

Nama & NPM	Topik:	Tanggal:
Nabila Azizah Mutiara S G1F024031	FOR dan WHILE Java	8 september 2024

[No. 1] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

Diketahui dari soal :

Contoh 1: Salin dan tempel kode program berikut ke Eclipse.

```
public class ContohFor{
    public static void main(String[] args) {
        for (int y = 0; y <= 10; ++y) {    //ubah 1
            if (y % 2 == 1)                //ubah 2
                continue;    //baris 1
            else if (y == 8)              //ubah 3
                break;    //baris 2
            else
                System.out.println(y + " ");
        } } }
```

Contoh 2: Salin dan tempel kode program berikut ke Eclipse

```
public class ForBersarang {
    public static void main(String[] args) {
        pertama:
            for( int i = 1; i < 5; i++) {

                kedua:
                for(int j = 1; j < 3; j++) {
                    System.out.println("i = " + i + "; j = " + j);
                    if ( i == 2)
                        break kedua;    //ubah1
                } } } }
```

Contoh 3: Salin dan tempel kode program berikut ke Eclipse.

```
import java.util.Scanner;

public class ForBersarang {
    public static void main(String[] args){
        //Instance Input Scanner
        Scanner input = new Scanner(System.in);
        System.out.print("Masukan Input: ");
        int tinggi = input.nextInt(); //Mendapatkan Input Dari User
        for(int t=tinggi; t>=1; t--){
            //Menghitung Jumlah Tinggi Piramida
            for(int s=tinggi; s>=t; s--){
                //Menghitung Jumlah Spasi per Baris
                System.out.print(" ");
            }
            System.out.println(); //Membuat Baris Baru
        }
    }
}
```


1.1

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

```
java -cp /tmp/zF6h3pAQy6/ContohFor
0
2
4
6

=== Code Execution Successful ===
```

Analisis :

- a. Pada ubah 1 program akan mencetak angka genap dari 0 hingga 15.dengan adanya break saat y sam dengan 8 maka output akan berhenti.

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




```
java -cp /tmp/43Z8bTT1Sb/ContohFor
1
3
5
7
9

=== Code Execution Successful ===
```

Analisis :

- b. Pada ubah 2 program ini sudah tidak dapat mencetak angka genap sehingga output hanya akan mencetak angka ganjil dari 0 hingga 15. Namun karena pernyataan break ada saat y sama dengan 8 program akan berhenti diangka 7 dikarenakan 8 merupakan bilangan genap.

Main.java



Run

Output

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

```
java -cp /tmp/k8BrlKaDbB/ContohFor
1
3
5
7

=== Code Execution Successful ===
```

Analysis:

- c. Pada ubah 3 dengan mengubah kondisi ini,program kan mencetak angka genap hingga 8 dan berhenti saat y mencapai 9
Namun apabila semua telah tiubah maka codingan akan mengeluarkan output angka ganjil sebelum angka Sembilan

1.2

A.

```
Main.java
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-                 if (i == 2)
9-                     continue pertama; // Melanjutkan ke iterasi
                                   berikutnya dari outer loop
10-            }
11-        }
12-    }
13- }
14- }
```

```
Output
java -cp /tmp/vT4wxxkG/ForBersarang
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
=== Code Execution Successful ===
```

Analisis : ketika kita mengganti break kedua; dengan continue pertama;, yang terjadi adalah: Saat i bernilai 2, program akan "melompat" kembali ke awal loop luar (pertama), sehingga semua iterasi untuk j masih akan dieksekusi untuk i yang lain (yaitu 3 dan 4). Hasilnya, kita mendapatkan semua nilai dari i dan j, termasuk j = 1 dan j = 2 untuk i = 2. Jadi, meskipun kita memiliki kondisi untuk i = 2, program tetap mencetak semua kombinasi i dan j.

