

**MODUL PRAKTIKUM**  
**PEMROGRAMAN BERORIENTASI OBYEK**



**UIN SUNAN AMPEL**  
**S U R A B A Y A**

**DOSEN :**

**Dwi Rolliawati**

**DISUSUN OLEH :**

**M.Khotibul Umam (09020620031)**

**UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA**

**FAKULTAS SAINT DAN TEKNOLOGI**

**PRODI SISTEM INFORMASI**

**2020/2021**

## Daftar Isi

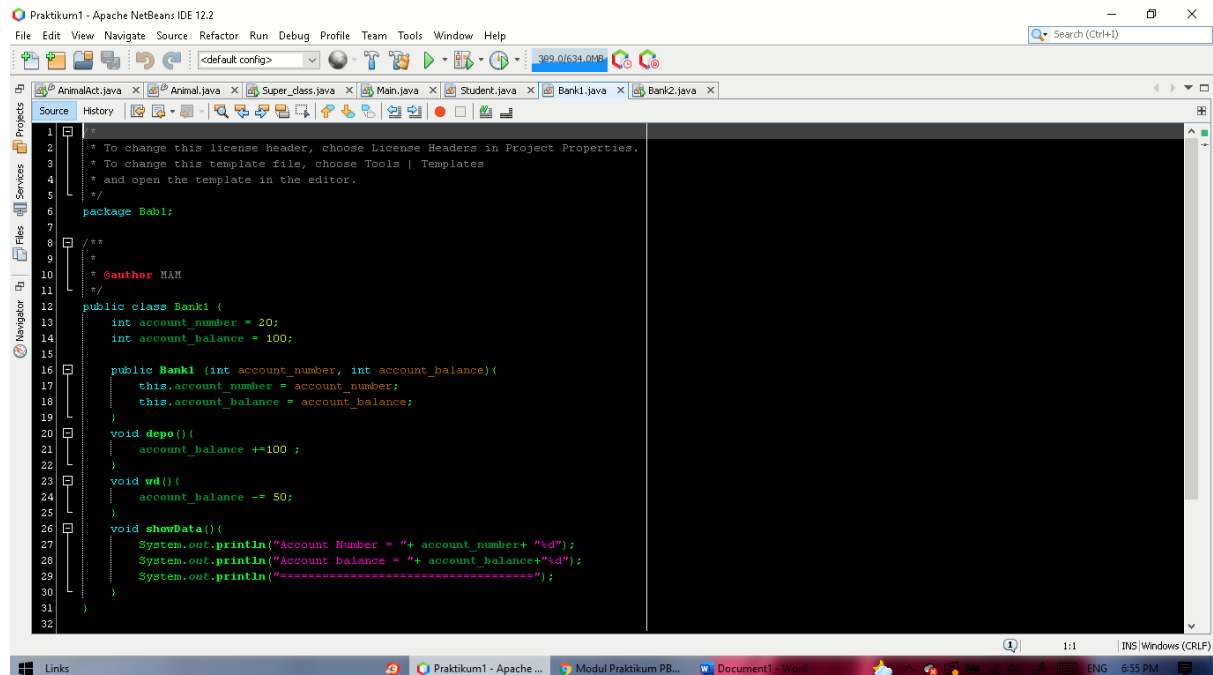
BAB 1 KONSEP PEMROGRAMAN BERORIENTASI OBYEK .....	3
Latihan 1 .....	3
Latihan 2 .....	3
Latihan 3 .....	4
BAB 2 CLASS DAN OBJECT.....	4
Latihan 1 .....	4
Latihan 2 .....	6
Latihan 3 .....	8
BAB 3 ENKAPSULASI.....	9
Latihan 1 .....	9
Latihan 2 .....	10
Latihan 3 .....	12
BAB 4 INHERITANCE .....	13
Latihan 1 .....	13
Latihan 2 .....	14

# BAB 1 KONSEP PEMROGRAMAN BERORIENTASI OBYEK

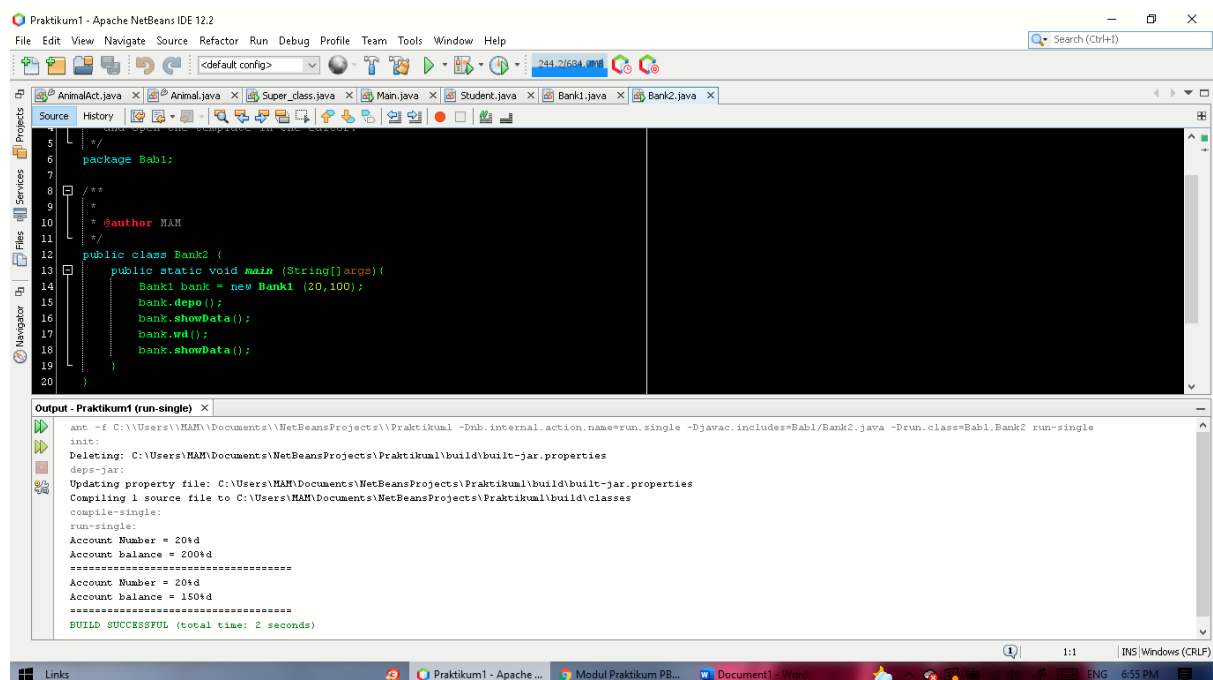
## Latihan 1

Perbedaan yang mendasar dari ketiga source code studi kasus aplikasi bank sederhana yaitu adanya method yang dipakai di source code ke 2 dan ke 3 yang mengakibatkan lebih praktis dan simple untuk memproses, juga mempersingkat waktu mengetik code dari pda yang digunakan di source code nomer 1 (kiri atas)

