**LAPORAN PERTEMUAN - 10**

**PEMROGRAMAN BERORIENTASI OBJEK**



**Materi : Pewarisan (Inheritance)**

**NAMA:**

**ARBAI - 223220066**

**TEKNIK INFORMATIKA-S1**

**FAKULTAS TEKNIK DAN INFORMATIKA**

**UNIVERSITAS AKI**

**2023**

**PENDAHULUAN**

Dalam kasus ini, kita memiliki sebuah perguruan tinggi (PT) yang terdiri dari dua jenis pegawai: Tenaga Pendidik (Dosen) dan Tenaga Kependidikan (Non Dosen). Setiap pegawai memiliki atribut data umum seperti id pegawai, nama pegawai, golongan, dan tanggal lahir. Atribut khusus yang dimiliki oleh pegawai dosen adalah NIDN (Nomor Induk Dosen Nasional).

Selain atribut data, setiap pegawai juga memiliki komponen gaji yang terdiri dari gaji pokok, tunjangan pegawai, dan tunjangan fungsional. Besaran gaji pokok dan tunjangan berbeda tergantung golongan dan usia pegawai. Tunjangan fungsional hanya diberikan kepada pegawai dosen dan besarannya tergantung usia pegawai.

**DASAR TEORI**

Dalam pemrograman Java, konsep pewarisan (inheritance) dapat digunakan untuk membuat hierarki class yang mirip dengan struktur hierarki dalam kasus ini. Dalam hal ini, class Pegawai merupakan superclass yang menjadi dasar untuk class Dosen dan class NonDosen.

Pewarisan memungkinkan class subclass (Dosen dan NonDosen) untuk mewarisi atribut dan metode dari superclass (Pegawai) sehingga kita tidak perlu menulis ulang kode yang sama. Class subclass dapat menambahkan atribut dan metode tambahan yang spesifik untuk jenis pegawai tersebut.

Selain itu, kita juga dapat menggunakan konstruktor untuk menginisialisasi objek dengan nilai atribut yang diberikan saat pembuatan objek. Dalam hal ini, kita dapat menggunakan konstruktor untuk menginisialisasi atribut id pegawai, nama pegawai, golongan, tanggal lahir, dan NIDN (untuk class Dosen).

Untuk menghitung total gaji, kita dapat membuat metode di class Pegawai yang menghitung gaji pokok, tunjangan pegawai, dan tunjangan fungsional (jika dosen) berdasarkan aturan yang telah ditentukan. Metode ini kemudian dapat digunakan oleh objek pegawai untuk menghitung total gaji mereka.

Dengan dasar teori ini, kita dapat melangkah ke tahap implementasi dengan membuat struktur class yang sesuai dengan UML yang diberikan.

**HASIL DAN PEMBAHASAN**

import java.time.LocalDate;

class Pegawai {

private String idPegawai;

private String namaPegawai;

private int golongan;

private LocalDate tanggalLahir;

public Pegawai(String idPegawai, String namaPegawai, int golongan, LocalDate tanggalLahir) {

this.idPegawai = idPegawai;

this.namaPegawai = namaPegawai;

this.golongan = golongan;

this.tanggalLahir = tanggalLahir;

}

public double hitungGajiPokok() {

if (golongan == 1) {

if (tanggalLahir.isBefore(LocalDate.now().minusYears(40))) {

return 4\_500\_000;

} else {

return 3\_500\_000;

}

} else if (golongan == 2) {

if (tanggalLahir.isBefore(LocalDate.now().minusYears(40))) {

return 6\_000\_000;

} else {

return 5\_000\_000;

}

}

return 0;

}

public LocalDate getTanggalLahir() {

return tanggalLahir;

}

public double hitungTunjanganPegawai() {

return 1\_000\_000;

}

public double hitungGajiTotal() {

return hitungGajiPokok() + hitungTunjanganPegawai();

}

}

class Dosen extends Pegawai {

private String nidn;

public Dosen(String idPegawai, String namaPegawai, int golongan, LocalDate tanggalLahir, String nidn) {

super(idPegawai, namaPegawai, golongan, tanggalLahir);

this.nidn = nidn;

}

@Override

public double hitungTunjanganPegawai() {

return 1\_500\_000;

}

public double hitungTunjanganFungsional() {

if (getTanggalLahir().isBefore(LocalDate.now().minusYears(40))) {

return 2\_000\_000;

} else {

return 1\_700\_000;

}

}

@Override

public double hitungGajiTotal() {

return hitungGajiPokok() + hitungTunjanganPegawai() + hitungTunjanganFungsional();

}

}

class NonDosen extends Pegawai {

public NonDosen(String idPegawai, String namaPegawai, int golongan, LocalDate tanggalLahir) {

super(idPegawai, namaPegawai, golongan, tanggalLahir);

}

}

public class Main {

public static void main(String[] args) {

LocalDate tanggalLahirDosen = LocalDate.of(1978, 5, 10);

Dosen dosen = new Dosen("D001", "John Doe", 2, tanggalLahirDosen, "1234567890");

System.out.println("Gaji Total Dosen: " + dosen.hitungGajiTotal());

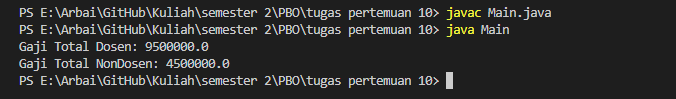
LocalDate tanggalLahirNonDosen = LocalDate.of(1990, 8, 25);

NonDosen nonDosen = new NonDosen("ND001", "Jane Smith", 1, tanggalLahirNonDosen);

System.out.println("Gaji Total NonDosen: " + nonDosen.hitungGajiTotal());

}

}



**Daftar Pustaka :**

https://blog.rosihanari.net/konsep-pewarisan-inheritance-di-java/

http://yuliana.lecturer.pens.ac.id/OOP/Inheritance/T%20-

%20Inheritance.pdf

https://drive.google.com/file/d/1ALckEppdGywqRhaqYlWLEChQBqCBMVt

2/view

https://zoneblog123.blogspot.com/2018/05/10-contoh-program-

inheritance-java.html