

## **MODUL 3**

### **Perulangan, Percabangan dan Array Pada Java**

#### **Tujuan :**

1. Mahasiswa mengetahui dan memahami perulangan dan percabangan di java
2. Mahasiswa mengetahui dan memahami array satu dimensi dan multi dimensi
3. Mahasiswa memahami cara mengakses array

#### **Tugas Pendahuluan :**

1. Apa yang anda ketahui tentang perulangan dan percabangan di Java
2. Sebutkan kegunaan dari perulangan dan percabangan di Java

### **DASAR TEORI**

#### **1.1. Percabangan**

##### **1.1.1 Percabangan if**

Pernyataan if merupakan salah satu bentuk pernyataan yang berguna untuk mengambil keputusan terhadap sebuah kemungkinan. Bentuk pernyataan if berupa

```
if(kondisi){  
  
    // yang akan dijalankan  
  
}
```

##### **1.1.2. Percabangan if-else**

Percabangan if-else merupakan percabangan yang sama dengan percabangan if namun memiliki kondisi false, artinya jika kondisi pada if tidak terpenuhi maka perintah pada else akan dijalankan. Bentuk pernyataan if-else berupa:

```
if(kondisi){  
    // jalankan jika kondixi true  
}else{  
    // jalankan jika kondisi false  
}
```

##### **1.1.3. Percabangan if bersarang**

Percabangan if bersarang merupakan gabungan beberapa if dan dapat pula digabung

dengan if-else. Bentuk pernyataan if bersarang adalah sebaga berikut:

```
if(kondisi1){  
    // perintah kondisi1  
}else if(kondisi2){  
    // perintah kondisi2  
}else if(kondisi3){  
    // perintah kondisi3  
}else{  
    // perintah jika semua kondixi tidak ada yang benar}
```

#### 1.1.4. Percabangan switch-case

Percabangan switch-case merupakan percabangan yang kondisinya hanya dapat menggunakan perbandingan == (sama dengan). Bentuk pernyataan percabangan switch-case adalah sebagai berikut :

```
switch(variabel){ case nilai1:  
    // jalankan instruksi break; // hentikan  
case nilai2:  
    // jalankan instruksi break; // hentikan  
case nilai2:  
    // jalankan instruksi break; // hentikan  
case nilai4:  
    // jalankan instruksi break; // hentikan  
default:  
    // jalankan instruksi break; // hentikan}
```

Pada

percabangan switch pertama, diperlukan sebuah variabel, setelah itu bada bagian case dibandingkan, jika sama, maka instruksi akan dijalankan sampai menemui tanda break.

## 1.2. Perulangan

### 1.2.1. Perulangan while

Pernyataan while berguna untuk melakukan proses perulangan untuk sebuah kondisi, selama kondisi tersebut bernilai benar (true), maka perulangan akan terus berjalan, dan terhenti ketika kondisi bernilai salah (false). Bentuk pernyataan while

seperti berikut ini :

```
while(kondisi){  
    // isi instruksi  
}
```

## **122. Perulangan do-while**

Perulangan do-while merupakan perulangan yang hampir mirip dengan perulangan while namun perbedaannya, pada perulangan do-while, maka minimal instruksi akan dijalankan sekali. Bentuk pernyataan do-while sebagai berikut :

```
do{  
    // instruksi  
}while(kondisi);
```

## **123. Perulangan for**

Perulangan for merupakan perulangan yang memiliki variabel untuk melakukan pengkondisian, berbeda dengan while dan do-while yang kita harus membuat sebuah variabel diluar untuk melakukan penkondisian, pada perulangan for, ditempatkan sebuah blok untuk membuat variabel dan melakukan proses pengkondisian. Bentuk pernyataan for seperti berikut :

```
for(inisialisasi;kondisi;penaikan/penurunan){  
  
    instruksi  
}
```

## **124. Perintah break**

Perintah break merupakan perintah yang dapat digunakan untuk menghentikan proses perulangan.

## **125. Perintah continue**

Perintah continue dapat digunakan untuk meloncati sebuah perulangan, maksudnya adalah instruksi yang seharusnya dapat dilewat, hal ini berarti instruksi tidak akan dijalankan.

### **1.3.1. Array**

Suatu array adalah sebuah struktur data yang terdiri atas banyak variabel dengan tipe data sama,

dimana masing-masing elemen variabel mempunyai nilai indeks. Setiap elemen array mampu untuk menyimpan satu jenis data (variabel).

Array merupakan sekumpulan obyek yang memiliki tipe data yang sama dan dapat di akses secara random dengan menggunakan index. Array mempunyai panjang yang tetap, artinya ketika kita mendeklarasikan suatu array dengan panjang 10, maka array tersebut panjangnya akan tetap 10 walaupun kita hanya memakai 5 elemen.

