

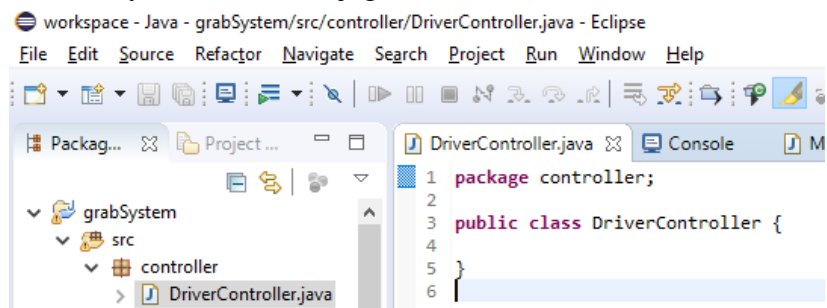
Encapsulation – LC027

Concept (Why using encapsulation)

Maksud dari encapsulation adalah membungkus class dan menjaga apa apa saja yang ada didalam class tersebut, baik method ataupun atribut, agar tidak dapat di akses oleh class lainnya. Untuk menjaga hal tersebut dalam Encapsulation dikenal nama Hak

Akses Modifier yang terdiri dari :

- a. `public` -> akan membuat member(atribut) dan class bisa diakses dari mana saja, jadi `public` memberikan hak akses kepada atribut atau method agar bisa diakses oleh siapapun (property atau class lain diluar class yang bersangkutan), artinya method atau atribut yang ada diclass A dapat diakses oleh siapaun baik itu class A, class B dan seterusnya.
- b. `protected` -> akan membuat member(atribut) dan class bisa diakses dari:
 - class itu sendiri
 - Sub class anak itu
 - Package (class yang berada satu package dengannya).Ket. Package -> digunakan untuk mengelompokkan kelas-kelas yang memiliki kemiripan fungsi. Kalo pake package, di class nya harus di tulis juga. Contohnya:



Jadi, `protected` memberikan hak akses kepada class itu sendiri dan class hasil turunannya (inheritance), artinya apa apa saja yang ada diclass A hanya bisa diakses oleh class A sendiri dan class yang meng Extends class A. Namun harus dipahami class lain yang berada dalam satu package dengan class A mampu mengakses tipe data `protected`, Sedangkan yang tidak mampu mengakses adalah class-class yang berada diluar package class A. untuk dapat mengaksesnya, class yang berada diluar package class A harus meng extends class A.

- c. `Private`
-> Modifier `private` akan membuat member(atribut) hanya bisa diakses oleh dari dalam class itu sendiri. Jadi, `private` akses modifier memberikan hak akses hanya pada class itu sendiri, artinya apa-apa saja yang ada di dalam class A baik itu

method ataupun atribut hanya bisa diakses oleh class A saja, class lain tidak bisa mengaksesnya.

Perlu diingat:

Modifier private tidak bisa diberikan kepada class, enum, dan interface. Modifier private hanya bisa diberikan kepada member class.

Accessor and Mutator (Setter & Getter)

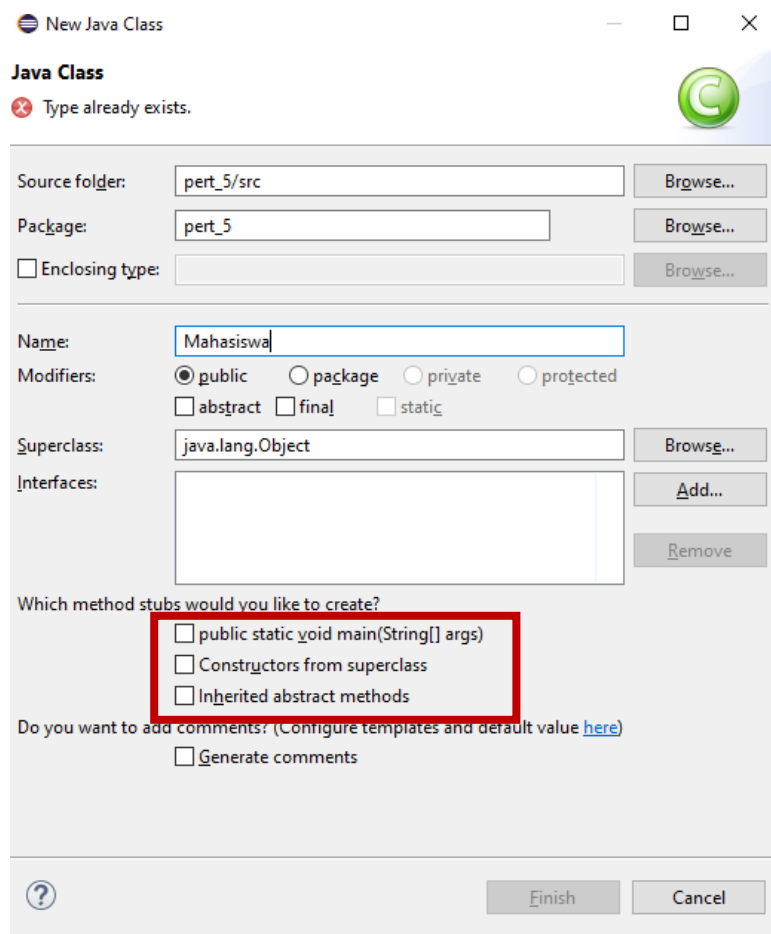
Method setter dan getter adalah dua method yang tugasnya untuk mengambil dan mengisi data ke dalam objek.

