

**LAPORAN PRAKTIKUM**  
**PEMROGRAMAN BERBASIS OBJEK**  
**PERTEMUAN 6**



Disusun oleh :

Nama : Fatimah Az Zahra

NIM : 2211102160

Kelas : IF-10-K

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**INSTITUT TEKNOLOGI TELKOM PURWOKERTO**  
**2024**

## MODUL 06

### A. Guided

#### 1. Binatang.java

##### Source Code

```
5 package Nama_Binatang;
6 //Fatimah Az Zahra_2211102160_IF10K
7 public abstract class Binatang {
8     private String jenis;
9
10    //constructor
11    Binatang (String jenis){
12        this.jenis = jenis;
13    }
14
15    //deklarasi metode
16    protected abstract void suara();
17
18    //pemanggilan polimorfism
19    public String toString(){
20        return "Seekor " + jenis;
21    }
22 }
```

#### Burung.java

##### Source Code

```
5 package Nama_Binatang;
6 //Fatimah Az Zahra_2211102160_IF10K
7 public class Burung extends Binatang { //child binatang atau turunan
8     Burung(String nama) {
9         super("Burung"); //pemanggilan kelas abstrak
10        this.nama = nama;
11    }
12
13    public void suara() { //metode suara
14        System.out.println("berkicau");
15    }
16
17    public String toString() {
18        return super.toString() + " " + nama; //pengembalian nilai polimorfism
19    }
20    private String nama;
21 }
```

#### Kucing.java

##### Source Code

```
5 package Nama_Binatang;
6 //Fatimah Az Zahra_2211102160_IF10K
7
8 public class Kucing extends Binatang { //child binatang atau turunan
9
10    Kucing(String nama) { //pemanggilan kelas abstrak
11        super("Kucing");
12        this.nama = nama;
13    }
14
15    public void suara() { //metode suara
16        System.out.println("mengeong");
17    }
18
19    public String toString() {
20        return super.toString() + " " + nama; //pengembalian nilai polimorfism
21    }
22    private String nama;
23 }
```

## Anjing.java

### Source Code

```
5 package Nama_Binatang;
6 //Fatimah Az Zahra_2211102160_IF10K
7
8 public class Anjing extends Binatang {
9
10     Anjing(String nama) {
11         super("Anjing"); //Pemanggilan kelas abstrak
12         this.nama = nama;
13     }
14
15     public void suara() { //metode suara
16         System.out.println("menggonggong");
17     }
18
19     public String toString() {
20         return super.toString() + " " + nama; //pengembalian nilai polimorfism
21     }
22     private String nama;
23 }
```

## Kambing.java

### Source Code

```
5 package Nama_Binatang;
6 //Fatimah Az Zahra_2211102160_IF10K
7
8 public class Kambing extends Binatang { //child binatang atau turunan
9
10     Kambing(String nama) {
11         super("Kambing"); //pemanggilan kelas abstrak
12         this.nama = nama;
13     }
14
15     public void suara() { //metode suara
16         System.out.println("mengembik");
17     }
18
19     public String toString() {
20         return super.toString() + " " + nama; //pengembalian nilai polimorfism
21     }
22     private String nama;
23 }
```

## CobaPolimorplic.java

### Source Code

```
5 package Nama_Binatang;
6 //Fatimah Az Zahra_2211102160_IF10K
7
8 import java.util.Random;
9
10 public class CobaPolimorplic {
11
12     public static void main(String[] args) {
13         Binatang[] peliharaanku = {new Burung("Kakak Tua"),
14             new Kambing("Etawa"),
15             new Anjing("Kintamani"),
16             new Kucing("Anggora")};
17
18         Binatang kesayangan;
19         Random pilihan = new Random(); //memilih secara acak
20
21         //random
22         kesayangan = peliharaanku[pilihan.nextInt(peliharaanku.length)]; //mengacak bilangan dari 0 sampai length-1 atau (i-1).
23         System.out.println(
24             "Binatang Kesayangan anda: "+kesayangan);
25         System.out.print("Suaranya: ");
26         kesayangan.suara(); //diambil dari kelas induk
27     }
28 }
```

