

Nama: Gusti Ayu Genta Bhuana Paramitha

NIM: 24060123130110

Kelas: C-Informatika (PBO Lab C2)

1. Screenshot kode yang dibuat, dan screenshot outputnya

- Pegawai.java

```
J Pegawai.java
1  class Pegawai {
2      protected String nama;
3      protected int gajiPokok = 5000000;
4
5      public Pegawai(String nama) {
6          this.nama = nama;
7      }
8
9      public void tampilData() {
10         System.out.println("Nama : " + nama + ", Gaji pokok : " + gajiPokok);
11     }
12 }
13
```

- Programmer.java

```
J Programmer.java
1  class Programmer extends Pegawai {
2      private int bonus = 450000;
3
4      public Programmer(String nama) {
5          super(nama);
6      }
7
8      @Override
9      public void tampilData() {
10         super.tampilData();
11         System.out.println("Bonus : " + bonus);
12     }
13 }
14
```

- Manajer.java

J Manajer.java

```
1  class Manajer extends Pegawai {
2      private int tunjangan = 700000;
3
4      public Manajer(String nama) {
5          super(nama);
6      }
7
8      @Override
9      public void tampilData() {
10         super.tampilData();
11         System.out.println("Tunjangan : " + tunjangan);
12     }
13 }
14
```

- TestPolimorfisme.java

J TestPolimorfisme.java

```
1  import java.util.ArrayList;
2
3  public class TestPolimorfisme {
4      public static void main(String[] args) {
5          Pegawai pegawai = new Programmer("Mira");
6          Pegawai pegawai2 = new Manajer("Joko");
7          Manajer pegawai3 = new Manajer("Argo");
8
9          ArrayList<Pegawai> emps = new ArrayList<>();
10         emps.add(pegawai);
11         emps.add(pegawai2);
12         emps.add(pegawai3);
13
14         for (Pegawai emp : emps) {
15             System.out.println();
16             emp.tampilData();
17         }
18     }
19 }
20
```

- Output

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\ZenBook 14 OLED\OneDrive\문서\Praktikum PBO\Pertemuan 6\Post Test> javac *.java
PS C:\Users\ZenBook 14 OLED\OneDrive\문서\Praktikum PBO\Pertemuan 6\Post Test> java TestPolimorfisme
```

```
Nama : Mira, Gaji pokok : 5000000
Bonus : 450000
```

```
Nama : Joko, Gaji pokok : 5000000
Tunjangan : 700000
```

```
Nama : Argo, Gaji pokok : 5000000
Tunjangan : 700000
```

2. Jelaskan manfaat polimorfisme pada kasus ini.

Jawaban:

Manfaat polimorfisme pada kasus ini adalah memudahkan pengelolaan objek-objek yang berasal dari kelas berbeda, seperti Programmer dan Manajer, melalui satu tipe referensi yang sama yaitu Pegawai. Dengan menggunakan polimorfisme, semua objek dapat disimpan dalam satu ArrayList bertipe Pegawai dan dipanggil method `tampilData()` tanpa perlu mengetahui secara spesifik tipe aslinya. Hal ini membuat program menjadi lebih sederhana, fleksibel, dan lebih mudah untuk dikembangkan di masa depan. Apabila suatu saat ditambahkan subclass baru, main program tidak perlu diubah karena cukup memastikan subclass tersebut meng-override method `tampilData()`.

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

Jawaban:

Apabila pada main program perlu menambahkan pegawai4 dan pegawai5 tanpa menggunakan polimorfisme, maka akan muncul permasalahan dalam pengelolaan data. Setiap tipe objek harus dipisahkan, misalnya dengan membuat daftar sendiri untuk Programmer dan Manajer, serta harus melakukan pengecekan tipe data sebelum memanggil method `tampilData()`. Hal ini akan membuat program menjadi lebih rumit, tidak efisien, sulit dipelihara, dan rawan kesalahan, terutama jika jumlah jenis pegawai semakin bertambah.