

MEMAHAMI DASAR - DASAR PEMROGRAMAN JAVA

Pemrograman Berorientasi Objek (PBO) atau *Object Oriented Programming* (OOP) menggunakan Bahasa Pemrograman Java. Bahasa Pemrograman Java memiliki beberapa komponen, antara lain:

- ▶ *Class* : tempat untuk mendeklarasikan tipe data
- ▶ *Object* : entiti yang memiliki keadaan/ tingkah laku
- ▶ *Attribute* : elemen dari sebuah objek yang berisi informasi tentang objek
- ▶ *Method* : tingkah laku dari objek/sub program
- ▶ *Construction* : method yang digunakan untuk membuat objek baru

MEMAHAMI DASAR - DASAR PEMROGRAMAN JAVA

Bahasa pemrograman java dikenal dengan *case sensitive*, artinya ada banyak aturan dalam Pemrograman Java yang harus benar-benar diperhatikan. Berikut ini beberapa aturan dalam pemberian nama *class*:

- a. Nama file **halaman kerja java** harus sama dengan **nama class**
- b. Nama class boleh terdiri dari huruf kecil dan huruf besar
- c. Nama class yang terdiri dari dua suku kata, dapat dituliskan dengan tanda underscore (_) sebagai pemisah. Contoh : Belajar_Java
- d. Nama class tidak boleh terdiri dari angka
- e. Nama class boleh ditulis dengan huruf abjad dan diikuti dengan angka.

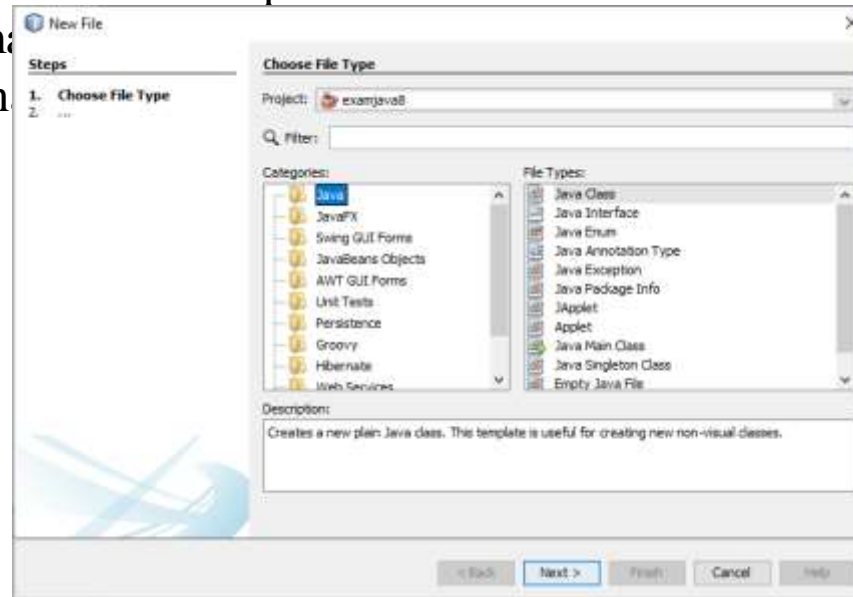
Contoh : Belajar_Java1

- f. Nama class tidak boleh menggunakan keyword (*public*, *class*, *private*, dan sebagainya)

