

Nama & NPM	Topik:	Tanggal:
Bagas Satrio Winata G1F024059	For dan While	8 oktober 2024

[No.1] Identifikasi masalah

Contoh 1:

```
public class ContohFor{
public static void main(String[] args) {
    for (double y <= 15; y = 0; y++) {
        if (y % 2 == 1) {    //kondisi 1
            // baris kode kosong 1
        } else if (y == 8) {    //kondisi 2
            // baris kode kosong 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) {
            // kode yang hilang
        } } } }
```

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
        }
    }
}
```

[No.1] Analisis dan Argumentasi

1) Saya mengusulkan permasalahan ini:

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!

2) Saya mengusulkan permasalahan ini dapat diselesaikan dengan cara :

1.1. Kesalahan Syntax error ini dikarenakan penulisan for loop terbalik. Pada bagian for (double y <= 15; y = 0; y++), inisialisasi seharusnya berada di awal, bukan di akhir.

Seharusnya kode yang benar adalah: for (double y = 0; y <= 15; y++).

Perbaiki kode program pada System.out.print(""); atau bisa dikosongkan untuk tidak melakukan apa pun saat kondisi terpenuhi. Pada metode break; untuk menghentikan loop ketika y == 8.

1.2. Untuk menghasilkan luaran yang diminta, dapat dilakukan dengan cara menambahkan break pada loop kedua (kedua:) ketika i == 2, sehingga loop kedua akan keluar setelah dua iterasi terjadi.

1.3. Untuk menghasilkan sebuah piramida yang simetris, dapat dilakukan dengan cara menambahkan metode for untuk spasi di depan setiap baris sehingga bintang dapat membentuk piramida yang simetris.

1.4. Analisis Flowchart:

Latihan 1.2:

Flowchart:

- Mulai
- Loop pertama: Inisialisasi i = 1
- Loop kedua: Inisialisasi j = 1, cetak i = x; j = y
- Jika i == 2, keluarkan break untuk keluar dari loop pertama
- Iterasi berlanjut sampai kondisi i < 5 terpenuhi.
- Selesai

Latihan 1.3:

Flowchart:

- Mulai

- Ambil input tinggi
- Loop pertama: Inisialisasi variabel t untuk tinggi piramida
- Loop kedua: Inisialisasi variabel s untuk menghitung spasi
- Loop ketiga: Inisialisasi variabel b untuk mencetak Bintang
- Cetak bintang dan spasi sesuai dengan jumlah yang dihitung
- Ulangi hingga piramida terbentuk
- Selesai

[No.1] Algoritma dan kode program

1) Algoritma

Contoh 1:

- Mulai program
- Mendeklarasikan public class contohfor
- Mendeklarasikan public static void main(String []args) didalam public class contohfor
- Mendeklarasikan metode if else didalam public static void main(String []args)
- Mendeklarasikan System.out.println didalam public static void main(String []args)
- Program selesai

Contoh 2:

- Mulai program
- Mendeklarasikan public class forbersarang
- Mendeklarasikan public static void main(String []args) didalam public class forbersarang
- Mendeklarasikan metode for dan if didalam public static void main(String []args)
- Program selesai

Contoh 3:

- Mulai program
- Mendeklarasikan import java.util.Scanner
- Mendeklarasikan public class forbersarang
- Mendeklarasikan public statis void main(String []args) didalam public class forbersarang
- Mendeklarasikan new Scanner(Syste m.in) didalam public statis void main(String []args)
- Mendeklarasikan variabel int didalam public statis void main(String []args)
- Medeklarasikan metode for didalam public statis void main(String []args)
- Mendeklarasikan System.out.println didalam public statis void main(String []args)
- Program selesai

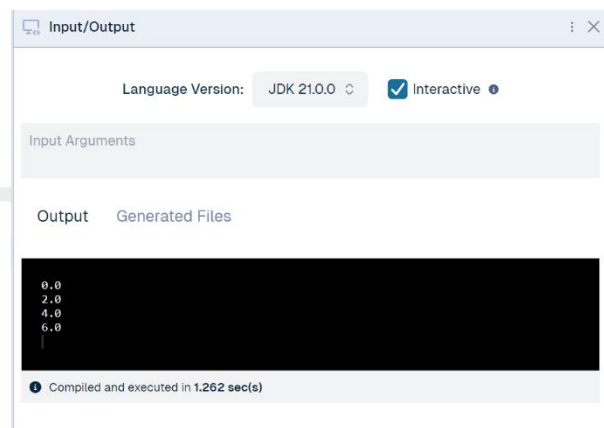
2) Kode program:

Contoh 1:

```

1 public class Contonfor {
2     public static void main(String[] args) {
3         for (double y = 0; y <= 15; y++) {
4             if (y % 2 == 1) { //kondisi 1
5                 // tidak melakukan apa-apa, atau bisa kosongkan
6             } else if (y == 8) { //kondisi 2
7                 break; // menghentikan loop saat y = 8
8             } else {
9                 System.out.println(y + " ");
10            }
11        }
12    }
13 }
14

```

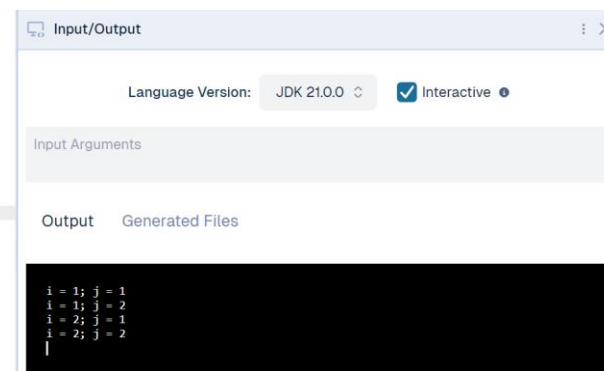


Contoh 2:

```

1 public class ForBersarang {
2     public static void main(String[] args) {
3         pertama:
4         for(int i = 1; i < 5; i++) {
5             kedua:
6             for(int j = 1; j < 3; j++) {
7                 System.out.println("i = " + i + "; j = " + j);
