

**LAPORAN**  
**PEMROGRAMAN BERBASIS OBJEK**



Nama : Achmad Zulfikar

Nim : 13020220007

Kelas : A1

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS MUSLIM INDONESIA**  
**MAKASSAR**  
**2024**

## 1. Kode Program

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

import javax.swing.*;

public class BacaString {

    public static void main(String[] args) throws IOException {
        // TODO Auto-generated method stub
        /* Kamus */
        String str;

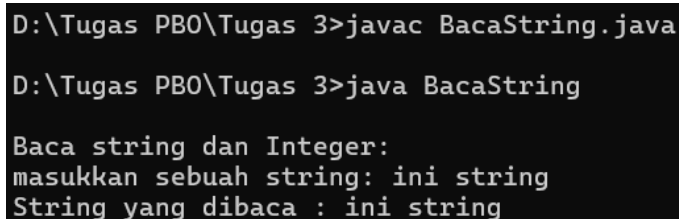
        BufferedReader dataIn = new BufferedReader(new
        InputStreamReader(System.in));

        /* Program */

        System.out.print ("\nBaca string dan Integer: \n");
        System.out.print("masukkan sebuah string: ");
        str= dataIn.readLine();
        System.out.print ("String yang dibaca : "+ str);

    }
}
```

- Output :



```
D:\Tugas PBO\Tugas 3>javac BacaString.java
D:\Tugas PBO\Tugas 3>java BacaString

Baca string dan Integer:
masukkan sebuah string: ini string
String yang dibaca : ini string
```

- Penjelasan :

### 1. Import Library:

- `java.io.BufferedReader`: Digunakan untuk membaca input dari aliran masukan, seperti dari keyboard.

- `java.io.IOException`: Digunakan untuk menangani pengecualian yang mungkin terjadi saat melakukan operasi masukan/keluaran (I/O).
- `javax.swing.JOptionPane`: Tidak digunakan dalam program ini, sehingga dapat dihapus.

## 2. Deklarasi Kelas:

- Kelas utama dalam program ini diberi nama `BacaString`.

## 3. Metode Utama:

- Metode utama dalam Java adalah `main(String[] args)`.
- `throws IOException`: Dideklarasikan pada metode `main` karena kita menggunakan `BufferedReader` yang dapat menimbulkan `IOException` jika terjadi kesalahan pada operasi masukan/keluaran.

## 4. Inisialisasi Objek `BufferedReader`:

- `BufferedReader datAIn = new BufferedReader(new InputStreamReader(System.in));`: Objek `BufferedReader` digunakan untuk membaca input dari sistem masukan (keyboard) dan diinisialisasi dengan objek `InputStreamReader` yang akan membaca input dari `System.in` (keyboard).

## 5. Meminta Input dari Pengguna:

- `System.out.print("masukkan sebuah string: ");`: Menginstruksikan pengguna untuk memasukkan sebuah string.
- `str = datAIn.readLine();`: Membaca string yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `str`.

## 6. Menampilkan String yang Dibaca:

- `System.out.print("String yang dibaca : " + str);`: Menampilkan kembali string yang dimasukkan oleh pengguna.

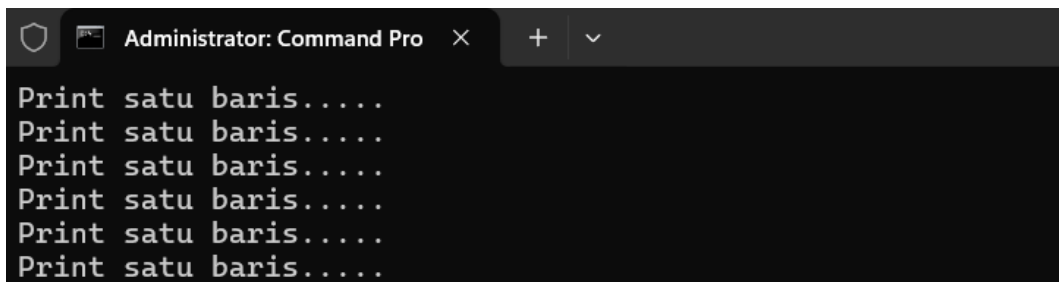
## Tujuan Program:

- Program ini dibuat untuk membantu pengguna dalam membaca sebuah string yang dimasukkan dari keyboard dan menampilkannya kembali ke layar.

## 2. Kode Program

```
public class ForEver {  
  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        /* Program */  
        System.out.println("Program akan looping, akhiri dengan ^c");  
        while (true)  
        { System.out.print ("Print satu baris.....\n");  
        }  
    }  
}
```

- Output :



- Penjelasan :

### 1. Deklarasi Kelas:

- Kelas utama dalam program ini diberi nama ForEver.

### 2. Metode Utama:

- Metode utama dalam Java adalah main(String[] args).

### 3. Pesan Awal:

- System.out.println("Program akan looping, akhiri dengan ^c");: Menampilkan pesan kepada pengguna bahwa program akan melakukan looping tak terbatas dan dapat diakhiri dengan menekan tombol "Ctrl + C".

### 4. Loop Tak Terbatas:

- while (true) { ... }: Ini adalah struktur loop yang akan terus berjalan selama kondisinya true. Karena kondisi true tanpa batas, maka loop ini akan terus berjalan tanpa berhenti.

- Di dalam loop, terdapat perintah untuk mencetak satu baris teks menggunakan `System.out.print("Print satu baris.....\n");`.

Tujuan Program:

- Tujuan dari program ini adalah untuk menunjukkan bagaimana membuat sebuah loop tak terbatas menggunakan struktur kontrol `while (true)`.

3. Kode Program :

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

```
public class If1 {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* Kamus */
        Scanner masukan=new Scanner(System.in);
        int a;

        /* Program */
        System.out.print ("Contoh IF satu kasus \n");
        System.out.print ("Ketikkan suatu nilai integer : ");
        a = masukan.nextInt();
        if (a >= 0)
            System.out.print ("\nNilai a positif "+ a);
        }
    }
}
```

- Output :

```
D:\Tugas PBO\Tugas 3>javac If1.java
D:\Tugas PBO\Tugas 3>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : 23