## Latihan 2



```
1 // To change this license header, choose License Headers in Project Properties.
2 // To change this template file, choose Tools | Templates
3 // and open the template in the editor.
4
5 package Bab1;
6
7 /**
8  *
9  * @author MAM
10  */
11
12 public class Bank1 {
13     int account_number = 20;
14     int account_balance = 100;
15
16     public Bank1 (int account_number, int account_balance){
17         this.account_number = account_number;
18         this.account_balance = account_balance;
19     }
20
21     void depo(){
22         account_balance +=100 ;
23     }
24
25     void wd(){
26         account_balance -= 50;
27     }
28
29     void showData(){
30         System.out.println("Account Number = "+ account_number+ "%d");
31         System.out.println("Account balance = "+ account_balance+"%d");
32         System.out.println("=====");
33     }
34 }
```



```
1 // To change this license header, choose License Headers in Project Properties.
2 // To change this template file, choose Tools | Templates
3 // and open the template in the editor.
4
5 package Bab1;
6
7 /**
8  *
9  * @author MAM
10  */
11
12 public class Bank2 {
13     public static void main (String[] args){
14         Bank1 bank = new Bank1 (20,100);
15         bank.depo ();
16         bank.showData ();
17         bank.wd ();
18         bank.showData ();
19     }
20 }
```

```
Output - Praktikum1 (run-single)
ant -f C:\Users\MAM\Documents\NetBeansProjects\Praktikum1\build-internal.action.name=run.single -Djavac.includes=Bab1/Bank2.java -Druntime.class=Bab1.Bank2 run-single
init:
Deleting: C:\Users\MAM\Documents\NetBeansProjects\Praktikum1\build\build-jar.properties
deps-jar:
Updating property file: C:\Users\MAM\Documents\NetBeansProjects\Praktikum1\build\build-jar.properties
Compiling 1 source file to C:\Users\MAM\Documents\NetBeansProjects\Praktikum1\build\classes
compile-single:
run-single:
Account Number = 20%d
Account balance = 200%d
=====
Account Number = 20%d
Account balance = 150%d
=====
BUILD SUCCESSFUL (total time: 2 seconds)
```