B.

```
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-                 if (i == 2)
9-                     break pertama; // Menghentikan semua loop
10-            }
11-        }
12-    }
13- }
14- }
```

```
java -cp /tmp/50Ragsjvt1/ForBersarang
i = 1; j = 1
i = 1; j = 2
i = 2; j = 1
=== Code Execution Successful ===
```

Analisa : Saat kita mengganti dengan break pertama;;

Ketika i sama dengan 2, program menghentikan seluruh loop. Artinya, tidak ada nilai lain yang akan dicetak setelah itu.

Hasilnya, hanya nilai untuk i = 1 yang dicetak. Kita tidak melihat apapun untuk i = 2, 3, atau 4. Ini seperti menghentikan semua kegiatan begitu kita mencapai nilai tertentu.

C.

```
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-                 if (i == 2)
9-                     continue kedua; // Melanjutkan ke iterasi
                                   berikutnya dari inner loop
10-            }
11-        }
12-    }
13- }
14- }
```

```
java -cp /tmp/mV328heQMM/ForBersarang
i = 1; j = 1
i = 1; j = 2
i = 2; j = 1
i = 2; j = 2
i = 3; j = 1
i = 3; j = 2
i = 4; j = 1
i = 4; j = 2
=== Code Execution Successful ===
```

Analisis : Ketika kita menggunakan continue kedua;;

Saat i sama dengan 2, program hanya akan "melompati" ke iterasi berikutnya dalam loop dalam (kedua), jadi hanya j = 1 yang dicetak untuk i = 2.

Namun, setelah itu, loop luar (pertama) tetap melanjutkan dan mencetak semua nilai untuk i selanjutnya (3 dan 4). Jadi kita melihat i = 3 dan i = 4 dengan semua nilai j.

1.3

```
1- import java.util.Scanner;
2
3- public class ForBersarang {
4-     public static void main(String[] args){
5-         //Instance Input Scanner
6-         Scanner input = new Scanner(System.in);
7-         System.out.print("Masukan Input: ");
8-         int tinggi = input.nextInt(); //Mendapatkan Input
9-         //Dari User
10-        for(int t=tinggi; t>=1; t--){
11-            //Menghitung Jumlah Tinggi Piramida
12-            for(int s=tinggi; s>=t; s--){
13-                //Menghitung Jumlah Spasi per Baris
14-                System.out.print(" ");
15-            }
16-            System.out.println(); //Membuat Baris Baru
17-        }
18-    }
19- }
```

```
java -cp /tmp/dX69mYyTlk/ForBersarang
Masukan Input: 7
*
**
***
****
*****
*****

=== Code Execution Successful ===
```

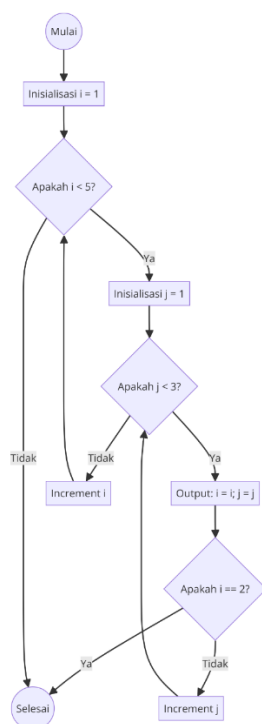
Analisis luaran dan code :

Kode ini bertujuan untuk mencetak piramida terbalik berdasarkan input tinggi dari pengguna.

Misalnya, jika pengguna memasukkan angka 7, kita ingin mencetak 7 baris bintang, mulai dari 7 bintang di baris pertama hingga 1 bintang di baris terakhir.

