Nama & NPM	Topik:	Tanggal:
Zahra Sari Fhadilah G1F024025	FOR	30 September 2024

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

[1.] Analisis dan Argumentasi

1) Uraikan rancangan solusi yang diusulkan.

➢ 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!

Penjelasan:

- Pada bagian pertama, saya mengubah baris //Ubah 1 menjadi for (int y = 0; y <= 15; y++). Setelah itu saya jalankan program dan amati outputnya.
- Pada bagian kedua, saya mengubah baris //Ubah 2 menjadi if (y % 2 == 0). Setelah itu saya jalankan program dan amati ouputnya.
- Yang terakhir, saya mengubah baris //Ubah 3 menjadi else if (y == 9). Setelah itu saya jalankan program dan amati outputnya.

➢ 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)!

Penjelasan:

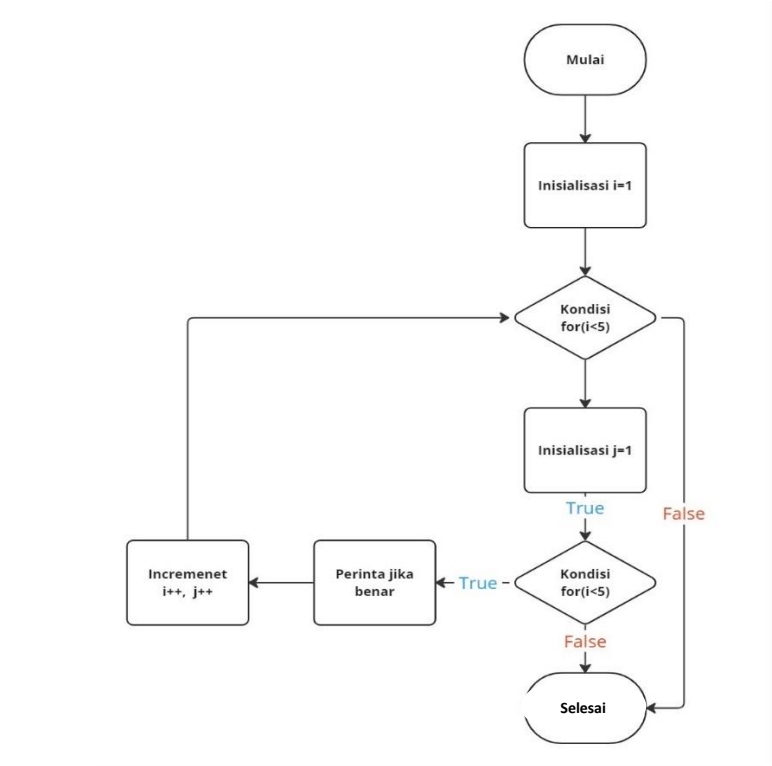
- Setelah menjalankan instruksi dari soal, saya menganalisis perbedaan dan perubahan kode program setelah melakukan beberapa perubaha.
- Pada perubahan a, Ketika i== 2, program melewati loop dalam untuk j dan langsung lanjuut ke iterasi berikutnya untuk I ==3.
- Pada perubahan poin b, Ketika I == 2 seluruh perulangan dihentikan, dan program tidak melanjutkan ke iterasi berikutnya.
- Pada perubahan ketiga atau poin c, program tetap menjalankan semua iterasi untuk j meskipun I == 2, hanya melewati satu langkah dan lanjut ke iterasi berikutnya dari j.

