

# **MODUL 3**

## **KELAS & OBJEK**

Praktikum Pemrograman Berorientasi Objek  
2023



## Tujuan Pembelajaran

1. Mengetahui konsep kelas dan objek
2. Mampu mengimplementasikan konsep kelas dan objek ke dalam bahasa pemrograman Java

## Pendahuluan

Segala sesuatu adalah **OBJEK**. Objek-objek yang sering kita temui di kehidupan sehari-hari tentu dapat dianggap juga sebagai sebuah objek. Objek memiliki 2 komponen, yaitu:

1. Atribut/Variabel/Field

Sebuah variabel yang melekat pada sebuah objek yang digunakan untuk menyimpan sebuah nilai/value/data yang dimiliki oleh sebuah objek.

2. Method/Operasi

Merupakan perilaku yang bisa dilakukan oleh objek tersebut. Implementasi dari perilaku adalah method (fungsi atau prosedur) yang digunakan untuk mengakses atau memanipulasi atribut sehingga menghasilkan suatu keluaran yang diharapkan.

Berikut adalah contoh Objek:

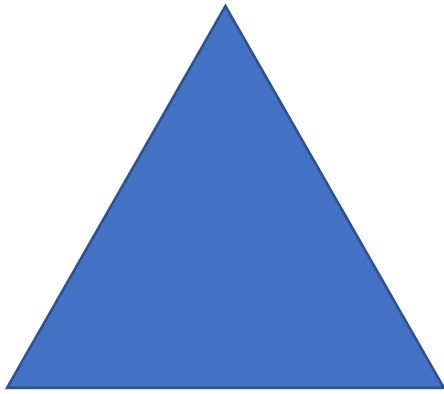


### Atribut:

- Warna : Biru
- Merk : ASUS
- Tahun Pembuatan : 2021

### Method:

- Mengerjakan Tugas
- Bermain Game



#### Atribut:

- Alas : 8
- Tinggi : 12
- Jenis Segitiga Sama Kaki

#### Method:

- Menghitung Luas Segitiga
- Menghitung Keliling Segitiga

Objek-Objek yang sama dapat disatukan menjadi sebuah **KELAS**. Kelas merupakan sebuah “Template”, yang digunakan untuk membentuk sebuah objek dengan karakteristik yang sama seperti template tersebut. Sebuah kelas memiliki beberapa komponen pembentuk utama, yaitu:

#### 1. Atribut

Seperti yang sudah dijelaskan sebelumnya, atribut merupakan tempat untuk menyimpan sebuah nilai yang dimiliki oleh sebuah objek. Pada saat sebuah objek dibuat dari sebuah kelas, maka atribut yang ada di kelas tersebut juga akan melekat dari objek yang diciptakan. Dalam implementasinya, atribut mungkin dapat juga berupa objek lain, atau rujukan ke objek lain. Dalam bahasa pemrograman Java, pendeklarasian atribut dapat dituliskan sebagai berikut:

```
private String nama;
```

Visibilitas      Tipe data      Nama atribut/variabel

#### 2. Konstruktor

Konstruktor merupakan method yang berfungsi untuk membuat objek dan menginisialisasikan nilai-nilai atribut dari objek yang terbentuk. Tujuan dari penggunaan konstruktor adalah untuk memastikan bahwa objek yang dihasilkan oleh kelas berada dalam sebuah keadaan yang valid. Suatu kelas dapat berisi lebih dari 1 konstruktor, dan karakteristik utamanya adalah tidak memiliki nilai balikan, dan nama konstruktor pasti sama dengan nama kelasnya. Dalam bahasa pemrograman Java, konstruktor dapat dituliskan sebagai berikut.

```
public Mahasiswa() {
    ...
}
```

Visibilitas

Nama Konstruktor

Konstruktor Default

```
public Mahasiswa(String NPM, String nama) {
    ...
}
```

Visibilitas

Nama Konstruktor

Parameter

Konstruktor dengan parameter

### 3. Method

Perilaku yang bisa dilakukan oleh objek tersebut. Implementasi dari perilaku adalah method yang digunakan untuk mengakses atau memanipulasi atribut sehingga menghasilkan suatu keluaran yang diharapkan. Method dapat berupa prosedur dan fungsi. Untuk fungsi, dapat memiliki nilai balikan berupa tipe data dasar & bentukan (int, double, String, vector, dll) dan dapat berupa kelas lain. Dalam bahasa pemrograman Java, method dapat dituliskan sebagai berikut.

```
public void showMahasiswa(String nama)
```

Visibilitas

Tipe balikan

Nama method

Parameter

Terdapat beberapa jenis Method, yaitu:

#### 1. Aksesori/Accessor/Getter

Sebuah method disebut dengan method aksesori jika fungsi dari method tersebut digunakan untuk mengembalikan nilai dari atribut sebuah objek. Tipe balikan method aksesori biasanya mengikuti tipe data atribut yang nilainya akan dikembalikan. (**mendapatkan value dari suatu atribut**)

Contohnya,

```
public String getName(){
    return nama;
}
```

Method **getName()** akan mengembalikan nilai atribut **nama**

## 2. Mutator/Setter

Sebuah method disebut dengan method mutator jika fungsi dari method tersebut adalah untuk mengubah nilai dari atribut tertentu. Method mutator biasanya tidak mengembalikan nilai apapun/void. (**mengubah value dari atribut tertentu**)

Contohnya,

```
public void setName(String nama){
    this.nama = nama;
}
```

Method **setName(...)** akan mengganti value dari variable **nama** dengan value dari parameter inputan **String nama**

## Visibilitas

Visibilitas atau Aksesabilitas bisa juga dibilang sebagai “batas” akses anggotanya, terdapat 3 tipe visibilitas yang ada, yaitu:

1. Private (-) : **TIDAK** dapat diakses oleh kelas lain.
2. Protected (#) : Dapat diakses oleh kelas turunan.
3. Public (+) : Dapat diakses oleh kelas lain.

## Jenis-Jenis Variable

### 1. Instance Variable/Variabel Objek

Variable ini merupakan variable yang memiliki sebuah value/nilai masing-masing atribut objek.

Contoh:

**Kelas** → Mahasiswa

**Objek** → mahasiswa1 → nama = “Felix”, umur = 20

                  mahasiswa2 → nama = “Yehdeya”, umur = 12

**Dari contoh di atas dapat kita lihat bahwa setiap objek memiliki value atributnya masing-masing.**

## 2. Class Variable/Variabel Kelas

Suatu variabel yang dimiliki pada suatu kelas dan variabel tersebut akan digunakan bersama-sama oleh semua objek dari kelas tersebut. Menggunakan keyword static

Contoh:

**Kelas** → Mahasiswa → **public static int jumlahObjekCreated**

**Objek** → mahasiswa1 → nama = "Felix", umur = 20

                  mahasiswa2 → nama = "Yehdeya", umur = 12

Terdapat class variable bernama **jumlahObjekCreated** nanti akan menampilkan value 2 (karena kita membuat 2 Objek), jadi value dari variabel **jumlahObjekCreated** nantinya akan mengikuti banyaknya jumlah objek yang kita buat.

## Konstanta

Sesuatu yang nilainya tetap, tidak bisa diubah dan nilainya akan selalu sama. Diberikan kepada variabel yang nilainya sudah pasti. Menggunakan keyword final, karena nilai yang sudah terdeklarasikan tidak akan bisa diubah lagi. Umumnya penulisan untuk variabel konstanta berisi huruf besar semuanya.

**Contoh:**

**Phi** pada lingkaran nilainya tidak akan berubah yaitu akan tetap bernilai 3.14

**Kelas** → Lingkaran

**Contoh pendeklarasian:**

**private static final double PHI = 3.14;**                   → Pendeklarasian konstanta

**private double jari\_jari;**                                   → Pendeklarasian atribut

```
public double luasLingkaran(){  
    return PHI * jari_jari * jari*jari;  
}
```

Bagaimana jika kita ingin menggunakan konstanta PHI yang ada di dalam kelas Lingkaran untuk digunakan pada kelas lainnya? sebagai contoh yaitu kelas Tabung. Kita akan menggunakan class variable/variabel kelas untuk dapat diakses oleh kelas lain, dengan keyword **static**

**Kelas → Lingkaran**

**Contoh pendeklarasian:**

```
public static final double PHI = 3.14;  
konstanta
```

→ Pendeklarasian static

```
private double jari_jari;
```

→ Pendeklarasian atribut

```
public double luasLingkaran(){  
    return PHI * jari_jari * jari*jari;  
}
```

**Kelas → Tabung**

**Contoh pendeklarasian:**


```
private double jari_jari;
```

→ Pendeklarasian atribut

```
private double tinggi;
```

→ Pendeklarasian atribut

```
public double volumeTabung(){  
    return Lingkaran.PHI * jari_jari * jari*jari * tinggi;  
}
```



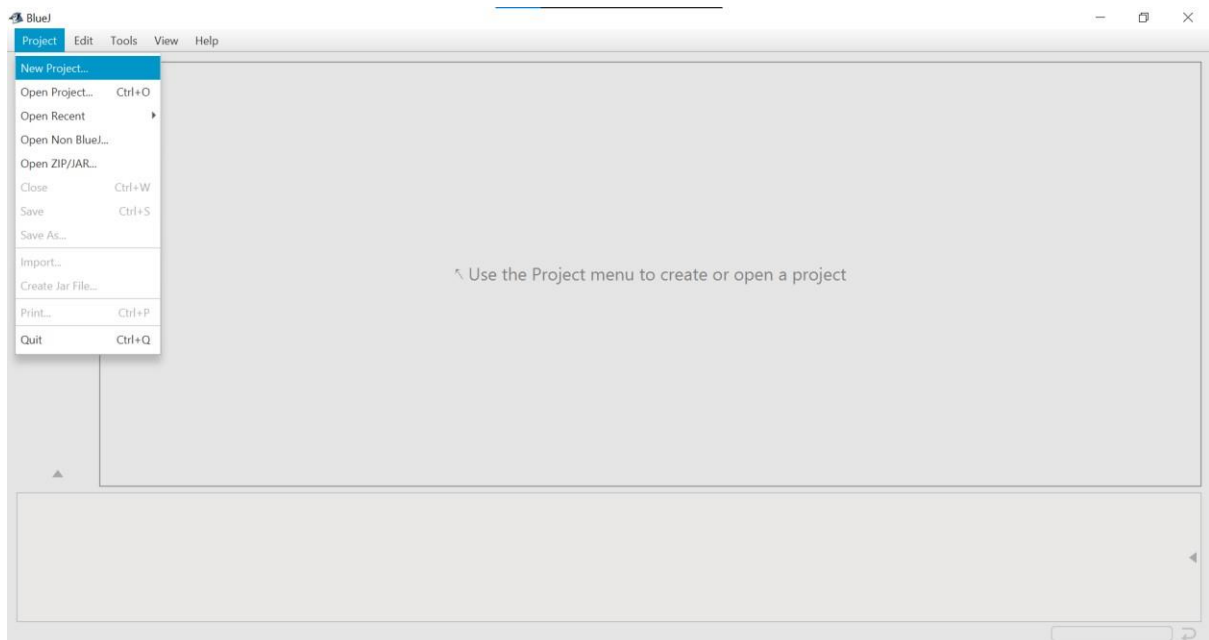
Memanggil variabel konstanta dengan nama **KELAS**nya (dot) namaVariabelKonstantanya

## GUIDED 1 – BLUE JAVA

Guided pertama ini, kita akan membuat sebuah kelas yang bernama PersegiPanjang dengan menggunakan salah satu tools untuk belajar bahasa pemrograman java, yaitu BlueJ. Secara umum sebuah persegi panjang mempunyai 2 buah atribut, yaitu panjang dan lebar, dan 2 method yang akan digunakan untuk menghitung luas dan keliling persegi panjang.