Nilai a positif 23
```

- Penjelasan :

1. Import Library:

- `java.util.Scanner`: Digunakan untuk membaca input dari pengguna.

## 2. Deklarasi Kelas:

- Kelas utama dalam program ini diberi nama `If1`.

## 3. Metode Utama:

- Metode utama dalam Java adalah `main(String[] args)`.

## 4. Deklarasi Variabel:

- `Scanner masukan = new Scanner(System.in);`: Membuat objek `Scanner` untuk membaca input dari pengguna melalui keyboard.
- `int a;`: Mendeklarasikan variabel `a` yang akan digunakan untuk menyimpan nilai integer yang dimasukkan oleh pengguna.

## 5. Pesan Awal:

- `System.out.print ("Contoh IF satu kasus \n");`: Menampilkan pesan kepada pengguna tentang contoh penggunaan struktur kontrol `if` dalam sebuah kasus.

## 6. Meminta Input dari Pengguna:

- `System.out.print ("Ketikkan suatu nilai integer : ");`: Meminta pengguna untuk memasukkan nilai integer.

## 7. Pengecekan Nilai:

- `a = masukan.nextInt();`: Membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `a`.
- `if (a >= 0)`: Memulai struktur kontrol `if` untuk memeriksa apakah nilai `a` lebih besar dari atau sama dengan 0.
- `System.out.print ("\nNilai a positif "+ a);`: Jika nilai `a` positif (atau sama dengan 0), maka pesan "Nilai a positif" beserta nilai `a` akan ditampilkan.

## Tujuan Program:

- Tujuan dari program ini adalah untuk memberikan contoh penggunaan struktur kontrol `if` dalam sebuah kasus sederhana, yaitu untuk menentukan apakah suatu nilai integer yang dimasukkan oleh pengguna positif atau tidak.

## 4. Kode Program :

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

```

/* contoh pemakaian IF dua kasus komplementer */
/* Membaca sebuah nilai, */
/* menuliskan 'Nilai a positif , nilai a', jika a >=0 */
/* 'Nilai a negatif , nilai a', jika a <0 */

public class If2 {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* Kamus */

        int a;
        Scanner masukan=new Scanner(System.in);
        /* Program */
        System.out.print ("Contoh IF dua kasus \n");
        System.out.print ("Ketikkan suatu nilai integer :");
        a=masukan.nextInt();

        if (a >= 0){
            System.out.println ("Nilai a positif "+ a);
        }else /* a< 0 */

        {
            System.out.println ("Nilai a negatif "+ a);
        }
    }
}

```

- Output :

```

D:\Tugas PB0\Tugas 3>javac If2.java
D:\Tugas PB0\Tugas 3>java If2
Contoh IF dua kasus
Ketikkan suatu nilai integer :34
Nilai a positif 34

```

- Penjelasan :

1. Import Library:

- `java.util.Scanner`: Digunakan untuk membaca input dari pengguna.

## 2. Deklarasi Kelas:

- Kelas utama dalam program ini diberi nama `If2`.

## 3. Metode Utama:

- Metode utama dalam Java adalah `main(String[] args)`.

## 4. Deklarasi Variabel:

- `int a`; Mendeklarasikan variabel `a` yang akan digunakan untuk menyimpan nilai integer yang dimasukkan oleh pengguna.
- `Scanner masukan = new Scanner(System.in)`; Membuat objek `Scanner` untuk membaca input dari pengguna melalui keyboard.

## 5. Pesan Awal:

- `System.out.print ("Contoh IF dua kasus \n");`; Menampilkan pesan kepada pengguna tentang contoh penggunaan struktur kontrol `if-else` dalam dua kasus.

## 6. Meminta Input dari Pengguna:

- `System.out.print ("Ketikkan suatu nilai integer :");`; Meminta pengguna untuk memasukkan nilai integer.

## 7. Pengecekan Nilai:

- `if (a >= 0) { ... }`: Jika nilai `a` lebih besar dari atau sama dengan 0, maka blok kode di dalam kurung kurawal akan dieksekusi. Program akan menampilkan pesan "Nilai `a` positif" beserta nilai `a`.
- `else { ... }`: Jika nilai `a` kurang dari 0, maka blok kode di dalam kurung kurawal setelah `else` akan dieksekusi. Program akan menampilkan pesan "Nilai `a` negatif" beserta nilai `a`.

## Tujuan Program:

- Tujuan dari program ini adalah untuk memberikan contoh penggunaan struktur kontrol `if-else` dalam dua kasus yang komplementer, yaitu untuk menentukan apakah suatu nilai integer yang dimasukkan oleh pengguna positif atau negatif.

## 5. Kode Program :

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

```
/* contoh pemakaian IF tiga kasus */ /* Membaca sebuah nilai, */
```



```

/* menuliskan 'Nilai a positif , nilai a', jika a >0 */
/* 'Nilai Nol , nilai a', jika a = 0 */
/* 'Nilai a negatif , nilai a', jika a <0 */

public class If3 {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* Kamus */
        int a;

        Scanner masukan=new Scanner(System.in);
        /* Program */
        System.out.print ("Contoh IF tiga kasus \n");
        System.out.print ("Ketikkan suatu nilai integer :");
        a=masukan.nextInt();
        if (a > 0){
            System.out.println ("Nilai a positif "+ a);
        }else if (a == 0){
            System.out.println ("Nilai Nol "+ a);
        }else /* a > 0 */ {
            System.out.println ("Nilai a negatif "+ a);
        }
    }
}

```

- Output :

```

D:\Tugas PBO\Tugas 3>javac If3.java
D:\Tugas PBO\Tugas 3>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer :45
Nilai a positif 45

```

- Penjelasan :

1. Import Library:

- `java.util.Scanner`: Digunakan untuk membaca input dari pengguna.

## 2. Deklarasi Kelas:

- Kelas utama dalam program ini diberi nama `If3`.

## 3. Metode Utama:

- Metode utama dalam Java adalah `main(String[] args)`.

## 4. Deklarasi Variabel:

- `int a;` Mendeklarasikan variabel `a` yang akan digunakan untuk menyimpan nilai integer yang dimasukkan oleh pengguna.
- `Scanner masukan = new Scanner(System.in);` Membuat objek `Scanner` untuk membaca input dari pengguna melalui keyboard.

## 5. Pesan Awal:

- `System.out.print ("Contoh IF tiga kasus \n");`  
Menampilkan pesan kepada pengguna tentang contoh penggunaan struktur kontrol `if-else if-else` dalam tiga kasus berbeda.

## 6. Meminta Input dari Pengguna:

- `System.out.print ("Ketikkan suatu nilai integer :");`  
Meminta pengguna untuk memasukkan nilai integer.

## 7. Pengecekan Nilai:

- `if (a > 0) { ... }`: Jika nilai `a` lebih besar dari 0, maka blok kode di dalam kurung kurawal pertama akan dieksekusi. Program akan menampilkan pesan "Nilai a positif" beserta nilai `a`.
- `else if (a == 0) { ... }`: Jika nilai `a` sama dengan 0, maka blok kode di dalam kurung kurawal kedua akan dieksekusi. Program akan menampilkan pesan "Nilai Nol" beserta nilai `a`.
- `else { ... }`: Jika nilai `a` kurang dari 0, maka blok kode di dalam kurung kurawal terakhir akan dieksekusi. Program akan menampilkan pesan "Nilai a negatif" beserta nilai `a`.

## Tujuan Program:

- Tujuan dari program ini adalah untuk memberikan contoh penggunaan struktur kontrol `if-else if-else` dalam tiga kasus yang berbeda terkait nilai integer yang dimasukkan oleh pengguna. Program ini memberikan pemahaman dasar tentang bagaimana menggunakan struktur kontrol tersebut untuk membuat keputusan berdasarkan kondisi yang berbeda.

6. Kode Program :

