

PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK
MODUL 4



Nama : NICKY JULYATRIKA SARI

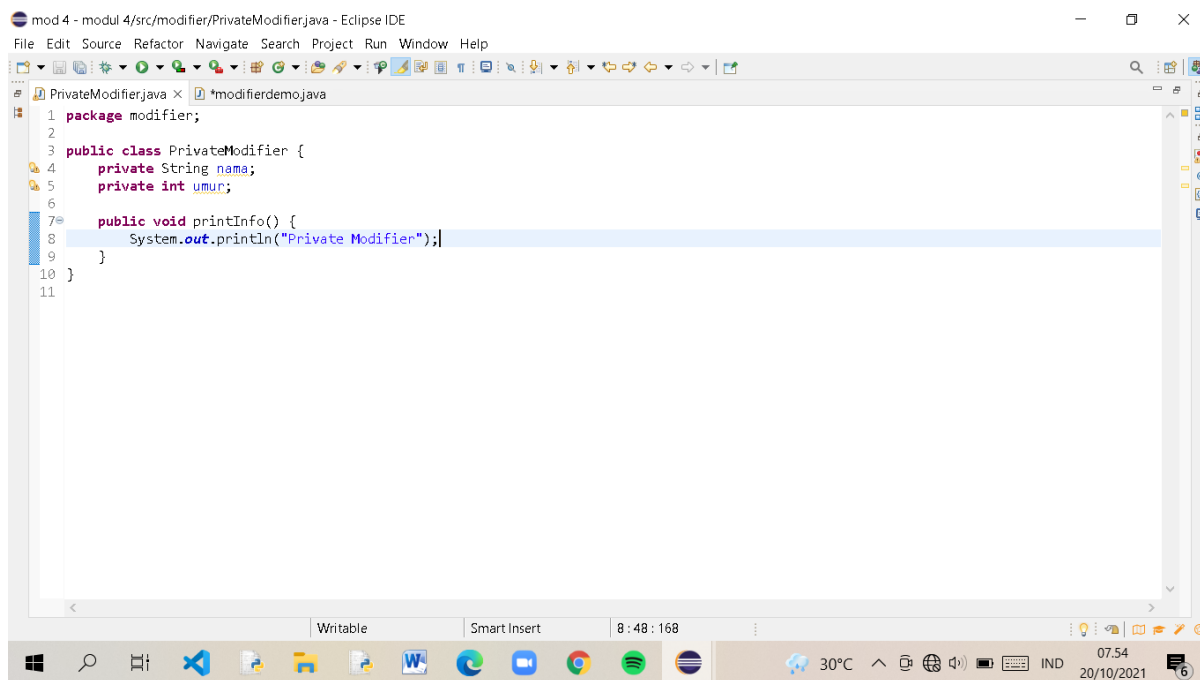
NIM : L200200101

PROGRAM STUDI
INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
TAHUN 2021/2022