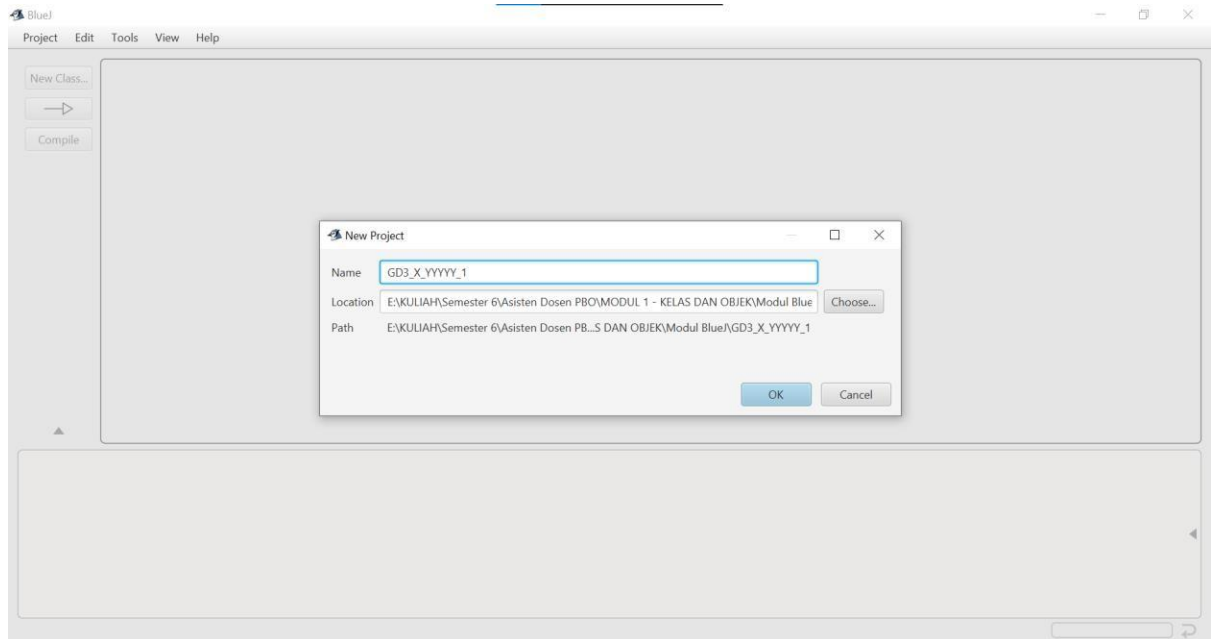
### LANGKAH – LANGKAH

1. Buka aplikasi BlueJ kalian. Kemudian pilih Project → New Project

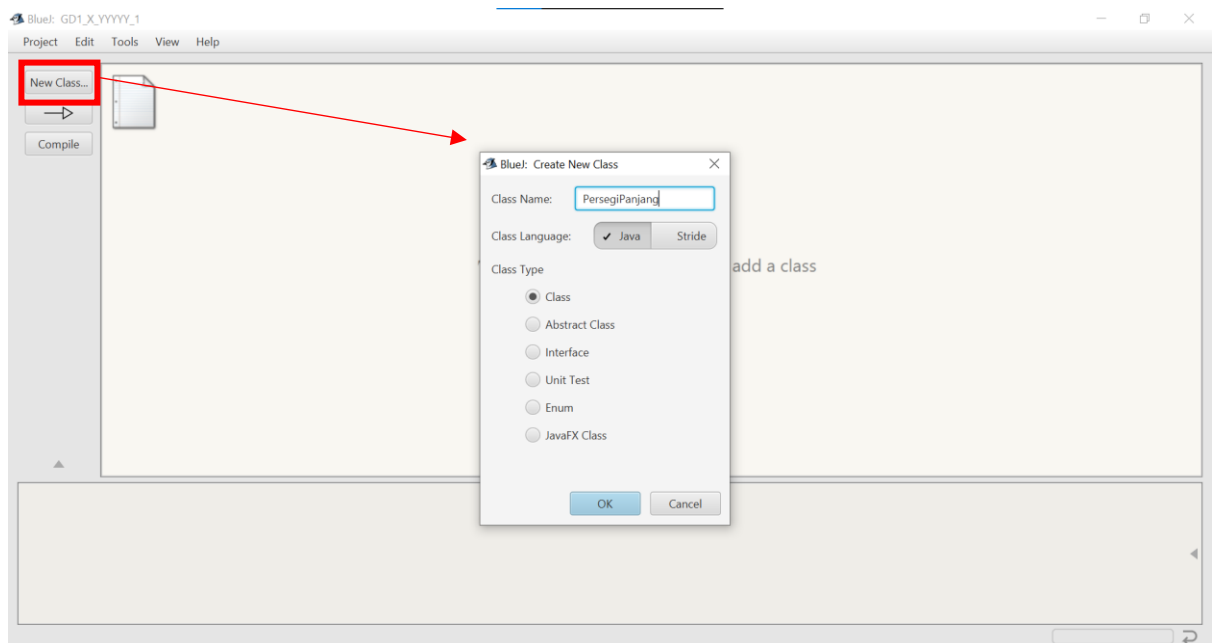


2. Isi **Name** dengan format **GD3\_X\_YYYYY\_1** (**X = Kelas**, **YYYYY = 5 Digit Belakang NPM**), lalu klik OK. Jendela BlueJ baru akan terbuka.

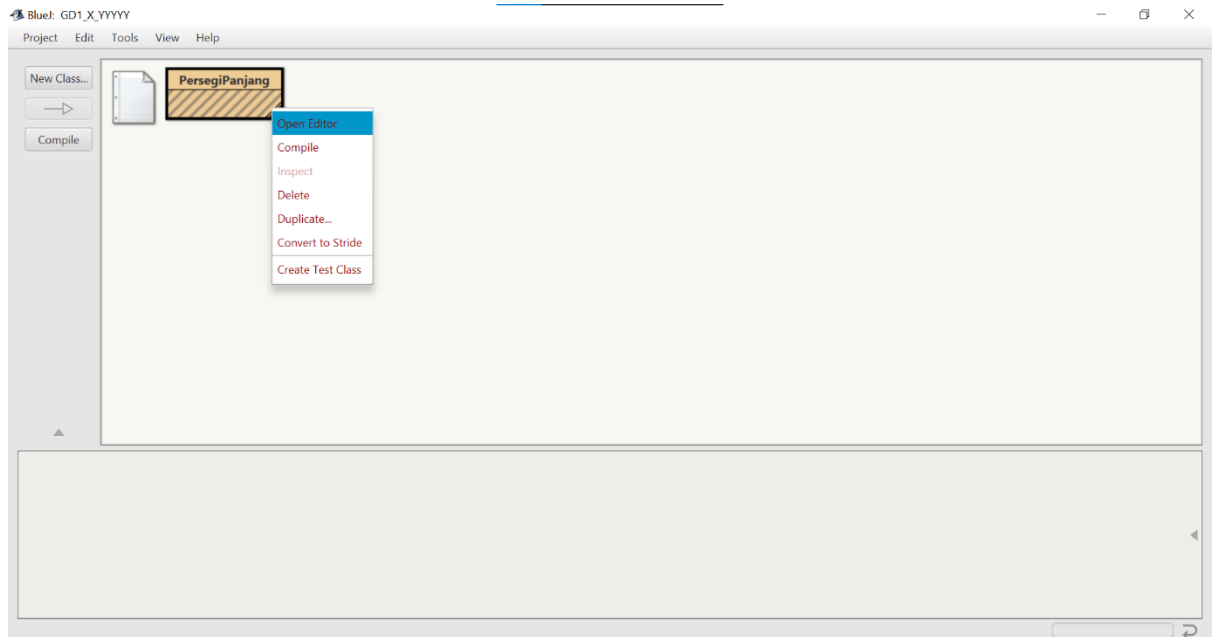




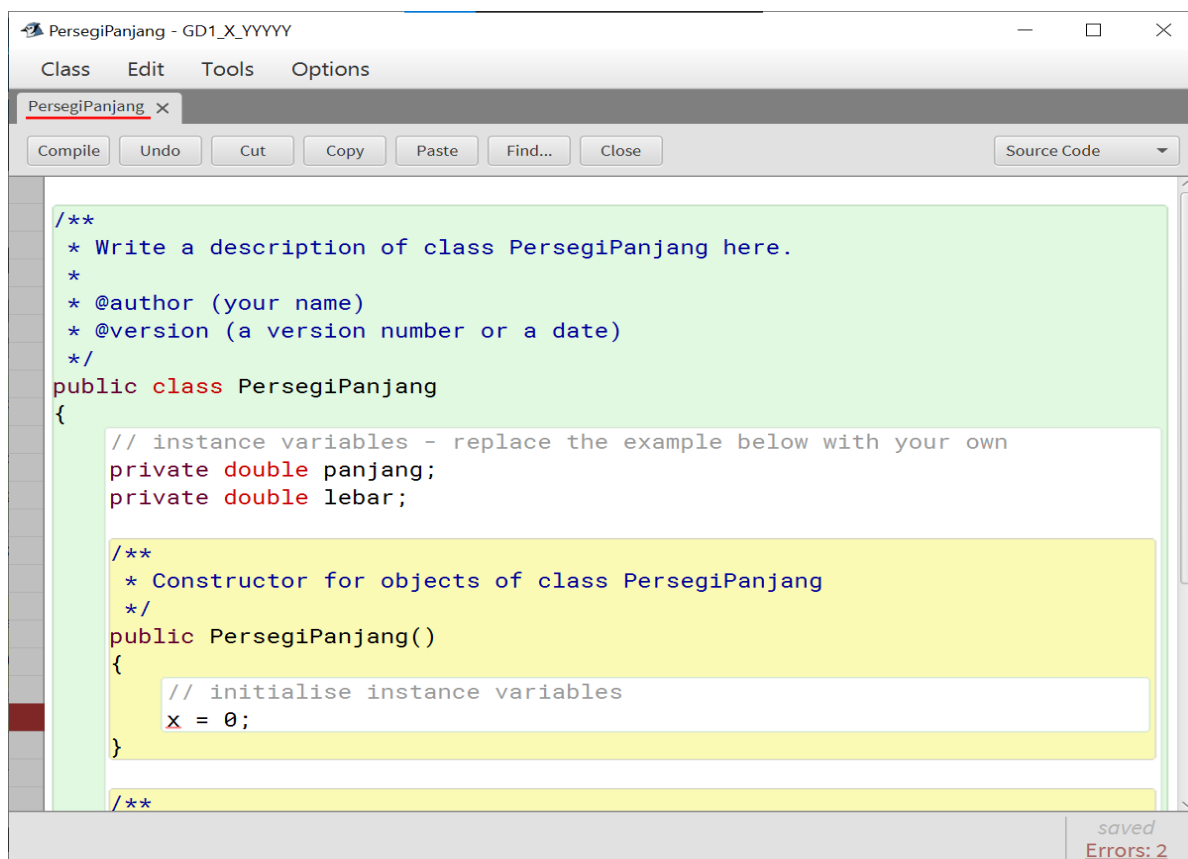
3. Sekarang, kita akan membuat sebuah kelas yang bernama PersegiPanjang, dengan caraklik tombol **New Class**, isi bagian **Class Name** dengan nama PersegiPanjang, lalu klikOK.



4. Sekarang, kita akan memulai mengedit code milik kelas PersegiPanjang, dengan caraKlik kanan pada kelas PersegiPanjang, klik Open Editor



5. Masukkan 2 Atribut yang dimiliki oleh persegi panjang, yaitu **panjang** dan **lebar** yang bertipe data double



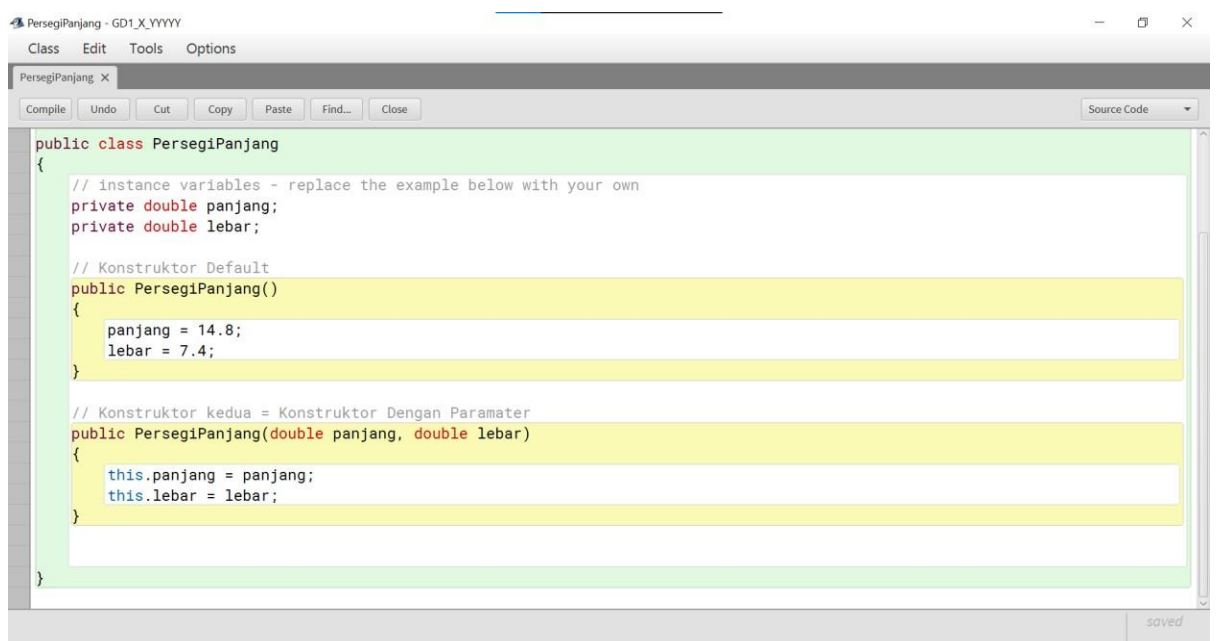
## UNTUK DIPERHATIKAN:

Dalam langkah ini, akan memunculkan error yang bisa kita lihat dengan adanya 2 kotak merah yang berada di samping kiri code (pada baris `x=0` & `return x+y`). Error tersebut muncul karena variabel yang kita pakai tidak kita lakukan pendeklarasian atribut. Maka dari itu, usahakan untuk memperhatikan penulisan code (pendeklarasian atribut, penggunaan semicolon, dll), sehingga tidak terjadi adanya syntax error yang menyebabkan program tidak dapat berjalan.

