

Praktikum 5

Interface

POKOK BAHASAN

- Interface
- Implements

TUJUAN BELAJAR

Setelah mempelajari materi dalam bab ini, mahasiswa diharapkan mampu:

- Memahami konsep Interface
- Memahami keywords Implements

DASAR TEORI

Interface

Interface adalah sekumpulan konstanta dan atau deklarasi method tanpa menyertakan/ menuliskan body methodnya. Interface biasa digunakan untuk mendeklarasikan koleksi method dan constants yang dapat digunakan oleh satu atau lebih class. Interface adalah kelas yang paling abstract, yang berisi daftar deklarasi method (seluruh method belum memiliki implementasi).

Interface dapat dianalogikan sebagai kontrak yang dapat dipakai oleh setiap kelas. Dalam kehidupan nyata dapat diketahui ada manusia yang bekerja sebagai dosen, tentara, penyanyi, pengacara, dan sebagainya, tentunya manusia-manusia tersebut selain harus memiliki method standard sebagai seorang manusia, juga harus memiliki method yang sesuai dengan pekerjaannya. Dengan demikian untuk membuat objek seorang Bobi bekerja sebagai dosen, harus dibuat kelas yang merupakan turunan kelas manusia yang mengimplementasikan interface dosen.

Bentuk umum deklarasi:

```
[modifier] interface NamaInterface {  
    // deklarasi konstanta  
    // deklarasi method  
} // catatan : modifier static tidak boleh  
digunakan dalam interface
```

Implements

Kelas yang memilki modifier implements artinya kelas tersebut mengimplementasikan satu atau lebih interface. Bila terdapat lebih dari satu interface, gunakan tanda koma di antara interface-interface tersebut.

Bentuk umum implementasi :

```
[modifier] class NamaKelas implements NamaInterface{
```

```
        // penggunaan konstanta
        // implementasi method
    }
```

Interface dan Class

komponen	interface	class
definisi	daftar deklarasi method	model objek
kandungan informasi	isi/implementasi semua methodnya berada di luar interface ini	mendefinisikan attribute dan method secara rinci dan konkret
instansiasi	tidak boleh	boleh

Interface dan Class Abstract

komponen	interface	abstract class
attribute	hanya berupa konstanta	bebas memiliki tipe data apa saja
method	berupa deklarasi	boleh deklarasi, boleh berupa method lengkap
syntax	seluruhnya abstract (berupa deklarasi)	sebagian abstract

TUGAS PRAKTIKUM

1. Buat kelas baru dan tulis kode berikut pada sebuah teks editor

```
public interface Speedometer {  
    public void tambahKecepatan();  
    public void kurangiKecepatan();  
}
```

Gambar 5.1

2. Buat kelas baru dan tulis kode berikut pada sebuah teks editor

```
public class Mobil implements Speedometer{  
    public void tambahKecepatan()  
    {  
        System.out.println("Cara tambah kecepatan mobil: ");  
        System.out.println("\t injak kopling lalu \n"  
            + "\t pindah ke gear yang lebih tinggi\n"  
            + "\t dan gas mobilnya");  
    }  
    public void kurangiKecepatan()  
    {  
        System.out.println("Cara mengurangi kecepatan mobil: ");  
        System.out.println("\t Rem mobil \n"  
            + "\t pindah ke gear yang lebih rendah");  
    }  
}
```

Gambar 5.2

3. Buat kelas baru dan tulis kode berikut pada sebuah teks editor

```
public class Motor implements Speedometer {  
    public void tambahKecepatan()  
    {  
        System.out.println("Cara tambah kecepatan motor: ");  
        System.out.println("\t pindah ke gear yang lebih tinggi\n"  
            + "\t dan gas motornya");  
    }  
    public void kurangiKecepatan()  
    {  
        System.out.println("Cara mengurangi kecepatan motor: ");  
        System.out.println("\t Rem motor \n"  
            + "\t pindah ke gear yang lebih rendah");  
    }  
}
```