```
public class KasusBoolean {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
        /* Kamus */  
        boolean bool;  
  
        /* Algoritma */  
        bool= true;  
        if(bool) {  
            System.out.print("true\n");  
        } else  
            System.out.print("false\n");  
        if(!bool) {  
            System.out.print("salah\n");  
        } else  
            System.out.print("benar\n");  
    }  
}
```

- Output :

```
D:\Tugas PBO\Tugas 3>javac KasusBoolean.java  
D:\Tugas PBO\Tugas 3>java KasusBoolean  
true  
benar
```

- Penjelasan :

1. Deklarasi Kelas:

- Kelas utama dalam program ini diberi nama KasusBoolean.

2. Metode Utama:

- Metode utama dalam Java adalah main(String[] args).

3. Deklarasi Variabel:

- boolean bool;; Mendeklarasikan variabel bool dengan tipe data boolean.

4. Inisialisasi Variabel:

- `bool = true;`: Menginisialisasi variabel `bool` dengan nilai `true`.

#### 5. Pengecekan Nilai Boolean:

- `if (bool) { ... }`: Jika nilai boolean `bool` adalah `true`, maka blok kode di dalam kurung kurawal pertama akan dieksekusi. Program akan menampilkan pesan `"true"`.
- `else { ... }`: Jika nilai boolean `bool` adalah `false`, maka blok kode di dalam kurung kurawal kedua akan dieksekusi. Namun, karena nilai `bool` dalam kasus ini adalah `true`, blok ini akan dilewati.
- `if (!bool) { ... }`: Jika nilai boolean `bool` adalah `false` (dengan menggunakan operator negasi `!`), maka blok kode di dalam kurung kurawal ketiga akan dieksekusi. Namun, karena nilai `bool` dalam kasus ini adalah `true`, blok ini akan dilewati.
- `else { ... }`: Jika nilai boolean `bool` adalah `true`, maka blok kode di dalam kurung kurawal keempat akan dieksekusi. Program akan menampilkan pesan `"benar"`.

#### Tujuan Program:

- Tujuan dari program ini adalah untuk memberikan contoh penggunaan tipe data boolean dan struktur kontrol `if-else` untuk mengecek dan menampilkan pesan berdasarkan nilai boolean yang diberikan. Program ini membantu dalam memahami bagaimana tipe data boolean dapat digunakan untuk mengekspresikan keadaan logika (`true/false`) dan bagaimana struktur kontrol `if-else` dapat digunakan untuk membuat keputusan berdasarkan nilai boolean.

#### 7. Kode Program :

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

```
public class KasusSwitch {
```

```
    public static void main(String[] args) {
```

```
        // TODO Auto-generated method stub
```

```
        /* Kamus */
```

```
        char cc;
```

```
        Scanner masukan=new Scanner(System.in);
```

```
        /* Program */
```

```
        System.out.print ("Ketikkan sebuah huruf, akhiri dengan  
RETURN \n");
```

```

cc=masukan.next().charAt(0);
switch (cc) {
    case 'a': { System.out.print (" Yang anda ketik adalah
a \n"); break;}
    case 'u': { System.out.print (" Yang anda ketik adalah
u \n"); break;}
    case 'e': { System.out.print (" Yang anda ketik adalah
e \n"); break;}
    case 'i': { System.out.print (" Yang anda ketik adalah
i \n"); break;}
    case 'o': { System.out.print (" Yang anda ketik adalah
o \n"); break;}

    default:
        System.out.print (" Yang anda ketik adalah huruf mati
\n");
    }
}
}

```

- Output :

```

D:\Tugas PB0\Tugas 3>javac KasusSwitch.java
D:\Tugas PB0\Tugas 3>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
ahmad
Yang anda ketik adalah a

```

- Penjelasan :

1. Deklarasi Kelas:

- Kelas utama dalam program ini diberi nama KasusSwitch.

2. Metode Utama:

- Metode utama dalam Java adalah main(String[] args).

3. Deklarasi Variabel:

- char cc;; Mendeklarasikan variabel cc dengan tipe data char untuk menyimpan huruf yang dimasukkan oleh pengguna.

4. Inisialisasi Scanner:

- `Scanner masukan = new Scanner(System.in);`: Membuat objek Scanner untuk membaca input dari pengguna melalui keyboard.

#### 5. Pesan Awal:

- `System.out.print ("Ketikkan sebuah huruf, akhiri dengan RETURN \n");`: Menampilkan pesan kepada pengguna untuk memasukkan sebuah huruf dan mengakhiri dengan menekan tombol "RETURN".

#### 6. Meminta Input dari Pengguna:

- `cc = masukan.next().charAt(0);`: Membaca huruf yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `cc`.

#### 7. Struktur Kontrol switch-case:

- `switch (cc) { ... }`: Menjalankan serangkaian pernyataan berdasarkan nilai `cc`.
- `case 'a': { ... }`: Jika nilai `cc` adalah 'a', maka blok kode di dalam kurung kurawal pertama akan dieksekusi. Program akan menampilkan pesan "Yang anda ketik adalah a".
- `case 'u': { ... }`: Jika nilai `cc` adalah 'u', maka blok kode di dalam kurung kurawal kedua akan dieksekusi. Program akan menampilkan pesan "Yang anda ketik adalah u".
- Kasus-kasus lainnya seperti 'e', 'i', 'o' akan menghasilkan pesan yang sesuai.
- `default: ...`: Jika nilai `cc` tidak cocok dengan kasus-kasus di atas, maka blok kode dalam default akan dieksekusi. Program akan menampilkan pesan "Yang anda ketik adalah huruf mati".

#### Tujuan Program:

- Tujuan dari program ini adalah untuk memberikan contoh penggunaan struktur kontrol switch-case untuk menangani beberapa kasus berbeda berdasarkan nilai yang dimasukkan oleh pengguna. Program ini memberikan pemahaman tentang bagaimana menggunakan switch-case untuk membuat pilihan berdasarkan nilai variabel tertentu.

#### 8. Kode Program :

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

