LAPORAN RESMI

PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK (PBO)

MODUL 3

"POLYMORPHISM"

Disusun Oleh :

TGL PRAKTIKUM	:	5 November 2012
NAMA	:	ACHMAD FAJAR NORULLAH
NRP	:	11.04.111.000.75
KELAS/KELOMPOK	:	B/3
DOSEN PENGAMPU	:	ERY SETIYAWAN JULLEV, S.KOM.
ASISTEN	:	SHOHIB

Disetujui : ...../...../...../Bangkalan
(SHOHIB) 10.04.111.00101

LABORATORIUM COMMON COMPUTING
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO MADURA
2012 / 2013

BAB I

PENDAHULUAN

1.1 Latar Belakang

Polymorphism merupakan metode pewarisan sifat yang mana dalam anak kelas terdapat method yang mengoverride method dalam induk kelas.

1.2 Tujuan

Tujuan mahasiswa mengikuti praktikum PBO Modul 2 ini adalah :

1. Mahasiswa dapat memahami tentang penggunaan dan bentuk - bentuk polymorphism.
2. Mahasiswa dapat menggunakan polymorphism untuk menyelesaikan suatu permasalahan.
3. Menghemat coding dengan penggunaan polymorphism.

BAB II

DASAR TEORI

2.1 Polymorphism

Metode disebut polimorfis jika aksi yang dilakukan oleh suatu metode berbeda-beda tergantung pada objek aktual pada saat metode itu dijalankan. Polimorfisme adalah salah satu fitur utama dalam pemrograman berorientasi objek.

BAB III

TUGAS PENDAHULUAN

3.1 Prepraktikum

1. Keuntungan apa yang bisa didapat dengan Polymorphism?

2. Bagaimana hubungan antar teknik overloading dengan Polymorphism?

3.2 Jawaban Prepraktikum

1. Dengan adanya sifat pewarisan polimorfis itu berarti kita bisa menghemat koding.
2. Overloading sangat lekat dengan polimorfis karena di dalam polimorfis harus terdapat overloading yaitu keadaan dimana memiliki method dengan nama yang sama namun parameter berbeda.

BAB IV

IMPLEMENTASI

4.1 Kegiatan Praktikum

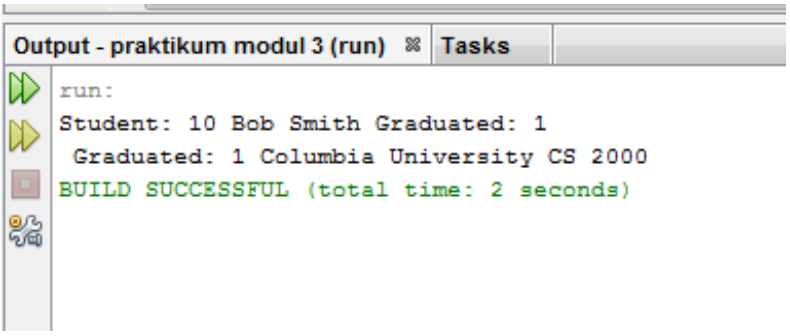
Pada gambar 1 kita membuat tiga object class dengan nama Student, subclass StudentGrad dan class StudentInfo. Class Student mempunyai method Write yang digunakan untuk melewatkan argumen untuk kemudian ditampilkan dengan menggunakan method Display.

```
class Student {
public void Write(int ID, int Grad, String Fname, String Lname) {
m_ID = ID;
m_Graduation = Grad;
m_First = Fname;
m_Last = Lname;
}
public void Display(){
System.out.println( "Student: " + m_ID + " " + m_First + " " + m_Last + "
Graduated: " + m_Graduation);
}
private int m_ID, m_Graduation;
private String m_First;
private String m_Last;
}

public class StudentInfo  {
   public static void main(String[] args) {
   StudentGrad myStudent = new StudentGrad();
   myStudent.Write(10, 1,"Bob","Smith", 2000,"Columbia University","CS");
   myStudent.Display();
}
}
```

```
class StudentGrad extends Student {
public void Write(int ID, int Grad, String Fname, String Lname, int yrGrad, String
unSch, String major) {
super.Write(ID, Grad, Fname, Lname);
m_UndergradSchool = unSch;
m_Major = major;
m_Grad = Grad;
YearGraduated = yrGrad;
}
   @Override
public void Display(){
super.Display();
System.out.println(" Graduated: " + m_Grad + " " + m_UndergradSchool + " " +
m_Major + " " + YearGraduated);
}
private Integer YearGraduated,m_Grad;
private String m_UndergradSchool;
private String m_Major;
}
```

4.2 Jawaban Kegiatan Praktikum

Hasil Running :