Gambar 5.3

4. Lakukan kompilasi/ build kemudian jalankan (run).
5. Amati hasil output!

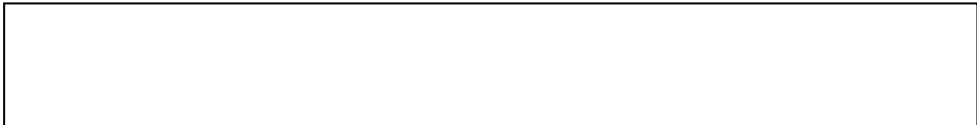


6. Tulis kode berikut ini pada sebuah teks editor

```
public class TesKendaraan {  
  
    public static void main(String [] args)  
    {  
        Mobil mobilku = new Mobil();  
        Motor motorku = new Motor();  
  
        mobilku.tambahKecepatan();  
        mobilku.kurangiKecepatan();  
  
        motorku.tambahKecepatan();  
        motorku.kurangiKecepatan();  
    }  
}
```

Gambar 5.4

- 7. Lakukan kompilasi/ build kemudian jalankan (run).
- 8. Amati hasil output!



9. Buat kelas baru dan tulis kode berikut pada sebuah teks editor

```
public abstract class Hewan {  
    protected String nama;  
    protected int jumKaki;  
  
    public Hewan(String nama, int kaki) {  
        this.nama = nama;  
        jumKaki = kaki;  
    }  
    public abstract void bersuara();  
  
    public void makan() {  
        System.out.println("nyam, nyam, nyam");  
    }  
    public void cetakAtribut() {  
        System.out.println("nama : "+nama);  
        System.out.println("jumlah kaki : "+jumKaki);  
    }  
}
```

Gambar 5.5

10. Buat kelas baru dan tulis kode berikut pada sebuah teks editor

```
interface Manusia {
    public void bernyanyi();
    public void tertawa();
}
```

Gambar 5.6

11. Buat kelas baru dan tulis kode berikut pada sebuah teks editor

```
public class Sapi extends Hewan {

    public Sapi() {
        super("sapi", 4);
    }
    public void bersuara() {
        System.out.println("\nhemmmoooh..., emmmouuh...");
    }
    public static void main(String[] args) {
        Sapi s = new Sapi();
        s.cetakAtribut();
        s.bersuara();
    }
}
```

Gambar 5.7

12. Lakukan kompilasi/ build kemudian jalankan (run).
13. Amati hasil output!

14. Buat kelas baru dan tulis kode berikut pada sebuah teks editor

```
public class Doraemon extends Hewan implements Manusia{
    public Doraemon() {
        super("Doraemon", 2);
    }
    public void bersuara() {
        System.out.println("Hallo Doraemon.....");
    }
    public void bernyanyi() {
        System.out.println("la la la lalala....");
    }
    public void tertawa() {
        System.out.println("hehehe...hehe....");
    }
    @Override
    public void makan() {
        System.out.println("hmmm...kue dorayaki....");
    }
    public void makan2() {
        super.makan();
    }
    public static void main(String[] args) {
        Doraemon d = new Doraemon();
        d.cetakAtribut();
        d.bersuara();
        d.bernayanyi();
        d.tertawa();
        d.makan();
        d.makan2();
    }
}
```

Gambar 5.8

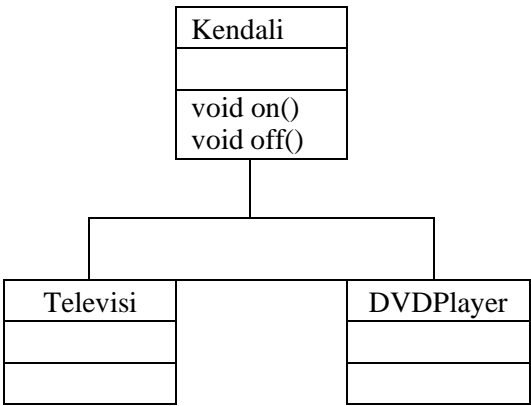
15. Lakukan kompilasi/ build kemudian jalankan (run).

16. Amati hasil output!



LATIHAN

1. Buatlah kode program dengan konsep interface berdasarkan class diagram berikut:



2. Buatlah kode program dengan class super Kendali dan class Anak Radio dan PlayStation

LAPORAN RESMI:

1. Buatlah summary Tugas Praktikum 5
2. Analisa latihan yang telah dilakukan.
3. Berikan kesimpulan dari praktikum ini.