

## Template Lembar Kerja Individu dan Kelompok

<b>Nama &amp; NPM</b>	<b>Topik:</b>	<b>Tanggal:</b>
<b>Vicky Andrean &amp; G1F024021</b>	<b>UNIT 1 FOR</b>	<b>09/10/2024</b>

**[1] Identifikasi Masalah:**

- 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!
- Buat perubahan kode pada Contoh 2 di baris `//Ubah1` menjadi
    - `continue` pertama; lalu running, periksa hasilnya
    - `break` pertama; lalu running, periksa hasilnya
    - `continue` kedua; lalu running, periksa hasilnyaAnalisa perbedaan perubahan kode pada Ubah 1 untuk setiap poin (a), (b), dan (c)!
- Cermati kode contoh 3. Apabila ingin menghasilkan luaran berikut:  
Luaran:  
Masukan Input: 7  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*  
\*\*\*  
\*\*  
\*  
  
Susunlah analisa kode untuk menghasilkan luaran tersebut!
- Analisa diagram flowchart dari Latihan 1.2 dan 1.3!

## [1] Analisis dan Argumentasi

- 1) 1.1 continue untuk melanjutkan kode dan break untuk memberhentikan kode di perintah yang di berikan  
Ubah 1, hasil yang di berikan tetap sama karena di break di angka 8  
Ubah 2, hasil yang awal nya 0 2 4 6 menjadi 1,3,5,7,9,11,13,15 karena yang dibagi tidak habis dari 2 kalo yang awal akan mengeluarkan hasil yang habis di bagi 2  
Ubah 3, hasil yang keluar berhenti di angka 7 karena akan di break di angka 9  
Dampak perubahan dari ketiga ubah adalah mengubah batas angka yang akan dijalankan, angka yang dikeluarkan dan mengubah dimana break akan dijalankan
- 2) 1.2 a. hasil yang di dikeluarkan sama seperti sebelumnya, i 2 j 2 akan di loncatkan karena if Langsung mengcontinue  
b. hasil yang dikeluarkan hanya batas angka 2 karena if i == 2 akan di break  
c. hasil akan keluar seperti layaknya perintah  
dampak perubahan yang di berikan adalah jika menggunakan break di pertama akan memberhentikan hanya batas pertama jika continue akan meloncat ke selanjutnya
- 3) 1.3 mengubah s=tinggi menjadi s=1 dan mengubah s=> menjadi s<= dan s—menjadi s++

### [1] Penyusunan Algoritma dan Kode Program

- 1) 1.1 Rancang desain solusi atau algoritma
  - a. Mulai
  - b. Inisialisasi

- c. Membuat loop
- d. Pengecekan if
- e. Pengecekan else if
- f. Tampilkan nilai
- g. Break / mengakhiri loop
- h. selesai

<pre> 1- public class ContohFor{ 2- public static void main(String[] args) { 3-     for (int y = 0; y &lt;= 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-     } } } </pre>	<pre> java -cp /tmp/TkPoT8RkYa/ContohFor 1 3 5 7 === Code Execution Successful === </pre>
--	---

Gambar 1.1 hasil dan output

## 1.2 Algoritma

- a.mulai
- b.inisialisasi perulangan i
- c.inisialisasi perulangan j
- d.pengecekan if
- e.selesai

<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-                 } } } </pre>	<pre> java -cp /tmp/w0y8J06RW6/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>
--	--

