

Nama & NPM	Topik:	Tanggal:
HANIFAH AZIZAH G1F024037	FOR WHILE JAVA	02/10/2024

[No.1] Identifikasi Masalah:

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 Penyebab kesalahan yang terjadi pada contoh 1 adalah: urutan inisialisasi dan evaluasi kondisi terbalik sehingga menyebabkan error. Untuk menampilkan luaran yang diinginkan maka Pada baris 1 yang kosong perlu ditambahkan fungsi continue, dan pada baris 2 yang kosong perlu ditambahkan fungsi break sehingga luaran yang dihasilkan 0,2,4,6. Dan tipe data yang digunakan harusnya int agar luaran yang dihasilkan tidak decimal.

1.2 Kode yang cocok untuk mengisi bagian yang kosong adalah break karena jika kode yang digunakan adalah continue maka luaran yang dihasilkan akan sampai i = 4.

1.3 Kode terdapat pada bagian algoritma dan kode program.

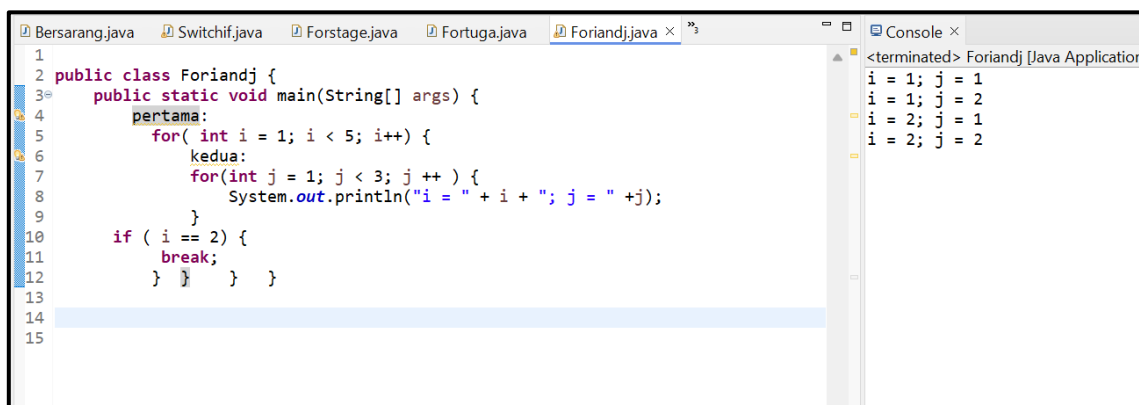
1.4 Diagram flowchart untuk Latihan 1.2 memiliki dua loop untuk dijalankan dengan pengecekan kondisi berulang dan penggunaan kode break agar luaran yang ditampilkan hanya batas i == 2 dan bukan 4 agar sesuai dengan luaran yang diinginkan pada soal.

Diagram flowchart untuk Latihan 1.3 menggunakan loop untuk membuat baris bintang sesuai dengan input dari pengguna. Perulangan program terus dijalankan hingga baris yang ditampilkan sesuai dengan jumlah baris yang diinput pengguna dengan luaran Bintang berurutan dimulai dengan 1 bintang pada baris pertama, dilanjutkan dengan 2 bintang pada baris kedua, dan seterusnya.

[No.1] Penyusunan Algoritma dan Kode Program

Algoritma Latihan 1.2

1. Mulai.
2. Inisialisasi program dengan deklarasi nama kelas.
3. Deklarasi main method.
4. Loop Pertama: Inisialisasi variabel i dengan nilai 1.
5. Cek kondisi $i < 5$. Jika true, lanjutkan ke langkah berikutnya; jika false, keluar dari loop.
6. Loop Kedua: Inisialisasi variabel j dengan nilai 1.
7. Cek kondisi $j < 3$. Jika true, lanjutkan ke langkah berikutnya; jika false, keluar dari loop kedua dan kembali ke loop pertama.
8. Cetak nilai i dan j: Cetak nilai i dan j ke luaran.
9. Setelah mencetak, j di-increment ($j++$). Kembali ke Loop Pertama.
10. Cek nilai i: Setelah Loop Kedua selesai (setelah j mencapai 3), cek apakah $i == 2$.
11. Jika true, lakukan break, yang akan keluar dari Loop Pertama dan akhiri program.
12. Jika false, kembalikan ke cek kondisi apakah $i < 5$.
13. Jika keluar dari Loop Kedua tanpa break, i di-increment ($i++$). kembalikan ke cek kondisi apakah $i < 5$.
14. Jika i mencapai 5, akhiri program.

