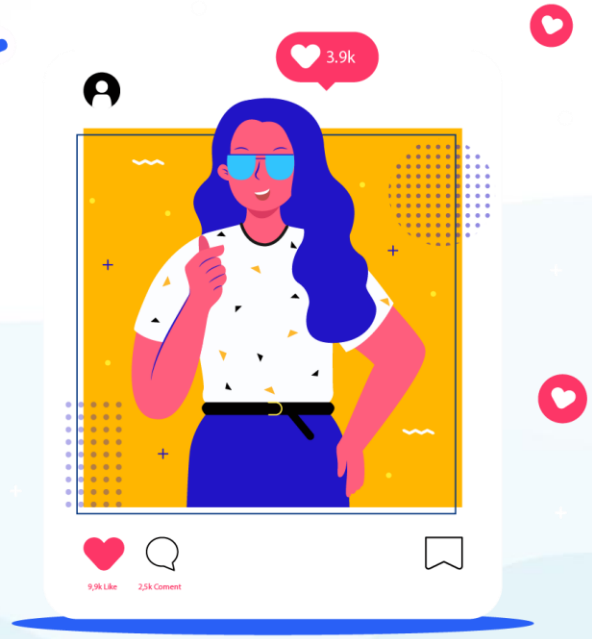
PROGRAM STUDI
SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS DIAN NUSWANTORO

MATA KULIAH
PEMROGRAMAN BERORIENTASI OBYEK



STRUKTUR KENDALI

-penyusun-

Team penyusun matkul PBO

2021

Capaian Pembelajaran

Mahasiswa dapat memahami berbagai macam struktur kontrol pada pemrograman java dan dapat menggunakannya dengan tepat.

Kemampuan Akhir yang Diharapkan

- Mahasiswa memiliki kemampuan menjelaskan penggunaan struktur kontrol pernyataan penentu keputusan if, if…else, dan switch
- Mahasiswa memiliki kemampuan menjelaskan penggunaan struktur control perulangan For, While, Do..While
- mahasiswa memiliki kemampuan menjelaskan penggunaan penanganan string serta dapat membuat program input data melalui keyboard (I/O *stream*)

Struktur Kendali / Follow of Control

"Follow of Control" merupakan suatu *keadaan* bagaimana memutuskan suatu statement dieksekusi, dimana dalam implementasinya hal tersebut dapat dituangkan dalam bentuk percabangan ataupun perulangan *(Looping)*.

Struktur kontrol digunakan untuk mengatur susunan proses eksekusi statement-statement di dalam program.

Struktur kontrol mempunyai dua tipe:

- Struktur kontrol keputusan
  Digunakan untuk memilih bagian dari code yang akan dieksekusi.

- Struktur kontrol pengulangan
  Digunanakan untuk mengeksekusi bagian tertentu sesuai dengan jumlah angka pengulangannya.

Struktur Kontrol Keputusan

Struktur kontrol keputusan
digunakan untuk memilih dan mengeksekusi block tertentu dari code yang dapat berpindah ke bagian lain.

Tipe-tipe:

- statement-if
- statement-if-else
- statement-if-else if

Contoh :

```
int time = 20;
if (time < 18) {
    System.out.println("Good day.");
} else {
    System.out.println("Good evening.");
}
```

Struktur Kontrol Keputusan

statement-if

Menspesifikasikan sebuah statement (atau block dari code) yang akan dieksekusi jika dan hanya jika statement boolean bernilai true.

Form statement-if:

```
if( boolean_ekspresi )
      statement;
```

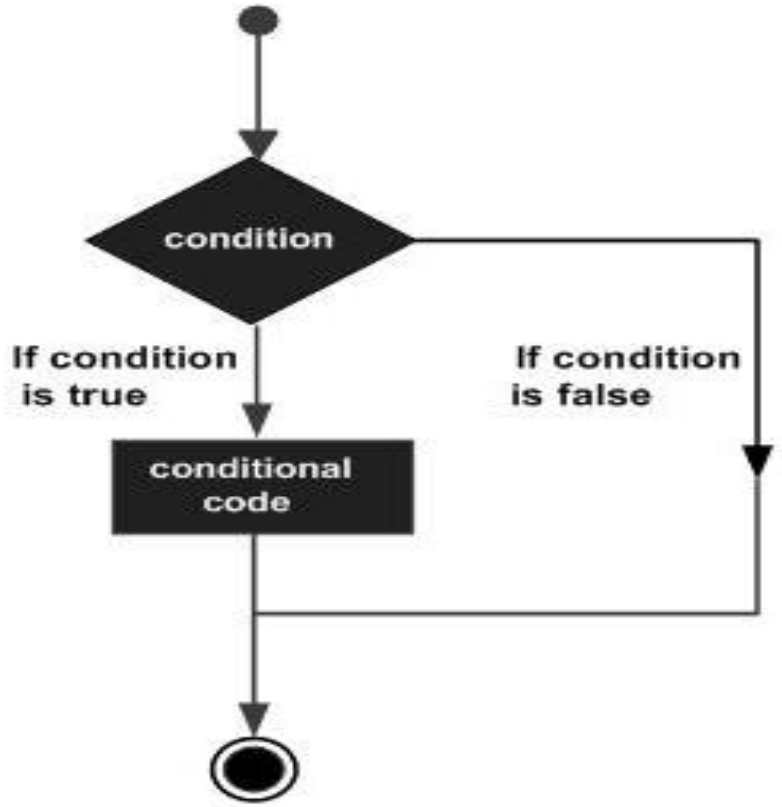
atau

```
if( boolean_ekspresi )
{
      statement 1;
      statement 2;
}
```

dimana,
boolean_ekspresi sama dengan boolean ekspresi atau boolean variabel.

