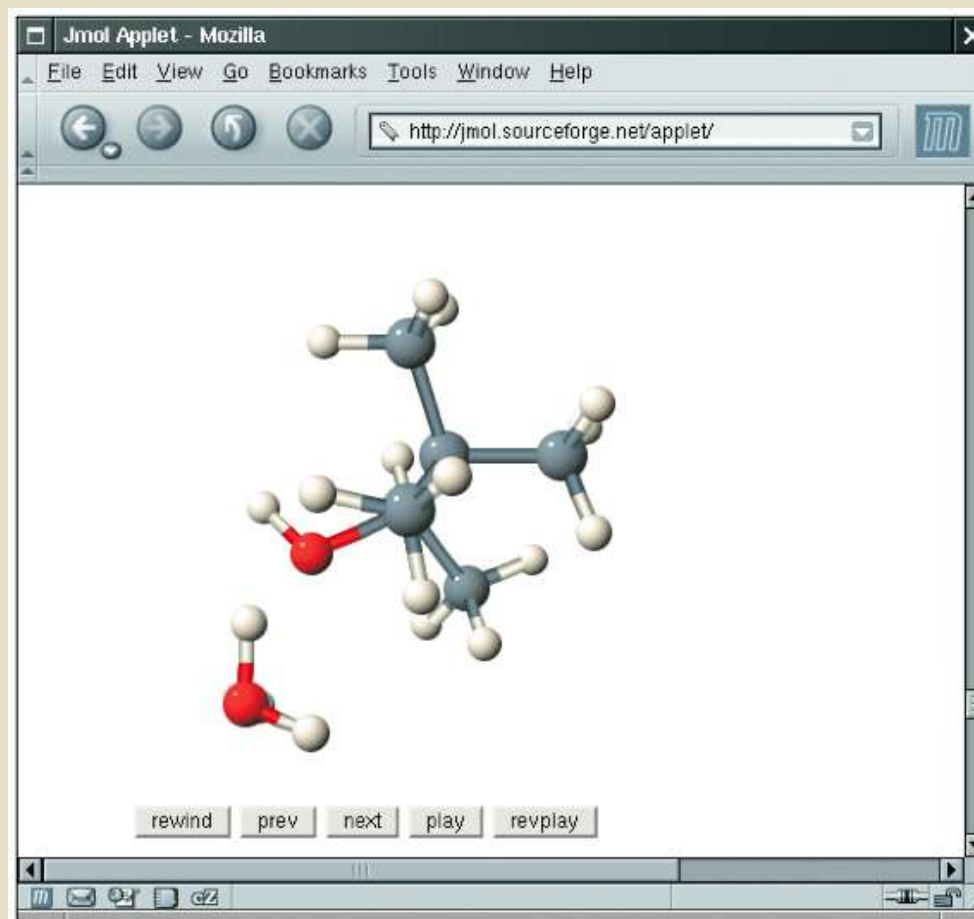


Figure 6 An Applet for Visualizing Molecules Running in a Browser (<http://jmol.sourceforge.net/applet/>)

Menulis Program Java

- Bentuk program:

1. **Text-Based Console Application** → menggunakan library non-GUI di Java
2. **GUI Application** → menggunakan AWT atau Swing untuk library GUI

- Suatu class bisa dieksekusi karena memiliki method **main**

public static void main(String[] args)

→ **Program Java mulai dari sini!**

Struktur Program Java

Program Java (.java)

Komentar program

Deklarasi package dan import

// Kelas pertama

```
class NamaKelas {  
    <pernyataan>  
}
```

// Kelas lainnya

```
class NamaKelasLain {  
    <pernyataan>  
}
```

- Dapat ditulis dalam **satu** file .java:
 - Berisi satu kelas
 - Berisi beberapa kelas
- Ditulis dalam **beberapa** file .java:
 - Satu file satu kelas
 - Satu file banyak kelas
- Hanya boleh ada satu fungsi utama pada setiap program objek.

