

| | | |
|--|----------------------|------------------------|
| Nama & NPM | Topik: | Tanggal: |
| Farrel Alvaro Alinskie .M G1F024024 | Unit satu FOR | 10 oktober 2024 |

[No. 1] Identifikasi Masalah:

```
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 + " ");
    } } }
```

Luaran contoh 1:

0
2
4
6

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
                } } } }
```

Luaran Contoh 2:

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

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
        }
    }
}

```

Luaran contoh 3:

Masukan Input: 7

```

*
**
***
****
*****
*****
*****

```

Latihan 1

- 1.1. Analisa tujuan penulisan kata kunci continue dan break pada Contoh 1!
 Buat perubahan nilai angka pada variabel di
 //Ubah 1 menjadi for (int y = 0; y <= 15; y++) { lalu running, periksa hasilnya
 //Ubah 2 menjadi if (y % 2 == 0) lalu running, periksa hasilnya
 //Ubah 3 menjadi else if (y == 9) lalu running, periksa hasilnya
 Analisa dampaknya perubahan ini terhadap luaran setelah running!
- 1.2. Buat perubahan kode pada Contoh 2 di baris //Ubah1 menjadi
 - a. continue pertama; lalu running, periksa hasilnya
 - b. break pertama; lalu running, periksa hasilnya
 - c. continue kedua; lalu running, periksa hasilnya
 Analisa perbedaan perubahan kode pada Ubah 1 untuk setiap poin (a), (b), dan (c)!
- 1.3. Cermati kode contoh 3. Apabila ingin menghasilkan luaran berikut:
 Luaran:
 Masukan Input: 7

```

*****
*****
*****
*****

```

**
*

Susunlah analisa kode untuk menghasilkan luaran tersebut!

1.4. Analisa diagram flowchart dari Latihan 1.2 dan 1.3!

[No.1]jawaban

1.1Kode Awal Contoh 1

java

Salin kode

```
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 + " ");
}
```

Analisa Kata Kunci

- continue: Memaksa loop untuk melanjutkan ke iterasi berikutnya. Dalam kasus ini, jika y adalah angka ganjil (dari kondisi if), continue akan mengabaikan sisa kode dalam loop untuk iterasi itu dan melanjutkan ke angka berikutnya.
- break: Menghentikan loop sepenuhnya. Jika y mencapai 8, maka break akan menghentikan eksekusi loop for.

Perubahan yang Dilakukan

1. Ubah 1: Mengubah nilai batas dari y <= 10 menjadi y <= 15:
 - Output yang dihasilkan: 0, 2, 4, 6, 8, 10, 12, 14.
2. Ubah 2: Mengubah kondisi menjadi if (y % 2 == 0):
 - Output yang dihasilkan: 0, 2, 4, 6, 8, 10, 12, 14. (Sekarang mencetak angka genap, tidak mencetak yang ganjil).
3. Ubah 3: Mengubah kondisi menjadi else if (y == 9):
 - Output yang dihasilkan: 0, 2, 4, 6, 8. (Ketika y mencapai 9, loop berhenti).

Dampak Perubahan

- Perubahan di Ubah 1 memperluas rentang iterasi, sehingga lebih banyak angka yang dicetak.

- Perubahan di Ubah 2 mengubah logika menjadi hanya mencetak angka genap.
- Perubahan di Ubah 3 mengubah titik berhenti dari 8 menjadi 9, sehingga semua angka genap di bawah 9 dicetak sebelum loop berhenti.

1.2 Analisis untuk Masing-Masing Ubah 1

1. a. continue pertama;

- Ketika i adalah 2, loop for untuk j akan melanjutkan ke iterasi berikutnya. Hasilnya, i = 2; j = 1 akan dicetak, tetapi j tidak akan mencetak 2, dan program melanjutkan ke i = 3.

2. b. break pertama;

- break pertama akan menghentikan loop for luar (loop i), sehingga tidak ada output setelah i = 2.

