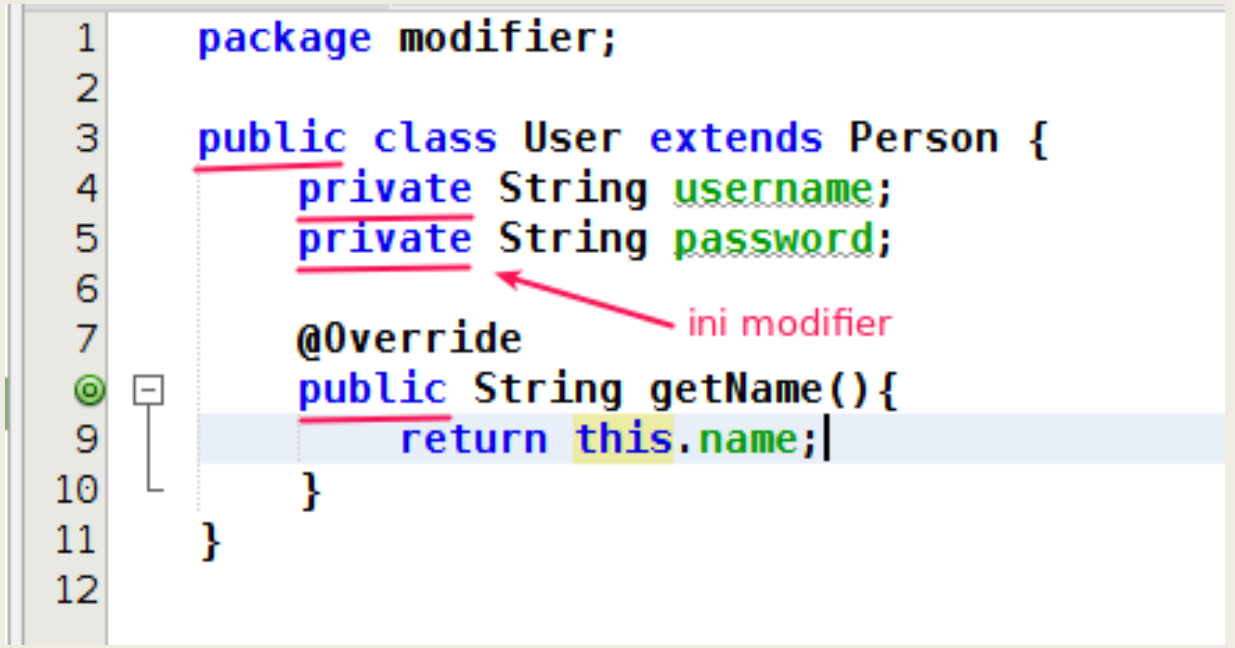
TINGKATAN AKSES MEMBER DAN CLASS (MODIFIER)

- Class dalam program Java dapat saling berhubungan dengan cara memberikan akses terhadap member mereka.
- Salah satu hubungan class yang pernah kita pelajari adalah inheritance (pewarisan).
- Semua yang ada di dalam class (atribut dan method) disebut member. Biasanya akan ada tingkatan akses yang disebut modifier.
- Pada hubungan inheritance, semua member di dalam class induk akan bisa diakses oleh class anak (subclass), kecuali memeber tersebut diberikan modifier private.
- Modifier tidak hanya bisa diberikan kepada member saja. Tapi juga bisa diberikan kepada interface, enum, dan class itu sendiri.

Perhatikan kode program berikut:



Ada 3 Macam Modifier dalam Java

- Secara umum ada 3 macam modifier yang digunakan dalam Java: public, private, dan protected.
- Apabila kita tidak menggunakan tiga kata kunci tersebut, maka member atau class itu tidak menggunakan modifier *(no modifier)*.
- Masing-masing modifier akan menentukan di mana saja member bisa diakses.

Berikut ini tabel jangkauan untuk masing-masing modifier:

Modifier	Class	Package	Subclass	World
public	Y	Y	Y	Y
protected	Y	Y	Y	N
no modifier	Y	Y	N	N
private	Y	N	N	N

Keterangan:
- Y artinya bisa diakses;
- N artinya tidak bisa diakses;
- Subclass artinya class anak;
- World artinya seluruh package di aplikasi.

