

Nama & NPM	Topik:	Tanggal:
Yulinda Samosir G1F024020	For dan While	5 Oktober 2024

Latihan 1. For

[No. 1] Identifikasi Masalah:

- 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) analisis kode program pada contoh 1 terdapat kesalahan sintaks pada pernyataan for. Lalu terdapat kesalahan pada penggunaan `<=` dan `=` karena tidak sesuai dalam konteks ini. Kode `for (double y <= 15; y = 0; y++) {` diubah menjadi `for (int y = 0; y <= 15; y++) {` agar penggunaanya benar dan berjalan.
- 2) Pada Contoh 2, karena soal meminta luaran

i = 1; j = 1

i = 1; j = 2

i = 2; j = 1

i = 2; j = 2

Maka kode program di awal tadi kita ubah untuk menghasilkan luaran yang diinginkan soal, yaitu dari pertama: `for(int i = 1; i < 5; i++)` kedua: `for(int j = 1; j < 3; j ++ System.out.println("i = " + i + "; j = " +j));` diubah menjadi pertama: `for(int i = 1; i < 3; i++) {` kedua: `for(int j = 1; j < 3; j ++) { System.out.println("i = " + i + "; j = " +j);` maka setelah ini luaran yang dihasilkan akan sesuai.

- 3) Pada contoh 3, karena soal meminta luaran berbentuk piramida:

Masukan Input: 7

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

Maka kode program di awal tadi kita ubah untuk menghasilkan luaran yang diinginkan soal, yaitu diubah menjadi `for (int t = 1; t <= tinggi; t++) { // Menghitung Jumlah Spasi per` dari `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("*");` Baris `for (int s = tinggi; s > t; s--) { System.out.print(" "); // Mencetak spasi } // Menghitung Jumlah Bintang per Baris` `for (int b = 1; b <= (2 * t - 1); b++) {` . Dengan perubahan kode program yang dilakukan, maka luaran yang diinginkan soal akan berhasil.

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

- (a) Deklarasikan variabel y sebagai integer.
- (b) Buat loop dari y = 0 hingga y <= 15 dengan langkah 1.
- (c) Jika y adalah angka ganjil (y % 2 == 1):
- (d) Lanjutkan ke iterasi berikutnya (skip output).
- (e) Jika y sama dengan 8:
- (f) Cetak "Eight".
- (g) Jika tidak memenuhi kondisi di atas:
- (h) Cetak nilai y.
- (i) Selesai

2). Algoritma Contoh 2

- (a) Mulai program.
- (b) Buat loop Pertama dari i = 1 hingga i < 5 dengan langkah 1.
- (c) Buat loop Kedua dari j = 1 hingga j < 3 dengan langkah 1.
- (d) Cetak "i = i; j = j".

- (e) Jika i sama dengan 2: lanjutkan ke iterasi berikutnya dari loop i.
- (f) selesai

3). Algoritma Ketiga

- (a) Mulai program
- (b) Buat loop pertama dari i = 1 hingga i < 5 dengan langkah 1
- (c) Buat loop Kedua dari j = 1 hingga j < 3 dengan langkah 1
- (d) Cetak "i = i; j = j"
- (e) Jika i sama dengan 2: lanjutkan ke iterasi berikutnya dari loop i
- (f) Selesai.

2) Kode program dan luaran

Main.java	Output
<pre> 1- public class ContohFor{ 2- public static void main(String[] args) { 3- for (int y = 0; y < 10; y++) { 4- if (y % 2 == 1) { //kondisi 1 5- // baris kode kosong 1 6- } else if (y == 8) { //kondisi 2 7- // baris kode kosong 2 8- } else 9- System.out.println(y + " "); 10 } } } </pre>	<pre> java -cp /tmp/smBS50EnDX/ContohFor 0 2 4 6 === Code Execution Successful === </pre>

