

Template Lembar Kerja Individu dan Kelompok

Nama & NPM	Topik:	Tanggal:
Karina Hadiyah Ramadona G1F024040	FOR dan WHILE Java	10 Oktober 2024

[No.1] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variable

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

[No.1] Analisis dan Argumentasi

- 1) Uraikan rancangan solusi yang diusulkan.
 - 1.1 Kesalahan pada kode:

Deklarasi dan inialisasi variabel y dalam for loop salah:
Pada perulangan for, variabel y tidak diinisialisasi dengan benar.
Kondisi ganjil-genap menggunakan tipe data double:
Operasi modulus (%) digunakan untuk menentukan apakah angka ganjil atau genap.
Namun, karena variabel y bertipe double, operasi modulus tidak dapat digunakan secara langsung untuk pengecekan ganjil-genap.
 - 1.2 Kita ingin mencetak semua kombinasi untuk i = 1 dan i = 2, dan kita ingin menghentikan loop luar ketika i mencapai 3 atau lebih.
Setelah i mencapai 2, kita tidak perlu melakukan apapun, sehingga kita tidak ingin menggunakan break pada perulangan dalam (kedua), tetapi kita juga tidak ingin mencetak apa pun setelah i lebih dari 2.
- 2) Analisis solusi, kaitkan dengan permasalahan.
 - 1.1 diinisialisasi terlebih dahulu sebelum pengecekan kondisi dan perubahan nilai.
Ubah tipe data double menjadi int untuk memungkinkan penggunaan operasi modulus.
- 3) Perbaiki kode
 - 1.1

```
public class ContohFor {  
    public static void main(String[] args) {  
        for (int y = 0; y <= 15; y++) {  
            if (y % 2 == 1) {    // kondisi 1  
                continue; // Lewati angka ganjil  
            } else if (y == 8) { // kondisi 2  
                break; // Hentikan saat y == 8  
            } else {  
                System.out.println(y + " ");  
            }  
        }  
    }  
}
```
 - 1.2

```
public class ForBersarang {  
    public static void main(String[] args) {  
        pertama:  
        for (int i = 1; i < 5; i++) {  
            kedua:  
            for (int j = 1; j < 3; j++) {  
                System.out.println("i = " + i + "; j = " + j);  
            }  
            if (i == 2) {  
                continue; // Lanjutkan ke iterasi berikutnya  
            }  
        }  
    }  
}
```
 - 1.3

```
import java.util.Scanner;  
  
public class ForBersarang {  
    public static void main(String[] args) {
```

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

for (int t = 1; t <= tinggi; t++) { // Menghitung Jumlah Tinggi Piramida
    // Menghitung Jumlah Spasi per 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++) {
        System.out.print("*"); // Mencetak bintang
    }
    System.out.println(); // Membuat Baris Baru
}

input.close(); // Menutup scanner
}
```

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Rancang desain solusi atau algoritma
 - a) Mulai
 - b) Buka jdoodle
 - c) Perbaiki dan analisis luaran yang ada
 - d) Selesai.
- 2) Tuliskan kode program dan luaran
 - a) Beri komentar pada kode

1.1

```
public class ContohFor {
    public static void main(String[] args) {
        for (int y = 0; y <= 15; y++) {
            if (y % 2 == 1) { // kondisi 1
                continue; // Lewati angka ganjil
            } else if (y == 8) { // kondisi 2
                break; // Hentikan saat y == 8
            } else {
                System.out.println(y + " ");
            }
        }
    }
}
```

Luaran

```
0
2
4
6
|
```

1.2

```
public class ForBersarang {
    public static void main(String[] args) {
        pertama:
        for (int i = 1; i < 5; i++) {
            kedua:
            for (int j = 1; j < 3; j++) {
                System.out.println("i = " + i + "; j = " + j);
            }
            if (i == 2) {
                continue; // Lanjutkan ke iterasi berikutnya
            }
        }
    }
}
```

Luaran 1.2

```

i = 1; j = 1
i = 1; j = 2
i = 2; j = 1
i = 2; j = 2
i = 3; j = 1
i = 3; j = 2
i = 4; j = 1
i = 4; j = 2

```

1.3

```

import java.util.Scanner;

public class ForBersarang {
    public static void main(String[] args) {
        // Instance Input Scanner
        Scanner input = new Scanner(System.in);
        System.out.print("Masukan Input: ");
        int tinggi = input.nextInt(); // Mendapatkan Input Dari User

        for (int t = 1; t <= tinggi; t++) { // Menghitung Jumlah Tinggi Piramida
            // Menghitung Jumlah Spasi per 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++) {
                System.out.print("*"); // Mencetak bintang
            }
            System.out.println(); // Membuat Baris Baru
        }

        input.close(); // Menutup scanner
    }
}

```

Luaran

Output Generated Files

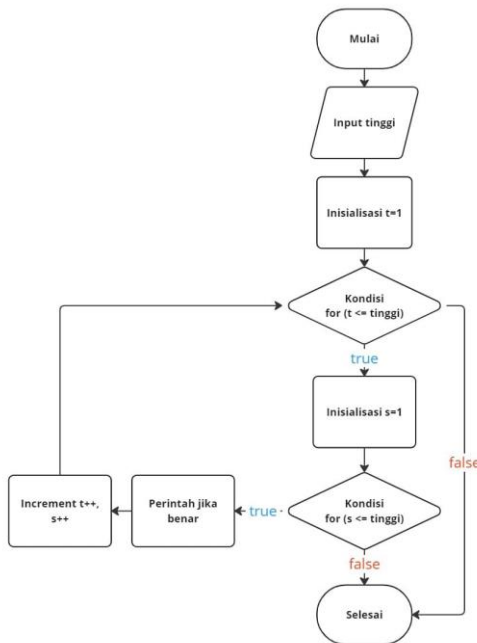
```

