

Nama & NPM	Topik:	Tanggal:
Ivanaly Dees Tumangger G1F024033	FOR	2 Oktober 2024
[Nomor Soal] Identifikasi Masalah:		
<p>1) Uraikan permasalahan dan variabel</p> <pre> 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 + " "); } } } </pre> <p>Luaran contoh 1:</p> <pre> 0 2 4 6 </pre> <p>Contoh 2: Salin dan tempel kode program berikut ke Eclipse.</p> <pre> 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 } } } } </pre> <p>Luaran Contoh 2:</p> <pre> 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 </pre> <p>Contoh 3: Salin dan tempel kode program berikut ke Eclipse.</p> <pre> import java.util.Scanner; public class ForBersarang { public static void main(String[] args){ //Instance Input Scanner </pre>		

```

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!

1.1. Analisa tujuan penulisan kata kunci continue dan break pada Contoh 1!

Jawaban: Tujuan penulisan kata kunci continue dan break pada Contoh 1 adalah untuk mengontrol aliran eksekusi perulangan:

- continue: Untuk mengontrol alur loop atau perulangan dengan mengubah kode pada contoh 1 dan menghasilkan luaran bilangan ganjil maka continue bertujuan untuk mencetak bilangan ganjil dan bilangan genap akan di lewati dan tidak tercetak.
- break: Membatasi perulangan hingga nilai tertentu (9) sehingga hanya bilangan ganjil dari 1-9 yang akan dicetak.

Secara keseluruhan, kode ini akan mencetak bilangan ganjil dari 1 hingga 9.

//Ubah 1 menjadi for (int y = 0; y <= 15; y++) { lalu running, periksa hasilnya

Main.java	Output
<pre>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 } 11 } 12 } 13 }</pre>	<pre>java -cp /tmp/QN3xEBZwyM/ContohFor 1 3 5 7 === Code Execution Successful ===</pre>

Dengan mengubah kode menjadi for (int y = 0; y <= 15; y++) maka perubahan nilai yang dicetak adalah kondisi perulangan dimana jumlah perulangan atau loop yang dilakukan akan berubah dan terjadi perubahan urutan di output.

//Ubah 2 menjadi if (y % 2 == 0) lalu running, periksa hasilnya

Main.java	Output
<pre>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 } } }</pre>	<pre>java -cp /tmp/rCbaTisHEg/ContohFor 1 3 5 7 9 === Code Execution Successful ===</pre>

Dengan mengubah kode menjadi if (y % 2 == 0) maka kondisi ini akan memeriksa apakah nilai y habis dibagi 2 (bilangan genap) jika tidak habis dibagi maka ouputnya akan berisi bilangan ganjil.

//Ubah 3 menjadi else if (y == 9) lalu running, periksa hasilnya

Main.java	Output
<pre>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 } } }</pre>	<pre>java -cp /tmp/hmtbqb1ref/ContohFor 0 2 4 6 === Code Execution Successful ===</pre>

Dengan mengubah kode menjadi else if (y == 9) Perulangan akan berhenti ketika nilai y mencapai 6 dan output akan berisi angka-angka dari 0 sampai 6.

- 1.2. Buat perubahan kode pada Contoh 2 di baris //Ubah1 menjadi
- continue pertama; lalu running, periksa hasilnya

The screenshot shows a Java IDE with a file named 'Main.java'. The code is as follows:

```
1 public class ForBersarang {
2     public static void main(String[] args) {
3         pertama:
4         for( int i = 1; i < 5; i++) {
5
6             kedua:
7             for(int j = 1; j < 3; j ++ ) {
8                 System.out.println("i = " + i + "; j = " + j);
9                 if ( i == 2)
10                     continue pertama; //ubah1
11             }
12         }
13     }
14 }
```

The output window shows the following results:

```
java -cp /tmp/fdFinzPjsP/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 ===
```

Dengan mengubah continue pertama maka akan menampilkan output yang berurutan dan berpasangan seperti pada gambar diatas.

- break pertama; lalu running, periksa hasilnya

The screenshot shows a Java IDE with a file named 'Main.java'. The code is as follows:

```
1 public class ForBersarang {
2     public static void main(String[] args) {
3         pertama:
4         for( int i = 1; i < 5; i++) {
5
6             kedua:
7             for(int j = 1; j < 3; j ++ ) {
8                 System.out.println("i = " + i + "; j = " + j);
9                 if ( i == 2)
10                     break pertama; //ubah1
11             }
12         }
13     }
14 }
```

The output window shows the following results:

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

Dengan mengubah break pertama maka akan menampilkan output yang lebih singkat daripada continue tetapi tetap berurutan seperti pada gambar diatas.

- continue kedua; lalu running, periksa hasilnya

The screenshot shows a Java IDE with a file named 'Main.java'. The code is as follows:

```
1 public class ForBersarang {
2     public static void main(String[] args) {
3         pertama:
4         for( int i = 1; i < 5; i++) {
5
6             kedua:
7             for(int j = 1; j < 3; j ++ ) {
8                 System.out.println("i = " + i + "; j = " + j);
9                 if ( i == 2)
10                     continue kedua; //ubah1
11             }
12         }
13     }
14 }
```

The output window shows the following results:

```
java -cp /tmp/1n7R1PbvPy/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 ===
```

Dengan menggunakan continue ke dua maka akan menghasilkan output yang lebih Panjang dari pada pertama.

Analisa perbedaan perubahan kode pada Ubah 1 untuk setiap poin (a), (b), dan (c)!

Jawab: Perbedaan pada kode (a), (b) dan (c) adalah pada continue pertama program akan langsung menampilkan output ke perulangan berikutnya dari perulangan yang berlabel pertama, dimana sisa perulangan pada perulangan kedua yang sedang berjalan akan diabaikan atau dilewati. Pada break pertama perulangan berhenti sebelum selesai, kombinasi nilai i dan j sebelum kondisi break terpenuhi akan dicetak sehingga perulangan pertama akan dihentikan sepenuhnya. Sedangkan pada continue kedua program akan langsung ke perulangan berikutnya dari loop atau perulangan kedua.

1.3. Cermati kode contoh 3. Apabila ingin menghasilkan luaran berikut:

```
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 Dari User
9
10        for (int t = tinggi; t >= 1; t--) {
11            // Menghitung jumlah tinggi bintang
12            for (int s = 1; s <= t; s++) {
13                System.out.print("*");
14            }
15            System.out.println(); // Membuat Baris Baru
16        }
17
18        input.close(); // Menutup Scanner
19    }
20 }
```

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

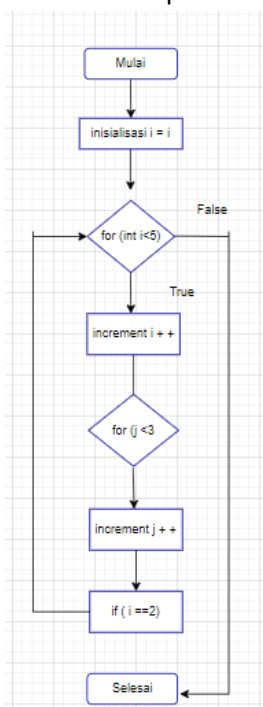
=== Code Execution Successful ===
```

Dengan memasukkan variabel tinggi Bintang untuk membentuk sebuah piramida , sehingga melakukan perulangan pertama for (int t = tinggi; t >= 1; t--) akan berulang dari tinggi yang dimasukkan sampai 1, yang artinya kode yang diinput kedalam program akan menghasilkan baris dari penuh ke kosong. Perulangan kedua for (int s = 1; s <= t; s++) akan mencetak bintang sesuai dengan nilai t untuk setiap perulangan. Pada perulangan pertama, t adalah tinggi yang dimasukkan, dan setiap perulangan berikutnya maka jumlah bintang akan selalu berkurang. Setelah perulangan kedua, System.out.println(); digunakan untuk mencetak baris sehingga program yang dijalankan akan mencetak seluruh bintang. Pada setiap iterasi jumlah Bintang yang di cetak menurun ,dimulai dari tertinggi sampai berkurang satu persatu setiap barisnya .dan menghasilkan piramida yang terbalik.