Aplikasi Java

- Stand alone
- Applets
- Aplikasi berbasis Windows (GUI)
- Java Servlet
- Java Server Pages (JSP)
- Java Beans dan Enterprise Java Beans
- Java Micro Edition

Kompilasi dan Eksekusi

- Kompilasi

javac <namafile.java>↵

Contoh:

javac HelloWorld.java↵

- Eksekusi

java <namafile hasil kompilasi> [<argumen baris perintah>]↵

Contoh:

java HelloWorld↵

java HelloWorld "Ini argumen baris perintah"↵

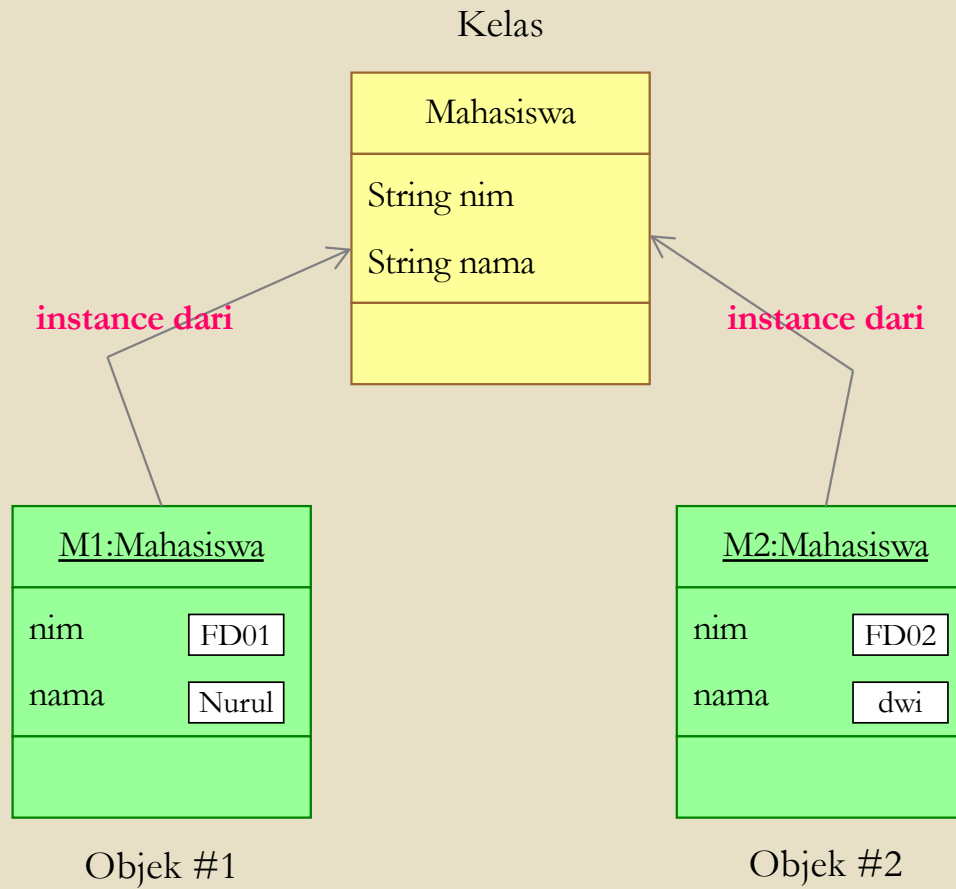


MATERI 2 : PEMBUATAN PROGRAM BERORIENTASI OBJEK DENGAN JAVA

Perbedaan Class dan Object

- Class: **konsep** dan **deskripsi** dari sesuatu
 - Class mendeklarasikan **method** yang dapat digunakan (dipanggil) oleh object
- Object: **instance dari class**, bentuk (contoh) nyata dari class
 - Object memiliki sifat **independen** dan dapat digunakan untuk memanggil method
- Contoh Class dan Object:
 - Class: **mobil**
 - Object: **mobilnya pak Joko, mobilku, mobil berwarna merah**

Kelas dan Objek (lanjutan)



