

TUGAS ACCES MODIFIER PEMROGRAMAN BERBASIS OBJECT

Disusun untuk Memenuhi Tugas Mata kuliah Pemrograman Berbasis Object yang Dibimbing
oleh YOGA DWITYA PRAMUDITA, S.Kom., M.Cs.



Disusun oleh:

Achmad Farid Alfa Waid

(190411100073)

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS TRUNOJOYO MADURA

2020

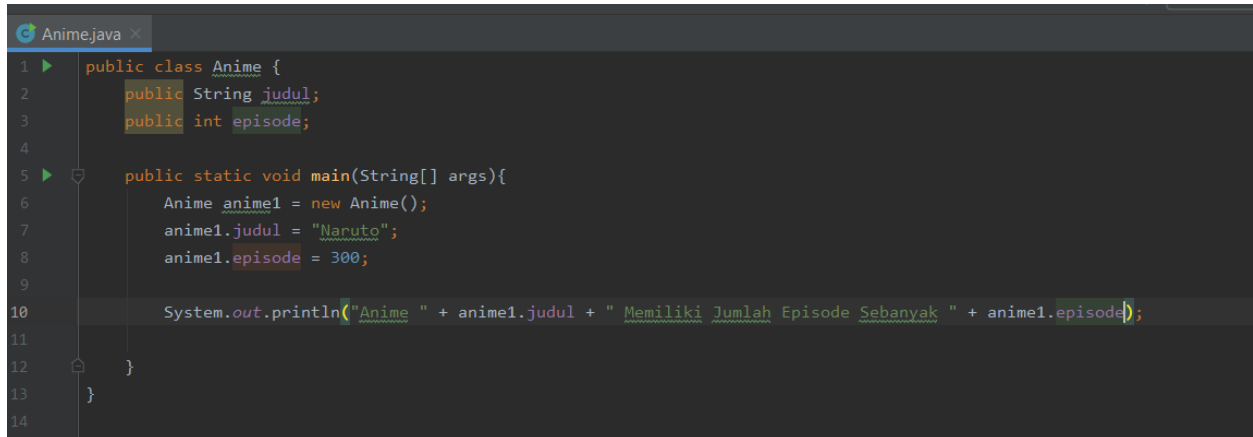
1. Java Acces Modifier: Public, Private, Protected, Default
2. Java Non – Acces Modifier: Final, Abstract, Static