Gambar 1.2 hasil dan output

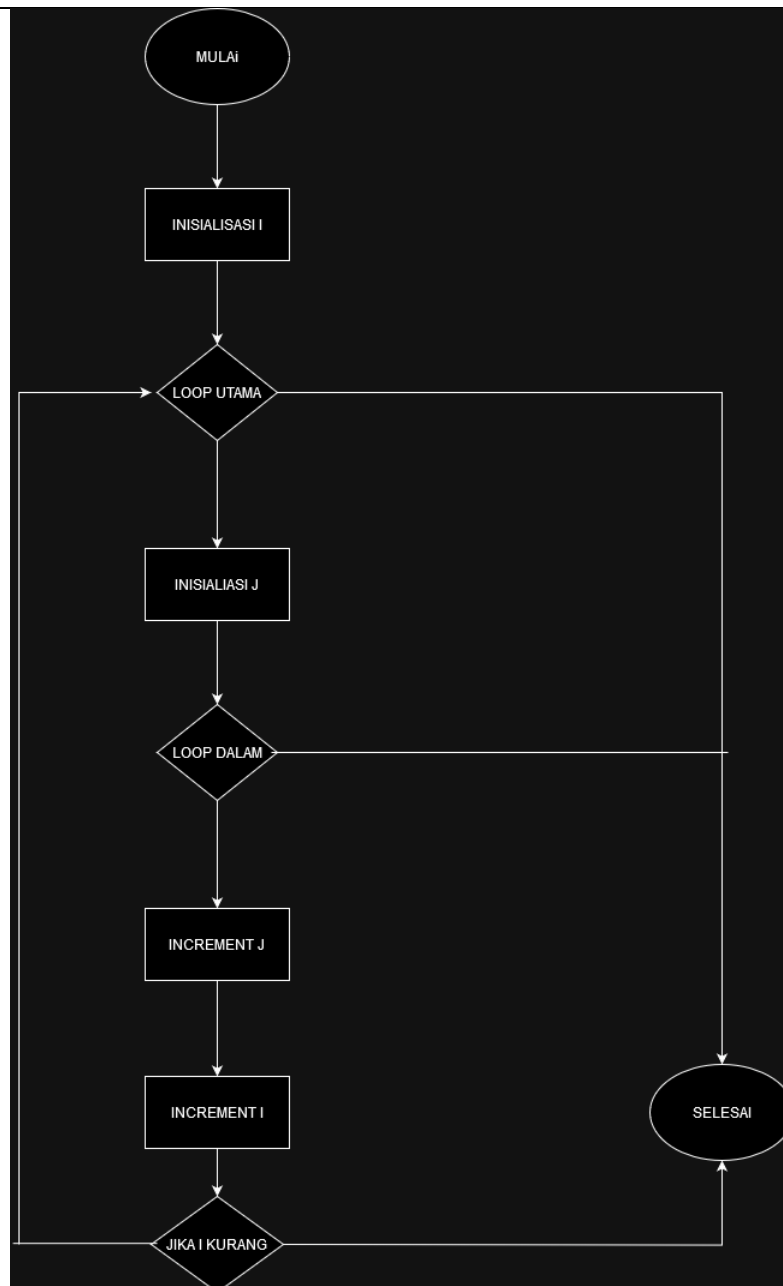
## 1.3 Algoritma

- a.mulai
- b.Deklarasi scanner
- c.masukan input
- d.membaca input
- e.loop tinggi
- f.selesai