PRAKTIKUM

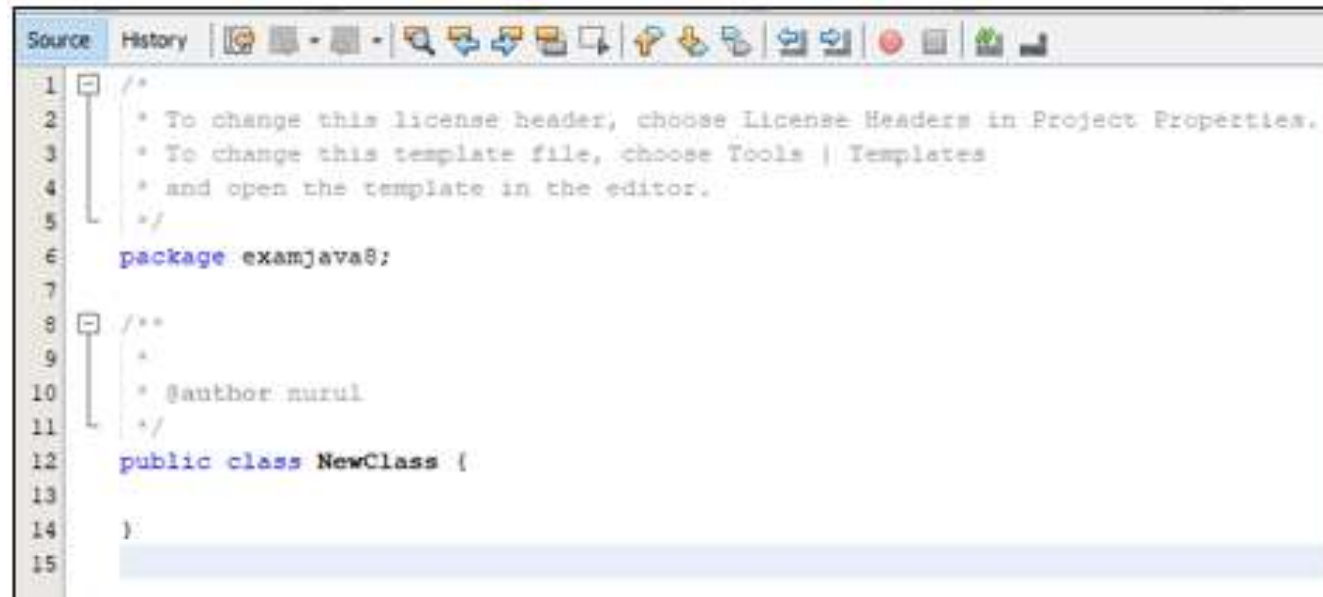
► Memulai Aplikasi Netbeans

Praktikum Pemrograman Berorientasi Objek (PBO) ini akan menggunakan software Java NetBeans. Halaman baru pada Java NetBeans dapat dimulai dari menu **New – New File – Java Class**. Kemudian berilah nama package dapat diisi sesuai dengan nama



PRAKTIKUM

Tampilan awal halaman kerja java dengan NewClass :

A screenshot of a Java IDE window titled 'Source'. The window shows a new class template for 'NewClass'. The code is as follows:

