

LAPORAN PRAKTIKUM KE – 1
PENGENALAN VARIABLE, TIPE DATA, OPERATOR DAN
METODE INPUT DI JAVA



Oleh :
Dimas Rizqi Agung Pratama (E31201887)
Semester 1

PROGRAM STUDI MANAJEMEN INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI JEMBER
2020

I. DASAR TEORI

1. Variabel Java

Variabel merupakan tempat atau wadah untuk menyimpan nilai / value pada bahasa pemrograman. Pada Pemrograman java, semua variabel harus dideklarasikan sebelum mereka dapat digunakan. Bentuk dasar dari sebuah deklarasi variabel yang ditampilkan di sini:

```
type identifier [ = value]
```

Jenis ini merupakan salah satu tipe data Jawa. Identifier adalah nama variabel. Menyatakan lebih dari satu variabel dari jenis tertentu, menggunakan daftar dipisahkan koma. Berikut adalah beberapa contoh deklarasi variabel dari berbagai jenis. Perhatikan bahwa beberapa mencakup inisialisasi.

```
int a, b, c;           // deklarasi 3 variabel a,b,c bertipe integer.
long d = 3, e, f = 5; // deklarasi 3 variabel d,e,f bertipe long
byte z = 22;          // deklarasi dan inisialisasi variable z.
char x = 'x';         // variable x yang diberikan value / nilai x.

String nama = "adhi"; // variable nama yang diberikan value / nilai
```

2. Metode Input Java

Pada pemrograman java tentunya memiliki metode input yang digunakan sebagai tempat menangkap value atau nilai pada variabel sehingga mampu menjadikan project anda bersifat dinamis, dalam pemrograman java memiliki banyak metode input namun diantara lain sebagai berikut :

- BufferedReade
- JoptionPane
- Scanner

3. Operator Java

Java menyediakan banyak set operator untuk memanipulasi variabel. Kita dapat membagi semua operator Java ke dalam kelompok berikut:

- Arithmetic Operators

Operator	Hasil
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Modulus
++	Increment
--	Decrement
+=	Persamaan penjumlahan
-=	Persamaan pengurangan

- **Relational Operators**

Operator	Penggunaan	Keterangan
>	op1 > op2	op1 lebih besar dari op2
>=	op1 >= op2	op1 lebih besar dari atau sama dengan op2
<	op1 < op2	op1 kurang dari op2
<=	op1 <= op2	op1 kurang dari atau sama dengan op2
==	op1 == op2	op1 sama dengan op2
!=	op1 != op2	op1 tidak sama dengan op2

- **Logical Operators**

Operator	Keterangan
&&	Operasi AND
	Operasi OR
^	Operasi XOR (exclusive OR)
!	Operasi NOT (negasi)

KEGIATAN PRAKTIKUM

1. Variabel

Berikut merupakan contoh project dari variabel yang sudah di jelaskan.

```

5 package hello.world.app;
6
7 /**
8  *
9  * @author SERVER
10 */
11 public class HelloWorldApp {
12
13     /**
14      * @param args the command line arguments
15      */
16     public static void main(String[] args) {
17         // TODO code application logic here
18         String nama = "Adhi Panca";
19         int nim = 11101420 ;
20         System.out.println("Nama Saya: "+nama);
21         System.out.println("Nim Saya: "+nim);
22     }
23 }
24

```

Output - Hello World App (run)

```

run:
Nama Saya: Adhi Panca
Nim Saya: 11101420
BUILD SUCCESSFUL (total time: 1 second)

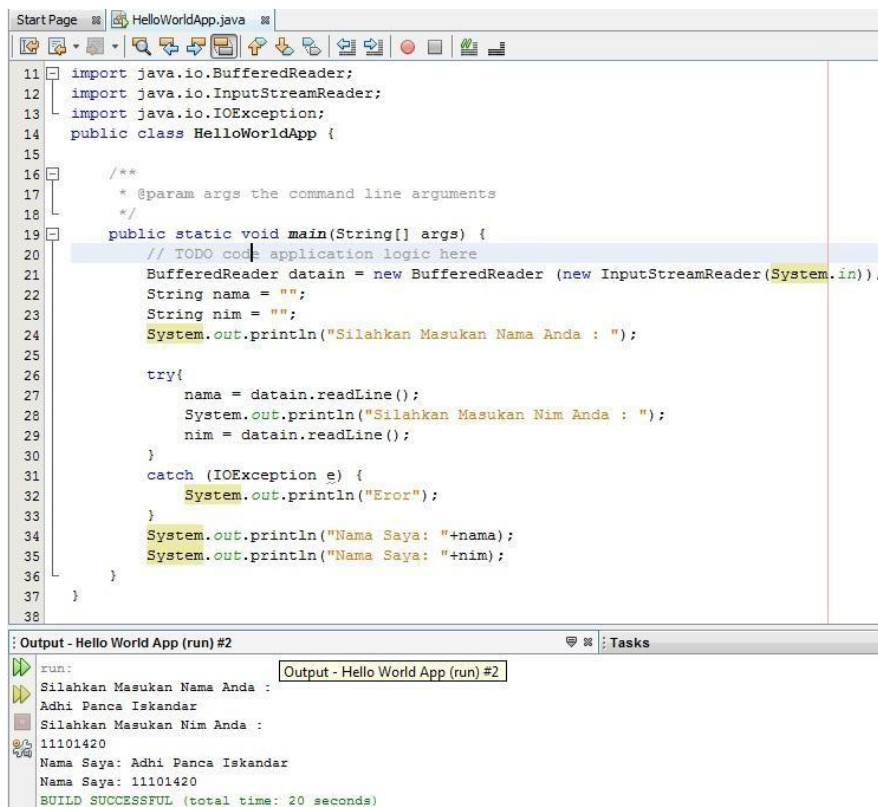
```