Penerapan

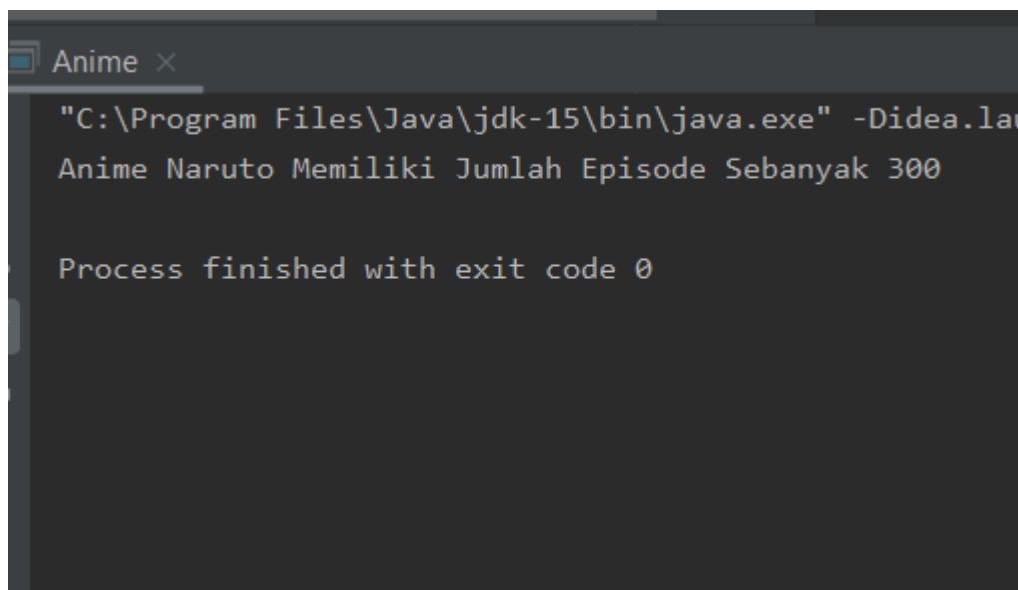
1. Java Acces Modifier

- **Public**

Modifier public akan membuat member dan class bisa di akses dari mana saja.



```
1 public class Anime {
2     public String judul;
3     public int episode;
4
5     public static void main(String[] args){
6         Anime anime1 = new Anime();
7         anime1.judul = "Naruto";
8         anime1.episode = 300;
9
10        System.out.println("Anime " + anime1.judul + " Memiliki Jumlah Episode Sebanyak " + anime1.episode);
11    }
12 }
13
14 }
```

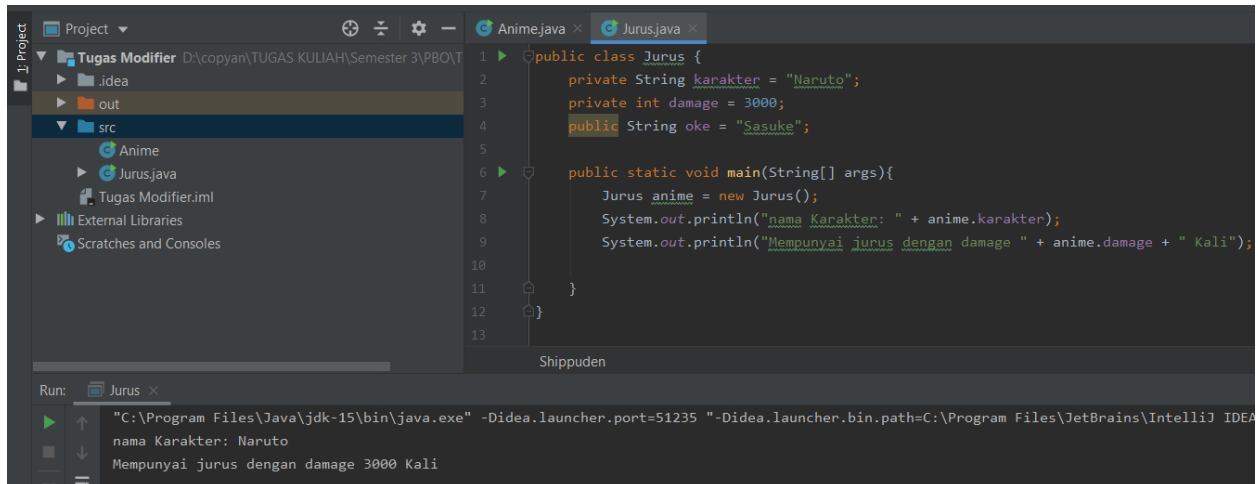


```
Anime x
"C:\Program Files\Java\jdk-15\bin\java.exe" -Didea.la
Anime Naruto Memiliki Jumlah Episode Sebanyak 300

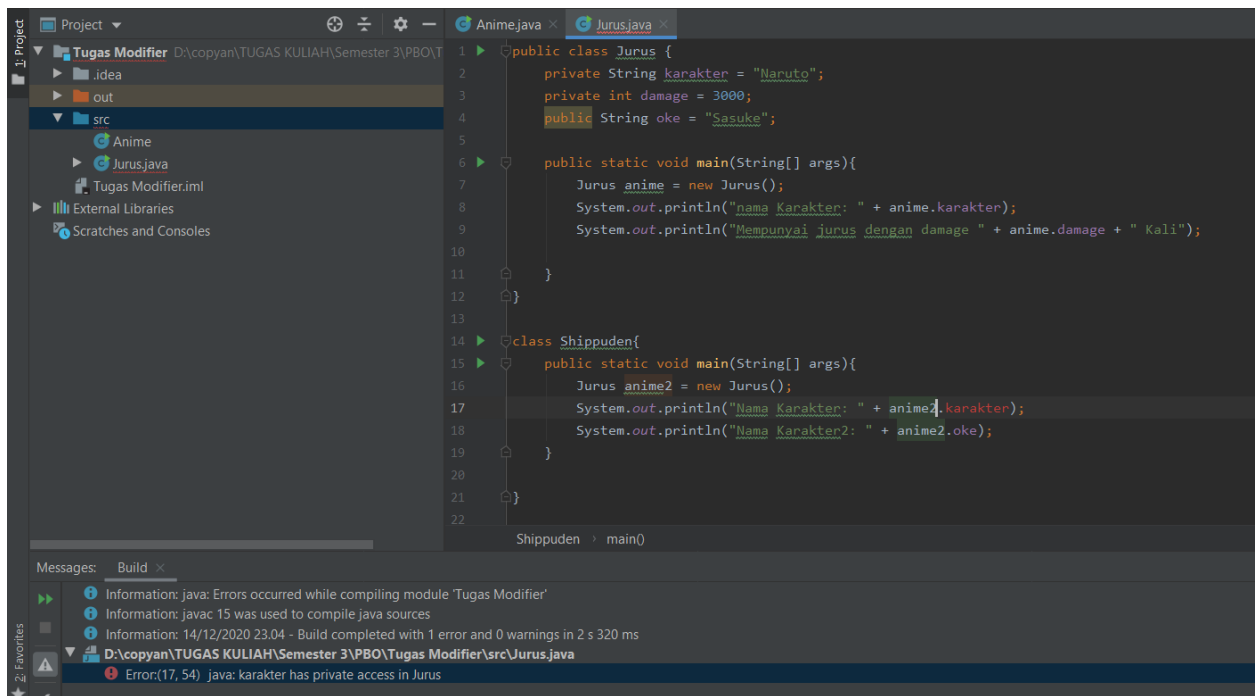
Process finished with exit code 0
```

