

Nama & NPM	Topik:	Tanggal:
Dewangga Nugroho Anwar G1F024045	FOR dan WHILE Java	9 oktober 2024

Latihan 1

[No. 1.1] Identifikasi Masalah:

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

Jawaban:

Tujuan penulisan kata kunci continue dalam program di bawah adalah untuk melanjutkan ke iterasi berikutnya jika nilai y adalah bilangan genap. Dalam hal ini, kondisi $\text{if } (y \% 2 == 0)$ memeriksa apakah y genap (yaitu, jika sisa pembagian y dengan 2 sama dengan 0). Jika kondisi ini terpenuhi, perintah continue akan dieksekusi, menyebabkan program melompat ke awal loop dan melewati sisa kode dalam loop untuk nilai tersebut. Di awal perulangan, saat y bernilai 0, kondisi $y \leq 15$ sudah terpenuhi (0 kurang dari atau sama dengan 15). Program kemudian memeriksa apakah y adalah bilangan genap. Jika ya, program akan melanjutkan ke iterasi berikutnya tanpa mencetak nilai y . Jika y adalah bilangan ganjil, program akan mengecek kondisi berikutnya, yaitu $\text{else if } (y == 9)$. Jika y sama dengan 9, perintah break akan dijalankan, yang menghentikan seluruh perulangan. Dalam kondisi ini, program tidak akan mencetak angka 9. Jika y adalah bilangan ganjil dan tidak sama dengan 9, maka program akan mencetak nilai y . Dengan demikian, output dari program ini akan mencetak semua bilangan ganjil dari 1 hingga 7, kecuali angka 9 karena program melakukan break saat pemeriksaan sampai pada angka ganjil 9, yang menyebabkan outputnya menjadi 1, 3, 5, 7.

[No.1.1] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan `System.out.println` di bawah kondisi $\text{if } (y \% 2 == 0)$ jika output adalah genap ataupun ganjil akan mengeluarkan hasil dari kedua nya sampai dengan fungsi break di jalankan.
- 2) Alasan solusi ini karena dapat memudahkan user untuk melihat bilangan ganjil ataupun genap.

[No.1.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
berikut adalah algoritma dari kode program 1.1
 - (a) Mulai
 - (b) Deklarasi class utama berupa ContohFor
 - (c) Deklarasi fungsi for hitung nilai y
 - (d) Deklarasi fungsi if hitung modulus %
 - (e) Deklarasi else if operator perbandingan $==$
 - (f) Deklarasi pernyataan break
 - (g) Deklarasi perintah output

2) Kode program dan luaran

a) Screenshot/ Capture potongan kode dan hasil luaran.

```
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
```

Kode input program sebelum melakukan kreasi.

```
1 public class ContohFor {
2     public static void main(String[] args) {
3         for (int y = 0; y <= 15; ++y) {
4             if (y % 2 == 0) { // Memeriksa apakah bilangan genap
5                 System.out.println(y + " adalah genap."); // Menampilkan bahwa bilangan genap
6                 continue; // Melanjutkan ke iterasi berikutnya
7             } else if (y == 9) { // Memeriksa jika y sama dengan 9
8                 break; // Menghentikan perulangan jika y adalah 9
9             } else {
10                System.out.println(y + " adalah ganjil."); // Menampilkan bahwa bilangan ganjil
11            }
12        }
13    }
14 }
```

Kode input program setelah melakukan kreasi.

```
1
3
5
7
|
```

Output program sebelum melakukan kreasi.

```
0 adalah genap.
1 adalah ganjil.
2 adalah genap.
3 adalah ganjil.
4 adalah genap.
5 adalah ganjil.
6 adalah genap.
7 adalah ganjil.
8 adalah genap.
|
```

Output program setelah melakukan kreasi.

b) Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun. Namun saya sedikit memasukan gambar dari program soal yang sudah saya kreasikan untuk menunjukkan bahwa ada metode yang lebih memudahkan user untuk mengetahui mana yang bilangan ganjil dan mana yang bilangan genap.