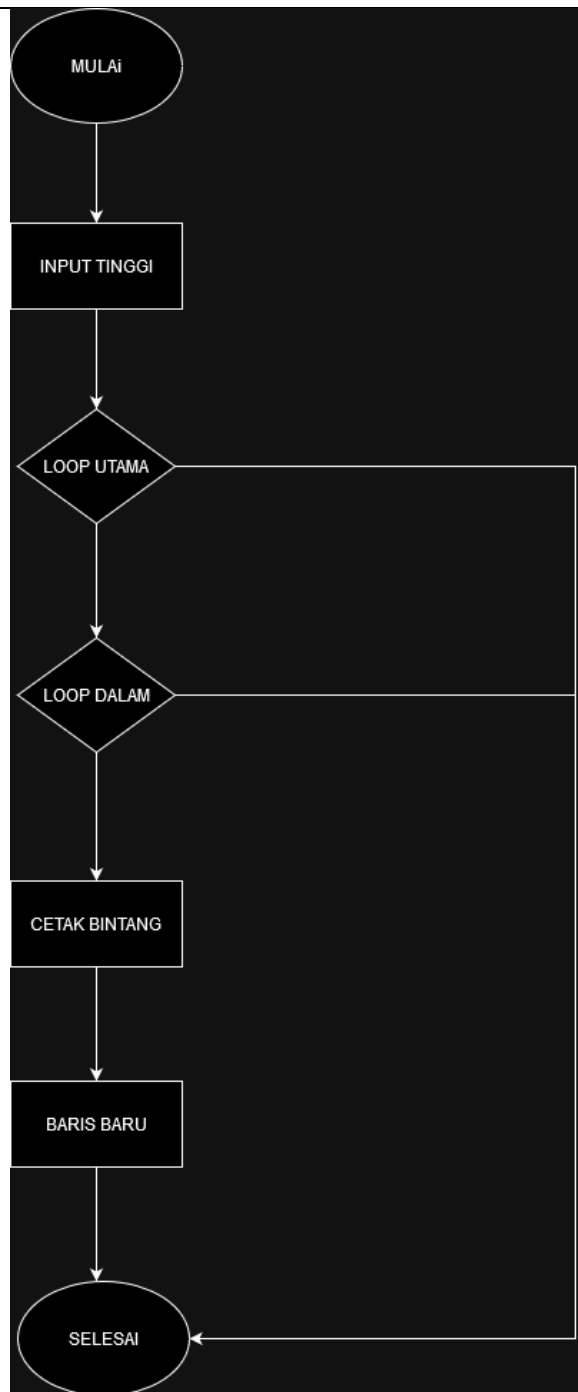
<pre> 1- import java.util.Scanner; 2- 3- public class ForBersarang { 4-     public static void main(String[] args){ 5-         //Instance Input Scanner 6-         Scanner input = new Scanner(System.in); 7-         System.out.print("Masukan Input: "); 8-         int tinggi = input.nextInt(); //Mendapatkan Input 9-         //Dari User 10-        for(int t=tinggi; t&gt;=1; t--){ 11-            //Menghitung Jumlah Tinggi Piramida 12-            for(int s=1; s&lt;=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/Kypappxz36/ForBersarang Masukan Input: 7 ***** ***** **** *** ** * === Code Execution Successful === </pre>
---	---

Gambar 1.3 hasil dan output

Gambar flowchart dari 1.2 dan 1.3



**Gambar 1.4 Flowchart 1.2**



**Gambar 1.5 flowchart 1.3**

**[1] Kesimpulan**

1) Evaluasi

- a) Apa konsekuensi dari skenario pemrograman ini?  
Jika terjadi salah input maka akan error

## Template Lembar Kerja Individu dan Kelompok

Nama & NPM	Topik:	Tanggal:
G1F024021	UNIT 2 WHILE	10/10/2024

### [2] Identifikasi Masalah:

- 5) 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!
- 6) 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!
- 7) 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!
- 8) 2.4. Analisa diagram flowchart dari Latihan 2.1, Contoh 5, dan Latihan 2.3!

### [2] Analisis dan Argumentasi

- 4) 2.1 hasil dari break adalah 1 2 3 sedangkan saat di ganti menjadi continue 1 2 3 4 5 6 karena di perintah mengeprint 6 – 1 dan saat di break berhenti di angka 4
- 5) 2.2 kode pertama output yang muncul adalah hasil yang di bagi Ketika if diganti dengan if (count % 5 == 0) maka output yang keluar adalah angka yang bisa di bagi 5
- 6) 2.3 saat ubah1 menjadi while (count < 0) { hasilnya adalah error karena variable count tidak ada, Ketika contoh 5 di ubah menjadi do while maka yang di output akan muncul 0 karena perintah akan menjalankan loop setidaknya sekali

### [2] Penyusunan Algoritma dan Kode Program

- 2) 2.1 Rancang desain solusi atau algoritma
  - a. Mulai
  - b. Inisialisasi variable
  - c. Mengecek while
  - d. Cetak nilai i
  - e. Menginkremen nilai
  - f. Cek if
  - g. Mengulang Kembali
  - h. Selesai

```

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- }
12- }
13- }
14-

```

```

java -cp /tmp/A26qQrmiky/ContohWhile
1
2
3
4
5
6
=== Code Execution Successful ===

```

Gambar 2.1 input dan output

- 2.2 algoritma
  - a.mulai

- b.inisialisasi variable
- c.cek kondisi while
- d.cek kondisi if
- e.menambah angka dengan inkremen
- f.kembali ke while
- g.selesai

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/HM1AyNdZpT/WhileBersarang 0 5 10 15  === Code Execution Successful === </pre>

