

Nama & NPM	Topik:	Tanggal:
Sevi Rina Pertiwi G1F024044	FOR dan WHILE Java	29 September 2024

Latihan 1

[No 1.1] Identifikasi Masalah:

1.1) Evaluasi penyebab kesalahan dan perbaiki kode pada Contoh!

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

Jawaban:

Penyebab kesalahan pada contoh satu yaitu menggunakan tipe data boolean operasi operator `<=` dengan nilai 15 dan menggunakan operator increment `y++`, perbaiki kode dilakukan dengan cara mengubah tipe data boolean menjadi int, menghapus `<` dari `<=` 15 dan mengubah operator increment `y++` menjadi decrement `y--`.

Rekomendasikan kata kunci yang tepat untuk diletakkan pada baris 1 adalah `continue` dan baris 2 adalah `break` lalu mengubah kode yang telah diperbaiki pada contoh 1 yaitu inisialisasi `y = 0` dengan kondisi `y <= 15` dan perubahan menggunakan increment `y++`. Perintah `continue`; untuk digunakan untuk baris kosong 1: lewati iterasi saat `y` ganjil dan menambahkan perintah `break` di bawah perintah `else if` yang digunakan untuk menghentikan loop ketika `y = 8`.

[No 1.1] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara memperbaiki kode program yang salah pada contoh 1 dan membuat kode baru untuk menampilkan luaran 0, 2, 4, 6 dengan mengubah isi kode program yang di perbaiki di contoh 1 yaitu inisialisasi `y = 0` dengan kondisi `y <= 15` dan perubahan menggunakan increment `y++`, lalu menggunakan perintah `continue`; untuk baris kosong 1: lewati iterasi saat `y` ganjil dan menambahkan perintah `break` di bawah perintah `else if` yang digunakan untuk menghentikan loop ketika `y = 8`.
- 2) Alasan solusi ini karena yang pertama karena ingin memperbaiki kode program contoh 1 dan membuat kode program dengan mengubah isi kode agar menampilkan luaran.
- 3) Perbaiki kode program dilakukan dengan cara mengubah tipe data boolean menjadi int, menghapus `<` dari `<= 15` dan mengubah operator increment `y++` menjadi decrement `y--`.

Rekomendasikan kata kunci yang tepat untuk diletakkan pada baris 1 adalah `continue` dan baris 2 adalah `break` lalu mengubah kode yang telah diperbaiki pada contoh 1 yaitu inisialisasi `y = 0` dengan kondisi `y <= 15` dan perubahan menggunakan increment `y++`. Perintah `continue`; untuk digunakan untuk baris kosong 1: lewati iterasi saat `y` ganjil dan menambahkan perintah `break` di bawah perintah `else if` yang digunakan untuk menghentikan loop ketika `y = 8`.

Kode program perbaikan:

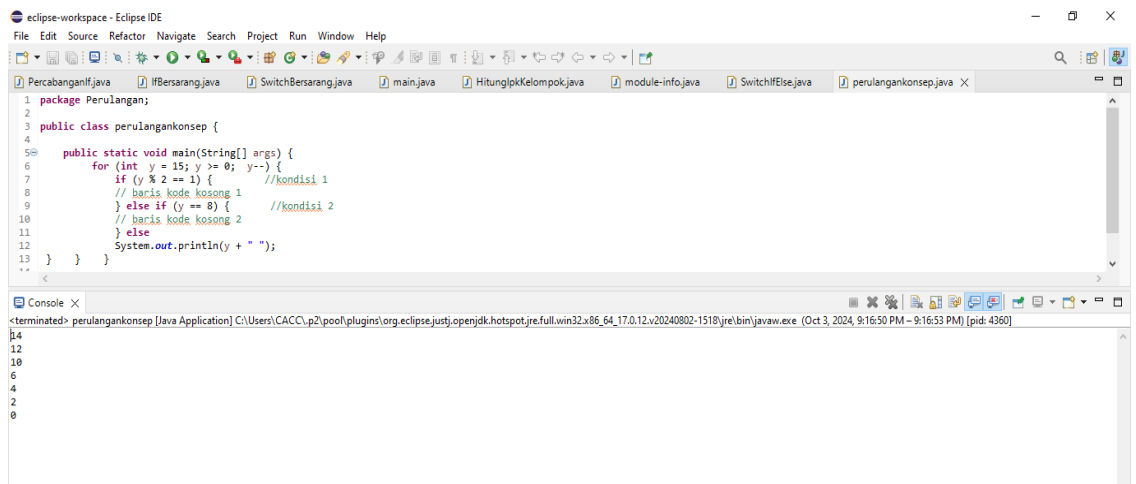
```
for (int y = 15; y >= 0; y--) {  
    if (y % 2 == 1) { //kondisi 1  
        // baris kode kosong 1  
    } else if (y == 8) { //kondisi 2  
        // baris kode kosong 2  
    } else  
        System.out.println(y + " ");  
} } }
```

Kode program dan kata kunci dibaris 1 dan 2:

```
for (int y = 0; y <= 15; y++){ // Perbaikan sintaks for-loop  
    if (y % 2 == 1) { // Kondisi untuk bilangan ganjil  
        continue; // Baris kode kosong 1: lewati iterasi saat y ganjil  
    } else if (y == 8) { // Kondisi untuk y = 8  
        break;  
    } else  
        System.out.println(y + " ");  
} } }
```

[No 1.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Mulai
 - (b) Inisialisasi variabel $y = 0$
 - (c) Selama y kurang dari atau sama dengan 15:
 - Jika y ganjil maka akan dilanjutkan proses berikutnya
 - Jika y sama dengan 8 maka proses akan dihentikan
 - Jika tidak y dicetak
 - (d) Hasil dari kode tersebut akan di print dan ditampilkan dalam luaran atau output
 - (e) Selesai
- 2) Kode program dan luaran
 - a) Kode program,komentar dan hasil luaran.

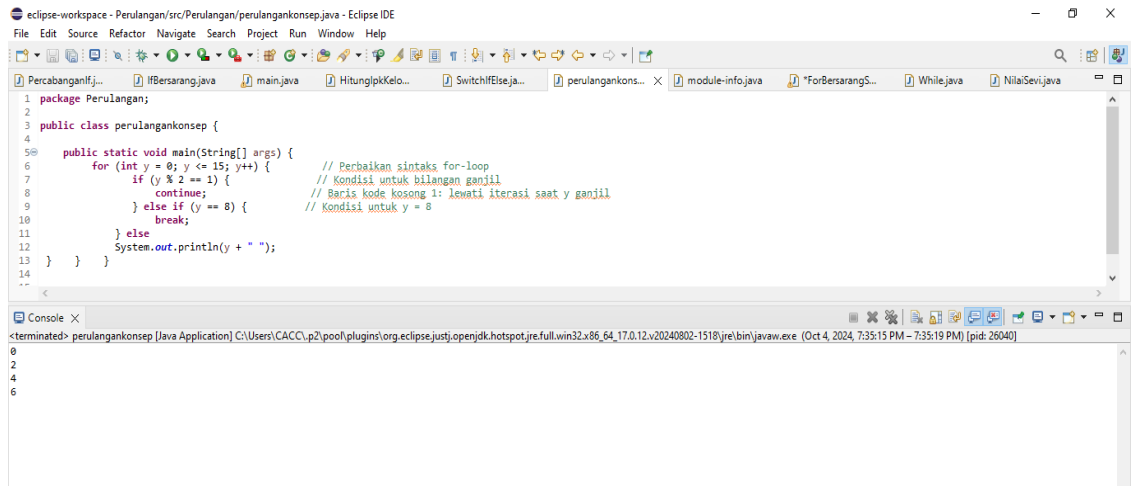


The screenshot shows the Eclipse IDE interface. The main editor displays a Java file named `perulangankonsep.java` with the following code:

```
1 package Perulangan;  
2  
3 public class perulangankonsep {  
4  
5     public static void main(String[] args) {  
6         for (int y = 15; y >= 0; y--) {  
7             if (y % 2 == 1) { //kondisi 1  
8                 // baris kode kosong 1  
9             } else if (y == 8) { //kondisi 2  
10                // baris kode kosong 2  
11            } else  
12                System.out.println(y + " ");  
13        } } }  
14
```

The console window at the bottom shows the output of the program, which is the numbers 15, 12, 10, 6, 4, 2, and 0, each on a new line.