6. Kalian boleh untuk menghapus code berikut, karena code tersebut tidak akan kita pakainantinya.

```
/**
 * An example of a method - replace this comment with your own
 *
 * @param y a sample parameter for a method
 * @return the sum of x and y
 */
public int sampleMethod(int y)
{
    // put your code here
    return x + y;
}
```

7. Mulai isi konstruktor `public PersegiPanjang() {}` dengan nilai default, tanpa parameter (karena kita tidak meminta inputan dari user), dan buatlah konstruktor kedua dengan parameter panjang dan lebar.



```
public class PersegiPanjang
{
    // instance variables - replace the example below with your own
    private double panjang;
    private double lebar;

    // Konstruktor Default
    public PersegiPanjang()
    {
        panjang = 14.8;
        lebar = 7.4;
    }

    // Konstruktor kedua = Konstruktor Dengan Paramater
    public PersegiPanjang(double panjang, double lebar)
    {
        this.panjang = panjang;
        this.lebar = lebar;
    }
}
```

this. sebagai penunjuk bahwa atribut yang dimaksud adalah atribut dari kelas tersebut,

BUKAN PARAMETER INPUTAN

Ada beberapa cara penulisan konstruktor, yaitu

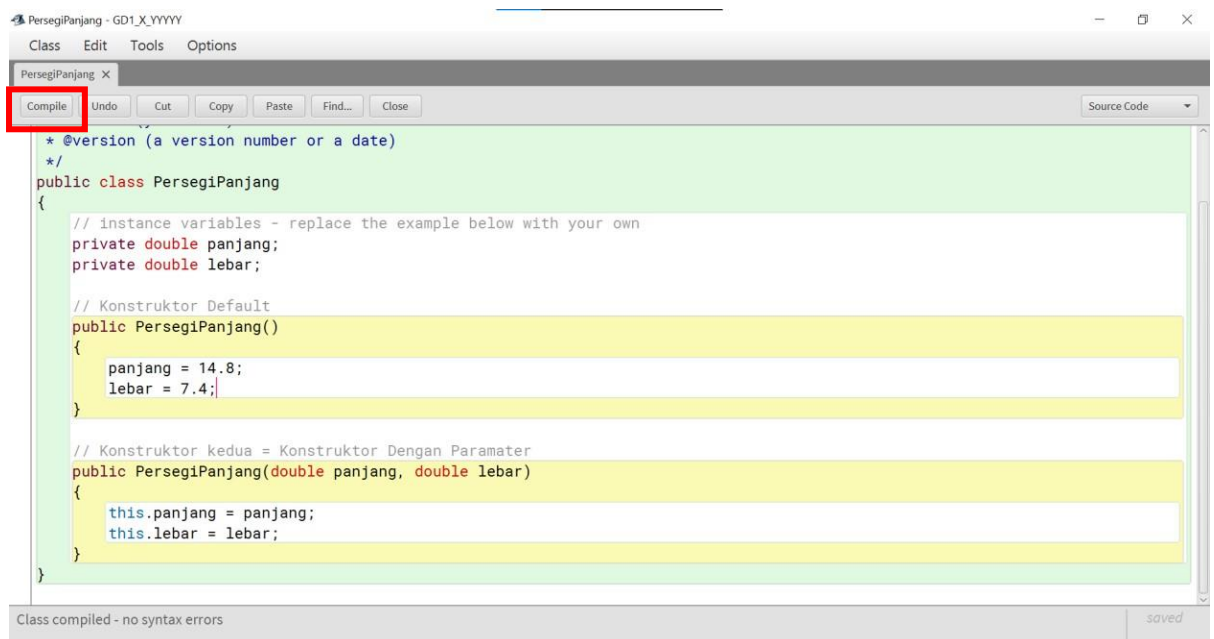
```
public PersegiPanjang(double panjang, double lebar){  
    this.panjang = panjang;  
    this.lebar = lebar;  
}
```

Atau

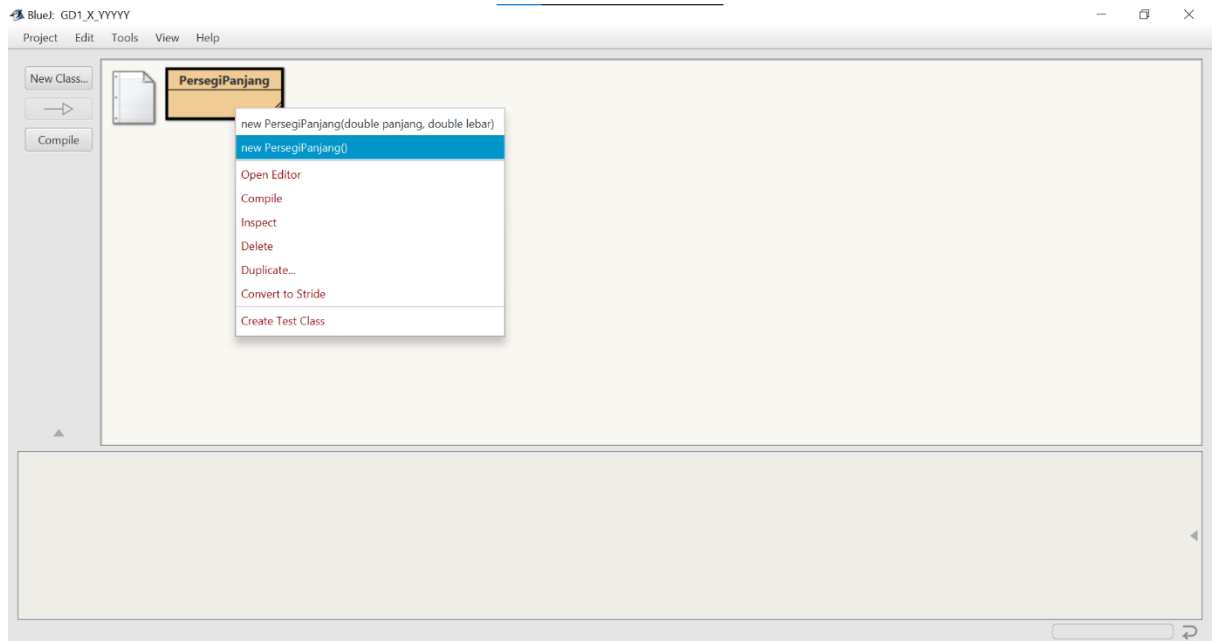
```
public PersegiPanjang(double inputanPanjang, double inputanLebar){  
    panjang = inputanPanjang;  
    lebar = inputanLebar;  
}
```

Kalian bebas untuk **MEMILIH SALAH SATU** cara penulisan konstruktor yang kalian inginkan.

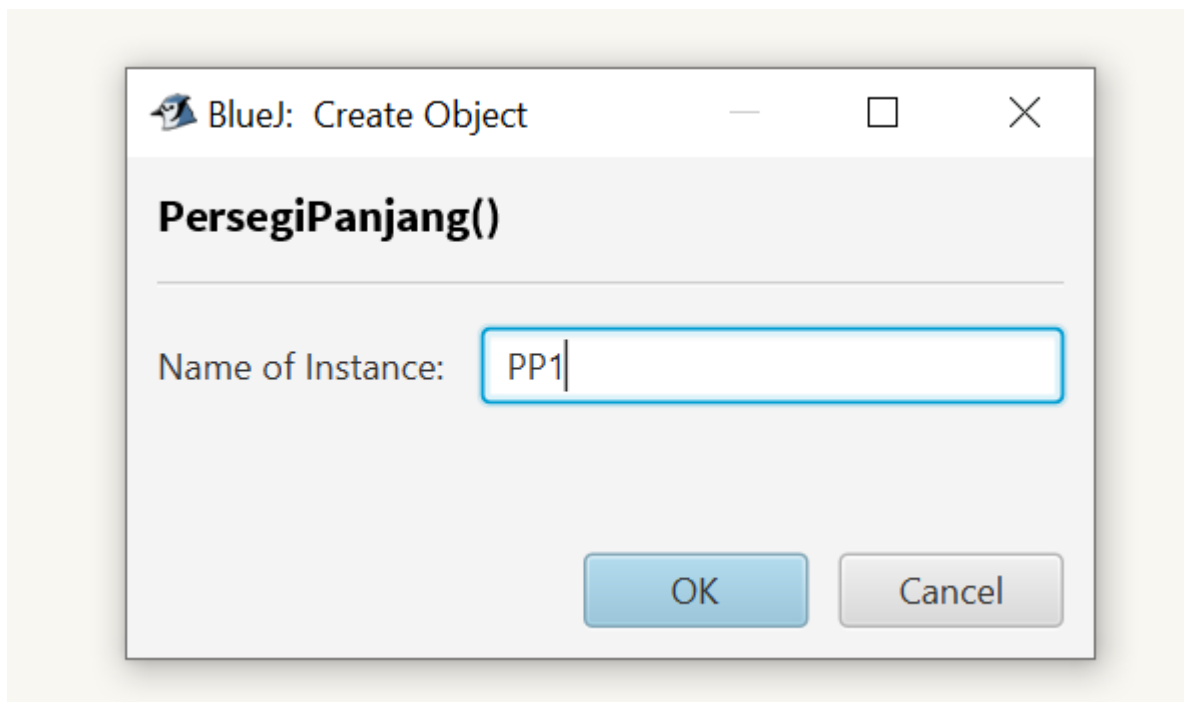
8. Setelah kalian menuliskan konstruktor default dan konstruktor dengan parameter, lalu klik **Compile**, lalu minimize jendela editornya



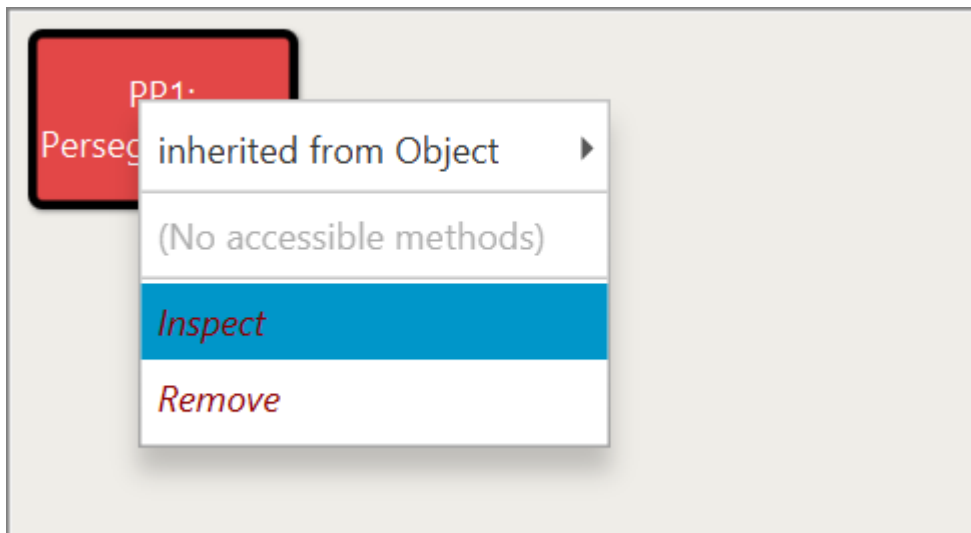
9. Setelah itu kita akan mulai untuk membuat objek dari kelas yang sudah kita buat. Caranya klik kanan di kelas PersegiPanjang lalu pilih New PersegiPanjang()



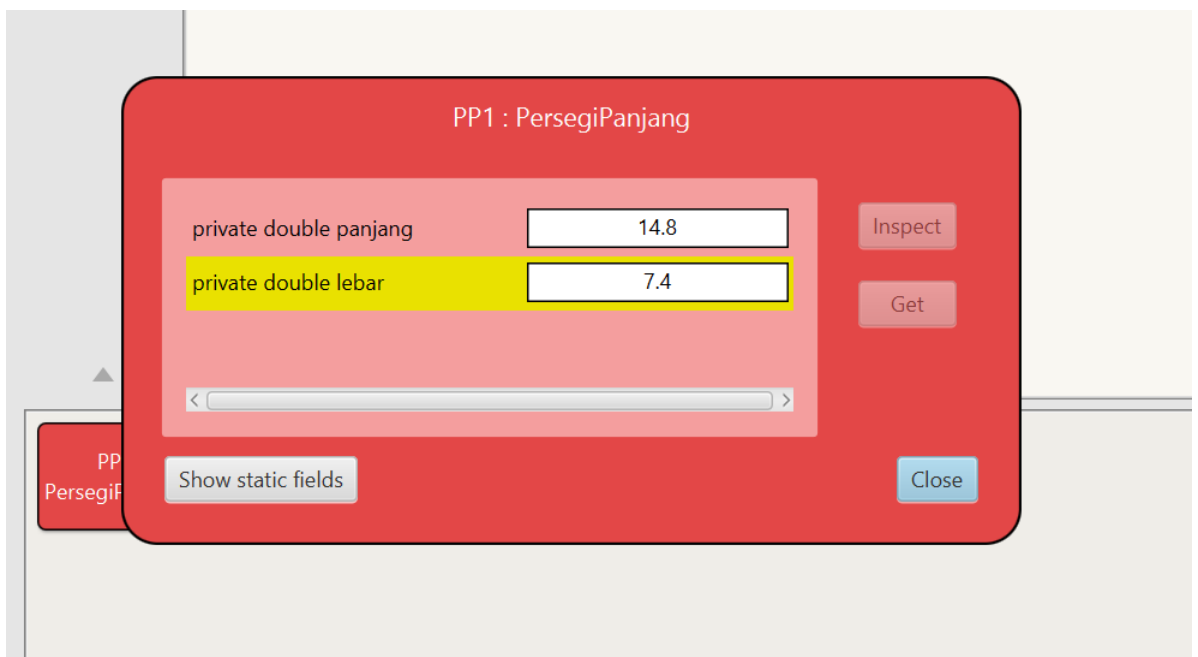
10. Beri nama objek tersebut dengan PP1, klik OK.



