

# Struktur Kontrol Pemrograman Java : **Perulangan**

Karmilasari

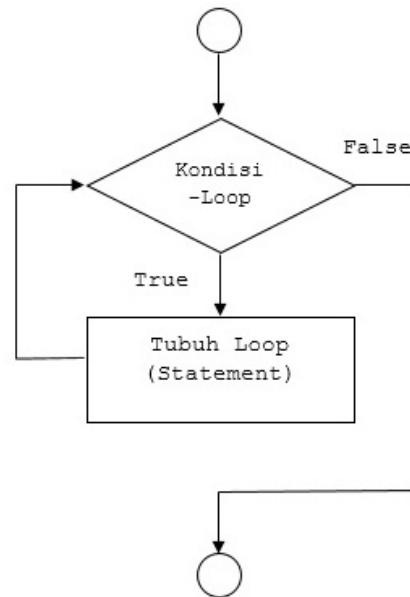
# Perulangan pada Pemrograman Java

- Struktur kontrol pengulangan adalah berupa pernyataan dari Java yang memungkinkan kita untuk mengeksekusi blok kode berulang-ulang sesuai dengan jumlah tertentu yang diinginkan.
- Ada tiga macam jenis dari struktur kontrol pengulangan yaitu **while, do-while dan for**.

# Perulangan While

- Pernyataan while adalah pernyataan atau blok pernyataan yang diulang-ulang selama kondisi -loop bernilai true. Sintaks untuk perulangan while adalah sebagai berikut:

```
while (kondisi-loop) {  
    //tubuh loop  
    statement-statement;  
}
```



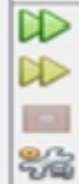
Setiap kondisi loop merupakan suatu ekspresi boolean yang mengendalikan tubuh loop. Kondisi tersebut dievaluasi untuk menentukan apakah tubuh loop di eksekusi atau tidak. Jika hasil evaluasi bernilai true, maka tubuh loop dieksekusi, sedangkan jika bernilai false, maka keseluruhan loop berhenti.

Pernyataan (statement) dalam tubuh loop akan dieksekusi secara berulang.

# Contoh Perulangan While

```
11  * @author LaptopKU
12  */
13  public class jirjis {
14      public static void main (String args[])
15  {
16      Scanner masuk = new Scanner(System.in);
17      int bil;
18      bil=1;
19      while (bil<=5) {
20          System.out.println(bil);
21          bil++;
22      }
23  }
24  }
```

Output - algoritma (run) ☒



run:

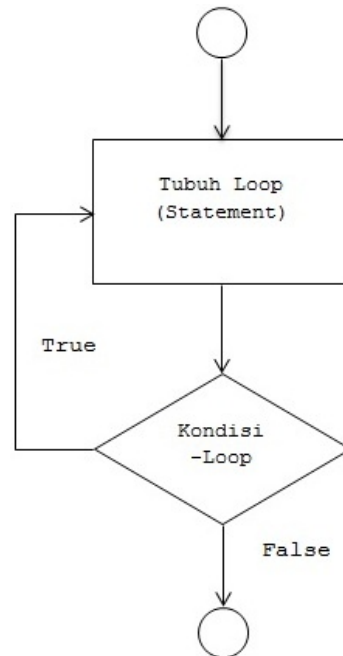
1  
2  
3  
4  
5

BUILD SUCCESSFUL (total time: 0 seconds)

# Perulangan Do - While

- Struktur perulangan do-while sebenarnya mirip dengan struktur perulangan while. Namun bedanya pada perulangan do-while ini, pemeriksaan kondisi-loop dilakukan pada akhir blok perulangan dan proses perulangan akan dilakukan minimal sekali meskipun ternyata kondisinya tidak terpenuhi (bernilai false).
- Bentuk umumnya :

```
do {  
    //tubuh loop  
    statement-statement;  
} while (kondisi-loop);
```

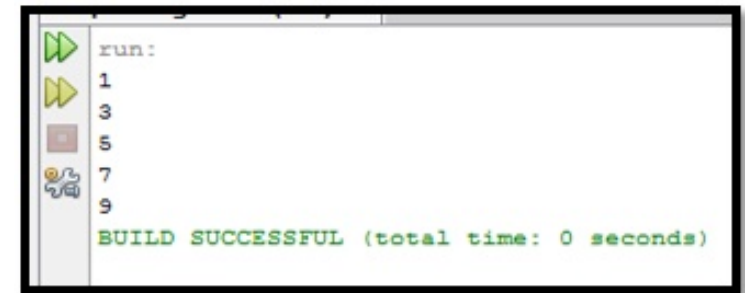


Pernyataan di dalam (tubuh loop) akan dieksekusi terlebih dahulu.