Kode program yang diperbaiki



```
1 package Perulangan;
2
3 public class perulangankonsep {
4
5     public static void main(String[] args) {
6         for (int y = 0; y <= 15; y++) {
7             if (y % 2 == 1) {
8                 continue;
9             } else if (y == 8) {
10                 break;
11             } else {
12                 System.out.println(y + " ");
13             }
14         }
15     }
16 }
```

Kode program dan kata kunci baris 1 dan 2

- b) Luaran yang dihasilkan sudah benar dan sesuai dengan soal yang diminta. Yaitu menampilkan luaran 0, 2, 4, 6 dengan menggunakan kata kunci continue pada baris 1 yang digunakan untuk menampilkan baris kosong jika y sama dengan ganjil dan kata kunci break pada baris 2 yang digunakan untuk menghentikan perulangan jika y sudah sama dengan 8.

[No 1.1] Kesimpulan

1) Analisa

Kode program menggunakan struktur for-loop untuk mengulangi proses pencetakan bilangan genap dari 0 sampai 8, dengan memanfaatkan operator modulus sisa bagi % untuk memeriksa apakah suatu bilangan ganjil atau genap. Jika bilangan ganjil ditemukan maka perintah continue dilewatkan agar tidak dicetak, dan ketika nilai y mencapai 8 maka perintah break digunakan untuk menghentikan loop. Kode program ini menggunakan kontrol alur seperti continue dan break serta pengkondisian yang telah dijelaskan dalam materi komputer dan pemrograman.

[No 1.2] Identifikasi Masalah:

- 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

Jawaban:

Bagaimana menghentikan loop setelah iterasi dengan i == 2 untuk memenuhi output yang diinginkan yaitu:

i = 1; j = 1

i = 1; j = 2

i = 2; j = 1

i = 2; j = 2

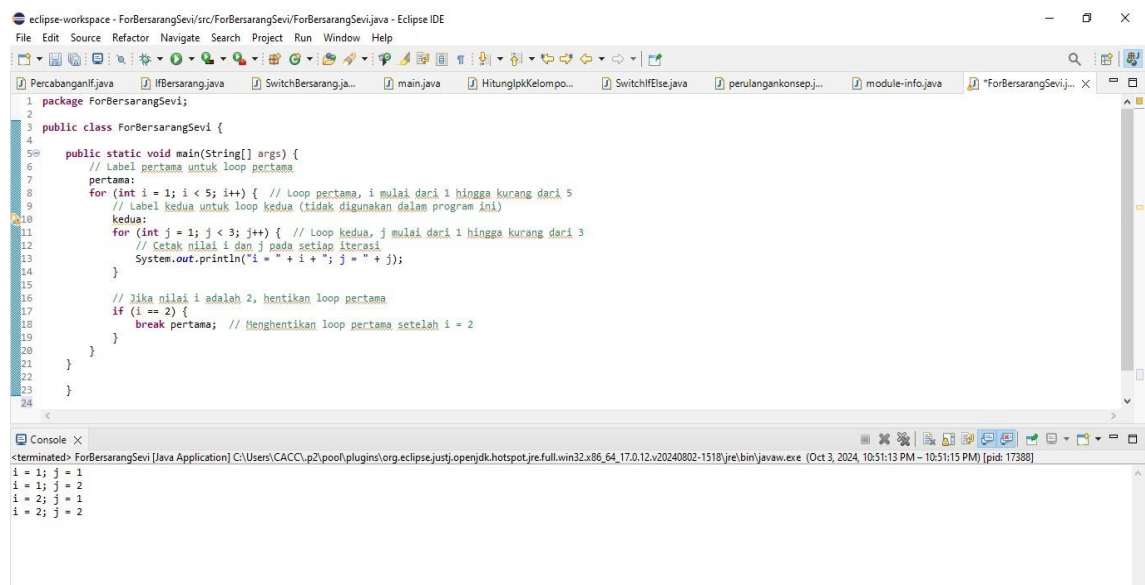
Saya merekomendasikan untuk menggunakan break pada kode yang hilang di loop for bersarang setelah kondisi tertentu terpenuhi yaitu (i == 2)

[No 1.2] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan perintah break pada komentar kode yang hilang.
- 2) Alasan solusi ini karena perintah break digunakan untuk menghentikan loop pertama setelah kondisi tertentu yaitu $i == 2$. Dengan menambahkan label pertama: pada loop luar dan menggunakan break pertama; setelah $i == 2$ untuk menghentikan iterasi lebih lanjut dan menghasilkan output yang sesuai.
- 3) Perbaikan kode program dilakukan dengan cara menambahkan kode break pertama; pada komentar kode yang hilang di bawah if ($i == 2$).

[No 1.2] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Mulai
 - (b) Loop pertama dimulai dengan variabel i yang diinisialisasi dengan nilai i
 - Loop pertama berjalan selama $i < 5$ yaitu selama nilai i kurang dari 5
 - (c) Untuk setiap nilai i:
 - Loop kedua dimulai dengan variabel j yang diinisialisasi dengan nilai 1
 - Loop kedua berjalan selama $j < 3$ selama nilai j kurang dari 3
 - (d) Cetak nilai i dan j dalam format "i = [nilai i] = [nilai j]"
 - (e) Setelah loop kedua selesai:
 - Periksa apakah nilai i sama dengan 2
 - Jika $i == 2$ hentikan loop pertama dengan menggunakan perintah break pada label pertama;
 - (f) Jika kondisi $i == 2$ tidak terpenuhi lanjutkan ke berikutnya dari loop pertama
 - (g) Loop pertama akan berakhir secara normal setelah nilai i mencapai 5 atau dihentikan lebih awal jika $i == 2$
 - (h) Akhiri program.
- 2) Kode program dan luaran
 - a) Kode program, komentar dan hasil luaran



The screenshot shows the Eclipse IDE with a Java project named 'ForBersarangSevi'. The main.java file contains the following code:

```
1 package ForBersarangSevi;
2
3 public class ForBersarangSevi {
4
5     public static void main(String[] args) {
6         // Label pertama untuk loop pertama
7         pertama:
8         for (int i = 1; i < 5; i++) { // Loop pertama, i mulai dari 1 hingga kurang dari 5
9             // Label kedua untuk loop kedua (tidak digunakan dalam program ini)
10            kedua:
11            for (int j = 1; j < 3; j++) { // Loop kedua, j mulai dari 1 hingga kurang dari 3
12                // Cetak nilai i dan j pada setiap iterasi
13                System.out.println("i = " + i + " ; j = " + j);
14            }
15
16            // Jika nilai i adalah 2, hentikan loop pertama
17            if (i == 2) {
18                break pertama; // Menghentikan loop pertama setelah i = 2
19            }
20        }
21    }
22 }
23
24 }
```

The console output shows the following results:

```
<terminated> ForBersarangSevi [Java Application] C:\Users\CACCI\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.12.v20240802-1518\jre\bin\java.exe (Oct 3, 2024, 10:51:13 PM - 10:51:15 PM) [pid: 17388]
i = 1; j = 1
i = 1; j = 2
i = 2; j = 1
i = 2; j = 2
```

- b) Luaran yang dihasilkan sudah benar dan sesuai dengan soal yang diminta.
Pada iterasi pertama, ketika $i = 1$, loop kedua berjalan dua kali dengan j bernilai 1 dan 2, sehingga dicetak " $i = 1; j = 1$ " dan " $i = 1; j = 2$ ". Pada iterasi kedua, ketika $i = 2$, loop kedua juga berjalan dua kali, mencetak " $i = 2; j = 1$ " dan " $i = 2; j = 2$ ". Setelah itu, karena nilai i sama dengan 2, perintah break menghentikan loop pertama. Akibatnya, tidak ada lagi iterasi untuk i yang lebih besar dari 2, dan program selesai. Luaran yang dihasilkan berisi total empat baris yang menampilkan kombinasi nilai i dan j dari iterasi yang berlangsung sebelum loop dihentikan.

[No 1.2] Kesimpulan

1) Analisa

Kode program "ForBersarangSevi" menggunakan dua loop bersarang untuk mencetak nilai i dan j . Loop pertama (i) berjalan dari 1 hingga kurang dari 5, dan loop kedua (j) dari 1 hingga kurang dari 3. Ketika i mencapai 2, program menghentikan seluruh loop pertama dengan perintah break, sehingga hanya mencetak nilai i dan j untuk $i = 1$ dan $i = 2$. Label kedua tidak digunakan dan bisa dihapus tanpa mempengaruhi hasil. Program berhenti lebih cepat karena break menghentikan iterasi lebih awal.

