

Tugas Praktikum Pemrograman Berorientasi Objek



Disusun Oleh :

Agil Deriansyah Hasan
4522210125

Dosen Pengampu :

Adi Wahyu Pribadi , S.Si., M.Kom
Prak. Pemrograman Berorientasi Objek - A

S1-Teknik Informatika
Fakultas Teknik
Universitas Pancasila 2023/2024

Person.java

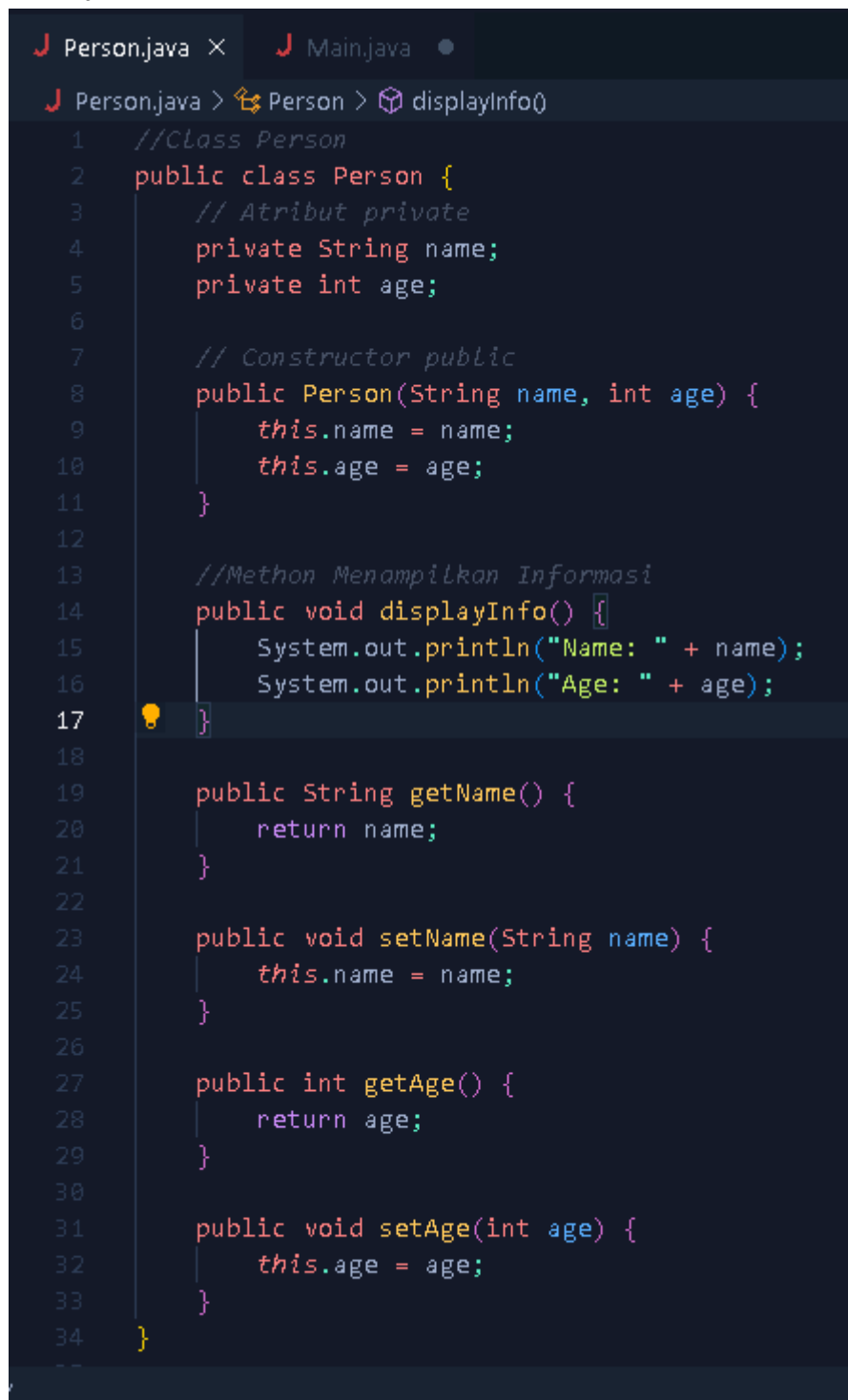
```
J Person.java x
J Person.java > Person > displayInfo()
1 //Class Person
2 public class Person {
3     // Atribut private
4     private String name;
5     private int age;
6
7     // Constructor public
8     public Person(String name, int age) {
9         this.name = name;
10        this.age = age;
11    }
12
13    //Methon Menampilkan Informasi
14    public void displayInfo() {
15        System.out.println("Name: " + name);
16        System.out.println("Age: " + age);
17    }
18
```

