

Nama & NPM	Topik:	Tanggal:
Michelia Erza Annadhira G1F024035	FOR Java	26 September 2024

[Nomor 1] Identifikasi Masalah:

1. Uraikan permasalahan dan variabel

Contoh 1: Salin dan tempel kode program berikut ke Eclipse.

```
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 + " ");
    } } }
```

Luaran:

Exception in thread "main" java.lang.Error: Unresolved compilation problems:

Syntax error on token "<=", = expected

Type mismatch: cannot convert from double to boolean

at ContohFor.main(ContohFor.java:5)

Contoh 2: Salin dan tempel kode program berikut ke Eclipse.

```
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: Salin dan tempel kode program berikut ke Eclipse.

```
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
}
}
}

```

Luaran:

Masukan Input: 7

```

*
**
***
****
*****
*****
*****

```

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!

2. Rincikan sumber informasi yang relevan (buku / webpage)

Video penjelasan pembelajaran dapat diakses pada [Chanel Youtube Rumah Ilmu](#)

[Raflesia](https://www.youtube.com/channel/UC8B9rghd3dBiS6OKonLMyIw)

<https://www.youtube.com/channel/UC8B9rghd3dBiS6OKonLMyIw>

[Video Materi 1 tentang FOR – https://www.youtube.com/watch?v=Ij9qLLblxEU](https://www.youtube.com/watch?v=Ij9qLLblxEU)

[Video Materi 2 tentang WHILE – https://www.youtube.com/watch?v=ORA4JyJMFss](https://www.youtube.com/watch?v=ORA4JyJMFss)

[Nomor 2] Analisis dan Argumentasi

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
```

- a. Saya mengusulkan permasalahan ini dapat diatasi dengan cara merevisi struktur loop dan logika kondisionalnya agar sesuai dengan tujuan, yaitu print bilangan genap dari 0 sampai 6
- b. Alasan solusi ini karena masalah utama ada pada urutan parameter dalam loop dan tipe data yang kurang tepat untuk bilangan bulat. Selain itu, logika yang memeriksa bilangan ganjil dan nilai tertentu tidak sesuai dengan hasil yang diminta
- c. Perbaiki kode program dengan cara :
 - Menggunakan tipe data 'int' untuk variabel 'y' karena kita hanya bekerja dengan bilangan bulat
 - Memulai iterasi dengan 'y = 0', menetapkan kondisi 'y <= 6' agar iterasi berhenti pada 6, dan menggunakan 'y++' untuk penambahan
 - Menggunakan pernyataan 'if (y % 2 == 0)' untuk memeriksa dan print bilangan genap

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
```

- a. Saya mengusulkan untuk menyelesaikan masalah ini dengan menambahkan perintah 'break' yang digunakan untuk keluar dari loop utama setelah 'i' mencapai nilai 2.
- b. Alasan solusi ini karena untuk mencegah loop luar melanjutkan ke iterasi berikutnya, yang akan membuat 'i' menjadi 3 dan seterusnya. Dengan menghentikan loop setelah 'i' bernilai 2, kita bisa mendapatkan hasil yang

diinginkan tanpa iterasi tambahan yang tidak perlu.

- c. Perbaikan kode dapat dilakukan dengan menambahkan 'break' di dalam blok 'if' yang memeriksa apakah 'i' sama dengan 2.

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

Luaran berbentuk piramida

Masukan Input: 7

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

Rekomendasikan kode untuk menghasilkan luaran tersebut!

- a. Ubah loop pertama `'for(int t = tinggi; t >= 1; t--)'` menjadi `'for(int t = 1; t ,= tinggi; t++)'` untuk memastikan kita mulai dari baris teratas ke bawah.
- b. Ubah loop kedua `'for(int s = tinggi; s >= t; s--)'` menjadi `'for(int s = tinggi; s > t; s--)'` untuk mencetak spasi yang diperlukan untuk setiap baris.
- c. Tambahkan loop ketiga `'for(int b = 1; b <= (2 * t - 1); b++)'` untuk mencetak bintang dalam jumlah yang sesuai untuk membuat piramida.

[Nomor 3] Penyusunan Algoritma dan Kode Program

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
```