## Output :

```
--- exec:3.1.0:exec (default-cli) @ 2211102160_Modul6 ---
Binatang Kesayangan anda : Seekor Kucing Anggora
Suaranya : mengeong
-----
BUILD SUCCESS
-----
Total time: 3.171 s
Finished at: 2024-04-29T10:08:37+07:00
-----
```

## Penjelasan program:

Program ini merupakan contoh penggunaan pewarisan dan polimorfisme dalam pemrograman berorientasi objek dengan menggunakan bahasa pemrograman Java.

- Package ``nama_binatang`` tidak di deklarasikan paket secara eksplisit, tapi kita dapat menyimpulkan bahwa semua kelas ini berada dalam sebuah paket bernama ``nama_binatang``.
- Kelas ``Binatang``: Kelas ini adalah kelas abstrak yang memiliki atribut ``jenis``, konstruktor untuk menginisialisasi ``jenis``, metode abstrak ``suara()``, dan metode ``toString()`` yang mengembalikan informasi tentang jenis binatang.
- Kelas ``Burung``, ``Kucing``, ``Anjing``, ``Kambing``: Kelas-kelas ini merupakan turunan dari kelas ``Binatang`` yang mengimplementasikan metode ``suara()`` sesuai dengan jenis binatangnya masing-masing. Masing-masing kelas memiliki atribut ``nama`` untuk menambahkan informasi tambahan tentang binatang tersebut.
- Kelas ``CobaPolimorpic``: Kelas ini merupakan kelas yang berisi method ``main()`` yang merupakan entry point untuk program. Pada method ``main()``, beberapa objek dari kelas-kelas turunan ``Binatang`` dibuat dan dimasukkan ke dalam sebuah array. Kemudian, objek binatang dipilih secara acak dari array tersebut untuk ditampilkan informasinya dan memanggil metode ``suara()`` secara polimorfik.

## 2. Direktur.Java

### Source Code

```
5 package Pegawai;
6 //Fatimah Az Zahra_2211102160_IF10K
7
8 public final class Direktur extends Pegawai {
9
10     private double gajiDirektur;
11     private double dividenSaham;
12
13     //Konstruktor Kelas Direktur
14     public Direktur(String nama, double gaji, double dividen) {
15         super(nama); //Memanggil konstruktor kelas Pegawai
16         setGajiDirektur(gaji);
17         setDividen(dividen);
18     }
19
20     //Mengeset gaji direktur (percabangan)
21     public void setGajiDirektur(double gaji)
22     {
23         if (gaji > 0) {
24             gajiDirektur = gaji;
25         } else {
26             gajiDirektur = 0;
27         }
28     }
29
30     //Mengeset hasil pembagian dividen keuntungan saham
31     public void setDividen(double dividen) {
32         if (dividen > 0) {
33             dividenSaham = dividen;
34         } else {
35             dividenSaham = 0;
36         }
37     }
38
39     }
40
41     public String nama()//Method yang mengembalikan nama
42     {
43         return super.namaPegawai();
44     }
45
46     public String jabatan()//Method yang mengembalikan jabatan
47     {
48         return "Direktur";
49     }
50
51     //Method yang mengembalikan besar gaji direktur
52     public double gajiPerBulan() {
53         return gajiDirektur;
54     }
```

```

56         //Method yang mengembalikan besar dividen saham
57     public double labaDividen() {
58         return dividenSaham;
59     }
60
61     //Pengimplementasian / Pendefinisian method abstract dari kelas Pegawai
62     //Method ini mengembalikan besar gaji direktur
63     public double income() {
64         return (gajiDirektur + dividenSaham);
65     }
66 }

