

## BAB 7

# POLIMORFISME

### Tujuan

1. Memberikan pemahaman kepada mahasiswa tentang polimorfisme
2. Dapat membedakan perbedaan antara polimorfisme dan inheritance

### Ringkasan Materi

#### A. Polymorfisme

Polimorfisme (memiliki banyak bentuk) menyatakan kemampuan untuk memperlakukan objek-objek dengan cara yang seragam meskipun objek-objek tersebut berbeda perilaku. Polimorfisme adalah suatu sifat yang merupakan efek langsung dari sifat inheritance pada OOP. Jika pada inheritance semua perilaku yang diturunkan kepada subclass memiliki bentuk yang sama, namun pada polimorfisme ini subclass dapat mendefinisikan perilaku yang sama dengan cara yang berbeda, dimana perilaku ini merupakan turunan dari super class.

Untuk bisa mengimplementasikan sifat polimorfisme, kita harus memakai method yang memungkinkan dibuat tanpa didefinisikan. Pada sub classnya nanti kita bisa mendefinisikan isi method tersebut sesuai tujuan masing-masing sub class. Dalam java kita dapat mendefinisikan method semacam ini dengan kata kunci **abstrak**, dengan syarat kelas yang menampung method tersebut juga harus berupa Kelas abstrak. Kelas abstrak harus mengandung satu atau lebih method abstrak tapi boleh ada method selain abstrak. Semua method abstrak yang ada di superclass harus dioverride di subclassnya

Cara pendeklarasian class abstract :

```
public abstract class <nama_kelas> {
    ...
}
```

Cara pendeklarasian method abstract :

```
<modifier> <return_value> abstract <nama_method>;
```

Sebagai contoh, ada kelas MakhlukHidup sebagai kelas induk, yang mempunyai method tertentu misalnya *bernafas()*, *makan()*, *tidur()*, dan *berjalan()*. Kelas MakhlukHidup memiliki kelas turunan Manusia, Sapi, dan Kanguru. Dengan konsep inheritance, semua class turunan MakhlukHidup harus mengimpementasikan semua perilaku yang dimiliki kelas super tersebut. Meskipun subclass (Manusia, Sapi, dan Kanguru) bisa mengimplementasikan semua perilaku yang didefinisikan, namun sub class tersebut mengimplementasikannya dengan cara yang berbeda. Misalnya perilaku *berjalan()*, Manusia, Sapi dan Kanguru berjalan dengan cara yang berbeda. Jika Manusia berjalan dengan dua kaki, Sapi berjalan dengan empat kaki dan Kanguru berjalan dengan cara melompat dengan dua kaki. Implementasi contoh diatas adalah sebagai berikut.

```
public abstract class MakhlukHidup {
    public void bernafas(){
        System.out.println("adalah Makhluk hidup yang dapat bernafas");
    }
    public abstract void berjalan();
}
```

Untuk subclass yang mengimplementasikan kelas MakhlukHidup :

- Manusia

```
public class Manusia extends MakhlukHidup {
    public void berjalan(){
        System.out.println("Manusi berjalan dengan dua kaki");
    }
}
```

```

    }
}
- Sapi
public class Sapi extends MakhlukHidup{
    public void berjalan(){
        System.out.println("Sapi berjalan dengan 4 kaki");
    }
}
- Kanguru
public class Kanguru {
    public void berjalan(){
        System.out.println("Kanguru berjalan dengan melompat pada 2
        kakinya");
    }
}

```

### Pelaksanaan Percobaan

#### Polimorfisme

Ketikkan program di bawah ini. Buatlah class-class berikut dalam package yang sama.

Employee.java	
1	public abstract class Employee {
2	private String name;
3	private String noKTP;
4	public Employee(String name, String noKTP){
5	this.name = name;
6	this.noKTP = noKTP;
7	}
8	public String getName(){
9	return name;
10	}
11	public String getNoKTP(){
12	return noKTP;
13	}
14	public String toString(){
15	return String.format("        "+getName()+"\nNo.        KTP
16	:"+getNoKTP());
17	}
18	public abstract double earnings();//pendapatan
19	}