### 1.3.2. Deklarasi Array satu dimensi

Untuk mendeklarasikan array dapat dilakukan dengan cara berikut ini :

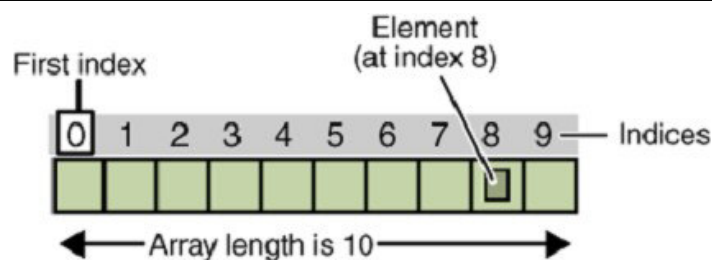
```
int value [];  
value = new int [100]; // menginisialisasikan  
                        //[100] merupakan batas ruang array
```

atau bisa deklarasi sekaligus menginisialisasikan

```
Int value []=new int [100];
```

Cara pendeklarasian tanpa batas

```
int value []={1,2,3}; //{1,2,3} adalah isi dari array
```



Gambar visual suatu array

Untuk mengetahui panjang array yaitu gunakan syntax **nama array.length** berdasarkan contoh diatas nama array = value maka untuk mengetahui panjang value kita gunakan syntax **value.length**. cara pemanggilan yaitu value[0] akan menampilkan angka 1.

### 1.3.3. Array Multidimensi

Array tidak hanya terdiri dari 1 dimensi, array juga bisa terdiri dari 2 dimensi, 3 dimensi dan n-dimensi. Array di atas 3 dimensi sangat jarang digunakan karena sangat sulit untuk di gambarkan. Array multidimensi diimplementasikan sebagai array dalam array. Untuk lebih jelas mengenai cara pendeklarasian array multidimensi dan penggunaanya, silakan ketikkan program di bawah ini.

**Array 2 dimensi:**

```
char multiChar[ ][ ]=new char[10][5];  
char multiChar2[ ][ ]={{'a','b','c'},{'d','e','f'}}; inisialisasi array 2 dimensi
```

**Array 3 dimensi:**

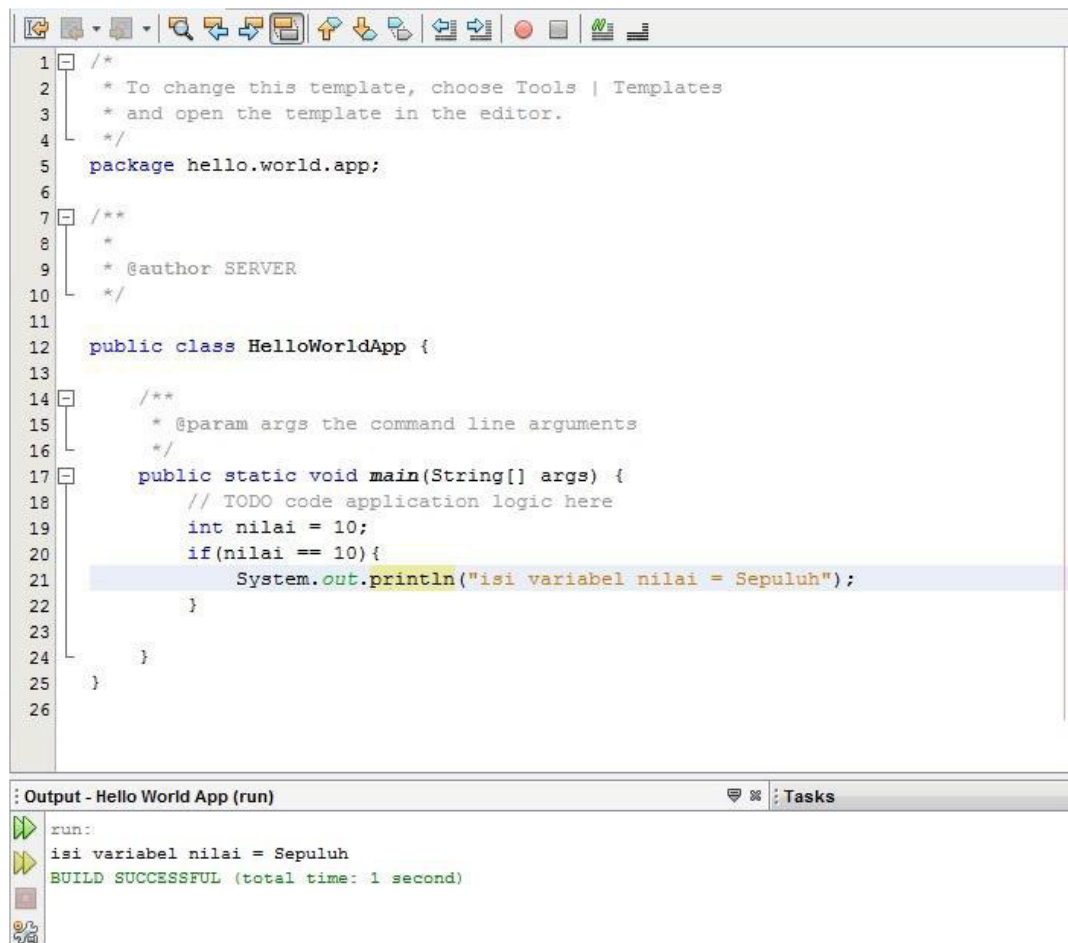
```
int tigaDimensi[ ][ ][ ]=new int[10][10][10];  
int tigaDimensi2[ ][ ][ ]={{{{1,2,3},{4,5,6}},{{1,2,3},{4,5,6}}}}; inisialisasi array 3 dimensi
```

**Tambahan:**

Array lebih dari 2-dimensi sangat jarang dipakai.