```

## Pegawai.Java

### Source Code

```

5 package Pegawai;
6 //Fatimah Az Zahra_2211102160_IF10K
7 public abstract class Pegawai {
8
9     private String namaPeg;
10
11     public Pegawai(String nama) { //metode untuk nama
12         namaPeg = nama;
13     }
14
15     public String namaPegawai() { //metode untuk mengembalikan nama pegawai
16         return namaPeg;
17     }
18     public abstract double income(); //polimorfism
19 }

```

## Test.Java

### Source Code

```

5 package Pegawai;
6 //Fatimah Az Zahra_2211102160_IF10K
7 import java.text.DecimalFormat;
8 public class Test {
9
10     public static void main(String args[]) {
11         Pegawai pgw;
12
13         //Membuat objek referensi dari kelas abstrak //Pegawai
14         String output = "";
15         Direktur d = new Direktur("Fatimah", 12000000.00, 7500000.00);
16         DecimalFormat digitPresisi = new DecimalFormat("0.00");
17         pgw = d;
18
19         /*objek referensi dari kelas abstrak pegawai (pgw) merefer objek
20         dari kelas Direktur (d) yang diturunkan dari kelas abstrak
21         pegawai */
22         System.out.println("\nDEMO INHERITANS, ENKAPSULASI, POLIMORFI");
23         System.out.println("-----\n");
24         // Mencetak informasi Direktur ke console
25         System.out.println("Nama : " + d.namaPegawai() + "\n"
26             + "Jabatan : " + d.jabatan() + "\n" + "Gaji : "
27             + digitPresisi.format(d.gajiPerBulan()) + "\n"
28             + "Dividen : " + digitPresisi.format(d.labaDividen()) + "\n"
29             + "Total : " + digitPresisi.format(d.income()) + "\n");
30         System.exit(0);
31     }
32 }

```

## Output:

```
--- exec:3.1.0:exec (default-cli) @ 2211102160_Modul6 ---

DEMO INHERITANS, ENKAPSULASI, POLIMORFI
-----

Nama : Fatimah
Jabatan : Direktur
Gaji : 12000000.00
Dividen : 7500000.00
Total : 19500000.00

-----

BUILD SUCCESS
-----

Total time: 3.448 s
Finished at: 2024-04-29T10:50:35+07:00
-----
```

## Penjelasan program:

Pada program tersebut terdapat dua kelas utama, yaitu `Pegawai` dan `Direktur`. Kelas `Pegawai` adalah kelas abstrak yang memiliki atribut nama pegawai dan method abstrak untuk menghitung pendapatan (`income`). Kelas `Direktur` merupakan turunan dari kelas `Pegawai`, yang memiliki atribut tambahan yaitu gaji direktur dan dividen saham. Melalui konsep inheritance, `Direktur` mewarisi sifat-sifat kelas `Pegawai` dan mengimplementasikan method abstrak `income`. Dalam kelas `Test`, dilakukan pengujian dengan membuat objek `Direktur` dan memanfaatkannya sebagai objek `Pegawai`. Program ini mengilustrasikan penggunaan konsep pewarisan, enkapsulasi, dan polimorfisme untuk memodelkan hubungan antara pegawai dan direktur dalam sebuah organisasi.

## 3. Ekspresi\_Wajah.java

### Source Code

```
5 package Ekspresi_Wajah;
6 //Fatimah Az Zahra_2211102160_IF10K
7
8 class EkspresiWajah {
9     public String respons() {
10         return ("Lihat Wajahku ini");
11     }
12 }
```

## Gembira.java

### Source Code

```
5 package Ekspresi_Wajah;
6 //Fatimah Az Zahra_2211102160_IF10K
7
8 class Gembira extends EkspresiWajah {
9
10     public String respons() {
11         return ("Ha..ha..saya lagi senang =)");
12     }
13 }
```

## Sedih.java

### Source Code

```
5 package Ekspresi_Wajah;
6 //Fatimah Az Zahra_2211102160_IF10K
7
8 class Sedih extends EkspresiWajah {
9
10     public String respons() {
11         return ("Hiks..hiks.. =(");
12     }
13 }
```

