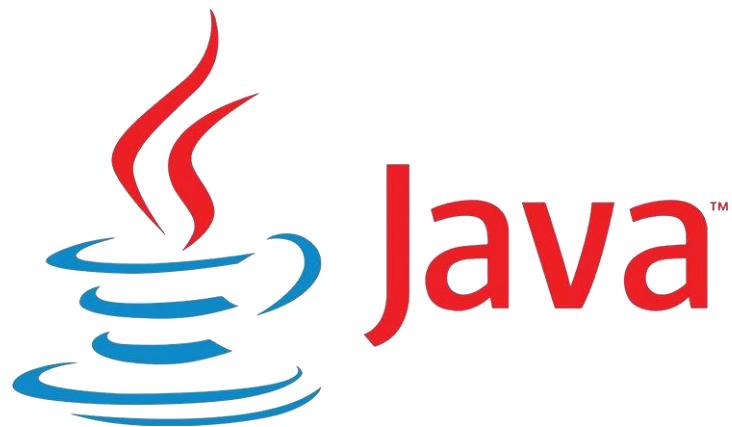


Pengenalan Tools Pemrograman Berorientasi Objek



Oleh Asisten Dosen Pemrograman Berorientasi Objek 2026/2027

Pendahuluan

Mata kuliah pemrograman berorientasi objek menggunakan 2 (dua) *Integrated Development Environment* (IDE) berupa Apache Netbeans dan BlueJ. Kedua IDE tersebut merupakan salah satu IDE Java yang cukup banyak digunakan untuk membantu pengembangan aplikasi dalam suatu proyek kecil hingga proyek yang cukup besar dan cenderung sudah dilengkapi dengan berbagai alat yang membantu *programmer* seperti *code editor*, *compiler*, *debugger*, *version control*, *testing*, dan peralatan lainnya yang dapat ditemukan dan dipelajari secara mandiri.

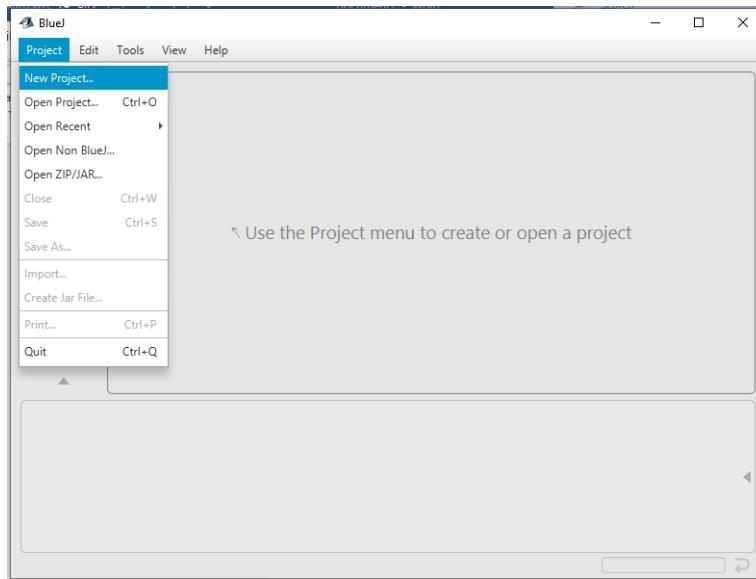
Bahasa pemrograman yang digunakan selama proses perkuliahan berlangsung merupakan bahasa pemrograman Java. Java merupakan salah satu bahasa pemrograman tingkat tinggi yang dirilis dan dikembangkan oleh Sun Microsystems sejak tahun 1995. Java disusun dengan menerapkan model *Object Oriented Programming* (OOP) yang akan berfokus pada penggunaan objek di dalam program. Java sendiri dapat digunakan untuk membuat berbagai jenis aplikasi *desktop*, *web*, ataupun *mobile*. Akan tetapi, fokus perkuliahan pemrograman berorientasi objek akan mengarah kepada aplikasi *desktop*.

Beberapa contoh aplikasi yang dikembangkan menggunakan bahasa pemrograman Java adalah Minecraft, Spotify, Netflix, Twitter dan aplikasi sehari-hari lainnya.

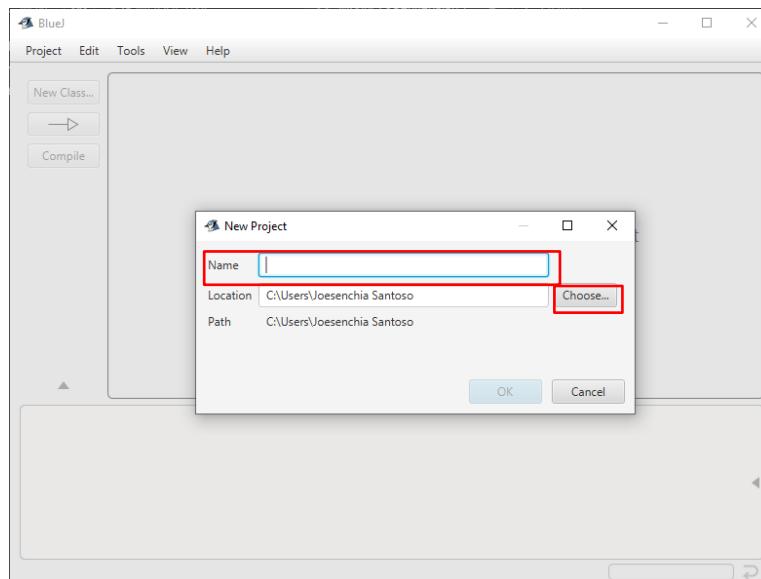
Pengenalan Aplikasi BlueJ

Membuat Proyek BlueJ

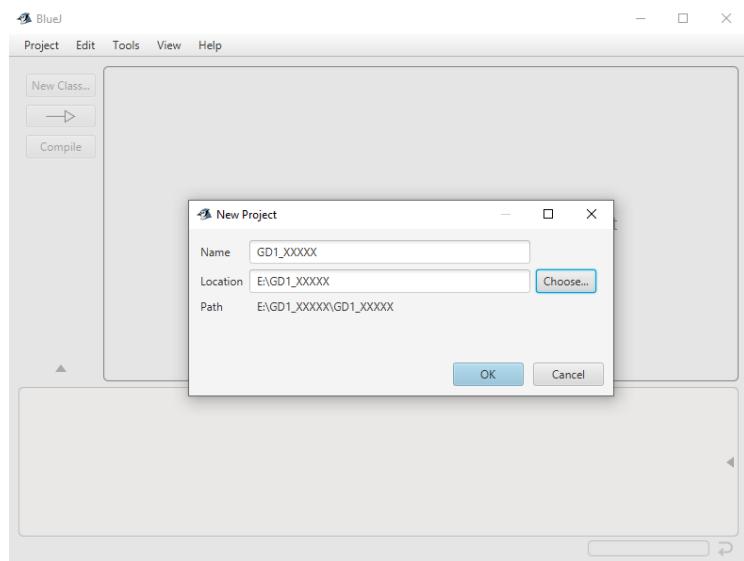
1. Silahkan membuka aplikasi BlueJ yang sudah terpasang untuk membuat proyek;
2. Silahkan klik tab *Project* kemudian pilih bagian *New Project*;



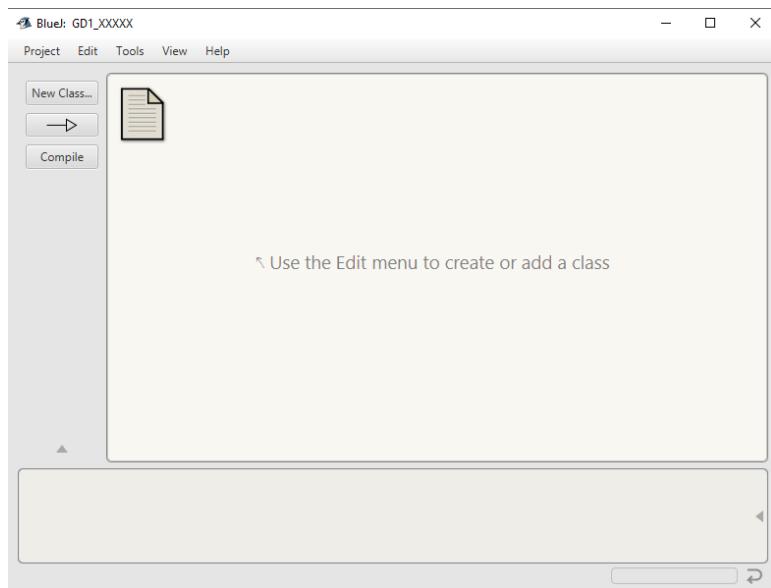
3. Silahkan mengisi nama proyek yang sudah ditentukan dan memilih lokasi sesuai dengan keinginan;



Jika sudah sesuai dengan yang diinginkan, silahkan klik *OK*;



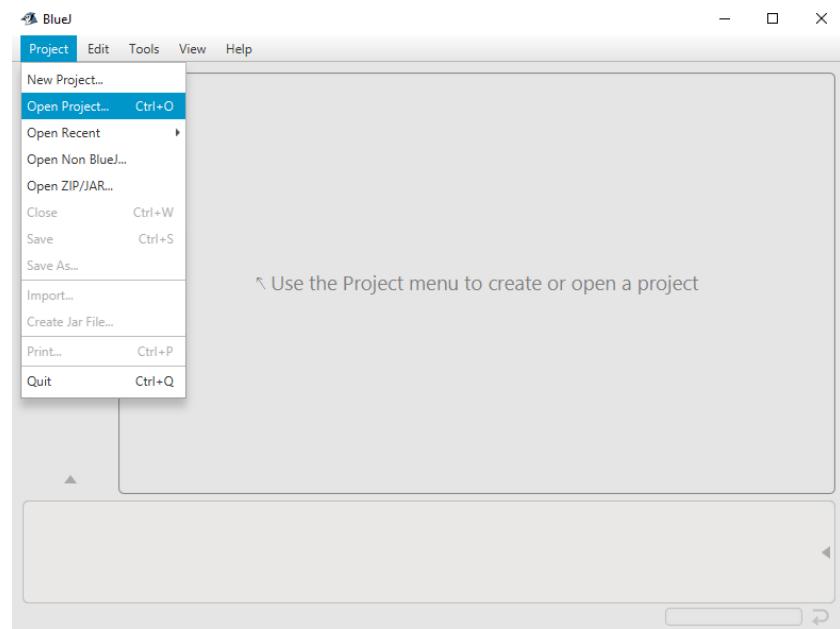
4. Apabila sudah terbuka tampilan seperti di bawah ini, selamat! Anda sudah dapat membuat proyek di dalam aplikasi BlueJ;



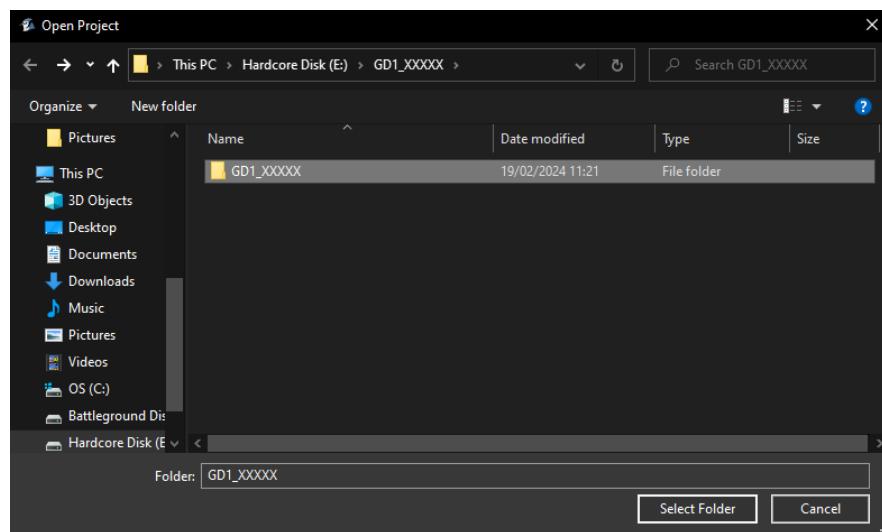
5. Enjoy :D.

