

Nama & NPM	Topik:	Tanggal:
Hedy Rafian Firdaus G1F024027	FOR	28 September 2024

**[No. 1] Identifikasi Masalah:**

**1) Uraikan permasalahan dan variabel**

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

**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!

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

[Video Materi 1 tentang FOR – https://www.youtube.com/watch?v=Ij9qLLblxEU](https://www.youtube.com/watch?v=Ij9qLLblxEU)

[No. 1] Analisis dan Argumentasi

1) Uraikan rancangan solusi yang diusulkan.

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!**

- Untuk percobaan pertama, saya mengganti baris //Ubah 1 menjadi **for (int y = 0; y <= 15; y++) {** sesuai dengan instruksi yang diberikan. Kemudian running kodenya.
- Untuk percobaan kedua, saya menggunakan kode sebelumnya dengan mengganti baris //Ubah 2 menjadi **if (y % 2 == 0)** sesuai dengan instruksi yang diberikan. Lalu running kodenya.
- Untuk percobaan ketiga, saya menggunakan kode sebelumnya dengan mengganti baris //Ubah 3 menjadi **else if (y == 9)** sesuai dengan instruksi yang diberikan. Lalu running kodenya.

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)!**

- a) Untuk percobaan pertama, saya mengganti baris //Ubah1 menjadi **continue pertama;** Lalu lakukan running pada kode, dan periksa hasilnya
- b) Untuk percobaan kedua, saya mengganti baris //Ubah1 menjadi **break pertama;** Lalu lakukan running pada kode, dan periksa hasilnya
- c) Untuk percobaan ketiga, saya mengganti baris //Ubah1 menjadi **continue kedua;** Lalu lakukan running pada kode, dan periksa hasilnya

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

Luaran:

Masukan Input: 7

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

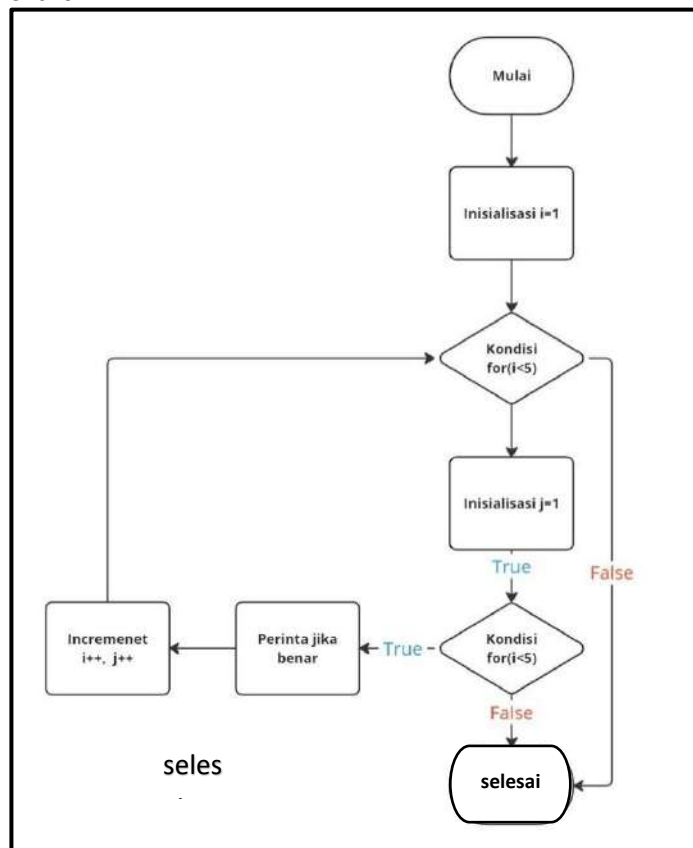
\*

**Susunlah analisa kode untuk menghasilkan luaran tersebut!**

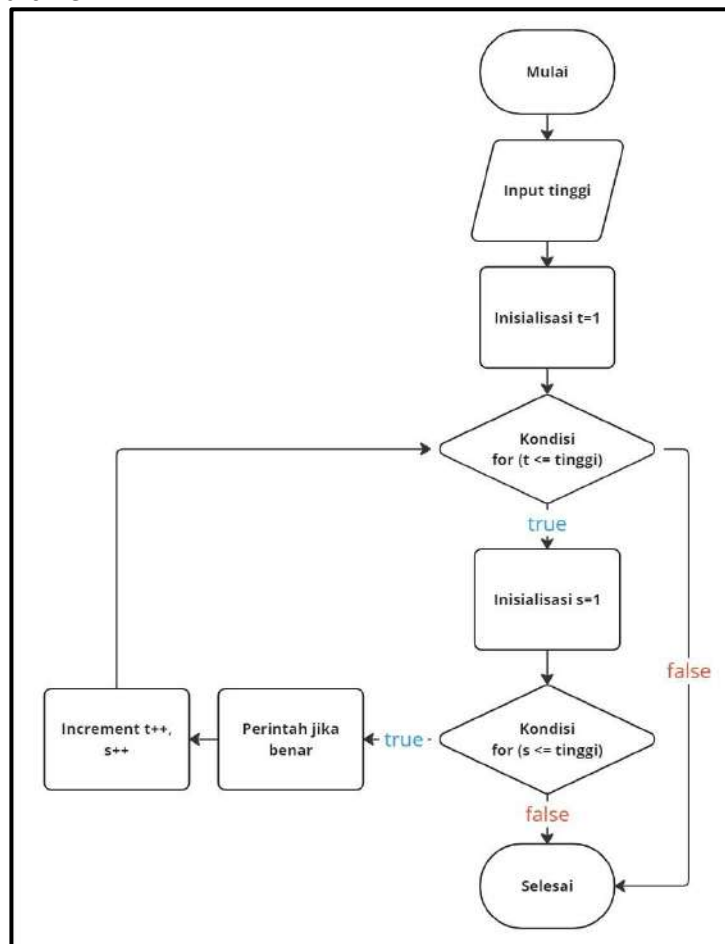
Untuk bagian ini, saya mengganti kode pada bagian Loop untuk menghasilkan bintang pada setiap baris.. Saya mengganti isi kodenya menjadi **for (int s = 1; s <= t; s++) {** sehingga hasilnya akan sesuai dengan yang diminta.

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

- Flowchart 1.2



- Flowchart 1.3



## [No. 1] Penyusunan Algoritma dan Kode Program

### 1) Rancang desain solusi atau algoritma

- Contoh 1
  - a) Mulai
  - b) Membuat nama kelas dan nama file ContohFor
  - c) Deklarasi method utama
  - d) Membuat kode for
  - e) Membuat kode if + continue
  - f) Membuat kode else if dan diakhir dengan break
  - g) Memunculkan hasil luaran
  - h) Selesai
  
- Contoh 2
  - a) Mulai
  - b) Membuat nama kelas dan nama file ForBersarang
  - c) Deklarasi method utama
  - d) Membuat kode for pertama
  - e) Membuat kode for kedua
  - f) Memasukan kode if
  - g) Memasukkan perintah continue/ break
  - h) Memunculkan luaran
  - i) Selesai
  
- Contoh 3
  - a) Mulai
  - b) Membuat nama kelas dan nama file ForBersarang
  - c) Deklarasi method utama
  - d) Membuat program Instance Input Scanner
  - e) Masukkan input scanner
  - f) Membuat kode loop untuk menghasilkan piramida
  - g) Memasukkan kode for
  - h) Memunculkan luaran
  - i) Selesai

### 2) Tuliskan kode program dan luaran

#### a) Contoh 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        }
11    }
```

Contoh 1.1

```
0
2
4
6
```

Luaran 1.1

Luaran yang dihasilkan telah sesuai dengan yang telah diprogram. Ini dibuktikan dengan hasil; yang didapat telah sesuai dengan kode program yang telah dilakukan.

```
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 == 8)                      //ubah 3
7             break;                           //baris 2
8         else
9             System.out.println(y + " ");
10    } } }
```

Contoh 1.2

```
1
3
5
7
9
11
13
15
```

Luaran 1.2

Luaran yang dihasilkan telah sesuai dengan yang telah diprogram. Ini dibuktikan dengan hasil; yang didapat telah sesuai dengan kode program yang telah dilakukan.