[No 1.3 & 4] Identifikasi Masalah:

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

Luaran berbentuk piramida

Masukan Input: 7

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

Rekomendasikan kode untuk menghasilkan luaran tersebut!

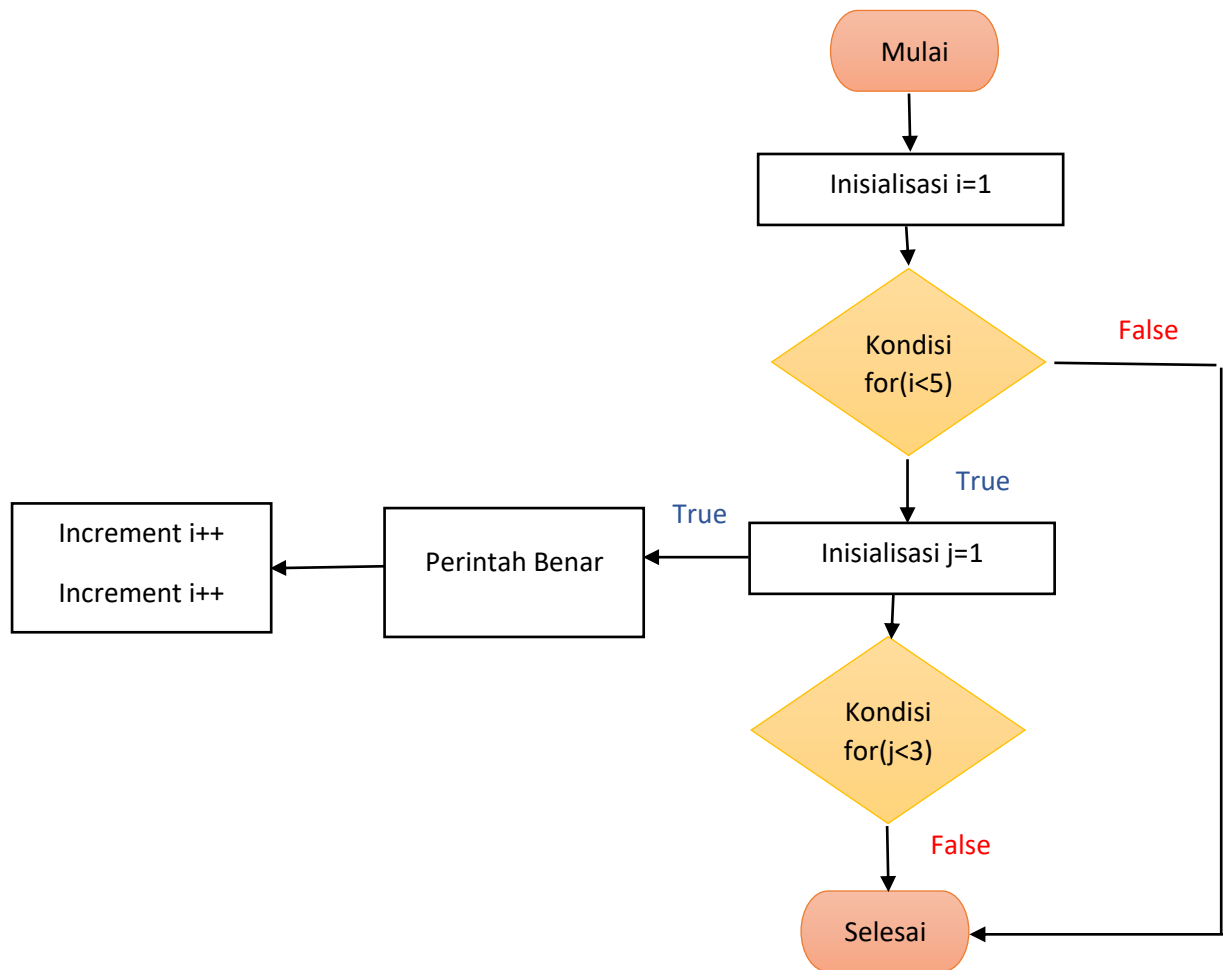
Jawaban:

Rekomendasi kode yang tepat untuk menghasilkan luaran piramida adalah dengan menambahkan kode berikut ke dalam contoh 3:

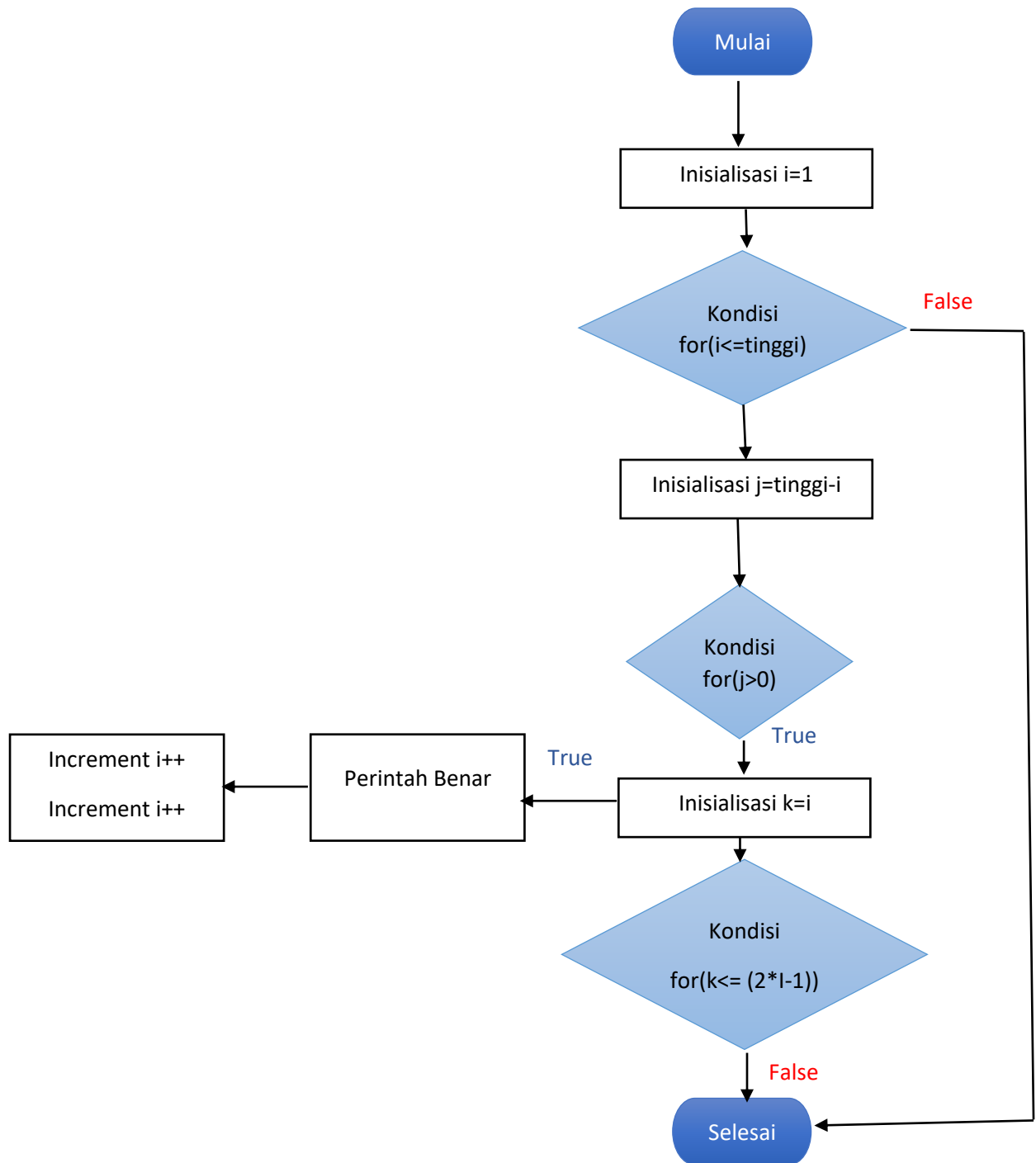
1. Diberis 10: Loop pertama (for (int t = 1; t <= tinggi; t++)) digunakan untuk mengatur tinggi piramida.
2. Diberis 12: Loop kedua (for (int s = tinggi; s > t; s--)) untuk menambahkan spasi, yang berkurang seiring bertambahnya baris.
3. Diberis 15: Loop ketiga (for (int b = 1; b <= (2 * t - 1); b++)) mencetak bintang dalam jumlah yang bertambah dengan pola piramida.