2. Metode Input BufferedReader

Pada bufferedReader memerlukan file bawaan java dimana kita harus mengimport file-file:

- import java.io.BufferedReader;
- import java.io.InputStreamReader;
- import java.io.IOException;

di luar main class atau pada baris 1,2,dan 3. Dan pada BufferedReader memiliki variabel dengan tipe Buffer yaitu Try dan catch yang berfungsi untuk menangkap value / nilai dari variabel utama. Seperti pada contoh berikut:



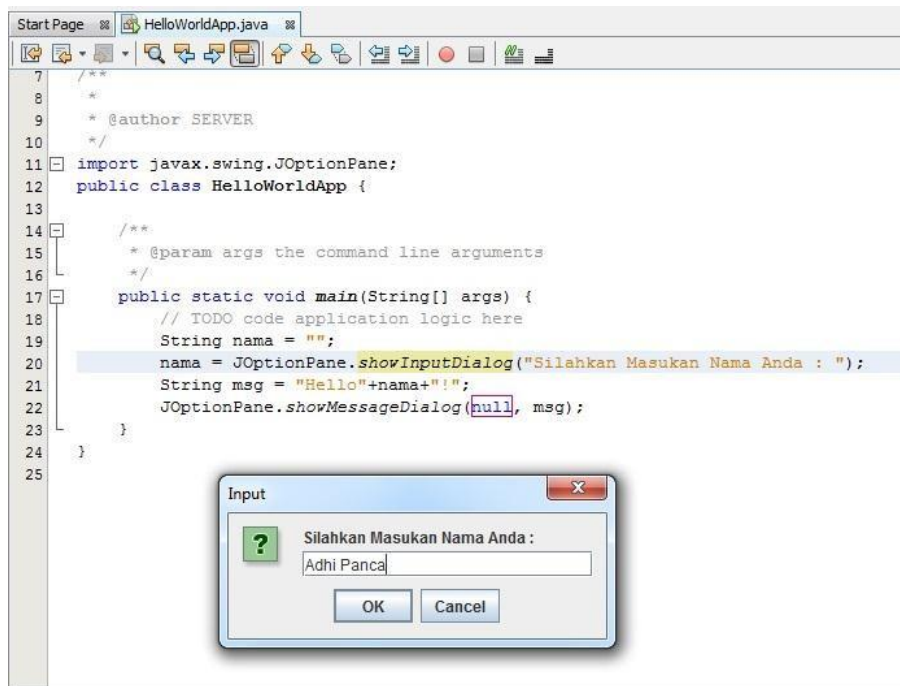
```
11 import java.io.BufferedReader;
12 import java.io.InputStreamReader;
13 import java.io.IOException;
14 public class HelloWorldApp {
15
16     /**
17      * @param args the command line arguments
18      */
19     public static void main(String[] args) {
20         // TODO code application logic here
21         BufferedReader datain = new BufferedReader (new InputStreamReader(System.in));
22         String nama = "";
23         String nim = "";
24         System.out.println("Silahkan Masukkan Nama Anda : ");
25
26         try{
27             nama = datain.readLine();
28             System.out.println("Silahkan Masukkan Nim Anda : ");
29             nim = datain.readLine();
30         }
31         catch (IOException e) {
32             System.out.println("Error");
33         }
34         System.out.println("Nama Saya: "+nama);
35         System.out.println("Nama Saya: "+nim);
36     }
37 }
38
```

Output - Hello World App (run) #2

```
run:
Silahkan Masukkan Nama Anda :
Adhi Panca Iskandar
Silahkan Masukkan Nim Anda :
11101420
Nama Saya: Adhi Panca Iskandar
Nama Saya: 11101420
BUILD SUCCESSFUL (total time: 20 seconds)
```

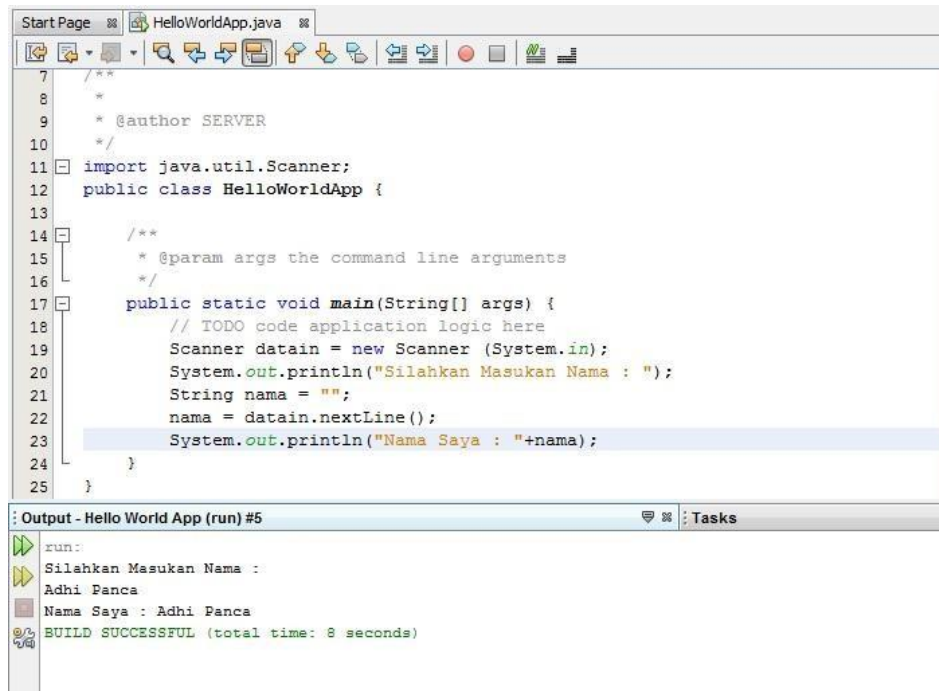
3. Metode Input JOptionPane

Pada JOptionPane memerlukan file bawaan java dimana kita harus mengimport file javax.swing.JOptionPane; tersebut diluar main class atau pada baris pertama. Seperti pada contoh beriku