8             }
9             if(i == 2) {
10                 break; // Keluar dari loop pertama saat i = 2
11             }
12         }
13     }
14 }
15

```

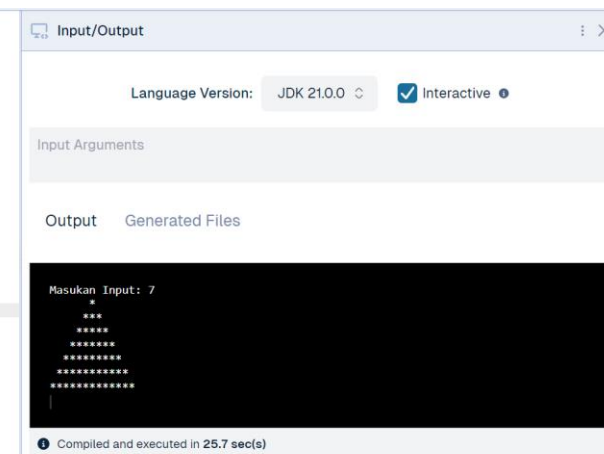


Contoh 3:

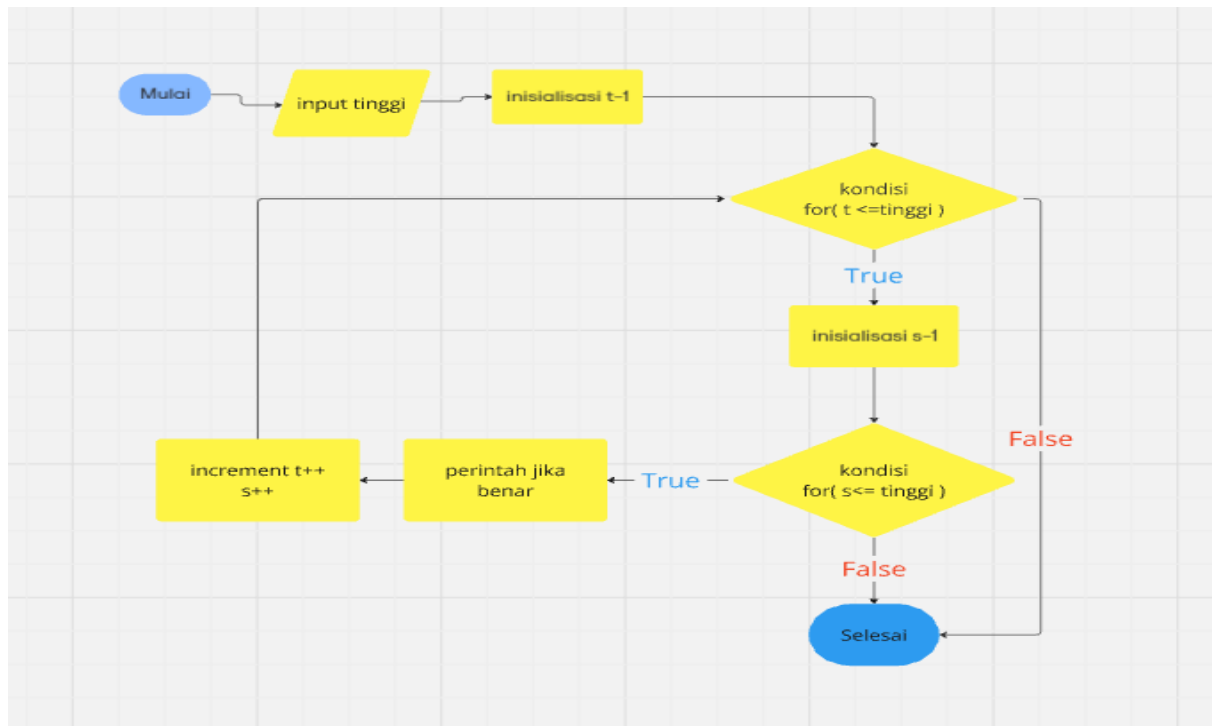
```

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         for(int t=1; t<=tinggi; t++){
10             for(int s=tinggi; s>=t; s--) {
11                 // Menghitung jumlah spasi per baris
12                 System.out.print(" ");
13             }
14             for(int b=1; b<=2*t-1; b++){
15                 // Menghitung jumlah bintang per baris
16                 System.out.print("*");
17             }
18             System.out.println(); // Membuat baris baru
19         }
20     }
21 }
22