- Pada tabel di atas... apabila kita tidak menggunakan modifier (no modifier), maka class dan member hanya akan bisa diakses dari Class itu sendiri dan package (class yang berada satu package dengannya).

- Agar bisa diakses dari mana saja, maka kita harus memberikan modifier public.

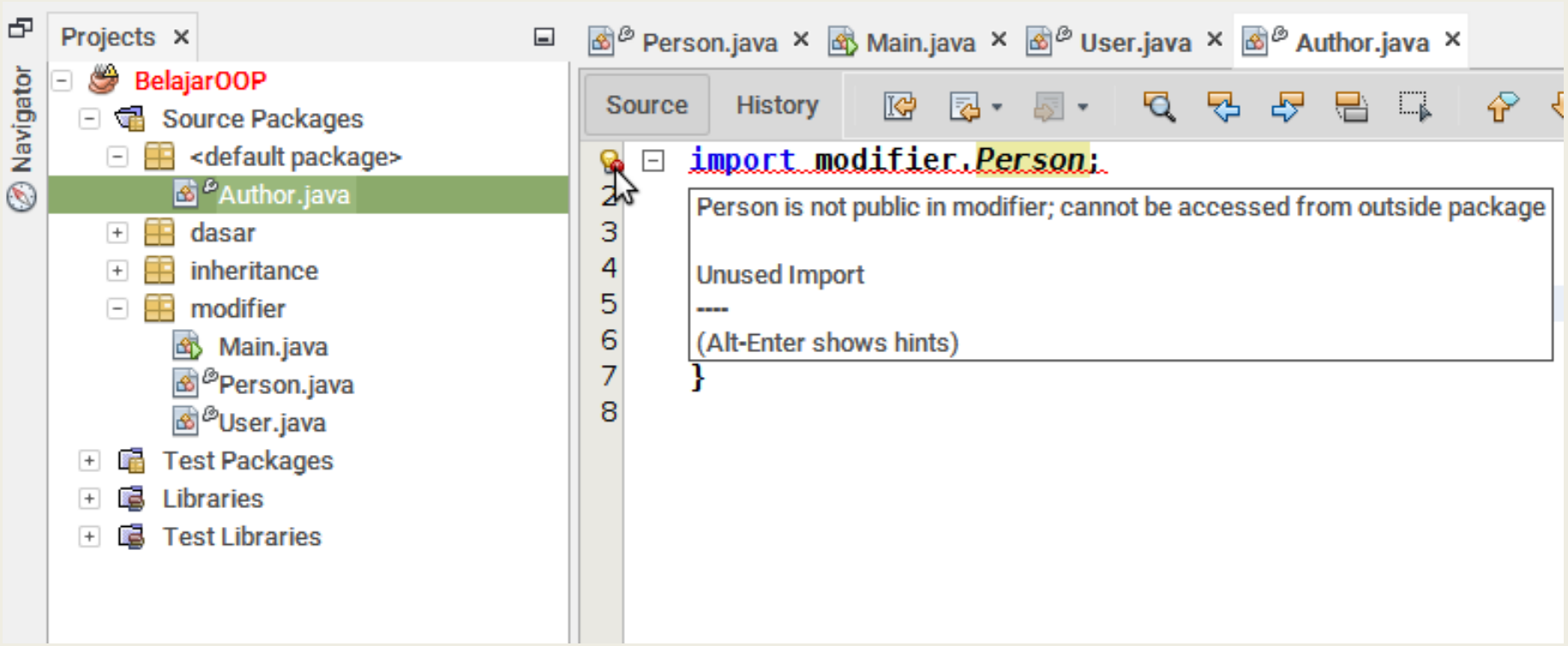
Contoh
1. Public

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

```
package modifier;

class Person {
    public String name;

    public changeName(String newName){
        this.name = newName;
    }
}
```

- Pada class Preson terdapat dua member, yaitu:
  - *atribut name*
  - *method changeName()*