1.4) Analisa diagram flowchart dari Latihan 1.2 dan 1.3!
Jawaban:

Flowchat 1.2



Flowchat 1.3



[No 1.3] Analisis dan Argumentasi

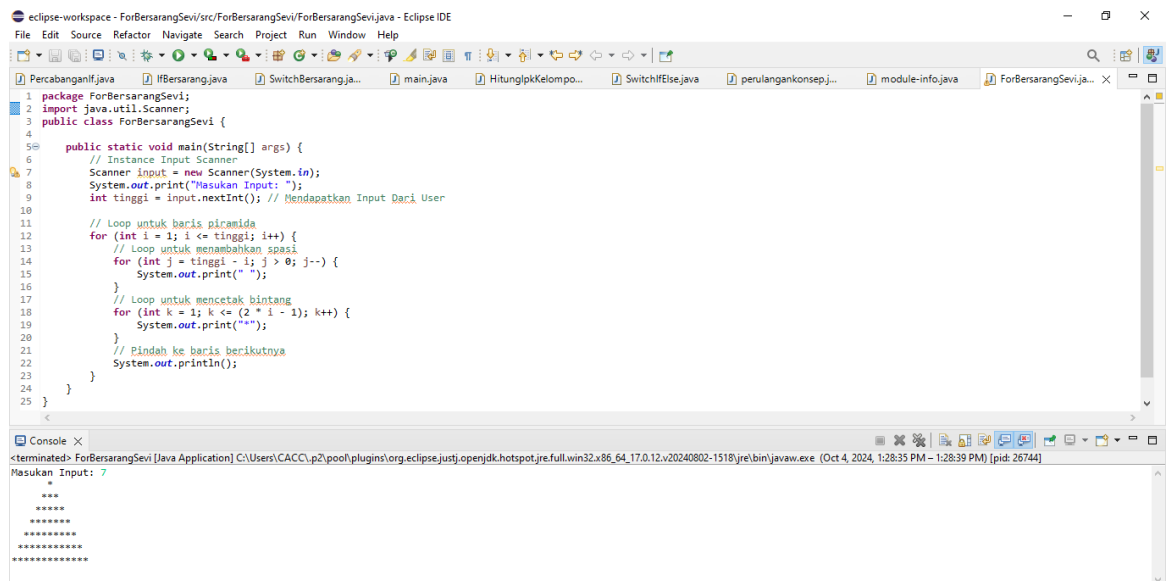
- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menambahkan pengaturan untuk spasi dan peraturan jumlah bintang di setiap baris. Karena kode awal di contoh 3 hanya mencetak jumlah bintang secara bertahap tanpa menggunakan spasi yang dibutuhkan agar bintang-bintang tersebut membentuk pola piramida.
- 2) Alasan solusi ini karena dalam membentuk piramida simetris, setiap baris membutuhkan dua hal:
 - a. Spasi di awal baris yang semakin berkurang dari baris atas ke bawah.
 - b. Bintang yang semakin bertambah sesuai dengan pola piramida, yang bertambah secara ganjil (1, 3, 5, 7, dan seterusnya).
- 3) Perbaikan kode program dilakukan dengan cara menambahkan loop pertama (for (int t = 1; t <= tinggi; t++)) digunakan untuk mengatur tinggi piramida. Loop kedua (for (int s = tinggi; s > t; s--)) untuk menambahkan spasi, yang berkurang seiring bertambahnya baris. Loop ketiga (for (int b = 1; b <= (2 * t - 1); b++)) mencetak bintang dalam jumlah yang bertambah dengan pola piramida. Berikut ini kodenya:

```
// Instance Input Scanner
Scanner input = new Scanner(System.in);
System.out.print("Masukan Input: ");
int tinggi = input.nextInt(); // Mendapatkan Input Dari User

// Loop untuk baris piramida
for (int i = 1; i <= tinggi; i++) {
    // Loop untuk menambahkan spasi
    for (int j = tinggi - i; j > 0; j--) {
        System.out.print(" ");
    }
    // Loop untuk mencetak bintang
    for (int k = 1; k <= (2 * i - 1); k++) {
        System.out.print("*");
    }
    // Pindah ke baris berikutnya
    System.out.println();
}
}
```

[No 1.3] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Mulai
 - (b) Buat objek scanner untuk membaca input dari pengguna
 - (c) Tampilkan pesan kepada pengguna untuk memasukkan nilai tinggi piramida
 - (d) Baca input dari pengguna
 - (e) Lakukan perulangan i dari 1 hingga tinggi untuk setiap baris piramida
 - (f) Ulangi langkah 5 hingga semua baris piramida selesai dicetak
 - (g) Program selesai
- 2) Kode program dan luaran
 - a) Kode program,komentar dan hasil luaran



```
1 package ForBersarangSevi;
2 import java.util.Scanner;
3 public class ForBersarangSevi {
4
5     public static void main(String[] args) {
6         // Instance Input Scanner
7         Scanner input = new Scanner(System.in);
8         System.out.print("Masukan Input: ");
9         int tinggi = input.nextInt(); // Mendapatkan Input Dari User
10
11         // Loop untuk baris piramida
12         for (int i = 1; i <= tinggi; i++) {
13             // Loop untuk menambahkan spasi
14             for (int j = tinggi - i; j > 0; j--) {
15                 System.out.print(" ");
16             }
17             // Loop untuk mencetak bintang
18             for (int k = 1; k <= (2 * i - 1); k++) {
19                 System.out.print("*");
20             }
21             // Pindah ke baris berikutnya
22             System.out.println();
23         }
24     }
25 }
```

Console ×
<terminated> ForBersarangSevi [Java Application] C:\Users\CACC\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.12.v20240802-1518\jre\bin\javaw.exe (Oct 4, 2024, 1:28:35 PM - 1:28:39 PM) [pid: 26744]
Masukan Input: 7
*
* *
* * *
* * * *
* * * * *
* * * * *
* * * * *

- b) Luaran yang dihasilkan sudah benar dan sesuai dengan soal yang diminta. Dengan input nilai 7, program akan menghasilkan piramida bintang yang terdiri dari 7 baris. Setiap baris piramida dimulai dengan sejumlah spasi yang berkurang dari baris pertama hingga baris terakhir. Pada baris pertama, terdapat 6 spasi diikuti oleh 1 bintang, sementara di baris kedua terdapat 5 spasi dan 3 bintang. Pola ini terus berlanjut dengan jumlah spasi yang semakin sedikit dan jumlah bintang yang semakin bertambah secara ganjil (1, 3, 5, 7, 9, 11, dan 13 bintang).

[No 1.3] Kesimpulan

1) Analisa

Kode program ini berhasil mencetak piramida bintang yang simetris dengan memanfaatkan pengaturan jumlah spasi dan bintang pada setiap baris. Program menerima input berupa nilai tinggi piramida dari pengguna, lalu menghasilkan piramida sesuai dengan tinggi tersebut. Setiap baris piramida terdiri dari spasi yang berkurang seiring bertambahnya baris, diikuti dengan bintang yang bertambah dalam pola ganjil. Dengan cara ini, piramida terlihat Pendekatan yang digunakan dalam kode ini efektif untuk membentuk piramida yang beraturan, menunjukkan pemahaman yang baik tentang konsep looping dan manipulasi karakter dalam Java.

Latihan 2

[No 2.1] Identifikasi Masalah:

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!

Jawaban:

Ubah1: Ganti baris `if(i==4)` menjadi `if(i % 3 == 0)`.

Ubah2: Ganti baris `break;` menjadi `continue;`

Sebelum perubahan (dengan `break`): Program berhenti pada nilai 3 karena `break` menghentikan loop ketika `i == 4`.