1. Latihan 1(Private Modifier)

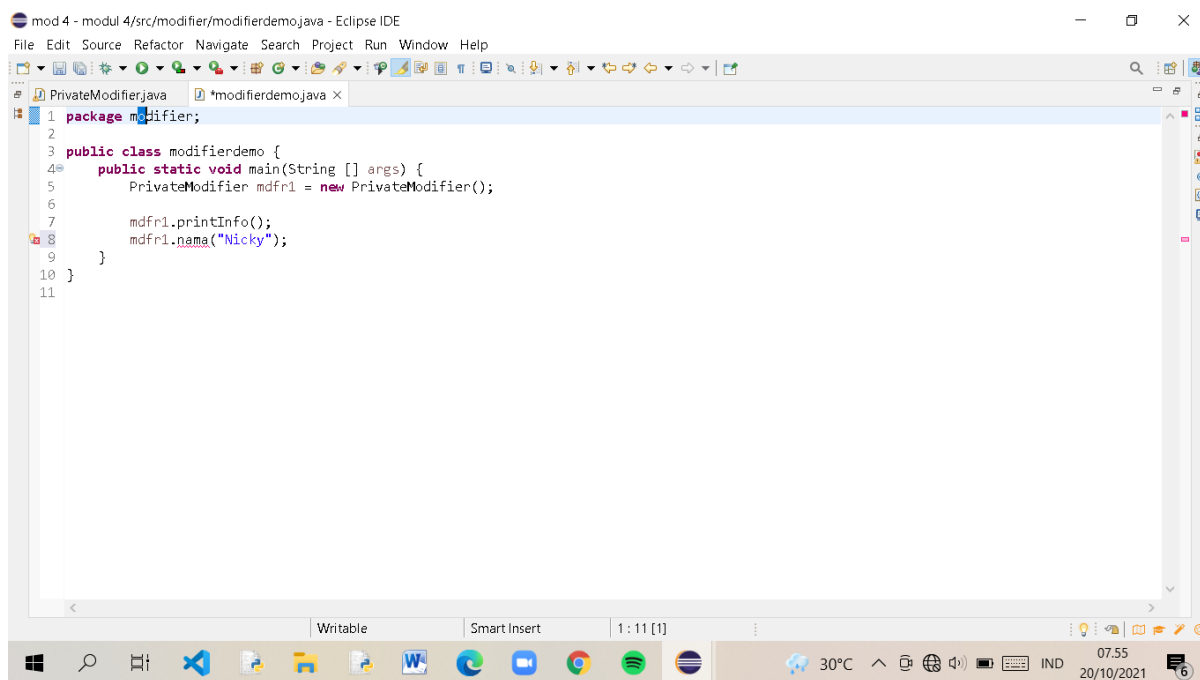
- a. Tuliskan kode di atas dan buatlah class baru untuk mengakses variable dari Program 1. di atas! Apakah variable tersebut dapat diakses dari class lain? Jelaskan!

i. Kode program



```
mod 4 - modul 4/src/modifier/PrivateModifier.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

1 package modifier;
2
3 public class PrivateModifier {
4     private String nama;
5     private int umur;
6
7     public void printInfo() {
8         System.out.println("Private Modifier");
9     }
10 }
11
```

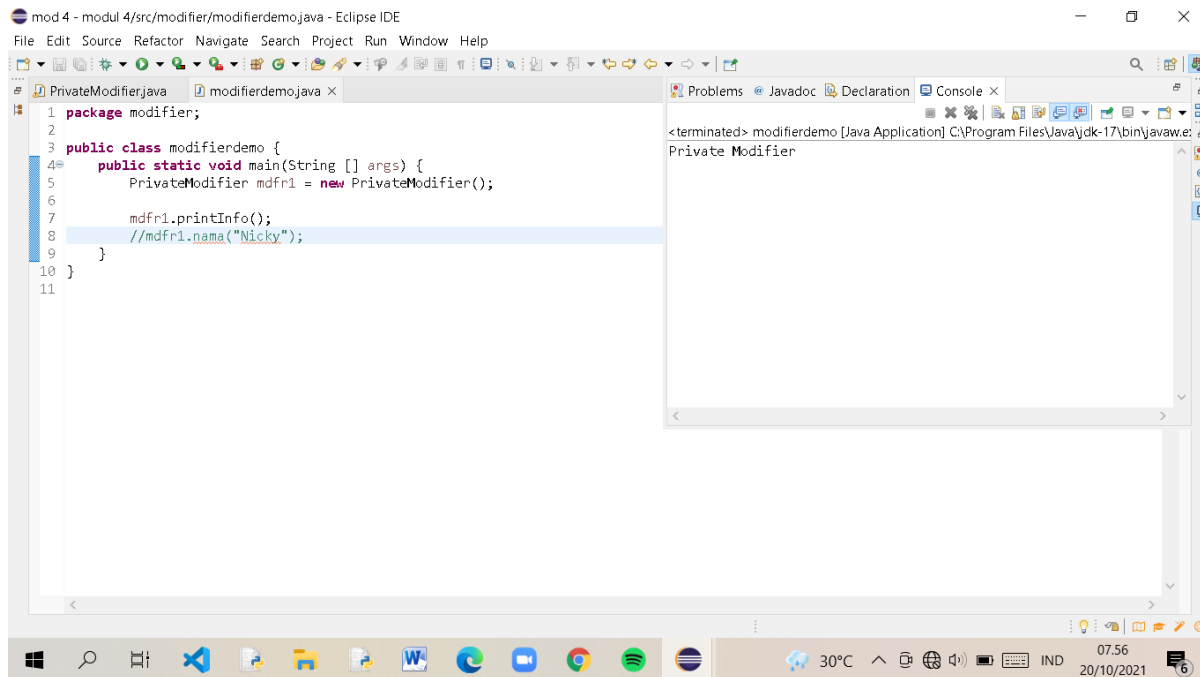


```
mod 4 - modul 4/src/modifier/modifierdemo.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help

1 package modifier;
2
3 public class modifierdemo {
4     public static void main(String [] args) {
5         PrivateModifier mdfn1 = new PrivateModifier();
6
7         mdfn1.printInfo();
8         mdfn1.nama("Nicky");
9     }
10 }
11
```