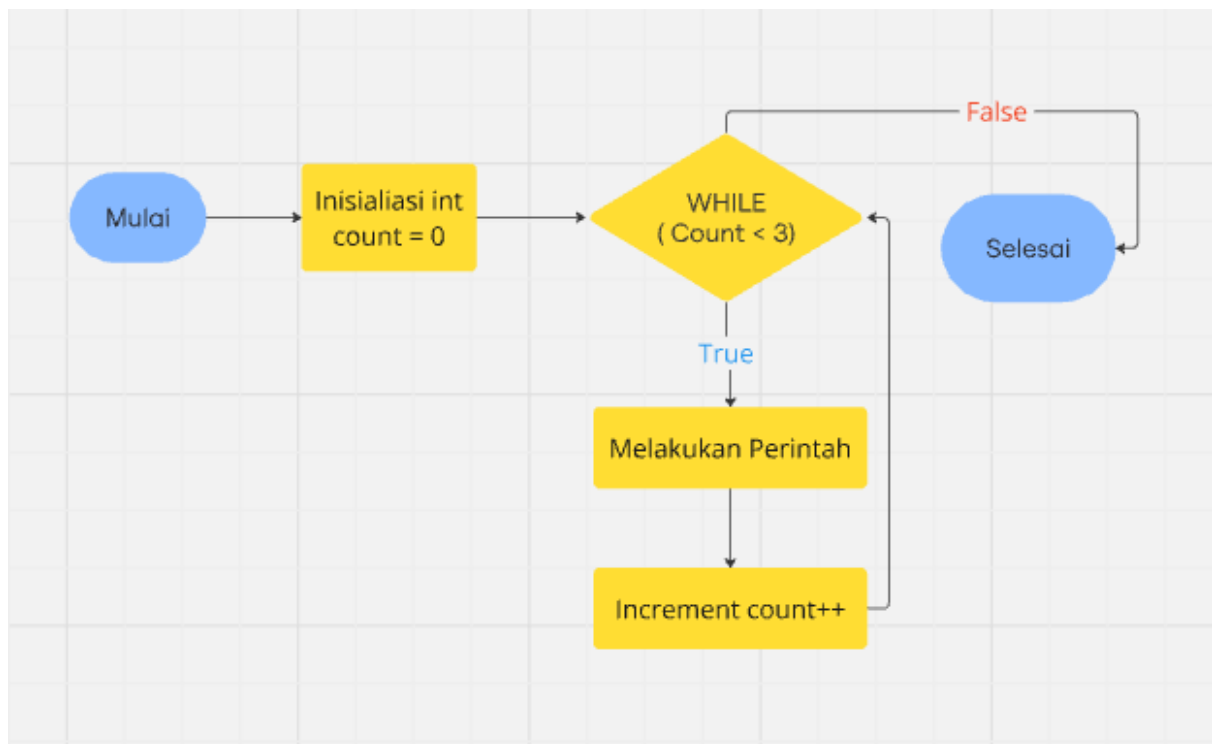
```



3) Flowchart
Flowchart 1.2:



Flowchart 1.3:



[No.1] Kesimpulan

Pada program ini menggunakan beberapa bentuk kelas publik karena Java yang mengharuskan setiap variabel sumber utama memiliki kelas publik dengan nama yang sama dengan nama variabel. Ini

adalah aturan Java untuk menjaga keteraturan dalam struktur kode program. Selain itu, penggunaan kelas publik memungkinkan akses eksternal ke elemen dalam kelas tersebut, yang biasanya memungkinkan Anda mengakses elemen dalam program Anda. Perbaiki program dengan menambahkan inisialisasi yang benar pada perulangan for. Hal ini karena struktur Java memerlukan inisialisasi, kondisi, dan iterasi untuk ditulis dengan benar dan dalam urutan yang benar dalam satu baris. Artinya, inisialisasi variabel terlebih dahulu, lalu periksa kondisinya, dan terakhir modifikasi variabelnya. Tanpa inisialisasi, loop tidak dapat berfungsi dengan baik dan akan menghasilkan kesalahan atau keluaran yang salah.

[No.2] Identifikasi Masalah

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 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
        }
    }
}
```

[No.2] Analisis dan Argumentasi

1) Saya mengusulkan permasalahan ini:

2.1. Ubahlah baris kode pada Contoh 4 //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. 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!

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

2) Saya mengusulkan permasalahan ini dapat diselesaikan dengan cara:

2.1 modifiikasi contoh 4

- Ubah `if(i == 4)` menjadi `if(i % 3 == 0)` untuk menghentikan loop pada kelipatan 3.
- Ubah `break` menjadi `continue` agar perulangan tetap berjalan dengan melewati angka
- Sebelum diubah: perulangan berhenti di angka 3 karena `break`.
- Setelah diubah: perulangan melewati angka kelipatan 3, namun tetap berjalan hingga akhir.