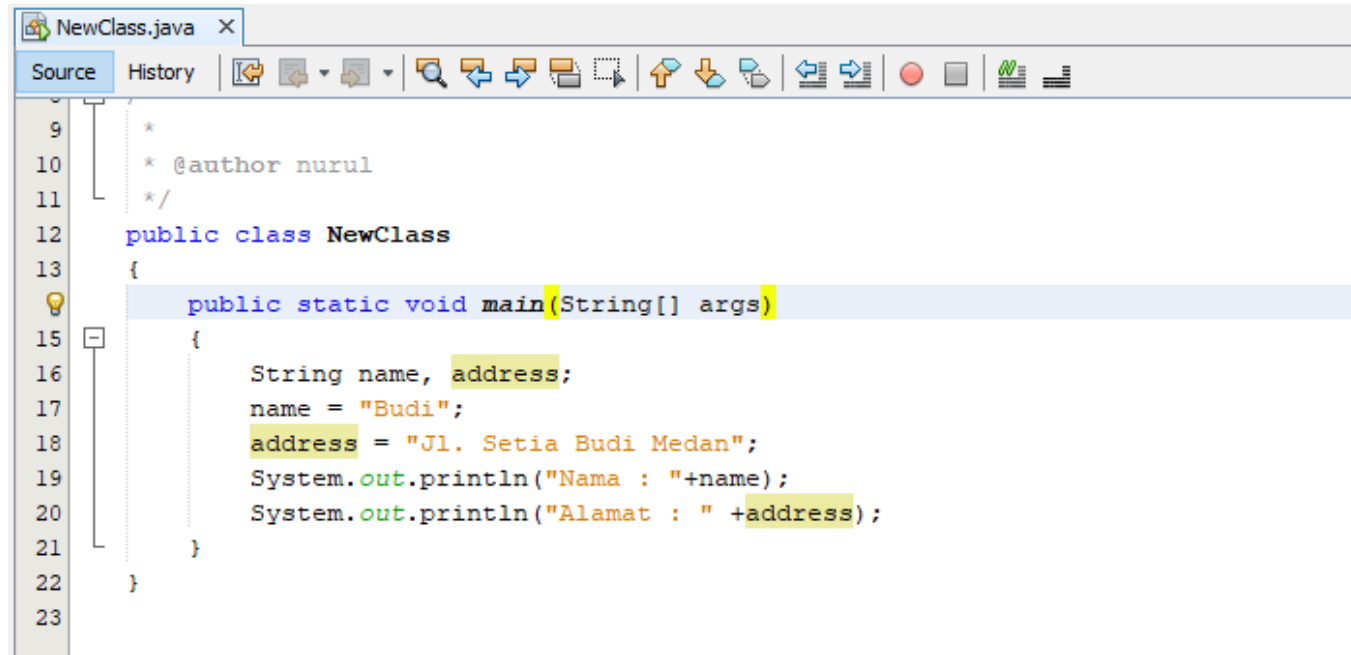
```
1  /*  
2   * To change this license header, choose License Headers in Project Properties.  
3   * To change this template file, choose Tools | Templates  
4   * and open the template in the editor.  
5   */  
6  package examjava8;  
7  
8  /**  
9   *  
10   * @author murul  
11   */  
12  public class NewClass {  
13  
14  }  
15
```

The code is color-coded: package names are blue, keywords like 'package', 'public', and 'class' are blue, and comments are in a light gray font. The line numbers 1 through 15 are visible on the left side of the editor.

PRAKTIKUM

2. Membuat Class

❖ Code Pemrograman:

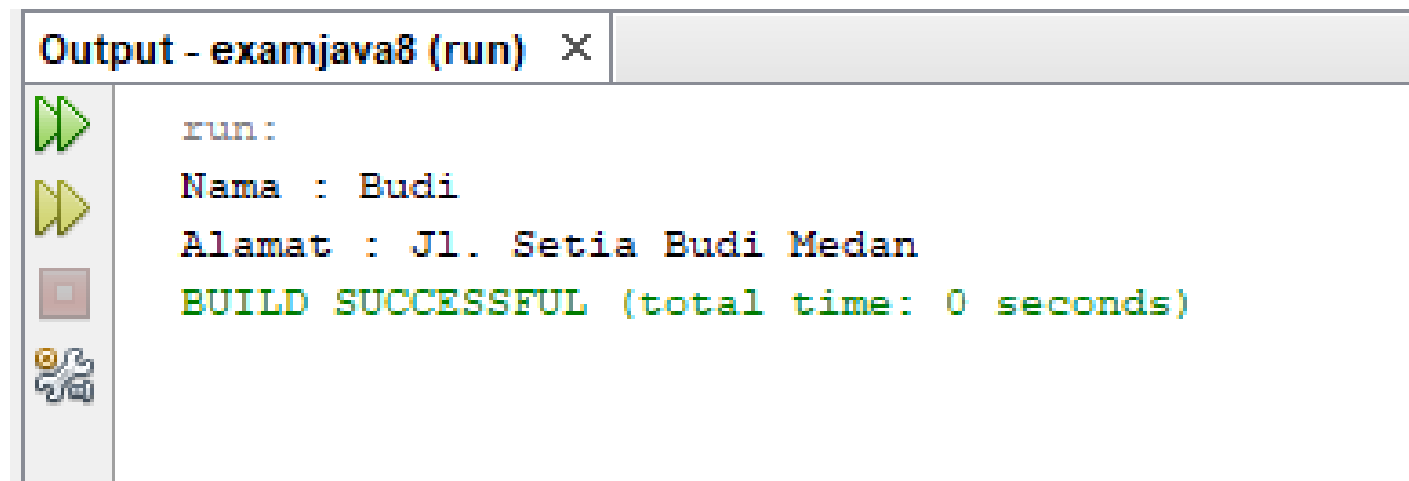


```
NewClass.java x
Source History
9      *
10     * @author nurul
11     */
12     public class NewClass
13     {
14     public static void main(String[] args)
15     {
16         String name, address;
17         name = "Budi";
18         address = "Jl. Setia Budi Medan";
19         System.out.println("Nama : "+name);
20         System.out.println("Alamat : " +address);
21     }
22 }
23
```

