

Pemrograman Berorientasi Objek

MODUL 2 ACCESS MODIFIER DAN ENKAPSULASI
ASISTEN LABORATORIUM

Tujuan Pembelajaran :

1. Mengetahui apa itu Access Modifier
2. Mengetahui apa fungsi dari setiap access modifier
3. Mengetahui bagaimana apa itu enkapsulasi
4. Mengetahui bagaimana contoh dari enkapsulasi

Access Modifier

Seperti namanya access modifier membantu untuk membatasi ruang lingkup dari sebuah kelas, dari konstruktor, variable, method dan data member. Ada 4 jenis access modifier :

1. Default / Package
2. Private
3. Public
4. Protected

	default	private	protected	public
Same Class	Yes	Yes	Yes	Yes
Same package subclass	Yes	No	Yes	Yes
Same package non-subclass	Yes	No	Yes	Yes
Different package subclass	No	No	Yes	Yes
Different package non-subclass	No	No	No	Yes

1. Default / Package – saat dimana sebuah method/variable tidak menggunakan access modifier yang di spesifikasikan didalam kelas. Contoh :

```
package package1;

public class Cat {
    String name;
    String type;
    int age;

    void sound(){
        System.out.println(this.name+" sound is so beautiful");
    }
}
```

```
}  
  
package package1;  
  
import package1.Cat;  
  
public class Result {  
    public static void main(){  
        Cat myPet = new Cat();  
        myPet.name = "Ars";  
        myPet.type = "Persia";  
        myPet.age = 17;  
        myPet.sound();  
    }  
}
```

2. Private – access modifier yang method, variable hanya dapat diakses didalam class tersebut. Tidak ada yang dapat mengakses method/variable yang menggunakan access modifier walaupun itu didalam package yang sama.
3. Protected – access modifier yang method, variable gunakan hanya dapat diakses didalam packagenya dan sub-class di luar dari package.
4. Public – access modifier yang method, variable gunakan yang dapat digunakan dipackage manapun saat di buat di luar dari package.