## KEGIATAN PRAKTIKUM

### 2.1. Simple Project If



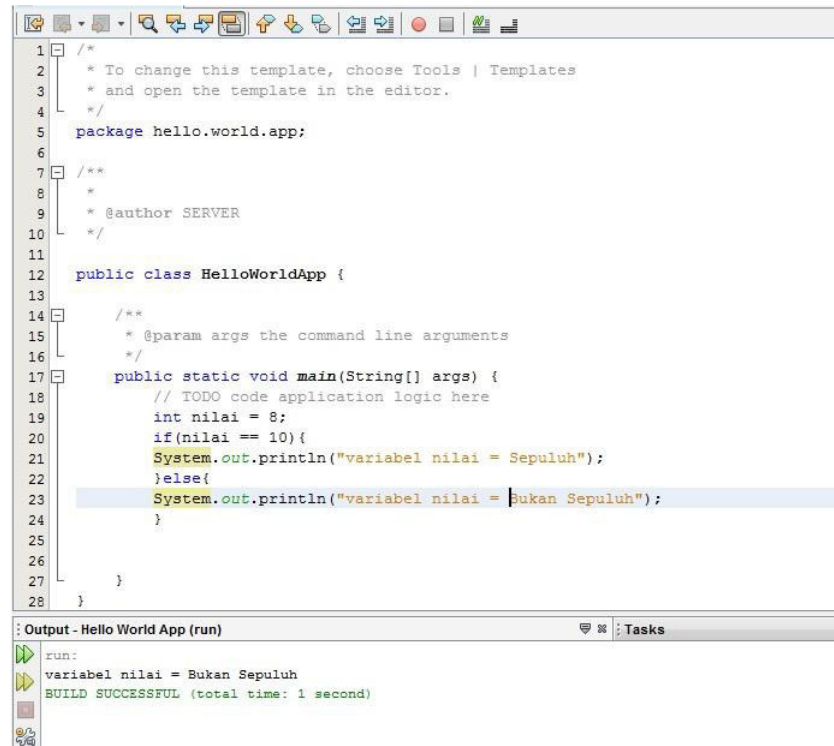
```
1  /*  
2   * To change this template, choose Tools | Templates  
3   * and open the template in the editor.  
4   */  
5  package hello.world.app;  
6  
7  /**  
8   *  
9   * @author SERVER  
10  */  
11  
12  public class HelloWorldApp {  
13  
14      /**  
15       * @param args the command line arguments  
16       */  
17      public static void main(String[] args) {  
18          // TODO code application logic here  
19          int nilai = 10;  
20          if(nilai == 10){  
21              System.out.println("isi variabel nilai = Sepuluh");  
22          }  
23      }  
24  }  
25  
26
```

Output - Hello World App (run)

```
run:  
isi variabel nilai = Sepuluh  
BUILD SUCCESSFUL (total time: 1 second)
```

Jika program diatas dijalankan, maka hasilnya adalah tulisan “variabel nilai = Sepuluh” karena kondisi pada if bernilai true, jika kondisi bernilai salah, misal nilai == 100, maka program tidak akan menghasilkan tulisan apa-apa.

## 2.2. Simple Project If else



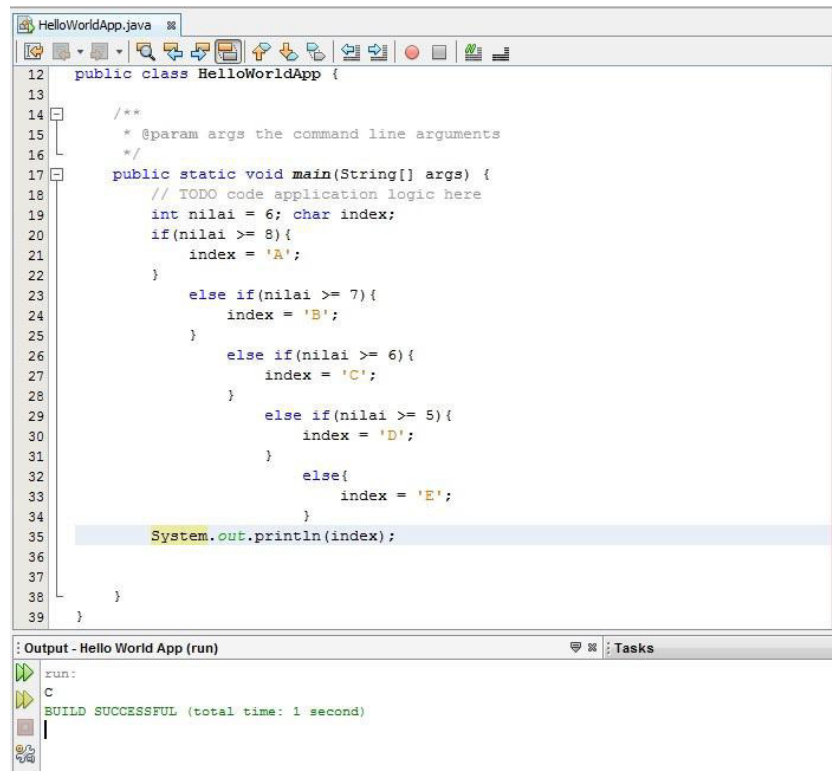
```
1  /*
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package hello.world.app;
6
7  /**
8   *
9   * @author SERVER
10  */
11
12  public class HelloWorldApp {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19          int nilai = 8;
20          if(nilai == 10){
21              System.out.println("variabel nilai = Sepuluh");
22          }else{
23              System.out.println("variabel nilai = Bukan Sepuluh");
24          }
25      }
26
27  }
28  }
```

