

Nama & NPM	Topik:	Tanggal:
Alif Alfarizi G1F024069	For	9 Oktober 2024

Nomor 1 Identifikasi Masalah:

1) Uraikan permasalahan dan variable

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 :

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 :

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 :

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!

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

<https://www.youtube.com/watch?v=lj9qLLblxEU> – Video materi For

Nomor 1 Analisis dan Argumentasi

Contoh 1 :

1.1 pada kode program tersebut ketika saya mengganti // ubah 1 menjadi for (int y = 0; y <= 15; y++) { setelah saya jalankan hasil yang keluar 0,2,4,6 karena program ini hanya mencetak angka genap yang lebih kecil dari 8, Perintah continue melewati nilai ganjil, dan break menghentikan perulangan pada nilai tertentu (8). Lalu selanjutnya kita mengganti kode program // ubah 2

menjadi if (y % 2 == 0) //ubah 2 maka hasil yang keluar adalah angka ganjil saja kode perintah continue hanya mencetak angka ganjil sedangkan break tidak bekerja karena pada kode tersebut karena kita melakukan break pada angka genap sehingga tidak terjadi pemberhentian di pada program namun ketika kita mengganti kode break dengan angka 9 maka luaran yang ditampilkan adalah angka ganjil yang lebih kecil dari angka 9.

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

Luaran :

1
3
5
7

1.2 pada saat kita mengganti kode program menjadi continue pertama; maka akan melompati sisa iterasi sisa perulangan saat i = 2 dan melanjutkan iterasi i berikutnya kemudian kalau kode diganti menjadi break pertama; maka akan mengakhiri seluruh perulangan for saat i mencapai 2 kemudian bila kita mengganti kode break menjadi continue kedua maka tidak akan mempengaruhi output karena hanya melanjutkan iterasi j berikutnya.

```
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 kedua; //ubah1
            } } } }
```

Luaran :

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 pada kode program tersebut kita mengganti kode //menghitung jumlah tinggi piramida dengan kode for(int t=1; t<=tinggi; t++){ maka hasil luarannya seperti berikut

```
import java.util.Scanner;
```

```
public class ForBersarang2 {
    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++){ //lebar
    //Menghitung Jumlah Tinggi Piramida
    for(int s=tinggi; s>=t; s--){ //panjang
        //Menghitung Jumlah Spasi per Baris
        System.out.print(" ");
    }
    System.out.println(); //Membuat Baris Baru
}
}
}

```

Luaran :