PRAKTIKUM

2. Membuat Class

❖ Hasil Pemrograman :



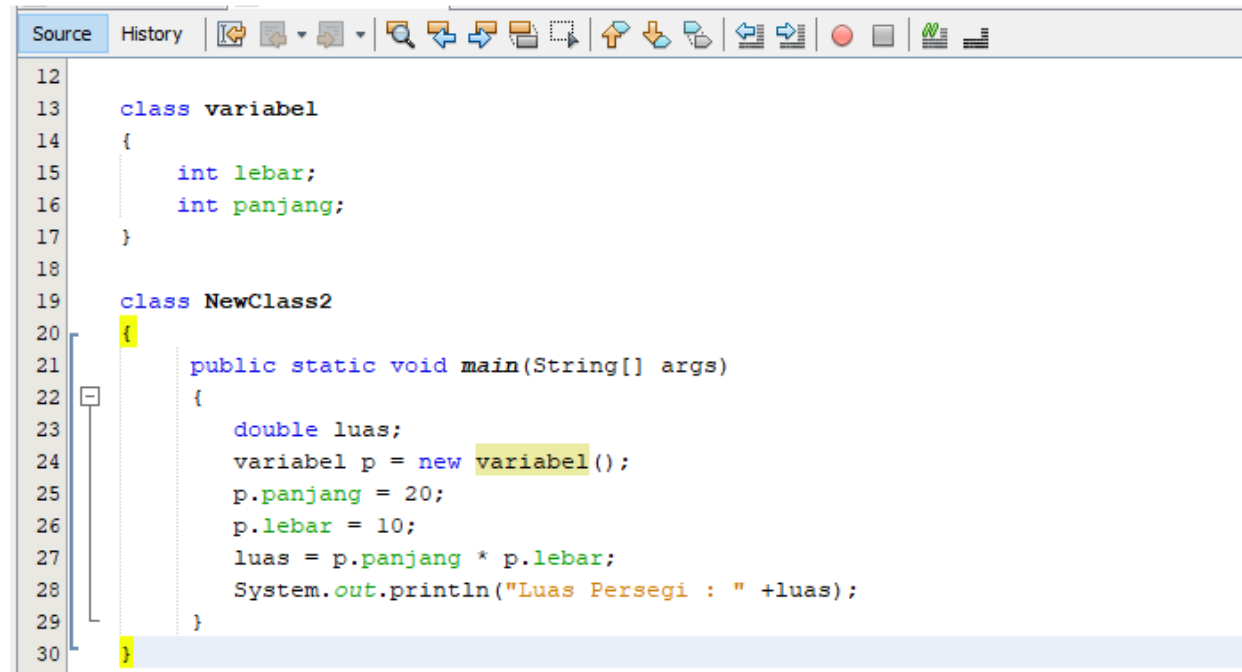
The screenshot shows a window titled "Output - examjava8 (run) X". On the left side of the window, there is a vertical toolbar with four icons: a green double arrow pointing right, a yellow double arrow pointing right, a red square, and a white bug icon. The main area of the window displays the following text in a monospaced font:

```
run:  
Nama : Budi  
Alamat : Jl. Setia Budi Medan  
BUILD SUCCESSFUL (total time: 0 seconds)
```