Membuka Proyek BlueJ

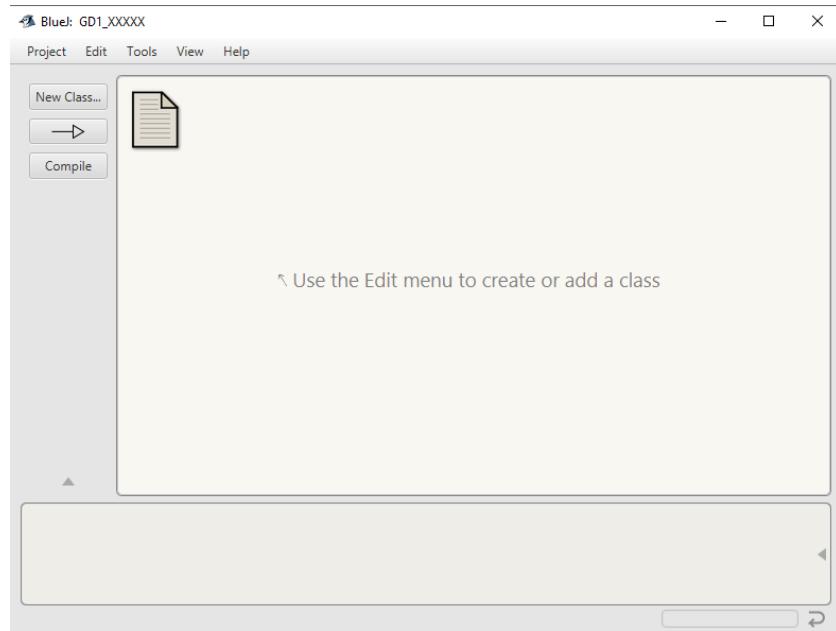
1. Silahkan membuka aplikasi BlueJ yang sudah terpasang untuk membuka proyek yang sudah dibuat;
2. Silahkan klik tab *Project* kemudian pilih *Open Projek* atau bisa langsung dilakukan melalui *shortcut key* *Ctrl + O*;



3. Silahkan memilih folder lokasi anda menyimpan proyek BlueJ yang ingin dibuka kembali. Kemudian klik *Select Folder*;



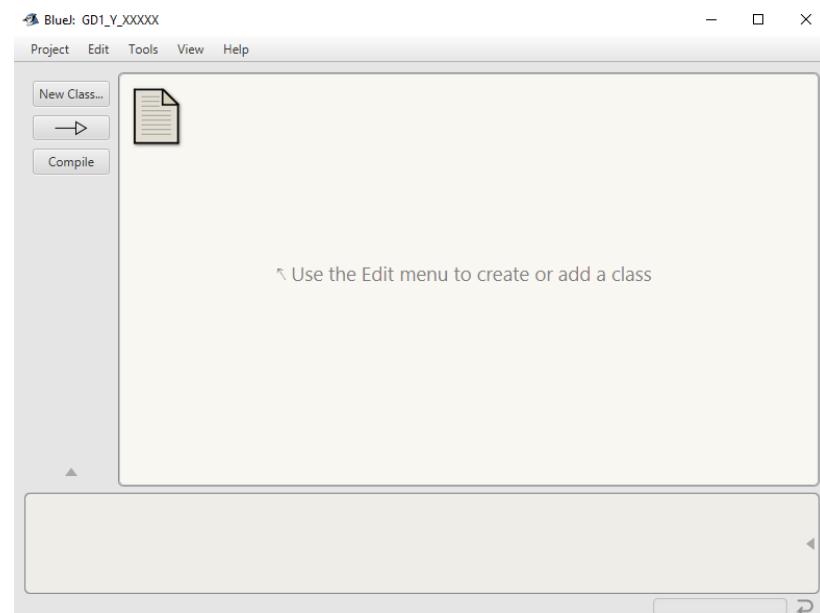
4. Apabila tampilan sudah berubah seperti di bawah ini, selamat! Anda sudah dapat membuka kembali proyek BlueJ;



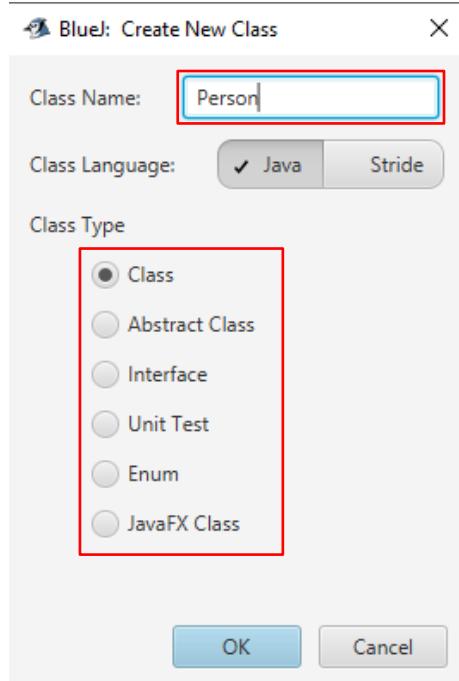
5. Enjoy :D.

Membuat Kelas Dalam Proyek BlueJ

1. Silahkan membuka proyek BlueJ yang masih kosong pada laptop anda;
2. Apabila sudah muncul tampilan seperti gambar di bawah ini. Klik pada bagian *New Class...*;

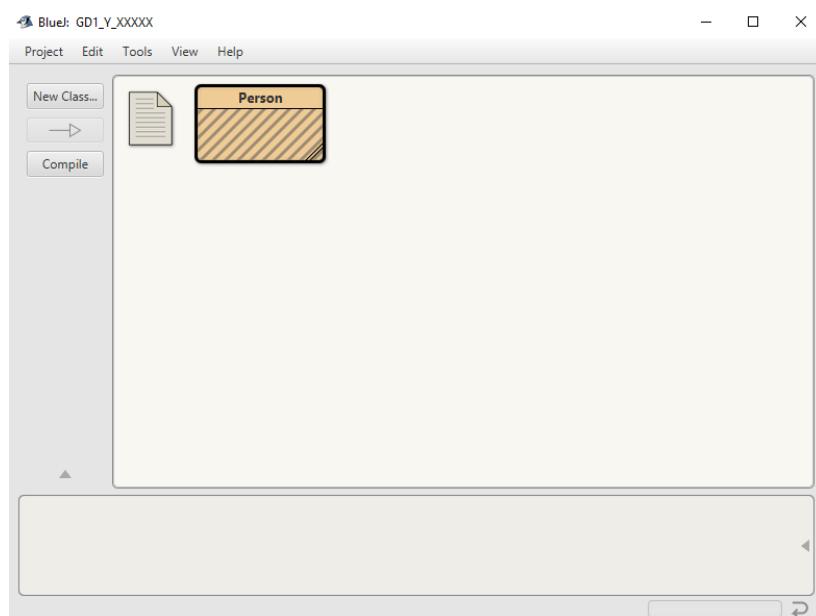


3. Pastikan tampilan yang muncul seperti gambar di bawah ini.

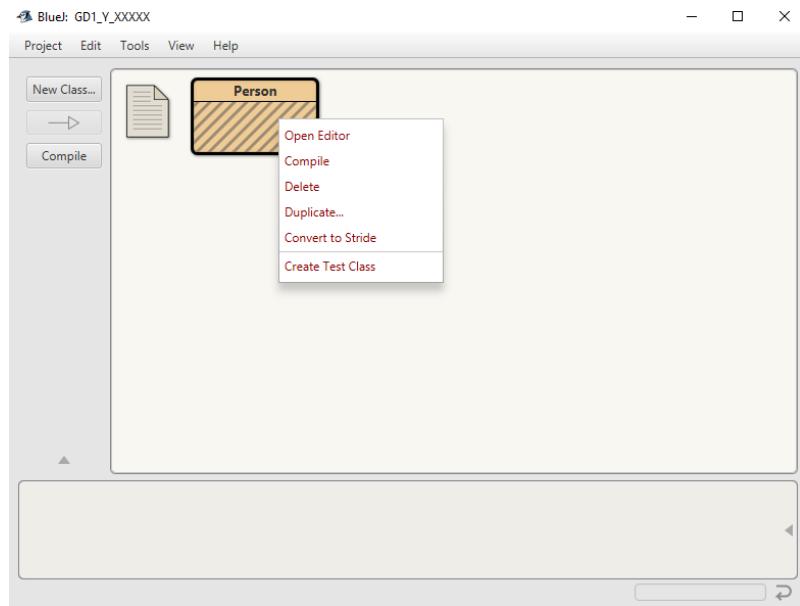


Anda sekalian dapat mengisi nama kelas sesuai dengan keinginan anda dan melakukan pemilihan terhadap bentuk kelas yang anda inginkan.

4. Pastikan tampilan yang muncul seperti gambar berikut ini.



Silahkan klik kanan pada bagian kelas yang sudah dibuat seperti pada gambar di bawah.



Silahkan klik pada *Open Editor* sehingga muncul tampilan seperti gambar di bawah ini;

A screenshot of the "Person.java" code editor window. The window has a title bar "Person - GD1_Y_XXXXX" and a menu bar with "Class", "Edit", "Tools", and "Options". Below that is a toolbar with "Compile", "Undo", "Cut", "Copy", "Paste", "Find...", and "Close" buttons. A dropdown menu "Source Code" is open. The main area contains the following Java code:

```
/** * Write a description of class Person here. * * @author (your name) * @version (a version number or a date) */public class Person {    // instance variables - replace the example below with your own    private int x;    /**     * Constructor for objects of class Person     */    public Person()    {        // initialise instance variables        x = 0;    }    /**     * An example of a method - replace this comment with your own     *     * @param y a sample parameter for a method     */}
```

The code is color-coded: comments are in green, keywords in blue, and variables in red.