[No.1.1] Kesimpulan

1) Kreasi

a) Pada program 1.1 saya sedikit memberikan kreasi yaitu penambahan `System.out.println(y + " adalah genap.");` pada line5 di bawah kondisi if, alasan nya agar memudahkan user dalam mengetahui mana bilangan genap dan mana bilangan ganjil.

2) Analisa

a) Dampak yang terjadi dengan perubahan angka pada fungsi for menjadi `for (int y = 0; y <= 15; ++y)`, fungsi if menjadi `if (y % 2 == 1)`, dan `else if (y == 8)` akan membuat output menjadi bilangan ganjil karena pada fungsi else if (`y % 2 == 0`) memeriksa apakah y adalah bilangan genap jika genap program akan melanjutkan perulangan tanpa mencetak y, namun jika ganjil program akan mencetak y dan melakukan perulangan sampai menyentuh angka 9 program akan melakukan break dan output yang di hasilkan yaitu 1,3,5,7

[No. 1.2] Identifikasi Masalah:

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

- a. continue pertama; lalu running, periksa hasilnya
- b. break pertama; lalu running, periksa hasilnya
- c. continue kedua; lalu running, periksa hasilnya

Jawaban:

a) Hasil dari running perubahan pertama //ubah1 menjadi continue pertama adalah, continue pertama aka menghentikan iterasi loop pertama dan langsung melanjutkan ke iterasi berikutnya dari loop tersebut. Pada saat variable waktu `i == 2`, program akan mengabaikan sisa iterasi dari loop kedua, output yang di keluarkan sebagai berikut:

```
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
```

b) Hasil dari running perubahan kode //ubah1 menjadi break pertama adalah. Perintah break pertama akan menghentikan fungsi loop for karena terdapat kode `if (i == 2)`, yang berarti jika looping iterasi pertama `i == 2` maka program akan melakukan break dan tidak akan melanjutkan looping walaupun nilai pada i masih dapat memenuhi proses loop.

```
i = 1; j = 1
i = 1; j = 2
i = 2; j = 1
```

c) Hasil dari running perubahan kode //ubah 1 menjadi continue kedua adalah fungsi for melakukan loop dengan mengabaikan statement `if (i == 2)`, dan melakukan looping sampai batas terpenuhi nya iterasi pertama jika `i < 5`.

```

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

```

[No.1.2] Analisis dan Argumentasi

- 1) Program tersebut menggunakan fungsi for, yang merupakan perulangan atau looping yang me-looping iterasi pertama i dan kedua j.

[No.1.2] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Mulai
 - (b) Inisialisasi loop pertama dengan variable i
 - Loop pada iterasi pertama berjalan selama nilai $i < 5$ “selama nilai i kurang dari 5”
 - (c) Untuk setiap nilai i:
 - Loop kedua dimulai dengan variabel j yang diinisialisasi dengan nilai 1
 - Loop kedua berjalan selama $j < 3$
 - (d) Cetak nilai i dan j dalam format “i = [nilai i] = [nilai j]”
 - (e) Setelah loop kedua selesai:
 - Periksa apakah nilai i sama dengan 2
 - Jika $i == 2$ hentikan loop pertama dengan menggunakan perintah break pada label pertama;
 - (f) Jika kondisi $i == 2$ tidak terpenuhi lanjutkan ke berikutnya dari loop pertama
 - (g) Loop pertama akan berakhir secara normal setelah nilai i mencapai 5 atau dihentikan lebih awal jika $i == 2$
 - (h) Akhir program
- 2) Kode program dan luaran.
 - a) Kode program hasil luaran.

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; //ubah1 11 } } } } </pre>	<pre> java -cp /tmp/w0FgWtqBvy/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 === </pre>

Kode program 1.1 continues pertama.

