

Control Flow

Hafidz Mulia
Teosofi Hidayah Agung

Departemen Matematika
Institut Teknologi Sepuluh Nopember

21 Oktober 2024



Masalah

Dalam hidup, penting untuk memahami kapan waktunya berhenti, melanjutkan, atau mengulang kembali ke titik awal. Hal tersebut perlu kita lakukan untuk mengontrol setiap langkah yang kita ambil. Jangan sampai kita terjebak dalam lingkaran yang sama atau bahkan tidak tahu kapan harus melangkah ke jalan yang benar.

Daftar isi

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

```
public class SaveDate {
```

1 Break

2 Continue

3 Return

4 Latihan

```
    public static void main(String args) {
        FileOutputStream fos = new FileOutputStream("date");
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        Date date = new Date();
        oos.writeObject(date);
        oos.flush();
        oos.close();
        fos.close();
    }
}
```

Break

Definisi

Break adalah sebuah perintah yang digunakan untuk menghentikan atau keluar dari sebuah perulangan atau switch-case yang sedang berjalan. Biasanya *break* berada dalam *if* atau *else jika* di dalam perulangan.

Listing: Contoh penggunaan *break* dalam *for*

```
34 for (...) {  
35     ...  
36     if (...)  
37         break; // Keluar dari loop  
38     ...  
39 }
```

Break

Contoh

Program untuk menentukan bilangan prima atau bilangan komposit.

```
3  int bilangan = input.nextInt();
4  for (int i = 2; i < bilangan; i++) {
5      if (bilangan % i == 0) {
6          System.out.println(bilangan + " bukan bilangan prima");
7          break;
8      }
9      if (i == bilangan - 1)
10         System.out.println(bilangan + " adalah bilangan prima");
11     }
12 System.out.println("Program selesai");
```

Masukkan bilangan: 21 ↵

21 bukan bilangan prima

Program selesai

Break

Labeled Break

Label

Label adalah sebuah penanda yang diletakkan di suatu baris pada kode program yang berfungsi sebagai penanda untuk suatu blok kode tertentu. Label digunakan untuk mengidentifikasi blok kode yang akan dieksekusi oleh perintah `break` atau `continue`.

Listing: Labeled Break

```
9  outerloop:  
10 for (...) {  
11     for (...) {  
12         if (...) {  
13             ...  
14             break outerloop; // Keluar dari semua loop di bawah label  
15         }  
16     }  
17 }
```

Break

Labeled Break

Perbedaan

break biasa hanya akan menghentikan satu perulangan saja, sedangkan break dengan label akan menghentikan perulangan yang diberi label (Bisa lebih dari satu perulangan, tergantung pada berapa perulangan yang berada dibawah label tersebut).

Break

Labeled Break

Listing: Non-Labeled Break

```
70  for (int i = 0; i < 3; i++) {  
71      for (int j = 0; j < 3; j++) {  
72          if (j == 1) {  
73              break; // Hanya menghentikan loop bagian dalam  
74          }  
75          System.out.println("i = " + i + ", j = " + j);  
76      }  
77  }
```

```
i = 0, j = 0  
i = 1, j = 0  
i = 2, j = 0
```

Break

Labeled Break

Listing: Labeled Break

```
83 outerLoop: // Label untuk loop luar
84 for (int i = 0; i < 3; i++) {
85     for (int j = 0; j < 3; j++) {
86         if (j == 1) {
87             break outerLoop; // Menghentikan loop luar
88         }
89         System.out.println("i = " + i + ", j = " + j);
90     }
91 }
```

```
i = 0, j = 0
```

Daftar isi

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

1 public class SaveDate {

Break

2 Continue

3 Return

4 Latihan

```
    public static void main(String args) {
        FileOutputStream fos = new FileOutputStream("date");
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        Date date = new Date();
        oos.writeObject(date);
        oos.flush();
        oos.close();
        fos.close();
    }
}
```

Continue

Definisi

Continue adalah sebuah perintah yang digunakan untuk melanjutkan perulangan berikutnya tanpa mengeksekusi kode di bawahnya. Berbeda dengan *break*, *continue* hanya memberhentikan atau melompati 1 iterasi saja. Program akan tetap berjalan hingga salah satu kondisi untuk berhenti terpenuhi.