Class = Method + Variable

Class Sepeda

gir

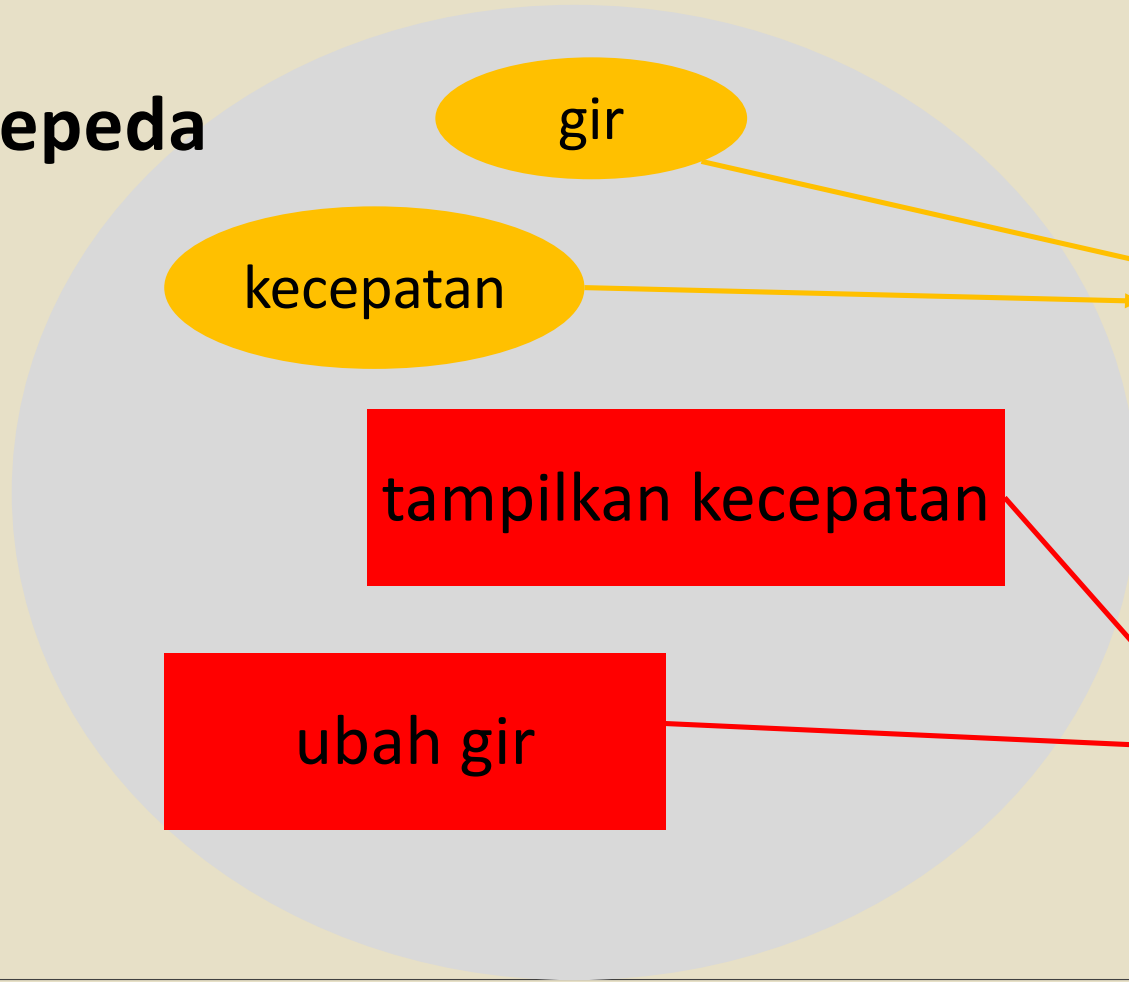
kecepatan

tampilkan kecepatan

ubah gir

variable

method



Object = Method + Variable yg Memiliki Nilai

Object Sepedaku

gir = 3

kecepatan = 10km/jam

tampilkan kecepatan ()
kecepatan = 10 km/jam

ubah gir (2)
gir = 5

instance variable

instance
method

Property Kelas / Objek

- Atribut

Data yang mendeskripsikan kelas/objek.

