

# 1

## Pengenalan Object Oriented Programming

### Tujuan Instruksional Khusus:

- *Praktikan ini bertujuan untuk memberi pemahaman mengenai Object Oriented Programming*

## Teori

### 1. Class

Didefinisikan Class sebagai sebuah blue print, atau prototipe, yang mendefinisikan variable-variabel dan metode-metode yang umum untuk semua objek dari jenis tertentu. Class mendefinisikan atribut dan perilaku objek yang dibuatnya. Class merupakan definisi formal suatu abstraksi. Class berlaku sebagai template untuk pembuatan objek-objek. Class berisi abstraksi yang terdiri dari nama class, atribut dan service/behaviour.

Bagian-bagian dari sebuah Class secara umum penulisan class terdiri atas 2 bagian yakni:

#### 1.1 Class Declaration

Bentuk Umum :

```
[modifier] class <nama_kelas>
{
    ...
    ...
    <class body>
    ...
    ...
}
```

[modifier] adalah pengaturan level akses terhadap kelas tersebut. Dengan kata lain, modifier ini akan menentukan sejauh mana kelas ini dapat digunakan oleh kelas atau package lainnya. Adapun macam-macam modifier ialah :

- **kosong / default / not specified**  
Kelas tersebut dapat diakses oleh kelas lain dalam satu package.
- **public**  
Kelas tersebut dapat dipakai dimanapun, maupun kelas lain atau package lain.

- `private`  
Kelas tersebut tidak dapat diakses oleh kelas manapun.

## 1.2 Class Body

Class Body merupakan bagian dari kelas yang mendeklarasikan kode program java. Class Body tersusun atas:

- a. Konstruktor
- b. Variable Instance (Atribut)
- c. Method (dikenal juga sebagai function atau def)

Untuk dapat menggunakan kelas yang telah didefinisikan, anda harus membuat sebuah objek dari kelas tersebut (instance class), dengan syntax:

```
NamaKelas namaObjek = new NamaKelas ( [parameter] );
```

Contoh:

```
Hitungluas segitiga = new Hitungluas();
```

## 1.3 Instance Variables (Atribut)

Suatu objek dapat dibedakan berdasarkan sifat (behavior) yang berbeda. objek juga dapat dibedakan berdasarkan atributnya. Misalnya burung dapat dibedakan berdasarkan suara kicauan, warna bulu, bentuk tubuh, dsb. . Dalam bahasa lain dikenal juga sebagai property yang mana merupakan ciri-ciri dari sebuah objek.

Atribut yang membedakan suatu instance objek burung yang satu dengan yang lainnya disebut instance variable.

Bentuk Umum :

```
[modifier] <type_data> <nama_variabel> = [nilai_default];
```

Contoh :

```
public double tinggi;  
private int berat = 70;
```

Modifier untuk atribut, yaitu `public`, `protected`, `private`. Penjelasan modifier atribut serupa dengan penjelasan modifier pada kelas.

Adapun perbedaan local dan instance variable adalah :

1. Instance variable dideklarasikan di dalam kelas tetapi tidak di dalam method.

```
class segitiga {  
    double tinggi = 15.2; // ini adalah variabel instance  
    String jenis; // ini adalah variabel instance  
    int tambah() {  
        return 3;  
    }  
}
```

2. Local variable dideklarasikan di dalam method.

```
int tambah() {  
    int total = tinggi * 2;    // total adalah variabel local  
    return total; }
```

#### 1.4 Method

Sebuah method adalah bagian-bagian kode yang dapat dipanggil oleh kelas, badan program atau method lainnya untuk menjalankan fungsi yang spesifik di dalam kelas. Secara umum method dalam java adalah sebuah fungsi.

Berikut adalah karakteristik dari method :

1. Dapat mengembalikan / melaporkan nilai balikkan (return value) atau tidak (void)
2. Dapat diterima beberapa parameter yang dibutuhkan atau tidak ada parameter sama sekali. Parameter bisa juga disebut sebagai argumen dari fungsi. Parameter berguna sebagai nilai masukkan yang hendak diolah oleh fungsi.
3. Setelah method telah selesai dieksekusi, dia akan kembali pada method yang memanggilnya.

