

## Acara 5

**Pokok Bahasan : Variabel dan Tipe Data**

**Acara Praktikum/Praktek: Minggu 2/1**

**Tempat : Politeknik Negeri Jember**

**Alokasi Waktu : 100 menit**

### **a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa dapat memahami tentang Variabel dan tipe data pada pemrograman Java
2. Mahasiswa dapat mendeklarasikan variabel dan tipe data pada pemrograman Java

### **b. Indikator**

Kemampuan mahasiswa dalam memahami variabel

### **c. Dasar Teori**

#### **Dasar pemrograman Java dan Variabel/Identifier**

1. **Pernyataan** adalah satu atau lebih baris kode yang diakhiri dengan semicolon. Contoh untuk pernyataan tunggal :

```
System.out.println("Hello world");
```

2. **Blok** adalah satu atau lebih pernyataan yang terbentang antara kurung kurawal buka dan kurung kurawal tutup yaitu sekumpulan pernyataan sebagai satu unit kesatuan. Contoh dari suatu blok adalah :

```
public static void main( String[] args ){  
    System.out.println("Hello");  
    System.out.println("world");  
}
```

3. **Variabel** adalah unit dasar (tempat) penyimpanan di program Java. Variabel didefinisikan menggunakan kombinasi *identifier*, *type* dan juga sekaligus diinisialisasi. Di dalam Java, semua variabel harus dideklarasikan sebelum digunakan. Deklarasi variabel adalah sebuah perintah agar komputer menyediakan variabel yang akan kita pakai. Pemndeklarasian variabel harus diawali dengan tipe variabel, nama variabel kemudian diakhiri dengan titik koma ; .

Deklarasi dan Inisialisasi Variabel untuk deklarasi variabel adalah sebagai berikut :

```
<data tipe> <name> [=initial value]
```

Dalam memberi nama variabel ada beberapa aturan yang harus dipenuhi, antara lain:

- a. Diawali dengan huruf atau karakter underscore \_
- b. Nama variabel bersifat case sensitive (huruf besar dianggap berbeda dengan huruf kecil), Hal ini berarti bahwa Identifier :  
Hello tidak sama dengan hello.
- c. Tidak boleh menggunakan spasi (sebagai pengganti dapat digunakan tanda underscore atau penggunaan huruf besar pada awal kata berikutnya)
- d. Tidak boleh menggunakan karakter khusus seperti <, >, \*, ?, = dan sebagainya.
- e. Apabila nama variabel lebih dari satu kata maka kata yang setelahnya diawali huruf besar.
- f. Identifier tidak dapat menggunakan kata kunci dalam Java seperti class, public, void, dsb.
- g. Kata kunci adalah identifier yang telah dipesan untuk didefinisikan oleh Java sebagai tujuan tertentu.
- h. Keyword tidak dapat digunakan sebagai nama variabel, class, method, dsb.
- i. Berikut ini adalah daftar dari kata kunci dalam Java (Java Keywords).

**Tabel: Daftar Keyword dalam Java**

| Keywords in Java |         |            |              |           |
|------------------|---------|------------|--------------|-----------|
| abstract         | default | if         | private      | this      |
| assert           | do      | implements | protected    | throw     |
| boolean          | double  | import     | public       | throws    |
| break            | else    | instanceof | return       | transient |
| byte             | enum    | int        | short        | try       |
| case             | extends | interface  | static       | void      |
| catch            | final   | long       | strictfp     | volatile  |
| char             | finally | native     | super        | while     |
| class            | float   | new        | switch       |           |
| continue         | for     | package    | synchronized |           |

- j. True, false, dan null bukan termasuk kata kunci akan tetapi mereka termasuk kata-kata khusus.