Masukan Input: 7

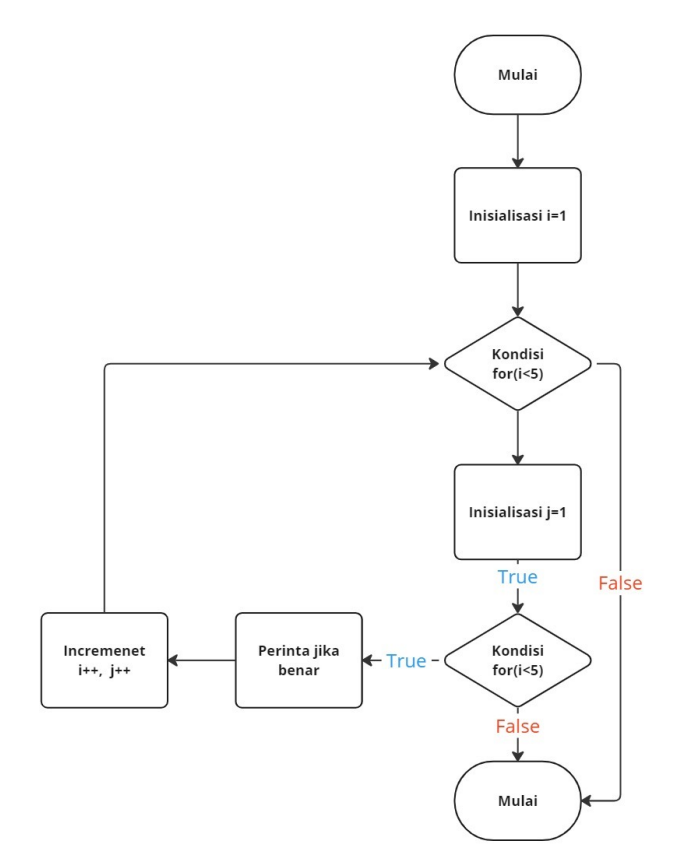
```

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

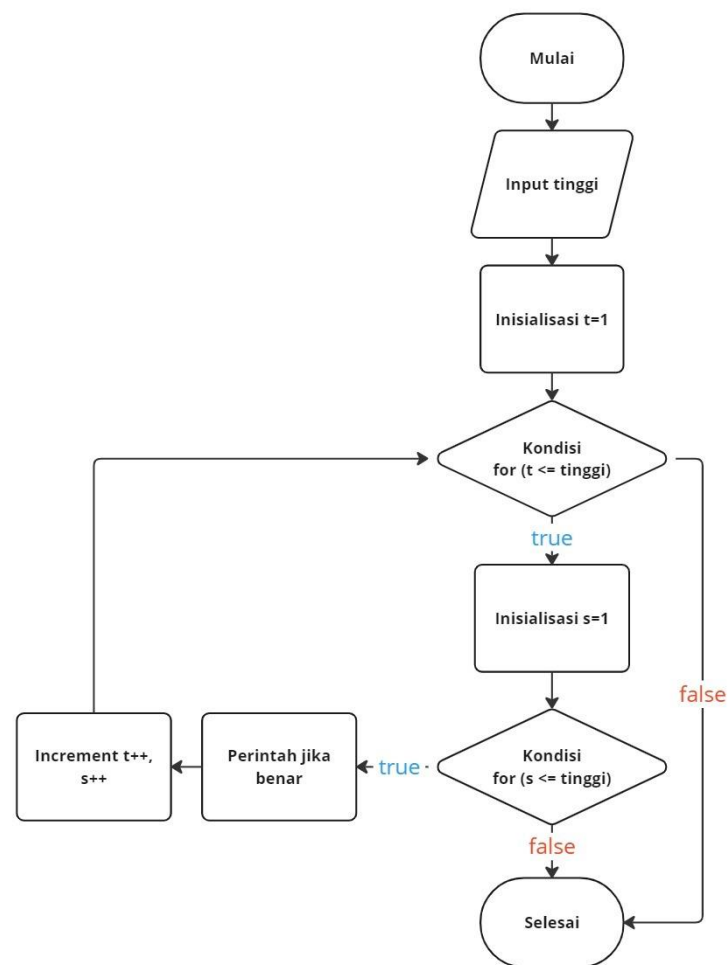
```

1.4 flowhart latihan 1.2 dan 1.3

Flowchart 1.2



Flowchart 1.3



Nomor 1 Penyusunan Algoritma dan Kode Program

- 1) Rancang desain solusi atau algoritma
 1. Mulai
 2. Masukan kelas method
 3. Masukan varibel For
 4. Luaran
 5. Selesai

- 2) Tuliskan kode program dan luaran

Contoh 1 :

```

1  public class ContohFor{
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
2  public static void main(String[] args) {
3      for (int y = 0; y <= 15; ++y) { //ubah 1
4          if (y % 2 == 0) //ubah 2
5              continue; //baris 1
6          else if (y == 9) //ubah 3
7              break; //baris 2
8          else
9              System.out.println(y + " ");
10 }
  
```

Luaran :

```
1
3
5
7
```

Contoh 2 :

```
1 public class ForBersarang {
2     public static void main(String[] args) {
3         pertama:
4             for( int i = 1; i < 5; i++) {
5
6                 kedua:
7                     for(int j = 1; j < 3; j++) {
8                         System.out.println("i = " + i + " ; j = " + j);
9                         if ( i == 2)
10                             continue kedua; //ubah1
11                     } } } }
```

