

**LAPORAN PRAKTIKUM
MODUL 4
ACCESS MODIFIER
PRAKTIKUM PEMROGRAMAN
BERORIENTASI OBJEK**

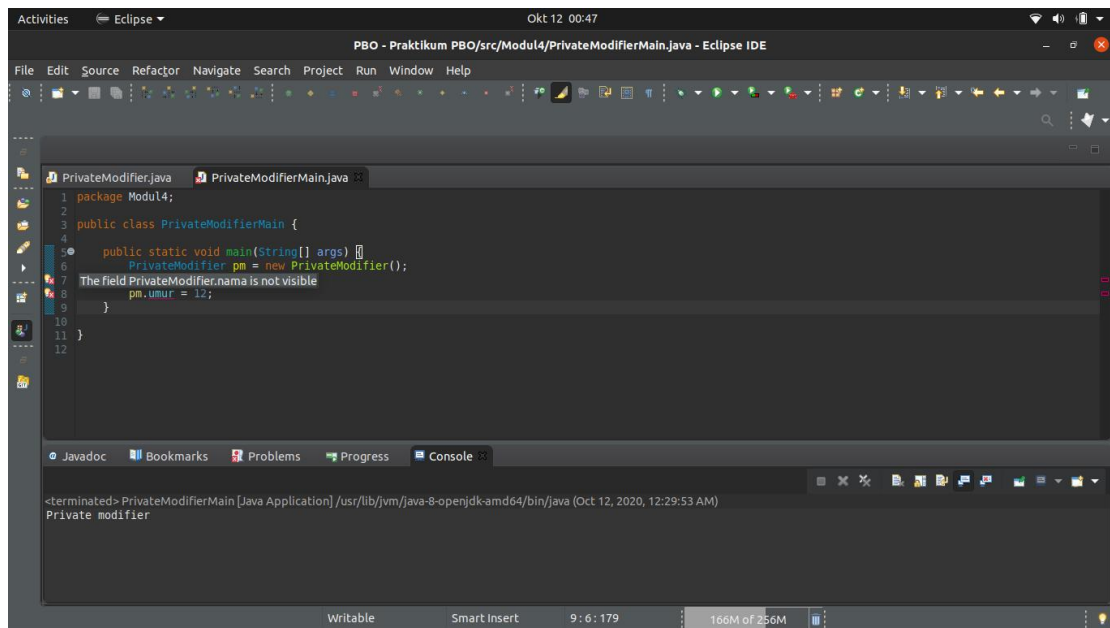


**Senopati Bkti W
L200190143
Praktikum PBO D**

INFORMATIKA
FAKULTAS KOMUNIKASI DAN
INFORMATIKA
UNIVERSITAS MUHAMMADIYAH
SURAKARTA

2020

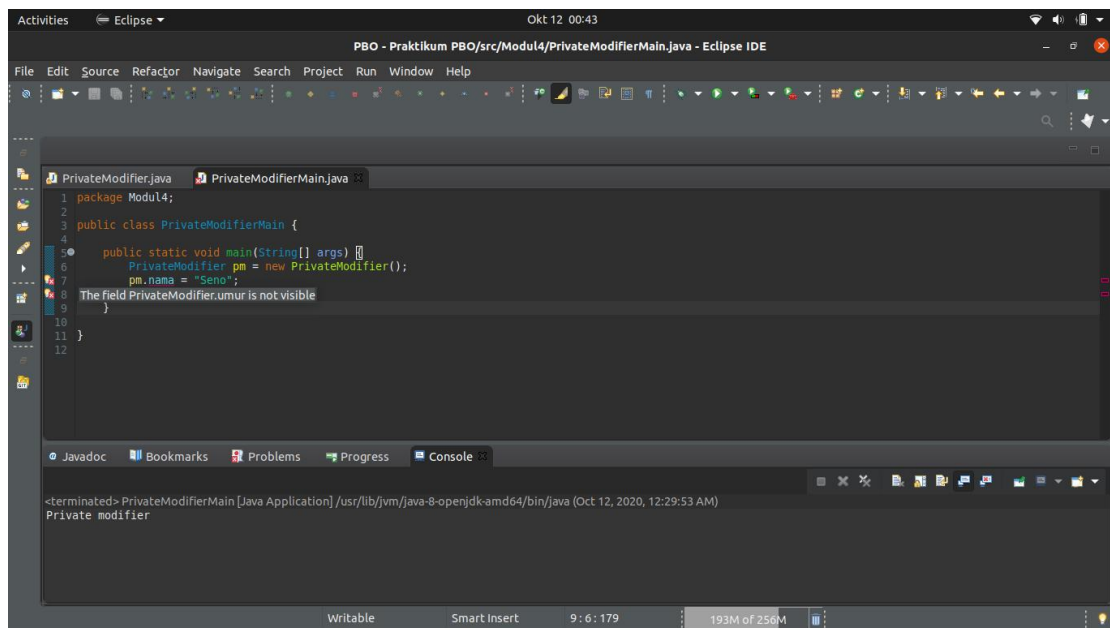
Latihan 1



The screenshot shows the Eclipse IDE interface. The main editor displays the file `PrivateModifierMain.java` with the following code:

```
1 package Modul4;
2
3 public class PrivateModifierMain {
4
5     public static void main(String[] args) {
6         PrivateModifier pm = new PrivateModifier();
7         pm.umur = 12;
8     }
9 }
10
11
12
```

A red squiggly line under `pm.umur` on line 7 indicates a compilation error. The error message in the `Problems` view is: "The field PrivateModifier.nama is not visible". The `Console` view shows the output: "<terminated> PrivateModifierMain [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (Oct 12, 2020, 12:29:53 AM) Private modifier".