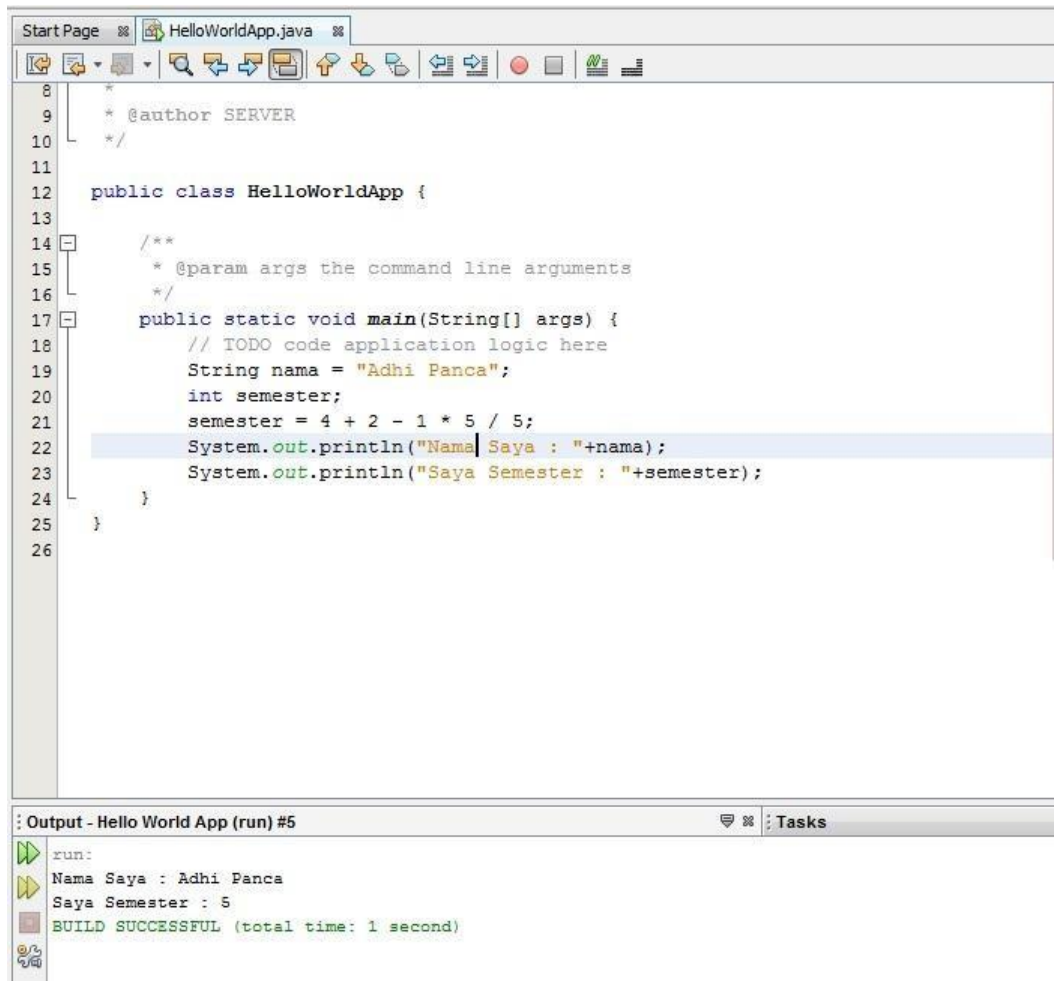
4. Metode Input Scanner

Pada Scanner memerlukan file bawaan java dimana kita harus mengimport file java.util.Scanner; tersebut diluar main class atau pada baris pertama. Seperti pada contoh berikut :



5. Operator Pada Java

Pada contoh Operator Java kita gunakan operator aritmatika, karena operator relational dan operator logika biasa digunakan pada sebuah kondisi atau perulangan, maka untuk contoh operator tersebut akan di cantumkan pada penjelasan percabangan dan perulan