Struktur Kontrol Keputusan

statement-if flowcart




```
public class IfExample
{
    public static void main(String[] args)
    {
        int age=20;

        if(age>18)
        {
            System.out.print("Age is greater than 18");
        }
    }
}
```

Struktur Kontrol Keputusan

statement if-else

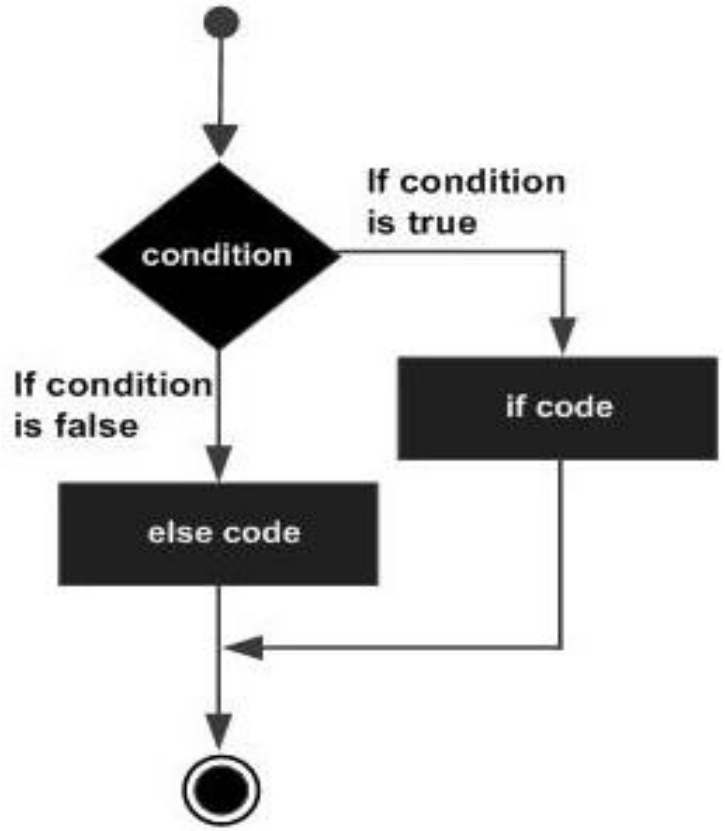
Digunakan ketika kita akan mengeksekusi sebuah statement jika kondisinya true, dan statement yang lain jika berkondisi false.

Form statement if-else:

```
if( boolean_ekspresi ){
      statement1;
      statement2;
      . . .
}
else{
      statement3;
      statement4;
      . . .
}
```

Struktur Kontrol Keputusan

statement-if-else flowcart




```java
public class IfExample02
{
    public static void main(String[] abcd)
    {
        int nilai = 60;

        if(nilai >= 60)
        {
            System.out.println("Lulus");
        }
        else
        {
            System.out.println("Gagal");
        }
    }
}
```

Struktur Kontrol Keputusan

statement if-else-if

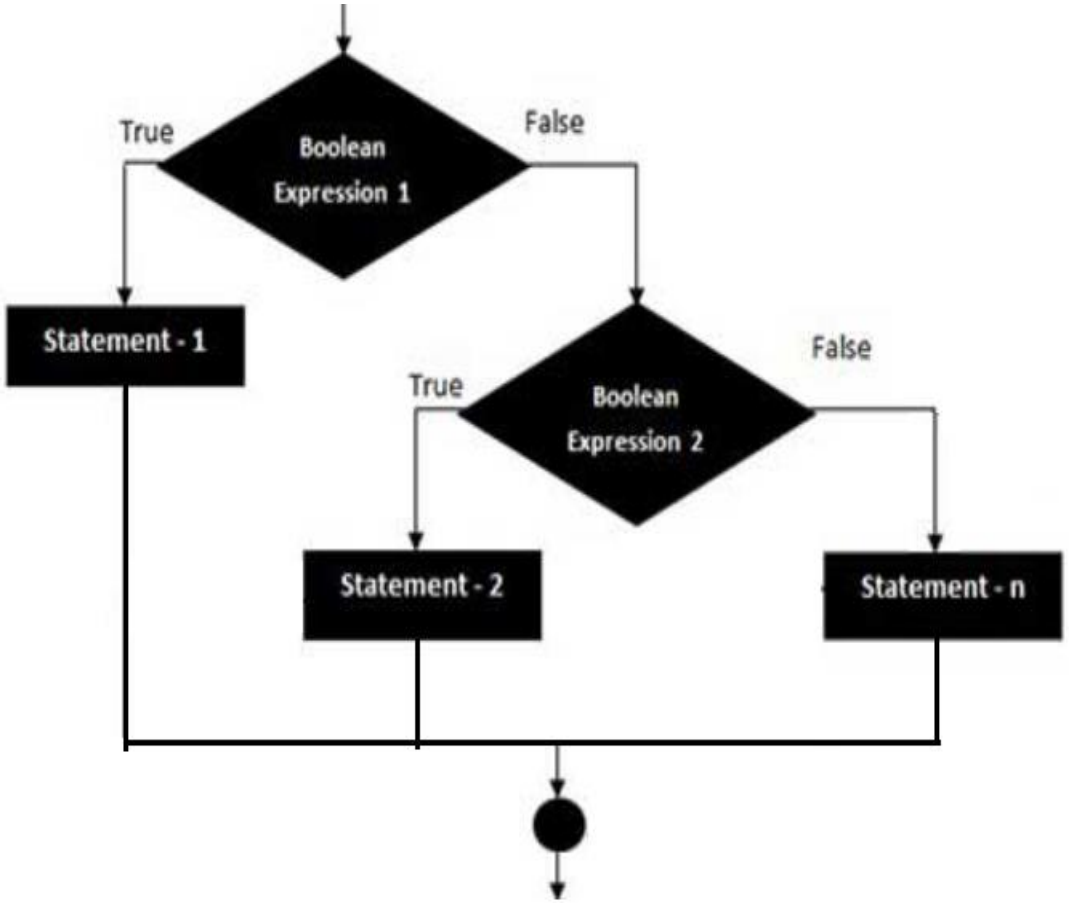
statement pada klausa else dari sebuah blok if-else dapat menjadi struktur if-else yang lain.