SalariedEmployee.java	
1	public class SalariedEmployee extends Employee {
2	private double weeklySalary; //gaji/minggu
3	public SalariedEmployee(String name, String noKTP, double
	salary) {
4	super(name, noKTP);
5	setWeeklySalary(salary);
6	}
7	public void setWeeklySalary(double salary) {
8	weeklySalary = salary;
9	}
10	public double getWeeklySalary() {
11	return weeklySalary;

```

12     }
13     public double earnings() {
14         return getWeeklySalary();
15     }
16     public String toString() {
17         return String.format("Salaried employee: " +
18 super.toString() +
19         "\nweekly salary:" + getWeeklySalary());
20     }
21 }

```

#### HourlyEmployee.java

```

1 public class HourlyEmployee extends Employee {
2     private double wage; //upah per jam
3     private double hours; //jumlah jam tiap minggu
4     public HourlyEmployee(String name, String noKTP,
5         double hourlyWage, double hoursWorked) {
6         super(name, noKTP);
7         setWage(hourlyWage);
8         setHours(hoursWorked);
9     }
10    public void setWage(double hourlyWage){
11        wage = hourlyWage;
12    }
13    public double getWage(){
14        return wage;
15    }
16    public void setHours(double hoursWorked){
17        hours = hoursWorked;
18    }
19    public double getHours(){
20        return hours;
21    }
22    public double earnings(){
23        if(getHours() <= 40)
24            return getWage() * getHours();
25        else
26            return 40 * getWage() + (getHours() - 40) *
27            getWage() * 1.5;
28    }
29    public String toString(){
30        return String.format("Hourly employee:
31 "+super.toString()
32        +"\nhourly wage"+getWage()+"\nhours worked:
33 "+getHours());
34    }
35 }

```

#### Commission.java

```

1 public class CommissionEmployee extends Employee {
2     private double grossSales; //penjualan per minggu
3     private double commissionRate; //komisi
4     public CommissionEmployee(String name, String noKTP, double

```

```

5 sales, double rate){
6     super(name, noKTP);
7     setGrossSales(sales);
8     setCommissionRate(rate);
9 }
10 public void setGrossSales(double sales){
11     grossSales = sales;
12 }
13 public double getGrossSales(){
14     return grossSales;
15 }
16 public void setCommissionRate(double rate){
17     commissionRate = rate;
18 }
19 public double getCommissionRate(){
20     return commissionRate;
21 }
22 public double earnings(){
23     return getCommissionRate()*getGrossSales();
24 }
25 public String toString(){
26     return String.format("Commision      employee:
27 "+super.toString()+"\ngross      sales:
28 "+getGrossSales()+"\ncommission rate"+getCommissionRate());
29 }
30 }
31

```

## BasePlusCommissionEmployee.java

```

1 public class BasePlusCommissionEmployee extends
2 CommissionEmployee {
3
4     private double baseSalary;//gaji pokok tiap minggu
5
6     public BasePlusCommissionEmployee(String name, String
7 noKTP, double sales, double rate, double salary) {
8         super(name, noKTP, sales, rate);
9         setBaseSalary(salary);
10    }
11
12    public void setBaseSalary(double salary) {
13        baseSalary = salary;
14    }
15
16    public double getBaseSalary() {
17        return baseSalary;
18    }
19
20    public double earnings() {
21        return getBaseSalary() + super.earnings();
22    }
23
24    public String toString() {

```

25	return String.format("Base-Salaried " +
26	super.toString() + "\nbase salary " + getBaseSalary());
27	}
28	}

### Data dan Analisis hasil percobaan

#### Pertanyaan

##### 1. Ketikkan kode ini.

Main.java	
1	public class Main {
2	public static void main(String[] args) {
3	Employee employee = new Employee();
4	}
5	}

Jalankan Main.java untuk polymorfisme Employee, analisis dan jelaskan keluaran program tersebut!

.....

.....