❖ Algoritma :

1. Mulai
2. Buat sebuah loop yang akan mengulangi blok kode di dalamnya. Kita mulai dengan 'y' yang diatur ke 0
3. Loop akan berjalan selama 'y' kurang dari atau sama dengan 15. Setiap kali melalui loop, 'y' akan bertambah satu
4. Periksa bilangan ganjil :
 - Kita cek apakah 'y' adalah bilangan ganjil. Kita lakukan ini dengan melihat apakah sisa pembagian 'y' dengan 2 adalah $(y \% 2 == 1)$
 - Jika 'y' adalah bilangan ganjil, kita tidak melakukan apa-apa dan langsung lanjut ke iterasi berikutnya

5. Cek apakah 'y' adalah 8 :

- Jika 'y' mencapai 8, kita keluar dari loop sepenuhnya. Itu artinya program berhenti memproses lebih jauh

6. Print bilangan genap :

- Jika 'y' adalah bilangan genap dan belum mencapai 8, kita print angka tersebut di layar, diikuti oleh spasi. Ini dilakukan untuk 'y' yang bernilai 0, 2, 4 dan 6

7. Setelah print, kita kembali ke awal loop dan menambahkan 1 ke 'y'. proses ini berlanjut hingga kita keluar dari loop

8. Ketika 'y' mencapai 8, kita keluar dari loop. Program tidak mencetak angka setelah 6 karena loop berhenti

9. Jalankan kode untuk memeriksa apakah program bisa dijalankan dengan baik dan tidak ada kesalahan

10. Simpan kode program yang telah diperbaiki atau catat perubahan yang telah dilakukan

11. Selesai

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

Luaran(Output) :

Output Generated Files

```
0
2
4
6
```

Compiled and executed in 1.859 sec(s)

Kode program sudah disusun dengan struktur yang benar. Hal ini dapat dilihat dari luaran (output) yang sesuai dengan kode program dan tidak terjadi error.

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
```

❖ Algoritma :

1. Mulai
2. Buat sebuah loop yang menggunakan variabel 'i'. nilainya dimulai dari 1 dan akan terus berputar selama 'i' kurang dari 5. Beri nama loop ini 'pertama' agar mudah diidentifikasi
3. Di dalam loop pertama, ada loop kedua yang menggunakan variabel 'j', mulai dari 1 hingga kurang dari 3. Loop ini menjalankan tugasnya untuk setiap nilai 'i'
4. Cetak kombinasi 'i' dan 'j'
5. Periksa kondisi
6. Keluar dari loop, kita hentikan loop utama dengan menggunakan 'break'.
7. Kalau 'i' belum mencapai 2, kita biarkan loop terus berjalan hingga syaratnya terpenuhi
8. Jalankan kode untuk memeriksa apakah program bisa dijalankan dengan baik dan tidak ada kesalahan
9. Simpan kode program yang telah diperbaiki atau catat perubahan yang telah dilakukan
10. Selesai

```
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 pertama; // kode yang hilang  
11            } } }  
}
```

Luaran(Output) :

Output Generated Files

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

Kode program suda disusun dengan struktur yang benar. Hal ini dapat dilihat dari luaran (output) yang sesuai dengan kode program dan tidak terjadi error.

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

Luaran berbentuk piramida

Masukan Input: 7

```

*
***
*****
*****
*****
*****
*****
*****

```

Rekomendasikan kode untuk menghasilkan luaran tersebut!

❖ Algoritma :

1. Mulai
2. Persiapan input
3. Baca input
4. Looping baris
5. Cetak spasi
6. Cetak bintang
7. Pindah ke baris baru
8. Jalankan kode untuk memeriksa apakah program bisa dijalankan dengan baik dan tidak ada kesalahan
9. Simpan kode program yang telah diperbaiki atau catat perubahan yang telah dilakukan
10. Selesai

```

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 Tinggi Piramida
12             for(int s=tinggi; s>t; s--){
13                 //Menghitung Jumlah Spasi per Baris
14                 System.out.print(" ");
15             }
16
17             for(int b=1; b<=(2*t-1); b++){
18                 //Menghitung jumlah bintang untuk membentuk piramida
19                 System.out.print("*");
20             }
21             System.out.println(); //Membuat Baris Baru
22         }
23     }
24 }

```

Luaran(Output) :

