

Percobaan 5

Paket dan Import, Interface

Pokok Bahasan

- Konsep paket
- Deklarasi paket
- Penggunaan import
- Pengertian Interface
- Pendeklarasian interface
- Pewarisan Interface
- Penggunaan interface

Tujuan Belajar

Dengan praktikum ini mahasiswa diharapkan dapat:

- Mengenalkan tentang konsep paket pada bahasa pemrograman java
- Dapat memahami tentang cara mendeklarasikan paket
- Dapat memahami tentang cara menggunakan Import
- Menjelaskan tentang pengertian Interface
- Menjelaskan tentang cara mendeklarasikan Interface
- Memahami tentang pewarisan Interface
- Dapat menjelaskan tentang cara menggunakan Interface

Dasar Teori

A. Interface

Terkadang dalam membuat suatu class, kita ingin melakukan abstraksi, dalam arti bahwa kita ingin menentukan apa yang harus atau dapat dilakukan oleh suatu class tetapi tidak menentukan bagaimana cara untuk melakukannya. Dalam hal ini kita dapat menggunakan interface untuk melakukannya. Pada dasarnya interface mirip dengan class biasa, namun properti yang dimiliki bersifat *final*, dan hanya memiliki deklarasi method tanpa memiliki implementasi, mirip seperti method abstrak.

Untuk mengimplementasi suatu interface, suatu class harus mengimplementasi semua method yang telah ditentukan oleh interface tersebut. Namun, detil cara implementasinya ditentukan sendiri oleh class yang bersangkutan.

Pendeklarasian Interface

Secara umum bentuk pendeklarasian interface adalah:

```
Access Spescifier interface NamaInetrface{
Access Specifier tipe-data namaVariabell=value 1;
.
.
.

Access Specifier tipe-data namaVariabell=value N;
Access Specifier tipe-data namaMethod1(parameter);
.
.
.
Access Specifier tipe-data namaMethodN(parameter);

}
```

Berikut ini aturan yang harus diingat tentang pendeklarasian interface:

- Access specifier yang dapat digunakan hanya public atau tidak ada sama sekali. Jika tidak ada access specifier, maka interface tersebut hanya dapat digunakan dalam package yang sama dimana ia dideklarasikan.
- Semua property yang dideklarasikan di dalam interface secara otomatis akan dianggap sebagai static final.
- Semua metod yang dideklarasikan tidak mempunyai implementasi

Penggunaan Interface

Interface digunakan dengan cara mengimplementasikannya ke suatu class menggunakan keyword implements. Bentuk umum pemakaian sebagai berikut:

```
Access Specifier class NamaClass implements
NamaInterface1,..,NamaInterfaceN{
//class implementation
}
```

Contoh:

```
interface infrared{
    Void kirimdatainfra(String data);
}

interface Bluetooth{
    Void kirimdatablue(String data);
}

class handphone implements infrared,Bluetooth{
//class implementasi
}
```

Pewarisan pada Interface

Sebagaimana layaknya class biasa, anda juga dapat melakukan pewarisan pada interface. Ini berarti semua class yang mengimplementasi interface hasil pewarisan tersebut harus mengimplementasi semua metod yang terdapat dalam interface itu sendiri beserta semua metode yang terdapat dalam interface darimana interface tersebut diturunkan.

Contoh:

```
interface A{
    void coba();
}

interface B extends A{
    void cobalagi();
}
```

B. Package dan Import

Package adalah suatu cara untuk memenej class-class yang kita buat. Package akan sangat bermanfaat jika class-class yang kita buat sangat banyak sehingga perlu dikelompokkan berdasarkan kategori tertentu. Contoh:

```
package it;

public class Siswa {
    ...
}
```

```
package telkom;

public class Siswa {
    ...
}
```

Yang perlu kita perhatikan pada saat deklarasikan package, bahwa class tersebut harus disimpan pada suatu direktori yang sama dengan nama package-nya.

