

## PERTEMUAN KE – 8

### *Package dan Acces modifier*

#### A. TUJUAN :

- Dapat menggunakan paket java
- Dapat membuat dan menggunakan package yang dibuat
- Dapat mengatur hak akses member

#### B. TEORI SINGKAT

Seperti yang diketahui, Java adalah pemrograman dengan menggunakan banyak kelas. Kelas-kelas tersebut dikelompokkan ke dalam kategori tertentu yang berhubungan disebut dengan *packages* (paket) . Misalnya saja paket `javax.swing.*` berarti semua kelas yang berhubungan dengan `javax.swing` berada dalam direktori tersebut.

Langkah-langkah untuk membuat kelas yang *reusable* adalah sebagai berikut:

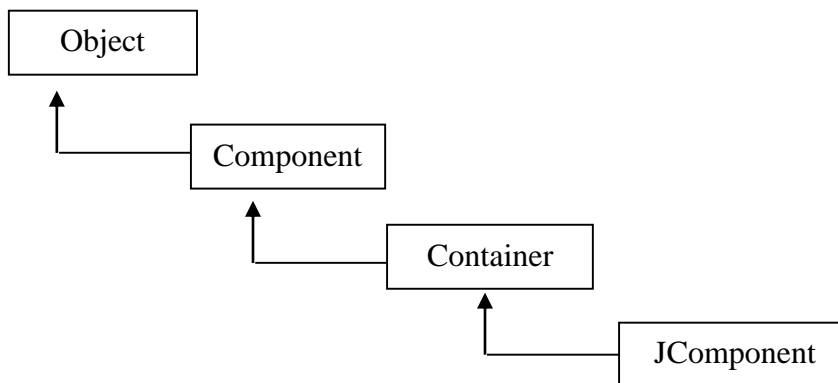
1. Deklarasikan *public class*. Jika kelas tidak bersifat *public*, dia hanya bisa dipergunakan oleh kelas lain di dalam paket yang sama.
2. Pilih nama paket, dan tambahkan sebuah *package declaration* ke file *source code* untuk deklarasi kelas *reusable*. Hanya bisa ada satu deklarasi paket saja di dalam sebuah file *source code* Java dan dia harus mendahului semua deklarasi dan pernyataan lain di dalam file tersebut.
3. Kompilasi kelas tersebut sehingga dia ditempatkan di dalam struktur direktori paket yang sesuai.
4. Import *reusable class* ke dalam sebuah program, dan gunakan kelas tersebut.

Fitur ini menyediakan mekanisme untuk mengatur class dan interface dalam jumlah banyak dan menghindari konflik pada penamaan.

Menggunakan Salah satu Paket Miliknya Java : Komponen Swing

Komponen	Deskripsi
JLabel	Area dimana teks yang tidak bisa diedit atau sebuah icon dapat ditampilkan
TextField	Area dimana user menginputkan data dari keyboard. Area ini juga bisa menampilkan informasi
JButton	Area yang memicu sebuah event ketika diklik dengan mouse

JCheckBox	Komponen GUI yang bisa dipilih atau tidak dipilih
JComboBox	Sebuah drop-down list dari item dari mana user dapat membuat pilihan dengan meng-klik sebuah item dalam list atau mungkin dengan menuliskan dalam box
JList	Area yang berisi sebuah list dari item dari mana user dapat membuat pilihan dengan meng-klik pada beberapa elemen dalam list. Beberapa elemen dapat dipilih
JPanel	Sebuah kontainer dalam mana komponen dapat ditempatkan atau diorganisasi



Gambar 1. Superclass untuk banyak komponen Swing

Gambar 1 memperlihatkan sebuah hirarki inheritance yang berisi class yang mendeklarasikan atribut dan lingkungan yang berhubungan dengan banyak komponen Swing. Class object adalah superclass dari hirarki class Java. Class Component (package java.awt) adalah subclass dari Object, class Container (package java.awt) adalah subclass dari Component, dan class Jcomponent (package javax.swing) adalah subclass dari Container.

### **Struktur Package**

```

package nm_package;

import registration.processing.*;
import java.util.List;
import java.lang.*; //imported by default

class NmClass {
    /* details of NmClass */
}
  
```

### **Access Modifier**

**Public** : Menyatakan bahwa kelas/method/attribute tersebut dapat diakses oleh kelas lain dimanapun.

**Protected** : Menyatakan bahwa kelas/method/attribute tersebut dapat diakses oleh kelas lain yang berada dalam satu package atau kelas lain tersebut merupakan turunannya.

