

MODUL 5

INHERITANCE

5.1 Inheritance

Inheritance merupakan proses pewarisan data dan method dari suatu class yang telah ada kepada suatu class baru. Class yang mewariskan disebut dengan **superclass / parent class / base class**, sedangkan class yang mewarisi (class yang baru) disebut dengan **subclass / child class / derived class**.

Subclass tidak dapat mewarisi anggota private dari superclass-nya. Dengan inheritance, class yang baru (subclass) akan mirip dengan class yang lama (superclass) namun memiliki karakteristik yang baru. Dalam Java, subclass hanya bisa memiliki satu superclass (single inheritance) sedangkan superclass bisa memiliki satu subclass atau lebih. Untuk menerapkan inheritance, gunakan statement “*extends*”.

Keyword “*super*” digunakan oleh subclass untuk memanggil constructor, atribut dan method yang ada pada superclass-nya.

Contoh untuk memanggil constructor milik superclass-nya :

```
super ()  
super (parameter)
```

Contoh untuk memanggil atribut dan method milik superclass-nya :

```
super.namaAtribut  
super.namaMethod(parameter)
```

5.2. Method Overriding

Overriding method adalah kemampuan dari subclass untuk memodifikasi method dari superclass-nya, yaitu dengan cara menumpuk (mendefinisikan kembali) method superclass-nya. Contoh overriding method dapat dilihat pada subclass “Mobil” yang mendefinisikan kembali method keterangan() dan hapus() dari class “Kendaraan”.

5.3 Input Data

Untuk menginputkan data dari keyboard ada 2 cara, yaitu :

- Input dari mode console, yaitu dengan memanfaatkan class **BufferedReader** dan **InputStreamReader**.

Untuk bisa mengakses class `BufferedReader` maka perlu mengimpor dari package **java.io.*** dan menambahkan statemen **throws IOException** pada header method main.

Contoh :

```
import java.io.*;
class CobaInput1
{
    public static void main (String []args) throws IOException
    {
        BufferedReader br = new BufferedReader (new InputStreamReader(System.in));
        String nama, kota;

        System.out.print("Nama Anda : ");
        nama = br.readLine();
        System.out.print("Kota Asal : ");
        kota = br.readLine();
    }
}
```

b) Inputan dengan memanfaatkan class **JOptionPane**.

Untuk bisa menggunakan class `JOptionPane`, maka perlu mengimpor dari package **javax.swing.*** dan gunakan method **showInputDialog()** yang terdapat pada class `JOptionPane`. Contoh :

```
import javax.swing.*;
class CobaInput2
{
    public static void main (String [] args)
    {
        String nama, kota;

        nama = JOptionPane.showInputDialog("Nama Anda :");
        kota = JOptionPane.showInputDialog("Kota Asal :");

        System.out.println("Selamat Datang "+ nama +" dari "+ kota);
        System.exit(0);
    }
}
```

Catatan :

Semua data yang diinputkan dianggap sebagai suatu nilai `String` meskipun data tersebut hanya terdiri atas angka saja. Untuk menampung data yang diinputkan ke dalam variabel bertipe numerik (misal : `int`, `long`, `double`), maka data harus terlebih dahulu diubah ke tipe data numerik.

Contoh :

```
String sAngka;  
int a = Integer.parseInt(sAngka);  
long b = Long.parseLong(sAngka);  
double c = Double.parseDouble(sAngka);
```

Silahkan coba source code dibawah ini:

Kendaraan.java

```
1  public class Kendaraan {  
2      private String merk, warna;  
3      protected String namaClass = "Kendaraan";  
4  
5      protected void setMerk(String merk) {  
6          this.merk = merk;  
7          merk = null; // menghapus variable parameter dari memory  
8      }  
9  
10     protected String getMerk() {  
11         return merk;  
12     }  
13  
14     protected void setWarna(String warna) {  
15         this.warna = warna;  
16         warna = null; // menghapus variable parameter dari memory  
17     }  
18  
19     protected String getWarna() {  
20         return warna;  
21     }  
22  
23     protected void tampil(String a)  
24     { System.out.println(a);  
25         a = null;  
26     }  
27  
28     protected String keterangan()  
29     { return ("Ini adalah class "+namaClass);  
30     }  
31  
32     protected void hapus()  
33     { // menghapus variable private dari memory  
34         warna = null;  
35         merk = null;  
36         namaClass = null;  
37     }  
38 }
```

Mobil.java

```
1 public class Mobil extends Kendaraan {
2     private long harga;
3     protected String namaClass = "Mobil";
4
5     protected void setHarga(String harga) {
6         this.harga = Long.parseLong(harga);
7         harga = null;
8     }
9
10    protected long getHarga() {
11        return harga;
12    }
13
14    protected String keterangan()
15    { return (namaClass+" : Ini adalah class "+namaClass);
16    }
17
18    public String keterangan Kendaraan()
19    { // mengakses atribut/variable & method parent (class Kendaraan)
20        return super.namaClass+" : "+super.keterangan();
21    }
22
23    protected void hapus()
24    { // menghapus variable private dari memory
25        harga = 0;
26        // menghapus variable private parent (class Kendaraan)
27        super.hapus();
28    }
29 }
```

MainMobil.java

```
1 import javax.swing.JOptionPane;
2
3 public class MainMobil extends Mobil {
4     public static void main (String []args) {
5         MainMobil ob = new MainMobil();
6         String merk, warna, harga ;
7
8         merk = JOptionPane.showInputDialog("Merk Mobil :");
9         ob.setMerk(merk);
10        ob.tampil("Merk Mobil : "+ob.getMerk());
11
12        warna = JOptionPane.showInputDialog("Warna Mobil :");
13        ob.setWarna(warna);
14        ob.tampil("Warna Mobil : "+ob.getWarna());
15
16        harga = JOptionPane.showInputDialog("Harga Mobil :");
17        ob.setHarga(harga);
18        ob.tampil("Harga Mobil : "+ob.getHarga());
19
20        ob.hapus();
21        merk=null;
22        warna=null;
23        harga=null;
24        ob = null;
25
26        System.exit(0);
27    }
28 }
```

MODUL 5

LATIHAN

SOAL

1. Buatlah aplikasi rental VCD sederhana sebagai berikut:

Masukkan data VCD Film

sebanyak X kali Judul : judul
film.

Aktor : nama-nama aktor di dalam film

tersebut. Sutradara : nama sutradara

film tersebut.

Publisher : yang memproduksi film tersebut.

Kategori : SU = Semua Umur, D = Dewasa, R = Remaja, A =

Anak-anak Stok : jumlah stok VCD film tersebut.

Judul, Aktor, Sutradara, Publisher, Kategori dan Stok dilooping sebanyak X kali.

Desainlah aplikasi rental VCD tersebut dengan konsep inheritance dan tentukan parent class serta child class-nya. Setelah itu, implementasikan class-class yang telah didesain dengan membuat program sederhana yang memiliki fasilitas entri data VCD film dan melihat daftar VCD film yang telah dientrikan.