#### 4. **Tipe Data**, Tipe data Java dibagi dalam dua kategori:

**A. Tipe Data Sederhana / Primitif:** Tipe data sederhana adalah tipe data inti. Tipe sederhana tidak diturunkan dari tipe lain. Tipe ini adalah tipe primitif, tipe primitif diantaranya adalah:

- a. Tipe bilangan bulat: byte, short, int, long

| Tipe  | Deskripsi   |
|-------|---|
| Byte  | -128 s/d +127 menempati 8 bits di memori                                  |
| Short | -32768 s/d +32767 menempati 16 bits di memori                             |
| Int   | -2147483648 s/d +2147483647 menempati 32 bits di memori                   |
| Long  | -9223372036854775808 s/d +9223372036854775807 menempati 64 bits di memori |

- b. Tipe angka titik mengambang (floating point): float dan double

| Tipe   | Deskripsi   |
|--------|---|
| Float  | $-3.4 \times 10^8$ s/d $+3.4 \times 10^8$         |
| Double | $-1.7 \times 10^{308}$ s/d $+1.7 \times 10^{308}$ |

- c. CharChar adalah karakter tunggal yang pendefinisian di awal dan akhir menggunakan tanda petik tunggal ( ' ). Tipe char mengikuti aturan Unicode, sehingga bisa dapat menggunakan kode untuk kemudian diikuti bilangan dari 0 sampai 65535, tetapi yang biasa digunakan adalah bilangan heksadesimal dari 0000 sampai FFFF.
- d. Boolean Tipe data Boolean terdiri dari dua nilai saja, yaitu true dan false. Boolean sangat penting untuk mengevaluasi suatu kondisi.

**Tabel: Tipe data**

| Nama Type | Nilai                      | Alokasi memori | Jangkauan ukuran  |
|-----------|----------------------------|----------------|---|
| boolean   | true or false              | 1 byte         | not applicable  |
| char      | single character (unicode) | 2 bytes        | all unicodes characters   |
| byte      | integer                    | 1 byte         | -128 to 127   |
| short     | integer                    | 2 bytes        | -32768 to 32767   |
| int       | integer                    | 4 bytes        | -2147483648 to 2147483647   |
| long      | integer                    | 8 bytes        | -9223372036854775808 to 9223372036854775807   |
| float     | floating-point number      | 4 bytes        | $-3.40282347 \times 10^{38}$ to $1.40239846 \times 10^{-46}$                            |
| double    | floating-point number      | 8 bytes        | $\pm 1.76769313486231570 \times 10^{308}$ to $\pm 4.94065645841246544 \times 10^{-324}$ |

Tipe data dalam Java

**B. Tipe Data Komposit**, Tipe data komposit disusun dari tipe data sederhana. Tipe ini diantaranya adalah *string*, *array*, *class* dan *interface*.

**5. Casting Tipe Data**, Casting seringkali diperlukan ketika fungsi mengirim tipe berbeda dengan tipe yang diperlukan operasi. Tindakan casting dilakukan dengan dengan

menempatkan tipe yang diharapkan di dalam tanda pasangan kurung di sebelah kiri nilai yang ingin dikonversi

**d. Alat dan Bahan**

1. PC/Laptop
2. Text Editor (IntelliJ, Netbeans, vscode, Notepad++, dan lainnya)

**e. Hasil dan Pembahasan**

1. Rangkuman tentang variabel dan tipe data dalam pemrograman dasar
2. Rangkuman berupa file .doc dikumpulkan pada <http://jti.polije.ac.id/elearning>

**f. Kesimpulan**

Mahasiswa memahami dan mengimplementasi variable dan tipe data pada Java.

**g. Rubrik Penilaian**

| No          | Indikator  | Skor* |   |   |   |
|-------------|--|-------|---|---|---|
| 1           | Ketepatan waktu dan kelengkapan dalam mengumpulkan tugas yang ditunjang dengan referensi | 1     | 2 | 3 | 4 |
| 2           | Ketepatan waktu dalam mengumpulkan tugas dengan tidak dilengkapi referensi               | 1     | 2 | 3 | 4 |
| 3           | Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan tugas                         | 1     | 2 | 3 | 4 |
| 4           | Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan tugas               | 1     | 2 | 3 | 4 |
| Jumlah skor |  |       |   |   |   |

