

## Template Lembar Kerja Individu

Nama & NPM	Topik:	Tanggal:
Nafiisah Calista Hersa Agustine (G1F024077)	For dan While	07 Oktober 2024

### [Nomor 1] Identifikasi Masalah:

1) Uraikan permasalahan dan variable

Contoh 1 :

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

Contoh 2 :

```
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
                    }
            }
    }
}
```

Contoh 3 :

```
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
        }
    }
}
```

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

```
public class ContohFor{
public static void main(String[] args) {
    for (int y = 0; y <= 15; y++) {
        if (y % 2 == 1)
            continue;
```

```

        else if (y == 8)
            break;
        else
            System.out.println(y + " ");
    } } }

```

Luarannya :

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

```

public class ContohFor{
public static void main(String[] args) {
    for (int y = 0; y <= 15; y++) {    //ubah 1
        if (y % 2 == 0)                //ubah 2
            continue;    //baris 1
        else if (y == 8)              //ubah 3
            break;    //baris 2
        else
            System.out.println(y + " ");
    } } }

```

Luarannya :

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

```

public class ContohFor{
public static void main(String[] args) {
    for (int y = 0; y <= 15; y++) {    //ubah 1
        if (y % 2 == 0)                //ubah 2
            continue;    //baris 1
        else if (y == 9)              //ubah 3
            break;    //baris 2
        else
            System.out.println(y + " ");
    } } }

```

Luarannya :

Analisa dampaknya perubahan ini terhadap luaran setelah running!

Secara keseluruhan, perubahan ini membuat program lebih selektif dalam outputnya, hanya menampilkan angka ganjil sebelum 9 dan tidak mencetak angka 9 atau angka ganjil setelahnya. Jika tujuan untuk menghentikan loop lebih awal dan menghindari pencetakan angka yang lebih tinggi, perubahan ini mencapai tujuan tersebut dengan efektif.

1.2. Buat perubahan kode pada Contoh 2 di baris //Ubah1 menjadi

a. `continue` pertama; lalu running, periksa hasilnya

```

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

```

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

```

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

```

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

```

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

```

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

Pada (a) dan (c), pakai continue bikin kedua nilai j tetap dicetak pas i sama dengan 2, jadi outputnya sama. Di (b), pakai break bikin loop kedua berhenti pas i 2, jadi Cuma cetak satu nilai j dan lanjut ke i berikutnya.

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

```

Luaran:
Masukan Input: 7
*****
*****
*****
****
***
**
*

```

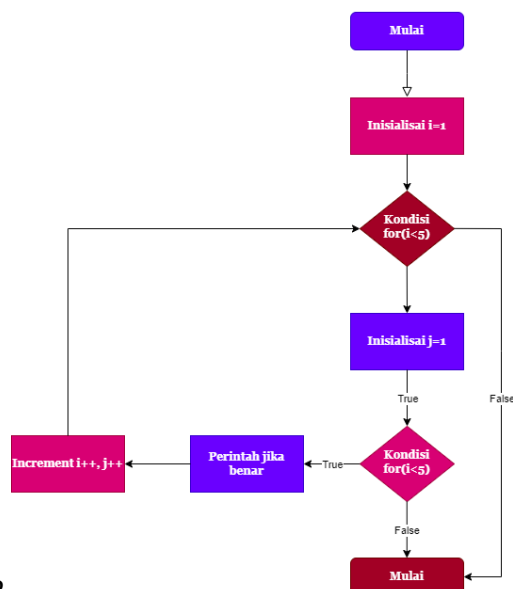
Susunlah analisa kode untuk menghasilkan luaran tersebut!

