

UJIAN TENGAH SEMESTER PRAKTIKUM RPL BO GENAP 2022/2023

Hari/Tgl : Jumat / 31 Maret 2023

Waktu: 15.30 – 17.30 WIB

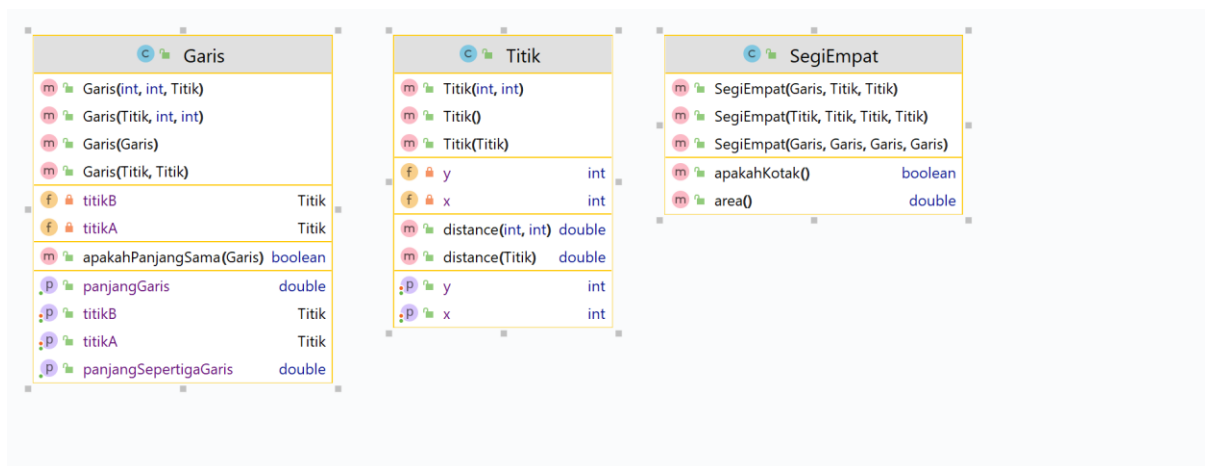
Dosen : Antonius Rachmat C., S.Kom., M.Cs. dan I Kadek Dendy Senapatha, S.T, M.Eng

Tempat : Laboratorium A, B, dan H

PETUNJUK UMUM:

1. Kerjakan soal-soal berikut, perhatikan point nilai masing-masing soal.
2. Silahkan **memilih** soal yang ingin anda kerjakan, perhatikan poinnya, **poin maksimal 100**. Misalnya anda hanya mengerjakan soal 1 berarti nilai maksimal 70, jika anda mengerjakan soal 1 dan 3 maka nilai maksimal $70+30 = 100$.
3. Tuliskan nomor soal yang Anda pilih dan kerjakan pada **README.MD** di Repo.

PROJECT TITIK, GARIS, DAN SEGIEMPAT (POIN 70)



Buatlah sebuah class Titik, Garis, dan Kotak. Soal ini memiliki topik relasi antar kelas (dependency). Berikut adalah spesifikasi dari class Titik yang harus dipenuhi:

Class **Titik** memiliki dua atribut, yaitu **x** dan **y** yang masing-masing bertipe integer.

Class **Titik** memiliki tiga constructor, yaitu:

- Titik(int, int), langsung mengisi nilai x dan y.
- Titik(), berarti membuat sebuah titik pada koordinat (0,0).
- Titik(Titik), membuat titik berdasarkan Titik lain yang sudah ada sebelumnya.

Class **Titik** memiliki beberapa method, yaitu:

- Method **distance**(int, int), untuk menghitung jarak antara titik sekarang dengan suatu titik di koordinat (x,y). Rumusnya jarak = $\sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2}$
- Method **distance**(Titik), untuk menghitung jarak antara titik sekarang dengan suatu titik yang lain.
- **Setter** dan **getter** untuk atribut x dan y.

