MODUL 2 PENGENALAN *CLASS, OBJECT* & ENKAPSULASI

Class merupakan cetak biru (blue print) dari objek atau dengan kata lain sebuah Class menggambarkan ciri-ciri objek secara umum. Sebagai contoh Suzuki Smash, Yamaha VegaR, Honda SupraFit, dan Kawasaki KazeR merupakan objek dari Class sepeda motor. Suzuki Smash dan objek lainnya juga memiliki kesamaan atribut (merk, tipe, berat, kapasitas bensin, tipe mesin, warna, harga) dan method untuk mengakses data pada atributnya (misal fungsi untuk menginputkan data merk, tipe, berat, dsb serta fungsi untuk mencetak data merk, tipe, berat, dsb).

Contoh :

```
1  public class SepedaMotor {
2     private String merk;
3     private long harga;
4
5     public void setMerk(String merkMotor) {
6        merk = merkMotor;
7     }
8
9     public String getMerk(){
10       return merk;
11    }
12
13    public long Harga(long hargaMotor) {
14       return harga = hargaMotor;
15    }
16 }
```

A. Object

Objek (*Object*) merupakan segala sesuatu yang ada di dunia ini, yaitu manusia, hewan, tumbuhan, rumah, kendaraan, dan lain sebagainya. Contoh-contoh objek yang telah disebutkan diatas merupakan contoh objek nyata pada kehidupan kita.

Pada pemrograman berorientasi objek, kita akan belajar bagaimana membawa konsep objek dalam kehidupan nyata menjadi objek dalam dunia pemrograman. Setiap objek dalam dunia nyata pasti memiliki 2 elemen penyusunnya, yaitu keadaan (*state*) dan perilaku/sifat (*behaviour*). Sebagai contoh, sepeda memiliki keadaan yaitu warna, merk, jumlah roda, ukuran roda. Dan perilaku/sifat sepeda adalah berjalan, berhenti, belok, menambah kecepatan, mengerem.

Pada saat objek diterjemahkan ke dalam konsep PBO, maka elemen penyusunnya juga terdiri atas 2 bagian, yaitu :

Atribut, merupakan ciri-ciri yang melekat pada suatu objek (*state*).

Method, merupakan fungsi-fungsi yang digunakan untuk memanipulasi nilai-nilai pada atribut atau untuk melakukan hal-hal yang dapat dilakukan suatu objek (*behaviour*).

Objek dalam konsep PBO memiliki keadaan dan perilaku yang sama seperti halnya objek di dunia nyata, karena objek dalam konsep PBO merupakan representasi objek dari dunia nyata.

Objek dalam PBO merepresentasikan keadaan melalui variabel-variabel (Atribut), sedangkan perilakunya direpresentasikan dengan method (yang merupakan suatu fungsi yang berhubungan dengan perilaku objek tersebut maupun berhubungan dengan atribut dari objek tersebut). Objek yang memiliki kesamaan atribut dan method dapat dikelompokkan menjadi sebuah Class. Dan objek-objek yang dibuat dari suatu class itulah yang disebut dengan Instant of class.

Untuk menginstansi (membuat) objek dari class, gunakan operator *new*.

Sintaks membuat objek dari suatu class :

```
namaClass namaObjek = new namaClass()
```

Class utama dari program :

```
1 public class Main {
2     public static void main (String []args){
3         SepedaMotor motor = new SepedaMotor();
4         motor.setMerk("Suzuki");
5         System.out.println("Motor ini bermerk " +motor.getMerk());
6         System.out.println("Motor ini berharga " +motor.Harga(11000000));
7     }
8 }
```

Perhatikan class Main diatas !

Nama objek dari class SepedaMotor adalah motor.

Silahkan dicoba untuk melihat hasilnya !

B. Enkapsulasi

Enkapsulasi (*encapsulation*) merupakan cara untuk melindungi property (atribut) / method tertentu dari sebuah kelas agar tidak sembarangan diakses dan dimodifikasi oleh suatu bagian program. Cara untuk melindungi data yaitu dengan menggunakan *access modifiers* (hak akses). Ada 4 hak akses yang tersedia, yaitu default, public, protected, private.

