

BAB 8

INTERFACE

Tujuan

1. Mampu memahami konsep Interface dalam pemrograman Java
2. Mampu mengimplementasikan konsep Interface dalam menyelesaikan permasalahan

Ringkasan Materi

A. Pengertian Interface

Pada bab sebelumnya telah dijelaskan bahwa polymorphism hanya memungkinkan untuk mewarisi sifat dari satu kelas abstrak saja, namun tidak jarang juga diperlukan untuk mewarisi sifat lebih dari satu class abstrak, atau biasa disebut juga dengan multiple inheritance. Dalam Java, tidak mendukung multiple inheritance, artinya sebuah class hanya bisa mewarisi sifat dari sebuah superclass atau yang biasa disebut dengan single inheritance. Sebuah Java class tersebut tidak mendukung prinsip multiple inheritance. Dalam memenuhi prinsip kebutuhan multiple inheritance, pada Java terdapat Interfaces yang memiliki aspek seperti multiple inheritance namun berbeda dengan abstract class. Sebuah interface hanya memiliki konstanta dan abstract method. Syntax untuk mendeklarasikan interface adalah sebagai berikut:

```
Modifier interface Interface_name {
    /** Konstanta */
    /** Abstract Method */
}
```

B. Implementasi Interface

Sebuah kelas dapat mengimplementasikan sebuah interface dengan menggunakan kata kunci *implements* sebagai berikut.

```
Modifier class class_name implements Interface_name {
    /** Attribute */
    /** Method */
}
```

Penggunaan kata kunci implements dapat lebih dari 2

```
Modifier class class_name implements Interface_name1,
interface_name2 {
    /** Attribute */
    /** Method */
}
```

Pelaksanaan Percobaan

A. Interface

Identitas.java	
1	package latihaninterface;
2	public interface Identitas {
3	public void tampilkanNama();
4	public void tampilkanUmur();
5	}

Manusia.java

```

1 package latihaninterface;
2 public class Manusia implements MakhlukHidup, Identitas {
3
4     private String nama;
5     private int umur;
6
7     @Override
8     public void makan() {
9         System.out.println("Makan pakai sendok garpu");}
10
11     @Override
12     public void berjalan() {
13         System.out.println("Jalan pakai dua kaki");}
14
15     @Override
16     public void bersuara() {
17         System.out.println("Suaranya merdu");}
18
19     @Override
20     public void tampilkanNama() {
21         System.out.println("Nama saya: " + this.nama);}
22
23     @Override
24     public void tampilkanUmur() {
25         System.out.println("Umur saya: " + this.umur);}
26
27 }

```

Hewan.java

```

1 package latihaninterface;
2 public class Hewan implements MakhlukHidup, Identitas {
3
4     @Override
5     public void makan() {
6         System.out.println("Makan pakai tangan dan mulut");
7     }
8
9     @Override
10    public void berjalan() {
11        System.out.println("Jalan pakai 4 kaki");
12    }
13
14    @Override
15    public void bersuara() {
16        System.out.println("Suaranya nggak jelas");
17    }
18
19    public void tampilkanNama (){}
20
21    public void tampilkanUmur () {}
22 }

```

MakhlukHidup.java

```

1 package latihaninterface;
2 public interface MakhlukHidup
3 {
4     public void makan();
5     public void berjalan();
6     public void bersuara();
7 }

```

Data dan Analisis hasil percobaan

Pertanyaan

```

1 package praktikumpl;
2
3 public interface Colorable {
4     public void howToColor();
5 }
6 public interface Comparable
7
8 {
9     public void compareTo(Object obj);
10 }
11 public class Rectangle implements Colorable, Comparable{ // lass
12     rectangle
13     private String warna;
14     private int kategori;
15
16     public Rectangle() {
17     }
18
19     public Rectangle(String warna) {
20         this.warna = warna;
21     }
22
23     public void howToColor() {
24         if(this.warna == null){
25             System.out.println("tidak ada warna, warna bangun
26 kotak masih polos");
27         }
28         else{
29             System.out.println("bangun kotak sudah diwarnai dengan
30 warna "+this.warna);
31         }
32     }
33
34     public void compareTo(Object obj) {
35         this.kategori = (int) obj;
36         if(this.kategori == 0){
37             System.out.println("ukuran cat yang cocok untuk bangun
38 kotak dengan ukuran kategori " +this.kategori+" yaitu 2.5L" );
39         }
40         else{
41             System.out.println("ukuran cat yang cocok untuk bangun kotak
42 dengan ukuran kategori " +this.kategori+" yaitu 6.5L" );
43         }
44     }
45 }
46 public static void main(String[] args) {
47
48     Rectangle kotak1 = new Rectangle("merah");
49     Rectangle kotak2= new Rectangle();
50     Rectangle kotak3 = new Rectangle();
51     kotak1.howToColor();
52     kotak2.howToColor();
53     kotak3.compareTo(4);
54 }
55