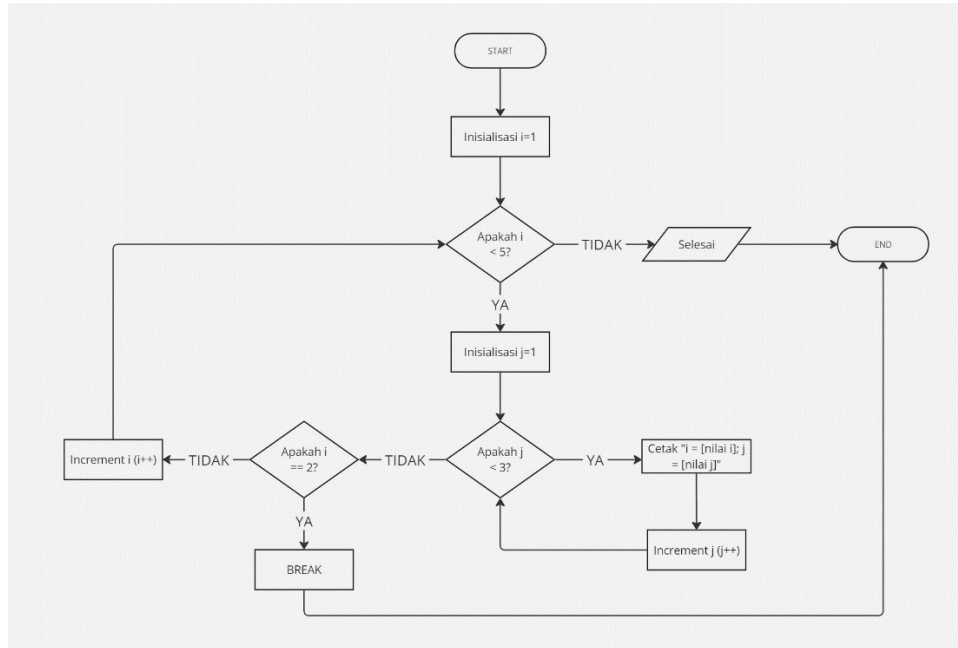


```
1 public class Foriandj {
2     public static void main(String[] args) {
3         pertama:
4             for( int i = 1; i < 5; i++) {
5                 kedua:
6                     for(int j = 1; j < 3; j ++ ) {
7                         System.out.println("i = " + i + "; j = " +j);
8         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

Console Output:

```
<terminated> Foriandj [Java Application]
i = 1; j = 1
i = 1; j = 2
i = 2; j = 1
i = 2; j = 2
```

Flowchart Latihan 1.2



Algoritma Latihan 1.3

1. Mulai.
2. Inisialisasi program dengan deklarasi nama kelas.
3. Deklarasi main method.
4. Inisialisasi scanner untuk membaca input pengguna.
5. Tampilkan pesan "Masukan Input:"
6. Membaca nilai tinggi dari pengguna.
7. Loop Pertama (t): Mulai loop untuk t dari 1 hingga tinggi.
8. Loop Kedua (s): Mulai loop untuk mencetak spasi, selama s lebih besar dari t.
9. Cetak spasi untuk posisi segitiga.
10. Loop Ketiga (b): Mulai loop untuk mencetak bintang, selama b kurang dari atau sama dengan $2 * t - 1$.
11. Cetak bintang (*) untuk membentuk pola segitiga.
12. Setelah selesai mencetak bintang untuk satu baris, program newline.
13. Setelah menyelesaikan satu iterasi untuk t, program meng-increment t.
14. Tutup scanner setelah semua proses selesai.
15. Akhiri program.

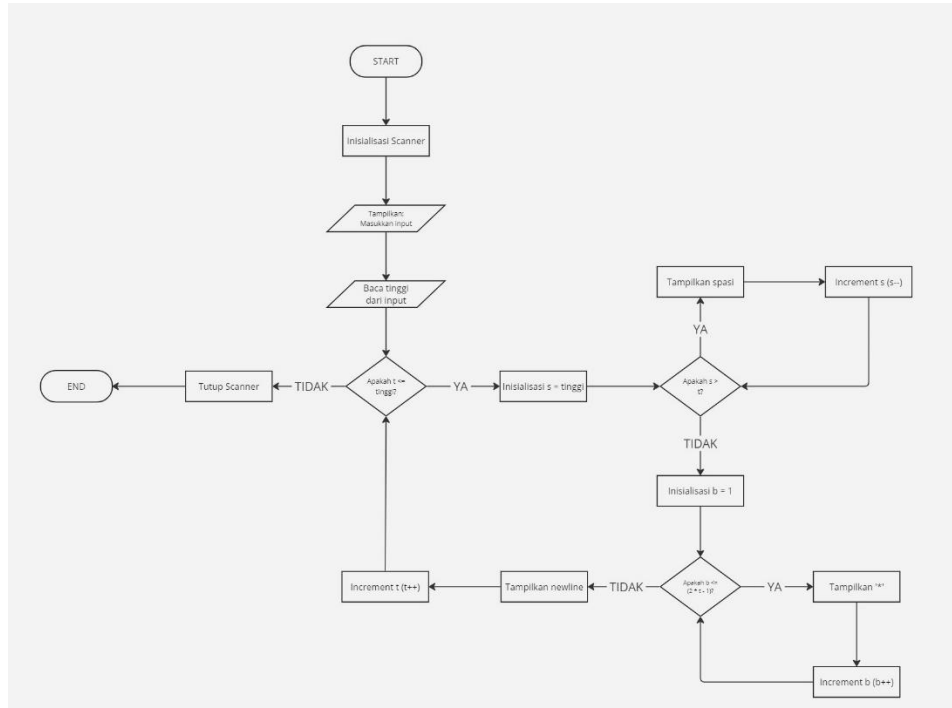
```