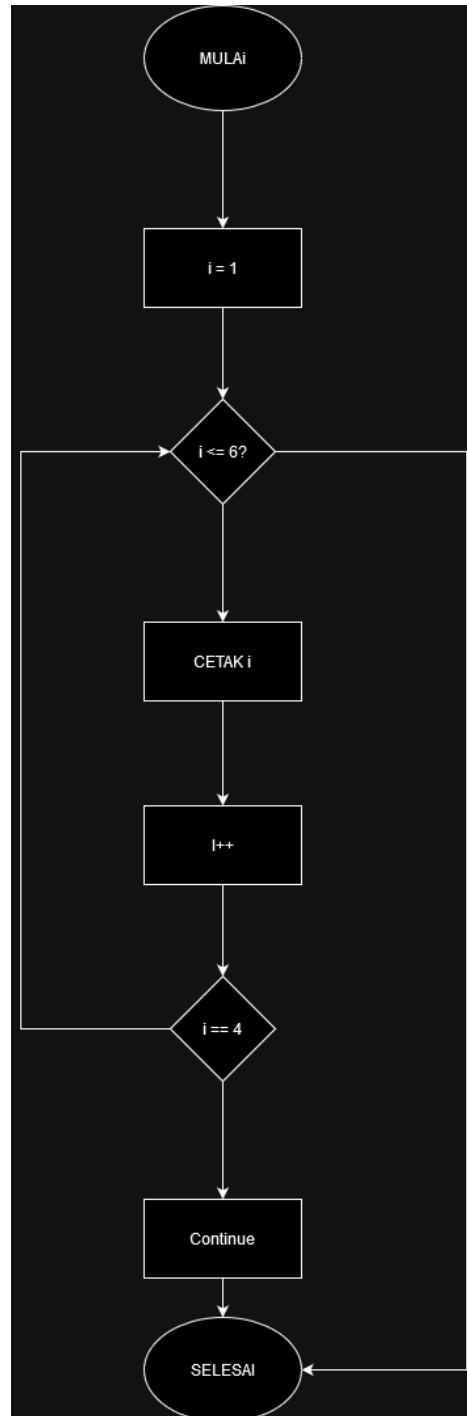
Gambar 2.2 Hasil dan output

- 2.3 algoritma
- a.mulai
- b.inisialisasi variable
- c.jalankan do
- d.cek kondisi while
- e.selesai

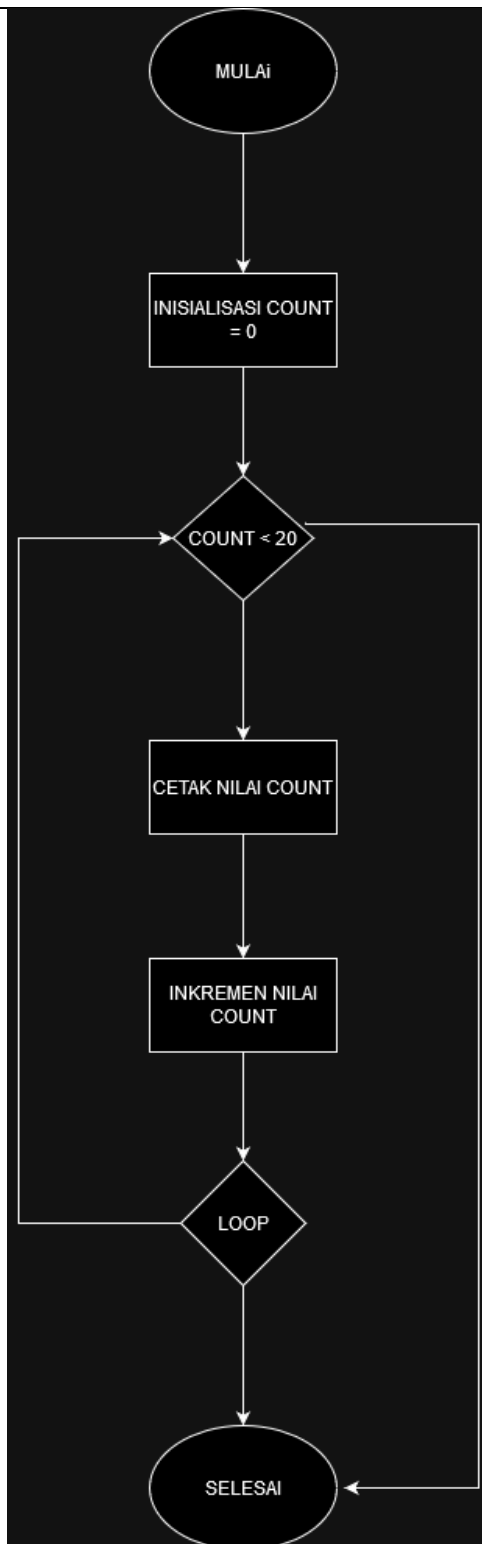
Main.java	Output
<pre> 1 public class WhileBersarang { 2     public static void main(String[] args) { 3         int count = 0; 4         do { 5             if (count % 3 == 0) //ubah2 6                 System.out.println(count); 7                 count++; 8             } while (count &lt; 0); 9         } 10    } </pre>	<pre> java -cp /tmp/GU8UJ3ipf6/WhileBersarang 0  === Code Execution Successful === </pre>

