

MODUL 5 – Pewarisan

5.1. CAPAIAN PEMBELAJARAN

- 1 Mahasiswa memahami konsep dasar Pewarisan dalam pemrograman berorientasi objek.
- 2 Mahasiswa mampu mengimplementasikan pewarisan pada Pemrograman Berorientasi Objek.

5.2. ALAT DAN BAHAN

- 1 Seperangkat komputer lengkap/Laptop dengan koneksi internet
- 2 Web Browser (Chrome/Firefox/Opera/Edge/Safari/dll)
- 3 Aplikasi Kantor (Microsoft Office/Libre Office/WPS Office/etc)
- 4 JDK (<https://www.oracle.com/java/technologies/downloads/>)
- 5 Netbeans (<https://netbeans.apache.org/front/main/download/>)

5.3. DASAR TEORI

5.3.1. Inheritance Pewarisan

Inheritance adalah konsep dalam pemrograman berorientasi objek (OOP) di mana sebuah kelas (disebut subclass atau child class) dapat mewarisi atribut dan metode dari kelas lain (disebut superclass atau parent class). Hal ini memungkinkan kita untuk membuat hierarki kelas yang menggambarkan hubungan "is-a" (adalah).

5.3.2. Konsep Turunan

Dalam konsep turunan, saat obyek dari subclass dikonstruksi, obyek dari superclass juga ikut dikonstruksi. Jadi setiap instance dari subclass adalah juga instance dari superclass. Apabila Anda mendeklarasikan metoda dengan parameter dari tipe superclass, Anda diperbolehkan untuk memberi argumen berupa obyek subclass yang merupakan turunan dari superclass tersebut. Sedangkan apa yang dimaksud dengan polymorphism sendiri, sebenarnya sulit untuk didefinisikan. Sejalan dengan contoh yang diberikan, Anda diharapkan dapat mengerti dan memahami konsep polymorphism itu sendiri.

5.3.3. Deklarasi Inheritance

Deklarasi inheritance dilakukan saat kelas dideklarasikan, Dengan menambahkan kata kunci extends setelah deklarasi nama class, kemudian diikuti dengan nama parent class-nya. Kata kunci extends tersebut memberitahu kompiler Java bahwa kita ingin melakukan perluasan class.

Bentuk umum untuk mendeklarasikan kelas turunan adalah sebagai berikut :

```
public class anak extends induk {  
    .....  
}
```

5.3.4. Keuntungan menggunakan konsep pewarisan

Keuntungan dari Inheritance adalah Reusability. Sekali perilaku(method) didefinisikan pada super class, maka perilaku tersebut secara otomatis diwariskan ke subclass. Sehingga hanya perlu menulis method sekali dan bisa digunakan untuk semua subclass. Sekali properti/field di definisikan di superclass, maka semua properti di wariskan ke semua subclass. Superclass dan subclass berbagi properti. Subclass hanya perlu mengimplementasikan jika ada perbedaan dengan parentnya.

5.3.5. Kapan menggunakan kelas turunan?

Kelas turunan dapat kita terapkan saat kita membutuhkan perluasan dari sebuah kelas, saat detail tertentu dari suatu kelas harus kita berikan sehingga membutuhkan kelas lain yang lebih detail. Sebagai contoh dalam lingkungan universitas terdapat beberapa entitas yang yaitu mahasiswa dan dosen. Entitas tersebut dapat kita perluas dari satu entitas manusia. Sehingga kita butuh untuk membuat kelas induk (manusia) dan kelas anak (mahasiswa dan dosen).

```
public class manusia {  
    public String NIK;  
    public String Nama;  
    public String JK;  
    public int umur;  
}  
  
public class Dosen extends manusia {  
    public String NIP;  
    public String NIDN;  
}  
  
public class Mahasiswa extends manusia {  
    public String NPM;  
}
```

5.4. PRAKTIKUM

5.4.1. Perisapan

- 1) Buka Netbeans yang sudah terinstall pada komputer
- 2) Buat proyek baru dengan nama PraktikumPBO_5.

- 3) Buat package baru dengan nama praktikum5.