codingan 1.1

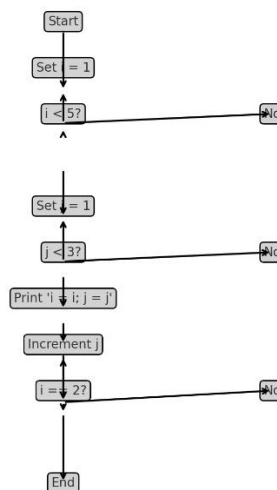
Main.java	Output
<pre> 1- public class ForBersarang { 2- public static void main(String[] args) { 3- pertama: 4- for(int i = 1; i < 3; i++) { 5- kedua: 6- for(int j = 1; j < 3; j ++) { 7- System.out.println("i = " + i + "; j = " +j); 8- } 9- if (i == 2) { 10- // kode yang hilang 11- } } } } </pre>	<pre> java -cp /tmp/rJZjE7OREq/ForBersarang i = 1; j = 1 i = 1; j = 2 i = 2; j = 1 i = 2; j = 2 === Code Execution Successful === </pre>

Codingan 1.2

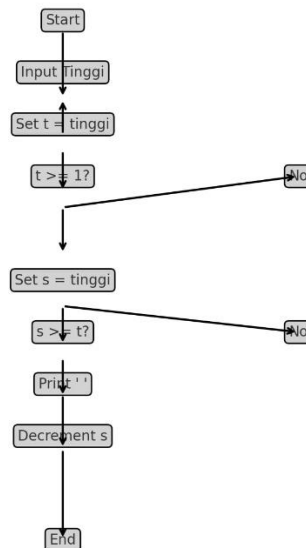
Main.java	Output
<pre> 1- import java.util.Scanner; 2- 3- public class ForBersarang { 4- public static void main(String[] args) { 5- // Instance Input Scanner 6- Scanner input = new Scanner(System.in); 7- System.out.print("Masukan Input: "); 8- int tinggi = input.nextInt(); // Mendapatkan Input Dari User 9- 10- for (int t = 1; t <= tinggi; t++) { 11- // Menghitung Jumlah Spasi per Baris 12- for (int s = tinggi; s > t; s--) { 13- System.out.print(" "); // Mencetak spasi 14- } 15- // Menghitung Jumlah Bintang per Baris 16- for (int b = 1; b <= (2 * t - 1); b++) { 17- System.out.print("*"); // Mencetak bintang 18- } 19- System.out.println(); // Membuat Baris Baru 20- } 21- } 22- } </pre>	<pre> java -cp /tmp/2jbhUbljs/ForBersarang Masukan Input: 7 * *** ***** ***** ***** ***** ***** === Code Execution Successful === </pre>

Codingan 1.3

Flowchart Contoh 2: ForBersarang



Flowchart Contoh 3: Piramida



a) Analisa luaran yang dihasilkan

Luaran yang dihasilkan sudah sesuai dengan program yang disusun.

Kondisi For yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

[No.1] Kesimpulan

Analisa

a) Susunlah kesimpulan!

Kesimpulan dari semua kondisi if yang telah dibahas adalah

Contoh 1: Menggunakan loop untuk mencetak angka genap dari 0 hingga 15, dengan penanganan khusus untuk angka 8. Kesalahan sintaks diperbaiki untuk menghasilkan keluaran yang diinginkan.

Contoh 2: Mencetak pasangan nilai i dan j dalam loop bersarang, dengan penggunaan pernyataan continue untuk menghindari pencetakan ketika i sama dengan 2.

Contoh 3: Menggunakan loop bersarang untuk menggambar piramida bintang berdasarkan tinggi yang dimasukkan oleh pengguna, dengan penanganan yang tepat untuk mencetak spasi dan bintang. Jadi kesimpulannya melalui analisis dan perbaikan kode, kita bisa melihat dan mengamati tentang pentingnya sintaks yang benar, penggunaan loop dan pengkondisian, serta bagaimana mendesain algoritma yang efisien untuk menyelesaikan masalah tertentu.

Latihan 2. WHILE

[No. 1] Identifikasi Masalah:

Uraikan permasalahan dan variabel!