1 import java.util.Scanner;
2
3 public class Fortuga {
4     public static void main(String[] args) {
5
6         Scanner input = new Scanner(System.in);
7         System.out.print("Masukan Input: ");
8         int tinggi = input.nextInt();
9         for (int t = 1; t <= tinggi; t++) {
10
11             for (int s = tinggi; s > t; s--) {
12                 System.out.print(" ");
13             }
14
15             for (int b = 1; b <= (2 * t - 1); b++) {
16                 System.out.print("*");
17             }
18             System.out.println();
19         }
20         input.close();
21     }
22 }
23
24

```

Console > Fortuga [Java Application] C:\Users\...
Masukan Input: 7

Flowchart Latihan 1.3



[No.1] Kesimpulan

Analisa:

Soal-soal yang diberikan berkaitan dengan materi FOR yang memiliki permasalahan berbeda di setiap Latihan soalnya. Permasalahan yang terdapat pada Latihan 1 berkaitan dengan urutan inisialisasi dan evaluasi kondisi, juga berkaitan dengan penggunaan fungsi continue dan break. Pada baris yang kosong perlu ditentukan penggunaan continue dan break agar luaran yang ditampilkan berhenti pada angka 6. Tipe data yang digunakan juga harus diganti yang awalnya double untuk menyimpan bilangan decimal diganti dengan int untuk menyimpan bilangan bulat.

Nama & NPM	Topik:	Tanggal:
HANIFAH AZIZAH G1F024037	FOR WHILE JAVA	09/10/2024

[No.2] Identifikasi Masalah:

Latihan 2

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!

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

[No.2] Analisis dan Argumentasi

2.1 Ketika kode pada Ubah 1 yang awalnya berupa `if(i==4){` diubah menjadi `if(i % 3 == 0){`, dan kode pada ubah 2 yang awalnya `break;` diubah menjadi `continue;` maka luaran yang awalnya berupa angka 1, 2, 3 berubah menjadi 1, 2, 3, 4, 5, 6 dikarenakan perbedaan perintah yang diterima oleh program. Kode yang baru memerintahkan program untuk menghitung angka yang merupakan kelipatan 3, jika angka kelipatan 3 maka luaran true dan ditampilkan pada luaran, `continue` melanjutkan eksekusi program karena ditambah setelah `i++` sehingga angka kelipatan 3 akan ditampilkan.