Suatu class dapat meng-import class lainnya sesuai dengan nama package yang dipunyainya. Contoh:

```
import it.Siswa;
public class IsiData {
    ...
}
```

Satu hal yang perlu kita ketahui, pada saat kita ingin meng-import suatu class dalam suatu package, pastikan letak package tersebut satu direktori dengan class yang ingin meng-import.

Soal Pendahuluan

1. Keuntungan apa saja yang dapat diperoleh dengan package dalam program?
2. Jelaskan fungsi keyword import dalam hubungannya dengan penggunaan package!

Percobaan

Percobaan 1 : Menggunakan package dan import .

Untuk memahami konsep import dan package . Compile dan eksekusilah program ini!

```
package package1;

public class packageA{
    protected String test;

    public void method(){
        System.out.println("metode dari class
                             packageA");
    }
}
```

```
package package2;

public class packageB{

public void method(){
System.out.println("metode dari class
packageB");
}
}
```

```
package demo;

import package1.packageA;
import package2.packageB;

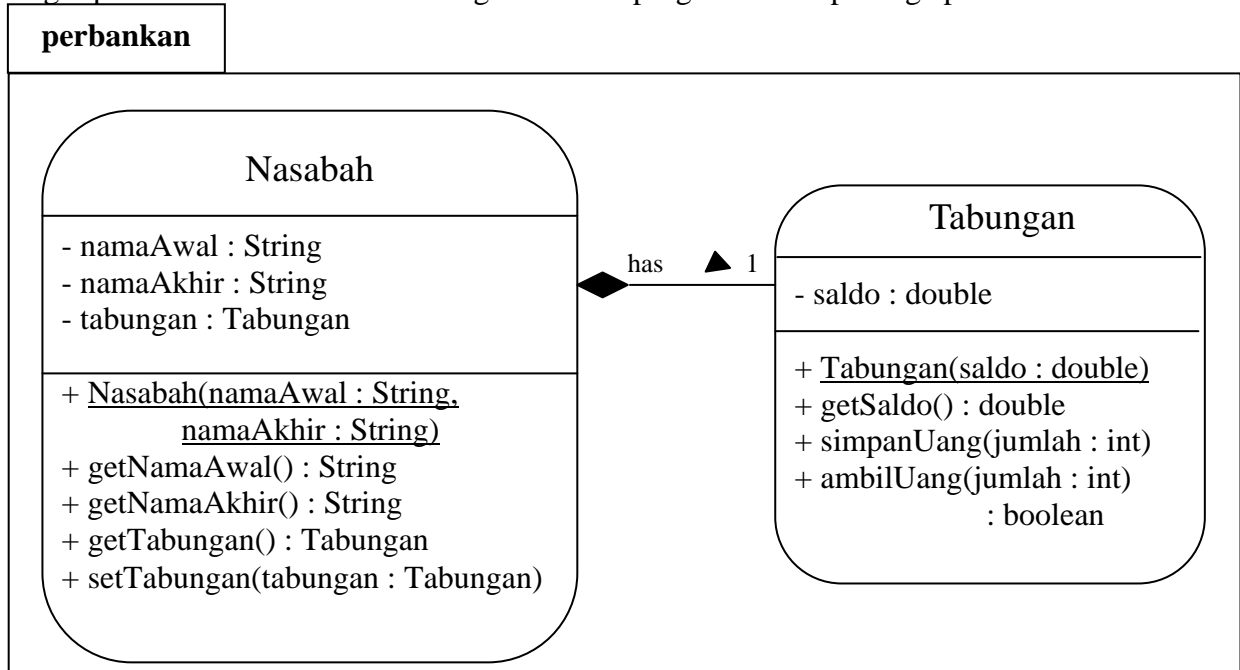
public class demopackage{
public static void main(String args[]){
packageA varA=new packageA();
packageB varB=new packageB();
varA.method();
varB.method();
}
}
```