ii. Penjelasannya

Variabel dari kelas private modifier tidak dapat diakses oleh kelas lain, karena privat modifier variabel atau method hanya dapat diakses dari dalam class itu sendiri. sehingga kode diatas akan mengalami eror.



The screenshot shows the Eclipse IDE interface. The main editor window displays a Java file named `modifierndemo.java` with the following code:

```
1 package modifier;  
2  
3 public class modifierndemo {  
4     public static void main(String [] args) {  
5         PrivateModifier mdfri1 = new PrivateModifier();  
6  
7         mdfri1.printInfo();  
8         //mdfri1.nama("Nicky");  
9     }  
10 }  
11
```

The right-hand side of the IDE shows the `Problems` view, which displays the following error message:

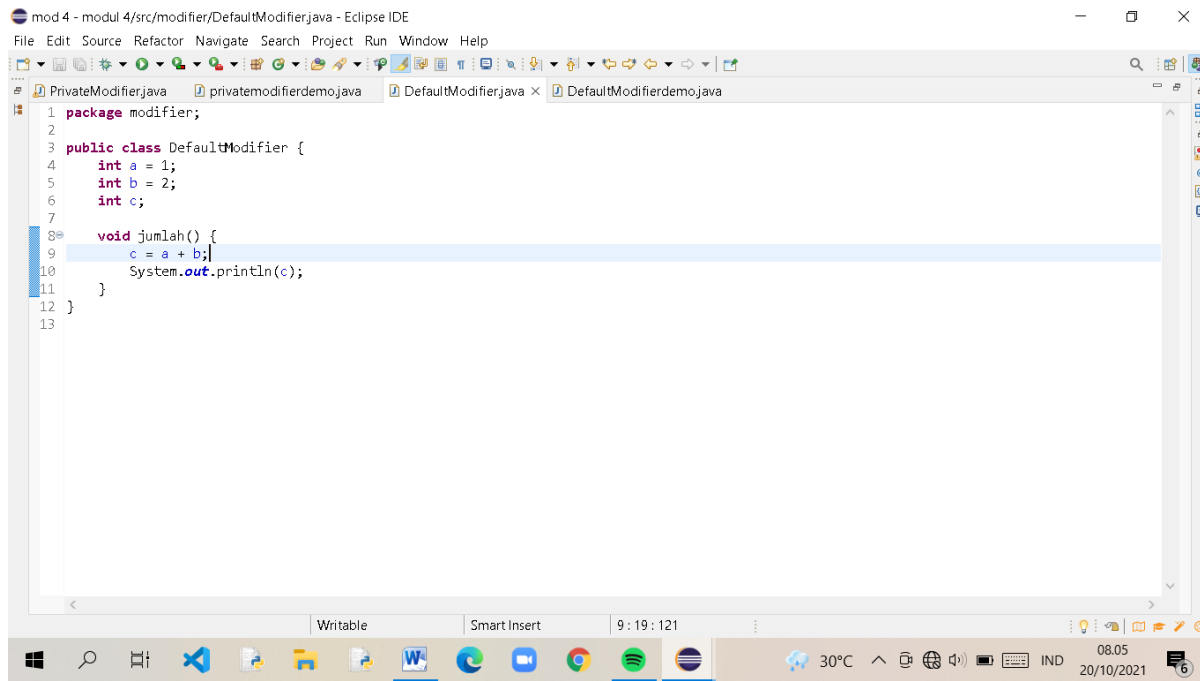
```
<terminated> modifierndemo [Java Application] C:\Program Files\Java\jdk-17\bin\javaw.exe  
Private Modifier
```

The error message indicates that the code is terminated and that there is a problem with the `Private Modifier`. The status bar at the bottom of the IDE shows the system temperature as 30°C, the time as 07:56, and the date as 20/10/2021.