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

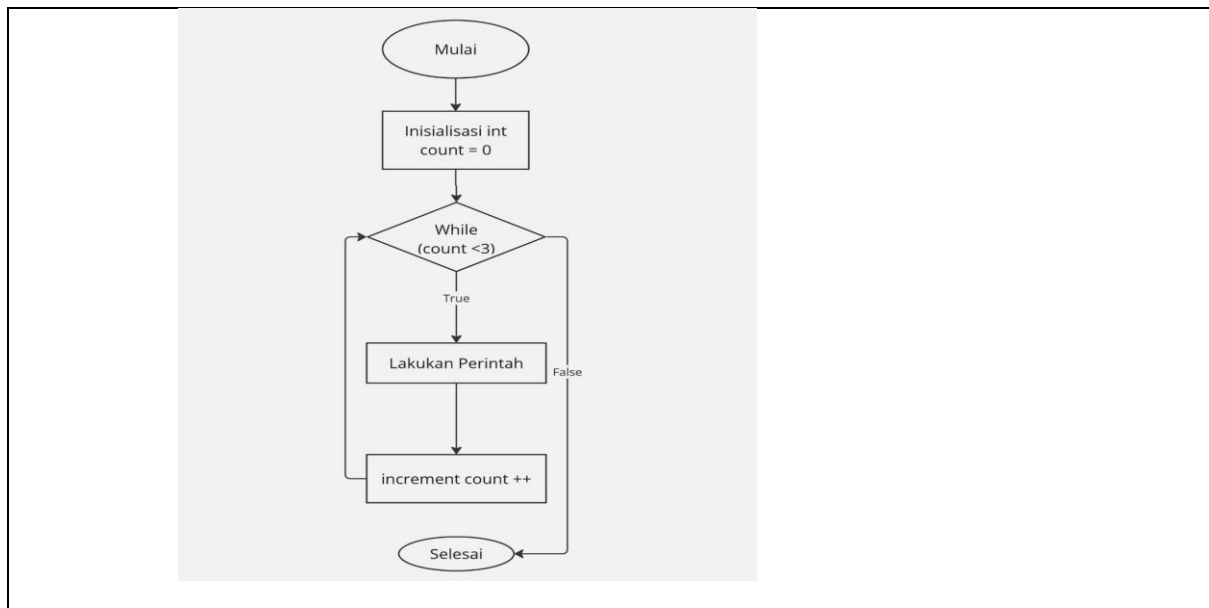
```

Flowchart

1.2



1.3



[No.1] Kesimpulan

1) Analisa

Contoh 1

Perbaikan dilakukan dengan menambahkan inisialisasi, kondisi, dan pembaruan yang tepat dalam loop for. Struktur dasar Java mengharuskan setiap loop untuk mengikuti urutan yang jelas: pertama adalah inisialisasi variabel, kemudian kondisi yang harus dipenuhi, dan terakhir pembaruan variabel.

Contoh 2

Penggunaan continue memungkinkan kita untuk melewati iterasi saat kondisi tertentu terpenuhi, sehingga iterasi berikutnya di loop luar tetap dapat dijalankan. Ini diterapkan untuk memperbaiki perilaku loop bersarang, sehingga program dapat melanjutkan eksekusi tanpa terhenti pada kondisi yang tidak diinginkan.

Contoh 3

Perbaikan dilakukan dengan menambahkan logika yang tepat untuk penempatan spasi dan bintang dalam loop bersarang. Ini memastikan bahwa program dapat menghasilkan pola piramida bintang sesuai dengan input yang diberikan oleh pengguna, sehingga output menjadi lebih sesuai dengan yang diharapkan.

Nama & NPM	Topik:	Tanggal:
Karina Hadiyah Ramadona G1F024040	FOR dan WHILE Java	10 Oktober 2024

[No.2] Identifikasi Masalah:

1) Uraikan permasalahan dan variable

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 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)

[No.2] Analisis dan Argumentasi

1) Uraikan rancangan solusi yang diusulkan.

Saya mengusulkan permasalahan ini dapat diatasi dengan cara

[2.3] Membuat variabel untuk menyimpan ID pelajaran, nilai pelajaran, dan rata-rata. Dan meminta pengguna untuk memasukkan jumlah pelajaran. Menggunakan loop while untuk meminta input nilai pelajaran sesuai dengan jumlah yang dimasukkan. Dengan menghitung rata-rata secara bertahap dengan menambahkan nilai baru dan membagi dengan total pelajaran yang telah

2) Alasan solusi ini karena dengan inisialisasi variabel bisa menyimpan ID Pelajaran, nilai Pelajaran, dan rata-rata. Dan juga menggunakan Input jumlah Pelajaran agar pengguna untuk memasukkan jumlah Pelajaran. Output Rata-Rata menampilkan hasil perhitungan rata-rata kepada pengguna.

3) Evaluasi dan Kesimpulan

2.1 Kode awal berhenti mencetak ketika i mencapai 4, menghasilkan output 1, 2, 3. Perintah break menghentikan loop sepenuhnya ketika kondisi terpenuhi.

Dengan continue;:

Setelah perubahan, perintah continue membuat loop melanjutkan iterasi berikutnya tanpa menghentikan seluruh loop. Dalam hal ini, semua nilai dari i dicetak, termasuk 4, 5, dan 6, karena tidak ada kondisi yang menghentikan loop.

2.1 kesimpulan

Perubahan dari break ke continue berfungsi untuk mengubah perilaku loop. break menghentikan loop sepenuhnya, sedangkan continue hanya melompati satu iterasi saat kondisi tertentu terpenuhi, sehingga memungkinkan loop untuk terus berjalan.

2.2 Penggunaan while lebih cocok ketika Anda ingin mengecek kondisi terlebih dahulu dan mungkin tidak ingin menjalankan kode jika kondisi awal tidak terpenuhi.

Penggunaan do-while lebih tepat ketika Anda ingin memastikan kode dalam blok dieksekusi setidaknya satu kali, terlepas dari kondisi awal yang dihadapi.

[No.2] Penyusunan Algoritma dan Kode Program

1) Rancang desain solusi atau algoritma

- a) Mulai
- b) Buka jdoodle
- c) Perbaiki dan analisis luaran yang ada
- d) Selesai.

2) Tuliskan kode program dan luaran

a) Beri komentar pada kode

2.1 ubah 1

```
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                 break; //ubah1
9             }
10        }
11    }
12 }
13 }
```

Luaran

```
1
2
|
```

Ubah 2

```
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; //ubah2
9             }
10        }
11    }
12 }
13 }
```

Luaran

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

2.3

```
import java.util.Scanner;

public class RataRataNilai {
    public static void main(String[] args) {
        // Inisialisasi Scanner untuk input pengguna
        Scanner scanner = new Scanner(System.in);

        // Inisialisasi variabel
        int idPelajaran = 0; // Penghitung jumlah pelajaran yang telah diproses
        double totalNilai = 0; // Total nilai pelajaran
        double nilaiRataRata; // Rata-rata nilai

        // Minta pengguna untuk menuliskan jumlah pelajaran
        System.out.print("Masukkan jumlah pelajaran: ");
        int jumlahPelajaran = scanner.nextInt();