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 break pertama; //ubah1 11 } } } </pre>	<pre> java -cp /tmp/B5y8HVItxu/ForBersarang i = 1; j = 1 i = 1; j = 2 i = 2; j = 1 === Code Execution Successful === </pre>

Kode program 1.2 break pertama

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 kedua; //ubah1 11 } } } </pre>	<pre> java -cp /tmp/0hqqabQGeI/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 === </pre>

Kode program 1.3 continue kedua

- b) Luran yang di hasilkan sudah benar dengan soal yang diminta.

Pada gambar 1.1 mengubah kode program yang sebelum nya *break kedua* menjadi continues pertama, yang menyebabkan continue pertama aka menghentikan iterasi loop pertama dan langsung melanjutkan ke iterasi berikutnya dari loop tersebut. Pada saat variable waktu $i == 2$, program akan mengabaikan sisa iterasi dari loop kedua. Untuk gambar 1.2 . Perintah break pertama akan menghentikan fungsi loop for karena terdapat kode $if (i == 2)$, yang berarti jika looping iterasi pertama $i == 2$ maka program akan melakukan break dan tidak akan melanjutkan looping walaupun nilai pada i masih dapat memenuhi proses loop. Begitupun pada gambar 1.3 fungsi for melakukan loop dengan mengabaikan statement $if (i == 2)$, dan melakukan looping sampai batas terpenuhi nya iterasi pertama jika $i < 5$.

[No 1.2] Kesimpulan

1) Analisa

Kode program for bersarang tersebut menggunakan 2 fungsi loop bersarang untuk mencetak nilai i dan j . loop pertama (i) berjalan dari 1 hingga kurang dari 5, dan loop kedua (j) daru 1 hingga kurang daru 3. Ketika i mencapai 2, program menghentikan sleuruh loop pertama dengan perintah break.

[No 1 1.3] Identifikasi masalah

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

Luaran:

Masukan Input: 7

```
*****
*****
*****
****
***
**
*
```

Jawaban:

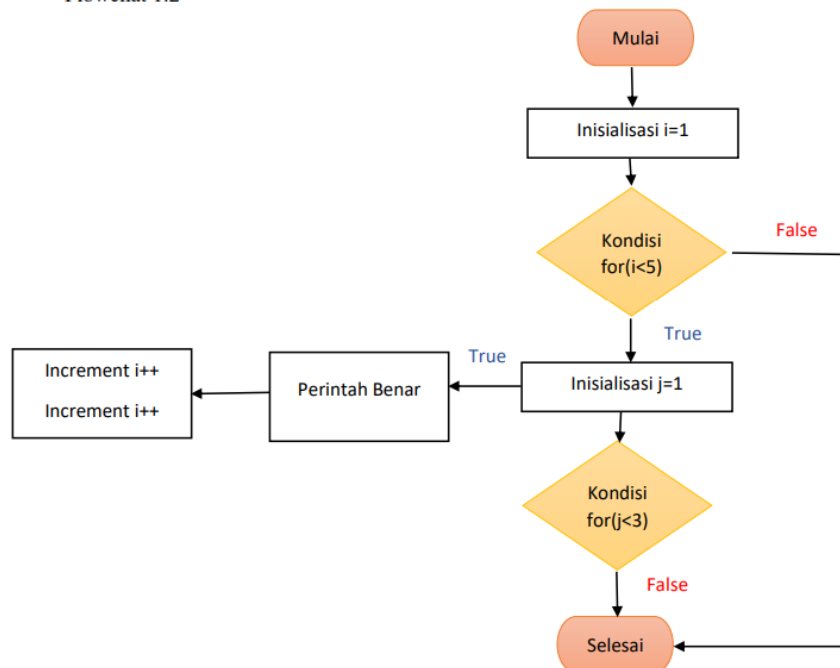
Rekomendasi kode program untuk membalik hasil output di atas adalah:

Hanya mengubah pada kode line 12 : `for(int s=1; s<=t; s++)` yang sebelum nya post-decrement menjadi post-decrement. Variable `s = tinggi` menjadi `s = 1` dan `s>=t` menjadi `s<=t` “menjadi s kurang dari sama dengan t”.