Luaran :

```
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
```

Contoh 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++){ //lebar
10             //Menghitung Jumlah Tinggi Piramida
11             for(int s=tinggi; s>=t; s--){ //panjang
12                 //Menghitung Jumlah Spasi per Baris
13                 System.out.print(" ");
14             }
15             System.out.println(); //Membuat Baris Baru
16         }
17     }
18 }
```

Luaran :

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

Semua kode program berjalan sesuai perintah latihan tanpa ada masalah

Nomor 1 Kesimpulan
1) Analisa
a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program! Pada program program diatas menunjukan pentingnya alur perulanga for dengan menggunakan break dan continue pada contoh 1 dan 2 lalu pada contoh 3 memberikan contoh penggunaan struktur perulangan untuk menghasilkan pola seperti segitiga terbalik.

Nama & NPM	Topik:	Tanggal:
Alif Alfarizi G1F024069	While	9 Oktober 2024
Nomor 2 Identifikasi Masalah:		
<p>1) Uraikan permasalahan dan variabel</p> <p>Contoh 4 :</p> <pre> 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 } } } </pre> <p>Luaran:</p> <pre> 1 2 3 </pre> <p>Contoh 5 :</p> <pre> 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++; } } } </pre> <p>Luaran:</p> <pre> 0 3 6 9 12 15 18 </pre> <p>Latihan 2</p> <p>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!</p> <p>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!</p>		

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!

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

<https://www.youtube.com/watch?v=ORA4JyJMFss> – Video materi While

Nomor 2 Analisis dan Argumentasi

1) Uraikan rancangan solusi yang diusulkan.

2.1 pada program ini ketika kita mengganti kode menjadi continue tidak akan mempengaruhi output secara langsung karena kode sudah dinaikan dan dicetak sebelum continue dieksekusi sedangkan jika kode diganti break maka program akan mencetak nilai 1,2,3 sesuai dengan perulangan.

```
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  
            }  
        }  
    }  
}
```

Luaran :

1
2
3
4
5
6

2.2 Operator modulus (%) sangat berguna untuk mengecek apakah sebuah angka habis dibagi angka tertentu, yang dapat membantu dalam kondisi selektif. Output akan berubah sesuai dengan nilai pembagi yang digunakan dalam kondisi modulus. Ini memungkinkan program mencetak hanya angka yang sesuai dengan kriteria pembagi yang diinginkan, seperti kelipatan tertentu.