```

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

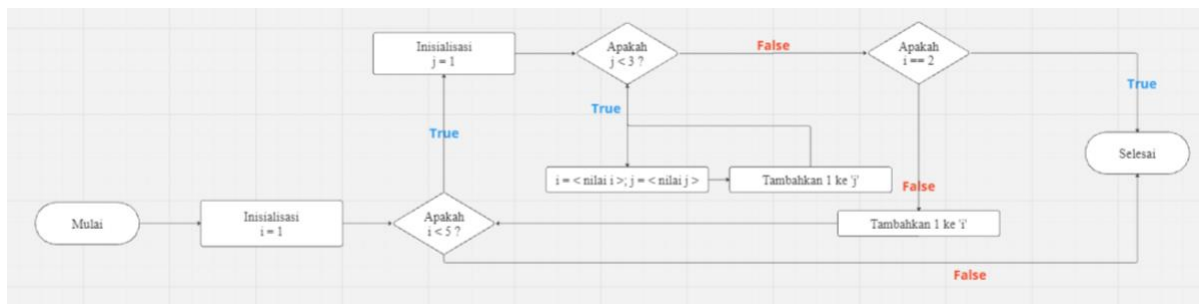
```

Compiled and executed in 4.269 sec(s)

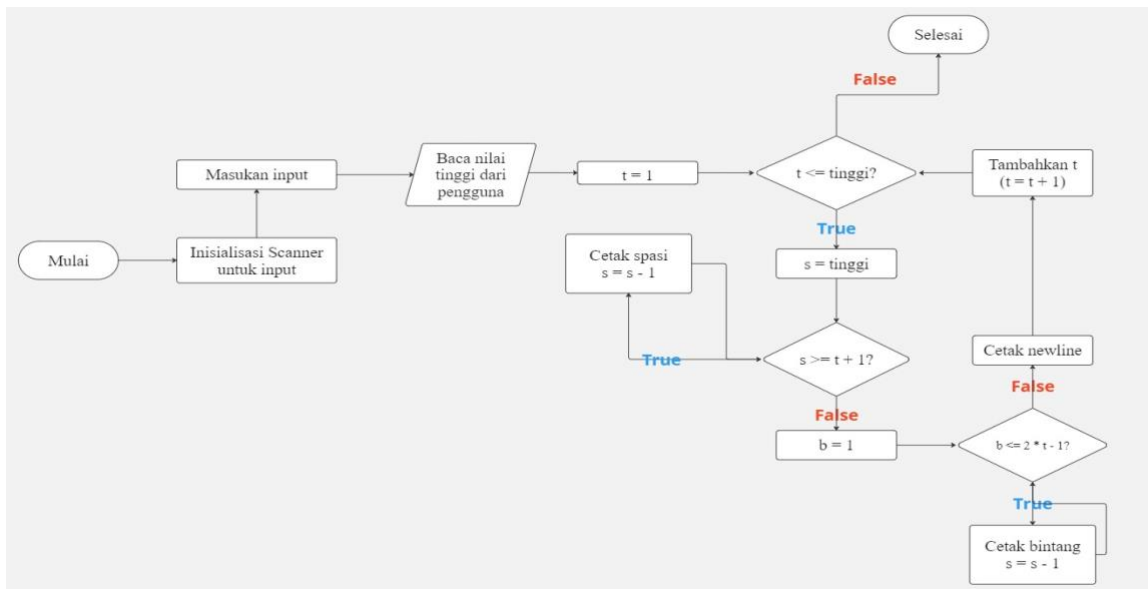
Kode program sudah disusun dengan struktur yang benar. Hal ini dapat dilihat dari luaran (output) yang sesuai dengan kode program dan tidak terjadi error.

1.4 Analisa diagram flowchart dari Latihan 1.2 dan 1.3!

❖ Flowchart Latihan 1.2



❖ Flowchart Latihan 1.3



[Nomor 4] Kesimpulan

- Pada program itu saya menggunakan 'for' loop karena struktur ini memungkinkan kontrol yang baik atas jumlah iterasi yang diinginkan dan memudahkan penyesuaian logika untuk menghasilkan pola yang diinginkan.
- Perbaiki program dengan menggunakan tipe data yang tepat dan logika kontrol ('continue' dan 'break') karena struktur Java mengharuskan penggunaan sintaks yang benar untuk menghindari kesalahan komplikasi dan mencapai hasil yang diinginkan.

Nama & NPM	Topik:	Tanggal:
Michelia Erza Annadhira G1F024035	WHILE Java	26 September 2024

[Nomor 1] Identifikasi Masalah:

1. Uraikan permasalahan dan variabel

Contoh 4: Salin dan tempel kode program berikut ke Eclipse.

