

Lab 1

Tutorial IDE

Dasar-Dasar Pemrograman 2

CSGE601021

Semester Genap 2016/2017

Batas waktu pengumpulan:

Senin, 13 Februari 2017 pukul 18.00 Waktu Scele (Lab Senin)

Rabu, 15 Februari 2017 pukul 18.00 Waktu Scele (Lab Rabu)

Tujuan dari Lab ini adalah melatih Anda agar menguasai bahan kuliah yang diajarkan di kelas. Mahasiswa diperbolehkan untuk berdiskusi, tetapi Anda tetap harus **menuliskan sendiri** solusi/kode program dari soal yang diberikan tanpa bantuan orang lain. Belajarlah menjadi mahasiswa yang mematuhi integritas akademik. **Sikap Jujur merupakan sebuah sikap yang dimiliki mahasiswa Fasilkom UI.**

Peringatan: Jangan mengumpulkan pekerjaan beberapa menit menjelang batas waktu pengumpulan karena ada kemungkinan pengumpulan gagal dilakukan atau koneksi internet terputus!

Lab 1

Tutorial IDE

Halo! Selamat datang pasukan Dasar-Dasar Pemrograman 2! Anda telah memasuki zona 'perang' pada mata kuliah ini. Untuk mempersiapkan peperangan, kalian harus mempersiapkan senjata yang akan kalian bawa dan kalian gunakan ke depannya. *Let us introduce you to our lovely weapons: Eclipse IDE and IntelliJ IDE!* Pada tutorial kali ini kami akan mengajarkan kalian bagaimana cara menggunakan kedua senjata tersebut.

Semangat mempelajari Tutorial IDE! Salam damai dan salam pemrograman.

IDE VS Text Editor

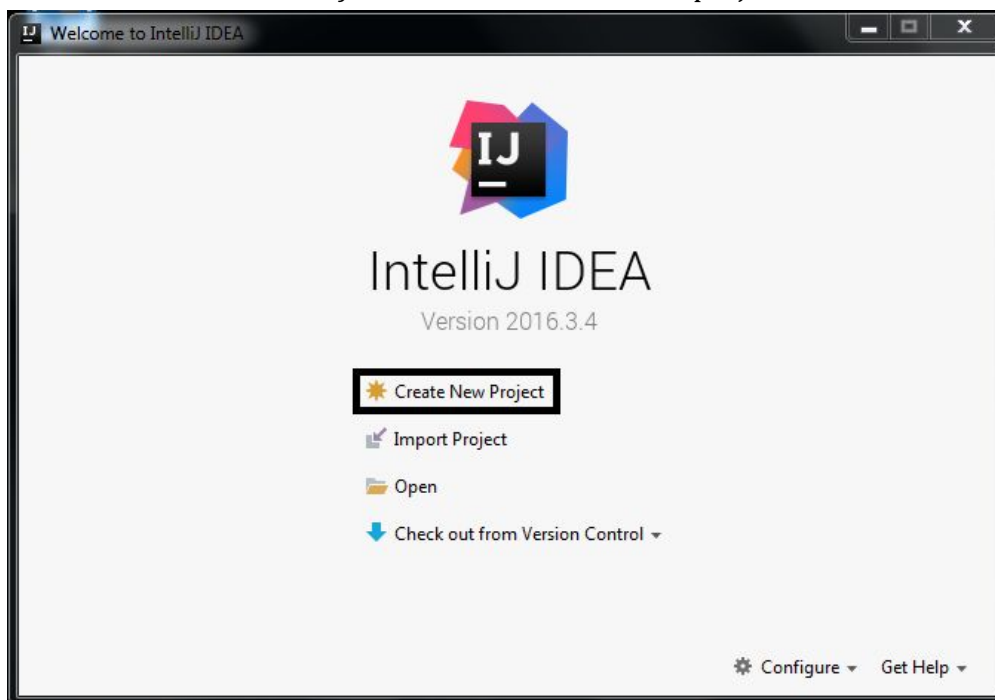
IDE (*Integrated Development Environment*) merupakan salah satu tools untuk software development dimana dengan IDE kita bisa langsung membuat kode menjadi *final product* (contoh: *compiled program, web app, dll.*). Ada beberapa hal yang membedakan IDE dengan *text editor*, yaitu:

- *Convenience Features*
Banyak IDE yang memasukkan *convenience features* (seperti, navigasi yang mudah, *code auto-completion, class explorers, hierarchi diagrams, dll.*) dan tools yang membantu kamu untuk *Automate development* (contoh: *source version control, testing tools, dll.*) Tetapi tidak semua IDE memiliki tools seperti yang disebutkan.
- *Support Most of Programming Language*
Setiap IDE *support* berbagai macam bahasa pemrograman seperti Java, C, C++, Python, dan lain-lain. Banyak bahasa pemrograman yang populer dapat didevelop di IDE. Kalian tidak perlu takut untuk tidak kebagian bahasa pemrograman. *You're favorite language will always be in IDE.*
- *Integrated Development*
Apa yang membuat IDE begitu berguna yakni karena ia *Integrated*. Di *integrated environment* kamu diberikan kebebasan untuk bisa melakukan apapun contohnya seperti *debugger*. Jadi secara tidak langsung ketika kamu menulis kode, IDE akan men-*debug* secara bersamaan. Tanpa IDE kamu harus menulis kodemu dalam *text editor* dan men-*debug*-nya dengan *compiler*.

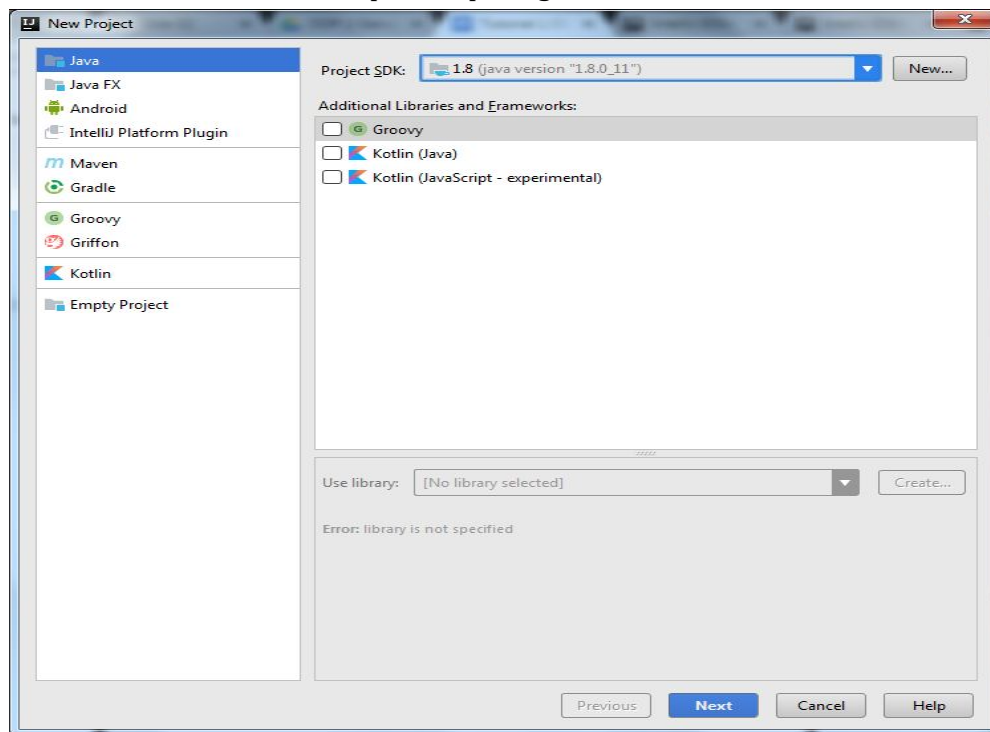
IntelliJ

IntelliJ adalah IDE yang dapat digunakan untuk mengembangkan program atau aplikasi komputer. IntelliJ dapat digunakan untuk melakukan pemrograman dengan bahasa Java, Groovy, atau Scala. Aplikasi ini dapat diunduh melalui [link https://www.jetbrains.com/idea/download/](https://www.jetbrains.com/idea/download/) secara gratis, namun disediakan juga IntelliJ versi berbayar untuk mendapatkan keuntungan lainnya. Kemudian, pilih *Operating System* yang sesuai dengan komputer/laptop Anda dan selamat menunggu hasil unduhan. Berikut adalah langkah-langkah dasar yang diperlukan untuk menggunakan IntelliJ.

