

VERSION 2.0
FEBRUARI, 2023



PEMROGRAMAN BERORIENTASI OBJEK

MODUL 2 – INHERITANCE DAN PACKAGE

DISUSUN OLEH:

Muhammad Nizar Zulmi Rohmatulloh

Jody Yuantoro

DIAUDIT OLEH:

Aminudin, S.Kom., M.Cs.

PRESENTED BY: TIM LAB. IT

UNIVERSITAS MUHAMMADIYAH MALANG

PEMROGRAMAN BERORIENTASI 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

1. Java Development Kit.
 2. Text Editor / IDE (Visual Studio Code, Netbeans, IntelliJ IDEA, atau yang lainnya).
-

KEYWORDS

- Inheritance
- Package
- Overloading
- Overriding
- Super Keyword

TEORI

- **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.

Contoh inheritance menggunakan keyword extends:

```
// Parent Class
public class parentClass {
    public void parentMethod(){
        System.out.println("This is from Parent Class");
    }
}
```

```
// Child Class
public class childClass extends parentClass {
    public static void main(String[] args){
        parentMethod();
    }
}
```

- **Package**

Package adalah sebuah cara untuk mengelompokkan class. Tujuannya menghindari bentrok nama class (jika ada yang bernama sama) serta memudahkan pengelolaan kode program, terutama untuk aplikasi besar, kita bisa membuat nama class yang sama selama berada di dalam package yang berbeda.

- **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.

- Contoh Overloading jumlah dari parameter

```
// Parent Method
void namaMethod(int params1){
}
```

```
// Overloading Method
void namaMethod(int params1, int params2){
}
```

- Contoh Overloading tipe data dari parameter

```
// Parent Method
void namaMethod(int params1){
}
```

```
// Overloading Method
void namaMethod(string params1){
}
```

- **Method Overriding**

Method overriding adalah sebuah kemampuan yang membolehkan untuk mereplikasi body method utama, yang membedakan parent method dengan override method terletak pada isi method nya. Singkatnya method Override adalah membuat method dengan

parameter yang sama, tapi dibedakan dari isi kode dalam method tersebut. Method Override memiliki ciri ciri dengan adanya anotasi `@Override` pada atas nama method nya

- Contoh Override

```
// Parent Method
void namaMethod(int params1){
    System.out.println("This is from Parent method");
}
```

```
// Override Method
@Override
void namaMethod(int params1){
    System.out.println("This is from Override method");
}
```

- **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.

Contoh penggunaan keyword super:

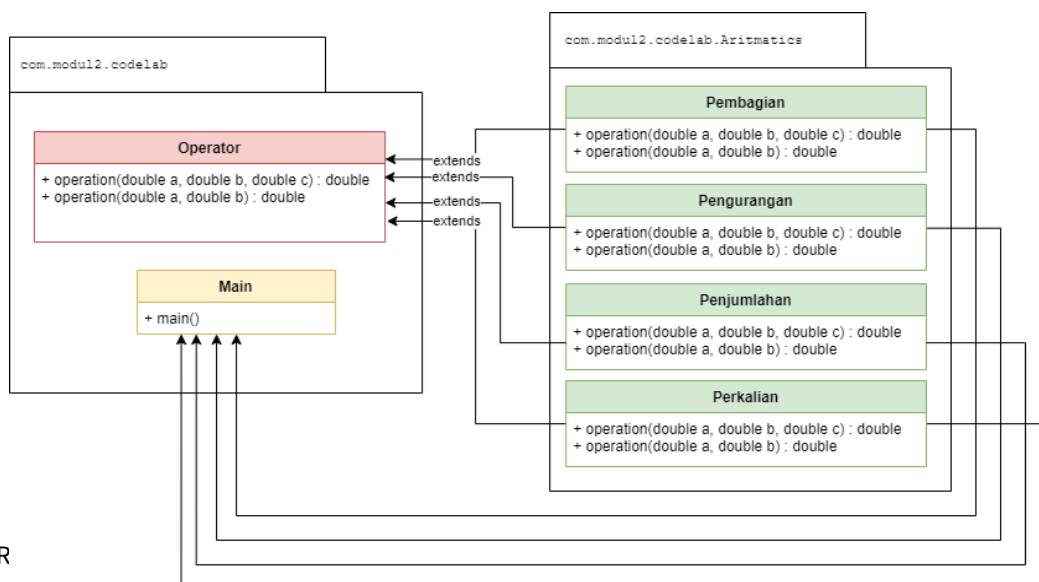
```
// Parent Class
public class parentClass {
    public void parentMethod(){
        System.out.println("This is from Parent
Class");
    }
}
```

```
// Child Class
public class childClass extends parentClass {
    @Override
    public void parentMethod(){
        super.parentMethod();
        System.out.println("This is from Child
Class");
    }
}
```

