

UNIT 1: FOR

[No. 1] Identifikasi Masalah :

1) Uraikan Permasalahan dan Variabel:

- Soal: Pada contoh 1, kita memiliki `for loop` yang menggunakan `continue` dan `break`. Tujuan dari soal adalah memahami fungsi kedua kata kunci tersebut.
- Variabel: Pada contoh kode terdapat variabel `y`, yang diinisialisasi dengan nilai 0 dan kemudian bertambah dalam perulangan.

[No. 1] Analisis dan Argumentasi :

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara melakukan perubahan pada nilai variabel dan kondisi di dalam `if` untuk memahami efek dari penggunaan `continue` dan `break`.
- 2) Alasan solusi ini karena dengan perubahan, kita dapat menganalisis dampaknya terhadap alur program.
- 3) Perbaiki kode program dengan cara:
 - Ubah 1 : Mengubah batas nilai `y` dari 10 menjadi 15.
 - Ubah 2 : Mengubah kondisi `if` dari `(y % 2 == 1)` menjadi `(y % 2 == 0)` untuk melihat dampak dari perubahan logika.
 - Ubah 3 : Mengubah kondisi `else if` dari `y == 8` menjadi `y == 9`.

[No. 1] Penyusunan Algoritma dan Kode Program :

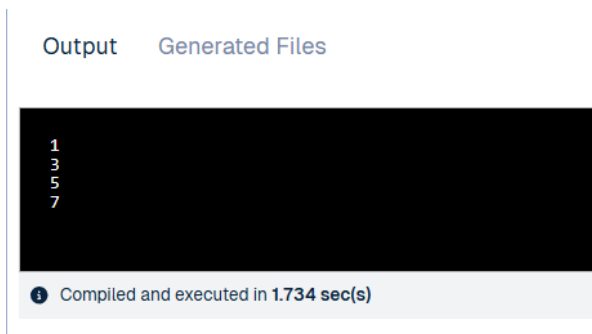
1) Algoritma:

- Inisialisasi variabel `y` dengan nilai 0.
- Periksa apakah `y` memenuhi kondisi perulangan hingga 15.
- Gunakan `continue` untuk melewati nilai-nilai yang memenuhi kondisi tertentu.
- Gunakan `break` untuk menghentikan perulangan saat kondisi tercapai.

2) Kode Program dan Luaran:

```
1 public class ContohFor {  
2     public static void main(String[] args) {  
3         // Ubah 1: for (int y = 0; y <= 15; y++)  
4         for (int y = 0; y <= 15; y++) {  
5             // Ubah 2: if (y % 2 == 0)  
6             if (y % 2 == 0)  
7                 continue;  
8             // Ubah 3: else if (y == 9)  
9             else if (y == 9)  
10                break;  
11            else  
12                System.out.println(y + " ");  
13        }  
14    }  
15 }  
16 }
```

Luaran:



[No. 1] Kesimpulan :

1) Analisa :

- Dalam program ini, `continue` digunakan untuk melewati nilai genap (sesuai perubahan pada Ubah 2), sehingga hanya bilangan ganjil yang dicetak. Penggunaan `break` menghentikan loop saat nilai mencapai 9.
- Penggunaan `continue` dan `break` penting untuk mengendalikan alur perulangan berdasarkan kondisi yang diinginkan.

[No. 2] Identifikasi Masalah :

1) Uraikan Permasalahan dan Variabel:

- Soal: Pada contoh 2, terdapat nested `for loop` yang menggunakan label dan kata kunci `continue` serta `break`.
- Variabel: Terdapat dua variabel, yaitu `i` dan `j`, di mana `i` mengontrol loop pertama dan `j` mengontrol loop kedua.