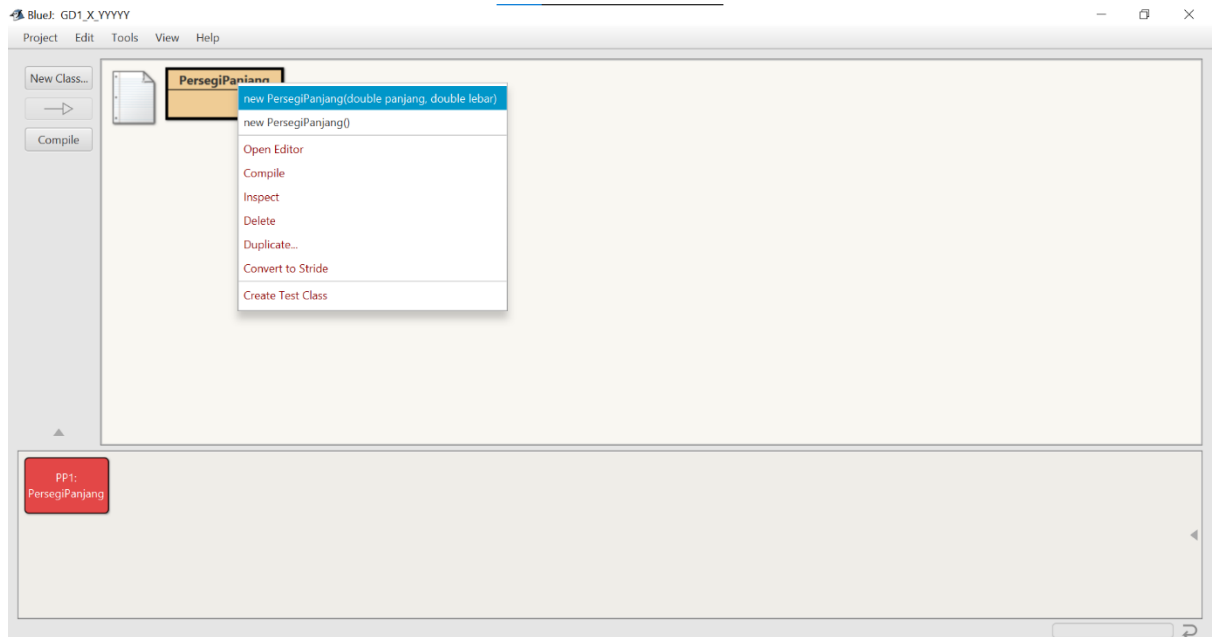
11. Lalu akan muncul kotak merah pada field bagian bawah dengan nama PP1, yang berartikita sudah berhasil untuk membuat objek dari kelas PersegiPanjang. Untuk kalian yangingin mengecek nilai dari atribut-atribut yang ada, silakan klik kanan pada kotak merahlalu klik **Inspect**.



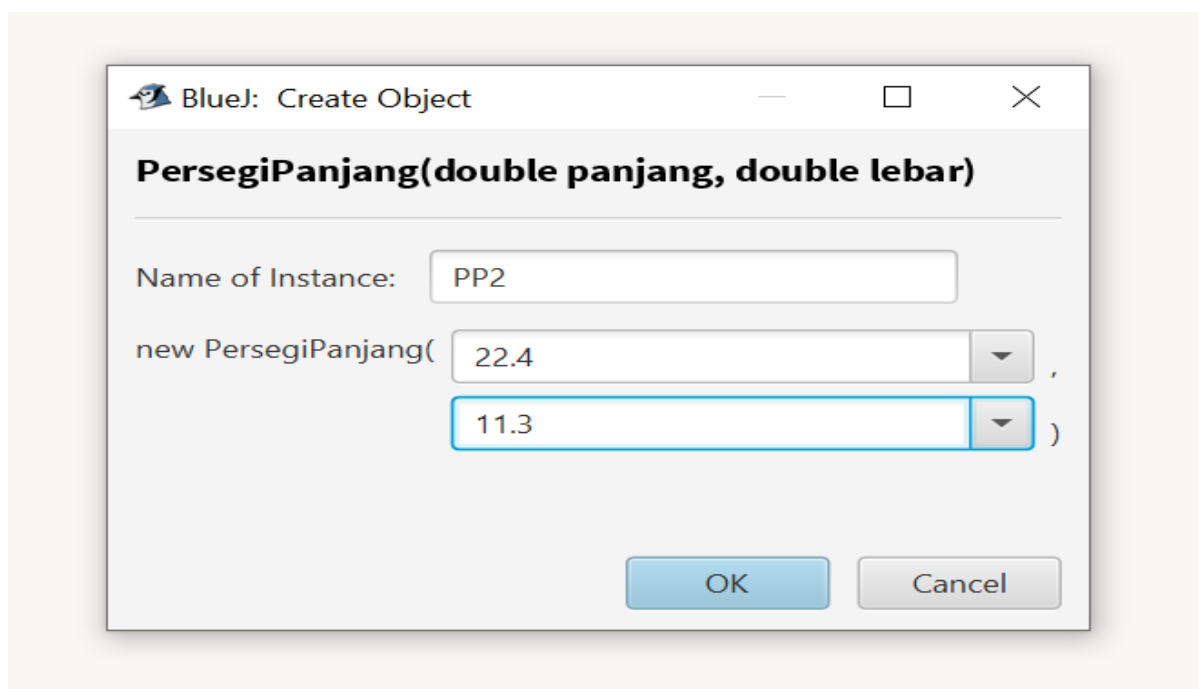
12. Lalu akan muncul sebuah kotak berwarna Merah lagi. Namun kotak merah tersebut berisi informasi tentang atribut-atribut dan nilai-nilai dari konstruktor default yang telah kita buat.



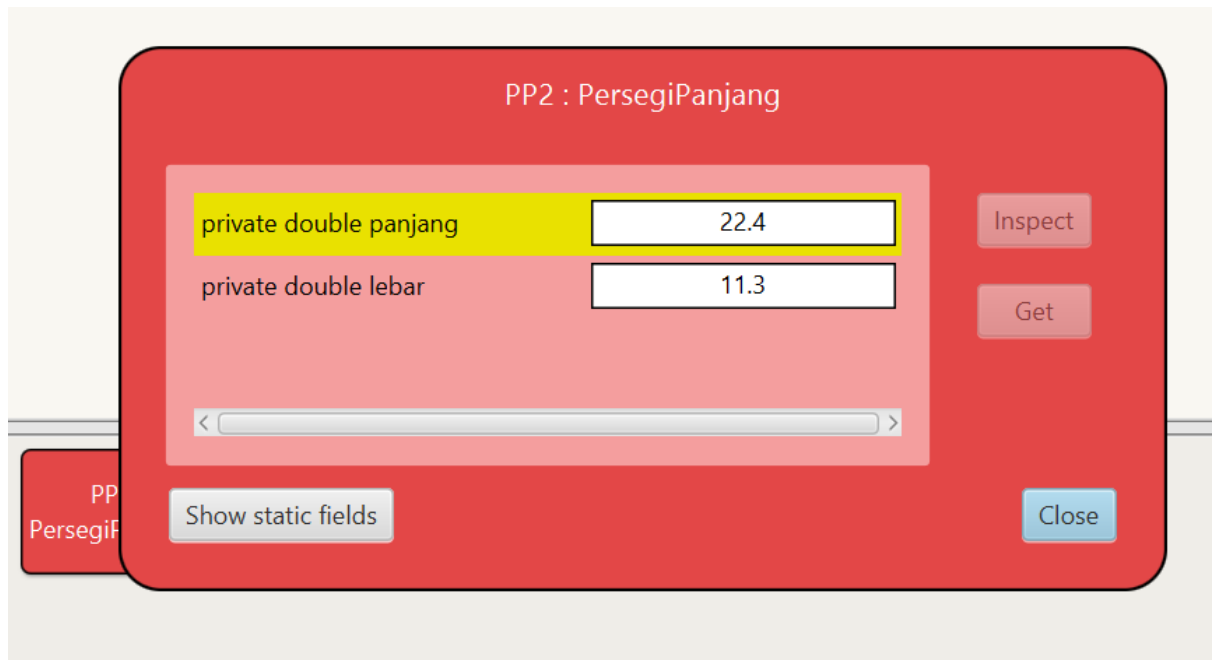
13. Setelah kita mencoba melihat konstruktor default, sekarang kita akan mencoba konstruktor kedua, yaitu konstruktor dengan parameter. Dengan cara yang sama, yaitu klik kanan pada kelas PersegiPanjang, lalu pilih New PersegiPanjang(double panjang, double lebar)



14. Sebuah jendela baru akan muncul **Name of Instance**, dan ada field yang meminta user untuk menginputkan nilai panjang dan lebar. Silakan menginputkan panjang dan lebar sesuai dengan keinginan kalian. Jika Sudah lalu klik OK



15. Lalu apabila kalian ingin melihat nilai dari atributnya silakan lakukan cara yang sama dengan langkah ke-11. Lalu lihatlah hasil nilai tersebut apakah nilainya sama dengan nilai yang kalian inputkan tadi.



16. Sekarang kita akan menambahkan beberapa method pada kelas PersegiPanjang. Method yang akan kita gunakan disini adalah method untuk menghitung luas dan keliling. Buka code editor yang kalian minimize sebelumnya, lalu tambahkan code di bawah ini.

```

PersegiPanjang - GD1_X_YYYY
Class Edit Tools Options
PersegiPanjang X
Compile Undo Cut Copy Paste Find... Close Source Code
{
    panjang = 14.8;
    lebar = 7.4;
}

// Konstruktor kedua = Konstruktor Dengan Paramater
public PersegiPanjang(double panjang, double lebar)
{
    this.panjang = panjang;
    this.lebar = lebar;
}

// Method Menghitung Luas
public double luas(){
    return panjang*lebar;
}

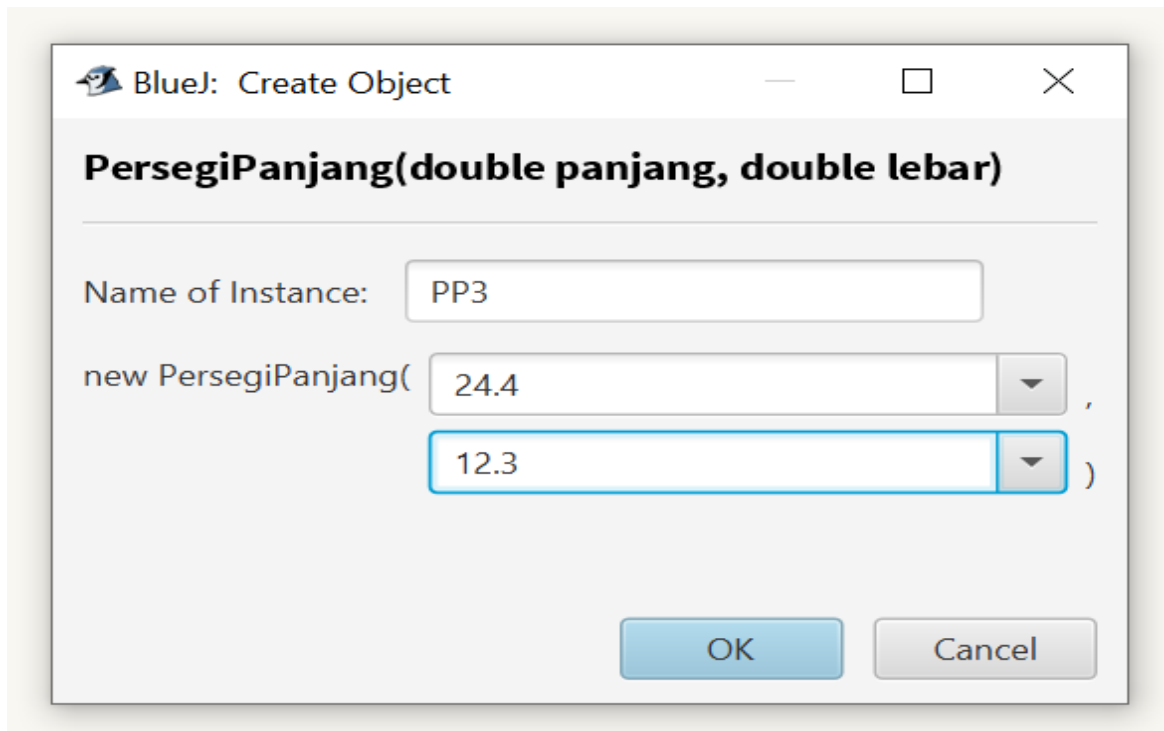
// Method Menghitung Keliling
public double keliling(){
    return 2*(panjang+lebar);
}
}