2.2 Evaluasi Contoh 5 dengan jumlah 0

Jika jumlah yang diulang adalah 0, maka dengan `while`, kode tidak akan mencetak apa pun. Namun, jika kita menggunakan `do-while`, kode akan tetap mencetak minimal satu kali meskipun jumlahnya 0. Perbedaan `while` memeriksa kondisi awal sedangkan, `do while` menjalankan perintah setidaknya sekali sebelum memeriksa

2.3 Rekomendasi kode program:

```
import java.util.Scanner;

public class RataRataPelajaran {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in)

        int idPelajaran = 0;

        double nilaiRataRata = 0;

        System.out.print("Masukkan jumlah pelajaran: ");

        int jumlahPelajaran = input.nextInt();
```

```

while (idPelajaran < jumlahPelajaran) {

    System.out.print("Masukkan nilai pelajaran: ");

    double nilaiPelajaran = input.nextDouble();

    // Menghitung nilai rata-rata

    nilaiRataRata = (nilaiPelajaran + nilaiRataRata) / 2

    // Menambah idPelajaran

    idPelajaran++;

}

// Menampilkan nilai rata-rata

System.out.println("Nilai rata-rata: " + nilaiRataRata);

}

```

2.4 flowachart dilampirkan pada gambar

[No.2] Algoritma dan Kode Program

1) Algoritma

Contoh 4:

- Mulai program
- Mendeklarasikan import java.util.Scanner;
- Mendeklarasikan public class ForBersarang
- Mendeklarasikan public static void main(String []args) didalam public class ForBersarang
- Mendeklarasikan new Scanner didalam public static void main(String []args)
- Mendeklarasikan system.out.print didalam public static void main(String []args)
- Mendeklarasikan variabel String dan int didalam public static void main(String []args)
- Mendeklarasikan metode do while didalam public static void main(String []args)
- Program selesai

Contoh 5:

- Mulai program
- Mendeklarasikan import java.util.Scanner;
- Mendeklarasikan public class NilaiRataRata
- Mendeklarasikan public static void main(String []args) didalam public class ForBersarang
- Mendeklarasikan new Scanner didalam public static void main(String []args)
- Mendeklarasikan variabel double dan int didalam public static void main(String []args)
- Mendeklarasikan system.out.print didalam public static void main(String []args)

- Mendeklarasikan variabel double dan int didalam public static void main(String []args)
- Mendeklarasikan metode while didalam public static void main(String []args)
- Program selesai
-

2) Kode Program

Contoh 4:

```

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;
14        do {
15            System.out.println(kata);
16            i++;
17        } while(i < jumlah);
18    }
19 }
20

```

Input/Output

Language Version: JDK 21.0.0 ☒ Interactive

Input Arguments

Output Generated Files

```

Masukkan Kata yang ingin diulang : hidupp Teknikkkk
Masukkan Jumlah ingin diulang : 6
hidupp Teknikkkk
hidupp Teknikkkk
hidupp Teknikkkk
hidupp Teknikkkk
hidupp Teknikkkk
hidupp Teknikkkk

```

Compiled and executed in 32.328 sec(s)

Contoh 5:

```

1 import java.util.Scanner;
2
3 public class NilaiRataRata {
4     public static void main(String[] args) {
5         Scanner input = new Scanner(System.in);
6
7         int idPelajaran = 1;
8         double totalNilai = 0;
9         double rataRata = 0;
10
11        System.out.print("Masukkan jumlah pelajaran : ");
12        int jumlahPelajaran = input.nextInt();
13
14        while(idPelajaran <= jumlahPelajaran) {
15            System.out.print("Masukkan nilai pelajaran ke- " + idPelajaran + " : ");
16            double nilaiPelajaran = input.nextDouble();
17            totalRata = (totalRata + nilaiPelajaran) / idPelajaran;
18            idPelajaran++;
19        }
20
21        System.out.println("Nilai rata-rata: " + rataRata);
22    }
23 }
24