Kemudian kondisi loop dievaluasi. Jika hasil evaluasi adalah true, maka tubuh loop dieksekusi kembali. Jika bernilai false maka pengulangan akan berhenti.

# Contoh Perulangan Do - While

```
import java.util.Scanner;
public class Iani {
    public static void main (String
args[])
{
    scanner masuk = new
scanner(System.in);
    int bil;
    bil=1;
    while (bil<=10){
        bil+=2
    }
}
}
```



```
run:
1
3
5
7
9
BUILD SUCCESSFUL (total time: 0 seconds)
```

# Perulangan For

# Perulangan Bersarang

- Struktur for pada umumnya digunakan untuk melakukan pengulangan yang banyaknya sudah pasti atau sudah diketahui sebelumnya. Dalam pengulangan for kita harus menentukan nilai awal pengulangan dan nilai akhir pengulangan.
- Pengulangan for tidak membutuhkan counter untuk menaikkan variabel karena sudah disebutkan pada salah satu parameter pengulangan. Bentuk umum pengulangan for adalah sebagai berikut:

```
for (nilai inisialisasi awal; kondisi loop; iterasi;) {  
    //tubuh loop, statement yang akan diulang  
}
```

- Proses pengulangan akan terus dilakukan selama kondisi loop bernilai true. Dengan kata lain, proses pengulangan hanya akan dihentikan apabila kondisi telah bernilai false atau sudah tidak terpenuhi lagi. Pengulangan for biasanya menggunakan suatu variabel untuk mengendalikan berapa kali tubuh loop akan dieksekusi dan menentukan kapan loop akan berhenti. Variabel ini disebut juga dengan variabel kontrol.
- Nilai inisialisasi awal merupakan variabel kontrol, proses inisialisasi nilai awal hanya akan dilakukan sekali saja. Iterasi berfungsi menaikkan (increment) nilai variabel kontrol dan kondisi loop mengevaluasi apakah kondisi perulangan bernilai true atau false.
- Java memungkinkan *loop* yang disarangkan di *loop* yang lain. Satu *loop* berada di dalam *loop* yang lainnya.

# Contoh Perulangan For

```
//Nama program: Perulangan1.java
public class Perulangan1 {
    public static void main(String[] args) {

        int counter;
        int batasPerulangan = 10;

        for (counter=1; counter<=batasPerulangan; counter++)
            System.out.println("Perulangan ke-"+counter);
        }
    }
}
```

## Output program:

```
Perulangan ke-1
Perulangan ke-2
Perulangan ke-3
Perulangan ke-4
Perulangan ke-5
Perulangan ke-6
Perulangan ke-7
Perulangan ke-8
Perulangan ke-9
Perulangan ke-10
```



# Contoh Perulangan For

```
//Nama program: Contoh Perulangan java ke-2 intermediat
public class Perulangan2 {

    public static void main(String[] args) {
        int bilangan1=6;
        int bilangan2=4;
        int hasil=0;
        String angka = "";

        for (int counter=1; counter<=bilangan2; counter++)
            hasil += bilangan1;
            angka = Integer.toString(bilangan1);
            System.out.println(angka);
        }

        System.out.println("----- +");
        System.out.println(hasil);

    }
}
```

## Output program:

```
6
6
6
6
----- +
24
```

# Contoh Perulangan For

```
public class perulanganlogika{  
  
    public static void main(String[] args) {  
  
        int bilangan1 = 1;  
        int bilangan2 = 10;  
        int counter;  
  
        for (counter=1; (counter<10) && (bilangan1<bilangan2  
            System.out.println("\t"+ bilangan1 + "\t" + bilangan2  
            bilangan1++;  
            bilangan2--;  
        }  
    }  
}
```

**Output program:**

1	10	1
2	9	3
3	8	5
4	7	7
5	6	9

# Contoh Perulangan Bersarang

```
public class Loopbersarang1{  
    public static void main(String[] args){  
        for(int i=0;i<10;i++){  
            for(int j=i;j<10;j++){  
                System.out.print("*");  
                System.out.println();  
            }  
        }  
    }  
}
```

**Hasil (*output*) :**

```
*  
**  
*  
**  
*  
**  
*
```

# LATIHAN

- Buatlah sebuah loop bersarang untuk menghasilkan beragam bentuk seperti di bawah ini. Gunakan JOptionPane untuk menanyakan kepada user tingkat/tinggi dari segitiga tersebut.

```
***** * ***** *****
**** ** * * *****
*** *** * * *****
** **** * * *****
* ***** *****

* ***** * * *****
** ***** * * *****
*** *** * *****
**** ** * * *****
***** * * * *****
```