

Nama & NPM	Topik:	Tanggal:
Muhammad Hovid Arman G1F024034	For While	7 Oktober 2024

[No. 1] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

Latihan 1

- 1.1. Evaluasi penyebab kesalahan dan perbaiki kode pada Contoh 1!

Rekomendasikan kata kunci yang tepat diletakkan pada baris kode yang kosong 1 dan 2 untuk dapat menghasilkan luaran berikut:

Luaran contoh 1:

```
0
2
4
6
```

- 1.2. Cermati contoh kode 2 pada kode //baris kode kosong.

Rekomendasikan kode yang tepat menggunakan break atau continue terhadap pertama atau kedua agar menghasilkan luaran berikut:

Luaran Contoh 2:

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

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

Luaran berbentuk piramida

Masukan Input: 7

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

Rekomendasikan kode untuk menghasilkan luaran tersebut!

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

[No.1] Analisis dan Argumentasi

- 1) 1.1 Kita melakukan perbaikan pada kode program dengan cara mengubah tipe data menjadi int agar lebih optimal dan mengubah urutan dari ekspresi, yaitu dengan cara mendefinisikan nilai dari variabel y terlebih dahulu baru melakukan pengecekan setelahnya dan juga melakukan perubahan ambang batas nilai y.
- 2) 1.2 Melakukan penambahan break pertama setelah loop kedua, break untuk ketika loop mencapai kondisi $i == 2 \ \&\& \ j == 2$ agar memberikan hasil sesuai yang dii=minta
- 3) 1.3 Menggunakan loop untuk mengatur jumlah baris piramida, mencetaj spasi, dan mencetak Bintang.

[No.1] Penyusunan Algoritma dan Kode Program

1) Algoritma

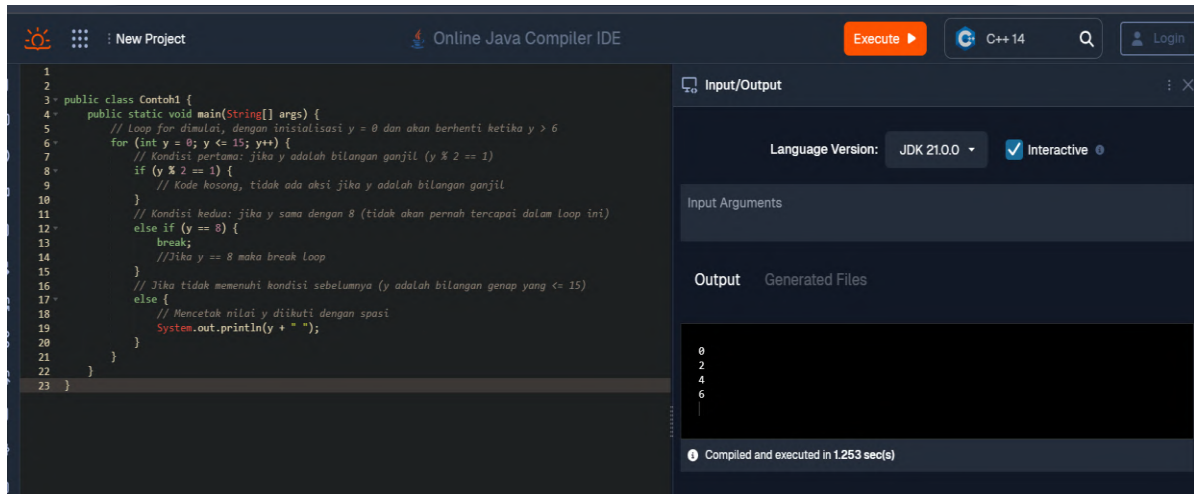
- 1.1
 1. Mulai
 2. Inisialisasi variabel y dengan nilai 0
 3. Cek variabel y apakah memenuhi kondisi $y \leq 15$:
 - Jika y adalah bilangan ganjil (kondisi: $y \% 2 == 1$), maka abaikan lanjutkan
 - Jika y sama dengan 8, maka hentikan loop (break)
 - Jika y adalah bilangan genap bukan ganjil dan bukan 8, cetak nilai y
 4. Tambahkan nilai y dengan 1.
 5. Ulangi langkah 3 hingga 4 sampai kondisi perulangan terpenuhi.
 6. Selesai
- 1.2
 1. Mulai
 2. Inisialisasi variabel i dengan nilai 1.
 3. Mulai loop luar dengan kondisi $i < 5$:
 4. Inisialisasi variabel j dengan nilai 1.
 5. Mulai loop dalam dengan kondisi $j < 3$:
 6. Cetak nilai i dan j.
 7. Jika kondisi $i == 2$ dan $j == 2$ terpenuhi:
 - Hentikan loop luar dan keluar dari semua perulangan (menggunakan break pada label pertama).
 - Jika kondisi tidak terpenuhi, tambahkan nilai j dengan 1.
 8. Setelah loop dalam selesai, tambahkan nilai i dengan 1.
 9. Ulangi langkah 3 hingga kondisi loop luar tidak terpenuhi (yaitu ketika $i \geq 5$ atau perulangan dihentikan dengan break).
 10. Selesai
- 1.3
 1. Mulai
 2. Membuat objek Scanner untuk membaca input dari pengguna.
 3. Minta pengguna memasukkan nilai tinggi piramida, dan simpan nilai tersebut dalam variabel tinggi.
 4. Loop luar:
 - a. Inisialisasi variabel t dengan nilai 1, yang mewakili baris pertama dari piramida.
 - b. Lakukan perulangan selama t kurang dari atau sama dengan tinggi
 - c. Loop untuk mencetak spasi:
 - i. Inisialisasi variabel s dengan nilai 1.
 - ii. Lakukan perulangan sebanyak tinggi - t kali untuk mencetak spasi, yang bertujuan untuk menyeimbangkan posisi bintang agar membentuk piramida.
 - d. Loop untuk mencetak bintang:
 - i. Inisialisasi variabel b dengan nilai 1.
 - ii. Lakukan perulangan sebanyak $2 * t - 1$ kali untuk mencetak bintang, yang bertujuan mencetak piramida dengan jumlah bintang bertambah di setiap baris.
 - e. Setelah mencetak bintang, pindahkan kursor ke baris berikutnya.
 - f. Tambahkan t dengan 1 untuk memproses baris piramida selanjutnya.

