

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Cut Fitriani G1F024079</b>	<b>UNIT 1 FOR</b>	<b>3 oktober 2024</b>

**[Nomor Soal] Identifikasi Masalah:**

- 1) Uraikan permasalahan dan variabel

```
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!

Jawab:

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

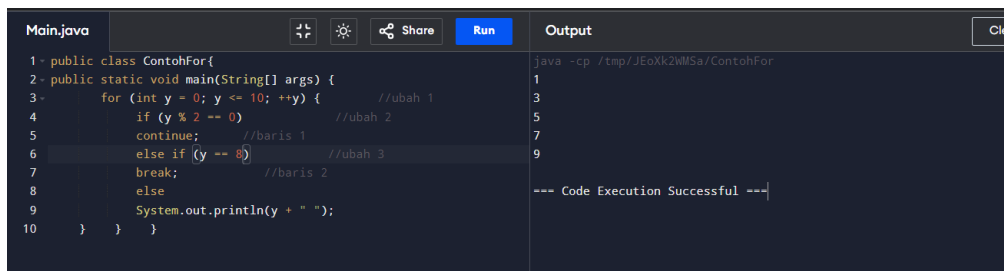
- continue: Untuk mengontrol alur loop atau perulangan dengan mengubah kode pada contoh 1 dan menghasilkan luaran bilangan ganjil maka continue bertujuan untuk mencetak bilangan ganjil dan bilangan genap akan di lewati dan tidak tercetak.
- break: Membatasi perulangan hingga nilai tertentu (9) sehingga hanya bilangan ganjil dari 1-7 yang akan dicetak.

Secara keseluruhan, kode ini akan mencetak bilangan genap dari 0 hingga 6.

- //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) Pada //Ubah 1 menjadi `for (int y = 0; y <= 15; y++)` , mengubah nilai yang dicetak adalah kondisi perulangan dimana jumlah perulangan yang dilakukan akan berubah dan terjadi perubahan urutan di output.
- 2) Pada //Ubah 2 menjadi `if (y % 2 == 0)`, kondisi ini akan memeriksa apakah nilai y habis dibagi 2 (bilangan genap) jika habis dibagi maka outputnya akan berisi bilangan genap.
- 3) Pada //Ubah 3 menjadi `else if (y == 9)`, perulangan akan berhenti ketika nilai y mencapai 9 dan output akan berisi angka-angka dari 0 hingga 10.

## 1.2 a. continue pertama; lalu running, periksa hasilnya

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

## b. break pertama; lalu running, periksa hasilnya

Main.java	Output
<pre>1- public class ForBersarang { 2-     public static void main(String[] args) { 3-         pertama: 4-             for( int i = 1; i &lt; 5; i++) { 5- 6-                 kedua: 7-                     for(int j = 1; j &lt; 3; j++) { 8-                         System.out.println("i = " + i + "; j = " + j); 9-                         if ( i == 2) 10-                             break pertama; //ubah1 11-                     } 12-             } 13-     } 14- }</pre>	<pre>java -cp /tmp/rdkNsYyMCY/ForBersarang i = 1; j = 1 i = 1; j = 2 i = 2; j = 1 === Code Execution Successful ===</pre>

## c. continue kedua; lalu running, periksa hasilnya

Main.java	Output