## Ekspresi.java

### Source Code

```
5 package Ekspresi_Wajah;
6 //Fatimah Az Zahra_2211102160_IF10K
7
8 public class Ekspresi {
9
10     public static void main(String args[]) {
11         System.out.println("DEMO POLIMORFISME");
12         System.out.println("=====");
13         EkspresiWajah objEkspresi = new EkspresiWajah();
14         Gembira objGembira = new Gembira();
15         Sedih objSedih = new Sedih();
16         EkspresiWajah[] ekspresi = new EkspresiWajah[3];
17         ekspresi[0] = objEkspresi;
18         ekspresi[1] = objGembira;
19         ekspresi[2] = objSedih;
20         System.out.println("Ekspresi[0]: " + ekspresi[0].respons());
21         System.out.println("Ekspresi[1]: " + ekspresi[1].respons());
22         System.out.println("Ekspresi[2]: " + ekspresi[2].respons());
23     }
24 }
```



## Output:

```
--- exec:3.1.0:exec (default-cli) @ 2211102160_Modul6 ---
DEMO POLIMORFISME
=====
Ekspresi[0]:Lihat Wajahku ini
Ekspresi[1]:Ha..ha..saya lagi senang =)
Ekspresi[2]:Hiks..hiks.. =(
-----
BUILD SUCCESS
-----
Total time: 3.859 s
Finished at: 2024-05-02T21:41:58+07:00
-----
```

## Penjelasan Program:

Program tersebut mengilustrasikan penggunaan polimorfisme dalam pemrograman Java untuk merepresentasikan berbagai ekspresi wajah. Terdapat tiga kelas yang terlibat: `EkspresiWajah` sebagai kelas dasar, dan dua subkelasnya, yaitu `Gembira` dan `Sedih`. Setiap subkelas memiliki method `respons()` yang di-overriding untuk mengembalikan string yang merepresentasikan ekspresi wajah masing-masing. Dalam kelas `Ekspresi`, tiga objek dibuat dan disimpan dalam array `EkspresiWajah`, lalu method `respons()` dipanggil untuk setiap objek dalam array tersebut. Melalui pendekatan ini, program menunjukkan kemampuan polimorfisme, di mana objek dari kelas yang berbeda dapat diakses dan memanggil method yang sesuai dengan perilaku kelas masing-masing melalui referensi yang sama.

## B. Unguided

### 1. Unguided1

#### Soal:

Dalam sebuah perusahaan, terdapat banyak employess, diantaranya Salaried Employee, Commission Employee, dan Project Planner.

1. Setiap Employee tersebut memiliki fungsi untuk menghitung gaji dan cetak informasi
2. Salaried Employee memiliki atribut nama, nip, upah mingguan
3. Commission Employee memiliki atribut nama, nip, gaji pokok, komisi dan total penjualan
4. Project Planner memiliki atribut nama, nip, gaji pokok, komisi dan total hasil proyek

Setiap kelas harus memiliki konstruktor, fungsi untuk menghitung gaji setiap pegawai, dan fungsi untuk menampilkan informasi yang dimiliki setiap jenis employee. Fungsi hitung gajinya adalah :

- Salaried Employee : gaji = upah mingguan
- Commission Employee : gaji = gaji pokok + (komisi \* total penjualan)
- Project Planner : gaji = gaji pokok + (komisi \* total hasil proyek) - pajak. Pajak dihitung 5 % dari gaji pokok.