Output - Hello World App (run)

```
run:
variabel nilai = Bukan Sepuluh
BUILD SUCCESSFUL (total time: 1 second)
```

Jika program diatas dijalankan, maka hasilnya adalah tulisan “variabel nilai = Bukan Sepuluh”, hal ini dikarenakan nilai bernilai 8, bukan 10.

## 2.3. Simple Project if else if



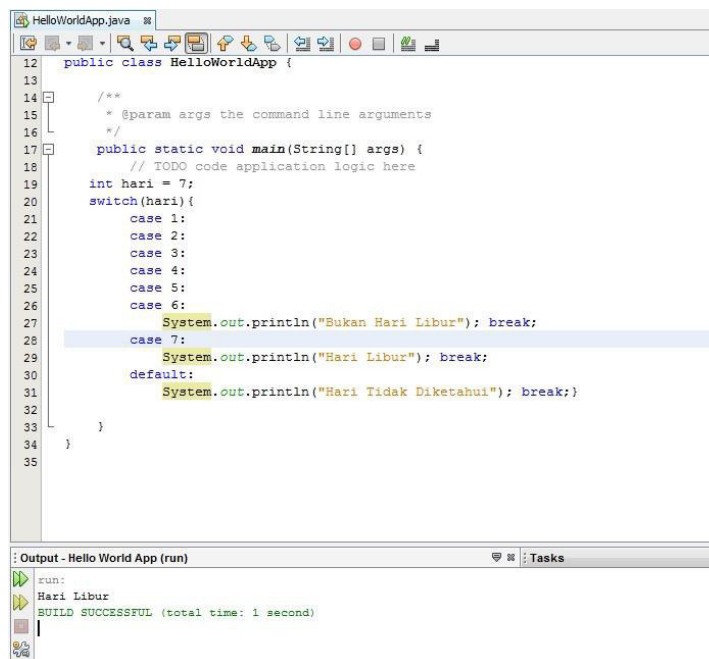
```
12 public class HelloWorldApp {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         // TODO code application logic here
19         int nilai = 6; char index;
20         if(nilai >= 8){
21             index = 'A';
22         }
23         else if(nilai >= 7){
24             index = 'B';
25         }
26         else if(nilai >= 6){
27             index = 'C';
28         }
29         else if(nilai >= 5){
30             index = 'D';
31         }
32         else{
33             index = 'E';
34         }
35         System.out.println(index);
36     }
37 }
38
39 }
```

Output - Hello World App (run)

run:  
BUILD SUCCESSFUL (total time: 1 second)

Jika program diatas dijalankan, maka hasilnya adalah „C“.

## 2.4. Simple Project Switch – Case



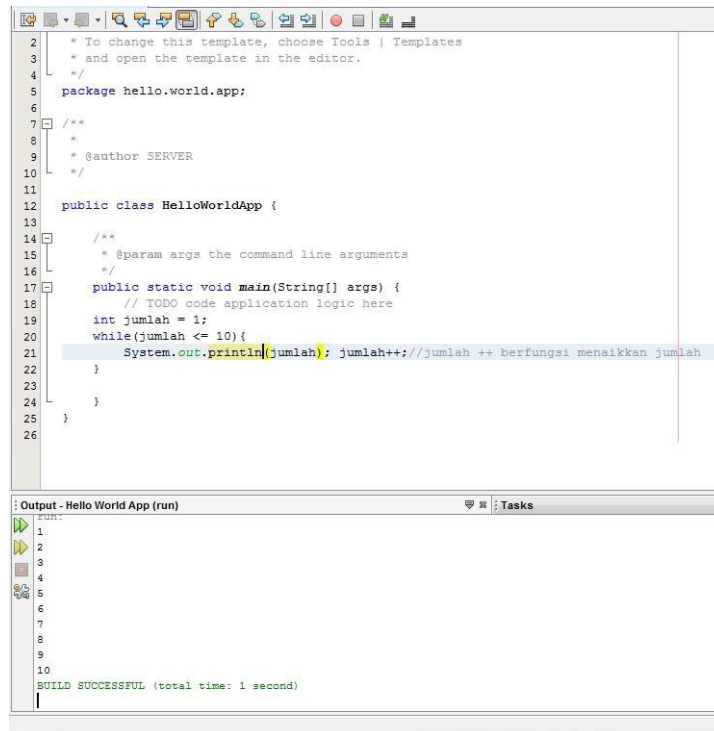
```
12 public class HelloWorldApp {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         // TODO code application logic here
19         int hari = 7;
20         switch(hari){
21             case 1:
22             case 2:
23             case 3:
24             case 4:
25             case 5:
26             case 6:
27                 System.out.println("Bukan Hari Libur"); break;
28             case 7:
29                 System.out.println("Hari Libur"); break;
30             default:
31                 System.out.println("Hari Tidak Diketahui"); break;}
32     }
33 }
34
35 }
```