```
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    } } }
```

Contoh 1.3

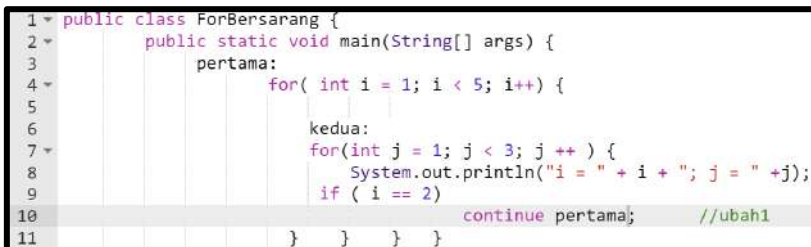


```
1
3
5
7
|
```

Luaran 1.3

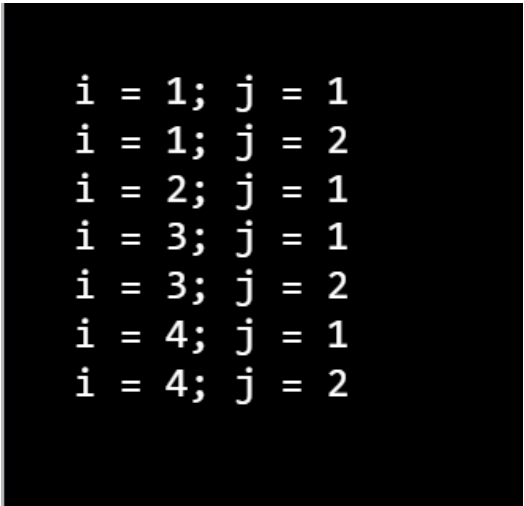
Luaran yang dihasilkan telah sesuai dengan yang telah diprogram. Ini dibuktikan dengan hasil; yang didapat telah sesuai dengan kode program yang telah dilakukan.

b) Contoh 2



```
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 }
```

Contoh 2.1



```
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
```

Luaran 2.1

Luaran yang dihasilkan telah sesuai dengan yang telah diprogram. Ini dibuktikan dengan hasil; yang didapat telah sesuai dengan kode program yang telah dilakukan.

```

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                     } } } }

```

Contoh 2.2

```

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

```

Luaran 2.2

Luaran yang dihasilkan telah sesuai dengan yang telah diprogram. Ini dibuktikan dengan hasi; yang didapat telah sesuai dengan kode program yang telah dilakukan.

```

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                     } } } }

```

Contoh 2.3



```
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
|
```

Luaran 2.3

Luaran yang dihasilkan telah sesuai dengan yang telah diprogram. Ini dibuktikan dengan hasil; yang didapat telah sesuai dengan kode program yang telah dilakukan.

c) Contoh 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 Dari User
9
10        // Loop untuk menghasilkan piramida
11        for (int t = tinggi; t >= 1; t--) {
12            // Loop untuk menghasilkan bintang pada setiap baris
13            for (int s = 1; s <= t; s++) {
14                System.out.print("*");
15            }
16            System.out.println(); // Membuat Baris Baru
17        }
18    }
19 }
```

Contoh 3

```
Masukan Input: 7
*****
*****
*****
****
***
**
*
```

Luaran Contoh 3

Luaran yang dihasilkan telah sesuai dengan yang telah diprogram. Ini dibuktikan dengan hasil; yang didapat telah sesuai dengan kode program yang telah dilakukan.