```

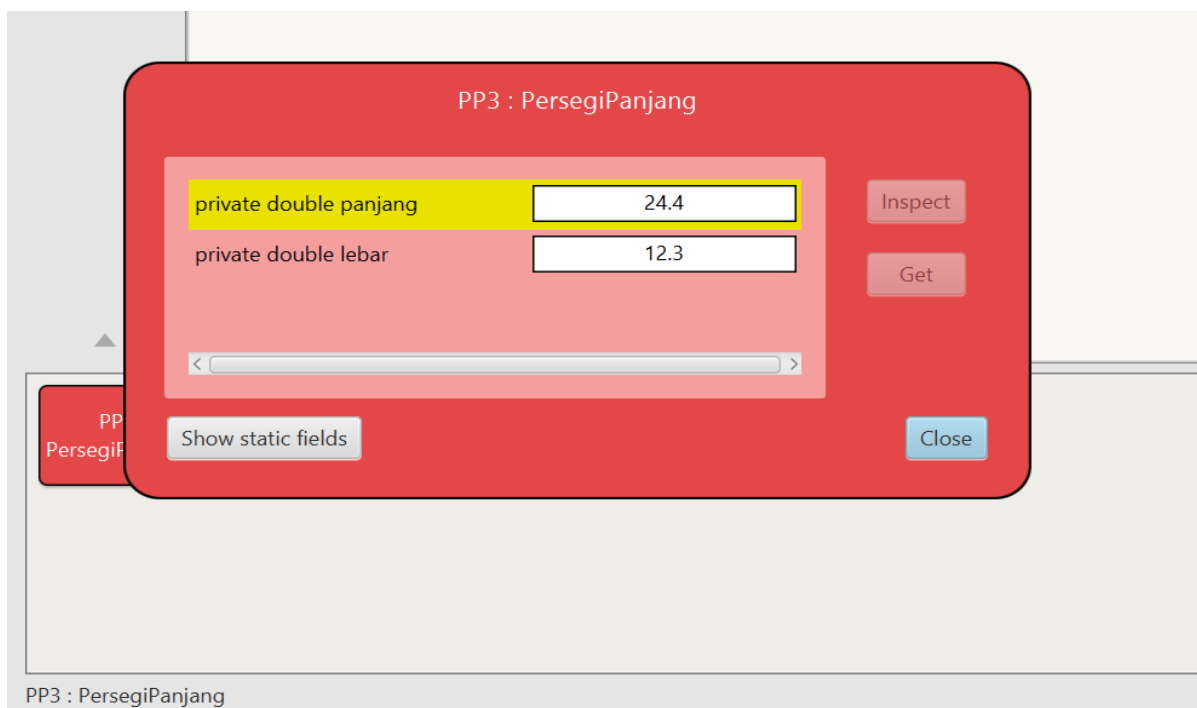
Jika sudah, silakan Klik **Compile** lagi.

17. Kita akan mencoba mengakses method-method tersebut. Buatlah objek baru dengan nama PP3, menggunakan konstruktor dengan parameter, pilihlah New PersegiPanjang(double panjang, double lebar)
18. Masukkan value panjang dan lebar sesuai dengan keinginan kalian. Jika sudah klik OK.

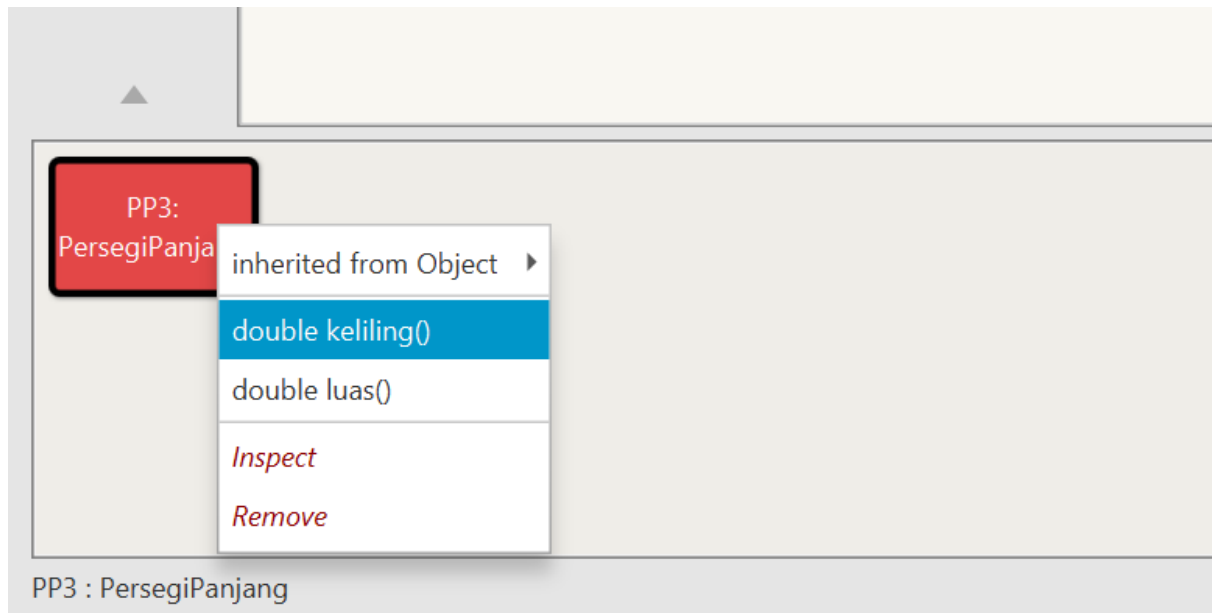




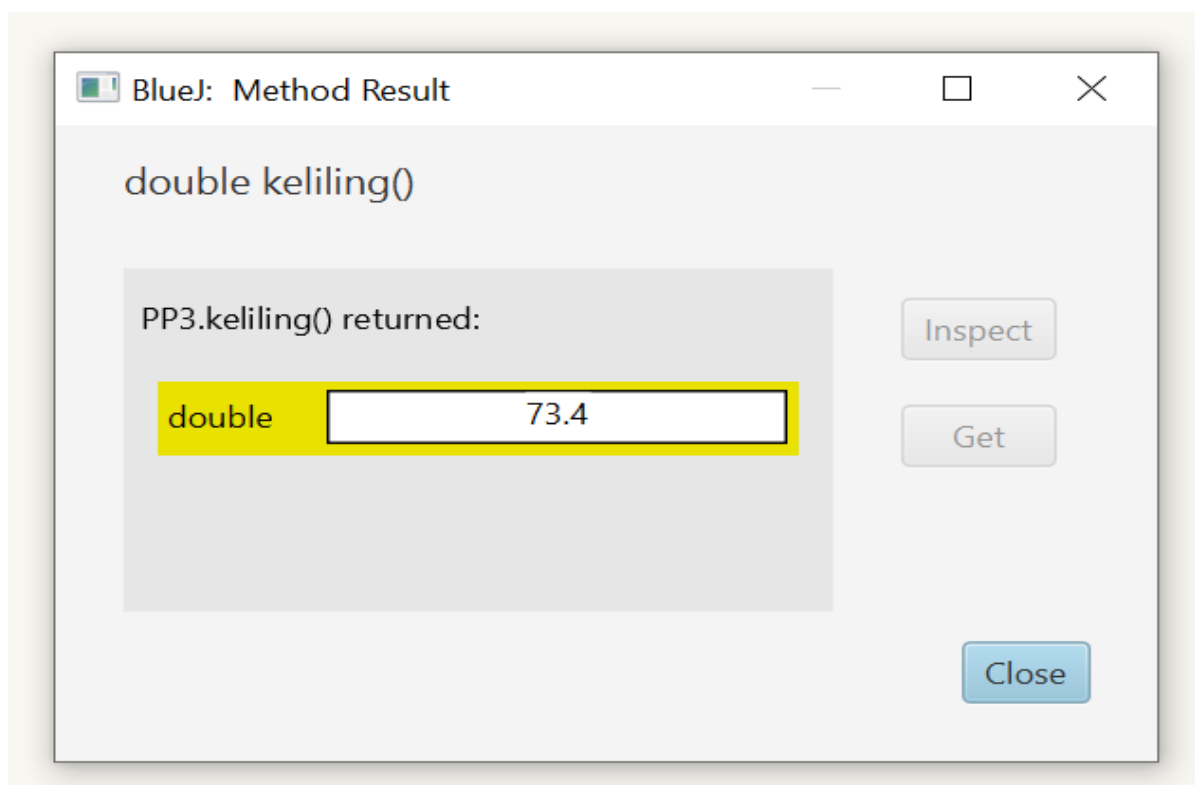
19. Jika sudah kalian boleh meng**Inspect** atribut-atribut atau value-value tersebut



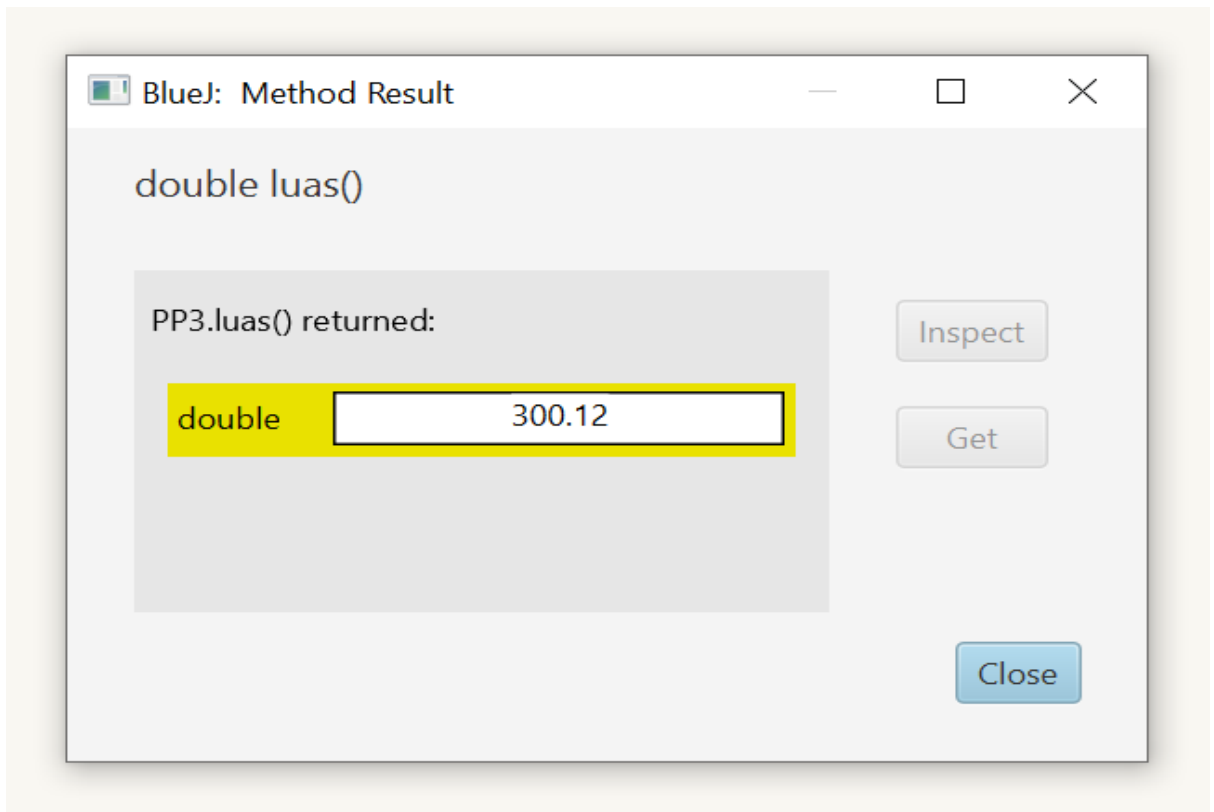
20. Sekarang kita akan mencoba melihat hasil dari Method yang telah kita buat yaitu Method `keliling()`, caranya klik kanan objek kalian lalu klik double `keliling()`



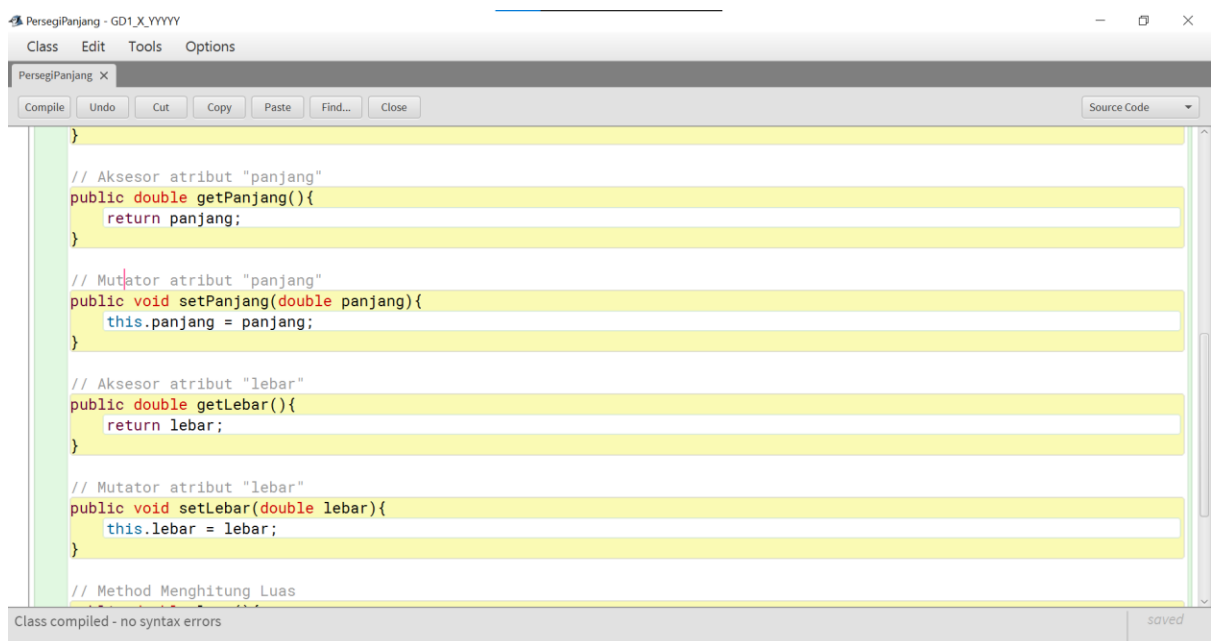
21. Lalu akan muncul jendela “Method Result” yang berisi hasil dari method `doublekeliling()` yang telah kalian buat.



