

# Algoritma dan Pemrograman

## Leon Andretti Abdillah

08

**Control Flow – Loop – While and Do While**

# Introduction

- There may be a situation when we need to execute a block of code several number of times, and is often referred to as a “loop”. Other terminology could be “repetition” or “iteration”.
- Java has very flexible three looping mechanisms. You can use one of the following three loops:
  1. **while** Loop
  2. **do...while** Loop
  3. **for** Loop

# while

- A while loop is a control structure that allows you to repeat a task a certain number of times
- The while statement continually executes a block of statements while a particular condition is true.

# While syntax and flowchart

- Its syntax can be expressed as:

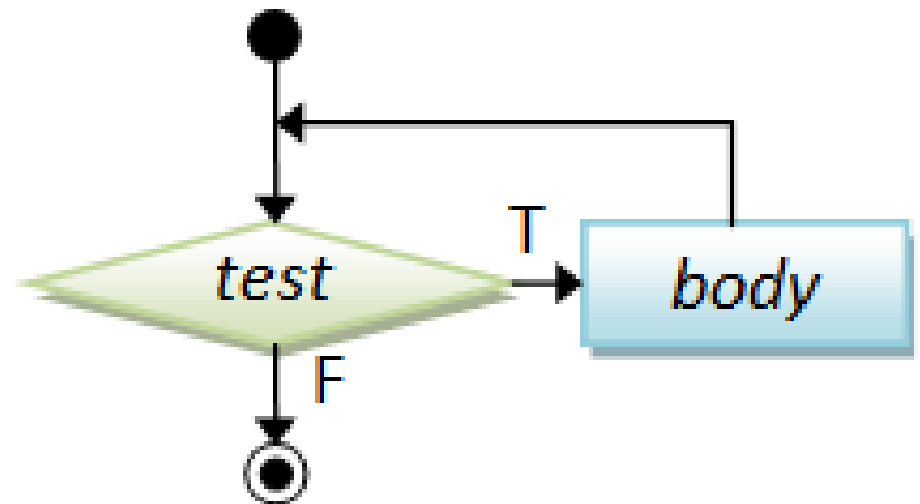
```
while (boolean_expression) {  
    statement(s)  
}
```

or

```
while ( test ) {  
    body;  
}
```

or

```
while (true){  
    // your code goes here  
}
```



# while statement - Test at beginning

- The *while* statement evaluates *expression*, which must return a boolean value. If the expression evaluates to true, the while statement executes the *statement(s)* in the while block.
- The *while* statement continues testing the expression and executing its block until the expression evaluates to false.
- The *while* statement is used to repeat a block of statements while some condition is true. The condition must become false somewhere in the loop, otherwise it will never terminate.

```
package Package05;

class WhileDemo {
    public static void main(String[] args){
        int count = 1;
        while (count <= 10) {
            System.out.println("Number : " + count);
            count++;
        }
    }
}
```

<terminated> WhileDemo [Java Application] C:\Program Files (x86)\Java\jre\bin\java.exe

```
Number : 1
Number : 2
Number : 3
Number : 4
Number : 5
Number : 6
Number : 7
Number : 8
Number : 9
Number : 10
```

```

package Package05;

import java.util.Scanner;

public class WhileSum {
    public static void main(String[] args) {
        Scanner scLim = new Scanner(System.in);

        System.out.print("Input limit number to sum? ");
        int limit = scLim.nextInt();

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

        int sum = 0, number = 1;
        while (number <= limit) {
            System.out.println(number);
            sum += number;
            number++;
        }
        System.out.println("-----");
        System.out.printf("Total number from 1 to %d is %d", limit, sum);
    }
}

```

```
<terminated> WhileSum [Java Application] C:\Program Files
```

```
Input limit number to sum? 5
```

```
-----
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
-----
```

```
Total number from 1 to 5 is 15
```



# Exercise

- Dengan menggunakan while statement coba Anda tampilkan bilangan genap dari 1 s.d. 20, kemudian hitunglah jumlah semua bilangan tersebut!

# do..while

- When you want to test at the end to see whether something should be repeated, the *do..while* statement is the natural choice.
- The difference between do-while and while is that do-while evaluates its expression at the bottom of the loop instead of the top. Therefore, the statements within the do block are always executed at least once.

# do..while syntax and flowchart

- Its syntax can be expressed as:

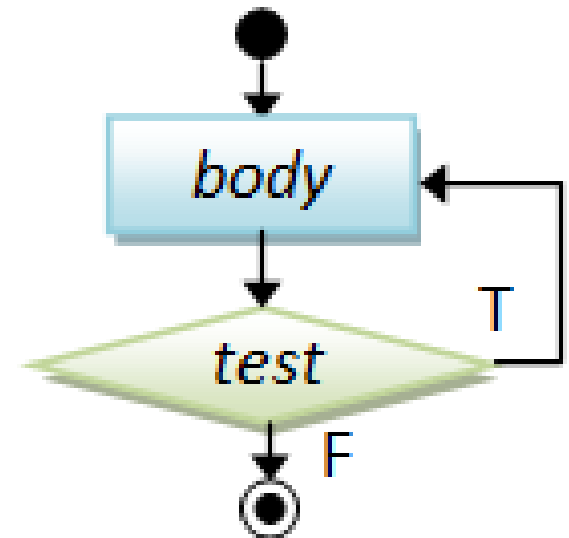
```
do {  
    statement(s)  
}  
while (boolean_expression);
```

or

```
do {  
    body;  
}  
while ( test ) ;
```

or

```
Do (  
    // your code goes here  
} while (true){
```



# do..while statement - Test at end

- A do...while loop is similar to a while loop, except that a do...while loop is guaranteed to execute at least one time.
- do - while loop executes all the statements first at once and then check the condition if the condition is true then all the statements are also executed in second times otherwise second times ignored all the statements.

```
package Package05;

public class DoWhileDemo {
    public static void main(String[] args){
        int count = 1;
        do {
            System.out.println("Count is: " + count);
            count++;
        } while (count <= 10);
    }
}
```

<terminated> DoWhileDemo [Java Application]

```
Count is: 1
Count is: 2
Count is: 3
Count is: 4
Count is: 5
Count is: 6
Count is: 7
Count is: 8
Count is: 9
Count is: 10
```

```

package Package05;

import java.util.Scanner;

public class DoWhileSum {

    public static void main(String[] args) {

        Scanner scLim = new Scanner(System.in);

        System.out.print("Input limit number to sum? ");
        int limit = scLim.nextInt();

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

        int sum = 0, number = 1;
        do {
            System.out.println(number);
            sum += number;
            number++;
        }
        while (number <= limit);

        System.out.println("-----");
        System.out.printf("Total number from 1 to %d is %d", limit, sum);
    }
}

```

---

```
<terminated> DoWhileSum [Java Application]
```

```
Input limit number to sum? 5
```

```
-----
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
-----
```

```
Total number from 1 to 5 is 15|
```

# Exercise

- Dengan menggunakan do while statement coba Anda tampilkan bilangan ganjil dari 1 s.d. 20, kemudian hitunglah jumlah semua bilangan tersebut!