```
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
```

Contoh 5: Salin dan tempel kode program berikut ke Eclipse.

```
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
Fakultas Teknik
Fakultas Teknik
Fakultas Teknik
```

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!

2. Rincikan sumber informasi yang relevan (buku / webpage)

Video penjelasan pembelajaran dapat diakses pada [Chanel Youtube Rumah Ilmu Raflesia](https://www.youtube.com/channel/UC8B9rghd3dBiS6OKonLMYIw)

<https://www.youtube.com/channel/UC8B9rghd3dBiS6OKonLMYIw>

[Video Materi 1 tentang FOR – https://www.youtube.com/watch?v=Ij9qLLblxEU](https://www.youtube.com/watch?v=Ij9qLLblxEU)

[Video Materi 2 tentang WHILE –](https://www.youtube.com/watch?v=ORA4JyJMFss)

<https://www.youtube.com/watch?v=ORA4JyJMFss>

[Nomor 2] Analisis dan Argumentasi

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!

- Awalnya, program mencetak angka 1 hingga 3 dan berhenti ketika mencapai 4 karena menggunakan 'break'. Ini menyebabkan perulangan berhenti sepenuhnya pada angka tersebut.
- Setelah modifikasi, kita mengganti kondisi '`if(i == 4)`' dengan '`if(i % 3`

`== 0)`’, dan ‘break’ dengan ‘continue’. Sekarang alih-alih menghentikan perulangan, kita melewati iterasi untuk angka yang merupakan kelipatan 3. Akibatnya, program mencetak semua angka dari 1 hingga 6, kecuali angka 3 dan 6 karena mereka dilewati. Perubahan ini mengubah cara kerja loop dari berhenti sepenuhnya pada angka tertentu menjadi hanya melewati angka tertentu, tetapi tetap melanjutkan perulangan.

- a. Saya mengusulkan permasalahan ini dapat diatasi dengan cara mengubah pendekatan kita terhadap perulangan. Alih-alih menghentikan perulangan sepenuhnya ketika mencapai angka tertentu, kita bisa melewatkan angka-angka yang tidak diinginkan.
- b. Alasan solusi ini karena kita ingin perulangan tetap berjalan sampai akhir, tetapi kita tidak ingin mencetak angka yang memenuhi kondisi tertentu, seperti kelipatan 3. Dengan begitu, kita bisa mengontrol alur perulangan tanpa harus menghentikannya.
- c. Perbaiki kode program dengan cara mengganti perintah ‘break’ dengan ‘continue’ dan memodifikasi kondisinya menjadi `if(i % 3 == 0)`. Dengan cara ini, kita bisa melewatkan angka-angka yang tidak diinginkan tanpa menghentikan perulangan secara keseluruhan.

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`!

- While, jika ‘jumlah’ adalah 0, kondisi ‘`i < jumlah`’ akan menjadi ‘false’ pada iterasi pertama (`0 < 0` adalah ‘false’). Makanya, blok kode di dalam ‘while’ tidak akan dieksekusi sama sekali dan tidak ada output yang dihasilkan.
 - Do. .while, blok kode di dalam ‘do’ akan dieksekusi setidaknya sekali, bahkan jika kondisi ‘while’ adalah ‘false’ pada iterasi pertama. Jadi, jika ‘jumlah’ adalah 0, program akan tetap mencetak ‘kata’ satu kali sebelum memeriksa kondisi ‘`i < jumlah`’.
- a. Jika menggunakan ‘while’, loop mengevaluasi kondisi sebelum mengeksekusi blok kode. Jika kondisi ‘false’ pada awalnya, blok kode tidak akan dieksekusi sama sekali.
 - b. Jika menggunakan ‘do. .while’, loop mengeksekusi blok kode setidaknya sekali sebelum mengevaluasi kondisi. Ini berarti blok kode akan dieksekusi satu kali, bahkan jika kondisi ‘false’ pada awalnya.
 - c. Perbedaan utama antara ‘while’ dan ‘do. .while’ adalah bahwa ‘do. .while’ menjamin bahwa blok kode akan dieksekusi setidaknya sekali, sedangkan ‘while’ mungkin tidak mengeksekusi blok kode sama sekali jika kondisi awalnya ‘false’.

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!

- a. Saya mengusulkan permasalahan ini dapat diatasi dengan cara pakai loop 'while' buat masukin nilai-nilai pelajarannya sampai semua nilai udah diinput.
- b. Alasan solusi ini karena loop 'while' itu gampang dan fleksibel banget buat urusan ngulang-ngulang. Kita bisa terus minta input nilai selama jumlah pelajaran yang kita mau belum tercapai.
- c. Perbaiki kode program dengan masukin loop 'while' buat ngumpulin nilai-nilai pelajaran. Kita mulai dari 'idPelajaran' yang kita set ke 0, terus kita tambahkan tiap kali kita dapat nilai dapet nilai baru, sambil kita jumlahin semua nilai itu ke 'totalNilai'. Setelah semua nilai masuk, kita tinggal bagi 'totalNilai' itu sama 'jumlahPelajaran' buat dapetin rata-ratanya.

[Nomor 3] Penyusunan Algoritma dan Kode Program

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!

❖ Algoritma :

1. Mulai
2. Mulai dengan angka 'i' yang bernilai 1
3. Periksa apakah masih di batas, selama 'i' masih kurang dari atau sama dengan 6
4. Cetak angka, tampilkan nilai 'i' ke layar
5. Tambah angka, naikan nilai 'i' dengan 1
6. Lompati kelipatan 3
7. Terus ulangi
8. Jalankan kode untuk memeriksa apakah program bisa dijalankan dengan baik dan tidak ada kesalahan
9. Simpan kode program yang telah diperbaiki atau catat perubahan yang telah dilakukan
10. Selesai

