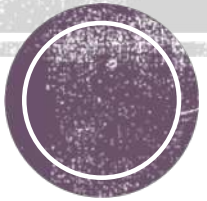
Pemrograman Berorientasi Objek :

Konsep Pemrograman Berorientasi Objek, PBO Pada Java.



Royana Afwani

Bahasa Pemrograman?

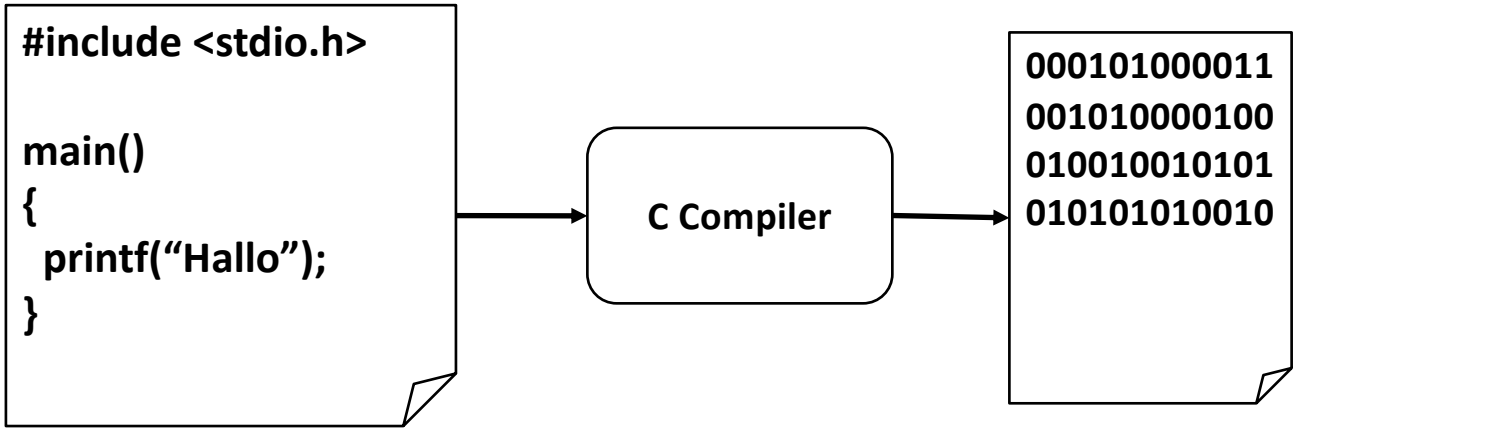
- Komputer bekerja seperti switching dan hanya mengenali 0 dan 1
- Manusia tidak (paham) berbicara dengan bahasa 0 dan 1
- Perlu bahasa pemrograman yang dapat menjadi perantara percakapan antara komputer dan manusia
- Bahasa pemrograman diubah ke dalam bahasa yang dipahami oleh komputer dengan menggunakan interpreter atau kompiler

Compiler or Interpreter?

1. Compiler:
   Mengkompilasi source code menjadi bentuk file yang bisa dieksekusi

2. Interpreter:
   Mengkompilasi dan menjalankan source code secara langsung

C Language (Compiler)



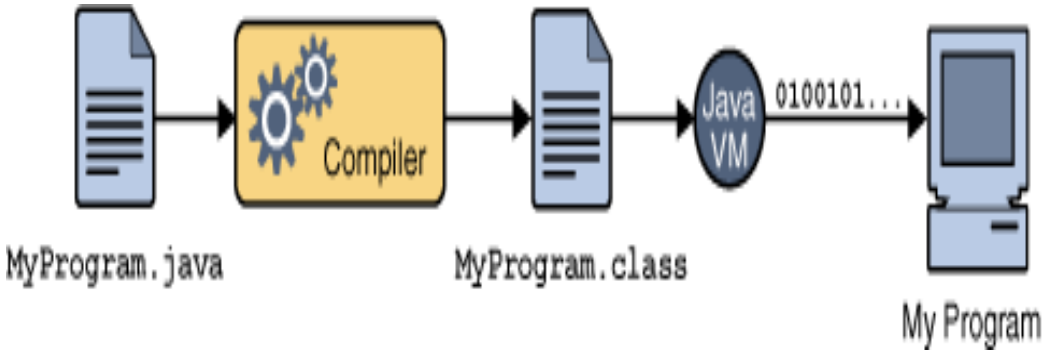
C Program

Machine language program
(executable ".exe" file)

Java Language (Compiler + Interpreter)

javac
(Java Compiler)

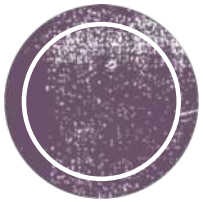
java
(Java Interpreter)



Paradigma Pemrograman

Sudut pandang dan style pemrograman berhubungan dengan bagaimana sebuah masalah diformulasikan dalam bahasa pemrograman

1. Functional Programming: Urutan fungsi secara sekuensial (Scheme, Lisp)
2. Procedural Programming: Pemecahan masalah berdasarkan prosedural kerja yg terkumpul dalam unit pemrograman bernama fungsi (C, Pascal)
3. Object-Oriented Programming: Koleksi object yang saling berinteraksi . Class adalah unit pemrograman (Java, C#, C++)



Konsep Dasar Pemrograman Berorientasi Objek

Class , Object, Method, Attribute

Objek?