<pre>1- public class ForBersarang { 2-     public static void main(String[] args) { 3-         pertama: 4-             for( int i = 1; i &lt; 5; i++) { 5- 6-                 kedua: 7-                     for(int j = 1; j &lt; 3; j++) { 8-                         System.out.println("i = " + i + "; j = " + j); 9-                         if ( i == 2) 10-                             continue kedua; //ubah1 11-                     } 12-             } 13-     } 14- }</pre>	<pre>java -cp /tmp/n7R1PbvPy/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>

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 melompat ke perulangan berikutnya dari perulangan yang berlabel pertama, sisa perulangan pada perulangan kedua yang sedang berjalan akan diabaikan. Pada break pertama perulangan berhenti sebelum selesai, kombinasi nilai i dan j sebelum kondisi break terpenuhi yang akan dicetak lalu perulangan pertama akan dihentikan sepenuhnya. Sedangkan pada continue kedua program akan langsung ke perulangan berikutnya dari loop kedua.

## 1.3 Susunlah analisa kode untuk menghasilkan luaran tersebut!

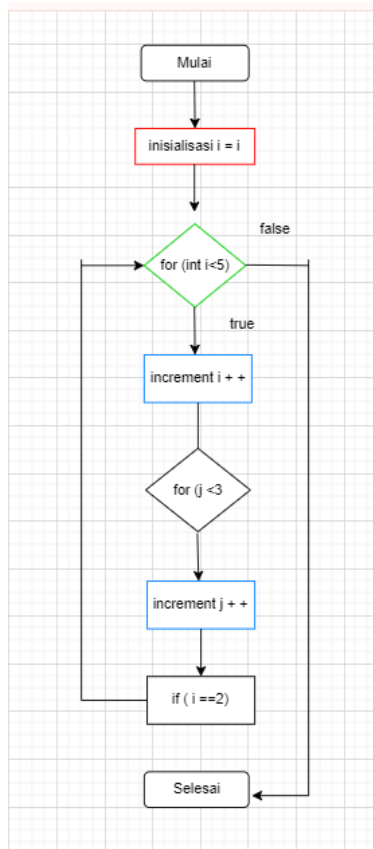
Main.java	Output
<pre>1- import java.util.Scanner; 2- 3- public class ForBersarang { 4-     public static void main(String[] args) { 5-         // Instance Input Scanner 6-         Scanner input = new Scanner(System.in); 7-         System.out.print("Masukan Input: "); 8-         int tinggi = input.nextInt(); // Mendapatkan Input Dari user 9- 10-        for (int t = tinggi; t &gt;= 1; t--) { 11-            // Menghitung jumlah tinggi piramida 12-            for (int s = 1; s &lt;= t; s++) { 13-                System.out.print(" "); 14-            } 15-            System.out.println(); // Membuat Baris Baru 16-        } 17- 18-        input.close(); // Menutup Scanner 19-    } 20- }</pre>	<pre>java -cp /tmp/nmfCoJnda/ForBersarang Masukan Input: 7 ***** ***** **** *** ** * === Code Execution Successful ===</pre>

Jawab: Dengan menginput variabel tinggi piramida, melakukan perulangan pertama `for (int t = tinggi; t >= 1; t--)` akan berulang dari tinggi yang dimasukkan sampai 1, yang berarti 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 iterasi. Pada perulangan pertama, `t` adalah tinggi yang dimasukkan, dan setiap perulangan berikutnya jumlah bintang berkurang. Setelah perulangan kedua, `System.out.println();` digunakan untuk mencetak baris baru setelah semua bintang dicetak.

