

Looping

Hafidz Mulia
Teosofi Hidayah Agung

Departemen Matematika
Institut Teknologi Sepuluh Nopember

7 Oktober 2024



Masalah

Melakukan sesuatu yang sama berulang kali adalah hal yang terkadang membosankan. Hal itu akan berakibat ketidakfokusan dalam prosesnya yang berujung pada kesalahan. Namun kita bisa tenang, karena komputer tidak memiliki sifat jenuh seperti manusia. Komputer dapat melakukan sesuatu berulang kali tanpa terjadinya kesalahan asalkan kita memberikan instruksi yang benar.

No... that can't be true.

Daftar isi

- 1 For
 - Nested

- 2 While

- 3 Do-While

- 4 Latihan

```
import java.util.Date;

public class SaveDate {

    public static void main(String arg[]) throws Exception {
        FileOutputStream fos = new FileOutputStream("data.txt");
        ObjectOutputStream oos = new ObjectOutputStream(fos);

        Date date = new Date();
        oos.writeObject(date);
        oos.flush();
        oos.close();
        fos.close();
    }
}
```

Definisi

For adalah salah satu struktur pengulangan yang digunakan untuk melakukan sesuatu berulang kali. Struktur ini memiliki tiga bagian, yaitu inisialisasi, kondisi, dan increment/decrement. Struktur ini biasanya digunakan ketika kita mengetahui berapa kali kita akan melakukan sesuatu.

Listing: Syntax For

```
for ("inisialisasi"; "kondisi"; "increment/decrement") {  
    // Perintah yang akan diulang  
}
```

For

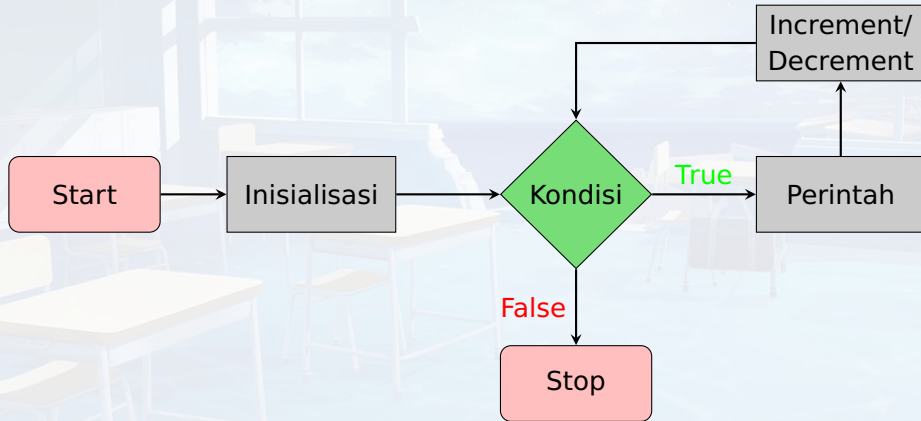


Figure: Flowchart For

For

Contoh

Menampilkan barisan aritmatika $-2, 1, 4, 7, \dots$ sebanyak 20 suku.

```
1  int suku_n = -2;
2  for (int n = 1; n <= 20; n++) {
3      System.out.print(suku_n+" ");
4      suku_n += 3;
5  }
```

-2 1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55

For

Nested

Contoh

Membuat segitiga siku-siku dengan simbol * sebanyak 5 baris.

```
1  for (int i = 1; i <= 5; i++) {  
2      for (int j = 1; j <= i; j++) {  
3          System.out.print("*");  
4      }  
5      System.out.println();  
6  }
```

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

Infinite Loop

Berhati-hatilah ketika menggunakan loop, karena jika tidak memperhatikan kondisi yang diberikan, maka akan terjadi *infinite loop* yang akan membuat program tidak berhenti hingga program mengalami *Stack Overflow*.

Listing: Infinite Loop

```
6  for (int i = 1; i <= 5; i--) {  
7      System.out.println(i);  
8  }
```


Daftar isi

```
import java.util.Date;
```

```
public class SaveDate {
```

1

For

• Nested

```
public static void main(String arg[]) throws Exception {
```

2

While

```
    FileOutputStream fos = new FileOutputStream("data.txt");
```

```
    ObjectOutputStream oos = new ObjectOutputStream(fos);
```

3

Do-While

```
    Date date = new Date();
```

```
    oos.writeObject(date);
```

4

Latihan

```
    oos.flush();
```

```
    oos.close();
```

```
    fos.close();
```

While

Definisi

While adalah salah satu struktur pengulangan yang digunakan untuk melakukan sesuatu berulang kali. Struktur ini hanya memiliki satu bagian, yaitu kondisi. Struktur ini biasanya digunakan ketika kita tidak mengetahui berapa kali kita akan melakukan sesuatu.

