

LAPORAN RESMI
MODUL III
INHERINTANCE & OVERRIDE
PEMROGRAMAN BERBASIS OBJEK



NAMA	: ANISYAFAAH
N.R.P	: 220441100105
DOSEN	: FIRMANSYAH ADIPUTRA, ST., M.Cs.
ASISTEN	: KUKUH COKRO WIBOWO
TGL PRAKTIKUM:	31 MARET 2023

Disetujui : 06 April 2023
Asisten

KUKUH COKRO WIBOWO
21.04.411.00102



LABORATORIUM BISNIS INTELIJEN SISTEM
PRODI SISTEM INFORMASI
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO MADURA

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era modern seperti saat ini, dimana perkembangan teknologi menjadi semakin pesat, masyarakat sudah banyak memanfaatkan teknologi dalam kehidupan sehari-hari. Berbagai sektorpun sudah memanfaatkan teknologi untuk membantu kegiatannya. Salah satu contoh perkembangan teknologi yang paling populer adalah programming atau pembuatan program. Dunia programming makin menarik banyak orang untuk terjun di dalamnya karena prospek kerjanya yang cerah. Selain itu, pekerjaan ini akan semakin diminati dan dicari perusahaan-perusahaan besar.

Pembuat program atau programmer bertugas mengembangkan perangkat lunak atau aplikasi. Contoh pekerjaan programmer adalah mengembangkan aplikasi absensi karyawan, membuat aplikasi kasir, dan sebagainya. Selain itu, programmer juga harus menguasai beberapa bahasa pemrograman atau setidaknya satu bahasa pemrograman sebagai pondasi pembangunan aplikasi, seperti bahasa pemrograman Java. Berikut ini adalah contoh pembuatan program dengan Java dan menggunakan Inheritance dan Override.

1.2 Tujuan

- Mahasiswa mampu memahami konsep Inheritance atau pewarisan dalam Pemrograman Berorientasi Objek serta mampu mengaplikasikannya
- Mahasiswa mampu memahami konsep override dalam Pemrograman Berorientasi Objek serta mampu mengaplikasikannya.

BAB II DASAR TEORI

2.1 Pengertian Inheritance

Inheritance adalah salah satu konsep dasar dalam java. Inheritance memiliki arti yaitu warisan jadi dalam java ada class yang akan mewariskan method atau fungsi dan ada yang akan menerima warisan itu. Pemberi warisan disebut superclass sedangkan untuk penerima warisan disebut subclass, atau biasa juga disebut sebagai induk dan anak class.

2.2 Pengertian Override

Override memiliki artian yaitu mengganti, dalam java method superclass dapat diganti oleh subclass dengan cara menuliskan lagi nama method yang sama pada subclass. Override biasanya dilakukan jika method superclassnya tidak memiliki makna yang sama atau isi dari method superclass harus diubah.

Contoh program :

- Class tumbuhan sebagai superclass semua method awal dituliskan disini
- Class tumbuhan memiliki subclass yaitu pohon-mangga, dan kakor
- Tumbuhan air memiliki subclass yaitu enceng-gondok, dan teratai

Class tumbuhan :

```
public class tumbuhan
{
    //membuat method
    public void daun() {
        System.out.println("memiliki daun");
    }
    //membuat method
    public void tempat_hidup() {
```

```

        System.out.println("hidup di darat");
    }
    // membuat method
    public void jenis_buah() {
        System.out.println("memiliki buah");
    }
}

```

Class tumbuhan membuat 3 method yang nantinya akan diturunkan ke setiap subclassnya.

Class pohon_mangga dan kaktus :

```

public class pohon_mangga extends tumbuhan
{
    // bidak perlu diisi karena prinsip inheritance mewariskan method
    // dari induk ke anaknya.
}

public class kaktus extends tumbuhan
{
    // menggunakan metode override menulis ulang method induk
    public void daun() {
        System.out.println("bidak memiliki daun");
    }
}

```

Class pohon_mangga menerima semua warisan dari class tumbuhan, sedangkan kaktus menerima (override) method daun karena method daun pada class tumbuhan tidak memiliki makna yang sama dan harus diganti.

Class tumbuhan_air :

```

public class tumbuhan_air extends tumbuhan
{
    // menggunakan metode override menulis ulang method induk

```



```

public void tempat_hidup() {
    System.out.println("hidup di air");
}
// menggunakan metode override menulis ulang method induk
public void jenis_buah() {
    System.out.println("tidak memiliki buah");
}
}

```

Class tumbuhan-air menimpa class tumbuhan pada method tempat hidup dan jenis-buah.