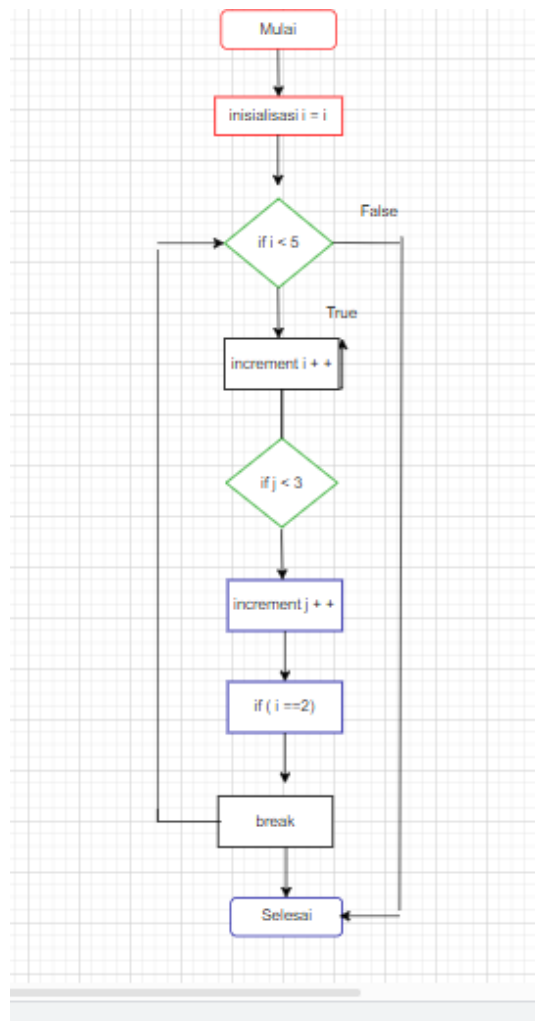
#### 1.4 Analisa diagram flowchart dari Latihan 1.2 dan 1.3!

##### Flowchart 1.2

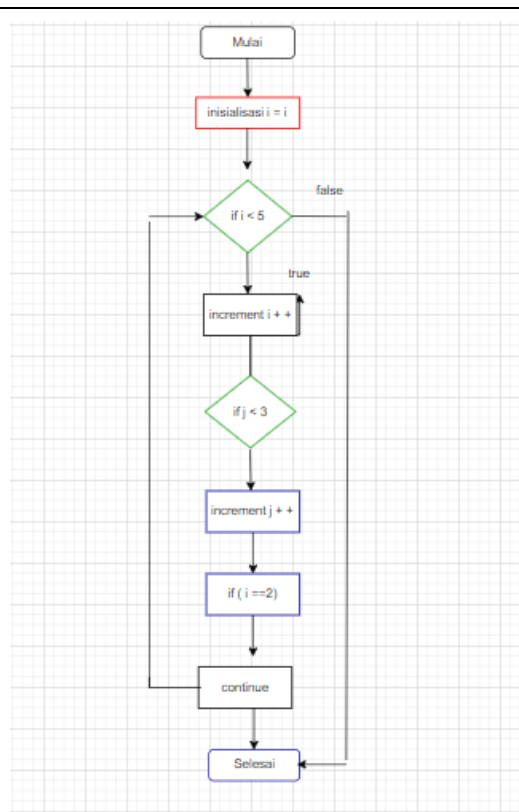
###### a. Continue pertama



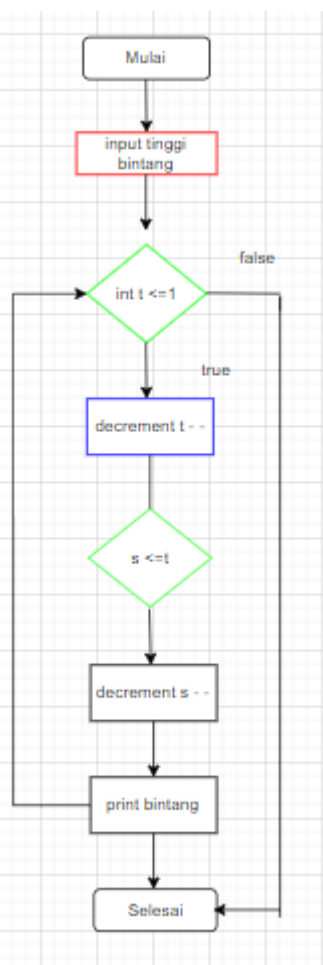
b. break pertama



C. continue kedua



Flowchart 1.3



- 2) Rincikan sumber informasi yang relevan (buku / webpage)  
Jawab: Informasi yang relevan ada pada youtube channel Rumah Ilmu Raflesia.
- 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.  
Jawab: Mengubah kode sesuai dengan kondisi tertentu dan akan mengeluarkan output yang diinginkan pada soal-soal tertentu.
- 2) Analisis solusi, kaitkan dengan permasalahan.  
Jawab: Agar output sesuai dengan yang diinginkan, kode pemrograman harus sesuai dengan kondisi-kondisi tertentu.

#### **[Nomor 3] Penyusunan Algoritma dan Kode Program**

- 1) Rancang desain solusi atau algoritma  
Algoritma 1.1:
  - a) Mulai program
  - b) Buat variabel y dan inisialisasi dengan 0
  - c) Buat sebuah loop for
  - d) Jika y dibagi 2 menghasilkan sisa 1 ( $y \% 2 == 1$ ), jika kondisi ini benar, gunakan continue untuk melanjutkan ke iterasi berikutnya tanpa mengeksekusi sisa blok dalam loop.
  - e) Jika y sama dengan 8 ( $y == 8$ ), jika kondisi ini benar, gunakan break untuk keluar dari loop dan tidak melanjutkan iterasi selanjutnya.
  - f) Lakukan perulangan hingga nilai y melebihi 10 atau loop dihentikan dengan break.
  - g) selesai.

Algoritma 1.2:

- a) Mulai program
- b) Menentukan nilai loop pertama
- c) Menentukan nilai loop kedua
- d) Mencetak nilai hasil dari loop pertama dan kedua
- e) Melakukan perulangan hingga semua iterasi selesai
- f) Selesai

Algoritma 1.3:

- a) Mulai program
  - b) Memasukkan nilai untuk menjalankan program
  - c) Melakukan perulangan atau loop yang berjalan
  - d) Mencetak bintang untuk menghasilkan piramida
  - e) Pindah ke baris baru setelah mencetak Bintang
  - f) Melakukan perulangan hingga seluruh baris piramida tercetak dari atas ke bawah
  - g) Setelah perulangan selesai tutup objek scanner untuk menghentikan input
  - h) selesai