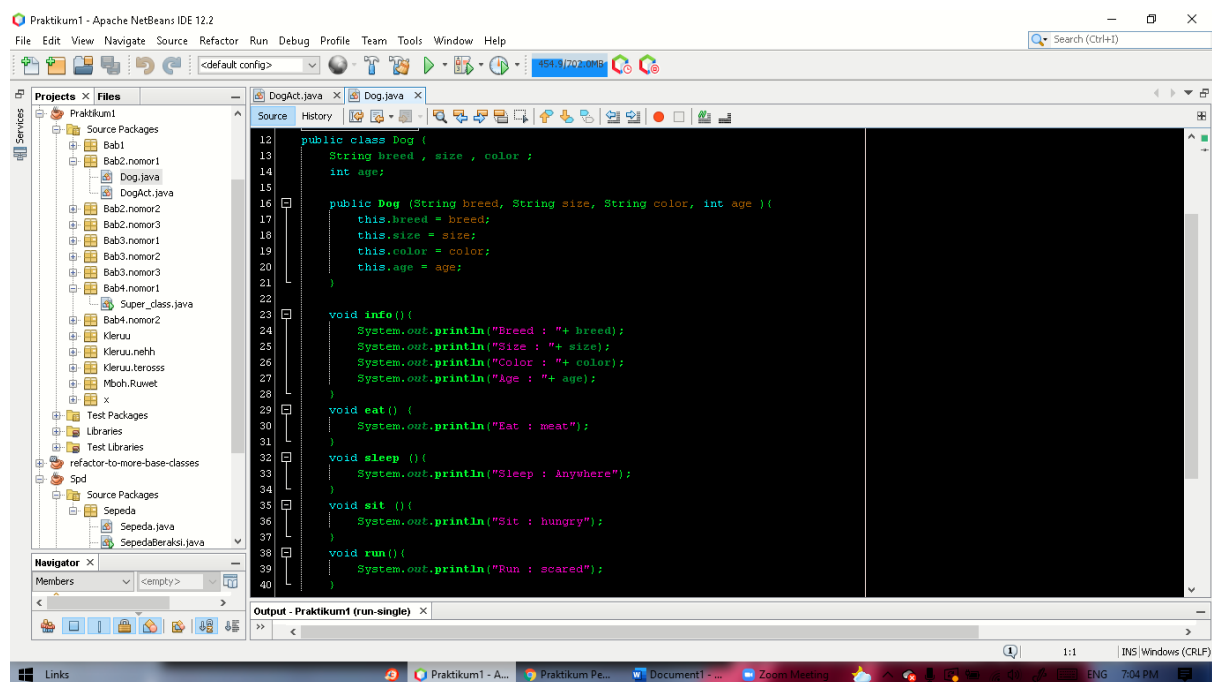
## Latihan 3

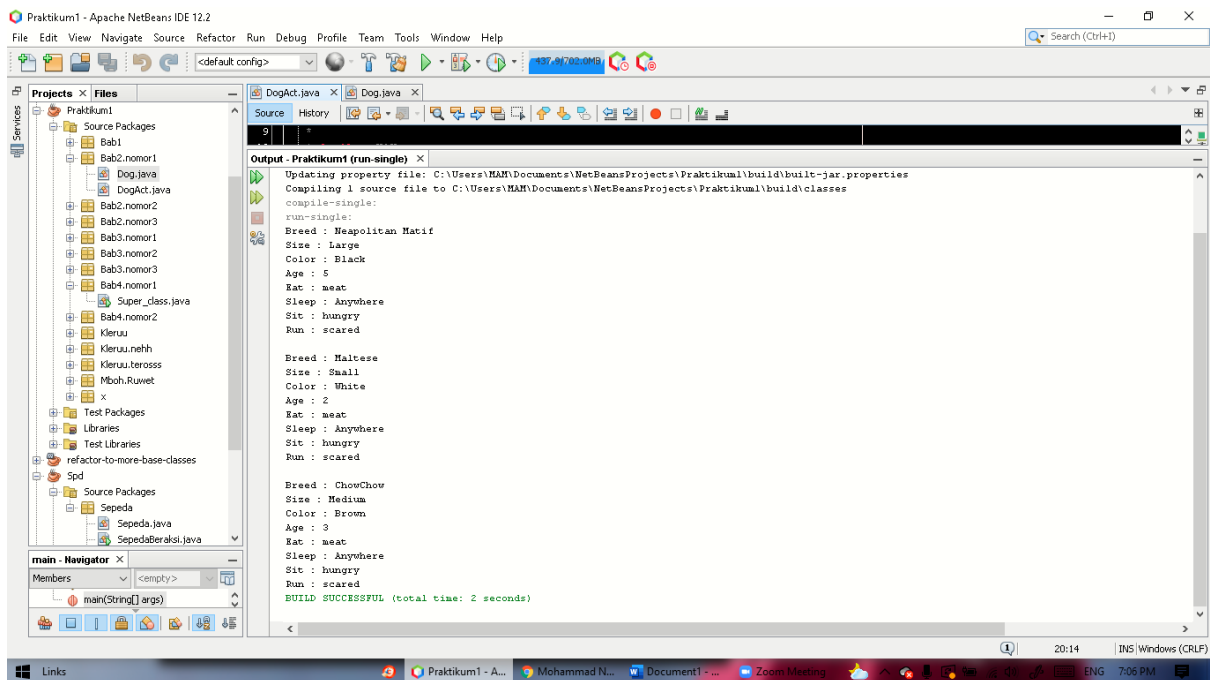
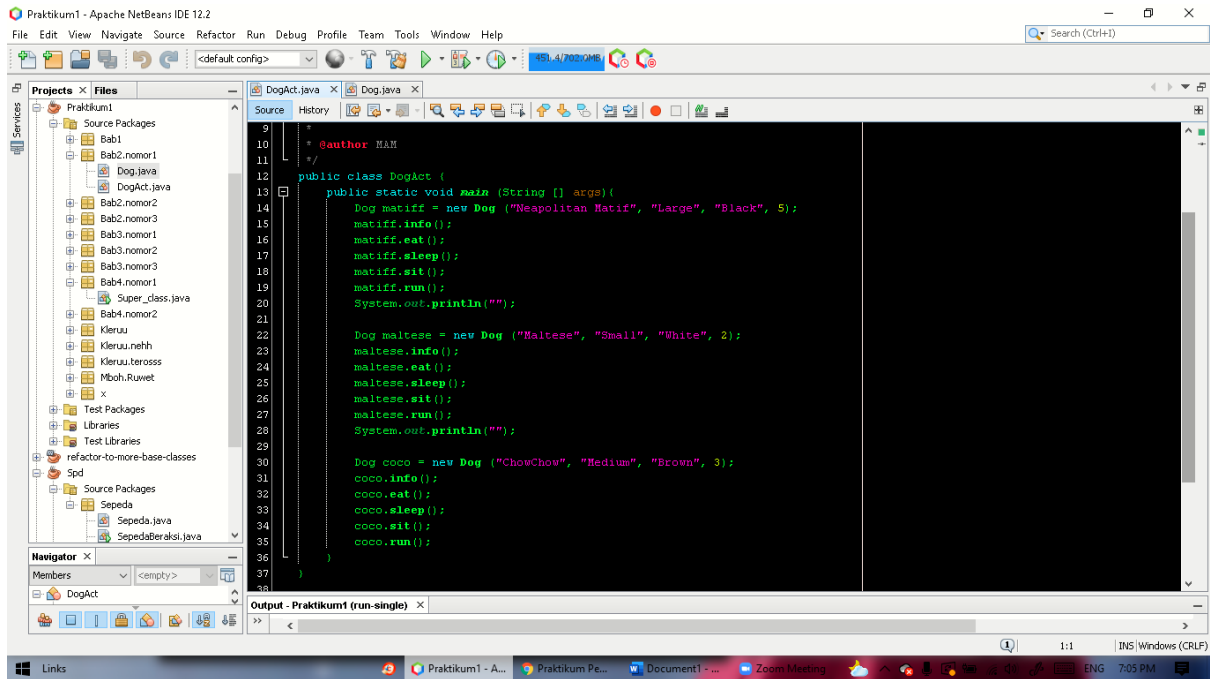
### Konsep dasar OOP/PBO :

- Class Merupakan suatu blueprint atau cetakan untuk menciptakan suatu instant dari object. Class juga merupakan grup suatu object dengan kemiripan attributes/properties, behavior dan relasi ke object lain.
- Object Merupakan instance dari class. Jika class secara umum merepresentasikan sebuah object, sebuah instance adalah representasi nyata dari class itu sendiri.
- Atribut Merupakan nilai data yang terdapat pada suatu object yang berasal dari class. Atribut merepresentasikan karakteristik dari suatu object.
- Method Merupakan suatu operasi berupa fungsi-fungsi yang dapat dikerjakan oleh suatu object. Method didefinisikan pada class akan tetapi dipanggil melalui object. Metode menentukan perilaku objek, yakni apa yang terjadi Ketika objek itu dibuat serta berbagai operasi yang dapat dilakukan objek sepanjang hidupnya.
- Constructor Merupakan suatu method yang akan memberikan nilai awal pada saat suatu objek dibuat. Pada saat program dijalankan, constructor akan bekerja dengan constructor.

## BAB 2 CLASS DAN OBJECT

### Latihan 1





## Latihan 2

The screenshot shows the Apache NetBeans IDE interface. The main editor window displays a Java file named `Person.java`. The code defines a `Person` class with attributes `fname`, `lname`, `email`, and `age`. A `main` method is also present, which creates a `Person` object and prints its details. The output window at the bottom shows the successful execution of the program, displaying the name, email, and age of the person.

```
7  /**
8   *
9   * @author MAM
10  */
11
12  public class Person {
13      String fname = "John";
14      String lname = "Doe";
15      String email = "john@doe.com";
16      int age = 24;
17
18      public static void main(String[] args) {
19          Person myObj = new Person();
20          System.out.println("Name: " + myObj.fname + " " + myObj.lname);
21          System.out.println("Email: " + myObj.email);
22          System.out.println("Age: " + myObj.age);
23      }
24  }
```

Output - Praktikum1 (run-single) X

```
Deleting: C:\Users\MAM\Documents\NetBeansProjects\Praktikum1\build\build-jar.properties
deps-jar:
Updating property file: C:\Users\MAM\Documents\NetBeansProjects\Praktikum1\build\build-jar.properties
Compiling 1 source file to C:\Users\MAM\Documents\NetBeansProjects\Praktikum1\build\classes
compile-single:
run-single:
Name: John Doe
Email: john@doe.com
Age: 24
BUILD SUCCESSFUL (total time: 2 seconds)
```

Analisa : Program pertama termasuk dalam tipe program yang hanya memiliki satu class dan proses pemanggilan ada di class yang sama.