Pada umumnya didefinisikan dengan *access modifier* **private** (konsep *information hiding*).

- Data dengan tipe data dasar: **integer**, **float**, **boolean**, **char**
- Data majemuk: **array**, **record**, dll.
- **Kelas** lain

- Layanan / Metode (Method)

Prosedur atau **fungsi** yang mencirikan sifat atau kelakuan dari kelas/objek.

Pada umumnya didefinisikan dengan *access modifier* **public**.

- Konstruktor: penamaan **sama** dengan nama kelasnya
- Accessor: penamaan diawali dengan kata **get**
- Mutator: penamaan diawali dengan kata **set**
- Prosedur atau fungsi tertentu: penamaan **bebas**
- Program utama: harus menggunakan nama **main()**

METHOD : CONSTRUCTOR

- Fungsi yang akan dijalankan/dieksekusi secara otomatis begitu sebuah objek diciptakan.
- Constructor merupakan method yang dieksekusi untuk melakukan inisialisasi nilai suatu atribut objek dengan nilai *default*.
- Copy Constructor (Cctor) merupakan method yang dieksekusi untuk melakukan inisialisasi nilai suatu atribut objek berdasarkan *parameter tertentu*.
- Diberi nama sesuai dengan nama kelasnya.

Contoh:

```
public SegiEmpat() {  
    // pernyataan  
}
```

```
public SegiEmpat(int p, int l) {  
    // pernyataan  
}
```

METHOD : SELECTOR/ACCESSOR

- Fungsi yang mempunyai kegunaan khusus untuk **mengambil nilai** dari atribut yang dimiliki sebuah objek.
- Pada umumnya menggunakan penamaan dengan format:
getNamaAtribut

Contoh:

```
public int getPanjang() {  
    return (this.panjang);  
}
```

METHOD : MUTATOR

- Fungsi yang mempunyai kegunaan khusus untuk **memberi nilai** kepada atribut yang dimiliki sebuah objek.
- Pada umumnya menggunakan penamaan dengan format:
setNamaAtribut

Contoh:

```
public void setPanjang(int p) {  
    this.panjang = p;  
}
```


FUNCTION main()

- Berfungsi sebagai **program utama** yang akan menjadi **titik awal eksekusi**.
- Format penulisan:

```
public static void main(String arg[]) { ... }
```

- Isi fungsi main() pada umumnya adalah:
 - Penciptaan (instansiasi) objek
 - Manipulasi objek: penyalinan (copy), penggunaan

Deklarasi Kelas

```
[public] class NamaKelas {  
    // Deklarasi atribut  
    ...  
  
    // Definisi layanan/method  
    <constructor>  
    <accessor>  
    <mutator>  
    <fungsi lain>  
  
    // Program utama  
    public static void main(String arg[]) {  
        ...  
    }  
}
```

Penciptaan Objek

- Deklarasi reference atau variabel:
Mahasiswa m;
- Alokasi objek ke variabel:
m = new Mahasiswa();

Atau

- Deklarasi dan sekaligus alokasi objek:
Mahasiswa m = new Mahasiswa();

Membuat dan Memanggil Method

```
public class Mobil2{  
    String warna;  
    int tahunProduksi;  
  
    void printMobil(){  
        System.out.println("Warna: " + warna);  
        System.out.println("Tahun: " + tahunProduksi);  
    }  
}
```

Mobil2.java

```
public class Mobil2Beraksi{  
    public static void main(String[] args){  
        Mobil2 mobilku = new Mobil2();  
  
        mobilku.warna = "Hitam";  
        mobilku.tahunProduksi = 2006;  
        mobilku.printMobil();  
    }  
}
```

Mobil2Beraksi.java

Kata Kunci this

Digunakan pada pembuatan class dan digunakan untuk **menyatakan object sekarang**

```
public class Mobil{  
    String warna;  
    int tahunProduksi;  
  
    void isiData(String warna,  
                 int tahunProduksi){  
  
        this.warna = warna;  
        this.tahunProduksi = tahunProduksi;  
    }  
}
```