5. Ulangi langkah 4 hingga semua baris piramida dicetak.
6. Selesai

2) Kode program, luaran, dan Flowchart

- a) Screenshot/ Capture potongan kode dan hasil luaran
Beri komentar pada kode yang di Screenshot

1.1



```
1 public class Contoh1 {
2     public static void main(String[] args) {
3         // Loop for dimulai, dengan inisialisasi y = 0 dan akan berhenti ketika y > 6
4         for (int y = 0; y <= 15; y++) {
5             // Kondisi pertama: jika y adalah bilangan ganjil (y % 2 == 1)
6             if (y % 2 == 1) {
7                 // Kode kosong, tidak ada aksi jika y adalah bilangan ganjil
8             }
9             // Kondisi kedua: jika y sama dengan 8 (tidak akan pernah tercapai dalam Loop ini)
10            else if (y == 8) {
11                break;
12            }
13            // Jika y == 8 maka break loop
14            // Jika tidak memenuhi kondisi sebelumnya (y adalah bilangan genap yang <= 15)
15            else {
16                // Mencetak nilai y diikuti dengan spasi
17                System.out.println(y + " ");
18            }
19        }
20    }
21 }
```

Input/Output

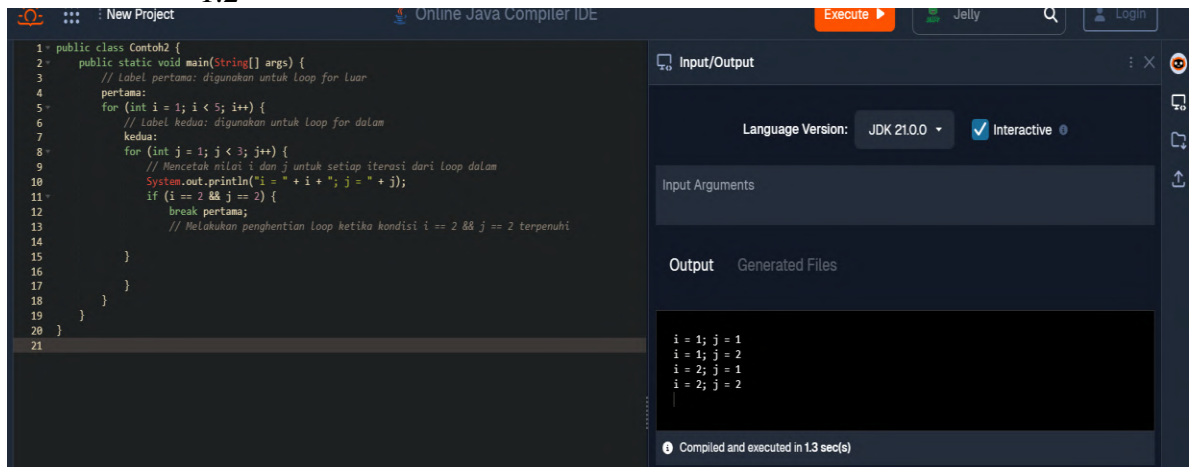
Language Version: JDK 21.0.0 Interactive

Output

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

Compiled and executed in 1.253 sec(s)

1.2



```
1 public class Contoh2 {
2     public static void main(String[] args) {
3         // Label pertama: digunakan untuk Loop for Luar
4         pertama:
5         for (int i = 1; i < 5; i++) {
6             // Label kedua: digunakan untuk Loop for dalam
7             kedua:
8             for (int j = 1; j < 3; j++) {
9                 // Mencetak nilai i dan j untuk setiap iterasi dari Loop dalam
10                System.out.println("i = " + i + "; j = " + j);
11                if (i == 2 && j == 2) {
12                    break pertama;
13                }
14            }
15        }
16    }
17 }
```

Input/Output

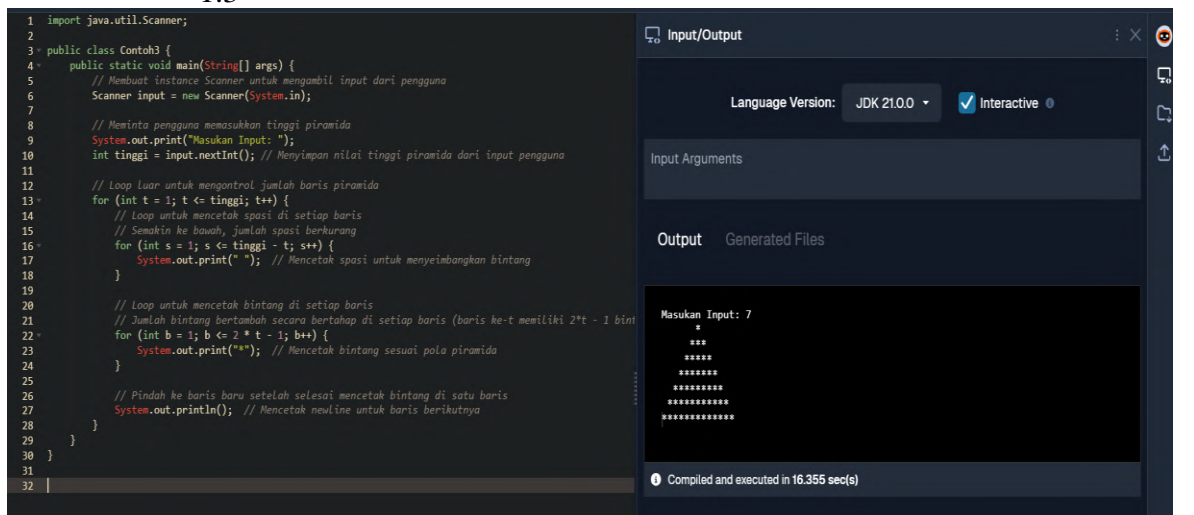
Language Version: JDK 21.0.0 Interactive