22. Lakukan hal yang sama dengan langkah ke-20. Cobalah untuk melihat method `doubleluas()` yang telah kalian buat juga, maka hasilnya akan sebagai berikut.

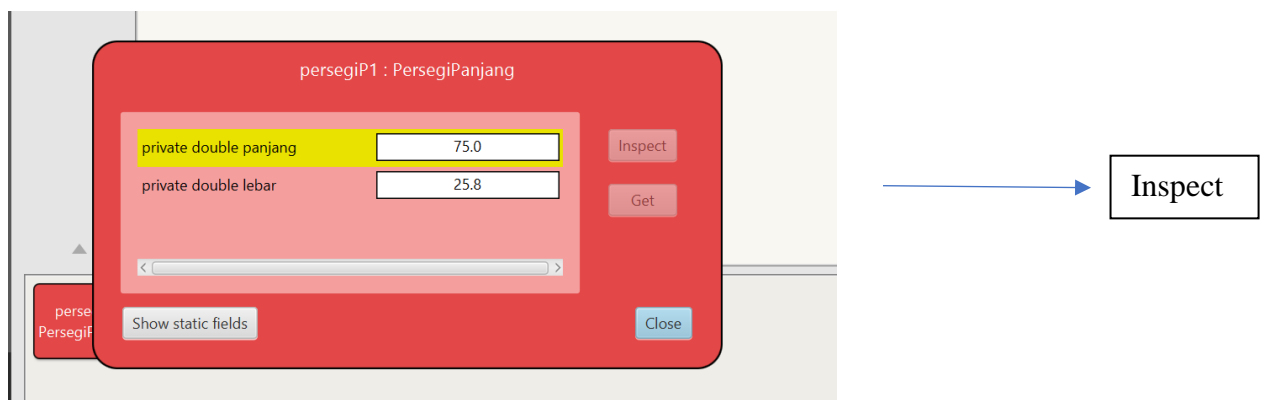
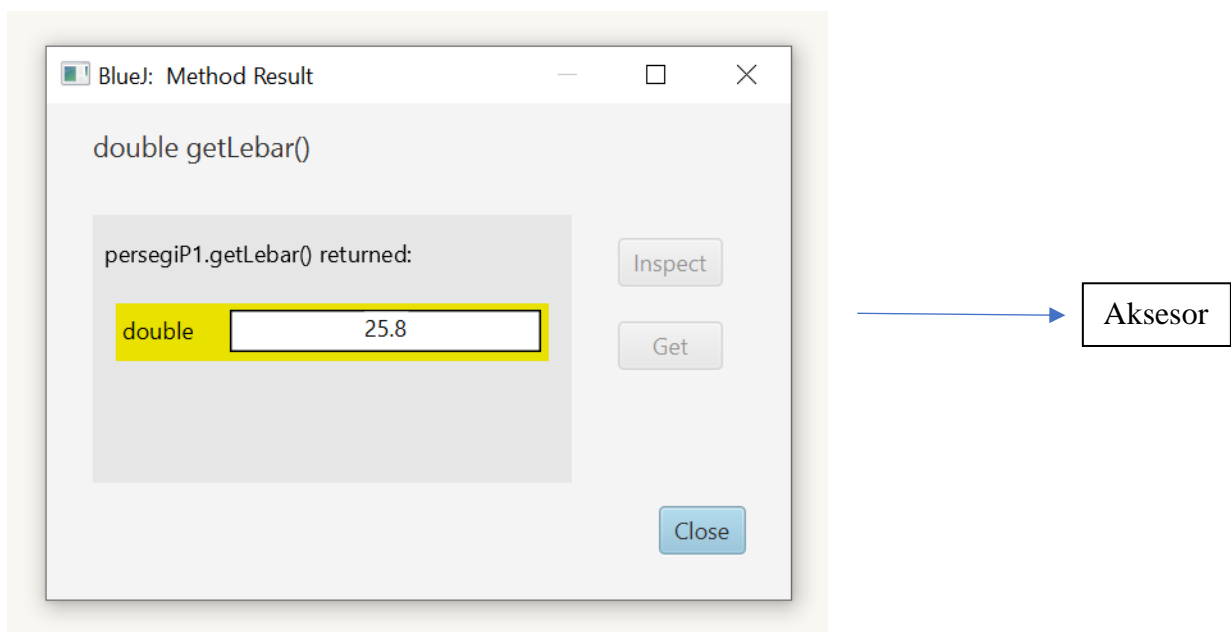
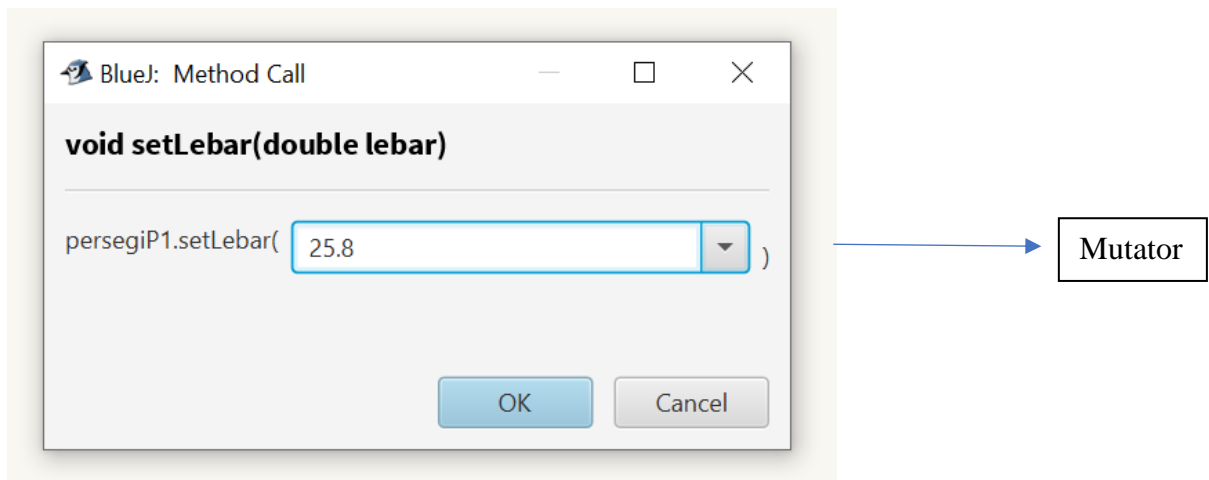


23. Setelah kita menambahkan konstruktor dan method pada kelas PersegiPanjang. Janganlupa kita juga bisa menambahkan aksesor dan mutator untuk dapat mengakses dan memanipulasi value dari masing-masing atribut.



24. Jika sudah menambahkan Aksesor dan mutator, **Compile** code kalian. Lalu cobalah untuk membuat objek baru bernama PP4, dengan menggunakan konstruktor berparameter New PersegiPanjang(double panjang, double lebar) dan

cobalah method aksesor dan mutator dari masing-masing atribut, dan juga mengakses method `luas()` dan `keliling()`.

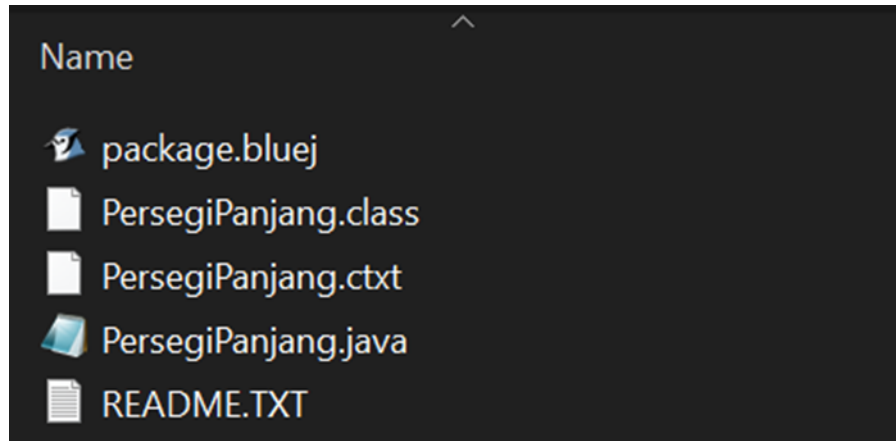


Jika kalian mempunyai waktu yang longgar dan ingin berlatih lagi kalian bebas

untuk menambah bangun datar yang lainnya. tidak perlu membuat project baru, cukup menambahkankelas baru saja.

### **Ketentuan Pengumpulan Guided**

1. Jika kalian mengikuti cara yang sama maka pada file yang terbentuk akan seperti pada gambar di bawah ini.



**Jumlah file tergantung dari berapa jumlah kelas yang kamu buat pada 1 project!**

2. Jadikan satu file-file tersebut ke dalam satu folder yang bernama GD3\_X\_YYYYY\_1
3. Kemudian zip folder tersebut dengan nama GD3\_X\_YYYYY\_1.zip

Keterangan

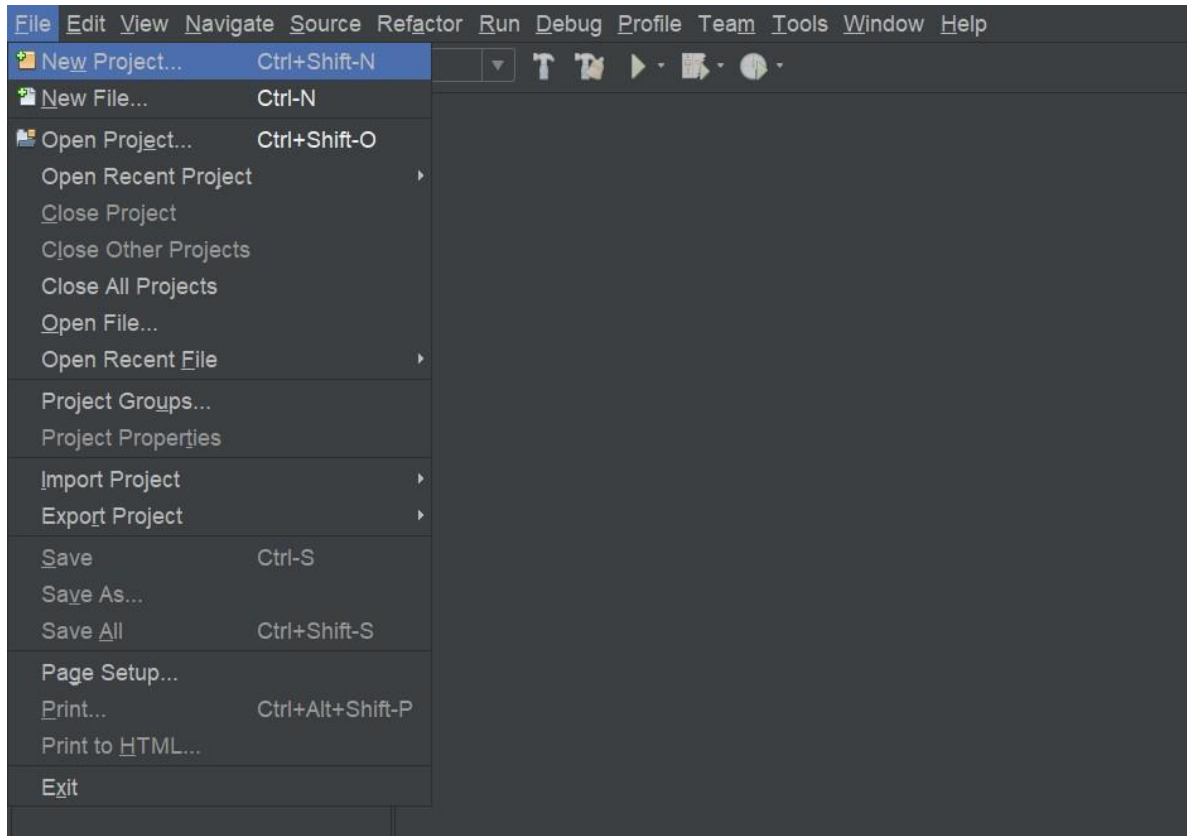
4. X = Kelas, YYYYYY = 5 Digit terakhir NPM