5. Silahkan tuliskan kode sebagai berikut ini.

The screenshot shows a Java IDE window titled "Person - GD1_Y_XXXXX". The menu bar includes "Class", "Edit", "Tools", and "Options". A toolbar below the menu has buttons for "Compile", "Undo", "Cut", "Copy", "Paste", "Find...", and "Close". A dropdown menu "Source Code" is open. The main area contains the following Java code:

```
public class Person
{
    // instance variables - replace the example below with your own
    private int a;
    private int b;

    /**
     * Constructor for objects of class Person
     */
    public Person()
    {
        a = 3;
        b = 2;
    }

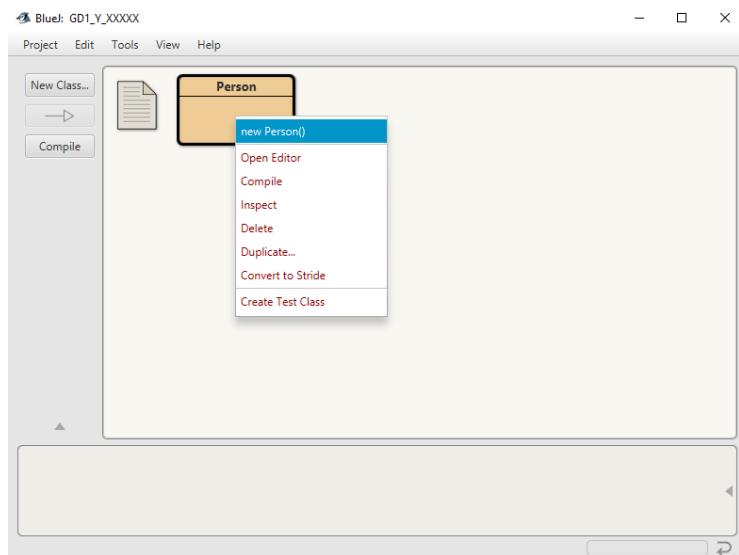
    /**
     * An example of a method - replace this comment with your own
     *
     * @param y  a sample parameter for a method
     * @return   the sum of x and y
     */
    public int penjumlahan()
    {
        // put your code here
        return a + b;
    }
}
```

At the bottom of the IDE window, a status bar displays "Class compiled - no syntax errors" and "saved".

Setelah selesai menuliskan kode, silahkan klik pada bagian *Compile* dan tunggu hingga selesai;

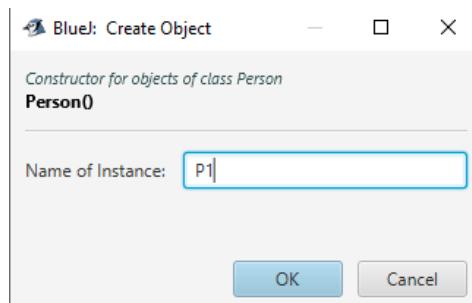
The screenshot shows the same Java IDE window as before, but the "Compile" button in the toolbar is now highlighted with a red box. The rest of the interface and code are identical to the first screenshot.

6. Kembali pada jendela tampilan awal lalu klik kanan pada kelas Person.



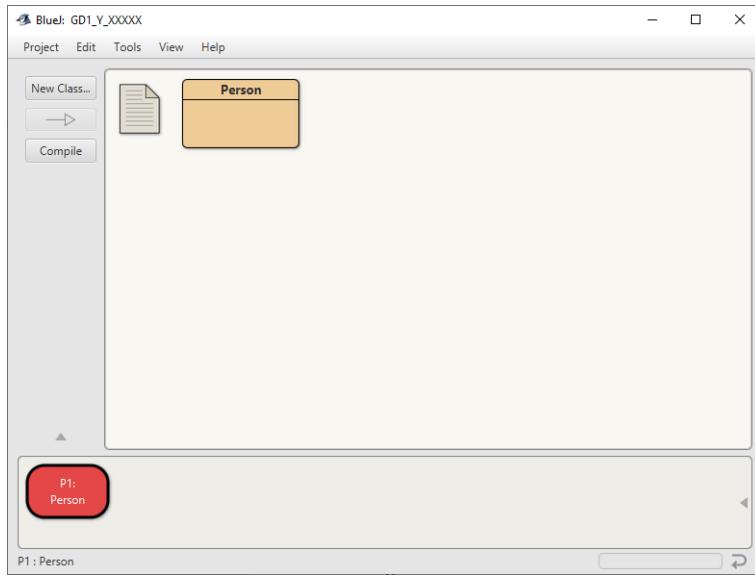
Kita akan membuat objek dari kelas Person. Silahkan pilih pada bagian *new Person()* yang akan memanggil konstruktor pada kelas Person yang telah kita buat;

7. Silahkan isi dengan nama sesuai dengan keinginan anda.



Kemudian silahkan klik *OK*;

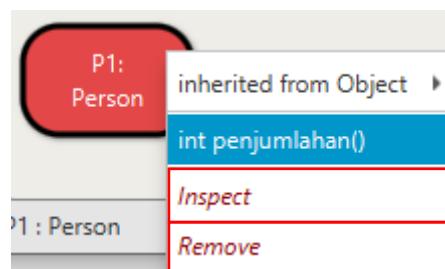
8. Tampilan akan berubah menjadi seperti di bawah ini.



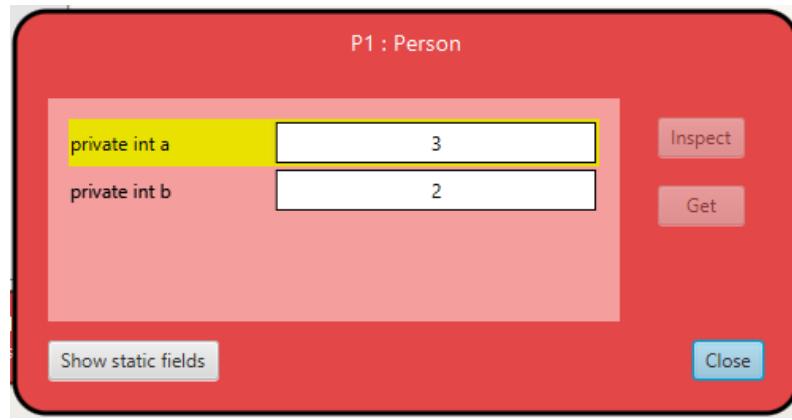
Bagian yang berwarna merah menandakan bahwa sebuah objek sudah berhasil dibentuk. Silahkan klik kanan pada bagian yang berwarna merah sehingga muncul tampilan sebagai berikut.



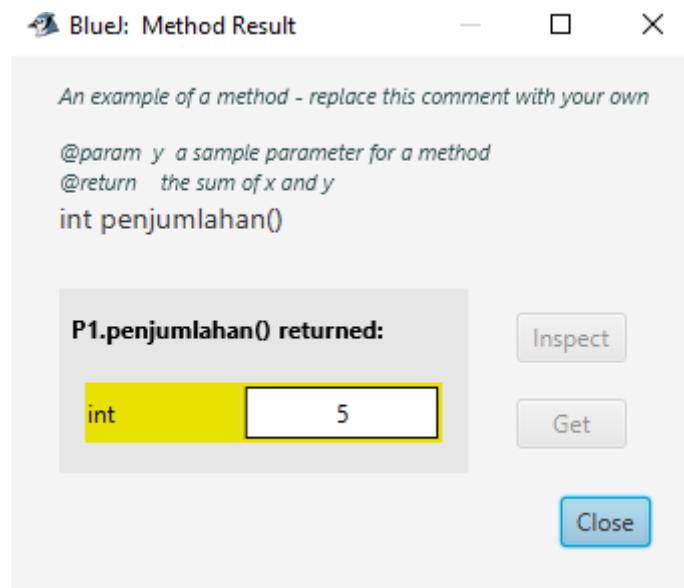
Pada menu *dropdown*, terdapat beberapa pilihan menu yang dapat diakses sesuai dengan kebutuhan *programmer*. Bagian *Inspect* akan menampilkan atribut yang sudah dibuat beserta dengan nilainya. Bagian *Remove* akan melakukan penghapusan terhadap objek yang sudah dibuat.



Berikut ini merupakan tampilan pada bagian *Inspect*.



Silahkan klik pada bagian *int penjumlahan()* untuk melihat hasil penjumlahan yang sudah anda buat sebelumnya seperti pada tampilan di bawah ini.

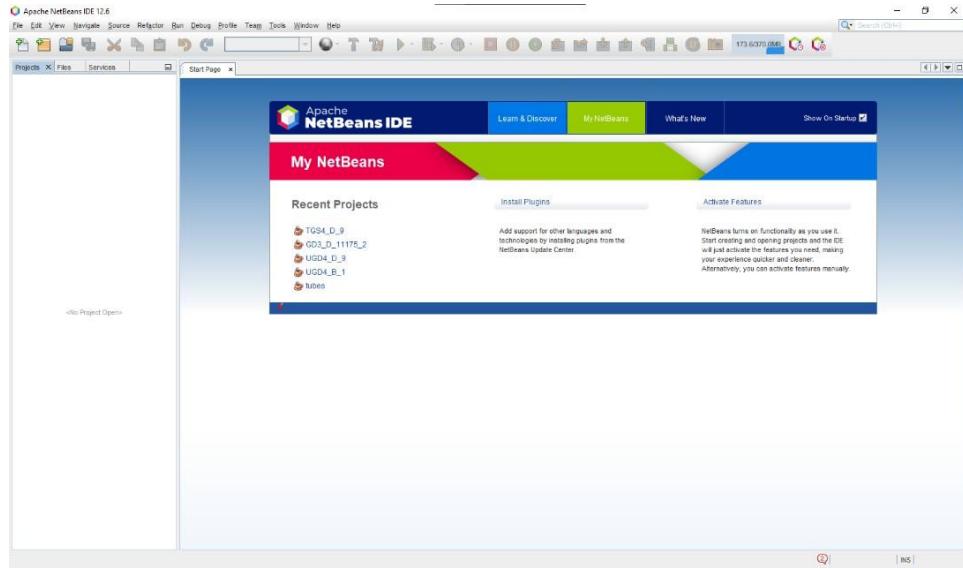


9. Selamat! Anda sudah dapat menggunakan aplikasi BlueJ sebagai sarana untuk membuat program berbahasa pemrograman Java;
10. Enjoy :D.