#### Deklarasi sebuah method

Method terdiri atas dua bagian yakni :

1. Method declaration
2. Method Body

Method dapat digambarkan sebagai sifat (behavior) dari suatu class. Untuk mendefinisikan method pada dalam class digunakan syntax :

```
[modifier] <tipe_data_return> nama_method( [parameter] )  
{  
    ...  
    ...  
    ...  
    return <tipe_data_return>;  
}
```

Contoh :

```
public int Perkalian ( int y;int z )  
{  
    return y * z ;  
}
```

### 1.5 Modifier pada method

Modifier menentukan level pengaksesan sebuah method. Hal ini menentukan apakah sebuah method biasa diakses oleh objek lain, objek anak, objek dalam satu paket atau tidak dapat diakses oleh suatu object sama sekali berikut adalah beberapa jenis level access:

- **Public**  
Atribut ini menunjukkan bahwa fungsi/method dapat diakses oleh kelas lain.
- **Private**  
Atribut ini menunjukkan bahwa fungsi atau method tidak dapat diakses oleh kelas lain
- **Protected**  
Atribut ini menunjukkan bahwa fungsi atau method bisa diakses oleh kelas lain dalam satu paket dan hanya kelas lain yang merupakan subclass nya pada paket yang berbeda.
- **Tanpa modifier**  
Atribut ini menunjukkan bahwa method dapat diakses oleh kelas lain dalam paket yang sama.

### 1.6 Method tanpa nilai balikan

Method ini merupakan method yang tidak mengembalikan nilai. Maka dari itu, kita harus mengganti tipe kembalian dengan kata kunci void. Berikut ini kode program yang dimaksud:

	Program Latihan Praktikum 1.1
1	class Kotak{
2	double panjang;
3	double lebar;
4	double tinggi;
5	//mendefinisikan method void (tidak mengembalikan nilai)
6	void cetakVolume(){
7	System.out.println("Volume kotak = " +(panjang*lebar*tinggi));
8	}
9	}
10	
11	class DemoMethod1{
12	public static void main(String[] args){
13	Kotak k1, k2, k3;
14	//instansiasi objek
15	k1=new Kotak();
16	k2=new Kotak();
17	k3=new Kotak();
18	
19	//mengisi data untuk objek k1
20	k1.panjang=4;
21	k1.lebar=3;
22	k1.tinggi=2;
23	//mengisi data untuk objek k2
24	k2.panjang=6;
25	k2.lebar=5;
26	k2.tinggi=4;

```
27 //mengisi data untuk objek k3
28 k3.panjang=8;
29 k3.lebar=7;
30 k3.tinggi=6;
31 //memanggil method cetakVolume() untuk masing-masing
32 //objek
33 k1.cetakVolume();
34 k2.cetakVolume();
35 k3.cetakVolume();
36 }
37 }
```

### Latihan 1.1

1. Jalankan program diatas
2. Berapakah volume yang ditampilkan untuk ketiga kotak diatas?

### 1.7 Method dengan nilai balikan

Di sini, kita akan memodifikasi program sebelumnya dengan mengganti method cetakVolume() menjadi method hitungVolume() yang akan mengembalikan nilai dengan tipe double. Berikut ini kode program yang dimaksud:

	Program Latihan Praktikum 1.2
1	class Kotak{
2	double panjang;
3	double lebar;
4	double tinggi;
5	//mendefinisikan method yang mengembalikan tipe double
6	double hitungVolume(){
7	//menghitung volume
8	double vol = panjang*lebar*tinggi;
9	//mengembalikan nilai
10	return vol;
11	}
12	}
13	class DemoMethod2{
14	public static void main(String[] args){
15	Kotak k1, k2, k3;
16	//instansiasi objek
17	k1=new Kotak();
18	k2=new Kotak();
19	k3=new Kotak();
20	//mengisi data untuk objek k1
21	k1.panjang=4;
22	k1.lebar=3;
23	k1.tinggi=2;
24	//mengisi data untuk objek k2
25	k2.panjang=6;
26	k2.lebar=5;
27	k2.tinggi=4;
28	//mengisi data untuk objek k3
29	k3.panjang=8;
30	k3.lebar=7;
31	k3.tinggi=6;

32	System.out.println("Volume k1 = "+k1.hitungVolume());
33	System.out.println("Volume k2 = "+k2.hitungVolume());
34	System.out.println("Volume k3 = "+k3.hitungVolume());
35	}
36	}

### Latihan 1.2

1. Jalankan program diatas
2. Berapakah volume yang ditampilkan untuk ketiga kotak diatas?

### 1.8 Parameter

Dengan adanya parameter, sebuah method dapat bersifat dinamis dan general. Artinya, method tersebut dapat mengembalikan nilai yang beragam sesuai dengan nilai parameter yang dilewatkan. Terdapat dua istilah yang perlu anda ketahui dalam bekerja dengan method, yaitu parameter dan argumen. Parameter adalah variabel yang didefinisikan pada saat method dibuat, sedangkan argumen adalah nilai yang digunakan pada saat pemanggilan method. Dalam referensi lain, ada juga yang menyebut parameter sebagai parameter formal dan argumen sebagai parameter aktual. Perhatikan kembali definisi method berikut :

```
int luasPersegiPanjang(int panjang, int lebar){
    return panjang * lebar;
}
```

Di sini, variabel panjang dan lebar disebut parameter.