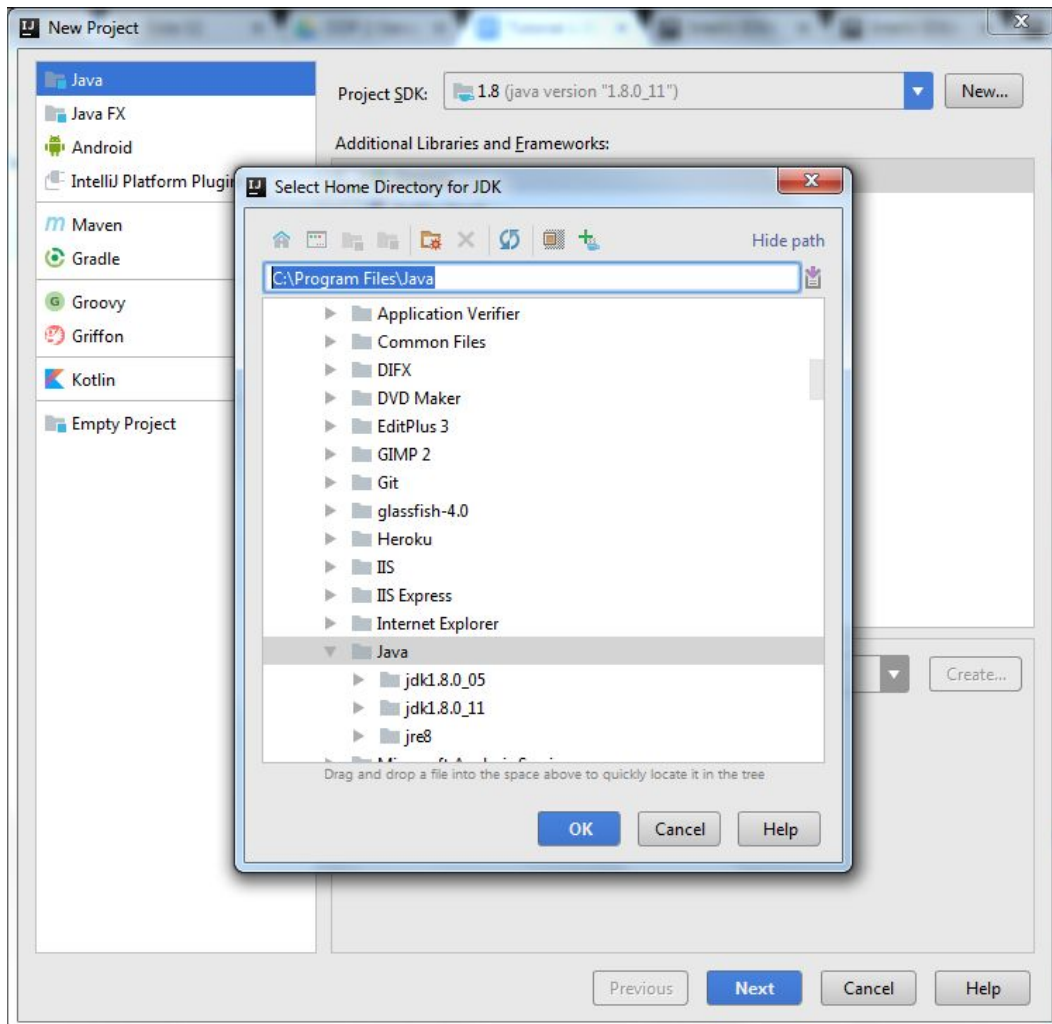
1. Jalankan IDE IntelliJ.
2. Pilih menu *Create New Project* untuk memulai sebuah project.



3. Kemudian, akan muncul tampilan seperti gambar di bawah.



- Pilih Project SDK yang sesuai dengan gambar langkah nomor 3. Jika tidak ada pilihan Project SDK yang sesuai, maka klik tombol New lalu pilih menu JDK. Kemudian, pilih directory C:\Program Files\Java dan pilih folder jdk dengan versi terbaru sesuai dengan komputer masing - masing.



- Klik tombol Next dua kali berturut - turut.
- Isi nama project pada lab kali ini dengan Lab1 lalu klik tombol Finish.
- Klik menu src pada sidebar Project.



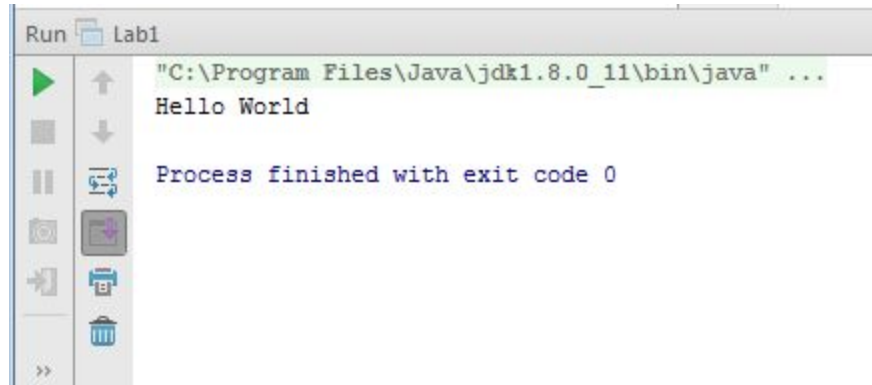
- Buat file Java dengan cara klik kanan pada folder src -> New -> Java Class atau klik tab File -> New -> Java Class.

9. Untuk nama class pada lab kali ini adalah Lab1. Kemudian, buat program anda sesuai dengan gambar di bawah. Program akan melakukan pencetakan "Hello World".

```
public class Lab1 {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

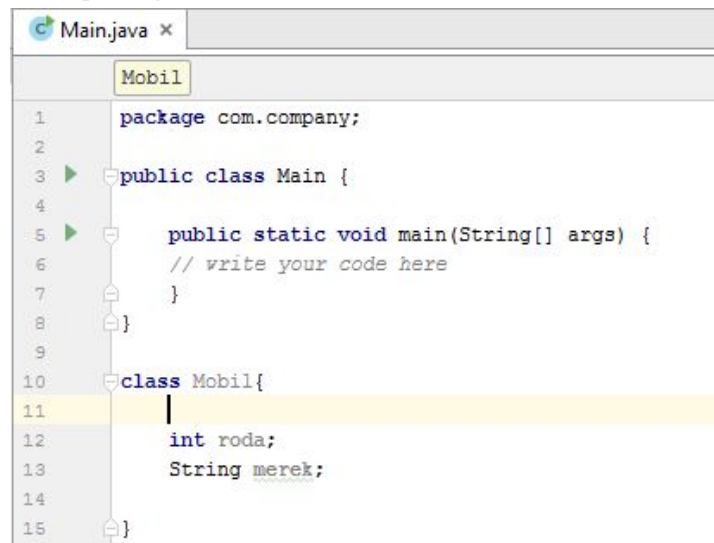
10. Pilih menu Run kemudian pilih file Lab1 untuk dijalankan.

11. Kemudian, akan muncul hasil dari program sesuai dengan gambar di bawah.



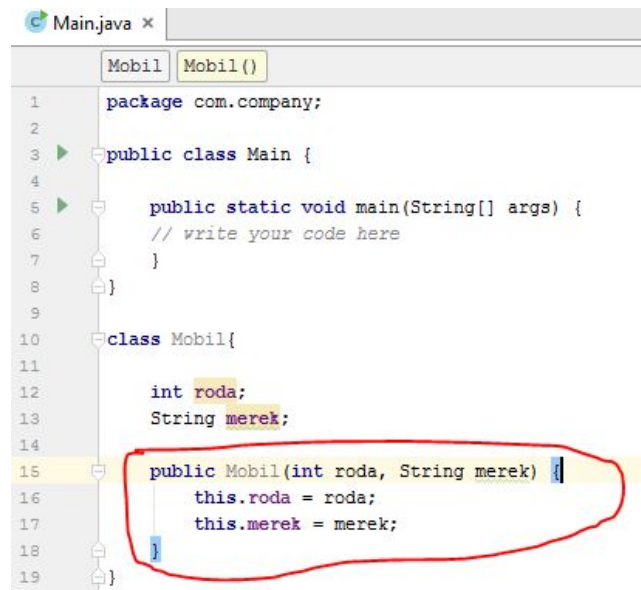
Berikut adalah langkah-langkah untuk melakukan *generate constructor*.

1. Buat class Mobil seperti gambar di bawah:



2. Tekan *alt+insert* dan pilih *constructor*. Blok roda dan merek lalu pilih OK.

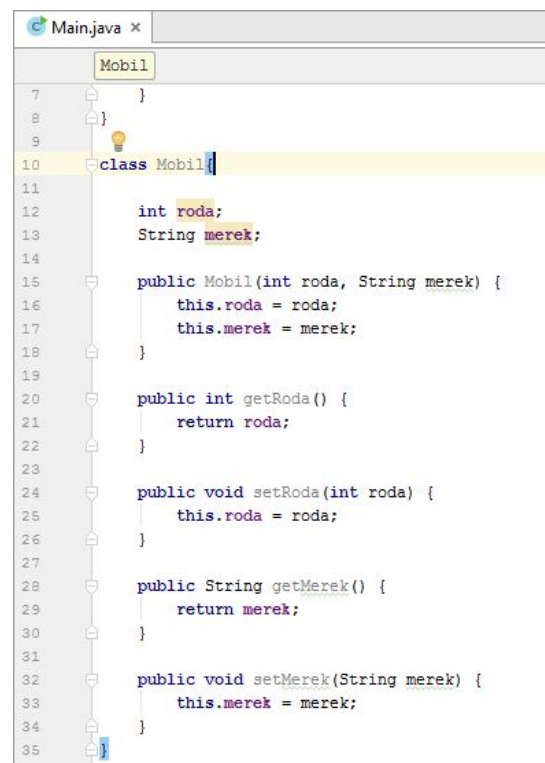
3. *Constructor* secara otomatis akan di-generate oleh IDE dan akan terlihat seperti gambar di bawah.