Output

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

Compiled and executed in 1.3 sec(s)

1.3



```
1 import java.util.Scanner;
2
3 public class Contoh3 {
4     public static void main(String[] args) {
5         // Membuat instance Scanner untuk mengambil input dari pengguna
6         Scanner input = new Scanner(System.in);
7
8         // Meminta pengguna memasukkan tinggi piramida
9         System.out.print("Masukan Input: ");
10        int tinggi = input.nextInt(); // Menyimpan nilai tinggi piramida dari input pengguna
11
12        // Loop Luar untuk mengontrol jumlah baris piramida
13        for (int t = 1; t <= tinggi; t++) {
14            // Loop untuk mencetak spasi di setiap baris
15            // Semakin ke bawah, jumlah spasi berkurang
16            for (int s = 1; s <= tinggi - t; s++) {
17                System.out.print(" "); // Mencetak spasi untuk menyeimbangkan bintang
18            }
19
20            // Loop untuk mencetak bintang di setiap baris
21            // Jumlah bintang bertambah secara bertahap di setiap baris (baris ke-t memiliki 2*t - 1 bintang)
22            for (int b = 1; b <= 2 * t - 1; b++) {
23                System.out.print("**"); // Mencetak bintang sesuai pola piramida
24            }
25
26            // Pindah ke baris baru setelah selesai mencetak bintang di satu baris
27            System.out.println(); // Mencetak newline untuk baris berikutnya
28        }
29    }
30 }
31
32
```

Input/Output

Language Version: JDK 21.0.0 Interactive

Output

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

Compiled and executed in 16.355 sec(s)

b) Analisa luaran yang dihasilkan

- 1.1. Luaran menghasilkan bilangan genap dari 1-15, namun kode membuat perulangan berhenti ketika mencapai $y == 8$.

- 1.2. Luaran menghasilkan

$i = 1; j = 1$

$i = 1; j = 2$

$i = 2; j = 1$

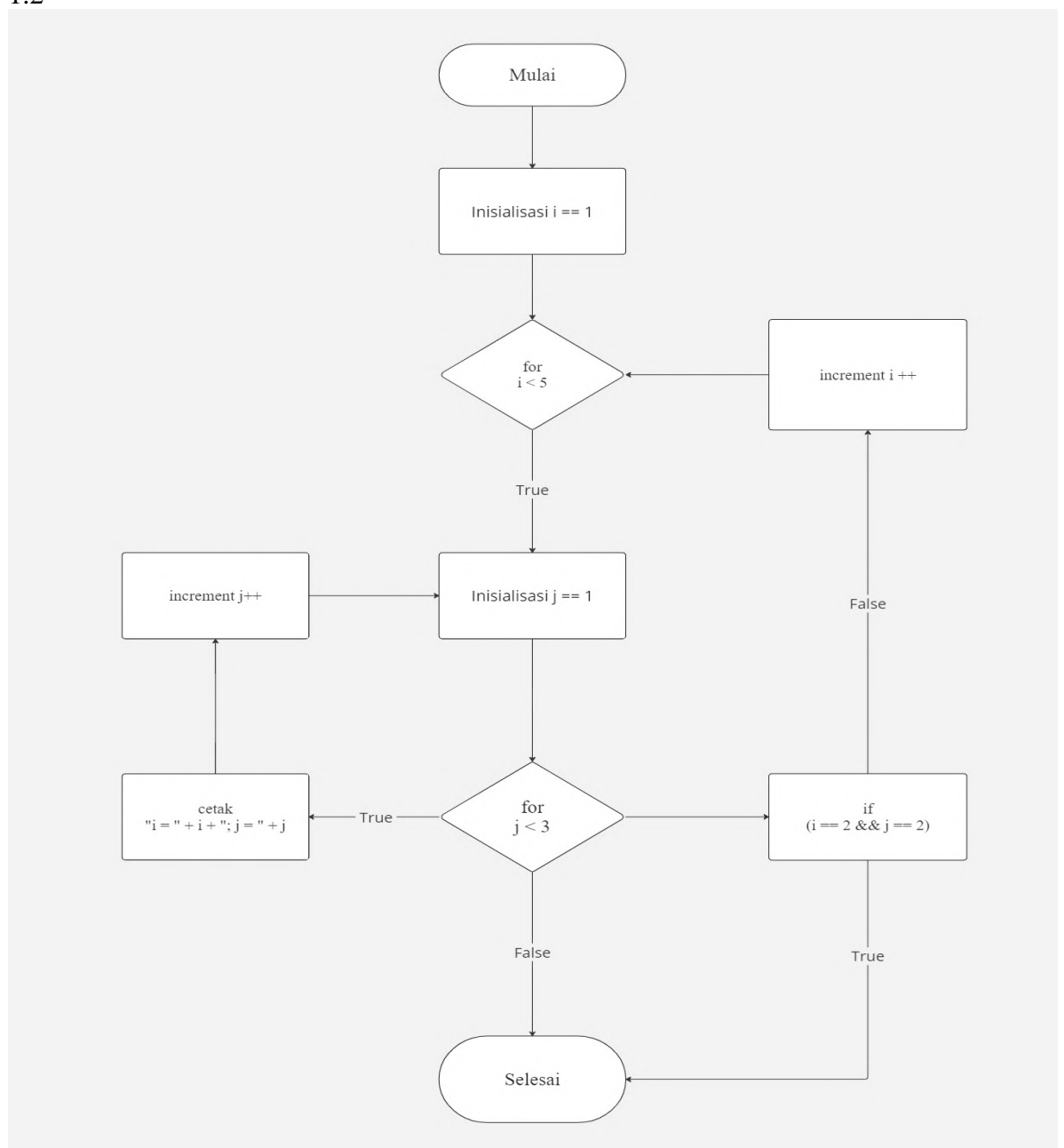
$i = 2; j = 2$

karena kode disusun untuk membatasi nilai $j < 3$, dan meski i disusun < 5 , namun kode disusun untuk break ketika mencapai kondisi $(i == 2 \ \&\& \ j == 2)$.