5.4.2. Membuat Kelas Turunan Kendaraan

- 1) Buat kelas induk "Kendaraan" dengan atribut nama, kecepatan, dan metode tampilkanInfo() untuk menampilkan informasi kendaraan.
- 2) Buat kelas turunan "Mobil" yang mewarisi dari kelas "Kendaraan", dan tambahkan atribut jumlahPintu serta metode tampilkanInfoMobil().
- 3) Lakukan overriding pada metode tampilkanInfo() di kelas Mobil untuk menampilkan informasi yang lebih spesifik.
- 4) Buat kelas turunan baru "SepedaMotor" yang juga mewarisi dari Kendaraan dengan atribut jenisMesin.

5.4.3. Studi Kasus Sederhana

Anda diminta membuat sistem untuk berbagai jenis kendaraan dari kelas kendaraan yang telah dibuat pada praktikum hari ini.

- 1) Mobil: Tambahkan atribut jumlahPintu.
- 2) SepedaMotor: Tambahkan atribut jenisMesin.
- 3) Buat objek dari masing-masing kelas dan cetak informasi menggunakan metode dari kelas induk dan turunan.

```
// Kelas Induk
class Kendaraan {
    String nama;
    int kecepatan;

    public void tampilkanInfo() {
        System.out.println("Nama Kendaraan: " + nama);
        System.out.println("Kecepatan: " + kecepatan + "
km/jam");
    }
}

// Kelas Turunan Mobil
class Mobil extends Kendaraan {
    int jumlahPintu;

    @Override
    public void tampilkanInfo() {
        super.tampilkanInfo();
        System.out.println("Jumlah Pintu: " +
jumlahPintu);
    }
}
```

```

}

// Kelas Turunan SepedaMotor
class SepedaMotor extends Kendaraan {
    String jenisMesin;

    @Override
    public void tampilkanInfo() {
        super.tampilkanInfo();
        System.out.println("Jenis Mesin: " +
jenisMesin);
    }
}

public class Main {
    public static void main(String[] args) {
        Mobil mobil = new Mobil();
        mobil.nama = "Toyota";
        mobil.kecepatan = 180;
        mobil.jumlahPintu = 4;
        mobil.tampilkanInfo();

        SepedaMotor motor = new SepedaMotor();
        motor.nama = "Yamaha";
        motor.kecepatan = 120;
        motor.jenisMesin = "2-tak";
        motor.tampilkanInfo();
    }
}

```

4) Override Metode

Dalam Java, kelas turunan dapat mengganti metode dari kelas induk (method overriding). Dengan overriding, metode di kelas turunan memiliki implementasi yang berbeda meskipun nama dan parameternya sama dengan metode di kelas induk.

```

@Override
public void tampilkanInfo() {
    // Kode baru untuk kelas turunan
}

```

Pada contoh di atas, metode tampilkanInfo() di kelas Mobil dan SepedaMotor menggantikan implementasi dari kelas Kendaraan. Keyword super digunakan untuk memanggil metode dari kelas induk.

5.5. TUGAS MODUL 3

5.5.1 Soal

- Perwarisan

- Buat kelas induk "Hewan" dengan atribut nama dan jenis.
- Buat dua kelas turunan: "Kucing" dan "Anjing".
- Implementasikan metode untuk menampilkan informasi hewan, dan lakukan overriding jika diperlukan.
- Overriding Method
 - Tambahkan metode di kelas Kucing dan Anjing yang menampilkan suara khas masing-masing hewan, dan lakukan overriding terhadap metode tampilkanInfo() di kelas induk.
- Hierarki Pewarisan
 - Buat hierarki pewarisan yang lebih kompleks dengan tiga level: kelas induk "Kendaraan", kelas menengah "KendaraanDarat", dan kelas turunan seperti "Mobil" dan "SepedaMotor".
- Push ke github masing-masing (Tautan Github cantumkan pada laporan medium)

5.5.2 **Petunjuk Pengerjaan**

a) Laporan:

- Buatlah laporan akhir sesuai dengan praktikum yang dilakukan
- Tuliskan laporan ke dalam akun masing-masing web medium
- Publikasikan laporan tersebut
- Kirimkan tautan laporan ke elita.
- **Batas Pengumpulan:** Sebelum Pertemuan Praktik Ke 6.