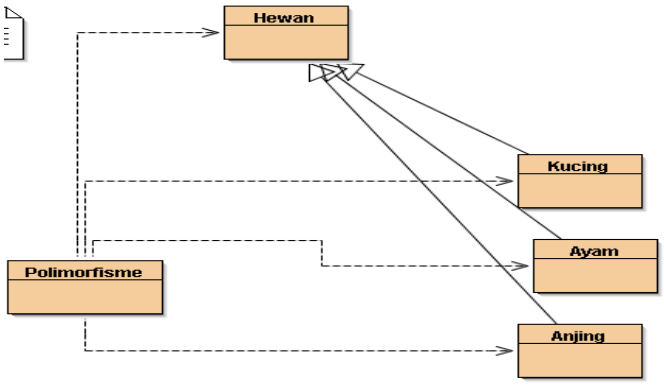


BAB V

TUGAS AKHIR

5.1 Tugas Akhir

Kucing, Ayam, dan Anjing merupakan makhluk hidup yang masuk kategori hewan, tiap hewan pasti mempunyai identitas baik itu berupa tipe hewan dan suaranya.



Gambar 2 : Class Hewan dan subclass nya

Potongan sintaks bisa dituliskan seperti di bawah ini:

```
class Hewan {
        private String tipe;
        public Hewan(String tipe){
        this.tipe = tipe;
        }
        public String toString(){
        return "Ini adalah "+tipe;
        }
        public void bersuara(){
        }
}
```

Sintaks class polimorfisme

```
import java.util.Random;
class Polimorfisme{
        public static void main(String args[]){
        Hewan[] hewannya = {
        new Anjing("Broni", "Herder"),
        new Kucing("Kiti", "Persia"), new Ayam("Toki","Buras")
        };
        Hewan pilihHewan;
        Random pilih = new Random();
        for (int i = 0; i<5 ; i++){
```

```
pilihHewan=hewannya[pilih.nextInt(hewannya.length)];
System.out.println("\nPilihan Anda:\n" +pilihHewan);
pilihHewan.bersuara();
}
}
}
```

Sintaks class Anjing

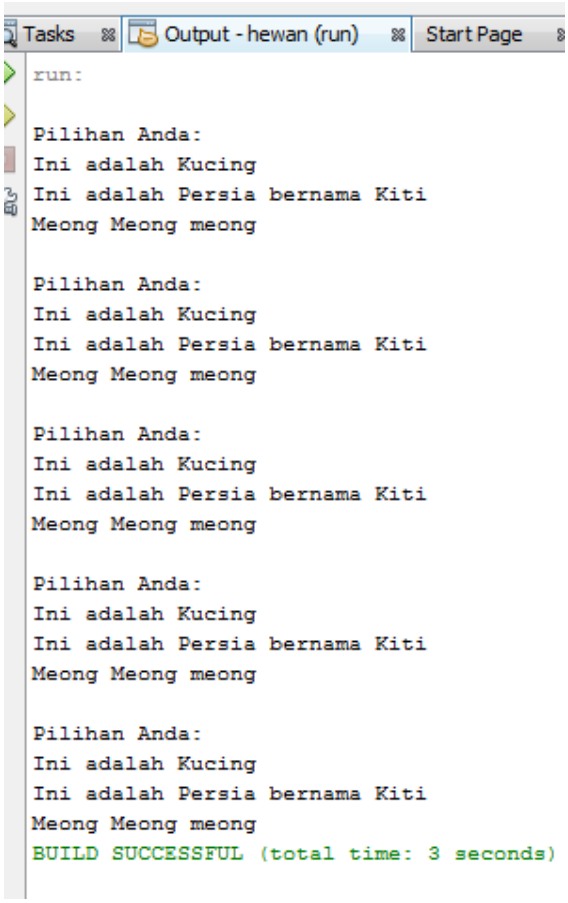
```
class Anjing extends Hewan{

//sintaks private object
public Anjing(String nama){
//sintaks menampilkan nama
}
public Anjing(String nama, String jenis){
//sintaks menampilkan nama dan jenis hewan
}
public void bersuara(){
//sintaks menampilkan suara hewan
}
ublic String toString(){
return super.toString()+"/nIni adalah "+jenis+" bernama
"+nama;
}
}
```