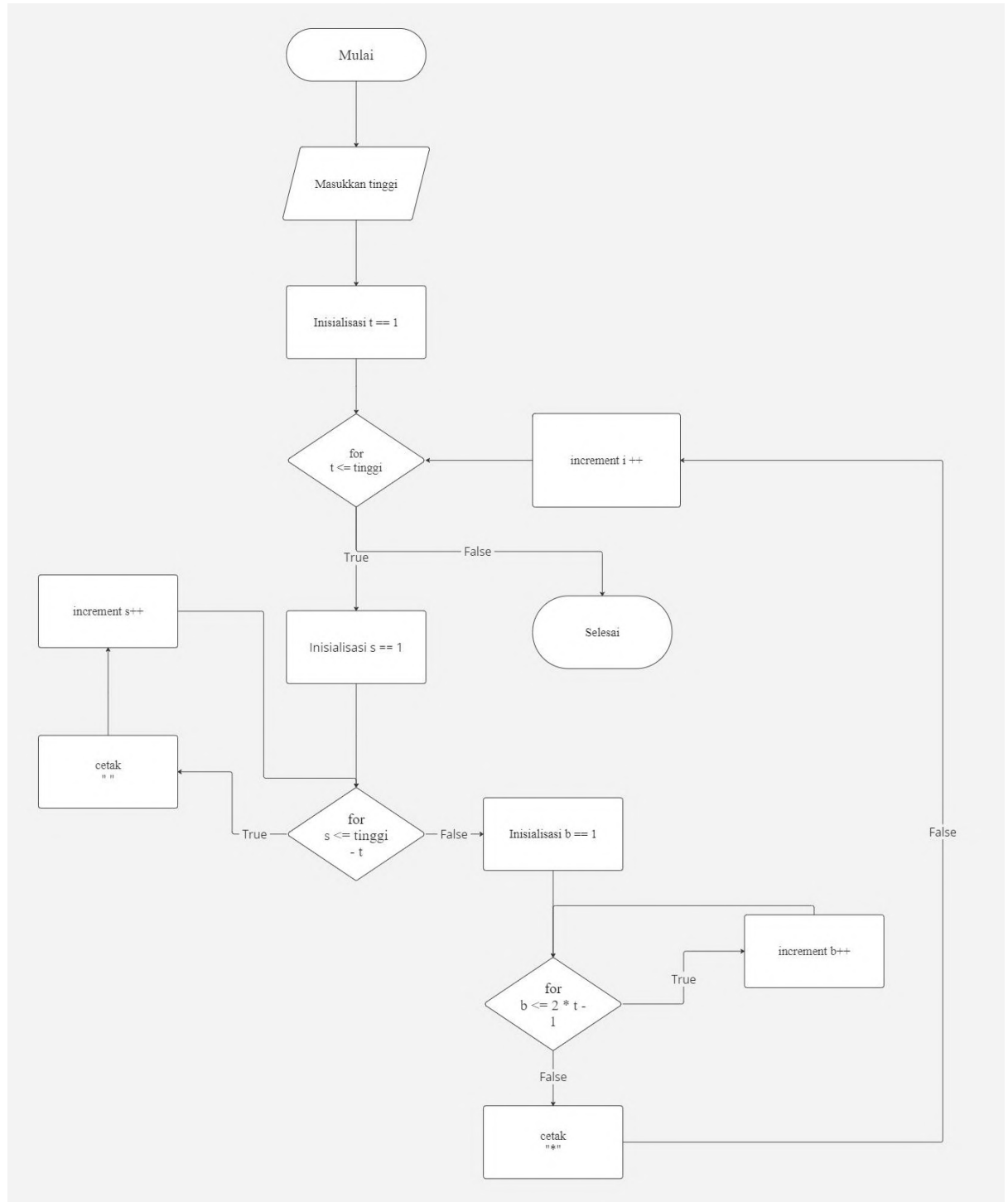
- 1.3 Luaran yang dihasilkan adalah segitiga yang disusun oleh Bintang-bintang dengan tinggi 7 baris

3) Flowchart

- 1.2



- 1.3



[No.1] Kesimpulan

1) Kreasi

Contoh 1 menggunakan perulangan for untuk mencetak bilangan genap dari 0 hingga 15 namun memberhentikan operasi dengan break ketika mencapai angka 8, dengan logika continue untuk melewati bilangan ganjil. Contoh 2 memanfaatkan dua loop bersarang dengan label, mencetak $i < 5$ dan $j < 3$ di mana break digunakan untuk menghentikan perulangan ketika mencapai kondisi tertentu. Contoh 3 mencetak pola piramida dengan loop bersarang yang menyesuaikan jumlah spasi dan bintang berdasarkan input pengguna.

[No. 2] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

Latihan 2

- 2.1. Ubahlah baris kode pada 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
        }
    }
}
```

Luaran:

1
2
3

//Ubah1 menjadi if(i % 3 == 0){ ◇ running, periksa hasilnya

//Ubah2 menjadi continue; ◇ running, periksa hasilnya

Evaluasi perbandingan luaran sebelum dan setelah diubah! Simpulkan maksud dari perubahan tersebut!

- 2.2. Cermati Contoh 5.