```
/* Membaca jari-jari, menghitung luas lingkaran */
```

```
/* latihan pemakaian konstanta */
```

```

public class Konstant {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* Kamus */
        final float PHI = 3.1415f;
        float r;
        Scanner masukan=new Scanner(System.in);
        /* program */ /* baca data */
        System.out.print ("Jari-jari lingkaran =");
        r = masukan.nextFloat();

        /* Hitung dan tulis hasil */
        System.out.print ("Luas lingkaran = "+ (PHI * r * r)+"\n");
        System.out.print ("Akhir program \n");
    }
}
- Output :

```

```

D:\Tugas PBO\Tugas 3>javac Konstant.java

D:\Tugas PBO\Tugas 3>java Konstant
Jari-jari lingkaran =21
Luas lingkaran = 1385.4014
Akhir program

```

- Penjelasan :

1. Deklarasi Kelas:

- Kelas utama dalam program ini diberi nama Konstant.

2. Metode Utama:

- Metode utama dalam Java adalah main(String[] args).

3. Deklarasi Variabel Konstanta:

- final float PHI = 3.1415f; Mendeklarasikan konstanta PHI dengan tipe data float dan nilai 3.1415. Kata kunci final digunakan untuk membuat konstanta yang tidak dapat diubah setelah nilai awalnya ditetapkan.

4. Deklarasi Variabel:

- `float r`;: Mendeklarasikan variabel `r` dengan tipe data `float` untuk menyimpan nilai jari-jari lingkaran.

#### 5. Inisialisasi Scanner:

- `Scanner masukan = new Scanner(System.in)`;: Membuat objek Scanner untuk membaca input dari pengguna melalui keyboard.

#### 6. Meminta Input dari Pengguna:

- `System.out.print("Jari-jari lingkaran =")`;: Menampilkan pesan kepada pengguna untuk memasukkan nilai jari-jari lingkaran.
- `r = masukan.nextFloat()`;: Membaca nilai jari-jari yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `r`.

#### 7. Menghitung dan Menampilkan Luas Lingkaran:

- `System.out.print("Luas lingkaran = " + (PHI * r * r) + "\n")`;: Menghitung luas lingkaran dengan rumus  $\pi * r^2$  dan menampilkannya ke layar. Nilai  $\pi$  (pi) diambil dari konstanta PHI.

#### 8. Pesan Akhir:

- `System.out.print("Akhir program \n")`;: Menampilkan pesan akhir program.

#### Tujuan Program:

- Tujuan dari program ini adalah untuk memberikan contoh penggunaan konstanta dalam pemrograman Java serta untuk menghitung luas lingkaran berdasarkan nilai jari-jari yang dimasukkan oleh pengguna. Dengan menggunakan konstanta PHI, program ini mempermudah penggunaan nilai  $\pi$  dalam perhitungan luas lingkaran.

#### 9. Kode Program :

```
import java.util.Scanner;

/* Maksimum dua bilangan yang dibaca */

public class Max2 {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* Kamus */
        int a, b;
```



```

Scanner masukan=new Scanner(System.in);
/* Program */
System.out.print ("Maksimum dua bilangan : \n");
System.out.print ("Ketikkan dua bilangan, pisahkan dg RETURN
: \n");
a=masukan.nextInt();
b=masukan.nextInt();
System.out.println ("Ke dua bilangan : a = "+ a+" b = "+b);
    if (a >= b){
        System.out.println ("Nilai a yang maksimum "+ a);
    }else /* a > b */{
        System.out.println ("Nilai b yang maksimum: "+
b);
    }
}
}

```

- Output :

```

D:\Tugas PB0\Tugas 3>javac Max2.java
D:\Tugas PB0\Tugas 3>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
23 34
Ke dua bilangan : a = 23 b = 34
Nilai b yang maksimum: 34

```

- Penjelasan :

1. Deklarasi Kelas:

- Kelas utama dalam program ini diberi nama Max2.

2. Metode Utama:

- Metode utama dalam Java adalah main(String[] args).

3. Deklarasi Variabel:

- int a, b;; Mendeklarasikan dua variabel bertipe integer a dan b yang akan digunakan untuk menyimpan dua bilangan yang dimasukkan oleh pengguna.

4. Inisialisasi Scanner:

- `Scanner masukan = new Scanner(System.in);`: Membuat objek Scanner untuk membaca input dari pengguna melalui keyboard.

#### 5. Meminta Input dari Pengguna:

- `System.out.print("Maksimum dua bilangan : \n");`: Menampilkan pesan kepada pengguna untuk mengetahui tujuan dari program.
- `System.out.print("Ketikkan dua bilangan, pisahkan dg RETURN : \n");`: Menampilkan pesan kepada pengguna untuk memasukkan dua bilangan, dipisahkan oleh tombol "RETURN".
- `a = masukan.nextInt();`: Membaca bilangan pertama yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `a`.
- `b = masukan.nextInt();`: Membaca bilangan kedua yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `b`.

#### 6. Menentukan Bilangan Terbesar:

- `if (a >= b) { ... }`: Jika nilai `a` lebih besar atau sama dengan nilai `b`, maka blok kode di dalam kurung kurawal pertama akan dieksekusi. Program akan menampilkan pesan "Nilai `a` yang maksimum" berserta nilai `a`.
- `else { ... }`: Jika nilai `a` lebih kecil dari nilai `b`, maka blok kode di dalam kurung kurawal kedua akan dieksekusi. Program akan menampilkan pesan "Nilai `b` yang maksimum" berserta nilai `b`.

#### Tujuan Program:

- Tujuan dari program ini adalah untuk membantu pengguna menentukan bilangan terbesar dari dua bilangan yang dimasukkan. Program ini bisa digunakan sebagai contoh sederhana untuk memahami bagaimana menggunakan struktur kontrol `if-else` dalam Java untuk membuat Keputusan.

#### 10. Kode Program :

```
import java.util.Scanner;

/* Baca N, Print 1 s/d N dengan FOR */

public class PriFor {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* Kamus */
```

```

int i,N;
Scanner masukan=new Scanner(System.in);
/* Program */
/* program */
System.out.print ("Baca N, print 1 s/d N ");
System.out.print ("N = ");
N=masukan.nextInt();
for (i = 1; i <= N; i++){
System.out.println (i); }
System.out.println ("Akhir program \n");
}
}

```

- Output :