```
1 package com.company;
2
3 public class Main {
4
5     public static void main(String[] args) {
6         // write your code here
7     }
8 }
9
10 class Mobil{
11
12     int roda;
13     String merek;
14
15     public Mobil(int roda, String merek) {
16         this.roda = roda;
17         this.merek = merek;
18     }
19 }
```

Berikut adalah langkah-langkah untuk melakukan *generate setter getter*.

1. Lanjutkan dari kode sebelumnya, tekan *alt+insert* dan pilih “*setter and getter*”
2. Blok semua *variable* dan klik OK.
3. *Setter getter* akan secara otomatis di-generate oleh IDE dan akan terlihat seperti gambar di bawah:



```
7     }
8 }
9
10 class Mobil{
11
12     int roda;
13     String merek;
14
15     public Mobil(int roda, String merek) {
16         this.roda = roda;
17         this.merek = merek;
18     }
19
20     public int getRoda() {
21         return roda;
22     }
23
24     public void setRoda(int roda) {
25         this.roda = roda;
26     }
27
28     public String getMerek() {
29         return merek;
30     }
31
32     public void setMerek(String merek) {
33         this.merek = merek;
34     }
35 }
```

Berikut adalah langkah-langkah untuk melakukan *refactoring*. (jika belum tau apa itu *refactoring*, *refactoring* yakni proses merubah suatu kode tanpa mengubah fungsi dari kode itu sendiri)

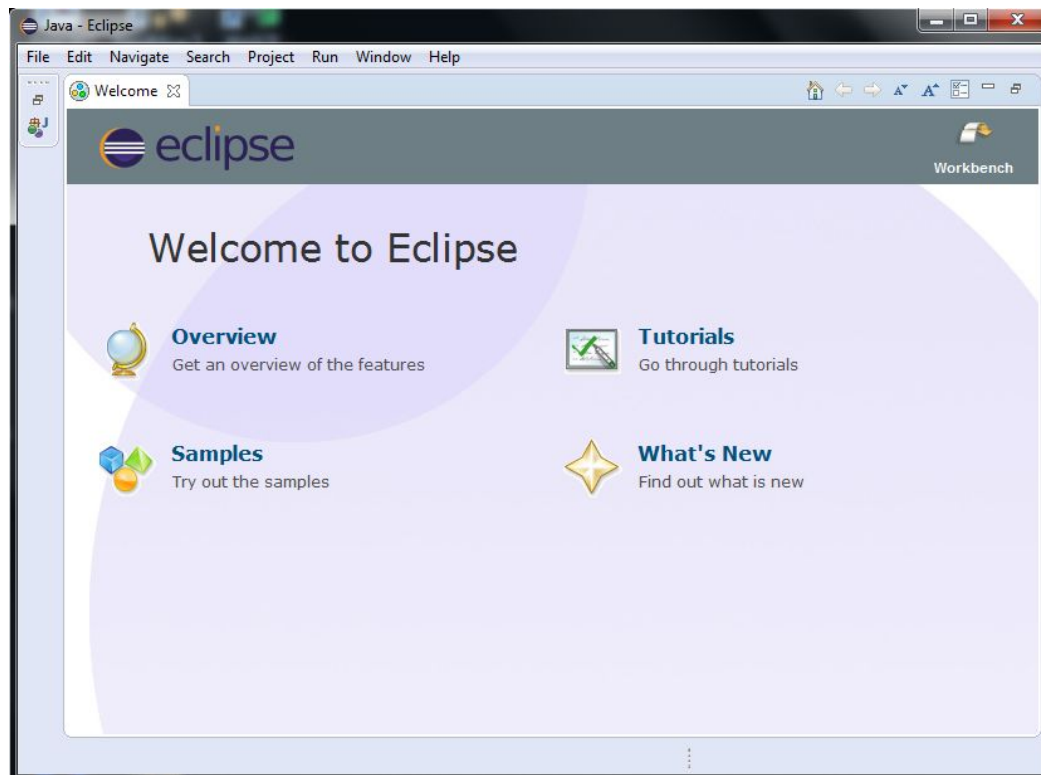
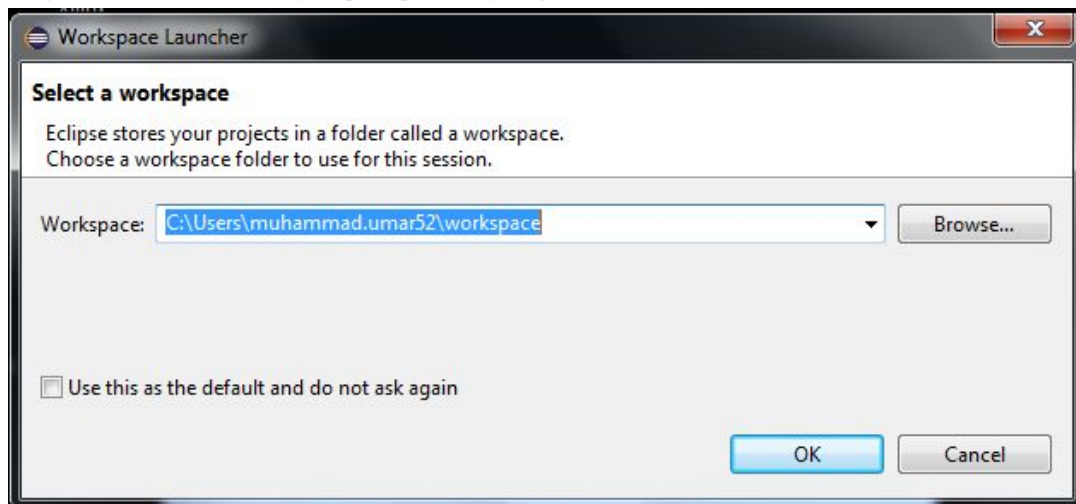
1. Blok nama *variable* roda.
2. Tekan Shift + F6
3. Ubah nama *variable* roda menjadi ban.