```
import java.util.Scanner;

public class ForBersarang {
    public static void main(String[] args) {
        Scanner dataKata = new Scanner(System.in);
        System.out.print("Masukkan Kata yang ingin diulang : ");
        String kata = dataKata.nextLine();

        Scanner dataJumlah = new Scanner(System.in);
        System.out.print("Masukkan Jumlah ingin diulang : ");
        int jumlah = dataJumlah.nextInt();

        int i = 0; //Inisialisasi batas dasar
        while(i < jumlah){
            System.out.println(kata);
            i++; //Faktor pengulang Increment
        }
    }
}
```

Luaran Contoh 5:

Masukkan Kata yang ingin diulang : Fakultas Teknik

Masukkan Jumlah ingin diulang : 5

Fakultas Teknik

Fakultas Teknik

Periksa luaran, bila ketika di eksekusi, jumlah yang diulang = 0!
Evaluasi luaran, bila kode diubah menjadi do ... while dengan masukan sama jumlah yang diulang = 0.
Simpulkan perbedaan while dan do ... while!

2.3. Bila diketahui pernyataan pseudocode berikut:

- [1] inisiasi idPelajaran
- [2] inisiasi nilai pelajaran
- [3] inisiasi nilai rata-rata
- [4] Minta pengguna untuk menuliskan jumlah pelajaran
- [5] Ketika idPelajaran lebih kecil dari jumlah pelajaran
- [6] Minta pengguna untuk menuliskan nilai pelajaran
- [7] Hitung nilai rata-rata = (nilai pelajaran + nilai rata-rata) / 2
- [8] Tambah satu ke idPelajaran
- [9] Tampilkan nilai rata-rata

Rekomendasikan kode untuk menyelesaikan Pseudocode tersebut!

2.4. Rancang diagram flowchart dari Latihan 2.1, Latihan 2.2, dan Latihan 2.3!

[No.2] Analisis dan Argumentasi

- 1) Contoh 4 mengubah kode dan melakukan analisa pada setia perubahan yang dilakukan pada kode program yang dilakukan.
- 2) Contoh 5 melakukan pemeriksaan kode dan evaluasi pada program dengan kondisi tertentu.

[No.2] Penyusunan Algoritma, Kode Program, dan Flowchart

- 1) Algoritma
 - Contoh 4
 2. Mulai
 3. Inisialisasi variabel i dengan nilai 1.
 4. Loop While:
 5. Lakukan perulangan selama i kurang dari atau sama dengan 6:
 - a. Cetak nilai i ke layar.
 - b. Tambahkan i dengan 1.
 - c. Jika i adalah kelipatan 3 ($i \% 3 == 0$): Lompat iterasi saat ini dan kembali ke awal loop (menggunakan continue).
 6. Ulangi langkah 3 hingga kondisi loop tidak terpenuhi (ketika i lebih dari 6).
 7. Selesai
 - Contoh 5
 1. Mulai
 2. Inisialisasi objek Scanner untuk menerima input dari pengguna.
 3. Tampilkan pesan "Masukkan Kata yang ingin diulang".
 4. Ambil input kata dari pengguna dan simpan dalam variabel kata.
 5. Tampilkan pesan "Masukkan Jumlah ingin diulang".
 6. Ambil input jumlah pengulangan dari pengguna dan simpan dalam variabel jumlah.

7. Inisialisasi variabel i dengan nilai 0 (sebagai batas awal pengulangan).
8. Ulangi.
9. Selesai

2) Kode program dan luaran

a) Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

Contoh 4, ubah 1

```
1 public class Contoh4 {
2     public static void main(String[] args) {
3         int i = 1; // Inisialisasi variabel i dengan nilai awal 1
4
5         // Memulai loop while selama i kurang dari atau sama dengan 6
6         while (i <= 6) {
7             System.out.println(i); // Mencetak nilai i ke layar
8             i++; // Meningkatkan nilai i sebesar 1
9
10            // Jika i adalah kelipatan 3 (yaitu i % 3 == 0), hentikan loop
11            if (i % 3 == 0) {
12                break; // Menghentikan perulangan saat i adalah kelipatan 3
13            }
14        }
15    }
16 }
```

Input/Output

Language Version: JDK 21.0.0 ☒ Interactive

Input Arguments

Output

```
1
2
|
```

Generated Files

Compiled and executed in 1.242 sec(s)

Contoh 4, ubah 2

```
1 public class ContohWhile {
2     public static void main(String[] args) {
3         int i = 1; // Inisialisasi variabel i dengan nilai awal 1
4
5         // Memulai loop while selama i kurang dari atau sama dengan 6
6         while (i <= 6) {
7             if (i % 3 == 0) { // Jika i adalah kelipatan 3, lompat ke iterasi berikutnya
8                 i++; // Tingkatkan nilai i terlebih dahulu
9                 continue; // Melompati pencetakan nilai i ketika i adalah kelipatan 3
10            }
11            System.out.println(i); // Mencetak nilai i ke layar
12            i++; // Meningkatkan nilai i sebesar 1
13        }
14    }
15 }
16 }
```

Input/Output

Language Version: JDK 21.0.0 ☒ Interactive

Input Arguments

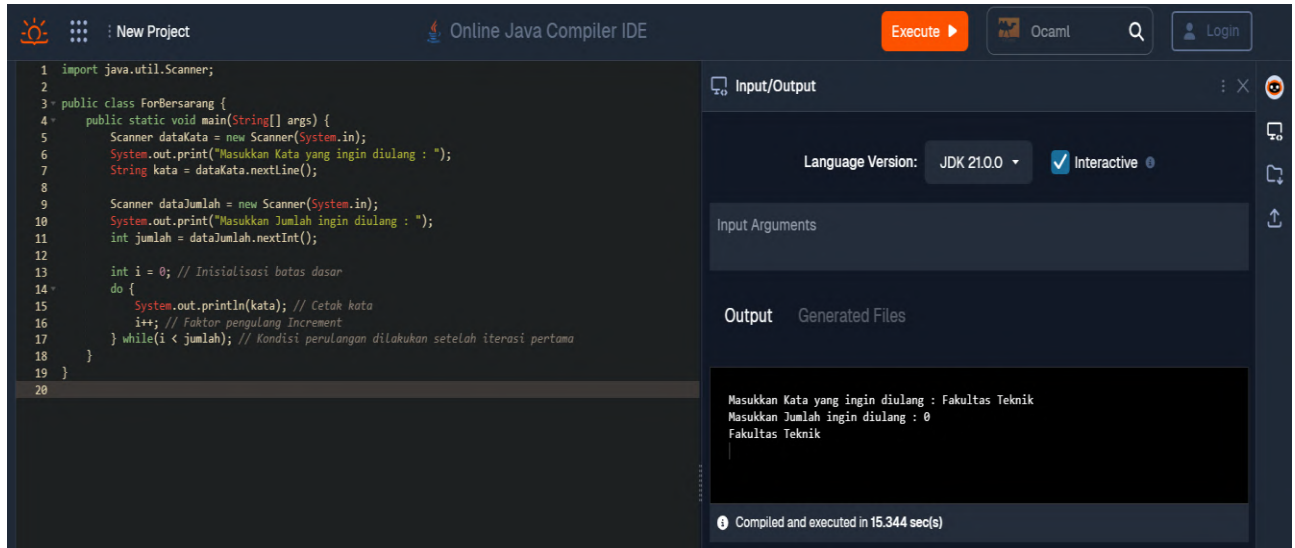
Output