PRAKTIKUM

3. Membuat *Object*

❖ Code Pemrograman :

A screenshot of a Java IDE window. The window has a menu bar with 'Source' and 'History'. Below the menu bar is a toolbar with various icons for editing and running code. The main area shows a code editor with the following Java code:

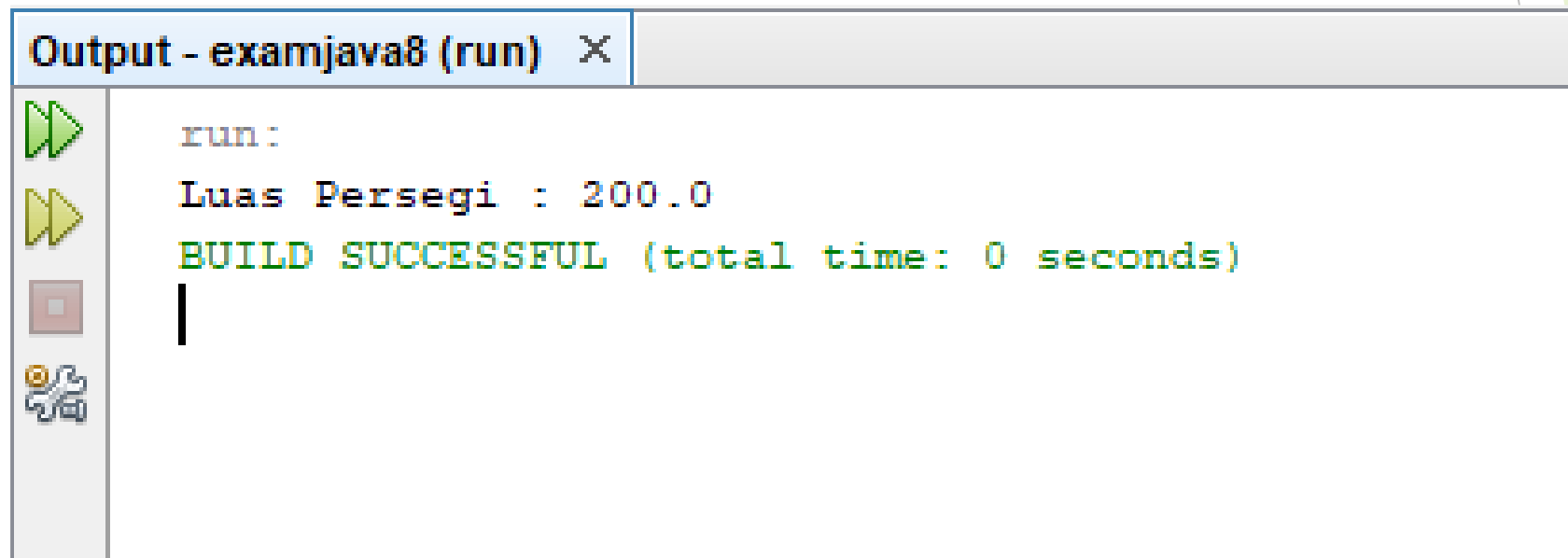
```
12
13  class variabel
14  {
15      int lebar;
16      int panjang;
17  }
18
19  class NewClass2
20  {
21      public static void main(String[] args)
22      {
23          double luas;
24          variabel p = new variabel();
25          p.panjang = 20;
26          p.lebar = 10;
27          luas = p.panjang * p.lebar;
28          System.out.println("Luas Persegi : " +luas);
29      }
30  }
```

The code defines a class named 'variabel' with two integer attributes, 'lebar' and 'panjang'. It then defines a second class named 'NewClass2' which contains a 'main' method. In the 'main' method, an instance of the 'variabel' class is created and named 'p'. The 'panjang' attribute of 'p' is set to 20, and the 'lebar' attribute is set to 10. The area of a square is calculated by multiplying 'p.panjang' by 'p.lebar' and stored in a variable named 'luas'. Finally, the area is printed to the console using 'System.out.println'.