2.2 Ketika input jumlah yang ingin diulang diisi dengan angka 0 maka tidak ada luaran yang ditampilkan atau kalimat yang ingin ditampilkan tidak muncul diluaran. Ketika kode `while` diubah menjadi kode `do while`, dengan input jumlah yang ingin diulang sama yaitu 0, maka luaran yang dihasilkan adalah kalimat yang diinput sebanyak 1 buah. Maka dapat diketahui jika perbedaan dari `while` dan `do while` adalah `while` memproses input dan menentukan kondisi true atau false kemudian melakukan perintah, sedangkan `do while` melakukan perintah terlebih dahulu baru memproses input sehingga `do while` akan mengeksekusi dan menampilkan luaran paling tidak satu kali.

2.3 Kode terdapat pada bagian penyusunan algoritma dan kode program.

2.4 Flowchart terdapat pada bagian penyusunan algoritma dan kode program.

[No.2] Penyusunan Algoritma dan Kode Program

Kode Latihan 2.3

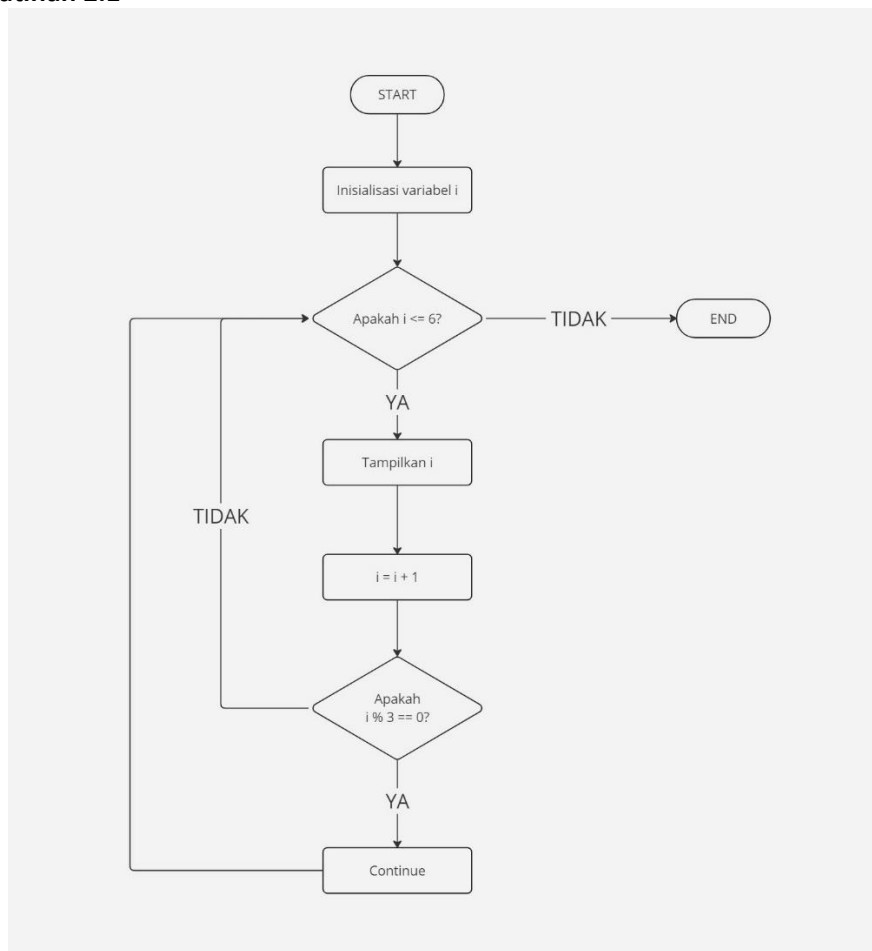
```
# Bersarang.java Breakremove.java Forstage.java Fortuga.java Forandj.java Whilee.java Whileethree.java Whiletwo.java x
1 import java.util.Scanner;
2 public class Whiletwo {
3     public static void main(String[] args) {
4         Scanner scanner = new Scanner(System.in);
5         int idPelajaran = 0; //inisialisasi idPelajaran
6         double nilaiPelajaran; //inisialisasi nilaiPelajaran
7         double nilaiRataRata = 0; //inisialisasiRataRata
8
9         System.out.print("Masukkan jumlah pelajaran: ");
10        int jumlahPelajaran = scanner.nextInt(); //Minta pengguna untuk menuliskan jumlah pelajaran
11
12        while (idPelajaran < jumlahPelajaran) { //Ketika idPelajaran lebih kecil dari jumlah pelajaran
13            System.out.print("Masukkan nilai pelajaran ke-" + (idPelajaran + 1) + ": ");
14            nilaiPelajaran = scanner.nextDouble(); //Minta pengguna untuk menuliskan nilai pelajaran
15            nilaiRataRata = (nilaiPelajaran + nilaiRataRata * idPelajaran) / (idPelajaran + 1);