## Acara 6

**Pokok Bahasan : Operator**

**Acara Praktikum/Praktek: Minggu 2/2**

**Tempat : Politeknik Negeri Jember**

**Alokasi Waktu : 100 menit**

### a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa dapat memahami tentang Operator pada pemrograman Java
2. Mahasiswa dapat mendeklarasikan Operator pada pemrograman Java

### b. Indikator

Kemampuan mahasiswa dalam memahami operator pada pemrograman Ja

### c. Dasar Teori

Setiap instruksi dalam kode program java disebut juga dengan statement. Setiap statement dalam bahasa Java diakhiri dengan tanda ; titik koma . Beberapa statement dapat dikelompokkan ke dalam grup atau blok yang berbeda. Caranya, digunakan tanda kurung kurawal untuk mengawali dan mengakhiri grup.

- 1. Operator,** Operator merupakan simbol yang biasa digunakan dalam menulis suatu pernyataan (statement) dalam bahasa pemrograman apapun. Operator akan melakukan suatu operasi terhadap operand sesuai dengan fungsinya. Jenis-jenis operator dapat kita lihat dalam tabel dibawah ini

| Jenis Operator                  | Simbol               |
|---------------------------------|----------------------|
| Increment/decrement             | ++, -                |
| Unary operators                 | +, -, ~, !, (cast)   |
| Perkalian/pembagian/modulus     | *, /, %              |
| Penambahan/pengurangan          | +, -                 |
| Operator shift                  | <<, >>, >>>          |
| Perbandingan                    | <, <=, >, >=, =, !=  |
| Bitwise                         | AND, OR, XOR &,  , ^ |
| Kondisional                     | AND, OR &&,          |
| Operator Ternary                | ? :                  |
| Operator penugasan (assignment) | =                    |

- 1) Operator Aritmatika,** Arithmetic operator (operator aritmatika) adalah operator yang berfungsi untuk operasi aritmatika.

| Nama        | Simbol | Contoh |
|-------------|--------|--------|
| Penjumlahan | +      | a + b  |
| Pengurangan | -      | a - b  |
| Perkalian   | *      | a * b  |

|           |   |       |
|-----------|---|-------|
| Pembagian | / | a / b |
| Sisa Bagi | % | a % b |

#### Contoh Penerapan

```
int a = 6, b = 3;
System.out.println(a+b); //hasilnya 9
System.out.println(a-b); //hasilnya 3
System.out.println(a*b); //hasilnya 18
System.out.println(a/b); //hasilnya 2
System.out.println(a%b); //hasilnya 0
```

**2) Operator Assignment,** Operator *assignment* atau penugasan dilakukan menggunakan operator "=". Sintaks yang digunakan adalah

```
identifier = ekspresi;
identifier = literal;
```

Meskipun mirip, operasi assignment berbeda dengan operasi "sama dengan" dalam Matematika. Berikut contohnya:

```
a = a + 1;
```

Statement diatas bukanlah dalam artian matematika, bukan memiliki arti a sama dengan a ditambah 1. Statement tersebut berarti tambahkan nilai a dengan angka 1 lalu jadikan nilai tersebut menjadi nilai a yang baru.

a+1 → a

| Nama                      | Simbol | Contoh |
|---------------------------|--------|--------|
| Pengisian Nilai           | =      | a = b  |
| Pengisian dan Penambahan  | +=     | a += b |
| Pengisian dan Pengurangan | -=     | a -= b |
| Pengisian dan Perkalian   | *=     | a *= b |
| Pengisian dan Pembagian   | /=     | a /= b |
| Pengisian dan Sisa bagi   | %=     | a %= b |

Contoh pengaplikasian:

```
int a, b;
a = 6;
b = 3;
System.out.println(a); //hasilnya 6
a += b;
System.out.println(a); //hasilnya 9
a -= b;
System.out.println(a); //hasilnya 6
a *= b;
System.out.println(a); //hasilnya 18
a /= b;
System.out.println(a); //hasilnya 6
a %= b;
System.out.println(a); //hasilnya 0
```

- 3) Operator Relasional,** Untuk membandingkan 2 nilai (variabel) atau lebih digunakan operator relasional, dimana operator ini akan mengembalikan atau menghasilkan nilai *True* atau *False*.