```
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++;  
        }  
    }  
}
```

Luaran :

0
5
10
15

2.3 pada program tersebut ketika menggunakan while tanpa do menyebabkan loop tidak dijalankan sama sekali jika kondisi awal tidak terpenuhi, sedangkan do...while memastikan blok kode dijalankan minimal sekali sebelum mengecek kondisi.

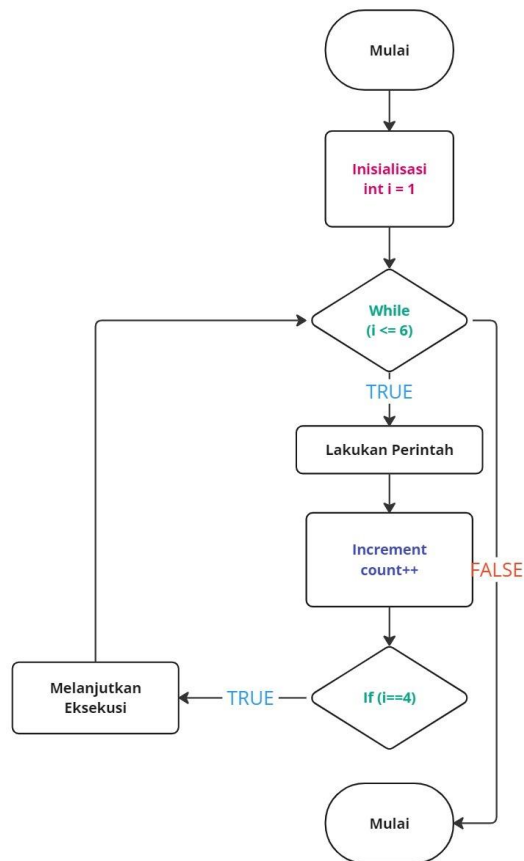
```
public class WhileBersarang {  
    public static void main(String[] args) {  
        int count = 0; //ubah1  
        do {  
            if (count % 5 == 0) //ubah2  
                System.out.println(count);  
            count++;  
        } while (count < 0);  
    }  
}
```

Luaran :

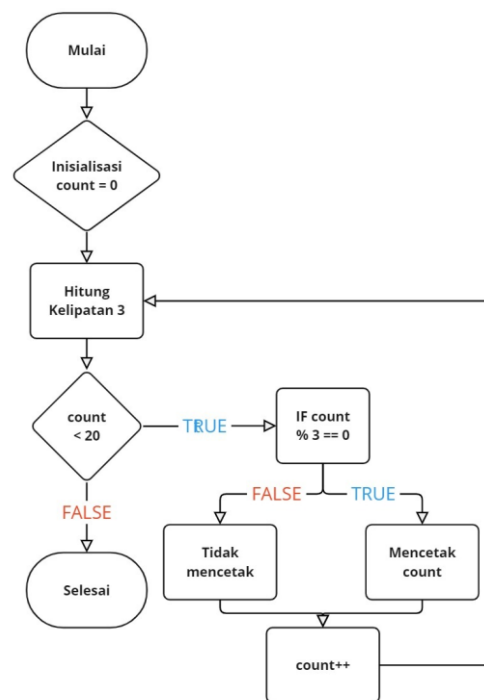
0

2.4 Flowchart latihan 2.1, contoh 5, dan latihan 2.3

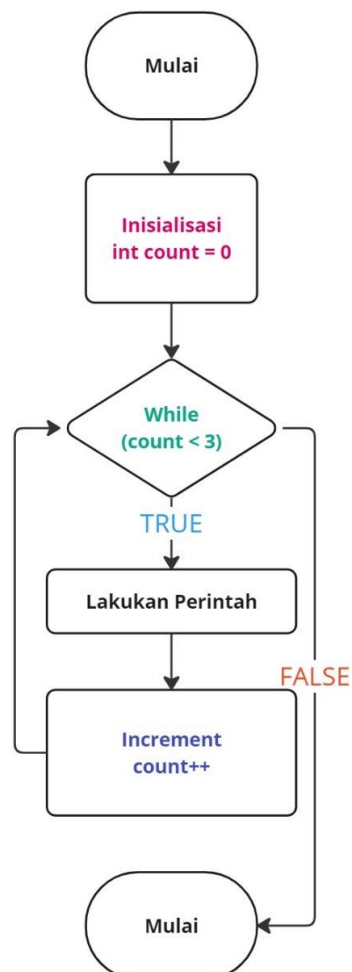
Flowchart 2.1



Flowchart contoh 5



Flowchart 2.3



Nomor 2 Penyusunan Algoritma dan Kode Program

- 1) Rancang desain solusi atau algoritma
 1. mulai
 2. inisialisasi data
 3. while
 4. increment/decrement
 5. for
 6. luaran
 7. selesai

- a. Tuliskan kode program dan luaran

Contoh 4 :

```
1 public class ContohWhile{
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
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==4){
8                 continue; //ubah1
9             }}
10    }
```

Luaran

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

Contoh 5 :

```
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 % 5 == 0) //ubah2
6                 System.out.println(count);
7             count++;
8         } while (count < 0);
9     }
10 }
```

Luaran :

```
0
```

Kode program berjalan sesuai dengan perintah dan menghasilkan luaran

Nomor 2 Kesimpulan
1) Analisa <ul style="list-style-type: none">a) Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program! Pada kode program di atas permasalahan hanya menambah kode baru dan menampilkan keluaran sesuai dengan perintah latihan dan maka akan menghasilkan keluaran yang sesuai dengan yang diinginkan.