Untuk lebih jelasnya, silahkan lihat kedua table berikut ini :

No	Modifier	Pada class dan interface	Pada method dan variabel
1	Default (tidak ada modifier)	Dapat diakses oleh yang sepaket	Diwarisi oleh subkelas dipaket yang sama, dapat diakses oleh method-method yang sepaket
2	Public	Dapat diakses dimanapun	Diwarisi oleh subkelasnya, dapat diakses dimanapun
3	Protected	Tidak bisa diterapkan	Diwarisi oleh subkelasnya, dapat diakses oleh method-method yang sepaket
4	private	Tidak bisa diterapkan	Tidak dapat dikases dimanapun kecuali oleh method-method yang ada dalam kelas itu sendiri

Aksesabilitas	public	private	protected	default
Dari kelas yang sama	Ya	Ya	Ya	Ya
Dari sembarang kelas dalam paket yang sama	Ya	Tidak	Ya	Ya
Dari sembarang kelas di luar paket	Ya	Tidak	Tidak	Tidak
Dari subkelas dalam paket yang sama	Ya	Tidak	Ya	Ya
Dari subkelas di luar paket	Ya	Tidak	Ya	Tidak

Perhatikan keyword "***this***" di bawah ini (lihat pada class Enkapsulasi). Untuk membedakan variabel alas pada parameter dan variabel alas pada atribut class Enkapsulasi, digunakanlah keyword "*this*". Sehingga untuk menggunakan atribut alas pada class Enkapsulasi digunakan : *this*.alas

Contoh:

Nama file : **Enkapsulasi.java**

```java
public class Enkapsulasi {
    private int alas, tinggi;
    private double luasSegitiga;

    public void setAlas(int alas) {
        this.alas = alas;
    }

    public int getAlas() {
        return alas;
    }

    public void setTinggi(int tinggi) {
        this.tinggi = tinggi;
    }

    public int getTinggi() {
        return tinggi;
    }

    public void setLuasSegitiga(int alas, int tinggi) {
        luasSegitiga = 0.5 * (double)(alas * tinggi);
    }

    public double getLuasSegitiga() {
        return luasSegitiga;
    }
}
```

Nama file : **MainEnkapsulasi.java**

```
1  public class MainEnkapsulasi {
2      public static void main (String args[])
3      {   Enkapsulasi ob = new Enkapsulasi();
4          ob.setAlas(5);
5          ob.setTinggi(7);
6          System.out.println("Alas Segitiga : "+ob.getAlas());
7          System.out.println("Tinggi Segitiga : "+ob.getTinggi());
8          ob.setLuasSegitiga(ob.getAlas(), ob.getTinggi());
9          System.out.println("Luas Segitiga : "+ob.getLuasSegitiga());
10     }
11 }
```

Silahkan dicoba untuk melihat hasilnya !

C. Overloading

Overloading adalah diperbolehkannya dalam sebuah class memiliki lebih dari satu nama function/method yang sama tetapi memiliki parameter/argument yang berbeda.

Contoh **overloading**:

```
1  public class Overloading {
2      public void Tampil(){
3          System.out.println("I love Java");
4      }
5      public void Tampil(int i){
6          System.out.println("Method dengan 1 parameter = "+i);
7      }
8      public void Tampil(int i, int j){
9          System.out.println("Method dengan 2 parameter = "+i+" & "+j);
10     }
11     public void Tampil(String str){
12         System.out.println(str);
13     }
14
15     public static void main(String a[]){
16         Overloading objek = new Overloading();
17         objek.Tampil();
18         objek.Tampil(8);
19         objek.Tampil(6,7);
20         objek.Tampil("Hello world");
21     }
22 }
```

Silahkan dicoba untuk melihat hasilnya !

MODUL 2 - LATIHAN

Latihan Soal

1. Seorang penjual alat tulis menjual 10 bolpoint, 10 pensil dan 10 penghapus. 1 biji bolpoint harganya Rp. 2000, 1 biji pensil harganya Rp. 1.000 dan 1 penghapus harganya Rp. 500. Gunakanlah objek untuk menyelesaikan soal dibawah ini!
   a. Buatlah method untuk memasukkan *(setter)* nama, stok, harga satuan, dan harga (stok x harga satuan) alat tulis tersebut! (40 point)
   b. Buatlah method untuk menampilkan *(getter)* nama, stok, harga satuan, dan harga (stok x harga satuan) alat tulis tersebut! (40 point)
   c. Buatlah method Total Harga *(setter getter)* untuk menampilkan uang yang diterima penjual jika semua alat tulis tersebut terjual semuanya! (20 point)

Catatan untuk asisten:

a. Soal dibuat berbeda antara hari senin, selasa, dan rabu. Untuk hari senin bisa memakai soal diatas, untuk hari selasa, dan rabu dapat diubah jumlah, harga dan jenisnya (misalnya mobil Toyota, Kijang, Panther, Karimun) terserah. Boleh lebih dari 3 variable.

b. Output

Nama Alat Tulis : Bolpoint
Stok : 10
Harga Satuan: 2000
Harga Bolpoint : 20000

Nama Alat Tulis : Pensil
Stok : 10
Harga Satuan: 1000
Harga Pensil : 10000

Nama Alat Tulis : Penghapus
Stok : 10
Harga Satuan: 500
Harga Penghapus : 5000

Total Harga: 35000