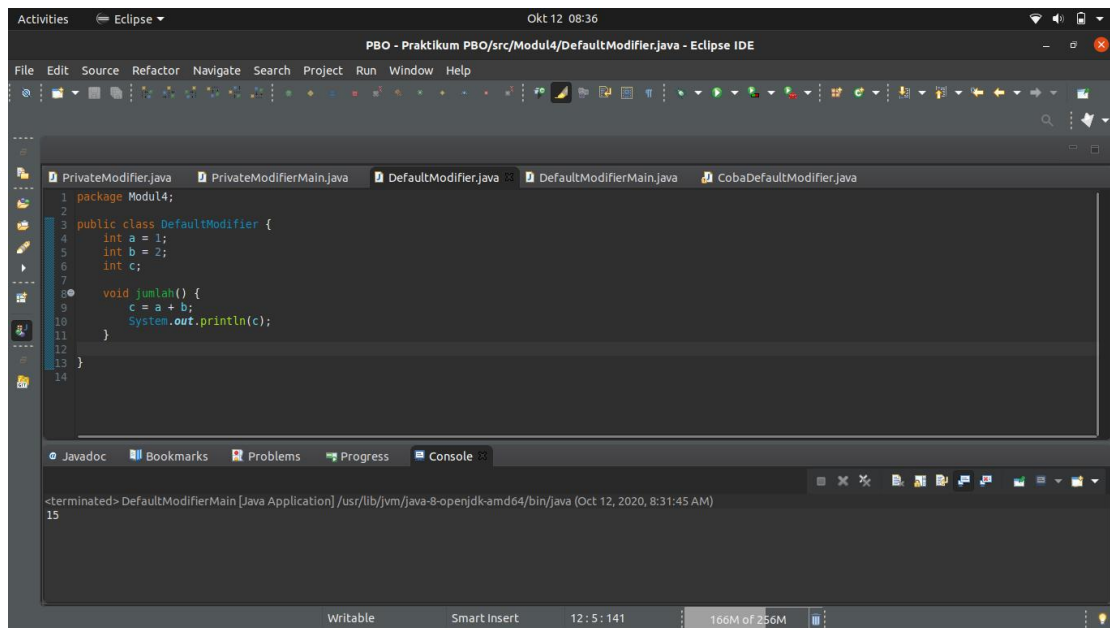
The screenshot shows the Eclipse IDE interface. The main editor displays the file `PrivateModifierMain.java` with the following code:

```
1 package Modul4;
2
3 public class PrivateModifierMain {
4
5     public static void main(String[] args) {
6         PrivateModifier pm = new PrivateModifier();
7         pm.nama = "Seno";
8     }
9 }
10
11
12
```

A red squiggly line under `pm.nama` on line 7 indicates a compilation error. The error message in the `Problems` view is: "The field PrivateModifier.umur is not visible". The `Console` view shows the output: "<terminated> PrivateModifierMain [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (Oct 12, 2020, 12:29:53 AM) Private modifier".

Variable pada class “PrivateModifier.java” mempunyai variable nama dan umur akan tetapi keduanya tidak bisa diakses pada kelas lain karena kedua variable tersebut mempunyai **akses modifier private yang dimana variable tersebut hanya bisa diakses dalam class tempat ia dideklarasikan.**