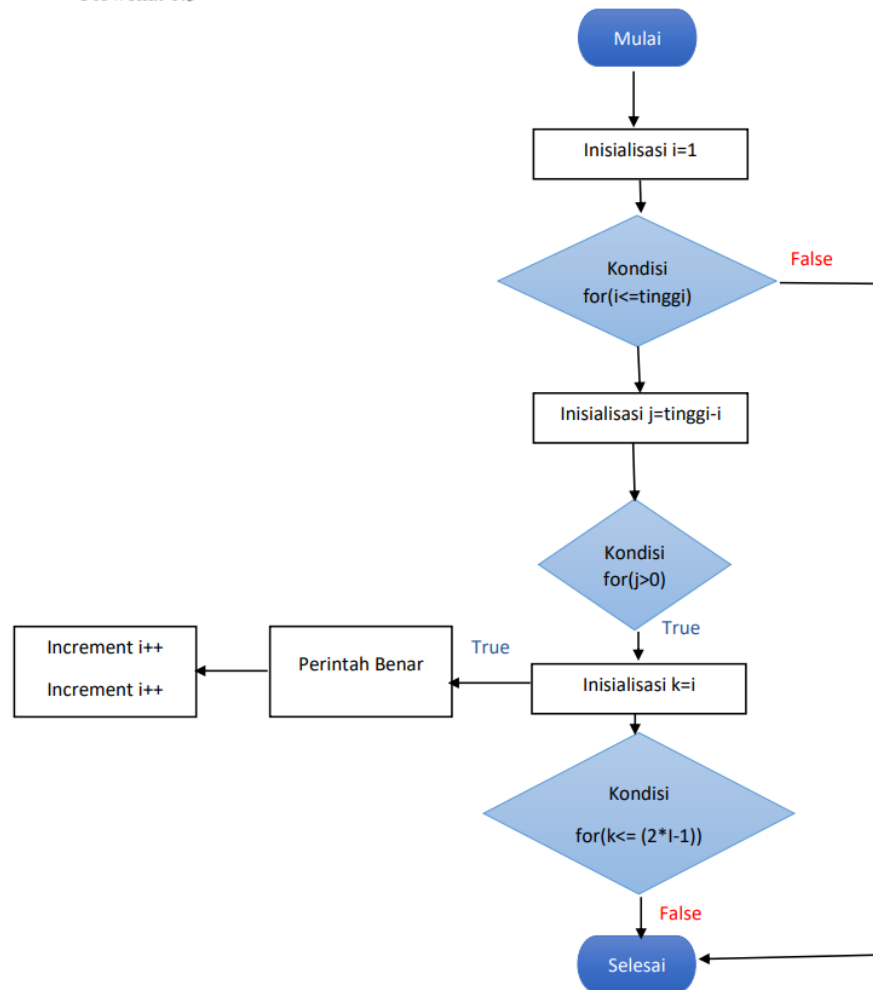
[No 1.2 & 3] Diagram Flowchart:

1.4. Analisa diagram flowchart dari Latihan 1.2 dan 1.3!

Flowchat 1.2



Flowchat 1.3



[No. 1.3] Analisis dan Argumentasi:

- 1) Saya mengusulkan permasalahan ini dengan melihat pada perintah soal untuk membalikan urutan Bintang pada program dengan cara, pada line 12 *for(int s=tinggi; s>=t; s--)* di ubah pada variable interger s=tinggi menjadi s=1 e=mengatur nilai awal s ke 1, artinya loop akan mulai dari s yang bernilai 1 pada iterasi pertama.
- 2) Perbaiki kode program di lakukan dengan cara mengganti loop kedua yang sebelumnya *for(int s=tinggi; s>=t; s--)* menjadi *for(int s=1; s<=t; s++)*. Menjadi kebalikan dari versi kode awal, variable s akan mulai dari 1 dan bertambah hingga mencapai nilai t.