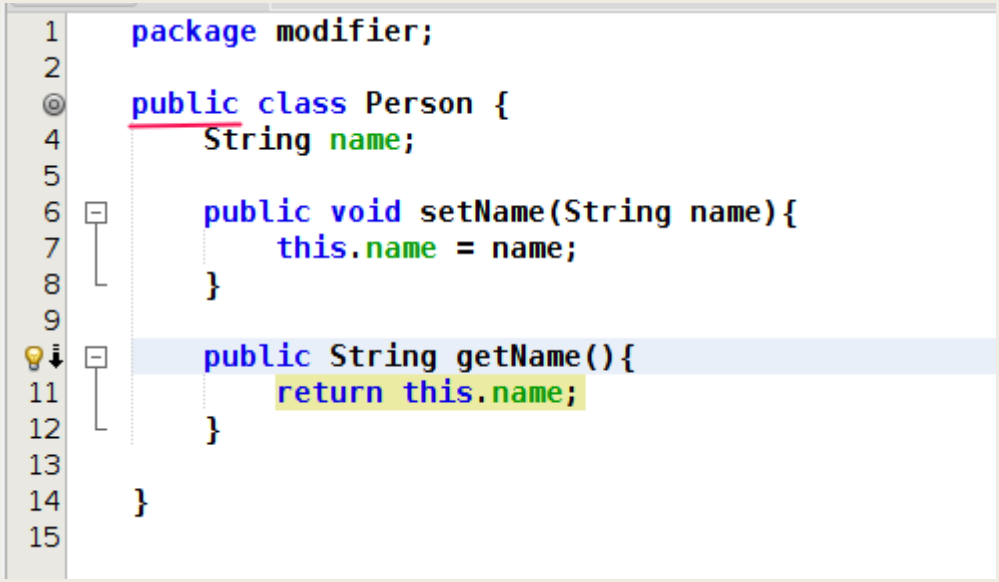
- Kedua member tersebut kita berikan modifier public. Artinya mereka akan bisa diakses dari mana saja.
- Namun, class Person tidak kita berikan modifier. Maka yang akan terjadi adalah class tersebut tidak akan bisa diimpor (diakses) dari luar package.



- Class Person berada di dalam package modifier, lalu kita coba akses dari default package, maka yang akan terjadi adalah error seperti gambar di atas.

Solusinya

- kita harus menambahkan modifier public ke dalam class Person.

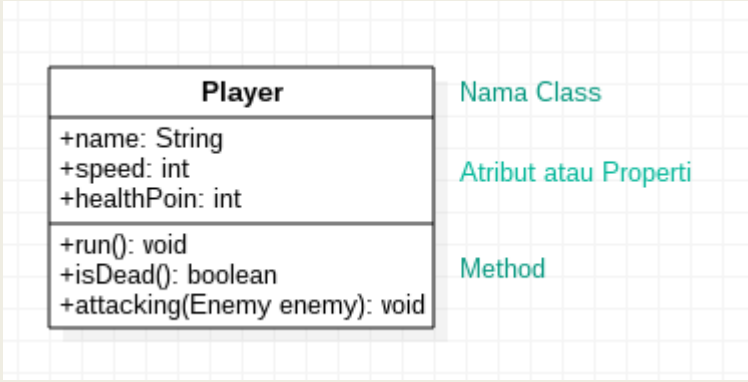


```
1   package modifier;
2
4   public class Person {
4       String name;
5
6       public void setName(String name){
7           this.name = name;
8       }
9
11      public String getName(){
11          return this.name;
12      }
13
14  }
15
```

- Maka error akan menghilang dan class Person akan bisa diimpor dari package manapun.

Pada class diagram

- Pada class diagram, modifier public digambarkan dengan simbol plus (+).



- Semua member dalam class Player memiliki modifier public. Perhatikan simbol + yang ada di depannya.

Contoh :
2. Private

- Modifier private akan membuat member hanya bisa diakses oleh dari dalam class itu sendiri.