- **Private**

Hanya bisa di akses di classnya sendiri, diluar class tersebut, tidak akan bisa diakses

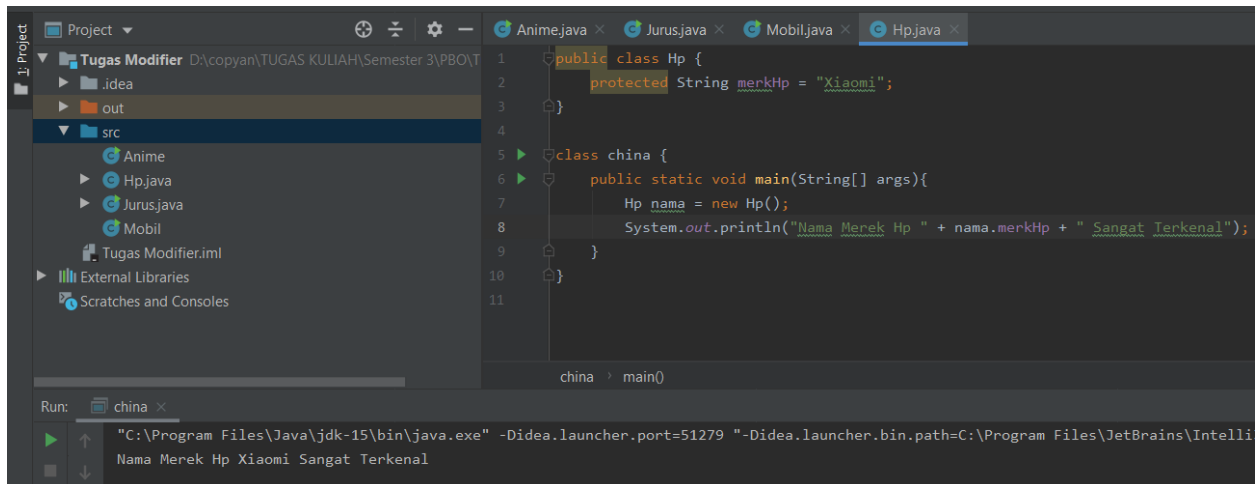


Jika diakses diluar classnya akan eror seperti gambar berikut:



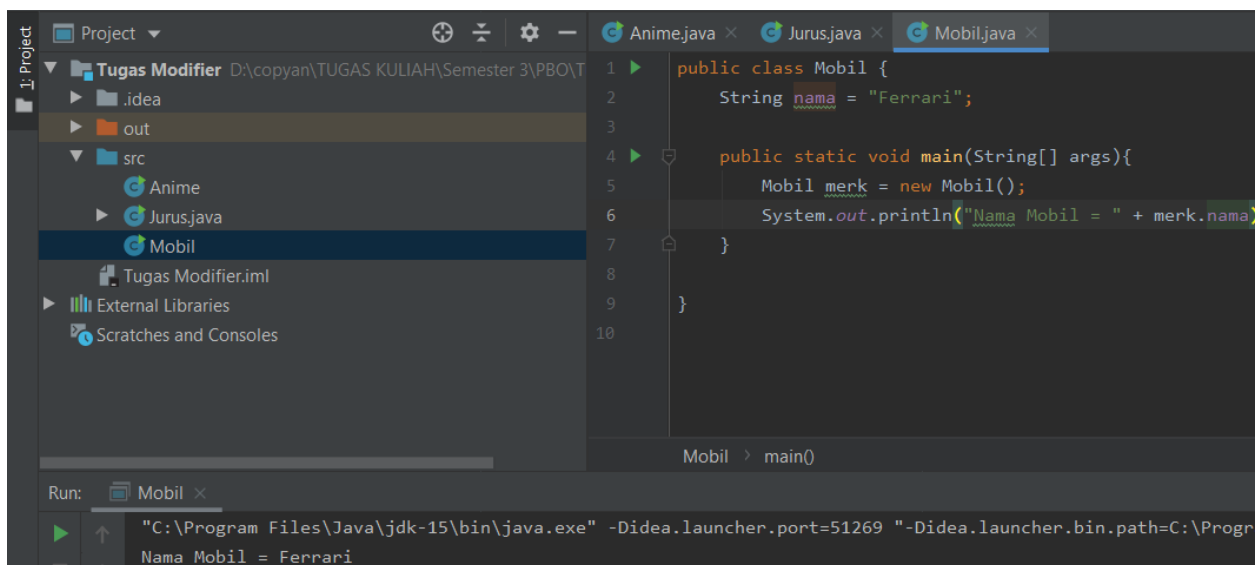
- **Protected**

code dapat diakses pada package yang sama dan subclass (yang artinya harus mewarisi dari class parent). Pada contoh berikut modifier protector diwarisi dari class parent, jadi bisa mengakses modifier tersebut.