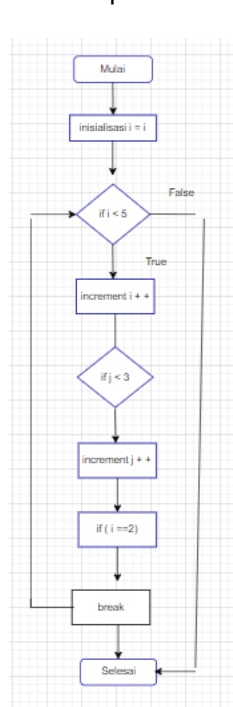
1.4. Analisa diagram flowchart dari Latihan 1.2 dan 1.3!

Flowchart 1.2

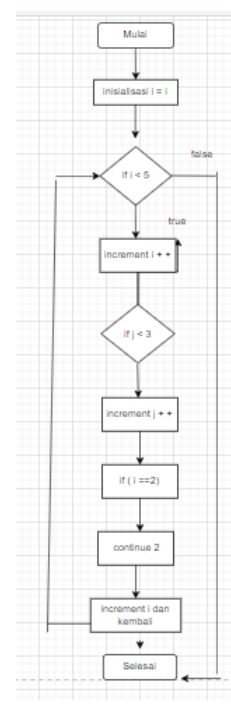
a. Continue pertama



b. break pertama

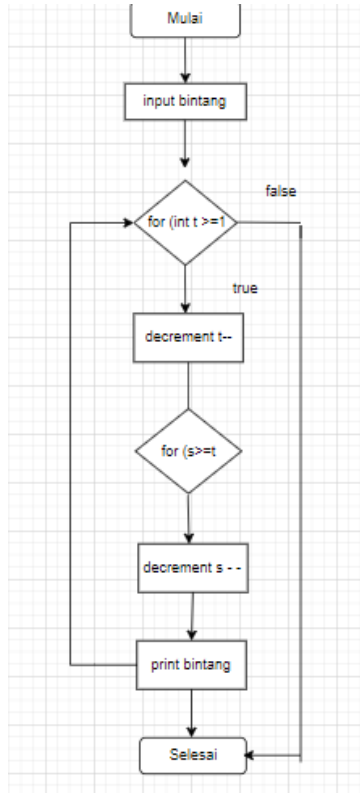


c. continue kedua



Analisis dari ketiga flowchart itu adalah menggambarkan proses input pengguna Dimana memasukkan variabel dan di tentukan false atau true, jika false maka program berhenti jika true maka program dijalankan.

Flowchart 1.3



- 2) Rincikan sumber informasi yang relevan (buku / webpage)
- 3) Uraikan rancangan solusi yang diusulkan (jika ada).
- 4) Analisis susunan solusi, parameter solusi (jika ada).

[Nomor Soal] Analisis dan Argumentasi

- 1) Uraikan rancangan solusi yang diusulkan.

Jawaban: Rancangan Solusi yang di usulkan pada contoh 1.1 adalah setelah perubahan kode pada program kita mengetahui bahwa peran continue adalah untuk melewati angka yang memenuhi kondisi tertentu . Break untuk menghentikan loop jika angka tertentu tercapai.

Contoh 1.2 solusinya adalah program dengan nested loop dapat diatur dengan continue dan break untuk mengontrol jalannya perulangan baik untuk melompati atau menghentikan loop sesuai kondisi .

Contoh 1.3 untuk menghasilkan piramida yang terbalik maka Solusi yang tepat adalah mengubah loop for yang mencetak jumlah Bintang menurun, dimulai dari angka yang bernilai besar sampai angka terkecil.

- 2) Analisis solusi, kaitkan dengan permasalahan.

Jawaban : Solusi yang diusulkan pada kode program sudah tepat agar permasalahan kode yang terjadi dapat terselesaikan dan kode dapat berjalan sesuai harapan . Dimana permasalahannya tentang bagaimana penggunaan continue dan break mempengaruhi suatu alur program dalam perulangan. Dimana dengan perulangan yang dilakukan kita bisa menganalisis bagaimana suatu perulangan terjadi dan kapan perulangan harus di lanjutkan dan di hentikan.