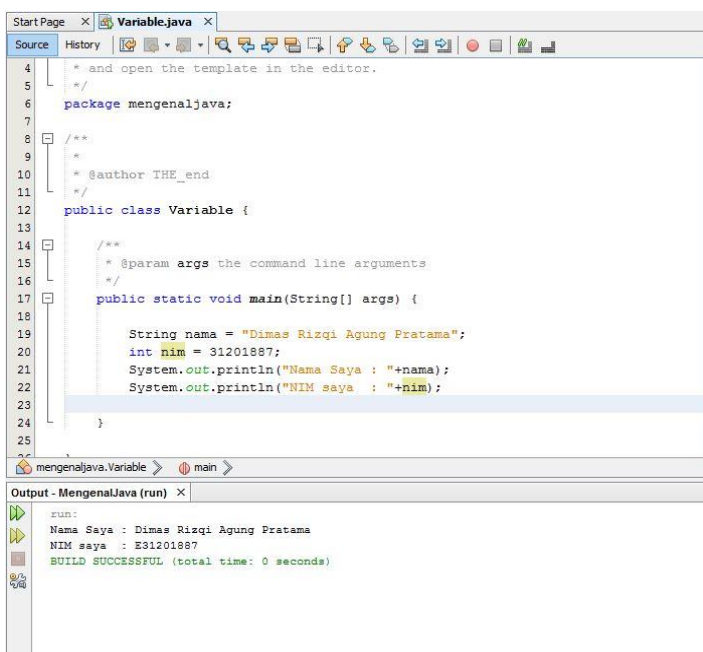
```
8  *
9  * @author SERVER
10 */
11
12 public class HelloWorldApp {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         // TODO code application logic here
19         String nama = "Adhi Panca";
20         int semester;
21         semester = 4 + 2 - 1 * 5 / 5;
22         System.out.println("Nama Saya : "+nama);
23         System.out.println("Saya Semester : "+semester);
24     }
25 }
26
```

Output - Hello World App (run) #5

```
run:
Nama Saya : Adhi Panca
Saya Semester : 5
BUILD SUCCESSFUL (total time: 1 second)
```

II. HASIL KERJA PRAKTIKUM

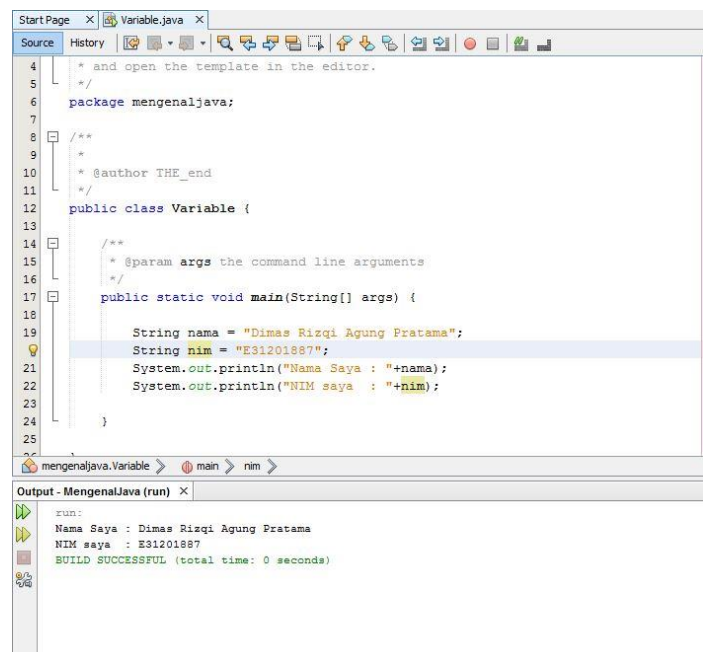
1. Variable



```
4  * and open the template in the editor.
5  */
6  package mengenaljava;
7
8  /**
9   *
10  * @author THE_end
11  */
12  public class Variable {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18
19          String nama = "Dimas Rizqi Agung Pratama";
20          int nim = 31201887;
21          System.out.println("Nama Saya : "+nama);
22          System.out.println("NIM saya : "+nim);
23      }
24  }
25
```

Output - MengenalJava (run)

```
run:
Nama Saya : Dimas Rizqi Agung Pratama
NIM saya : E31201887
BUILD SUCCESSFUL (total time: 0 seconds)
```

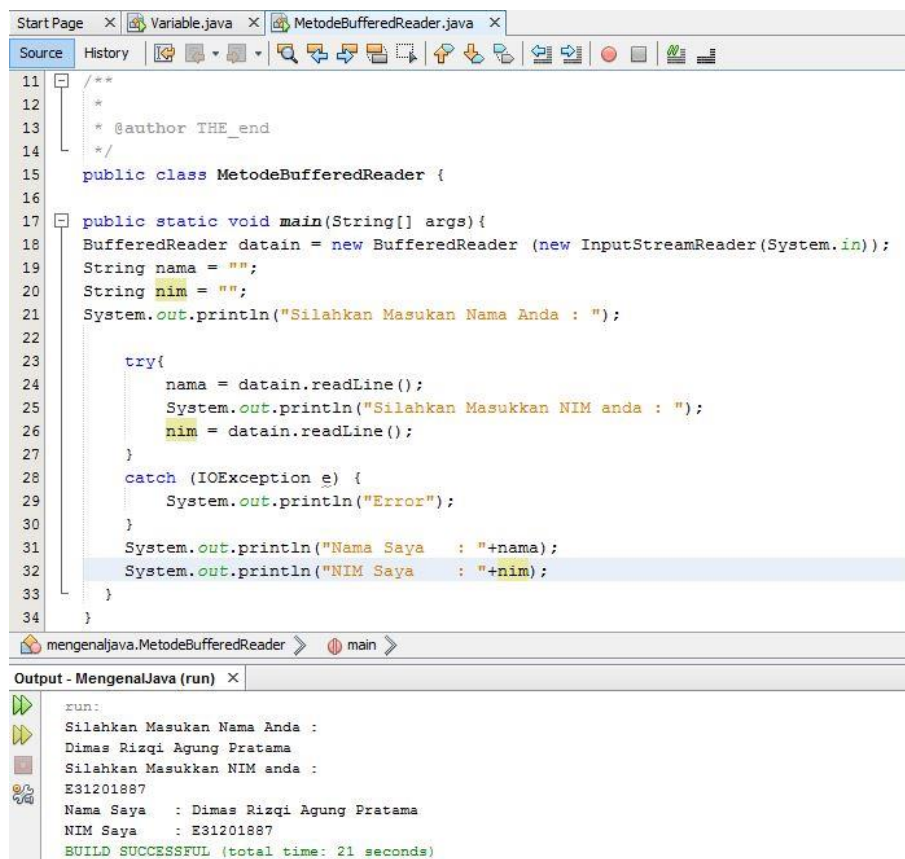


```
4  * and open the template in the editor.
5  */
6  package mengenaljava;
7
8  /**
9   *
10  * @author THE_end
11  */
12  public class Variable {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18
19          String nama = "Dimas Rizqi Agung Pratama";
20          String nim = "E31201887";
21          System.out.println("Nama Saya : "+nama);
22          System.out.println("NIM saya : "+nim);
23      }
24  }
25
```