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.1] Analisis dan Argumentasi

2.1 Analisis menghapus break pada contoh 3

- 1) Analisis kode program pada contoh 4, perubahan kode program awal dan setelah di ubah memperlihatkan hasil luaran yang sangat berbeda. Bisa kita liat pada kode program awal tadi menghasilkan keluaran 1,2,3, namun setelah kode programnya di ubah menghasilkan keluaran 1,2,4,5. Hal tersebut dikarenakan penggunaan kode program if (i % 3 == 0) dan continue, yang dimana kode program ini melewati angka kelipatan 3, yang artinya kode tersebut menunjukkan bagaimana penggunaan continue dapat mengontrol alur eksekusi dalam loop.
- 2) Analisis kode program pada contoh 5, perbedaan hasil keluaran while ketika dieskusi dengan jumlah yang diulang 0, dengan hasil keluaran do while Ketika di eksekusi dengan jumlah yang diulang 0 sangat berbeda. Bisa kita liat pada kode program while ketika kita masukkan jumlah perulangannya 0 program tidak menampilkan apa pun karena kondisi while (count > 0) tidak terpenuhi. Sedangkan kode program do while ketika kita masukkan jumlah perulangan 0 blok kode di do akan di jalankan setidaknya sekali.
- 3) Untuk rekomendasi Kode Pseudocode, kita menggunakan import scanner, inisialisasi, input jumlah Pelajaran, loop(while).

[No.1] Penyusunan Algoritma dan Kode Program

- 1). Algoritma Contoh 2.1
 - (a) Buat variabel i dan set nilainya ke 1.
 - (b) Gunakan loop for atau while untuk iterasi dari 1 hingga 10.
 - (c) Tentukan kondisi loop (i <= 10).

- (d) di dalam loop, tambahkan pernyataan if untuk memeriksa apakah i adalah kelipatan 3 ($i \% 3 == 0$). Jika benar, gunakan continue untuk melewati sisa iterasi dan kembali ke langkah awal loop.
- (e) Jika i bukan kelipatan 3, cetak nilai i.
- (f) Tambahkan 1 ke i setelah setiap iterasi.
- (g) Setelah loop selesai, program berakhir

2). Algoritma 2.2

- (a) Buat variabel count dan set nilainya ke 0.
- (b)Gunakan loop while dengan kondisi $count > 0$.
- (c) Di dalam loop, tambahkan pernyataan untuk mencetak "Count: count".
- (d) Tambahkan pernyataan untuk mengurangi nilai count sebesar 1
- (e) Karena nilai count dimulai dari 0, loop tidak akan pernah dieksekusi, sehingga program langsung berakhir tanpa output

2). Kode program dan luaran

Main.java	Output
<pre> 1- public class ContohWhile { 2- public static void main(String[] args) { 3- int i = 1; 4- while (i <= 6) { 5- if (i % 3 == 0) { // Ubah1: Cek apakah i adalah kelipatan 6- i++; // Jangan lupa untuk increment i 7- continue; // Melewatkan iterasi jika i adalah kelipatan 3 8- } 9- System.out.println(i); // Cetak nilai i jika bukan kelipatan 3 10- i++; 11- } 12- } 13- } 14- </pre>	<pre> java -cp /tmp/jaAUsDwkFE/ContohWhile 1 2 4 4 5 === Code Execution Successful === </pre>

Codingan 2.1

Main.java	Output
<pre> 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- while(i < jumlah){ 15- System.out.println(kata); 16- i++; //Faktor pengulang Increment 17- } 18- } 19- } 20- </pre>	<pre> java -cp /tmp/lbzAmKd5I2/ForBersarang Masukkan Kata yang ingin diulang : Fakultas Teknik Masukkan Jumlah ingin diulang : 0 === Code Execution Successful === </pre>