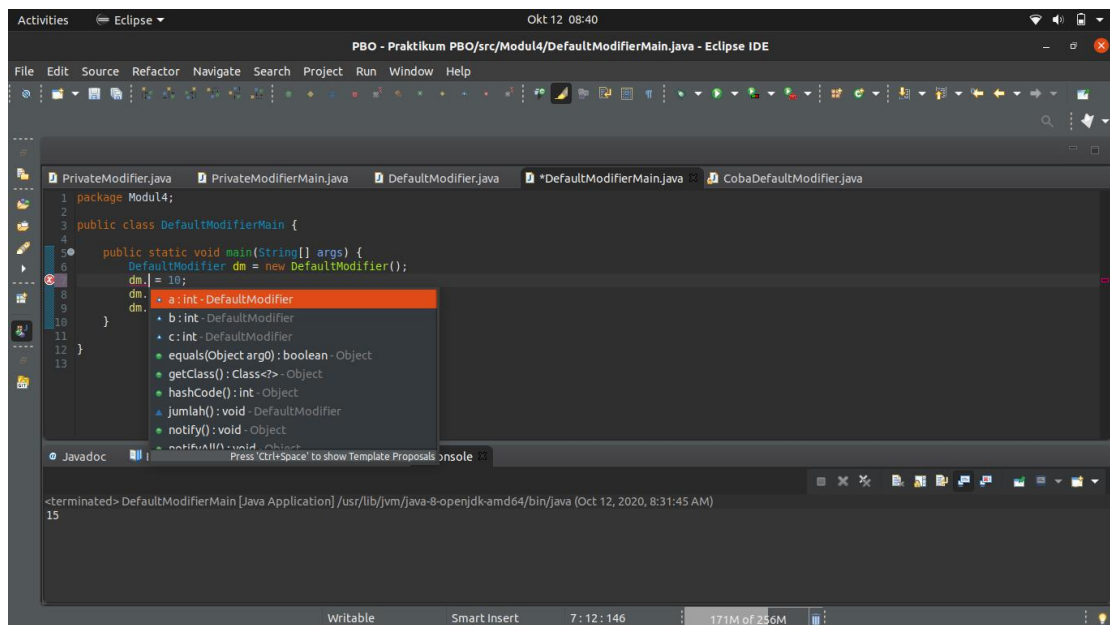
Latihan 2



The screenshot shows the Eclipse IDE interface. The top toolbar includes icons for File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The main editor window displays the code for `DefaultModifier.java` in the `Modul4` package. The code defines a public class `DefaultModifier` with three integer variables (`a`, `b`, `c`) and a `jumlah()` method that calculates the sum of `a` and `b` and prints it. The bottom console shows the output of the program, which is the number 15.

```
1 package Modul4;
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 }
14
```

Console output: <terminated> DefaultModifierMain [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (Oct 12, 2020, 8:31:45 AM) 15

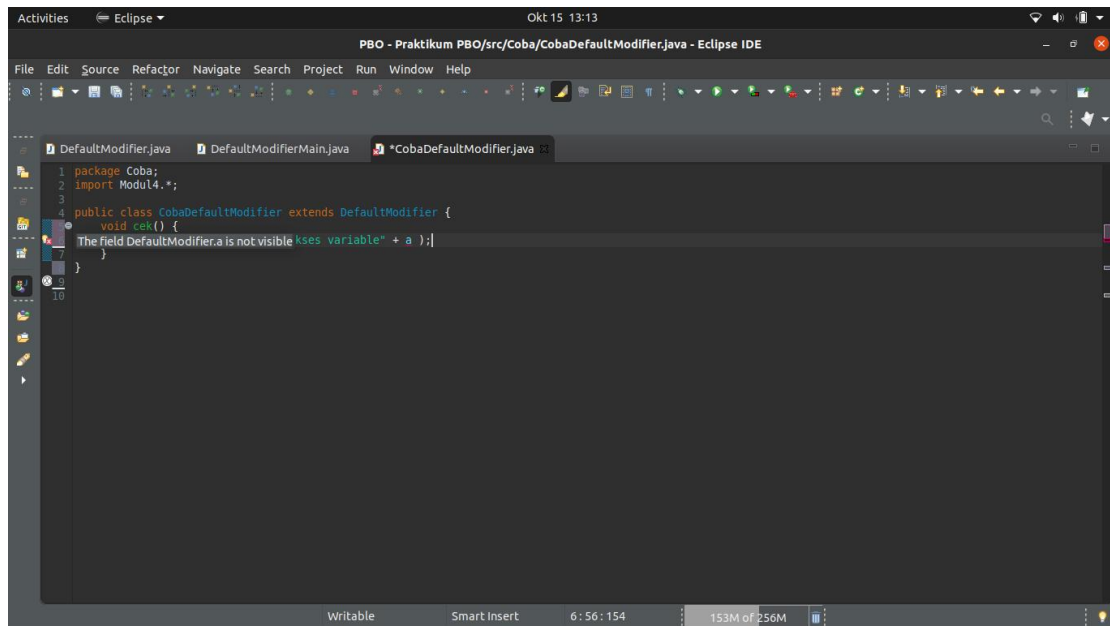


The screenshot shows the Eclipse IDE interface. The top toolbar includes icons for File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The main editor window displays the code for `DefaultModifierMain.java` in the `Modul4` package. The code defines a public class `DefaultModifierMain` with a `main` method that creates an instance of `DefaultModifier` and calls its `jumlah` method. The bottom console shows the output of the program, which is the number 15.

```
1 package Modul4;
2
3 public class DefaultModifierMain {
4
5     public static void main(String[] args) {
6         DefaultModifier dm = new DefaultModifier();
7         dm.a = 10;
8         dm.b = 10;
9         dm.jumlah();
10    }
11
12 }
13
```

Console output: <terminated> DefaultModifierMain [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (Oct 12, 2020, 8:31:45 AM) 15