- **Default**

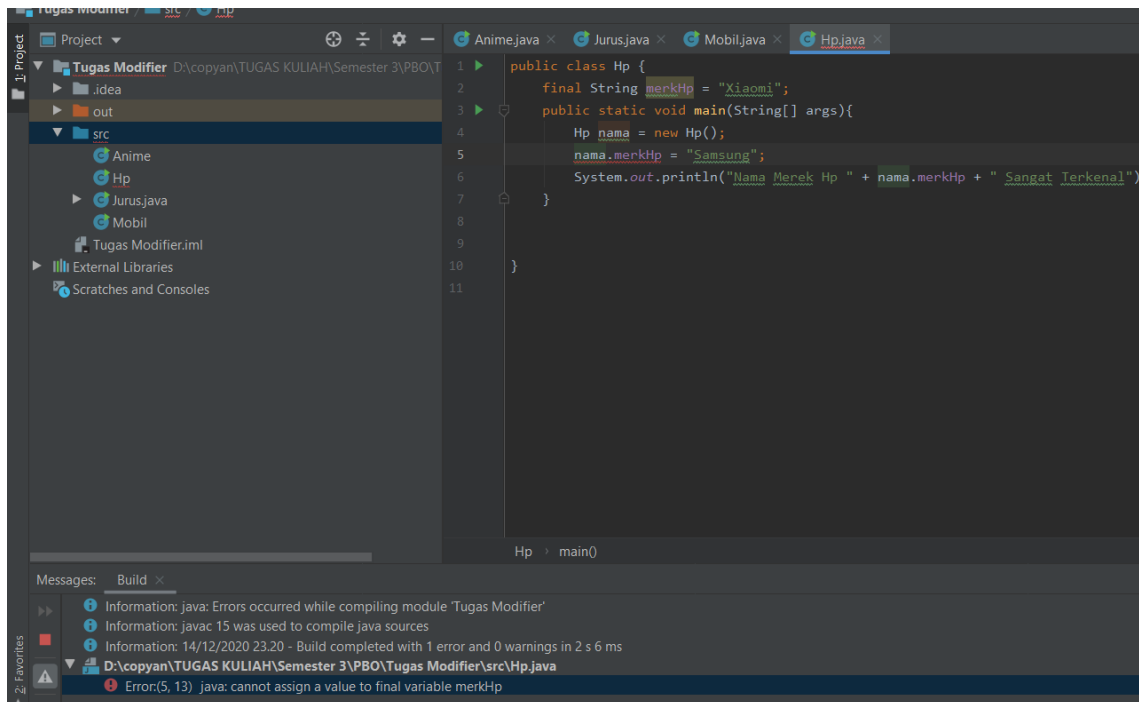
Hanya dapat digunakan pada package yang sama, modifier ini digunakan saat programmer tidak memberikan detail dari modifier



2. Java Non – Acces Modifier

- **Final**

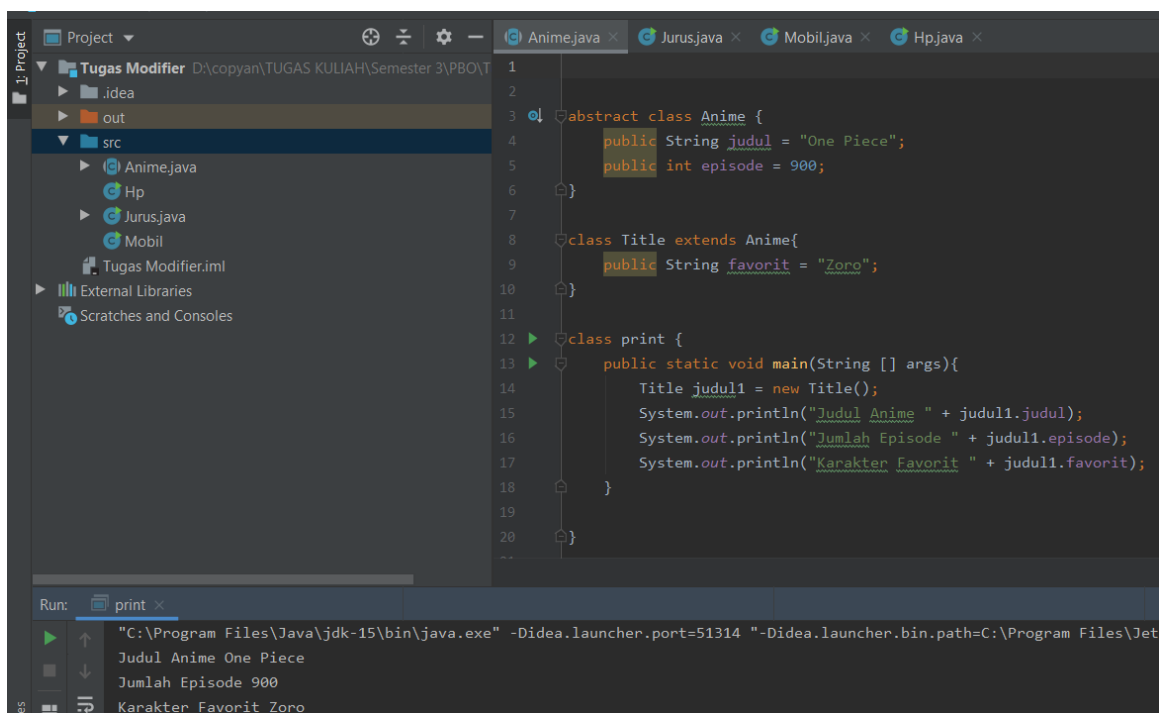
Modifier final digunakan agar suatu atribut atau method bersifat final atau tidak bisa diubah nilainya. Modifier ini digunakan untuk membuat konstanta di Java.