**Buatlah program tersebut dalam bahasa Java menggunakan konsep polymorphism. Setiap kelas dibuat dalam file yang berbeda**

#### Employee.Java

##### Source code

```
5 package Employees;
6 //Fatimah Az Zahra_2211102160_IF10K
7
8 public abstract class Employee {
9     private String name;
10    private String nip;
11
12    // Konstruktor untuk inisialisasi atribut nama dan NIP
13    public Employee(String name, String nip) {
14        this.name = name;
15        this.nip = nip;
16    }
17
18    // Method untuk mendapatkan nama pegawai
19    public String getName() {
20        return name;
21    }
22
23    // Method untuk mendapatkan NIP pegawai
24    public String getNip() {
25        return nip;
26    }
27
28    // Method abstrak untuk menghitung gaji pegawai
29    public abstract double calculatePay();
30
31    // Method abstrak untuk menampilkan informasi pegawai
32    public abstract void displayInfo();
33 }
```

## SalariedEmployee.Java

### Source Code

```
5 package Employees;
6 //Fatimah Az Zahra_2211102160_IF10K
7 public class SalariedEmployee extends Employee {
8     private double weeklySalary;
9
10    // Konstruktor untuk inisialisasi atribut nama, NIP, dan upah mingguan
11    public SalariedEmployee(String name, String nip, double weeklySalary) {
12        super(name, nip); // Memanggil konstruktor kelas induk (Employee)
13        this.weeklySalary = weeklySalary;
14    }
15
16    // Implementasi method untuk menghitung gaji pegawai
17    @Override
18    public double calculatePay() {
19        return weeklySalary; // Gaji Salaried Employee sama dengan upah mingguan
20    }
21
22    // Implementasi method untuk menampilkan informasi pegawai
23    @Override
24    public void displayInfo() {
25        System.out.println("Name: " + getName());
26        System.out.println("NIP: " + getNip());
27        System.out.println("Weekly Salary: " + weeklySalary);
28    }
29 }
```

## CommissionEmployee.Java

### Source Code

```
5 package Employees;
6 //Fatimah Az Zahra_2211102160_IF10K
7
8 public class CommissionEmployee extends Employee {
9     private double baseSalary;
10    private double commissionRate;
11    private double totalSales;
12
13    // Konstruktor dengan argumen sesuai dengan deskripsi
14    public CommissionEmployee(String name, String nip, double baseSalary, double commissionRate, double totalSales) {
15        super(name, nip); // Memanggil konstruktor kelas induk (Employee)
16        this.baseSalary = baseSalary;
17        this.commissionRate = commissionRate;
18        this.totalSales = totalSales;
19    }
20
21    // Override method calculatePay untuk menghitung gaji pegawai
22    @Override
23    public double calculatePay() {
24        // Gaji Commission Employee = Gaji pokok + (komisi * total penjualan)
25        return baseSalary + (commissionRate * totalSales);
26    }
27
28    // Override method calculatePay untuk menghitung gaji pegawai
29    @Override
30    public double calculatePay() {
31        // Gaji Commission Employee = Gaji pokok + (komisi * total penjualan)
32        return baseSalary + (commissionRate * totalSales);
33    }
34
35    // Override method displayInfo untuk menampilkan informasi pegawai
36    @Override
37    public void displayInfo() {
38        System.out.println("Name: " + getName()); // Menggunakan method getName() dari kelas Employee
39        System.out.println("NIP: " + getNip()); // Menggunakan method getNip() dari kelas Employee
40        System.out.println("Base Salary: " + baseSalary);
41        System.out.println("Commission Rate: " + commissionRate);
42        System.out.println("Total Sales: " + totalSales);
43    }
44 }
```