```
1 public class ContohWhile{
2     public static void main(String[] args) {
3         int i=1;
4         while(i<=6){
5             System.out.println(i);
6             i++;
7             if(i % 3 == 0){
8                 continue; //ubah1
9             }
10        }
11    }
12 }
```

Luaran(Output) :



```
1
2
3
4
5
6
|
```

Compiled and executed in 1.32 sec(s)

Kode program sudah disusun dengan struktur yang benar. Hal ini dapat dilihat dari luaran (output) yang sesuai dengan kode program dan tidak terjadi error.

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!

❖ Algoritma dengan while :

1. Mulai
2. Minta pengguna masukan kata
3. Minta pengguna masukan jumlah pengulangan
4. Siapkan penghitung
5. Ulangi dengan 'while'
6. Jalankan kode untuk memeriksa apakah program bisa dijalankan dengan baik dan tidak ada kesalahan
7. Simpan kode program yang telah diperbaiki atau catat perubahan yang telah dilakukan
8. Selesai

```
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 }
```

Luaran(Output) :



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

Compiled and executed in 14.437 sec(s)

Kode program sudah disusun dengan struktur yang benar. Hal ini dapat dilihat dari

luaran (output) yang sesuai dengan kode program dan tidak terjadi error.

❖ Algoritma dengan do . . while :

1. Mulai
2. Minta pengguna masukan kata
3. Minta pengguna masukan jumlah pengulangan
4. Siapkan penghitung
5. Ulangi dengan 'do . . while'
6. Jalankan kode untuk memeriksa apakah program bisa dijalankan dengan baik dan tidak ada kesalahan
7. Simpan kode program yang telah diperbaiki atau catat perubahan yang telah dilakukan
8. Selesai