Class enceng-gondok dan teratai :

// membuat class yang terhubung dengan class tumbuhan-air
// sebagai superclassnya

```

public class enceng-gondok extends tumbuhan-air
{

```

// tidak perlu diiri karena prinsip inheritance mewariskan method
// dari tumbuhan-air ke class ini

```

}

```

// membuat class teratai yang terhubung dengan class tumbuhan-
// air sebagai superclassnya

```

public class teratai extends tumbuhan-air
{

```

// tidak perlu diiri karena prinsip inheritance mewariskan
// method dari tumbuhan air ke class ini

```

}

```

Class enceng-gondok dan teratai terhubung dengan class tumbuhan air sehingga sudah tidak terpengaruh dengan class tumbuhan lagi, jadi method yang digunakan adalah method yang dihirup di class tumbuhan-air dan method warisan yang tidak ditulis ulang.

Running program :

Class kaktus :

Pernahkan pada inherited from tumbuhan jadi diturunkan dari tumbuhan. pada method daun ada keterangan redefined in kaktus, jadi method daun telah ditimpa atau ditulis ulang pada class kaktus.

Class pohon_mangga :

Pada inherited from tumbuhan pada semua methodnya tidak ada redefined berarti class ini mewarisi semua yang ada di class tumbuhan.

Class tumbuhan-air :

Class tumbuhan-air menimpa atau menulis ulang class tumbuhan pada bagian method jenisbuah dan tempathidup.

Class enceng-gondok dan teratai :

Pada class enceng-gondok dan teratai dapat dilihat mereka menerima method dari class tumbuhan air karena pada method jenisbuah dan tempathidup telah ditulis ulang oleh class tumbuhan air sedangkan untuk daun masih warisan dari class tumbuhan.

BAB III

TUGAS PENDAHULUAN

3.1 Soal

1. Saat ini industri mobile gaming merupakan industri yang sangat menjanjikan. Fakta ini tidak terbantahkan. Banyak pengembang gim yang awalnya fokus di desktop kini merambah ke perangkat mobile. Faktor ini jugalah yang kemudian akhirnya membuat smartphone terus berkembang dengan spesifikasi yang tinggi. Sebagai seorang developer, jelaskan secara detail penggunaan konsep Inheritance & Override dalam pembuatan (developer) sebuah game!

3.2 jawab

1. Konsep Inheritance dan Override adalah dua konsep yang sangat penting dalam pembuatan game dan software pada umumnya. Dalam inheritance, sebuah kelas dapat menerima perilaku atau sifat dari kelas lain, sedangkan dalam override, sebuah kelas dapat menerima perilaku atau sifat dari kelas induknya. Dalam pengembangan game, konsep Inheritance dan Override digunakan untuk membuat hierarki kelas yang berguna untuk mengatur perilaku objek dan karakter dalam game.
 - Dalam pengembangan game, inheritance dapat digunakan untuk membuat kelas-kelas yang terorganisasi dengan baik. Sebagai contoh, dalam game RPG, anda dapat membuat kelas lain seperti "Hero", "Warrior", "Mage", dan "Thief" yang mewarisi sifat-sifat dari kelas "Character" sebagai kelas induk. Dalam hal ini, "Character" dapat memiliki sifat-sifat seperti "health", "attack", dan defense, sedangkan kelas turunan seperti "Hero", "Warrior", dan "Thief" dapat menambahkan sifat-sifat khusus mereka sendiri, seperti "spell power" untuk mage dan "stealth" untuk Thief.

- Dalam pengembangan game, override digunakan untuk mengubah perilaku atau sifat dari kelas induk dalam kelas turunan. Sebagai contoh, anda dapat membuat kelas "Enemy" sebagai kelas induk, kemudian membuat kelas-kelas turunan seperti "Goblin", "Orc", dan "Troll".

Dalam contoh tersebut inheritance dan override membantu mengorganisir kelas-kelas dalam hierarki yang terstruktur dan memungkinkan anda untuk membuat objek dan karakter yang berbeda dengan sifat-sifat yang unik. Hal ini sangat berguna dalam pengembangan game, karena memudahkan pengembang untuk mengelola dan mengubah perilaku karakter dan objek dalam game.

BAB V IMPLEMENTASI

4.1 Soal

Buallah aplikasi perpustakaan sederhana sebagai berikut :
Masukkan data buku sebanyak x kali

1. Judul : Judul Buku
2. Penulis : Nama penulis film tersebut
3. Publisher : Perusahaan yang menerbitkan buku tersebut
4. Kategori : SU = Semua Umur, D = Dewasa, R = Remaja, A = Anak-anak
5. Stok : jumlah buku tersebut
6. Tahun Terbit : Tahun buku tersebut diterbitkan

Judul, Penulis, Publisher, Kategori, stok, dan Tahun Terbit di looping sebanyak x kali

Desainlah aplikasi perpustakaan tersebut dengan konsep inheritance dan tentukan parent class serta child class-nya.

Setelah itu, implementasikan class-class yang telah didesain dengan membuat program sederhana yang memiliki fasilitas entri data buku dan melihat daftar buku yang telah di-entri-kan.