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

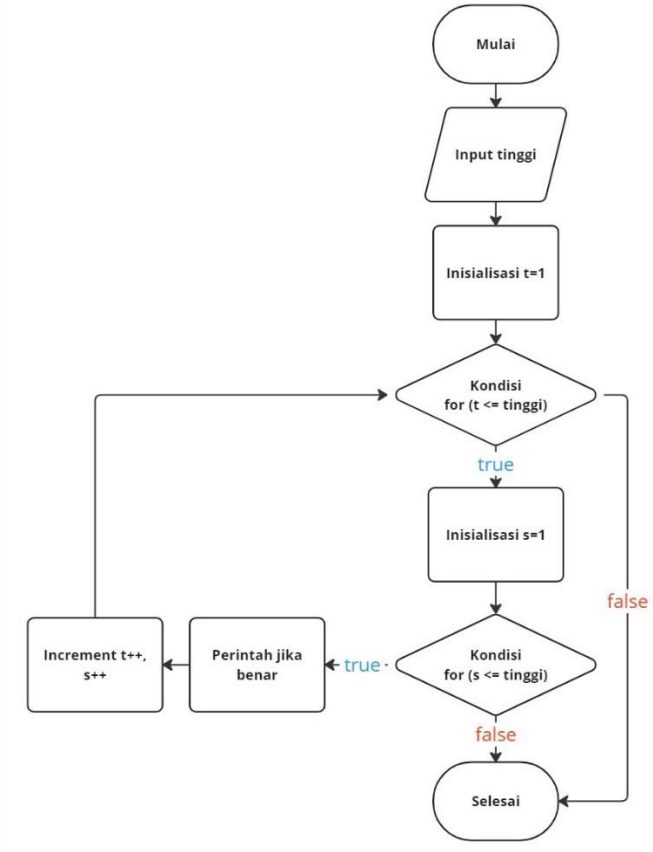
Luaran:
Masukan Input: 7
****
******
*****
****
***
**
*

- Analisa diagram flowchart dari Latihan 1.2 dan 1.3!

Flowchart latihan 1.2



Flowchart Latihan 1.3



[1.] Kode Program

Menjalankan program sesuai dengan instruksi dalam soal.

1) Tuliskan kode program dan luaran
   a) Kode program
      - Contoh 1

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

Contoh 1.1

Pada contoh 1, kode program perulangan akan menampilkan angka genap dari angkaa 0 hingga angka 6. Dimana, ketika y bernilai ganjil, maka perintah continue akan dilewati. Dan jika y == 8, perulangan akan dihentikan oleh break. Luaran yang ditampilkan adalah:

```
Output
java -cp /tmp/ddEEzqH08s/ContohFor
0
2
4
6

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

Luaran 1.1

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

Contoh 1.2

Pada kode program diatas. Ketika y diubah menjadi y <= 15, perulangan akan berjalan hingga y == 15, tetapi terhenti pada y == 9 karena ada break. Luaran yang ditampilkan adalah:

```
Output
java -cp /tmp/lt05fjk8hF/ContohFor
0
2
4
6

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

Luaran 1.2

```
1  public class ContohFor{
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 == 8)           //ubah 3
7              break;            //baris 2
8              else
9              System.out.println(y + " ");
10     }    }    }
11
```

Contoh 1.3

Pada kode program diatas, ketika y % 2 == 0, maka program hanya akan mencetak angka yang ganjil. Luaran yang ditampilkan adalah:

```
Output
java -cp /tmp/DoUwTmQFRi/ContohFor
1
3
5
7
9
11
13
15

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

Luaran 1.3

```
1  public class ContohFor{
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     }    }    }
11
```

Contoh 1.4

Pada kode program diatas ketika baris 6 diubah menjadi else if (y == 9), maka kode program akan berhenyi ketika y == 9. Luaran yang ditampilkan adalah:

```
Output
java -cp /tmp/4fQBJnJOCw/ContohFor
1
3
5
7

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

Luaran 1.4

- 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                                 break kedua;      //ubah1
11                 }    }    }   }
```

Contoh 2.1

Pada kode program ini, terdapat dua perulangan bersarang. Pada perulangan pertama, terdapat kondisi ketika i == 2, maka break akan menghentikan perulangan dalam loop kedua. Luaran yang ditampilkan adalah:

```
Output
java -cp /tmp/YAAa7TQqqQ/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 ===
```

Luaran 2.1

```
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 pertama;      //ubah1
11             }   }   }   }
```

Contoh 2.2

continue pertama; melanjutkan perulangan luar (pertama) tanpa menyelesaikan yang dalam.