Cara membuat Setter Getter (lihat pada contoh kodingan)

1. Buat class Main (**abaikan terlebih ArrayListnya**).
Buat menu di dalam Constructor untuk insert, view, update, delete seperti pada latihan ArrayList/Vector
Dan menggunakan Function / Method.

```
DriverController.java  Console  Main.java  Mahasiswa.java
1 package pert_5;
2
3 import java.util.ArrayList;
4 import java.util.Scanner;
5
6 public class Main {
7     static Scanner scan = new Scanner(System.in);
8     ArrayList<Mahasiswa> mhsList = new ArrayList<>();
9
10    public Main() {
11        // TODO Auto-generated constructor stub
12
13        // ## Cara penggunaan object
14        // Mahasiswa mhs = new Mahasiswa("will" , "2000");
15        // Mahasiswa.semester = 3;
16        // mhs.nama = "william";
17        // mhs.nim = "22000000";
18        //
19        // System.out.println(mhs.nama);
20        // System.out.println(mhs.getNama());
21        // mhs.printIdentitas();
22
23        // ## Latihan
24        int menu = 0;
25        do{
26            menu();
27            menu = scan.nextInt();
28            scan.nextLine();
29
30            switch(menu){
31                case 1: insertMahasiswa();
32                    break;
33                case 2: viewMahasiswa();
34                    break;
35                case 3: deleteMahasiswa();
36                    break;
37                case 4: updateMahasiswa();
38                    break;
39            }
40        }
41    }
42}
```

2. Buat new class untuk object-nya. Berdasarkan contoh kodingan ko VI, dia membuat Object Mahasiswa yang memiliki atribut nama dan nim.



Tidak usah mencentang apa-apa pada saat membuat class object yang nantinya akan di jadikan Encapsulation.