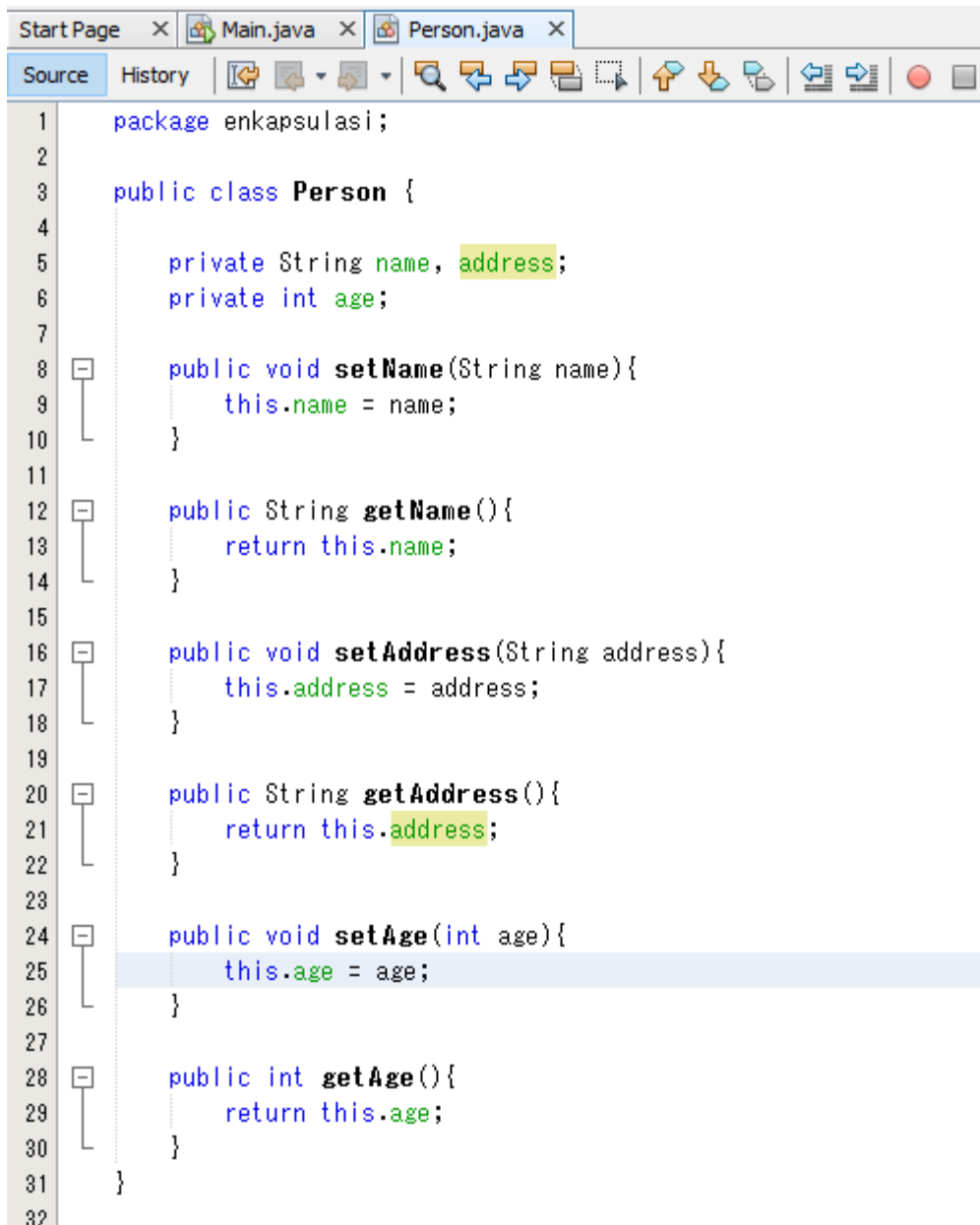
Enkapsulasi

Enkapsulasi di definisikan sebagai pembungkus data di dalam sebuah unit. Ini adalah sebuah mekanisme untuk mengikat dan memanipulasi semua data untuk melindungi dan menghalangi kode diakses dari luar.

- Biasanya enkapsulas, variable atau data yang berada didalam kelas akan disembunyikan dari kelas lainnya dan hanya dapat diakses melalui fungsi yang sudah disediakan.
- Enkapsulasi dapat dideklarasikan saat menggunakan access modifier private pada variable dan untuk menulis dan mengambilnya menggunakan public method.

