

VERSION 1.1

MARET, 2022



PEMROGRAMAN BERORIENTASI OBJEK

INHERITANCE, PACKAGE MODUL 3

DISUSUN OLEH :

- Muhammad Rahadian Arya Saputra
- Riyan Putra Firjatullah

DIAUDIT OLEH :

- Ali Sofyan Kholimi, M.Kom.

PRESENTED BY: TIM LAB-IT

UNIVERSITAS MUHAMMADIYAH MALANG

PEMROGRAMAN BEROIRENTASI OBJEK

TUJUAN

1. Mahasiswa mampu membangun aplikasi sederhana menggunakan paradigma pemrograman terstruktur.
2. Mahasiswa mampu membangun aplikasi sederhana menggunakan paradigma object oriented.

TARGET MODUL

Mahasiswa mampu memahami & menerapkan Object Oriented Programming

PERSIAPAN SOFTWARE/APLIKASI

1. Compiler java (JDK), JRE.
2. Editor Java (NetBeans, Visual Studio Code, Gel, Eclipse, Jcreator, dll).

KEYWORDS

Inheritance

Is-a Java

Overloading

Super Keyword

Akses Modifier

Diagram UML

PERSIAPAN MATERI

1. Konsep inheritance

Inheritance ini merupakan konsep pemrograman dimana sebuah class dapat mewariskan atau menurunkan property, method atau data data yang dimilikinya kepada class yang bertindak sebagai child. Jadi class tersebut dapat mengakses data-data dari class utamanya yang bertindak sebagai parent. contohnya disini saya telah buat class utama yakni class Tranportasi.

```
class Tranportasi{  
    //Class Utama  
}
```

buat class baru nama classnya "Mobil" atau class anaknya dan untuk menggunakan konsep inheritance tambahkan kata extends lalu nama class utamanya yaitu "Tranportasi".

```
public class Mobil extends Tranportasi {  
    //Class Anak  
}
```

Jadi sekarang jika class utamanya memiliki data-data, Maka data tersebut dapat kita akses dari class Mobil.

```
public class Mobil extends Tranportasi {  
    Run | Debug  
    public static void main(String[] args) {  
        //Method Main  
    }  
}
```

Dan pastikan nama_filanya harus sama dengan nama class yang memiliki metode static voidmainnya. Pertama kita akan buat property dan method pada class utama sebagai berikut:

```
class Transportasi{
    String merk = "Toyota";
    double harga = 2000.000;
}
```

setelah itu buat objek mobil dari Class Mobil dan seperti biasa kita cetak variabel merknya

```
Mobil mobil = new Mobil();
System.out.println("Merk mobil ini adalah "+mobil.merk);
```

Jadi dari contoh di atas, class main dapat mengakses field pada class parentnya.

2. Konsep is-a

Di dalam Object Oriented, konsep dari IS-A adalah berdasar pada class turunan atau pengimplementasian interface. IS-A adalah sebuah cara untuk mengatakan “object ini adalah type dari object yang itu”. Sebagai contoh, seekor Mustang adalah- Kuda, maka dalam pembahasan Object Oriented kita dapat mengatakan “Mustang IS-A Kuda”. Subaru IS-A Mobil. Brokoli IS-A Sayuran. Kalian dapat meng-expresiasikan analogi hubungan IS-A ini dalam java melalui keyword extends (untuk inheritance)dan implements (untuk implementasi interface).

3. Method overloading

Method Overloading adalah sebuah kemampuan yang membolehkan sebuah class mempunyai 2 atau lebih method dengan nama yang sama, yang membedakan adalah parameternya. Singkatnya Method overloading adalah membuat beberapa method dengan nama yang sama, tapi dibedakan dari jumlah dan/atau tipe parameter.

- Jumlah parameter
- Tipe data dari parameter
- Tipe data dari parameter

4. Keyword super

Keyword super merupakan keyword yang digunakan untuk mengakses superclass. Di Java, keyword ini dapat digunakan untuk melakukan beberapa hal, seperti berikut ini:

- Mengakses constructor superclass oleh subclass.
- Menunjuk anggota superclass oleh subclass.
- Mengakses method superclass oleh subclass.

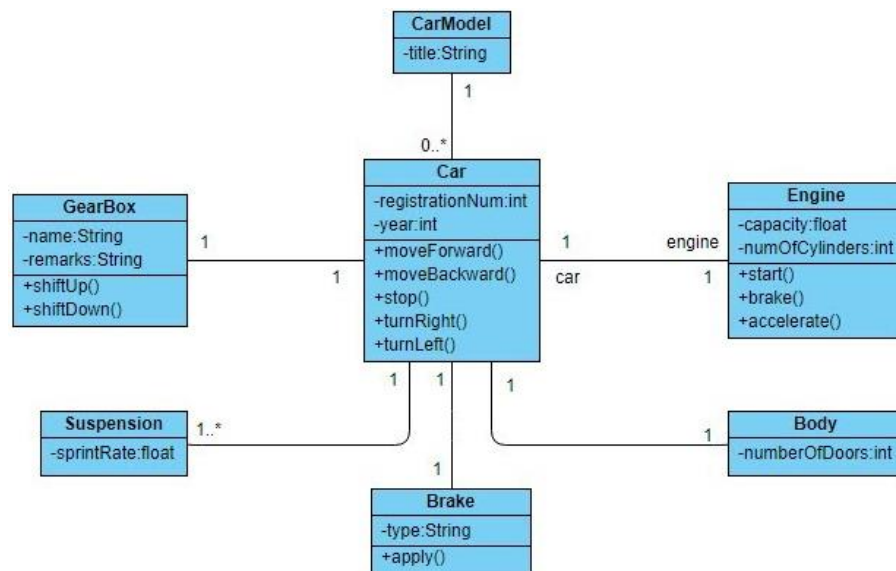
5. Akses modifier

Access Modifier adalah sebuah “hak akses” yang diberikan kepada sebuah variabel/method/class dengan tujuan untuk menjaga integritas dari data tersebut ketika ingin diakses object lain. Hak akses sendiri diberikan oleh pembuat program sebagai salah satu bentuk implementasi dari OOP itu sendiri.

