PRAKTIKUM PARADIGMA PEMROGRAMAN

MODUL 9

Disusun oleh :

Nama : Fidelia Ping

NIM : 245410012

Kelas : Informatika 1

PROGRAM STUDI INFORMATIKA

PROGRAM SARJANA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA

YOGYAKARTA

2025

MODUL 9
KELAS ABSTAK

A. TUJUAN PRAKTIKUM

Dapat membuat dan menggunakan kelas asbtrak.

B. PEMBAHASAN LISTING

PRATIKUM

-

LATIHAN

-

C. PEMBAHASAN TUGAS

Class Employee

```
abstract class Employee {
    protected String nama;
    protected String id;
public Employee(String nama, String id) {
        this.nama = nama;
        this.id = id;
    }
    public abstract double hitungGaji();
    public abstract String getJenis();
    public void tampilIdentitas() {
        System.out.println("Nama : " + nama);
        System.out.println("ID   : " + id);
        System.out.println("Karyawan: " + getJenis());
    }
    public void cetakSlipGaji() {
        System.out.println("============== Slip Gaji ==============");
        tampilIdentitas();
        System.out.println("Gaji : Rp " + hitungGaji());
        System.out.println("=======================================\n");
    }
}
```

Pembahasan program : kelas *Employee* adalah kelas abstrak yang menjadi dasar bagi semua jenis karyawan, menyimpan atribut umum berupa *nama* dan *id*, serta menyediakan konstruktor untuk menginisialisasi keduanya; kelas ini mendefinisikan dua metode abstrak, yaitu *hitungGaji()* dan *getJenis()*, yang wajib dioverride oleh subclass sesuai karakteristik masing-masing, serta menyediakan metode *tampilIdentitas()* untuk menampilkan data karyawan dan metode *cetakSlipGaji()* untuk mencetak slip gaji lengkap berisi identitas dan hasil perhitungan gaji, sehingga kelas ini berfungsi sebagai kerangka utama yang mengatur struktur dan perilaku dasar semua karyawan.

Class FullTimeEmployee

```
class FullTimeEmployee extends Employee {
    private double gajiBulanan;
    private double tunjangan;
public FullTimeEmployee(String nama, String id, double gajiBulanan, double
tunjangan) {
        super(nama, id);
        this.gajiBulanan = gajiBulanan;
        this.tunjangan = tunjangan;
    }
    @Override
    public double hitungGaji() {
        return gajiBulanan + tunjangan;
    }
    @Override
    public String getJenis() {
        return "Tetap (Full Time)";
    }
}
```

Pembahasan program : Kelas *FullTimeEmployee* adalah subclass dari *Employee* yang merepresentasikan karyawan tetap dengan dua atribut utama, yaitu *gajiBulanan* dan *tunjangan*, yang keduanya diterima melalui konstruktor bersama nama dan ID yang dikirim ke superclass; kelas ini meng-override metode *hitungGaji()* untuk menghitung total gaji berdasarkan penjumlahan gaji bulanan dan tunjangan, serta meng-override metode *getJenis()* untuk mengembalikan informasi bahwa karyawan tersebut berstatus "Tetap (Full Time)", sehingga perilakunya sesuai dengan karakteristik karyawan tetap dalam sistem.

Class DialyEmployee

```
class DailyEmployee extends Employee {
    private int hariKerja;
    private double upahHarian;
public  DailyEmployee(String  nama,  String  id,  int  hariKerja,  double
upahHarian) {
        super(nama, id);
        this.hariKerja = hariKerja;
        this.upahHarian = upahHarian;
    }
    @Override
    public double hitungGaji() {
        return hariKerja * upahHarian;
    }
    @Override
    public String getJenis() {
        return "Harian";
    }
}
```

Pembahasan program : Kelas *DailyEmployee* merupakan turunan dari *Employee* yang menggambarkan karyawan harian dengan atribut *hariKerja* dan *upahHarian* yang disimpan melalui konstruktor bersama nama dan ID yang dikirim ke superclass; kelas ini meng-override metode *hitungGaji()* untuk menghitung total gaji berdasarkan hasil perkalian jumlah hari kerja dengan upah per hari, serta meng-override metode *getJenis()* untuk mengembalikan jenis karyawan sebagai "Harian", sehingga perilakunya mencerminkan sistem penggajian karyawan harian.