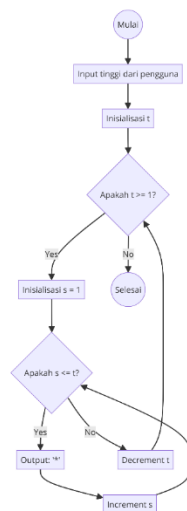
Yang pertama yaitu mengambil input dari pengguna seperti contoh diatas input yang masuk adalah 7 Loop pertama (mencetak tinggi pyramid), dimulai dari tinggi (7) dan berkurang hingga 1. Jadi, loop ini akan menjalankan 7 iterasi (1 untuk setiap baris). Loop kedua (mencetak Bintang) Loop kedua mulai dari 1 hingga nilai t (yang saat ini merupakan tinggi baris yang sedang dicetak).Ini berarti, di baris pertama (t=7), kita mencetak 7 bintang; di baris kedua (t=6), kita mencetak 6 bintang; dan seterusnya hingga baris terakhir (t=1) di mana kita mencetak 1 bintang. Setelah mencetak bintang untuk satu baris, kita menambahkan perintah untuk membuat baris baru (System.out.println();), sehingga bintang berikutnya akan dicetak di baris yang berbeda.

1.4 analisis flowchart 1.2 dan 1.3



Analisis : Flowchart akan memiliki dua loop bersarang, di mana loop i bersifat eksternal dan loop j bersifat internal. Label pertama dan kedua ditandai pada loop ini. Penggunaan continue atau break mengubah jalur eksekusi di flowchart dengan melewati iterasi atau menghentikan eksekusi loop.

Flowchart 1.3



Analisis : Flowchart dimulai dari input pengguna untuk tinggi piramida. Dua loop bersarang digunakan, dengan loop eksternal mengatur tinggi piramida dan loop internal mengatur jumlah bintang. Flowchart ini menunjukkan bagaimana setiap baris baru dihasilkan dengan jumlah bintang yang berkurang seiring berjalannya loop.

program dibuat telah sesuai dengan output yang diinginkan

[No.1] Kesimpulan

(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)

1) Analisa

- Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
- Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawab

- Saya dapat menyimpulkan penggunaan kata kunci continue dan break dalam program java seperti halnya pada contoh 1 kata kunci continue digunakan untuk melompati iterasi tertentu, sedangkan break menghentikan loop sepenuhnya. mengubah nilai dan kondisi dalam loop menyebabkan output yang berbeda, seperti mencetak angka ganjil atau angka genap hingga nilai tertentu. kemudian pemahaman tentang looping bersarang pada piramida menggunakan loop bersarang untuk menghasilkan pola piramida berdasarkan input pengguna. Dengan mengatur pengurangan jumlah bintang di setiap baris, pola piramida terbentuk, dan setiap baris baru dicetak dengan satu bintang lebih sedikit dari baris sebelumnya.
- Pengambilan Keputusan didasari oleh pemahaman tentang control alur program menggunakan kata kunci continue dan break, serta penerapan loop bersarang untuk memecahkan masalah mencetak sebuah piramida.

[No. 2] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variable

Contoh 4: Salin dan tempel kode program berikut ke Eclipse.

```
public class ContohWhile{
    public static void main(String[] args) {
        int i=1;
        while(i<=6){
            System.out.println(i);
            i++;
            if(i==4){
                break;    //ubah1
            }
        }
    }
}
```

Contoh 5: Salin dan tempel kode program berikut ke Eclipse.

```
public class WhileBersarang {
    public static void main(String[] args) {
        int count = 0; //ubah1
        while (count < 20) {
            if (count % 3 == 0) //ubah2
                System.out.println(count);
            count++;
        }
    }
}
```

Latihan 2

- 2.1. Buat perubahan nilai angka pada variabel di Contoh 4
//Ubah 1 menjadi continue; lalu running, periksa hasilnya
Analisa dampaknya perubahan terhadap luaran setelah running dan uraikan kegunaan break dan continue!
- 2.2. Buat perubahan nilai angka pada variabel di Contoh 5
//Ubah2 menjadi if (count % 5 == 0) lalu running, periksa hasilnya
Analisa dampaknya perubahan terhadap luaran setelah running dan uraikan kegunaan % untuk angka yang berbeda pada perintah tersebut!
- 2.3. Buat perubahan nilai angka pada variabel di
//Ubah1 menjadi while (count < 0) { lalu running, periksa hasilnya
Ubahlah baris kode while pada Contoh 5 menjadi do ... while dengan persyaratan yang sama while (count < 0). Bandingkan hasil luaran antara menggunakan while dan do ... while!
- 2.4. Analisa diagram flowchart dari Latihan 2.1, Contoh 5, dan Latihan 2.3!