16            // Hitung nilai rata-rata = (nilai pelajaran + nilai rata-rata) / 2
17            idPelajaran++; // Tambah satu ke idPelajaran
18        }
19
20        System.out.printf("Nilai rata-rata: %.2f\n", nilaiRataRata); //
21        scanner.close();
22    }
23 }
24
```

Console x

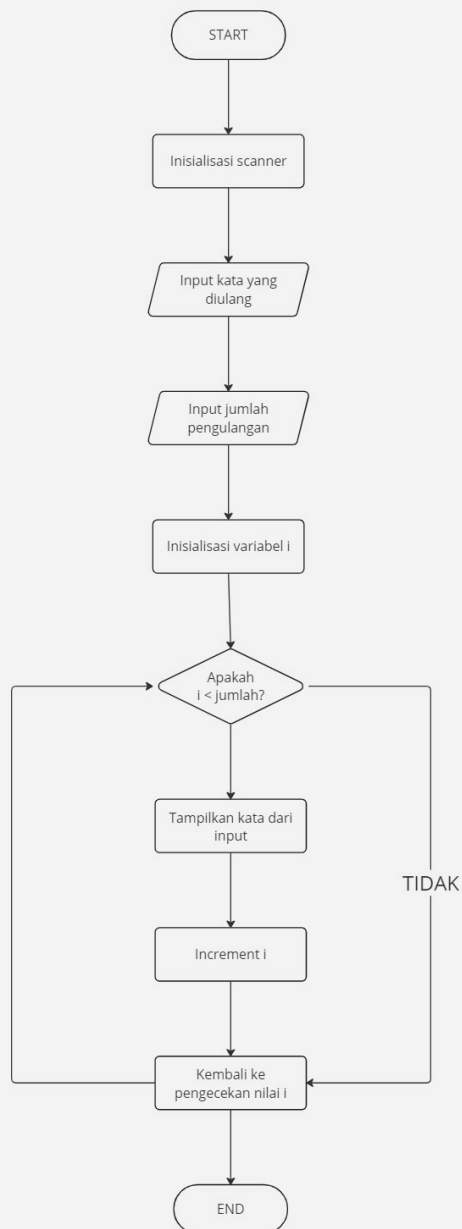
```
<terminated> Whiletwo [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_22.0.2.v20240802-1626\jre\bin\javaw.exe (9 Oct 2
Masukkan jumlah pelajaran: 3
Masukkan nilai pelajaran ke-1: 10
Masukkan nilai pelajaran ke-2: 9
Masukkan nilai pelajaran ke-3: 8
Nilai rata-rata: 9,00
```

Kode di atas merupakan kode yang disarankan sesuai dengan pseudocode yang ada dengan dapat menerima input jumlah Pelajaran dan jumlah nilai dari masing-masing Pelajaran sehingga dapat menghitung rata-rata sesuai dengan jumlah Pelajaran yang ada dan menampilkan luaran rata-rata dengan tipe data double sesuai dengan inisialisasi variabel nilai rata-rata.

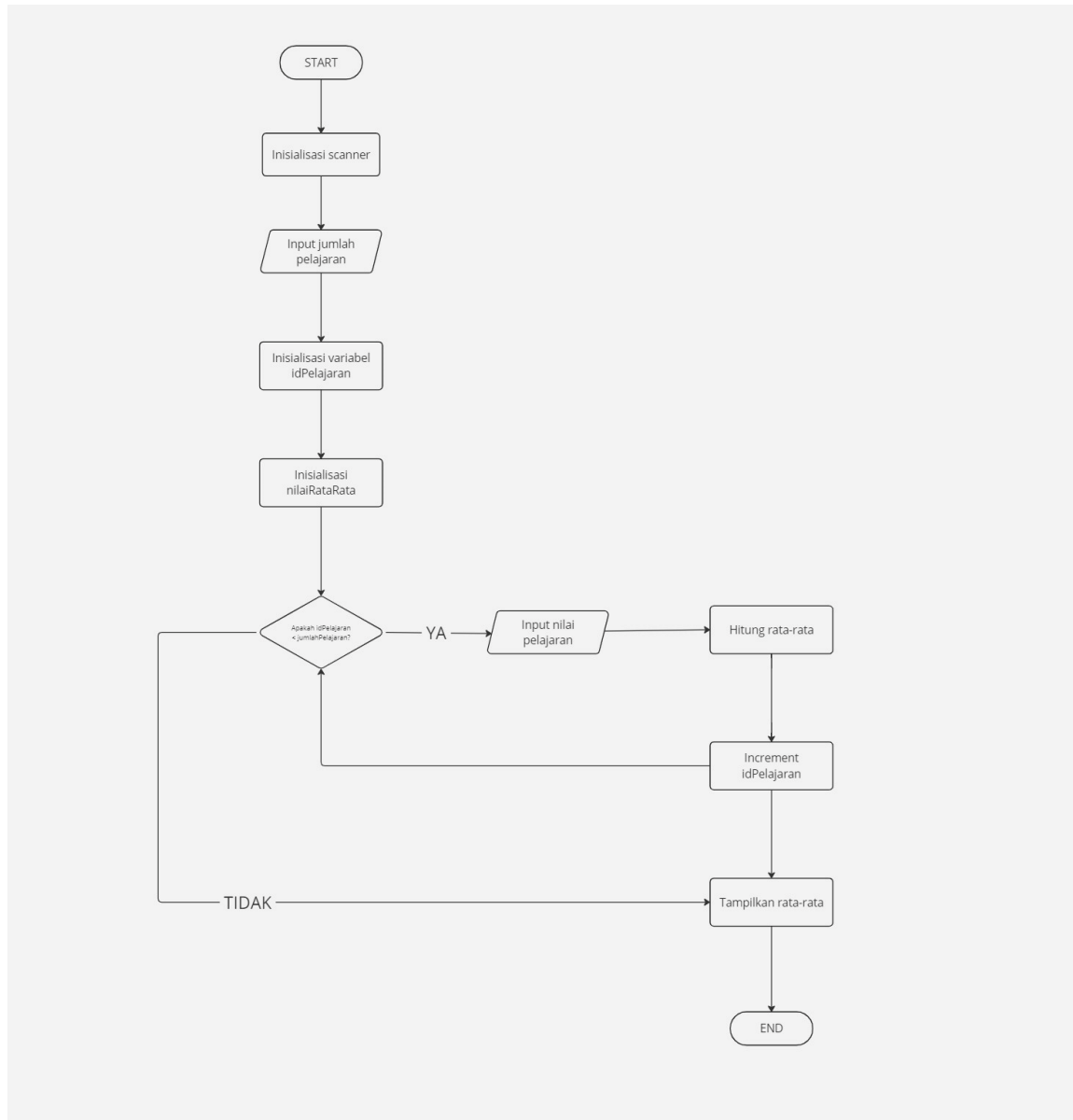
Flowchart Latihan 2.1



Flowchart Latihan 2.2



Flowchart Latihan 2.3



[No.2] Kesimpulan

Berdasarkan Latihan-latihan pada soal nomor 2, diketahui bahwa do while dan while memiliki perbedaan pada eksekusi perintah pada program. Jika while memproses input dan menentukan true false sebelum mengeksekusi perintah, do while akan mengeksekusi perintah terlebih dahulu kemudian memproses input sehingga do while akan tetap mencetak setidaknya satu kali luaran walaupun hasilnya false.

Refleksi

Pengalaman belajar yang didapatkan selama minggu materi FOR dan WHILE adalah mendapatkan pengetahuan terkait bagaimana penggunaan dari kedua perintah tersebut dan bagaimana penggunaannya dapat memengaruhi luaran yang dihasilkan. Tantangan yang dihadapi berupa penggunaan FOR dan WHILE yang perlu disesuaikan dengan kondisi tertentu, dan bagaimana menentukan DO WHILE atau WHILE yang perlu digunakan dalam suatu susunan program. Materi kali ini membahas tentang perintah FOR, WHILE, DO WHILE, susunan kode pseudocode, dan looping flowchart.