[No. 2] Analisis dan Argumentasi :

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara mencoba perubahan pada perintah `continue` dan `break` di berbagai posisi.
- 2) Alasan solusi ini karena dengan mencoba variasi penggunaan kata kunci tersebut, kita bisa melihat pengaruhnya terhadap nested loop.
- 3) Perbaiki kode program dengan cara:
 - Ubah 1a: Mengganti `break` dengan `continue pertama`;
 - Ubah 1b: Mengganti `continue` dengan `break pertama`;
 - Ubah 1c: Mengganti `break` dengan `continue kedua`;

[No. 2] Penyusunan Algoritma dan Kode Program :

- 1) Algoritma:
 - Mulai perulangan pertama dengan nilai `i` dari 1 hingga kurang dari 5.
 - Dalam loop pertama, jalankan loop kedua dengan nilai `j` dari 1 hingga kurang dari 3.
 - Coba berbagai variasi `continue` dan `break` untuk melihat efeknya.
- 2) Kode Program dan Luaran:

```
1 public class ForBersarang {
2     public static void main(String[] args) {
3         pertama:
4         for (int i = 1; i < 5; i++) {
5             kedua:
6             for (int j = 1; j < 3; j++) {
7                 System.out.println("i = " + i + "; j = " + j);
8                 // Ubah 1a: continue pertama;
9                 // continue pertama;
10
11                 // Ubah 1b: break pertama;
12                 // break pertama;
13
14                 // Ubah 1c: continue kedua;
15                 // continue kedua;
16
17                 if (i == 2)
18                     break kedua;
19             }
20         }
21     }
22 }
23
```

Luaran:

Output Generated Files

```
i = 1; j = 1
i = 1; j = 2
i = 2; j = 1
i = 3; j = 1
i = 3; j = 2
i = 4; j = 1
i = 4; j = 2
|
```

Compiled and executed in 1.807 sec(s)

[No. 2] Kesimpulan :

1) Evaluasi :

- Perbedaan pada Ubah 1a: `continue pertama` membuat loop pertama tetap berjalan meskipun loop kedua dihentikan sementara.
- Perbedaan pada Ubah 1b: `break pertama` menghentikan kedua loop lebih awal.
- Perbedaan pada Ubah 1c: `continue kedua` hanya melanjutkan loop kedua, sehingga pengulangan dilanjutkan dengan nilai `i` berikutnya.

[No. 3] Identifikasi Masalah :

1) Uraikan Permasalahan dan Variabel:

- Soal: Pada contoh 3, kita ingin mengubah pola piramida menjadi bentuk terbalik.
- Variabel: Terdapat variabel `tinggi` yang diinput oleh pengguna untuk menentukan jumlah baris dalam pola.

[No. 3] Analisis dan Argumentasi :

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara memodifikasi kondisi perulangan sehingga piramida terbentuk terbalik.
- 2) Alasan solusi ini karena perubahan kondisi dalam loop akan membalik arah pencetakan karakter ` `.
- 3) Perbaiki kode program dengan cara:

- Mengubah kondisi perulangan dari `for(int s=tinggi; s>=t; s--)` menjadi `for(int s=1; s<=t; s++)`.

[No. 3] Penyusunan Algoritma dan Kode Program :

1) Algoritma:

- Minta input dari pengguna untuk menentukan tinggi piramida.
- Jalankan loop dari nilai tertinggi ke terendah untuk mencetak setiap baris.
- Pada setiap baris, kurangi jumlah karakter ` ` yang dicetak.

2) Kode Program dan Luaran:

```
1 import java.util.Scanner;
2
3 public class ForBersarang {
4     public static void main(String[] args) {
5         Scanner input = new Scanner(System.in);
6         System.out.print("Masukan Input: ");
7         int tinggi = input.nextInt();
8
9         for(int t = tinggi; t >= 1; t--){
10             for(int s = 1; s <= t; s++){
11                 System.out.print(" ");
12             }
13             System.out.println(); // Membuat baris baru
14         }
15     }
16 }
17
```

Luaran:



Output Generated Files

```
Masukan Input: 7
*****
*****
*****
****
***
**
*
|
```

Compiled and executed in 11.472 sec(s)

[No. 3] Kesimpulan :

1) Analisa:

- Dengan perubahan kondisi dalam loop, pola piramida berhasil dibalik menjadi bentuk yang menurun dari atas ke bawah.