Output - MengenalJava (run)

```
run:
Nama Saya : Dimas Rizqi Agung Pratama
NIM saya : E31201887
BUILD SUCCESSFUL (total time: 0 seconds)
```


2. Metode Input BufferedReader



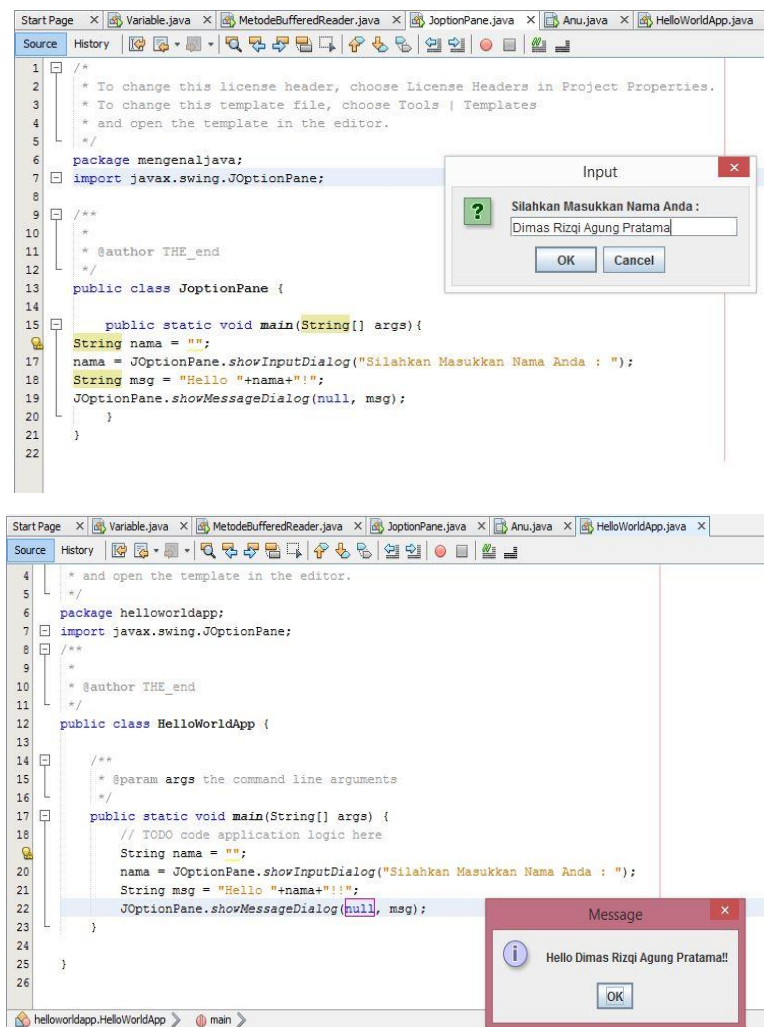
The screenshot shows an IDE with two tabs: `Variable.java` and `MetodeBufferedReader.java`. The `MetodeBufferedReader.java` file contains the following code:

```
11 /**
12  *
13  * @author THE_end
14  */
15 public class MetodeBufferedReader {
16
17     public static void main(String[] args){
18         BufferedReader datain = new BufferedReader (new InputStreamReader(System.in));
19         String nama = "";
20         String nim = "";
21         System.out.println("Silahkan Masukkan Nama Anda : ");
22
23         try{
24             nama = datain.readLine();
25             System.out.println("Silahkan Masukkan NIM anda : ");
26             nim = datain.readLine();
27         }
28         catch (IOException e) {
29             System.out.println("Error");
30         }
31         System.out.println("Nama Saya : "+nama);
32         System.out.println("NIM Saya : "+nim);
33     }
34 }
```

The output window shows the execution results:

```
run:
Silahkan Masukkan Nama Anda :
Dimas Rizqi Agung Pratama
Silahkan Masukkan NIM anda :
E31201887
Nama Saya : Dimas Rizqi Agung Pratama
NIM Saya : E31201887
BUILD SUCCESSFUL (total time: 21 seconds)
```

3. Metode Input JOptionPane



The first screenshot shows the `JOptionPane.java` file with the following code:

```
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 mengenaljava;
7 import javax.swing.JOptionPane;
8
9 /**
10  *
11  * @author THE_end
12  */
13 public class JOptionPane {
14
15     public static void main(String[] args){
16         String nama = "";
17         nama = JOptionPane.showInputDialog("Silahkan Masukkan Nama Anda : ");
18         String msg = "Hello "+nama+"!";
19         JOptionPane.showMessageDialog(null, msg);
20     }
21 }
22
```