```
class Mobil {  
    int ban;  
    String merek;  
  
    public Mobil(int roda, String merek) {  
        this.ban = roda;  
        this.merek = merek;  
    }  
  
    public int getRoda() {  
        return ban;  
    }  
  
    public void setRoda(int roda) {  
        this.ban = roda;  
    }  
  
    public String getMerek() {  
        return merek;  
    }  
  
    public void setMerek(String merek) {  
        this.merek = merek;  
    }  
}
```

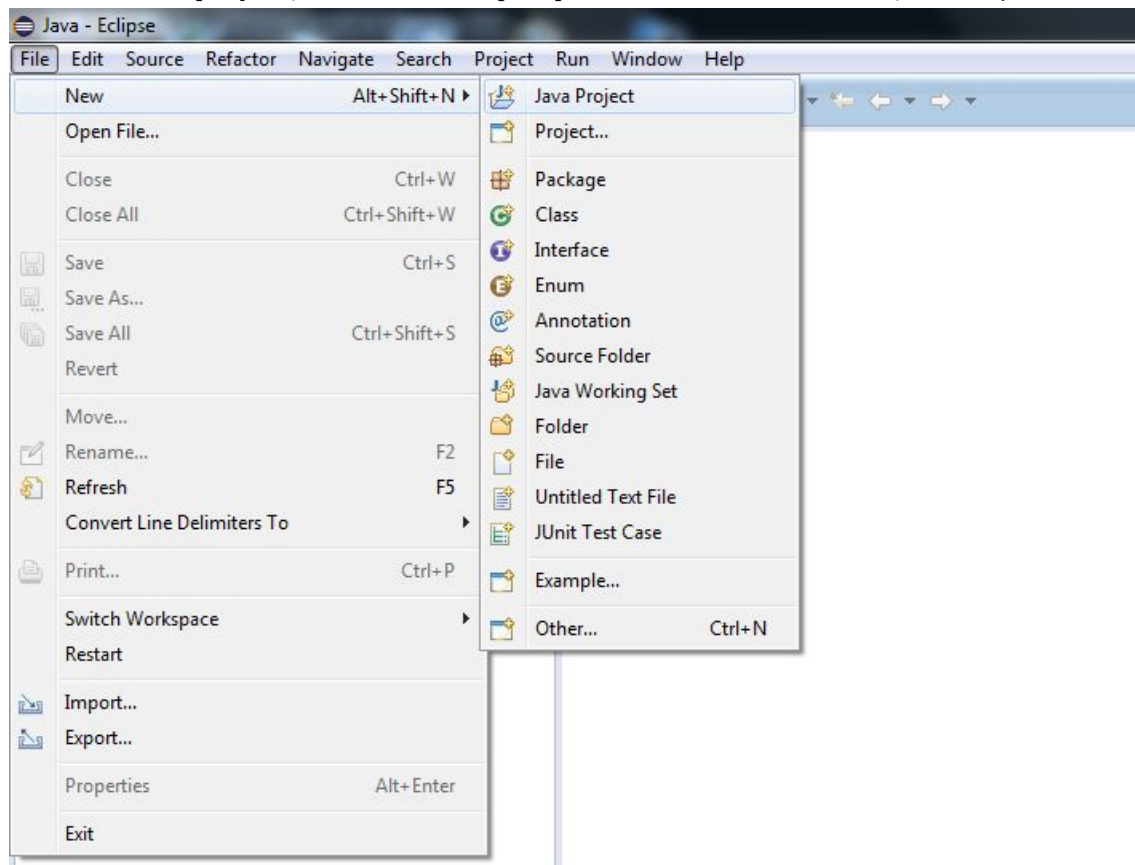

Eclipse

Eclipse IDE adalah IDE yang juga dapat digunakan untuk mengembangkan program atau aplikasi komputer. Eclipse dapat digunakan untuk melakukan pemrograman dengan bahasa Java, C, dan lainnya. Aplikasi ini dapat diunduh melalui [link https://eclipse.org/downloads/](https://eclipse.org/downloads/) secara gratis. Berikut adalah langkah-langkah dasar yang diperlukan untuk menggunakan IntelliJ.

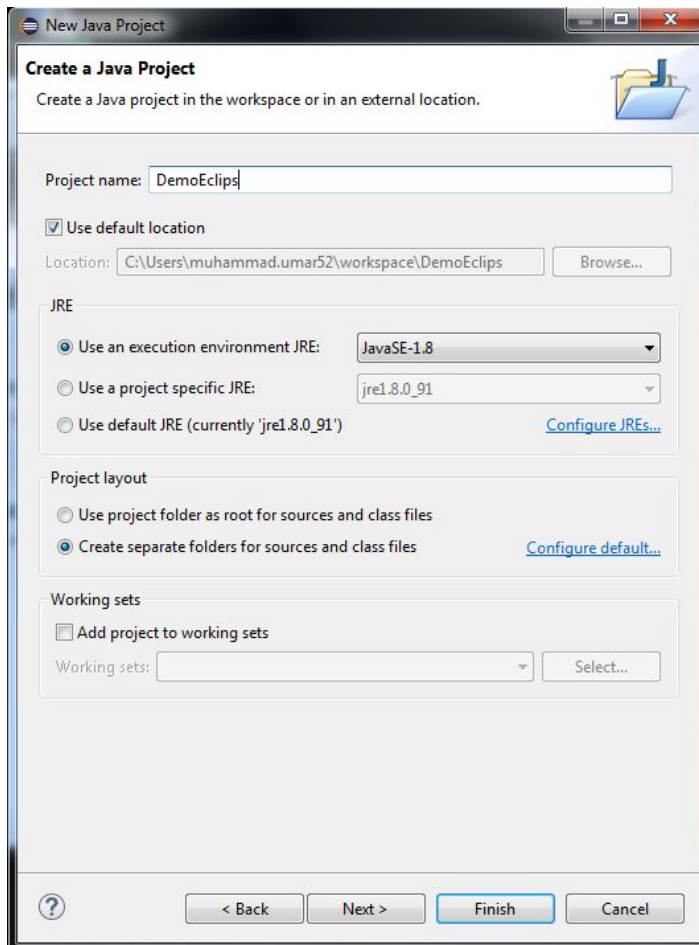
1. Klik icon eclipse yang ada di desktop, startmenu, atau program file, dan pilih direktori yang akan digunakan untuk menyimpan projek yang akan dibuat.