Output - Hello World App (run)

run:  
Hari Libur  
BUILD SUCCESSFUL (total time: 1 second)

Jika program diatas dijalankan, maka program akan menghasilkan tulisan “Hari Libur”.

## 2.5. Simple Project While



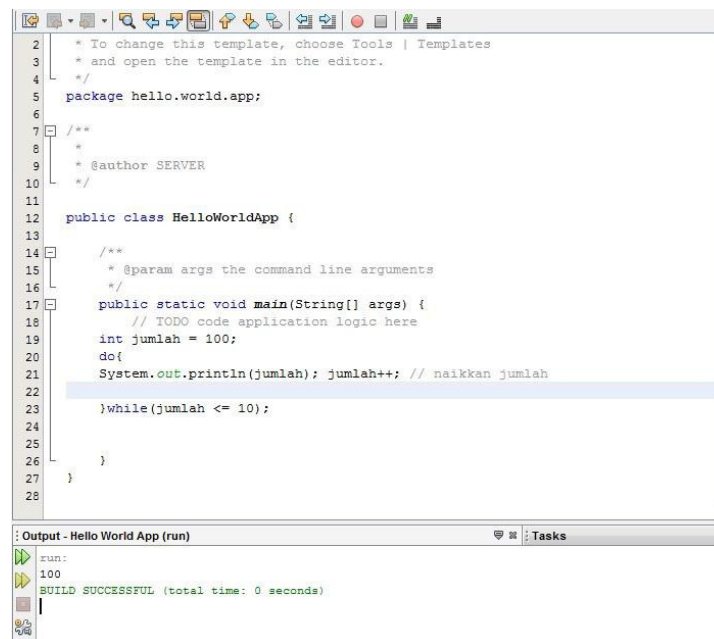
```
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package hello.world.app;
6
7  /**
8   *
9   * @author SERVER
10  */
11
12  public class HelloWorldApp {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19          int jumlah = 1;
20          while(jumlah <= 10){
21              System.out.println(jumlah); jumlah++; // jumlah ++ berfungsi menaikkan jumlah
22          }
23      }
24  }
25
26
```

Output - Hello World App (run)

```
run:
1
2
3
4
5
6
7
8
9
10
BUILD SUCCESSFUL (total time: 1 second)
```

Jika program tersebut dijalankan, maka hasilnya adalah tulisan dari 1 sampai 10. Dan saat jumlah bernilai 11, maka perulangan akan berhenti dikarenakan kondisi bernilai false (11 <= 10).

## 2.6. Simple Project Do – while



```
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package hello.world.app;
6
7  /**
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          int jumlah = 100;
20          do{
21              System.out.println(jumlah); jumlah++; // naikan jumlah
22          }while(jumlah <= 10);
23      }
24  }
25
26
27
28
```

Output - Hello World App (run)

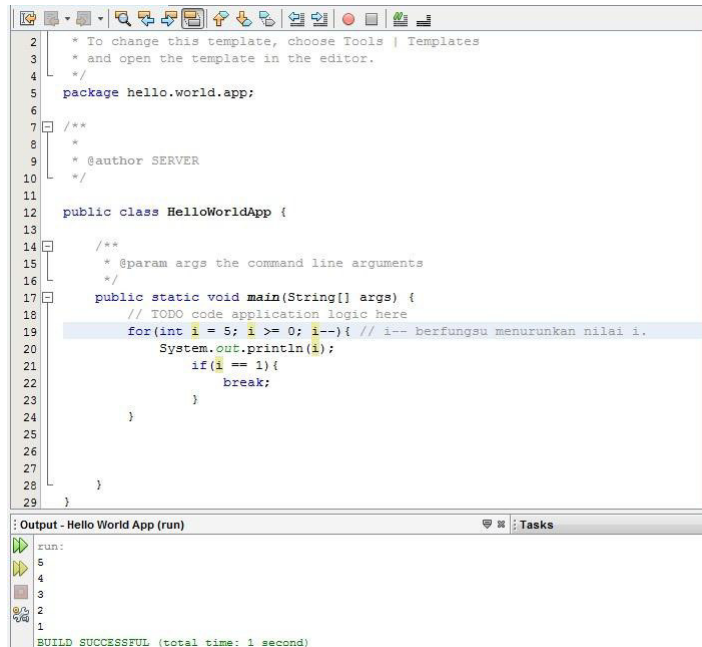
```
run:
100
BUILD SUCCESSFUL (total time: 0 seconds)
```

Jika program tersebut dijalankan, maka akan menghasilkan keluaran 100, artinya walaupun kondisi salah, namun minimal isi instruksi akan dijalankan sekali, hal ini dikarenakan proses do-while berbeda dengan while, dimana do-while pertama melakukan instruksi baru mengecek



kondisi, sedangkan while pertama mengecek kondisi baru melakukan instruksi.