```

D:\Tugas PBO\Tugas 3>javac PriFor.java
D:\Tugas PBO\Tugas 3>java PriFor
Baca N, print 1 s/d N N = 13
1
2
3
4
5
6
7
8
9
10
11
12
13
Akhir program

```

- Penjelasan :

1. Deklarasi Kelas:

- Kelas utama dalam program ini diberi nama PriFor.

2. Metode Utama:

- Metode utama dalam Java adalah main(String[] args).

3. Deklarasi Variabel:

- `int i, N;` Mendeklarasikan dua variabel bertipe integer `i` dan `N` yang akan digunakan untuk iterasi dan menyimpan nilai `N` yang dimasukkan oleh pengguna.

4. Inisialisasi Scanner:

- `Scanner masukan = new Scanner(System.in);`: Membuat objek Scanner untuk membaca input dari pengguna melalui keyboard.

#### 5. Meminta Input dari Pengguna:

- `System.out.print("Baca N, print 1 s/d N ");`: Menampilkan pesan kepada pengguna tentang tujuan program.
- `System.out.print("N = ");`: Menampilkan pesan kepada pengguna untuk memasukkan nilai N.
- `N = masukan.nextInt();`: Membaca nilai N yang dimasukkan oleh pengguna dan menyimpannya dalam variabel N.

#### 6. Loop For:

- `for (i = 1; i <= N; i++) { ... }`: Loop for digunakan untuk mencetak angka dari 1 hingga N. Variabel i diinisialisasi dengan nilai 1, kemudian akan terus ditambah satu setiap iterasi sampai nilai i mencapai atau melebihi nilai N. Setiap iterasi, program akan mencetak nilai i ke layar menggunakan `System.out.println(i);`.

#### 7. Pesan Akhir:

- `System.out.println("Akhir program \n");`: Menampilkan pesan akhir program setelah selesai mencetak angka dari 1 hingga N.

#### Tujuan Program:

- Tujuan dari program ini adalah untuk memberikan contoh penggunaan loop for dalam Java untuk mencetak angka dari 1 hingga N, dimana N adalah nilai yang dimasukkan oleh pengguna. Program ini membantu pengguna untuk memahami konsep dasar penggunaan loop for dalam pengulangan dan penggunaan variabel iterasi untuk mengontrol jumlah iterasi dalam loop.

#### 11. Kode Program :

```
import java.util.Scanner;

/* Baca N, */

/* Print i = 1 s/d N dengan ITERATE */

public class PrintIterasi {

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        /* Kamus : */

        int N;
```

```

int i;
Scanner masukan=new Scanner(System.in);

/* Program */
System.out.print ("Nilai N >0 = "); /* Inisialisasi*/
N = masukan.nextInt();
i = 1; /* First Elmt */
    System.out.print ("Print i dengan ITERATE : \n");
for (;){
    System.out.println(i); /* Proses */
    if (i == N)
        /* Kondisi Berhenti */ break;
    else {
        i++; /* Next Elmt */
    }
    } /* (i == N) */
}
}

```

- Output :

```

D:\Tugas PBO\Tugas 3>javac PrintIterasi.java

D:\Tugas PBO\Tugas 3>java PrintIterasi
Nilai N >0 = 8
Print i dengan ITERATE :
1
2
3
4
5
6
7
8

```

- Penjelasan :

1. Deklarasi Kelas:

- Kelas utama dalam program ini diberi nama PrintIterasi.

2. Metode Utama:

- Metode utama dalam Java adalah main(String[] args).

3. Deklarasi Variabel:

- `int N;` Mendeklarasikan variabel N yang akan digunakan untuk menyimpan nilai N yang dimasukkan oleh pengguna.
- `int i;` Mendeklarasikan variabel i yang akan digunakan sebagai indeks dalam iterasi.

#### 4. Inisialisasi Scanner:

- `Scanner masukan = new Scanner(System.in);` Membuat objek Scanner untuk membaca input dari pengguna melalui keyboard.

#### 5. Meminta Input dari Pengguna:

- `System.out.print("Nilai N >0 = ");` Menampilkan pesan kepada pengguna untuk memasukkan nilai N yang diinginkan.
- `N = masukan.nextInt();` Membaca nilai N yang dimasukkan oleh pengguna dan menyimpannya dalam variabel N.

#### 6. Inisialisasi Variabel Iterasi:

- `i = 1;` Menginisialisasi variabel iterasi i dengan nilai awal 1.

#### 7. Loop Tanpa Batas dengan Instruksi Break:

- `for (;;) { ... }` Loop tanpa batas yang akan terus berjalan hingga nilai i mencapai nilai N.
- `System.out.println(i);` Mencetak nilai i ke layar.
- `if (i == N) break;` Instruksi untuk keluar dari loop jika nilai i sudah mencapai nilai N.
- `else { i++; }` Jika nilai i belum mencapai N, nilai i akan ditambah satu untuk iterasi selanjutnya.

#### Tujuan Program:

- Tujuan dari program ini adalah untuk memberikan contoh penggunaan loop tanpa batas dengan instruksi break di dalamnya untuk menghentikan iterasi ketika kondisi tertentu terpenuhi.

#### 12. Kode Program :

```
import java.util.Scanner;

public class PrintRepeat {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* Kamus : */
    }
}
```

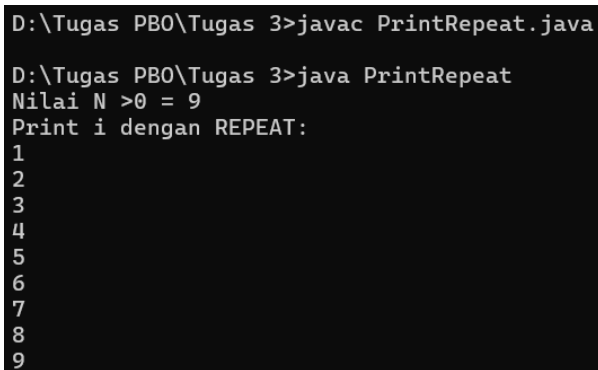
```

int N;
int i;
Scanner masukan=new Scanner(System.in);
/* Program */
System.out.print ("Nilai N >0 = ");

/* Inisialisasi */
N = masukan.nextInt();
i = 1;
/* First Elmt */
System.out.print ("Print i dengan REPEAT: \n");
do{
    System.out.print (i+"\n"); /* Proses */
    i++; /* Next Elmt */
}
while (i <= N); /* Kondisi pengulangan */
}
}

```

- Output :