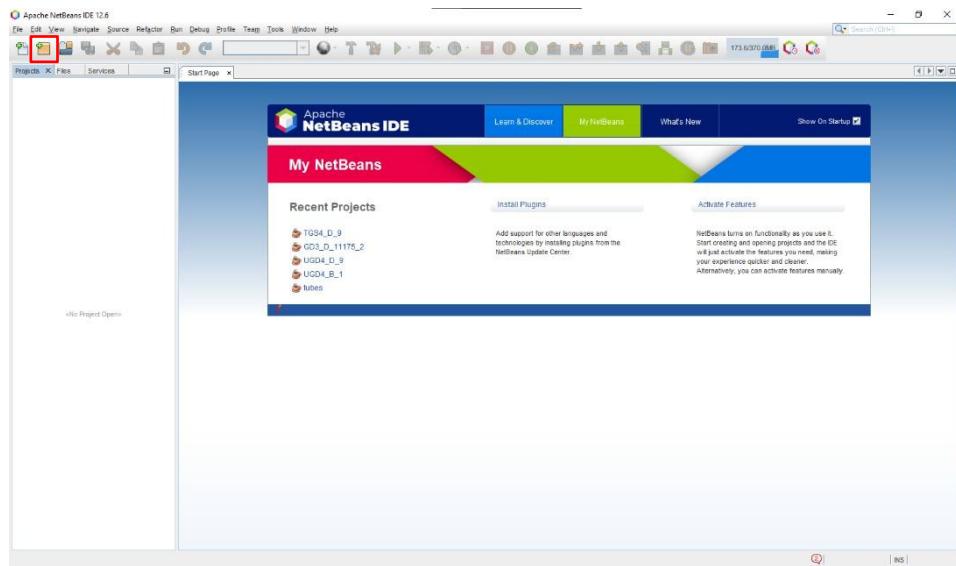
Pengenalan Aplikasi NetBeans

Membuat Projek Netbeans

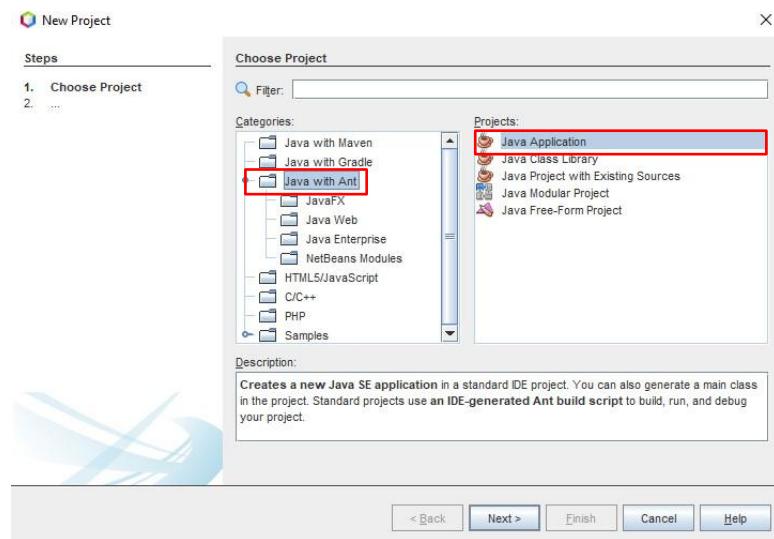
1. Silahkan membuka aplikasi pada NetBeans pada laptop masing-masing;
2. Tampilan awal akan berupa seperti gambar di bawah ini.



Silahkan pilih ikon yang bertanda merah;

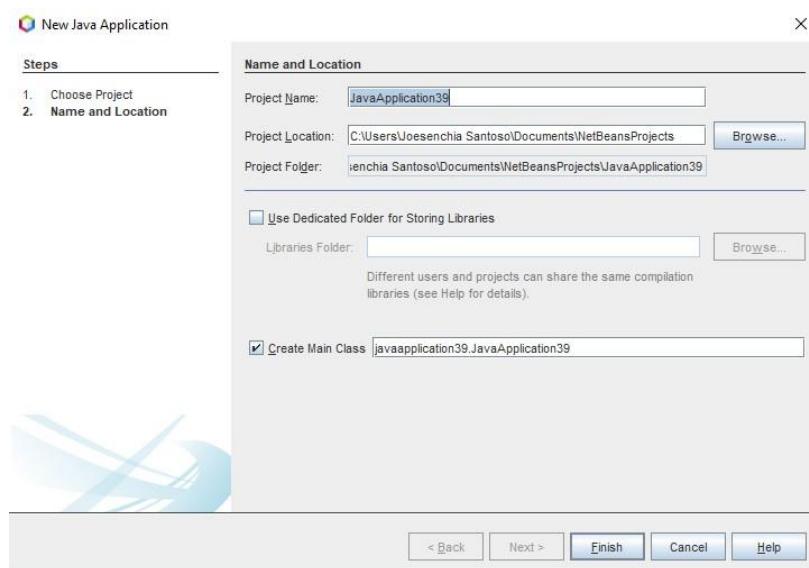


3. Jika sudah muncul tampilan seperti gambar di bawah ini, silahkan pilih sesuai dengan pilihan pada gambar.

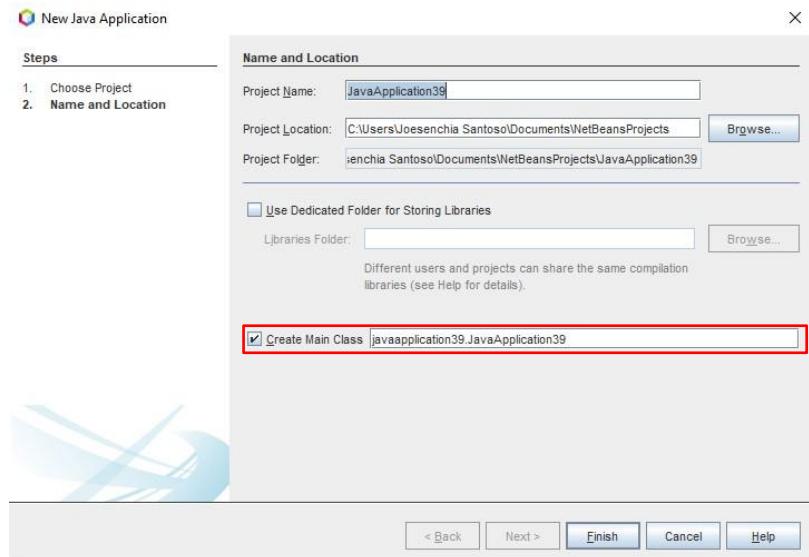


Kemudian silahkan klik *Next/Lanjut*;

4. Apabila aplikasi meminta anda untuk melakukan instalasi tambahan, ikuti saja sesuai dengan keterangan pada bagian tersebut dan tunggu hingga benar-benar selesai (pastikan koneksi internet menyala dan stabil)
5. Apabila sudah muncul tampilan seperti di bawah ini, silahkan mengisi nama proyek dan lokasi penyimpanannya.

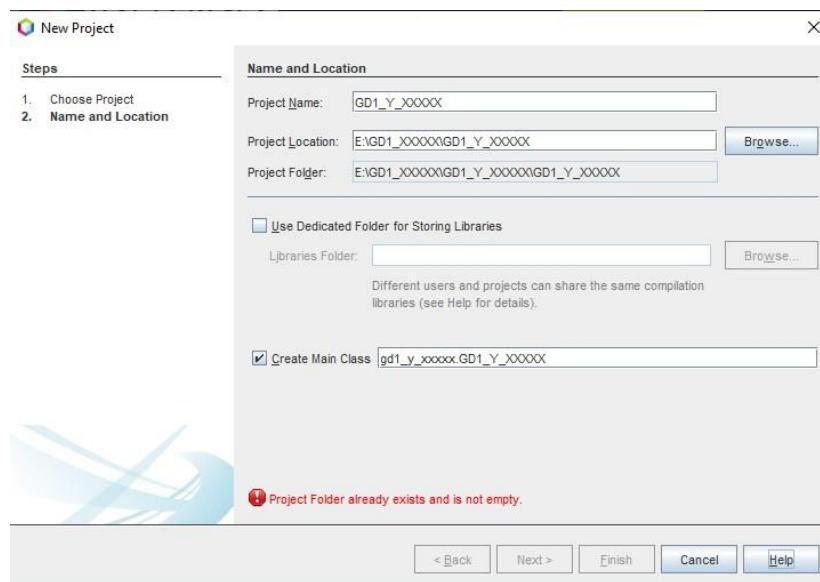


Apabila anda tidak ingin membuat *Main Class* secara otomatis, silahkan hilangkan tanda centang pada bagian yang bertanda merah. Akan tetapi, disarankan untuk tidak menghilangkan tanda centang kecuali diatur dalam ketentuan modul.

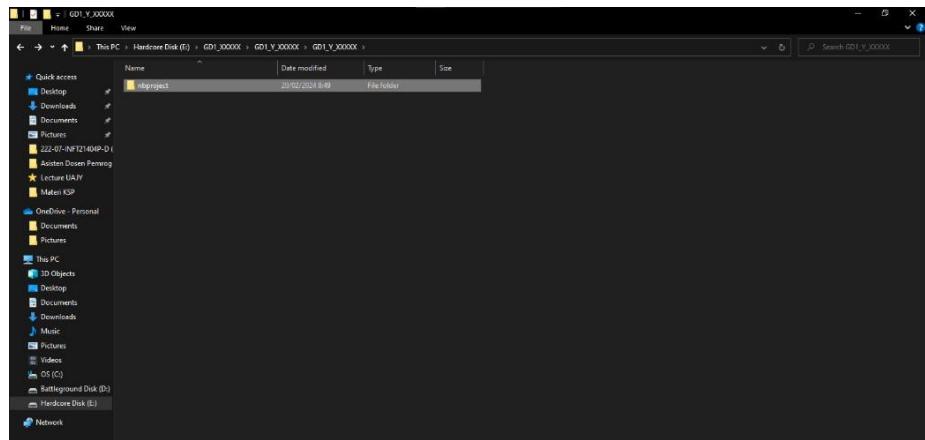


Setelah itu, silahkan klik *Finish*;

6. Apabila tampilan berubah menjadi seperti ini, silahkan tutup tab dengan klik *Cancel* atau tanda X pada pojok atas kanan.



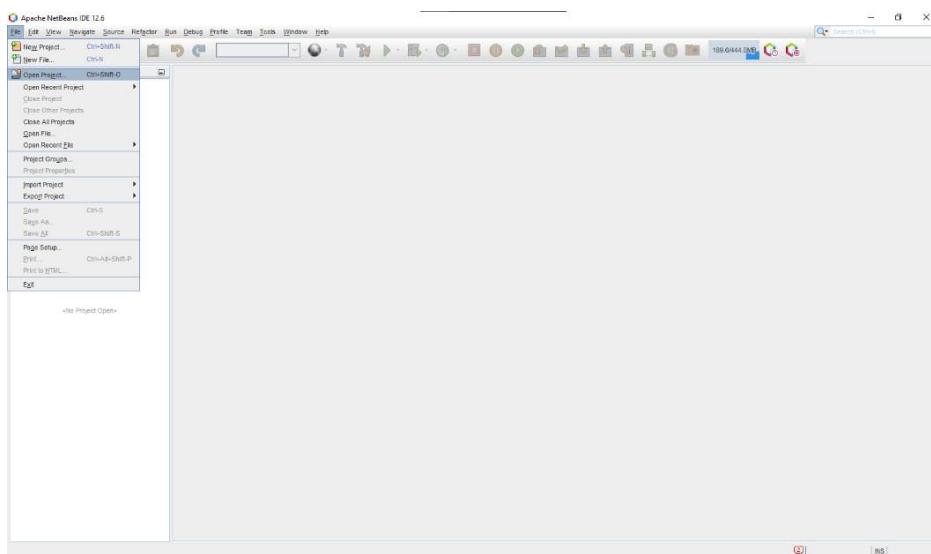
Hal tersebut menandakan proyek sudah berhasil dibuat pada lokasi folder yang anda inginkan sehingga terlihat seperti tampilan di bawah ini.