```
Output
java -cp /tmp/RV5S7P5xOz/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 ===
```

Luaran 2.2

```java
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 pertama;      //ubah1
                  }    }    }   }
```

Contoh 2.3

break pertama; menghentikan seluruh perulangan luar.

```
Output

java -cp /tmp/UDagK39iMw/ForBersarang
i = 1; j = 1
i = 1; j = 2
i = 2; j = 1

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

Luaran 2.3

```java
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
                  }    }    }   }
```

Contoh 2.4

continue kedua; melanjutkan perulangan dalam tanpa menyelesaikan loop dalam.

```
Output

java -cp /tmp/dRjCck3uK1/ForBersarang
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

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

Luaran 2.4

- Contoh 3

```java
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=1; s<=t; s++){
                    //Menghitung Jumlah Spasi per Baris
                    System.out.print("*");
                }
                 System.out.println(); //Membuat Baris Baru
            }
        }
    }
```

Contoh 3.1

Pada kode program diats tengah mencetak piramida bintang sesuai dengan tinggi yang dimasukkan pengguna. Luaran yang ditampilkan adalah:

```
Output
java -cp /tmp/y7vudxPvw0/ForBersarang
Masukan Input: 7
*******
******
*****
****
***
**
*

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

Luaran 3.1

[1.] Kesimpulan

1) Kode program ini menunjukkan pengimplementasian fungsi for dalam kode program
2) Output atau luaran yang ditampilkan di layer menunjukkan hasil yang sesaui dengan kode program yang diinpput. Meskipun ada berbagai percobaan yang diinstruksikan oleh soal, akan tetapi kode program tetap dapat berjalan dan menghasilkan output yang sesuai.

