

LAPORAN PERTEMUAN 3

NAMA : AHMAD FIKRI ZAKARIA
NIM : H1D024062
MATERI : INHERITANCE (PEWARISAN)

A. Alur Kerja (Workflow)

Pada pertemuan ini, saya menyelesaikan studi kasus sistem kepegawaian "TechMaju" yang menerapkan konsep pewarisan (*inheritance*) untuk membedakan hak akses dan perilaku antara Karyawan biasa dengan Manajer. Langkah-langkah penggeraan adalah sebagai berikut:

1. Pembuatan Class Induk (Karyawan):

- Membuat class Karyawan sebagai *superclass*.
- Mendefinisikan atribut nama dan gajiPokok menggunakan *access modifier protected* agar dapat diakses langsung oleh kelas turunannya.
- Membuat method tampilInfo() untuk menampilkan informasi dasar pegawai.

2. Pembuatan Class Turunan (Manajer):

- Membuat class Manajer yang mewarisi class Karyawan menggunakan keyword extends.
- Menambahkan atribut khusus tunjangan.
- Membuat constructor yang menggunakan keyword super(...) untuk menginisialisasi atribut milik *superclass*.
- Melakukan **Overriding** pada method tampilInfo() untuk menampilkan informasi tambahan berupa tunjangan dan total pendapatan.

3. Pengujian Program (UjiKaryawan):

- Membuat objek karyawan1 (Staf Biasa) dan manajer1 (Manajer).
- Memanggil method tampilInfo() pada masing-masing objek untuk memverifikasi perbedaan perilaku (polimorfisme) antara keduanya.

B. Penjelasan Kode & Fungsi

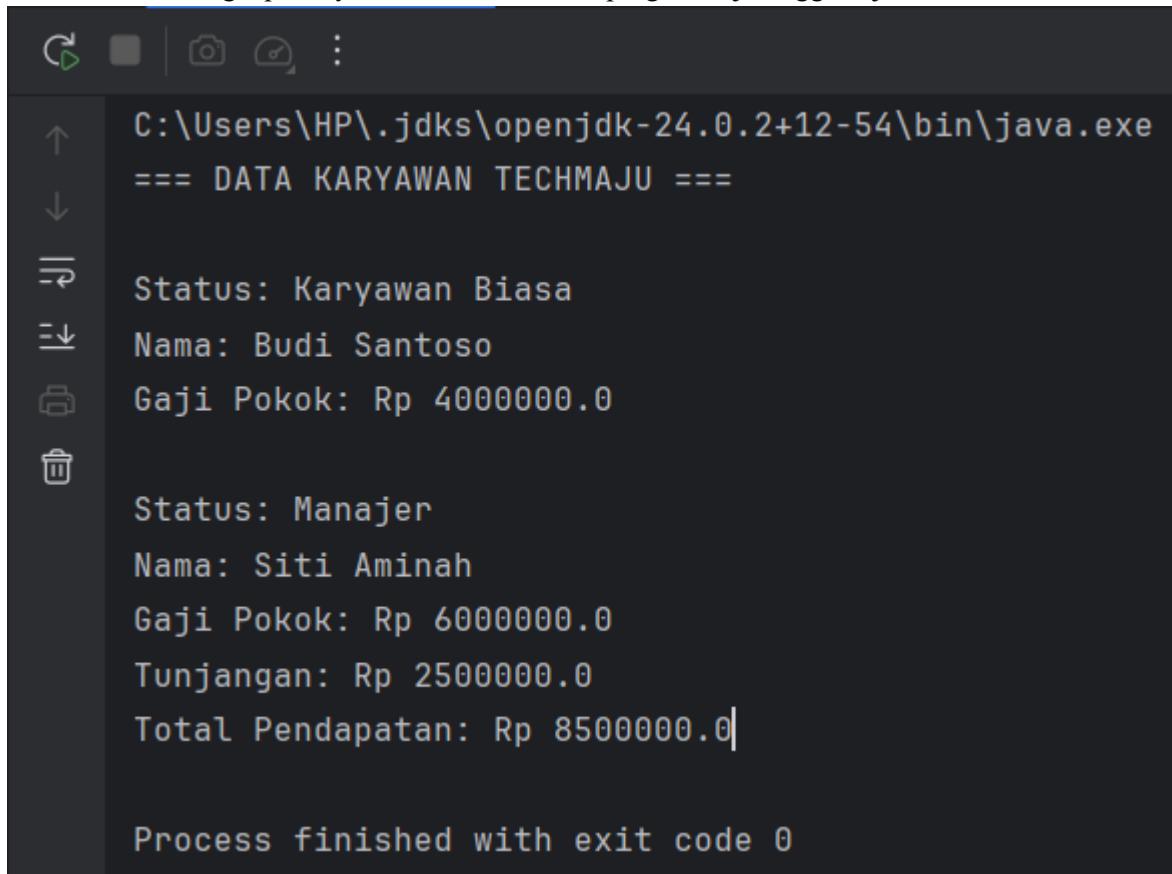
Berikut adalah analisis mendalam mengenai konsep teknis yang diterapkan dalam kode program:

1. **Inheritance (extends)** Class Manajer dideklarasikan dengan sintaks public class Manajer extends Karyawan. Hal ini menandakan bahwa Manajer adalah *subclass* yang secara otomatis mewarisi seluruh atribut dan method dari Karyawan. Keuntungannya adalah efisiensi kode (*reusability*), karena kita tidak perlu menulis ulang deklarasi nama dan gajiPokok.
2. **Keyword super**
 - **Di Constructor:** super(nama, gajiPokok) digunakan pada baris pertama constructor Manajer. Fungsinya adalah memanggil constructor Karyawan untuk memastikan atribut dasar pegawai diinisialisasi terlebih dahulu sebelum atribut khusus manajer.
 - **Di Method:** super.tampilInfo() digunakan untuk memanggil logika penampilan data dasar milik *parent class*, sehingga kode di *subclass* menjadi lebih ringkas.
3. **Method Overriding** Method tampilInfo() ditulis ulang di dalam class Manajer dengan nama dan parameter yang sama persis dengan di Karyawan. Ini disebut *Overriding*. Tujuannya adalah memodifikasi perilaku method agar sesuai dengan konteks Manajer (menampilkan tunjangan dan total gaji), tanpa mengubah kode di class induk.
4. **Access Modifier protected** Atribut nama dan gajiPokok di class Karyawan diubah menjadi protected. Modifier ini memberikan hak akses kepada class itu sendiri dan class turunannya

(subclass), sehingga Manajer bisa mengakses this.nama atau this.gajiPokok secara langsung untuk keperluan perhitungan gaji.

C. Hasil Output Program

Berikut adalah tangkapan layar/teks hasil eksekusi program UjiPengguna.java:



```
C:\Users\HP\.jdks\openjdk-24.0.2+12-54\bin\java.exe
== DATA KARYAWAN TECHMAJU ==
Status: Karyawan Biasa
Nama: Budi Santoso
Gaji Pokok: Rp 4000000.0

Status: Manajer
Nama: Siti Aminah
Gaji Pokok: Rp 6000000.0
Tunjangan: Rp 2500000.0
Total Pendapatan: Rp 8500000.0

Process finished with exit code 0
```