```
1
2
4
5
|
```

Generated Files

Compiled and executed in 1.424 sec(s)

Contoh 5



```
1 import java.util.Scanner;
2
3 public class ForBersarang {
4     public static void main(String[] args) {
5         Scanner dataKata = new Scanner(System.in);
6         System.out.print("Masukkan Kata yang ingin diulang : ");
7         String kata = dataKata.nextLine();
8
9         Scanner dataJumlah = new Scanner(System.in);
10        System.out.print("Masukkan Jumlah ingin diulang : ");
11        int jumlah = dataJumlah.nextInt();
12
13        int i = 0; // Inisialisasi batas dasar
14        do {
15            System.out.println(kata); // Cetak kata
16            i++; // Faktor pengulang Increment
17        } while(i < jumlah); // Kondisi perulangan dilakukan setelah iterasi pertama
18    }
19 }
20
```

Input/Output

Language Version: JDK 21.0.0 ☒ Interactive

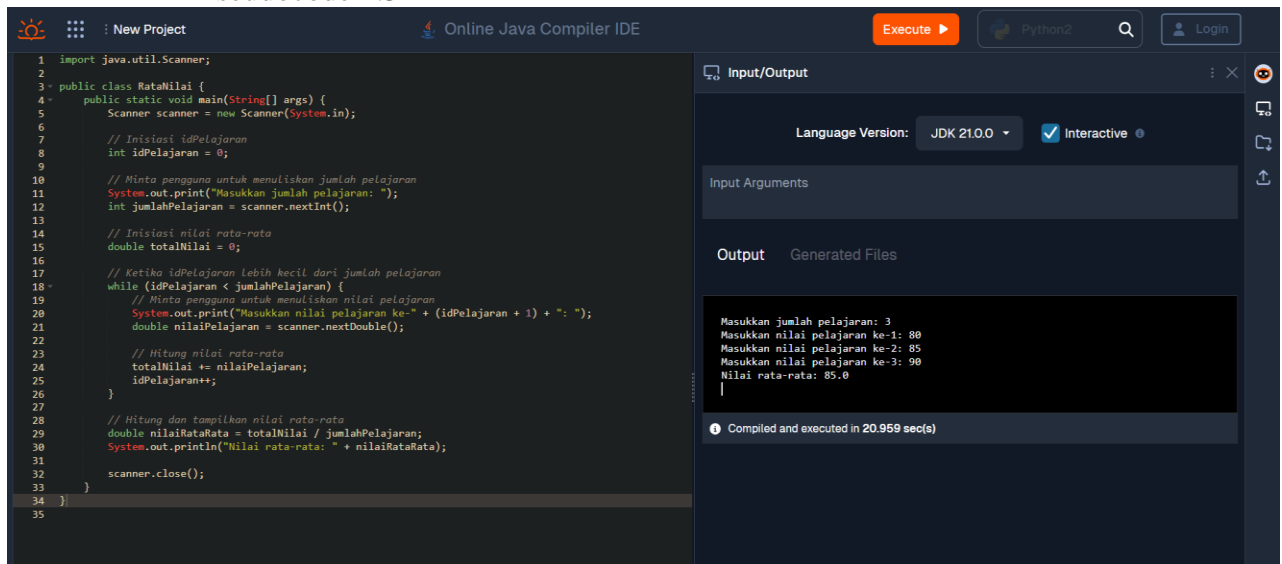
Input Arguments

Output Generated Files

Masukkan Kata yang ingin diulang : Fakultas Teknik
Masukkan Jumlah ingin diulang : 0
Fakultas Teknik

Compiled and executed in 15.344 sec(s)

Pseudocode 2.3