Berorientasi Objek?

Attribute:

Topi, Baju, Jaket,
Tas Punggung,
Tangan, Kaki, Mata

Behavior:

Cara Jalan ke Depan
Cara Jalan Mundur
Cara Belok ke Kiri
Cara Memanjat

Berorientasi Objek?

Attribute (State):
- Ban, Stir, Pedal Rem, Pedal Gas,
- Warna, Tahun Produksi

Behavior:
- Cara Menghidupkan Mesin
- Cara Manjalankan Mobil
- Cara Memundurkan Mobil

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

Pengertian OOP

- Teknik atau cara untuk membuat program objek yang saling berinteraksi satu sama lainnya
- Objek yaitu suatu unit terkecil dari program yang mengandung data dan fungsi yang bekerja atas objek tersebut.
- Subtansi materi OOP:
  - Konsep/ilmu tentang OOP
  - Bahasa pemrograman objek: Java
  - Teknis pengoperasian tools untuk membuat program: J2SDK, IDE for Java, dll.

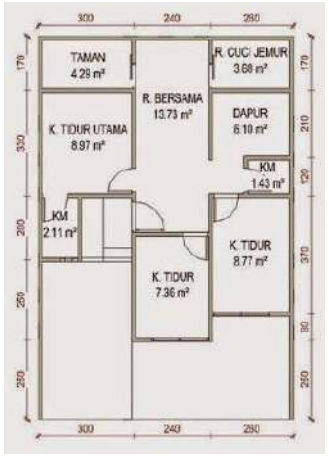
PERANGKAT LUNAK : BERORIENTASI OBJEK

- Perangkat lunak yang dibangun dari kelas-kelas dan objek-objek yang saling berinteraksi satu sama lainnya.
  - Kelas

    Deskripsi statis dari sesuatu.
  - Objek

    Sesuatu yang diciptakan (instansiasi) dari kelas.