```

1. Lakukan percobaan diatas dan benahi jika menemukan kesalahan serta jelaskan!
tidak ditemukan kesalahan

2. Apakah class yang berbentuk Interface bisa diinstansiasi menjadi sebuah objek? Jelaskan alasannya!

Tidak, interface tidak bisa diinstansiasi menjadi sebuah objek. Karena interface dalam pemrograman berorientasi objek adalah tipe class yang hanya mendefinisikan metode tanpa memberikan implementasi. Interface bertujuan untuk mendefinisikan kontrak atau blueprint yang harus diikuti oleh class yang mengimplementasikannya. Interface secara alami bertipe abstract dan tidak memiliki sebuah konstruktor.

3. Apakah suatu class dapat mengimplementasi class interface yang jumlahnya lebih dari satu? Jelaskan alasannya!

Iya, suatu class dapat mengimplementasikan class interface yang jumlahnya lebih dari satu. Hal ini dikarenakan mengimplementasikan beberapa interface memungkinkan class untuk menggunakan kembali kode yang ada dan menggabungkan berbagai perilaku tanpa harus menulis ulang kode. Hal ini dapat menghindari kompleksitas yang terkait dengan multiple inheritance untuk class, dengan tetap mendukung konsep polimorfisme.

4. Pada interface Identitas.java hapus method tampilkan nama, amati apa yang terjadi dan mengapa demikian?

```
1 public interface Identitas {
2     // public void tampilkanNama();
3
4     public void tampilkanUmur();
5 }
6
```

Manusia.java 1

Yang terjadi adalah error pada class Manusia.java. Hal ini dikarenakan pada class manusia terdapat implementasi method yang berasal dari class identitas.

5. Jika pada class hewan kita hanya ingin mengimplements interface MakhlukHidup saja apa yang terjadi? Jelaskan

```
public class Hewan implements MakhlukHidup {
    @Override
    public void makan() {
        System.out.println(x:"Makan pakai tangan dan mulut");
    }

    @Override
    public void berjalan() {
        System.out.println(x:"Jalan pakai 4 kaki");
    }

    @Override
    public void bersuara() {
        System.out.println(x:"Suaranya nggak jelas");
    }

    public void tampilkanNama() {
    }

    public void tampilkanUmur() {
    }
}
```

Tidak terjadi apa-apa. Hal ini dikarenakan class hewan mengimplementasikan method yang ada di class makhluk hidup. Akan tetapi, method tampilkanNama dan method tampilkanUmur tidak akan berfungsi karena class Hewan tidak lagi mengimplementasikan interface Identitas. Setelah perubahan ini, objek Hewan hanya akan memiliki perilaku yang ditentukan oleh method di MakhlukHidup.

6. Buatlah konstruktor pada manusia dengan parameter umur dan nama kemudian panggil pada Class Main dengan menginstan objek bernama nama anda!

```
public Manusia(String nama, int umur) {
    this.nama = nama;
    this.umur = umur;
}
```

```
public class MainAnalisis {
    Run | Debug
    public static void main(String[] args) {
        Manusia nafa = new Manusia( nama:"Nafasyah Rizky Anandita", umur:19);
    }
}
```

7. Ubah source code diatas menjadi proses meminta inputan dari user dan buat menjadi interaktif!

```
public void informasi(){
    Scanner in = new Scanner(System.in);
    System.out.print(s:"Masukkan nama Anda: ");
    this.nama = in.nextLine();
    System.out.print(s:"Masukkan umur Anda: ");
    this.umur = in.nextInt();
}
```

```
MainAnalisis.java > MainAnalisis > main(String[])
1  import java.util.*;
2
3  public class MainAnalisis {
    Run | Debug
4  public static void main(String[] args) {
5      Scanner in = new Scanner(System.in);
6      Manusia nafa = new Manusia( nama:"Nafasyah Rizky Anandita", umur:19);
7
8      Manusia manusia = new Manusia(nama:"null", umur:0);
9      manusia.informasi();
10
11     manusia.tampilkanNama();
12     manusia.tampilkanUmur();
13
14     manusia.berjalan();
15     manusia.bersuara();
16     manusia.makan();
17 }
18
19
20
```