PRAKTIKUM

3. Membuat *Object*

❖ Hasil Pemrograman :



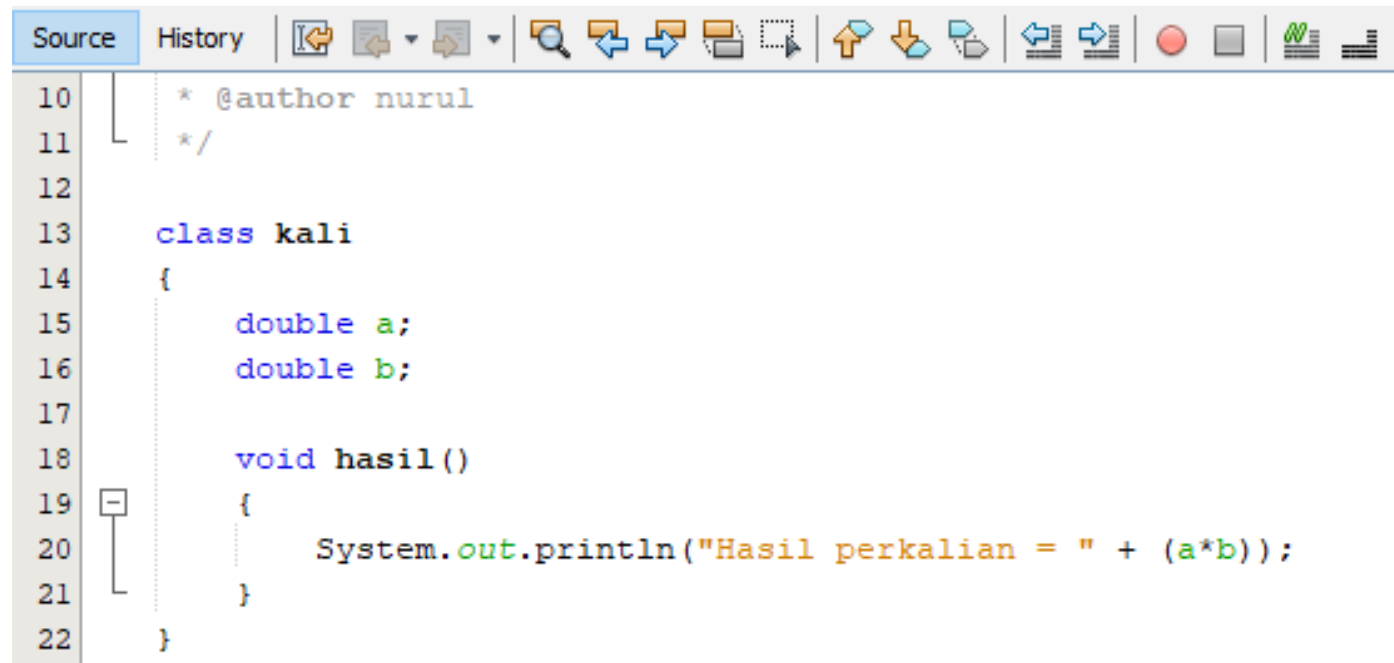
The screenshot shows a window titled "Output - examjava8 (run)" with a close button. On the left side of the window is a vertical toolbar containing four icons: a green double arrow, a yellow double arrow, a red square, and a gear icon. The main area of the window displays the following text:

```
run:
Luas Persegi : 200.0
BUILD SUCCESSFUL (total time: 0 seconds)
|
```