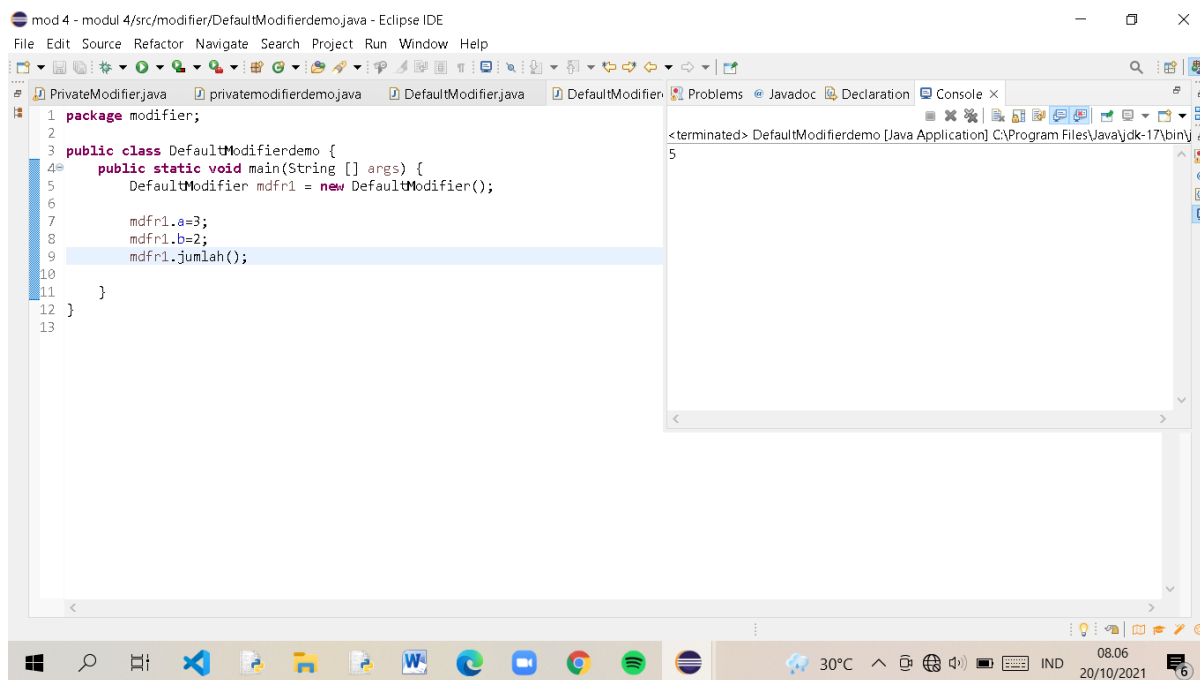
iii. Kode program diatas tidak eror karena variabel/methodnya di akses oleh kelas itu sendiri.

2. Latihan 2(Default Modifier)

- Untuk membuktikan sifat dari tipe akses default, maka buat satu class baru dan coba akses variabel dan method dari Program 2! Amati dan jelaskan hasil yang Saudara dapatkan!
- Buat package lain dan beri nama yang unik sebagai pembeda! Tambahkan satu class di dalamnya. Kemudian lakukan pengaksesan variable dan method Program 2. melalui class baru tersebut! Amati dan jelaskan hasil yang Saudara dapatkan!
 - Kode program



```
1 package modifier;
2
3 public class DefaultModifier {
4     int a = 1;
5     int b = 2;
6     int c;
7
8     void jumlah() {
9         c = a + b;
10        System.out.println(c);
11    }
12 }
13
```



```
1 package modifier;
2
3 public class DefaultModifierdemo {
4     public static void main(String [] args) {
5         DefaultModifier mdfn1 = new DefaultModifier();
6
7         mdfn1.a=3;
8         mdfn1.b=2;
9         mdfn1.jumlah();
10    }
11 }
12 }
13
```