[No 1.3] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Mulai
 - (b) Buat objek iport scanner
 - (c) Kode masukan input oleh user
 - (d) Membaca input dari pengguna
 - (e) Lakukan perulangan i dari 1 hingga tinggi untuk setiap baris piramida
 - (f) Ulangi langkah 5 hingga semua baris piramida selesai dicetak
 - (g) Program selesai

- 2) Kode program dan luaran
 - a) Input program dan output

Main.java	Output
<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 bintang per baris 12 for(int s=1; s<=t; s++){ 13 System.out.print("*"); 14 } 15 System.out.println(); // Membuat Baris Baru 16 } 17 } 18 } </pre>	<pre> java -cp /tmp/aEYYvDFxzZ/ForBersarang Masukan Input: 7 ***** ***** ***** **** *** ** * === Code Execution Successful === </pre>

- b) Luaran yang di hasilkan sudah sesuai dengan yang di minta. Dengan input nilai 7 dan bentuk output yang di hasilkan hanya terbalik urutan nya dari output kode sebelumnya.

[No 1.3] Kesimpulan

- 1) Analisa
Kode program ini berhasil mencetak output yang diinginkan dengan merubah sedikit kode program. Program menerima input berupa nilai tinggi piramida dari pengguna, lalu menghasilkan piramida sesuai dengan tinggi yang di input oleh user.

Latihan 2

[No.2.1] Identifikasi Masalah:

- 2.1) Buat perubahan nilai angka pada variable di contoh 4
//Ubah 1 menjadi continue; lalu running, periksa hasilnya
Analisa dampaknya perubahan terhadap luaran setelah running dan uraikan kegunaan break dan continue!

Jawaban:

Jika kita mengganti statement break menjadi continue, maka saat perulangan sudah mencapai nilai *if(i==4)* akan di abaikan dan statement *continue* akan melanjutkan perulangan i sampai ke akhir nilai yaitu 6.

[No 2.1] Analisis dan Argumentasi

- 1) Saya mengajukan permasalahan ini dapat diatasi dengan menambahkan kurung kurawal penutup } pada akhir program karena menurut Analisa saya pada soal masih kurang penutup kurung kurawal. Sebenarnya untuk perubahan statement tergantung pada kebutuhan atau keinginan user, jika user ingin nilai variable output yang di hasilkan kurang dari 4 *i == 4* maka kita tidak perlu mengubah statement break. Namun jika output yang kita inginkan merupakan angka 1, 2, 3, 4, 5, 6,

maka ada dua cara, yang pertama kita mengubah statement break menjadi continue, cara yang kedua menghapus total kode:

```
if(i==4){
    break;
}
```

- 2) Perbaiki kode dengan menambahkan kurung kurawal penutup pada akhir program }.

[No 2.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Mulai
 - (b) Inisialisasi i
 - (c) Selama nilai i <= 6 maka:
 - Cetak nilai i
 - Tambahkan nilai i sebesar 1
 - Jika nilai i adalah kelipatan 4, lewati iterasi dan gunakan continue
 - (d) Ulangi perulangan sampai nilai i == 6.
 - (e) Selesai
- 2) Kode program dan luaran
 - a) Kode program



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

```
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    }
```

The output window on the right shows the following output:

```
java -cp /tmp/1x730tUVXu/ContohWhile
1
2
3
4
5
6
=== Code Execution Successful ===
```

- b) Luaran yang dihasilkan sudah benar dan sesuai dengan soal yang diminta. Dengan output 1, 2, 3, 4, 5, 6 dari hasil perintah kode program tersebut.