Nama & NPM	Topik:	Tanggal:
Zahra Sari Fhadilah G1F024025	WHILE	1 Oktober 2024

[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.

```
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!

2) Rincikan sumber informasi yang relevan (buku / webpage)
Video Materi 2 tentang WHILE –
https://www.youtube.com/watch?v=ORA4JyJMFss

[2.] Analisis dan Argumentasi

1) Uraikan rancangan solusi yang diusulkan.

- 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!
  Penjelasan:
  Ketika break diganti menjadi continue maka kode program tidak akan menghentikan perulangan melainkan tetap melanjutkan perulangan akan tetapi tidak mencetak atau menampilkan angka 4.
  Kegunaan break adalah untuk menghentikan seluruh perulangan, sementara continue untuk melewati iterasi saat ini (4) dan melanjutkan ke iterasi berikutnya.

- 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!
  Penjelasan:
  Karena perubahan kondisi pada kode program ini membuat output yang ditampilkam dan atau di cetak adalah kelipatan dari 5. Penggunaan operator % atau operator modulus berguna untuk mengecek apakah suatu angka dapat dibagi dengan angka tertentu atau tidak.

- 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!
  Penjelasan:

  Menganalisis Struktur Kontrol While: Konstruk While memeriksa kondisi sebelum blok kode dieksekusi.
  Jika kondisi tidak terpenuhi, blok kode di dalamnya tidak akan dijalankan sama sekali.
  Pada contoh yang diberikan, menyetel kondisi pada count < 0 tidak akan menghasilkan output apa pun karena nilai awal count adalah 0.
  Dengan kata lain, 0 tidak kurang dari 0, sehingga blok while tidak akan dijalankan.
  Struktur kendali.
  while: do .

Dalam konstruksi while, kode di dalam blok do selalu dieksekusi setidaknya sekali, dan baru kemudian kondisinya diperiksa.

Jika kondisi masih belum terpenuhi setelah eksekusi pertama, maka tidak ada iterasi lebih lanjut yang terjadi.

Dalam contoh yang sama, kondisi count < 0 tetap tidak terpenuhi, namun kode dalam do hanya dieksekusi satu kali.

Namun, karena penghitungan tetap pada 0, kondisi pencetakan tidak terpenuhi dan data tidak dihasilkan.

Contoh kondisi: Dengan hitungan <;0 sebagai kondisi untuk kedua struktur menunjukkan bahwa tidak ada iterasi yang valid untuk menghasilkan keluaran.

Ini menunjukkan betapa pentingnya memilih kondisi yang tepat agar kode Anda berfungsi sebagaimana mestinya.

Argumen Kelebihan dan Kekurangan: while direkomendasikan bila Anda ingin kode Anda dijalankan hanya jika kondisi tertentu terpenuhi sejak awal.

Hal ini menghindari eksekusi kode yang tidak perlu dan bisa lebih efisien dalam beberapa kasus.

Lakukan.

while cocok jika Anda ingin menjamin bahwa kode Anda akan dieksekusi setidaknya sekali, terlepas dari apakah kondisi awal terpenuhi.

Ini berguna ketika meminta masukan dari pengguna atau melakukan operasi yang harus dilakukan setidaknya sekali.

Arti pemilihan kondisi: Pemilihan kondisi yang benar sangat penting untuk mengendalikan aliran program.

Jika kondisi tidak diatur dengan benar, perilaku yang tidak diinginkan dapat terjadi, misalnya tidak ada keluaran yang dihasilkan.

Misalnya, jika Anda ingin mengeluarkan nilai yang habis dibagi 3, Anda harus memulai dengan penghitungan yang memiliki rentang nilai positif.

Prinsip pemrograman yang baik: Pilihan antara while dan do.

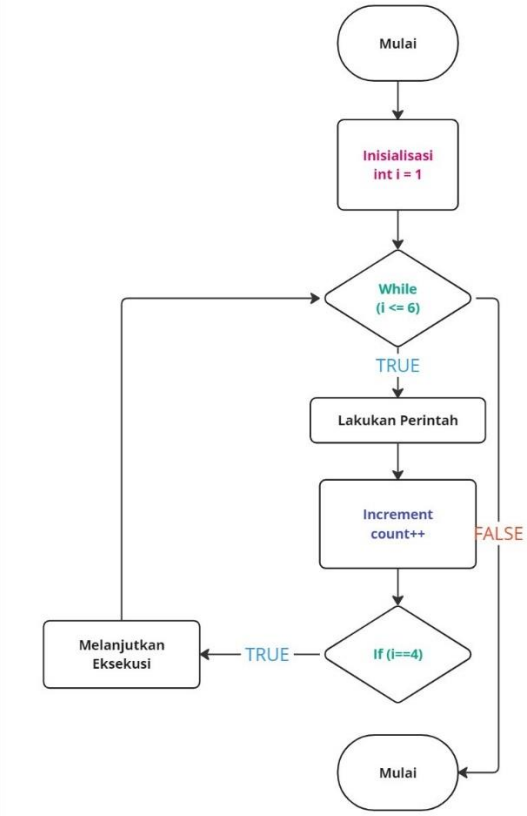
while harus didasarkan pada persyaratan logika program.

Saat memprogram dengan baik, penting untuk menulis kode yang jelas, mudah dipahami, dan efisien.

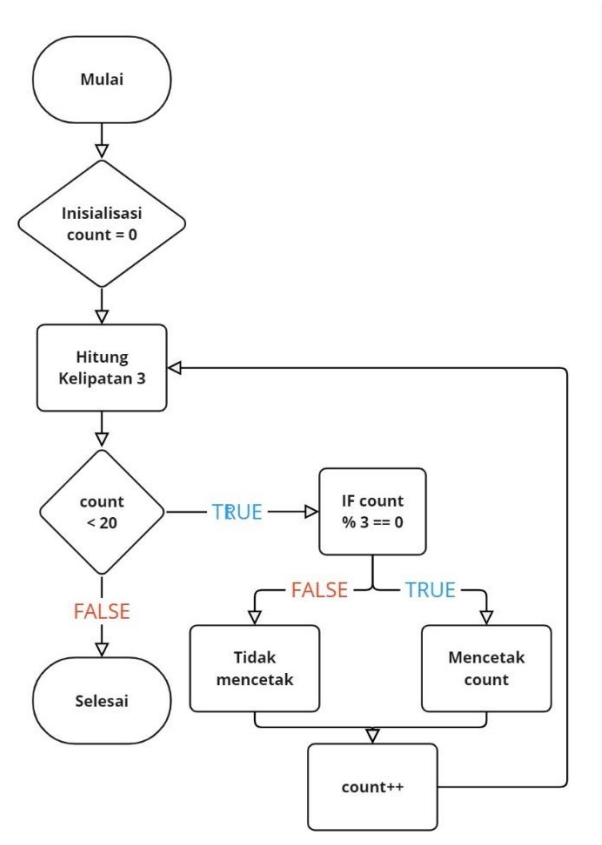
Menggunakan struktur kontrol yang tepat akan meningkatkan keterbacaan dan pemeliharaan kode Anda.

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

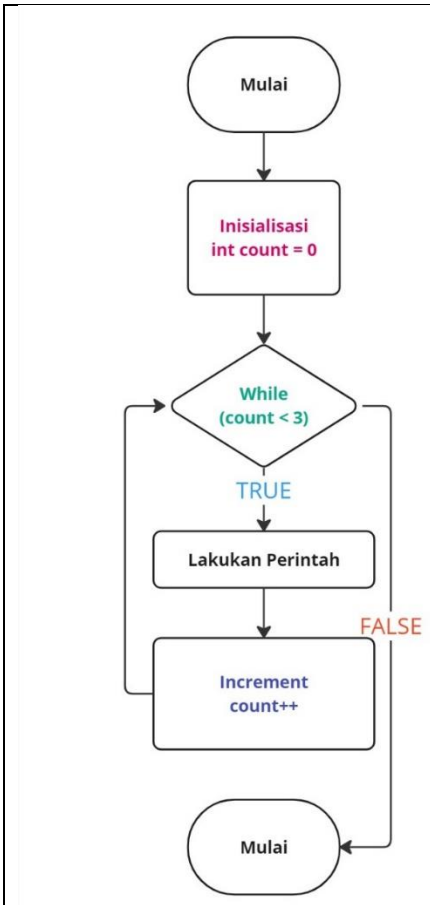
Latihan 2.1



Contoh 5



Latihan 2.3



[2.] Kode Program

1) Tuliskan kode program dan luaran
   a) Kode program

➢ Contoh 4