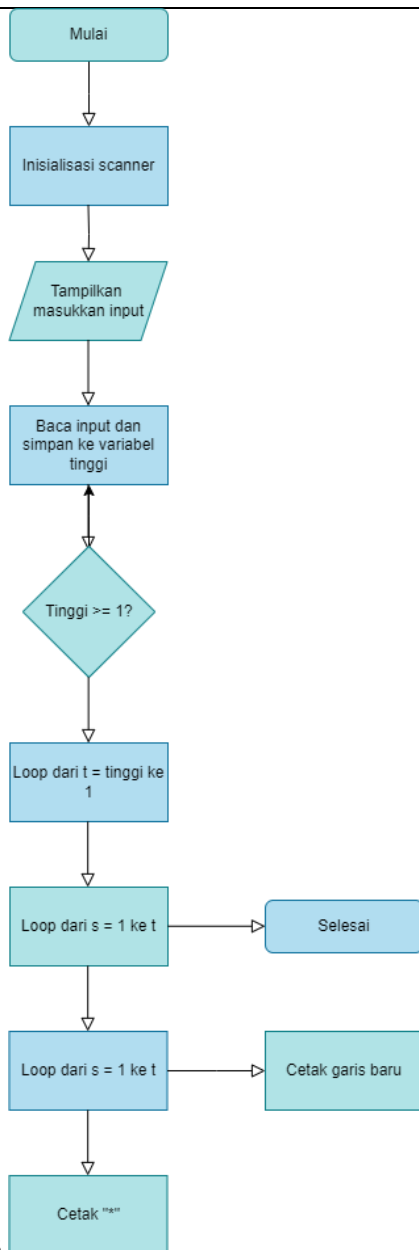
```

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 = 1; s <= t; s++){
                //Menghitung Jumlah Spasi per Baris
                System.out.print(" ");
            }
            System.out.println(); //Membuat Baris Baru
        }
    }
}

```

1.4. Analisa diagram flowchart dari Latihan 1.2 dan 1.3!





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)

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

### [Nomor 1] Analisis dan Argumentasi

1) Uraikan rancangan solusi yang diusulkan.

Contoh 1 :

Rancangan solusi adalah program java yang mencetak angka ganjil dari 0 hingga 15, mengabaikan angka genap dan menghentikan output saat angka mencapai angka 9 dengan menggunakan control alur continue dan break.

Contoh 2 :

Rancangan solusi pakai loop bersarang buat mencetak kombinasi nilai i dan j, dengan control pakai continue buat melewati iterasi tertentu atau buat break buat berhenti di loop dalam berdasarkan nilai i.

Contoh 3 :

Program ini minta pengguna masukan tinggi piramida, lalu pakai loop bersarang buat cetak pola bintang terbalik sesuai tinggi yang diminta, dengan setiap baris nampilinnya jumlah bintang yang sesuai.

2) Analisis solusi, kaitkan dengan permasalahan.

Contoh 1 :

Solusi ini efektif dalam memenuhi tujuan pencetakan angka ganjil, meskipun terbatas pada rentang 0 hingga 15, jika rentang berubah, program perlu disesuaikan untuk tetap berfungsi dengan baik.

Contoh 2 :

Analisisnya, pakai continue bikin semua kombinasi tetap dicetak saat i 2, sedangkan break Cuma cetak satu iterasi, jadi pilihan antara keduanya ngaruh ke hasil output sesuai kebutuhan.

Contoh 3 :

Solusi ini efektif dan fleksibel, jadi pengguna bisa cetak pola dengan ukuran yang berbeda beda. Dengan struktur yang sederhana, program ini berhasil menyelesaikan masalah dengan menghasilkan pola yang diinginkan secara akurat dan cepat.

#### **[Nomor 1] Penyusunan Algoritma dan Kode Program**

1) Rancang desain solusi atau algoritma

Contoh 1 :

1. Inisialisasi
2. Loop
3. Cek ganjil atau genap
4. Selesai

Contoh 2 :

1. Inisialisasi
2. Loop utama
3. Loop dalam
4. Kontrol aliran
5. Akhiri loop
6. Selesai

Contoh 3 :

1. Inisialisasi
2. Input dari pengguna
3. Menghitung baris piramida
4. Selesai

2) Tuliskan kode program dan luaran

a) Beri komentar pada kode

Contoh 1 :

Kode java ini mencetak semua angka ganjil dari 0 hingga 15, tetapi berhenti ketika mencapai angka 9. Jika angka yang sedang diperiksa adalah genap, proses akan melanjutkan ke iterasi berikutnya tanpa mencetaknya.

Contoh 2 :

Kode ini pakai dua loop bersarang buat cetak kombinasi nilai i (1 sampai 4) dan j (1 sampai 2), dengan control aliran pakai label. Jadi, kalau i 2, bisa pakai continue buat lewat iterasi, atau break buat keluar dari loop dalam.

