

LAPORAN PRAKTIKUM
PEMROGRAMAN BERBASIS OBJEK
PERTEMUAN 4



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 04

A. Guided

1. Point.java

Source Code

```
5 package GUIDED1;
6
7 //Fatimah Az Zahra_2211102160_IF10K
8 public class Point {
9     //deklarasi atribut
10    protected float x, y;
11
12    // Konstruktor Point
13    public Point(float a, float b) {
14        System.out.println("Konstruktor dijalankan");
15        x = a;
16        y = b;
17    }
18
19    //Membuat method cekPoint
20    public void cekPoint() {
21        System.out.println("Poin: [" + x + ", " + y + "]");
22    }
23 }
```

Circle.Java

Source Code

```
5 package GUIDED1;
6
7 //Fatimah Az Zahra_2211102160>If10K
8 public class Circle extends Point {
9     private float radius;
10
11    // Konstruktor Circle
12    public Circle(float r, float a, float b) {
13        super(a, b);
14        radius = r;
15        System.out.println("Konstruktor Circle dijalankan");
16    }
17
18    // Method cetakPoint untuk mencetak informasi titik dan radius lingkaran
19    public void cetakPoint() {
20        super.cekPoint(); // Memanggil method cekPoint dari superclass Point
21        System.out.println("Radius: " + radius);
22    }
23 }
```

Main.Java

Source Code

```
5 package GUIDED1;
6 //Fatimah Az Zahra_2211102160>If10K
7 public class Main {
8     public static void main(String[] args){
9         Circle circle1 = new Circle(6.5f, 8.2f, 1.9f);
10        circle1.cetakPoint();
11        System.out.println(" ");
12
13        Circle circle2 = new Circle(10, 5, 5);
14        circle2.cetakPoint();
15    }
16 }
```

Output :

```
--- exec:3.1.0:exec (default-cli) @ 2211102160_Modul4 ---
Konstruktor dijalankan
Konstruktor Circle dijalankan
Poin: [8.2, 1.9]
Radius: 6.5

Konstruktor dijalankan
Konstruktor Circle dijalankan
Poin: [5.0, 5.0]
Radius: 10.0
-----
BUILD SUCCESS
-----
Total time: 3.377 s
Finished at: 2024-03-25T10:09:49+07:00
```

Penjelasan program:

Program di atas merupakan contoh sederhana dari konsep pewarisan (*inheritance*) dalam bahasa pemrograman Java. Dalam program ini, terdapat dua kelas utama, yaitu `Point` dan `Circle`. Kelas `Point` merepresentasikan sebuah titik dalam koordinat dua dimensi dengan atribut `x` dan `y`, serta memiliki konstruktor untuk menginisialisasi nilai titik dan method untuk mencetak koordinat titik. Kelas `Circle` merupakan turunan dari kelas `Point` dan menambahkan atribut `radius` yang merepresentasikan panjang jari-jari lingkaran. Konstruktor kelas `Circle` digunakan untuk menginisialisasi nilai titik dan radius lingkaran, sementara method `cetakPoint()` mencetak informasi tentang koordinat titik dan radius lingkaran. Kelas `Main` digunakan untuk menjalankan program dengan membuat objek `Circle` dan mencetak informasi titik dan radius lingkaran dari setiap objek yang dibuat. Dengan demikian, program ini memberikan pemahaman tentang cara menggunakan pewarisan dalam Java untuk memperluas fungsionalitas kelas dan mendukung hierarki kelas yang lebih kompleks.

2. Mammal.Java

Source Code

```
5 package GUIDED2;
6 //Fatimah Az Zahra_2211102160_If10K
7 public class Mammal {
8     protected String name; // Atribut nama yang dilindungi
9     // Metode untuk tidur
10    public void sleep(){
11        System.out.println("zzzz zzzz zzzz");
12    }
13 }
```

Dog.Java

Source Code

```
5 package GUIDED2;
6 //Fatimah Az Zahra_2211102160_If10K
7 // Kelas anak yang mewarisi kelas Mammal, merepresentasikan Dog
8 public class Dog extends Mammal{
9     // Metode untuk bersuara
10    public void speak(){
11        System.out.println("Arf! Arf!");
12    }
13 }
```