Listing: Contoh penggunaan continue dalam for

```
59  for (...) {  
60      ...  
61      if (...)  
62          continue; // Melanjutkan ke iterasi berikutnya  
63      ...  
64  }
```

Continue

Contoh

Program untuk menampilkan faktor dari sebuah bilangan.

```
1 int bilangan = input.nextInt();
2 for (int i = 1; i <= bilangan; i++) {
3     if (bilangan % i != 0)
4         continue;
5     System.out.print(i+" ");
6 }
7 System.out.println("\nProgram selesai");
```

Masukkan bilangan: 12

1 2 3 4 6 12

Program selesai

Continue

Labeled Continue

Label

Sama seperti break, continue juga bisa diberi label. Namun, penggunaan continue dengan label tidak terlalu umum.

Listing: Labeled Continue

```
9  outerLoop: // Label untuk loop luar
10 for (int i = 0; i < 3; i++) {
11     for (int j = 0; j < 3; j++) {
12         if (j == 1) {
13             continue outerLoop; // Melompat ke iterasi berikutnya dari
14                         // loop luar (i)
15         }
16         System.out.println("i = " + i + ", j = " + j);
17     }
```

Daftar isi

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

1 public class SaveDate {

Break

2 public static void main(String args) {

FileOutputStream fos = new FileOutputStream("date.dat");

ObjectOutputStream oos = new ObjectOutputStream(fos);

Date date = new Date();

oos.writeObject(date);

oos.flush();

oos.close();

fos.close();

Return

Definisi

Return adalah sebuah perintah yang digunakan untuk mengembalikan nilai atau menghentikan proses dari suatu fungsi. Karena kita masih belum membahas fungsi/method, intinya return digunakan untuk menghentikan semua proses yang ada di dalam method `main` (untuk sekarang).

Return

Listing: Contoh penggunaan return

```
3 public static void main(String[] args) {
4     int x = input.nextInt();
5     if (x % 2 == 0) {
6         System.out.println("Bilangan genap");
7         return;
8     }
9     System.out.println("Bilangan ganjil");
10 }
```

Masukkan bilangan: 12

Bilangan genap

Masukkan bilangan: 7

Bilangan ganjil

Daftar isi

1. `import java.util.Date;`

2. `public class SaveDate {`

Break

3. `public static void main(String args) {`

Continue

4. `FileOutputStream fos = new FileOutputStream("date.dat");`

Return

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

Latihan

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

7. `oos.writeObject(date);`

8. `oos.flush();`

9. `oos.close();`

10. `fos.close();`

Latihan 1

Program yang menerima inputan kata sandi dari user. Jika kata sandi yang dimasukkan benar, maka program akan menampilkan "Kata sandi benar" dan berhenti. Jika kata sandi yang dimasukkan salah, maka program akan menampilkan "Kata sandi salah" dan meminta inputan kata sandi kembali. Program akan terus berjalan hingga maksimal 3 kali kesempatan untuk memasukkan kata sandi.

Latihan 2

Diberikan bilangan bulat random p yang termuat pada interval $[-50, 100]$. Buatlah program yang menerima input tebakan dari user dan akan berhenti jika tebakan benar. Jika tebakan salah, maka akan ada dua kondisi berikut:

- ▷ Jika tebakan berjarak lebih dari 10 angka dari p , maka program akan menampilkan "Tebakan salah".
- ▷ Jika tebakan berada di interval $[p - 10, p + 10]$, maka program akan menampilkan "Tebakan hampir benar".

Program akan terus berjalan hingga tebakan benar dan tak ada batas maksimal percobaan. (Petunjuk: Gunakan `Math.Random()` untuk menghasilkan bilangan random).