

# BAB 1

## CLASS DAN OBJECT

### Tujuan

1. Praktikan mampu memahami konsep OOP (Object Oriented Programming) dari bahasa Java
2. Praktikan mampu membuat class sebagai bentuk implementasi dari bab ini

### Ringkasan Materi

#### A. Pengenalan OOP

OOP adalah sebuah konsep/cara pemrograman dengan menggunakan objek sebagai elemen dasar dari program. Jika kita memperhatikan dunia nyata, kita dapat menemukan beragam objek disekitar kita seperti mobil, singa, manusia dan seterusnya. Objek yang dimaksud di sini, dikarakterisasi oleh atribut dan tingkah lakunya.

Contohnya, objek sebuah mobil mempunyai atribut tipe transmisi, warna dan manufaktur. Objek Mobil juga mempunyai tingkah laku berbelok, mengerem dan berakselerasi. Dengan cara yang sama pula kita dapat mendefinisikan perbedaan sifat dan tingkah laku dari objek singa. Coba perhatikan tabel dibawah ini sebagai contoh perbandingan :

Objek	Atribut	Tingkah Laku
Mobil	Setir/kemudi Rem Pegas	Berbelok Mengerem Mempercepat
Singa	mata kaki Taring	Tidur Berlari Memangsa

Dengan deskripsi ini, objek pada dunia nyata dapat secara mudah diasumsikan sebagai objek perangkat lunak menggunakan atribut sebagai data dan tingkah laku sebagai method. Data dan method dapat digunakan dalam pemrograman game atau perangkat lunak interaktif untuk membuat simulasi objek pada dunia nyata. Contohnya adalah perangkat lunak objek mobil dalam permainan balap mobil ataupun singa dalam perangkat lunak pendidikan interaktif untuk anak-anak.

#### B. Class

Class adalah struktur dasar dari OOP. Class inilah yang nantinya digunakan sebagai *template* atau cetakan dari sebuah objek. Pembentukan objek dilakukan dengan menggunakan class. Class terdiri dari 2 dua komponen yang disebut dengan field (menggambarkan atribut/properti) dan method (menggambarkan tingkah laku). Field merupakan tipe data yang didefinisikan oleh class, sementara method merupakan operasi. Sedangkan objek adalah sebuah instance dari class. Untuk dapat membedakan class dan objek, dapat dilihat contoh pada tabel dibawah ini :

Class Mobil	Objek Mobil A	Objek Mobil B
Nomor plat	N 7221 WZ	N 2010 KT
Warna	Biru	Merah
Manufaktur	Mitsubishi	Toyota
Kecepatan	50 km/h	100 km/h

Ketika diinisialisasi, setiap objek mendapat satu set variabel yang baru. Bagaimanapun, implementasi dari method dibagi diantara objek pada class yang sama. Class menyediakan keuntungan dari *reusability* artinya programmer perangkat lunak dapat menggunakan sebuah class beberapa kali untuk membuat objek (blueprint).

Untuk mendefinisikan class dapat dituliskan sebagai berikut

```

<modifier> class <name> {
    <attribut declaration>
    <constructor declaration>
    <method declaration>
}
Contoh :
public class
    mobil{ int a;
    public mobil(int nilai){
        a = nilai;
    }
    Publig int
    getNilai(){ return
        a;
    }
}

```

## Instansiasi

Instansiasi adalah proses untuk membuat objek dari sebuah class. Membuat instan Objek dari sebuah class dilakukan dengan menggunakan kata kunci **new**. Contohnya pada suatu kasus kita memiliki Class bernama mobil dan kita ingin menginstan objek dari class Mobil pada class mainMobil dan kita beri nama mobil\_A.

```

Mobil.java
public class Mobil{
}

mainMobil.java
public class mainMobil{
    public static void main(String[]
        args){ Mobil mobil_A = new Mobil();
    }
}

```

## Pelaksanaan Percobaan

### A. Class

Ketikkan program di bawah ini

Mobil.java	
1	public class Mobil {
2	private String noPlat;
3	private String warna;
4	private String manufaktur;
5	private int kecepatan;
6	public void setNoPlat(String s){
7	noPlat = s;
8	}
9	public void setWarna(String s){
10	warna = s;
11	}
12	public void setManufaktur(String s){
13	manufaktur = s;
14	}
15	public void setKecepatan(int i){

```

16         kecepatan = i;
17     }
18     public void displayMessage() {
19         System.out.println("Mobil anda adalah bermerek
"+manufaktur);
20         System.out.println("mempunyai nomor plat "+noPlat);
21         System.out.println("serta memililki warna "+warna);
22         System.out.println("dan mampu menempuh kecepatan
"+kecepatan);
23     }
24 }