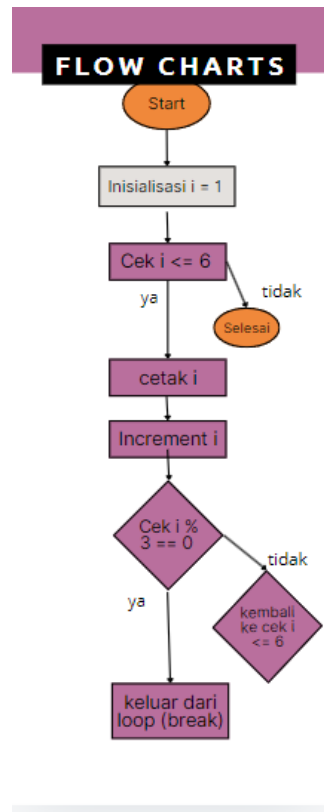
Codingan 2.2 While

Main.java	Output
<pre> 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 System.out.print("Masukkan Jumlah ingin diulang : "); 10 int jumlah = dataKata.nextInt(); // Menggunakan Scanner yang sama 11 12 int i = 0; // Inisialisasi batas dasar 13 do { 14 System.out.println(kata); 15 i++; // Faktor pengulang Increment 16 } while(i < jumlah); 17 } 18 } 19 20 </pre>	<pre> java -cp /tmp/UwikhNssel/ForBersarang Masukkan Kata yang ingin diulang : Fakultas Teknik Masukkan Jumlah ingin diulang : 0 Fakultas Teknik === Code Execution Successful === </pre>

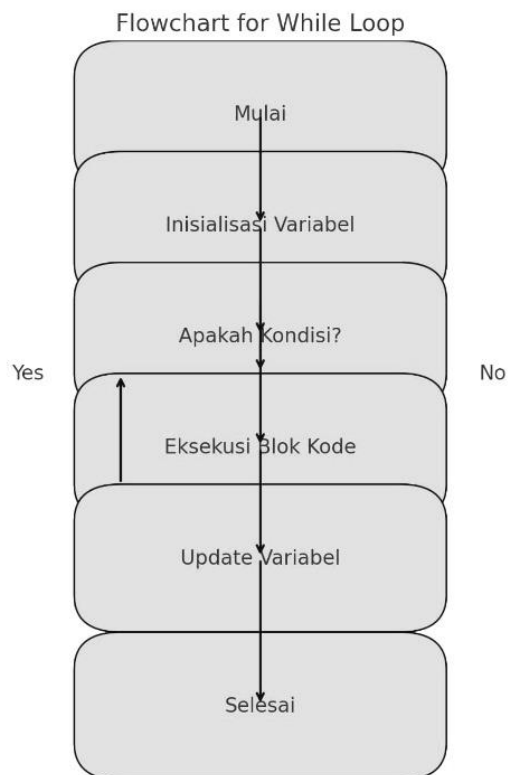
Codingan 2.2 Do While

Main.java	Output
<pre> 1- import java.util.Scanner; 2 3- public class RataRataNilai { 4- public static void main(String[] args) { 5 Scanner scanner = new Scanner(System.in); 6 7 // Inisiasi variabel 8 int idPelajaran = 0; 9 double nilaiPelajaran = 0; 10 double nilaiRataRata = 0; 11 12 // Minta pengguna untuk menuliskan jumlah pelajaran 13 System.out.print("Masukkan jumlah pelajaran: "); 14 int jumlahPelajaran = scanner.nextInt(); 15 16 // Ketika idPelajaran lebih kecil dari jumlah pelajaran 17 while (idPelajaran < jumlahPelajaran) { 18 // Minta pengguna untuk menuliskan nilai pelajaran 19 System.out.print("Masukkan nilai pelajaran ke-" + 20 (idPelajaran + 1) + ": "); 21 nilaiPelajaran = scanner.nextDouble(); 22 23 // Hitung nilai rata-rata 24 nilaiRataRata = (nilaiPelajaran + (nilaiRataRata * 25 idPelajaran)) / (idPelajaran + 1); 26 27 // Tambah satu ke idPelajaran 28 idPelajaran++; 29 30 // Tampilkan nilai rata-rata 31 System.out.printf("Nilai rata-rata: %.2f\n", nilaiRataRata); 32 33 scanner.close(); 34 } 35 } 36 } </pre>	<pre> java -cp /tmp/3xxwGGJmB4/RataRataNilai Masukkan jumlah pelajaran: 3 Masukkan nilai pelajaran ke-1: 80 Masukkan nilai pelajaran ke-2: 85 Masukkan nilai pelajaran ke-3: 87 Nilai rata-rata: 84.00 === Code Execution Successful === </pre>