Contoh 3 :

Kode java ini meminta pengguna memasukan tinggi piramida, mengecek apakah inputnya valid, lalu mencetak piramida terbalik dari asterisk sesuai tinggi yang dimasukkan, sambil menutup sumber daya setelah selesai.

b) Uraikan luaran yang dihasilkan

Contoh 1 :

1  
3  
5  
7

Contoh 2 :

a.

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

b.

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

c.

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

Contoh 3 :

Masukan Input: 7

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*

\*

c) Screenshot/ Capture potongan kode dan hasil luaran

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 == 0) //ubah 2
5-             continue;
6-         else if (y == 9) //ubah 3
7-             break;
8-         else
9-             System.out.println(y + " ");
10-     } }
11 }
```

Input/Output

Language Version: JDK 21.0.0 ☒ Interactive

Input Arguments

Output Generated Files

```
1
3
5
7
```

Contoh 2 :  
1.2 (a)

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

Input/Output

Language Version: JDK 21.0.0 ☒ Interactive

Input Arguments

Output Generated Files

```
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
```

1.2 (b)

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

Input/Output

Language Version: JDK 21.0.0 ☒ Interactive

Input Arguments

Output Generated Files

```
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
```

1.2 (c)

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

Input/Output

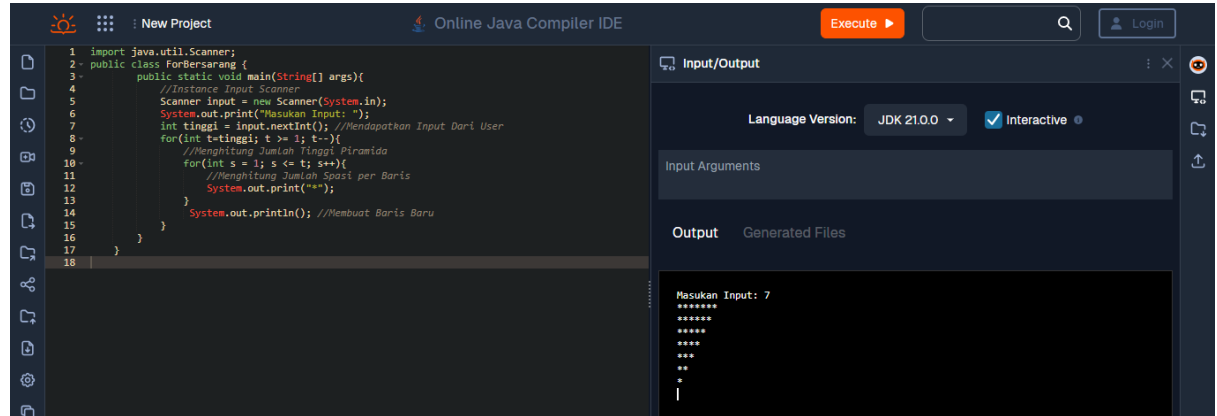
Language Version: JDK 21.0.0 ☒ Interactive

Input Arguments

Output Generated Files

```
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
```

### Contoh 3 :



```
1 import java.util.Scanner;
2 public class ForBersarang {
3     public static void main(String[] args){
4         //Instance Input Scanner
5         Scanner input = new Scanner(System.in);
6         System.out.print("Masukan Input: ");
7         int tinggi = input.nextInt(); //Mendapatkan Input Dari User
8         for(int t=tinggi; t >= 1; t--){
9             //Menghitung Jumlah Tinggi Piramida
10            for(int s = 1; s <= t; s++){
11                //Menghitung Jumlah Spasi per Baris
12                System.out.print(" ");
13            }
14            System.out.println(); //Membuat Baris Baru
15        }
16    }
17 }
18
```

Input/Output

Language Version: JDK 21.0.0 ☒ Interactive

Input Arguments

Output Generated Files

Masukan Input: 7