## 2.7. Simple Project Break



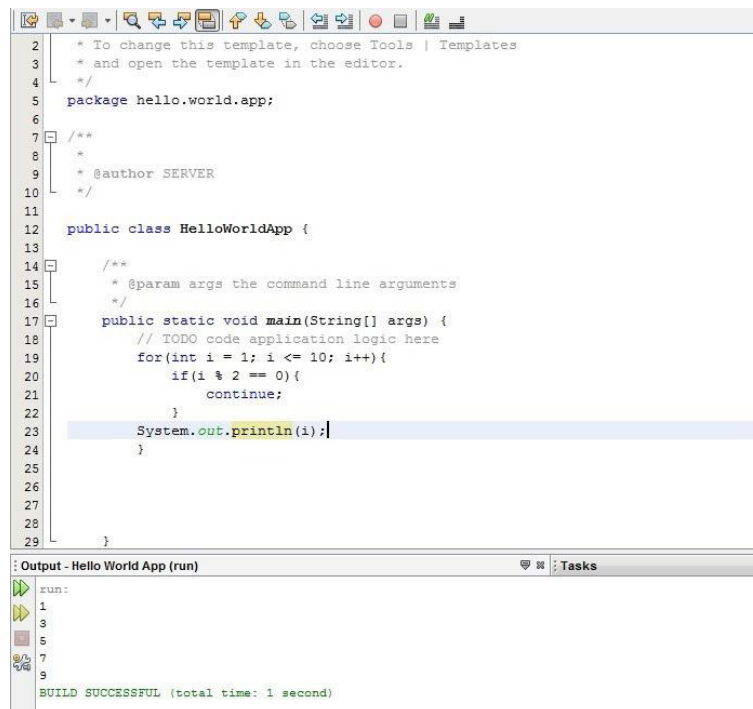
```
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package hello.world.app;
6
7  /**
8   *
9   * @author SERVER
10  */
11
12  public class HelloWorldApp {
13
14      /**
15       * @param args the command line arguments
16       */
17      public static void main(String[] args) {
18          // TODO code application logic here
19          for(int i = 5; i >= 0; i--){ // i-- berfungsi menurunkan nilai i.
20              System.out.println(i);
21              if(i == 1){
22                  break;
23              }
24          }
25      }
26  }
27
28  }
29  }
```

Output - Hello World App (run)

```
run:
5
4
3
2
1
BUILD SUCCESSFUL (total time: 1 second)
```

Maka program hanya akan menampilkan angka dari 5 sampai 1, karena pada saat *i* mencapai 1, program dihentikan oleh perintah `break`.

## 2.8. Simple Project Continue



```
2  * To change this template, choose Tools | Templates
3  * and open the template in the editor.
4  */
5  package hello.world.app;
6
7  /**
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          for(int i = 1; i <= 10; i++){
20              if(i % 2 == 0){
21                  continue;
22              }
23              System.out.println(i);
24          }
25      }
26  }
27
28  }
29  }
```

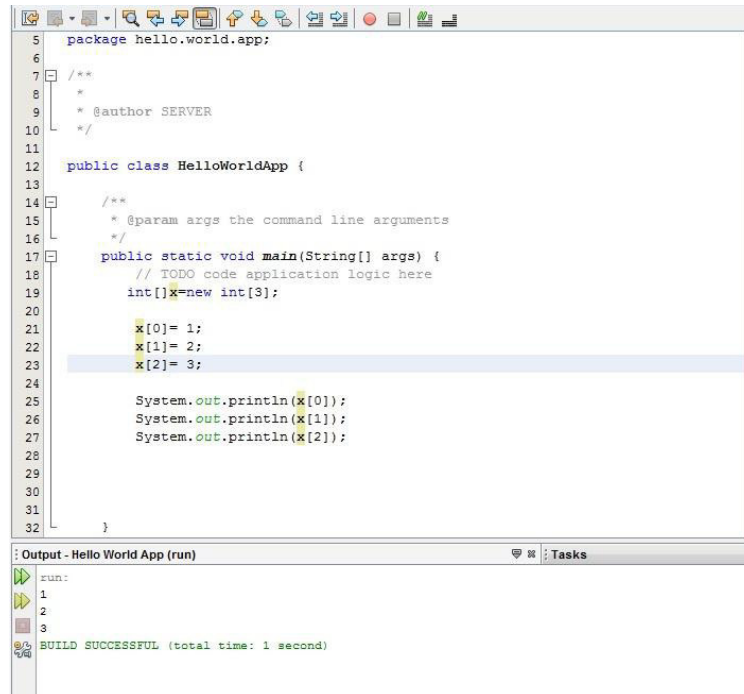
Output - Hello World App (run)

```
run:
1
3
5
7
9
BUILD SUCCESSFUL (total time: 1 second)
```

Jika program diatas dijalankan, maka hasilnya akan menampilkan angka-angka ganjil saja, hal ini dikarenakan saat nilai *i* merupakan angka genap, maka perintah `continue` membuat program

tidak menampilkan angka genap.