[Nomor Soal] Penyusunan Algoritma dan Kode Program

1) Rancang desain solusi atau algoritma

Jawaban: Algoritma adalah Langkah Langkah dalam pemograman untuk mencapai tujuan yang diinginkan

Algoritma 1.1

- Mulai program
- Menentukan batas loop yang akan dijalankan
- Melakukan pengecekan apakah y genap atau ganjil
- Mengecek apakah y sama dengan 9
- Mencetak nilai y
- Akhiri program

Algoritma 1.2

- Mulai program
- Menentukan nilai loop pertama
- Menentukan nilai loop kedua
- Mencetak nilai hasil dari loop pertama dan kedua
- Melakukan perulangan hingga semua iterasi selesai
- Akhiri program.

Algoritma 1.3

- Mulai program
- Memasukkan nilai untuk menjalankan program
- Melakukan perulangan atau loop yang berjalan dari t
- Mencetak Bintang untuk menghasilkan piramida terbalik
- Pindah ke baris baru setelah mencetak Bintang
- Melakukan perulangan hingga seluruh baris piramida tercetak dari atas ke bawah
- Setelah perulangan selesai tutup objek scanner untuk menghentikan input
- Akhiri program

2) Tuliskan kode program dan luaran

- Beri komentar pada kode

Jawaban : komentar pada kode yaitu kode yang digunakan sudah sesuai pada program sehingga Program yang dijalankan menghasilkan luaran yang sesuai. Seperti import scanner Digunakan untuk mengimpor kelas scanner yang diperlukan untuk membaca input Dari pengguna.

- Uraikan luaran yang dihasilkan

Luaran yang dihasilkan dari program sudah tepat tergantung pada nilai input yang Diberikan oleh pengguna dari setiap masing masing soal.

- Screenshot/ Capture potongan kode dan hasil luaran

Main.java	Output
<pre>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 } 11 } 12 } 13</pre>	<pre>java -cp /tmp/QN3xEBZwyM/ContohFor 1 3 5 7 === Code Execution Successful ===</pre>

Main.java	Output
<pre> 1- public class ForBersarang { 2- public static void main(String[] args) { 3- pertama: 4- for(int i = 1; i < 5; i++) { 5- 6- kedua: 7- for(int j = 1; j < 3; j ++) { 8- System.out.println("i = " + i + "; j = " + j); 9- if (i == 2) 10- continue pertama; //ubah! 11- } } } </pre>	<pre> java -cp /tmp/fdFinzPjsP/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 === </pre>

<pre> 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 Dari User 9 10- for (int t = tinggi; t >= 1; t--) { 11- // Menghitung jumlah tinggi bintang 12- for (int s = 1; s <= t; s++) { 13- System.out.print("*"); 14- } 15- System.out.println(); // Membuat Baris Baru 16- } 17 18- input.close(); // Menutup Scanner 19- } </pre>	<pre> java -cp /tmp/nnWfCoinUa/ForBersarang Masukan Input: 7 ***** ***** **** *** ** * === Code Execution Successful === </pre>
---	---

[Nomor Soal] Kesimpulan

1) Analisa

- Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program! Kesimpulannya adalah program ini efisien dalam hal kinerja karena jumlah iterasi dalam perulangan ditentukan secara langsung oleh input pengguna .ini memastikan bahwa program tidak akan melakukan lebih banyak iterasi dari pada yang diperlukan untuk mencapai output yang diinginkan. Dengan pendekatan yang sistematis dan terstruktur maka program akan dapat dikembangkan dan ditampilkan dalam menciptakan Solusi yang efisien.
- Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini? Dasar pengambilan keputusan yaitu penggunaan algoritma berbasis loop bersarang adalah Solusi yang tepat untuk mencetak pola piramida terbalik karena menciptakan struktur yang jelas dalam output dan memudahkan pengulangan dalam konteks yang lebih kompleks. Keputusan untuk menggunakan continue dan break pada loop adalah for adalah untuk mengontrol aliran program secara efektif.