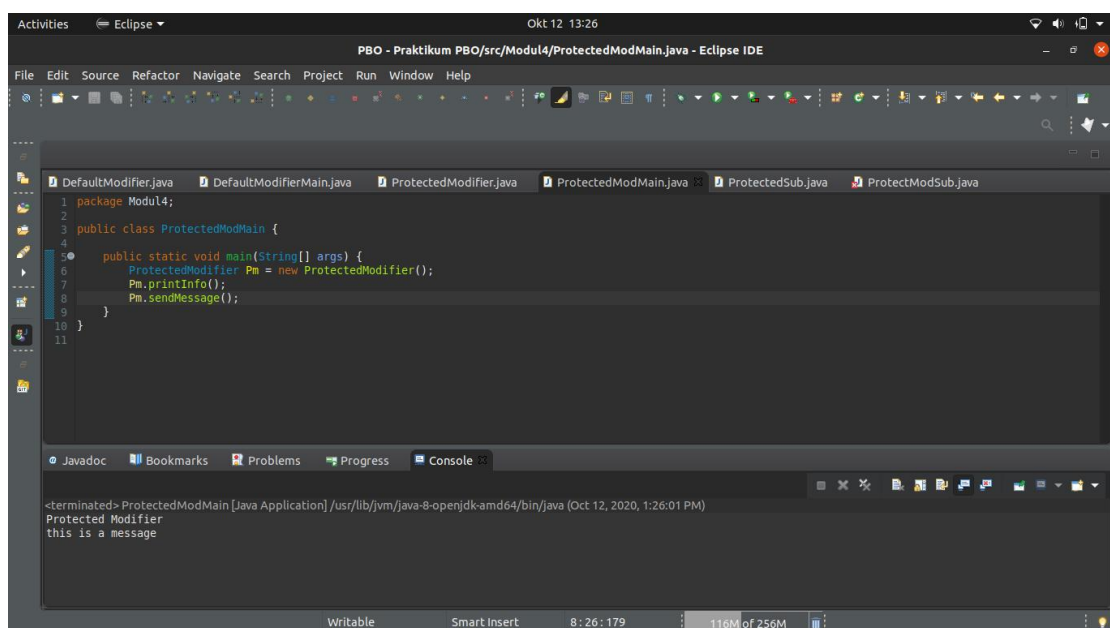
Default modifier (Variable dan method) dapat diakses pada class lain dalam satu package. Dapat dibuktikan pada main class yang berada dalam satu package dapat mengakses Variable dan Method yang dimiliki class “DefaultModifier.java”



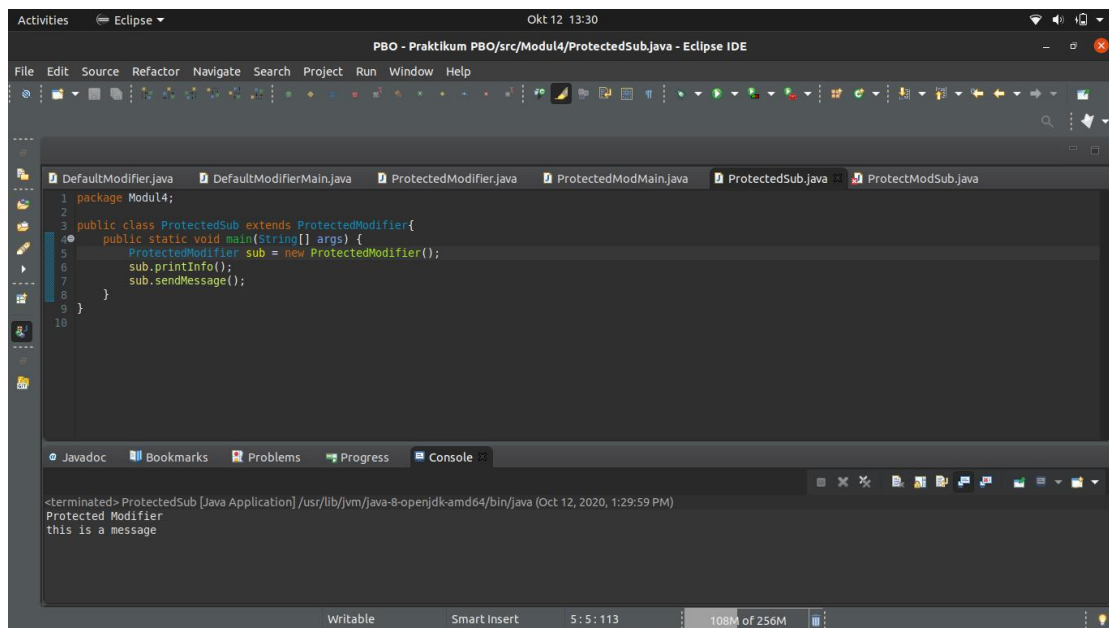
Membuat class “*CobaDefaultModifier.java*” pada package bernama “Coba” (Diluar package Modul4). Variable a , b dan method jumlah() tidak bisa diakses melalui class “*CobaDefaultModifier.java*” karena Variable a , b dan method jumlah() merupakan **Default Modifier** artinya hanya dapat diakses pada class lain akan tetapi lokasinya masih dalam satu package.

Latihan 3

Dieksekusi pada Main Class “ProtectedModMain.java” berada dalam satu package yaitu Modul4



Pada kondisi ini , dieksekusi oleh Subclass (ProtectedSub.java) yang berada satu package dengannya dari Parent Class (ProtectedModifier.java)