```
*****
****
***
**
*
|
```

#### [Nomor 1] Kesimpulan

##### 1) Evaluasi

- a) Apa konsekuensi dari skenario pemrograman ini?

##### Contoh 1 :

Program ini hanya mencetak angka ganjil dari 0 hingga 7 dan menghentikan proses saat mencapai angka 9, yang bias membingungkan pengguna yang berharap melihat angka ganjil lebih tinggi.

##### Contoh 2 :

Pakai control aliran seperti continue dan break bisa ngaruh ke output yang dihasilkan dan bikin kode jadi lebih rumit, jadi penting buat paham cara kerjanya supaya nggak bingung.

##### Contoh 3 :

Program mencetak piramida terbalik berdasarkan input tinggi dari pengguna, tetapi tidak menangani kasus input negated, nol, atau karakter non integer, yang dapat menyebabkan hasil yang tidak diharapkan atau error.

- b) Evaluasi input, proses, dan luaran yang dihasilkan! (jika ada)

##### Contoh 1 :

Program tidak mengambil input dinamis, menggunakan loop untuk memeriksa angka dalam rentang statis, dan hanya mencetak angka ganjil yang kurang dari 9, yaitu 1,3,5,dan 7.

##### Contoh 2 :

Tidak ada input dari pengguna, program ini jalanin dua loop bersarang buat cetak kombinasi nilai i dan j. luaran yang dihasilkan tergantung pemakaian continue atau break.

##### Contoh 3 :

Input berupa integer tinggi piramida, proses menggunakan dua loop bersarang untuk mencetak bintang, dan luaran berupa piramida terbalik yang sesuai dengan tinggi yang diberikan, seperti contoh untuk input 5 menghasilkan lima bintang di baris pertama dan berkurang satu bintang di setiap baris berikutnya.



### Template Lembar Kerja Individu

Nama & NPM	Topik:	Tanggal:
Nafiisah Calista Hersa Agustine (G1F024077)	For dan While	07 Oktober 2024

#### [Nomor 2] Identifikasi Masalah:

1) Uraikan permasalahan dan variable

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){
            break;        //ubah1
        }
    }
}
```

Contoh 5 :

```
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++;
        }
    }
}
```

2.1. Buat perubahan nilai angka pada variabel di Contoh 4  
//Ubah 1 menjadi continue; lalu running, periksa hasilnya

```
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
        }
    }
}
```

Hasilnya :

1  
2  
3  
4  
5  
6

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

Perubahan dari break ke continue di loop bikin cara program berjalan berbeda. Dengan break, loop berhenti total pas kondisi tertentu terpenuhi. Sedangkan continue, Cuma melewati iterasi itu dan lanjut ke yang berikutnya. Jadi, break sering dipakai buat berhentiin pencarian atau proses, sementara continue buat menghindari pemrosesan tertentu. Keduanya berguna banget buat ngatur alur program sesuai kebutuhan.

2.2. Buat perubahan nilai angka pada variabel di Contoh 5

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

```
public class WhileBersarang {  
    public static void main(String[] args) {  
        int count = 0; //ubah1  
        while (count < 20) {  
            if (count % 5 == 0) //ubah2  
                System.out.println(count);  
            count++;  
        }  
    }  
}
```

Hasilnya :

0  
5  
10  
15

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

Mengganti angka di `if (count % 5 == 0)` bakal bikin program mencetak angka yang kelipatan dari x. operator % itu buat ngecek sisa bagi, jadi kita bisa tahu apakah angka itu kelipatan, genap, atau ganjil.

2.3. Buat perubahan nilai angka pada variabel di

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

Tidak ada output atau hasilnya, Karena program tidak mencetak apa apa karena count mulai dari 0, dan kondisi `count < 0` adalah salah

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`!