Kelas vs Objek

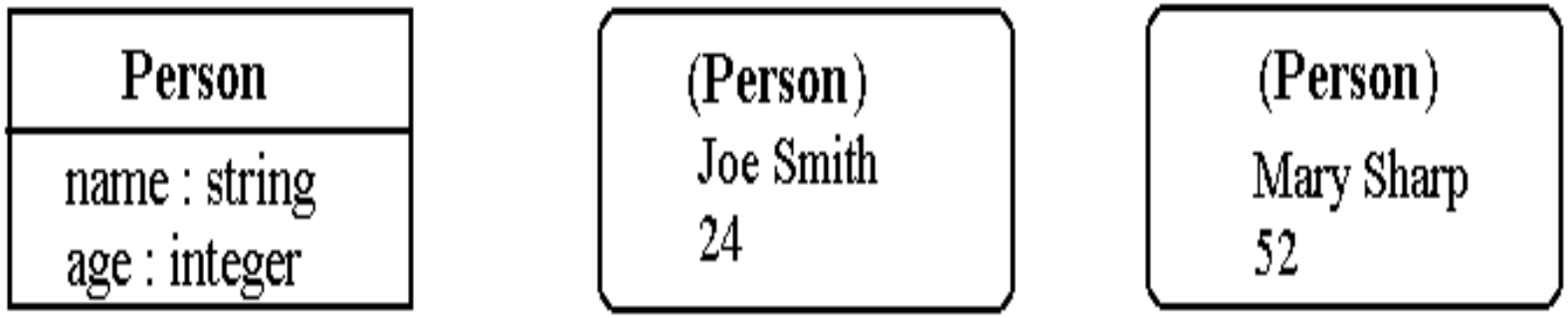


Perbedaan Class dan Object

- 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

Perbedaan Class dan Object

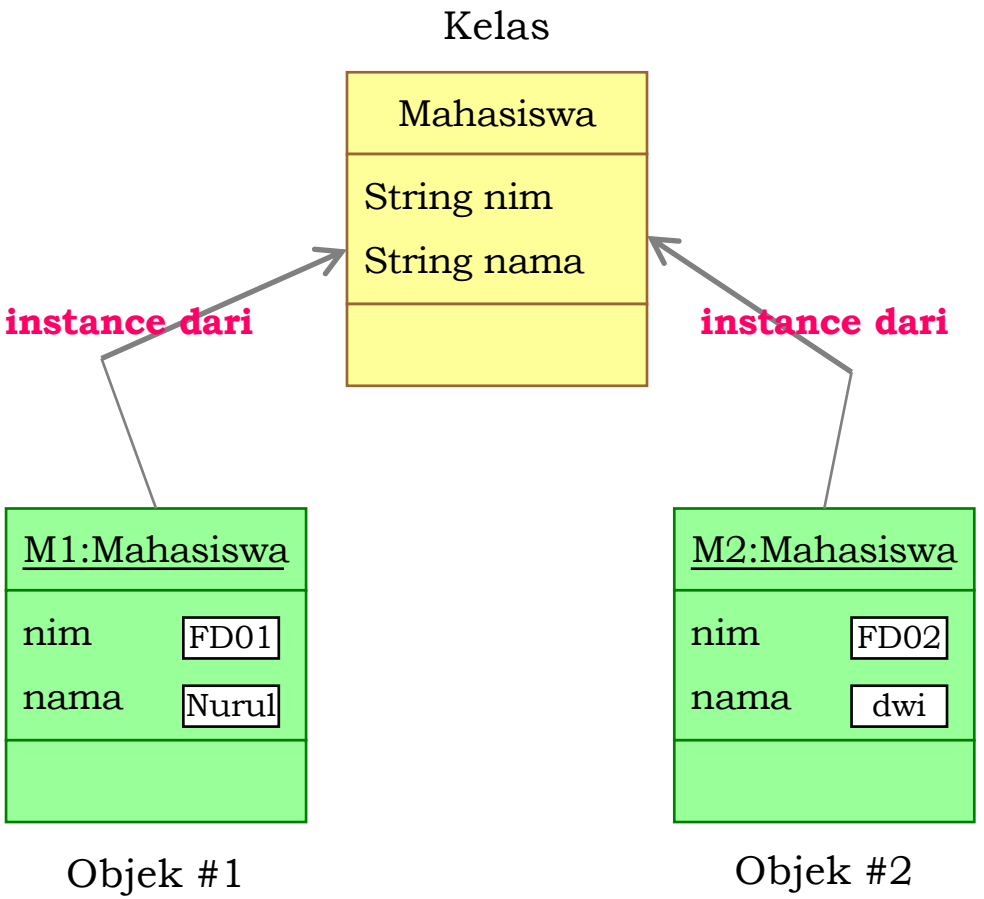
- Class seperti cetakan kue, dimana kue yg dihasilkan dari cetakan kue itu adalah object
- Warna kue bisa bermacam-macam meskipun berasal dari cetakan yang sama (object memiliki sifat independen)