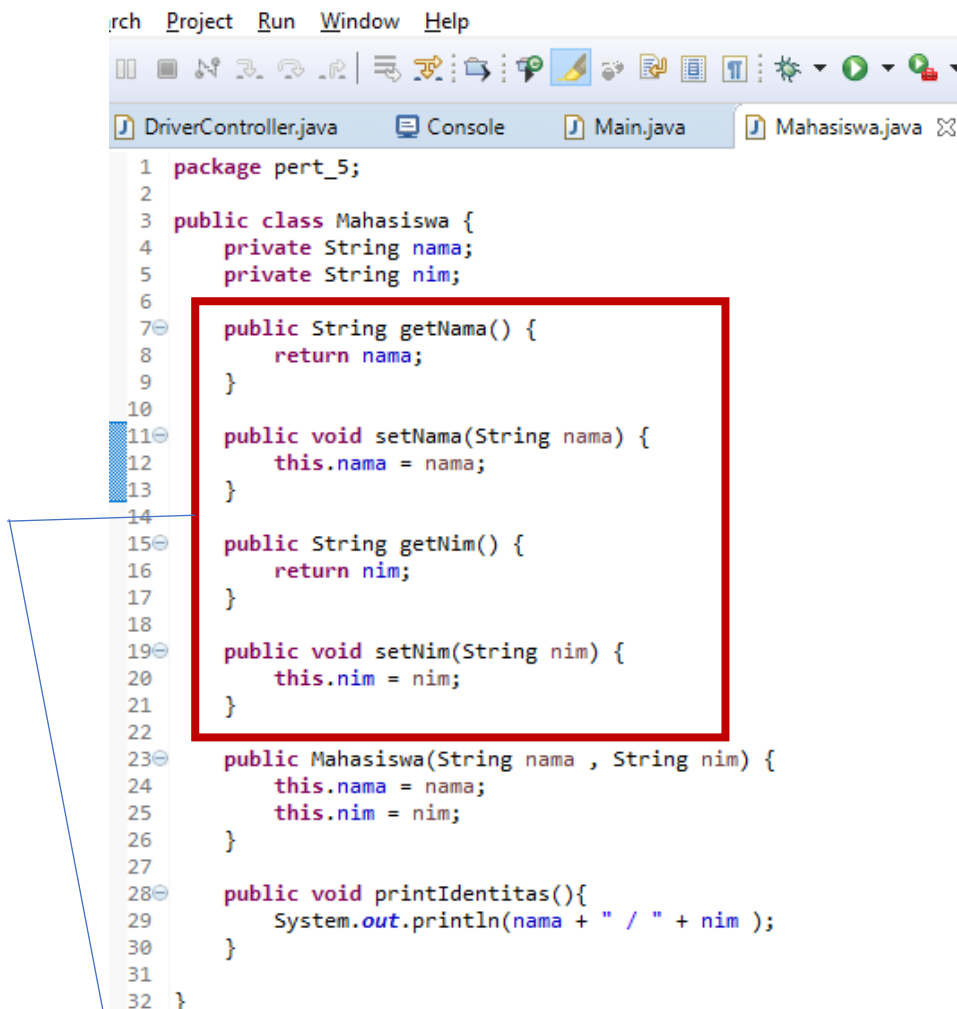
3. Jika sudah terbuat classnya, lalu berikan atribut nama dan nim. Dan berikan juga **access modifier** nya.

wa.java - Eclipse

rch Project Run Window Help

```
DriverController.java Console Main.java *Mahasiswa.java
1 package pert_5;
2
3 public class Mahasiswa {
4     private String nama;
5     private String nim;
6 }
```

4. Lalu buat Setter Getter nya
Membuat setter getter bisa dengan 2 cara.
 - a. Pertama dengan mengetik sendiri.

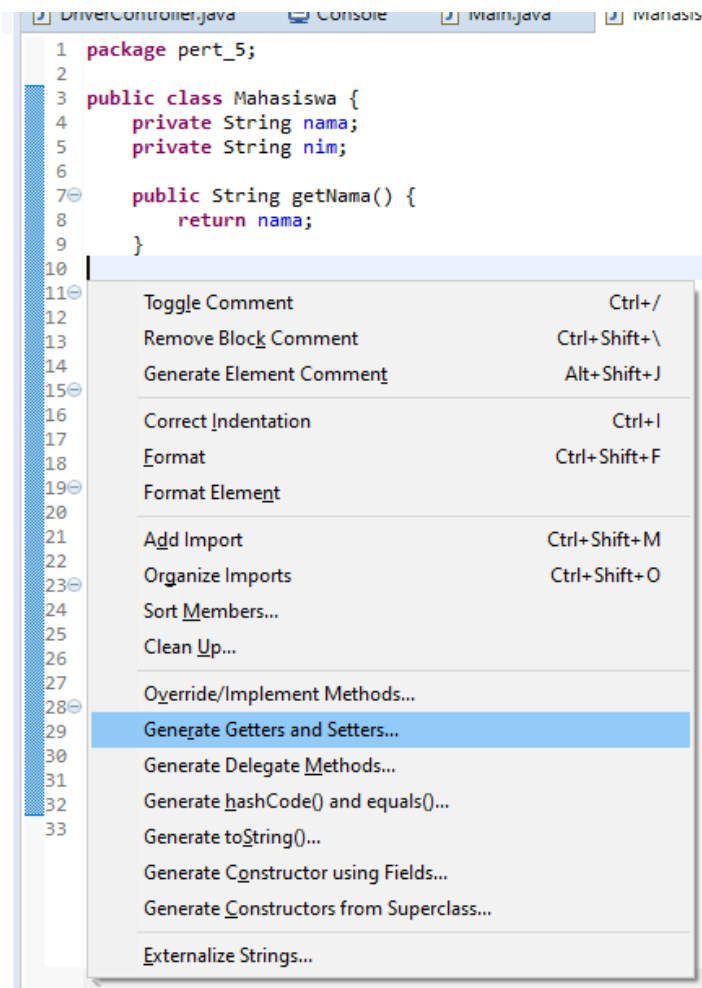


```
1 package pert_5;
2
3 public class Mahasiswa {
4     private String nama;
5     private String nim;
6
7     public String getNama() {
8         return nama;
9     }
10
11     public void setNama(String nama) {
12         this.nama = nama;
13     }
14
15     public String getNim() {
16         return nim;
17     }
18
19     public void setNim(String nim) {
20         this.nim = nim;
21     }
22
23     public Mahasiswa(String nama , String nim) {
24         this.nama = nama;
25         this.nim = nim;
26     }
27
28     public void printIdentitas(){
29         System.out.println(nama + " / " + nim );
30     }
31
32 }
```