## [No. 1] Kesimpulan

### Analisa

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

1. Masalah:
  - Masalah utama dalam proses ini adalah memahami penggunaan kata "continue" dan "break" dalam nesting. seperti bagaimana sifat-sifat lingkaran dan transformasi logika mempengaruhi hasil suatu program. Selain itu, perlu dipahami cara mengurutkan keluaran video (seperti pola bintang) menggunakan loop bersarang.
2. Proses:
  - Proses siklus terbalik terstruktur dengan baik menggunakan struktur kontrol seperti "for", "if" dan "else". Istilah "continue" memungkinkan program untuk melewati logika perulangan ketika kondisi tertentu terpenuhi, sedangkan "break" menyebabkan perulangan berhenti sama sekali. Pada model nested loop untuk model bintang, algoritma ini menyesuaikan jumlah bintang pada setiap baris dengan mengurangi jumlah iterasi pada setiap loop.
3. Kode Program:
  - Pada contoh pertama, perubahan batas loop dan logika kontrol ("if", "else if") menunjukkan pengaruh kekuatan struktur terhadap aliran retakan . Dengan mengubah kondisi "y % 2 == 0" program hanya akan memproses bilangan genap, dan menambahkan kondisi "if (y == 9)" akan memaksa loop mencapai nilai "y". » Berhenti sebelumnya. 9.
  - Pada contoh kedua, penerapan kata kunci "lanjutkan" dan "break" lebih memahami kontrol aliran program. "lanjutkan" akan melewati instruksi yang tersisa dalam perulangan, sedangkan "break" akan menghentikan seluruh perulangan.
  - Pada contoh ketiga, penggunaan loop bersarang dapat menghasilkan contoh yang diinginkan. Atur loop luar untuk mengontrol jumlah baris dan loop dalam untuk mencetak bintang sesuai logika yang diinginkan.

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

1. Menggunakan "continue" dan "Break":
  - Saya memutuskan untuk menggunakan "continue" di beberapa tempat untuk menunjukkan bagaimana program dapat melewati beberapa bagian siklus jika kondisi tertentu terpenuhi. Ini digunakan ketika saya hanya ingin melanjutkan loop berikutnya tanpa memproses semua logika dalam loop.
  - "Break" digunakan untuk menghentikan perulangan jika kondisi tertentu terpenuhi. Diputuskan untuk menggunakan "jeda" untuk meningkatkan efisiensi dengan menghentikan proses konversi lebih awal daripada melanjutkan.
2. Perubahan Logika Loop:
  - Mengubah batas atas iterasi dari 10 menjadi 15 untuk menguji seberapa mudah retakan mengubah kondisi.
  - Menggunakan kondisi "jika (y % 2 == 0)" membantu memperjelas konsep bahwa hanya bilangan genap yang diproses. Hal ini dilakukan untuk memfokuskan analisis pada bagian data tertentu.
  - Penambahan "else if (y == 9)" digunakan untuk menunjukkan bagaimana loop berakhir ketika suatu nilai tercapai, dan juga menunjukkan bagaimana kontrol mengalir melalui metode sarang.

3. Bersarang untuk pola bintang:

- Algoritma bersarang untuk menghasilkan pola bintang berdasarkan pola masukan yang diberikan. Keputusan untuk menggunakan loop bersarang adalah keputusan yang logis karena menghasilkan keluaran yang dibangun berdasarkan jumlah iterasi yang diperlukan.

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Hedy Rafian Firdaus G1F024027</b>	<b>SWITCH</b>	<b>19 september 2024 2024</b>

