



# Modul Praktikum **PBO**



## MODUL 2 CLASS, OBJECT & CONSTRUCTOR

### TUJUAN PEMBELAJARAN

- A. Mahasiswa dapat memahami konsep Class, Object, dan Constructor
- B. Mahasiswa diharapkan me-implementasikan Class, Object, dan Constructor

### DASAR TEORI

#### 1. CLASS

**Class** adalah cetak biru atau blueprint dari **object**. Class digunakan hanya untuk membuat kerangka dasar. Yang akan kita pakai nanti adalah hasil cetakan dari class, yakni **object**.

Sebagai analogi, class bisa diibaratkan dengan **laptop** atau **notebook**, yakni gambaran umum tentang sebuah benda. Di dalam pemrograman nanti, contoh class seperti **User**, **Item**, **Siswa**, **Validate**, dll.

Dalam bahasa Java, penulisan **class** diawali dengan keyword **class**, kemudian diikuti dengan nama dari class tersebut. Aturan penulisan nama **class** sama seperti aturan penulisan variabel di Java (lebih tepatnya aturan **identifier**), yakni tidak boleh diawali angka dan tidak boleh mengandung spasi.

Namun kebiasaan programmer Java, nama class ditulis menggunakan **PascalCase** atau **UpperCamelCase**. Yakni setiap kata diawali dengan huruf besar, termasuk kata pertama. Berikut adalah contoh penulisan **class** dalam bahasa Java:

```
1  class Laptop {  
2      // isi class laptop  
3      // isi class laptop  
4  }
```



## a. Properti (Property)

**Property** (atau kadang juga dengan atribut atau field) adalah data yang terdapat dalam sebuah **class**. Melanjutkan analogi tentang laptop, **property** dari laptop bisa berupa merk, warna, harga, jenis processor, ukuran layar, dan lain-lain. Berikut adalah contoh penulisan **class** dengan penambahan **property**:

```
1  class Laptop {
2      String merk;
3      String warna;
4      int harga;
5  }
```

## b. Method

Method adalah tindakan yang bisa dilakukan di dalam class. Jika menggunakan analogi **class Laptop**, maka contoh method adalah: menghidupkan laptop, mematikan laptop, atau mengganti cover laptop.

Method pada dasarnya adalah **function** yang berada di dalam **class**. Seluruh sifat function bisa diterapkan ke dalam method seperti bisa di isi argument/parameter, mengembalikan nilai (dengan keyword **return**), dan lain-lain.

Berikut adalah contoh penulisan **class** dengan penambahan **property**:

```
1  class Laptop {
2      // property
3      String merk;
4      String warna;
5      int harga;
6      //method
7      void hidupkanLaptop() {
8          System.out.println("Laptop hidup");
9      }
10
11     void matikanLaptop() {
12         System.out.println("Laptop mati");
13     }
14 }
```



## 2. OBJECT / OBJEK

**Objek** adalah hasil cetak dari class, atau bisa juga disebut hasil konkrit dari class. Masih menggunakan analogi **class Laptop**, maka object dari **class Laptop** bisa berupa: **laptopRudi**, **laptopLisa**, atau **laptopDuniailkom**.

Sebuah object dari class Laptop akan memiliki seluruh ciri-ciri laptop, termasuk property dan method-nya. Proses mencetak object dari class ini disebut dengan **instansiasi** (atau **instantiation**).

Dalam bahasa Java, proses instansiasi dilakukan dengan menulis nama class, lalu diikuti nama variabel yang akan menampung object tersebut, tanda sama dengan "=", keyword new dan kembali nama class dengan tambahan tanda kurung "()".

Hasil cetakan **class** akan disimpan dalam variabel untuk selanjutnya dipakai selama penulisan kode program. Variabel inilah yang akan berisi **object** dari sebuah class.

### a. Cara membuat Objek

Sebagai contoh, berikut cara membuat object **laptopRudi** dan **laptopLisa** yang dibuat dari **class Laptop**:

*\*buat file java baru bernama Main.java\**

```
1 public class Main {
2     public static void main(String[] args) {
3         // buat object dari class Laptop (instansiasi)
4         Laptop laptopRudi = new Laptop();
5         Laptop laptopLisa = new Laptop();
6     }
7 }
```



## b. Cara mengakses Objek

Cara mengakses object yang dimaksud disini adalah cara untuk mengakses 'isi' dari sebuah **object**, yakni property dan method. Agar lebih mudah dipahami, berikut revisi contoh **class Laptop** sebelumnya:

```
1 public class Main {
2     public static void main(String[] args) throws Exception{
3         // buat object dari class laptop (instansiasi)
4         Laptop laptopRudi = new Laptop();
5         Laptop laptopLisa = new Laptop();
6         // set nilai property
7         laptopRudi.merk = "Lenovo";
8         laptopRudi.warna = "Hitam";
9         laptopRudi.harga = 5000000;
10        // tampilkan property object laptop
11        System.out.println("Merk: " + laptopRudi.merk);
12        System.out.println("Warna: " + laptopRudi.warna);
13        System.out.println("Harga: " + laptopRudi.harga);
14    }
15 }
16
```

## 3. CONSTRUCTOR

Constructor adalah **method khusus** yang dijalankan secara otomatis pada saat sebuah object dibuat, yakni saat proses instansiasi. Constructor biasa dipakai untuk membuat proses awal dalam persiapan object, seperti memberi nilai kepada property, memanggil method internal serta beberapa proses lain yang dirasa perlu.

Dalam bahasa Java, constructor dibuat dengan cara menulis sebuah method yang namanya sama dengan nama class. Sebagai contoh jika kita memiliki class Laptop, maka function constructor juga harus bernama **Laptop()**.

Sebuah constructor tidak mengembalikan nilai sehingga tidak perlu menulis tipe data sebelum nama function. Constructor juga harus memiliki hak akses public, yang apabila tidak ditulis dianggap **public** secara default. Berikut contoh kode program cara membuat **constructor** dalam bahasa Java:



*\*pada file Laptop.java\**

```
1  class Laptop {
2      //property
3      String merk;
4      String warna;
5      int harga;
6
7      // method khusus
8      public Laptop(String merk, String warna, int harga) {
9          this.merk = merk;
10         this.warna = warna;
11         this.harga = harga;
12     }
13     //method
14     void hidupkanLaptop(){
15         System.out.println("Laptop Hidup");
16     }
17     void matikanLaptop(){
18         System.out.println("Laptop Mati");
19     }
20 }
21
```

*\*pada file Main.java\**

```
1  public class Main {
2      public static void main(String[] args) throws Exception{
3          // buat object dari class laptop (instansiasi)
4          Laptop laptopRudi = new Laptop("Lenovo", "Hitam", 5000000);
5          Laptop laptopLisa = new Laptop("Asus", "Putih", 6000000);
6          // set nilai property
7          // laptopRudi.merk = "Lenovo";
8          // laptopRudi.warna = "Hitam";
9          // laptopRudi.harga = 5000000;
10
11
12         // tampilkan property object laptop
13         System.out.println("Merk: " + laptopRudi.merk);
14         System.out.println("Warna: " + laptopRudi.warna);
15         System.out.println("Harga: " + laptopRudi.harga);
16         System.out.println("=====");
17         System.out.println("Merk: " + laptopLisa.merk);
18         System.out.println("Warna: " + laptopLisa.warna);
19         System.out.println("Harga: " + laptopLisa.harga);
20         // tampilkan hasil method object Laptop
21         laptopRudi.hidupkanLaptop();
22         laptopRudi.matikanLaptop();
23     }
24 }
```