Untuk class **Garis**, berikut ini adalah spesifikasi yang harus dipenuhi:

Class Garis memiliki 2 atribut yaitu **titikA** dan **titikB** yang bertipe **Titik**

Class Garis memiliki empat constructor yaitu:

- `Garis(Titik, Titik)` untuk mengisi titik A dan titik B
- `Garis(Garis)` untuk mengisi titik dengan Garis lain
- `Garis(Titik,int,int)` untuk mengisi titik A dengan parameter pertama dan titik B dengan koordinat x,y
- `Garis(int,int,Titik)` untuk mengisi titik A dengan koordinat x,y dan titik B dengan Titik

Method dari class **Garis** adalah:

- **`getPanjangGaris()`** untuk menghitung panjang garis tersebut. Rumus sama dengan distance.
- **`getPanjangSepertigaGaris()`** untuk menghitung $1/3 * \text{panjang garis tersebut}$.
- **`apakahPanjangSama(Garis)`** untuk mengetahui apakah panjang garis sama dengan garis yang input parameter (return boolean true / false).
- Lengkapi dengan **getter** dan **setter** untuk atribut `titikA` dan `titikB`.

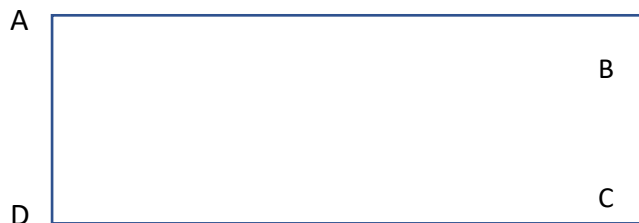
Class **SegiEmpat** memiliki 4 atribut, yaitu **`garisA`**, **`garisB`**, **`garisC`** dan **`garisD`** yang merupakan keempat Garis dari segi empat tersebut. Setiap atribut bertipe **Garis**.

Class **SegiEmpat** memiliki 3 constructor:

- `SegiEmpat(Garis,Garis,Garis,Garis)` untuk mengisi 4 garis yang membentuk segiempat
- `SegiEmpat(Titik,Titik,Titik,Titik)` untuk menerima 4 titik sudut yang membentuk segiempat
- `SegiEmpat(Garis,Titik,Titik)` untuk menerima 1 buah garis (`garisA` ke `B`) dan 2 buah titik (`titik c` dan `titik d`) yang membentuk segi empat

Class **SegiEmpat** memiliki method **`area()`** untuk menghitung luas segiempat (rumusnya adalah $\text{panjang} * \text{lebar}$) dan **`apakahKotak()`** untuk mengetahui apakah segi empat itu berbentuk kotak (bujur sangkar) atau bukan (return boolean true/false). Lengkapi dengan **getter** dan **setter** untuk seluruh atribut `garisA`, `garisB`, `garisC`, dan `garisD`.

Contoh ilustrasi bentuk segi empat:



- Ketentuan project maven:
 - `GroupId = com.rplbo.uts1`
 - `ArtifactId = SegiEmpat`
 - `Version = 1.0-SNAPSHOT`
 - `Name = SegiEmpat`

Class Main yang diberikan pada project dapat dikembangkan untuk dicoba. Class `AppTest` juga sudah disediakan. Saat mencoba, class `AppTest` bisa dikomen dulu.

Poin Penilaian:

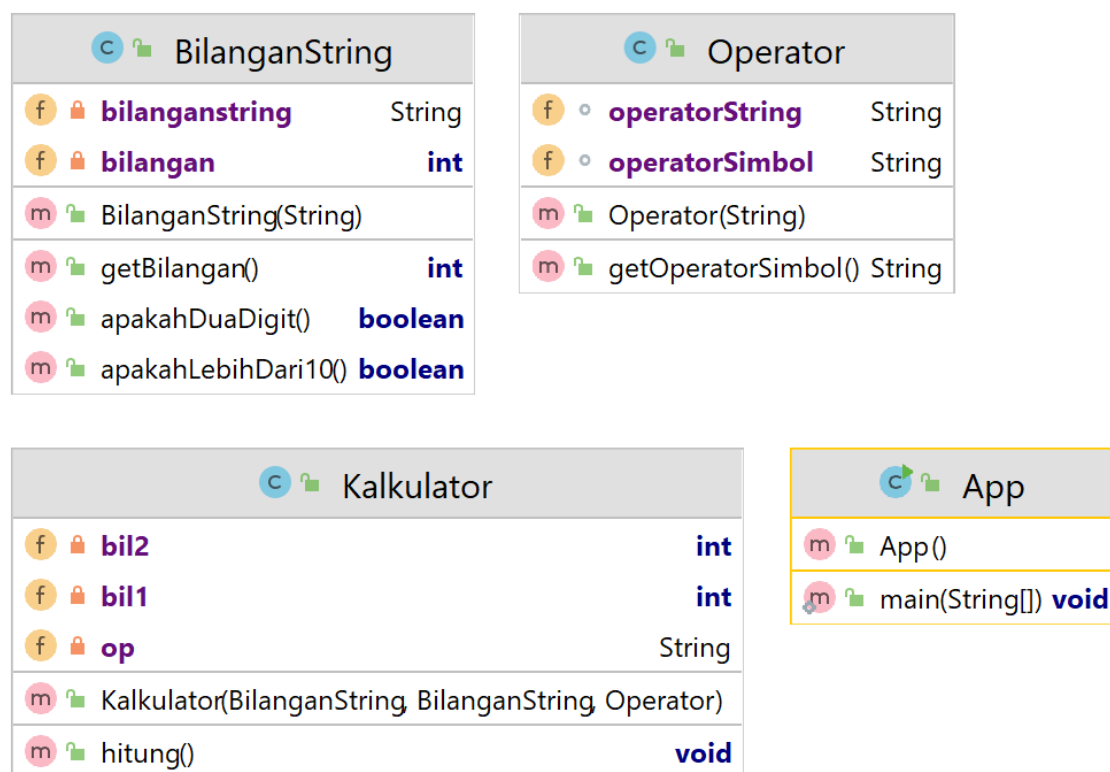
1. Penggunaan Maven : 15
2. Pembuatan class Titik dengan semua spesifikasinya: 25
3. Pembuatan class Garis dengan semua spesifikasinya : 25

4. Pembuatan class SegiEmpat dengan semua spesifikasinya: 25
5. Program bisa running tanpa error: 10

PROJECT KALKULATOR (POIN 70)

Project Kalkulator merupakan kalkulator sederhana untuk menghitung perkalian, penjumlahan, pembagian, pengurangan, dan pemangkatan dari dua buah bilangan. Istimewanya, aplikasi ini menerima bilangan berupa string bahasa Indonesia dengan nilai bilangan maksimal adalah **20**. Operator yang digunakan juga berupa teks bahasa Indonesia dalam bentuk String. Anda harus membuatkan method hitung pada Kalkulator untuk menghitung hasil kalkulasinya.

Class diagram aplikasi ini adalah:



Keterangan:

- Ketentuan project maven:
 - Grupid = com.rplbo.uts2
 - Artifactid = Kalkulator
 - Version = 1.0-SNAPSHOT
 - Name = Kalkulator
- 1. Class **BilanganString** merupakan class yang menerima bilangan dalam bentuk teks berbahasa Indonesia dengan nilai maksimal 20. Misalnya adalah “dua belas”. Class ini mengkonversi bilangan dalam teks ke dalam bilangan dalam bentuk integer. Kemudian terdapat method **apakahDuaDigit()** untuk mengembalikan true/false jika bilangan tersebut dua digit. Method **apakahLebihDari10()** akan mengembalikan true/false jika lebih dari 10 atau tidak. Input teks tidak harus huruf kecil.

2. Class **Operator** merupakan class yang menerima jenis operator dalam bentuk teks berbahasa Indonesia, hanya “kali”, “bagi”, “tambah”, “kurang”, dan “pangkat”. Input teks tidak harus huruf kecil.
3. Class **Kalkulator** adalah class utama untuk menerima input **bilangan1**, **bilangan2**, dan **operator** dengan tipe class **BilanganString** dan **Operator**. Class ini memiliki method **hitung** untuk menghitung hasil akhir dari kalkulasi bilangan dan operatornya.
4. Gunakan Maven!
5. *Class Main yang diberikan pada project dapat dikembangkan untuk dicoba. Class AppTest juga sudah disediakan. Saat mencoba, class AppTest bisa dikomen dulu.*

Contoh Main Class (App) adalah:

```
package com.rplbo.utskalkulator;

public class App
{
    public static void main( String[] args )
    {
        BilanganString b1 = new BilanganString("DuA Belas");
        BilanganString b2 = new BilanganString("tiga BELAS");
        System.out.println("Apakah 2 digit? " + b1.apakahDuaDigit());
        System.out.println("Apakah lebih dari 10? " +
b2.apakahLebihDari10());
        Operator op = new Operator("TaMbah");
        Kalkulator k = new Kalkulator(b1,b2,op);
        k.hitung();
        Kalkulator k2 = new Kalkulator(new BilanganString("dua"),new
BilanganString("tiga"),new Operator("pangkat"));
        k2.hitung();
    }
}
```