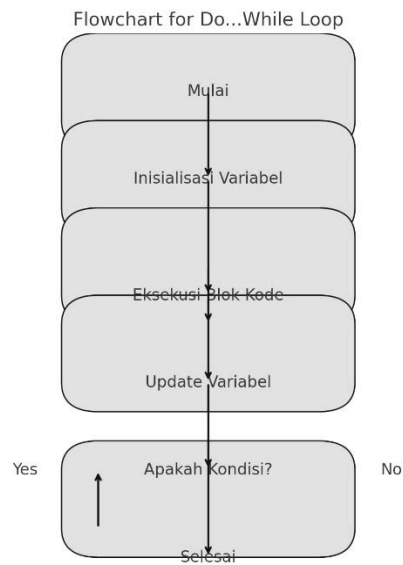
Codingan 2.3



Flowchart 2.1

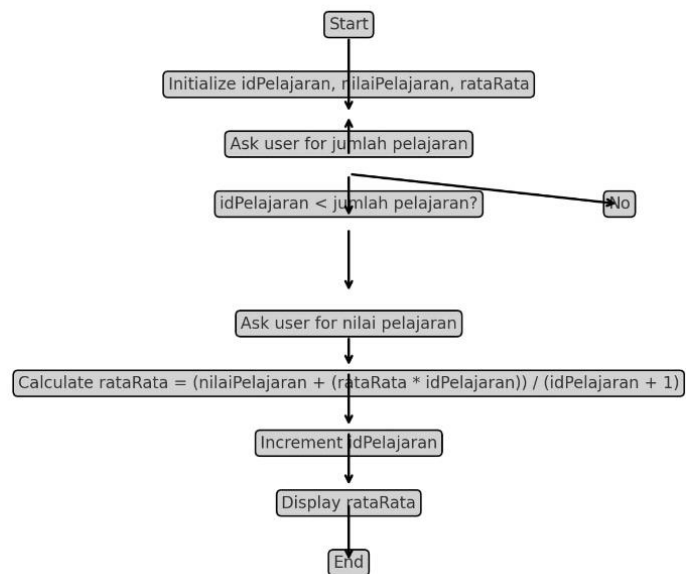


Flowchart while



Flowchart do while

Flowchart: Rata-rata Nilai Pelajaran



a. Analisa luaran yang dihasilkan:

Luaran yang dihasilkan sudah sesuai dengan program yang disusun.

Kondisi While yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

[No.1] Kesimpulan

Analisa

A. Susunlah kesimpulan!

Kesimpulannya pada contoh 4 diatas bahwa break itu fungsinya untuk menghentikan perulangan sepenuhnya Ketika kondisi terpenuhi, sedangkan continue fungsinya untuk melewati iterasi saat kondisi terpenuhi tapi tetap melanjutkan perulangan untuk nilai berikutnya. Lalu pada contoh 5 diatas tentang penggunaan while dan do while. While itu fungsinya untuk mengecek kondisi terlebih dahulu sebelum menjalankan perintah di dalam loop. Jika kondisi tidak terpenuhi perintah tidak akan dijalankan sama sekali. Sedangkan do while menjalankan perintah di dalam loop setidaknya sekali, meskipun kondisi tidak terpenuhi. Kemudian untuk rekomendasi kode Pseudocode diatas, hal tersebut menggambarkan proses pengambilan input dari pengguna, perulangan untuk menghitung nilai rata-rata, dan kemudian menampilkan hasilnya.