```
1  public class ContohWhile {
2      public static void main(String[] args) {
3          int i = 1;
4          while(i <= 6) {
5              if(i == 4) {
6                  i++;
7                  continue;           // Ubah 1
8              }
9              System.out.println(i);
10             i++;
11         }
12     }
13 }
14
```

Contoh 4.1

Pada kode program diatas ketika mengubah break menjadi continue, maka kode program tidak akan menghentikan perulangan akan tetapi akan melanjutkan perulangan tanpa mencetak nilai 4. Luaran yang ditampilkan adalah:

```
Output

java -cp /tmp/S6TBZqF5AR/ContohWhile
1
2
3
5
6

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

Luaran 4.1

- Contoh 5

```
1  public class WhileBersarang {
2      public static void main(String[] args) {
3          int count = 0;  //ubah1
4          while (count < 20) {
5          if (count % 5 == 0)  //ubah2
6          System.out.println(count);
7          count++;
8          }
9          }
10     }
```

Contoh 5.1

Ketika mengubah count % 3 == 0 menjadi count % 5== 0, kode program akan membuat kode program akan menampilkan luaran berupa kelipatan 5. Luaran yang ditampilkan adalah:

```
Output

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

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

Luaran 5.1

```java
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);

        }
    }
```

Contoh 5.2

Ketika menggunakan do..while maka kode program akan menjalankan perulangan meskipun konduisinya salah. Luaran yang ditampilkan adalah:

```
Output
java -cp /tmp/gRYklur7UY/WhileBersarang
0

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

Luaran 5.2

[2.] Kesimpulan

1) Secara keseluruhan, algoritma dan implementasi kode program ini memberikan pemahaman yang baik tentang penggunaan operator penugasan dalam Java, serta menunjukkan pentingnya kejelasan dan efisiensi dalam penulisan kode.