## 2.9. Simple Project Array 1 Dimensi



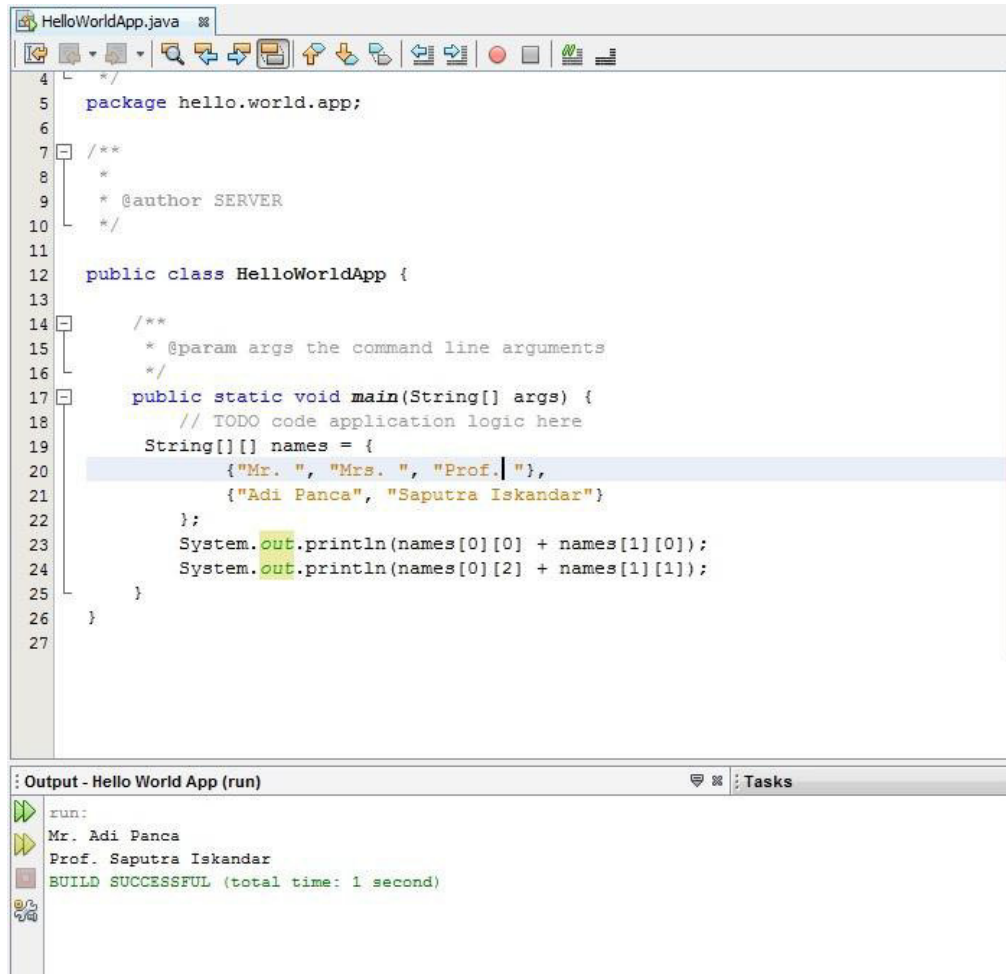
```
5 package hello.world.app;
6
7 /**
8  *
9  * @author SERVER
10 */
11
12 public class HelloWorldApp {
13
14     /**
15      * @param args the command line arguments
16      */
17     public static void main(String[] args) {
18         // TODO code application logic here
19         int[] x=new int[3];
20
21         x[0]= 1;
22         x[1]= 2;
23         x[2]= 3;
24
25         System.out.println(x[0]);
26         System.out.println(x[1]);
27         System.out.println(x[2]);
28
29
30
31
32 }
```

Output - Hello World App (run)

```
run:
1
2
3
BUILD SUCCESSFUL (total time: 1 second)
```

Program yang ditampilkan adalah 1,2, dan 3 sebab array x telah dideklarasikan 1,2,3.

