

| | | |
|---|------------------|--------------------------|
| Nama & NPM | Topik: | Tanggal: |
| Khairul Aji Pangestu G1F024042 | For While | 28 September 2024 |

[No. 1] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

UNIT 1 FOR

Contoh 1:

```
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:

```
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:

```
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 hasilnya
 Analisa 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!

[No.1] Analisis dan Argumentasi

- 1) Pada contoh 1
 - a. Jika //Ubah 1 diubah menjadi `for (int y = 0; y <= 15; y++) {` lalu running, maka hasilnya akan tetap sama.
 - b. Jika //Ubah 2 menjadi `if (y % 2 == 0)` lalu running, maka hasilnya menunjukkan angka ganjil dari 1 hingga 15
 - c. Jika //Ubah 3 menjadi `else if (y == 9)` lalu running, maka hasilnya menunjukkan angka ganjil dari 1 sampai 7
- 2) Pada Contoh 2 di baris //Ubah1 diubah menjadi
 - a. Jika diubah menjadi `continue` pertama; lalu running, maka hasilnya:
`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`
 - b. Jika diubah menjadi `break` pertama; lalu running, maka hasilnya:
`i = 1; j = 1`
`i = 1; j = 2`
`i = 2; j = 1`
 - c. Jika diubah menjadi `continue` kedua; lalu running, maka hasilnya:
`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`

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Mulai
 - (b) Buka Java Compiler
 - (c) Buat Project UnitFor
 - (d) Deklarasi Class ContohFor
 - (e) Deklarasi fungsi Public static
 - (f) Buat kode perulangan dengan For
 - (g) Run
 - (h) Selesai
- 2) Kode program dan luaran
 - a) Potongan kode dan hasil luaran Latihan 1.3