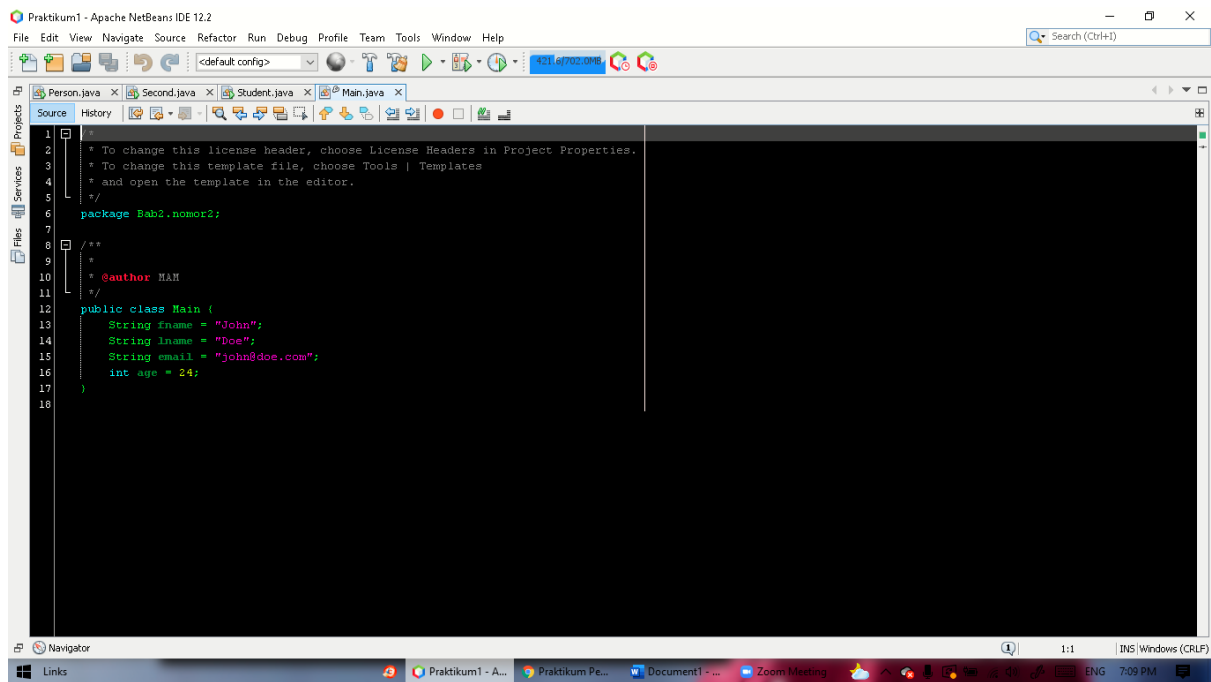
The screenshot shows the Apache NetBeans IDE interface. The main editor window displays a Java file named `Person.java`. The code defines a `Person` class with attributes `fname`, `lname`, `email`, and `age`. A `Student` class is also present, which extends the `Person` class and has its own `main` method. The output window at the bottom shows the successful execution of the program, displaying the name, email, age, and graduation year of the student.

```
6  package Bab2.nomor2;
7
8  /**
9   *
10  * @author MAM
11  */
12
13  class Person {
14      String fname = "John";
15      String lname = "Doe";
16      String email = "john@doe.com";
17      int age = 24;
18  }
19
20  public class Student extends Person {
21      private int graduationYear = 2018;
22      public static void main(String[] args) {
23          Student myObj = new Student();
24          System.out.println("Name: " + myObj.fname + " " + myObj.lname);
25          System.out.println("Email: " + myObj.email);
26          System.out.println("Age: " + myObj.age);
27          System.out.println("Graduation Year: " + myObj.graduationYear);
28      }
29  }
```

Output - Praktikum1 (run-single) X

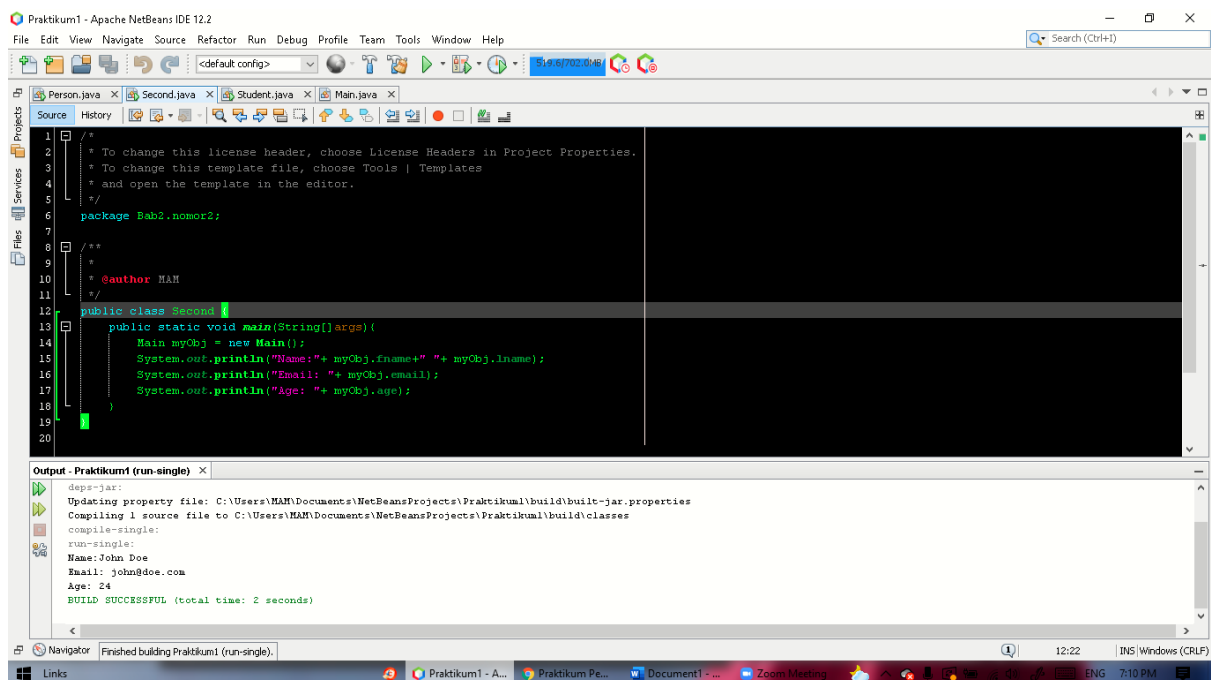
```
deps-jar:
Updating property file: C:\Users\MAM\Documents\NetBeansProjects\Praktikum1\build\build-jar.properties
Compiling 1 source file to C:\Users\MAM\Documents\NetBeansProjects\Praktikum1\build\classes
compile-single:
run-single:
Name: John Doe
Email: john@doe.com
Age: 24
Graduation Year: 2018
BUILD SUCCESSFUL (total time: 2 seconds)
```

Analisa : Program kedua memiliki dua class yang berbeda, dan proses pemanggilan ada pada class yang berbeda namun package nya sama, keduanya saling berhubungan karena pada salah satu class menggunakan extends untuk mengambil fungsi pada class satunya. Protected fungsinya untuk melindungi konstruktor, mencegah pengguna membuat instance class diluar package.



The screenshot shows the Apache NetBeans IDE with the 'Main.java' file open. The code defines a package 'Bab2.nomor2' and a class 'Main' with a 'main' method. The 'main' method initializes variables for 'fname' (John), 'lname' (Doe), 'email' (john@doe.com), and 'age' (24).

```
1  *  
2  * To change this license header, choose License Headers in Project Properties.  
3  * To change this template file, choose Tools | Templates  
4  * and open the template in the editor.  
5  */  
6  package Bab2.nomor2;  
7  
8  /**  
9   *  
10  * @author MAM  
11  */  
12  public class Main (  
13      String fname = "John";  
14      String lname = "Doe";  
15      String email = "john@doe.com";  
16      int age = 24;  
17  )  
18  
```



The screenshot shows the Apache NetBeans IDE with the 'Second.java' file open. The code defines a package 'Bab2.nomor2' and a class 'Second' with a 'main' method. The 'main' method creates an instance of the 'Main' class and prints its attributes. Below the code editor, the 'Output' window shows the successful execution of the program, displaying the output: 'Name: John Doe', 'Email: john@doe.com', and 'Age: 24'.

```
1  /*  
2  * To change this license header, choose License Headers in Project Properties.  
3  * To change this template file, choose Tools | Templates  
4  * and open the template in the editor.  
5  */  
6  package Bab2.nomor2;  
7  
8  /**  
9   *  
10  * @author MAM  
11  */  
12  public class Second {  
13      public static void main(String[] args){  
14          Main myObj = new Main();  
15          System.out.println("Name:" + myObj.fname + " " + myObj.lname);  
16          System.out.println("Email: " + myObj.email);  
17          System.out.println("Age: " + myObj.age);  
18      }  
19  }  
20  
```

Output - Praktikum1 (run-single) X

```
deps-jar:  
Updating property file: C:\Users\MAM\Documents\NetBeansProjects\Praktikum1\build\build-jar.properties  
Compiling 1 source file to C:\Users\MAM\Documents\NetBeansProjects\Praktikum1\build\classes  
compile-single:  
run-single:  
Name: John Doe  
Email: john@doe.com  
Age: 24  
BUILD SUCCESSFUL (total time: 2 seconds)
```

Analisa : Program ketiga memiliki dua class berbeda namun dalam satu package. Proses untuk pemanggilan fungsi terletak pada class kedua, variabel yang digunakan pada public supaya dapat mudah mengakses fungsi.

## Latihan 3

Praktikum1 - Apache NetBeans IDE 12.2

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config> 488.5/700.0 MB
```

Source History

```
10 //
11 // @author MAM
12 //
13 public class Venicle {
14     String brandName;
15     String color;
16     int numberOfWheels, numberGear;
17     double price;
18     void info (String brandName, String color, int numberOfWheels, double price){
19         this.brandName = brandName;
20         this.color = color;
21         this.numberOfWheels = numberOfWheels;
22         this.price = price;
23     }
24     void print (){
25         System.out.println(" Brand Name : " + brandName);
26         System.out.println(" Color : " + color);
27         System.out.println(" Number Of Wheels : " + numberOfWheels);
28         System.out.println(" Price : " + price + "Juta");
29     }
30     void start(int Gear){
31         this.numberGear = Gear;
32         System.out.println(" Total Gear : " + Gear);
33     }
34     void changeGear (int UpGear){
35         this.UpGear = UpGear;
36         int Up = UpGear + 1;
37         System.out.println(" Change Gear : From " + UpGear + " To " + Up );
38         System.out.println("-----");
39     }
40 }
41
```

15:38 INS|Windows (CRLF)

Praktikum1 - Apache NetBeans IDE 12.2

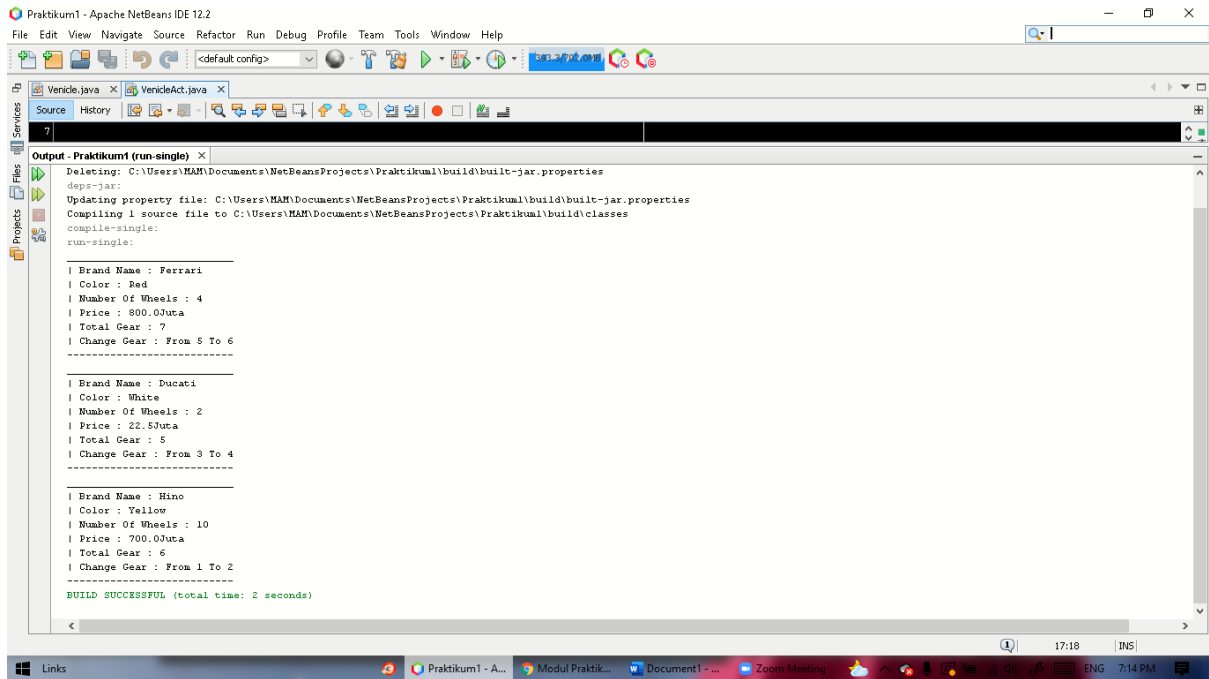
```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config> 488.5/700.0 MB
```

Source History