## ProjectPlanner.Java

### Source Code

```
5 package Employees;
6 //Fatimah Az Zahra_2211102160_IF10K
7
8 public class ProjectPlanner extends Employee {
9     private double baseSalary;
10    private double commissionRate;
11    private double totalProjectResults;
12
13    // Konstruktor untuk inisialisasi atribut nama, NIP, gaji pokok, komisi, dan total hasil proyek
14    public ProjectPlanner(String name, String nip, double baseSalary, double commissionRate, double totalProjectResults) {
15        super(name, nip); // Memanggil konstruktor kelas induk (Employee)
16        this.baseSalary = baseSalary;
17        this.commissionRate = commissionRate;
18        this.totalProjectResults = totalProjectResults;
19    }
20
21    // Implementasi method untuk menghitung gaji pegawai
22    @Override
23    public double calculatePay() {
24        double tax = 0.05 * baseSalary; // Menghitung pajak (5% dari gaji pokok)
25        // Gaji Project Planner = Gaji pokok + (komisi * total hasil proyek) - pajak
26        return baseSalary + (commissionRate * totalProjectResults) - tax;
27    }
28
29    // Implementasi method untuk menampilkan informasi pegawai
30    @Override
31    public void displayInfo() {
32        System.out.println("Name: " + getName());
33        System.out.println("NIP: " + getNip());
34        System.out.println("Base Salary: " + baseSalary);
35        System.out.println("Commission Rate: " + commissionRate);
36        System.out.println("Total Project Results: " + totalProjectResults);
37    }
38 }
```

## Main.Java

### Source Code

```
5 package Employees;
6 //Fatimah Az Zahra_2211102160_IF10K
7
8 public class Main {
9     public static void main(String[] args) {
10        // Membuat objek SalariedEmployee
11        SalariedEmployee salariedEmployee = new SalariedEmployee("John Doe", "123456", 1000.0);
12        System.out.println("Salaried Employee:");
13        salariedEmployee.displayInfo();
14        System.out.println("Pay: $" + salariedEmployee.calculatePay());
15        System.out.println();
16
17        // Membuat objek CommissionEmployee
18        CommissionEmployee commissionEmployee = new CommissionEmployee("Jane Smith", "789012", 500.0, 0.1, 2000.0);
19        System.out.println("Commission Employee:");
20        commissionEmployee.displayInfo();
21        System.out.println("Pay: $" + commissionEmployee.calculatePay());
22        System.out.println();
23
24        // Membuat objek ProjectPlanner
25        ProjectPlanner projectPlanner = new ProjectPlanner("Alice Johnson", "345678", 800.0, 0.05, 10000.0);
26        System.out.println("Project Planner:");
27        projectPlanner.displayInfo();
28        System.out.println("Pay: $" + projectPlanner.calculatePay());
29    }
30 }
```

## Output :

```
--- exec:3.1.0:exec (default-cli) @ 2211102160_Modul6 ---
Salaried Employee:
Name: John Doe
NIP: 123456
Weekly Salary: 1000.0
Pay: $1000.0

Commission Employee:
Name: Jane Smith
NIP: 789012
Base Salary: 500.0
Commission Rate: 0.1
Total Sales: 2000.0
Pay: $700.0

Project Planner:
Name: Alice Johnson
NIP: 345678
Base Salary: 800.0
Commission Rate: 0.05
Total Project Results: 10000.0
Pay: $1260.0

-----
BUILD SUCCESS
-----

Total time: 3.943 s
Finished at: 2024-05-03T18:33:33+07:00
-----
```

## Penjelasan Program:

Program diatas merupakan implementasi sistem penggajian karyawan menggunakan bahasa pemrograman Java dengan materi polimorfism. Ada empat kelas utama yang digunakan: ``Employee``, ``SalariedEmployee``, ``CommissionEmployee``, dan ``ProjectPlanner``. Kelas ``Employee`` adalah kelas abstrak yang menjadi induk dari semua jenis karyawan, dengan method untuk menghitung gaji dan menampilkan informasi. Sementara itu, kelas-kelas lainnya mewakili jenis-jenis karyawan yang berbeda, seperti karyawan dengan upah mingguan tetap (``SalariedEmployee``), karyawan dengan gaji pokok ditambah komisi dari penjualan (``CommissionEmployee``), dan karyawan dengan gaji pokok ditambah komisi dari hasil proyek (``ProjectPlanner``). Setiap kelas memiliki method untuk menghitung gaji dan menampilkan informasi karyawan sesuai dengan jenisnya. Dengan menggunakan polimorfisme, program dapat dengan mudah mengelola berbagai jenis karyawan dan menghitung gaji mereka secara efisien.