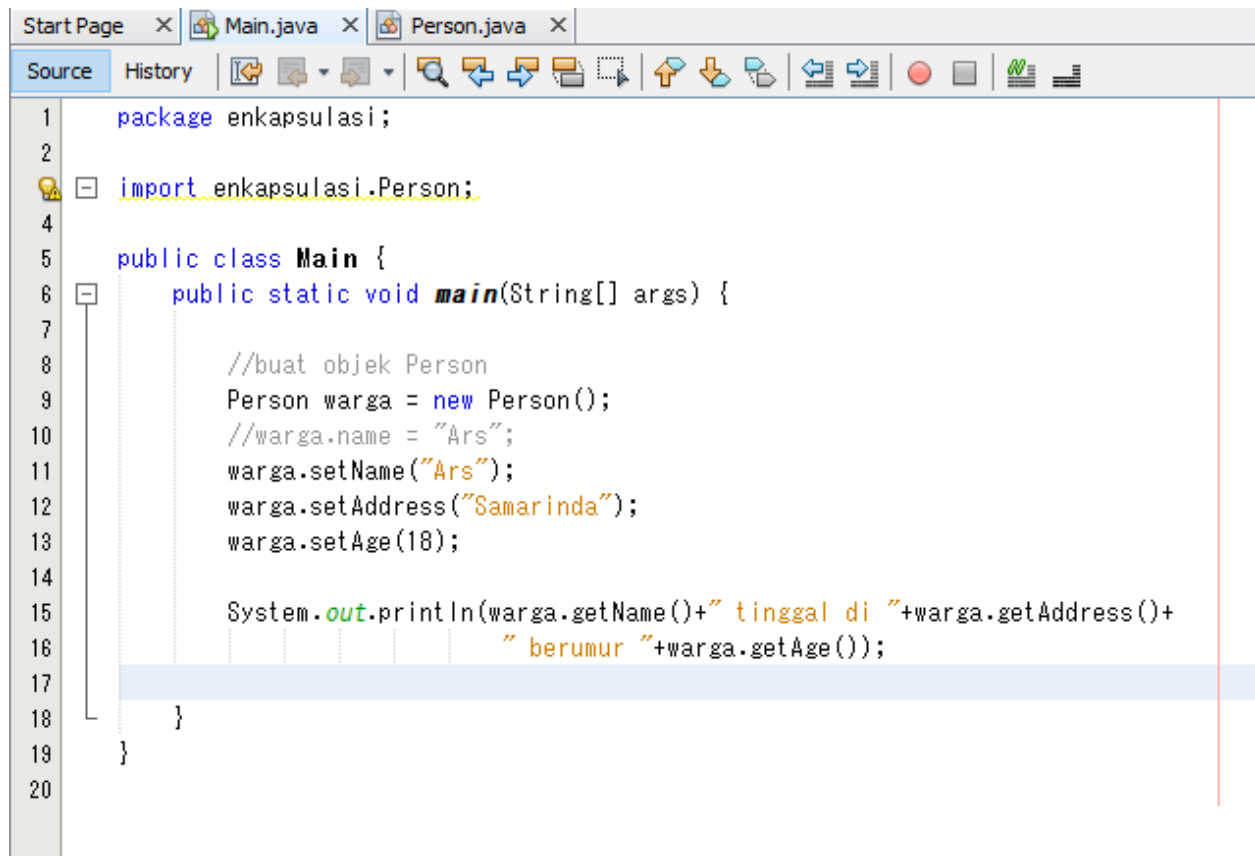
Setter dan Getter

Setter dan getter adalah method yang digunakan enkapsulasi untuk mengakses data / variable yang disembunyikan di dalam sebuah kelas. Set adalah method yang digunakan untuk memberikan nilai kedalam variable pada objek serta getter adalah sebuah method yang digunakan untuk mengambil sebuah nilai dari variable pada objek. Contoh :



```
1 package enkapsulasi;
2
3 public class Person {
4
5     private String name, address;
6     private int age;
7
8     public void setName(String name){
9         this.name = name;
10    }
11
12    public String getName(){
13        return this.name;
14    }
15
16    public void setAddress(String address){
17        this.address = address;
18    }
19
20    public String getAddress(){
21        return this.address;
22    }
23
24    public void setAge(int age){
25        this.age = age;
26    }
27
28    public int getAge(){
29        return this.age;
30    }
31 }
32
```

MODUL 2 ACCESS MODIFIER DAN ENKAPSULASI



```
1 package enkapsulasi;
2
3 import enkapsulasi.Person;
4
5 public class Main {
6     public static void main(String[] args) {
7
8         //buat objek Person
9         Person warga = new Person();
10        //warga.name = "Ars";
11        warga.setName("Ars");
12        warga.setAddress("Samarinda");
13        warga.setAge(18);
14
15        System.out.println(warga.getName()+" tinggal di "+warga.getAddress()+
16                           " berumur "+warga.getAge());
17    }
18 }
19
20
```

Dari syntax yang di atas bahwa kita mengetahui semua variable yang berada pada class person menggunakan private sehingga tidak dapat diakses di class Main sehingga untuk mengakses variable yang berada di class person harus menggunakan method setter dan getter yang sudah kita buat di class person.