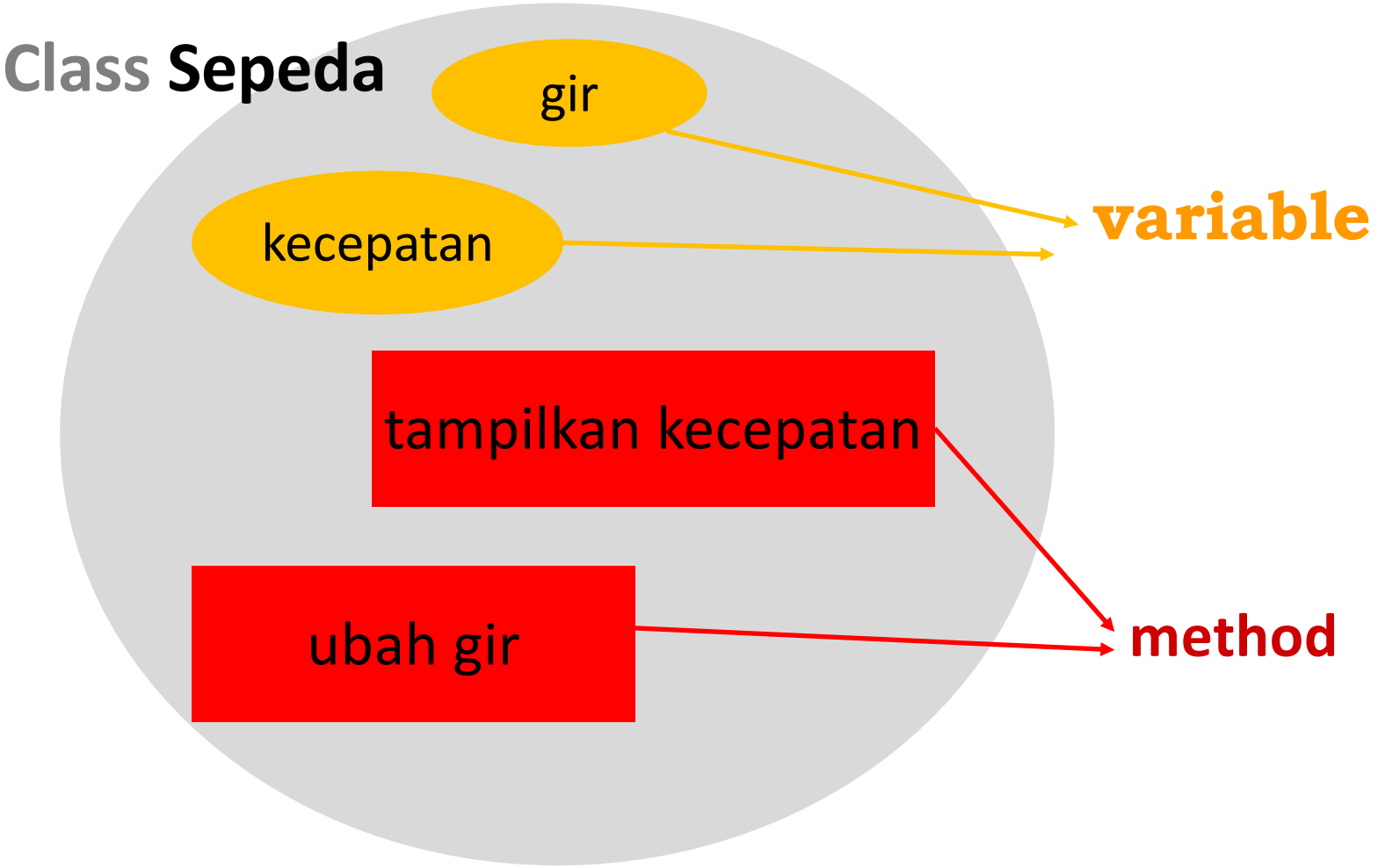
Class with Attributes

Objects with Values

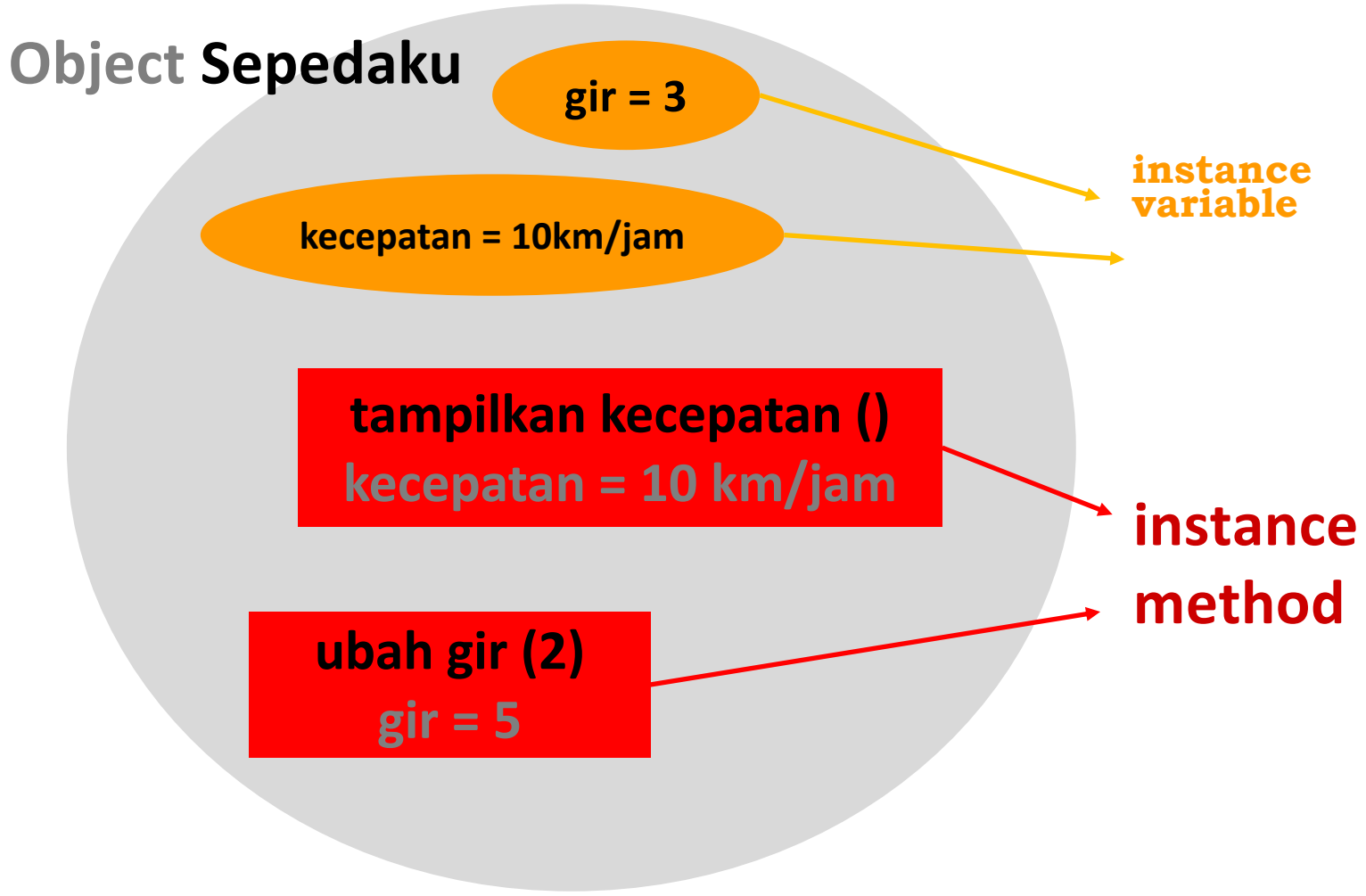
Kelas dan Objek (lanjutan)



Class = Method + Variable



Object = Method + Variable yg Memiliki Nilai



Contoh Kelas dan Objek ?