```

D:\Tugas PBO\Tugas 3>javac PrintRepeat.java
D:\Tugas PBO\Tugas 3>java PrintRepeat
Nilai N >0 = 9
Print i dengan REPEAT:
1
2
3
4
5
6
7
8
9

```

- Penjelasan :

1. Deklarasi Kelas:

- Kelas utama dalam program ini diberi nama PrintRepeat.

2. Metode Utama:

- Metode utama dalam Java adalah main(String[] args).

3. Deklarasi Variabel:

- `int N`;: Mendeklarasikan variabel `N` yang akan digunakan untuk menyimpan nilai `N` yang dimasukkan oleh pengguna.
- `int i`;: Mendeklarasikan variabel `i` yang akan digunakan sebagai indeks dalam iterasi.

#### 4. Inisialisasi Scanner:

- `Scanner masukan = new Scanner(System.in)`;: Membuat objek `Scanner` untuk membaca input dari pengguna melalui keyboard.

#### 5. Meminta Input dari Pengguna:

- `System.out.print("Nilai N >0 = ")`;: Menampilkan pesan kepada pengguna untuk memasukkan nilai `N` yang diinginkan.
- `N = masukan.nextInt()`;: Membaca nilai `N` yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `N`.

#### 6. Inisialisasi Variabel Iterasi:

- `i = 1`;: Menginisialisasi variabel iterasi `i` dengan nilai awal 1.

#### 7. Loop do-while:

- `do { ... } while (i <= N)`;: Loop do-while digunakan untuk mencetak nilai `i` dari 1 hingga `N`. Pertama-tama, program akan mencetak nilai `i`, kemudian nilai `i` akan ditambah satu. Iterasi akan terus berlanjut selama nilai `i` kurang dari atau sama dengan `N`.

#### Tujuan Program:

- Tujuan dari program ini adalah untuk memberikan contoh penggunaan loop do-while dalam Java untuk melakukan pengulangan dengan kondisi di bagian bawah. Program ini membantu pengguna untuk memahami konsep dasar penggunaan loop do-while dan bagaimana cara menggunakan loop ini untuk melakukan pengulangan dengan kondisi di akhir iterasi.

#### 13. Kode Program :

```
import java.util.Scanner;

/* Baca N, Print i = 1 s/d N dengan WHILE */
public class PrintWhile {
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        /* Kamus : */
```



```

int N;
int i;
Scanner masukan=new Scanner(System.in);

/* Program */
System.out.print ("Nilai N >0 = "); /* Inisialisasi */
N = masukan.nextInt();
i = 1; /* First Elmt */
System.out.print ("Print i dengan WHILE: \n");
while (i <= N) /* Kondisi pengulangan */
{ System.out.println (i); /* Proses */
  i++; /* Next Elmt */
}; /* (i > N) */
}
}
- Output :

```

```

D:\Tugas PBO\Tugas 3>javac PrintWhile.java
D:\Tugas PBO\Tugas 3>java PrintWhile
Nilai N >0 = 6
Print i dengan WHILE:
1
2
3
4
5
6

```

- Penjelasan :
  1. Deklarasi Kelas:
    - Kelas utama dalam program ini diberi nama PrintWhile.
  2. Metode Utama:
    - Metode utama dalam Java adalah main(String[] args).
  3. Deklarasi Variabel:
    - int N;; Mendeklarasikan variabel N yang akan digunakan untuk menyimpan nilai N yang dimasukkan oleh pengguna.
    - int i;; Mendeklarasikan variabel i yang akan digunakan sebagai indeks dalam iterasi.

#### 4. Inisialisasi Scanner:

- `Scanner masukan = new Scanner(System.in);`: Membuat objek Scanner untuk membaca input dari pengguna melalui keyboard.

#### 5. Meminta Input dari Pengguna:

- `System.out.print("Nilai N >0 = ");`: Menampilkan pesan kepada pengguna untuk memasukkan nilai N yang diinginkan.
- `N = masukan.nextInt();`: Membaca nilai N yang dimasukkan oleh pengguna dan menyimpannya dalam variabel N.

#### 6. Inisialisasi Variabel Iterasi:

- `i = 1;`: Menginisialisasi variabel iterasi i dengan nilai awal 1.

#### 7. Loop while:

- `while (i <= N) { ... }`: Loop while digunakan untuk mencetak nilai i dari 1 hingga N. Pertama-tama, program akan memeriksa apakah nilai i kurang dari atau sama dengan N. Jika benar, maka blok kode di dalam kurung kurawal akan dieksekusi, yaitu mencetak nilai i dan kemudian menambahkan nilai i. Proses ini akan terus berlanjut selama nilai i kurang dari atau sama dengan N.

#### Tujuan Program:

- Tujuan dari program ini adalah untuk memberikan contoh penggunaan loop while dalam Java untuk melakukan pengulangan dengan kondisi di bagian atas. Program ini membantu pengguna untuk memahami konsep dasar penggunaan loop while dan bagaimana cara menggunakan loop ini untuk melakukan pengulangan dengan kondisi di awal iterasi.

#### 14. Kode Program :

```
import java.util.Scanner;

/* Baca N, */

/* Print i = 1 s/d N dengan while (ringkas) */

public class PrintWhile1 {

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        /* Kamus : */

        int N;

        int i = 1;
```

```

Scanner masukan=new Scanner(System.in);

/* Program */
System.out.print ("Nilai N >0 = ");
N = masukan.nextInt();
System.out.print ("Print i dengan WHILE (ringkas): \n");
while (i <= N){
System.out.println (i++);
} /* (i > N)*/
}
}

```

- Output :

```

D:\Tugas PBO\Tugas 3>javac PrintWhile1.java

D:\Tugas PBO\Tugas 3>java PrintWhile1
Nilai N >0 = 10
Print i dengan WHILE (ringkas):
1
2
3
4
5
6
7
8
9
10

```

- Penjelasan :

1. Deklarasi Kelas:

- Kelas utama dalam program ini diberi nama PrintWhile1.

2. Metode Utama:

- Metode utama dalam Java adalah main(String[] args).

3. Deklarasi Variabel:

- int N;: Mendeklarasikan variabel N yang akan digunakan untuk menyimpan nilai N yang dimasukkan oleh pengguna.
- int i = 1;: Mendeklarasikan variabel i dengan nilai awal 1. Ini adalah inisialisasi langsung untuk variabel i.

#### 4. Inisialisasi Scanner:

- `Scanner masukan = new Scanner(System.in);`: Membuat objek Scanner untuk membaca input dari pengguna melalui keyboard.

#### 5. Meminta Input dari Pengguna:

- `System.out.print("Nilai N >0 = ");`: Menampilkan pesan kepada pengguna untuk memasukkan nilai N yang diinginkan.
- `N = masukan.nextInt();`: Membaca nilai N yang dimasukkan oleh pengguna dan menyimpannya dalam variabel N.

#### 6. Loop while:

- `while (i <= N) { System.out.println(i++); }`: Loop while digunakan untuk mencetak nilai i dari 1 hingga N. Pada setiap iterasi, program akan mencetak nilai i dan kemudian menambahkan nilai i. Proses ini akan terus berlanjut selama nilai i kurang dari atau sama dengan N.

#### Tujuan Program:

- Tujuan dari program ini adalah untuk memberikan contoh penggunaan loop while dalam Java untuk melakukan pengulangan dengan kondisi di bagian atas. Program ini memberikan solusi yang lebih ringkas dengan menginisialisasi variabel iterasi i secara langsung pada saat deklarasinya dan menambahkan nilai variabel i dalam satu baris pada saat pencetakan nilai.

#### 15. Kode Program :