<terminated> DefaultModifierdemo [Java Application] C:\Program Files\Java\jdk-17\bin\j5

ii. Penjelasan

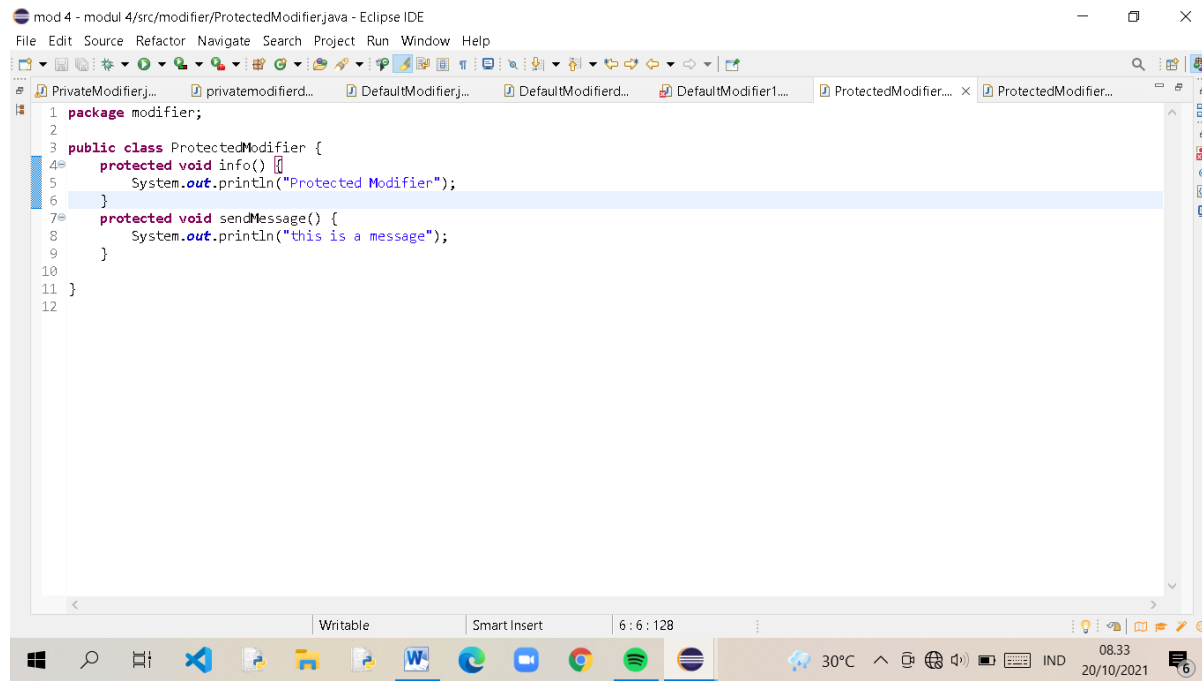
Dalam kode program diatas tidak terjadi eror karena Default modifier, atau tanpa penentu akses, yang berarti setiap variabel atau method dapat diakses oleh class yang berada dalam package yang sama.

```
1 package modifierbaru;
2
3 import modifier.DefaultModifier;
4
5 public class DefaultModifier1 {
6     public static void main(String [] args) {
7         DefaultModifier mdfr1 = new DefaultModifier();
8
9         mdfr1.jumlah();
10    }
11 }
12
```

- iii. Kode diatas adalah kode program didalam package yang berbeda dengan kode sebelumnya.kode program akan eror karena default modifier hanya bisa mengakses variabel/method dalam package yang sama.