```

Selanjutnya kita akan membuat main class dengan MainMobil dan menginstan objek baru dari class tersebut. Perhatikan pada baris 4 dan 12 terdapat deklarasi **new** yang artinya perintah untuk menginstan objek baru dari class mobil

MainMobil.java

```

1  public class MainMobil {
2      public static void main(String[] args) {
3          //instan objek bernama m1
4          Mobil m1 = new Mobil();
5          m1.setKecepatan(50);
6          m1.setManufaktur("Toyota");
7          m1.setNoPlat("AB 1231 UA");
8          m1.setWarna("Merah");
9          m1.displayMessage();
10         System.out.println("=====");
11         //instan objek baru bernama m2
12         Mobil m2 = new Mobil();
13         m2.setKecepatan(100);
14         m2.setManufaktur("Mitsubishi");
15         m2.setNoPlat("N 1134 AG");
16         m2.setWarna("Pink");
17         m2.displayMessage();
18         System.out.println("=====");
19         //merubah warna dari objek m1
20         System.out.println("mobil pada objek m1 di rubah menjadi warna
hijau");
21         m1.setWarna("Hijau");
22         //menampilkan hasil perubahan
23         m1.displayMessage();
24     }
25 }

```

## Data dan Analisis hasil percobaan

### A. Class

Pertanyaan

1. Apakah yang disebut dengan variabel instance dan lokal variabel? Jelaskan perbedaanya!

.....

.....

2. Lakukan percobaan diatas dan benahi jika menemukan kesalahan!

.....

- .....
3. Rubah kode pada mainMobil diatas menjadi proses meminta masukan dari user dan buat menjadi interaktif!  
.....  
.....
  4. Tambahkan method pada class mobil bernama setWaktu yang berparameter double, yang kemudian disimpan pada variabel waktu! (Ketetuannya adalah user harus menginputkan dalam satuan jam)  
.....  
.....
  5. Tambahkan method bernama rubahSekon mempunyai parameter bertipe double dan hanya dapat dipanggil pada class mobil. Method ini memiliki fungsi untuk merubah masukan user yaitu jam menjadi sekon. Method tersebut di panggil pada method setWaktu dengan nilai parameter adalah nilai dari variabel parameter method setWaktu!  
.....  
.....
  6. Tambahkan method pada class mobil dan hanya dapat dipanggil pada class mobil bernama rubahKecepatan yang mempunyai fungsi untuk merubah format kecepatan yang awalnya km/h menjadi m/s. Dipanggil di method setKecepatan!  
.....  
.....
  7. Tambahkan method pada class mobil bernama hitungJarak yang mempunyai aksi untuk menghitung jarak yang dapat di tempuh oleh mobil dengan rumus jarak = kecepatan \* waktu!  
.....  
.....
  8. Tambahkan informasi jarak yang dapat ditempuh pada method displayMessage kemudian rubah satuannya yang awalnya m (meter) menjadi km (kilometer)!  
.....  
.....
  9. Mahasiswa A ingin menulis pada sebuah buku tulis yang ingin dia miliki, isi lembar buku tersebut adalah 50 lembar. Setiap harinya ia menulis sebanyak 100 kata perhari yang cukup untuk 1/2 halaman buku. Buatlah rumus untuk menghitung berapa lama ia menghabiskan 1 buku tersebut serta identifikasilah objek, dan karakteristiknya kemudian implementasikan dalam bentuk class.  
.....  
.....

**Tugas Praktikum**

Suatu perpustakaan di kampus X memiliki banyak koleksi buku, dan buku-buku tersebut dikategorikan berdasarkan jenisnya. Ada 7 kategori dalam perpustakaan tersebut, yaitu teknologi, filsafat, sejarah, agama, psikologi, politik, dan fiksi. Setiap kategori pasti memiliki banyak buku. Setiap buku ditulis oleh setidaknya 1 penulis, walaupun tak menutup kemungkinan buku tersebut ditulis oleh banyak penulis.

Dari studi kasus di atas, tentukan entitas-entitas yang terlibat beserta propertiesnya dan implementasikan ke dalam kode program, serta tampilkan ke layar nilai dari properties dari entitas tersebut. (Minimal tiap kategori ada 5 buku)