Gambar 2.3 hasil dan output

GAMBAR FLOWCHART LATIHAN 2.4

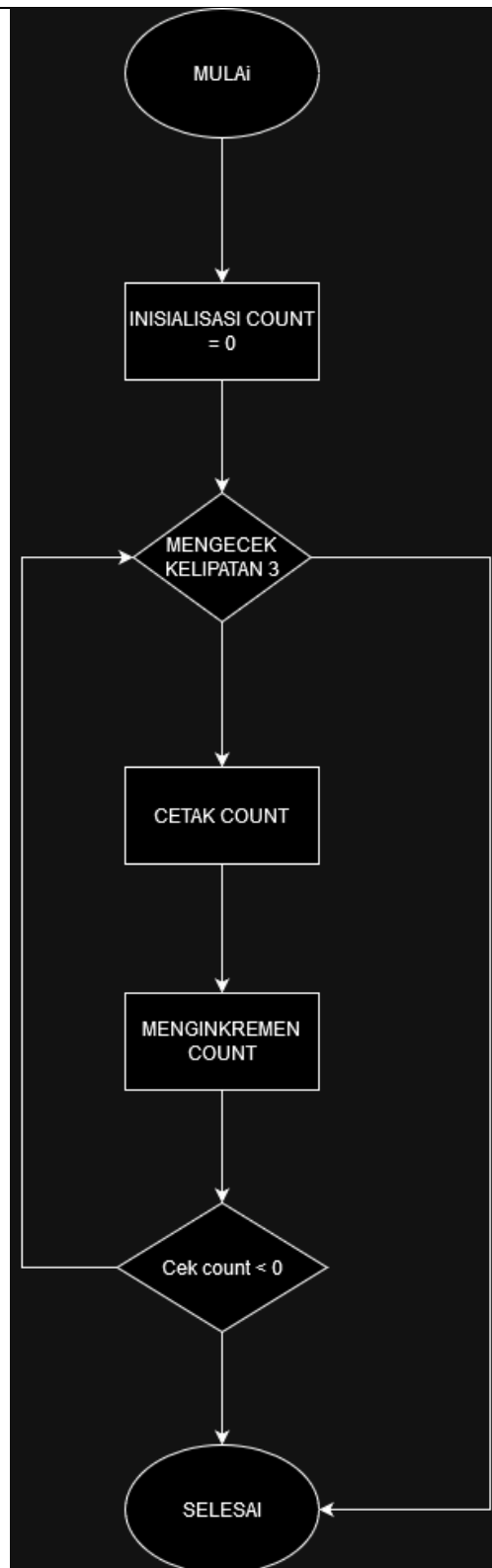


Gambar 2.4 Flowchart 2.1



Gambar 2.5 Flowchart contoh 5





Gambar 2.6 Flowchart 2.3

**[Nomor Soal] Kesimpulan**

2) Evaluasi

Penggunaan kode yang tidak efisien, input yang di berikan harus di rubah manual