- 2) Tuliskan kode program dan luaran
  - a) Beri komentar pada kode
  - b) Uraikan luaran yang dihasilkan
  - 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 &lt;= 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 ForBersarang { 2-     public static void main(String[] args) { 3-         pertama: 4-             for( int i = 1; i &lt; 5; i++) { 5- 6-                 kedua: 7-                     for(int j = 1; j &lt; 3; j ++ ) { 8-                         System.out.println("i = " + i + "; j = 9-                             " +j); 10-                        if ( i == 2) 11-                            break kedua; //ubah1 </pre>	<pre> java -cp /tmp/yVsC8Kx2fy/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>

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 9-         //Dari User 10-         for(int t=tinggi; t&gt;=1; t--){ 11-             //Menghitung Jumlah Tinggi Piramida 12-             for(int s=tinggi; s&gt;=t; s--){ 13-                 //Menghitung Jumlah Spasi per Baris 14-                 System.out.print(" "); 15-             } 16-             System.out.println(); //Membuat Baris Baru 17-         } 18-     } </pre>	<pre> java -cp /tmp/ZtQgEzem8d/ForBersarang Masukan Input: 7 * ** *** **** ***** ***** === Code Execution Successful === </pre>

#### [Nomor 4] Kesimpulan

##### 1) Analisa

- a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!

Jawab:

Pemrograman ini menunjukkan penggunaan struktur kontrol dasar dalam bahasa pemrograman, seperti if-else dan for. Pada 1.1 lebih sederhana dan fokus pada pengkondisian sederhana, pada 1.2 lebih kompleks dengan penggunaan loop, pada 1.3 berhasil mencetak sebuah piramida sesuai dengan input pengguna. Kode ini relatif mudah dipahami karena menggunakan penamaan variabel yang jelas dan struktur kontrol yang sederhana. Hasil output dari ketiga program sesuai dengan logika yang terdapat dalam kode

- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawab: Untuk memenuhi output yang diinginkan dan kode program dapat berjalan, tidak adanya error saat di running.

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Cut Fitriani G1F024079</b>	<b>UNIT 2 WHILE</b>	<b>3 oktober 2024</b>

**[Nomor Soal] Identifikasi Masalah:**

- 1) Uraikan permasalahan dan variabel

