

Modul Praktikum III PBO

INHERITANCE

A. TUJUAN

1. Memahami konsep inheritance (pewarisan)
2. Mendefinisikan superclasses dan subclasses
3. Override method dari superclass
4. Menerapkan keyword this

B. DASAR TEORI

1. Pewarisan

Pewarisan merupakan konsep dalam pemrograman berorientasi objek yang memungkinkan untuk membuat suatu kelas yang didasarkan pada kelas yang sudah ada sehingga mewarisi semua metode dan variabelnya.

2. Keyword super

Keyword **super** digunakan oleh subclass untuk memanggil konstuktur yang berada pada superclass.

3. Keyword this

Kata kunci **this** dipergunakan pada pembuatan kelas dan digunakan untuk menyatakan objek sekarang

4. Overriding

Overriding menulis kembali method sama persis. Sama mulai dari nama method dan isinya dan mengimplementasi kembali di subclassnya. Overriding dipakai saat kita menggunakan method yang sama tapi berbeda implementasinya. Jadi overriding method mempunyai nama method yang sama, jumlah parameter dan tipe parameter serta nilai kembalian (return) method yang di override.

C. LATIHAN

Latihan 1

- a. Tuliskan ketiga class berikut dalam 3 file yang berbeda

```
class Pertama
{
    private int a = 10;

    protected void terprotek ()
    {
        System.out.println ("Method ini hanya untuk anaknya");
    }

    public void info ()
    {
        System.out.println ("a = "+a);
        System.out.println ("Dipanggil pada = "+this.getClass ().getName());
    }
}
```

```
class Kedua extends Pertama
{
    private int b = 8;

    protected void BacaSuper ()
    {
        System.out.println ("Nilai b :"+b);
        terprotek ();
        info ();
    }
}
```

```
class TestPertamaKedua
{
    public static void main (String [] args)
    {
        Kedua D2 = new Kedua ();
        D2.BacaSuper ();
        D2.info ();

        Pertama S1 = new Pertama ();
        S1.terprotek ();
        S1.info ();
    }
}
```

- b. Kompilasi ketiga class tersebut secara berurutan, hasil kompilasi adalah :

- c. Hilangkan baris S1.terprotek() dari class TestPertamaKedua
- d. Kompilasi ketiga class tersebut secara berurutan, Hasil kompilasi adalah :

- e. Hapus semua file yang berektensi .class dari folder
- f. Kompilasi class TestPertamaKedua saja. Perhatikan file apa yang dihasilkan dari kompilasi tersebut :

- g. Jelaskan fungsi dari perintah this.getClass.getName();

Latihan 2

Pada latihan 2 ini kita akan menggunakan keyword super untuk memanggil konstuktur dari kelas induk.

- a. Tuliskan ketiga class berikut dalam 3 file yang berbeda

```
class Person
{
    protected String name;
    protected int age;

    public Person(String name, int age)
    {
        this.name = name;
        this.age = age;
    }

    //metode
    public void info()
    {
        System.out.println ("Nama : "+this.name);
        System.out.println ("Usia : "+this.age);
    }
    //akhir kelas program
}
```

```

class Employ extends Person
{
    private String noKaryawan;

    //konstruktor
    public Employ(String noKaryawan, String name, int age)
    {
        super (name, age);
        this.noKaryawan = noKaryawan;
    }

    //metode
    public void info ()
    {
        System.out.println ("No.karyawan :"+this.noKaryawan);
        super.info ();
    }
}

public class KonstruktorSuperKelas
{
    public static void main (String [] args)
    {
        Employ programmer1 = new Employ ("12345678", "Yanto", 32);
        programmer1.info();
    }
}

```

- b. Jalankan program, keluaran dari program diatas adalah :

Latihan 3

- a. Tuliskan program berikut :

```

public class Hewan
{
    public static void testClassMethod()
    {
        System.out.println ("The Class Method in Hewan...");
    }

    public void testInstanceMethod()
    {
        System.out.println ("The Instance Method in Hewan...");
    }
}

//meng-overwrite method pada class Hewan

public class Gajah extends Hewan
{
    public static void testClassMethod ()
    {
        System.out.println ("The class method in Hewan...");
    }

    //mengoverride method pada class Hewan
    public void testInstancemethod ()
    {
        System.out.println ("The Instance method in Gajah...");
    }

    public static void main (String [] args)
    {
        Kucing myKucing = new Kucing ();
        Hewan myHewan = myKucing;
        Hewan.testClassMethod ();
        myHewan.testInstanceMethod();
    }
}