```
7 //
8 //
9 // @author MAM
10 //
11 //
12 public class VenicleAct {
13     public static void main(String[] args){
14         Venicle car = new Venicle();
15         car.info("Ferrari", "Red", 4, 800);
16         car.print();
17         car.start(7);
18         car.changeGear(5);
19
20         Venicle bike = new Venicle();
21         bike.info("Ducati", "White", 2, 22.500);
22         bike.print();
23         bike.start(5);
24         bike.changeGear(3);
25
26         Venicle truck = new Venicle();
27         truck.info("Hino", "Yellow", 10, 700);
28         truck.print();
29         truck.start(6);
30         truck.changeGear(1);
31     }
32 }
33
```

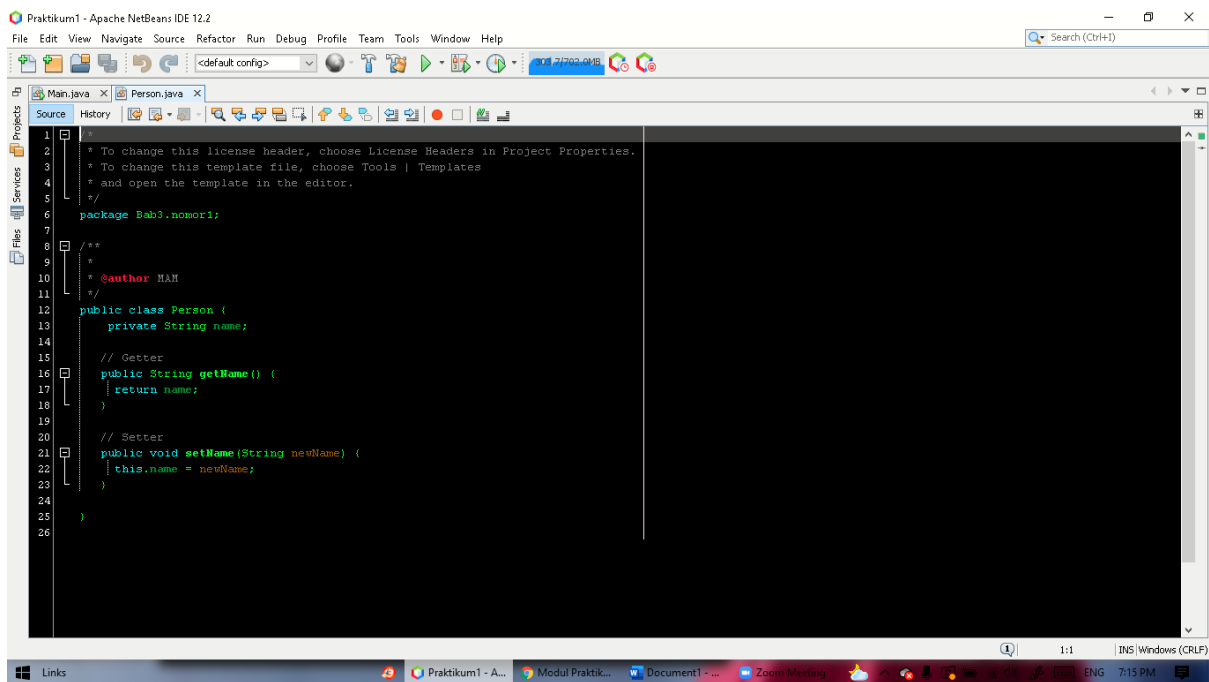
17:18 INS|Windows (CRLF)

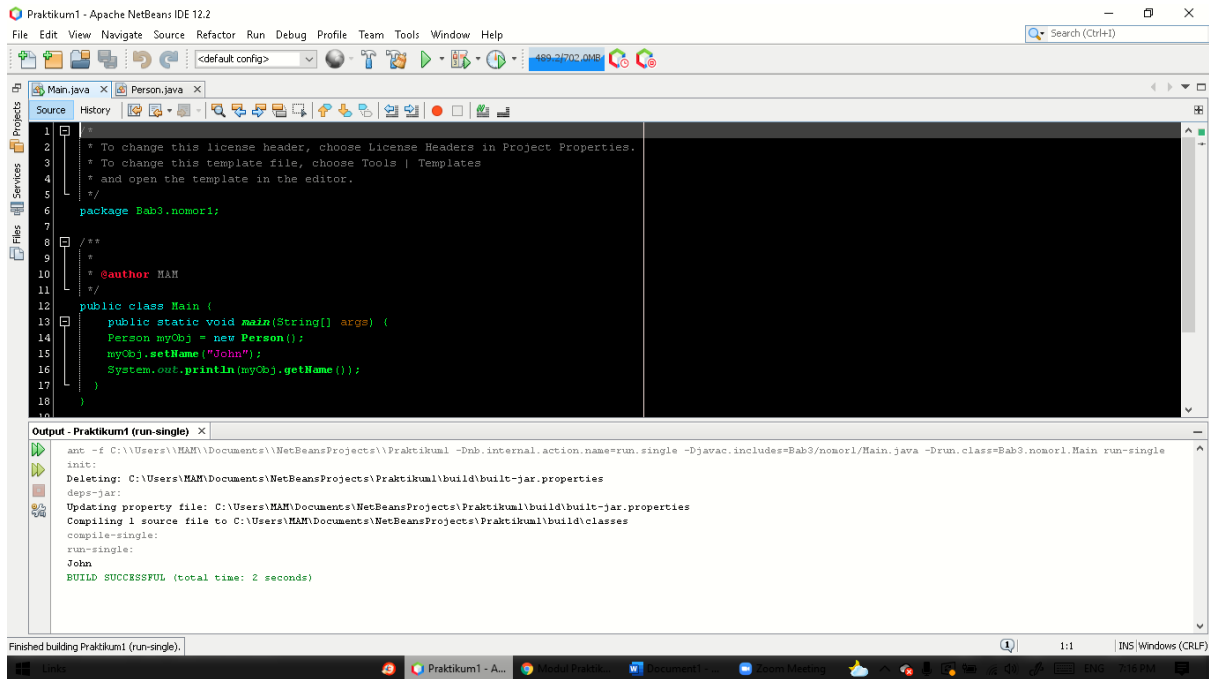




## BAB 3 ENKAPSULASI

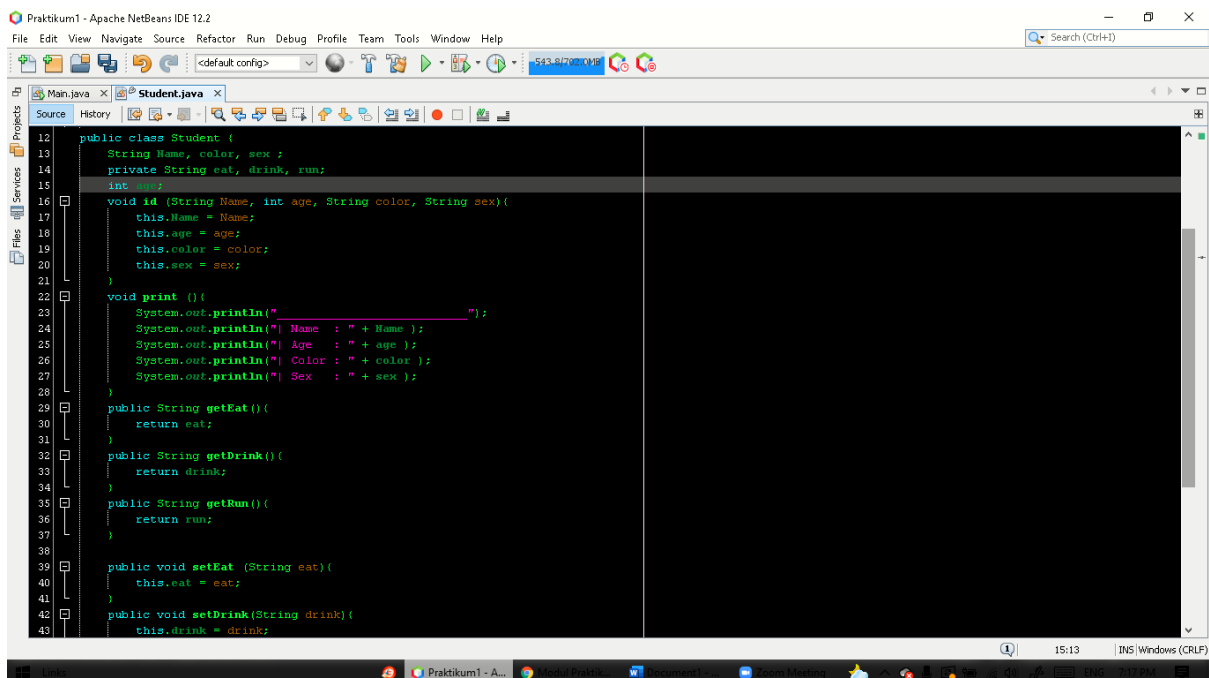
### Latihan 1





Analisa : pada source code awal (masih ada error pada kelas “main”), dikarenakan adanya variabel pada kelas “person”, yang menggunakan access modifier dengan nama “private”. Karena itulah variabel pada kelas “person” tidak bisa diakses pada kelas lain untuk mengisi variable dengan data yang ada “jhon”. Maka dari itu access modifier pada variable yang berada di kelas “person”, yang awalnya bernama “private” harus diganti dengan “protected” atau yang lain, agar dapat diakses di kelas lain.

## Latihan 2



Praktikum1 - Apache NetBeans IDE 12.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Source History

Projects

Files

Services

Main.java Student.java