Fungsi dari access modifier pada Java adalah untuk membatasi scope dari sebuah class, constructor, variabel, method, atau anggota data lainnya yang terdapat dalam suatu program Java.

Modifier	Class	Package	Subclass	World
public	✓	✓	✓	✓
protected	✓	✓	✓	✗
no modifier*	✓	✓	✗	✗
private	✓	✗	✗	✗

6. Diagram UML



UML (Unified Modelling Language) adalah suatu metode dalam pemodelan secara visual yang digunakan sebagai sarana perancangan sistem berorientasi objek. UML juga dapat didefinisikan sebagai suatu bahasa standar visualisasi, perancangan, dan pendokumentasian sistem, atau dikenal juga sebagai bahasa standar penulisan blueprint sebuah software.

MATERI PRAKTIKUM

1. Overloading

```
class Penjumlahan{

    public void jumlah(int a, int b){
        System.out.println("int a + int b : " + (a + b));
    }

    public void jumlah(double a, int b){
        System.out.println("double a + int b : " + (a + b));
    }

    public void jumlah(int a, double b){
        System.out.println("int a + double b : " + (a + b));
    }

    public void jumlah(double a, double b){
        System.out.println("double a + double b : " + (a+b));
    }

}
```

Memanggil pada kelas Main:

```
public class Overloading {
    Run | Debug
    public static void main(String[] args) {
        Penjumlahan tambah = new Penjumlahan();
        tambah.jumlah(2, 4);
        tambah.jumlah(2.9,1);
        tambah.jumlah(5,9.2);
        tambah.jumlah(2.2, 3.3);
    }
}
```

2. Is-a

Membuat kelas Parent:

```
class Parent{
    String nama;
    public void tampilNama(){
        nama = "Pajero";
        System.out.println("Mobil : "+nama);
    }
}
```

Membuat kelas Child yang extends ke kelas Parent:

```
class Child extends Parent{
    int kecepatan;
    public void tampilKecepatan(){
        kecepatan = 2442;
        System.out.println("Kecepatan : "+kecepatan+" cc");
    }
}
```

Kelas Main:

```
public class IsA {
    Run | Debug
    public static void main(String[] args) {
        Parent p = new Parent(); //Membuat objek parent
        p.tampilNama();

        Child c = new Child(); //Membuat objek child
        c.tampilNama(); //Contoh relasi Is-a
        c.tampilKecepatan();
    }
}
```

3. Keyword Super

Membuat Kelas Utama

```
class luasPersegi{
    public int luas(int sisi) {
        return (sisi*sisi);
    }

    public void printLuas(int sisi) {
        System.out.println("Luas persegi adalah : "+this.luas(sisi));
    }
}
```

Membuat kelas Anak yang extends pada kelas Utama

```
class kelilingPersegi extends luasPersegi{ //extends ke kelas parent (luasPersegi)
    public int keliling(int sisi) {
        return (sisi*4);
    }
    public void printKeliling(int sisi) {
        System.out.println("Keliling persegi adalah : "+this.keliling(sisi));
    }
    //Memanggil method pada class parent (luasPersegi) menggunakan super keyword
    public void printLuasPersegi(int sisi) {
        System.out.println("Luas persegi adalah : "+super.luas(sisi));
    }
}
```

Panggil method pada kelas Utama dan kelas Anak pada Kelas Main

```

public class Super {
    Run | Debug
    public static void main(String[] args) {
        kelilingPersegi persegi = new kelilingPersegi(); // Membuat objek persegi
        int[] sisi = new int[1];

        sisi[0] = 10;

        persegi.printLuas(sisi[0]);
        persegi.printKeliling(sisi[0]);
        persegi.printLuasPersegi(sisi[0]);
    }
}

```

Berikut referensi untuk menambah pemahaman kalian mengenai Turunan :

<https://www.petanikode.com/java-oop-inheritance/>

LEMBAR KERJA

KEGIATAN 1

Pilihlah salah satu dari soal berikut:

- A. Buatlah program dengan Class Parent Hero (Pahlawan) yang menerapkan konsep inheritance dengan ketentuan class parent menggunakan overloading constructor (minimal 4 constructor dalam 1class).
- B. Buatlah program dengan class parent yang memiliki 3 buah sub class atau child class yang mana di class mainnya menggunakan atribut dan method pada class parentnya menggunakan inheritance.

KEGIATAN 2

Buatlah satu class child di package yang berbeda (menerapkan konsep is_a) dengan class parent dan gunakan keyword super dan terapkan overloading constructor. Serta buatlah class drivernya (Main method) berada di dalam package yang sama dengan parent class. Gunakan modifier private, protected dan public (Opsional).

KEGIATAN 3

Berdasarkan kegiatan 1 dan 2 buatlah class diagram UML nya, class diagram harus memuat konsep inheritance di dalamnya. Dimana terdapat minimal 5 class child yang saling berhubungan dengan class parentnya.

Jelaskan kepada asisten kelebihan dan kekurangan masing-masing paradigm.

*Jika ada source code yang identik, maka akan dilakukan pengurangan nilai.

RUBRIK PENILAIAN

Aspek Penilaian	Poin
Kegiatan 1 A Kegiatan 1 B	20 25
Kegiatan 2	35
Kegiatan 2 (Opsional)	10
Kegiatan 3	10
Total	100