Duck.Java

Source Code

```
5 package GUIDED2;
6 //Fatimah Az Zahra_2211102160_If10K
7 // Kelas anak yang mewarisi kelas Mammal, merepresentasikan Duck
8 public class Duck extends Mammal {
9     // Metode untuk bersuara
10    public void speak(){
11        System.out.println("Quack! Quack!");
12    }
13 }
```

Horse.Java

Source Code

```
5 package GUIDED2;
6 //Fatimah Az Zahra_2211102160_If10K
7 // Kelas anak yang mewarisi kelas Mammal, merepresentasikan Horse
8 public class Horse extends Mammal{
9     // Metode untuk bersuara
10    public void speak(){
11        System.out.println("Whinny! Whinny!");
12    }
13 }
```

Main.Java

Source Code

```

5 package GUIDED2;
6 //Fatimah Az Zahra_2211102160_If10K
7 // Kelas utama yang berisi metode main
8 public class Main {
9     public static void main(String[] arguments){
10         // Membuat objek-objek
11         Dog guffy = new Dog(); // Sebuah objek Dog
12         Duck donald = new Duck(); // Sebuah objek Duck
13         Horse lala = new Horse(); // Sebuah objek Horse
14
15         // Menginisialisasi atribut nama untuk setiap objek
16         guffy.name = "Guffy";
17         donald.name = "Donald";
18         lala.name = "Lala";
19
20         // Menampilkan suara dan nama hewan-hewan
21         System.out.println("First we'll get the dog to speak: ");
22         guffy.speak(); // Dog bersuara
23         System.out.println(" ");
24
25         System.out.println("Now, the duck will speak: ");
26         donald.speak(); // Duck bersuara
27         System.out.println(" ");
28
29         System.out.println("Now, the horse will speak: ");
30         lala.speak(); // Horse bersuara
31         System.out.println(" ");
32
33         // Memanggil metode sleep untuk menunjukkan polimorfisme
34         guffy.sleep(); // Dog tidur
35         donald.sleep(); // Duck tidur
36         lala.sleep(); // Kuda tidur
37     }
38 }

```

Output:

```

--- exec:3.1.0:exec (default-cli) @ 2211102160_Modul4 ---
First we'll get the dog to speak:
Arf! Arf!

Now, the duck will speak:
Quack! Quack!

Now, the horse will speak:
Whinny! Whinny!

zzzz zzzz zzzz
zzzz zzzz zzzz
zzzz zzzz zzzz
-----
BUILD SUCCESS
-----
Total time: 2.171 s
Finished at: 2024-03-25T12:48:38+07:00
-----

```

Penjelasan program:

Objek `guffy`, `donald`, `morley`, dan `woodsy` dapat mengakses atribut ``name`` dan method ``sleep`` dari kelas ``Mammal`` karena kelas-kelas tersebut adalah turunan dari kelas ``Mammal``. Hal ini sesuai dengan konsep pewarisan (inheritance) yang memungkinkan kelas anak (*subclasses*) untuk mewarisi atribut dan perilaku dari kelas induk (*superclass*).

Dalam Program di atas, kelas ``Dog``, ``Duck``, ``MorleySafer``, dan ``Owl`` merupakan turunan dari kelas ``Mammal``. Oleh karena itu, mereka mewarisi semua atribut dan method yang diwariskan oleh kelas ``Mammal``. Karena atribut ``name`` dan method ``sleep`` didefinisikan dalam kelas ``Mammal`` dan bersifat ``protected`` atau ``public``, maka kelas-kelas anak dapat mengaksesnya.

Ketika objek-objek seperti ``guffy``, ``donald``, ``morley``, dan ``woodsy`` diciptakan, mereka memiliki akses langsung ke atribut ``name`` dan method ``sleep`` yang didefinisikan dalam kelas ``Mammal`` karena kedua elemen tersebut telah diwarisi oleh kelas anak. Ini menggambarkan prinsip pewarisan dalam pemrograman berorientasi objek, di mana kelas anak dapat menggunakan dan memperluas perilaku dari kelas induknya.