Luas1 = luasPersegiPanjang(10, 5);

Adapun pada statemen diatas, nilai 10 dan 5 disebut argumen. Sekarang, kita akan mengimplementasikan konsep di atas ke dalam kelas Kotak. Di sini data panjang, lebar, dan tinggi akan kita isikan melalui sebuah method. Berikut ini kode program yang dapat digunakan untuk melakukan hal tersebut.

Program Latihan Praktikum 1.3	
1	class Kotak{
2	double panjang;
3	double lebar;
4	double tinggi;
5	
6	//mendefinisikan method dengan parameter
7	void isiData(double p, double l, double t){
8	panjang = p;
9	lebar = l;
10	tinggi = t;
11	}
12	double hitungVolume(){
13	return (panjang*lebar*tinggi);
14	}
15	}

```

16 class DemoMethod3{
17     public static void main(String[] args){
18         Kotak k;
19         //instansiasi objek
20         k = new Kotak();
21
22         //memanggil method isiData()
23         k.isiData(4,3,2);
24
25         System.out.println("Volume kotak = " + k.hitungVolume());
26     }
27 }

```

### Latihan 1.3

1. Jalankan program diatas
2. Berapakah volume yang ditampilkan untuk kotak diatas?

Bagaimana jika bagian lain dari program ingin tahu juga nilai volume itu tetapi tidak ingin menampilkannya (mencetaknya). Apabila terdapat suatu fungsi yang tidak menghasilkan suatu nilai apapun maka bagian return type ini diganti dengan void .

### Contoh penggunaan return:

	Program Latihan Praktikum 1.4
1	package cobaCoba;
2	import java.util.Scanner;
3	
4	class balok {
5	int p, l, t;
6	int volume( int p, int l, int t)
7	{
8	return (p*l*t);
9	}
10	
11	public static void main(String args[]) {
12	Scanner masuk = new Scanner(System.in);
13	//fungsi untuk menginputkan suatu nilai
14	System.out.print("Panjang = "); int a=masuk.nextInt();
15	System.out.print("Lebar = "); int b=masuk.nextInt();
16	System.out.print("Tinggi = "); int c=masuk.nextInt();
17	
18	balok coba = new balok();
19	System.out.print("\nVolume balok = "+ coba.volume(a,b,c));
20	}
21	}

### Latihan 1.4

1. Jalankan program diatas
2. Berapakah volume yang ditampilkan balok diatas?

### 1.9 Method Static

Sebuah method static dapat diakses tanpa harus melakukan instantiasi terlebih dahulu.

Pemanggilan method static dilakukan dengan format :

Nama\_kelas.nama\_method();

Nama\_kelas diberikan bila method tersebut dipanggil dari kelas yang berbeda.

Contoh :

	Program Latihan Praktikum 1.5
1	public class persegi {
2	static int hitungluas(int p, int l){
3	return p*l;
4	}
5	public static void main(String[] args) {
6	int z=0;
7	z=hitungluas(3,2);
8	System.out.println(z);
9	}
10	}

### Latihan 1.5

1. Jalankan program diatas
2. Berapakah luas yang ditampilkan persegi diatas?

### Tugas 1

1. Buatlah program yang dapat merepresentasikan sebuah buku dan atributnya, dimana pada program tersebut dapat mengetahui biaya sewa dan biaya denda dari buku tersebut!

#### Input contoh (main) :

	Program Latihan Praktikum 1.5
1	Buku doraemon = new Buku("Doraemon", "Komik", 1000, 100);
2	
3	System.out.println("Peminjaman    "+    doraemon.getTipe()    +"
4	"+doraemon.getJudul() + " selama 4 hari akan menghasilkan biaya
5	sewa    sebesar    Rp"+doraemon.getBiayaSewa(4)+"",00.    Apabila
6	terlambat 2hari, maka peminjam wajib membayar denda sebesar
7	Rp"+doraemon.getDenda(2)+"",00.");

#### Output :

Peminjaman komik doraemon selama 4 hari akan menghasilkan biaya sewa sebesar Rp4000,00. Apabila terlambat 2hari, maka peminjam wajib membayar denda sebesar Rp200,00.

**Buatlah contoh input lain yang berbeda dari contoh diatas!**