**Tabel : Operator Relasional**

| Operator | Keterangan                   |
|----------|------------------------------|
| ==       | Sama dengan                  |
| !=       | Tidak sama dengan            |
| >        | Lebih besar                  |
| <        | Lebih kecil                  |
| >=       | Lebih besar atau sama dengan |
| <=       | Lebih kecil atau sama dengan |

- 4) Operator Kondisional,** Operator ini menghasilkan nilai yang sama dengan operator relasional, hanya saja penggunaannya lebih pada operasi – operasi Boolean.

**Tabel: Operator Kondisional**

| Operator | Keterangan           |
|----------|----------------------|
| &&       | Operasi AND          |
|          | Operasi OR           |
| ^        | Operasi XOR          |
| !        | Operasi NOT (Negasi) |

**Tabel: Contoh Operasi Kondisional**

| A     | B     | A&&B  | A  B  | A^B   | !A    |
|-------|-------|-------|-------|-------|-------|
| True  | True  | True  | True  | False | False |
| TRUE  | False | False | True  | True  | False |
| False | True  | False | True  | True  | True  |
| False | False | False | False | False | True  |

- 5) Operator Shift dan Bitwise,** Kedua operator ini digunakan untuk memanipulasi nilai dari bitnya, sehingga diperoleh nilai yang lain.

**Tabel Operator Shift dan Bitwise**

| Operator | Keterangan  |
|----------|---|
| &        | Operasi bitwise AND                                 |
|          | Operasi bitwise OR                                  |
| ^        | Operasi bitwise XOR                                 |
| ~        | Operasi bitwise NOT                                 |
| >>       | Operasi shift right (geser ke kanan sebanyak n bit) |
| >>>      | Operasi shift right zero fill                       |

&lt;&lt;

Operasi shift left (geser ke kiri sebanyak n bit)

**Tabel** Contoh Operasi Bitwise

| A | B | A&B | A B | A^B | ~A |
|---|---|-----|-----|-----|----|
| 1 | 1 | 1   | 1   | 0   | 0  |
| 1 | 0 | 0   | 1   | 1   | 0  |
| 0 | 1 | 0   | 1   | 1   | 1  |
| 0 | 0 | 0   | 0   | 0   | 1  |

**d. Hasil dan Pembahasan**

1. Rangkuman tentang operator dalam pemrograman dasar
2. Rangkuman berupa file .doc dikumpulkan pada <http://jti.poliije.ac.id/elearning>

**e. Kesimpulan**

Mahasiswa memahami operator dalam pemrograman Java



**f. Rubrik Penilaian**

| No          | Indikator   | Skor* |   |   |   |
|-------------|---|-------|---|---|---|
| 1           | Ketepatan waktu dan kelengkapan dalam menjelaskan tugas yang ditunjang dengan bukti gambar                      | 1     | 2 | 3 | 4 |
| 2           | Ketepatan waktu dalam mengumpulkan tugas dan 75 % langkah - langkah yang dilakukan tepat                        | 1     | 2 | 3 | 4 |
| 3           | Ketepatan waktu akan tetapi kurang tepat dalam menjelaskan tugas dan 50% langkah - langkah yang dilakukan tepat | 1     | 2 | 3 | 4 |
| 4           | Keterlambatan pengumpulan tugas dan ketidaktepatan dalam menjelaskan tugas                                      | 1     | 2 | 3 | 4 |
| Jumlah skor |   |       |   |   |   |

## Acara 7

**Pokok Bahasan : Output dan Input**

**Acara Praktikum/Praktek: Minggu 2/3**

**Tempat : Politeknik Negeri Jember**

**Alokasi Waktu : 100 menit**

### **a. Capaian Pembelajaran Mata Kuliah (CPMK)**

1. Mahasiswa mampu memahami Output dan Input pada pemrograman java

### **b. Indikator**

1. Kemampuan mahasiswa dalam memahami Output dan Input pada pemrograman Java

### **c. Dasar Teori**

**1) Output,** Setiap program pasti menghasilkan output. Salah satu cara untuk menampilkan output adalah dengan menampilkannya ke layar. Untuk menampilkan ke layar, ada beberapa cara:

- a. `System.out.print("Hello world");` Perintah ini akan menampilkan kata Hello world ke layar, atau apapun yang kita tuliskan didalam tanda petik.
- b. `System.out.println("Hello world");` Perintah ini akan menampilkan kata Hello world ke layar, atau apapun yang kita tuliskan didalam tanda petik, sekaligus memberi perintah ganti baris di akhir kata/kalimat.
- c. `System.out.println(panjang);` Perintah ini akan menampilkan isi variabel panjang ke layar. Perhatikan bahwa untuk menampilkan isi variabel, tidak perlu menggunakan tanda petik ( " ).
- d. `System.out.println("Panjang segi empat: " + panjang);` Perintah ini akan menampilkan kalimat "Panjang segi empat: " kemudian disambung dengan isi variabel panjang ke layar. Perhatikan untuk menyambung kalimat dengan isi variabel, digunakan tanda plus ( + ).

#### **Contoh:**

1. Menggunakan `System.out.print()`

```
System.out.print("Hello World!");  
System.out.print("Nama saya Sari");
```

#### **Hasil**

```
Hello World!Nama saya Sari
```

2. Menggunakan `System.out.println()`

```
System.out.println("Hello World!");  
System.out.println("Nama saya Sari");
```

#### **Hasil**

```
Hello World!  
Nama saya Sari
```

3. Menampilkan isi variabel panjang

```
int panjang = 10;  
System.out.println(panjang);
```

Hasil

```
10
```

4. Menampilkan kalimat "Panjang segi empat" dan disambung dengan isi variabel panjang

```
int panjang = 10;  
System.out.println("Panjang segi empat: " + panjang);
```

Hasil

```
Panjang segi empat: 10
```

Pada output program terdapat beberapa karakter khusus. Karakter-karakter ini dikenal dengan istilah escape sequence. Karakter-karakter yang umum digunakan dapat dilihat pada tabel berikut ini:

| Escape Sequence | Keterangan         |
|-----------------|--------------------|
| \n              | baris baru         |
| \               | Karakter backslash |
| %%              | Karakter %         |
| \t              | Karakter tab       |
| \"              | Karakter "         |

**2) Input,** Dalam membaca input dari keyboar, diperlukan sebuah library yaitu Scanner. Scanner Class Scanner merupakan class yang dapat mengambil data yang diketikkan pada keyboard untuk disimpan ke dalam variabel pada program Java. Pertama-tama kita pastikan bahwa library Scanner sudah kita sertakan. Caranya adalah dengan menuliskan perintah `import java.util.Scanner;` di baris paling atas dari kode program kita. Selanjutnya kita tuliskan perintah deklarasi scanner berikut ini didalam fungsi `main()`:

```
Scanner sc = new Scanner(System.in);
```

Selanjutnya, tergantung dari jenis input yang akan kita masukkan, berupa bilangan bulat (int), bilangan koma (float/double), atau karakter (String).

1. Jika input berupa bilangan bulat, maka perintahnya adalah: `nextInt()`;
2. Jika input berupa bilangan koma, maka perintahnya adalah: `nextFloat()`;
3. Jika input berupa teks, maka perintahnya adalah: `nextLine()`;

Contoh:

### 1. Input berupa bilangan bulat:

```
import java.util.Scanner;

class Hello {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        int usia;
        System.out.print("Masukkan usia anda: ");
        usia = sc.nextInt();
        System.out.print("Usia anda adalah: " + usia);
    }
}
```

Hasil

Masukkan usia anda: 20

Usia anda adalah 20

### 2. Input berupa dua buah bilangan:

```
import java.util.Scanner;

class Hello {
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        int panjang, lebar;
        System.out.print("Masukkan panjang: ");
        panjang = sc.nextInt();
        System.out.print("Masukkan lebar: ");
        lebar = sc.nextInt();
        System.out.print("Nilai dari panjang: " + panjang);
        System.out.print("Nilai dari lebar: " + lebar);
    }
}
```