Struktur ini memperbolehkan kita untuk membuat pilihan yang lebih kompleks. Form statement if-else- if:

```
if( boolean_ekspresi1 )
     statement1;
else if( boolean_ekspresi2 )
     statement2;
else
     statement3;
```

Struktur Kontrol Keputusan

statement-if-else-if flowcart




```java
public class IfExample03
{
    public static void main(String[] abcd)
    {
        int nilai = 95;

        if(nilai >= 80) {
            System.out.println("Istimewa");
        }
        else {
            if(nilai >= 60) {
                System.out.println("Bagus");
            }
            else {
                System.out.println("Biasa");
            }
        }
    }
}
```

statement-switch

Switch

Memperbolehkan percabangan pada multiple outcomes.

Form statement-switch:

```
switch ( switch_ekspresi ){
      case case_pilihan1:
            statement1;//
            statement2;//blok 1
            break;
      case case_pilihan2:
            statement1;//
            statement2;//blok 2
            break;
      default:
            statement1;//
            statement2;//blok n
}
```

Dimana, ekspresi switch merupakan integer atau karakter ekspresi
case_pilihan1, case_pilihan2 dan yang lainnya, merupakan integer unique atau karakter tetap.

statement-switch

Ketika sebuah switch digunakan,
Java akan menilai ekspresi switch, kemudian berpindah ke case yang pilihan dari pemilih sesuai dengan nilai dari ekspresi.

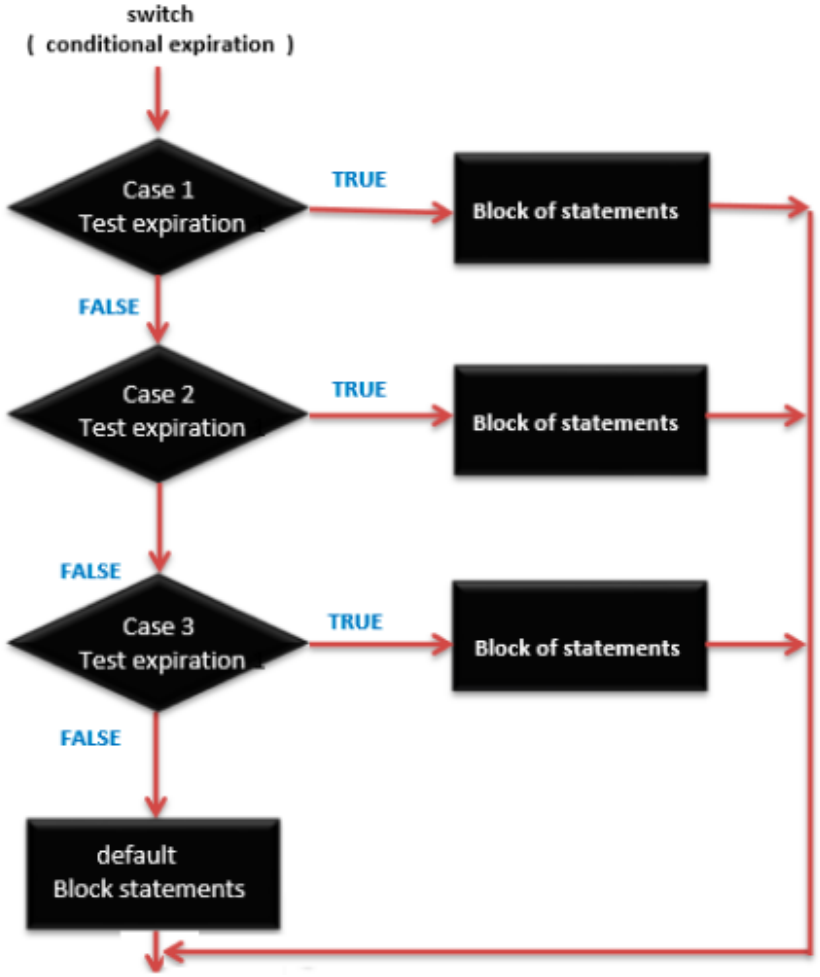
Program mengeksekusi statement yang diminta dari point sebuah case sampai statement break dibaca, kemudian pindah ke statement awal setelah membaca akhir dari struktur switch.