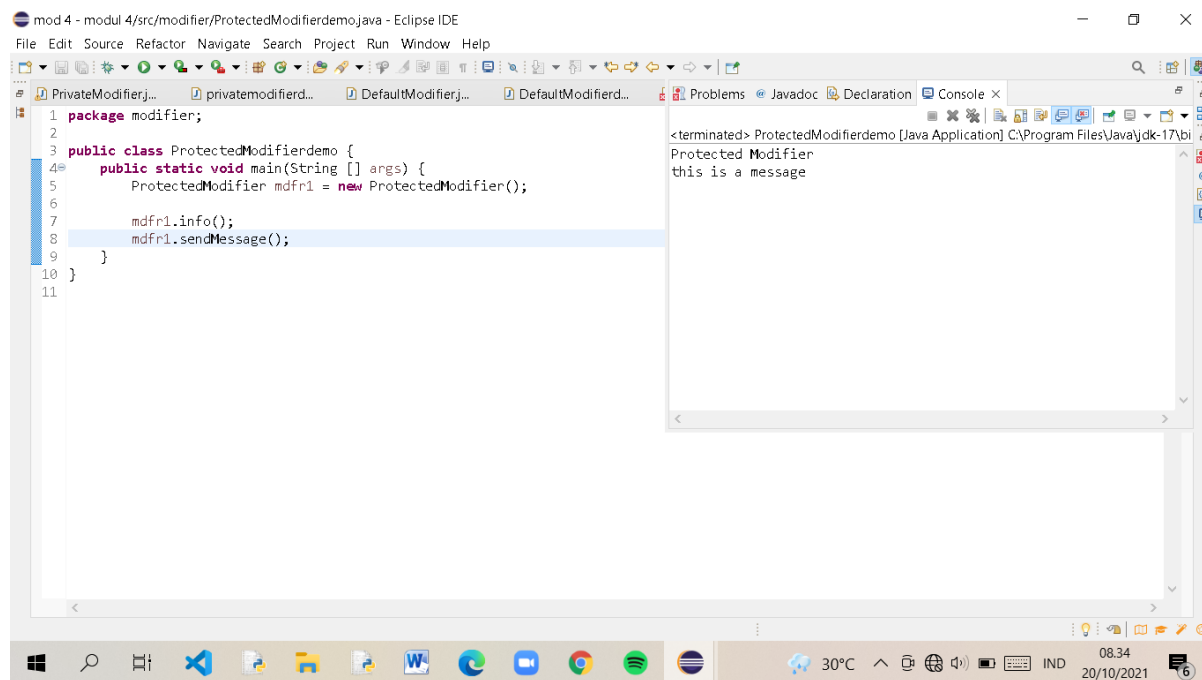
3. Latihan 3(Protected Modifier)

- a. Buatlah class baru dan object-nya untuk mengakses method dari Program 3., dan bandingkan dengan dua modifier sebelumnya!
 - i. Kode program



The screenshot shows the Eclipse IDE with a Java file named ProtectedModifier.java. The code defines a package 'modifier' and a public class 'ProtectedModifier'. The class has two methods: 'info()' and 'sendMessage()', both marked with the 'protected' access modifier. The 'info()' method prints 'Protected Modifier' and the 'sendMessage()' method prints 'this is a message'.

```
1 package modifier;
2
3 public class ProtectedModifier {
4     protected void info() {
5         System.out.println("Protected Modifier");
6     }
7     protected void sendMessage() {
8         System.out.println("this is a message");
9     }
10 }
11
12
```



The screenshot shows the Eclipse IDE with a Java file named ProtectedModifierno.java. The code defines a package 'modifier' and a public class 'ProtectedModifierno'. The class has a main method that creates an instance of 'ProtectedModifier' and calls its 'info()' and 'sendMessage()' methods. The console output shows the execution of the program, displaying 'Protected Modifier' and 'this is a message'.

```
1 package modifier;
2
3 public class ProtectedModifierno {
4     public static void main(String [] args) {
5         ProtectedModifier mdfn1 = new ProtectedModifier();
6
7         mdfn1.info();
8         mdfn1.sendMessage();
9     }
10 }
11
```

Console Output:

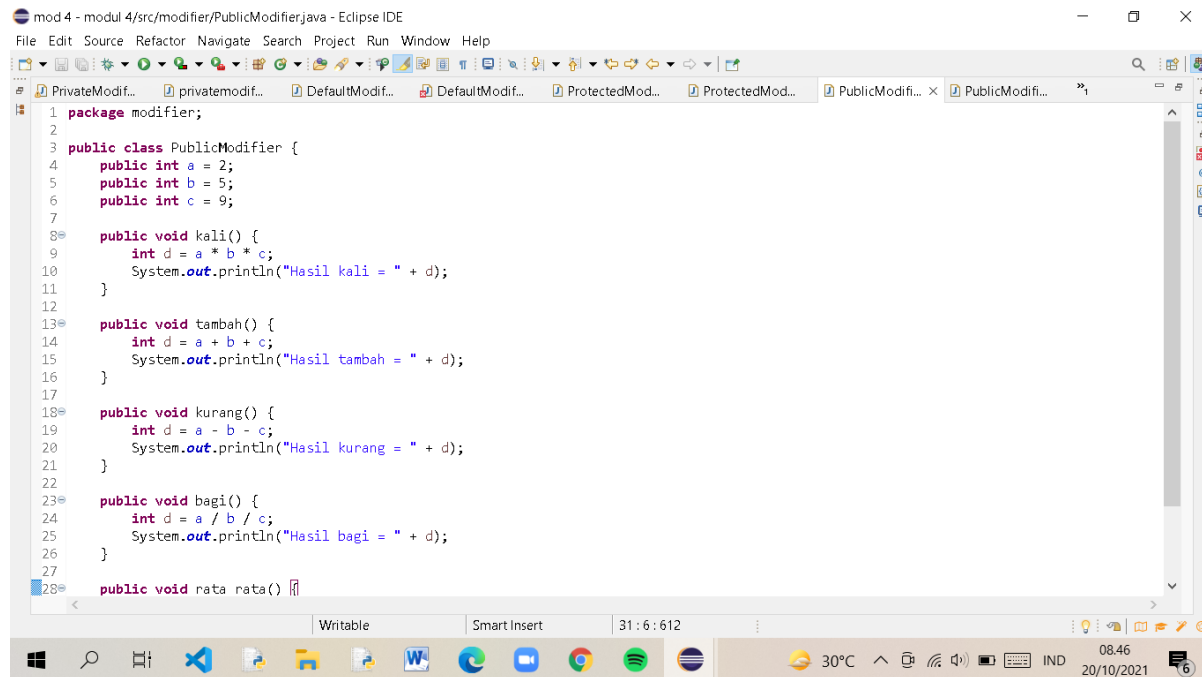
```
<terminated> ProtectedModifierno [Java Application] C:\Program Files\Java\jdk-17\bin
Protected Modifier
this is a message
```

ii. Penjelasan

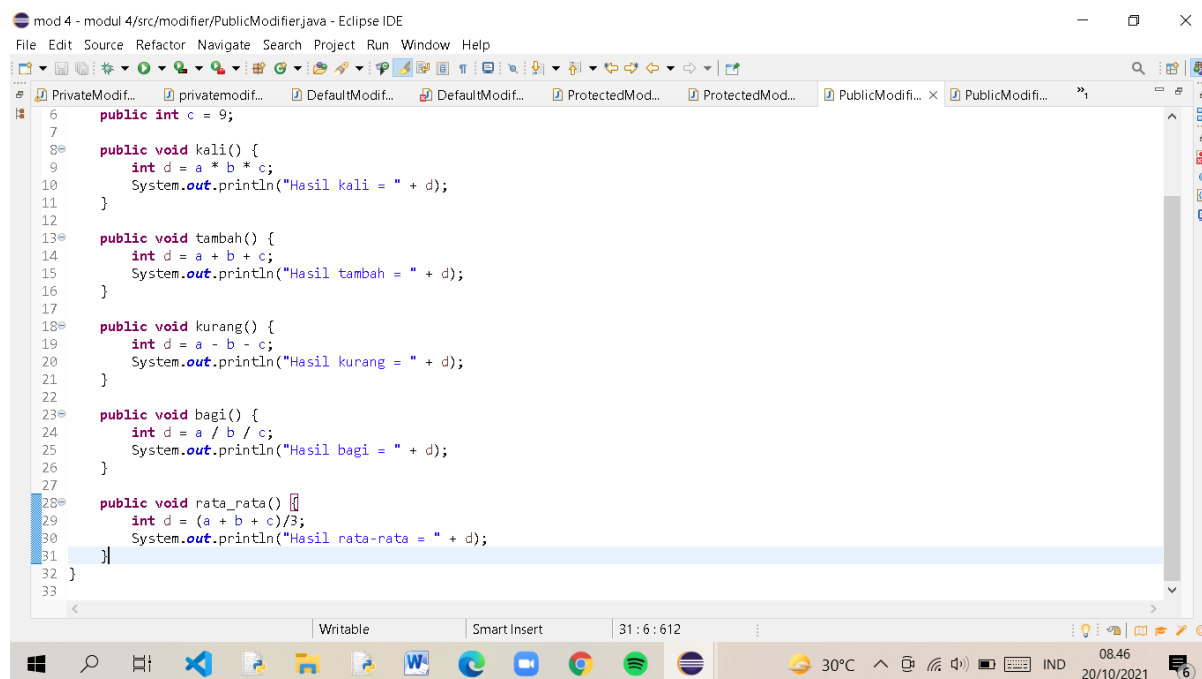
Pada kode program diatas tidak terjadi eror karena Protected modifier, digunakan untuk membuat class tersebut dapat diakses hanya oleh class lain dan turunan dalam satu package.