Contoh : Kegunaan Kelas

```
11 public class Ayah {
12     private String nama;
13
14     public String getNama(){
15         return nama;
16     }
17     public void setNama(String input){
18         if (input.length() > 5){
19             nama = input;
20         }
21         else{
22             nama = "Anonim";
23         }
24     }
25 }
```

```
11 public class Konsep_PBO {
12
13     /**
14      * @param args the command line arguments
15      */
16     public static void main(String[] args) {
17         // TODO code application logic here
18         Ayah a1 = new Ayah();
19         a1.setNama("Superman");
20         System.out.println("Hello " + a1.getNama());
21     }
22 }
```

Contoh Kegunaan Konstruktor dan Copy Konstruktor

```
11 public class Ayah {
12     private String nama;
13
14     public Ayah() { //constructor
15         nama = "default";
16     }
17
18     public Ayah(String nama) { //copy constructor
19         this.nama = nama;
20     }
21
22     public String getNama() { ... }
25     public void setNama(String input) { ... }
33 }
```

**Perhatikan
Penggunaan
this**

```
16 public static void main(String[] args) {
17     // TODO code application logic here
18     Ayah a1 = new Ayah();
19     Ayah a2 = new Ayah("Keren");
20
21     System.out.println("Halo " + a1.getNama() + " dan " + a2.getNama());
22 }
```

Latihan

- Buat class **Handphone**,
 - Class Handphone berisi **empat method** di bawah:
 1. `hidupkan()`
 2. `lakukanPanggilan()`
 3. `kirimSMS()`
 4. `matikan()`
 - Isi masing-masing method dengan tampilan status menggunakan **`System.out.println()`**
- Buat class **HandphoneBeraksi**, dan panggil method-method diatas dalam class tersebut

Latihan: Hasil Tampilan

Handphone hidup ...

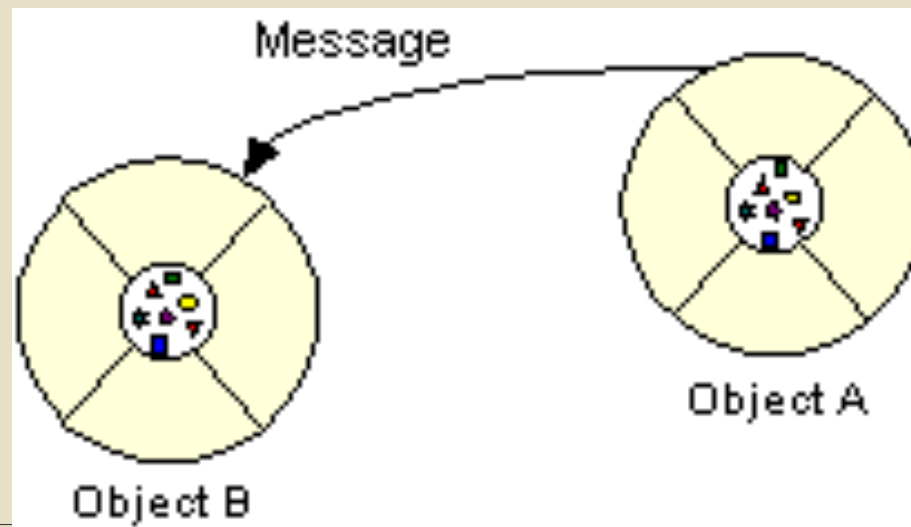
Kring, kring, kring ... panggilan dilakukan

Dung, dung ... sms berhasil terkirim

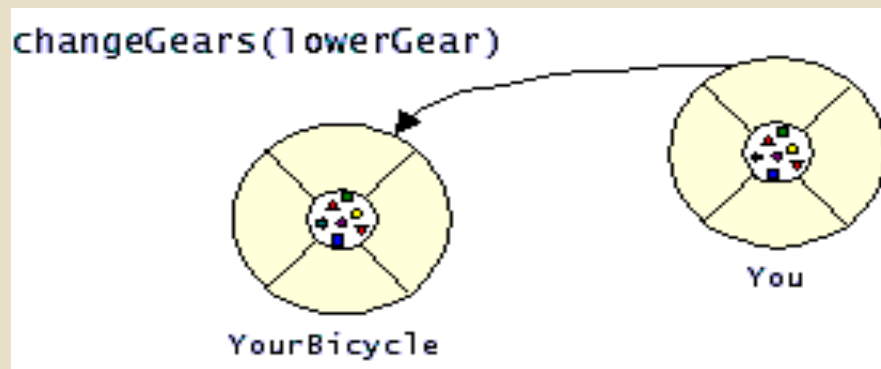
Handphone mati ...