```

1 import java.util.Scanner;
2
3 public class ForBersarang2 {
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         for(int t=1; t<=tinggi; t++){
10             //Menghitung Jumlah Tinggi Piramida
11             for(int s=1; s<=t; s++){
12                 //Menghitung Jumlah Spasi per Baris
13                 System.out.print(" ");
14             }
15             System.out.println(); //Membuat Baris Baru
16         }
17     }
18 }
19
20

```

Input/Output

Language Version:

Input Arguments

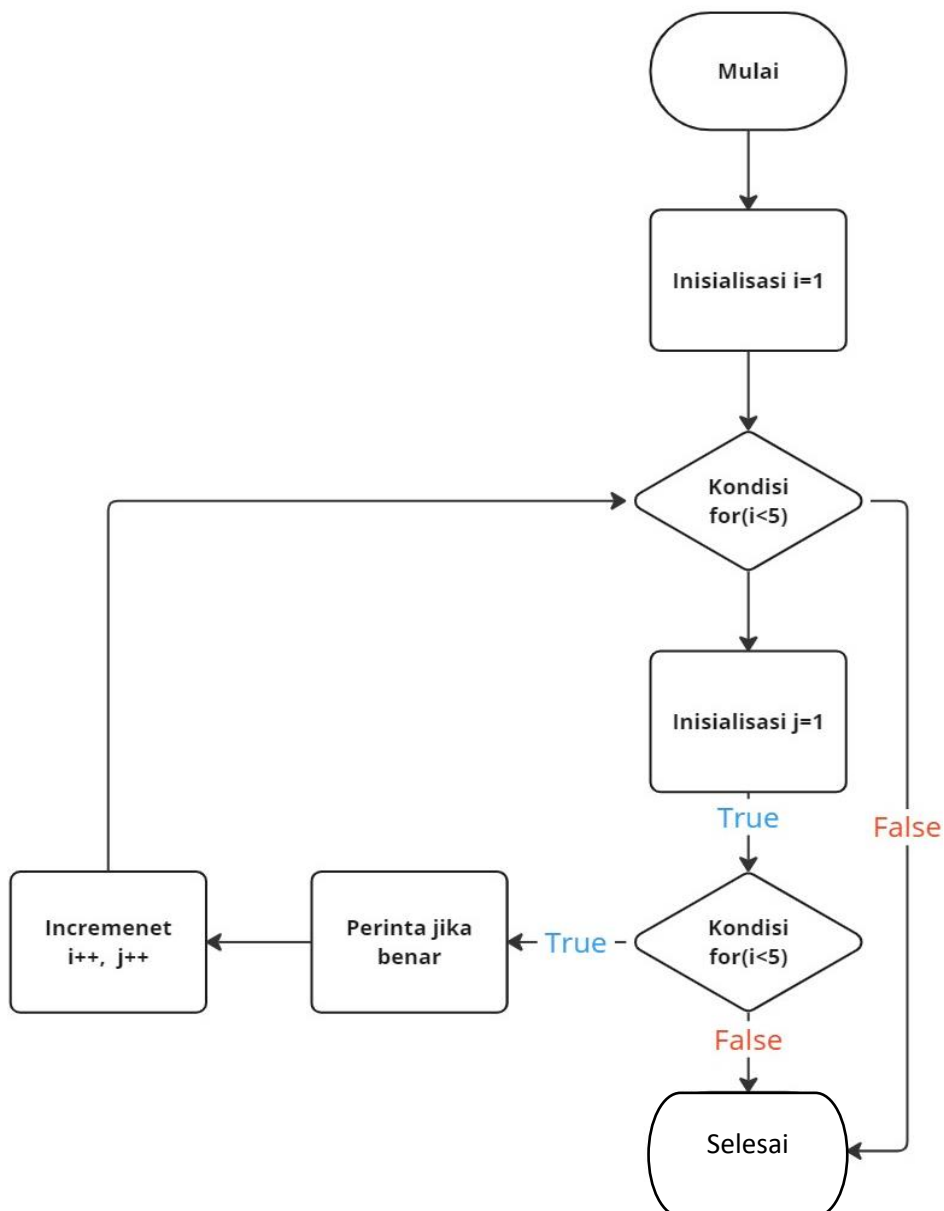
Output Generated Files

Masukan Input: 7
 *
 **

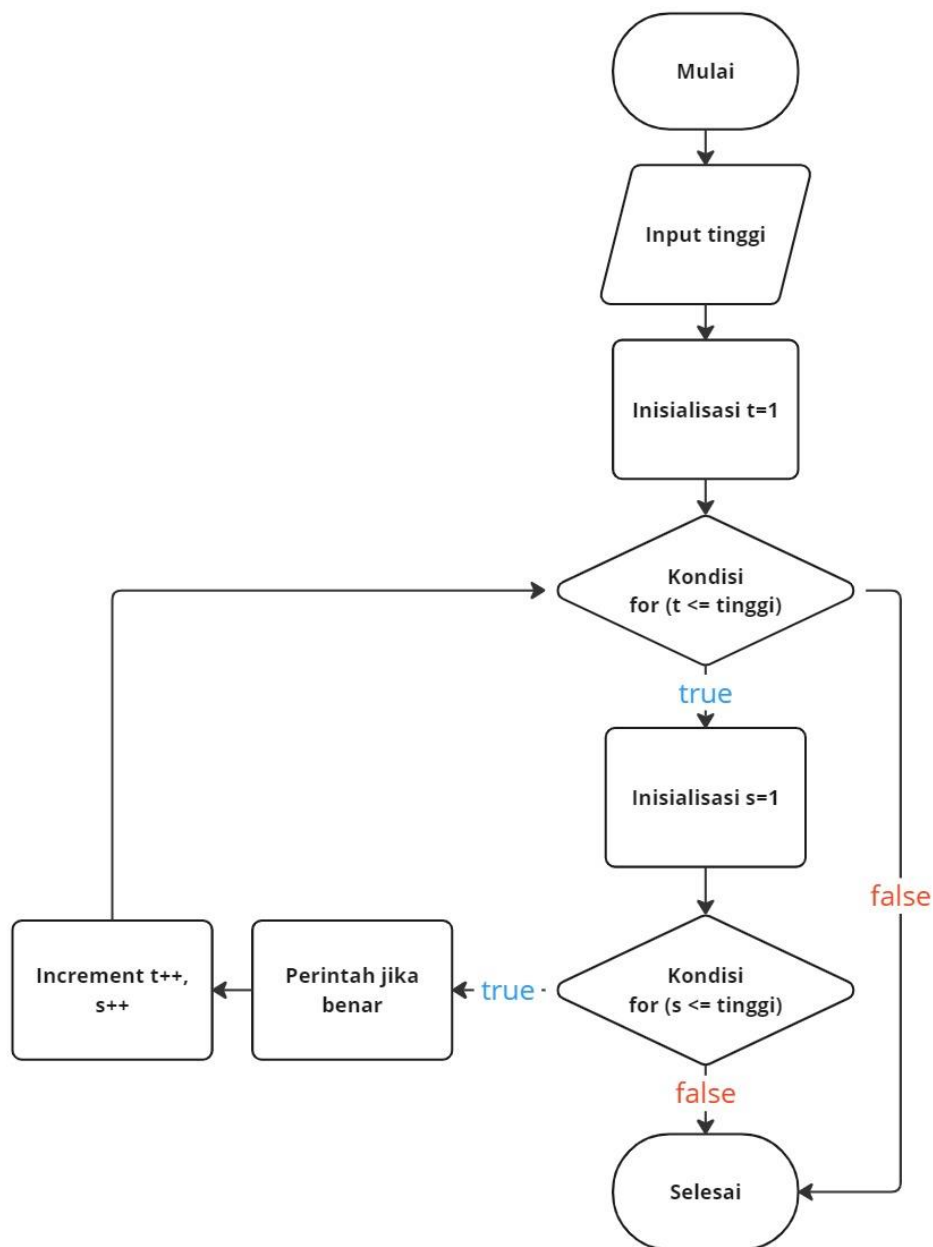
 |

Compiled and executed in 11.452 sec(s)

b) Flowchart 1.2



c) Flowchart 1.3



- d) Analisa luaran yang dihasilkan
Luaran sudah sesuai dengan program yang disusun.
Kode dibuat untuk membuat implementasi perulangan for. Kode dan luaran sudah benar dan tepat.

[No.1] Kesimpulan

1) Analisa

- Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
Pada contoh 1 dan 2, penggunaan continue dan break berfungsi untuk mengatur alur perulangan:
 - continue melewati iterasi tertentu tanpa menyelesaikan kode di dalam loop.
 - break menghentikan perulangan sepenuhnya saat kondisi tercapai.
- Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?
 - continue digunakan untuk melewati kondisi yang tidak relevan.
 - break digunakan untuk menghentikan perulangan guna menghindari iterasi yang tidak perlu, sehingga lebih efisien.

[No. 2] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

UNIT 1 WHILE

Contoh 4:

```
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:

```
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!

[No.2] Analisis dan Argumentasih

- 1) Pada Contoh 4 jika //Ubah 1 diubah menjadi continue; lalu running, maka hasilnya angka 1 sampai dengan 6.

Dalam kode tersebut break digunakan untuk menghentikan eksekusi suatu kasus setelah menjalankan statement yang sesuai. Sedangkan continue untuk melanjutkan perulangan, tetapi melewati sisa instruksi yang ada pada iterasi saat ini dan langsung beralih ke iterasi berikutnya.

- 2) Pada Contoh 5 jika diubah menjadi //Ubah2 menjadi if (count % 5 == 0) lalu running, maka hasilnya adalah

0
5
10
15

Kegunaan % pada perintah tersebut untuk mencari angka yang habis dibagi 5, angka lain misalnya %2==0 untuk mencari nilai genap atau nilai kelipatan 2, dan seterusnya.

- 3) Pada Contoh 5 jika diubah menjadi //Ubah1 menjadi while (count < 0) lalu running, maka hasilnya tidak akan ada hasilnya. Dan jika kode tersebut diubah menjadi do ... while maka hasil luaranya adalah 0.

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Mulai
 - (b) Buka Java Compiler
 - (c) Buat Project UnitFor
 - (d) Deklarasi Class While
 - (e) Deklarasi fungsi Public static
 - (f) Buat kode perulangan dengan While atau Do ... While
 - (g) Run
 - (h) Selesai
- 2) Kode program dan luaran
 - a) Potongan kode dan hasil luaran Latihan 2.3

```

Codeium: Refactor | Explain
1 public class WhileBersarang {
  Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
2   public static void main(String[] args) {
3     int count = 0; //ubah1
4     do {
5       if (count % 3 == 0)
6         System.out.println(count);
7       count++;
8     } while (count < 0); //ubah2
9   }
10 }