## [No. 2] Identifikasi Masalah:

### 1) Uraikan permasalahan dan variabel

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
            }
        }
    }
}
```

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!

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

[Video Materi 2 tentang WHILE – https://www.youtube.com/watch?v=ORA4JyJMFss](https://www.youtube.com/watch?v=ORA4JyJMFss)

## [No. 2] Analisis dan Argumentasi

### 1) Uraikan rancangan solusi yang diusulkan.

#### 2.1. Buat perubahan nilai angka pada variabel di Contoh //Ubah 1 menjadi

**continue; lalu running, periksa hasilnya. Analisa dampaknya perubahan terhadap luaran setelah running dan uraikan kegunaan break dan continue!**

Untuk pemrograman ini, saya menggunakan kode yang disediakan. Lalu, saya mengubah untuk baris **//Ubah 1** menjadi continue. Setelah melakukan pemrograman, melihat fungsi dari break dan continue.

- Break** : Mengakhiri loop while atau for secara keseluruhan.
- Continue** : Mengakhiri eksekusi pernyataan dalam loop while atau for, dan melanjutkan loop pada iterasi berikutnya

#### 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!

Untuk contoh 5, saya menggunakan kode yang telah disediakan. Saya mencoba mengganti kode pada baris **//Ubah2** menjadi **(count %5 == 0)**. Setelah kode dijalankan. Hasilnya pun keluar. Untuk fungsi “%” pada angka yang berbeda adalah :

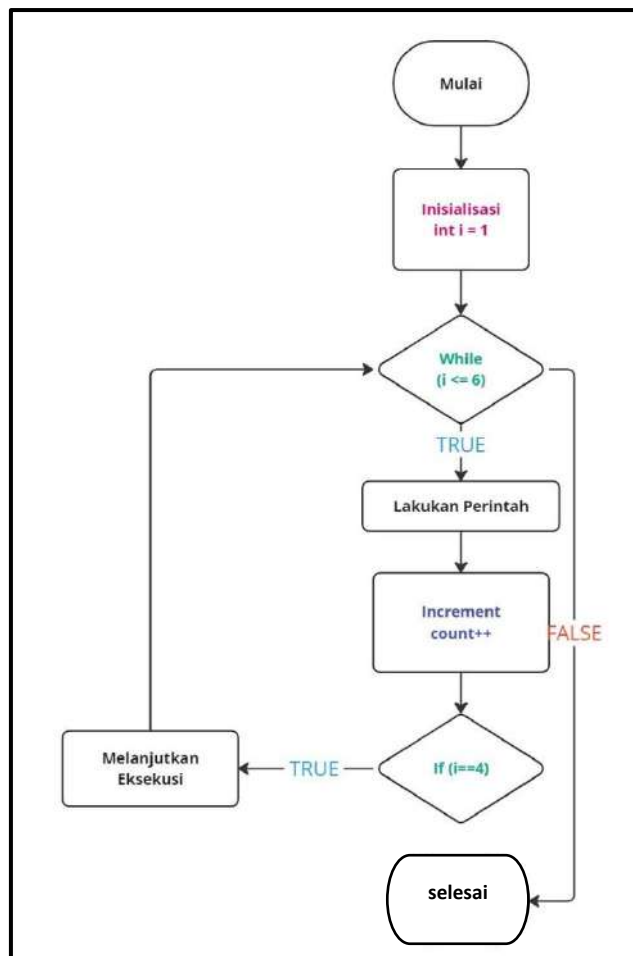
- a) Mendapatkan sisa dari pembagian dua bilangan
- b) Pengecekan genap atau ganjil
- c) Lopping dengan modulus
- d) Penggunaan dalam bilangan negatif

**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!**

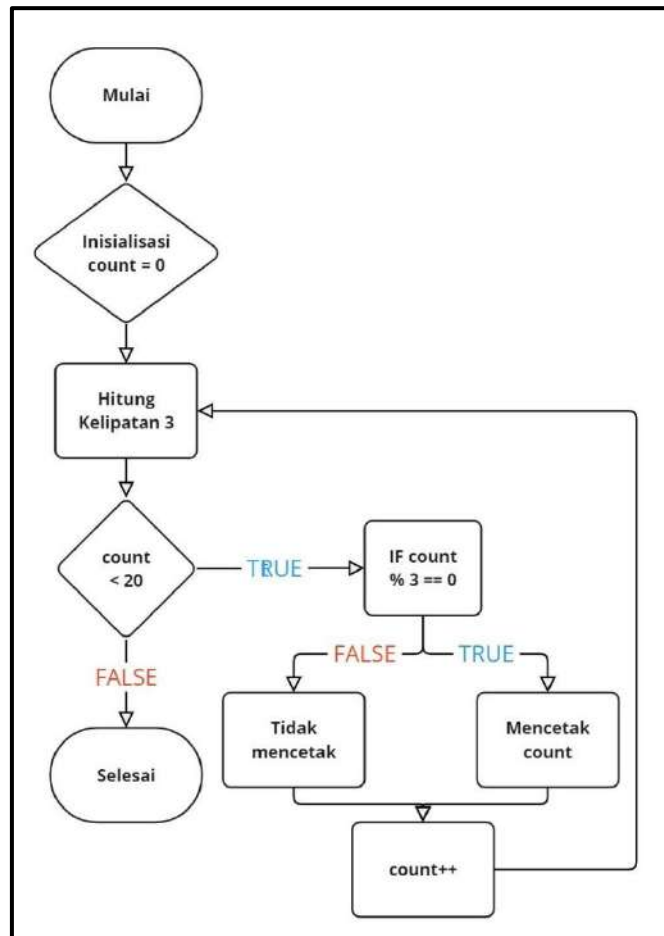
- a) Saya menggunakan kode yang disediakan. Lalu, saya mengganti variabel pada baris **//Ubah1** menjadi **while (count < 0) {**, lalu running. Hasilnya adalah 0, 5, 10, 15.
- b) Saya menggunakan format kode percobaan sebelumnya. Saya menambahkan kode program Do-While dengan persyaratan yang sama, yaitu **while (count < 0)** lalu running. Hasilnya adalah 0.

