

## MODUL 3 – Kelas dan Objek

### 3.1. CAPAIAN PEMBELAJARAN

- 1 Mahasiswa memahami konsep dasar class dan object dalam pemrograman berorientasi objek.
- 2 Mahasiswa mampu mengimplementasikan class dan membuat object di dalam program.

### 3.2. ALAT DAN BAHAN

- 1 Seperangkat komputer lengkap/Laptop dengan koneksi internet
- 2 Web Browser (Chrome/Firefox/Opera/Edge/Safari/dll)
- 3 Aplikasi Kantor (Microsoft Office/Libre Office/WPS Office/etc)
- 4 JDK (<https://www.oracle.com/java/technologies/downloads/>)
- 5 Netbeans (<https://netbeans.apache.org/front/main/download/>)

### 3.3. DASAR TEORI

#### 3.3.1. Class

Kelas (*class*) adalah sebuah cetak biru (*blueprint*) atau model yang digunakan untuk membuat instansiasi dari suatu objek. Kelas juga dapat diartikan sebagai kumpulan objek yang memiliki kesamaan dalam atribut atau properti, perilaku, dan hubungan dengan objek lainnya. (Anif & Samsinar, 2023)

Sintaks :

```
[modifier1] Class NamaKelas [modifier2] {  
    //body class  
}
```

Kelas dan method terdiri dari 2 (dua) jenis :

#### 1) Kelas dan Method standar

Kelas dan metode standar adalah kelas yang sudah didefinisikan sebelumnya, yang disediakan oleh Java sebagai bagian dari pustaka referensi. Ada banyak kelas standar yang dapat digunakan untuk berbagai aplikasi spesifik. Kumpulan kelas ini sering disebut sebagai API (Application Programming Interface).

#### 2) Kelas dan Method yang didefinisikan sendiri

Selain kelas dan metode standar, kadang-kadang kita perlu membuat model objek menjadi sebuah kelas dan mendefinisikan data serta metode yang dimiliki. Untuk aplikasi yang besar dan kompleks, kita bisa membaginya ke

dalam beberapa paket, di mana setiap paket berisi beberapa kelas.

## A. Deklarasi Kelas

Kelas dideklarasikan sebagai berikut :

### 1) Deklarasi kelas secara sederhana

```
Class NamaKelas {  
    //Body class  
}
```

### 2) Deklarasi kelas secara lengkap

```
Modifier1 tipeData NamaClass modifier2  
[NamaKelas/NamaInstance ...] {  
    //Body class  
}
```

### 3) Modifier Pada Kelas

Modifier1 digunakan untuk menentukan sifat suatu kelas serta hak akses (privilege) yang dimiliki oleh kelas lain terhadapnya. Terdapat 5 (lima) modifier pada kelas yaitu : public, private, protected, abstract, dan final. Modifier public memungkinkan akses dari kelas lain, baik di dalam maupun di luar paket yang sama. Modifier private mencegah akses dari kelas lain, baik dari paket yang sama maupun berbeda (hanya dapat diakses di dalam kelas dan paket yang sama). Modifier private membatasi akses dari kelas lain dan hanya dapat diakses oleh subclass yang mewarisinya atau kelas lain dalam paket yang sama. Modifier abstract menunjukkan bahwa kelas tersebut tidak dapat diinstansiasi langsung menjadi objek. Kelas ini digunakan pada puncak hirarki kelas dan hanya dapat digunakan melalui pewarisan. Dengan kata lain, kelas ini tidak boleh memiliki objek. Modifier final mencegah kelas tersebut untuk diturunkan menjadi subclass.

Modifier2 digunakan untuk menentukan relasi (extends atau implement) dengan kelas yang lain. Modifier extends digunakan untuk pewarisan (inheritance) sedangkan modifier implements digunakan untuk mengimplementasikan interface pada suatu kelas.

## B. Deklarasi Method

Method adalah fungsi yang dirancang untuk menjalankan tugas tertentu.

