

Tujuan :

- a. Memahami unsur-unsur pembentuk class dalam Java
- b. Mempraktekkan pembuatan class beserta objeknya
- c. Memahami dan mempraktekkan kegunaan package

Alokasi Waktu : 2 x 60 menit

A. Dasar Teori :

Bayangkan sebuah benda, kita akan menyebutnya objek. *Objek* apapun tentunya memiliki penyusun dan rancangan yang membuatnya terealisasi, dan *kelas* adalah rancangan atau *blue print* dari objek tersebut. Dalam pemrograman, rancangan mengenai objek dipaparkan dalam bentuk source code. Komponen yang sering dijumpai dalam kelas yakni atribut (sifat atau ciri yang dimiliki objek yang membantu kita membayangkan wujud dari objek tersebut) dan method (kata kerja yang dapat dilakukan oleh/pada objek dan khas bagi objek tersebut).

Contoh kelas misalnya manusia, burung, dan rambutan. Objek yang bisa diinstansiasi dari masing-masing kelas tersebut misalnya Nanda, Shiva, Hitler, lalu Beo, Kenari, Merpati, dan RambutanRafia, RambutanAce, RambutanJawa. Method dari kelas manusia bisa berupa bernafas, berlari, melompat, untuk kelas burung bisa berupa terbang, berkicau, juga bernafas, sedangkan untuk kelas rambutan misalnya membusuk, dipetik, dan dimakan.

2.1. CLASS

Deklarasi class pada Java:

```
[modifier1] class NamaKelas [modifier2] {  
    class body;  
}
```

Keterangan :

1. Tanda [] bersifat optional.
2. Modifier1 dapat berupa : `public`, `abstract`, `final`
3. Modifier2 dapat berupa : `extends`, `implements`
4. Class body merupakan isi dari program yang terdiri dari constructor, atribut dan method.

2.2 METHOD

Deklarasi method:

```
[modifier] tipeNilaiKembalian namaMethod (parameter input) {  
    method body;  
}
```

Keterangan :

- a) Tanda [] bersifat optional.
- b) Modifier dapat berupa : `public`, `protected`, `private`, `static`, `abstract`, `final`, `native` dan `synchronized`
- c) Dilihat dari parameter input, method dibagi menjadi dua yaitu membutuhkan input atau tidak. Bila membutuhkan input, maka tipe data dari parameter input harus dituliskan. Gunakan koma untuk memisahkan setiap parameter input. Bila tidak membutuhkan input, maka bagian parameter input dikosongkan.

Cara pemanggilan method:

- 1. `NamaKelas>NamaMethod(parameterInput);`
- 2. `NamaObjek>NamaMethod(parameterInput);`

2.3. CONSTRUCTOR

Constructor merupakan method yang namanya sama dengan nama kelas dimana method tersebut berada. Constructor dieksekusi pertama kali saat suatu kelas diinstansiasi menjadi objek. Biasanya, Constructor digunakan untuk menginisialisasi nilai awal (memberikan nilai default) pada atribut-atribut yang dimiliki oleh suatu objek saat objek itu pertama kali dibuat.

2.4. MODIFIER

Modifier adalah sifat yang dimiliki oleh setiap atribut, method maupun kelas dalam java. Modifier akses adalah modifier yang selalu digunakan, modifier akses terdiri dari `private`, `default`, `protected` dan `public`. Kriteria modifier-modifier tersebut adalah :

Wilayah Akses	public	protected	default	Private
Kelas yang sama	ya	ya	ya	ya
Antar kelas dalam package yang sama	ya	ya	ya	Tidak
SubKelas di package yang berbeda	ya	ya	Tidak	Tidak
Bukan subkelas, package yang berbeda	ya	Tidak	Tidak	Tidak

Beberapa modifier penting lainnya :

static yaitu modifier yang digunakan agar suatu atribut maupun method dapat diakses oleh objek atau kelas lain meski tanpa dilakukan instansiasi terhadap kelas dimana atribut maupun method itu berada, method `main` adalah salah satu contoh method bermodifier `static` yang sering digunakan.

final yaitu modifier yang digunakan untuk mencegah kemungkinan modifikasi terhadap atribut maupun method, dengan modifier ini suatu atribut akan berlaku sebagai konstanta

2.5. OBJECT

Sebuah kelas dapat digunakan untuk membuat banyak objek. Setiap objek dapat diperlakukan secara berbeda oleh objek-objek lain yang menggunakannya. Dibutuhkan operator `new` untuk membuat objek dari suatu kelas.

Objek dibuat dengan cara menuliskan :

```
1. NamaKelas NamaObjek = new NamaConstructor();
2. Nama Kelas NamaObjek;
   NamaObjek = new NamaConstructor();
```

Proses diatas disebut **instansiasi**. Ketika objek selesai dibuat, objek tersebut disimpan di dalam memori dan dapat diakses oleh objek-objek lain melalui `NamaObjek` nya.

2.6. PACKAGE

Package digunakan untuk mengelompokkan file kelas (*.class) yang terkait (karena jenisnya, fungsinya, atau karena alasan lainnya) pada folder yang sama, dimana di dalam setiap kelasnya terdapat directive (statement java dalam source code yang digunakan untuk membuat kelas) package yang mengacu pada folder tersebut.