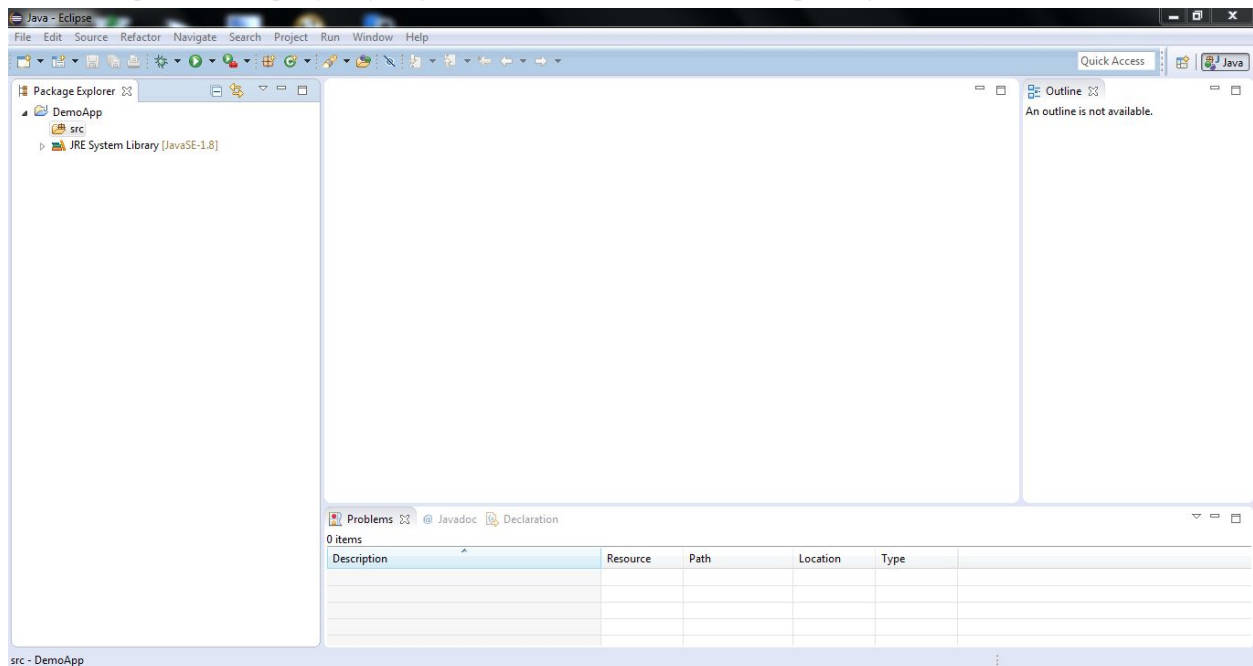
2. Untuk membuat proyek Java baru di Eclipse, pilih menu File => New => Java Project.



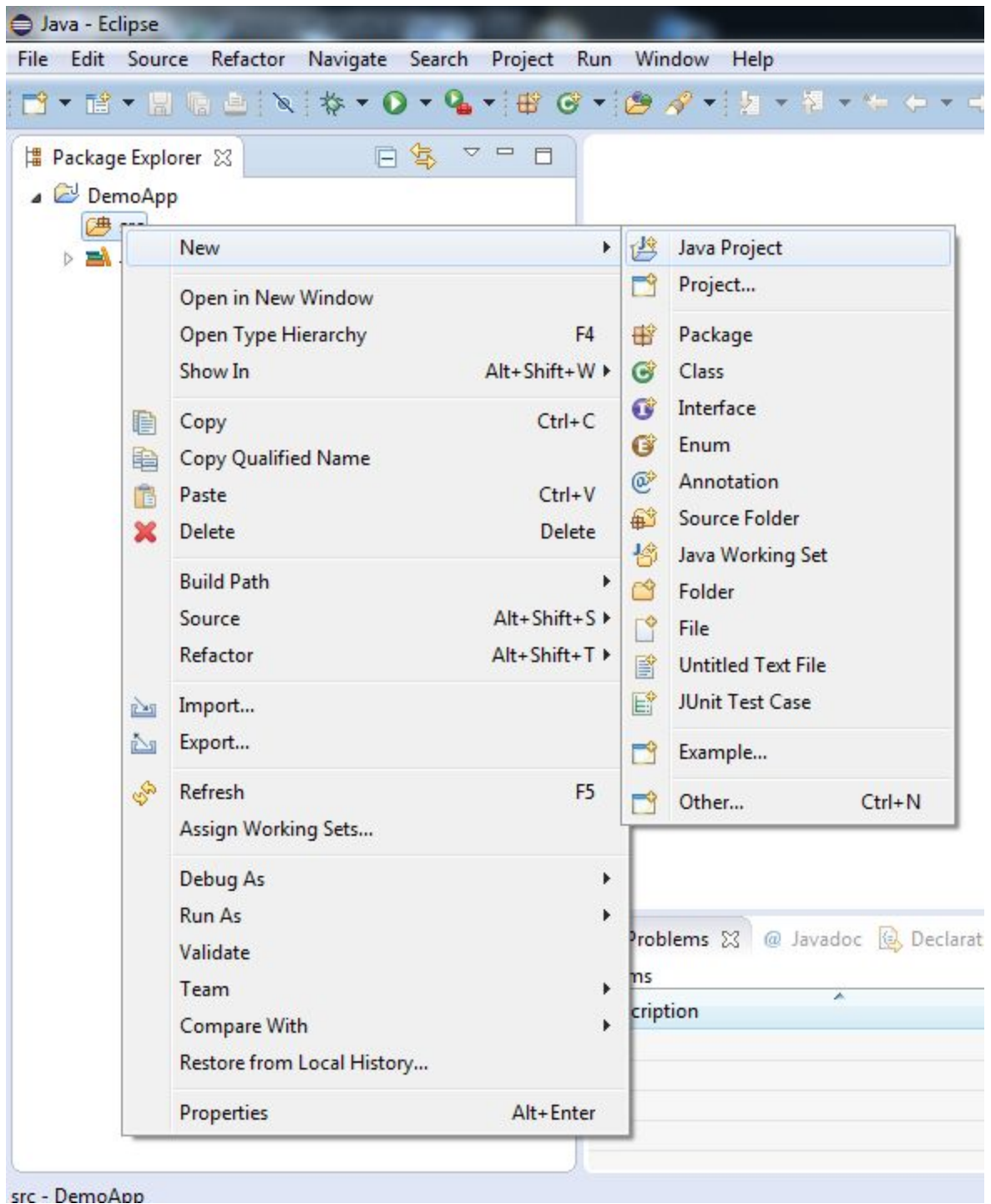
3. Isi nama proyek yang diinginkan, contohnya DemoApp.



4. Tampilan awal proyek yang telah dibuat akan terlihat seperti gambar di bawah.



5. Untuk membuat suatu class Java di eclipse, klik kanan pada NamaProyek/src, contohnya DemoApp/src.



6. Isi nama class dengan awalan huruf besar, contohnya Main.

New Java Class

Java Class

The use of the default package is discouraged.

Source folder:

Package:

☐ Enclosing type:

Name:

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass:

Interfaces:

Which method stubs would you like to create?

☐ public static void main(String[] args)

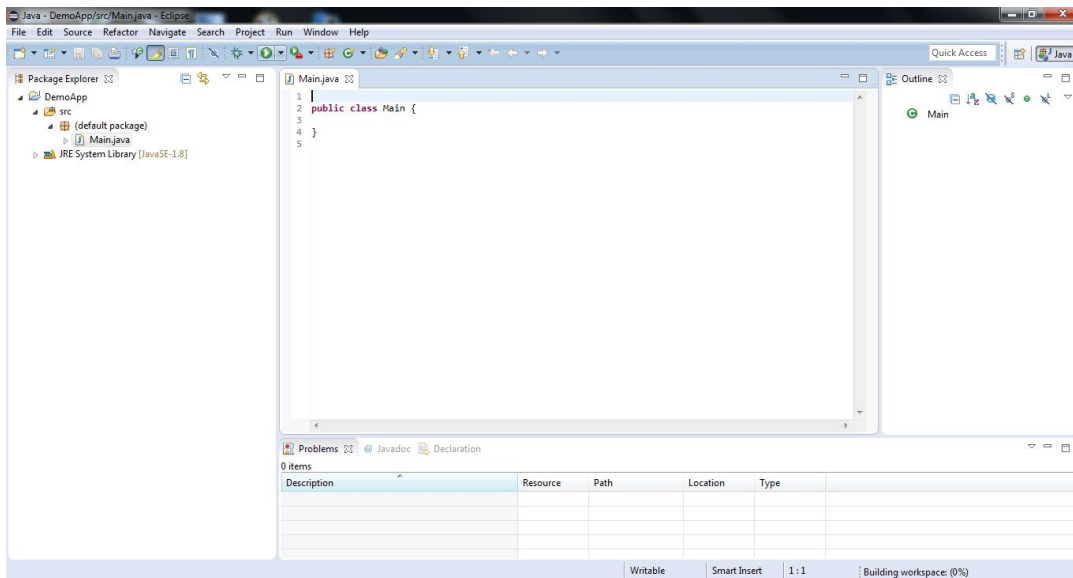
☐ Constructors from superclass

☒ Inherited abstract methods

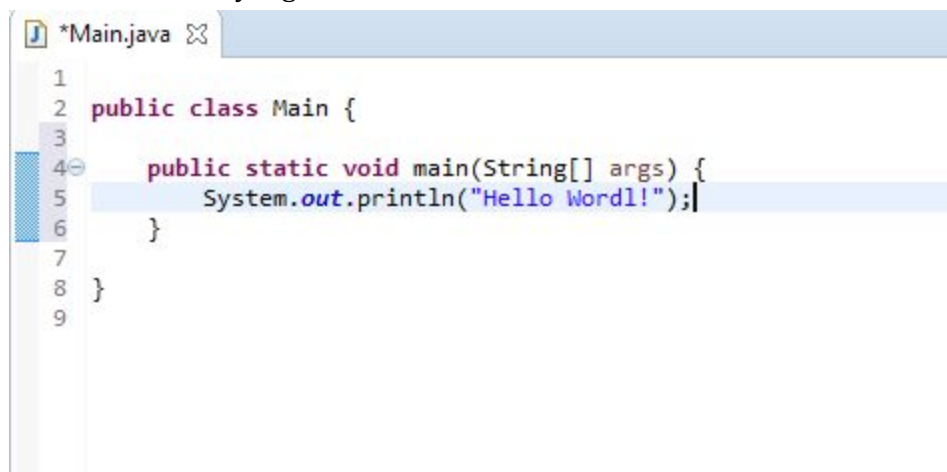
Do you want to add comments? (Configure templates and default value [here](#))