4. Latihan 4(Public Modifier)

- a. Lakukan percobaan seperti pada Latihan 4.2.1. dan bandingkan hasilnya!
- b. Tambahkan method baru diantaranya tambah(), kurang(), bagi(), dan rata_rata().
 - i. Kode program



```
1 package modifier;
2
3 public class PublicModifier {
4     public int a = 2;
5     public int b = 5;
6     public int c = 9;
7
8     public void kali() {
9         int d = a * b * c;
10        System.out.println("Hasil kali = " + d);
11    }
12
13    public void tambah() {
14        int d = a + b + c;
15        System.out.println("Hasil tambah = " + d);
16    }
17
18    public void kurang() {
19        int d = a - b - c;
20        System.out.println("Hasil kurang = " + d);
21    }
22
23    public void bagi() {
24        int d = a / b / c;
25        System.out.println("Hasil bagi = " + d);
26    }
27
28    public void rata_rata() {
```



```
28    public void rata_rata() {
29        int d = (a + b + c)/3;
30        System.out.println("Hasil rata-rata = " + d);
31    }
32 }
33
```

```
1 package modifier;
2
3 public class PublicModfierdemo {
4     public static void main(String [] args) {
5         PublicModifier mdfnr1 = new PublicModifier();
6
7         mdfnr1.kali();
8         mdfnr1.tambah();
9         mdfnr1.kurang();
10        mdfnr1.bagi();
11        mdfnr1.rata_rata();
12    }
13 }
14
```

<terminated> PublicModfierdemo [Java Application] C:\Program Files\Java\jdk-17\bin\java.exe
Hasil kali = 90
Hasil tambah = 16
Hasil kurang = -12
Hasil bagi = 0
Hasil rata-rata = 5

ii. Kode program di package yang berbeda

```
1 package modifierbaru;
2
3 import modifier.PublicModifier;
4
5 public class PublicModifier1 {
6     public static void main(String [] args) {
7         PublicModifier mdfnr1 = new PublicModifier();
8
9         mdfnr1.kali();
10        mdfnr1.tambah();
11        mdfnr1.kurang();
12        mdfnr1.bagi();
13        mdfnr1.rata_rata();
14    }
15 }
16
```

<terminated> PublicModifier1 [Java Application] C:\Program Files\Java\jdk-17\bin\java.exe
Hasil kali = 90
Hasil tambah = 16
Hasil kurang = -12
Hasil bagi = 0
Hasil rata-rata = 5

iii. Penjelasan

Kode program diatas akan berjalan dan tidak terjadi eror,karena Public modifier, variabel atau methodnya dapat diakses pada semua class yang menggunakan class yang berisi variabel tersebut meskipun berada pada package yang berbeda.