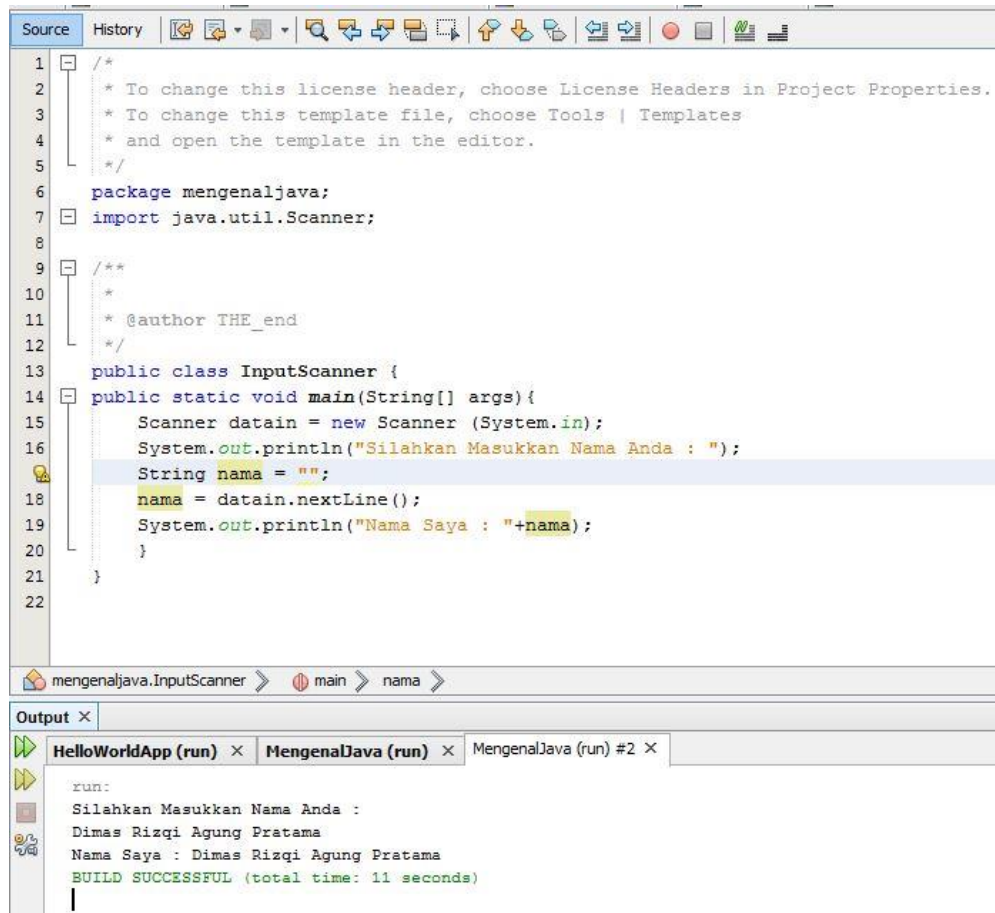
An `Input` dialog box is shown with the title "Input" and the message "Silahkan Masukkan Nama Anda :". The input field contains "Dimas Rizqi Agung Pratama".

The second screenshot shows the `HelloWorldApp.java` file with the following code:

```
4 /**
5  * and open the template in the editor.
6  */
7 package helloworldapp;
8 import javax.swing.JOptionPane;
9
10 /**
11  * @author THE_end
12  */
13 public class HelloWorldApp {
14
15     /**
16      * @param args the command line arguments
17      */
18     public static void main(String[] args) {
19         // TODO code application logic here
20         String nama = "";
21         nama = JOptionPane.showInputDialog("Silahkan Masukkan Nama Anda : ");
22         String msg = "Hello "+nama+"!";
23         JOptionPane.showMessageDialog(null, msg);
24     }
25 }
26
```

A `Message` dialog box is shown with the title "Message" and the message "Hello Dimas Rizqi Agung Pratama!!".

4. Metode Input Scanner



The screenshot shows an IDE with the source code for `mengenajava.InputScanner`. The code includes a package declaration, an import for `java.util.Scanner`, and a class `InputScanner` with a `main` method. The `main` method uses `Scanner` to read input from `System.in`. The output window shows the program's execution, displaying the prompt "Silahkan Masukkan Nama Anda :", the user input "Dimas Rizqi Agung Pratama", and the output "Nama Saya : Dimas Rizqi Agung Pratama". The build is successful.

```
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 mengenajava;
7  import java.util.Scanner;
8
9  /**
10   *
11   * @author THE_end
12   */
13  public class InputScanner {
14      public static void main(String[] args){
15          Scanner datain = new Scanner (System.in);
16          System.out.println("Silahkan Masukkan Nama Anda : ");
17          String nama = "";
18          nama = datain.nextLine();
19          System.out.println("Nama Saya : "+nama);
20      }
21  }
```