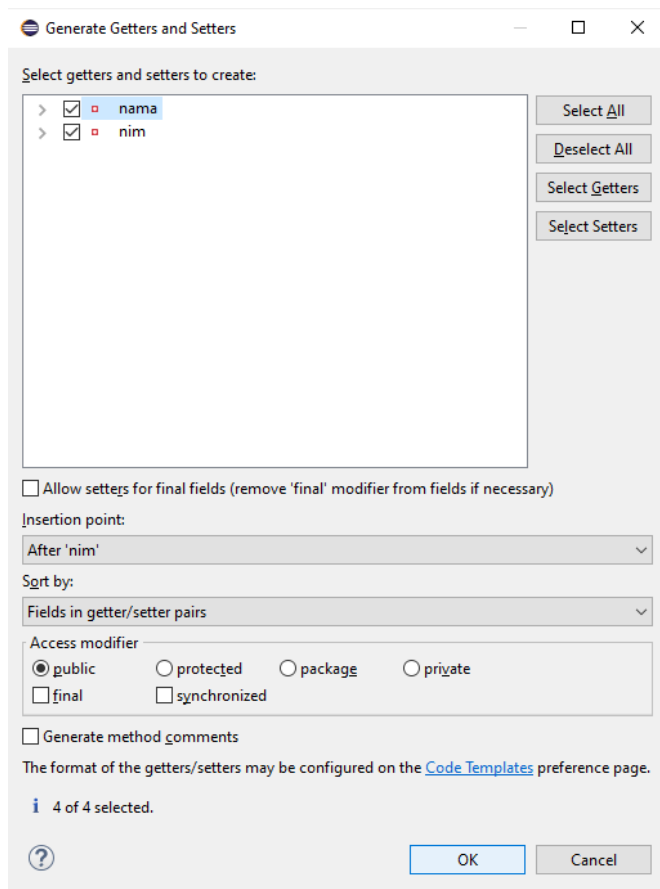
Ini adalah Bagian Setter Getter.

b. Menggunakan shortcut

Tekan -> alt + shift + s + r

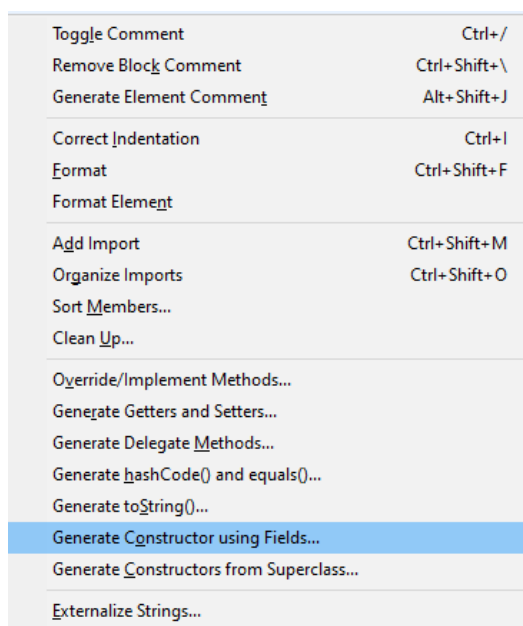


Lalu pilih atribut mana yang akan dibuat getter setter (select All juga bisa), dan tekan OK