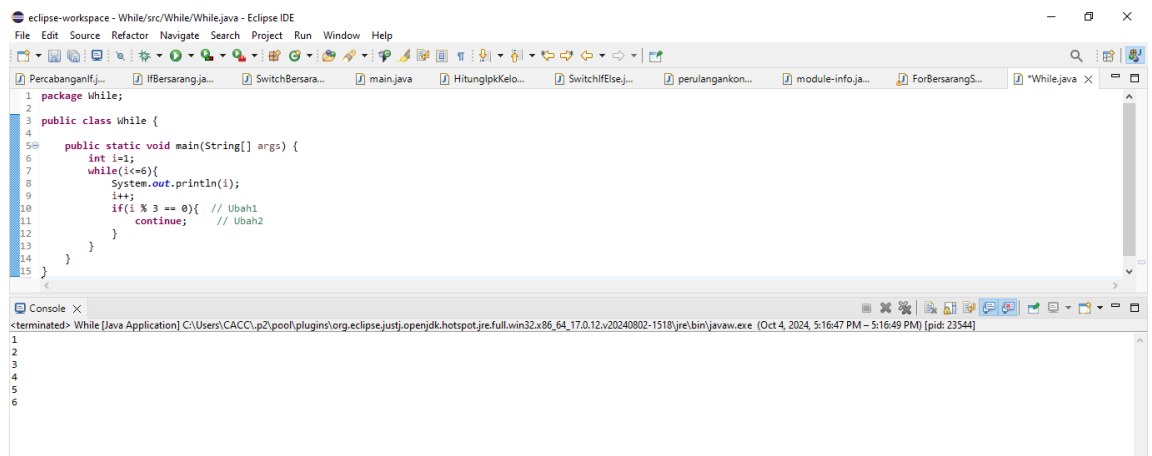
Setelah perubahan (dengan continue): Program melewati angka 3 karena continue melompati iterasi ketika $i \% 3 == 0$, tetapi melanjutkan mencetak angka 4, 5, dan 6.

[No 2.1] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara mengganti perintah break; dengan continue;, serta mengubah kondisi pengecekan dari $\text{if}(i == 4)$ menjadi $\text{if}(i \% 3 == 0)$.
- 2) Alasan solusi ini karena menggunakan break; akan menghentikan loop sepenuhnya begitu kondisi terpenuhi, yang menyebabkan iterasi terhenti lebih awal dan output tidak lengkap. Sementara itu, menggunakan continue; memungkinkan loop tetap berjalan, namun melewati iterasi tertentu (dalam hal ini, saat i adalah kelipatan dari 3), sehingga kita bisa mengontrol kapan angka tertentu dilewatkan tanpa menghentikan keseluruhan proses perulangan.
- 3) Perbaikan kode program dilakukan dengan cara mengganti baris $\text{if}(i == 4)$ menjadi $\text{if}(i \% 3 == 0)$, untuk memeriksa kelipatan 3 daripada nilai spesifik 4. Dan break; menjadi continue;, agar loop tidak berhenti tetapi melewati iterasi tertentu.

[No 2.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Mulai
 - (b) Inisialisasi $i = 1$
 - (c) Selama nilai dari $i \leq 6$
 - Cetak nilai i
 - Tambahkan nilai i sebesar 1
 - Jika nilai i adalah kelipatan 3, lewati iterasi dan gunakan continue
 - (d) Ulangi langkah 3 sampai $i > 6$
 - (e) Selesai.
- 2) Kode program dan luaran
 - a) Kode program, komentar dan hasil luaran



```
1 package While;
2
3 public class While {
4
5     public static void main(String[] args) {
6         int i=1;
7         while(i<=6){
8             System.out.println(i);
9             i++;
10            if(i % 3 == 0){ // Ubah1
11                continue; // Ubah2
12            }
13        }
14    }
15 }
```

Console

```
<terminated> While [Java Application] C:\Users\CACCI\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.12.v20240802-1518\jre\bin\java.exe (Oct 4, 2024, 5:16:47 PM - 5:16:49 PM) [pid: 23544]
1
2
3
4
5
6
```

- b) Luaran yang dihasilkan sudah benar dan sesuai dengan soal yang diminta. Dengan output 1, 2, 3, 4, 5, 6 dari hasil perintah kode program tersebut. Program mencetak angka 1, 2, 4, 5 dan melewati angka 3 serta 6 karena kondisi $i \% 3 == 0$ terpenuhi, menyebabkan iterasi dilewatkan dengan continue.

[No 2.1] Kesimpulan

1) Analisa

Kode program ini menggunakan while loop untuk mencetak angka dari 1 hingga 6, namun dengan pengecualian untuk angka yang merupakan kelipatan dari 3. Pada setiap iterasi, jika nilai variabel *i* adalah kelipatan dari 3, seperti pada angka 3 dan 6, program menjalankan perintah continue yang menyebabkan iterasi tersebut dilewati tanpa mencetak angka yang bersangkutan. Sebagai hasilnya, program hanya mencetak angka 1, 2, 4, dan 5, sementara angka 3 dan 6 dilewatkan. Penggunaan continue memungkinkan program untuk melewati iterasi tertentu tanpa menghentikan keseluruhan loop, sehingga program tetap berjalan hingga kondisi akhir tercapai, yaitu saat *i* lebih besar dari 6.

[No 2.2] Identifikasi Masalah:

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!

Jawaban:

a. Luaran Saat Jumlah Diulang = 0 dengan while:

Ketika program dijalankan dengan menggunakan while loop dan jumlah pengulangan yang dimasukkan adalah 0, maka while loop tidak akan dieksekusi sama sekali karena kondisi awal $i < \text{jumlah}$ (di mana jumlah adalah 0) tidak pernah benar. Oleh karena itu, program tidak akan mencetak apa pun ke layar. Loop tidak dijalankan, sehingga tidak ada output yang dihasilkan.

b. Luaran Saat Diubah ke do...while dengan Jumlah Diulang = 0:

Jika kode diubah menjadi do...while, program akan tetap mencetak satu kali kata yang dimasukkan oleh pengguna, bahkan jika jumlah pengulangan adalah 0. Hal ini terjadi karena do...while akan mengeksekusi blok kode setidaknya sekali, tanpa memeriksa kondisi pada awalnya. Setelah iterasi pertama, baru kondisi $i < \text{jumlah}$ diperiksa, dan loop akan berhenti karena jumlah yang diulang adalah 0.

Perbedaan while dan do while adalah while: Kondisi dievaluasi sebelum memasuki loop, sehingga jika kondisi awal tidak terpenuhi (misalnya jumlah pengulangan = 0), loop tidak akan dijalankan sama sekali. Sedangkan do...while: Kondisi dievaluasi setelah satu iterasi pertama, sehingga loop selalu dijalankan setidaknya sekali, bahkan jika kondisi awal tidak terpenuhi (seperti saat jumlah pengulangan = 0).

[No 2.2] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara mengubah pengulangan dari while menjadi do...while. Dengan cara ini, program akan selalu mencetak kata setidaknya satu kali, terlepas dari nilai yang dimasukkan untuk jumlah pengulangan.
- 2) Alasan solusi ini karena penggunaan do...while memberikan fleksibilitas lebih dalam situasi di mana pengguna mungkin ingin mengulang kata meskipun jumlah pengulangan yang dimasukkan adalah nol. Dengan menggunakan do...while, kita dapat memastikan bahwa kata tetap dicetak minimal satu kali, sehingga memberikan pengalaman pengguna yang lebih baik.
- 3) Perbaikan kode program dilakukan dengan cara mengubah dari while ke do...while. Pada program yang diperbaiki, kita melakukan inisialisasi variabel *i* dan kemudian menjalankan blok kode yang mencetak kata setidaknya satu kali sebelum memeriksa kondisi. Dengan cara ini, program tetap berfungsi meskipun pengguna memasukkan

jumlah pengulangan yang tidak positif, sekaligus memenuhi tujuan awal program untuk mencetak kata yang diinput oleh pengguna.