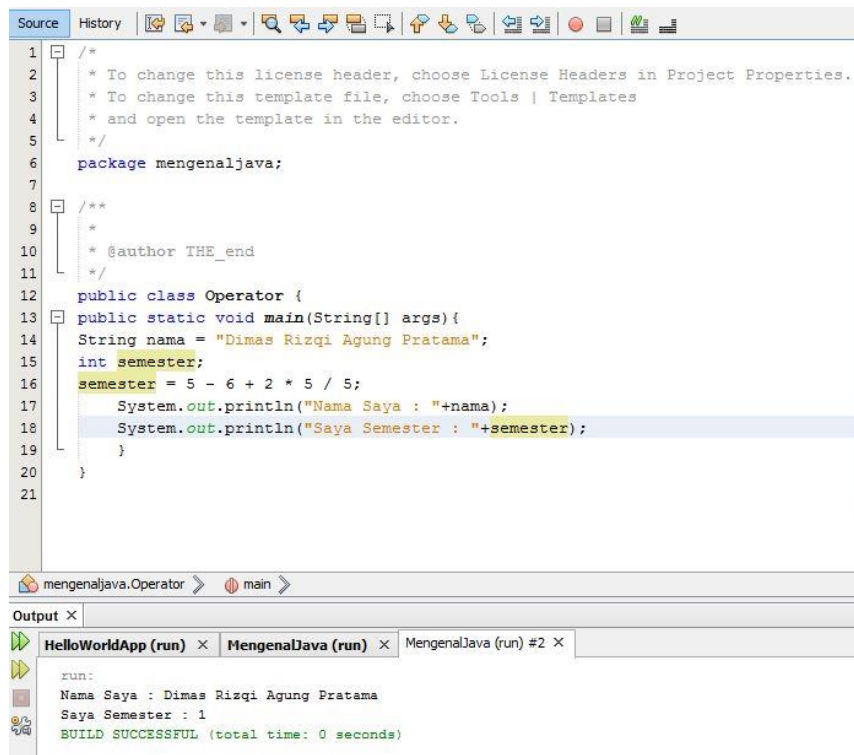
mengenajava.InputScanner > main > nama >

Output x

HelloWorldApp (run) x MengenalJava (run) x MengenalJava (run) #2 x

run:
Silahkan Masukkan Nama Anda :
Dimas Rizqi Agung Pratama
Nama Saya : Dimas Rizqi Agung Pratama
BUILD SUCCESSFUL (total time: 11 seconds)

5. Operator



The screenshot shows an IDE with the source code for `mengenajava.Operator`. The code includes a package declaration and a class `Operator` with a `main` method. The `main` method declares a `String` variable `nama` and an `int` variable `semester`, then prints their values. The output window shows the program's execution, displaying "Nama Saya : Dimas Rizqi Agung Pratama" and "Saya Semester : 1". The build is successful.

```
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 mengenajava;
7
8  /**
9   *
10   * @author THE_end
11   */
12  public class Operator {
13      public static void main(String[] args){
14          String nama = "Dimas Rizqi Agung Pratama";
15          int semester;
16          semester = 5 - 6 + 2 * 5 / 5;
17          System.out.println("Nama Saya : "+nama);
18          System.out.println("Saya Semester : "+semester);
19      }
20  }
```

mengenajava.Operator > main >

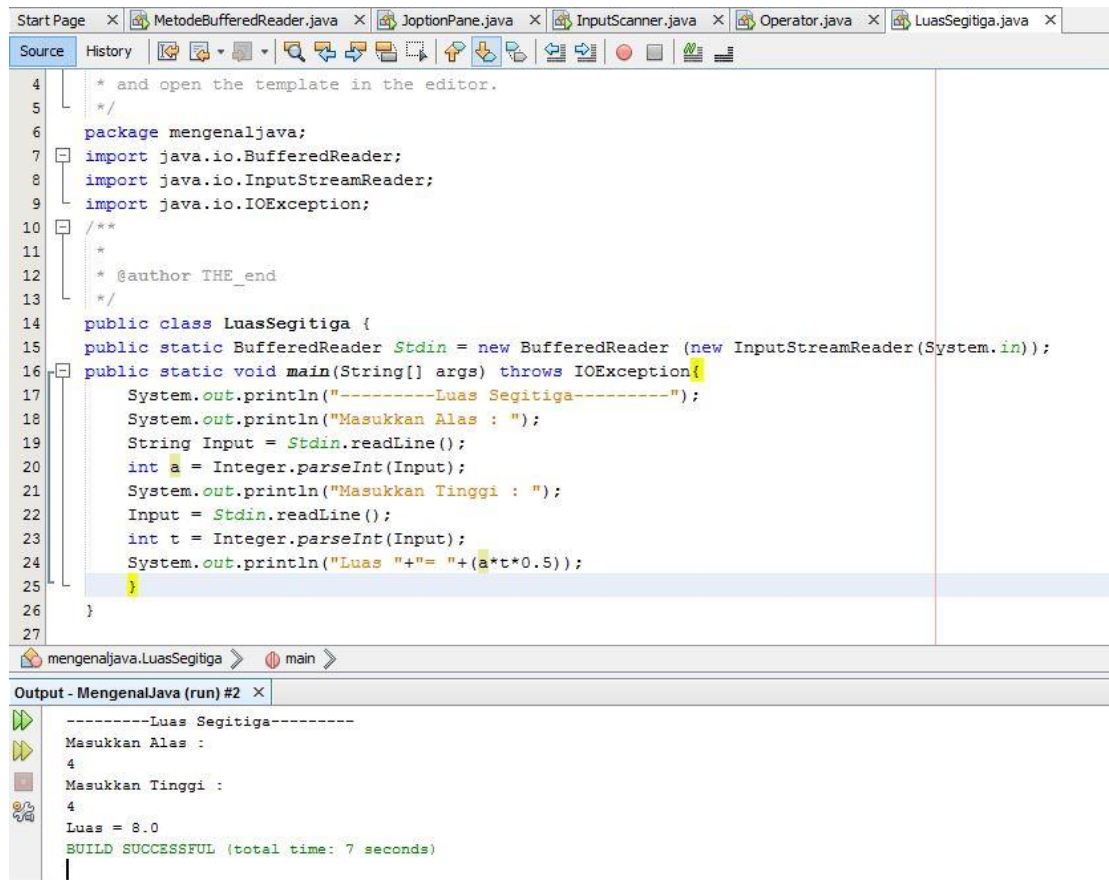
Output x

HelloWorldApp (run) x MengenalJava (run) x MengenalJava (run) #2 x

run:
Nama Saya : Dimas Rizqi Agung Pratama
Saya Semester : 1
BUILD SUCCESSFUL (total time: 0 seconds)