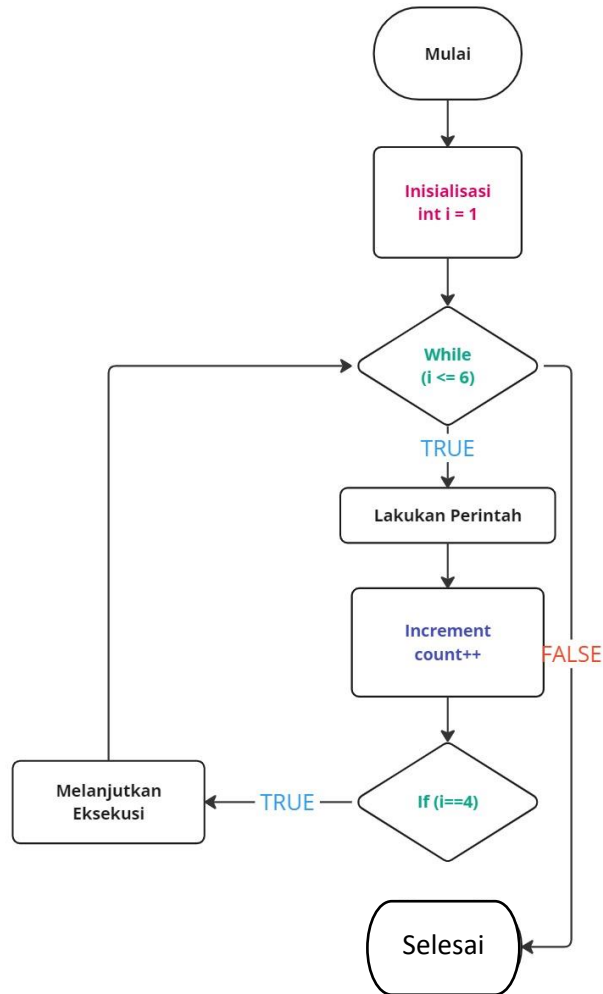
```

```

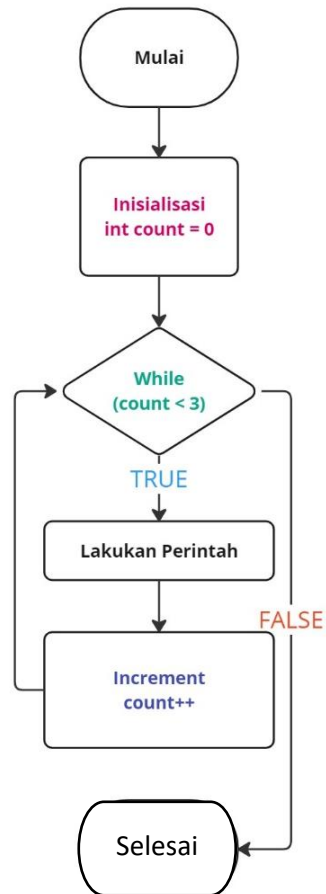
e\User\workspaceStorage
0
PS D:\vscode>

```

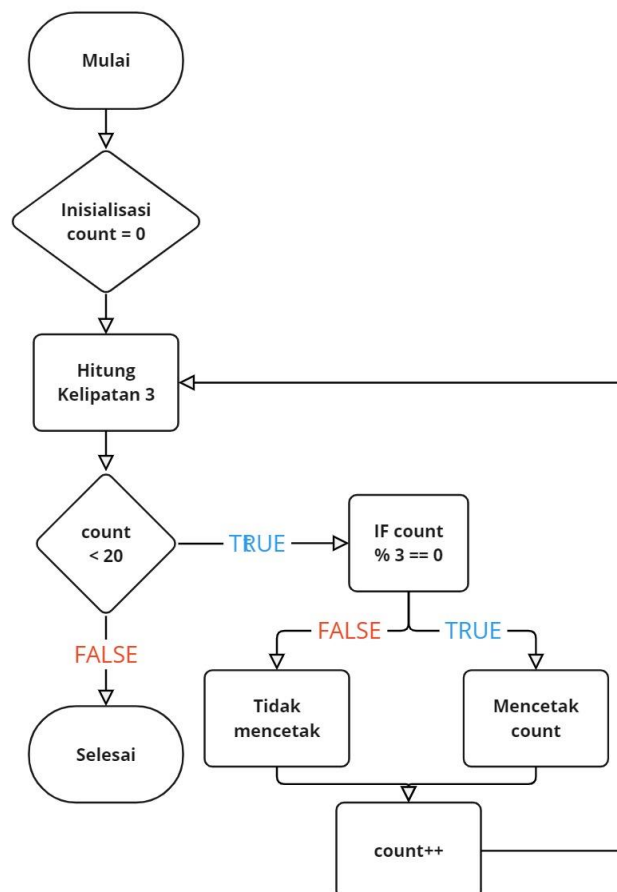
b) Flowchart Latihan 2.1



c) Flowchart Contoh 5



d) Flowchart Latihan 2.3



- a) Analisa luaran yang dihasilkan
Luaran sudah sesuai dengan program yang disusun.
Kode dibuat untuk membuat implementasi perulangan WHILE. Kode dan luaran sudah benar dan tepat.

[No.2] Kesimpulan

1) Analisa

- a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!
Pada contoh 4 dan 5, break digunakan untuk menghentikan loop, sedangkan operator modulo (%) membantu memfilter angka yang memenuhi syarat tertentu. Perbedaan antara *while* dan *do-while* terlihat pada cara pengecekan kondisi dilakukan.
- b) Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?
- break dipilih untuk menghentikan iterasi saat batas tercapai.
 - Modulo (%) digunakan untuk menemukan angka yang spesifik (kelipatan), menjaga hasil sesuai kebutuhan.

Refleksi

Materi minggu ini cukup mudah, tetapi sangat disayangkan ini adalah materi terakhir pada e-Learning **2024 Komputer dan Pemrograman**, sejauh ini saya cukup bisa memahami berbagai materi tersebut, Terimakasih bu Endina Purwandari, Dr., ST., M.Kom..