Hasil yang diharapkan:

```
Apakah 2 digit? true
Apakah lebih dari 10? true
Hasil: 25.0
Hasil: 8.0
```

Poin Penilaian:

1. Penggunaan Maven : 15
2. Pembuatan class **BilanganString** (dengan semua spesifikasinya): 25
3. Pembuatan class **Operator** (dengan semua spesifikasinya) : 25
4. Pembuatan class **Kalkulator** (dengan semua spesifikasinya): 25
5. Program bisa running tanpa error: 10

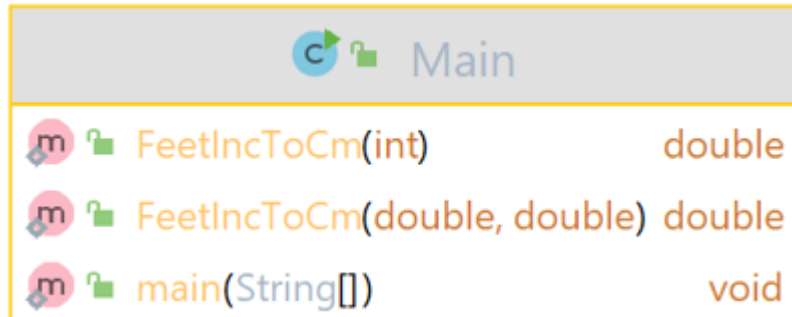
Project Feet Inch to Cm (POINT 30)

Materi: Overloading method

Buatlah sebuah program java untuk menghitung nilai konversi dari feet dan inci ke centimeter(cm) dengan ketentuan berikut:

- Ketentuan project maven:

- Grupid = com.rplbo.uts3
- Artifactid = FeetIncToCm
- Version = 1.0-SNAPSHOT
- Name = FeetIncToCm
- Hanya ada 1 class saja Main



- Buat method bernama **FeetIncToCm** yang memiliki 2 parameter yaitu **feet** dan **inch**.
 - Validasi nilai kedua parameter tidak boleh kurang atau sama dengan 0, nilai inch harus kurang atau sama dengan 12. Return -1 jika tidak memenuhi syarat.
 - Jika valid, maka konversi nilai dari feet dan inch ke cm.
 - Return nilai **centimeter**-nya.
- Buat method dengan nama yang sama yaitu **FeetIncToCm** tetapi hanya memiliki 1 parameter **inc**.
 - Validasi parameter inch tidak boleh kurang atau sama dengan 0. Return -1 jika tidak memenuhi.
 - Jika valid, maka hitunglah nilai centimeternya dengan **memanggil fungsi FeetIncToCm (double feet, double inch)** yang sudah dibuat sebelumnya.
 - Hint: inch dibagi menjadi feet dan sisanya menjadi inch.
 - Return nilai **centimeternya**
- Panggil kedua method di main method.

CATATAN: 1 inch = 2.54 cm dan 1 foot = 12 inch

| No | Test Case | Output |
|----|---|--|
| 1 | FeetIncToCm(1, 3); FeetIncToCm(15); | Output: 38.1 Output: 38.1 *Baris pertama merupakan output dari FeetIncToCm(1, 3); *Baris kedua dan ketiga merupakan output dari FeetIncToCm(15); |
| 2 | FeetIncToCm(12, 5); FeetIncToCm(30); | Output: 378.46 Output: 76.2 *Baris pertama merupakan output dari FeetIncToCm(12, 5); *Baris kedua dan ketiga merupakan output dari FeetIncToCm(30); |

Poin penilaian:

1. Struktur project benar: 15
2. Terdapat 2 method FeetToCm overloading: 30
3. Logic benar: 20
4. Me-reuse method FeetToCm yang pertama: 20
5. Output benar/ tidak error: 15

Vehicle (POIN 70)

Buatlah implementasi dari class diagram berikut ini:



Penilaian:

1. Implementasi masing-masing class sesuai dengan class diagram (25 poin)
2. Implementasi relasi antar class benar. (15 poin)
3. Implementasi nama variabel sesuai dan menerapkan encapsulation pada semua variable. (10 poin)
4. Constructor masing-masing class benar (10 poin)
5. Method masing-masing class benar. (20 poin)
6. Test case berjalan benar. (20 poin)