PRAKTIKUM

4. Membuat *Method*

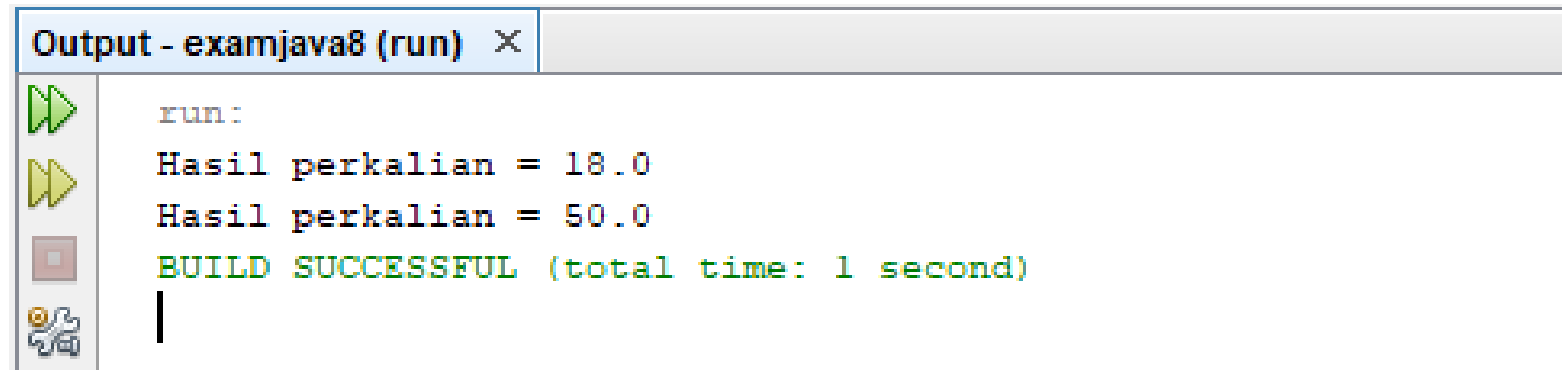
❖ Code :



```
Source History | [Icons]
10      * @author nurul
11      */
12
13      class kali
14      {
15          double a;
16          double b;
17
18          void hasil()
19          {
20              System.out.println("Hasil perkalian = " + (a*b));
21          }
22      }
```

PRAKTIKUM

❖ Output :



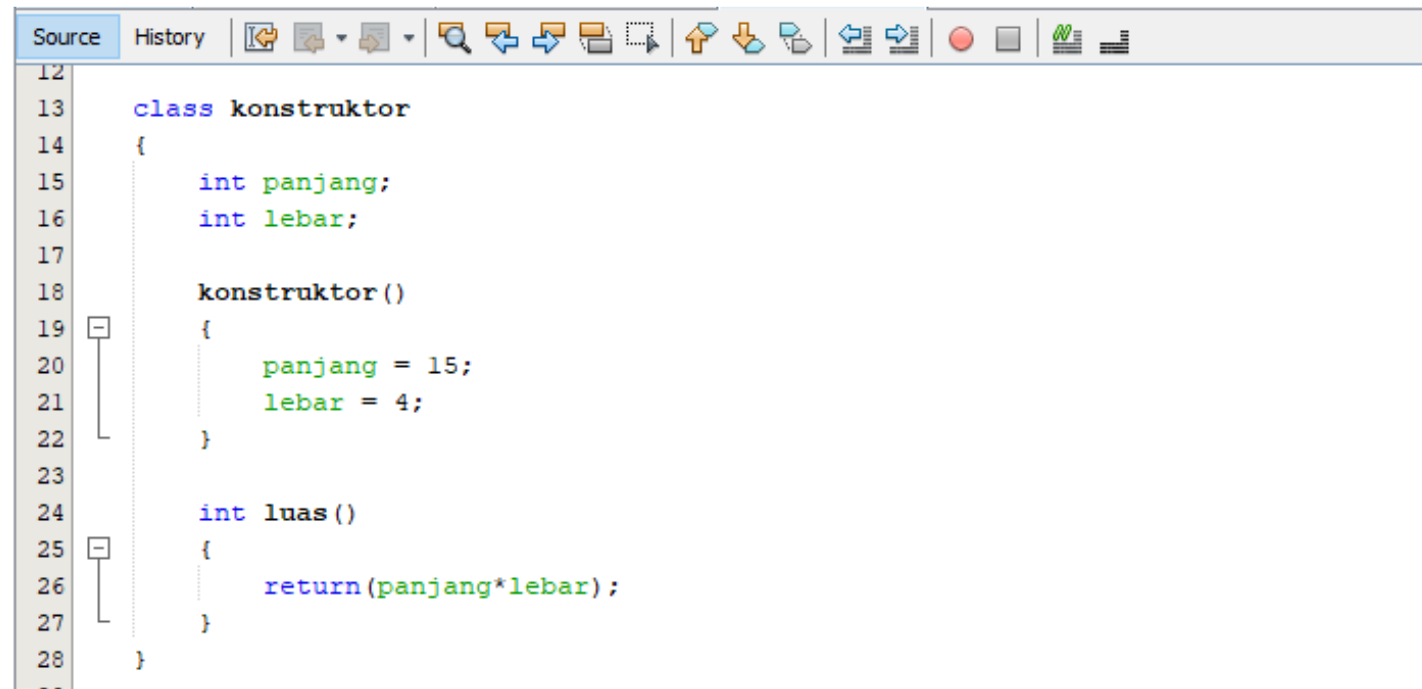
The screenshot shows an IDE output window titled "Output - examjava8 (run)". On the left side of the window, there is a vertical toolbar with four icons: a green double arrow pointing right, a yellow double arrow pointing right, a red square, and a gear icon. The main area of the window contains the following text:

```
run:  
Hasil perkalian = 18.0  
Hasil perkalian = 50.0  
BUILD SUCCESSFUL (total time: 1 second)  
|
```

