

Nama : Yunan Faila Sofi

NIM : H1D024033

ShiftLama : G

ShiftBaru : C

Class Karyawan

package pert3;

```
public class Karyawan {
    // Gunakan 'protected' agar variabel ini dapat diakses langsung oleh subclass
    protected String nama;
    protected double gajiPokok;

    // Constructor untuk inisialisasi nama dan gajiPokok
    public Karyawan(String nama, double gajiPokok) {
        this.nama = nama;
        this.gajiPokok = gajiPokok;
    }

    // Method untuk menampilkan informasi dasar karyawan
    public void tampilInfo() {
        System.out.println(
            "Nama: " + this.nama +
            " | Gaji Pokok: Rp " + this.gajiPokok
        );
    }
}

/**
 * Subclass (Child Class) yang mewarisi Karyawan
 */
class Manajer extends Karyawan {
    // Deklarasikan variabel tambahan khusus Manajer
    private double tunjangan;

    // Constructor untuk inisialisasi Manajer
    public Manajer(String nama, double gajiPokok, double tunjangan) {
        // Panggil constructor dari parent class (Karyawan)
        super(nama, gajiPokok);
    }
}
```

```

        this.tunjangan = tunjangan;
    }

    // Method Override tampilInfo
    // Menambahkan informasi tunjangan dan total gaji
    @Override
    public void tampilInfo() {
        // Opsi 1: Panggil method tampilInfo() dari parent class untuk mencetak nama dan gaji pokok
        // super.tampilInfo();

        // Opsi 2: Langsung mengakses variabel protected dan mencetak semua data
        double totalGaji = this.gajiPokok + this.tunjangan;

        System.out.println(
            "Nama: " + this.nama +
            " | Gaji Pokok: Rp " + this.gajiPokok +
            " | Tunjangan: Rp " + this.tunjangan
        );
        System.out.println("Total Pendapatan: Rp " + totalGaji);
    }
}

```

1. Alur Kerja Program (Workflow)

Program ini berjalan secara berurutan mulai dari *entry point* di kelas UjiKaryawan. Berikut adalah langkah-langkah logisnya:

1. Inisialisasi Program: Program dimulai dari method main di dalam kelas UjiKaryawan.
2. Pencetakan Header: Program mencetak judul "=== DATA KARYAWAN TECHMAJU ===".
3. Instansiasi Objek Parent (Karyawan):
 - Objek budi dibuat dari kelas Karyawan.
 - Constructor Karyawan dipanggil untuk mengisi variabel nama ("Budi Santoso") dan gajiPokok (4.000.000).
4. Pemanggilan Method Parent:

- Program memanggil `budi.tampilInfo()`. Karena `budi` adalah objek `Karyawan`, method yang dijalankan adalah versi dasar (hanya menampilkan nama dan gaji pokok).

5. Instansiasi Objek Child (Manajer):

- Objek `siti` dibuat dari kelas `Manajer`.
- Constructor `Manajer` dipanggil. Di dalamnya, kata kunci `super(...)` mengirim nama dan `gajiPokok` ke kelas induk (`Karyawan`), sedangkan tunjangan diatur khusus di kelas `Manajer`.

6. Pemanggilan Method Override:

- Program memanggil `siti.tampilInfo()`.
- Program mendeteksi bahwa kelas `Manajer` memiliki method `tampilInfo` sendiri (Override).
- Program menjalankan logika di dalam `Manajer`, yaitu menghitung total gaji (`gajiPokok + tunjangan`) dan menampilkannya.

2. Fungsi dan Komponen yang Digunakan

Berikut adalah rincian elemen teknis dalam kode tersebut:

a) Kelas `Karyawan` (Superclass / Parent Class)

Ini adalah kerangka dasar.

- `protected`: *Access modifier* ini digunakan pada variabel `nama` dan `gajiPokok`. Fungsinya agar variabel ini bisa diakses langsung oleh kelas turunannya (`Manajer`) tanpa harus menggunakan *getter/setter*.
- Constructor `Karyawan`: Menerima parameter input untuk mengisi data awal saat objek dibuat.
- Method `tampilInfo()`: Menampilkan data standar.

b) Kelas `Manajer` (Subclass / Child Class)

Kelas ini mewarisi sifat dari `Karyawan` namun lebih spesifik.

- `extends Karyawan`: Keyword ini untuk melakukan pewarisan. Artinya, `Manajer` otomatis memiliki variabel `nama` dan `gajiPokok` milik `Karyawan`.
- `super(nama, gajiPokok)`: Digunakan dalam constructor untuk memanggil constructor milik parent (`Karyawan`). Ini menghindari penulisan ulang kode inisialisasi.
- `@Override`: Anotasi yang menandakan bahwa method `tampilInfo()` di sini menggantikan fungsi method bernama sama di kelas parent.
- Logika Penjumlahan: Di dalam method override, dilakukan operasi aritmatika `double totalGaji = this.gajiPokok + this.tunjangan;`

c) Kelas UjiKaryawan (Driver Class)

Kelas ini berfungsi sebagai tempat pengujian.

- `public static void main`: Fungsi utama untuk mengeksekusi program.
- `new`: Keyword untuk menciptakan objek baru di memori (instansiasi).

Class UjiKaryawan

package pert3;

```
public class UjiKaryawan {
    public static void main(String[] args) {
        // Tampilkan Header
        System.out.println("=== DATA KARYAWAN TECHMAJU ===");

        // 1. Buat objek Karyawan biasa (Budi)
        // Nama: "Budi Santoso", Gaji Pokok: Rp 4.000.000
        Karyawan budi = new Karyawan("Budi Santoso", 4000000.0);

        // Tampilkan info Karyawan biasa
        System.out.println("Status: Karyawan Biasa");
        budi.tampilInfo();

        System.out.println(); // Baris baru/jarak

        // 2. Buat objek Manajer (Siti)
        // Nama: "Siti Aminah", Gaji Pokok: Rp 6.000.000, Tunjangan: Rp 2.500.000
        Manajer siti = new Manajer("Siti Aminah", 6000000.0, 2500000.0);

        // Tampilkan info Manajer
        System.out.println("Status: Manajer");
        siti.tampilInfo();
    }
}
```

3. Hasil Output Program

```
=== 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
```

Analisis Output:

- Pada bagian Budi, hanya muncul Nama dan Gaji Pokok karena ia menggunakan method asli dari Karyawan.
- Pada bagian Siti, muncul informasi tambahan (Tunjangan) dan hasil penjumlahan (Total Pendapatan) karena ia menggunakan method yang sudah dimodifikasi (Override) di kelas Manajer.