##### 2. Jalankan program dengan main sebagai berikut.

Main.java	
1	public class Main {
2	public static void main(String[] args) {
3	SalariedEmployee salariedEmployee = new
4	SalariedEmployee("Daniel", "135", 800.00);
5	HourlyEmployee hourlyEmployee = new
6	HourlyEmployee("Karina", "234", 16.75, 40);
7	CommissionEmployee commissionEmployee = new
8	CommissionEmployee("Keanu", "145", 10000, .06);
9	BasePlusCommissionEmployee basePlusCommissionEmployee =
10	new BasePlusCommissionEmployee("Bondan", "234", 5000, .04,
11	300);
12	System.out.println("Employees diproses secara
13	terpisah:\n");
14	System.out.printf("%s\n%s: \$%,.2f\n\n",
15	salariedEmployee, "pendapatan:",
16	salariedEmployee.earnings());
17	System.out.printf("%s\n%s: \$%,.2f\n\n",
18	hourlyEmployee, "pendapatan:",
19	hourlyEmployee.earnings());
20	System.out.printf("%s\n%s: \$%,.2f\n\n",
21	commissionEmployee, "pendapatan:",
22	commissionEmployee.earnings());
23	System.out.printf("%s\n%s: \$%,.2f\n\n",
24	basePlusCommissionEmployee,
25	"earned",
26	basePlusCommissionEmployee.earnings());
27	
28	Employee[] employees = new Employee[4];
29	employees[0] = salariedEmployee;
30	employees[1] = hourlyEmployee;
31	employees[2] = commissionEmployee;

```

32         employees[3] = basePlusCommissionEmployee;
33         System.out.println("Employees      diproses      secara
34 polimorfisme:\n");
35         for (Employee currentEmployee : employees) {
36             System.out.println(currentEmployee);
37             if (currentEmployee instanceof
38 BasePlusCommissionEmployee) {
39                 BasePlusCommissionEmployee employee =
40 (BasePlusCommissionEmployee) currentEmployee;
41                 employee.setBaseSalary(1.10 *
42 employee.getBaseSalary());
43                 System.out.printf(
44                     "Gaji pokok setelah dinaikkan 10%% :
45 $%,.2f\n",
46                     employee.getBaseSalary());
47             }
48             System.out.printf("pendapatan:      $%,.2f\n\n",
49 currentEmployee.earnings());
50         }
51         for (int j = 0; j < employees.length; j++) {
52             System.out.printf("Employee   %d   =   %s\n",    j,
53 employees[j].getClass().getName());
54         }
55     }

```

Analisis dan jelaskan output program (berdasarkan konsep polimorfisme)!

.....

3. Buat objek dari method Employee? Jelaskan hasil dari output program tersebut!

.....

4. Tambahkan atribut tanggal lahir di Kelas Employee, serta tambahkan method pendukungnya (accesor dan mutator). Modifikasi program agar sesuai. Asumsikan gaji yang diterima adalah per bulan, buat kelas uji untuk menguji program yang sudah anda modifikasi, kemudian buat objek dari semua class (salariedEmployee, hourlyEmployee, commissionEmployee, basePlusCommissionEmployee dan hitung gajinya secara polimorfisme, serta tambahkan gajinya sebesar 100.000 jika bulan ini adalah bulan ulang tahunnya.

.....

5. Perusahaan yang mengaplikasikan program polimorfisme diatas ingin menambahkan kriteria baru untuk penggajian karyawannya, yaitu penggajian berdasarkan banyaknya barang yang diproduksi. Dengan ketentuan gaji karyawan tersebut adalah hasil dari banyaknya barang yang diproduksi per minggu dikalikan upah per barangnya.
- Analisis dan jelaskan proses modifikasi program diatas (dimulai dari pemilihan jenis class, perancangan class, dan penempatan class)
  - Implementasi hasil analisis tersebut ke dalam program dan buat kelas uji dengan minimal 4 objek yang dibentuk.

.....

## Tugas Praktikum

I. Buatlah sebuah kelas **abstract** Kue yang memiliki atribut dan method sebagai berikut

- nama : String
- harga : double
- + *hitungHarga()*\*\* : double
- + toString : String (*menampilkan nama kue dan harga*)

**\*\* abstract**

II. Buatlah 2 subklas dari klas Kue yaitu

a. KuePesanan

- berat : double
- + hitungHarga() : double
- Hitung harga berdasarkan harga x berat

b. KueJadi

- jumlah : double
- + hitungHarga() : double
- Hitung harga berdasarkan harga x jumlah x 2

III. Berdasarkan 2 kelas tersebut, buatlah :

1. Array yang terdiri dari 20 kue
2. Isikan 20 objek kue dengan berbagai jenis kue (KuePesanan atau KueJadi)
3. Dari array tersebut :
  - a. Tampilkan semua kue dan harus ditampilkan jenis kue nya
  - b. Hitung total harga yang didapat dari semua jenis kue
  - c. Hitung total harga dan total berat dari KuePesanan
  - d. Hitung total harga dan total jumlah dari KueJadi
  - e. Tampilkan informasi kue dengan harga (harga akhir) terbesar