☐ Generate comments

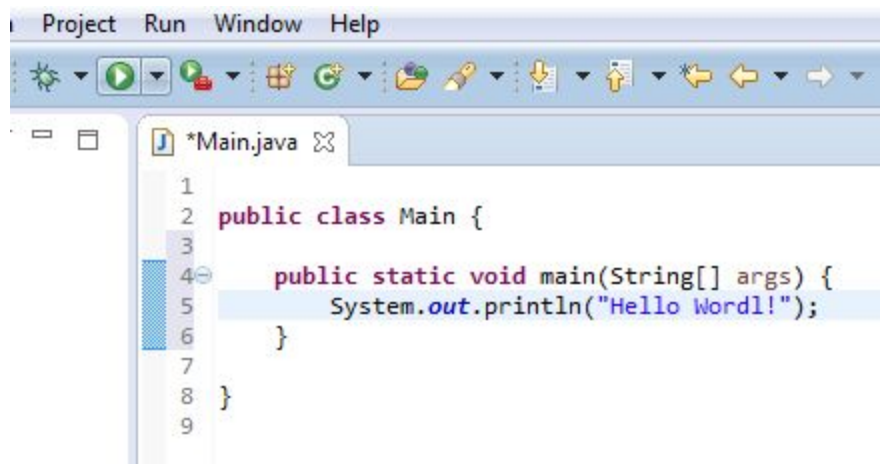
7. Tampilan awal kelas yang telah dibuat akan terlihat seperti gambar di bawah.



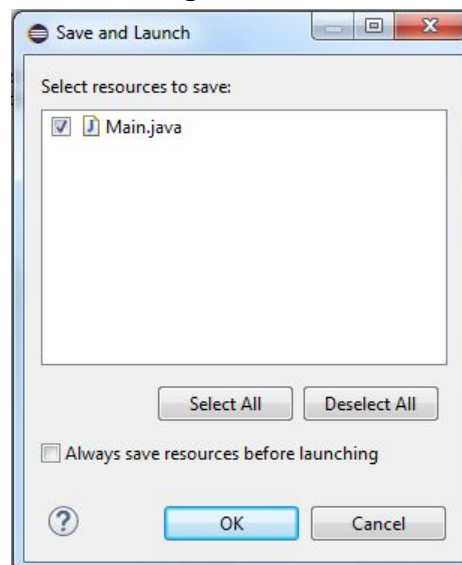
8. Membuat method Main yang berisi kode untuk mencetak “Hello Word!”.



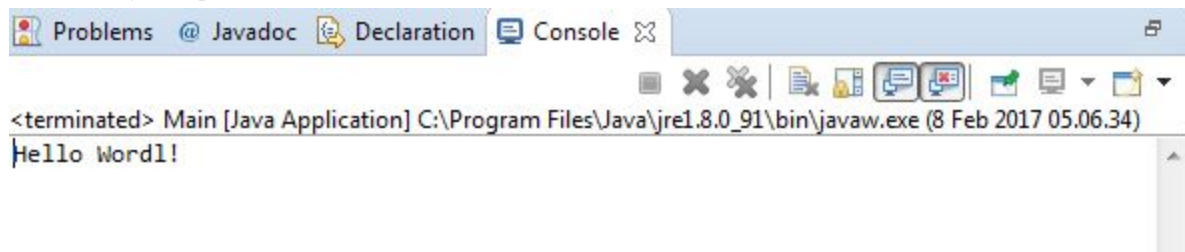
9. Untuk menjalankan program, klik tombol Run sesuai dengan gambar di bawah ini.



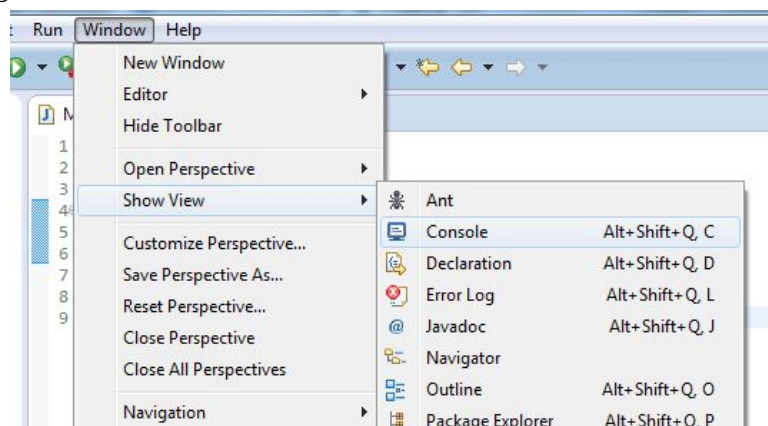
10. Pilih file yang ingin dijalankan, file dengan method main.



11. Hasilnya dapat dilihat di window Console.



12. Jika window Console tidak terbuka secara otomatis pada langkah 11, dapat dibuka secara manual dengan klik Window => Show View => Console

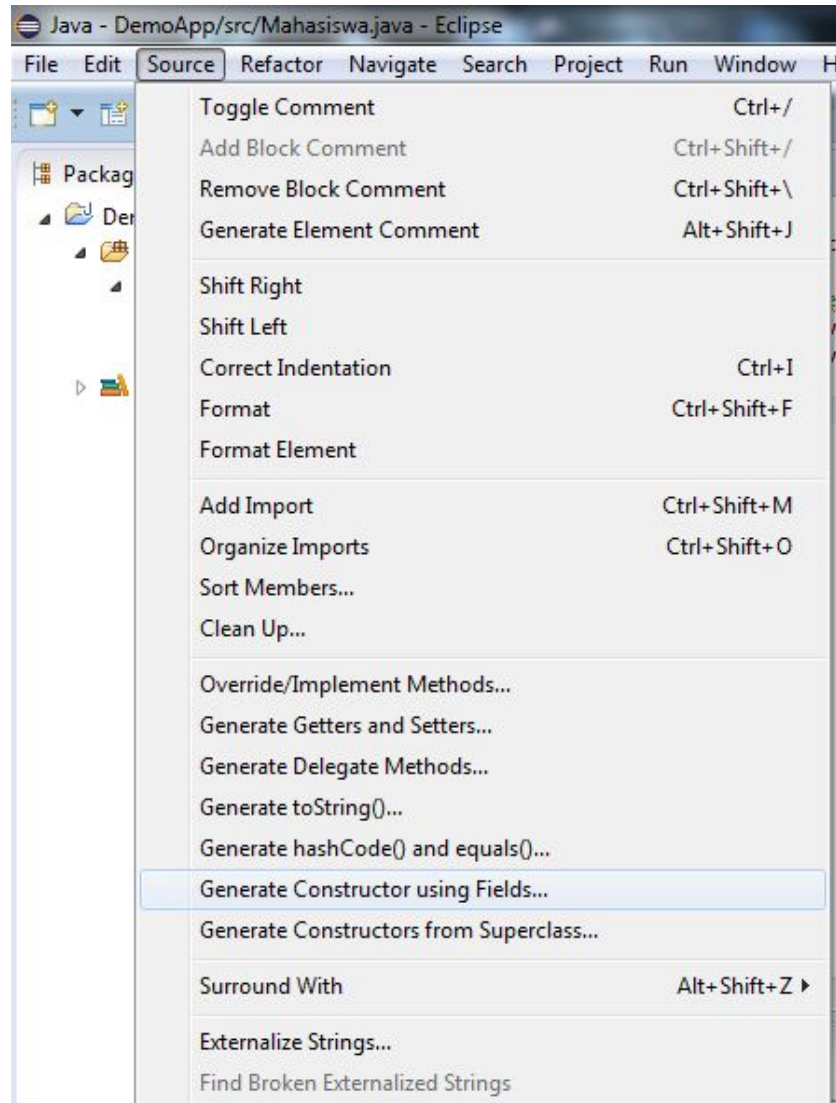


Berikut adalah langkah-langkah untuk melakukan *generate constructor*.

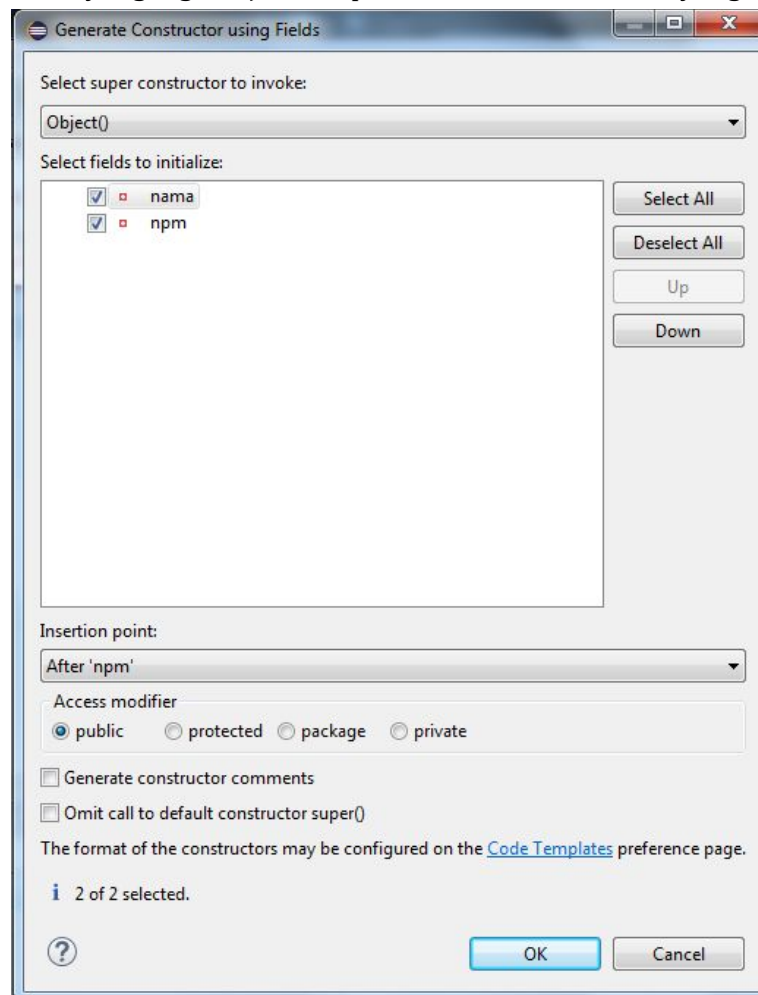
1. Buat class Mahasiswa dengan cara yang sama ketika membuat kelas Main sebelumnya. Tambahkan beberapa atribut/variabel di class Mahasiswa tersebut.

