

Praktikum 6

Polimorfisme

1. Tujuan

Mahasiswa mampu menerapkan konsep polimorfisme dan implementasinya ke dalam program.

2. Landasan Teori

Polimorfisme merupakan kemampuan entitas (class, object, method, parameter) untuk memiliki wujud/perilaku yang berbeda dalam konteks yang berbeda. Polimorfisme dapat dikelompokkan menjadi :

1. Polimorfisme Ad Hoc
 - a. Overloading Polymorphism
 - b. Coercion Polymorphism
2. Polimorfisme Universal
 - a. Inclusion Polymorphism
 - b. Parametric Polymorphism (tidak dibahas disini)

3. Langkah Praktikum

- a. { 15 menit } Polimorfisme Ad Hoc: Overloading

Pada prinsipnya Polimorfisme jenis ini memanfaatkan kemampuan method overloading (yang telah dibahas pada topik awal). Silahkan diskusikan kembali **contoh dan syarat-syarat** yang harus dipenuhi untuk mendapatkan method overloading.

- b. { 15 menit } Polimorfisme Ad Hoc: Coercion

No	Program	System.out.println(output);	Apakah Berhasil Dieksekusi Atau Gagal? Catatlah Penjelasan Terkait Pemahaman tentang Casting dan Coercion
1	int output = 'a';		
2	double x = 15.5; int output = x;		
3	int y = 25; double output = y;		
4	int z = 78; char output = (char) z;		
5	char a = 'a'; double output = a;		

c. { 20 menit } Polimorfisme Universal: Inclusion

Perhatikan contoh berikut ini:

Super Class	
<pre>public class Vehicle { void calRent(int distance, float price){ float fare=distance* price; System.out.println("vehicle price = "+fare); } }</pre>	
Sub Class	
<pre>public class Car extends Vehicle{ void calRent(int jarak, float harga){ float fare=jarak*harga; fare=fare-100.00f; System.out.println("harga sewa mobil = "+fare); } }</pre>	<pre>public class Bus extends Vehicle { }</pre>
Main Class	
<pre>public class Sewa { public static void main(String[] args) { Vehicle kendaraan = new Vehicle(); Vehicle mobil = new Car(); Vehicle bis = new Bus(); kendaraan.calRent(50, 1000); mobil.calRent(50, 1000); bis.calRent(50, 1000); } }</pre>	

- Eksekusilah kode tersebut, perhatikan bahwa ketiga objek yang dibuat diarahkan sebagai tipe Vehicle.
- Anggap jika kita mempunyai method **hitungSewa(Vehicle v)**, analisislah keuntungan penggunaan teknik polimorfisme inclusion.

- d. { 20 menit } Optional jika ada waktu

Perhatikan kode berikut ini:

```
1 public class Person
2 {
3     private String name;
4
5     public Person(String name) { this.name = name; }
6     public boolean isAsleep(int hr) { return 22 < hr || 7 > hr; }
7     public String toString() { return name; }
8
9     public void status( int hr )
10    {
11        if ( this.isAsleep( hr ) )
12            System.out.println( "Now offline: " + this );
13        else
14            System.out.println( "Now online: " + this );
15    }
16 }
```

```
1 public class Student extends Person
2 {
3     public Student(String name) {
4         super(name);
5     }
6
7     public boolean isAsleep( int hr ) // override
8     { return 2 < hr && 8 > hr; }
9 }
```

Jika dieksekusi main method sebagai berikut:

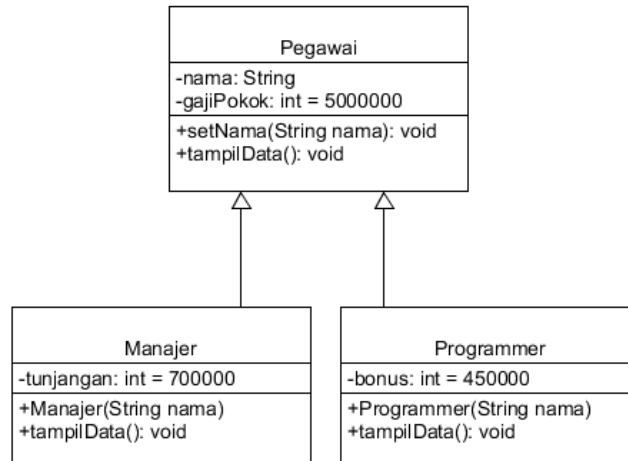
```
1 Person p;
2 p = new Student("Sally");
3 p.status(1);
```

Maka output yang didapat adalah:

.....
Diskusikan bagaimana hasil tersebut didapatkan (hint: apakah "this" pada kelas person dalam method status dieksekusi ketika compile time atau runtime) .

e. { 30 menit } **Post Test**

Implementasikan kelas diagram berikut kedalam bahasa pemrograman java agar diperoleh *main program* dan *output* seperti yang dicontohkan.



Main program:

```
import java.util.ArrayList;

public class TestPolimorfisme {

    public static void main(String[] args) {

        Pegawai pegawai = new Programmer("Mira");
        Pegawai pegawai2 = new Manajer("Joko");
        Manajer pegawai3 = new Manajer("Argo");

        ArrayList<Pegawai> emps = new ArrayList<>();
        emps.add(pegawai);
        emps.add(pegawai2);
        emps.add(pegawai3);

        for (Pegawai emp : emps) {
            emp.tampilData();
        }
    }
}
```

Output:

```
Output - PBOpraktikum (run) ×
run:
Nama : Mira, Gaji pokok : 5000000
Bonus : 450000
Nama : Joko, Gaji pokok : 5000000
Tunjangan : 700000
Nama : Argo, Gaji pokok : 5000000
Tunjangan : 700000
```

Catat hasil Post Test pada file doc:

1. *Screenshoot* kode yang dibuat, dan *screenshoot* outputnya
2. Jelaskan manfaat polimorfisme pada kasus ini.
3. Apabila pada main program perlu menambahkan pegawai4 dan pegawai5! Apa permasalahan yang muncul jika diterapkan tanpa polimorfisme (inclusion)?

Kumpulkan Post Test ke asisten dan submit ke ioclass.