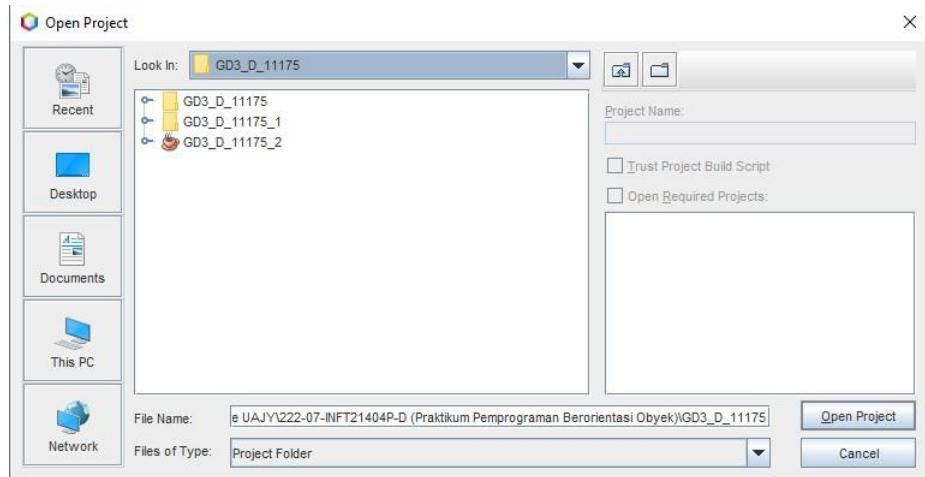
7. Enjoy :D.

Membuka Proyek Netbeans

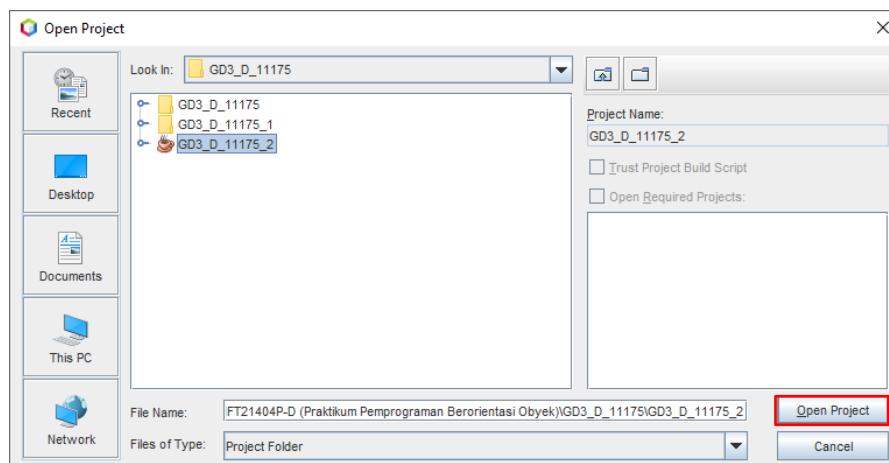
1. Silahkan membuka aplikasi Netbeans pada laptop masing-masing;
2. Pilih tab *File* lalu klik *Open Project* sesuai pada tampilan di bawah;



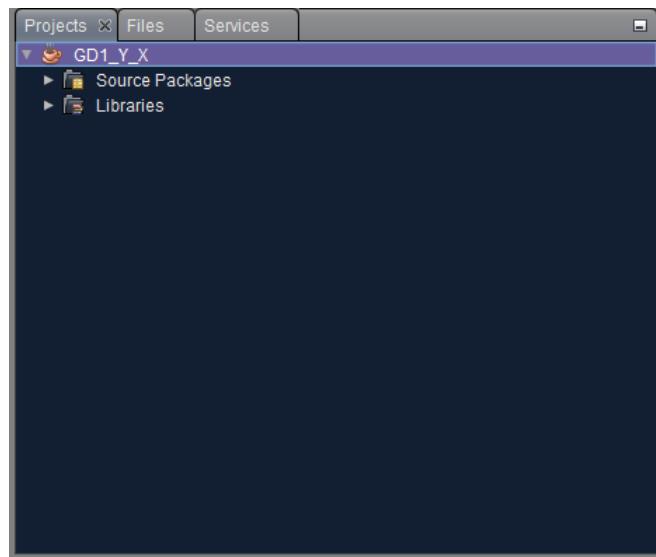
3. Apabila sudah muncul tampilan seperti gambar di bawah. Silahkan klik pada bagian yang memiliki ikon cangkir kopi yang ada pada *storage* anda.



Kemudian klik *Open Project*;



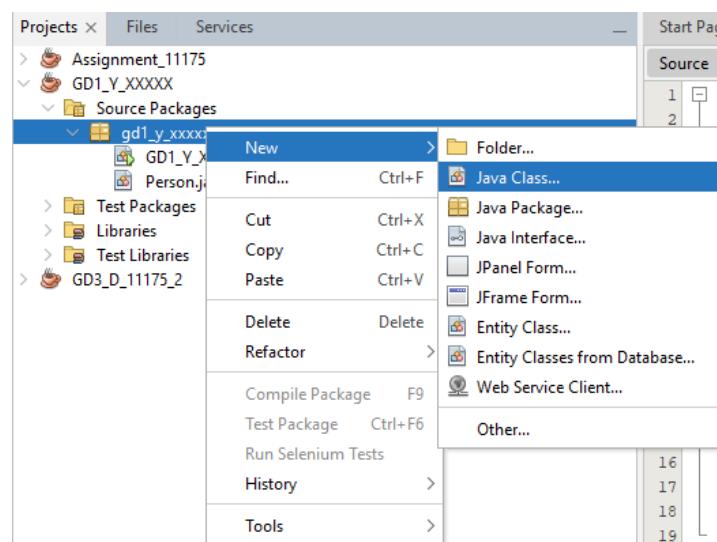
4. Apabila sudah muncul tampilan seperti gambar di bawah ini pada bagian pojok kiri anda, selamat! Anda sudah dapat membuka proyek NetBeans.



5. Enjoy :D.

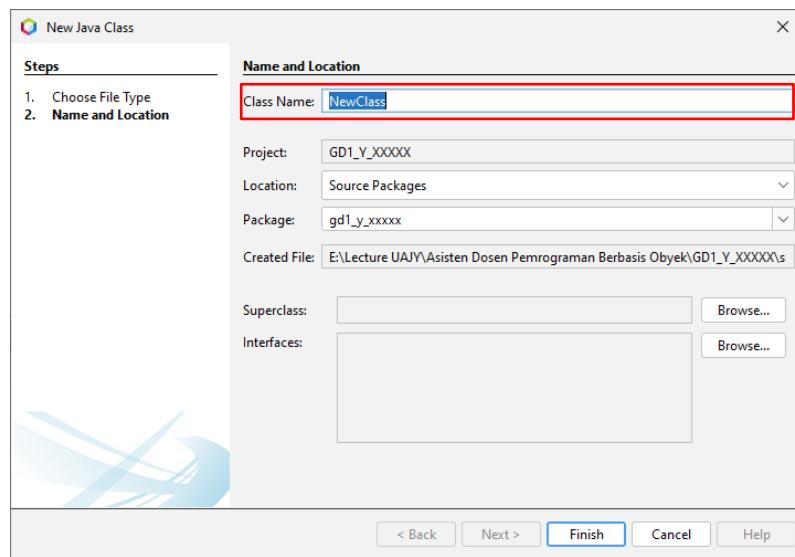
Membuat Class dan Main Class

1. Silahkan membuka proyek yang sudah pernah dibuat;
2. Silahkan membuka folder di dalam proyek hingga seperti tampilan di bawah ini;
3. Silahkan klik kanan pada bagian *package* sehingga tampil seperti di bawah ini.

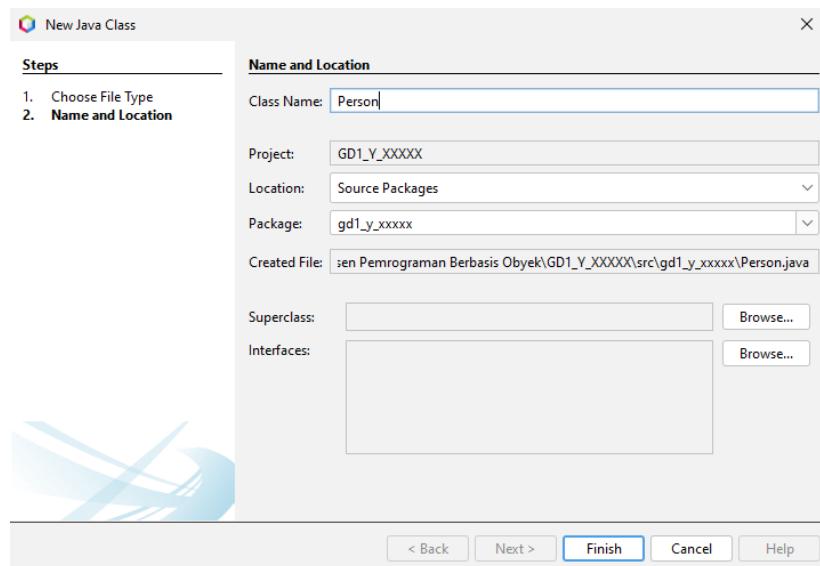


Silahkan klik pada bagian *New* kemudian pilih bagian *Java Class*;

4. Kemudian akan muncul jendela seperti tampilan di bawah ini.



Silahkan isi pada bagian *Class Name* sesuai dengan keinginan anda sekalian. Kemudian klik *Finish*;