Jawab :

```
class Hewan {
private String tipe;
public Hewan(String tipe){
this.tipe = tipe;
}
   @Override
public String toString(){
return "Ini adalah "+tipe;
}
public void bersuara(){
}
}

class Polimorfisme{
public static void main(String args[]){
Hewan[] hewannya = {
new anjing("Broni", "Herder"),
new kucing("Kiti", "Persia"),
new ayam("Toki","Buras")
};
Hewan pilihHewan;
Random pilih = new Random();
```

```
        for (int i = 0; i<5 ; i++){
        pilihHewan=hewannya[pilih.nextInt(hewannya.length)];
System.out.println("\nPilihan Anda:\n" +pilihHewan); pilihHewan.bersuara();
        }
        }

        }
        class anjing extends Hewan {
            private String nama;
            private String jenis;

            public anjing(String nama, String jenis){
                super("anjing");
                this.nama=nama;
                this.jenis=jenis;
            }

            @Override
            public void bersuara(){
                System.out.println("Guk Guk Guk");
            }

            @Override
            public String toString(){
                return super.toString()+"\nIni adalah "+jenis+" bernama "+nama;
                    }
        }
        class ayam extends Hewan {
            private String nama;
            private String jenis;

            public ayam(String nama, String jenis){
                super("ayam");
                this.nama=nama;
                this.jenis=jenis;
            }

            @Override
            public void bersuara(){
                System.out.println("Kukuruyuk Petok Petok Petok");
            }

            @Override
            public String toString(){
                return super.toString()+"\nIni adalah "+jenis+" bernama "+nama;
                    }
        }
        class kucing extends Hewan {
            private String nama;
            private String jenis;
```

```
    public kucing(String nama, String jenis){
        super("Kucing");
        this.nama=nama;
        this.jenis=jenis;
    }

    @Override
    public void bersuara(){
        System.out.println("Meong Meong meong");
    }

    @Override
    public String toString(){
        return super.toString()+"\nIni adalah "+jenis+" bernama "+nama;
            }
}
```



5.2 Tugas Tambahan

1. Apa yang dimaksud dengan methode overloading, accessor, dan mutator pada java?

2. Jelaskan Perbedaan antara overloading dengan overriding?
3. Buatlah program sederhana dari overloading, overriding, accessor, dan mutator (dalam satu program polymorphisme) 1 superclass dan 1 subclass !
4. Tambahkan 1 methode pada program soal no 3 yang tidak bisa di overriding di subclass !

Jawab :

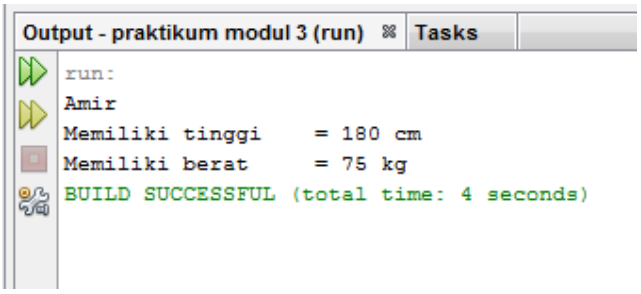
1. Overloading : adalah keadaan dimana pada kelas anak terdapat method dengan nama yang persis sama dengan kelas induk namun memiliki parameter yang berbeda..

   Accessor : method yang digunakan untuk membaca nilai variabel pada class. Biasanya memiliki penulisan seperti get.

   Mutator : method yang digunakan untuk memberi atau mengubah data pada variabel dan biasanya menggunakan penulisan set.
2. Perbedaanya adalah terletak pada return value dan penulisannya, pada overriding penulisannya persis sama dan memiliki return value yang berbeda sedangkan overloading penulisan methodnya sama namun memiliki parameter yang berbeda sehingga bisa memiliki keluaran yang sama ataupun bisa juga berbeda.
3.

```
class orangnya {
String tinggi;
String berat;

   public void settinggi(){
      this.tinggi = "160 cm";
   }
   public void setberat(){
      this.berat = "50 kg";
   }

   public String gettinggi(){
      return tinggi;
   }
   public String getberat(){
      return berat;
   }
}

class amir extends orangnya{
```

```
    @Override
    public void settinggi(){
        this.tinggi = "180 cm";
    }
    @Override
    public void setberat(){
        this.berat = "75 kg";
    }

    @Override
    public String gettinggi(){
        return tinggi;
    }
    @Override
    public String getberat(){
        return berat;
    }
}

public class orang{
public static void main(String[] args) {
        amir tampil = new amir();
        tampil.settinggi();
        tampil.setberat();
        System.out.println("Amir");
        System.out.println("Memiliki    tinggi          =   "   +
tampil.gettinggi());
        System.out.println("Memiliki    berat               =   "
tampil.getberat());



    }
}
```



4. Ditambahkan method

```
public final void besar(){
System.out.println("gak bisa di override");
```

}

Karena method besar merupakan final method yang tidak bisa di override lagi

```
44      @Override
45      public String gettinggi() {
46
48
49
50
52      void besar() {
53      }
53  }
54
```

besar() in Tugastambahan.amir cannot override besar() in Tugastambahan.orangnya
overridden method is final

Add @Override Annotation

(Alt-Enter shows hints)

BAB VI
PENUTUP

6.1 Kesimpulan

Polimorfis memiliki kesamaan dengan inheritance yaitu pewarisan sifat namun memiliki perbedaan yaitu terletak pada overriding methodnya.

6.2 Saran

- Diharap asisten untuk menjelaskan mengenai tugas praktikum dan kejelasan tentang tugas akhir supaya jelas dan dapat dimengerti.