CODELAB

Implementasi Package, Override dan Overload

- Buatlah package **com.modul2.codelab** sebagai main package / root, dan package **operator** di dalam package **com.modul2.codelab**.
- Buat class **Driver dan Operator** pada package **com.modul2.codelab**,
- Pada class **Operator** buat method operation, dan buat juga untuk method overload parameter nya.
- Pada class Pembagian, Pengurangan, Penjumlahan, dan Perkalian extends dari class Operator, lalu buat method override dari method Operator dengan rumus yang sesuai
- Pada class Main isi main method sebagai driver method yang sudah dibuat.



TUGAS

Buatlah sebuah program menghitung Luas Keliling dan Volume dari sebuah bangun datar dan bangun ruang dengan spesifikasi yang ada pada diagram UML dibawah, yang dimana akan terdapat **3 package** yaitu :

- **modul2**

Package ini merupakan root package dari program ini yang akan berisi class driver main, dan class parent LKBangun2D, dan LKBangun3D.

- **Main**, class tersebut akan berfungsi sebagai driver dari class dan method pada package-package yang telah dibuat, method driver tersebut perlu diorganisir dengan cara dipecah ke dalam method sebagai berikut :
 - **hitung2D()**, method tersebut akan berisi menu / driver untuk handle class dan method perhitungan luas dan keliling yang ada pada dalam package **modul2.bangun2d**.
 - **hitung3D()**, method tersebut akan berisi menu / driver untuk handle class dan method menghitung volume yang ada pada dalam package **modul2.bangun3d**.
- **LKBangun2D**, class tersebut akan diturunkan ke pada class yang terdapat pada package **modul2.bangun2D**, class tersebut memiliki beberapa method sebagai berikut :
 - **infoBangun()**, method ini difungsikan sebagai output dari sebuah bangun datar dari class child, method ini perlu di overload karena setiap bangun memiliki sisi yang berbeda beda, misalkan persegi memiliki 1 atribut yaitu sisi, sedangkan persegi panjang memiliki 2 atribut yaitu panjang dan lebar, jadi pada class **Persegi** perlu method yang memiliki 1 parameter untuk

sisi, sedangkan pada class **PersegiPanjang** perlu memiliki method yang mempunyai 2 parameter untuk Panjang dan Lebar.