Jika tidak ada case yang sesuai, maka blok default akan dieksekusi. Dimana bagian default merupakan pilihan.
Catatan :

- Tidak sama dengan statement-if, statement multiple dieksekusi pada statement-switch, tanpa membutuhkan statement percabangan (braches statement).
- Ketika sebuah case pada statement-switch sesuai, semua statement yang ada didalam case tersebut akan dieksekusi. Tidak hanya itu, statement yang berhubungan dengan case tersebut juga akan dieksekusi.
- Untuk mencegah program dari pengeksekusian statement pada case sebelumnya, kita menggunakan statement-break sebagai statement akhir.

statement-switch

Switch flowchat




```java
int day = 4;
switch (day)
{
    case 1:
        System.out.println("Senin");
        break;
    case 2:
        System.out.println("Selasa");
        break;
    case 3:
        System.out.println("Rabu");
        break;
    default:
        System.out.println("diluar pilihan");
}
```

Struktur Kontrol Pengulangan

Struktur kontrol pengulangan

Pada statement Java, kita dapat menentukan nilai pengulangan yang akan dilakukan.
Tipe:

- Pengulangan-while
- Pengulangan-do-while
- Pengulangan-for

Contoh :

```
int i = 0;
while (i < 5) {
    System.out.println(i);
    i++;
}
```

Struktur Kontrol Pengulangan

Pengulangan while

Merupakan statement atau blok dari statement yang diulang selama kondisinya sesuai.

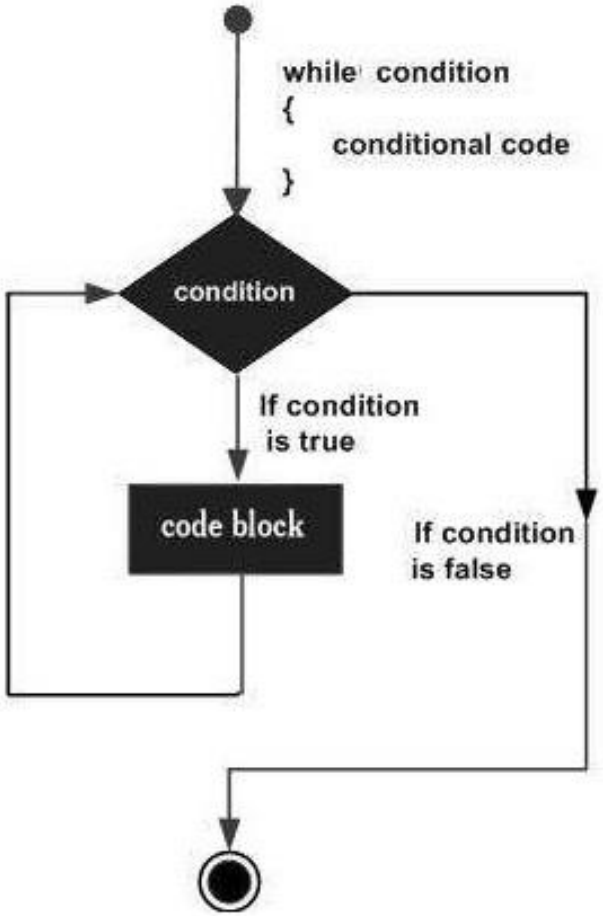
Form pengulangan while:

```
while( boolean_ekspresi ){
      statement1;
      statement2;
      . . .
}
```

statement didalam pengulangan while akan dieksekusi selama boolean_ekspresi bernilai true.

Struktur Kontrol Pengulangan

Flowchat while




```
public class Loop01
{
    public static void main(String abcd[])
    {
        int a = 0;
        while (a < 10) {
            System.out.println(a);
            a++;
        }
    }
}
```

Struktur Kontrol Pengulangan

Pengulangan do-while

statement-do-while
Sama dengan pengulangan-while, statement didalam pengulangan do-while akan dieksekusi beberapa kali selama kondisinya sesuai dengan ekspressi yang diberikan.

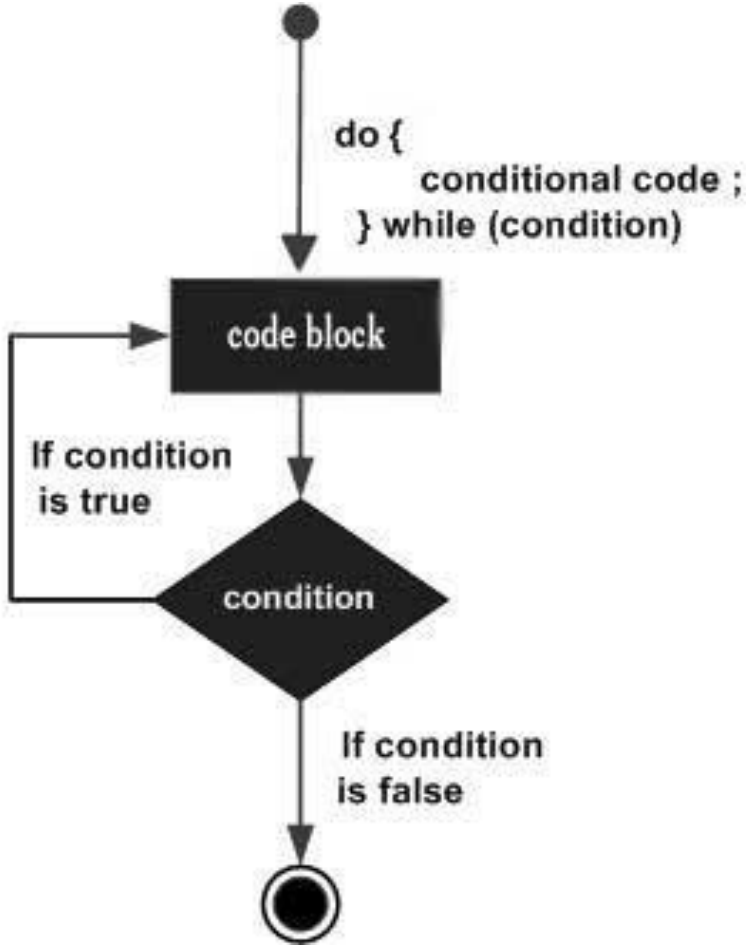
Hal utama yang membedakan antara pengulangan while dan do-while :
statement didalam pengulangan do-while loop setidaknya dieksekusi satu kali.