[No 2.2] Penyusunan Algoritma dan Kode Program

1) Algoritma dengan while

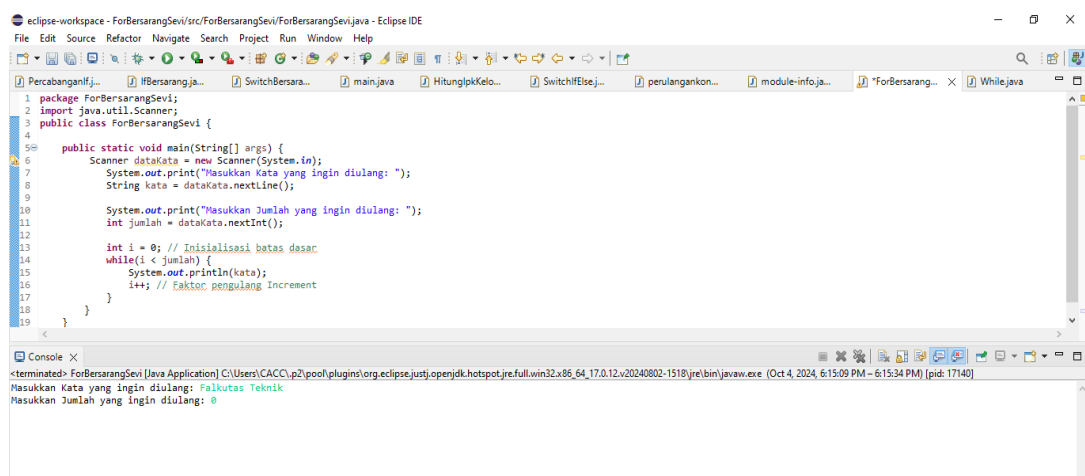
- (a) Mulai
- (b) Inisialisasi Scanner untuk input.
- (c) Tampilkan pesan: "Masukkan Kata yang ingin diulang:"
- (d) Baca input kata dari pengguna.
- (e) Tampilkan pesan: "Masukkan Jumlah yang ingin diulang:"
- (f) Baca input jumlah dari pengguna.
- (g) Inisialisasi variabel i dengan 0.
- (h) Selama $i < \text{jumlah}$, lakukan langkah berikut:
 - Cetak kata
 - Tambahkan i dengan 1
- (i) Selesai.

Algoritma dengan do while

- (a) Mulai
- (b) Inisialisasi Scanner untuk input.
- (c) Tampilkan pesan: "Masukkan Kata yang ingin diulang:"
- (d) Baca input kata dari pengguna.
- (e) Tampilkan pesan: "Masukkan Jumlah yang ingin diulang:"
- (f) Baca input jumlah dari pengguna.
- (g) Inisialisasi variabel i dengan 0.
- (h) Lakukan langkah berikut:
 - Cetak kata
 - Tambahkan i dengan 1
- (i) Selama $i < \text{jumlah}$, ulang langkah 8.
- (j) Selesai.

2) Kode program dan luaran

a) Kode program,komentar dan hasil luaran



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

Console Output:

```
<terminated> ForBersarangSevi [Java Application] C:\Users\CACCO\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.12.v20240802-1518\jre\bin\javaw.exe (Oct 4, 2024, 6:15:09 PM - 6:15:34 PM) [pid: 17140]
Masukkan Kata yang ingin diulang: Falkutas Teknik
Masukkan Jumlah yang ingin diulang: 0
```

Kode program while

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

Console Output:

```
<terminated> ForBersarangSevi [Java Application] C:\Users\CACCI\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.12.v20240802-1518\jre\bin\javaw.exe (Oct 4, 2024, 6:16:14 PM - 6:16:30 PM) [pid: 9472]
Masukkan Kata yang ingin diulang: Fakultas Teknik
Masukkan Jumlah yang ingin diulang: 0
Fakultas Teknik
```

Kode program do while

- b) Luaran yang dihasilkan sudah benar dan sesuai dengan soal yang diminta.
- Ketika program menggunakan while dengan jumlah pengulangan yang dimasukkan adalah 0, program tidak akan mencetak apa pun. Hal ini disebabkan oleh fakta bahwa kondisi awal loop $i < \text{jumlah}$ tidak terpenuhi (karena $0 < 0$ adalah salah). Oleh karena itu, blok kode di dalam loop while tidak akan dieksekusi sama sekali. Sebagai hasilnya, tidak ada output yang dihasilkan, dan program akan langsung selesai setelah meminta input dari pengguna. Sebaliknya, ketika menggunakan do while, program akan mencetak kata setidaknya satu kali meskipun jumlah pengulangan yang dimasukkan adalah 0. Ini karena blok kode di dalam do dieksekusi terlebih dahulu sebelum kondisi diperiksa. Setelah kata dicetak, nilai i ditambahkan 1. Kemudian, kondisi $i < \text{jumlah}$ diperiksa (di mana jumlah adalah 0), dan loop akan berhenti. Oleh karena itu, dengan jumlah = 0, output program akan mencetak kata satu kali.

[No 2.2] Kesimpulan

1) Analisa

Kode program yang menggunakan while dan do while, terdapat perbedaan mendasar dalam cara kedua loop tersebut beroperasi, terutama ketika jumlah pengulangan yang dimasukkan adalah 0. Dalam while, kondisi diperiksa sebelum loop dijalankan, sehingga jika jumlah pengulangan adalah 0, loop tidak akan dijalankan sama sekali dan tidak ada output yang dihasilkan. Sebaliknya, pada do while, blok kode akan dieksekusi setidaknya sekali sebelum kondisi diperiksa. Akibatnya, meskipun jumlah pengulangan 0, program tetap mencetak kata yang diinput oleh pengguna satu kali. Perbedaan ini menegaskan bahwa do while lebih cocok digunakan dalam situasi di mana aksi perlu dilakukan setidaknya sekali, sedangkan while lebih sesuai jika aksi hanya perlu dilakukan ketika kondisi awal terpenuhi.

[No 2.3 & 4] Identifikasi Masalah:

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 = $(\text{nilai pelajaran} + \text{nilai rata-rata}) / 2$

[8] Tambah satu ke idPelajaran

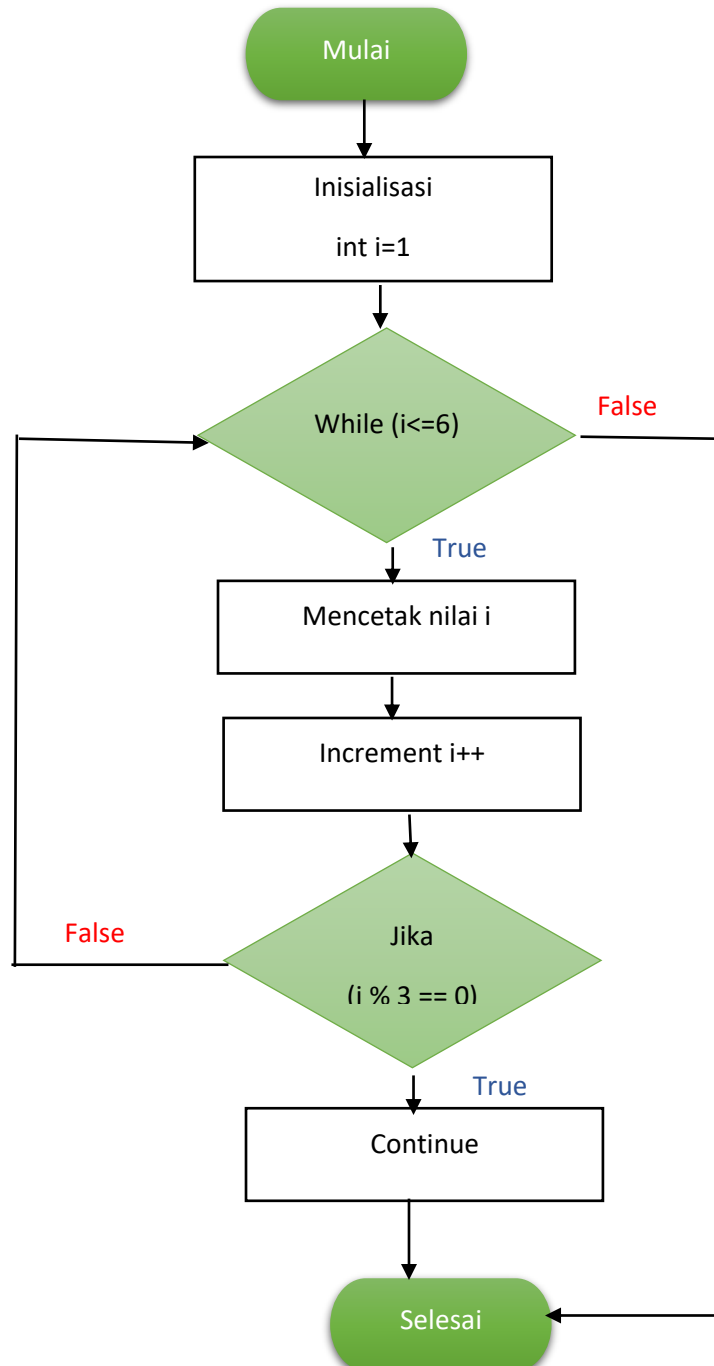
[9] Tampilkan nilai rata-rata

Rekomendasikan kode untuk menyelesaikan Pseudocode tersebut!