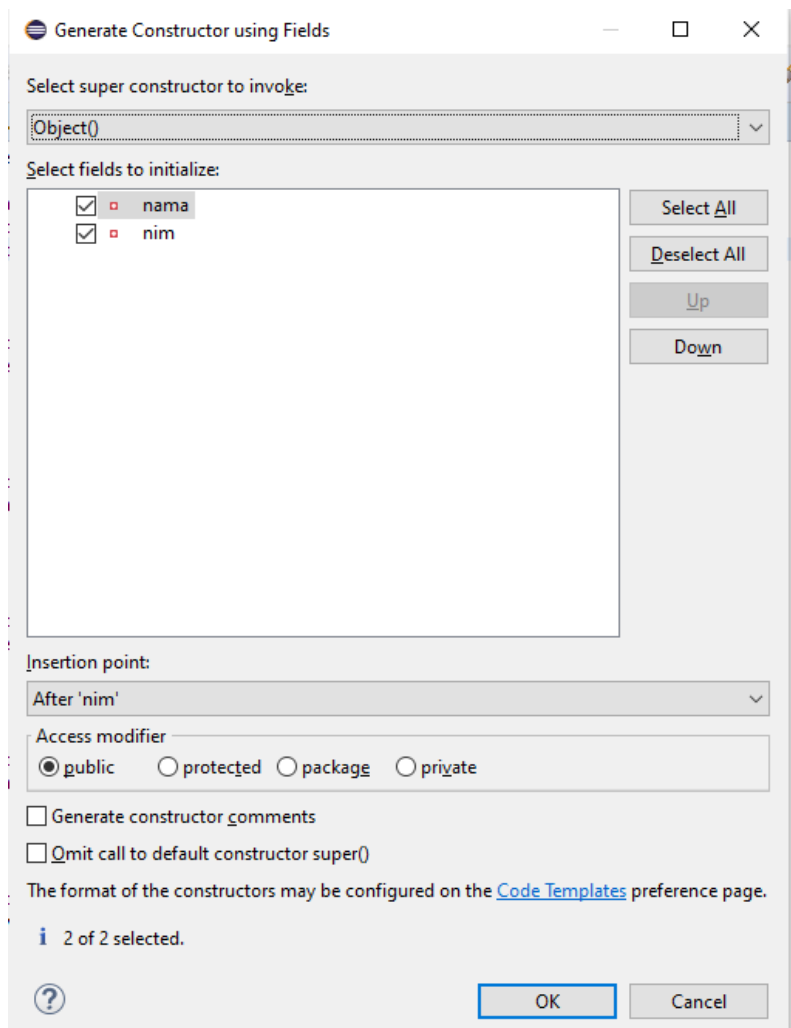


5. Kemudian buat Constructor dari Class tersebut

Cara cepat untuk membuat Constructor. **Alt + shift + s + o**



Lalu pilih atribut yang akan di buat sebagai Object(s) lalu klik OK



Lalu object akan terbuat seperti ini.

```
public Mahasiswa(String nama, String nim) {  
    super();  
    this.nama = nama;  
    this.nim = nim;  
}
```

6. Lalu pada Class Main, buat ArrayList dengan tipe data yang menggunakan Class object Mahasiswa. Menjadi variable global.

```
ArrayList<Mahasiswa> mhsList = new ArrayList<>();
```

7. Lalu buat Object Class Mahasiswa dengan keyword **new**

```
Mahasiswa mhs = new Mahasiswa(nama, nim);
```

Sesuai dengan Object dalam Constructor yang dibuat tadi di Class Mahasiswa.

Variable nama dan nim berupa Variabel String yang sebelumnya sudah di validasi untuk meminta inputan user.

```
private void insertMahasiswa() {  
    // TODO Auto-generated method stub  
    String nama = "", nim = "";
```

8. Lalu Masukan Object yang sudah di buat dengan nama variable 'mhs' kedalam ArrayList

```
private void insertMahasiswa() {  
    // TODO Auto-generated method stub  
    String nama = "", nim = "";  
    System.out.print("insert nama : ");  
    nama = scan.nextLine();  
    System.out.print("insert nim : ");  
    nim = scan.nextLine();  
  
    Mahasiswa mhs = new Mahasiswa(nama, nim);  
    mhsList.add(mhs);  
  
    System.out.println("success...");  
    scan.nextLine();  
}
```

References : <https://www.petanikode.com/java-oop-setter-getter/>