PRATIKUM POLIMORFISME

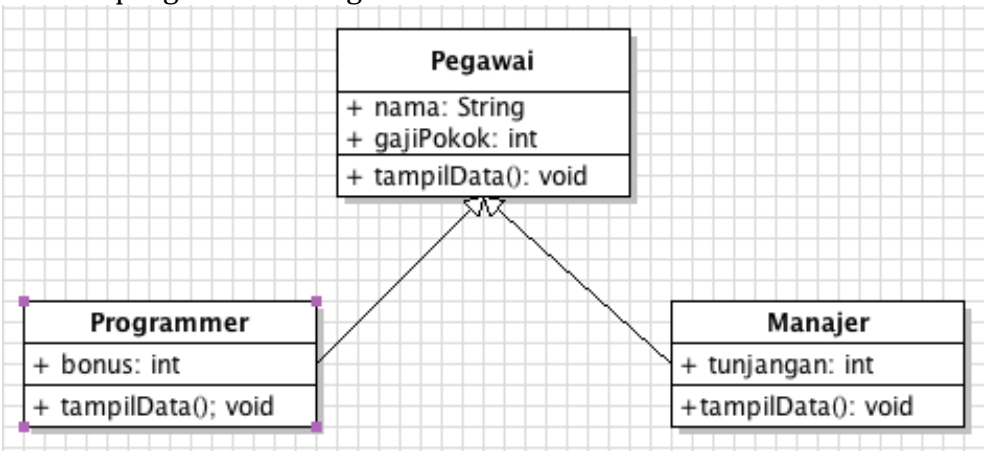
PERCOBAAN 1: KONSEP DASAR POLIMORFISME

Tujuan:

1. Memahami konsep polimorfisme
2. Bisa membuat program di Java yang menerapkan konsep polimorfisme.

Prosedur Percobaan:

1. Buatlah program dari diagram klas berikut ini:



Keterangan:

- Method tampilData() di klas Pegawai di dalamnya berisi perintah untuk menampilkan informasi atau data nama dan gajiPokok pegawai.
- Method tampilData() di klas Programmer meng-override method tampilData() yang diwarisi dari superklas. Isi method tampilData() di klas Programmer adalah pemanggilan method tampilData() milik superklas nya dan menampilkan data bonus.
- Method tampilData() di klas Manajer serupa dengan method tampilData() di Programmer.

2. Tampilan programnya sebagai berikut:

Klas Pegawai

```
10   */
11  public class Pegawai {
12      public String nama;
13      public int gajiPokok;
14
15      public void tampilData(){
16          System.out.println("NAMA = "+nama);
17          System.out.println("GAJI POKOK = "+gajiPokok);
18      }
19  }
20
```

Klas Programmer

```
10   */
11  public class Programmer extends Pegawai{
12      public int bonus;
13
14      @Override
15      public void tampilData(){
16          super.tampilData();
17          System.out.println("BONUS = "+bonus);
18      }
19  }
20
```

Klas Manajer

```
10   */
11  public class Manajer extends Pegawai{
12      public int tunjangan;
13
14      @Override
15      public void tampilData(){
16          super.tampilData();
17          System.out.println("Tunjangan = "+tunjangan);
18      }
19  }
20
```

3. Buat Main class

```
10   */
11  public class Main {
12      public static void main(String[] argv){
13          Pegawai p = new Pegawai();
14          p = new Programmer(); //polimorfisme
15          p = new Manajer(); //polimorfisme
16
17          Manajer m = new Programmer();//tidak bisa karena programmer
18                                       //bukan subklas manajer
19
20      }
21
22  }
23
```

4. Jawab pertanyaan di bawah ini:
   a. Perhatikan pada baris 13, 14, 15. Pada baris 13, dideklarasikan objek p dari klas Pegawai dan diinstansiasi dari klas Pegawai. Tetapi pada baris 14 dan 15 objek P diinstansiasi dari klas Programmer dan Manajer. Terlihat tidak error, mengapa bisa demikian?

      ……
      ……
      ……
      ……
      ……

   b. Pada baris 15, mengapa terjadi error? Apa yang harus dilakukan agar tidak terjadi error?

      ……
      ……
      ……
      ……
      ……

   c. Dari percobaan, buat kesimpulan tentang apa itu polimorfisme, pada klas-klas yang bagaimana polimorfisme bisa diterapkan, ciri/format polimorfisme.

      ……
      ……
      ……
      ……
      ……
      ……
      ……

PERCOBAAN 2: IMPLEMENTASI POLIMORFISME

Tujuan:

Memahami contoh-contoh implementasi polimorfisme, diantaranya:

a. Virual Method Invocation – VMI
b. Heterogeneous Collection
c. Polimorphic Argument
d. InstanceOf
e. Casting Object

Prosedur Percobaan:

1. Lihat klas Main yang telah dibuat pada Percobaan 1 hingga seperti di bawah ini:

```
10    */
11   public class Main {
12       public static void main(String[] argv){
             Pegawai p = new Pegawai();
             p = new Programmer(); //polimorfisme
             p = new Manajer(); //polimorfisme
16
17           //Manajer m = new Programmer();//tidak bisa karena programmer
18                                         //bukan subklas manajer
19           Manajer m = new Manajer();
20           m.gajiPokok = 10000;
21           m.nama = "Sai";
22           m.tunjangan = 5000;
23           p = m;//Polimorfisme
24           p.tampilData();//VIRTUAL METHOD INVOCATION-VMI
25
26           m.tampilData();//Bukan VMI, krm m bukan objek polimorfisme
27
28       }
29
30   }
```

2. Jalankan klas Main, dan akan tampil hasil sbb:

```
run:
NAMA = Sai
GAJI POKOK = 10000
Tunjangan = 5000
NAMA = Sai
GAJI POKOK = 10000
Tunjangan = 5000
BUILD SUCCESSFUL (total time: 0 seconds)
```

3. Jawab pertanyaan tentang VMI di bawah ini:
   a. Mengapa hasil output dari running program antara p.tampilData() dan m.tampilData() bisa sama?