- **hitungKeliling()**, method ini difungsikan sebagai tempat untuk pengoperasian perhitungan keliling dari class child, method ini perlu dioverride sesuai dengan rumus yang sesuai dari class child nya, pada method parent hanya terdapat kode output untuk menampilkan keliling, yang nantinya pada class child dapat memanggilnya dengan keyword **super**.
- **hitungLuas()**, method ini difungsikan sebagai tempat untuk pengoperasian perhitungan luas dari class child, method ini perlu dioverride sesuai dengan rumus yang sesuai dari class child nya, pada method parent hanya terdapat kode output untuk menampilkan luas, yang nantinya pada class child dapat memanggilnya dengan keyword **super**.
- **LKBangun3D**, class tersebut akan diturunkan ke pada class yang terdapat pada package **modul2.bangun3D**, class tersebut memiliki beberapa method sebagai berikut :
 - **infoBangun()**, method ini difungsikan sebagai output dari sebuah bangun ruang dari class child, method ini perlu di overload karena setiap bangun memiliki atribut yang berbeda beda, misalkan Tabung memiliki 2 atribut yaitu jari jari dan tinggi, sedangkan bola hanya memiliki 1 atribut yaitu jari jari, jadi pada class **Tabung** perlu method yang memiliki 2 parameter untuk jari jari dan tinggi, sedangkan pada class **Bola** cukup membutuhkan method yang mempunyai 1 parameter untuk jari - jari.
 - **hitungVolume()**, method ini difungsikan sebagai tempat untuk pengoperasian perhitungan volume dari class child, method ini perlu dioverride sesuai dengan rumus yang sesuai dari class child nya, pada

method parent hanya terdapat kode output untuk menampilkan volume, yang nantinya pada class child dapat memanggilnya dengan keyword **super**.

- **modul2.bangun2d**

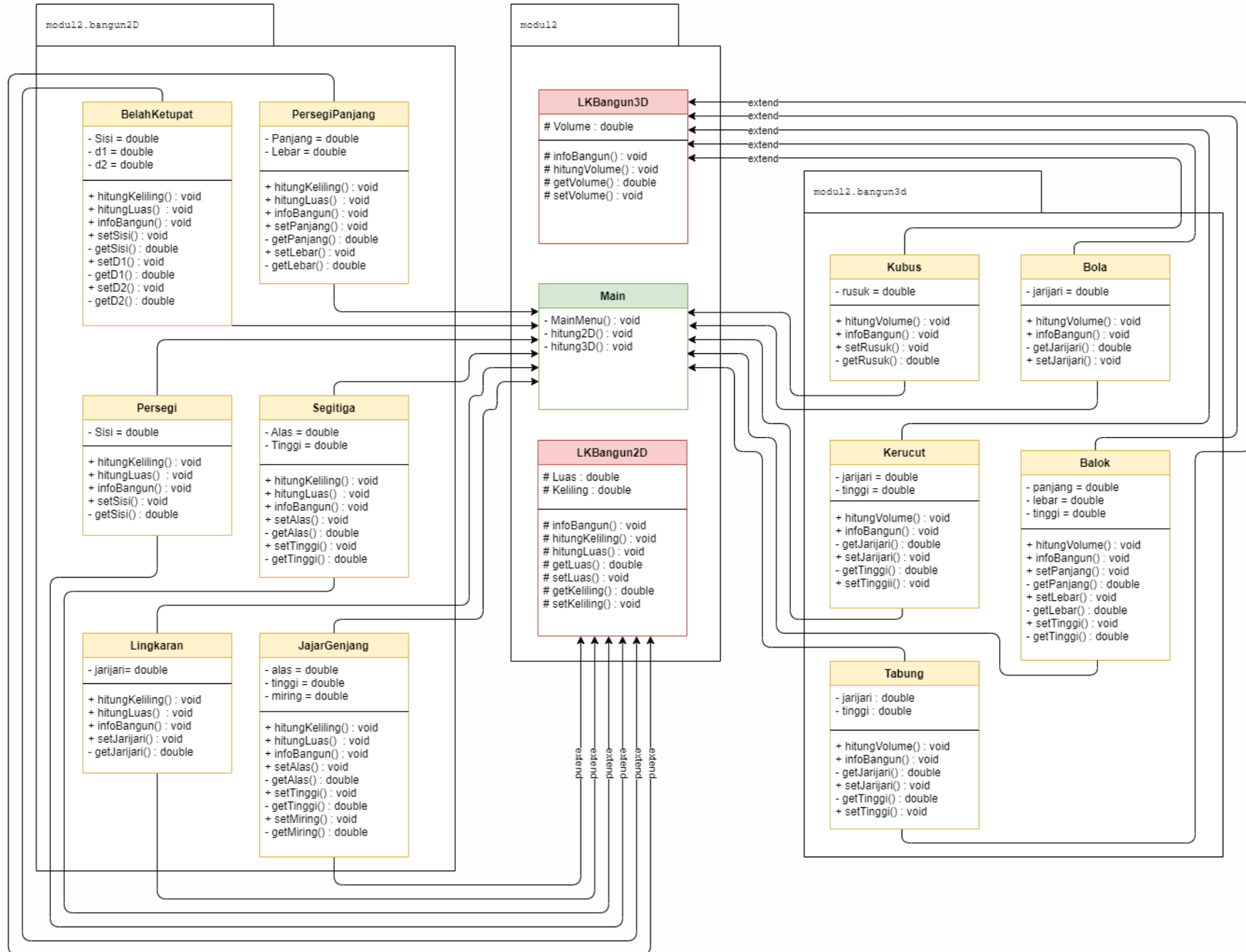
Package ini akan berisi berbagai class bangun datar yang extends ke class parent **LKBangun2D** pada package **modul2**, class tersebut memiliki method **@Override** dari class parent, yaitu :

