

Dasar-Dasar Pemrograman 2

Lab 04

Intro to Object-Oriented Programming (OOP)



FAKULTAS
ILMU
KOMPUTER

Object Oriented Programming

Object Oriented Programming adalah suatu paradigma *programming* yang mengikuti konsep dari *object*. Dalam paradigma ini kita memodelkan objek di dunia nyata sebagai sebuah *class* di dalam program. Objek tersebut memiliki *variable*, *constructor*, dan *method*.

Sebagai contoh, kita akan membuat suatu *class* yang memodelkan objek **Mobil** di dunia nyata. Mobil dapat dilihat sebagai sebuah objek yang memiliki atribut **merek**, **model**, **warna**, dan **kelajuan**. Kelajuan mobil dapat dipercepat (**akselerasi**) maupun diperlambat (**deselerasi**). Berikut implementasi *class* yang dibuat.

```
public class Mobil{  
    private String merek;  
    private String model;  
    private String warna;  
    private double kelajuan;  
  
    //class' Constructor  
    public Mobil(String merek, String model, String warna){  
        this.merek = merek;  
        this.model = model;  
        this.warna = warna;  
        this.kelajuan = 0;  
    }  
}
```

```

//class' method
public void akselerasi(double perubahan){
    this.kelajuan += perubahan;
}

//class' method
public void deselerasi(double perubahan){
    this.kelajuan -= perubahan;
}
}

```

Catatan:

Penggunaan **this** pada kode di atas (Java) ekuivalen dengan penggunaan **self** pada kode Python.

Constructor

Constructor adalah *method* untuk membuat instansiasi objek dari *class*. Constructor adalah *method* khusus yang akan dieksekusi pada saat pembuatan objek (*instance*). Pada umumnya, *method* ini berisi inisialisasi variabel untuk objek.

```

public class Mobil{

    //...inisialisasi variabel

    //class' Constructor
    public Mobil(String merek, String model, String warna){
        this.merek = merek;
        this.model = model;
        this.warna = warna;
        this.kelajuan = 0;
    }
}

```

Constructor di atas akan dijalankan jika objek mobil dibuat

```
Mobil mobil = new Mobil("Daihatsu", "Xenia", "Putih");
```

Setter & Getter

Setter adalah *method* yang dipakai suatu objek untuk mengubah *value* dari variabel yang dimiliki sebuah objek, sedangkan **getter** adalah *method* yang dipakai untuk mengambil *value* dari sebuah variabel yang dimiliki objek.

Setter dan *getter* digunakan bersamaan terhadap variabel yang disembunyikan oleh objek, biasanya dengan memakai tipe **private** pada atribut tersebut (Contohnya semua variabel pada kelas **Mobil** di atas). Hal ini dilakukan untuk menghindari akses secara langsung dari kelas lain yang bisa menyebabkan kebocoran dan perubahan terhadap data yang disimpan oleh suatu objek

Contoh pengaplikasian *setter* dan *getter*:

```
//setter method
public void setKelajuan(int laju){
    this.kelajuan = laju;
}

//getter method
public double getKelajuan(){
    return this.kelajuan;
}
```

Contoh penggunaan *setter* dan *getter*:

```
Mobil mobil = new Mobil("Daihatsu", "Xenia", "Putih");
System.out.println(mobil.getKelajuan());
mobil.setKelajuan(40);
System.out.println(mobil.getKelajuan());
```

Output yang dihasilkan:

```
0
40
```

toString()

toString adalah *method* yang digunakan untuk mengatur representasi suatu objek ke dalam String. Jika dalam suatu *class* tidak terdapat method **toString()**, maka saat melakukan print, yang muncul di output program adalah nilai kode hash dari objek tersebut

```
Mobil mobil = new Mobil("Daihatsu", "Xenia", "Putih");
System.out.println(mobil);
```

Output:

Mobil@d716361

Dengan *method* **toString()** kita dapat mengukur representasi String dari objek tersebut. Contohnya dengan menambahkan *method* **toString()** dalam *class* **Mobil**, seperti di bawah ini:

```
class Mobil{

    //.....

    //toString method
    public String toString(){
        return "Mobil " + merek + " " + model + " berwarna " + warna;
    }
}
```

Output ketika kita melakukan print:

Mobil Daihatsu Xenia berwarna Putih

Soal Lab 04

Toko Baru Dek Depe



Belakangan ini Dek Depe sangat senang karena akhirnya ia dapat membuka toko kelontong yang selama ini diidam-idamkannya. Ia berniat untuk memberikan potongan harga Rp 1.000,00 bagi produk - produk berlabel bintang. Namun ternyata Dek Depe mengalami musibah, mesin kasir bekas yang dibelinya ternyata tidak bekerja dengan baik. Hal ini menyebabkan lamanya antrian saat pembayaran di tokonya sehingga penjualan menjadi menurun. Sebagai teman yang baik, kamu berniat membantu Dek Depe untuk mengatasi masalah ini dengan membuat sebuah program java yang dapat berfungsi layaknya mesin kasir.

Soal Lab:

Terdapat dua buah class yang akan digunakan untuk membantu pembuatan program mesin kasir ini. Tugas kalian adalah mengimplementasikan class Keranjang dan class Barang dengan ketentuan berikut:

1. Class Barang

Attribut : nama barang dan harga barang

2. Class Keranjang

Attribut : List dari barang - barang yang akan dibeli.

Method :

- banyakBarangDiskon()

Method ini akan mengembalikan banyaknya barang - barang yang didiskon.

- totalHarga()

Method ini akan mengembalikan total harga dari barang - barang yang ada di dalam keranjang. (Jangan lupa beri potongan harga apabila barang memiliki label berbintang)

Input:

Pertama - tama anda akan diminta untuk memasukkan banyaknya barang yang akan dibayar

Kemudian anda akan memasukkan barang ke dalam keranjang sebanyak N kali sesuai input banyaknya barang.

Formatnya adalah sebagai berikut:

Masukkan jumlah produk: N

[NAMA_BARANG] [HARGA_BARANG]

Output:

Anda diminta untuk mencetak banyaknya barang yang diskon beserta total harga seluruh produk yang dibeli

Contoh input dan output

No.	Input	Output
1.	Masukkan jumlah produk: 3 SABUN 5000 SAMPO 20000 SUSU 6000	Banyaknya barang yang didiskon adalah 0 dengan total harga 31000
2.	Masukkan jumlah produk: 4 *GULA 10000	Banyaknya barang yang didiskon adalah 2 dengan total harga 49000

	GARAM 8000 *TEPUNG 25000 LADA 8000	
3.	Masukkan jumlah produk: 4 *SUSU 6000 *MARGARIN 10000 *GULA 10000 *LADA 8000	Banyaknya barang yang didiskon adalah 4 dengan total harga 30000

Catatan:

1. Diasumsikan nama produk hanya terdiri dari sebuah kata dan harga produk pasti lebih dari Rp 1.000,00
2. Untuk mengecek apakah barang tersebut diskon, dapat dilakukan dengan mengecek apakah nama barang tersebut diawali dengan tanda bintang (' * ').
Hint: gunakan method substring.

Komponen Penilaian

20% Implementasi class Barang
40% Implementasi class Keranjang
30% Implementasi class Kasir
10% Dokumentasi dan kerapian kode