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

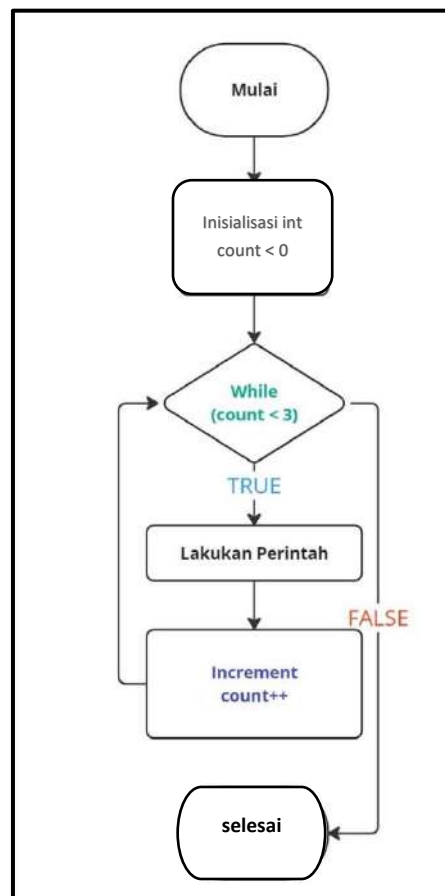
- **Flowchart 2.1**



- Flowchart Contoh 5



- Flowchart 2.3



## [No. 2] Penyusunan Algoritma dan Kode Program

### 1) Rancang desain solusi atau algoritma

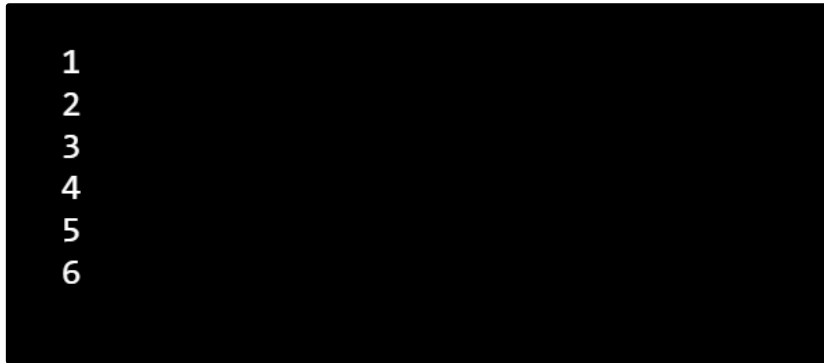
- Kode Contoh 4
  - a) Mulai
  - b) Memberi nama kelas dan nama file ContohWhile
  - c) Deklarasi method utama
  - d) Menginput variabel integer i = 1
  - e) Menginput While dengan (i <= 6)
  - f) Melakukan print untuk variabel i
  - g) Menginput decrement i++;
  - h) Membuat program if dengan (i == 4)
  - i) Menginput kode break/continue
  - j) Menghasilkan luaran
  - k) Selesai
  
- Kode Contoh 5.1
  - a) Mulai
  - b) Memberi nama kelas dan nama file WhileBersarang
  - c) Deklarasi method utama
  - d) Menginput variabel integer dengan count = 0;
  - e) Menginput While dengan (count < 20)
  - f) Membuat program if dengan (count % 5 == 0)
  - g) Menginput Melakukan print untuk variabel count
  - h) decrement count++;
  - i) Menghasilkan luaran
  - j) Selesai
  
- Kode Contoh 5.2
  - a) Mulai
  - b) Memberi nama kelas dan nama file WhileBersarang
  - c) Deklarasi method utama
  - d) Menginput variabel integer count = 0;
  - e) Menginput program do{
  - f) Melakukan print untuk variabel count
  - g) Menginput decrement count++
  - h) Menginput While dengan (count < 0)
  - i) Membuat program if dengan (count % 5 == 0)
  - j) Menghasilkan luaran
  - k) Selesai

### 2) Tuliskan kode program dan luaran

#### a) Contoh 4

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

Gambar Contoh 4

A black rectangular box containing the numbers 1, 2, 3, 4, 5, and 6, each on a new line, representing the output of a program.

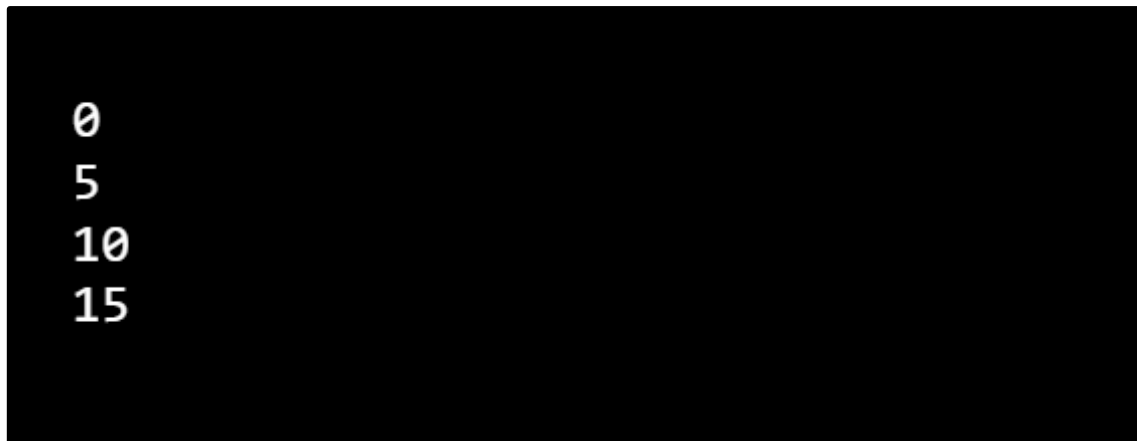
Luaran Contoh 4

Luaran yang dihasilkan telah sesuai dengan yang telah diprogram. Ini dibuktikan dengan hasil; yang didapat telah sesuai dengan kode program yang telah dilakukan.

**b) Contoh 5**