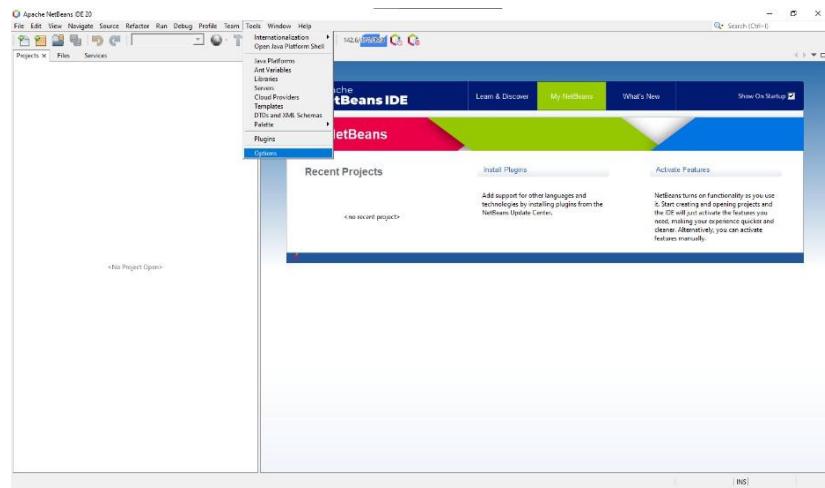


5. Selamat! Anda sudah dapat membuat sebuah kelas pada *package* di dalam proyek;
6. Enjoy :D.

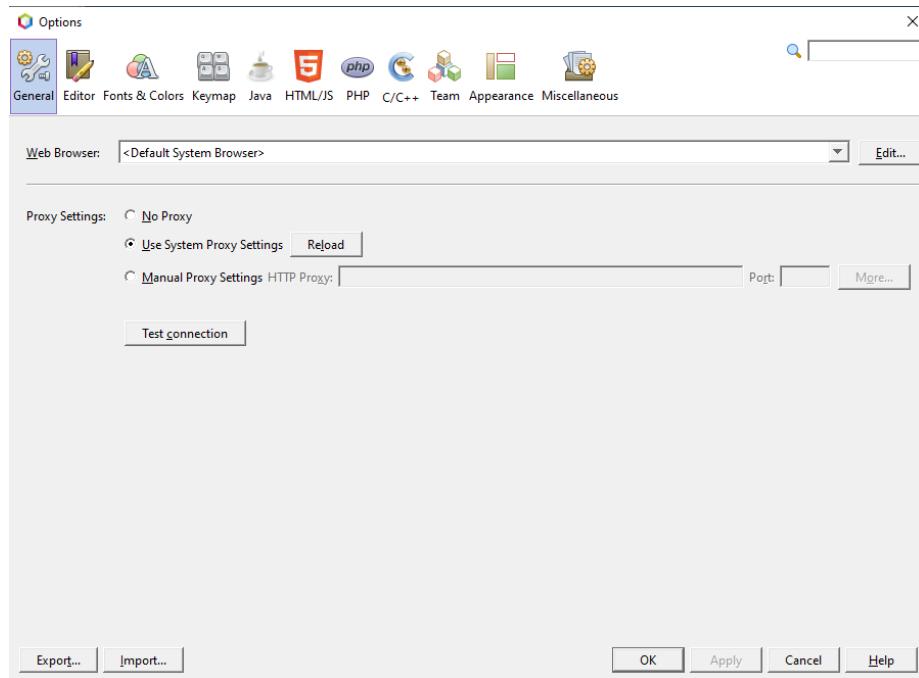
Mengganti Tema Aplikasi NetBeans

1. Silahkan membuka aplikasi NetBeans yang sudah terinstal pada laptop masing-masing;

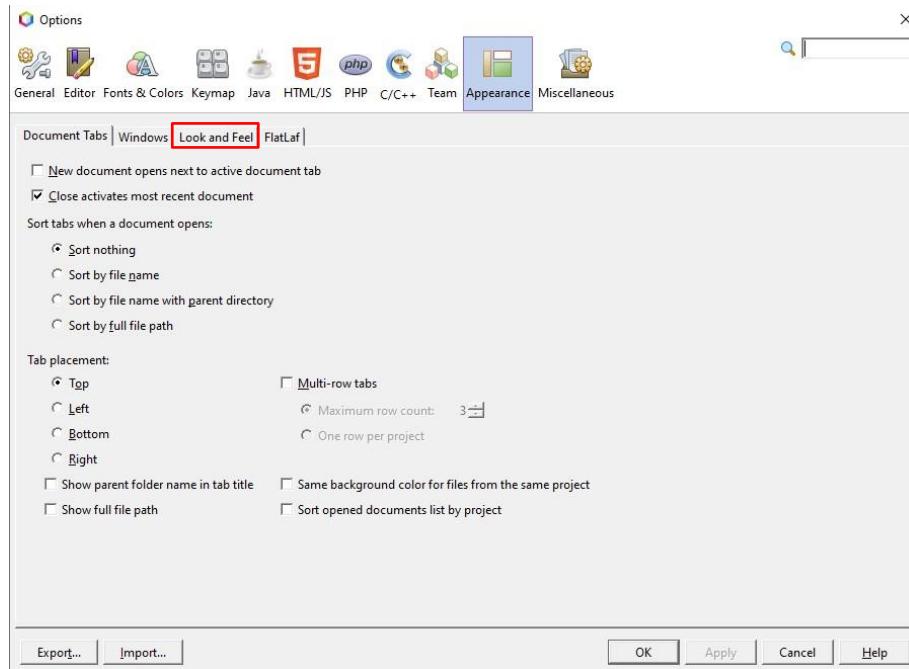
2. Pilih tab *Tools* dan klik pada bagian *Options*;



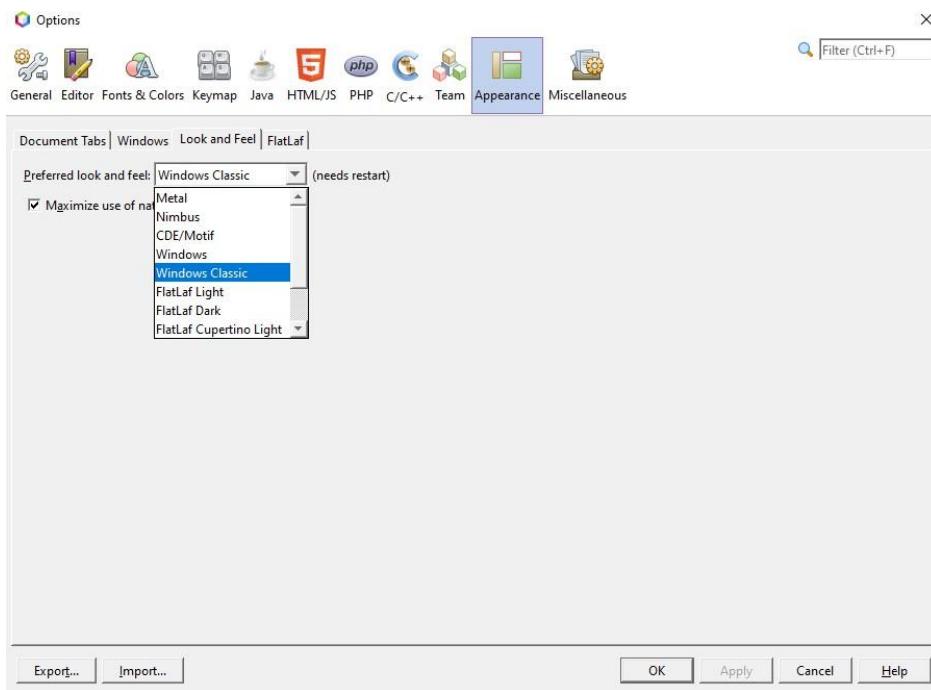
3. Setelah muncul tampilan seperti gambar di bawah ini, silahkan klik pada bagian *Appearance*;



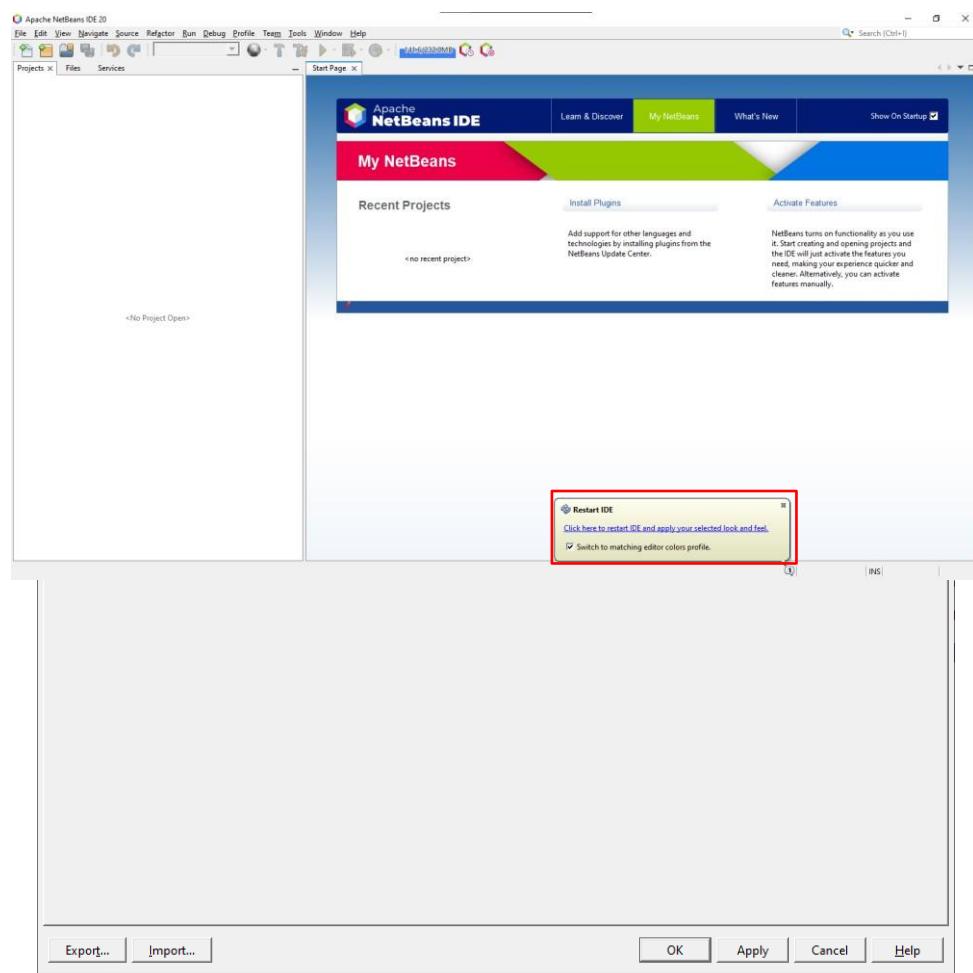
4. Klik pada tab *Look and Feel*;



5. Silahkan pilih sesuai dengan keinginan anda pada bagian dropdown. Kemudian klik *Apply*;



6. Lakukan *restart* pada aplikasi NetBeans secara manual atau langsung dengan melakukan klik pada balon notifikasi seperti gambar di bawah.