```
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);
16            i++; // Faktor pengulang Increment
17        } while(i < jumlah);
18    }
19 }
```

Luaran(Output) :



Kode program sudah disusun dengan struktur yang benar. Hal ini dapat dilihat dari luaran (output) yang sesuai dengan kode program dan tidak terjadi error.

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!

❖ Algoritma :

1. Mulai
2. Siapkan variabel :
 - 'idPelajaran' mulai dari 0
 - 'nilaiPelajaran'
 - 'nilaiRataRata' mulai dengan 0.0
3. Tanya jumlah pelajaran
4. Proses input dan hitung rata-rata
5. Tampilkan hasil, tunjukkan rata-ratanya ke pengguna
6. Simpak kode program yang telah diperbaiki atau catat perubahan yang telah dilakukan
7. Selesai

```
1 import java.util.Scanner;
2
3 public class HitungRataRata {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7         // Inisiasi idPelajaran
8         int idPelajaran = 0;
9
10        // Inisiasi nilai pelajaran
11        double nilaiPelajaran;
12
13        // Inisiasi nilai rata-rata
14        double nilaiRataRata = 0.0;
15
16        // Minta pengguna untuk menuliskan jumlah pelajaran
17        System.out.print("Masukkan jumlah pelajaran: ");
18        int jumlahPelajaran = scanner.nextInt();
19
20        // Ketika idPelajaran lebih kecil dari jumlah pelajaran
21        while (idPelajaran < jumlahPelajaran) {
22            // Minta pengguna untuk menuliskan nilai pelajaran