Contoh Method (Behaviour) dan variabel (atributnya) ?

Property Kelas / Objek

- Atribut

  Data yang mendeskripsikan kelas/objek.

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

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

- Layanan / Metode (Method)

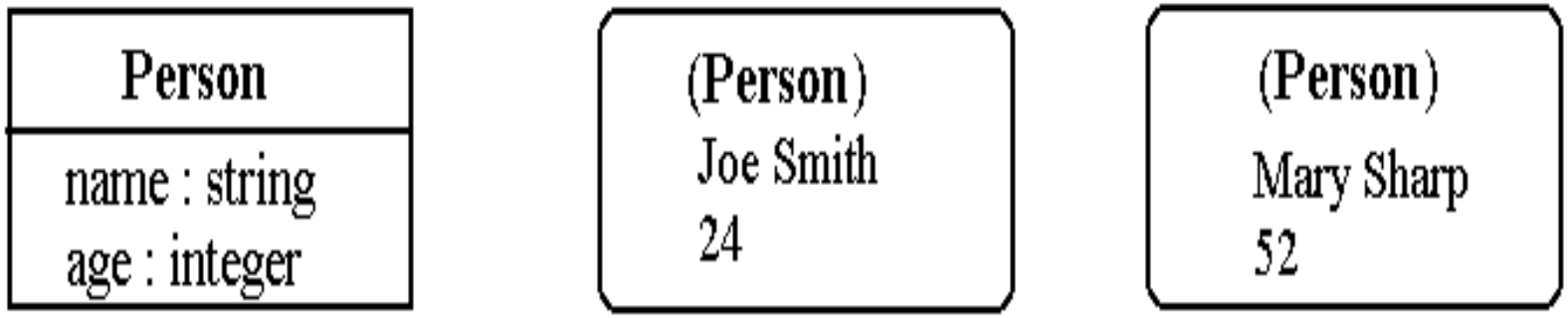
  Prosedur atau **fungsi** yang mencirikan sifat atau kelakuan dari kelas/objek.

  Pada umumnya didefinisikan dengan *access modifer* **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()

Attribute

- Variable yang mengitari class, dengan nilai datanya bisa ditentukan di object
- Variable digunakan untuk menyimpan nilai yang nantinya akan digunakan pada program
- Variable memiliki jenis (tipe), nama dan nilai
- Name, age, dan weight adalah atribute (variabel) dari class Person



Class with Attributes

Objects with Values

Deklarasi Kelas

```
[public] class NamaKelas {
  // Deklarasi atribut
  ...

  // Definisi layanan/method
  <constructor>
  <accessor>
  <mutator>
  <fungsi lain>

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

METHOD : CONSTRUCTOR

- Fungsi yang akan dijalankan/dieksekusi secara otomatis begitu sebuah objek diciptakan.
- 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[]) { ... }
```

  - **public**: fungsi dapat diakses dari dalam maupun luar kelas
  - **static**: sama untuk semua instan dari kelas
  - **void**: fungsi tidak memberikan *return value*
  - **arg[]**: array String yang menjadi argumen fungsi (diambil dari lingkungan eksekusi)
- Isi fungsi main() pada umumnya adalah:
  - Penciptaan (instansiasi) objek
  - Manipulasi objek: penyalinan (copy), penggunaan

Penciptaan Objek

- Deklarasi reference atau variabel:

  Mahasiswa m;

- Alokasi objek ke variabel:

  m = new Mahasiswa();

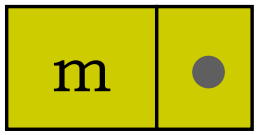
Atau

- Deklarasi dan sekaligus alokasi objek:

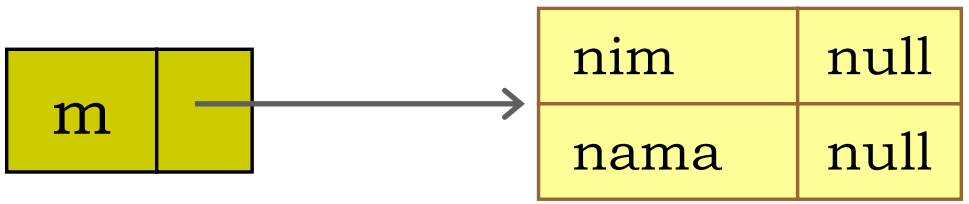
  Mahasiswa m = new Mahasiswa();

Penciptaan Objek (lanjutan)

- Mahasiswa m; // deklarasi variabel



- m = new Mahasiswa(); // alokasi objek



PENGGUNAAN OBJEK

- Dilakukan dengan cara mengirim pesan (*message passing*) ke objek.
- Implementasinya dalam bentuk pemanggilan method yang dipunyai objek.

Contoh:

```
s.setPanjang(17);
s.setLebar(8);
System.Out.Println("Luas = " + s.Luas());
```

nama objek

nama method

Membuat Class, Object dan Memanggil Atribut

Mobil.java

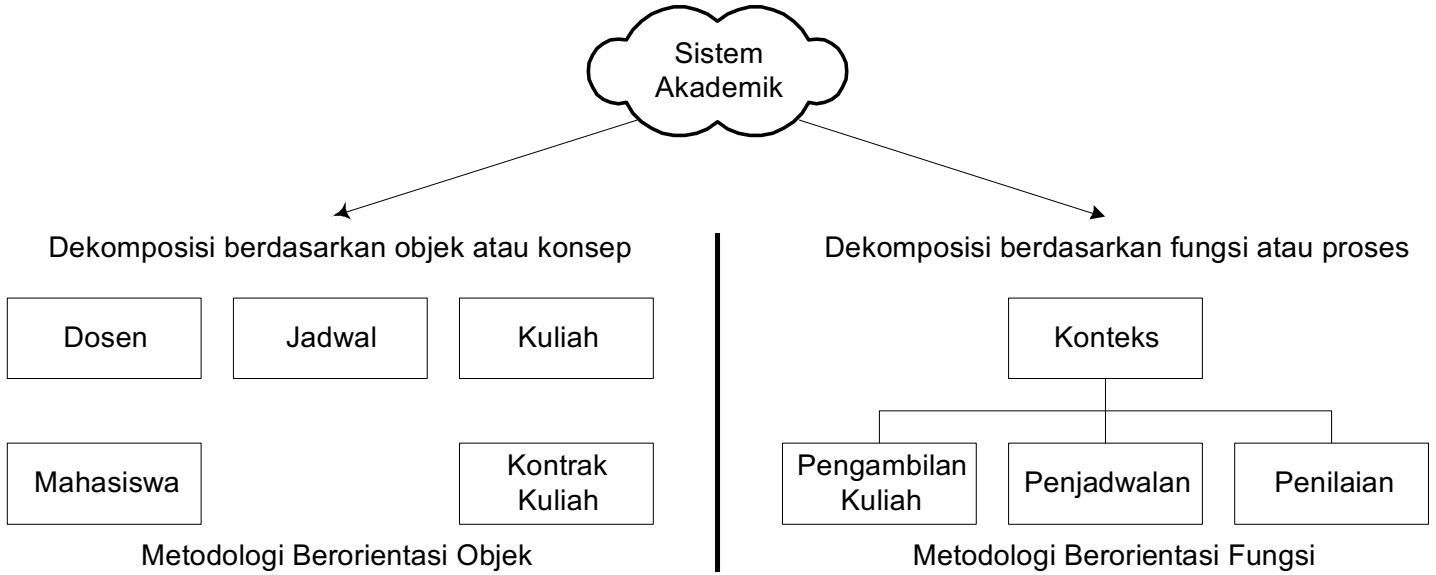
```
public  class  Mobil {
  String  warna;
  int  tahunProduksi;
}
```

MobilBeraksi.java

```
public  class  MobilBeraksi{
    public  static  void  main(String[] args){
          // Membuat object
         Mobil  mobilku = new Mobil();

          /* memanggil atribut  dan memberi nilai */
         mobilku.warna = "Hitam";
         mobilku.tahunProduksi = 2006;
         System.out.println("Warna: " + mobilku.warna);
         System.out.println("Tahun: " + mobilku.tahunProduksi);
    }
}
```

OOP VS PROCEDURAL



PENGENALAN BAHASA PEMROGRAMAN JAVA

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

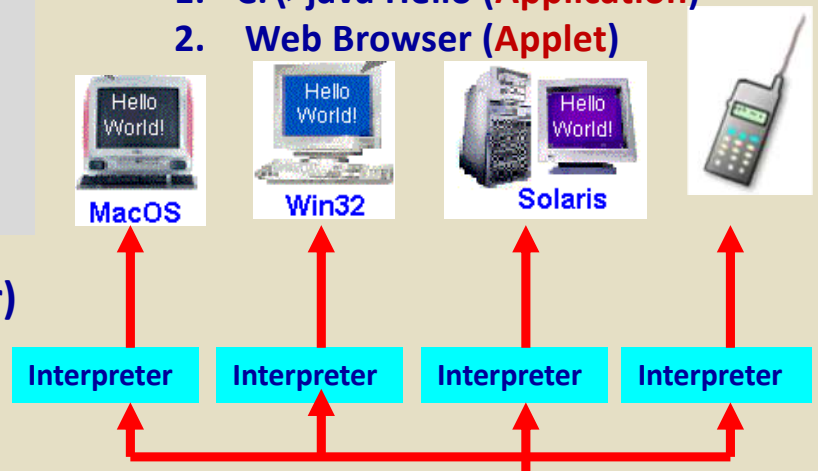
javac (java compiler)

Kompiler memproduksi
Bytecode (Class)

Hello.class

Jalankan dengan:

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



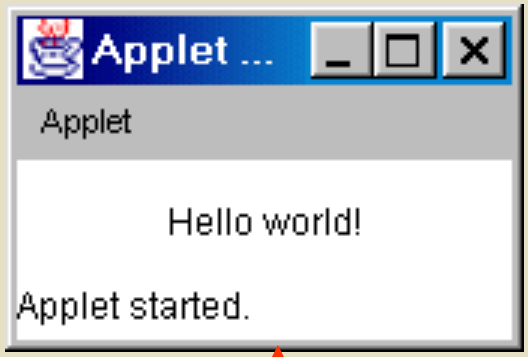
Web Server

Write Once Run Everywhere !

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:\appletviewer Hello.html
```

```
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>
```

Applet on a Web Page

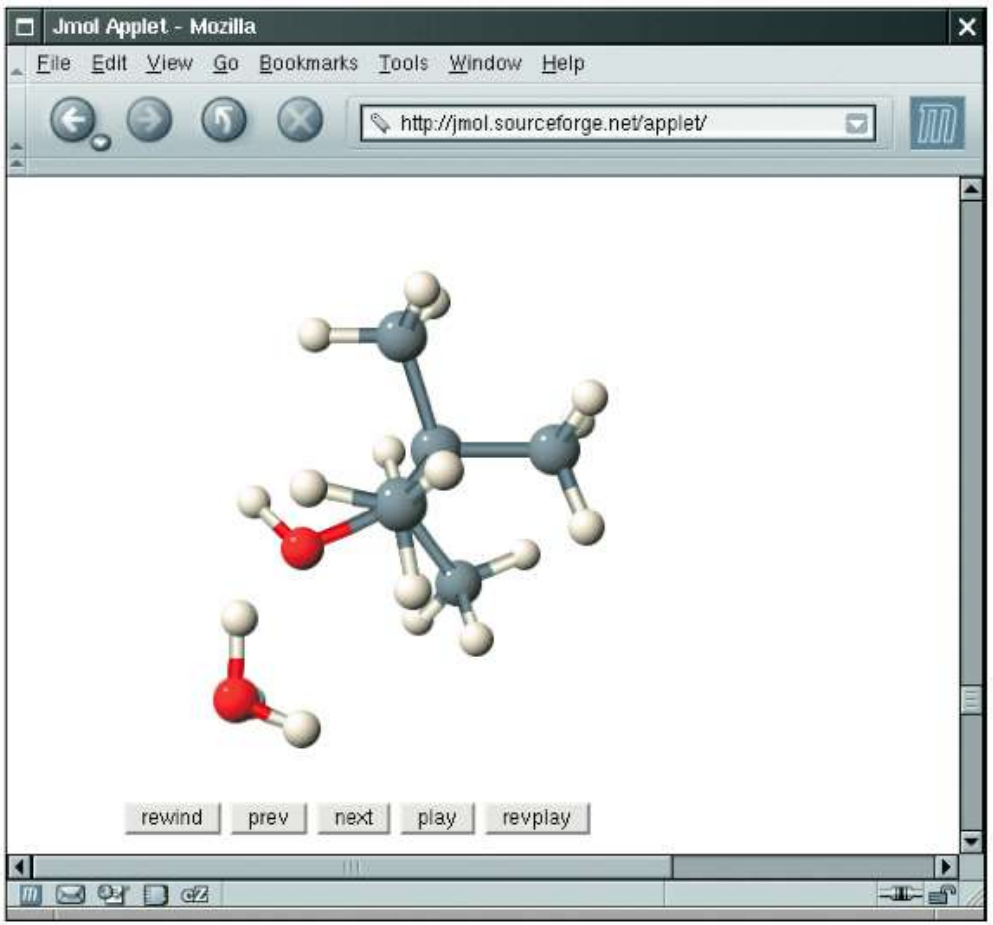


Figure 6 An Applet for Visualizing Molecules Running in a Browser (*http://jmol.sourceforge.net/applet/*)

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”↵

PEMBUATAN PROGRAM BERORIENTASI OBJEK DENGAN JAVA

Review : Property Kelas / Objek

- Atribut

  Data yang mendeskripsikan kelas/objek.

  Pada umumnya didefinisikan dengan *access modifer* **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 modifer* **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
}
```

```
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  }
```

```
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      }
```

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);
}
```

```
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  }
```

```
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      }
```

- 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;
}
```

METHOD : MUTATOR

```
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  }
```

```
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      }
```

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

```
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  }
```

```
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      }
```

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

Mobil2.java

```
public class Mobil2{
   String warna;
   int tahunProduksi;

   void  printMobil(){
            System.out.println("Warna: " + warna);
            System.out.println("Tahun: " + tahunProduksi);
   }
}
```

Mobil2Beraksi.java

```
public class Mobil2Beraksi{
     public static void main(String[] args){
       Mobil2 mobilku = new Mobil2();

        mobilku.warna = "Hitam";
       mobilku.tahunProduksi = 2006;
       mobilku.printMobil();
      }
}
```

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;
}
```

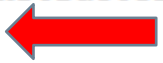
Contoh : Kegunaan Kelas

```
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 Construktor

```
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 …

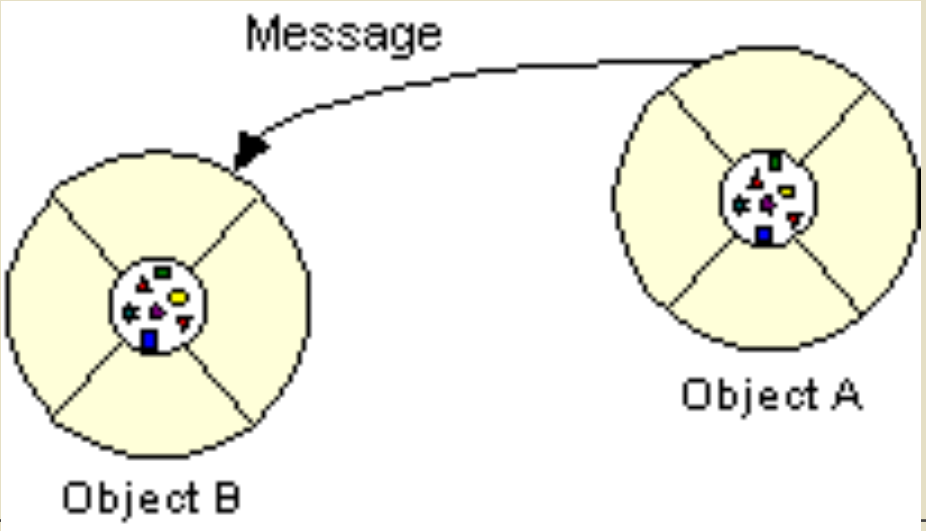
Kring, kring, kring … panggilan dilakukan

Dung, dung … sms berhasil terkirim

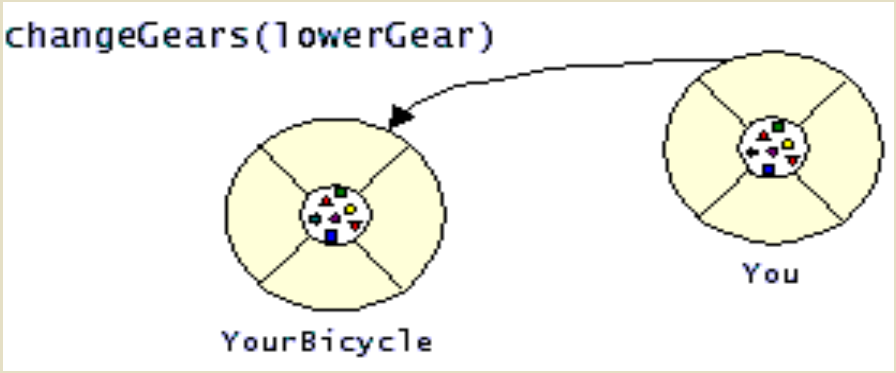
Handphone mati …

Konsep : Parameter

- Sepeda akan berguna apabila ada object lain yang berinterasi 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 :
  - Constractor default
  - Constractor dengan parameter sisi
  - Constractor 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 constractor parameter sisi
  - Pembuatan objek persegiPanjang1 dengan constractor parameter default
  - Pembuatan objek persegiPanjang2 dengan constractor 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.

Latihan: Buatlah Class BangunRuang

- Buat Class bernama BangunRuang, yang berisi method :
  - Constractor default
  - Constractor dengan parameter sisi
  - Constractor dengan parameter jari-jari
  - Constractor dengan parameter panjang, lebar, dan tinggi
  - Assesor dan Mutator untuk variable sisi, panjang, lebar, tinggi, jari-jari
  - Method hitungLuasKubus, hitungLuasBalok, hitungLuasBola, hitungVolumeKubus, hitungVolumeBalok, dan hitungVolumeBola.
- Buat Class bernama BangunRuangBeraksi yang memiliki method main, dengan di dalamnya terdapat :
  - Pembuatan objek kubus dengan constractor parameter sisi
  - Pembuatan objek balok dengan constractor parameter default
  - Pembuatan objek balok2 dengan constractor parameter panjang dan lebar
  - Pembuatan objek bola dengan constractor parameter jari-jari
  - Perhitungan volume dan luas dari masing-masing objek. Dan outputkan hasilnya dengan detail, misalnya : “volume balok dengan panjang 10, lebar 5, dan tinggi 7 yaitu = 350”, dan begitu juga untuk hasil perhitungan volume dan luas untuk objek yang lain.

TERIMA KASIH