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

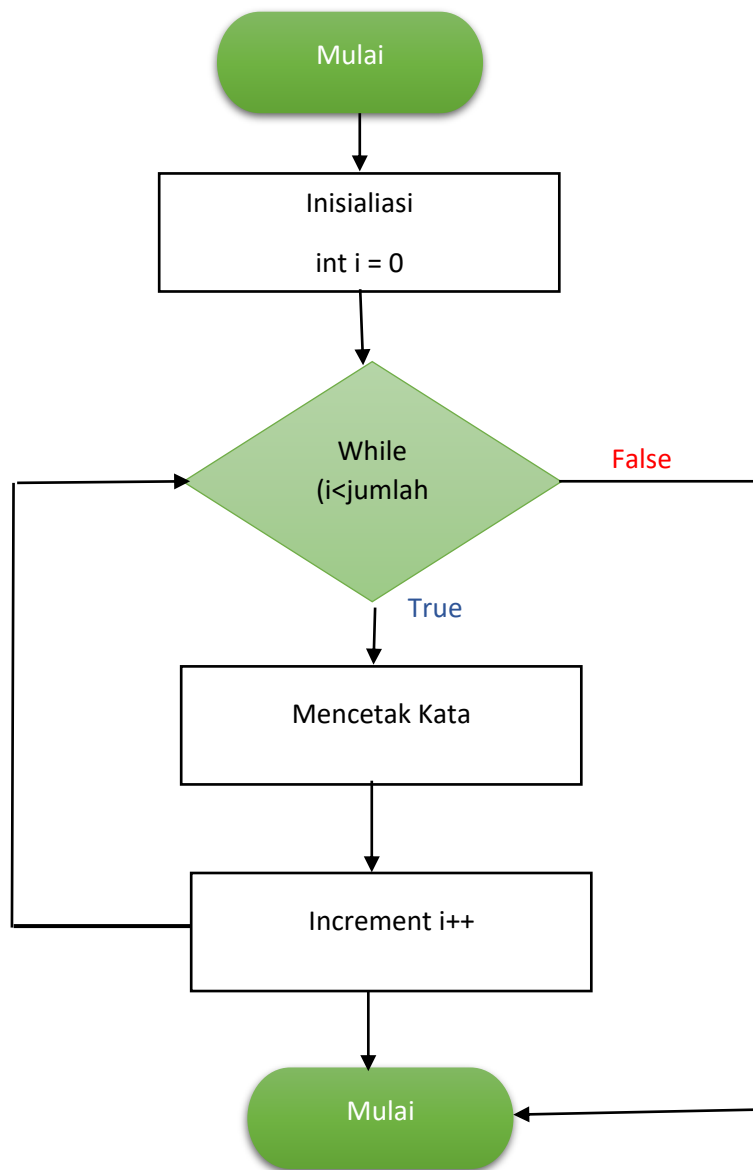
Jawaban:

Flowchat 2.1

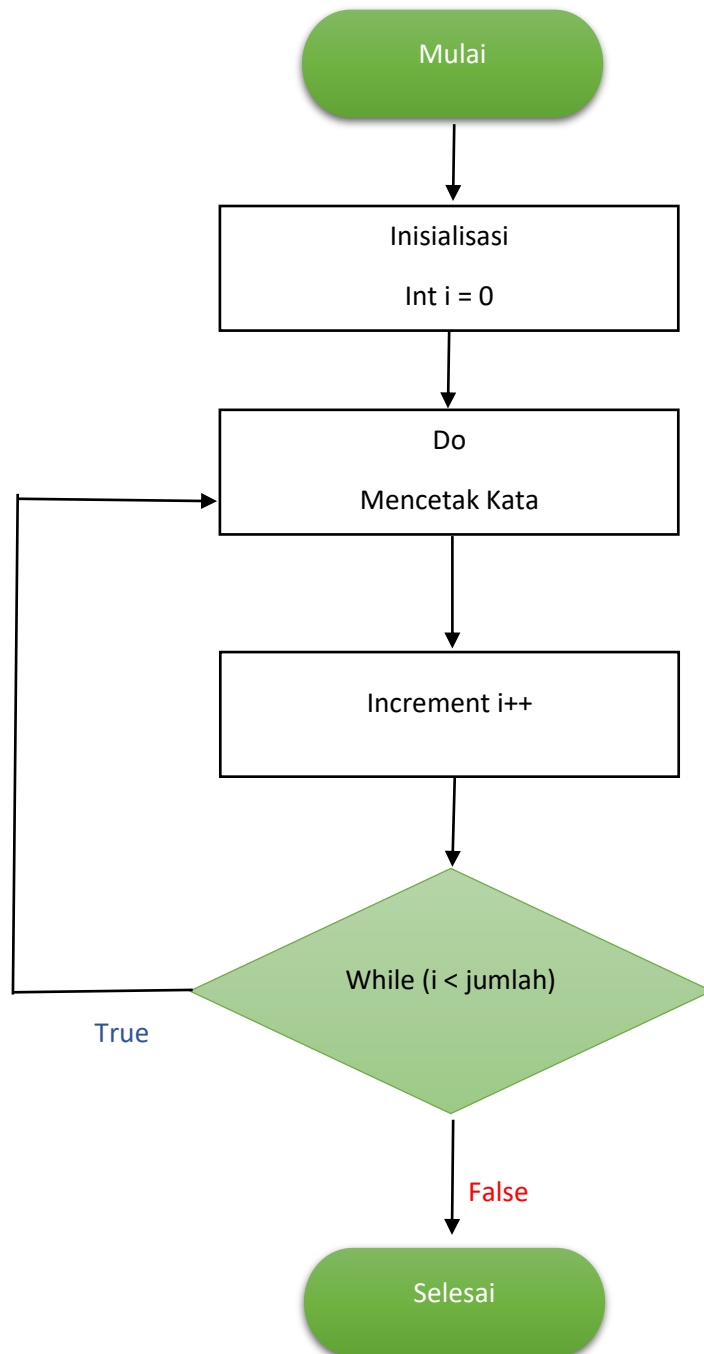


Flowchat 2.2

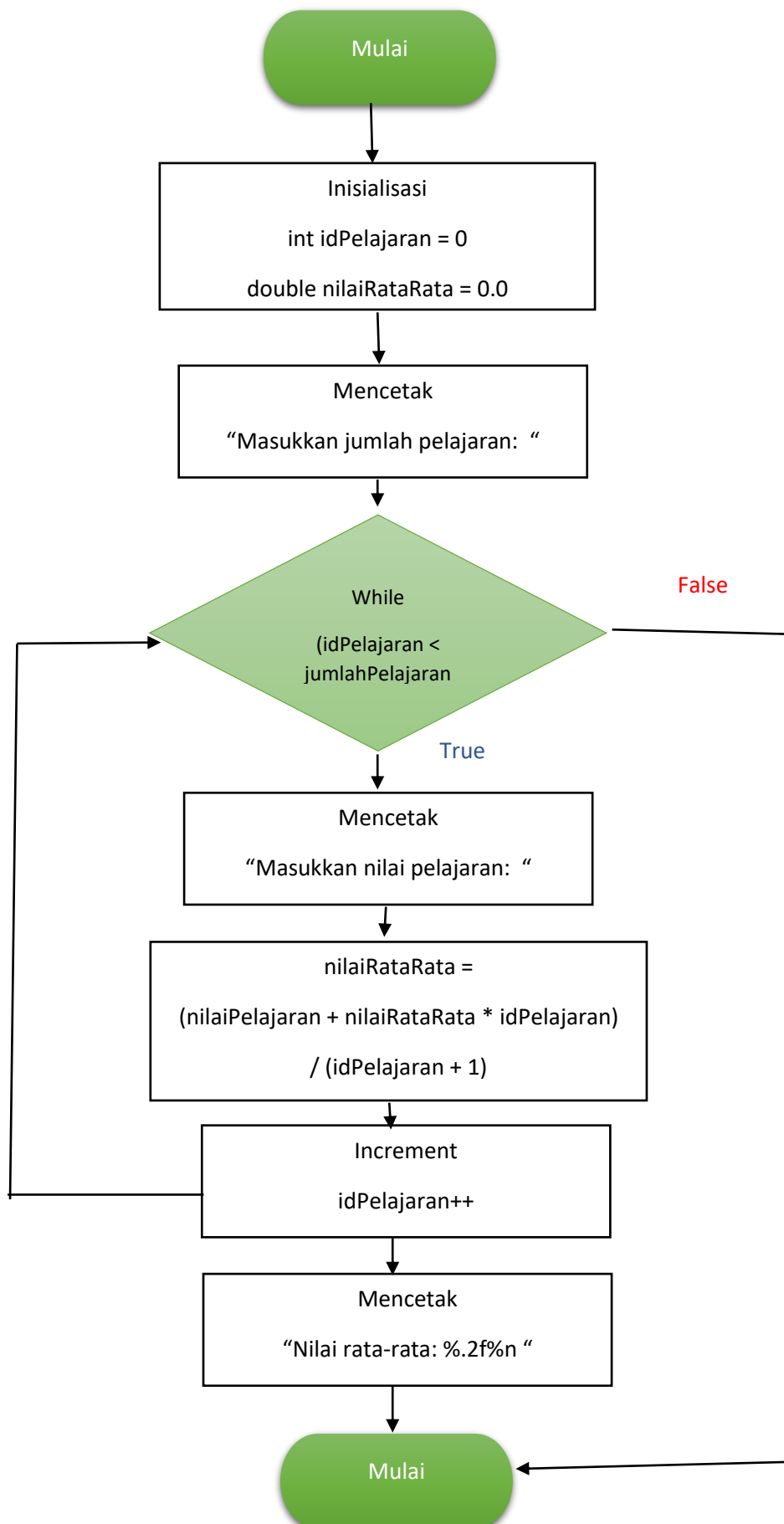
Flowchat untuk kode program while:



Flowchat untuk kode program do while:



Flowchat 2.3



[No 2.3] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara menggunakan pendekatan berbasis input dan perhitungan yang sistematis melalui pemrograman. Dengan menggunakan bahasa pemrograman seperti Java, kita dapat membuat program yang memungkinkan pengguna untuk memasukkan jumlah pelajaran dan nilai-nilai terkait, serta menghitung rata-rata secara otomatis.
- 2) Alasan solusi ini karena pendekatan berbasis pemrograman tidak hanya memudahkan pengguna dalam melakukan perhitungan rata-rata, tetapi juga mengurangi kemungkinan kesalahan manusia dalam perhitungan manual. Selain itu, dengan menggunakan program, pengguna dapat dengan cepat menghitung rata-rata nilai untuk berbagai jumlah pelajaran tanpa harus mengulangi perhitungan secara manual setiap kali. Ini membuat solusi ini efisien dan praktis, terutama ketika jumlah pelajaran yang harus dihitung cukup besar.
- 3) Perbaikan kode program dilakukan dengan cara:
 - a. Memastikan bahwa input yang diterima dari pengguna divalidasi dengan baik untuk menghindari kesalahan, seperti memasukkan nilai yang tidak valid (misalnya, huruf atau angka negatif).
 - b. Menggunakan rumus yang benar untuk menghitung rata-rata, yang memperhitungkan semua nilai yang telah dimasukkan sebelumnya tanpa mengurangi akurasi perhitungan.
 - c. Menambahkan komentar yang jelas dan terstruktur dalam kode untuk meningkatkan pemahaman tentang fungsi masing-masing bagian dari program, sehingga mudah dipahami oleh orang lain yang mungkin membaca atau memodifikasi kode di masa depan.
 - d. Mengoptimalkan penggunaan memori dengan menutup objek Scanner setelah selesai digunakan.

[No 2.3] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Mulai
 - (b) Inisialisasi:
 - $idPelajaran = 0$
 - $nilaiRataRata = 0$
 - (c) Input:
 - Minta pengguna memasukkan jumlah pelajaran dan disimpandi $jumlahPelajaran$
 - (d) Perulangan selama $idPelajaran < jumlahPelajaran$
 - Minta pengguna memasukkan nilai pelajaran ke $idPelajaran + 1$
 - Update $nilaiRataRata = (nilaiRataRata * idPelajaran + nilaiPelajaran) / (idPelajaran + 1)$
 - Tambah 1 ke $idPelajaran$
 - (e) Output tampilkan $nilaiRataRata$
 - (f) Program selesai
- 2) Kode program dan luaran
 - a) Kode program, komentar dan hasil luaran

The screenshot displays the Eclipse IDE environment. The main editor window shows a Java file named `NilaiSevi.java` with the following code:

```
1 import java.util.Scanner;
2 public class NilaiSevi {
3     public static void main(String[] args) {
4         // Inisiasi variabel
5         int idPelajaran = 0;
6         double nilaiRataRata = 0.0;
7
8         // Scanner untuk input dari pengguna
9         Scanner scanner = new Scanner(System.in);
10
11         // Minta pengguna untuk menuliskan jumlah pelajaran
12         System.out.print("Masukkan jumlah pelajaran: ");
13         int jumlahPelajaran = scanner.nextInt();
14
15         // Mengulangi sampai idPelajaran kurang dari jumlah pelajaran
16         while (idPelajaran < jumlahPelajaran) {
17             // Minta pengguna untuk menuliskan nilai pelajaran
18             System.out.print("Masukkan nilai pelajaran ke-" + (idPelajaran + 1) + ": ");
19             double nilaiPelajaran = scanner.nextDouble();
20
21             // Hitung nilai rata-rata
22             nilaiRataRata = (nilaiPelajaran + nilaiRataRata * idPelajaran) / (idPelajaran + 1);
23
24             // Tambah satu ke idPelajaran
25             idPelajaran++;
26         }
27
28         // Tampilkan nilai rata-rata
29         System.out.printf("Nilai rata-rata adalah: %.2f\n", nilaiRataRata);
30
31         // Tutup scanner
32         scanner.close();
33     }
34 }
35
```

The console window below the editor shows the program's execution:

```
<terminated> NilaiSevi [Java Application] C:\Users\CACCA\p2\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.12.x20240802-1518\jre\bin\javaw.exe (Oct 4, 2024, 6:52:58 PM - 6:53:22 PM) [pid: 21256]
Masukkan jumlah pelajaran: 3
Masukkan nilai pelajaran ke-1: 90
Masukkan nilai pelajaran ke-2: 85
Masukkan nilai pelajaran ke-3: 95
Nilai rata-rata adalah: 90.00
```

- b) Luaran yang dihasilkan sudah benar dan sesuai dengan soal yang diminta. Dengan input jumlah pelajaran sebanyak 3, pengguna diminta untuk memasukkan tiga nilai pelajaran. Nilai rata-rata dihitung secara bertahap setelah setiap input. Akhirnya, program menampilkan nilai rata-rata dari ketiga nilai yang telah dimasukkan. Dalam contoh ini, nilai rata-rata yang diperoleh adalah 90.00.

[No 2.3] Kesimpulan

1) Analisa

Kode program ini efektif dalam menghitung rata-rata nilai pelajaran berdasarkan input pengguna, memungkinkan pengguna untuk memasukkan jumlah pelajaran dan nilai-nilai terkait dengan mudah. Algoritma yang terstruktur memastikan setiap nilai diperhitungkan dengan benar dalam perhitungan rata-rata. Hasil dari contoh perhitungan dengan tiga pelajaran menunjukkan keandalan program dalam memberikan nilai rata-rata yang tepat.

Refleksi

Pengalaman belajar saya pada materi ke 5 ini sangat banyak, yaitu saya dapat membuat kode program dengan for, while dan do while. Kode program adalah proses perulangan blok kode dengan jumlah angka yang sudah ditentukan, kode program while adalah proses perulangan blok kode dengan jumlah angka tidak terhitung lalu proses pengulangan blok pernyataan dilakukan selama kondisinya bernilai true dan kode program do while adalah proses perulangan blok kode dengan jumlah angka tidak terhitung dan proses pengulangan suatu blok pernyataan dieksekusi paling tidak satu kali dan selama kondisinya bernilai true. Tantangan saya mungkin tidak terlalu banyak hanya terkadang lupa menambahkan titik koma dan import java.util.scanner pada awal membuat program lalu harus lebih lama dalam membuat flowchat. Pengalaman belajar ini sangat berharga karena dapat membantu saya dalam memahami materi for dan while ini.