Listing: Syntax While

```
while ("kondisi") {  
    // Perintah yang akan diulang  
}
```

While

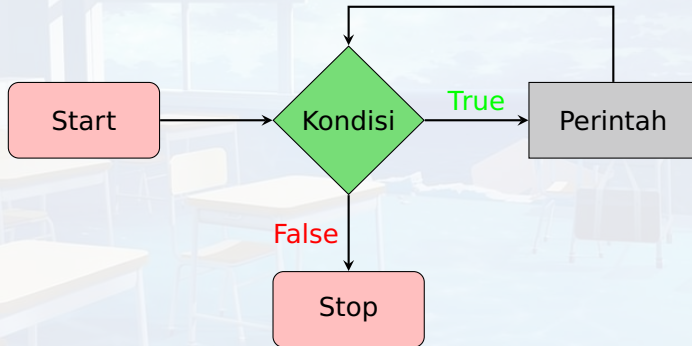


Figure: Flowchart While

While

Contoh

Program untuk menghentikan hacker yang sedang mencuri data warga negara.

```
1  int i = 1;
2  boolean mencuri = true;
3  while (mencuri) {
4      System.out.println("Hacker jangan mencuri!");
5      if (i == 3) {
6          mencuri = false;
7      } else {
8          i++;
9      }
10 }
11 System.out.println("Selamat data warga negara telah aman!");
```

Daftar isi

```
import java.util.Date;
```

```
public class SaveDate {
```

1

For

• Nested

```
    public static void main(String arg[]) throws Exception {
```

2

While

```
        FileOutputStream fos = new FileOutputStream("data.txt");
```

```
        ObjectOutputStream oos = new ObjectOutputStream(fos);
```

3

Do-While

```
        Date date = new Date();
```

```
        oos.writeObject(date);
```

4

Latihan

```
        oos.flush();
```

```
        oos.close();
```

```
        fos.close();
```

Do-While

Definisi

Do-While adalah salah satu struktur pengulangan yang digunakan untuk melakukan sesuatu berulang kali. Struktur ini memiliki dua bagian, yaitu perintah dan kondisi. Struktur ini biasanya digunakan ketika kita ingin melakukan sesuatu minimal satu kali.

Listing: Syntax Do-While

```
do {  
    // Perintah yang akan diulang  
} while ("kondisi");
```

Do-While

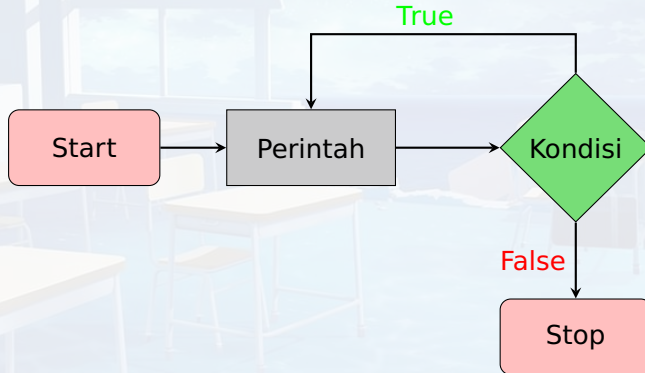


Figure: Flowchart Do-While

Do-While

Perbedaan

Perbedaan utama antara **While** dan **Do-While** adalah **While** melakukan pengecekan kondisi sebelum melakukan perintah, sedangkan **Do-While** melakukan perintah terlebih dahulu sebelum melakukan pengecekan kondisi. Jadi **Do-While** pasti akan melakukan perintah minimal satu kali.

Do-While

Contoh

Countdown sebelum kuis.

```
12  int hari = 4;
13  boolean malas = true;
14  do{
15      System.out.println("Santai masih "+hari+" hari lagi");
16      if (hari == 2) {
17          malas = false;}
18      hari--;
19      }while(malas);
20  System.out.println("Waduh besok udah kuis, ayo belajar!");
```

Daftar isi

```
import java.util.Date;
```

```
public class SaveDate {
```

1

For

• Nested

```
public static void main(String arg[]) throws Exception {
```

2

While

```
FileOutputStream fos = new FileOutputStream("data.txt");
```

```
ObjectOutputStream oos = new ObjectOutputStream(fos);
```

3

Do-While

```
Date date = new Date();
```

```
oos.writeObject(date);
```

4

Latihan

```
oos.flush();
```

```
oos.close();
```

```
fos.close();
```

Latihan 1

Buatlah program untuk menampilkan barisan Fibonacci dengan input n adalah banyaknya suku yang diinginkan.

Fun Fact

Barisan Fibonacci juga bisa disebut Deret Fibonacci, padahal deret dan barisan merupakan dua hal yang berbeda. Ada yang tau alasannya?

Latihan 2

Buatlah program for-loop yang masing-masing menampilkan pola berikut:

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

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

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