```
8  /**
9   *
10  * @author MAM
11  */
12  public class Main {
13      public static void main(String[] args) {
14          Student john = new Student();
15          john.id("John", 12, "Fair", "Male");
16          john.print();
17          john.setEat("More");
18          john.setDrink("More");
19          john.setRun("Fast");
20          System.out.println(" | " + john.Name + " Can Eat " + john.getEat());
21          System.out.println(" | " + john.Name + " Can Drink " + john.getDrink());
22          System.out.println(" | " + john.Name + " Can Run " + john.getRun());
23          System.out.println("=====");
24
25          Student sophia = new Student();
26          sophia.id("sophia", 10, "Fair", "Female");
27          sophia.print();
28          sophia.setEat("Less");
29          sophia.setDrink("Less");
30          sophia.setRun("Slow");
31          System.out.println(" | " + john.Name + " Can Eat " + sophia.getEat());
32          System.out.println(" | " + john.Name + " Can Drink " + sophia.getDrink());
33          System.out.println(" | " + john.Name + " Can Run " + sophia.getRun());
34          System.out.println("=====");
35
36          Student Lili = new Student();
37          Lili.id("Lili", 11, "Dark", "Female");
38          Lili.print();
39          Lili.setEat("More");
```

1:1 | INS | Windows (CRUF)

Praktikum1 - Apache NetBeans IDE 12.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

Source History

Projects

Files

Services

Main.java Student.java

```
28  sophia.print();
29  sophia.setEat("Less");
30  sophia.setDrink("Less");
31  sophia.setRun("Slow");
32  System.out.println(" | " + john.Name + " Can Eat " + sophia.getEat());
33  System.out.println(" | " + john.Name + " Can Drink " + sophia.getDrink());
```

Output - Praktikum1 (run-single)

```
| Name : John
| Age : 12
| Color : Fair
| Sex : Male
| John Can Eat More
| John Can Drink More
| John Can Run Fast
=====
| Name : sophia
| Age : 10
| Color : Fair
| Sex : Female
| John Can Eat Less
| John Can Drink Less
| John Can Run Slow
=====
| Name : Lili
| Age : 11
| Color : Dark
| Sex : Female
| John Can Eat More
| John Can Drink More
| John Can Run Fast
=====
BUILD SUCCESSFUL (total time: 2 seconds)
```

38:22 | INS

## Latihan 3

Praktikum1 - Apache NetBeans IDE 12.2

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config> 316.8/702.0 MB
```

Projects: MakYam.java, MakYamAct.java

```
12 public class MakYam {
13     private String menu;
14     private int harga, diskon, ongkir, potongan;
15
16     void setMenu(String Menu){
17         this.menu = Menu;
18     }
19     public String getMenu(){
20         return menu;
21     }
22     void setHarga(int Harga){
23         this.harga = Harga;
24     }
25     public int getHarga(){
26         return harga;
27     }
28     void setDisc(int Disc){
29         diskon = harga - Disc;
30         System.out.println("-Diskon      : " + diskon);
31     }
32     public int getDisc (){
33         return diskon;
34     }
35     void setOngkir (int potongan){
36         this.ongkir = potongan;
37     }
38     public int getOngkir(){
39         return ongkir;
40     }
41     void setPotongan(int Potongan){
42         potongan = ongkir - Potongan;
43         System.out.println("Diskon Ongkir sd 7k : " + potongan);
44     }
45 }
```

15:1 | INS | Windows (CTRL)

Praktikum1 - Apache NetBeans IDE 12.2

```
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
<default config> 316.8/702.0 MB
```

Projects: MakYam.java, MakYamAct.java

```
10  * @author MAH
11  */
12  public class MakYamAct {
13      public static void main(String[] args) {
14          MakYam nota = new MakYam();
15          nota.setMenu("Bakso" + " Es Teh " + "Lolipop ");
16          System.out.println("=====");
17          System.out.println("Diskon 35% sd 50% + potongan ongkir 7k");
18          System.out.println("=====");
19          System.out.println("Menu yang Diambil : " + nota.getMenu());
20          nota.setHarga(15000);
21          System.out.println("Total      : " + nota.getHarga());
22          nota.setDisc(5500);
23          nota.setOngkir(12000);
24          System.out.println("Ongkir      : " + nota.getOngkir());
25          nota.setPotongan(7000);
26          System.out.println("Total semua : " + nota.getDisc());
27      }
28  }
```

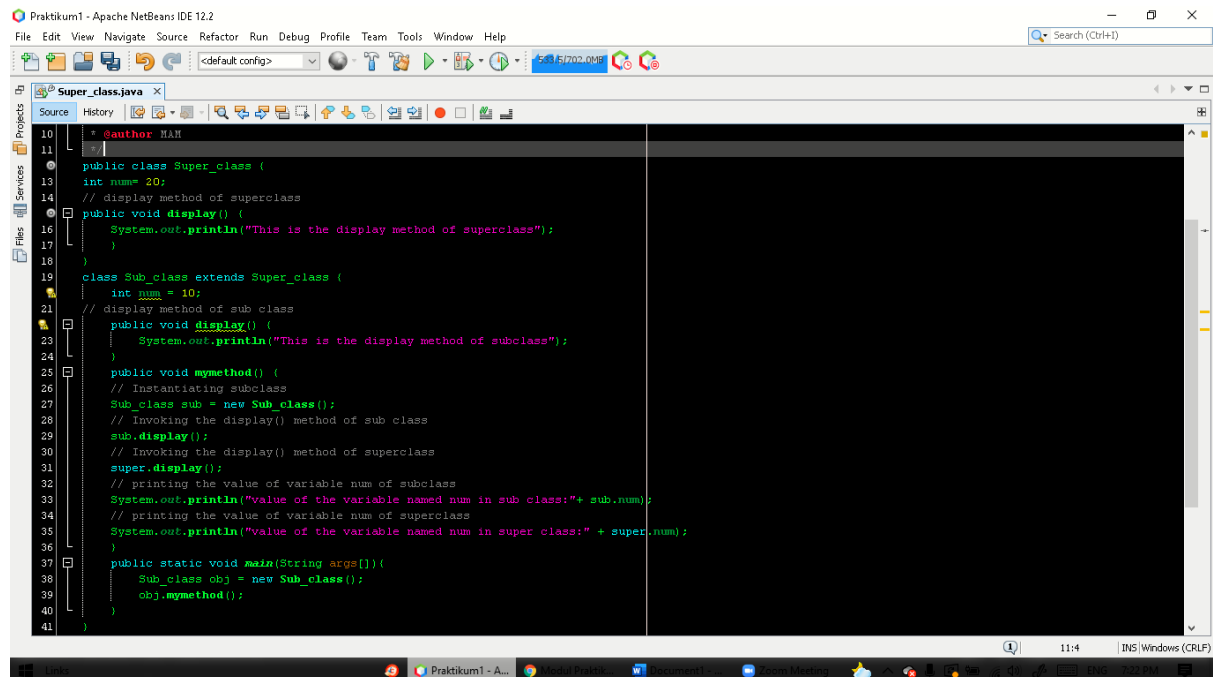
Output - Praktikum1 (run-single)

```
run-single:
=====
Diskon 35% sd 50% + potongan ongkir 7k
=====
Menu yang Diambil : Bakso Es Teh Lolipop
Total      : 15000
-Diskon      : 9500
Ongkir      : 12000
Diskon Ongkir sd 7k : 5000
Total semua : 9500
BUILD SUCCESSFUL (total time: 2 seconds)
```