## 2.10. Simple Project Array 2 Dimensi / Multidimensi



The screenshot shows an IDE window titled 'HelloWorldApp.java'. The code is as follows:

```
4  /**
5  package hello.world.app;
6
7  /**
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[][] names = {
20             {"Mr. ", "Mrs. ", "Prof. | "},
21             {"Adi Panca", "Saputra Iskandar"}
22         };
23         System.out.println(names[0][0] + names[1][0]);
24         System.out.println(names[0][2] + names[1][1]);
25     }
26 }
27
```

Below the code editor is the 'Output - Hello World App (run)' window, which displays the following output:

```
run:
Mr. Adi Panca
Prof. Saputra Iskandar
BUILD SUCCESSFUL (total time: 1 second)
```

Program yang ditampilkan adalah Mr. Adi Panca | Prof. Saputra Iskandar sebab pemanggilan pada array names [0] [0] = **Mr** dan names [1][0] = **Adi Panca**. Demikian pun dengan pemanggilan kedua pada Prof. Saputra Iskandar.

## Percobaan

### Percobaan 1 : Percabangan menggunakan if, if-else dan else-if

```
class IfElseName {
    public static void main (String args[]) {
        char firstInitial = 'a';

        if (firstInitial == 'a')
            System.out.println("Nama anda pasti Asep!");
        else if (firstInitial == 'b')
            System.out.println("Nama anda pasti Brodin!");
        else if (firstInitial == 'c')
            System.out.println("Nama anda pasti Cecep!");
        else
            System.out.println("Nama anda tidak terkenal!");
    }
}
```

### ***Percobaan 2 : Percabangan menggunakan switch***

```
class SwitchName {
    public static void main (String args[]) {
        char firstInitial = 'a';

        switch (firstInitial) {
            case 'a':
                System.out.println("Nama anda pasti Asep!");
            case 'b':
                System.out.println("Nama anda pasti Brodin!");
            case 'c':
                System.out.println("Nama anda pasti Cecep!");
            default:
                System.out.println("Nama anda tidak terkenal!");
        }
    }
}
```

### ***Percobaan 3 : Percabangan menggunakan switch dengan break***

```
class SwitchNameBreak {
    public static void main (String args[]) {
        char firstInitial = 'a';

        switch (firstInitial) {
            case 'a':
                System.out.println("Nama anda pasti Asep!");
                break;
            case 'b':
                System.out.println("Nama anda pasti Brodin!");
                break;
            case 'c':
                System.out.println("Nama anda pasti Cecep!");
                break;
            default:
                System.out.println("Nama anda tidak terkenal!");
        }
    }
}
```

### ***Percobaan 4 : Perulangan menggunakan for***

```
class ForCount {
    public static void main (String args[]) {
        int count=1;
        for (int i=0; i<9; i++) {
            for (int j=0; j<i+1; j++) {
                System.out.print(count);
            }
            count++;
            System.out.println();
        }
    }
}
```

### ***Percobaan 5 : Perulangan menggunakan while***

```
class WhileCount {
    public static void main (String args[]) {
        int count=1;
        int i=0;
        while (i<9) {
            int j=0;
            while (j<i+1) {
                System.out.print(count);
                j++;
            }
            count++;
            System.out.println();
            i++;
        }
    }
}
```

### ***Percobaan 6 : Perulangan dengan break***

```
class BreakLoop {
    public static void main (String args[]) {
        int i = 0;
        do {
            System.out.println("Iterasi ke " + i);
            i++;
            if (i > 10) break;
        }
        while (true);
    }
}
```

### ***Percobaan 7 : Perulangan dengan continue***

```
public class ContinueLoop {
    public static void main(String args[]) {
        int a, b;
        for(a=0;a<2;a++)
            for(b=0;b<3;b++) {
                if (b==1) continue;
                System.out.println("a=" + a + " ; b=" + b);
            }
    }
}
```

### ***Percobaan 8 : Pemakaian label pada kondisi break***

```
public class BreakLabel {
    public static void main(String args[]) {
        int a, b;
        Mulai:
        for(a=0;a<2;a++)
            for(b=0;b<3;b++) {
```

```

        if (b==1) break Mulai;
        System.out.println("a=" + a + " ; b=" + b);
    }
}

```

### ***Percobaan 9 : Pemakaian label pada kondisi continue***

```

public class ContinueLabel {
    public static void main(String args[]) {
        int a, b;
        Mulai:
        for(a=0;a<2;a++)
            for(b=0;b<3;b++) {
                if (b==1) continue Mulai;
                System.out.println("a=" + a + " ; b=" + b);
            }
    }
}

```

### **TUGAS**

1. Buatlah program mencari nilai rata rata,nilai terbesar dimana nilai yg diinput oleh user, minimal inputan 10x.
2. Buatlah program mencari bilangan berpangkat, dengan ketentuan user menginput bilangan bulat, kemudian user menginput bilangan pangkat, maka akan muncul hasil dari kedua bilangan tersebut.
3. Buatlah program untuk menentukan suatu tahun kabisat atau bukan dimana tahun dibatasi mulai dari tahun 1900 sampai dengan tahun 2020

Contoh tampilan:

Masukkan tahun (1900-2005) : 1923  
1923 bukan tahun kabisat

Masukkan tahun (1900-2005) : 1898  
Maaf, tahun input dibawah 1900

Masukkan tahun (1900-2005) : 1996  
1996 adalah tahun kabisat

Masukkan tahun (1900-2005) : 2008  
Maaf, tahun input diatas 2005