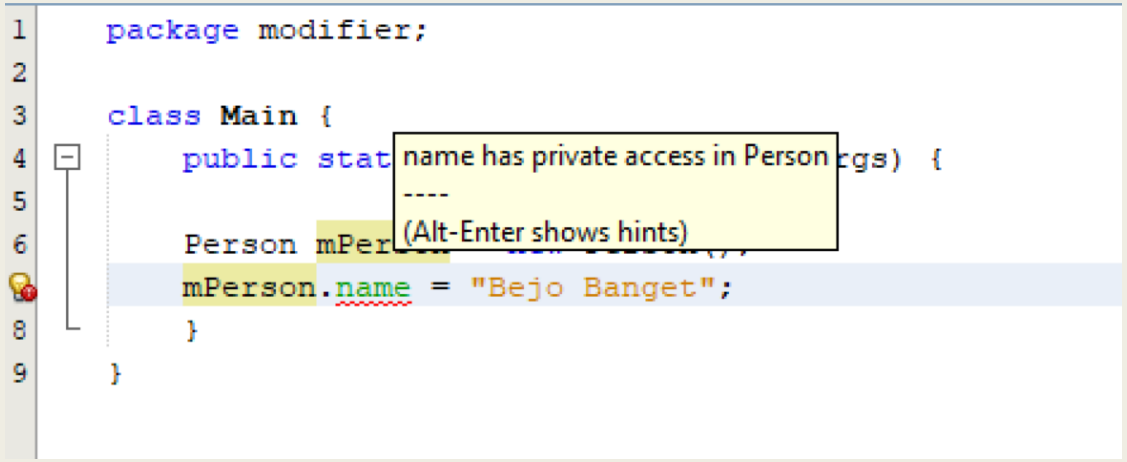
Perlu diingat:

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

```
class Person {
    private String name;

    public void setName(name){
        this.name = name;
    }

    public String getName(){
        return this.name;
    }
}
```

- Pada contoh di atas, kita memberikan modifier private pada atribut name dan modifier public pada method setName() dan getName().
- Apabila kita coba mengkases langsung atribut name seperti ini:

```
Person mPerson = new Person()
mPerson.name = "Bejo Banget"; // <- maka akan terjadi error di sini
```

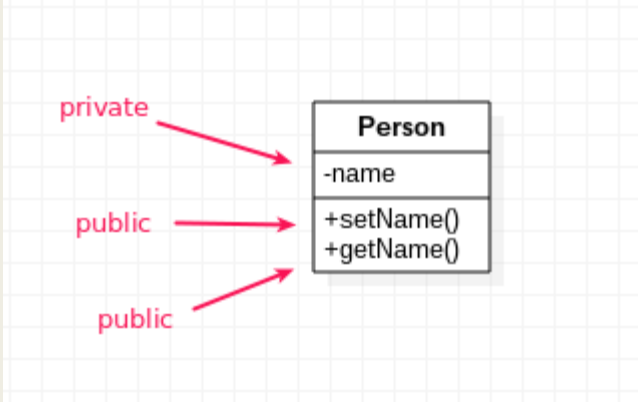


- Lalu, bagaimana cara mengakses member private dari luar class?
- Kita bisa memanfaatkan method setter dan getter. Karena, method ini akan selalu diberikan modifier public.
- Contoh:

```
Person mPerson = new Person();
mPerson.setName("Bejo Banget");

System.out.println("Person Name: " + mPerson.getName());
```

Class Diagarm

- Pada class diagram, modifier private digambarkan dengan simbol minus (-).



Contoh
3. Protected

- Modifier protected akan membuat member dan class hanya bisa diakses dari:
  - *Class itu sendiri;*
  - *Sub class atau class anak;*
  - *Package (class yang berada satu package dengannya).*

- Modifier protected juga hanya boleh digunakan pada member saja.