```

Input/Output

Language Version: JDK 21.0.0 ☒ Interactive

Input Arguments

Output Generated Files

```

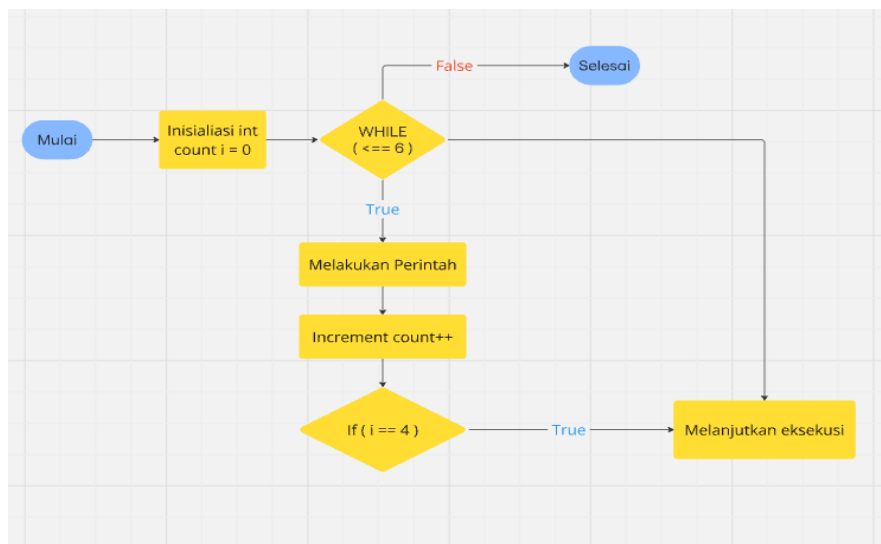
Masukkan jumlah pelajaran: 4
Masukkan nilai pelajaran ke-1: 85
Masukkan nilai pelajaran ke-2: 69
Masukkan nilai pelajaran ke-3: 97
Masukkan nilai pelajaran ke-4: 93
Nilai rata-rata: 112.1875

```

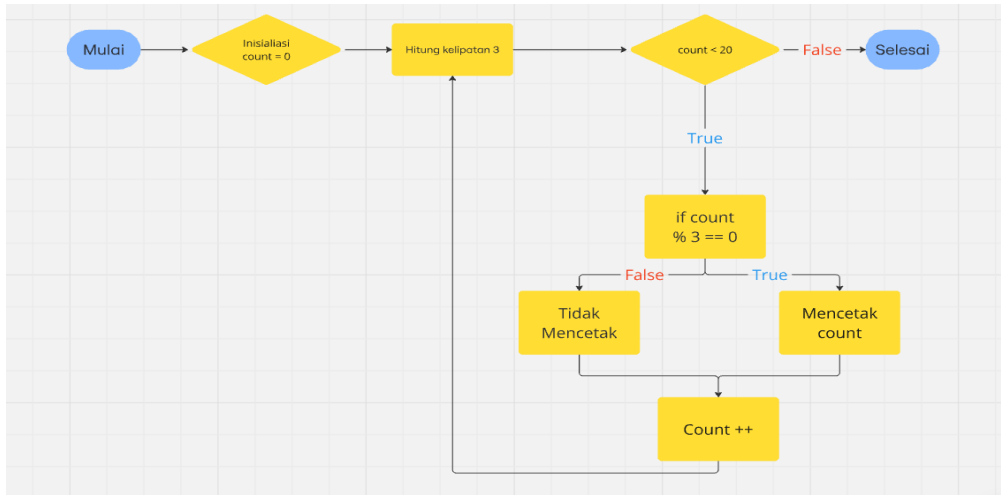
Compiled and executed in 33.323 sec(s)

3) Flowchart

Flowchart 2.1:



Flowchart 2.2:



Flowchart 2.3:

