

# Pemrograman Berorientasi Objek

## ENCAPSULATION

Oleh :  
SRI HERAWATI, S.KOM

D3 Manajemen Informatika  
Fakultas Teknik  
Universitas Trunojoyo  
2009

## Sub Topik

- Class
- Object
- Method
- Package

## Konsep Pemrograman Berorientasi Objek

- Pemrograman berorientasi objek merupakan teknik membuat suatu program berdasarkan objek.
- Contoh objek : manusia, Meja, kursi, jam dinding

## Class dan Objek

- Objek merupakan suatu bentuk atau model yang tergantung dari cara pandang objek dimaksud.
- Meja, kursi dan jam dinding berada dalam objek ruangan. Ruang A, ruang B, dan Ruang C berada dalam objek rumah

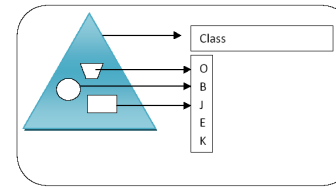
## Class dan Objek

- Setiap objek memiliki atribut (state) dan method (behaviour)
- Atribut merupakan segala sesuatu yang berhubungan dengan karakteristik objek seperti : tipe, warna
- Method adalah fungsi atau segala sesuatu yang dapat dilakukan objek, seperti melindungi dari panas, hujan.

## Class dan objek

- Ruangan dan rumah ini yang dinamakan *class*
- *Class* adalah prototipe yang mendefinisikan variabel-variabel dan method-method secara umum.
- Objek merupakan instansiasi dari class.

## Class dan Objek



## Class dan Objek

- Objek yang mengandung variabel dan method dapat ditentukan hak aksesnya kemudian dibungkus dalam bagian yang terlindungi yang dinamakan **encapsulation**

## Class dan Objek

- Deklarasi objek  
Objek diciptakan melalui **new**.
- Sintak:  

```
class namaclass {  
    // definisi class  
}
```

## Class dan Objek

```
class mobil{  
    String warna;  
    int tahunproduksi;  
}  
public class coba {  
  
    public static void  
    main(String[]args)  
    {  
        mobil mobilku= new  
        mobil();  
    }  
}
```

// mengisi variabel instan  
mobilku.warna="Hitam";  
mobilku.tahunproduksi=2006;  
  
//menampilkan isi variabel  
instan  
System.out.println("Warna : " +  
mobilku.warna);  
System.out.println("Tahun  
Produksi : " +  
mobilku.tahunproduksi);  
}

## Method

- Method merupakan blok yang berisi kumpulan perintah, memiliki nama method, memiliki nilai balik dan dapat diatur hak aksesnya.

## Kata kunci this

- this digunakan pada pembuatan class yang menyatakan objek sekarang.

## Contoh :

```
Class Lingkup{  
    //variabel instan  
    String warna = "Merah";  
  
    //metode  
    void infoLingkup(){  
        String warna = "Biru";  
        System.out.println("warna pada  
metode : "+ warna);  
        System.out.println("Warna pada  
kelas : "+this.warna);  
    }  
}
```

```
public class lingkupvariabel {  
  
    public static void main(String[]  
args){  
        Lingkup varx = new Lingkup();  
        varx.infoLingkup();  
    }  
}
```

## Penentu akses : public dan private

- Public berarti pengaksesan suatu variabel instan atau metode dapat dilakukan dari luar kelas
- Private berarti pengaksesan suatu variabel instan atau metode hanya dapat dilakukan di dlm kelas, tidak bisa dari luar kelas.

Modifier	Class	Package	Subclass	Lainnya
public	✓	✓	✓	✓
protected	✓	✓	✓	
default modifier	✓	✓		
private	✓			

## Paket (Package)

- Paket merupakan pengelompokan sejumlah kelas yang terkait ke dlm sebuah unit.