PRAKTIKUM

5. Membuat *Constructor*

❖ Code :

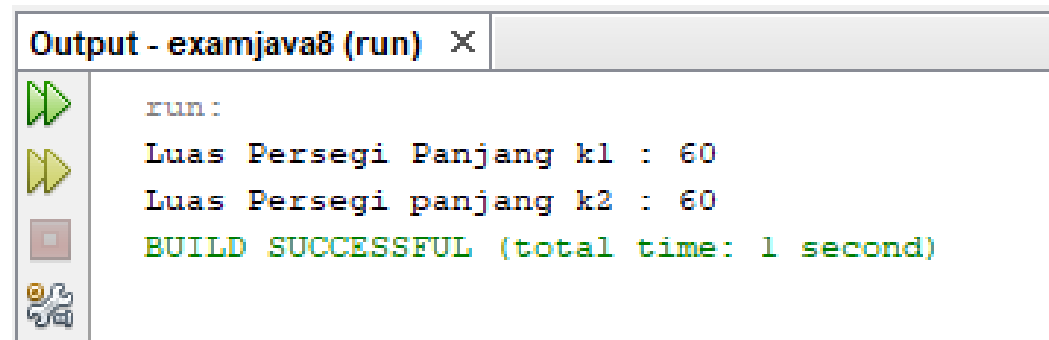


```
12
13  class konstruktor
14  {
15      int panjang;
16      int lebar;
17
18      konstruktor()
19      {
20          panjang = 15;
21          lebar = 4;
22      }
23
24      int luas()
25      {
26          return (panjang*lebar);
27      }
28  }
```

PRAKTIKUM

```
30 public class hitung
31 {
32     public static void main(String[] args)
33     {
34         konstruktor k1, k2;
35         k1 = new konstruktor();
36         k2 = new konstruktor();
37
38         System.out.println("Luas Persegi Panjang k1 : " + k1.luas());
39         System.out.println("Luas Persegi panjang k2 : " + k2.luas());
40     }
41 }
```

❖ Output



```
run:
Luas Persegi Panjang k1 : 60
Luas Persegi panjang k2 : 60
BUILD SUCCESSFUL (total time: 1 second)
```

TUGAS

1. Dengan menerapkan object dan method, buatlah program java menghitung :
 - a. Luas isi tabung
 - b. Luas persegi panjang
 - c. Luas trapesium
2. Buatlah program prosedural dan Program Berorientasi Objek (OOP) yang dapat menghitung volume Tabung. Tunjukkan dimana letak perbedaan pemrograman prosedural dan OOP.