Pada contoh diatas dengan modifier final, kemudin selanjutnya valuenya akan dirubah, maka akan error.

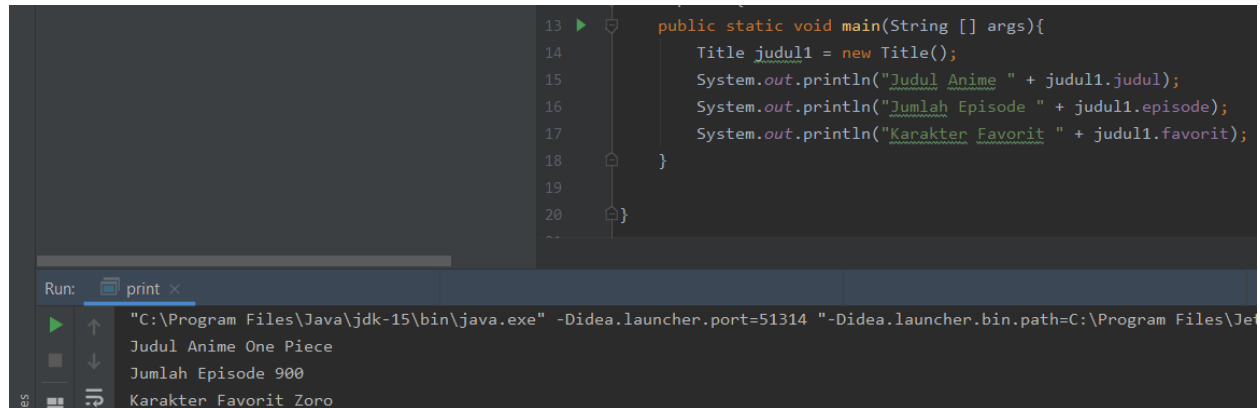
- **Abstract**

Modifier Abstract tidak bisa digunakan untuk membuat object, jadi harus menurunkan ke class lain yang mewaris class modifier abstract. Llau membuat class lain untuk mengakses nilai dari object class turunan.



- **Static**

Modifier Static digunakan saat kita ingin membuat class yang tujuannya tidak sebagai object, misal untuk membuat object dari class lain dan mengeluarkan value object tersebut.



The screenshot shows an IDE with a Java code editor and a console window. The code defines a `Title` class with attributes `judul`, `episode`, and `favorit`. The `main` method creates an instance of `Title` and prints its details.

```
13 public static void main(String [] args){  
14     Title judul1 = new Title();  
15     System.out.println("Judul Anime " + judul1.judul);  
16     System.out.println("Jumlah Episode " + judul1.episode);  
17     System.out.println("Karakter Favorit " + judul1.favorit);  
18 }  
19  
20 }
```

The console window shows the output of the program:

```
Run: print x  
"C:\Program Files\Java\jdk-15\bin\java.exe" -Didea.launcher.port=51314 "-Didea.launcher.bin.path=C:\Program Files\Jet  
Judul Anime One Piece  
Jumlah Episode 900  
Karakter Favorit Zoro
```