```
import java.util.Scanner;

/* contoh baca nilai x, */
/* Jumlahkan nilai yang dibaca dengan ITERATE */
public class PrintXinterasi {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* Kamus : */
        int Sum=0;
        int x;
        Scanner masukan=new Scanner(System.in);
```

```

        /* Program */
        System.out.print ("Masukkan nilai x (int), akhiri dg
999 : ");
        x = masukan.nextInt(); /* First Elmt */
        if (x == 999){
            System.out.print ("Kasus kosong \n");
        }else{ /* Minimal ada satu data yang dijumlahkan */
            Sum = x; /* Inisialisasi; invariant !! */
            for (;;) {
                System.out.print ("Masukkan nilai x (int), akhiri dg
999 : ");
                x = masukan.nextInt(); /* Next Elmt */
                if(x==999)
                    break;
                else{
                    Sum = Sum + x; /* Proses */
                }
            }
            System.out.println("Hasil penjumlahan = "+ Sum);
            /* Terminasi */
        }
    }
}

```

- Output :

```

D:\Tugas PBO\Tugas 3>javac PrintXinterasi.java

D:\Tugas PBO\Tugas 3>java PrintXinterasi
Masukkan nilai x (int), akhiri dg 999 : 55
Masukkan nilai x (int), akhiri dg 999 : 45
Masukkan nilai x (int), akhiri dg 999 : 50
Masukkan nilai x (int), akhiri dg 999 : 20
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 170

```

- Penjelasan :

1. Deklarasi Kelas:

- Kelas utama dalam program ini diberi nama PrintXinterasi.

## 2. Metode Utama:

- Metode utama dalam Java adalah `main(String[] args)`.

## 3. Deklarasi Variabel:

- `int Sum = 0;` Mendeklarasikan variabel `Sum` yang akan digunakan untuk menyimpan jumlah dari nilai-nilai `x`.
- `int x;` Mendeklarasikan variabel `x` yang akan digunakan untuk menyimpan nilai `x` yang dimasukkan oleh pengguna.
- `Scanner masukan = new Scanner(System.in);` Membuat objek `Scanner` untuk membaca input dari pengguna melalui keyboard.

## 4. Meminta Input dari Pengguna dan Menghitung Jumlah:

- Program akan menampilkan pesan "Masukkan nilai x (int), akhiri dg 999 : " untuk meminta pengguna memasukkan nilai `x`.
- Program akan membaca nilai `x` yang dimasukkan oleh pengguna menggunakan `masukan.nextInt()`.
- Jika nilai `x` yang dimasukkan sama dengan 999, program akan menampilkan pesan "Kasus kosong" karena tidak ada nilai yang dimasukkan.
- Jika nilai `x` yang dimasukkan tidak sama dengan 999, program akan menjumlahkan nilai `x` dengan `Sum` menggunakan operasi `Sum = Sum + x`.
- Proses ini akan terus berulang hingga pengguna memasukkan nilai 999 untuk mengakhiri proses penginputan.

## 5. Menampilkan Hasil Penjumlahan:

- Setelah pengguna memasukkan nilai 999 dan proses penginputan selesai, program akan menampilkan pesan "Hasil penjumlahan = " diikuti dengan nilai dari variabel `Sum`, yang merupakan hasil penjumlahan dari semua nilai `x` yang dimasukkan oleh pengguna.

## Tujuan Program:

- Tujuan dari program ini adalah untuk memberikan contoh penggunaan loop `for` tanpa kondisi terminasi yang spesifik, dengan menggunakan instruksi `break` untuk menghentikan proses pengulangan.

## 16. Kode Program :

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

```

/* contoh baca nilai x, */
/* Jumlahkan nilai yang dibaca dengan ITERATE */

public class PrintXRepeat {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* Kamus : */
        int Sum;
        int x;
        Scanner masukan=new Scanner(System.in);

        /* Program */
        System.out.print ("Masukkan nilai x (int), akhiri dg
999 : ");
        x = masukan.nextInt(); /* First Elmt */
        if (x == 999){
            System.out.print("Kasus kosong \n");
        }else { /* MInimal ada satu data yang dijumlahkan */
            Sum = 0; /* Inisialisasi; invariant !! */
            do{
                Sum = Sum + x; /* Proses */
                System.out.print ("Masukkan nilai x (int), akhiri dg
999 : ");
                x = masukan.nextInt(); /* Next Elmt */
            } while (x != 999); /* Kondisi pengulangan */
            System.out.println ("Hasil penjumlahan = "+Sum);
            /* Terminasi */
        }
    }
}

```

- Output :

```
D:\Tugas PBO\Tugas 3>javac PrintXRepeat.java
D:\Tugas PBO\Tugas 3>java PrintXRepeat
Masukkan nilai x (int), akhiri dg 999 : 60
Masukkan nilai x (int), akhiri dg 999 : 40
Masukkan nilai x (int), akhiri dg 999 : 100
Masukkan nilai x (int), akhiri dg 999 : 30
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 230
```

- Penjelasan :

1. Deklarasi Kelas:

- Kelas utama dalam program ini diberi nama PrintXRepeat.

2. Metode Utama:

- Metode utama dalam Java adalah main(String[] args).

3. Deklarasi Variabel:

- int Sum;; Mendeklarasikan variabel Sum yang akan digunakan untuk menyimpan jumlah dari nilai-nilai x.
- int x;; Mendeklarasikan variabel x yang akan digunakan untuk menyimpan nilai x yang dimasukkan oleh pengguna.
- Scanner masukan = new Scanner(System.in);: Membuat objek Scanner untuk membaca input dari pengguna melalui keyboard.

4. Meminta Input dari Pengguna dan Menghitung Jumlah:

- Program akan menampilkan pesan "Masukkan nilai x (int), akhiri dg 999 : " untuk meminta pengguna memasukkan nilai x.
- Program akan membaca nilai x yang dimasukkan oleh pengguna menggunakan masukan.nextInt().
- Jika nilai x yang dimasukkan sama dengan 999, program akan menampilkan pesan "Kasus kosong" karena tidak ada nilai yang dimasukkan.
- Jika nilai x yang dimasukkan tidak sama dengan 999, program akan menjumlahkan nilai x dengan Sum menggunakan operasi Sum = Sum + x.
- Proses ini akan terus berulang hingga pengguna memasukkan nilai 999 untuk mengakhiri proses penginputan, yang ditentukan oleh kondisi while (x != 999).

5. Menampilkan Hasil Penjumlahan:

- Setelah pengguna memasukkan nilai 999 dan proses penginputan selesai, program akan menampilkan pesan "Hasil penjumlahan = " diikuti dengan nilai dari



variabel Sum, yang merupakan hasil penjumlahan dari semua nilai x yang dimasukkan oleh pengguna.

Tujuan Program:

- Tujuan dari program ini adalah untuk memberikan contoh penggunaan loop do-while dalam Java untuk melakukan pengulangan dengan kondisi di bagian bawah. Program ini memberikan solusi yang lebih sederhana dan ringkas untuk menangani kasus di mana ada minimal satu data yang dijumlahkan.

17.Kode Program :