Main.java

```
J Person.java x J Main.java •
J Main.java > Main > main(String[])
1 public class Main {
2     Run | Debug
3     public static void main(String[] args) {
4         // Membuat objek Person
5         Person person1 = new Person(name:"Agil", age:20);
6
7         person1.displayInfo();
8
9         person1.setName(name:"Seira");
10
11        person1.setAge(age:21);
12
13        // Menampilkan informasi setelah dimodifikasi
14        person1.displayInfo();
15    }
16
```

Pada tahap ini, kita membuat class dasar bernama Person yang memiliki atribut dasar berupa name dan age. Keduanya bersifat private untuk memastikan enkapsulasi, sehingga atribut ini hanya dapat diakses melalui metode (getter/setter). Constructor digunakan untuk menginisialisasi nilai atribut ketika objek diciptakan.

Menambahkan Setter dan Getter Person.java



```
Person.java X Main.java
Person.java > Person > displayInfo()
1 //Class Person
2 public class Person {
3     // Atribut private
4     private String name;
5     private int age;
6
7     // Constructor public
8     public Person(String name, int age) {
9         this.name = name;
10        this.age = age;
11    }
12
13    //Methon Menampilkan Informasi
14    public void displayInfo() {
15        System.out.println("Name: " + name);
16        System.out.println("Age: " + age);
17    }
18
19    public String getName() {
20        return name;
21    }
22
23    public void setName(String name) {
24        this.name = name;
25    }
26
27    public int getAge() {
28        return age;
29    }
30
31    public void setAge(int age) {
32        this.age = age;
33    }
34 }
```

Setter digunakan untuk mengubah nilai atribut private, dan getter digunakan untuk mengambil nilai atribut tersebut. Misalnya, kita dapat menggunakan `person.setName("Agil")` untuk mengubah nama objek `Person`, dan `person.getName()` untuk mengambil nilai nama yang disimpan.

Modifikasi dan Debugging

```
Person.java
1 //Class Person
2 public class Person {
3     // Atribut private
4     private String name;
5     private int age;
6     private String address;
7
8     // Constructor public
9     public Person(String name, int age,String address)
10         this.name = name;
11         this.age = age;
12         this.address = address;
13     }
14
15     //Methon Menampilkan Informasi
16     public void displayInfo() {
17         System.out.println("Name: " + name);
18         System.out.println("Age: " + age);
19         System.out.println("Address:" + address);
20     }
21
22     public String getName() {
23         return name;
24     }
25
26     public void setName(String name) {
27         this.name = name;
28     }
29
30     public int getAge() {
31         return age;
32     }
33
34     public void setAge(int age) {
```

```
Main.java
1 public class Main {
2     Run | Debug
3     public static void main(String[] args) {
4         // Membuat objek Person
5         Person person1 = new Person(name:"Agil", age:20, address:"Bekasi");
6         person1.displayInfo();
7
8         System.out.println(x:"Mengubah Data");
9
10        person1 = new Person(name:"Seira",age:21,address:"Jakarta");
11        // Menampilkan informasi setelah dimodifikasi
12        person1.displayInfo();
13    }
14 }
15
```

```
D:\PBO>java Main.java
Name: Agil
Age: 20
Address:Bekasi
Mengubah Data
Name: Seira
Age: 21
Address:Jakarta
```

Pada tahap ini, kita menambahkan atribut address untuk menyimpan informasi alamat. Constructor diubah agar bisa menginisialisasi address saat objek dibuat. Method displayInfo juga dimodifikasi untuk menampilkan nilai alamat selain name dan age. Selain itu, setter dan getter untuk address juga ditambahkan, memungkinkan kita untuk mengubah dan mengambil nilai address.

Kesimpulan

- Dari tahap pembuatan class dasar, penambahan setter dan getter, hingga modifikasi class dengan atribut baru, proses ini menunjukkan bagaimana cara kita bisa menjaga prinsip enkapsulasi di dalam pemrograman berorientasi objek. Dengan menggunakan getter dan setter, kita dapat mengontrol akses dan perubahan atribut dalam class secara aman.
- Proses modifikasi dan debugging membantu kita mengembangkan program agar lebih kompleks dan fungsional, serta memastikan semua bagian kode berjalan dengan baik tanpa error.