Output:

```
Masukkan nama Anda: Nafasyah Rizky Anandita
Masukkan umur Anda: 19
Nama saya: Nafasyah Rizky Anandita
Umur saya: 19
Jalan pakai dua kaki
Suaranya merdu
Makan pakai sendok garpu
PS C:\Users\Joe\Desktop\PB0Sem2\Tugasqu\Tugas9>
```

8. Buat objek selain objek diatas dengan menggunakan method yang berbeda dengan yang diatas! (min.1 contoh)

```
public void informasi(){
    Scanner in = new Scanner(System.in);
    System.out.print(s:"Masukkan nama Anda: ");
    this.nama = in.nextLine();
    System.out.print(s:"Masukkan umur Anda: ");
    this.umur = in.nextInt();
}
```

```
public class Hewan implements MakhlukHidup, Identitas {
    private String nama;
    private int umur;

    public Hewan(String nama, int umur) {
        this.nama = nama;
        this.umur = umur;
    }
}
```

```
public void tampilkanNama() {
    System.out.println("Nama hewan saya: " + this.nama);
}

public void tampilkanUmur() {
    System.out.println("Umur hewan saya: " + this.umur);
}

public void informasi(){
    Scanner in = new Scanner(System.in);
    System.out.print(s:"Masukkan nama Hewan: ");
    this.nama = in.nextLine();
    System.out.print(s:"Masukkan umur Hewan: ");
    this.umur = in.nextInt();
}
```

```
import java.util.*;

public class MainAnalisis {
    Run | Debug
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        Manusia nafa = new Manusia( nama:"Nafasyah Rizky Anandita", umur:19);

        Manusia manusia = new Manusia(nama:"null", umur:0);
        manusia.informasi();

        manusia.tampilkanNama();
        manusia.tampilkanUmur();

        manusia.berjalan();
        manusia.bersuara();
        manusia.makan();

        System.out.println(x:"");
        Hewan hewan = new Hewan(nama:"", umur:0);

        hewan.informasi();

        hewan.tampilkanNama();
        hewan.tampilkanUmur();

        hewan.berjalan();
        hewan.bersuara();
        hewan.makan();
    }
}
```

Output:

```
Masukkan nama Anda: Nafasyah Rizky Anandita
Masukkan umur Anda: 19
Nama saya: Nafasyah Rizky Anandita
Umur saya: 19
Jalan pakai dua kaki
Suaranya merdu
Makan pakai sendok garpu

Masukkan nama Hewan: Kucing
Masukkan umur Hewan: 2
Nama hewan saya: Kucing
Umur hewan saya: 2
Jalan pakai 4 kaki
Suaranya nggak jelas
Makan pakai tangan dan mulut
PS C:\Users\Joe\Desktop\PB0Sem2\Tugasqu\Tugas9>
```

Tugas Praktikum

Perusahaan NV. Meneer memiliki koperasi karyawan yang memungkinkan karyawannya berbelanja di koperasi tersebut. Tentunya, karyawan tersebut bisa membayar belanjanya tersebut di akhir bulan melalui pemotongan gaji. Ada 2 kelas yang terlibat disini, Invoice dan Employee. Kedua class tadi mengimplementasikan **interface Payable** yang mana ia hanya memiliki satu method yang harus diimplementasikan di kedua class, yaitu **getPayableAmount()**. Program harus bisa mengolah gaji karyawan di akhir bulan beserta invoice belanjaan karyawan yang nantinya gaji karyawan perbulannya dikurang total harga belanjanya secara polimorfis. Tampilkan informasi dari karyawan tersebut beserta total gaji setelah dipotong hutang belanjaan di koperasi dan tampilkan pula detail belanjanya secara polimorfis pula.

1. Atribut dari Invoice:

String productName, Integer quantity, Integer pricePerItem

2. Atribut dari Employee:

Integer registrationNumber, String name, Integer salaryPerMonth, Invoice[] invoices