Hasil

Masukkan panjang: 15

Masukkan lebar: 7

Nilai dari panjang: 15

Nilai dari lebar: 7

**d. Alat dan Bahan**

1. Komputer
2. Netbeans
3. JDK

**e. Hasil dan Pembahasan**

1. Laporan hasil praktikum
2. Laporan berupa file word dikumpulkan pada <http://jti.poliije.ac.id/elearning>

**f. Kesimpulan**

Mahasiswa memahami dan mampu mengimplementasikan input output pada pemrograman java netbeans.

**g. Rubrik Penilaian**

| No          | Indikator   | Skor* |   |   |   |
|-------------|---|-------|---|---|---|
| 1           | Ketepatan dalam menjalankan serta menjelaskan dari tugas            | 1     | 2 | 3 | 4 |
| 2           | Kurang dan ketepatan dalam menjalankan serta menjelaskan dari tugas | 1     | 2 | 3 | 4 |
| 3           | Kurang tepat dalam menjelaskan tugas                                | 1     | 2 | 3 | 4 |
| 4           | Keterlambatan dan ketidaktepatan dalam menjelaskan tugas            | 1     | 2 | 3 | 4 |
| Jumlah skor |   |       |   |   |   |

## Acara 8

**Pokok Bahasan : Input Output**

**Acara Praktikum/Praktek: Minggu 2/4**

**Tempat : Politeknik Negeri Jember**

**Alokasi Waktu : 100 menit**

### a. Capaian Pembelajaran Mata Kuliah (CPMK)

1. Mahasiswa mampu memahami dan mampu mengimplementasikan Input dan Output pada pemrograman java
2. Mahasiswa mampu mengimplementasikan input dan output pada Netbeans

### b. Indikator

Kemampuan mahasiswa mengimplementasikan program java sederhana

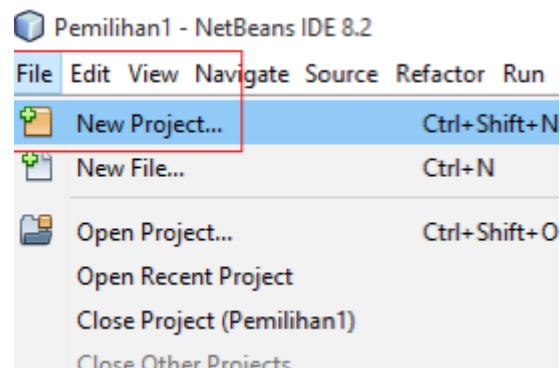
### c. Dasar Teori

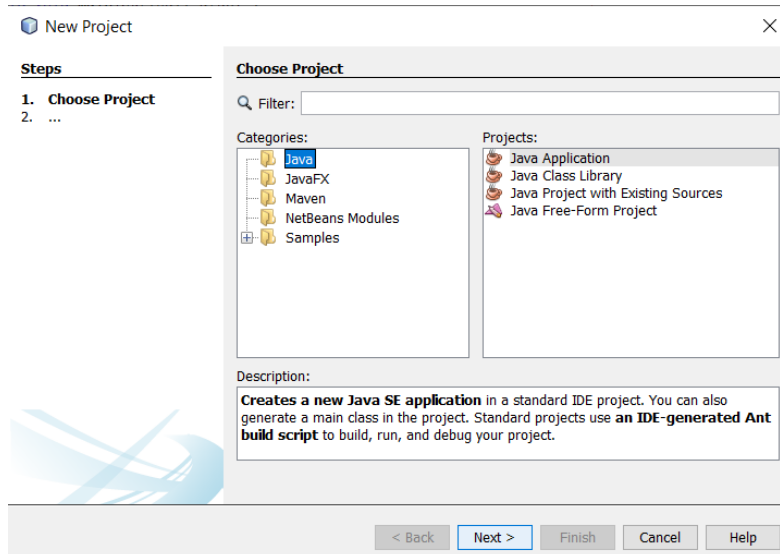
Setelah kita mempelajari tentang dasar-dasar dari pemrograman dasar maka dalam pertemuan ini kita akan memulai membuat sebuah pemrograman java sederhana.