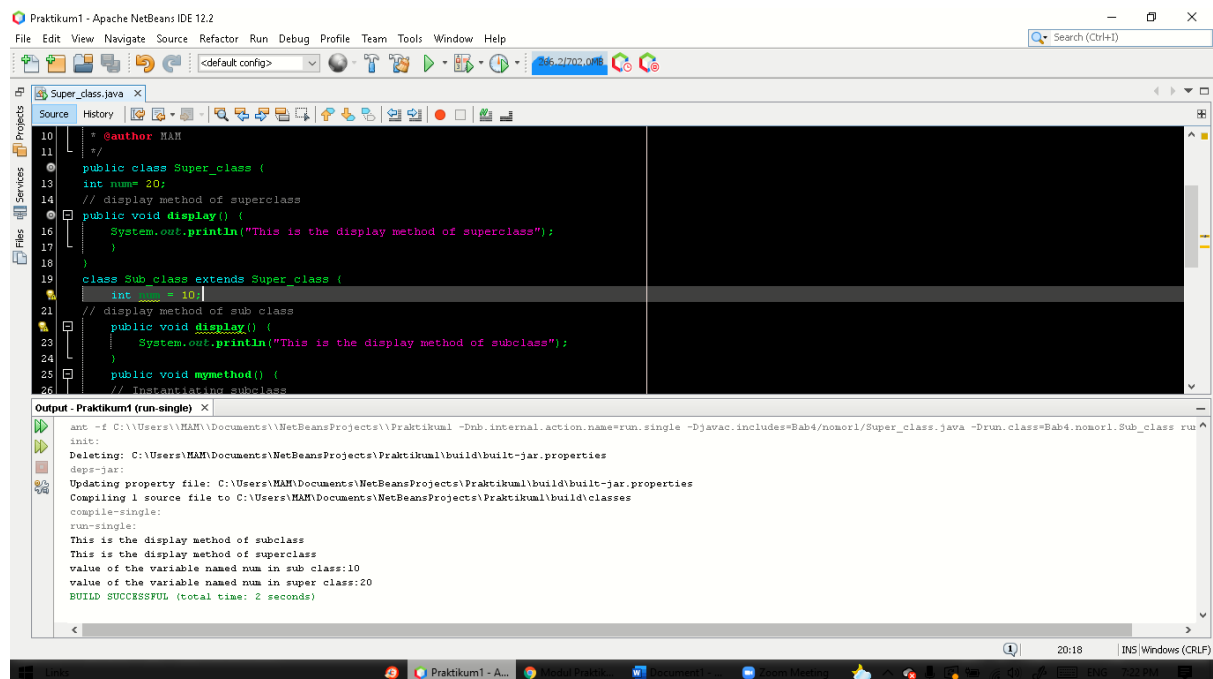
1:1 | INS

# BAB 4 INHERITANCE

## Latihan 1



```
10  * @author MAM
11  */
12  public class Super_class {
13      int num= 20;
14      // display method of superclass
15      public void display() {
16          System.out.println("This is the display method of superclass");
17      }
18  }
19  class Sub_class extends Super_class {
20      int num = 10;
21      // display method of sub class
22      public void display() {
23          System.out.println("This is the display method of subclass");
24      }
25      public void mymethod() {
26          // Instantiating subclass
27          Sub_class sub = new Sub_class();
28          // Invoking the display() method of sub class
29          sub.display();
30          // Invoking the display() method of superclass
31          super.display();
32          // printing the value of variable num of subclass
33          System.out.println("Value of the variable named num in sub class:"+ sub.num);
34          // printing the value of variable num of superclass
35          System.out.println("Value of the variable named num in super class:" + super.num);
36      }
37      public static void main(String args[]){
38          Sub_class obj = new Sub_class();
39          obj.mymethod();
40      }
41  }
```



```
Output - Praktikum1 (run-single) x
ant -f C:\Users\MAM\Documents\NetBeansProjects\Praktikum1\build\build-jar.properties
init:
Deleting: C:\Users\MAM\Documents\NetBeansProjects\Praktikum1\build\build-jar.properties
Dep-jar:
Updating property file: C:\Users\MAM\Documents\NetBeansProjects\Praktikum1\build\build-jar.properties
Compiling 1 source file to C:\Users\MAM\Documents\NetBeansProjects\Praktikum1\build\classes
compile-single:
run-single:
This is the display method of subclass
This is the display method of superclass
value of the variable named num in sub class:10
value of the variable named num in super class:20
BUILD SUCCESSFUL (total time: 2 seconds)
```

Analisa : Pada hasil uji coba program tersebut mengimplementasikan inheritance, dengan ditandai extends. Program ini memiliki 2 class, class yang pertama merupakan inheritance dari class kedua. Pada class pertama terdapat 1 method, class yang kedua ada 3 method. Method tersebut digunakan untuk mengembalikan nilai. Super di class kedua digunakan sebagai objek untuk mengambil nilai dari variabel kelas induk.

## Latihan 2

Praktikum1 - Apache NetBeans IDE 12.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

<default config>

463.0/702.0MB

```
4  * and open the template in the editor.
5
6  package Bab4.nomor2;
7
8
9  /**
10   *
11   * @author MAM
12   */
13
14  class Animal{
15      public void MakananHewan(){
16          System.out.println("Klarifikasi Hewan berdasarkan jenis makanan \n");
17      }
18  }
19
20  class herbivora extends Animal{
21      @Override
22      public void MakananHewan(){
23          System.out.println("Hewan Herbivora atau hewan pemakan Tumbuhan");
24      }
25  }
26
27  class rabbit extends herbivora{
28      @Override
29      public void MakananHewan(){
30          System.out.println("Makanan rabbit : wortel, sawi, rumput \n");
31      }
32  }
33
34  class carnivora extends Animal{
35      @Override
36      public void MakananHewan(){
37          System.out.println("Hewan Carnivora atau hewan pemakan Daging");
38      }
39  }
40
41  class omnivora extends Animal{
42      @Override
43      public void MakananHewan(){
44          System.out.println("Hewan Omnivora atau hewan pemakan segala");
45      }
46  }
47
48  class man extends omnivora{
49      @Override
50      public void MakananHewan(){
51          System.out.println("Makanan manusia : semua masuk sih \n");
52      }
53  }
```

5:4 | INS | Windows (CRLF)

Praktikum1 - Apache NetBeans IDE 12.2

File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help

Search (Ctrl+I)

<default config>

414.0/12.0MB

```
35
36  class lion extends carnivora{
37      @Override
38      public void MakananHewan(){
39          System.out.println("Makanan lion : daging ");
40      }
41  }
42
43  class hyena extends carnivora{
44      @Override
45      public void MakananHewan(){
46          System.out.println("Makanan hyena : daging \n");
47      }
48  }
49
50  class omnivora extends Animal{
51      @Override
52      public void MakananHewan(){
53          System.out.println("Hewan Omnivora atau hewan pemakan segala");
54      }
55  }
56
57  class man extends omnivora{
58      @Override
59      public void MakananHewan(){
60          System.out.println("Makanan manusia : semua masuk sih \n");
61      }
62  }
```

5:4 | INS | Windows (CRLF)

