

Nama : Mohammad Izza Hakiki

NIM : 24060123140139

Kelas : C1

Pengumpulan Post Test PBO Lab C1

1. Screenshoot kode yang dibuat, dan screenshoot outputnya

Pegawai.java

```
Pegawai.java > ...
1 public class Pegawai {
2     protected String nama;
3     protected int gajiPokok = 5000000;
4
5     // Constructor
6     public Pegawai(String nama) {
7         this.nama = nama;
8     }
9
10    // Method to set the name
11    public void setName(String nama) {
12        this.nama = nama;
13    }
14
15    // Method to display the employee data
16    public void tampilData() {
17        System.out.println("Nama : " + nama + ", Gaji pokok : " + gajiPokok);
18    }
19 }
```

Manager.java

```
Manager.java > ...
1 public class Manager extends Pegawai {
2     private int tunjangan = 700000; Field tunjangan can be final
3
4     // Constructor
5     public Manager(String nama) {
6         super(nama); // Call the superclass (Pegawai) constructor
7     }
8
9     // Override tampilData method to include the additional attribute
10    @Override
11    public void tampilData() {
12        System.out.println("Nama : " + nama + ", Gaji pokok : " + gajiPokok + ", Tunjangan : " + tunjangan);
13    }
14 }
15
```

Programmer.java

```
Programmer.java > ...
1 public class Programmer extends Pegawai {
2     private int bonus = 450000; Field bonus can be final
3
4     // Constructor
5     public Programmer(String nama) {
6         super(nama); // Call the superclass (Pegawai) constructor
7     }
8
9     // Override tampilData method to include the additional attribute
10    @Override
11    public void tampilData() {
12        System.out.println("Nama : " + nama + ", Gaji pokok : " + gajiPokok + ", Bonus : " + bonus);
13    }
14 }
15
```

TestPolimorfisme.java

```
Pegawai.java  Manager.java 1  Programmer.java 1  TestPolimorfisme.java X
TestPolimorfisme.java > ...
1  import java.util.ArrayList;
2
3  public class TestPolimorfisme {
    Run main | Debug main
4      public static void main(String[] args) {
5          // Create objects of Manager and Programmer
6          Pegawai pegawai1 = new Programmer("Mira");
7          Pegawai pegawai2 = new Manager("Joko");
8          Manager pegawai3 = new Manager("Argo");
9
10         // Create an ArrayList to store Pegawai objects
11         ArrayList<Pegawai> emps = new ArrayList<>();
12         emps.add(pegawai1);
13         emps.add(pegawai2);
14         emps.add(pegawai3);
15
16         // Loop through the ArrayList and call tampilData() method
17         for (Pegawai emp : emps) {
18             emp.tampilData();
19         }
20     }
21 }
22
```

Outputnya :

```
PROBLEMS 2  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS
PS D:\UNDIP\KULIAH UNDIP\Semester 4\Praktikum\Pemrograman Berorientasi Objek\Week6\post test> cd "d:\UNDIP\KULIAH UN
javac TestPolimorfisme.java } ; if ($?) { java TestPolimorfisme }
Nama : Mira, Gaji pokok : 5000000, Bonus : 450000
Nama : Joko, Gaji pokok : 5000000, Tunjangan : 700000
Nama : Argo, Gaji pokok : 5000000, Tunjangan : 700000
o PS D:\UNDIP\KULIAH UNDIP\Semester 4\Praktikum\Pemrograman Berorientasi Objek\Week6\post test> |
```

2. Jelaskan manfaat polimorfisme pada kasus ini.

Polimorfisme disini membuat kelas turunan yaitu programmer dan manajer lebih fleksibel seakan-aka mereka kelas induk. Dalam ArrayList<Pegawai>, kita bisa menyimpan objek dari kelas Manajer dan Programmer secara bersamaan tanpa harus membuat dua array terpisah. Selain itu, pemanggilan metode juga akan otomatis disesuaikan oleh java, seperti saat emp.tampilData() dipanggil, Java secara otomatis akan memilih implementasi metode dari kelas yang sesuai (Programmer atau Manager), walaupun referensinya bertipe Pegawai.

3. Apabila pada main program perlu menambahkan pegawai4 dan pegawai5! Apa permasalahan yang muncul jika diterapkan tanpa polimorfisme (inclusion)?

Jika tanpa polimorfisme, maka list untuk menyimpan objek pegawai dan manajer juga harus dibedakan. Selain itu, pemanggilan dari tiap objek harus terpisah sesuai kelas nya yaitu manajer atau pegawai.