```

b. Output program tersebut adalah :

Latihan 4

a. Tulis dan simpan code program berikut (perhatikan keyword super pada program)

```
class A
```

```
{
```

```
    private int a;
```

```
    public void setA (int nilai)
```

```
    {
```

```
        a = nilai;
```

```
    }
```

```
    public int getA()
```

```
    {
```

```
        return a;
```

```
    }
```

```
    public void tampilkanNilai()
```

```
    {
```

```
        System.out.println ("Nilai a : "+getA());
```

```
    }
```

```
}
```

```
class B extends A
```

```
{
```

```
    private int b;
```

```
    public void setB (int nilai)
```

```
    {
```

```
        b = nilai;
```

```
    }
```

```
    public int getB ()
```

```
    {
```

```
        return b;
```

```
    }
```

```
    //melakukan override terhadap method tampilkanNilai ()
```

```
    //yang terdapat pada kelas A
```

```
    public void tampilkanNilai ()
```

```
    {
```

```
        super.tampilkanNilai (); //memanggil method dalam kelas A
```

```
        System.out.println ("Nilai b : "+getB());
```

```
    }
```

```
}
```

```
class DemoOverride2
{
    public static void main (String [] args)
    {
        B obj = new B ();
        obj.setA(50);
        obj.setB(150);
        //akan memanggil method yang terdapat pada kelas B
        obj.tampilkanNilai ();
    }
}
```

- b. Keluaran dari program tersebut adalah :

Latihan 5

- a. Tulis dan simpan code program berikut :

```
class Truck{
    //the Truck class has three fields
    public int cadence;
    public int gear;
    public int speed;

    //the Truck class has one constructor
    public void Truck(int startCadence, int startSpeed, int startGear){
        gear = startGear;
        cadence = startCadence;
        speed = startSpeed;
    }

    //the Truck class has four methods
    public void setCadence(int newValue){
        cadence = newValue;
    }

    public void setGear(int newValue){
        gear = newValue;
    }

    public void applyBrake(int decrement){
        speed -= decrement;
    }

    public void speedUp(int increment){
        speed += increment;
    }
}
```



```

public class MountainTruck extends Truck{
    //the MountainTruck subclass adds one field
    public int seatHeight;

    //the MountainTruck subclass has one constructor
    public void MountainTruck(int startHeight, int startCadence,
        int startSpeed, int startGear){
        super.Truck(startHeight, startCadence, startGear);
        seatHeight = startHeight;
    }

    //the MountainTruck subclass adds one method
    public void setHeight(int newValue){
        seatHeight = newValue;
    }

    public static void main(String[] abc){
        System.out.println("Mulai Berjalan");
        MountainTruck MB = new MountainTruck();
        MB.MountainTruck(0,0,0,0);
        System.out.println("gear = "+gear);
        System.out.println("speed = "+speed);
        MB.setGear(2);
        MB.speedUp(2);
        System.out.println("gear = "+gear);
        System.out.println("speed = "+speed);
        MB.speedUp(2);
        System.out.println("gear = "+gear);
        System.out.println("speed = "+speed);
    }
}

```

b. Keluaran dari program tersebut adalah :

c. Modifikasi class MountainBike sedemikian sehingga mensimulasikan proses perjalanan sepeda dari posisi diam sampai berjalan normal, kemudian diikuti dengan berjalan di tanjakan jalan, dilanjutkan lagi menuruni tanjakan dan diakhiri dengan berhenti. Code yang dihasilkan :

d. Output dari point C adalah :



Tugas Praktikum

Lengkapi program dibawah ini dengan membuat satu method pada kelas turunan yang meng-override method pada kelas induk sehingga anda dapat membedakan antara overload method dengan override method! Kemudian buat satu kelas lagi untuk menguji coba fungsionalitas kedua method tersebut!

```
class Induk{
    // mendefinisikan method test() tanpa parameter
    public void test(){
        System.out.println("Method di dalam kelas Induk");
    }
}

class Turunan extends Induk{
    // melakukan overload terhadap method test(), bukan override .
    public void test(String s){
        System.out.println("Method di dalam kelas Turunan");
        System.out.println("s : \"\" + s + \"\"");
    }
}
```

Tugas Rumah

Buatlah sebuah program dengan menerapkan “inheritance” dalam kehidupan perkuliahan!

Contoh :

Induk : Fakultas

Anak : Jurusan