```
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	Output
<pre>1 public class ContohWhile{ 2 public static void main(String[] args) { 3     int i=1; 4     while(i&lt;=6){ 5         System.out.println(i); 6         i++; 7         if(i==4){ 8             continue; //ubah1 9         } 10    }</pre>	<pre>java -cp /tmp/9HjVEG20gu/ContohWhile 1 2 3 4 5 6 === Code Execution Successful ===</pre>

Analisa dampaknya perubahan terhadap luaran setelah running dan uraikan kegunaan break dan continue!

Jawab:

Program akan melewati sisa perulangan dari loop saat kondisi tertentu terpenuhi. Dalam hal ini, ketika i menjadi 4, pernyataan continue akan menyebabkan program melewati bagian sisa dari loop dan langsung kembali ke awal loop, di mana nilai i sudah ditingkatkan menjadi 5. Program kemudian melanjutkan dan mencetak 5 dan 6.

- Kegunaan break dan continue:  
Break digunakan untuk keluar dari loop sepenuhnya dan saat ingin menghentikan proses loop lebih awal berdasarkan suatu kondisi.

Continue digunakan untuk melewati iterasi saat ini dan melanjutkan ke iterasi berikutnya dan saat ingin mengabaikan beberapa proses di dalam loop untuk kondisi tertentu, tetapi tetap ingin loop berjalan.

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

Main.java	Output
<pre>1 public class WhileBersarang { 2     public static void main(String[] args) { 3         int count = 0; //ubah1 4         while (count &lt; 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>

Analisa dampaknya perubahan terhadap luaran setelah running dan uraikan kegunaan % untuk angka yang berbeda pada perintah tersebut!

Jawab: Dengan kondisi ini, program mencetak angka yang merupakan kelipatan dari 5, yaitu: 0, 5, 10, dan 15. Keluaran ini mencakup angka dari 0 hingga 19 yang habis dibagi 5.

Kegunaan %, kita dapat dengan mudah memeriksa apakah count adalah kelipatan dari 20. Jika hasilnya 0, itu berarti count dapat dibagi habis oleh 20.

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

Main.java	Output
<pre>1- public class WhileBersarang { 2-     public static void main(String[] args) { 3-         int count = 0; //ubah1 4-         while (count &lt; 0) { 5-             if (count % 3 == 0) //ubah2 6-                 System.out.println(count); 7-             count++; 8-         } 9-     } }</pre>	<pre>java -cp /tmp/tgfv0yHglm/WhileBersarang === Code Execution Successful ===</pre>

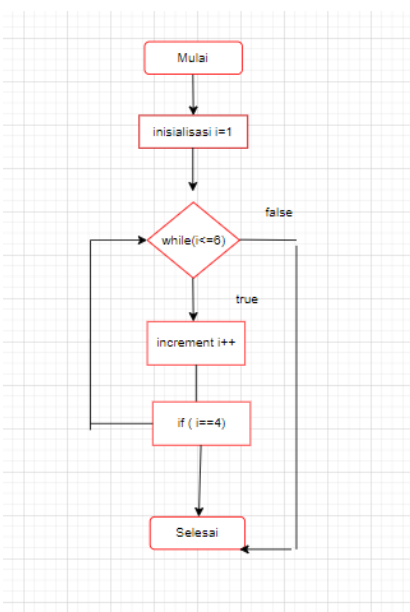
Ubahlah baris kode while pada Contoh 5 menjadi do ... while dengan persyaratan yang sama while (count < 0). Bandingkan hasil luaran antara menggunakan while dan do ... while!

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 &lt; 0); // kondisi ini 9-     } 10- } 11-</pre>	<pre>java -cp /tmp/nm6820GurC/WhileBersarang 0 === Code Execution Successful ===</pre>

- Ketika menggunakan while tidak ada luaran. Program tidak mencetak apa pun dan langsung keluar karena kondisi count < 0 tidak terpenuhi (nilai awal count adalah 0).
- Ketika menggunakan do ... while, blok kode di dalam do akan dijalankan setidaknya sekali sebelum memeriksa kondisi. Nilai awal count adalah 0, dan mencetak hasil luaran 0.

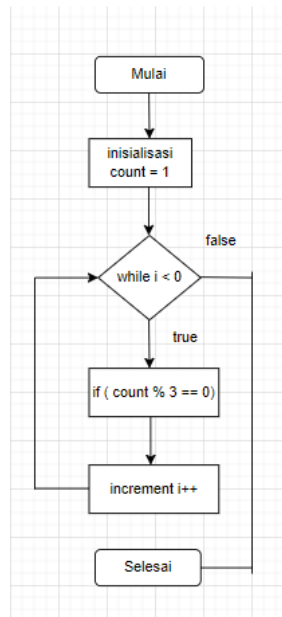
2.4. Analisa diagram flowchart dari Latihan 2.1, Contoh 5, dan Latihan 2.3!