23            System.out.print("Masukkan nilai pelajaran: ");
24            nilaiPelajaran = scanner.nextDouble();
25
26            // Hitung nilai rata-rata
27            nilaiRataRata = (nilaiRataRata * idPelajaran + nilaiPelajaran) / (idPelajaran + 1);
28
29            // Tambah satu ke idPelajaran
30            idPelajaran++;
31        }
32
33        // Tampilkan nilai rata-rata
34        System.out.println("Nilai rata-rata: " + nilaiRataRata);
35    }
36 }
```

Luaran(Output) :

Output Generated Files

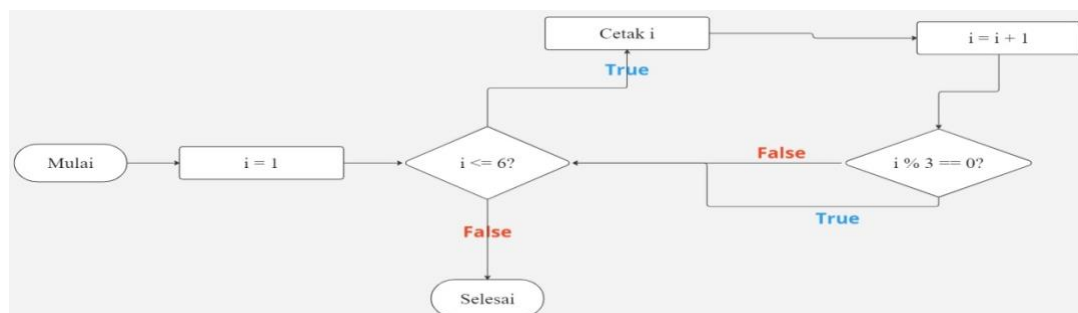
```
Masukkan jumlah pelajaran: 7
Masukkan nilai pelajaran: 95
Masukkan nilai pelajaran: 70
Masukkan nilai pelajaran: 78
Masukkan nilai pelajaran: 87
Masukkan nilai pelajaran: 93
Masukkan nilai pelajaran: 80
Masukkan nilai pelajaran: 88
Nilai rata-rata: 84.42857142857143
```

Compiled and executed in 51.757 sec(s)

Kode program sudah disusun dengan struktur yang benar. Hal ini dapat dilihat dari luaran (output) yang sesuai dengan kode program dan tidak terjadi error.

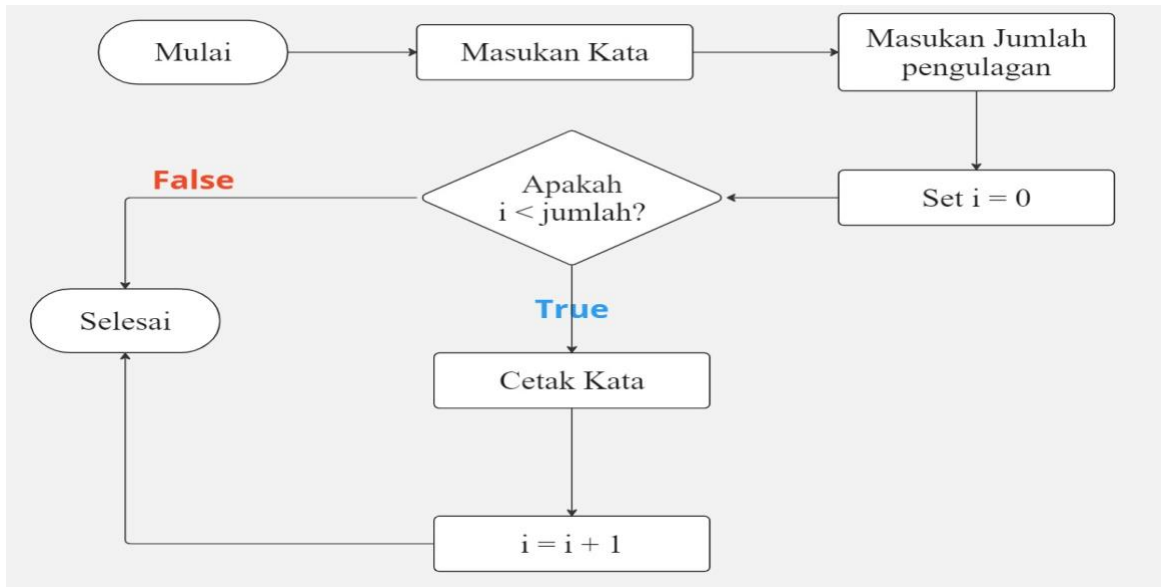
2.4 Rancang diagram flowchart dari Latihan 2.1, Latihan 2.2, dan Latihan 2.3!

❖ Flowchart Latihan 2.1

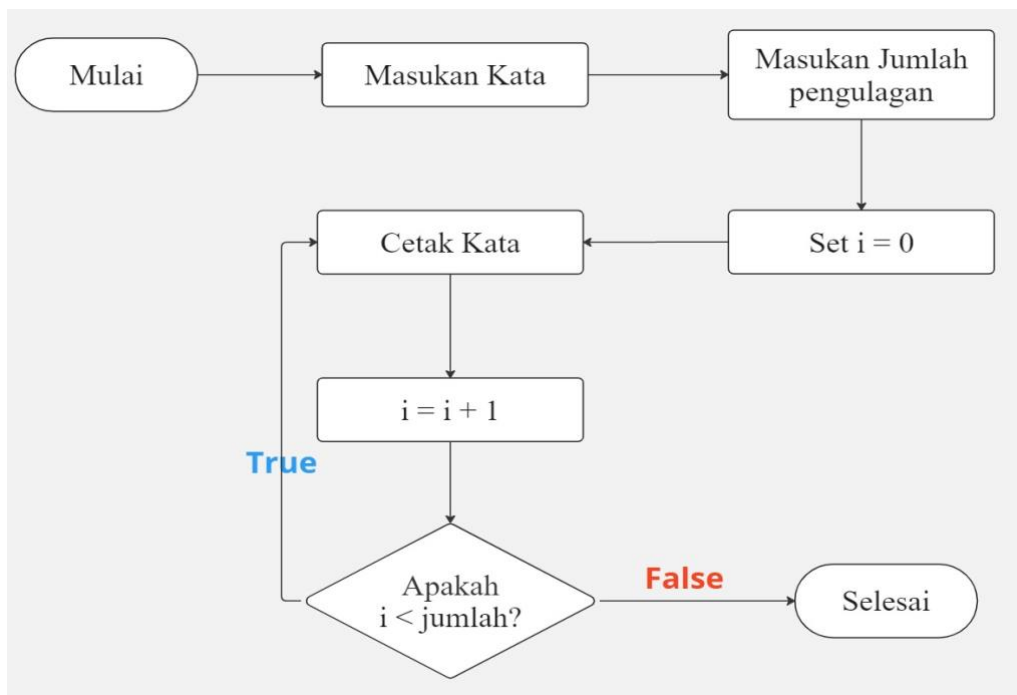


❖ Flowchart Latihan 2.2

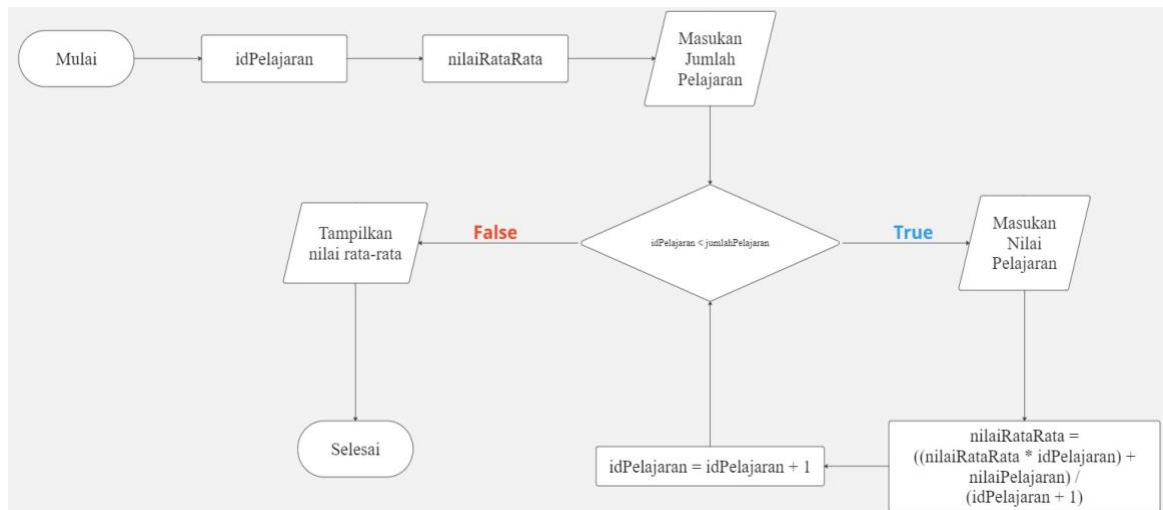
➤ Flowchart While



➤ Flowchart Do . . While



❖ Flowchart Latihan 2.3



[Nomor 4] Kesimpulan

