

## MODUL 9

### INTERFACE

#### 9.1 Pengertian Object Interface

Secara sederhana, Object Interface adalah sebuah ‘kontrak’ atau perjanjian implementasi method. Bagi *class* yang menggunakan *object interface*, *class* tersebut harus mengimplementasikan ulang seluruh *method* yang ada di dalam *interface*. Dalam pemrograman objek, penyebutan *object interface* sering disingkat dengan ‘*Interface*’ saja.

Jika anda telah mempelajari *abstract class*, maka *interface* bisa dikatakan sebagai bentuk lain dari *abstract class*. Walaupun secara konsep teoritis dan tujuan penggunaannya berbeda. Sama seperti *abstract class*, *interface* juga hanya berisi *signature* dari *method*, yakni hanya nama *method* dan *parameternya* saja (jika ada). Isi dari *method* akan dibuat ulang di dalam *class* yang menggunakan *interface*.

Jika kita menganggap *abstract class* sebagai ‘*kerangka*’ atau ‘*blue print*’ dari *class-class* lain, maka *interface* adalah implementasi *method* yang harus ‘*tersedia*’ dalam sebuah objek. *Interface* tidak bisa disebut sebagai ‘*kerangka*’ *class*.

Jika terdapat *class* komputer, *interface* bisa dicontohkan dengan ‘*mouse*’, atau ‘*keyboard*’. Di dalam *interface mouse*, kita bisa membuat *method* seperti *klik\_kiri()*, *klik\_kanan()*, dan *double\_klik()*. Jika *class laptop* ‘*menggunakan*’ *interface mouse*, maka *class* tersebut harus membuat ulang *method* *klik\_kiri()*, *klik\_kanan()*, dan *double\_klik()*.

#### 9.2. Deklarasi interface

Untuk mendeklarasikan sebuah *interface* gunakan sintaks :

```
interface namaInterface
{
    ..... //isi dari interface namaInterface
}
```

Berikut ini adalah contoh membuat *interface mouse*:

```
public interface mouse
{
    public void klik_kanan();
    public void klik_kiri();
    public void double_klik();
}
```

### 9.3. Implementasi interface

Cara menggunakan suatu interface adalah dengan mengimplementasikan interface tersebut pada class yang menggunakannya. Selain itu, anda juga harus mendefinisikan secara detail method-method yang ada pada interface tersebut.

```
public class NamaClass implements namaInterface
{
    ..... //method dan isi method pada namaInterface
}
```

### 9.4. Perbedaan Interface dengan Abstract Class

Salah satu yang membedakan interface dengan abstract class adalah kita tidak bisa membuat method normal / biasa di dalam Interface.

Perbedaan lain antara Interface dengan Abstract Class adalah: Sebuah class bisa menggunakan lebih dari 1 interface, sedangkan untuk abstract class, kita hanya bisa menggunakan 1 abstract class dalam sekali penurunan class.

### 9.5. Fungsi Interface

*Interface* lebih berperan untuk *menyeragamkan method*. Ia tidak masuk kedalam struktur class seperti *abstract class*. Jika kita menggunakan *abstract class komputer* sebagai ‘*konsep class*’ untuk kemudian diturunkan kepada class lain seperti *class laptop*, *class pc*, dan *class netbook*, maka interface hanya ‘*penyedia method*’. *Interface* tidak termasuk kedalam pewarisan class.

Contoh source code:

#### Mouse.java

```
1 public interface Mouse {
2     String jenis = "Laser Mouse";
3     // Method Interface harus public
4     public void klik_kanan();
5     public void klik_kiri();
6     // Tidak boleh ada method normal / biasa
7     // yang terdapat isi method-nya
8 }
```

### MouseBaru.java

```
1 // Interface dapat diturunkan (Inherit)
2 public interface MouseBaru extends Mouse {
3     public void double_klik();
4 }
```

### Keyboard.java

```
1 public interface Keyboard {
2     public void tekan_enter();
3 }
```

### Komputer.java

```
1 public class Komputer implements MouseBaru {
2
3     public void klik_kanan(){
4         System.out.println("Mouse Komputer : Klik Kanan...");
5     }
6
7     public void klik_kiri(){
8         System.out.println("Mouse Komputer : Klik Kiri...");
9     }
10
11     public void double_klik(){
12         System.out.println("Mouse Komputer : Double Klik...");
13     }
14
15 }
```

### PC.java

```
1 public class PC extends Komputer implements Keyboard {
2
3     public void tekan_enter(){
4         System.out.println("Keyboard PC : Tekan Enter...");
5     }
6
7 }
```

## Laptop.java

```
1 // sebuah class dapat menggunakan banyak interface
2 public class Laptop extends Komputer implements MouseBaru, Keyboard {
3
4     public void jenis(){
5         System.out.println("Jenis Mouse Laptop : "+jenis);
6     }
7
8     public void klik_kanan(){
9         System.out.println("Mouse Laptop : Klik Kanan...");
10    }
11
12    public void klik_kiri(){
13        System.out.println("Mouse Laptop : Klik Kiri...");
14    }
15
16    public void double_klik(){
17        System.out.println("Mouse Laptop : Double Klik...");
18    }
19
20    public void tekan_enter(){
21        System.out.println("Keyboard Laptop : Tekan Enter...");
22    }
23 }
```

## Cetak.java

```
1 public class Cetak {
2     public static void main (String []args) {
3         System.out.println("Komputer : ");
4         Komputer komputer_baru = new Komputer();
5         komputer_baru.klik_kanan();
6         komputer_baru.klik_kiri();
7         komputer_baru.double_klik();
8         System.out.println("=====");
9
10        System.out.println("PC : ");
11        PC pc_baru = new PC();
12        pc_baru.tekan_enter();
13        System.out.println("=====");
14
15        System.out.println("Laptop : ");
16        Laptop laptop_baru = new Laptop();
17        laptop_baru.jenis();
18        laptop_baru.klik_kanan();
19        laptop_baru.klik_kiri();
20        laptop_baru.double_klik();
21        laptop_baru.tekan_enter();
22    }
23 }
```

## MODUL 9

### LATIHAN

### SOAL

- a. Masukkanlah source code pada modul 9 ke dalam konsep interface, abstract, final dan polimorfisme !