Bentuk penulisan pengulangan-do-while:

```
do{
        statement1;
        statement2;
        . . .
}while( boolean_ekspresi );
```

Struktur Kontrol Pengulangan

Flowchat do-while




```
public class ntLoop02
{
    public static void main(String abcd[])
    {
        int a = 0;
        do {
            System.out.println(a);
            a++;
        } while (a < 10);
    }
}
```

Struktur Kontrol Pengulangan

Pengulangan for

Digunakan untuk mengeksekusi code yang bernilai sama, berulang-ulang.

Form pengulangan-for:

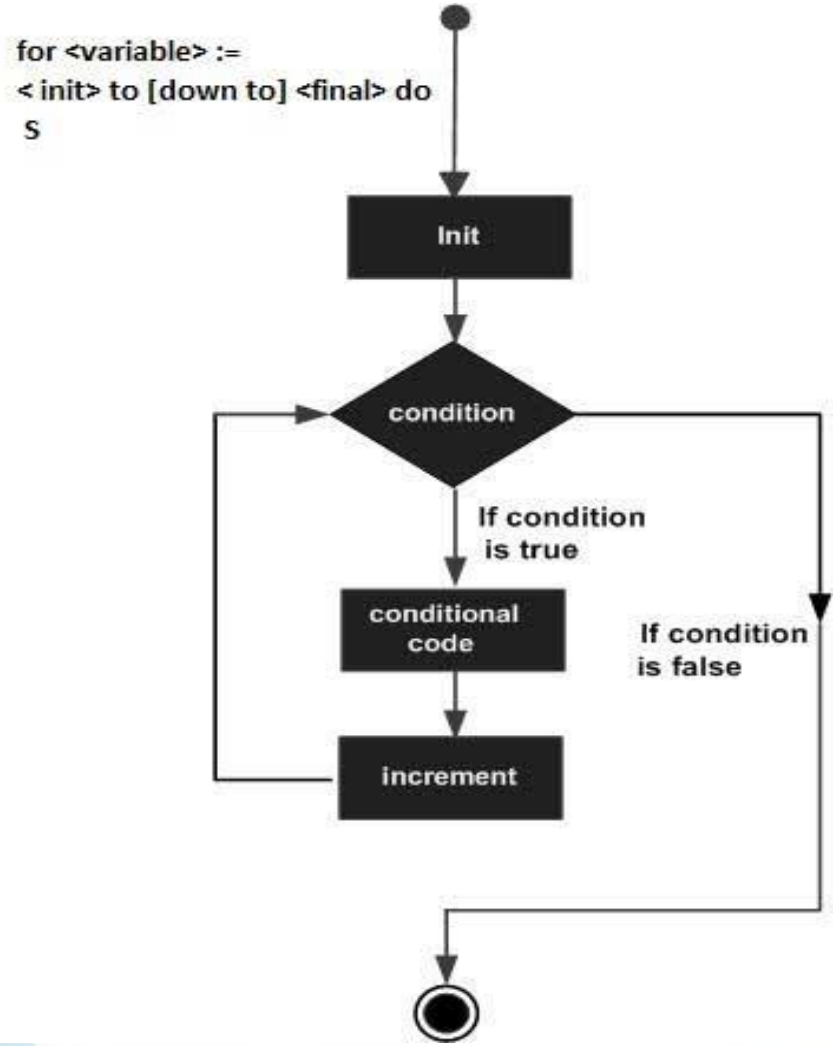
```
for(InisialisasiEkspresi;KondisiPengulangan;StepEkspresi)
{
    statement1;
    statement2;
    . . .
}
```

dimana,

- InisialisasiEkspresi – menginisialisasi variabel pengulangan.
- KondisiPengulangan – membandingkan variabel pengulangan dengan nilai limit.
- StepEkspresi – memperbarui variabel pengulangan.

Struktur Kontrol Pengulangan

Flowchat for




```
public class ntLoopFor
{
    public static void main(String abcd[])
    {
        int i;
        for(i=0; i<10; i++){
            System.out.println(i);
        }
    }
}
```

Branching statement

statement branching dapat digunakan untuk mengatur flow dari pengeksekusian program.

Java menyediakan tiga statement branching:

- break
- continue
- return.

Branching statement

Unlabeled break

Digunakan :

- Mengakhiri statement switch
- Juga dapat digunakan untuk mengakhiri pengulangan for, while, atau do-while

```
int day = 4;
switch (day)
{
    case 1:
        System.out.println("Senin");
        break;
    case 2:
        System.out.println("Selasa");
        break;
    case 3:
        System.out.println("Rabu");
        break;
    default:
        System.out.println("diluar pilihan");
}
```

```
public class ntLoopFor01
{
    public static void main(String abcd[])
    {
        int i;
        for(i=0; i<10; i++){
            if (i == 5) {
                break;
            }
            System.out.println(i);
        }
    }
}
```