Deklarasi package:

```
package namapackage;
```

Untuk bisa mengakses kelas yang berbeda package, diperlukan pernyataan import baik pada kelas yang ingin diakses, maupun package yang menampung kelas tersebut, berikut ini contohnya:

```
1. import namaPackage.namaKelas; // untuk mengimport suatu kelas
2. import namaPackage.*;         // untuk mengimport semua kelas dalam package tersebut
```

B. Langkah-Langkah Praktikum :

1. Membuat folder `/home/student/123xx0xxx/bab2/makhlukhidup/hewan`

- Buat folder **bab2** sebagai folder utama.
- Kemudian buatlah folder **makhlukhidup** didalam folder **bab2**
- Terakhir buatlah folder **hewan** didalam folder **makhlukhidup**.

2. Membuat Program **Manusia.java**, **Burung.java** dan **MainMakhlukHidup.java**

- Klik **Application > Accessories > gedit**
- Klik **File > New** atau tekan **Ctrl+N** untuk membuat dokumen baru
- Ketik Source Code di bawah ini pada dokumen baru tersebut

Source 2.1. Manusia.java

```
1: //file class harus berada dalam folder makhlukhidup
2: package makhlukhidup;
3:
4: public class Manusia {
5:
6:     public String nama = "";
7:
8:     public Manusia(String nama) {
9:         this.nama += nama;
10:    }
11:    //baris kosong
12: }
```

d) Simpan dengan nama **Manusia.java** dalam folder **/home/student/123xx0xxx/bab2/makhlukhidup**

e) Klik **File > New** atau tekan **Ctrl+N** untuk membuat dokumen baru

f) Ketik Source Code di bawah ini pada dokumen baru tersebut

Source 2.2. Burung.java

```
1: //file ini harus berada dalam folder makhlukhidup/hewan
2: package makhlukhidup.hewan;
3:
4: public class Burung {
5:     //static: tidak perlu diinstansiasi
6:     //final: nilai konstanta
7:
8:     public static final boolean bisaTerbang = true;
9:     public String nama = "";
10:
11:     public Burung (String nama) {
12:         this.nama = nama;
13:         System.out.println("Nama burung: " + nama);
14:     }
15:
16:     public void simpanNama(String nama) { //method penyimpanan nama
17:         this.nama = nama;
18:     }
19:
20:     public void hapusNama() { //method penghapus nama
21:         nama = "";
22:     }
23:
24:     public String bacaNama() { //method pengambil nama
25:         return nama;
26:     }
27: }
```

g) Simpan dengan nama **Burung.java** dalam folder **/home/student/123xx0xxx/bab2/makhlukhidup/hewan**

h) Klik **File > New** atau tekan **Ctrl+N** untuk membuat dokumen baru

i) Ketik Source Code di bawah ini pada dokumen baru tersebut

Source 2.3. **MainMakhlukHidup.java**

```

1:  ///file ini harus berada dalam folder bab2
2:  import makhlukhidup.hewan.Burung;
3:
4:  public class MainMakhlukHidup {
5:
6:      public static void main(String[] args) {
7:          makhlukhidup.Manusia m = new
8:  makhlukhidup.Manusia("Maurer");
9:          System.out.println("nama: " + m.nama);
10:         System.out.println("Semua burung pasti bisa terbang? " +
11:  Burung.bisaTerbang);
12:         Burung b = new Burung("Beo");
13:         b.hapusNama();
14:         b.simpanNama("Kasuari");
15:         System.out.println("Nama baru burung: " + b.bacaNama());
16:     }
17: }

```

j) Simpan dengan nama **MainMakhlukHidup.java** dalam folder **/home/student/123xx0xxx/bab2**

k) Klik **Application > Accessories > Terminal**

l) Compile file java di Terminal:

javac MainMakhlukHidup.java

javac makhlukhidup/Manusia.java

javac makhlukhidup/hewan/Burung.java

m) Jalankan program di Terminal:

java MainMakhlukHidup

3. Mengimplementasikan **Constructor** pada class Manusia.java, dan Method pada class Hewan.java

a) Buka file **Manusia.java** di **/home/student/123xx0xxx/bab2/makhlukhidup** dengan **Gedit**.

b) Ubah source code pada baris ke 6 dengan source code dibawah ini

c) Ketik Source Code di bawah ini pada dokumen baru tersebut

```

4:         public String nama = "Mario";

```

d) Simpan perubahan pada file **Manusia.java**

e) Klik **Application > Accessories > Terminal**

f) Compile file java di Terminal:

javac makhlukhidup/Manusia.java

g) Jalankan program di Terminal:

java MainMakhlukHidup

h) Amati perubahan yang terjadi.

i) Buka lagi file **Manusia.java** di **/home/student/123xx0xxx/bab2/makhlukhidup** dengan **Gedit**.

j) Tambahkan source code berikut mulai baris ke 11 seperti dibawah ini

```
11:         public Manusia() {  
12:             nama = "Saverio";  
13:         }
```

k) Simpan perubahan pada file **Manusia.java**

l) Buka file **MainMakhlukHidup.java** dengan **Gedit**.

m) Ubah source code pada baris ke 7 dengan source code dibawah ini

```
7:         makhlukhidup.Manusia m = new makhlukhidup.Manusia();
```

n) Simpan perubahan pada file **MainMakhlukHidup.java**

o) Klik **Application > Accessories > Terminal**

p) Compile file java di Terminal:

javac makhlukhidup/Manusia.java

javac MainMakhlukHidup.java

q) Jalankan program di Terminal:

java MainMakhlukHidup

C. Tugas

Buatlah 2 (dua) package yang masing-masing memiliki 2 (dua) kelas dengan tema bebas. Hubungkan setiap kelas dengan minimal 1 (satu) kelas dapat dieksekusi.