4.2 Jawaban

a) Hasil

===== ENTRI DATA BUKU =====

Judul : Main Kundang

Penulis : Anisaa

Publisher : Gramedia Pustaka Utama

Tahun Terbit : 2001

===== Kategori =====

1. SU = Semua Umur
2. D = Dewasa
3. R = Remaja

4. A = Anak - anak

Silahkan Masukkan Pilihan Anda (1/2/3/4): 4

Stok = 5

===== HASIL ENTRI DATA =====

Judul Buku : Malin Kundang

Penulis : Anisna

Publisher : Gramedia Pustaka Utama

Tahun Terbit : 2001

Kategori : Anak - anak

Stok : 5

=====

Apakah Ingin Mengulang Kembali? (1/2):

1 = Ya / 2 = Tidak

Silahkan Masukkan Pilihan Anda :

2

Terima Kasih!

BUILD SUCCESSFULL (total time : 55 seconds)

b) Source Code

1. Sewa

```
Package Project3;
```

```
public class Sewa {
```

```
    String judul;
```

```
    String publisher;
```

```
    int stok;
```

```
    Sewa (String judul, String publisher, int stok) {
```

```
        this.judul = judul;
```

```
        this.publisher = publisher;
```

```
        this.stok = stok;
```

```
    }
```

```
    Sewa () {
```

```
    }
```

```
}
```

2. Perpustakaan

```
Package Project3;  
public class Perpustakaan extends Sewa {  
    String penulis ;  
    String kategori ;  
    int tahun terbit ;  
    perpustakaan (String penulis, String kategori, String judul,  
        String publisher, int tahun terbit, int stok) {  
        super (judul, publisher, stok);  
        this.penulis = penulis ;  
        this.kategori = kategori ;  
        this.tahun terbit = tahun terbit ;  
    }  
    Perpustakaan () {  
    }  
}
```

3. Project3

```
Package Project3;  
import java.util. Scanner;  
public class Project3 {  
    public static void main (String[] args) {  
        int user = 0, repeat;  
        do {  
            Scanner input = new Scanner (System.in);  
            Perpustakaan daftar = new Perpustakaan;  
            System.out.println (" ===== ENTRI DATA BUKU =====");  
            System.out.println (" Judul : ");  
            daftar.judul = input.nextLine();  
            System.out.println (" Penulis : ");  
            daftar.penulis = input.nextLine();  
            System.out.println (" Publisher : ");  
            daftar.publisher = input.nextLine();  
        }  
    }  
}
```

```

System.out.println (" Tahun Terbit:");
daftar.tahunTerbit = input.nextLine();
System.out.println ("1. SD = Semua Umur\n2. D =
Dewasa\n3. R = Remaja\n4. A = Anak-anak");
System.out.print ("Silahkan Masukkan Pilihan Anda :");
;
user = input.nextInt();
if (user == 1) {
    daftar.kategori = " Semua Umur";
}
else if (user == 2) {
    daftar.kategori = " Dewasa"; }
else if (user == 3) {
    daftar.kategori = " Remaja";
}
else if (user == 4) {
    daftar.kategori = " Anak-anak";
}
System.out.print (" stok : ")
daftar.stok = input.nextInt();

```

```

System.out.println ("==== HASIL ENTRI DATA ====");
System.out.println (" Judul Buku : " + daftar.judul);
System.out.println ("Penulis      : " + daftar.penulis);
System.out.println (" Publisher    : " + daftar.publisher);
System.out.println (" Tahun Terbit : " + daftar.tahun
Terbit);
System.out.println (" Kategori     : " + daftar.kategori);
System.out.println (" Stok         : " + daftar.stok);
System.out.println ("-----");

```

```

System.out.println ("Apakah Ingin Mengulang Kembali?
(1/2) : ");

```



```

System.out.println ("1 > Ya / 2 > Tidak");
System.out.print ("Silahkan Masukkan Pilihan Anda : ");
System.out.println ("\n");
repeat = input.nextInt();
}
while (repeat < 2);
System.out.println ("Terima Kasih!");
}
}

```

c) Penjelasan

Pada program untuk menampilkan beberapa data buku dalam program perpustakaan, pengguna harus memasukkan data yang terdiri dari judul buku, penulis, publisher, tahun terbit, kategori, dan stok.

BAB V

PENUTUP

5.1 Analisa

Dari hasil praktikum, praktikan menganalisa bahwa dalam membuat program menggunakan Inheritance dan Override terdapat beberapa hal yang perlu diperhatikan, yaitu penggunaan Inheritance dan Override. Konsep dari Inheritance sendiri yaitu untuk membuat struktur class pada pemrograman yang dimana struktur tersebut terdapat sebuah superclass sebagai induk kelas dan subclass sebagai anak kelas. Sedangkan konsep Override yaitu pembuatan ulang metode yang ada di subclass agar lebih spesifik.

5.2 Kesimpulan

1. Inheritance adalah proses pewarisan sifat yang dimiliki oleh class untuk diwariskan kepada kelas-kelas yang lain.
2. Override adalah proses menerima dari superclass yang diterima oleh subclass dengan cara menuliskan lagi nama method yang sama pada subclass.
3. Program yang telah diimplementasikan menggunakan konsep Inheritance dan Override.