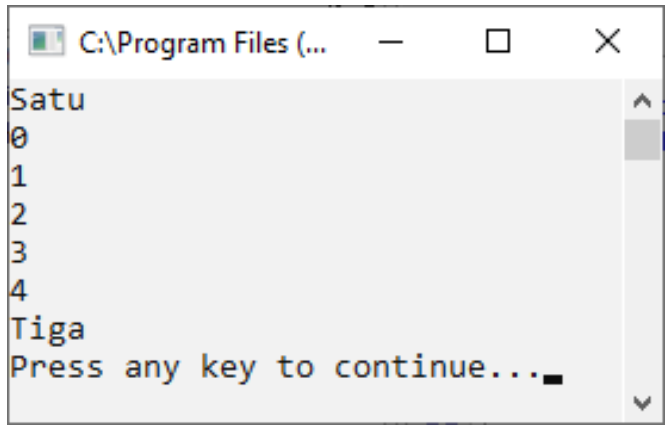
Branching statement

Labeled break statement

Digunakan :

- Mengakhiri sebuah statement, yang diidentifikasi oleh spesifikasi label pada statement break.

```
public class ntLoopFor01
{
    public static void main(String abcd[])
    {
        int a, i;
        System.out.println("Satu");

        searchLabel:
        for(a=0; a<100; a++){
            for(i=0; i<10; i++){
                if (i == 5) {
                    break searchLabel;
                }
                System.out.println(i);
            }
            System.out.println("Dua");
        }
        System.out.println("Tiga");
    }
}
```

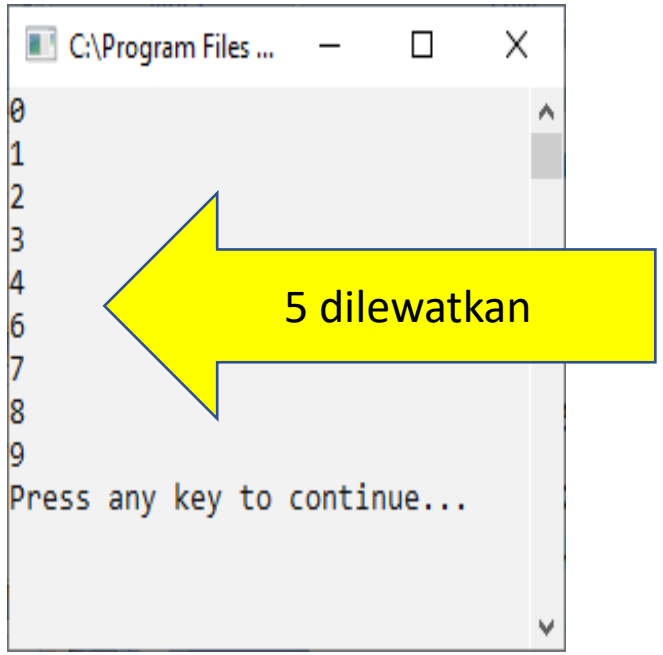


Branching statement

unlabeled continue statement

Digunakan :
Pindah ke akhir dari bagian pengulangan dan memberikan nilai boolean ekspresi yang mengontrol pengulangan tersebut, pada dasarnya perpindahan merupakan pengingat(remainder) dari iterasi yang berasal dari pengulangan.

```
public class ntContinue01
{
    public static void main(String abcd[])
    {
        for(int i=0; i<10; i++)
        {
            if (i == 5) continue;
            System.out.println(i);
        }
    }
}
```

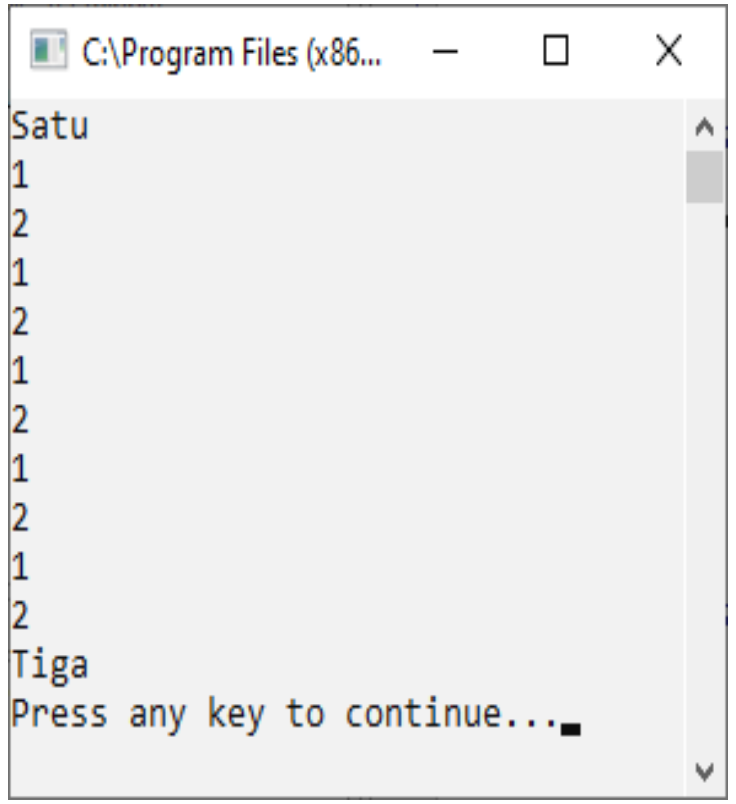


Branching statement

labeled continue statement

Digunakan : untuk memindahkan alur perulangan menuju label yang dituju.

```
public class ntContinue02
{
    public static void main(String abcd[])
    {
        System.out.println("Satu");

        outerLoop:
        for(int a=1; a<=5; a++)
        {
            for(int i=1; i<=100; i++)
            {
                if (i == 3) continue outerLoop;
                System.out.println(i);
            }
            System.out.println("Dua");
        }
        System.out.println("Tiga");
    }
}
```



Branching statement

Return statement

Digunakan : untuk keluar dari method. Mengikuti kontrol return dari statement pada method yang memanggilnya.