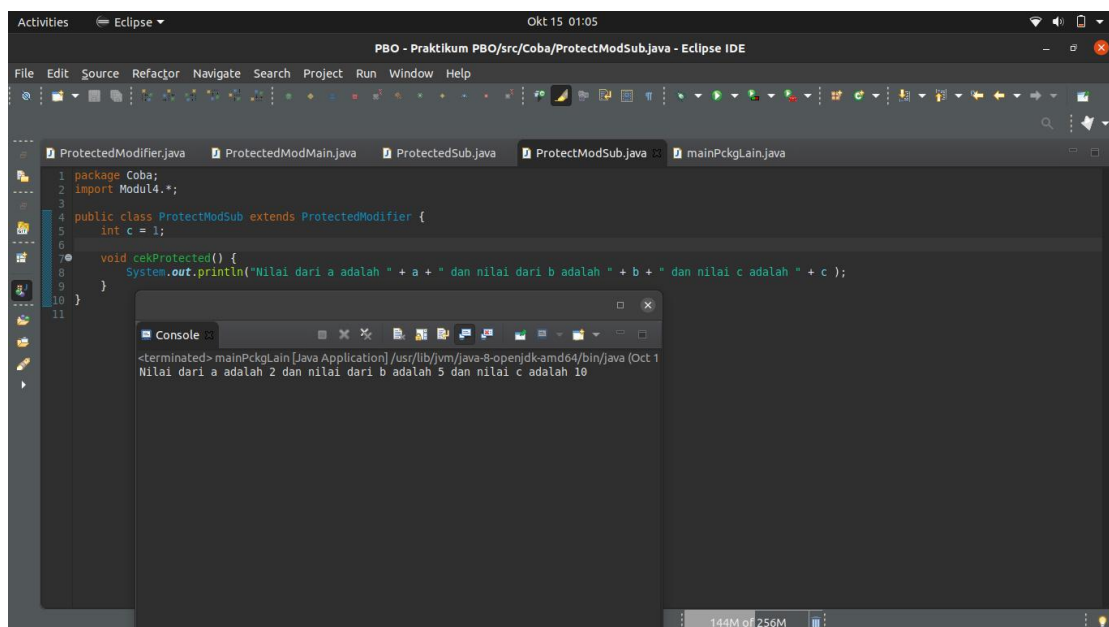
The screenshot shows the Eclipse IDE with the file `ProtectedSub.java` open. The code defines a package `Modul4` and a class `ProtectedSub` that extends `ProtectedModifier`. The `main` method creates an instance of `ProtectedModifier` and calls `sub.printInfo()` and `sub.sendMessage()`. The console output shows the execution of `ProtectedSub` printing "Protected Modifier" and "this is a message".

```
package Modul4;

public class ProtectedSub extends ProtectedModifier{
    public static void main(String[] args) {
        ProtectedModifier sub = new ProtectedModifier();
        sub.printInfo();
        sub.sendMessage();
    }
}
```

```
<terminated> ProtectedSub [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (Oct 12, 2020, 1:29:59 PM)
Protected Modifier
this is a message
```

Variable a dan b dapat Diakses oleh class lain yang berada **diluar package “Modul4”**. Dibawah ini contoh bahwa private modifier bisa diakses pada subclass yang berada diluar package



The screenshot shows the Eclipse IDE with the file `ProtectModSub.java` open. The code defines a package `Coba` and a class `ProtectModSub` that extends `ProtectedModifier`. The `cekProtected` method prints the values of variables `a`, `b`, and `c`. The console output shows the execution of `mainPckgLain` printing "Nilai dari a adalah 2 dan nilai dari b adalah 5 dan nilai c adalah 10".

```
package Coba;
import Modul4.*;

public class ProtectModSub extends ProtectedModifier {
    int c = 1;

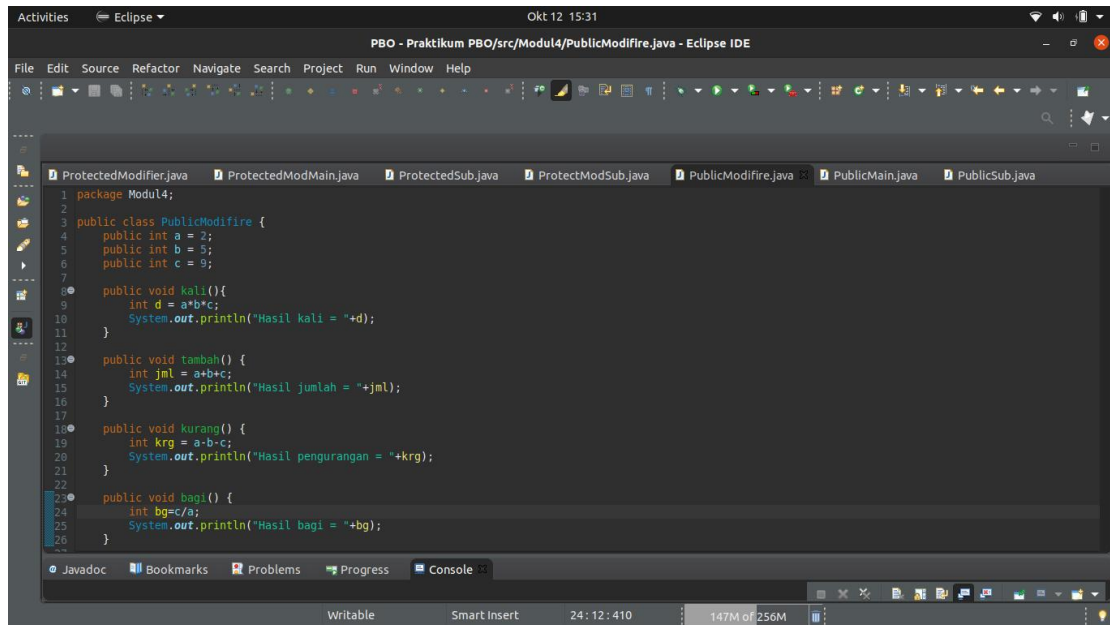
    void cekProtected() {
        System.out.println("Nilai dari a adalah " + a + " dan nilai dari b adalah " + b + " dan nilai c adalah " + c );
    }
}
```

```
<terminated> mainPckgLain [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (Oct 1
Nilai dari a adalah 2 dan nilai dari b adalah 5 dan nilai c adalah 10
```