        // Menggunakan loop untuk meminta nilai pelajaran
        while (idPelajaran < jumlahPelajaran) {
            System.out.print("Masukkan nilai pelajaran ke-" + (idPelajaran + 1) + ": ");
            double nilaiPelajaran = scanner.nextDouble(); // Minta pengguna memasukkan nilai pelajaran

            // Hitung total nilai
            totalNilai += nilaiPelajaran; // Tambahkan nilai pelajaran ke total

            // Tambah satu ke idPelajaran
            idPelajaran++; // Naikkan penghitung pelajaran
        }

        // Hitung nilai rata-rata
        nilaiRataRata = totalNilai / jumlahPelajaran;

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

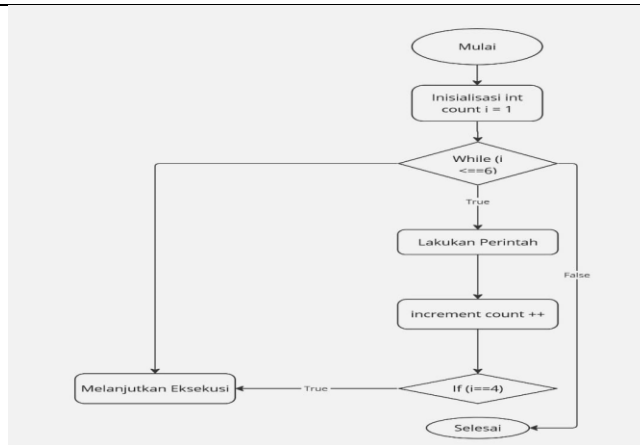
        // Tutup scanner
        scanner.close();
    }
}
```

Luaran

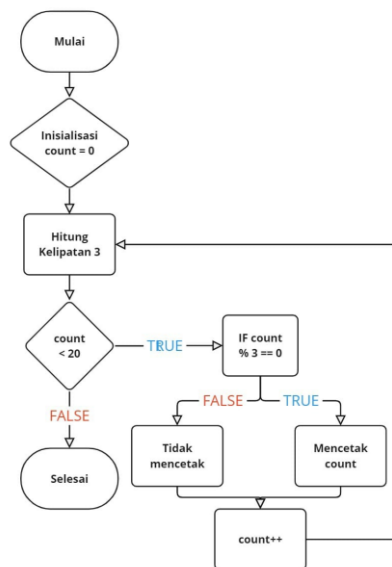
```
Masukkan jumlah pelajaran: 4
Masukkan nilai pelajaran ke-1: 80
Masukkan nilai pelajaran ke-2: 75
Masukkan nilai pelajaran ke-3: 60
Masukkan nilai pelajaran ke-4: 85
Nilai rata-rata: 75.0
```

Flowchart

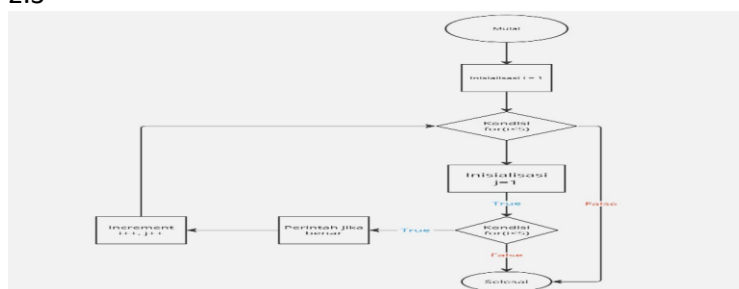
2.1



2.2



2.3



[No.2] Kesimpulan

1) Analisa

a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!

Kode pertama menggunakan loop while untuk mencetak angka dari 1 hingga 6. Namun, loop berhenti secara paksa menggunakan perintah break ketika nilai variabel i mencapai 4. Masalah di sini adalah pengulangan terhenti sebelum mencapai batas maksimal yang seharusnya. Perintah break digunakan sebagai penyelesaiannya untuk menghentikan loop lebih awal ketika kondisi tertentu ($i == 4$) terpenuhi.

Kesimpulan Kode Kedua: Kode kedua menggunakan loop while untuk mencetak kata yang dimasukkan pengguna beberapa kali, sesuai jumlah yang diminta. Program meminta input dari pengguna berupa kata dan jumlah pengulangan, kemudian mencetak kata tersebut sebanyak jumlah yang diinginkan. Masalah yang diselesaikan adalah bagaimana

mengulang string sesuai dengan input pengguna, yang diatasi dengan loop while yang berhenti ketika nilai i mencapai batas jumlah yang diminta.