

VERSION 2.0
FEBRUARI, 2021



PEMROGRAMAN BERORIENTASI OBJEK

INHERITANCE, PACKAGE MODUL 3

TIM PENYUSUN:

- GALIH WASIS WICAKSONO S.KOM,.M.CS.
- FARLI NAHRUL JAVIER
- MUHAMMAD NUR ICHSAN

PRESENTED BY: LAB. INFORMATIKA
UNIVERSITAS MUHAMMADIYAH MALANG

PEMROGRAMAN BERORIENTASI OBJEK

CAPAIAN PEMBELAJARAN MATA KULIAH

Mahasiswa mengetahui :

1. Konsep inheritance
2. Konsep is-a, berdasarkan hierarki inheritance
3. Method overloading
4. Pembuatan package

SUB CAPAIAN PEMBELAJARAN MATA KULIAH

Mahasiswa dapat melakukan :

1. Pemanggilan constructor milik superclass dengan keyword super
2. Pemanggilan method milik superclass dengan keyword super
3. Mengakses class dari package yang berbeda
4. Mengakses modifier dari berdasarkan package
5. Mengakses modifier dari berdasarkan inheritance dan package

KEBUTUHAN HARDWARE & SOFTWARE

1. Compiler java (JDK), JRE.
2. Editor Java (Intellij IDEA, NetBeans, Gel, Eclipse, Jcreator, dll).

MATERI POKOK

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 Gadget

```
class Gadget {  
    // Class Utama  
}
```

buat class baru nama classnya Laptop atau class anaknya dan untuk menggunakan konsep inheritance tambahkan kata extends lalu nama class utamanya

```
class Laptop extends Gadget {  
    // class anak  
}
```

jadi sekarang jika class utamanya memiliki data-data dan data tersebut dapat kita akses dari class Laptop.

```
class Laptop extends Gadget {
    public static void main(String[] args) {
        // method main
    }
}
```

dan pastikan nama filenya harus sama dengan nama class yang memiliki metode static void mainnya. Pertama kita akan buat property dan method pada class utama sebagai berikut

```
class Gadget {
    String merk = "lenovo";
    int harga = 30000;
}
```

setelah itu buat objek laptop dari Class Laptop dan seperti biasa kita cetak variabel merknya

```
Laptop laptop = new Laptop();
System.out.println("Merk Laptop Ini "+laptop.merk);
```

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-ekspresikan 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.

Pada method overloading perbedaan parameter mencakup :

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

Method Overloading juga dikenal dengan sebutan Static Polymorphism. Berikut ini contoh Class yang melakukan Overloading.

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. Dengan adanya Access Modifier, kita dapat membatasi resource-resource mana saja yang dapat diakses oleh object tertentu, turunannya, ataupun oleh method tertentu.

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

MATERI PRAKTIKUM

1. Overloading

Contoh overloading pada parameter :

```
public class ContohOverloading {
    public void jumlah (int a, int b){
        System.out.println("Jumlah 2 angka =" + (a + b));
    }
    //overloading perbedaan jumlah parameter
    public void jumlah (int a, int b, int c){
        System.out.println("Jumlah 3 angka =" + (a + b + c));
    }
    //overloading perbedaan tipe data parameter
    public void jumlah (double a, int b){
        System.out.println("Jumlah 2 angka (double+int) = " + (a + b));
    }
    //overloading perbedaan urutan tipe data parameter
    public void jumlah (int b, double a){
        System.out.println("Jumlah 2 angka (int+double) = " + (a + b));
    }
}
```

Penggunaan pada class utama :

```
public class PenggunaanOverloading {
    public static void main(String[] args) {
        ContohOverloading co = new ContohOverloading();
        co.jumlah(83,32);
        co.jumlah(34,454,432);
        co.jumlah(34.43,34);
        co.jumlah(28,33.23);
    }
}
```

2. Is-a

Contoh penggunaan relasi is-a Class Parent

```
package is_a;

public class Parent {
    String nama;
    protected void tampilNama() {
        nama = "Nama Saya";
        System.out.println("Tampil Nama : "+nama);
    }
}
```

Class Child turunan dari class Parent

```
package is_a;

public class Child extends Parent{
    String jeniskelamin;
    public void tampilJK(){
        jeniskelamin = "Laki-laki";
        System.out.println("Tampil enis eniselamin : "+jeniskelamin);
    }
}
```

Class main

```
package is_a;

public class main {
    public static void main(String[] args) {
        Parent p = new Parent(); //object class Parent
        System.out.println("---Class Parent---");
        p.tampilNama(); // tampil output dari method Parent

        Child c = new Child(); //object class Child
        System.out.println("---Class Child---");
        c.tampilNama(); //contoh relasi is-a
        c.tampilJK();
    }
}
```

3. Keyword Super

```
public class buah { //super class

    void pesan() {
        //method dari super class
        System.out.println("Rasa buahnya manis");
    }

    public static void main(String[] args) {
        mangga m = new mangga();
        m.pesan();
    }
}
```

Class buah merupakan super class

```
public class mangga extends buah{
    void pesan() {
        super.pesan(); //memanggil method pada class Super
        System.out.println("Rasa buahnya asam"); //menampilkan method pesan pada class turunan
    }
}
```

Class mangga adalah sub class dari class buah

LEMBAR KERJA

KEGIATAN 1

Pilihlah salah satu dari soal berikut:

- Buatlah program dengan Class Parent **Company** yang menerapkan konsep inheritance dengan ketentuan class parent menggunakan overloading constructor (minimal 4 constructor dalam 1 class).
- Buatlah program dengan Class Parent dengan nama class sembarang (tidak ada ketentuan nama class) yang menerapkan konsep 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 didalamnya. dimana terdapat minimal 5 class child yang saling berhubungan dengan class parentnya.

Catatan:

- Apabila terdapat source code yang identik, maka akan ada pengurangan nilai

RUBRIK PENILAIAN

KRITERIA	NILAI (%)
Kegiatan 1 A Kegiatan 1 B	40 20
Kegiatan 2	30
Kegiatan 2 (Opsional)	10
Kegiatan 3	20
Total	100