[No.2] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara memperbaiki logika pengulangan continue dan mengganti kondisi untuk mendapatkan hasil sesuai kebutuhan.
- 2) Alasan solusi ini karena continue memberikan memberikan fleksibilitas untuk melewati Batasan tertentu tanpa menghentikan perulangan sepenuhnya
- 3) Perbaiki kode program dengan cara menggunakan kondisi logika yang tepat seperti mengganti while dan do while untuk memastikan setidaknya satu iterasi dijalankan sebelum kondisi diperiksa.

[No.2] Penyusunan Algoritma dan Kode Program

1) Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

2.1 ubah break menjadi continue pada contoh 4

- (a) Mulai
- (b) Inisialisasi variable i dengan nilai 1
- (c) Selama i kurang dari atau sama dengan 6:
Cetak i, tambahkan nilai i dengan 1. Jika i sama dengan 4,, lewati iterasi berikutnya menggunakan continue
- (d) Akhiri program

2.2 Ubah kondisi if (count % 3 == 0) menjadi if (count % 5 == 0) pada Contoh 5

- (a) mulai
- (b) inisialisasi variable count dengan nilai 0
- (c) Selama count kurang dari 20:
Jika count habis dibagi 5, cetak nilai count. Tambahkan nilai count dengan 1.
- (d) akhiri program

2.3 Ubah while (count < 0) menjadi do-while (count < 0) pada Contoh 5

- (a) mulai
- (b) inisialisasi variable count dengan nilai 0
- (c) Jika count kurang dari 0:
Jika count habis dibagi 3, cetak nilai count. Tambahkan nilai count dengan 1.
- (d) akhiri program

2) Kode program dan luaran

a) Screenshot/ Capture potongan kode dan hasil luaran 2.1

<pre>1- public class ContohWhile { 2- public static void main(String[] args) { 3- int i = 1; 4- while(i <= 6){ 5- System.out.println(i); 6- i++; 7- if(i == 4){ 8- continue; //ubah1 9- } 10- } 11- } 12- } 13- }</pre>	<pre>java -cp /tmp/z86mje8FW9/ContohWhile 1 2 3 4 5 6 === Code Execution Successful ===</pre>
--	---

b) Analisa luaran yang dihasilkan

Sebelum perubahan (dengan break): Program berhenti ketika nilai i mencapai 4

Setelah perubahan (dengan continue): Saat nilai i sama dengan 4, program akan melewati cetak angka 4 dan lanjut ke iterasi berikutnya.

c) Screenshot/ Capture potongan kode dan hasil luaran 2.12

<pre>1- public class WhileBersarang { 2- public static void main(String[] args) { 3- int count = 0; //ubah1 4- while (count < 20) { 5- if (count % 5 == 0) //ubah2 6- System.out.println(count); 7- count++; 8- } 9- } 10- } 11- }</pre>	<pre>java -cp /tmp/0b50hIhKHZ/WhileBersarang 0 5 10 15 === Code Execution Successful ===</pre>
---	--

d) Analisa luaran yang dihasilkan

Sebelum perubahan (dengan % 3): Output akan menampilkan angka yang habis dibagi 3:
setelah diubah output mengasilkan angka yang habis dibagi 5