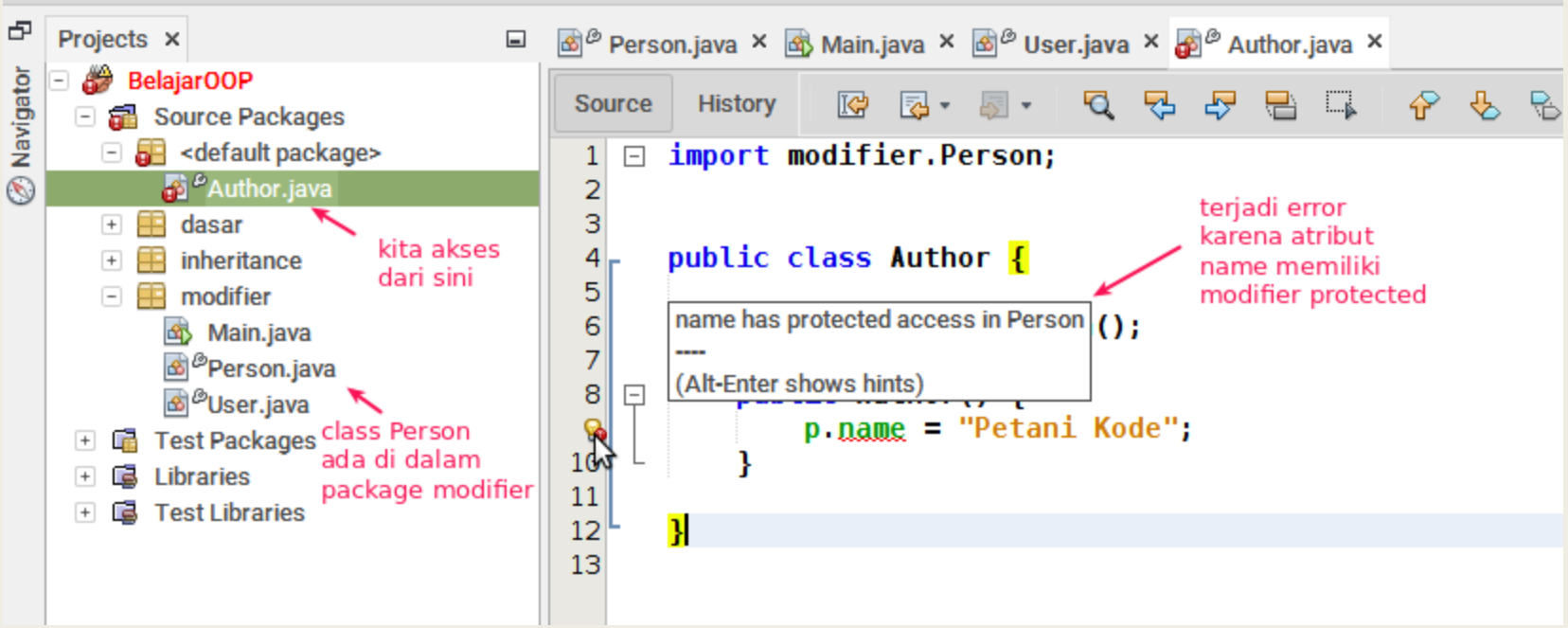
```java
package modifier;

public class Person {
    protected String name;

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

    public String getName(){
        return this.name;
    }
}
```

- Pada contoh di atas, kita memberikan modifier protected pada atribut name.

- Apabila kita coba mengakses dari class yang satu package dengannya, maka tidak akan terjadi error.

- Namun, apabila kita mencoba mengakses dari luar package seperti ini:

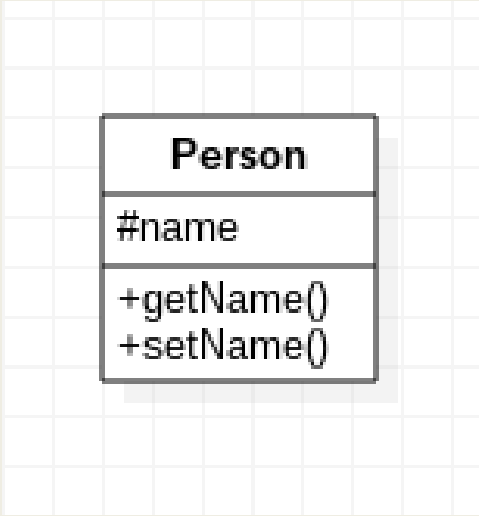
```
import modifier.Person;

public class Author {

    Person p = new Person();

    public Author() {
        // akan terjadi error di sini karena atribut name
        // telah diberikan modifier protected
        p.name = "Bejo Banget";
    }

}
```


Projects
Navigator
BelajarOOP
Source Packages
<default package>
Author.java
dasar
inheritance
modifier
Main.java
Person.java
User.java
Test Packages
Libraries
Test Libraries
kita akses dari sini
class Person ada di dalam package modifier
Person.java
Main.java
User.java
Author.java
Source
History
import modifier.Person;
public class Author {
name has protected access in Person

(Alt-Enter shows hints)
();
p.name = "Petani Kode";
}
terjadi error karena atribut name memiliki modifier protected

Class Diagaram

- Pada class diagram ,modifier protected digambarkan dengan tanda pagar (#).



Kesimpulan

- Kita sudah mengetahui 3 macam modifier dalam Java beserta contohnya.
- Masing-masing modifier akan menentukan batasan akses untuk member dan class.
- Kita juga sudah mengetahui cara menggambarkan sebuah class dalam bentuk Class Diagaram