Class FreelanceEmployee

```
class FreelanceEmployee extends Employee {
    private int jamKerja;
    private double ratePerJam;
public FreelanceEmployee(String nama, String id, int jamKerja, double
ratePerJam) {
        super(nama, id);
        this.jamKerja = jamKerja;
        this.ratePerJam = ratePerJam;
    }
    @Override
    public double hitungGaji() {
        return jamKerja * ratePerJam;
    }
    @Override
    public String getJenis() {
        return "Freelance";
    }
}
```

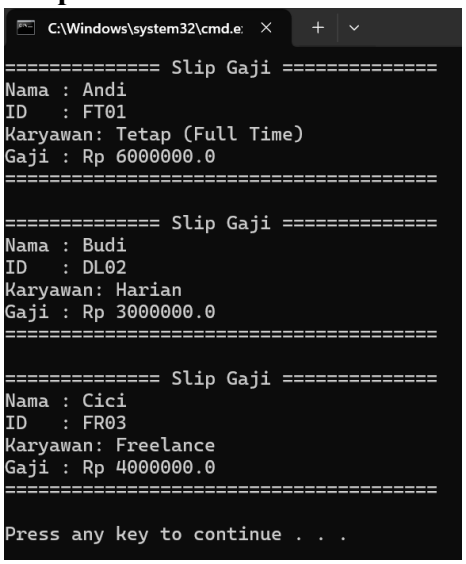
Pembahasan program : Kelas FreelanceEmployee merupakan turunan dari kelas abstrak Employee yang merepresentasikan karyawan freelance dengan dua atribut tambahan, yaitu jamKerja dan ratePerJam; melalui konstruktor, nilai-nilai tersebut disimpan bersama nama dan ID yang dikirim ke superclass, kemudian metode hitungGaji() dioverride untuk menghitung gaji berdasarkan perkalian jumlah jam kerja dengan tarif per jam, sedangkan metode getJenis() dioverride untuk mengembalikan jenis karyawan sebagai "Freelance", sehingga kelas ini menyesuaikan perilaku perhitungan gaji sesuai karakteristik karyawan freelance.

Class Main Employee

```
public class Main_Employee {
    public static void main(String[] args) {
        Employee e1 = new FullTimeEmployee("Andi", "FT01", 5000000,
1000000);
        Employee e2 = new DailyEmployee("Budi", "DL02", 25, 120000);
        Employee e3 = new FreelanceEmployee("Cici", "FR03", 80, 50000);
        e1.cetakSlipGaji();
        e2.cetakSlipGaji();
        e3.cetakSlipGaji();
    }
}
```

Pembahasan program : Program di atas membuat tiga objek karyawan dengan tipe berbeda FullTimeEmployee, DailyEmployee, dan FreelanceEmployee yang semuanya diturunkan dari kelas abstrak Employee, sehingga masing-masing memiliki cara perhitungan gaji yang berbeda sesuai aturannya, yaitu karyawan tetap dihitung dari gaji pokok dan tunjangan, karyawan harian dari jumlah hari kerja dikali upah harian, dan karyawan freelance dari jumlah jam kerja dikali rate per jam; kemudian setiap objek memanggil metode **cetakSlipGaji()** yang menampilkan nama, ID, jenis karyawan, serta total gaji, sehingga program menunjukkan penggunaan konsep pewarisan dan polimorfisme dalam OOP.

Output



Pembahasan output : Program tersebut menampilkan slip gaji tiga karyawan dengan menghitung gaji sesuai jenisnya karyawan tetap menjumlahkan gaji pokok dan tunjangan, karyawan harian mengalikan hari kerja dengan upah harian, dan freelancer menghitung gaji berdasarkan jam kerja dikali rate per jam lalu menampilkan nama, ID, jenis karyawan, serta total gajinya.

D. KESIMPULAN

Melalui praktikum kelas abstrak ini saya memahami bahwa kelas abstrak digunakan sebagai cetakan dasar yang menyatukan atribut dan perilaku umum semua objek, sementara detail perhitungannya wajib saya lengkapi di setiap subclass melalui metode abstrak, sehingga program menjadi lebih terstruktur, mudah dikembangkan, dan mendukung polimorfisme ketika saya memanggil metode yang sama tetapi menghasilkan perilaku berbeda sesuai jenis kelas turunannya.