Return value

Memberi nilai (atau sebuah ekspresi yang menghitung sebuah nilai) setelah keyword **return**.

Contoh :

```
return ++count;
```

atau

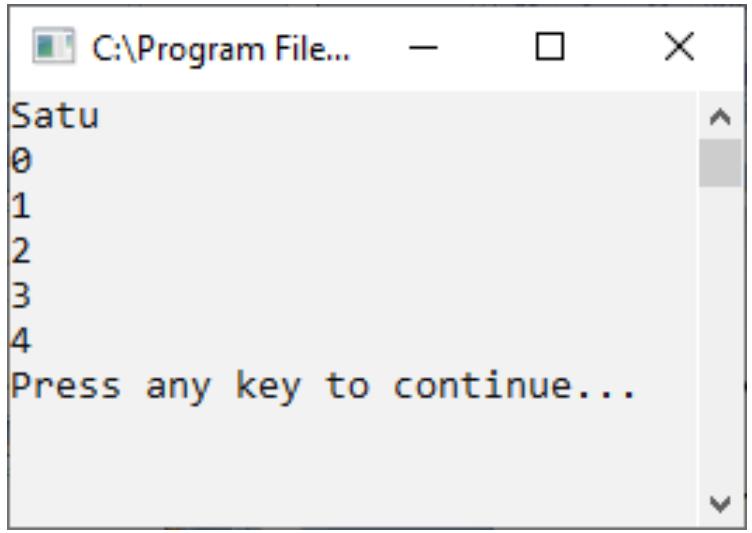
```
return "Hello";
```

Tipe data dari nilai dikembalikan oleh return harus sama dengan tipe dari pendeklarasian nilai dari method yang memanggilnya.

Branching statement

Return statement

```
public class ntReturn
{
    public static void main(String abcd[])
    {
        System.out.println("Satu");

        for(int a=0; a<100; a++)
        {
            if (a == 5) return;
            System.out.println(a);
        }

        System.out.println("Dua");
    }
}
```



Branching statement

Return statement

```
class ntReturn01
{
    public static void main(String abcd[])
    {
        ntReturn01 t = new ntReturn01();

        int tambah = t.addition(10,20);
        System.out.println("Hasil  = " + tambah);

    }

    int addition(int a,int b)
    {
        return a+b;
    }
}
```

```
public static char getGrade(double score)
{
    if (score >= 90.0)
        return 'A';
    else if (score >= 80.0)
        return 'B';
    else if (score >= 70.0)
        return 'C';
    else if (score >= 60.0)
        return 'D';
    else
        return 'F';
    }
}
```

Input dari keyboard

Pada Java terdapat beberapa cara yang dapat dilakukan untuk mendapatkan masukan dari keyboard, diantaranya yaitu menggunakan:

1. Kelas Scanner
2. Kelas BufferedREader
3. GUI (Graphical User Interface) JOptionPane

Perbedaan Scanner, BufferedReader dan GUI

- Untuk Scanner, ketika ada perhitungan matematika, maka variabel yang akan digunakan dalam perhitungan tidak perlu di konversikan lagi, bisa langsung dihitung.
- Lain halnya dengan fungsi (InputStreamReader + BufferedReader) yang perlu dikonversikan terlebih dahulu sebelum dilakukan perhitungan matematika pada variabel yang akan digunakan.
- JOptionPane yang merupakan packages dari javax.swing, digunakan untuk input dan output data berbasis GUI swing. Tampilannya memudahkan pengguna karena berupa dialog box.

Input dari keyboard

1. Kelas Scanner

Kelas Scanner berada pada paket java.util, maka harus mengimpornya terlebih dahulu dengan sintaks:

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

Metode-metode Untuk Objek Scanner :

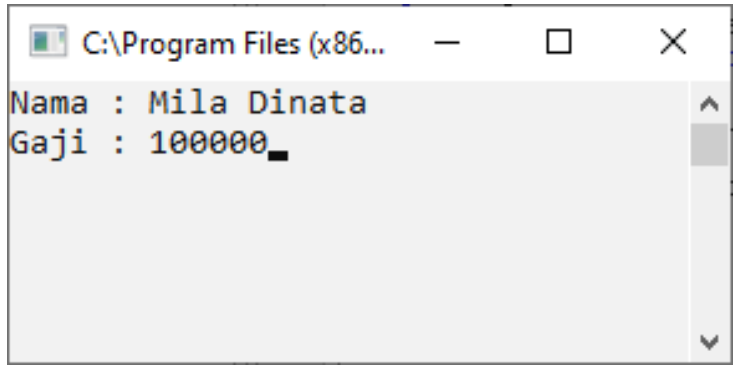
Metode	Penjelasan
nextByte()	Membaca suatu integer bertipe byte
nextShort()	Membaca suatu integer bertipe short
nextInt()	Membaca suatu integer bertipe Int
nextLong()	Membaca suatu integer bertipe long
nextFloat()	Membaca suatu angka pecahan bertipe float
nextDouble()	Membaca suatu angka pecahan bertipe double
next()	Membaca suatu string yang berakhir dengan karakter spasi
nextLine()	Membaca sebaris teks (suatu string yang berakhir dengan enter)

Input dari keyboard

1. Kelas Scanner

Contoh :

```
import java.util.Scanner;

public class InputCsannernt01 {

    public static void main(String[] args)
    {
        String nama, alamat;
        int usia, gaji;

        Scanner keyboard = new Scanner(System.in);

        System.out.print("Nama : ");
        nama = keyboard.nextLine();

        System.out.print("Gaji : ");
        gaji = keyboard.nextInt();
    }
}
```



Input dari keyboard

2. Kelas BufferedREader

Untuk menggunakan BufferedReader perlu mengimport :

- `import java.io.BufferedReader;`
- `import java.io.InputStreamReader;`
- `import java.io.IOException;`

Kelas diatas : BufferedReader, InputStreamReader dan IOException terletak pada package java.io.
Pada class BufferedReader terdapat method readLine() yang berfungsi sebagai penerima inputan dari keyboard.