Konsep : Parameter

- Sepeda akan berguna apabila ada object lain yang berinteraksi dengan sepeda tersebut
- Object software berinteraksi dan berkomunikasi dengan object lain dengan cara mengirimkan message atau pesan
- Pesan adalah suatu method, dan informasi dalam pesan dikenal dengan nama parameter



Pengiriman Pesan dan Parameter



1. **You** → object pengirim
2. **YourBicycle** → object penerima
3. **changeGears** → pesan berupa method yang dijalankan
4. **lowerGear** → parameter yang dibutuhkan method (pesan) untuk dijalankan

Sepeda.java

```
public class Sepeda{  
    int gir;  
  
    // method (mutator) dengan parameter  
    void setGir(int pertambahanGir) {  
        gir= gir+ pertambahanGir;  
    }  
  
    // method (accessor)  
    int getGir() {  
        return gir;  
    }  
}
```

SepedaBeraksi.java

```
public class SepedaBeraksi{  
    public static void main(String[] args) {  
        Sepeda sepedaku = new Sepeda();  
  
        sepedaku.setGir(1); // menset nilai gir = 1 (sebelumnya 0)  
        System.out.println("Gir saat ini: " + sepedaku.getGir());  
  
        sepedaku.setGir(3); // menambahkan 3 pada posisi gir saat ini (1)  
  
        System.out.println("Gir saat ini: " + sepedaku.getGir());  
    }  
}
```

Latihan: Class Matematika dan Parameter

- Buat Class bernama **Matematika**, yang berisi method dengan **dua parameter**:
 - `pertambahan(int a, int b)`
 - `pengurangan(int a, int b)`
 - `perkalian(int a, int b)`
 - `pembagian(int a, int b)`
- Buat Class bernama **MatematikaBeraksi**, yang mengeksekusi method dan menampilkan:
 - Pertambahan: $20 + 20 = 40$
 - Pengurangan: $10 - 5 = 5$
 - Perkalian: $10 * 20 = 200$
 - Pembagian: $21 / 2 = 10$

Variasi Tampilan

```
void pertambahan(int a, int b){  
    System.out.println(a + " + " + b + " = " + (a+b))  
}
```

```
void pertambahan(int a, int b){  
    System.out.println("Hasil = " + (a+b))  
}
```

```
void pertambahan(int a, int b){  
    int hasil = a + b;  
    System.out.println("Hasil = " + hasil)  
}
```

Latihan: Buatlah Class BangunDatar

- Buat Class bernama **BangunDatar**, yang berisi method :
 - Constructor default
 - Constructor dengan parameter sisi
 - Constructor dengan parameter panjang dan lebar
 - Assesor dan Mutator untuk variable sisi, Panjang, lebar
 - Method hitungLuasPersegi, hitungLuasPersegiPanjang, hitungKelilingPersegi, dan hitungKelilingPersegiPanjang.
- Buat Class bernama **BangunDatarBeraksi** yang memiliki method main, dengan di dalamnya terdapat :
 - Pembuatan objek persegi dengan constructor parameter sisi
 - Pembuatan objek persegiPanjang1 dengan constructor parameter default
 - Pembuatan objek persegiPanjang2 dengan constructor parameter panjang dan lebar
 - Perhitungan luas dan keliling dari masing-masing objek. Dan outputkan hasilnya dengan detail, misalnya : “luas persegi panjang dengan panjang 10 dan lebar 5 yaitu = 50”, dsj.

TERIMA KASIH