## GUIDED 1.2 – NETBEANS

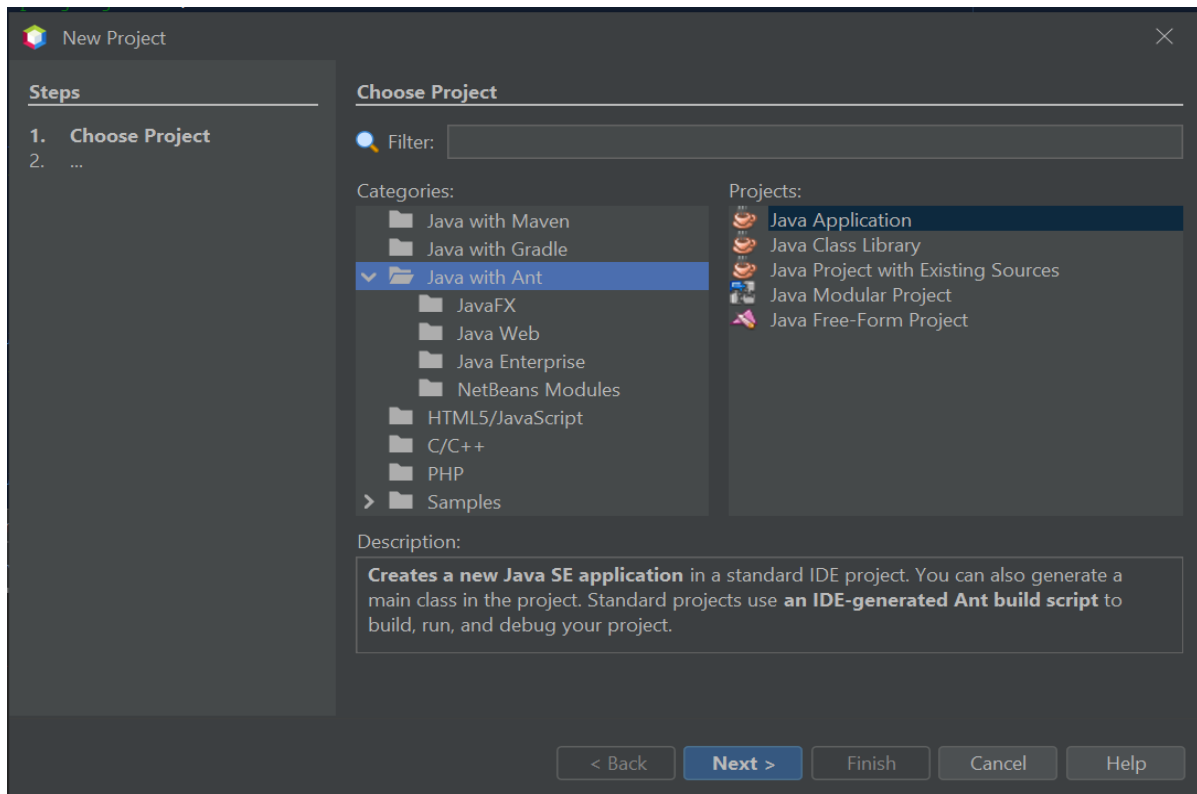
Pada Guided kedua ini kita akan mengimplementasikan code yang sudah kita tuliskan pada Guided 1 ke dalam netbeans.

### LANGKAH-LANGKAH

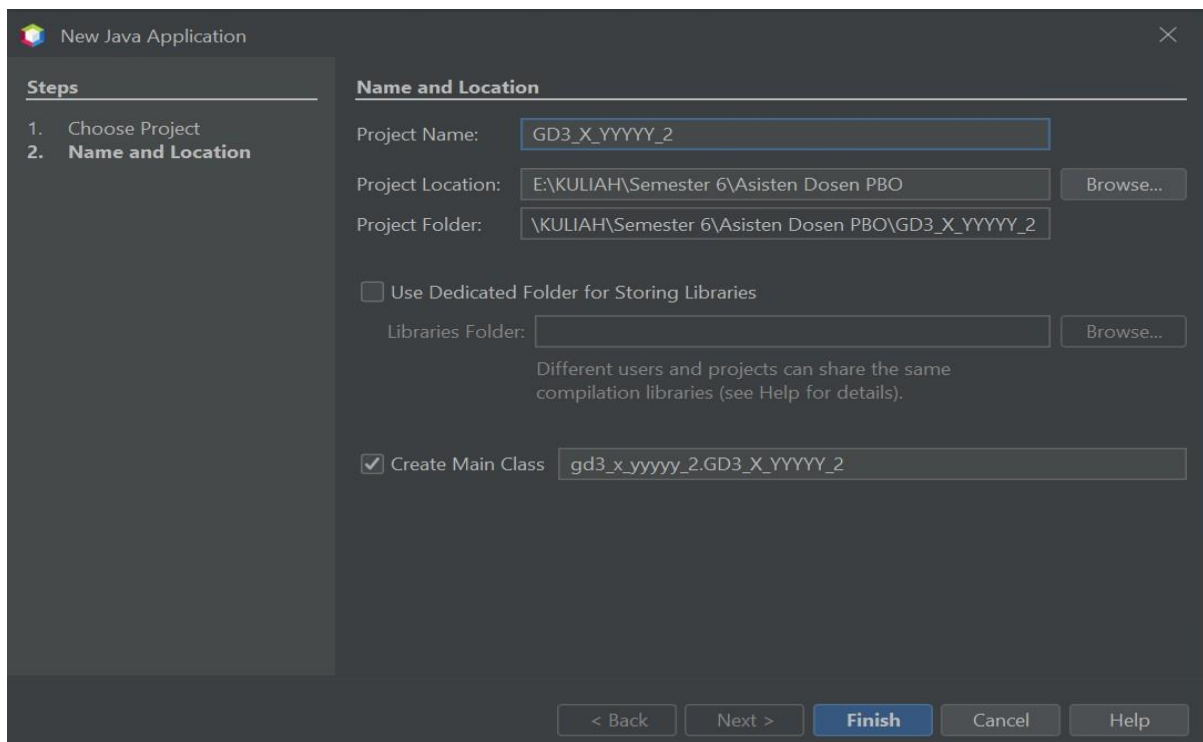
1. Buka NetBeans kalian, lalu klik File → New Project



2. Kemudian pada bagian **Choose Project** kalian pilihlah Java Application, lalu klik Next

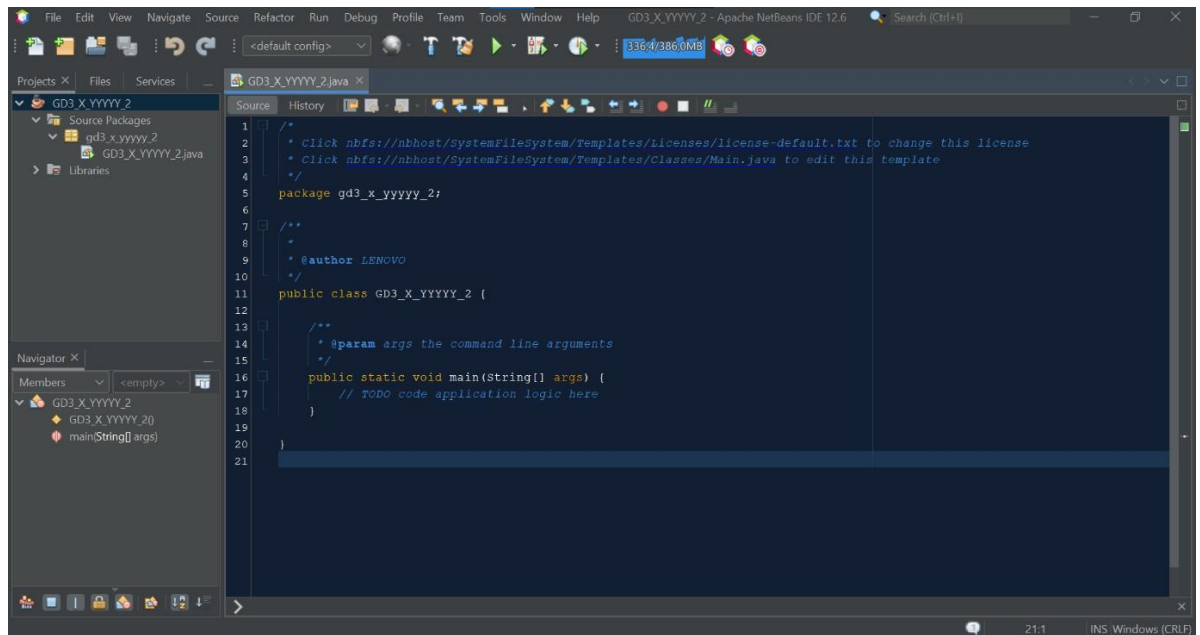


3. Isi bagian **Project Name** dengan nama **GD3\_X\_YYYYY\_2** (X=Kelas, YYYYYY=5Digit terakhir NPM)



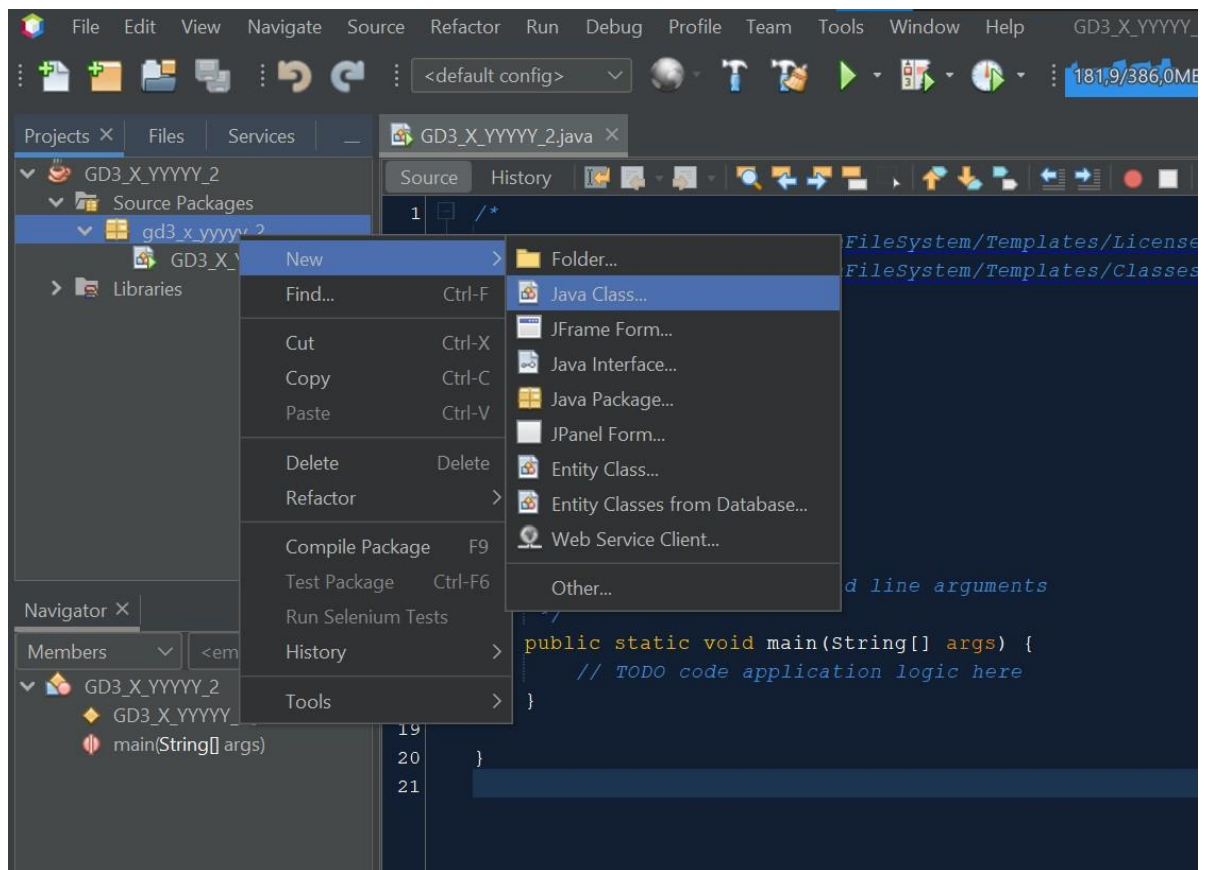
Pastikan bahwa “Create Main Class” kalian sudah dicentang. Ini bertujuan untuk kalian langsung dibuatkan main class pada saat membuat project baru.

4. Jika sudah Klik **Finish**
5. Jika sudah maka kalian akan dibawa ke project baru yang kalian buat. Kurang lebihnya akan seperti ini.