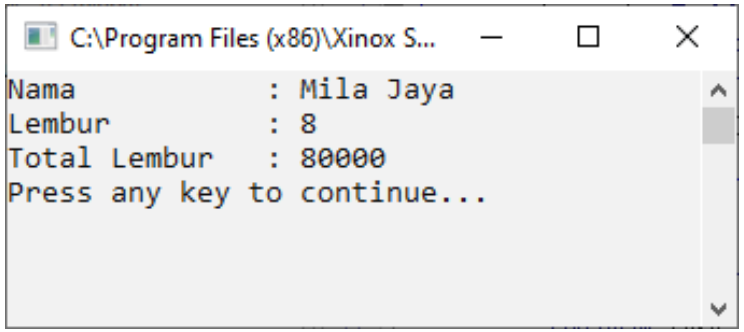
Contoh :

```
System.out.print("Masukkan Nama : ");
String name = input.readLine();
```

Input dari keyboard

Kelas BufferedREader

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class InputBufferReader
{
    public static void main(String[] args) throws IOException {

        // membuat objek bufferreader
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

        System.out.print("Nama            : ");
        String snama = br.readLine();

        System.out.print("Lembur          : ");
        String slembur = br.readLine();

        int lembur = Integer.parseInt(slembur);

        System.out.println("Total Lembur    : " + (lembur * 10000));
    }
}
```



Input dari keyboard

3. JOptionPane

Input Dengan Menggunakan GUI (Graphical User Interface) JOptionPane terletak pada javax.swing package. Dengan menggunakan JOptionPane dapat mempermudah dengan memunculkan dialog box yang dapat memasukkan sebuah nilai atau menginformasikan sesuatu.

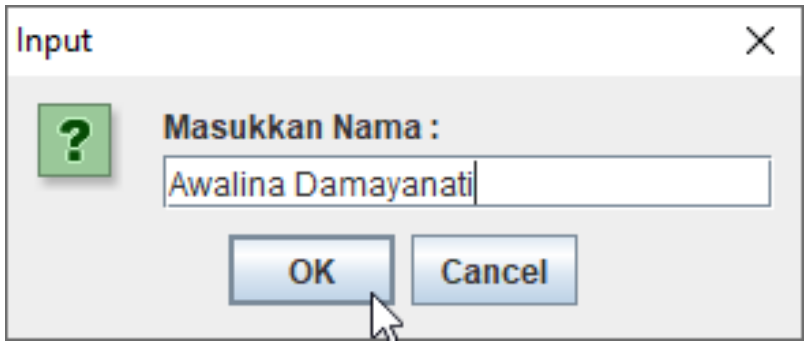
Contoh :

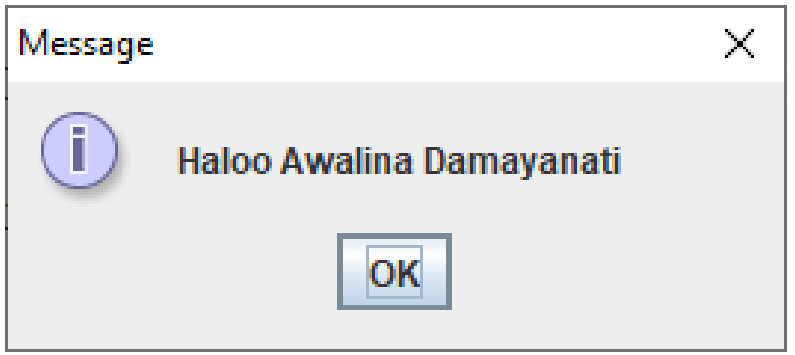
```
String nama = JOptionPane.showInputDialog("Masukkan Nama : ");
JOptionPane.showMessageDialog(null, "Haloo " + nama);
```

Input dari keyboard

3. JOptionPane

```
import javax.swing.JOptionPane;

public class ContohJOptionPane {
    public static void main(String abcdsaja[])
    {
        String nama = JOptionPane.showInputDialog("Masukkan Nama : ");
        JOptionPane.showMessageDialog(null, "Haloo " + nama);
    }
}
```





RANGKUMAN

Struktur program pada java dapat digunakan untuk membuat program agar dapat berjalan sesuai dengan rancangan alur yang dikehendali oleh pengguna.

Java memberikan beberapa pilihan untuk membuat inputan data melalui keyboard yang dapat digunakan sesuai dengan kebutuan. Dengan inputan ini maka struktur program yang digunakan dapat dikendalikan sesuai dengan masukan dari pemakai.

SUMBER PUSTAKA

- https://www.w3schools.com/java/java_type_casting.asp
- http://www.java2s.com/
- https://www.w3schools.com/java/java_conditions.asp
- https://www.javatpoint.com/java-if-else

Sumber gambar:
www.freepik.com

THANKS

ANY QUESTIONS?