[No 2.1] Kesimpulan

- 1) Analisa

Kode program ini menggunakan while loop untuk mencetak angka dari 1 hingga 6, namun jika proses perulangan i sampai pada nilai 4 maka program akan menghentikan proses looping dan output yang akan di keluarkan adalah 1, 2, 3. Namun jika kita menggunakan statement continue maka program akan mengabaikan kode if dan melanjutkan perulangan sampai semua nilai i terpenuhi atau habis.

[No 2.2] Identifikasi Masalah:

- 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!

Jawaban:

- a. . Luaran Saat Jumlah Diulang = 0 dengan while: Ketika program dijalankan dengan menggunakan while loop dan jumlah pengulangan yang dimasukkan adalah 0, maka while loop tidak akan dieksekusi sama sekali karena kondisi awal $i < \text{jumlah}$ (di mana jumlah adalah 0) tidak pernah benar. Oleh karena itu, program tidak akan mencetak apa pun ke layar. Loop tidak dijalankan, sehingga tidak ada output yang dihasilkan.

[No 2.2] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara mengubah pengulangan dari while menjadi do...while. Dengan cara ini, program akan selalu mencetak kata setidaknya satu kali, terlepas dari nilai yang dimasukkan untuk jumlah pengulangan.
- 2) Alasan solusi ini karena penggunaan do...while memberikan fleksibilitas lebih dalam situasi di mana pengguna mungkin ingin mengulang kata meskipun jumlah pengulangan yang dimasukkan adalah nol. Dengan menggunakan do...while, kita dapat memastikan bahwa kata tetap dicetak minimal satu kali, sehingga memberikan pengalaman pengguna yang lebih baik

[No 2.2] Penyusunan Algoritma dan Kode Program

1) Algoritma dengan while

- (a) Mulai
- (b) Inisialisasi class utama
- (c) Inisialisasi public class
- (d) Inisialisasi nilai yang di hitung
- (e) Inisialisasi menghitung nilai < 20
- (f) Inisialisasi fungsi if
- (g) Inisialisasi cetak output
- (h) Selesai

2) Kode program

- a) Kode program dan luaran

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 % 3 == 0) //ubah2 6 System.out.println(count); 7 count++; 8 } 9 } 10 }</pre>	<pre>java -cp /tmp/38aKdqf6J6/WhileBersarang 0 3 6 9 12 15 18 === Code Execution Successful ===</pre>

- b) Luaran yang dihasilkan sudah benar dan sesuai dengan soal yang diminta.

[No 2.2] Kesimpulan

1) Analisa

Kdoo program tersebut menggunakan while sebagai perulangan, sebenarnya kita dapa menggunakan do while agar lebih efisien dan user dapat melihat output sesuai dengan berapa jumlah nilai yang ingin di keluarkan dengan cara user menginputkan nilai secara manual.

[No 2.3 & 4] Identifikasi Masalah:

2.3 Buat perubahan nilai angka pada variable.

a) program dan output yang menggunakan while.

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 % 3 == 0) //ubah2 6 System.out.println(count); 7 count++; 8 } 9 } 10 }</pre>	<pre>java -cp /tmp/38aKdqf6J6/WhileBersarang 0 3 6 9 12 15 18 === Code Execution Successful ===</pre>

b) program dan output yang menggunakan do..while.