Hasil Eksekusi:

metode dari class packageA
metode dari class packageB

Percobaan 2 : Menggunakan package dan import

Mengimplementasikan UML class diagram dalam program untuk package perbankan



Transformasikan class diagram diatas ke dalam bentuk program? Jalankan file TesLatihan2.java.

```
import perbankan.*;

public class TesLatihan2 {
    public static void main(String args[]) {
        double tmp;
        boolean status;

        Nasabah nasabah=new Nasabah("Agus","Daryanto");
        System.out.println("Nasabah atas nama : " +
            nasabah.getNamaAwal() + " " +
            nasabah.getNamaAkhir());

        nasabah.setTabungan(new Tabungan(5000));

        tmp=nasabah.getTabungan().getSaldo();
        System.out.println("Saldo awal : " + tmp);

        nasabah.getTabungan().simpanUang(3000);
        System.out.println("Jumlah uang yang disimpan : 3000");

        status=nasabah.getTabungan().ambilUang(6000);
        System.out.print("Jumlah uang yang diambil : 6000");
        if (status)
            System.out.println("    ok");
        else
            System.out.println("    gagal");
    }
}
```

```

nasabah.getTabungan().simpanUang(3500);
System.out.println("Jumlah uang yang disimpan : 3500");

status=nasabah.getTabungan().ambilUang(4000);
System.out.print("Jumlah uang yang diambil : 4000");
if (status)
    System.out.println("    ok");
else
    System.out.println("    gagal");

status=nasabah.getTabungan().ambilUang(1600);
System.out.print("Jumlah uang yang diambil : 1600");
if (status)
    System.out.println("    ok");
else
    System.out.println("    gagal");

nasabah.getTabungan().simpanUang(2000);
System.out.println("Jumlah uang yang disimpan : 2000");

tmp=nasabah.getTabungan().getSaldo();
System.out.println("Saldo sekarang = " + tmp);
}
}

```

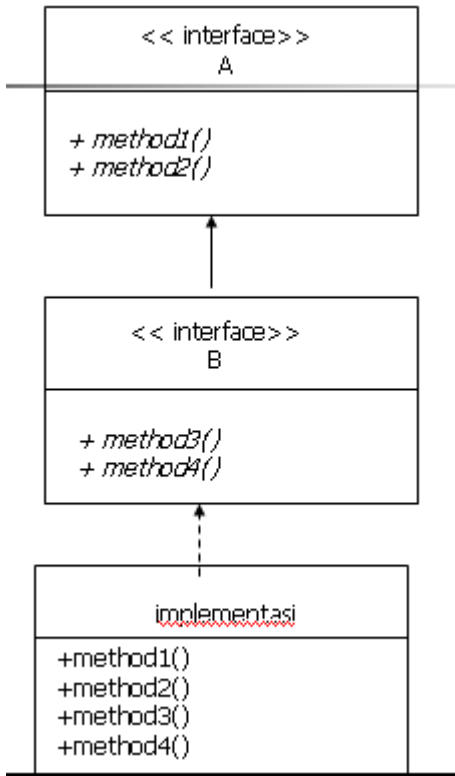
Jika tampilan di layar tampak seperti dibawah ini, maka program anda sudah benar. Jika tidak sama, benahi kembali program anda dan lakukan hal yang sama seperti diatas.

```

Nasabah atas nama : Agus Daryanto
Saldo awal : 5000
Jumlah uang yang disimpan : 3000
Jumlah uang yang diambil : 6000    ok
Jumlah uang yang disimpan : 3500
Jumlah uang yang diambil : 4000    ok
Jumlah uang yang diambil : 1600    gagal
Jumlah uang yang disimpan : 2000
Saldo sekarang = 3500

```

Percobaan 3 : Buatlah program dari UML class diagram dibawah ini!



```
class demo interface{
public static void main(String args[])
{
    implementasi impl=new implementasi();
    impl.method1();
    impl.method2();
    impl.method3();
    impl.method4();
}
}
```

Hasil Eksekusi:

Implementasi method 1.....

Implementasi method 2.....

Implementasi method 3.....

Implementasi method 4.....

Soal Laporan Resmi

1. Apa perbedaan antara Interface dan kelas abstrak? Berikan contoh program untuk memperjelas perbedaan tersebut