- **infoBangun()**, yaitu method override yang digunakan untuk mengoutput hasil dari data yang telah kita input. Yang dimana pada class parentnya juga memiliki parent method yang telah dioverload parameternya sesuai kebutuhan dari child class nya.
- **hitungKeliling()**, yaitu method override yang digunakan untuk menghitung keliling dengan rumus yang sesuai dengan nama bangun pada class tersebut sekaligus menampilkan output hasil dari perhitungan tersebut.
- **hitungLuas()**, yaitu method override yang digunakan untuk menghitung luas bangun dengan rumus yang sesuai dengan nama bangun pada class tersebut.

- **modul2.bangun3d**

Package ini akan berisi berbagai class bangun datar yang extends ke class parent **LKBangun3D** pada package **modul2**, class tersebut memiliki method **@Override** dari class parent, yaitu :

- **infoBangun()**, yaitu method override yang digunakan untuk mengoutput hasil dari data yang telah kita input. Yang dimana pada class parentnya juga memiliki parent method yang telah dioverload parameternya sesuai kebutuhan dari child class nya.
- **hitungVolume()**, yaitu method override yang digunakan untuk menghitung volume bangun ruang dengan rumus yang sesuai dengan nama bangun pada class tersebut sekaligus menampilkan output hasil dari perhitungan tersebut.



Contoh Output :

```

Run
6. Belah Ketupat
0. Kembali
Pilih menu : 0
=====
Selamat Datang di Aplikasi LKV
Menghitung Luas Keliling Volume Akan Menjadi Mudah
=====
1. Menghitung Bangun 2 Dimensi
2. Menghitung Bangun 3 Dimensi
0. Keluar
Pilih menu :

Run
0. Keluar
Pilih menu : 1
Aplikasi Penghitung Luas & Keliling Bangun Datar
1. Persegi
2. Persegi Panjang
3. Segitiga
4. Lingkaran
5. Jajar Genjang
6. Belah Ketupat
0. Kembali
Pilih menu :

Run
2. Menghitung Bangun 3 Dimensi
0. Keluar
Pilih menu : 2
Aplikasi Penghitung Volume Bangun Ruang
1. Kubus
2. Balok
3. Tabung
4. Bola
5. Kerucut
0. Kembali
Pilih bangun ruang yang ingin di hitung :

Run
5. Kerucut
0. Kembali
Pilih bangun ruang yang ingin di hitung : 2
Masukkan Panjang : 10
Masukkan Lebar : 40
Masukkan Tinggi : 56
=====
Hasil Bangun Balok
Panjang : 10.0—Lebar : 40.0—Tinggi : 56.0—
Volume dari Bangun Balok tersebut adalah 22400.0
Kembali ke menu ? Y/n :

```

RUBRIK PENILAIAN

Aspek Penilaian	Poin
Codelab	20
Tugas	35
Pemahaman	45
Total	100