3. c. continue kedua;

- continue kedua akan menyebabkan loop j untuk melanjutkan iterasi, tetapi tidak mempengaruhi loop i. Outputnya akan mencetak i = 1; j = 1, i = 1; j = 2, i = 2; j = 1, dan loop j untuk i = 2 akan melanjutkan, tetapi tidak mencetak i = 2; j = 2, sehingga output untuk i = 2 hanya sampai j = 1.

Perbandingan Perubahan Kode

- **continue pertama;** Mencetak semua nilai j kecuali ketika i adalah 2, sehingga tidak mencetak j = 2 untuk i = 2.
- **break pertama;** Menghentikan loop luar sepenuhnya, sehingga tidak ada output lebih lanjut setelah i = 2.
- **continue kedua;** Mencetak semua output sebelum j = 2 untuk i = 2, menghentikan eksekusi untuk j tetapi tidak mempengaruhi loop i.

1.3 Perubahan yang Diperlukan

- **Menghitung Jumlah Spasi:** Ubah loop kedua untuk menghitung jumlah bintang.

Susunan Kode yang Tepat

Ubah loop seperti berikut:

java

program:

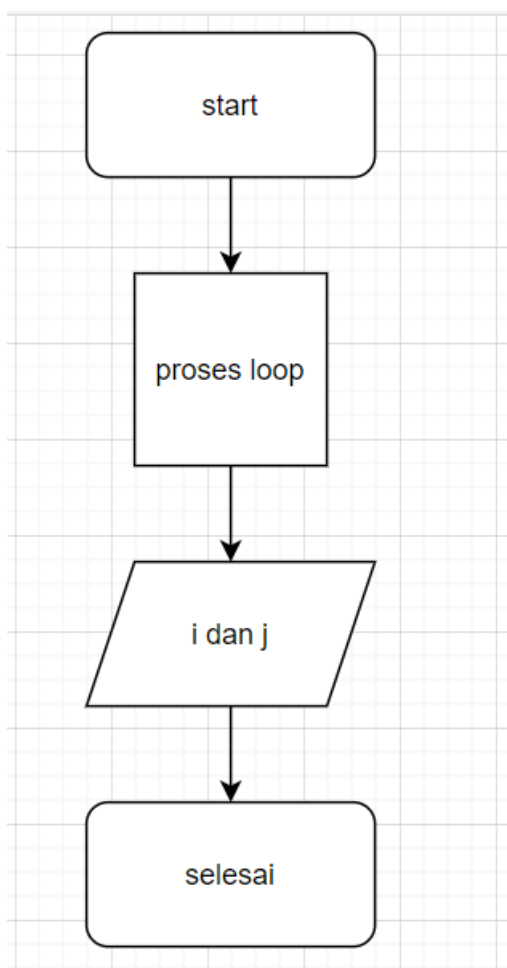
```
for (int t = tinggi; t >= 1; t--) {
    for (int s = 1; s <= t; s++) { // Mengubah loop ini
        System.out.print("*");
    }
}
```

```
System.out.println();  
}
```