Nama & NPM	Topik:	Tanggal:
Ivanaly Dees Tumangger G1F024033	WHILE	2 Oktober 2024

[Nomor 1] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variable

```
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!

Jawab:

2.1 //Ubah 1 menjadi continue; lalu running, periksa hasilnya

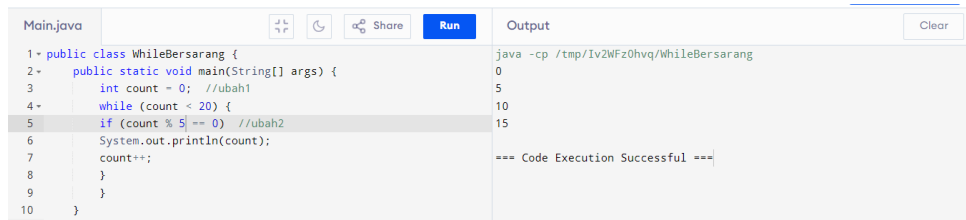


```
Main.java
1 public class ContohWhile{
2 public static void main(String[] args) {
3 int i=1;
4 while(i<=6){
5 System.out.println(i);
6 i++;
7 if(i==4){
8 continue; //ubah1
9 }
10 }
11 }
```

```
Output
java -cp /tmp/9HjVEG20gu/ContohWhile
1
2
3
4
5
6
=== Code Execution Successful ===
```

Pada gambar program diatas Dengan mengubah if(i==4) break; { menjadi if(i==4){ continue;} Perubahan yang terjadi adalah setelah menggunakan continue program melewati nilai 4 dan Melanjutkan perulangan dimana hasil outputnya adalah angka 1 sampai 6. Kegunaan break adalah untuk menghentikan seluruh perulangan saat kondisinya terpenuhi. Kegunaan continue adalah untuk menghentikan iterasi saat itu dan melanjutkan ke iterasi berikutnya,dan melewati perintah di bawahnya.

2.2 //Ubah2 menjadi if (count % 5 == 0) lalu running, periksa hasilnya

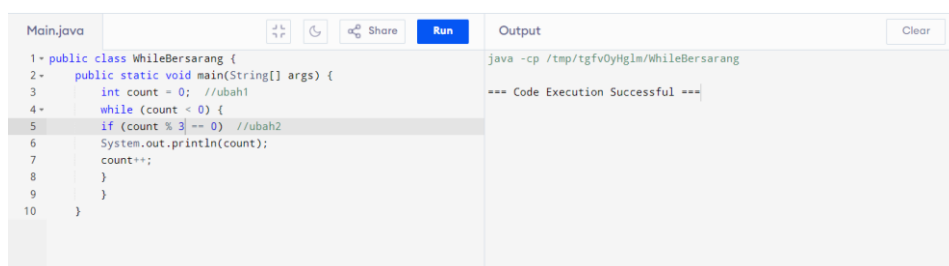


```
Main.java
1 public class WhileBersarang {
2 public static void main(String[] args) {
3 int count = 0; //ubah1
4 while (count < 20) {
5 if (count % 5 == 0) //ubah2
6 System.out.println(count);
7 count++;
8 }
9 }
10 }
```

```
Output
java -cp /tmp/Iv2WFz0hvq/WhileBersarang
0
5
10
15
=== Code Execution Successful ===
```

Pada gambar program tersebut diubah dari kode awal if (count % 3 == 0) menjadi if (count % == 0) maka hasil setelah perubahan adalah menampilkan angka yang habis dibagi 5, yaitu 0,5,10,15. Itulah hasil luaran yang akan ditampilkan. Kegunaan % adalah operator modulus (%) yang mengembangkan sisa hasil pembagian.

2.3 //Ubah1 menjadi while (count < 0) { lalu running, periksa hasilnya



```
Main.java
1 public class WhileBersarang {
2 public static void main(String[] args) {
3 int count = 0; //ubah1
4 while (count < 0) {
5 if (count % 3 == 0) //ubah2
6 System.out.println(count);
7 count++;
8 }
9 }
10 }
```

```
Output
java -cp /tmp/tgfv0yHgln/WhileBersarang
=== Code Execution Successful ===
```

Pada gambar diatas diubah kode awal dari while (count < 20) menjadi while (count < 0){dimana hasil perubahan karena count dimulai dari 0 dan kondisi while (count < 0) tidak terpenuhi sejak awal maka program tidak akan menghasilkan output apapun.

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!