```
public class Mahasiswa {  
  
    //Membuat atribut2 dari mahasiswa  
    private String nama;  
    private String npm;  
  
}
```

2. Buat constructor dari class Mahasiswa dengan cara klik menu Source => Generate Constructor using Field.



3. Memilih variabel yang ingin dijadikan parameter di constructor yang ingin dibuat..

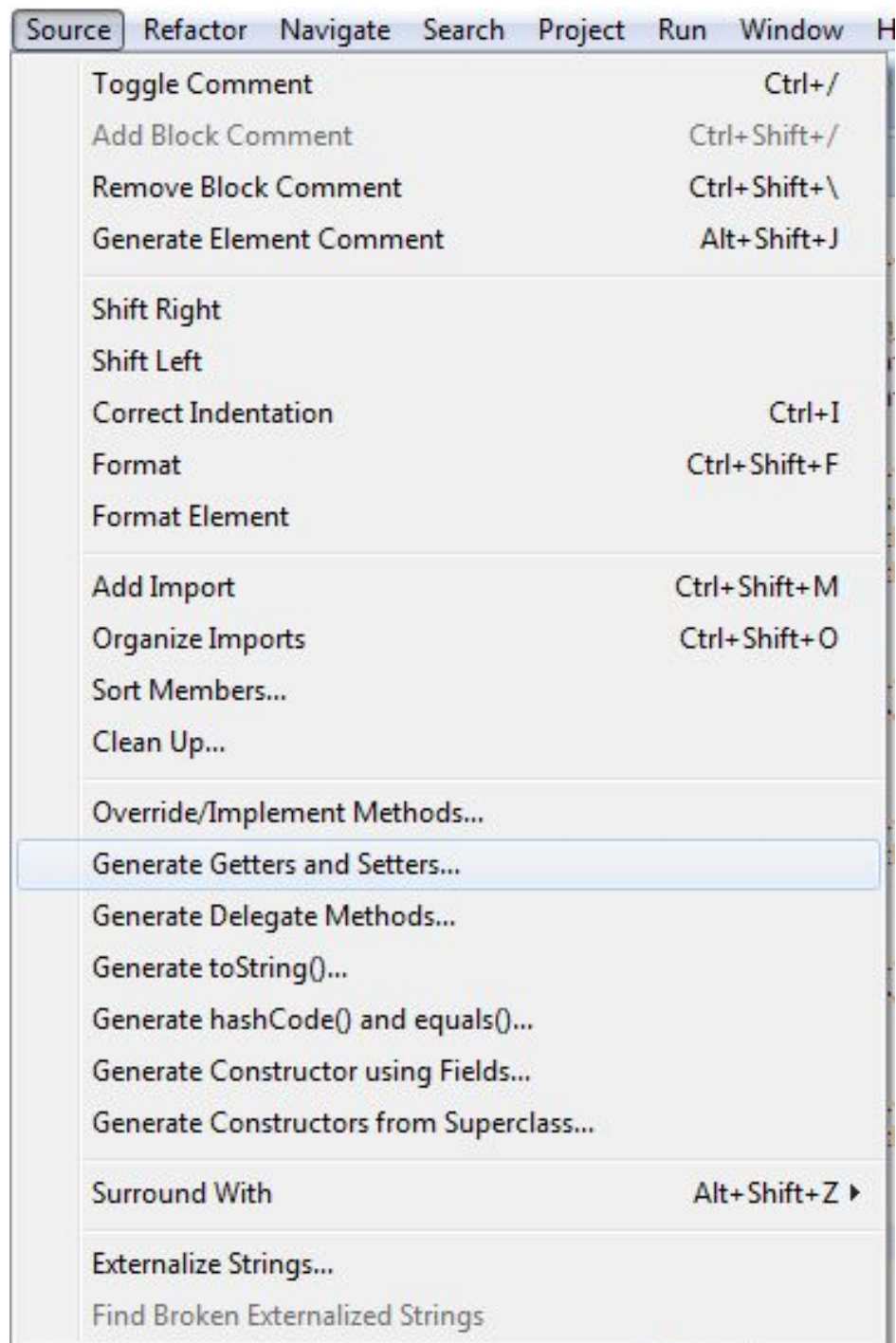


4. Hasil constructor yang telah dibuat seperti gambar di bawah.

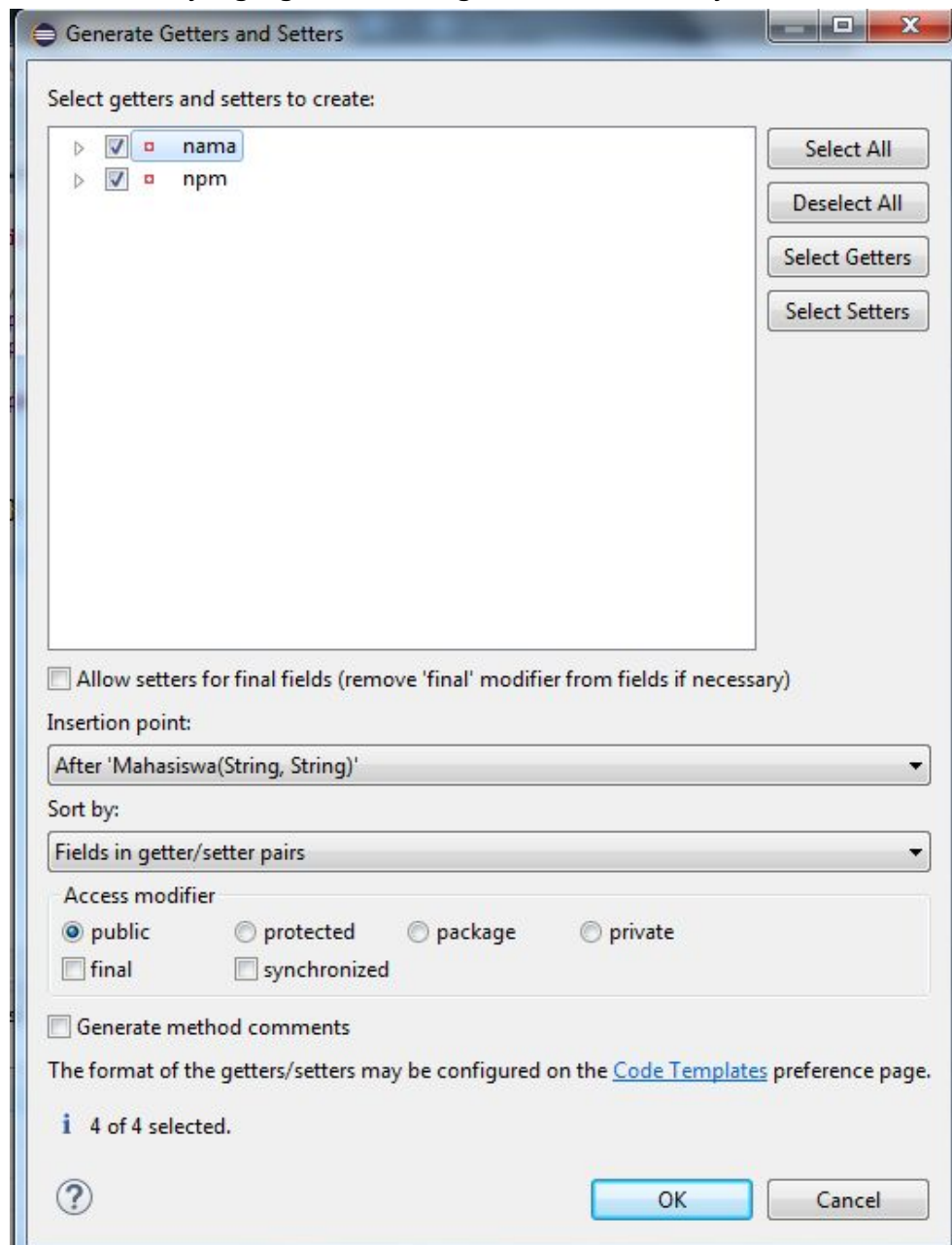
```
public class Mahasiswa {  
  