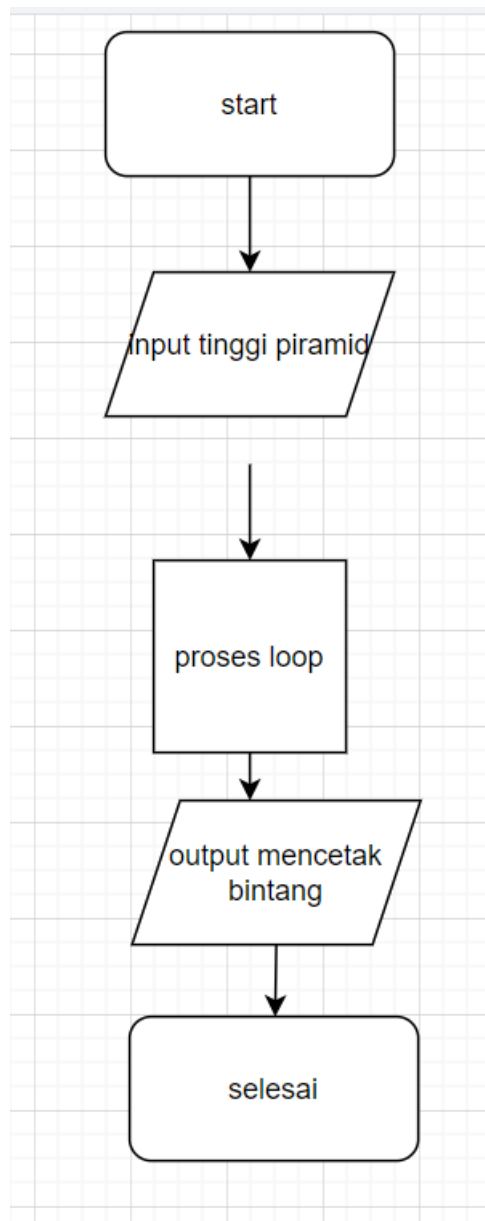
Penjelasan

- **Loop Pertama (t):** Mengurangi tinggi setiap kali, dimulai dari tinggi yang diberikan.
- **Loop Kedua (s):** Menghitung dari 1 sampai t, mencetak bintang berdasarkan nilai t untuk menghasilkan piramida terbalik.

1.4flowchart Latihan 1.2



b.flowchart Latihan 1.3



| Nama & NPM | Topik: | Tanggal: |
|--|----------------|-----------------|
| Farrel Alvaro Alinskie .M G1F024024 | Unit dua WHILE | 10 oktober 2024 |

[No. 2] Identifikasi Masalah:

```
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
        }
    }
}
```

Luaran:

1
2
3

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++;
        }
    }
}
```

Luaran:

0
3
6
9
12
15
18

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]jawaban

2.1

Jika kita mengubah break; menjadi continue;, kode menjadi:

java

```
if (i == 4) {  
    continue;    //ubah1  
}
```

Luaran Setelah Perubahan

1. Output yang Dihasilkan:

○ Sebelum:

1
2
3

○ Setelah:

1
2
3
5
6

Analisa Dampak

- Dengan menggunakan continue;, saat i mencapai 4, perulangan tidak berhenti. Sebaliknya, perulangan melanjutkan iterasi berikutnya, dan 4 tidak dicetak. Sehingga angka 5 dan 6 tetap dicetak.
- Kegunaan:
 - break: Menghentikan loop sepenuhnya ketika kondisi tertentu terpenuhi.
 - continue: Melewatkan iterasi saat ini dan melanjutkan ke iterasi berikutnya tanpa mengeksekusi kode di bawahnya.

2.2Ubah 2: if (count % 5 == 0)

Jika kita mengubah menjadi:


```
if (count % 5 == 0) //ubah2
```

Luaran Setelah Perubahan

1. Output yang Dihasilkan:

○ Sebelum:

```
0
3
6
9
12
15
18
```

○ Setelah:

```
0
5
10
15
```

Analisa Dampak

- Setelah perubahan, kode hanya mencetak angka yang merupakan kelipatan 5 dalam rentang dari 0 hingga 20. Hasilnya menunjukkan perubahan dalam pola output.
- **Kegunaan %:** Operator modulus (%) digunakan untuk menemukan sisa dari pembagian dua angka. Dalam konteks ini, `count % 3 == 0` mencetak angka yang habis dibagi 3, sedangkan `count % 5 == 0` mencetak angka yang habis dibagi 5.

2.3Ubah 1: while (count < 0)

Jika kita mengubah while menjadi:

```
while (count < 0) {
    // kode di sini
}
```

Ubah Menjadi do...while

Jika kita ingin mengubahnya menjadi do...while, kodenya akan menjadi:

```
java
do {
    if (count % 5 == 0)
```

```
System.out.println(count);  
  
count++;  
} while (count < 0);
```

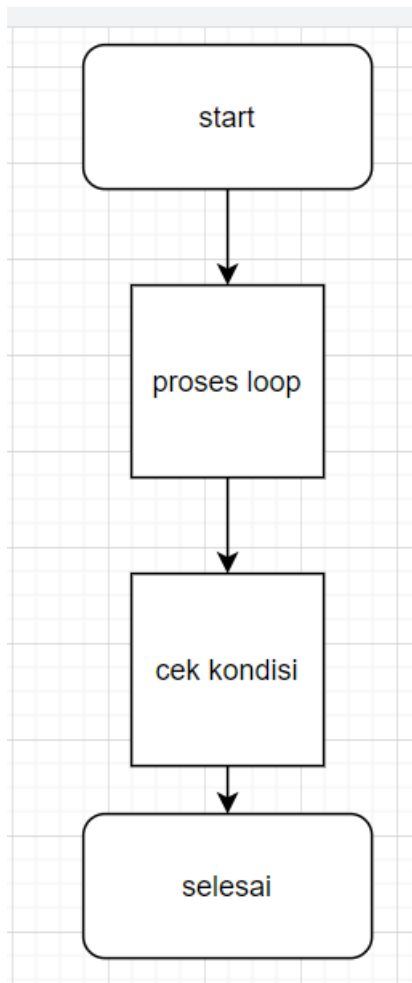
Luaran Setelah Perubahan

1. **Output untuk while (count < 0):** Tidak akan ada output, karena kondisi awal tidak terpenuhi (count dimulai dari 0).
2. **Output untuk do...while:**
 - o Walaupun kondisinya $\text{count} < 0$, do...while akan tetap menjalankan blok setidaknya sekali. Namun, karena count adalah 0, hasilnya tidak akan mencetak apapun dan tetap berhenti.

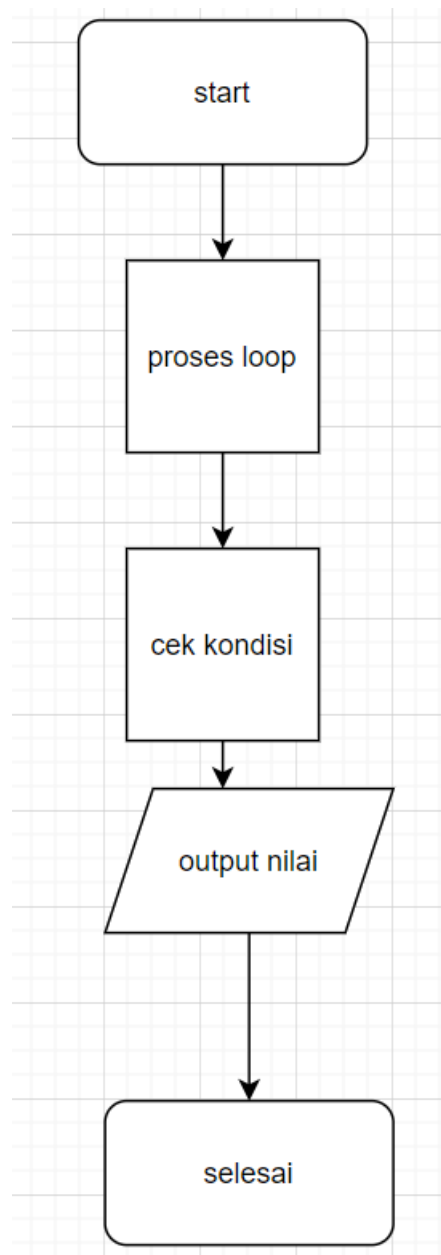
Bandingkan Hasil Luaran

- **while (count < 0):** Tidak ada output karena tidak memenuhi syarat.
- **do...while (count < 0):** Juga tidak ada output setelah iterasi pertama, tetapi do memastikan setidaknya satu kali eksekusi.

2.4 flowchart Latihan 2.1



Flowchart dari contoh 5:



Flowchart Latihan 2.3