Bentuk umum/ sintaks umum dari sebuah method adalah sebagai berikut :

```
Modifier tipeNilaiKembalian namaMethod(paramerter input)
throws exception {
    //Body Method
}
```

Modifier (opsional) yang digunakan pada metode sama dengan modifier yang digunakan untuk mendeklarasikan kelas, karena pada dasarnya modifier dapat diterapkan pada kelas, data, dan metode. Namun, efek penggunaan modifier pada ketiganya mungkin tidak selalu sama persis. final pada kelas berarti bahwa kelas tersebut tidak dapat diturunkan menjadi subclass. final pada metode berarti bahwa metode tersebut tidak dapat di-override oleh subclass lainnya. final pada variabel akan membuat variabel tersebut menjadi sebuah konstanta.

Seperti saat mengakses variabel, kita juga menggunakan notasi titik (.) untuk memanggil method. Bentuk umum pemanggilan metode adalah sebagai berikut:

```
namaObjek.namaMethod(argumen);
```

contoh :

```
objMahasiswa.isiMataKuliah();
```

### 3.3.2. Konstruktor

Konstruktor adalah suatu method yang berfungsi untuk menginisialisasi variabel-variabel instan yang akan dimiliki oleh objek. Konstruktor ini dipanggil saat objek diciptakan.

Ketentuan pembuatan konstruktor adalah sebagai berikut :

- 1) Namanya sama dengan nama Kelasnya.
- 2) Tidak mengembalikan suatu nilai
- 3) Satu kelas bisa memiliki lebih dari satu konstruktor, konstruktor satu dapat memanggil konstruktor lain.
- 4) Dapat dibubuhi modifier public, private, protected

Contoh :

```
Class Cetak {
    Public Cetak(int bil) {
        System.out.println("Nilai yang akan dikirim adalah"+bil);
    }
}
```

```

}
Public Cetak(String nama){
    System.out.println("Nama yang akan dikirim adalah
"+nama);
}
}
}

```

### 3.3.3. Objek

Untuk membuat sebuah objek dari sebuah kelas, diperlukan operator new. Operator ini digunakan untuk membentuk (menginstansiasi) objek dari kelas tersebut. Bentuk umumnya adalah sebagai berikut:

```
namaKelas namaObjek = new namaKelas();
```

Pada saat menggunakan new, terjadi beberapa proses internal sebagai berikut:

- 1) Objek baru tercipta
- 2) Memory dialokasikan untuk objek tersebut
- 3) Method konstruktor dipanggil untuk menginisialisasi objek

Untuk mengakses data/variabel dan method dalam sebuah class dengan menggunakan sintaks berikut :

```

namaObjek.namaVariable;
namaObjek.namaMethod();

```

### 3.3.4. Package

Package adalah cara untuk mengelompokkan bagian-bagian program Java menjadi satu kesatuan. Sebuah package dalam Java terdiri dari kumpulan kelas dan/atau interface. Dalam sebuah package, juga dimungkinkan untuk memiliki sub-package. Package dapat berupa package bawaan dari Java, atau package yang dibuat oleh pengguna. Package yang dibuat oleh pengguna ini sering disebut sebagai folder atau lokasi dalam media penyimpanan yang kita miliki.

Dalam package yang kita buat, dapat berisi sejumlah kelas, dan di dalam package tersebut juga dapat terdapat sub-package. Deklarasi sebuah package dimulai dengan menyebutkan nama package di bagian teratas dari sebuah program sumber.

Bentuk umum dari deklarasi sebuah package adalah sebagai berikut :

```

Package namaPackage;
Import package1;

```

```

Import package2;
Class kelas 1 {
}
Class kelas2 {
}
.....

```

### 3.4.PRAKTIKUM

#### 2.4.1. Perisapan

- 1) Buka Netbeans yang sudah terinstall pada komputer
- 2) Buat proyek baru dengan nama PraktikumPBO\_3.
- 3) Buat package baru dengan nama praktikum3.