```
1- public class WhileBersarang {
2-     public static void main(String[] args) {
3-         int count = 0; //ubah1
4-         do {
5-             if (count % 3 == 0) //ubah2
6-                 System.out.println(count);
7-             count++;
8-         } while (count < 0); // kondisi ini
9-     }
10- }
11- }
```

Output

```
java -cp /tmp/nm68Z0GurC/WhileBersarang
0
=== Code Execution Successful ===
```

Pada gambar kode program diatas maka diubah kode menjadi do while yaitu do {

if (count % 3 == 0) // ubah 2

System.out.println(count):

Count++;

} while (count < 0): // kondisi ini

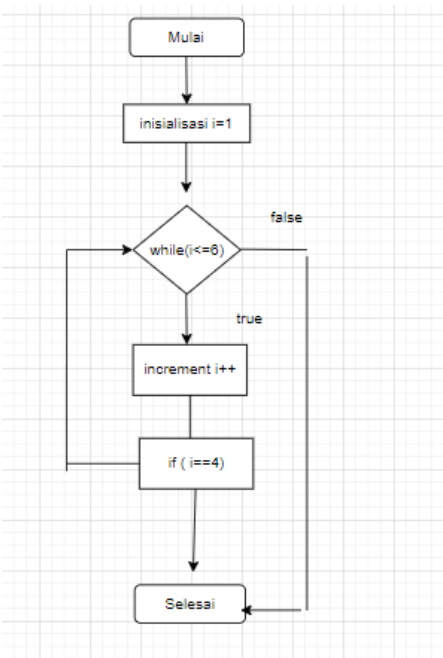
}

}

Pada kondisi menggunakan do while ,perulangan minimal dilakukan sekali walaupun kondisi while tidak terpenuhi . karena dalam do while kondisi dievaluasi setelah satu kali eksekusi blok kode. Sehingga outputnya menampilkan luaran 0.

2.4. Analisa diagram flowchart dari Latihan 2.1, Contoh 5, dan Latihan 2.3!

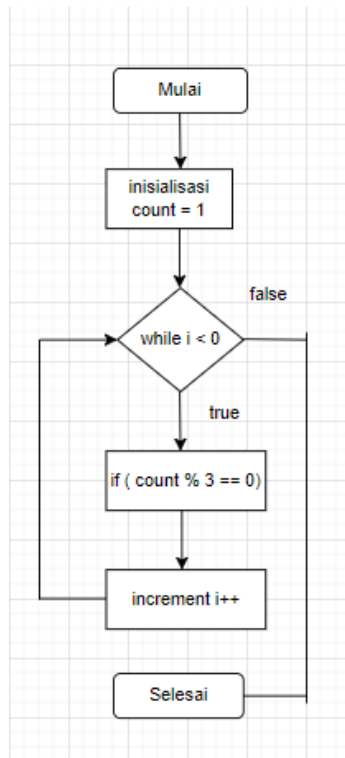
Flowchart 2.1



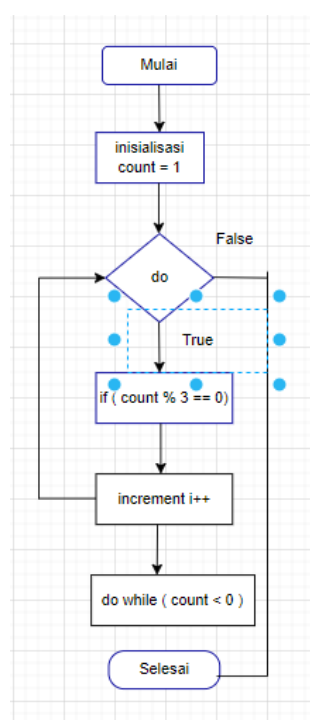
Analisa dari flowchart ini adalah menggambarkan proses atau langkah-langkah input pengguna yang memasukkan inisialisasi data dan variabel dan menentukan tipe data true atau false.

Flowchart 2.3

While



do while



Analisa dari flowchart while adalah memasukkan input pengguna dan memeriksa kondisi diawal Dimana jika kondisi terpenuhi sejak awal blok akan dieksekusi.

Analisa dari flowchart Do while adalah memasukkan variabel dan input pengguna dimana kondisi diperiksa di akhir blok kode didalam perulangan akan di eksekusi setidaknya sekali kemudian kondisi baru diperiksa.

- 2) Rincikan sumber informasi yang relevan (buku / webpage)
- 3) Uraikan rancangan solusi yang diusulkan (jika ada).
- 4) Analisis susunan solusi, parameter solusi (jika ada).

[Nomor 2] Analisis dan Argumentasi

- 1) Uraikan rancangan solusi yang diusulkan.

Jawaban: rancangan solusi yang diusulkan yaitu dengan mengubah break menjadi continue dimana letakkan perintah i++ pada pernyataan continue agar perulangan tidak masuk kedalam loop terbatas .

Dimana dengan menggunakan modulus % memungkinkan kita menyaring output berdasarkan kelipatan angka yang kita masukan. Karena mengubah angka dalam pernyataan modulus menghasilkan luaran yang berbeda.

- 2) Analisis solusi, kaitkan dengan permasalahan.

Jawaban : permasalahan pada soal 2.1 adalah mengubah break menjadi continue dan menganalisis dampaknya terhadap keluaran progam yang dijalankan,pada kode asli break menghentikan perulangan setelah kondisi i==4 tercapai dimana perulangan berhenti setelah mencetak angka 3.

Solusinya yaitu dengan mengganti break menjadi continue yang bertujuan untuk melewati angka 4 dan melanjutkan perulangan dimana perintah i++ harus ditempatkan sebelum continue agar perulangan dapat berjalan dengan lancar.