Flowchart 2.1

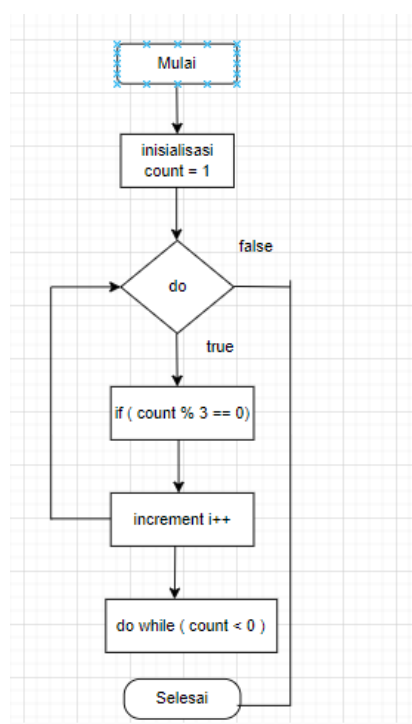


Flowchart 2.3

### While



### Do while



- 2) Rincikan sumber informasi yang relevan (buku / webpage)

Jawab: Informasi yang saya dapatkan ada pada youtube channel Rumah Ilmu Raflesia.

- 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.  
Jawab: Mengubah kode sesuai dengan kondisi tertentu dan akan mengeluarkan output yang diinginkan pada soal-soal tertentu.
- 2) Analisis solusi, kaitkan dengan permasalahan.  
Jawab: Agar output sesuai dengan yang diinginkan, kode pemrograman harus sesuai dengan kondisi-kondisi tertentu.

### [Nomor 3] Penyusunan Algoritma dan Kode Program

- 1) Rancang desain solusi atau algoritma  
Algoritma 2.1:
  - a) Mulai program
  - b) Menginialisasikan variabel
  - c) Membuat loop while
  - d) Mencetak nilai i
  - e) Memeriksa i sama dengan 4
  - f) Lakukan perulangan
  - g) Selesai  
Algoritma 2.2:
  - a) Mulai program
  - b) Menginialisasikan variabel
  - c) Membuat loop while
  - d) Memeriksa apakah count adalah kelipatan 3
  - e) Mencetak count
  - f) Selesai
- 2) Tuliskan kode program dan luaran
  - a) Beri komentar pada kode
  - b) Uraikan luaran yang dihasilkan
  - c) Screenshot/ Capture potongan kode dan hasil luaran

Main.java	Output
<pre>1- public class ContohWhile{ 2- public static void main(String[] args) { 3-     int i=1; 4-     while(i&lt;=6){ 5-         System.out.println(i); 6-         i++; 7-         if(i==4){ 8-             break;        //ubah1 9-         } 10-     } 11- }</pre>	<pre>java -cp /tmp/62GYgUnTag/ContohWhile 1 2 3  --- 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 &lt; 20) { 5-             if (count % 3 == 0) //ubah2 6-                 System.out.println(count); 7-             count++; 8-         } 9-     } 10- }</pre>	<pre>java -cp /tmp/4cfXqUY1L0/WhileBersarang 0 3 6 9 12 15 18  --- Code Execution Successful ---</pre>

#### [Nomor 4] Kesimpulan

1) Analisa

a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!

Jawab: Kedua program ini memiliki tujuan yang serupa, yaitu mencetak angka secara berulang. Dengan menunjukkan penggunaan loop while untuk perulangan. Loop ini bermanfaat ketika jumlah iterasi tidak diketahui sebelumnya. Break dan if, untuk mengontrol eksekusi program dengan lebih efisien dan hanya melakukan tindakan yang diperlukan. Penggunaan kondisi dalam loop (if, modulus) sangat penting untuk memfilter hasil dan mendapatkan output yang diinginkan. Keduanya menunjukkan dasar-dasar pengulangan dan kontrol alur dalam pemrograman, yang penting untuk membangun algoritma yang efektif dan efisien.

b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?

Jawab: Dengan menggunakan break, kita dapat menghentikan eksekusi loop lebih awal tanpa harus menunggu hingga kondisi  $i \leq 6$  tidak terpenuhi. Ini membuat program lebih efisien dan menghindari keluaran yang tidak perlu. Dan program untuk hanya mencetak angka yang relevan (yaitu, kelipatan 3) tanpa mencetak semua angka dari 0 hingga 19. Ini menunjukkan efisiensi dalam pengolahan data dan penggunaan kontrol yang tepat untuk menghasilkan output yang sesuai.

#### Refleksi

Dari pembelajaran ini saya dapat menyimpulkan bahwa dapat menentukan kondisi if bersarang saat dibutuhkan di sesuaikan dengan kondisi tertentu dan dapat membuat flowchart.