#### Percobaan 1

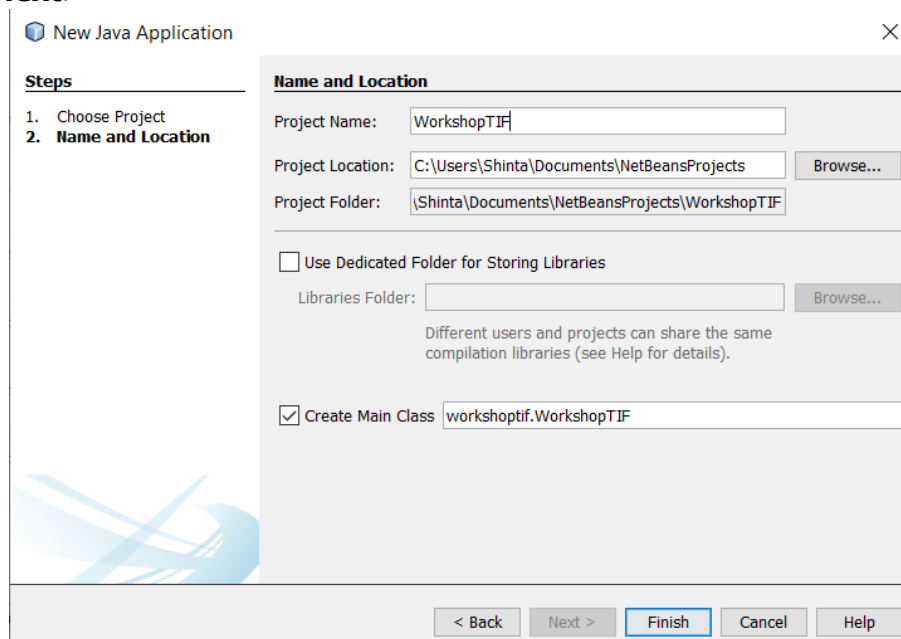
- Buka Netbeans yang sudah anda install
- Buat project baru dengan nama "WorkshopTIF" dengan cara

File -> New Project -> Java Application ->Next ->WorkshopTIF ->Finish

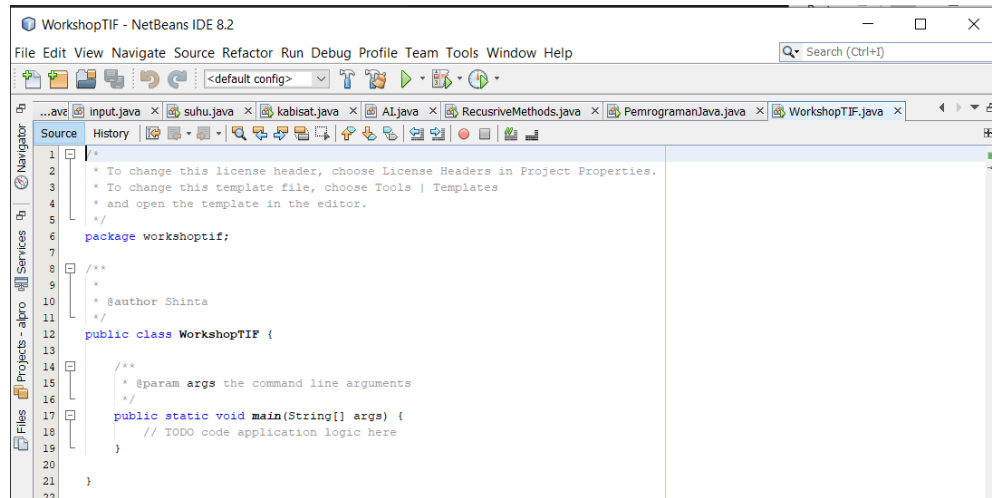




- Klik **Next>**



- Beri nama project WorkshopTIF ? klik **Finish**
- Maka akan muncul tampilan awal berikut ini