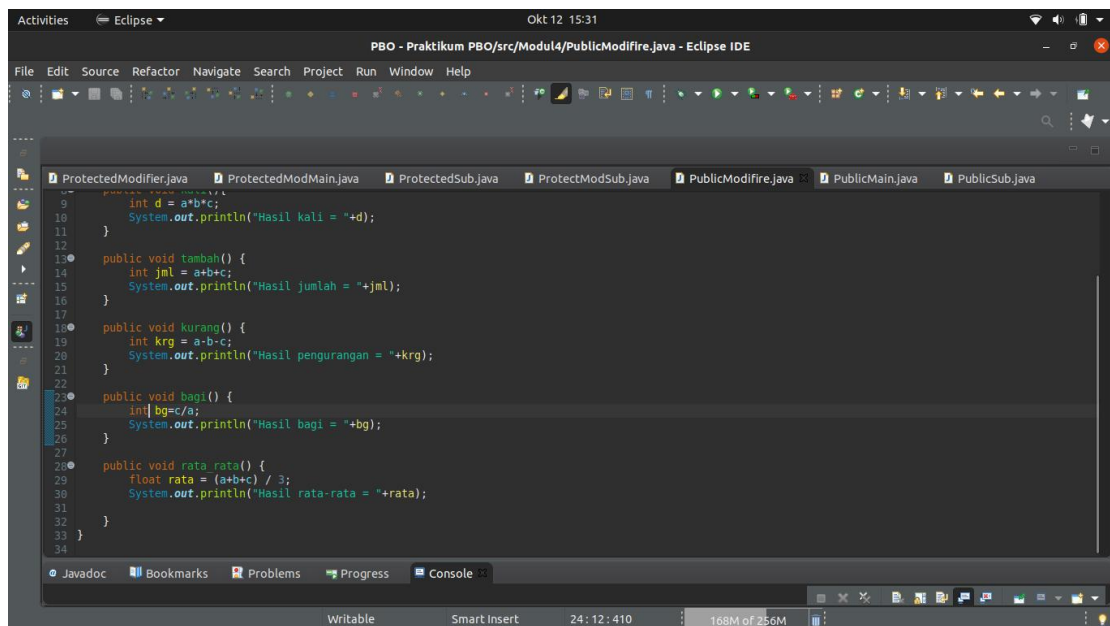
Dapat disimpulkan Modifier “protected” hanya bisa diakses pada Class maupun Subclass yang berada dalam satu package dengannya atau Subclass yang berada diluar packagenya

Latihan 4

Membuat class bernama PublicModifier.java sebagai cetakan (blueprint) dan ditambahkan method baru diantaranya tambah(), kurang(), bagi(), dan rata_rata()

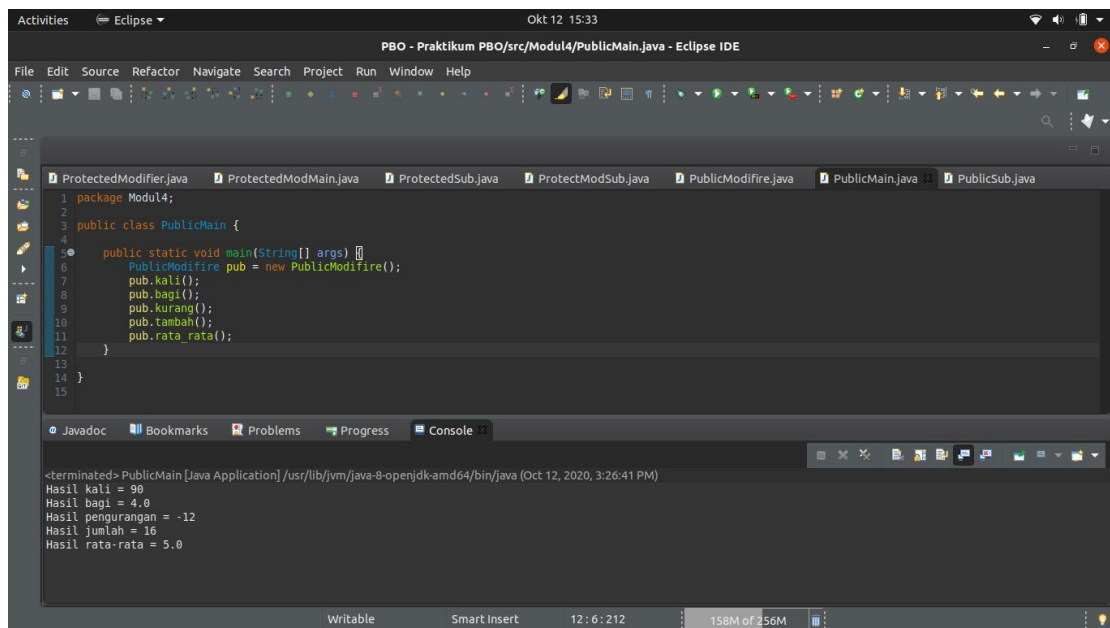


```
1 package Modul4;
2
3 public class PublicModifire {
4     public int a = 2;
5     public int b = 5;
6     public int c = 9;
7
8     public void kali(){
9         int d = a*b*c;
10        System.out.println("Hasil kali = "+d);
11    }
12
13    public void tambah() {
14        int jml = a+b+c;
15        System.out.println("Hasil jumlah = "+jml);
16    }
17
18    public void kurang() {
19        int krg = a-b-c;
20        System.out.println("Hasil pengurangan = "+krg);
21    }
22
23    public void bagi() {
24        int bg=c/a;
25        System.out.println("Hasil bagi = "+bg);
26    }
27 }
```



```
9         int d = a*b*c;
10        System.out.println("Hasil kali = "+d);
11    }
12
13    public void tambah() {
14        int jml = a+b+c;
15        System.out.println("Hasil jumlah = "+jml);
16    }
17
18    public void kurang() {
19        int krg = a-b-c;
20        System.out.println("Hasil pengurangan = "+krg);
21    }
22
23    public void bagi() {
24        int bg=c/a;
25        System.out.println("Hasil bagi = "+bg);
26    }
27
28    public void rata_rata() {
29        float rata = (a+b+c) / 3;
30        System.out.println("Hasil rata-rata = "+rata);
31    }
32
33 }
```