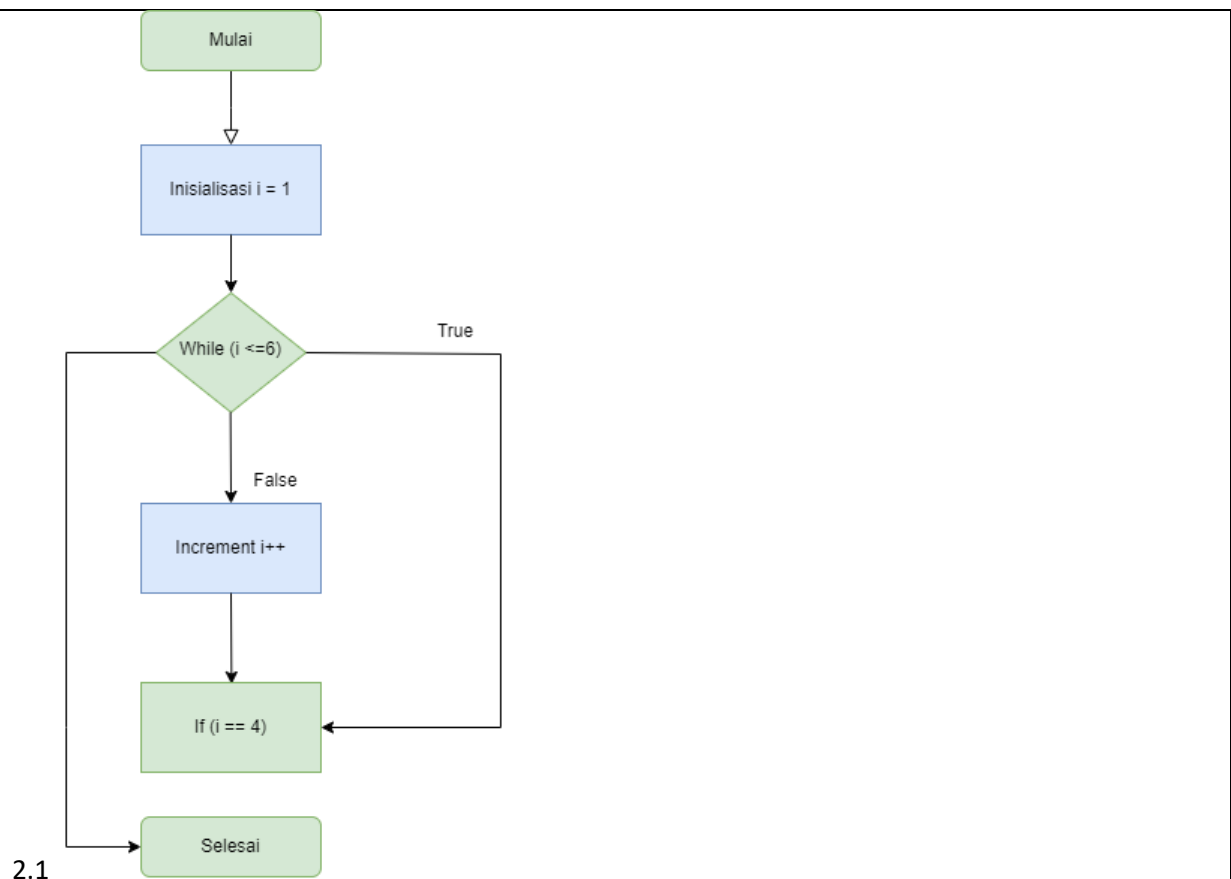
Hasilnya

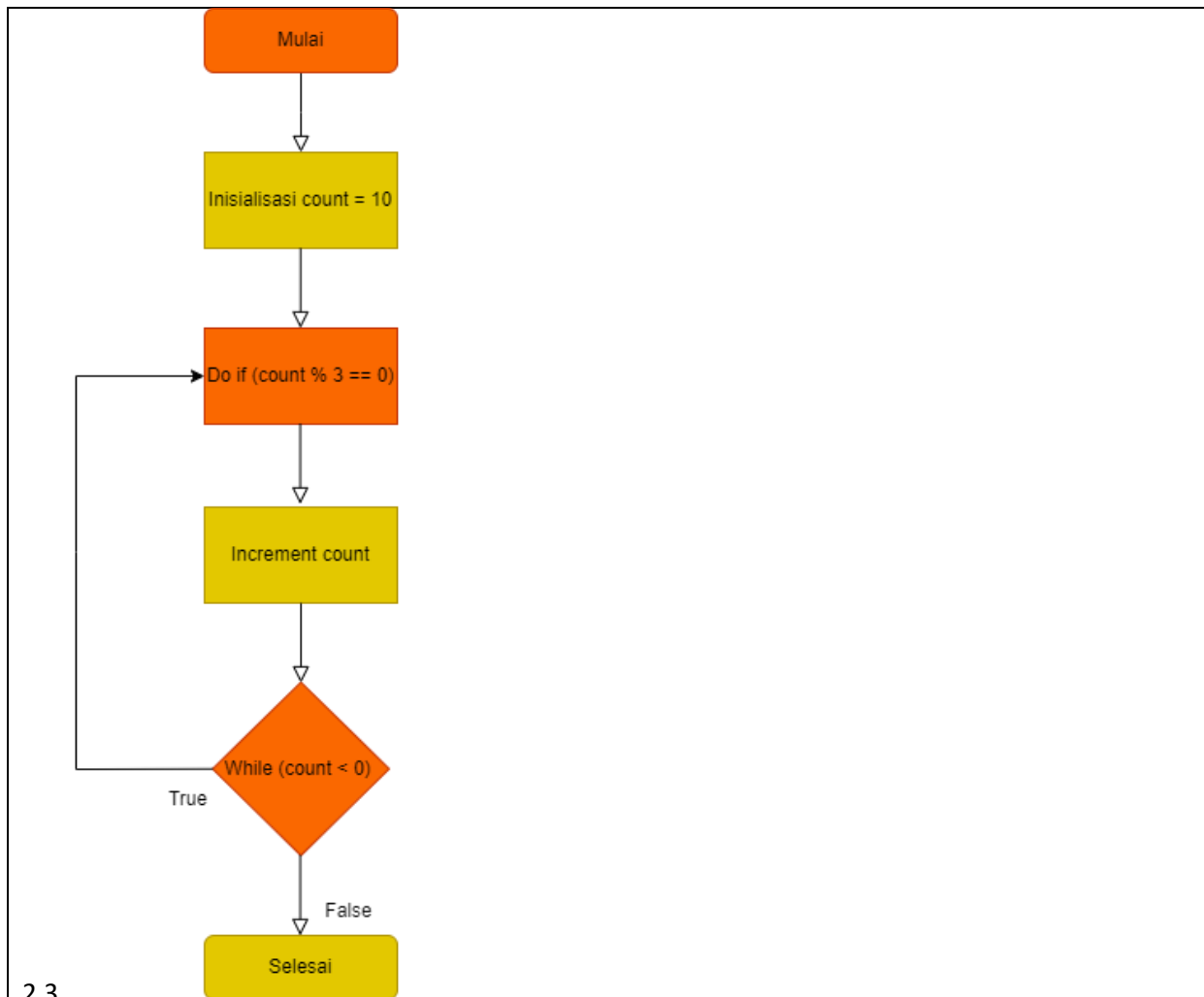
0

`while` tidak menjalankan kode sama sekali jika kondisi awal salah.

`do..while` menjalankannn kode setidaknya sekali, tetapi jika kondisi setelah iterasi pertama salah, tidak ada output.

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





2.3

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

[Video Materi 1 tentang FOR – https://www.youtube.com/watch?v=Ij9qLLbIxEU](https://www.youtube.com/watch?v=Ij9qLLbIxEU)

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

#### [Nomor Soal] Analisis dan Argumentasi

1) Uraikan rancangan solusi yang diusulkan.

Contoh 4 :

Rancangan solusi yang diusulkan adalah pakai loop buat iterasi angka, dimana break berhentiin loop saat sampai angka tertentu, dan continue buat melewati angka tertentu sambil tetap lanjut di loop.

Contoh 5 :

Bikin program buat cetak angka berdasarkan kelipatan tertentu pakai loop while dan do...while. Pertama, kita siapkan variable count, lalu cek angka yang mau dicetak. Ganti kondisi buat lihat perbedaan antara kedua loop. Terakhir, jalankan program dan cek hasilnya.

2) Analisis solusi, kaitkan dengan permasalahan.

Contoh 4 :

Analisis solusi menunjukkan kalau break cocok buat berhenti setelah angka tertentu, sementara continue lebih fleksibel buat ngindarin pencetakan angka yang gak mau diitampilin, jadi keduanya membantu ngatur alur program sesuai kebutuhan.

Contoh 5 :

Permasalahan yang dihadapi adalah memahami bagaimana while dan do...while berfungsi. Hasilnya, while nggak bakal jalan kalau kondisi awal salah, sedangkan do...while tetap jalan minimal sekali. Pemilihan loop tergantung kebutuhan program, jadi kita bisa proses data dengan cara yang tepat.