```
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    }
```

Gambar Contoh 5.1

A black rectangular box containing the numbers 0, 5, 10, and 15, each on a new line, representing the output of a program.

Luaran contoh 5.1

Luaran yang dihasilkan telah sesuai dengan yang telah diprogram. Ini dibuktikan dengan hasil; yang didapat telah sesuai dengan kode program yang telah dilakukan.

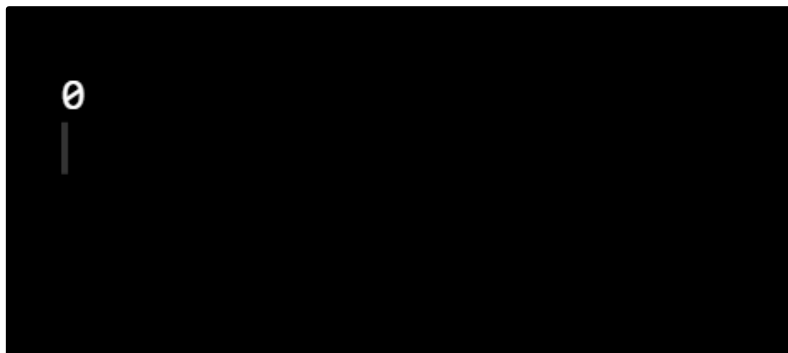