e) Screenshot/ Capture potongan kode dan hasil luaran 2.3

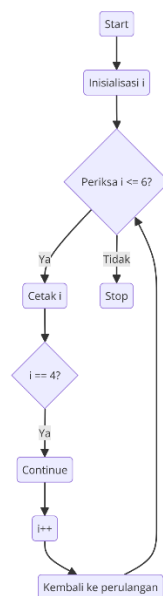
```
1- public class WhileBersarang {
2-     public static void main(String[] args) {
3-         int count = 0; //rubah!
4-     do {
5-         System.out.println(count);
6-         count++;
7-     } while (count < 0);
8-
9-     }
10 }
```

```
java -cp /tmp/etp1pUXvUK/WhileBersarang
0
=== Code Execution Successful ===
```

f) Analisa luaran yang dihasilkan

Sebelum perubahan (dengan while): Karena kondisi awal count = 0 dan tidak memenuhi syarat count < 0, tidak ada output yang dihasilkan. Loop tidak akan dijalankan sama sekali.
Setelah perubahan (dengan do ... while): Karena do ... while menjalankan blok kode setidaknya sekali sebelum memeriksa kondisi, output akan mencetak angka 0 satu kali, meskipun kondisi count < 0 tidak terpenuhi.

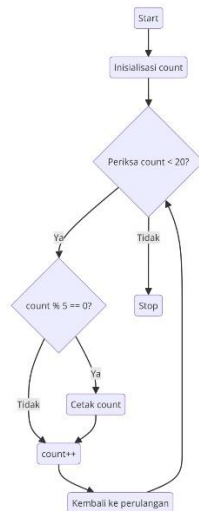
g) Flowchart Latihan 2.1 (Contoh 4):



h) Analisa luaran yang dihasilkan

Mulai atau start kemudian dilanjutkan dengan Inisialisasi i = 1. Jika ya cek kondisi i <= 6. Jika kondisi terpenuhi, cetak i. Tambahkan i dengan 1. Jika i == 4, lewati cetak dan lanjutkan iterasi berikutnya. Jika i <= 6, kembali ke langkah 4, jika tidak, berhenti.

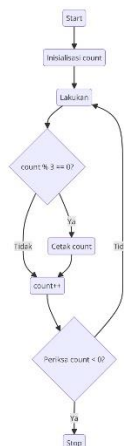
i) **Flowchart Contoh 5 (Latihan 2.2):**



j) **Analisa luaran yang dihasilkan**

Start atau mulai kemudian Inisialisasi count = 0. Cek kondisi count < 20. Jika kondisi terpenuhi, cek apakah count habis dibagi 5 (count % 5 == 0). Jika iya, cetak count. Tambahkan count dengan 1. Ulangi dari langkah 3 jika kondisi masih terpenuhi, jika tidak, berhenti.

k) **Flowchart Latihan 2.3:**



l) **Analisa luaran yang dihasilkan**

Mulai, Inisialisasi count = 0. Cetak count, tambahkan count dengan , cek kondisi count < 0, jika kondisi terpenuhi, kembali ke langkah 3, jika tidak, berhenti.

[No.2] Kesimpulan

(PILIH SALAH SATU ANDA INGIN MEMBAHAS DENGAN CARA ANALISA/ EVALUASI / KREASI)

1) Analisa

- Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
- Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

jawaban Analisa:

a) dalam lembaran soal ini dapat dianalisa berbagai kegunaan dari, break: Menghentikan loop sepenuhnya ketika kondisi dipenuhi. continue: Melewati iterasi yang sedang berlangsung dan langsung ke iterasi berikutnya tanpa menghentikan loop. break: Menghentikan loop sepenuhnya ketika kondisi dipenuhi. do ... while menjalankan blok kode setidaknya sekali sebelum memeriksa kondisi

b) pengambilan keputusan didasari dengan berbagai macam percobaan dan pemahaman materi selama dilaksanakannya praktikum

Refleksi

Saya banyak sekali belajar tentang pengcodingan java dengan benar melalui implementasi pengajaran selama berlangsungnya praktikum.