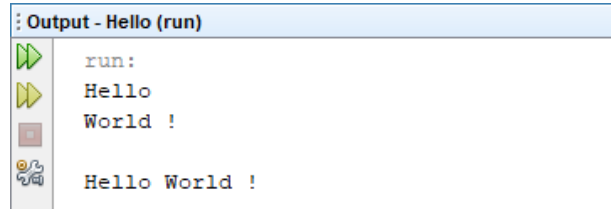


**a. Praktikkan code berikut ini untuk dasar pemrograman java**

```
System.out.println("Hello ");
System.out.println("World !");
```

```
System.out.print("Hello ");
System.out.print("World !");
```

Output:



**b. Praktikkan code berikut ini untuk variable dan tipe data**

```
public static void main(String[] args) {
    // TODO code application logic here

    String nama = "Indonesia";
    char abjad = 'a';
    int a = 6;
    double b = 6.9;

    System.out.println (nama);
    System.out.println (a);
}
```



### Output:

```
Output - materi_ajar_DDP (run)
run:
Indonesia
6
BUILD SUCCESSFUL (total time: 0 seconds)
```

### c. Praktikkan code berikut ini untuk class, object, dan method

```
class book {
    int price;
    int pages;

    public void set (int price, int pages) {
        this.price = price;
        this.pages = pages;
    }

    public void show () {
        System.out.println("books information");
        System.out.println("books price : " +price);
        System.out.println("number of pages : " +pages);
    }
}

public class Buku {
    public static void main(String[] args) {
        book javabook = new book();
        javabook.set(60000, 100);
        javabook.show();
    }
}
```

book merupakan nama class yang berisi 2 komponen: atribut dan method.

Atribut: price dan pages

Method: set dan show

Kelas Buku berisi main method.

book javabook = new book(); merupakan sebuah object dari class book. Yang mana object tersebut mendeklarasikan method set dan show.

Output:

```
Output - materi_ajar_DDP (run)

run:
books information
books price : 60000
number of pages : 100
BUILD SUCCESSFUL (total time: 0 seconds)
```

d. Praktikkan code berikut ini untuk masukan dari keyboard

```
package inputankeyboard;

import java.util.Scanner;

public class prak_Scanner {

    public static void main(String[] args) {
        // Buat scanner
        Scanner s1 = new Scanner(System.in);

        // Ambil nilai dari keyboard
        System.out.print("Masukkan nilail = ");
        int nilail = s1.nextInt();
        System.out.print("Masukkan nilai2 = ");
        int nilai2 = s1.nextInt();

        // Hitung dan Print
        int jumlah = nilail + nilai2;
        System.out.println("Jumlah = " + jumlah);
    }
}
```

Output:

```
Output - materi_ajar_DDP (run)

run:
Masukkan nilail = 10
Masukkan nilai2 = 15
Jumlah = 25
BUILD SUCCESSFUL (total time: 29 seconds)
```

#### d. Alat dan Bahan

1. Komputer
2. Netbeans
3. JDK

#### e. Hasil dan Pembahasan

1. Laporan hasil praktikum

2. Laporan berupa file word dikumpulkan pada <http://jti.poliije.ac.id/elearning>

**f. Kesimpulan**

Mahasiswa memahami dan mampu mengimplementasikan input output pada pemrograman java netbeans.

**g. Rubrik Penilaian**

| No          | Indikator   | Skor* |   |   |   |
|-------------|---|-------|---|---|---|
| 1           | Ketepatan dalam menjalankan serta menjelaskan dari tugas            | 1     | 2 | 3 | 4 |
| 2           | Kurang dan ketepatan dalam menjalankan serta menjelaskan dari tugas | 1     | 2 | 3 | 4 |
| 3           | Kurang tepat dalam menjelaskan tugas                                | 1     | 2 | 3 | 4 |
| 4           | Keterlambatan dan ketidaktepatan dalam menjelaskan tugas            | 1     | 2 | 3 | 4 |
| Jumlah skor |   |       |   |   |   |