**[Nomor Soal] Penyusunan Algoritma dan Kode Program**

1) Rancang desain solusi atau algoritma

Contoh 4 :

1. Inisialisasi
2. Looping
3. Tampilkan nilai
4. Increment
5. Kondisi break
6. Kondisi continue
7. Akhir loop

Contoh 5 :

1. Inisialisasi variable
2. Loop dengan while
3. Ganti kondisi
4. Loop dengan do...while
5. Perbandingan hasil
6. Selesai

2) Tuliskan kode program dan luaran

a) Beri komentar pada kode

Contoh 4 :

Kode di atas mulai dengan ngatur variable i ke 1 dan pakai loop while buat cetak nilai i dari 1 sampai 6, dimana i ditambah 1 di setiap putaran. Kalau i sama dengan 4, loop bakal berhenti pakai break, atau bisa juga melewati pencetakan angka 4 pakai continue. Komentar di kode bikin setiap langkah jadi lebih gampang dimengerti.

Contoh 5 :

Kode ini mulai dengan ngatur variable count ke 0 dan pakai loop while buat cetak angka kelipatan 5 selama kurang dari 20. Setelah itu, kondisi diubah jadi count < 0, yang nggak ngasilin output. Terus, ada juga do...while yang coba cetak count kalau kelipatan 5, tapi juga nggak ngasilin output karena kondisi nggak terpenuhi setelah iterasi pertama.

b) Uraikan luaran yang dihasilkan

Contoh 4 :

1  
2  
3  
4  
5  
6

Contoh 5 :

0

c) Screenshot/ Capture potongan kode dan hasil luaran

Contoh 4 :

```
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- }
```

Input/Output

Language Version: JDK 21.0.0 ☒ Interactive

Input Arguments

Output

Generated Files

1  
2  
3  
4  
5  
6

Compiled and executed in 1.828 sec(s)

Contoh 5 :

2.2

```
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-             count++;
10-         }
11-     }
12- }
```

Input/Output

Language Version: JDK 21.0.0 ☒ Interactive

Input Arguments

Output

Generated Files

0  
5  
10  
15

2.3 (a)

```
1- public class WhileBersarang {
2-     public static void main(String[] args) {
3-         int count = 0; //ubah1
4-         while (count < 10) {
5-             if (count % 3 == 0) { //ubah2
6-                 System.out.println(count);
7-                 count++;
8-             }
9-             count++;
10-         }
11-     }
12- }
```

Input/Output

Language Version: JDK 21.0.0 ☒ Interactive

Input Arguments

Output

Generated Files

1

2.3 (b)

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

Input/Output

Language Version: JDK 21.0.0 ☒ Interactive

Input Arguments

Output

Generated Files

0

[Nomor Soal] Kesimpulan

1) Evaluasi

a) Apa konsekuensi dari skenario pemrograman ini?

Contoh 4 :

Konsekuensi dari skenario pemrograman ini adalah control alur jadi fleksibel, tapi kalau nggak hati-hati, bias bikin logika program jadi membingungkan dan kode sulit dibaca, serta bias pengaruhi efisiensi.

Contoh 5 :

Scenario ini bikin programmer lebih paham perbedaan antara loop while dan do...while, plus dampak dari kondisi yang dipakai. Ganti kondisi jadi  $\text{count} < 0$  nunjukin bahwa nggak semua kondisi bakal ngasalin output, yang bisa bikin bingung, terutama buat yang baru belajar. Ini juga ngebahas pentingnya merencanakan logika dalam program.

b) Evaluasi input, proses, dan luaran yang dihasilkan! (jika ada)

Contoh 4 :

Inputnya angka dari 1 sampai 6, prosesnya melibatkan looping dan pencetakan nilai, sementara luaran tergantung pada penggunaan break atau continue. Dengan break, yang dicetak Cuma 1, 2, dan 3, sedangkan dengan continue, yang dicetak 1, 2, 3, 5, dan 6, jadi angka 4 dilewatkan.

Contoh 5 :

Tidak ada input dari pengguna karena program Cuma pake nilai awal untuk count. Prosesnya pakai loop buat cek kelipatan 5, jadi kita bisa lihat cara kerja dasar loop. Luaran dari loop while adalah angka 0, 5, 10, dan 15, tapi pas kondisi diubah ke  $\text{count} < 0$ , nggak ada output yang muncul, sama juga dengan do...while.