    //Membuat atribut2 dari mahasiswa  
    private String nama;  
    private String npm;  
  
    public Mahasiswa(String nama, String npm) {  
        super();  
        this.nama = nama;  
        this.npm = npm;  
    }  
  
}
```

Berikut adalah langkah-langkah untuk melakukan *generate setter getter*.

1. Dari class Mahasiswa yang sebelumnya dibuat, buat getters and setters dari class Mahasiswa dengan cara klik menu Source => Generate Getter and Setters.



- Memilih variabel yang ingin dibuatkan getters and setterstnya.

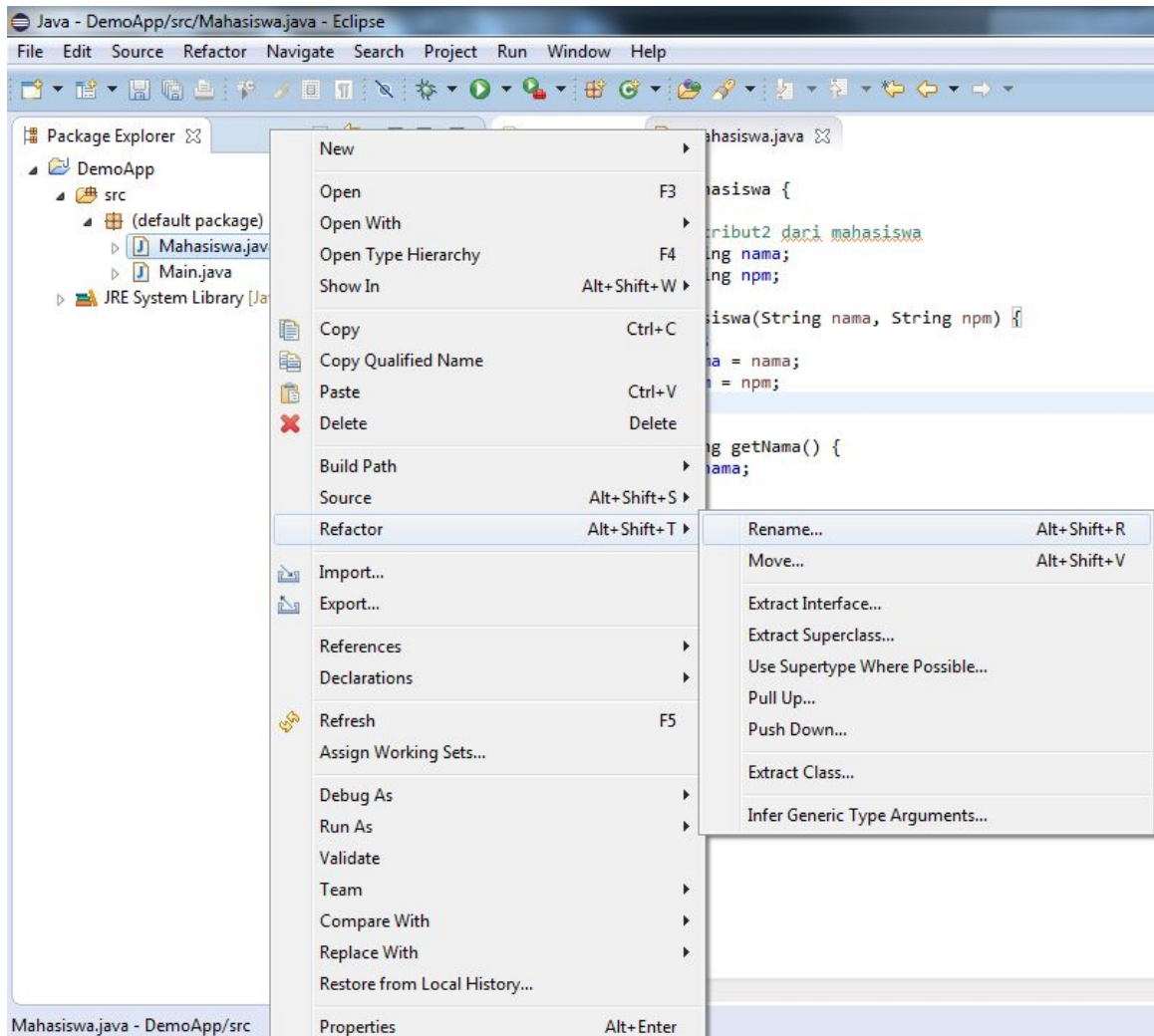


3. Hasil getter and setters yang telah dibuat.

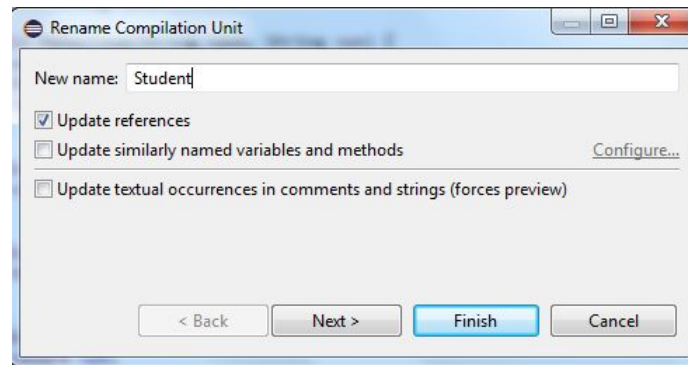
```
public class Mahasiswa {  
  
    //Membuat atribut2 dari mahasiswa  
    private String nama;  
    private String npm;  
  
    public Mahasiswa(String nama, String npm) {  
        super();  
        this.nama = nama;  
        this.npm = npm;  
    }  
  
    public String getNama() {  
        return nama;  
    }  
  
    public void setNama(String nama) {  
        this.nama = nama;  
    }  
  
    public String getNpm() {  
        return npm;  
    }  
  
    public void setNpm(String npm) {  
        this.npm = npm;  
    }  
}
```

Berikut adalah langkah-langkah untuk melakukan *refactoring*.

1. Pilih file yang ingin di-*refactor*, misalnya file Mahasiswa.java yang berisi implementasi kelas Mahasiswa. Setelah itu klik kanan pada file tersebut dan pilih menu Refactor. Dengan refactor, file tersebut dapat dilakukan hal-hal seperti mengganti nama. File Mahasiswa.java tersebut akan diganti namanya.



2. Ganti nama Mahasiswa menjadi Student dan klik Finish.



3. Hasil dari refactor dengan mengganti nama file dari Mahasiswa.java ke Student.java



Soal Tutorial

1. Buat getters dan setters untuk kelas Pengguna
2. Ubah class:
 - a. Pengguna menjadi User
 - b. PenggunaService menjadi UserService
 - c. PenggunaMapper menjadi UserMapper
3. Ubah method
 - a. getPenggunaByUsernameAndPassword menjadi getUserByUsernameAndPassword
 - b. validatePengguna menjadi validateUser
 - c. getPenggunaByUsername menjadi getUserByUsername
4. Pindahkan class
 - a. UserService ke dalam package soal.tutorial.satu.service;
 - b. UserMapper ke dalam package soal.tutorial.satu.model.mapper;
 - c. Hasher ke dalam package soal.tutorial.satu.util;

So, setelah kalian mempelajari IDE, menurut kalian apakah keunggulan dari penggunaan IDE dibanding dengan Editor biasa? Bagaimana jika soal diatas dikerjakan tanpa menggunakan IDE? Jabarkan ke pendapat kalian ke dalam tulisan dan kumpulkan dengan format .pdf (penamaan sesuai format pengumpulan di bawah)

Format Pengumpulan

- KELAS_NPM_TUTORIAL1.zip contoh: A_151617181_TUTORIAL1.zip

- Isi file KELAS_NPM_TUTORIAL.zip:
 - PROJECT_ECLIPSE.zip untuk project dengan Eclipse
 - PROJECT_INTELLIJ.zip untuk project IntelliJ
 - IDE.pdf untuk pendapat kalian mengenai IDE.