Algoritma dan Struktur Data
Leon Andretti Abdillah

09
Control Flow – Loop – For

2016
Introduction 1

• Structure control, consists of:
  1. Sequence
  2. Selection/Choice/Decision
     a) if..else
     b) switch..case
  3. Loop/Iteration/Repetition
     a) while
     b) do..while
     c) for..
The for statement provides a compact way to iterate over a range of values. Programmers often refer to it as the "for loop" because of the way in which it repeatedly loops until a particular condition is satisfied.

A for loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times.

A for loop is useful when you know how many times a task is to be repeated.

The for loop is a looping construct which can execute a set of instructions a specified number of times. It’s a counter controlled loop.
Syntax

```java
for(initialization; Boolean_expression; update) {
    //Statements
}
```

Or

```java
for (initialization_expression; loop_condition; increment_expression) {
    // statements
}
```
Comparison for and while

```c
for (cnt = 0; cnt < N; cnt++)
{
    inner code here
}
```

```c
while (cnt < N) {
    inner code here
    cnt++
}
```
Here is the flow of control in a for loop:

- The initialization step is executed first, and only once. This step allows you to declare and initialize any loop control variables. You are not required to put a statement here, as long as a semicolon appears.
- Next, the Boolean expression is evaluated. If it is true, the body of the loop is executed. If it is false, the body of the loop does not execute and flow of control jumps to the next statement past the for loop.
- After the body of the for loop executes, the flow of control jumps back up to the update statement. This statement allows you to update any loop control variables. This statement can be left blank, as long as a semicolon appears after the Boolean expression.
- The Boolean expression is now evaluated again. If it is true, the loop executes and the process repeats itself (body of loop, then update step, then Boolean expression). After the Boolean expression is false, the for loop terminates.
Notes:

- `init` is an initialization that will be performed before the first iteration.
- `booleanExpression` is a boolean expression which will cause the execution of statement(s) if it evaluates to true.
- `update` is a statement that will be executed after the execution of the statement block.
- `init`, `expression`, and `update` are optional.
• The for statement will stop only if one of the following conditions is met:
  • booleanExpression evaluates to false
  • A break or continue statement is executed
  • A runtime error occurs.
package Package05;

public class ForDemo {
    public static void main(String[] args){
        for(int i=1; i<=10; i++){
            System.out.println("Count is: "+i);
        }
    }
}

<terminated> ForDemo [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;

public class ForCounter {
    public static void main (String args[]) {
        int jumlah = 0;

        System.out.println("-----");
        for (int i=1; i<=10; i++) {
            System.out.println(i);
            jumlah = jumlah + i;
        }
        System.out.println("-----");
        System.out.println("Jumlah bilangan 1 s.d 10 : " + jumlah);
    }
}

<terminated> ForCounter [Java Application]

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

-----
Jumlah bilangan 1 s.d 10 : 55
For Demo 2

- Perhatikan inisialisasi (j++) dilemakan di atas (sebelum for)
For Demo 3

- Perhatikan update (k++) diletakkan di dalam body

```java
package Package05;

public class ForDemo3 {

    public static void main(String[] args) {
        int k = 1;
        for (; k <= 5;) {
            System.out.println(k);
            k++;
        }
    }
}

<terminated> ForDemo3 [Java Application]

1
2
3
4
5
```
For Demo 4

- Perhatikan condition 
  \((m>5)\) diletakkan di dalam body
For Demo 5

- If we would like to show 1..10, called forward by using increment (++), then
- If we would like to show 10..1, called backward by using decrement (--)
Enhanced for / for each/for in

- The for statement also has another form designed for iteration through Collections and arrays. This form is sometimes referred to as the enhanced for statement, and can be used to make your loops more compact and easy to read.

- Syntax
  ```java
  for (declaration : expression) {
      // Statements
  }
  
or
  for (type itr-var : iterableObj) 
  statement-block
  Or
  ```
package Package05;

enum Week {
    SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY
}

public class ForEnum {

    public static void main(String args[]) {

        System.out.println("Here are all week constants" +
                          " and their ordinal values: ");
        for (Week day : Week.values())
            System.out.println(day.ordinal() + " : " + day);
    }
}
• Note that each `enum` type has a static `values` method that returns an array containing all of the values of the enum type in the order (`ordinal`) they are declared.

• This method is commonly used in combination with the `for-each loop` to iterate over the values of an enumerated type.
Here are all week constants and:

0 : SUNDAY
1 : MONDAY
2 : TUESDAY
3 : WEDNESDAY
4 : THURSDAY
5 : FRIDAY
6 : SATURDAY
For Each Array

```java
package Package05;

public class.ForEachVowels {

    public static void main(String[] args) {
        char[] vowels = { 'a', 'e', 'i', 'o', 'u'};

        for(char ch: vowels) {
            System.out.println(ch);
        }
    }
}
```
<terminated> ForEachVowels [Java Application]

a
ei
ou
Exercise

• Dengan menggunakan for do statement coba Anda tampilkan:
  • NIM Ganjil → bilangan ganjil dari 1 s.d. 20, kemudian hitunglah jumlah semua bilangan tersebut!
  • NIM Genap → bilangan genap dari 1 s.d. 20, kemudian hitunglah jumlah semua bilangan tersebut!

• Write your answer in word docx format, and save the file with template → “Class_Looping_Group_Name.docx”
References

- https://docs.oracle.com/javase/tutorial/java/nutsandbolts/while.html