      .....
      .....
      .....
      .....
      .....

   b. Mengapa pada baris 24 disebut VMI sedangkan pada baris 26 bukan VMI?

      .....
      .....
      .....
      .....
      .....

c. Pada saat p.tampilData() pada baris 24, yang dijalankan apakah method tampilData() milik klas Pegawai atau tampilData() milik Manajer?Mengapa demikian?

……
……
……
……
……

d. Ambil kesimpulan terkait dengan pengertian dan proses VMI, serta syarat terjadinya VMI.

……
……
……
……
……

4. Ubah lagi klas Main hingga menjadi seperti di bawah ini

```
10  */
11  public class Main {
12      public static void main(String[] argv){
            Pegawai p = new Pegawai();
            p = new Programmer(); //polimorfisme
            p = new Manajer(); //polimorfisme
16
17          //Manajer m = new Programmer();//tidak bisa karena programmer
18                                         //bukan subklas manajer
19          Manajer m = new Manajer();
20          m.gajiPokok = 10000;
21          m.nama = "Sai";
22          m.tunjangan = 5000;
23          p = m;//Polimorfisme
24          p.tampilData();//VIRTUAL METHOD INVOCATION-VMI
25
26          m.tampilData();//Bukan VMI, krm m bukan objek polimorfisme
27
28          Pegawai[] arrPeg = new Pegawai[3];
29          Programmer pr = new Programmer();
30          pr.nama = "uchiha";pr.gajiPokok = 100000; pr.bonus = 120000;
31          arrPeg[0] = p;
32          arrPeg[1] = m;
33          arrPeg[2] = new Programmer();
34          arrPeg[0].tampilData();
35          arrPeg[1].tampilData();
36      }
37
38  }
```

5. Jalankan klas Main dan amati perubahan hasil outputnya.
6. Jawab pertanyaan di bawah ini:
   a. Pada baris berapa terjadi Heterogeneous Collection?

   ……
   ……

   b. Mengapa pada baris tersebut disebut sebagai Heterogeneous Collection?

   ……
   ……
   ……
   ……

   c. Mengapa array arrPeg bisa diisi baik dengan objek dari klas Programmer maupun objek dari klas Manajer?

   ……

.....
.....
.....

d. Ambil kesimpulan tentang Heterogeneous Collection.

.....
.....
.....
.....

7. Ubah kembali klas Main sbb:

```
10   */
11  public class Main {
12      public static void main(String[] argv){
13          Pegawai p = new Pegawai();
14          p = new Programmer(); //polimorfisme
15          p = new Manajer(); //polimorfisme
16
17          //Manajer m = new Programmer();//tidak bisa karena programmer
18                              //bukan subklas manajer
19          Manajer m = new Manajer();
20          m.gajiPokok = 10000;
21          m.nama = "Sai";
22          m.tunjangan = 5000;
23          p = m;//Polimorfisme
24          p.tampilData();//VIRTUAL METHOD INVOCATION-VMI
25
26          m.tampilData();//Bukan VMI, krm m bukan objek polimorfisme
27
28          Pegawai[] arrPeg = new Pegawai[3];
29          Programmer pr = new Programmer();
30          pr.nama = "uchiha";pr.gajiPokok = 100000; pr.bonus = 120000;
31          arrPeg[0] = p;
32          arrPeg[1] = m;
33          arrPeg[2] = new Programmer();
34          arrPeg[0].tampilData();
35          arrPeg[1].tampilData();
36
37          hitungGajiTotal(m);
38          hitungGajiTotal(pr);
39      }
40
41      public static void hitungGajiTotal(Pegawai p){
42
43          if(p instanceof Programmer){
44              Programmer prog = (Programmer) p;
45              int gajiTot = prog.bonus+prog.gajiPokok;
46              System.out.println("Gaji Total Programmer = "+gajiTot);
47          }else if(p instanceof Manajer){
48              Manajer man = (Manajer) p;
49              int gajiTot = man.tunjangan+man.gajiPokok;
50              System.out.println("Gaji Total Manajer = "+gajiTot);
51          }
52      }
53
54  }
```

8. Jalankan klas Main dan amati perubahan hasil outputnya
9. Jawab pertanyaan di bawah ini:
    a. Mengapa fungsi hitungGajiTotal() pada baris 41, bisa menerima argument berupa objek dari klas Programmer dan Manajer (contoh pemanggilan fungsi pada baris 37 dan 38, pada baris 37 dilewatkan argument **m** yang merupakan objek dari Manajer dan pada baris 38 dilewatkan argument objek **pr** yang merupakan objek dari Programmer) ?

.....
.....
.....

……
……
……
……
……

b. Pada baris 42 coba tambahi sintaks untuk mengakses atribut bonus atau tunjangan, seperti **p.bonus** atau **p.tunjangan**. Bisa apa tidak? Mengapa demikian?

……
……
……
……

c. Pada baris 43 dan 47 diberikan operator **instanceof**, untuk apakah operator **instanceof**?

……
……
……
……

d. Pada baris 44 dan 48 dilakukan casting objek atau merubah objek. Pada baris 44 merubah objek **p** agar menjadi objek dari klas **Programmer**, sedangkan pada baris 48 merubah objek **p** agar menjadi objek dari klas **Manajer**. Untuk apa perlu dilakukan casting objek?

……
……
……
……

e. Ambil kesimpulan tentang Polimorphic argument, instance of dan casting objek.

……
……
……
……
……
……
……
……