```
1 import java.util.Scanner;
2
3 public class RataNilai {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7         // Inisiasi idPelajaran
8         int idPelajaran = 0;
9
10        // Minta pengguna untuk menuliskan jumlah pelajaran
11        System.out.print("Masukkan jumlah pelajaran: ");
12        int jumlahPelajaran = scanner.nextInt();
13
14        // Inisiasi nilai rata-rata
15        double totalNilai = 0;
16
17        // Ketika idPelajaran Lebih kecil dari jumlah pelajaran
18        while (idPelajaran < jumlahPelajaran) {
19            // Minta pengguna untuk menuliskan nilai pelajaran
20            System.out.print("Masukkan nilai pelajaran ke-" + (idPelajaran + 1) + ": ");
21            double nilaiPelajaran = scanner.nextDouble();
22
23            // Hitung nilai rata-rata
24            totalNilai += nilaiPelajaran;
25            idPelajaran++;
26        }
27
28        // Hitung dan tampilkan nilai rata-rata
29        double nilaiRataRata = totalNilai / jumlahPelajaran;
30        System.out.println("Nilai rata-rata: " + nilaiRataRata);
31
32        scanner.close();
33    }
34 }
35
```

Input/Output

Language Version: JDK 21.0.0 ☒ Interactive

Input Arguments

Output Generated Files

Masukkan jumlah pelajaran: 3
Masukkan nilai pelajaran ke-1: 80
Masukkan nilai pelajaran ke-2: 85
Masukkan nilai pelajaran ke-3: 90
Nilai rata-rata: 85.0

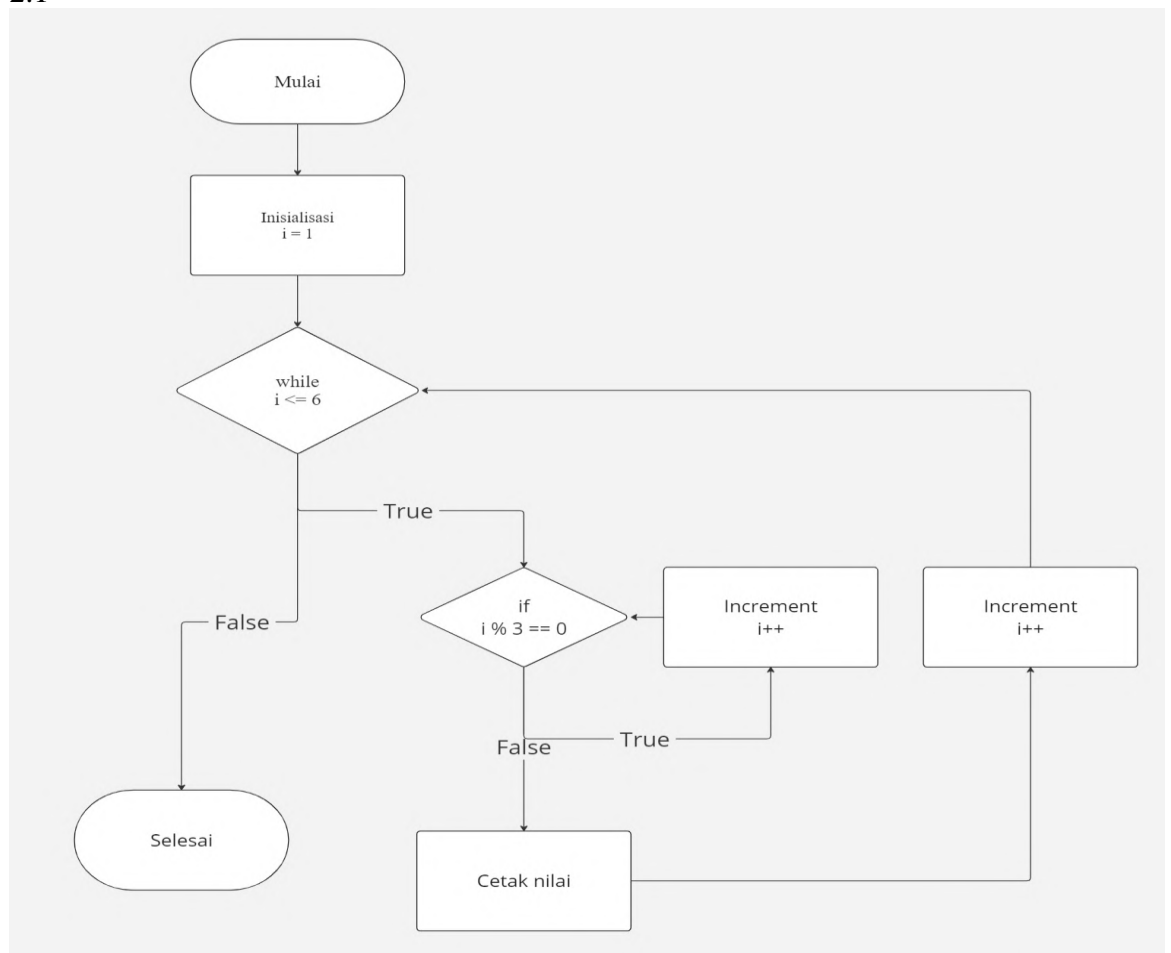
Compiled and executed in 20.959 sec(s)

b) Analisa luaran yang dihasilkan

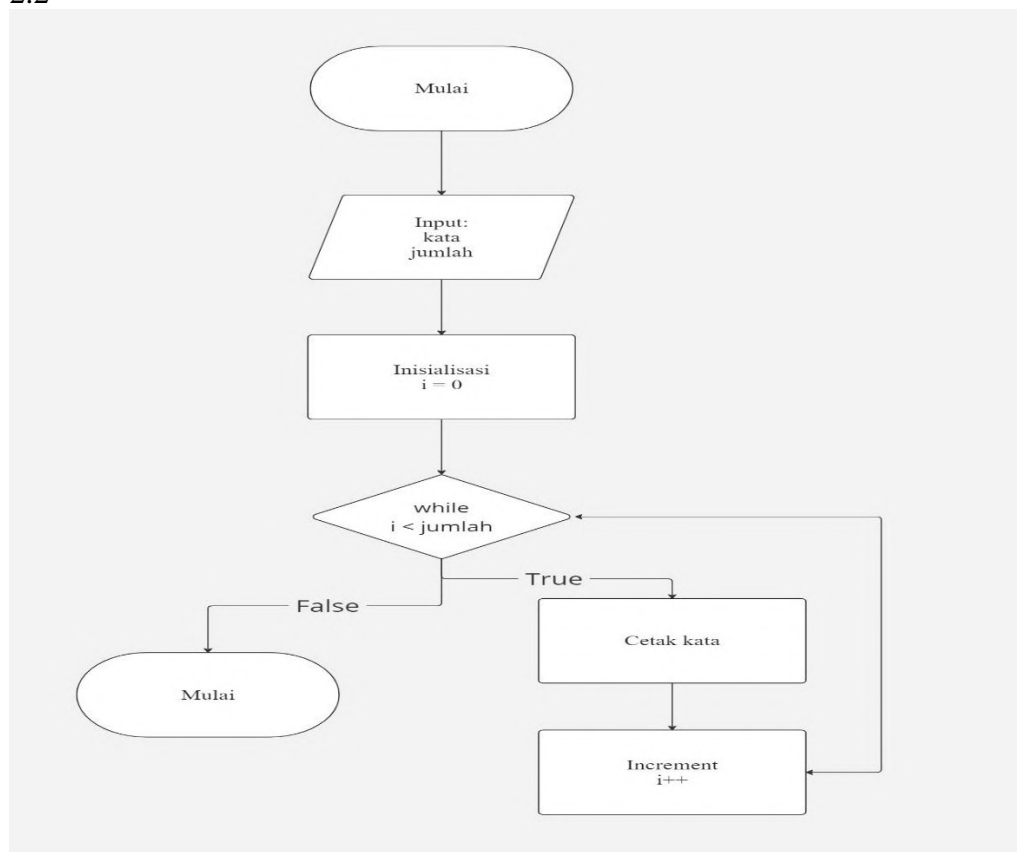
- Contoh 4 pada ubah 1 menghasilkan 1 2 karena kode melakukan break ketika mencapai 3
- Contoh 4 pada ubah 2 menghasilkan 1 2 4 5 karena kode melewati pencetakan ketika kondisi $I \% 3 == 0$ sehingga 3 6 tidak dicetak.
- Contoh 5 jika pengguna memasukkan jumlah pengulangan 0, maka kondisi `while(i < jumlah)` akan langsung false pada iterasi pertama, sehingga tidak ada output yang dihasilkan.