Silahkan tunggu beberapa saat hingga aplikasi terbuka kembali;

7. Enjoy :D.

Dasar-dasar Bahasa Java

Variabel

Variabel pada bahasa pemrograman Java digunakan untuk menyimpan sebuah nilai tertentu di dalam memori. Variabel tersebut dapat didefinisikan menggunakan sebuah nama dan tipe data yang spesifik. Variabel yang dikenal di dalam *sub-program* dikenal dengan nama variabel lokal, sedangkan variabel yang dikenal secara umum di dalam satu program disebut variabel global. Variabel yang digunakan oleh bahasa pemrograman Java memiliki persamaan dengan variabel yang digunakan dalam bahasa pemrograman C.

Syntax yang digunakan untuk melakukan pendeklarasian di dalam bahasa pemrograman Java adalah sebagai berikut.

```
tipedata nama_variabel;
```

Ataupun dapat dituliskan secara sambung dimulai dengan huruf kecil di depan kata pertama dan huruf kapital pada kata berikutnya seperti pada contoh berikut ini.

```
tipedata namaVariabel;
```

Apabila kita ingin melakukan deklarasikan sebuah variabel bertipe data integer dengan nama variabel ‘jumlah’, dapat dituliskan sebagai berikut.

```
int jumlah;
```

Tipe Data

Bahasa pemrograman Java memiliki 2 (dua) jenis tipe data. Tipe data pertama disebut dengan tipe data primitive atau tipe data dasar. Tipe data tersebut berupa byte (8-bit), short (16-bit), int (32-bit), long (64-bit), float, double, boolean dan char. Penggunaan pada setiap tipe data primitif tersebut tidak jauh berbeda dengan bahasa pemrograman C seperti pada contoh berikut ini.

```
// TODO code application logic here
byte ini_byte_plus = 127;
byte ini_byte_minus = -128;

System.out.println(ini_byte_plus);
System.out.println(ini_byte_minus);
```

Byte

```
// TODO code application logic here  
short iniShortPlus = 32767;  
short iniShortMinus = -32768;  
  
System.out.println(iniShortPlus);  
System.out.println(iniShortMinus);
```

Short

```
// TODO code application logic here  
long iniLongPlus = 10000L;  
long iniLongMinus = -10000L;  
  
System.out.println(iniLongPlus);  
System.out.println(iniLongMinus);
```

Long

```
// TODO code application logic here  
int ini_integer_plus = 100;  
int ini_integer_minus = -100;  
  
System.out.println(ini_integer_plus);  
System.out.println(ini_integer_minus);
```

Integer

```
// TODO code application logic here  
float iniFloatPlus = 199.9f;  
float iniFloatMinus = -199.9f;  
  
System.out.println(iniFloatPlus);  
System.out.println(iniFloatMinus);
```

Float

```
// TODO code application logic here  
double iniDoublePlus = 199.98D;  
double iniDoubleMinus = -199.98D;  
  
System.out.println(iniDoublePlus);  
System.out.println(iniDoubleMinus);
```

Double

```
// TODO code application logic here  
boolean iniBooleanYa = true;  
boolean iniBooleanTidak = false;  
  
System.out.println(iniBooleanYa);  
System.out.println(iniBooleanTidak);
```

Boolean

```
// TODO code application logic here  
char iniKarakterA = 'A';  
  
System.out.println(iniKarakterA);
```

Char

Tipe data berikutnya adalah tipe data objek atau tipe data referensi. Tipe data tersebut merujuk pada objek yang dibuat menggunakan kelas atau antarmuka. Tipe data referensi contohnya berupa String, Array, Double, Integer, dll, yang sudah dibuatkan oleh Java dan tinggal digunakan. Penggunaan String dan Array pada bahasa pemrograman Java tidak jauh berbeda dengan penggunaan pada bahasa pemrograman C. Tipe data referensi juga dapat berasal dari Class yang kita buat sendiri. Penjelasan lebih lanjut mengenai tipe data ini akan dipelajari lebih lanjut pada pertemuan berikutnya. Contoh penggunaan dari tipe-tipe data referensi dapat dilihat dari contoh berikut ini.

```
// TODO code application logic here
String iniString = "John Doe";

System.out.println(iniString);
```

String

```
// TODO code application logic here
int[] Number = {1, 2, 3, 4, 5};

System.out.println(Number[0]);
```

Array

```
// TODO code application logic here
Person person = new Person("John Doe", 225);

System.out.println(person.name);
System.out.println(person.number);
```

Kelas

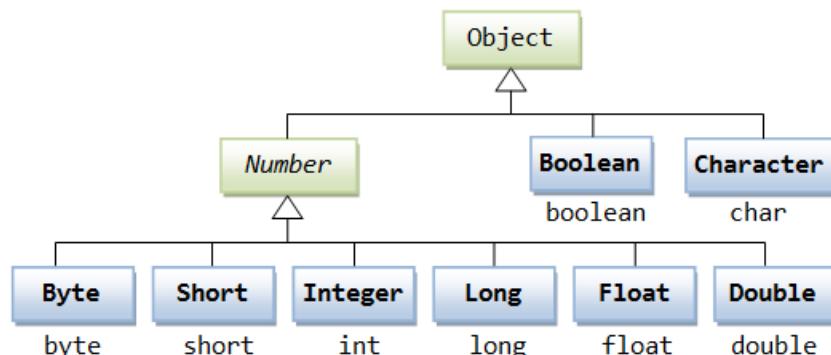
Pada kondisi tertentu, tipe data primitif dapat diubah menjadi tipe data objek atau referensi tanpa memerlukan adanya pembuatan kelas terlebih dahulu. Contohnya adalah sebagai berikut.

```
// TODO code application logic here
Integer iniObyekInteger = 10;
Double iniObyekDouble = 10.99;
Boolean iniObyekBoolean = true;

System.out.println(iniObyekInteger);
System.out.println(iniObyekDouble);
System.out.println(iniObyekBoolean);
```

Primitive to Reference

Hierarki dan pengelompokan objek pada bahasa pemrograman Java dapat direpresentasikan melalui gambar berikut ini.



Operator

Operator yang terdapat di dalam bahasa pemrograman Java memiliki kesamaan cara penggunaan dengan operator yang terdapat di dalam bahasa pemrograman C. Contoh dari operator dan penggunaannya adalah sebagai berikut.

```
// TODO code application logic here
int b = 10;

if (b == 10) {
    System.out.println("Ini Sepuluh");
} else {
    System.out.println("Ini Bukan Sepuluh");
}
```

if-else

```
// TODO code application logic here
int b = 10;

if (b < 10) {
    System.out.println("Ini Kurang dari Nol");
} else if (b > 0) {
    System.out.println("Ini Lebih dari Nol");
} else {
    System.out.println("Saya Tidak Tahu");
}
```

else-if

```
// TODO code application logic here
int menu = 0;

switch(menu) {
    case 1:
        System.out.println("Menu 1");
        break;
    default :
        System.out.println("Menu Default");
        break;
}
```

switch

```
// TODO code application logic here
for (int b = 0; b < 3; b++) {
    System.out.println(b);
}
```

for

```
// TODO code application logic here
int b = 0;

while (b == 2) {
    System.out.println(b);
    b++;
}
```

while

```
// TODO code application logic here
int b = 0;

do {
    System.out.println(b);
    b++;
} while (b == 2);
```

do-while

Method

Method atau metode merupakan sebuah blok yang berisikan beberapa kode untuk menjalankan tugas-tugas tertentu. Penggunaan metode di dalam bahasa pemrograman Java memiliki kemiripan dengan penggunaan prosedur dan fungsi di dalam bahasa pemrograman C.

Syntax pembuatan method atau metode di dalam bahasa pemrograman Java adalah sebagai berikut.

```
visibilitas (static) tipebalikan namaMethod (parameter1, parameter2, ...)
```

Method atau metode dapat dibagi menjadi tiga bagian utama. bagian pertama yaitu visibilitas, bagian kedua yaitu tipe data balikan, dan bagian terakhir yaitu nama metode beserta dengan parameternya. Terdapat bagian tambahan berupa static yang terletak di belakang visibilitas. Fungsi dari static yaitu memungkinkan pemanggilan metode tanpa membuat objek terlebih dahulu dari kelas tersebut.

Visibilitas merupakan cara sistem melihat bahwa metode tersebut dapat digunakan secara menyeluruh atau eksklusif hanya pada kelas-kelas tertentu. Visibilitas beserta jangkauannya terbagi sebagai berikut.

a. ***Public***

Visibilitas *public* memungkinkan semua kelas lain untuk menggunakan metode tersebut. Direpresentasikan dengan tanda tambah atau plus (+).

b. ***Protected***

Visibilitas *protected* hanya memungkinkan kelas turunan dari kelas tersebut untuk mengakses metode tersebut. Kelas yang bukan merupakan turunan dari kelas tersebut tidak dapat mengakses metode tersebut. Direpresentasikan dengan tanda pagar atau hashtag (#).

c. ***Private***

Visibilitas *private* hanya memungkinkan kelas yang memiliki metode tersebut yang dapat menggunakannya. Kelas turunan maupun kelas lainnya tidak dapat mengakses metode tersebut. Direpresentasikan dengan tanda kurang atau minus (-).

Bagian kedua atau tipe data balikan merupakan cara metode tersebut mengembalikan nilai yang sudah di proses. Tipe data balikan dapat berupa tipe data primitif atau tipe data dasar ataupun tipe data void. Contoh penggunaannya dapat dilihat pada tampilan berikut ini.

```
public class GD1_Y_XXXXX {

    public static int hitung (int a, int b) {
        return a + b;
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        System.out.println("2 + 3 = " + hitung(2, 3));

        Person person = new Person("Budi", 10);

        System.out.println("Nama : " + person.getName());
        System.out.println("Umur : " + person.getNumber();
    }
}
```

Komentar

Komentar atau comment merupakan salah satu cara yang cukup efektif untuk menambahkan catatan ataupun penjelasan mengenai metode-metode tertentu ataupun dapat digunakan untuk mencegah compiler membaca kode prematur yang dapat menyebabkan terjadinya error atau kesalahan di dalam program. Penulisan komentar dapat dilakukan dengan menambahkan *double slash* (//) untuk komentar sepanjang satu baris. Selain itu, penulisan komentar juga dapat dilakukan dengan *slash* dan *asterisk* /* & */ untuk komentar yang menggunakan lebih dari satu baris. Contoh penggunaannya dapat dilihat pada tampilan berikut ini.

```
public static void main(String[] args) {
    // TODO code application logic here
    System.out.println("2 + 3 = " + hitung(2, 3));

    Person person = new Person("Budi", 10);

    System.out.println("Nama : " + person.getName());
    System.out.println("Umur : " + person.getNumber();
}

//Bagian ini kosong
/*
Kalau bisa ditambahkan kode
sesuai keinginan kalian|
*/
```

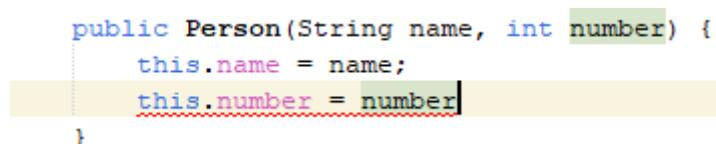
Kesalahan

Kesalahan atau *error* merupakan hal yang cukup lumrah dialami oleh programmer baik yang sudah mahir maupun yang masih pemula. Kesalahan-kesalahan dalam kode dapat membuat aplikasi yang kita kembangkan menjadi kurang sempurna dan seringkali menjadi tidak sesuai dengan harapan baik programmer dan klien.