Permasalahan pada 2.2 yaitu mengubah kondisi modulus dari count % 3 menjadi count % 5. Operator modulus(%) digunakan untuk mendapatkan sisa dari pembagian. dimana jika count % 3 == 0 maka hanya angka kelipatan 3 yang dicetak .
Jika account % 5== 0 maka akan menghasilkan kelipatan angka 5 pada output program.

Permasalahan soal 2.3 mengubah while 9 count < 0) menjadi do...while
While loop digunakan untuk memeriksa kondisi sebelum menjalankan perulangan ,dimana jika Kondisi tidak terpenuhi dari awal, perulangan tidak aan terjadi sama sekali.
Do...while digunakan untuk memastikan blok kode dijalankan satu kali sebelum kondisi di Evaluasi , dimana jika kondisi tidak terpenuhi pada awalnya, perulangan akan tetap terjadi Sekali.

[Nomor 3] Penyusunan Algoritma dan Kode Program

- 1) Rancang desain solusi atau algoritma
Algoritma adalah langkah atau prosedur awal dalam pemrograman untuk menghasilkan dan mencapai tujuan tertentu.

Algoritma 2.1

- a) Mulai program
- b) Melakukan inisialisasi pada variable I dengan nilai 1
- c) Melakukan perulangan I <= 6: dan cetak nilai
- d) Menambahkan I dengan 1
- e) Ulangi langkah C sampai I lebih dari 6
- f) Mencetak hasil program
- g) Selesai

Aloritma 2.2

- a) Mulai program
- b) Inisialisasi variable count dengan nilai 0
- c) Melakukan perulangan selama count < 20
- d) Melakukan count habis dibagi 5 dan cetak nilainya
- e) Ulangi langkah C hingga count >= 20
- f) Mencetak hasil program
- g) Selesai

Algoritma 2.3 while

- a) Mulai
- b) Inisialisasi variable count dengan nilai 0
- c) Lakukan perulangan selama count < 0
- d) Tambahkan nilai count dengan 1
- e) Ulangi langkah C hingga count >= 0
- f) Cetak hasil
- g) Selesai

Algoritma do while

- a) Mulai program
- b) Inisialisasi variabel dengan count 0
- c) Mealakukan perulangan jika count % 3 == 0 cetak nilai count
- d) Tambahkan count dengan 1
- e) Ulangi langkah C hingga kondisi count >= 0
- f) Cetak hasil
- g) selesai

2) Tuliskan kode program dan luaran

a) Beri komentar pada kode

Jawaban :int count =0: digunakan untuk menginisialisasi variabel count dengan nilai awal 0.
While (count < 0): digunakan untuk mengecek apakah nilai count lebih kecil dari 0.
Karena count dimulai dari 0 mak kondisi ini tidak pernah terpenuhi yang akibatnya
Perulangan tidak akan pernah dijalankan.
If (count % 5 == 0): yaitu blok yang akan dijalankan jika nilai count habis dibagi 5.
Count++; adalah penambahan variabel count.

b) Uraikan luaran yang dihasilkan


Jawaban: luaran yang dihasilkan dari setiap program yang dijalankan berbeda beda dari
Soal latihan 2.1 – 2.3 tetapi luaran yang ditampilkan pada output sudah sesuai dengan
Permintaan Kode yang dimasukkan.

c) Screenshot/ Capture potongan kode dan hasil luaran

Main.java	Output
<pre>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 == 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+ } } }</pre>	<pre>java -cp /tmp/pybnXvri93/ContohFor 0 2 4 6 === Code Execution Successful ===</pre>

