

## POST TEST PRAKTIKUM KE 6 PBO

**Nama : Muhammad Farhan Al Ghifari**

**NIM : 24060123140186**

**Lab : C2**

### Soal

1. Screenshoot kode yang dibuat, dan screenshoot outputnya
2. Jelaskan manfaat polimorfisme pada kasus ini.
3. Apabila pada main program perlu menambahkan pegawai4 dan pegawai5! Apa permasalahan yang muncul jika diterapkan tanpa polimorfisme (inclusion)?

### Jawaban

1. Screenshoot kode yang dibuat, dan screenshoot outputnya

#### Pegawa.java

```
1  /* Nama File   : Pegawai.java
2     Nama       : Muhammad Farhan Al Ghifari - 24060123140186
3     Tanggal    : 26 - April - 2025
4  */
5
6  public class Pegawai {
7      protected String nama;
8      protected int gajiPokok = 5000000;
9
10     public Pegawai(String nama) {
11         this.nama = nama;
12     }
13
14     public void tampilData() {
15         System.out.println("Nama : " + nama + ", Gaji pokok : " + gajiPokok);
16     }
17 }
```

#### Manager.java

```
1  /* Nama File   : Manager.java
2     Nama       : Muhammad Farhan Al Ghifari - 24060123140186
3     Tanggal    : 26 - April - 2025
4  */
5
6  public class Manager extends Pegawai {
7      private int tunjangan;
8
9      public Manager(String nama) {
10         super(nama);
11         this.tunjangan = 7000000;
12     }
13
14     @Override
15     public void tampilData() {
16         super.tampilData();
17         System.out.println("Tunjangan : " + tunjangan);
18     }
19 }
```

## Programmer.java

```
1  /* Nama File   : Programmer.java
2     Nama        : Muhammad Farhan Al Ghifari - 24060123140186
3     Tanggal     : 26 - April - 2025
4  */
5
6  public class Programmer extends Pegawai {
7      private int bonus;
8
9      public Programmer(String nama) {
10         super(nama);
11         this.bonus = 450000;
12     }
13
14     @Override
15     public void tampilData() {
16         super.tampilData();
17         System.out.println("Bonus : " + bonus);
18     }
19 }
```

## TestPolimorfisme.java

```
1  /* Nama File   : TestPolimorfisme.java
2     Nama        : Muhammad Farhan Al Ghifari - 24060123140186
3     Tanggal     : 26 - April - 2025
4  */
5
6  import java.util.ArrayList;
7
8  public class TestPolimorfisme {
9      public static void main(String[] args) {
10         Pegawai pegawai1 = new Programmer(nama:"Mira");
11         Pegawai pegawai2 = new Manager(nama:"Joko");
12         Manager pegawai3 = new Manager(nama:"Argo");
13
14         ArrayList<Pegawai> emps = new ArrayList<>();
15         emps.add(pegawai1);
16         emps.add(pegawai2);
17         emps.add(pegawai3);
18
19         for (Pegawai emp : emps) {
20             emp.tampilData();
21         }
22     }
23 }
```

## Output :

```
PROBLEMS 20 OUTPUT DEBUG CONSOLE TERMINAL PORTS
cd "d:\SMT 4\PRAKTIKUM PBO\Praktikum 06\Post Test\" ; if ($?) { javac TestPolimorfisme.java }
va ; if ($?) { java TestPolimorfisme }
Nama : Mira, Gaji pokok : 500000
Bonus : 450000
Nama : Joko, Gaji pokok : 500000
Tunjangan : 700000
Nama : Argo, Gaji pokok : 500000
Tunjangan : 700000
PS D:\SMT 4\PRAKTIKUM PBO\Praktikum 06\Post Test>
```

2. Kode lebih fleksibel dan mudah dikembangkan, bisa mengelola semua jenis pegawai (Programmer, Manager) lewat satu tipe umum yaitu Pegawai, menambah jenis pegawai baru misal Designer tidak perlu ubah kode utama tinggal buat subclass baru

3. jika tidak menggunakan polimorfisme dapat membuat beberapa permasalahan berikut :

- Setiap objek pegawai harus dicek satu per satu untuk mengetahui jenisnya (apakah Manager, Programmer, atau jenis lainnya)
- Program menjadi lebih panjang, rumit, dan sulit dipelihara
- Setiap kali ada penambahan jenis pegawai baru, program utama harus diubah lagi untuk menambahkan pengecekan terhadap tipe baru tersebut
- Risiko terjadinya kesalahan (error) dalam program juga menjadi lebih tinggi