B. Unguided

1. Unguided1

Soal:

Sebuah perusahaan Asuransi memiliki banyak pegawai antara lain Sales, Satpam dan Manajer dengan spesifikasi sbb :

- Setiap pegawai memiliki NIP, nama, alamat, dan tahun masuk kerja
- Data satpam meliputi NIP, nama, alamat, gaji pokok dan jam lembur
- Bonus diberikan bagi Satpam yang bersedia untuk lembur dengan perhitungan Rp.10.000/jam lembur
- Data Sales meliputi NIP, nama, alamat, gaji pokok dan jumlah pelanggan yang berhasil direkrut
- Komisi diberikan dengan perhitungan Rp.50.000 untuk setiap pelanggan yang berhasil direkrut
- Data Manajer meliputi nama, alamat, gaji pokok, divisi dan tunjangan jabatan
- Tunjangan jabatan bagi Manajer diberikan jika telah bekerja selama 3 tahun sebesar 5% dari gaji pokok dan

Jika telah bekerja selama lebih dari 5 tahun sebesar 10% dari gaji pokok. Buatlah sebuah Kelas Pegawai yang menjadi induk Kelas Satpam, Kelas Sales dan Kelas Manajer. Lalu implementasikan konsep inheritance untuk menghitung gaji akhir pegawai melalui method `HitungGajiAkhir()` dan menampilkan informasi data pegawai perusahaan Asuransi tersebut pada tahun 2015.

Ketentuan:

- Data Satpam, Sales dan Manajer diinputkan melalui konstruktornya. Setelah itu semua data pegawai ditampilkan.
- Gaji Akhir = GajiPokok + Bonus atau Komisi atau Tunjangan

Pegawai.Java

Source code

```
5 package UNGUIDED1;
6 //Fatimah Az Zahra_2211102160_IF10K
7 // Kelas induk Pegawai yang akan diwarisi oleh kelas-kelas turunannya
8
9 class Pegawai {
10     protected String NIP;
11     protected String nama;
12     protected String alamat;
13     protected int tahunMasukKerja;
14
15     // Konstruktor untuk inisialisasi data pegawai
16     public Pegawai(String NIP, String nama, String alamat, int tahunMasukKerja) {
17         this.NIP = NIP;
18         this.nama = nama;
19         this.alamat = alamat;
20         this.tahunMasukKerja = tahunMasukKerja;
21     }
22
23     // Method untuk menampilkan informasi pegawai
24     public void displayInfo() {
25         System.out.println("NIP: " + NIP);
26         System.out.println("Nama: " + nama);
27         System.out.println("Alamat: " + alamat);
28         System.out.println("Tahun Masuk Kerja: " + tahunMasukKerja);
29     }
30 }
```

Satpam.Java

Source Code

```
5 package UNGUIDED1;
6 //Fatimah Az Zahra_2211102160_IF10K
7 // Kelas turunan Satpam yang mewarisi kelas Pegawai
8 class Satpam extends Pegawai {
9     private double gajiPokok;
10    private int jamLembur;
11
12    // Konstruktor untuk inisialisasi data Satpam
13    public Satpam(String NIP, String nama, String alamat, int tahunMasukKerja, double gajiPokok, int jamLembur) {
14        super(NIP, nama, alamat, tahunMasukKerja);
15        this.gajiPokok = gajiPokok;
16        this.jamLembur = jamLembur;
17    }
18
19    // Override method displayInfo() dari superclass Pegawai
20    public void displayInfo() {
21        super.displayInfo();
22        System.out.println("Gaji Pokok: Rp " + String.format("%.0f", gajiPokok)); // Format gaji dengan pemisah ribuan dan mat
23        System.out.println("Jam Lembur: " + jamLembur);
24    }
25 }
```

Sales.Java

Source Code

```
5 package UNGUIDED1;
6 // Fatimah Az Zahra_2211102160_IF10K
7 // Kelas turunan Sales yang mewarisi kelas Pegawai
8 class Sales extends Pegawai {
9     private double gajiPokok;
10    private int jumlahPelanggan;
11
12    // Konstruktor untuk inisialisasi data Sales
13    public Sales(String NIP, String nama, String alamat, int tahunMasukKerja, double gajiPokok, int jumlahPelanggan) {
14        super(NIP, nama, alamat, tahunMasukKerja);
15        this.gajiPokok = gajiPokok;
16        this.jumlahPelanggan = jumlahPelanggan;
17    }
18
19    // Override method displayInfo() dari superclass Pegawai
20    public void displayInfo() {
21        super.displayInfo();
22        System.out.println("Gaji Pokok: Rp " + String.format("%.0f", gajiPokok)); // Format gaji dengan pemisah ribuan dan mata
23        System.out.println("Jumlah Pelanggan: " + jumlahPelanggan);
24    }
25 }
```