- Pada program ini saya menggunakan 'while' karena kita perlu mengecek kondisi sebelum iterasi dimulai. Perbaiki program dengan 'do. . .while' karena kita perlu memastikan loop berjalan setidaknya sekali, terlepas dari kondisi awal.
- Struktur Java mengharuskan pemahaman kontrol aliran seperti 'break' dan 'continue' untuk menulis program yang efisien dan sesuai kebutuhan.

Refleksi

(Tuliskan singkat tentang pengalaman belajar, pemaknaan pengetahuan yang baru, tantangan yang dihadapi pada minggu tersebut. Ringkasan singkat dari semua soal, bukan per soal)

Minggu ini, saya mendalami konsep perulangan dalam pemrograman terutama dengan menggunakan 'for' dan 'while'. Saya belajar bagaimana perulangan dapat mengontrol alur program dan bagaimana kita bias menggunakan 'break' dan 'continue' untuk mengarah jalannya perulangan. Salah satu hal baru yang saya pahami adalah perbedaan antara 'while' dan 'do. . .while', terutama bagaimana 'do. . .while' akan menjalankan setidaknya satu kali iterasi, bahkan jika kondisinya tidak terpenuhi.

Tantangan yang saya hadapi adalah memahami cara kerja perulangan bersarang dan memastikan variabel kontrol diatur dengan benar agar program berjalan sesuai keinginan. Ada juga ketika saya harus mengatasi kesalahan sintaks dan logika dalam kode, yang mengajarkan saya pentingnya menganalisis kesalahan kode program dengan teliti agar tidak terjadi error.