- Contoh 5 jika pengguna memasukkan jumlah pengulangan 0, maka dengan `do-while`, kode di dalam blok perulangan akan tetap dijalankan sekali sebelum memeriksa kondisi. Sehingga, kata akan dicetak satu kali meskipun jumlah pengulangan 0.
- Latihan 2.3 menghasilkan luara jumlah pelajran dan nilai peljaran sesuai dengan input pengguna dan menghitung rata-rata darinya.

3) Flowchart

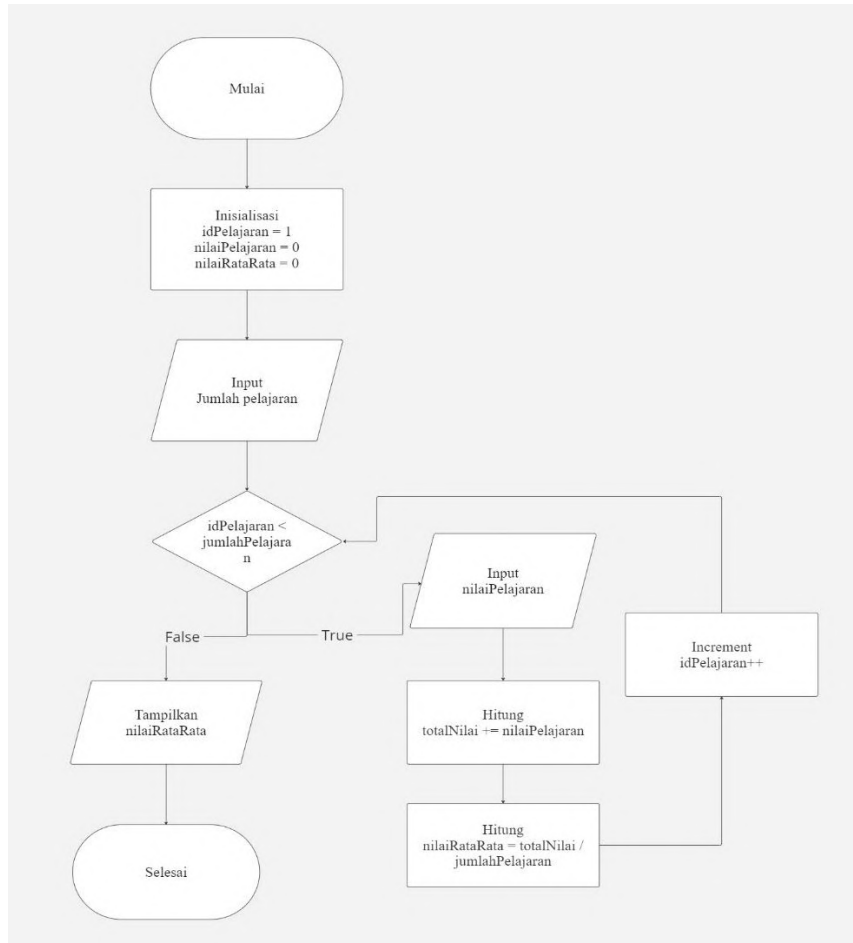
- 2.1



- 2.2



- 2.3



[No.2] Kesimpulan

1) Kreasi

- Pada Contoh 4, penggunaan while loop dengan kondisi $i \% 3 == 0$ dan perintah continue membuat program melewati pencetakan nilai i saat i merupakan kelipatan 3. Hal ini menyebabkan angka 3 dan 6 tidak tercetak, karena loop melompati iterasi tersebut. Contoh 5, perbedaan antara while dan do-while diperlihatkan ketika jumlah pengulangan diatur ke 0. Pada while, jika jumlah 0, tidak ada output karena kondisi diperiksa sebelum perulangan dimulai. Sebaliknya, dengan do-while, meskipun jumlah 0, kata tetap dicetak sekali karena kondisi diperiksa setelah iterasi pertama. Ini menunjukkan bahwa do-while selalu menjalankan setidaknya satu iterasi.
- Perbedaan while dan do...while adalah:
While tidak mencetak kondisi false sedangkan do...while mencetak sekali sebelum memeriksa kondisi.

Refleksi

Pada materi for dan while saya mendapatkan banyak pengetahuan baru, tentang bagaimana cara menyusun kode untuk program dengan tipe for dan while, saya juga menemukan banyak kesulitan sebab materi ini membutuhkan banyak focus dan ketelitian tinggi juga logika yang sulit. Saya melakukan banyak pencarian untuk menemukan cara mengerjakan tugas ini. Penyusunan flowchart juga cukup sulit bagi saya, karena pemahaman yang masih terbilang rendah. Saya juga ingin berterima kasih kepada dosen atas semua pengajaran yang diberikan, dan saya meminta maaf atas semua kekurangan yang ada pada penyusunan tugas saya.