NetBeans merupakan salah satu IDE berbahasa pemrograman Java yang mampu melakukan pendekripsi dini letak kesalahan kode program yang sudah dibuat. Pendekripsi kesalahan atau *error* dapat muncul dalam berbagai bentuk indikator yang dapat dilihat sebagai berikut.

1. *Red Underline*

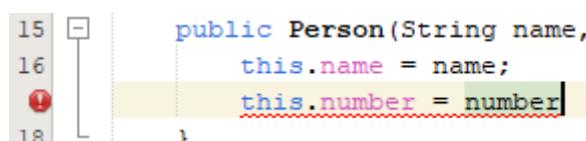
Indikator ini terletak tepat di bawah area kode yang salah atau error. Apabila kursor berada diatas daerah yang memiliki indikator ini, balon informasi akan muncul yang bertuliskan kesalahan yang dialami dalam daerah tersebut. *Shortcut key* Alt + ENTER akan memunculkan solusi-solusi umum yang dapat digunakan untuk menyelesaikan masalah pada daerah tersebut.



```
public Person(String name, int number) {  
    this.name = name;  
    this.number = number;  
}
```

2. *Alert Icon*

Indikator ini juga memiliki cara kerja yang sama ketika kursor berada di atas ikon merah bertanda seru yang terletak di barisan nomor baris. Indikator ini memudahkan para programmer untuk saling mengkomunikasikan letak kesalahan atau error di dalam program.



```
15  
16  
17  public Person(String name,  
18             this.name = name;  
                 this.number = number;  
             }
```

3. *Stacktrace*

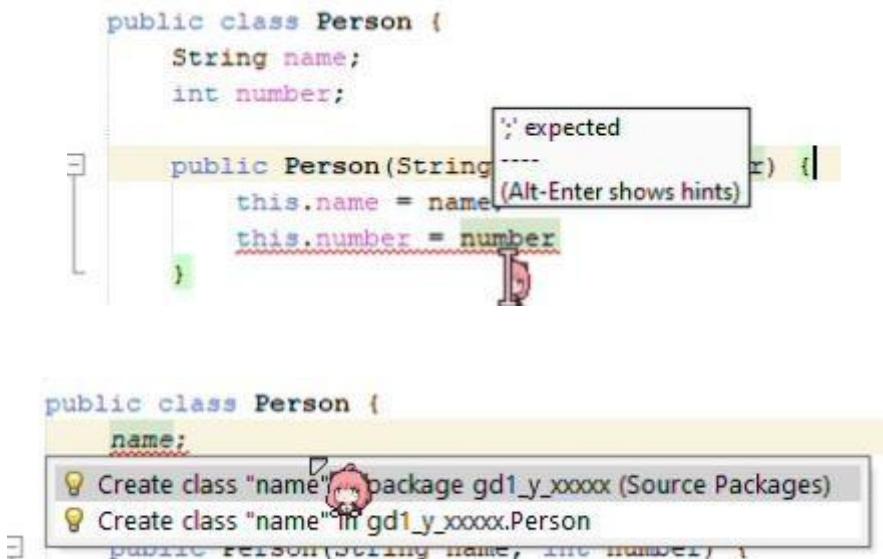
Indikator ini merupakan indikator yang hanya muncul setelah program selesai melakukan *compile*. Indikator ini memuat sebuah teks penjelasan terhadap kesalahan yang terjadi di dalam program yang ditampilkan pada terminal. Indikator ini seringkali dimuat dalam 2 (dua) warna, yaitu warna merah dan biru. Teks yang berwarna biru akan menjelaskan secara spesifik letak kesalahan yang terjadi di dalam program. Selain itu, teks tersebut dapat ditekan untuk langsung menuju ke lokasi baris kesalahan.

The screenshot shows the NetBeans IDE interface. In the top right corner, there is an 'Output' window titled 'Output - GD1_Y_XXXXX (run)'. The window displays the following error message:

```
run:
Exception in thread "main" java.lang.RuntimeException: Uncompilable code - ';' expected
    at gd1_y_xxxxxx.Person.<init>()
    at gd1_y_xxxxxx.GD1_Y_XXXXX.main(GD1_Y_XXXXX.java:44)
C:\Users\Joesenchia Santos\AppData\Local\NetBeans\Cache\20\executor-snippets\run.xml:111: The following error occurred while executing this line
C:\Users\Joesenchia Santos\AppData\Local\NetBeans\Cache\20\executor-snippets\run.xml:68: Java returned: 1
BUILD FAILED (total time: 2 seconds)
```

Below the output window, a tooltip is displayed over some code. The tooltip contains the text "' expected' and '(Alt-Enter shows hints)'.

Kesalahan-kesalahan tersebut dapat diperbaiki dengan melakukan *hover* di atas bagian yang memiliki indikator *red underline* ataupun *alert icon*. Sebuah balon informasi mengenai letak kesalahan dan solusi yang dapat menuntaskan permasalahan akan muncul. Balon informasi tersebut dapat memuat informasi yang variatif sesuai dengan tingkat kerumitan kesalahan kode. Mulai dari hanya sekedar menampilkan informasi berupa kata-kata ataupun menampilkan solusi otomatis yang akan tertulis di dalam kode. Contohnya dapat dilihat pada gambar di bawah ini.



Informasi Tambahan

Penulisan nama kelas dapat dituliskan seperti contoh berikut ini.

```
class NamaKelas
```

Penulisan nama metode dapat dituliskan seperti contoh berikut ini.

```
public int jumlahGajiPokok
```

Penulisan nama variabel dapat dituliskan seperti contoh berikut ini.

```
double totalLabaBersih
```

Penulisan tipe data obek atau referensi dituliskan dengan diawali oleh huruf kapital dan tidak disingkat.

```
// TODO code application logic here
Integer iniObyekInteger = 10;
Boolean iniObyekBoolean = true;|
```

Setiap baris kode diakhiri dengan semicolon atau titik koma (;).

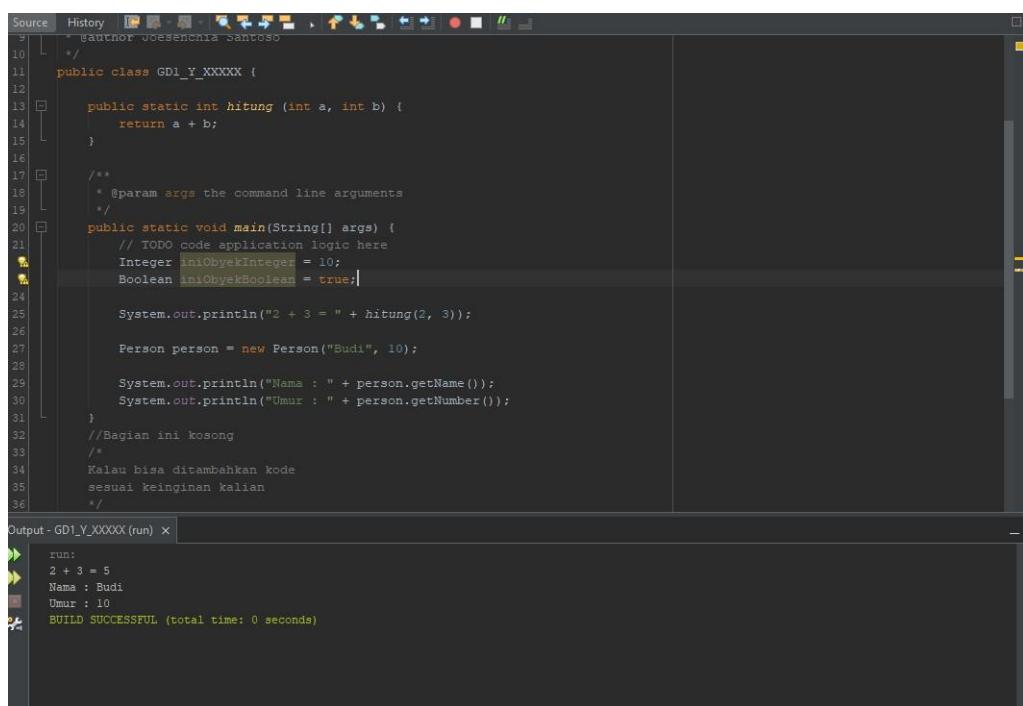
```
class Person {
    String name;
    int number;

    public Person(String name, int number) {
        this.name = name;
        this.number = number;
    }

    public String getName() {
        return name;
    }

    public int getNumber() {
        return number;
    }
}
```

Pencetakan hasil kode ke dalam terminal dapat dilakukan dengan `System.out.println` atau dalam bahasa pemrograman C seringkali dilakukan dengan `printf`.



The screenshot shows an IDE interface with a code editor and a terminal window. The code editor contains a Java file named 'GD1_Y_XXXXX.java' with the following content:

```
/*
 * Author: Joesenchia Santoso
 */
public class GD1_Y_XXXXX {

    public static int hitung (int a, int b) {
        return a + b;
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        Integer iniObyekInteger = 10;
        Boolean iniObyekBoolean = true;

        System.out.println("2 + 3 = " + hitung(2, 3));

        Person person = new Person("Budi", 10);

        System.out.println("Nama : " + person.getName());
        System.out.println("Umur : " + person.getNumber());
    }
    //Bagian ini kosong
    /*
    Kalau bisa ditambahkan kode
    sesuai keinginan kalian
    */
}
```

The terminal window below shows the output of the code execution:

```
Output - GD1_Y_XXXXX (run) ×
▶ run:
2 + 3 = 5
Nama : Budi
Umur : 10
BUILD SUCCESSFUL (total time: 0 seconds)
```

Daftar Pustaka

<https://codingstudio.id/blog/tipe-data-primitif#:~:text=Java%20sudah%20mendefinisikan%20tipe%20dari,float%2C%20char%2C%20dan%20double.>

<https://jayjay.co/variabel-dan-tipe-data-pada-java#:~:text=Variabel%20pada%20Java%20digunakan%20untuk,disimpan%20di%20dalam%20variabel%20tersebut.>

<https://www.petanikode.com/java-variabel-dan-tipe-data/>

<https://traspac.co.id/mengenal-tipe-data-pada-java/>

<https://medium.com/@furatamarizuki/mengenal-tipe-data-dalam-bahasa-pemrograman-java-panduan-lengkap-untuk-pemula-b595ca25a5fc>

<https://docs.oracle.com/javase/8/docs/api/overview-summary.html>