## Java class package

- Java memiliki beragam paket yang didalamnya terdapat kelas-kelas dengan beragam fungsi
- Java lang, Java AWT, Java swing, Java Util, java Math adalah diantara contoh paket yang jumlahnya sangat banyak karena lengkapnya library java

## Java Class Packages

```
public class Math
{
    public void Perkalian(long A, long B)
    {
        long C=A*B;
        System.out.println("Hasil Perkalian="+ C);
    }

    public void Pembagian(long A, long B)
    {
        long C=A/B;
        System.out.println("Hasil Pembagian="+C);
    }

    public void Penjumlahan(long A, long B)
    {
        long C=A+B;
        System.out.println("Hasil Penjumlahan="+C);
    }

    public void Pengurangan(long A, long B)
    {
        long C=A-B;
        System.out.println("Hasil Pengurangan="+C);
    }
}
```

## Java Class Packages

```
import java.util.Scanner;

public class AksesMath
{
    public static void main (String args [])
    {
        int X;
        int Y;
        int Pilih;

        Scanner input= new Scanner(System.in);

        System.out.println("Masukkan Nilai Pertama =");
        X= input.nextInt();
        System.out.println("Masukkan Nilai Kedua =");
        Y= input.nextInt();

        System.out.println("Menu Pilihan Test Math :");
        System.out.println("1. Penjumlahan");
        System.out.println("2. Pengurangan");
        System.out.println("3. Perkalian");
        System.out.println("4. Pembagian");
    }
}
```

## Java Class Packages

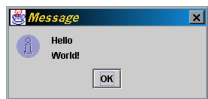
```
Pilih=input.nextInt();
Math FungsMath = new Math();

switch (Pilih)
{
    case 1:
    {
        FungsMath.Penjumlahan(X,Y);
        break;
    }
    case 2:
    {
        FungsMath.Pengurangan(X,Y);
        break;
    }
    case 3:
    {
        FungsMath.Perkalian(X,Y);
        break;
    }
    case 4:
    {
        FungsMath.Pembagian(X,Y);
        break;
    }
    default : System.out.println("Pilihan Anda Ngawut.");
}
```

## Java Class Packages

```
import javax.swing.JOptionPane;

public class Welcome2
{
    public static void main (String args [])
    {
        JOptionPane.showMessageDialog (null,
        "Hello\nWorld!");
        System.exit (0);
    }
}
```



## Java Class Packages

- Dari kedua contoh *Package* diatas, Paket java.util diakses kedalam kelas *java.Scanner* yang memiliki kelas dengan metode pembacaan input sedangkan untuk contoh kedua adalah penggunaan Package *java.Swing.JOptionPane* yang merupakan kelas window untuk menampilkan ataupun menerima pesan
- Class static JOptionPane memiliki metode *showMessageDialog ()* yang berfungsi menampilkan pesan window terdapat metode yang lain dari kelas ini yaitu *showInputDialog()*.

## Mendeklarasikan paket

```
Contoh 1 :
package kendaraan;
public class Mobil{
    public void info(){
        System.out.println("Mobil");
    }
}

Contoh 2 :
package kendaraan;
public class Sepeda{
    public void info(){
        System.out.println("Sepeda");
    }
}
```

## Mendeklarasikan paket

- Dari kedua contoh diatas menyatakan bahwa kelas berikutnya(mobil maupun sepeda) disatukan dalam sebuah paket kendaraan.
- Sebuah berkas hanya boleh mengandung sebuah pernyataan **package**.

## Menggunakan import

- **import** merupakan mekanisme dalam program utk mengakses kelas yang terdapat pada paket.
- Bentuk :  
**import** nama\_paket.nama\_berkas
- Apabila menemui pengimporan paket spt dibawah ini  
**import** nama\_paket.\*;  
tanda \* (*wildcard*) menyatakan semua kelas pada paket nama\_paket. Contoh :  
**import** kendaraan.\*;  
berarti semua kelas yang berada pada paket kendaraan.

## Menggunakan import

```
import kendaraan.Mobil;
import kendaraan.Sepeda;

public class TesPaket{
    public static void main(String[] args){
        Mobil obj1 = new Mobil();
        obj1.info();

        Sepeda obj2 = new Sepeda();
        obj2.info();
    }
}
```