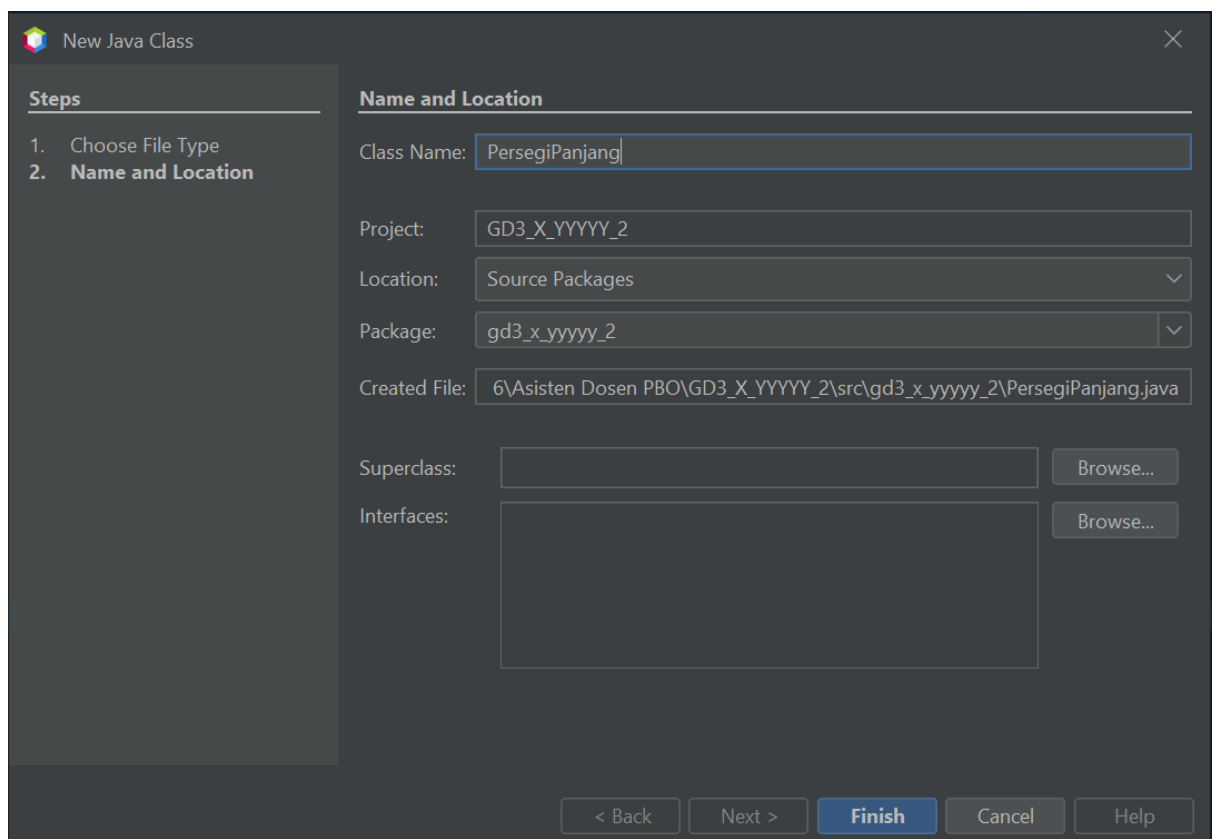


6. Sekarang kita akan membuat kelas baru bernama PersegiPanjang, Caranya Klik kanan pada package → New → Java Class

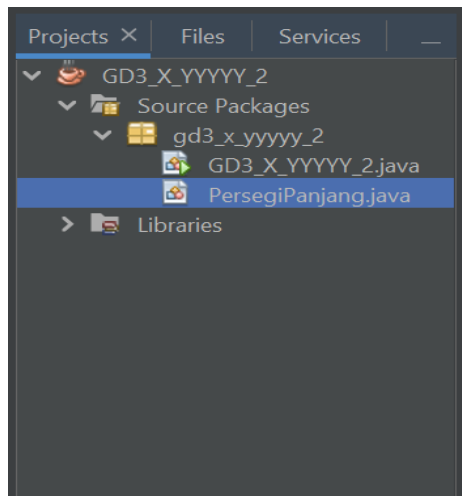




7. Kemudian akan muncul jendela. Isi **Class name** dengan nama PersegiPanjang, jikasudah Klik **Finish**.



8. Maka pada bagian package di sebelah kiri akan terbentuk kelas baru dengan namaPersegiPanjang.java



9. Lalu buka file PersegiPanjang.java tersebut lalu ketikkan code seperti di bawah ini

```
1 package gd1_x_yyyyy_2;
2
3 public class PersegiPanjang {
4     private double panjang;
5     private double lebar;
6
7     // Konstruktor default
8     public PersegiPanjang() {
9         panjang = 15;
10        lebar = 5;
11    }
12
13    // Konstruktor dengan parameter
14    public PersegiPanjang(double panjang, double lebar) {
15        this.panjang = panjang;
16        this.lebar = lebar;
17    }
18
19    // Method Keliling Persegi Panjang
20    public double keliling() {
21        return 2*(panjang+lebar);
22    }
23
24    // Method Luas Persegi Panjang
25    public double luas() {
26        return panjang*lebar;
27    }
28
29    // AKSESOR dan MUTATOR atribut-atribut
30    public double getPanjang() {
31        return panjang;
32    }
33
34    public void setPanjang(double panjang) {
35        this.panjang = panjang;
36    }
37
38    public double getLebar() {
39        return lebar;
40    }
41
42    public void setLebar(double lebar) {
43        this.lebar = lebar;
44    }
45 }
46
```

10. Kita menuju ke main class (GD3\_X\_YYYYY\_2.java). Deklarasikan 2 variable di mainkalian, panjang dan lebar dengan tipe data double.

```
1 package gd3_x_YYYYY_2;
2
3 import java.util.Scanner;
4
5 public class GD3_X_YYYYY_2 {
6
7     public static void main(String[] args) {
8         double panjang;
9         double lebar;
10        Scanner reader = new Scanner(System.in);
11    }
12
13 }
14
```

Ada beberapa cara untuk menerima input, salah satunya adalah menggunakan **Scanner**. Scanner ini merupakan suatu kelas yang terdapat pada library java. Jika ingin menggunakan Scanner **JANGAN LUPA UNTUK IMPORT**.

11. Buatlah 2 Objek dari kelas PersegiPanjang yang bernama PP1 dan PP2

```
1 package gd3_x_YYYYY_2;
2
3 import java.util.Scanner;
4
5 public class GD3_X_YYYYY_2 {
6
7     public static void main(String[] args) {
8         double panjang;
9         double lebar;
10        Scanner reader = new Scanner(System.in);
11
12        // Buat Objek PP1, menggunakan konstrkutor default
13        PersegiPanjang PP1 = new PersegiPanjang();
14
15        //Buat Objek PP2, menggunakan konstruktor dengan parameter
16        PersegiPanjang PP2 = new PersegiPanjang(68.2, 35.3);
17
18    }
19 }
20
```

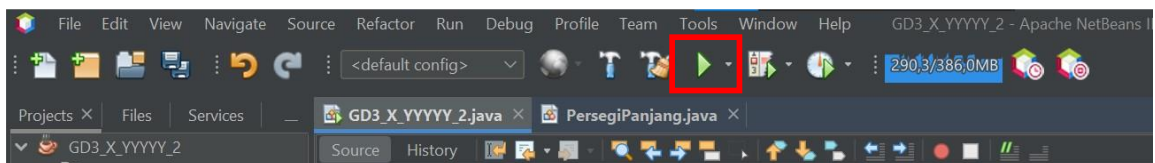
12. Kemudian ketikan code berikut ini di bawah code langkah ke-11. **Tetap di dalam main!**

```

17
18     System.out.println("---Persegi Panjang PP1 Lama---");
19     System.out.println("Panjang PP1 : "+PP1.getPanjang());
20     System.out.println("Lebar PP1 : "+PP1.getLebar());
21     System.out.println("Keliling PP1 : "+PP1.keliling());
22     System.out.println("Luas PP1 : "+PP1.luas());
23     System.out.println("-----\n");
24     System.out.println("---Persegi Panjang PP2 Lama---");
25     System.out.println("Panjang PP2 : "+PP2.getPanjang());
26     System.out.println("Lebar PP2 : "+PP2.getLebar());
27     System.out.println("Keliling PP2 : "+PP2.keliling());
28     System.out.println("Luas PP2 : "+PP2.luas());
29     System.out.println("-----\n");
30
31     System.out.println("--- Input Panjang dan Lebar BARU ---");
32     System.out.println("Masukkan nilai panjang yang baru : ");
33     panjang = reader.nextDouble();
34
35     System.out.println("Masukkan nilai lebar yang baru : ");
36     lebar = reader.nextDouble();
37
38     // Mengubah value atribut dari object PP2 melalui mutator/setter
39     PP2.setPanjang(panjang);
40     PP2.setLebar(lebar);
41
42     System.out.println("\n---Persegi Panjang PP1 Baru---");
43     System.out.println("Panjang PP1 baru : "+PP1.getPanjang());
44     System.out.println("Lebar PP1 baru : "+PP1.getLebar());
45     System.out.println("Keliling PP1 baru : "+PP1.keliling());
46     System.out.println("Luas PP1 baru : "+PP1.luas());
47     System.out.println("-----");
48
49     System.out.println("\n---Persegi Panjang PP2 Baru---");
50     System.out.println("Panjang PP2 baru : "+PP2.getPanjang());
51     System.out.println("Lebar PP2 baru : "+PP2.getLebar());
52     System.out.println("Keliling PP2 baru : "+PP2.keliling());
53     System.out.println("Luas PP2 baru : "+PP2.luas());
54     System.out.println("-----\n\n");

```

13. Jika sudah, **Run** program kalian dengan cara mengKlik tombol Play pada toolbar menudiatas. Bisa juga menggunakan **F6**.



14. Silakan cek value-value yang ada, apakah sesuai dengan apa yang kalian inputkan atautuliskan atau tidak, apakah Aksesori/getter mendapatkan value dari panjang dan lebar. Apakah Mutator/setter kalian dapat mengubah nilai sesuai yang kalian inputkan. Hasilnya kurang lebih akan seperti ini.

```
Output - GD1_X_YYYYY_2 (run) x
run:
---Persegi Panjang PP1 Lama---
Panjang PP1 : 15.0
Lebar PP1 : 5.0
Keliling PP1 : 40.0
Luas PP1 : 75.0
-----

---Persegi Panjang PP2 Lama---
Panjang PP2 : 68.2
Lebar PP2 : 35.3
Keliling PP2 : 207.0
Luas PP2 : 2407.46
-----

--- Input Panjang dan Lebar BARU ---
Masukkan nilai panjang yang baru :
14,4
Masukkan nilai lebar yang baru :
8,6

---Persegi Panjang PP1 Baru---
Panjang PP1 baru : 15.0
Lebar PP1 baru : 5.0
Keliling PP1 baru : 40.0
Luas PP1 baru : 75.0
-----

---Persegi Panjang PP2 Baru---
Panjang PP2 baru : 14.4
Lebar PP2 baru : 8.6
Keliling PP2 baru : 46.0
Luas PP2 baru : 123.84
-----
```

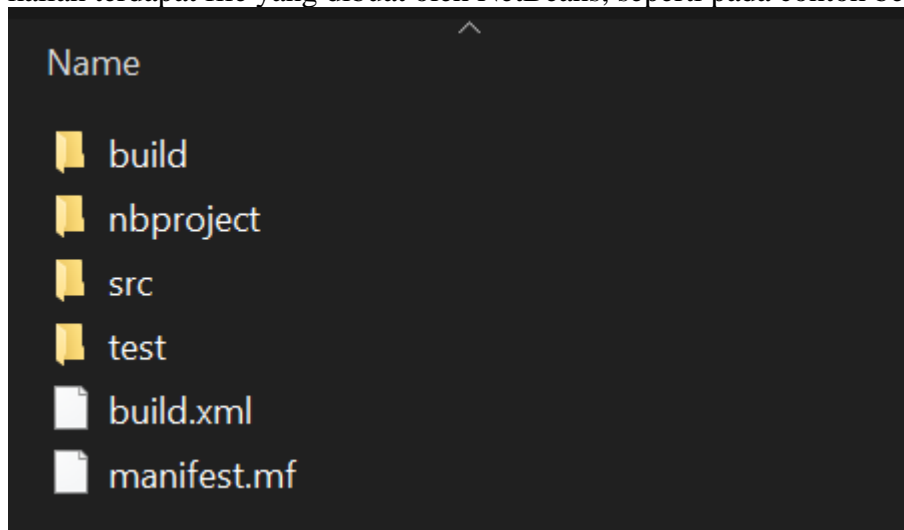
Dalam penulisan double jika kalian bisa menuliskan dengan titik(.) contoh 8.53 maka gunakanlah tanda titik, jika tidak bisa maka gunakanlah koma(,) seperti pada gambar

Dari kode di atas mengapa yang berubah hanya persegi panjang PP2? Karena kita hanya memanggil mutator/setter dari Objek PP2 saja, sedangkan PP1 kita tidak lakukan, maka dari itu value dari persegi panjang PP1 akan tetap seperti value di konstruktor default.

Jika kalian mempunyai waktu yang longgar dan ingin berlatih lagi kalian bebas untuk menambah bangun datar yang lainnya. tidak perlu membuat project baru, cukup menambahkan kelas baru saja

## Ketentuan Pengumpulan Guided

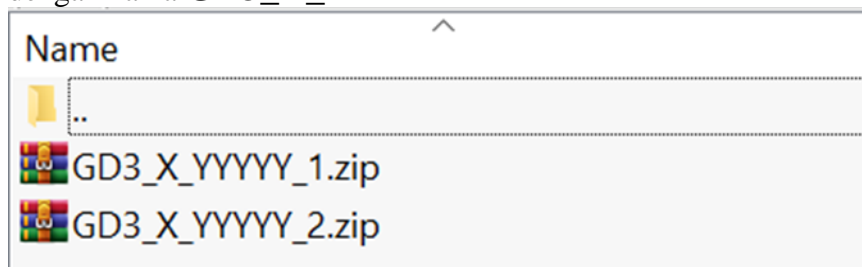
1. Ketika kalian sudah membuat project, maka didalam folder GD3\_X\_YYYYYY\_2 kalian terdapat file yang dibuat oleh NetBeans, seperti pada contoh berikut ini:



2. Zip lah folder GD3\_X\_YYYYYY\_2

## Ketentuan Pengumpulan Guided ke Situs Kuliah

1. **GD3\_X\_YYYYYY\_1.zip** dan **GD3\_X\_YYYYYY\_2.zip** dijadikan satu. Dizip lagi dengan nama **GD3\_X\_YYYYYY**



Isi di dalam GD3\_X\_YYYYYY