```

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

```

Contoh 5.2



Luaran 5.2

Luaran yang dihasilkan telah sesuai dengan yang telah diprogram. Ini dibuktikan dengan hasil; yang didapat telah sesuai dengan kode program yang telah dilakukan.

## [No. 2] Kesimpulan

### Analisa

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

##### 1. Perubahan pada perintah continue di dalam loop:

- **Masalah:** Mengubah baris kode dari //Ubah 1 menjadi continue akan mempengaruhi jalannya iterasi dalam sebuah loop. Dengan menggunakan continue, ketika kondisi tertentu terpenuhi, loop akan langsung melompat ke iterasi berikutnya tanpa mengeksekusi sisa pernyataan dalam loop tersebut.
- **Algoritma:** Algoritma tetap berjalan, namun ketika mencapai kondisi yang memicu continue, ia akan melewati pernyataan di dalam loop untuk iterasi tersebut dan melanjutkan ke iterasi berikutnya. Hal ini menyebabkan output tertentu tidak ditampilkan atau perintah tertentu tidak dieksekusi pada iterasi tersebut.
- **Kode Program:** Program tetap berjalan seperti biasa, tetapi dengan perilaku yang diubah pada iterasi di mana continue dipicu. Loop tidak dihentikan, tetapi ada sebagian kode yang tidak dieksekusi pada iterasi tertentu.

##### 2. Perubahan pada operator modulus % dalam kondisi if:

- **Masalah:** Mengubah kondisi menjadi if (count % 5 == 0) akan membuat program memeriksa apakah suatu bilangan dapat dibagi habis oleh 5. Jika iya, pernyataan di dalam blok if akan dijalankan.
- **Algoritma:** Operator modulus (%) digunakan untuk mendapatkan sisa dari pembagian. Jika sisa pembagian 0, itu berarti angka tersebut dapat dibagi habis oleh 5. Algoritma ini sering digunakan untuk menentukan kelipatan atau pola bilangan.

- **Kode Program:** Program akan mengeksekusi pernyataan dalam blok if hanya ketika count merupakan kelipatan 5.

### 3. Perbandingan antara while dan do-while:

- **Masalah:** Perubahan dari while menjadi do...while mengubah urutan evaluasi kondisi dalam loop.
- **Algoritma:**
  - **while:** Kondisi dievaluasi sebelum eksekusi perulangan dimulai, sehingga jika kondisi salah sejak awal, perulangan tidak akan dijalankan.
  - **Do-while:** Perintah dalam loop dijalankan setidaknya satu kali, karena kondisi dievaluasi setelah eksekusi perulangan pertama.
- **Kode Program:** Pada do-while, loop akan selalu dijalankan minimal sekali, terlepas dari kondisi yang diberikan, berbeda dengan while yang mungkin tidak akan dijalankan sama sekali jika kondisi awal tidak terpenuhi.

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

1. **Efisiensi eksekusi program:** Menggunakan continue dapat menghemat eksekusi jika ada kondisi di mana iterasi tidak perlu mengeksekusi sisa pernyataan dalam loop, sehingga memberikan performa yang lebih baik.
2. **Keterbacaan kode:** Penggunaan modulus % sering digunakan dalam situasi yang melibatkan bilangan kelipatan, sehingga perubahan pada kondisi if ( $\text{count} \% 5 == 0$ ) mempermudah untuk menangani kasus-kasus khusus seperti bilangan kelipatan lima.
3. **Keperluan deklarasi minimum:** Perubahan dari while ke do-while dibuat jika ingin memastikan bahwa loop dijalankan setidaknya sekali, terlepas dari kondisi awal. Ini memberikan jaminan bahwa kode di dalam loop dijalankan, yang tidak bisa dijamin oleh while.

#### Refleksi :

Setelah mengerjakan laporan ini, saya menjadi lebih tau mengenai penggunaan For dan While. Saya mampu mengerjakan tugas – tugas yang diberikan, walaupun ada sedikit kesulitan. Tetapi, saya bisa meminta bantuan dengan teman – teman saya sehingga saya dapat mengerjakannya dengan baik. Walaupun ada kekurangan dalam laporan ini, tapi saya yakin bahwa saya bisa lebih baik lagi.