#### 2.4.2. Membuat class dan objek sederhana

- 1) Buat class baru bernama Hewan dalam package praktikum1.
- 2) Tambahkan dua atribut String nama dan int umur.
- 3) Tambahkan method suara() yang mencetak "Hewan bersuara".
- 4) Buat class utama Main dan buat object Hewan dengan nama kucing, set nilai atribut, dan panggil method suara().
- 5) Kode program

```

5 package praktikum3;
6
7 /**
8  *
9  * @author HP 14
10 */
11 public class Hewan {
12     String nama;
13     int umur;
14
15     void suara() {
16         System.out.println("Hewan bersuara");
17     }
18 }

```

```

5    package praktikum3;
6
7    /**
8     *
9     * @author HP 14
10   */
11   public class Main {
12       public static void main(String[] args) {
13           Hewan kucing = new Hewan();
14           kucing.nama = "Mimi";
15           kucing.umur = 3;
16           kucing.suara();
17       }
18   }

```

- 6) Ubah program di atas agar selain method suara(), ada method info() yang menampilkan informasi lengkap dari object Hewan.

#### 2.4.3. Menggunakan Constructor dan Encapsulation

- 1) Tambahkan constructor pada class Hewan untuk menginisialisasi atribut nama dan umur.
- 2) Ubah atribut nama dan umur menjadi private.
- 3) Tambahkan method getter dan setter untuk kedua atribut tersebut.
- 4) Kode Program :

```

5 package praktikum3;
6 /**
7  *
8  * @author HP 14
9  */
10 public class Hewan {
11     private String nama;
12     private int umur;
13
14     public Hewan(String nama, int umur) {
15         this.nama = nama;
16         this.umur = umur;
17     }
18     public String getName() {
19         return nama;
20     }
21     public void setName(String nama) {
22         this.nama = nama;
23     }
24     public int getUmur() {
25         return umur;
26     }
27     public void setUmur(int umur) {
28         this.umur = umur;
29     }
30     void suara() {
31         System.out.println("Hewan bersuara");
32     }
33     void info() {
34         System.out.println("Nama: " + getName() + ", Umur: " + getUmur());
35     }
36 }

```

```

5 package praktikum3;
6 /**
7  *
8  * @author HP 14
9  */
10
11 public class Main {
12     public static void main(String[] args) {
13         Hewan kucing = new Hewan("Mimi", 3);
14         kucing.suara();
15         kucing.info();
16     }
17 }

```

5) Tambahkan satu method lagi dalam class Hewan bernama berlari(), yang mencetak "Hewan sedang berlari".

6) Buat object lain dengan nama anjing dan gunakan method berlari().

### 3.5.TUGAS MODUL 3

#### 2.5.1. Soal

- Buat class baru bernama Mobil yang memiliki atribut merk, model, dan tahun.
- Buat constructor yang menerima nilai untuk ketiga atribut tersebut.
- Tambahkan method displayInfo() yang menampilkan informasi lengkap mengenai mobil.

- Buat class Main dan ciptakan dua object dari class Mobil, lalu tampilkan informasi kedua object tersebut.
- Ubah class Mobil agar setiap atribut memiliki method getter dan setter.
- Buat method startEngine() yang mencetak "Mesin mobil [merk] menyala", dan panggil method ini pada setiap object.
- Modifikasi program yang telah dibuat untuk menambahkan satu atribut baru pada class Mobil, misalnya warna.
- Tambahkan method untuk mengubah warna mobil dan tampilkan perubahan tersebut dalam method displayInfo().
- Push Proyek tersebut ke dalam github pada folder pertemuan3.

### 2.5.2. **Petunjuk Pengerjaan**

#### a) Laporan:

- Buatlah laporan akhir sesuai dengan praktikum yang dilakukan
- Tuliskan laporan ke dalam akun masing-masing web medium
- Publikasikan laporan tersebut
- Kirimkan tautan laporan ke elita.
- **Batas Pengumpulan:** Sebelum Pertemuan Praktik Ke 4.