Dibuat main classnya bernama “PublicMain.java” sebagai tempat eksekusi perintah.



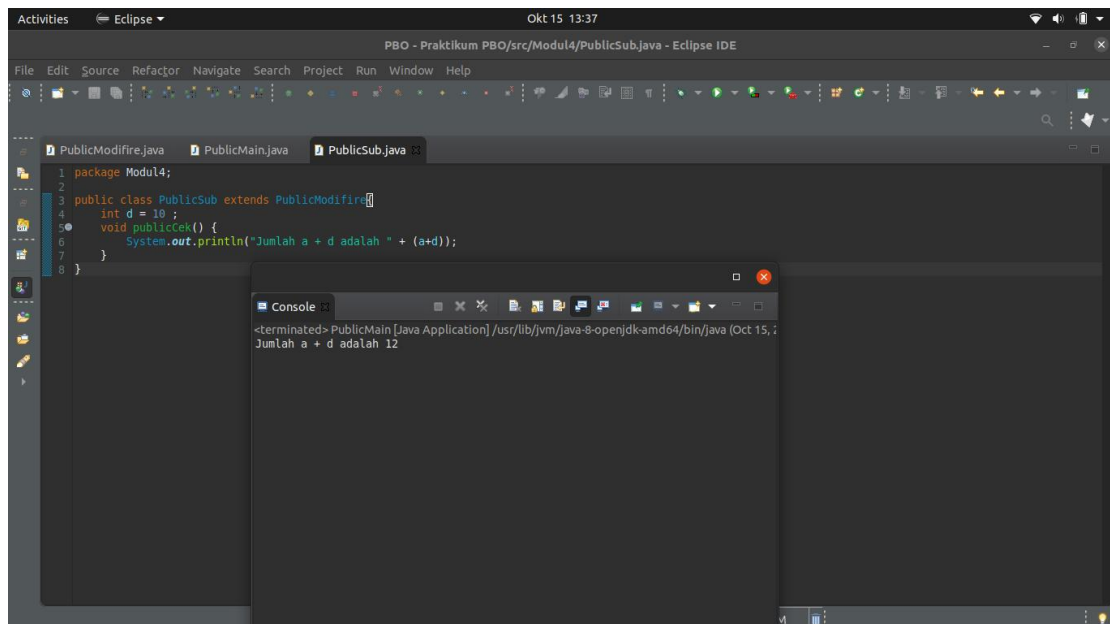
```
package Modul4;

public class PublicMain {

    public static void main(String[] args) {
        PublicModifire pub = new PublicModifire();
        pub.kali();
        pub.bagi();
        pub.kurang();
        pub.tambah();
        pub.rata_rata();
    }
}
```

```
<terminated> PublicMain [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (Oct 12, 2020, 3:26:41 PM)
Hasil kali = 90
Hasil bagi = 4.0
Hasil pengurangan = -12
Hasil jumlah = 16
Hasil rata-rata = 5.0
```

Dibuat SubClassnya bernama “PublicSub.java” dan dapat dieksekusi/diakses melalui subclassnya

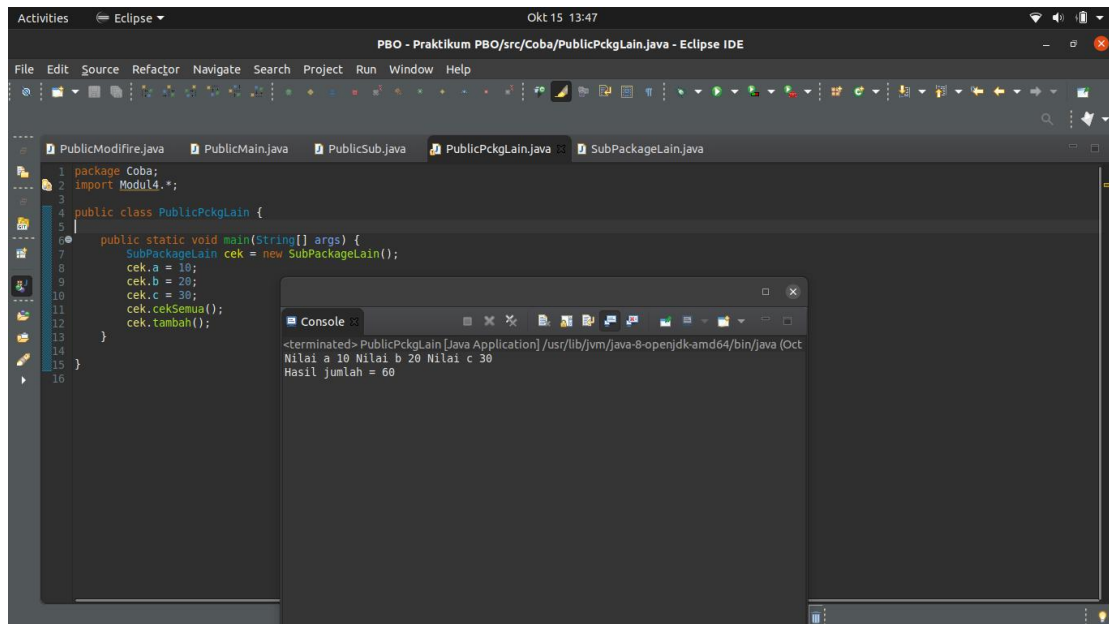


```
package Modul4;

public class PublicSub extends PublicModifire {
    int d = 10;
    void publicCek() {
        System.out.println("Jumlah a + d adalah " + (a+d));
    }
}
```

