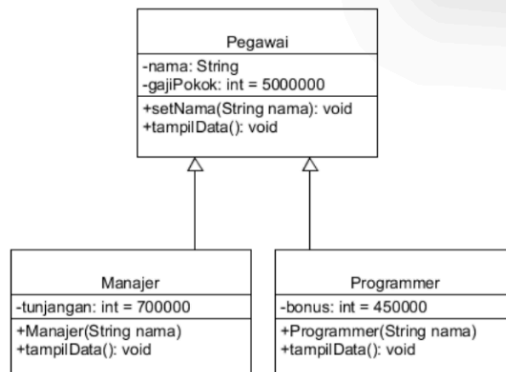


Nama: Jihan Saniyya Pudaliba

NIM: 24060123130101

Lab: C2

POST TEST PRAKTIKUM PBO



Main program:

```
import java.util.ArrayList;

public class TestPolimorfisme {

    public static void main(String[] args) {

        Pegawai pegawai = new Programmer("Mira");
        Pegawai pegawai2 = new Manajer("Joko");
        Manajer pegawai3 = new Manajer("Argo");

        ArrayList<Pegawai> emps = new ArrayList<>();
        emps.add(pegawai);
        emps.add(pegawai2);
        emps.add(pegawai3);

        for (Pegawai emp : emps) {
            emp.tampilData();
        }
    }
}
```

Output:

```
Output - PBOpraktikum (run) X
run:
Nama : Mira, Gaji pokok : 5000000
Bonus : 450000
Nama : Joko, Gaji pokok : 5000000
Tunjangan : 700000
Nama : Argo, Gaji pokok : 5000000
Tunjangan : 700000
```

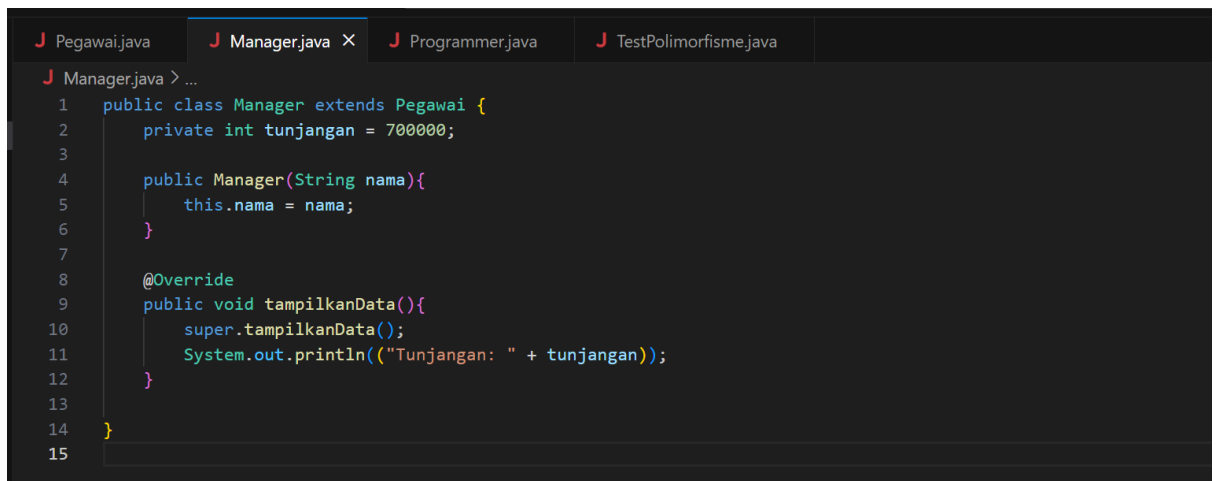
1. Screenshoot kode yang dibuat dan screenshoot outputnya

- Pegawai.java



```
J Pegawai.java X J Manager.java J Programmer.java J TestPolimorfisme.java
J Pegawai.java > ...
1 public class Pegawai {
2     protected String nama;
3     protected int gajiPokok = 5000000;
4
5     public void setNama(String nama){
6         this.nama = nama;
7     }
8
9     public void tampilkanData(){
10        System.out.println("Nama: " + nama + ", Gaji pokok: " + gajiPokok);
11    }
12 }
13
14
```

