

Template Lembar Kerja Individu dan Kelompok

Nama & NPM	Topik:	Tanggal:
Naufal Wafiq Syafri G1F024054	For dan while	10 Oktober 2024

[Nomor 1] Identifikasi Masalah:

Pernyataan for dapat mengendalikan proses berulang dengan jumlah perulangan yang sudah ditentukan.

Bentuk umum For:

```
for (inisialisasi; kondisi; iterasi) {  
    pernyataan;  
}
```

Makna bagian **for**:

1. inisialisasi untuk memberikan nilai kepada variabel yang digunakan untuk mengontrol pengulangan.
2. kondisi untuk mengontrol pengulangan untuk dilanjutkan atau diakhiri.
3. Penaikan (increment) atau penurunan (decrement) untuk menaikkan atau menurunkan nilai variabel perulangan.

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

```
public class ContohFor{  
    public static void main(String[] args) {  
        for (int y = 0; y <= 10; ++y) { //ubah 1  
            if (y % 2 == 1) //ubah 2  
                continue; //baris 1  
            else if (y == 8) //ubah 3  
                break; //baris 2  
            else  
                System.out.println(y + " ");  
        }  
    }  
}
```

Luaran contoh 1:

```
0  
2  
4  
6
```

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)  
                    break kedua; //ubah1  
            }  
        }  
    }  
}
```

Luaran Contoh 2:

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

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 contoh 3:

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

Latihan 1

- 1.1. Analisa tujuan penulisan kata kunci `continue` dan `break` pada Contoh 1!
Buat perubahan nilai angka pada variabel di
//Ubah 1 menjadi `for (int y = 0; y <= 15; y++)` { lalu running, periksa hasilnya
//Ubah 2 menjadi `if (y % 2 == 0)` lalu running, periksa hasilnya
//Ubah 3 menjadi `else if (y == 9)` lalu running, periksa hasilnya
Analisa dampaknya perubahan ini terhadap luaran setelah running!
- 1.2. Buat perubahan kode pada Contoh 2 di baris //Ubah1 menjadi
 - a. `continue` pertama; lalu running, periksa hasilnya
 - b. `break` pertama; lalu running, periksa hasilnya
 - c. `continue` kedua; lalu running, periksa hasilnyaAnalisa perbedaan perubahan kode pada Ubah 1 untuk setiap poin (a), (b), dan (c)!
- 1.3. Cermati kode contoh 3. Apabila ingin menghasilkan luaran berikut:

Luaran:

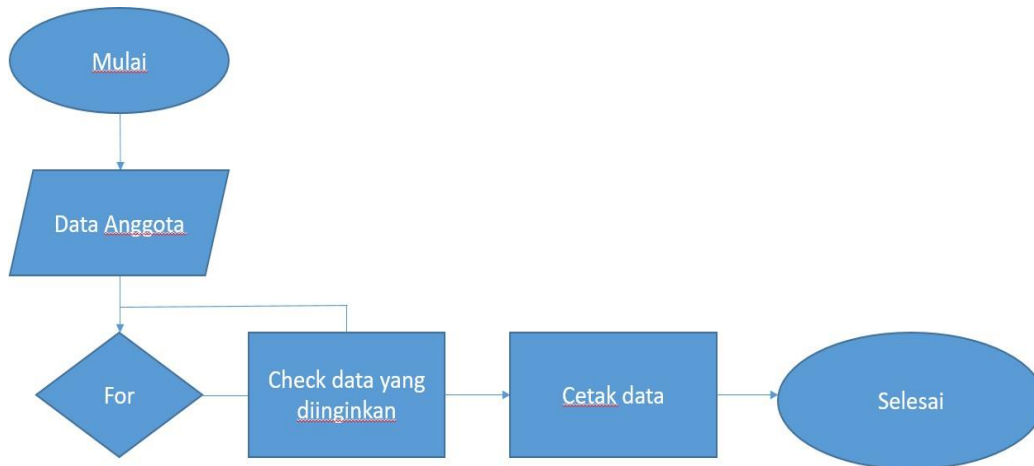
```
Masukan Input: 7
*****
*****
*****
****
***
**
*
```

Susunlah analisa kode untuk menghasilkan luaran tersebut!

- 1.4. Analisa diagram flowchart dari Latihan 1.2 dan 1.3!

[Nomor 1] Analisis dan Argumentasi

- 1) Dengan mengubah kondisi yang diinginkan maka luaran juga akan berubah.
- 2) Dengan menambahkan continue dan break pada kode maka prosesnya juga akan berubah.
- 3) Ganti kondisi for loop.



[Nomor 1] Penyusunan Algoritma dan Kode Program

<pre>public class ContohFor{ public static void main(String[] args) { for (int y = 0; y <= 15; y++) { //ubah 1 if (y % 2 == 0) //ubah 2 continue; //baris 1 else if (y == 9) //ubah 3 break; //baris 2 else System.out.println(y + " "); } } }</pre>	<pre>java -cp /tmp/gwHlMmrEW7/ContohFo 1 3 5 7 === Code Execution Successful ===</pre>
<pre>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) break kedua; //ubah1 } } } }</pre>	<pre>java -cp /tmp/18Van01Kbq/ForBersarang i = 1; j = 1 i = 1; j = 2 i = 2; j = 1 i = 3; j = 1 i = 3; j = 2 i = 4; j = 1 i = 4; j = 2 === Code Execution Successful ===</pre>
<pre>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--){ // Loop untuk mencetak bintang for(int b=1; b<=t; b++){ System.out.print("*"); } System.out.println(); // Pindah ke baris berikutnya } } }</pre>	<pre>java -cp /tmp/fUfHEGVdY/ForBersarang Masukan Input: 7 ***** ***** **** *** ** * === Code Execution Successful ===</pre>

[Nomor 1] Kesimpulan

- 1) Analisa
 - a) Kode sudah bisa berjalan dengan baik dan menampilkan data yang diinginkan.
 - b) Saya mengubah sedikit saja kodenya karena memang kode sudah bisa berjalan dengan baik.

Template Lembar Kerja Individu dan Kelompok

Nama & NPM	Topik:	Tanggal:
Naufal Wafiq Syafri G1F024054	For dan while	10 Oktober 2024

[Nomor 2] Identifikasi Masalah:

Pernyataan **while** berguna untuk melakukan proses yang berulang. Pernyataan akan dijalankan terus menerus selama kondisi bernilai true (benar). Oleh karena itu, kita harus membuat kondisi suatu saat ungkapan bernilai salah agar perulangan berakhir.

Bentuk umum **while** seperti:

```
while (kondisi) {  
    pernyataan;  
}
```

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.

```
public class WhileBersarang {  
    public static void main(String[] args) {  
        int count = 0; //ubah1  
        while (count < 20) {  
            if (count % 3 == 0) //ubah2  
                System.out.println(count);  
            count++;  
        }  
    }  
}
```

Luaran:

```
0  
3  
6  
9  
12  
15  
18
```

Latihan 2

2.1. Buat perubahan nilai angka pada variabel di Contoh 4
//Ubah 1 menjadi continue; lalu running, periksa hasilnya
Analisa dampaknya perubahan terhadap luaran setelah running dan uraikan kegunaan break dan continue!

2.2. Buat perubahan nilai angka pada variabel di Contoh 5
//Ubah2 menjadi `if (count % 5 == 0)` lalu running, periksa hasilnya

Analisa dampaknya perubahan terhadap luaran setelah running dan uraikan kegunaan % untuk angka yang berbeda pada perintah tersebut!

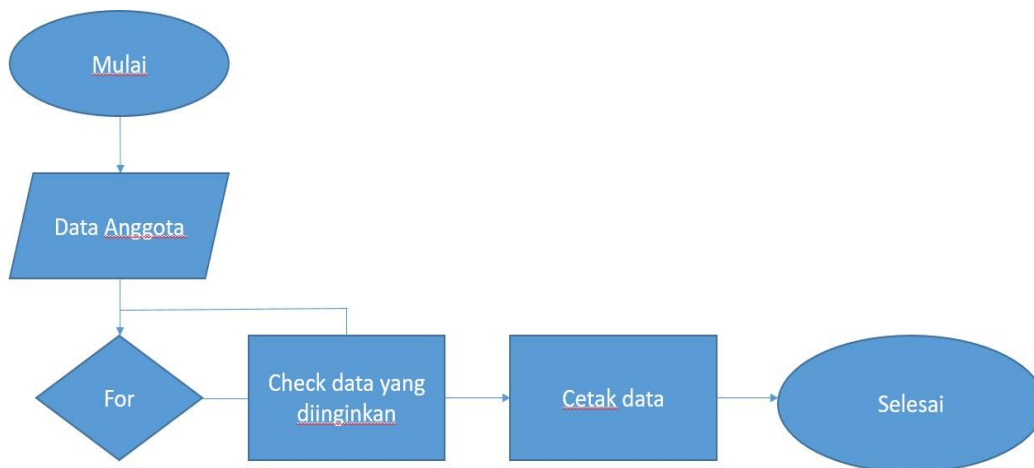
2.3. Buat perubahan nilai angka pada variabel di

//Ubah1 menjadi `while (count < 0) {` lalu running, periksa hasilnya
Ubahlah baris kode `while` pada Contoh 5 menjadi `do ... while` dengan persyaratan yang sama `while (count < 0)`. Bandingkan hasil luaran antara menggunakan `while` dan `do ... while`!

2.4. Analisa diagram flowchart dari Latihan 2.1, Contoh 5, dan Latihan 2.3!

[Nomor 2] Analisis dan Argumentasi

- 1) Dengan mengubah kondisi yang diinginkan maka luaran juga akan berubah.
- 2) Ganti kondisi while loop.



[Nomor 2] Penyusunan Algoritma dan Kode Program

- 1) Rancang desain solusi atau algoritma
Ubah kode;
Lihat luaran;
Jika error perbaiki;
- 2) Tuliskan kode program dan luaran

```
public class ContohWhile{
    public static void main(String[] args) {
        int i=1;
        while(i<=6){
            System.out.println(i);
            i++;
            if(i==4){
                continue; //ubah1
            }
        }
    }
}

public class WhileBersarang {
    public static void main(String[] args) {
        int count = 0; //ubah1
        while (count < 20) {
            if (count % 5 == 0) //ubah2
                System.out.println(count);
            count++;
        }
    }
}

public class WhileBersarang {
    public static void main(String[] args) {
        int count = 0; //ubah1
        do {
            if (count % 5 == 0) //ubah2
                System.out.println(count);
            count++;
        } while (count < 0); // Kondisi akan diperiksa setelah setidaknya sekali menjalankan loop
    }
}
```

```
java -cp /tmp/f1IqAC0tB3/ContohWhile
1
2
3
4
5
6

=== Code Execution Successful ===

java -cp /tmp/MV501XAQHy/WhileBersarang
0
5
10
15

=== Code Execution Successful ===

java -cp /tmp/wRUjLCcvAy/WhileBersarang
0

=== Code Execution Successful ===
```

[Nomor 2] Kesimpulan
1) Analisa <ul style="list-style-type: none">a) Sudah bisa berjalan dengan baik dan menampilkan data yang diinginkan.b) Mengubah sedikit saja kodenya karena memang kode sudah bisa berjalan dengan baik.

Refleksi

(Saya belajar menggunakan loop di Java :D)