Manajer.Java

Source Code

```
5 package UNGUIDED1;
6
7 // Fatimah Az Zahra_2211102160_IF10K
8 // Kelas turunan Manajer yang mewarisi kelas Pegawai
9 class Manajer extends Pegawai {
10    private double gajiPokok;
11    private String divisi;
12
13    // Konstruktor untuk inisialisasi data Manajer
14    public Manajer(String NIP, String nama, String alamat, int tahunMasukKerja, double gajiPokok, String divisi) {
15        super(NIP, nama, alamat, tahunMasukKerja);
16        this.gajiPokok = gajiPokok;
17        this.divisi = divisi;
18    }
19
20    // Override method displayInfo() dari superclass Pegawai
21    public void displayInfo() {
22        super.displayInfo();
23        System.out.println("Gaji Pokok: Rp " + String.format("%.0f", gajiPokok)); // Format gaji dengan pemisah ribuan dan mata
24        System.out.println("Divisi: " + divisi);
25    }
26 }
```


Main.Java

Source Code

```
5 package UNGUIDED1;
6 // Fatimah Az Zahra_2211102160_IF10K
7 // Kelas utama Main
8 public class Main {
9     public static void main(String[] args) {
10         // Membuat objek-objek pegawai
11         Satpam satpam = new Satpam("0042", "Rendra", "Jl. Itik 15", 2000, 3000000, 5);
12         Sales sales = new Sales("0185", "Wibisana", "Jl. Ayam 78", 2006, 5000000, 10);
13         Manajer manajer = new Manajer("0005", "Adi", "Jl. Angsa 56", 1999, 15000000, "Keuangan");
14
15         // Menampilkan informasi pegawai
16         System.out.println("== DISPLAY DATA KARYAWAN ==");
17         System.out.println("Satpam:");
18         satpam.displayInfo();
19         System.out.println();
20         System.out.println("Sales:");
21         sales.displayInfo();
22         System.out.println();
23         System.out.println("Manajer:");
24         manajer.displayInfo();
25     }
26 }
```

Output :

```
--- exec:3.1.0:exec (default-cli) @ 2211102160_Modul4 ---
== DISPLAY DATA KARYAWAN ==

Satpam:
NIP: 0042
Nama: Rendra
Alamat: Jl. Itik 15
Tahun Masuk Kerja: 2000
Gaji Pokok: Rp 3,000,000
Jam Lembur: 5

Sales:
NIP: 0185
Nama: Wibisana
Alamat: Jl. Ayam 78
Tahun Masuk Kerja: 2006
Gaji Pokok: Rp 5,000,000
Jumlah Pelanggan: 10

Manajer:
NIP: 0005
Nama: Adi
Alamat: Jl. Angsa 56
Tahun Masuk Kerja: 1999
Gaji Pokok: Rp 15,000,000
Divisi: Keuangan

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

Total time: 3.593 s
Finished at: 2024-03-25T12:54:54+07:00
-----
```

Penjelasan Program:

Terima kasih atas klarifikasinya. Program tersebut memang merupakan contoh implementasi dari materi pewarisan tunggal dalam konsep OOP. Pewarisan tunggal (single inheritance) terjadi ketika sebuah kelas turunan hanya mewarisi sifat dan perilaku dari satu kelas induk. Dalam program tersebut, kelas-kelas `Satpam`, `Sales`, dan `Manajer` mewarisi sifat dan perilaku dari kelas induk `Pegawai`, yang merupakan contoh dari pewarisan tunggal. Dengan menggunakan pewarisan tunggal, kita dapat mengatur struktur program dengan lebih terorganisir dan menghindari duplikasi kode, sehingga mempermudah pemeliharaan dan pengembangan kode di masa depan.