**Private** : Menyatakan bahwa kelas tersebut tidak dapat diakses sama sekali oleh kelas lain bahkan juga tidak dapat diturunkan.

### C. PRAKTIK :

1. Membuat sebuah paket sederhana

Tuliskan program berikut dalam folder mahasiswa yang berada di dalam direktori kerja anda. Jika belum ada buat dulu.

```
package mahasiswa;
public class paket
{
    public static void isiPaket(){
        System.out.println("ini adalah hasil import");
    }
}
```

Simpan dengan nama paket.java pada direktori mahasiswa.

2. Menggunakan paket hasil buatan sendiri. Cobalah program berikut. Simpan dalam direktori kerja anda (di luar folder mahasiswa).

```
import mahasiswa.*;
public class cobaPaket
{
    public static void main(String args[])
    {
        paket coba=new paket();
        coba.isiPaket();
    }
}
```

3. Modifikasi class paket menjadi seperti berikut :

```
package mahasiswa;
public class paket
{
    String nama;
    int umur;
    public static void isiPaket(){
        System.out.println("ini hasil import");
    }
    public void setData(int umur, String nama){
        this.umur = umur;
        this.nama = nama;
    }
    public void info(){
        System.out.println("Nama : "+nama);
    }
}
```

```

        System.out.println("Umur : "+umur);
    }
}

```

4. Panggil method tersebut dari class cobaPaket!
5. Menggunakan paket miliknya java. Cobalah program berikut yang menggunakan paket bawaannya java.

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class LabelTest extends JFrame {
    private JLabel label;

    public LabelTest()
    {
        super( "Mencoba JLabel" );
        Container container = getContentPane();
        container.setLayout( new FlowLayout() );
        label = new JLabel( "Label dengan text" );
        label.setToolTipText( "Ini adalah label1" );
        container.add( label );
        setSize( 500, 400 );
        setVisible( true );
    }

    public static void main( String args[] )
    {
        LabelTest aplikasi = new LabelTest();
        aplikasi.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}

```

## 6. Menggunakan hak *access modifier*

```

class orang
{
    private String nama = "";
    //Dideklarasikan sebagai private karena hanya dipanggil pada class orang
    protected String alamat = "";
    //Dideklarasikan sebagai protected karena dibutuhkan oleh class ini dan class
    dibawahnya

    public orang (String nm, String alm)
    //Dideklarasikan sebagai public karena diperlukan oleh class lainnya
    {
        nama = nm;
        alamat = alm;
    }
    public void setName (String nm)
    { this.nama = nm;}
    public void setAlamat (String alm)
    { this.alamat = alm;}
}

```

```

public String getNama()
{ return nama;}
public String getAlamat()
{ return alamat;}
public String displayInfo()
{ return "Nama : " + nama + ", Alamat : " + alamat;
}
}

class dosen extends orang
{ private String nip = "";
//Dideklarasikan sebagai private karena hanya dipanggil pada class dosen
public dosen (String nm, String np, String alm)
//Dideklarasikan sebagai public karena diperlukan oleh class lainnya
{ super (nm,alm);
nip = np;
}
public void setNIP (String np)
{ this.nip = np;}
public String getNIP()
{ return nip;}
public String displayInfo()
{ return super.displayInfo() + ", NIP : " + nip;
}
}

class mahasiswa extends orang
{ private String npm = "";
//Dideklarasikan sebagai private karena hanya dipanggil pada class mahasiswa
public mahasiswa (String nm, String ni, String alm)
//Dideklarasikan sebagai public karena diperlukan oleh class lainnya
{
super (nm,alm);
npm = ni;
}
public void setNpm (String ni)
{ this.npm = ni;}
public String getNpm()
{ return npm;}
public String displayInfo()
{ return super.displayInfo() + ", NPM : " + npm;
}
}

public class Perkuliahan
{
public static void main (String[] args)
{
dosen dsn = new dosen ("Arch", "666","Aincrad");
mahasiswa mhs = new mahasiswa ("Azwrath", "999","Alfheim");
System.out.println(dsn.displayInfo());
System.out.println(mhs.displayInfo());
}
}

```

#### D. LATIHAN

- Latihan diberikan oleh dosen pengampu pada saat praktikum.

- *Dikerjakan di laboratorium pada jam praktikum.*

#### **E. TUGAS**

- *Tugas diberikan oleh dosen pengampu pada akhir praktikum.*
- *Dikerjakan di rumah dan dilampirkan pada laporan.*