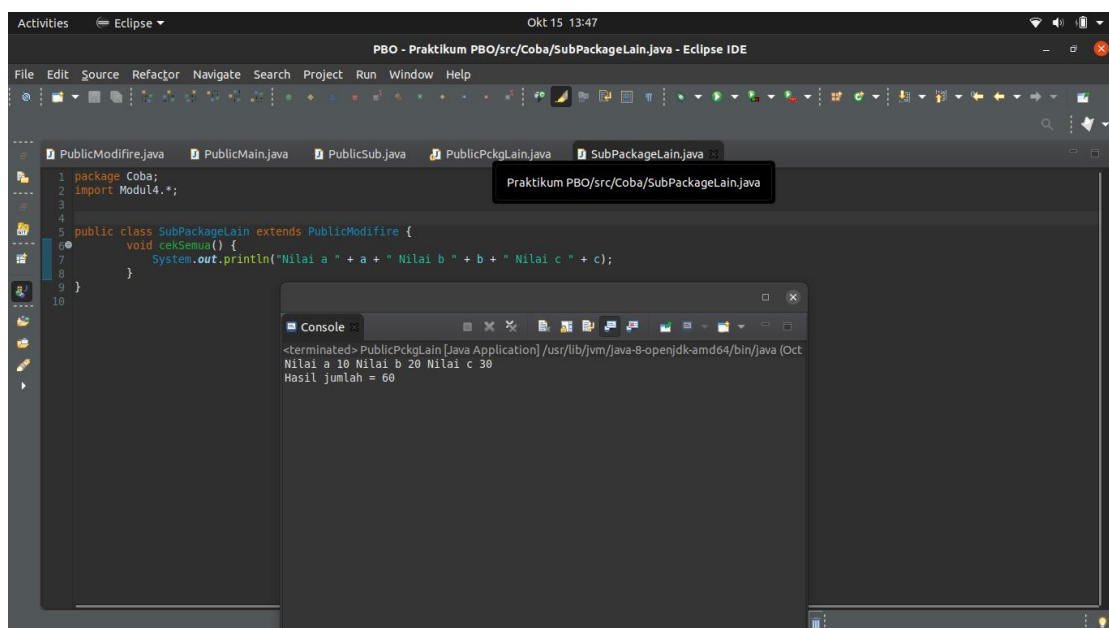
```
<terminated> PublicMain [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (Oct 15, 2020, 13:37:12)
Jumlah a + d adalah 12
```


Dibuat MainClass “PublicPckgLain.java” dan subclass “SubPackageLain.java” pada package Coba (diluar Package Modul4) dan dapat dilihat Properti (variable dan method) pada Class “PublicModifier.java” dapat diakses / dieksekusi .



```
1 package Coba;
2 import Modul4.*;
3
4 public class PublicPckgLain {
5
6     public static void main(String[] args) {
7         SubPackageLain cek = new SubPackageLain();
8         cek.a = 10;
9         cek.b = 20;
10        cek.c = 30;
11        cek.cekSemua();
12        cek.tambah();
13    }
14
15 }
16
```

```
<terminated> PublicPckgLain [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (Oct
Nilai a 10 Nilai b 20 Nilai c 30
Hasil jumlah = 60
```



```
1 package Coba;
2 import Modul4.*;
3
4
5 public class SubPackageLain extends PublicModifier {
6
7     void cekSemua() {
8         System.out.println("Nilai a " + a + " Nilai b " + b + " Nilai c " + c);
9     }
10 }

```

```
<terminated> PublicPckgLain [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (Oct
Nilai a 10 Nilai b 20 Nilai c 30
Hasil jumlah = 60
```

Dapat disimpulkan dari percobaan diatas , **Modifier Acces public dapat diartikan variabel atau method dapat diakses pada semua class yang menggunakan variabel atau methodnya meskipun berada pada package yang berbeda.**

KESIMPULAN DARI ACCESS MODIFIER

Access Modifier				
Modifier	Class	Package	Subclass	World
public	✓	✓	✓	✓
protected	✓	✓	✓	-
<i>no modifier</i>	✓	✓	-	-
private	✓	-	-	-