III. TUGAS

- 1) Buatlah sebuah program untuk menghitung luas segitiga dengan metode BufferedReader



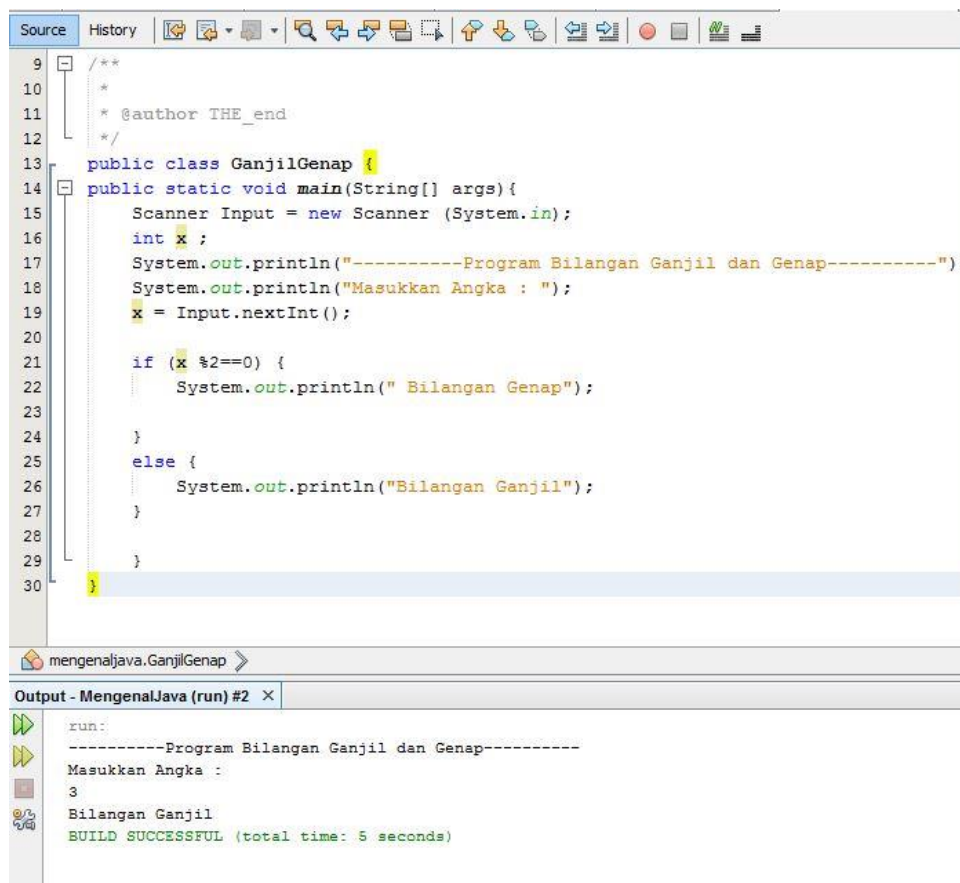
```
4  * and open the template in the editor.
5  */
6  package mengenaljava;
7  import java.io.BufferedReader;
8  import java.io.InputStreamReader;
9  import java.io.IOException;
10 /**
11  *
12  * @author THE_end
13  */
14 public class LuasSegitiga {
15     public static BufferedReader Stdin = new BufferedReader (new InputStreamReader(System.in));
16     public static void main(String[] args) throws IOException{
17         System.out.println("-----Luas Segitiga-----");
18         System.out.println("Masukkan Alas : ");
19         String Input = Stdin.readLine();
20         int a = Integer.parseInt(Input);
21         System.out.println("Masukkan Tinggi : ");
22         Input = Stdin.readLine();
23         int t = Integer.parseInt(Input);
24         System.out.println("Luas "+ " = "+ (a*t*0.5));
25     }
26 }
27
```

mengenaljava.LuasSegitiga > main >

Output - MengenalJava (run) #2 >

```
-----Luas Segitiga-----
Masukkan Alas :
4
Masukkan Tinggi :
4
Luas = 8.0
BUILD SUCCESSFUL (total time: 7 seconds)
```

- 2) Buatlah sebuah program untuk menampilkan bilangan genap dan ganjil dengan menggunakan metode Scanner.



```
9  /**
10  *
11  * @author THE_end
12  */
13 public class GanjilGenap {
14     public static void main(String[] args){
15         Scanner Input = new Scanner (System.in);
16         int x ;
17         System.out.println("-----Program Bilangan Ganjil dan Genap-----");
18         System.out.println("Masukkan Angka : ");
19         x = Input.nextInt();
20
21         if (x %2==0) {
22             System.out.println(" Bilangan Genap");
23         }
24         else {
25             System.out.println("Bilangan Ganjil");
26         }
27     }
28 }
29
30
```

mengenaljava.GanjilGenap >

Output - MengenalJava (run) #2 >

```
run:
-----Program Bilangan Ganjil dan Genap-----
Masukkan Angka :
3
Bilangan Ganjil
BUILD SUCCESSFUL (total time: 5 seconds)
```

IV. ANALISA DAN KESIMPULAN

Kesimpulannya adalah metode scanner lebih mudah digunakan daripada menggunakan metode BufferedReader. Karena, metode scanner hanya memasukkan syntax scanner satu kali saja dibandingkan dengan menggunakan metode bufferedreader yang memasukkan tiga kali.

V. SUMBER

Modul 2, Pengenalan Variable, Tipe data, Operator dan metode input di JAVA