- Manager.java



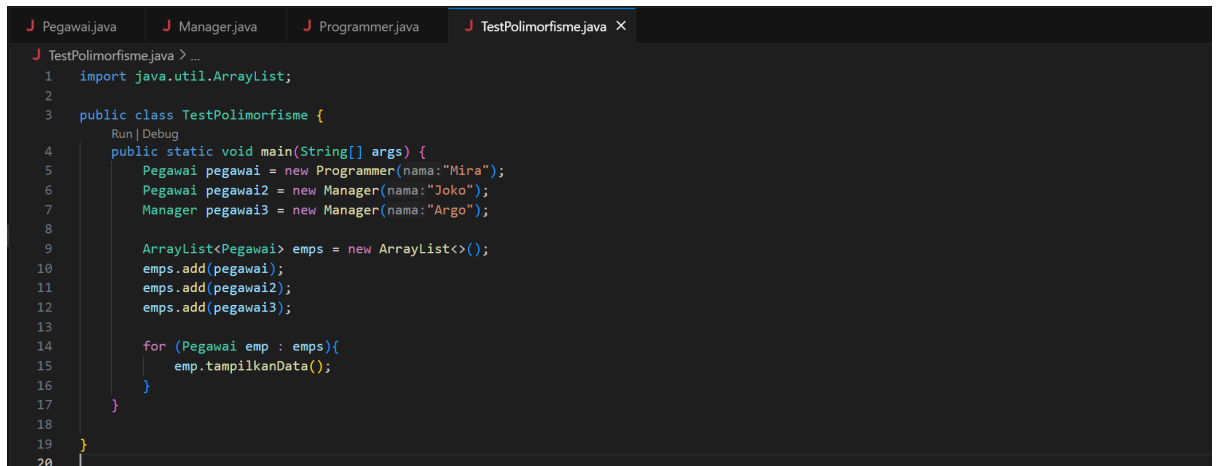
```
J Pegawai.java J Manager.java X J Programmer.java J TestPolimorfisme.java
J Manager.java > ...
1 public class Manager extends Pegawai {
2     private int tunjangan = 700000;
3
4     public Manager(String nama){
5         this.nama = nama;
6     }
7
8     @Override
9     public void tampilkanData(){
10        super.tampilkanData();
11        System.out.println(("Tunjangan: " + tunjangan));
12    }
13
14 }
15
```

- Programmer.java



```
J Pegawai.java J Manager.java J Programmer.java X J TestPolimorfisme.java
J Programmer.java > ...
1 public class Programmer extends Pegawai{
2     private int bonus = 450000;
3
4     public Programmer(String nama){
5         this.nama = nama;
6     }
7
8     @Override
9     public void tampilkanData(){
10        super.tampilkanData();
11        System.out.println("Bonus: " + bonus);
12    }
13
14 }
15
16
```

- TestPolimorfisme.java



```
1 import java.util.ArrayList;
2
3 public class TestPolimorfisme {
4     public static void main(String[] args) {
5         Pegawai pegawai = new Programmer(nama:"Mira");
6         Pegawai pegawai2 = new Manager(nama:"Joko");
7         Manager pegawai3 = new Manager(nama:"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            emp.tampilkanData();
16        }
17    }
18 }
19
20
```

- Output



```
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.

Manfaat polimorfisme pada kasus ini yaitu:

- Kode lebih fleksibel dan umum: Walaupun pegawai1, pegawai2, dan pegawai3 adalah objek dari subclass (Programmer dan Manager), namun semuanya dianggap sebagai Pegawai di dalam ArrayList<Pegawai>.
 - Pemanggilan metode yang sesuai: Saat pemanggilan emp.TampilkanData(), akan menjalankan versi metode yang sesuai dengan tipe asli objeknya.
 - Menyederhanakan logika program: Tidak perlu membuat if-else.
3. Apabila pada main program perlu menambahkan pegawai4 dan pegawai5! Apa permasalahan yang muncul jika diterapkan tanpa polimorfisme (inclusion)?

Permasalahan yang muncul yaitu program akan ribet, panjang, rentan error, dan susah dikembangkan jika ingin menambah pegawai baru.