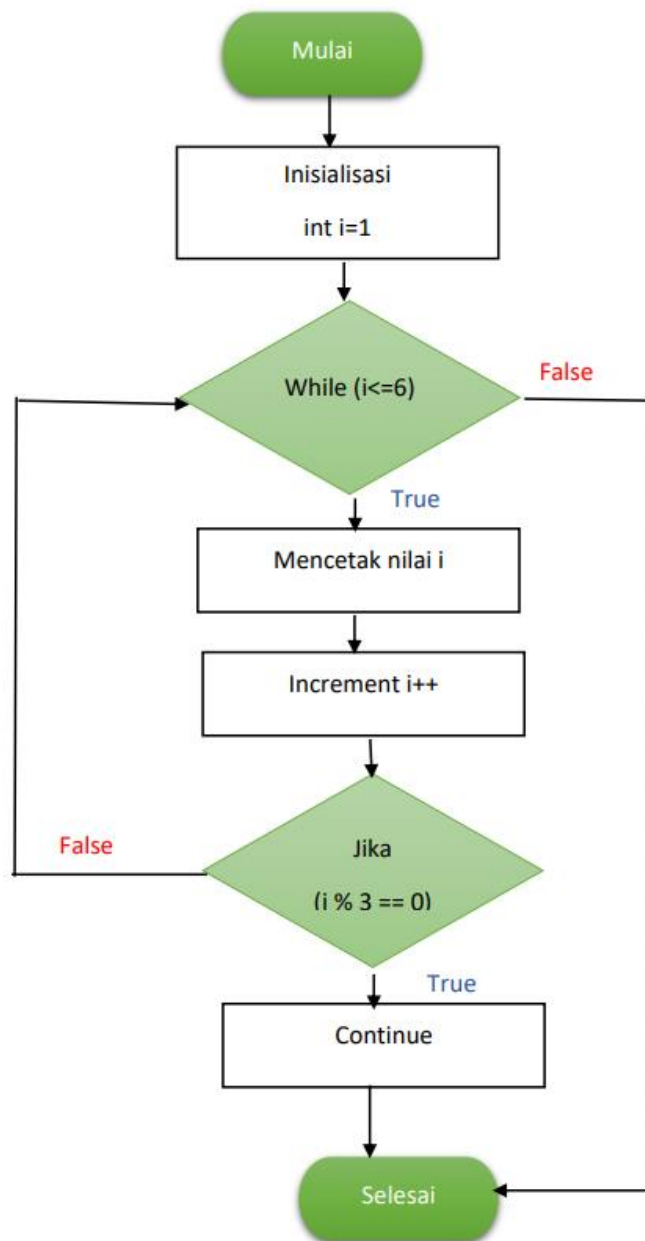
Main.java	Output
<pre>1 import java.util.Scanner; 2 public class forBersarang { 3 4 public static void main(String[] args){ 5 Scanner data = new Scanner(System.in); 6 System.out.print("masukan kata yang ingin di ulang: "); 7 String kata = data.nextLine(); 8 9 System.out.print("masukan jumlah yang ingin di ulang: "); 10 int jumlah = data.nextInt(); 11 12 int i = 0; 13 do { 14 System.out.println(kata); 15 i++; 16 } while(i < jumlah); 17 } 18 }</pre>	<pre>java -cp /tmp/71XnTobz1M/forBersarang masukan kata yang ingin di ulang: dewangga masukan jumlah yang ingin di ulang: 2 dewangga dewangga === Code Execution Successful ===</pre>

c) penjelasan.

Luaran yang dihasilkan sudah benar dan sesuai dengan soal yang diminta. Ketika program menggunakan while dengan jumlah pengulangan yang dimasukkan adalah 0, program tidak akan mencetak apa pun. Hal ini disebabkan oleh fakta bahwa kondisi awal loop $i < \text{jumlah}$ tidak terpenuhi (karena $0 < 0$ adalah salah). Oleh karena itu, blok kode di dalam loop while tidak akan dieksekusi sama sekali. Sebagai hasilnya, tidak ada output yang dihasilkan, dan program akan langsung selesai setelah meminta input dari pengguna. Sebaliknya, ketika menggunakan do while, program akan mencetak kata setidaknya satu kali meskipun jumlah pengulangan yang dimasukkan adalah 0. Ini karena blok kode di dalam do dieksekusi terlebih dahulu sebelum kondisi diperiksa. Setelah kata dicetak, nilai i ditambahkan 1. Kemudian, kondisi $i < \text{jumlah}$ diperiksa (di mana jumlah adalah 0), dan loop akan berhenti. Oleh karena itu, dengan jumlah = 0, output program akan mencetak kata satu kali.

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

Flowchat 2.1



Flowchat 2.3

