

Praktikum 7

Polimorfisme

1. Tujuan

Mahasiswa mampu menerapkan konsep polimorfisme dan implementasinya ke dalam program.

2. Landasan Teori

Polimorfisme adalah konsep yang memungkinkan objek subkelas diakses dalam lebih dari satu bentuk. Akibatnya, objek subclass dapat diinterpretasikan sebagai tipe dari superclass-nya atau sebagai tipenya sendiri.



Gambar2.1. Ilustrasi Polimorfisme

Ada beberapa macam kelompok polimorfisme, adhoc dan universal. Polimorfisme adhoc dibagi menjadi *overloading* dan *coercion*, sedangkan universal dibagi menjadi *parametric* dan *inclusion*. Polimorfisme jenis *overloading* dan *parametric* dibahas pada sesi praktikum tersendiri, sedangkan pada praktikum ini akan disampaikan materi polimorfisme jenis *inclusion*.

3. Langkah Praktikum

a. Contoh polimorfisme coercion

```
public class PolimorfismeCoercion {  
  
    public static int kuadrat(int bilangan) {  
        return bilangan*bilangan;  
    }  
  
    public static void main(String[] args) {  
        //deklarasi objek integer  
        Integer bilangan = 10;  
  
        //objek integer 'dipaksa' untuk diubah  
        //menjadi primitif  
        int hasilKuadrat = kuadrat(bilangan);  
  
        System.out.printf("Hasil kuadrat %d adalah %d",  
            bilangan,  
            hasilKuadrat);  
    }  
}
```

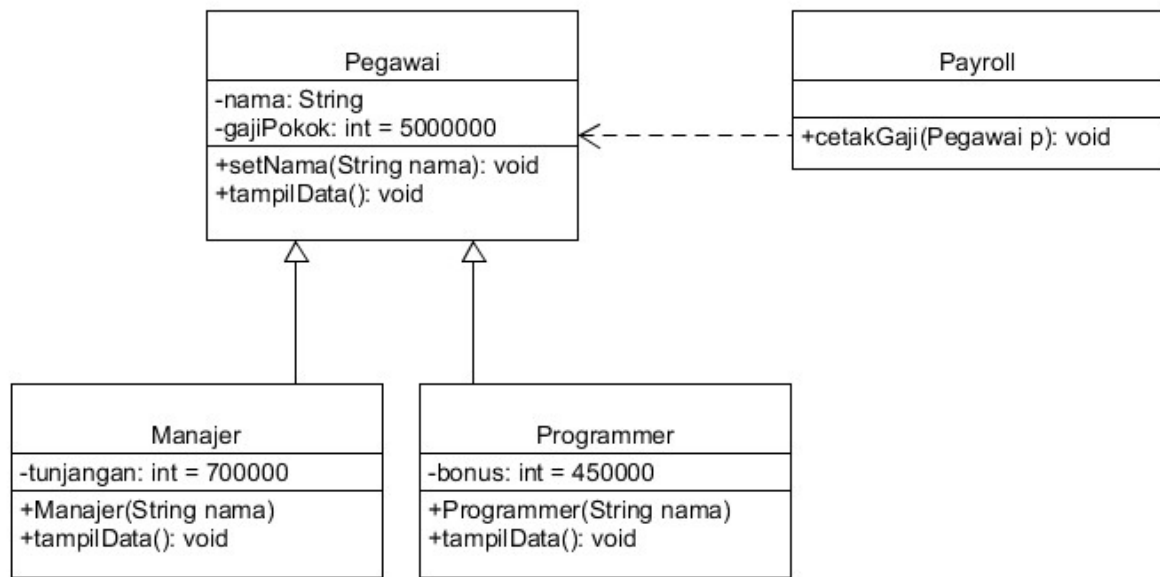
b. Contoh polimorfisme inclusion - 1

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>	

Amati kode di atas, lihat bahwa tipe **Vehicle** bisa digunakan untuk instan **Bus** dan **Car**.

c. Contoh polimorfisme inclusion - 1

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



Main program:

```

public class PegawaiDanGaji{
    public static void main(String args[]) {
        Pegawai pegawai = new Programmer("Mira");
        Pegawai pegawai2 = new Manajer("Joko");
        Pegawai pegawai3 = new Manajer("Argo");

        Payroll payroll = new Payroll();
        payroll.cetakGaji(pegawai);
        payroll.cetakGaji(pegawai2);
        payroll.cetakGaji(pegawai3);
    }
}
  
```

Output:

```

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

Cermati fungsi `cetakGaji` pada kelas `Payroll`, yang menerima parameter objek pegawai dan bisa digunakan untuk `Manajer` maupun `Programmer`.

--Selamat mengerjakan--