Main.java	Output
<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+ }</pre>	<pre>java -cp /tmp/Iv2WFz0hvfq/WhileBersarang 0 5 10 15 === Code Execution Successful ===</pre>

Main.java	Output
<pre>1+ public class WhileBersarang { 2+ public static void main(String[] args) { 3+ int count = 0; //ubah1 4+ do { 5+ if (count % 3 == 0) //ubah2 6+ System.out.println(count); 7+ count++; 8+ } while (count < 0); // kondisi ini 9+ } 10+ } 11+ }</pre>	<pre>java -cp /tmp/nm68ZOGurC/WhileBersarang 0 === Code Execution Successful ===</pre>

Main.java	   Share 	Output
<pre>1 public class WhileBersarang { 2 public static void main(String[] args) { 3 int count = 0; //ubah1 4 do { 5 if (count % 3 == 0) //ubah2 6 System.out.println(count); 7 count++; 8 } while (count < 0); // kondisi ini 9 } 10 } 11</pre>		<pre>java -cp /tmp/nm68ZOGurC/WhileBersarang 0 === Code Execution Successful ===</pre>

[Nomor 4] Kesimpulan

1) Analisa

- a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program! Kesimpulan adalah dalam penggunaan break dan continue memberikan dampak Hasil atau dampak yang berbeda pada setiap eksekusi program. Saat break dipakai Maka program menghentikan perulangan sedangkan pada saat menggunakan Continue program akan melewati iterasi dan melanjutkan ke iterasi berikutnya. Dengan menggunakan kode program if count modulus nilai variabel maka program Akan mencetak kelipatan angka yang dimasukkan pada input nilai . Ketika menggunakan kondisi while tidak ada iterasi yang terjadi sehingga tidak akan Menampilkan output apapun. Sebaliknya jika menggunakan kondisi do while maka perulangan akan terjadi minimal Satu kali meskipun kondisi awal tidak terpenuhi.
- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?
Jawaban : alasan pengambilan keputusan untuk kasus pemograman ini adalah berdasarkan pada analisi logika , struktur control dan hasil yang diharapkan dari setiap perubahan kode sesuai yang diinginkan selama menjalankan pemograman.

Refleksi

Melalui materi for dan while ini , saya sedikit paham tentang perulangan seperti kegunaan break dan continue pada pemograman. Saya mengetahui bahwa perulangan itu adalah struktur dasar dalam pemograman yang memungkinkan kita untuk menjalankan blok kode secara berulang. Saya paham bahwa perulangan for lebih cocok digunakan pada saat jumlah iterasi sudah diketahui sebelumnya. Dan perulangan while lebih cocok digunakan ketika jumlah iterasi tidak diketahui. Melalui latihan soal yang diberikan saya dapat memahami dan mengetahui cara penggunaan jenis perulangan diterapkan dalam kode, yang memberikan saya pemahaman baru tentang penggunaan break ,continue , modulus , while dan do while. Tetapi saya masih tidak paham dan sulit dalam membuat flowchart. Untuk lebih mengasah kemampuan saya maka harus lebih banyak belajar tentang pemograman dan mencari sumber sumber belajar yang cocok.