```
import java.util.Scanner;

/* contoh baca nilai x, */
/* Jumlahkan nilai yang dibaca dengan WHILE */

public class PrintXWhile {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* Kamus : */
        int Sum;
        int x;
        Scanner masukan=new Scanner(System.in);

        /* Program */
        Sum = 0; /* Inisialisasi */
        System.out.print ("Masukkan nilai x (int), akhiri dg
999 : ");
        x = masukan.nextInt();/* First Elmt*/
        while (x != 999) /* Kondisi berhenti */ {
            Sum = Sum + x; /* Proses */
            System.out.print("Masukkan nilai x (int), akhiri dg 999
: ");
            x = masukan.nextInt();/* First Elmt*/
        }
        System.out.println("Hasil penjumlahan = "+ Sum); /*
```

```

        Terminasi */
    }
}
- Output :

```

```

D:\Tugas PBO\Tugas 3>javac PrintXWhile.java
D:\Tugas PBO\Tugas 3>java PrintXWhile
Masukkan nilai x (int), akhiri dg 999 : 400
Masukkan nilai x (int), akhiri dg 999 : 34
Masukkan nilai x (int), akhiri dg 999 : 66
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 500

```

- Penjelasan :

#### 1. Deklarasi Kelas:

- Kelas utama dalam program ini diberi nama PrintXWhile.

#### 2. Metode Utama:

- Metode utama dalam Java adalah main(String[] args).

#### 3. Deklarasi Variabel:

- int Sum;; Mendeklarasikan variabel Sum yang akan digunakan untuk menyimpan jumlah dari nilai-nilai x.
- int x;; Mendeklarasikan variabel x yang akan digunakan untuk menyimpan nilai x yang dimasukkan oleh pengguna.
- Scanner masukan = new Scanner(System.in); Membuat objek Scanner untuk membaca input dari pengguna melalui keyboard.

#### 4. Inisialisasi Variabel:

- Sum = 0;; Menginisialisasi variabel Sum dengan nilai 0 sebelum dimulainya proses pengulangan.

#### 5. Loop while:

- while (x != 999) { ... }: Loop while digunakan untuk menjalankan proses pengulangan selama nilai x yang dimasukkan oleh pengguna tidak sama dengan 999.
- Di dalam loop, program akan menambahkan nilai x ke variabel Sum menggunakan operasi Sum = Sum + x, dan kemudian meminta pengguna memasukkan nilai x baru dengan menampilkan pesan "Masukkan nilai x (int), akhiri dg 999 : ".
- Proses ini akan terus berulang hingga pengguna memasukkan nilai 999 untuk mengakhiri proses penginputan.

#### 6. Menampilkan Hasil Penjumlahan:

- Setelah pengguna memasukkan nilai 999 dan proses penginputan selesai, program akan menampilkan pesan "Hasil penjumlahan = " diikuti dengan nilai dari variabel Sum, yang merupakan hasil penjumlahan dari semua nilai x yang dimasukkan oleh pengguna.

#### Tujuan Program:

- Tujuan dari program ini adalah untuk memberikan contoh penggunaan loop while dalam Java untuk melakukan pengulangan dengan kondisi di bagian atas. Program ini membantu pengguna untuk memahami konsep dasar penggunaan loop while dan bagaimana cara menggunakan loop ini untuk melakukan pengulangan dengan kondisi di awal iterasi.

#### 18. Kode Program :

```
import java.util.Scanner;

/* Contoh program yang mengandung prosedur dan fungsi */
/* prosedur/fungsi */

public class SubProgram {
    public static int maxab (int a, int b){
        /* mencari maksimum dua bilangan bulat */
        return ((a >= b) ? a : b);
    }

    public static void tukar (int a, int b)
    { /* menukar dua bilangan bulat */
        int temp;
        temp = a;
        a = b;
        b = temp;

        System.out.println ("Ke dua bilangan setelah tukar: a =
"+ a +" b = "+ b);
    }

    /*** Program Utama ***/
    public static void main(String[] args) {
        // TODO Auto-generated method stub
```

```

        /* Membaca dua bilangan integer */
        /* Menuliskan maksimum dua bilangan yang dibaca dg
memanggil fungsi */
        /* Menukar kedua bilangan dengan 'prosedur' */
        int a, b;
        Scanner masukan=new Scanner(System.in);
        System.out.print ("Maksimum dua bilangan \n");
        System.out.print ("Ketikkan dua bilangan, pisahkan dg
RETURN : \n");
        a = masukan.nextInt();
        b = masukan.nextInt();
        System.out.println ("Ke dua bilangan : a = "+ a +" b =
"+ b);
        System.out.println ("Maksimum = " + (maxab(a, b)));
        System.out.print("Tukar kedua bilangan... \n");
        tukar (a, b);
    }
}

```

- Output :

```

D:\Tugas PBO\Tugas 3>javac SubProgram.java

D:\Tugas PBO\Tugas 3>java SubProgram
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
55 45
Ke dua bilangan : a = 55 b = 45
Maksimum = 55
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 45 b = 55

```

- Penjelasan :

1. Deklarasi Kelas:

- Kelas utama dalam program ini diberi nama SubProgram.

2. Deklarasi Fungsi:

- `public static int maxab(int a, int b):` Mendefinisikan fungsi maxab yang mengambil dua parameter a dan b, dan mengembalikan nilai maksimum dari kedua parameter tersebut.
- Fungsi ini menggunakan operator ternary (?:) untuk membandingkan nilai a dan b, kemudian mengembalikan nilai a jika a lebih besar atau

sama dengan b, dan mengembalikan nilai b jika sebaliknya.

### 3. Deklarasi Prosedur:

- `public static void tukar(int a, int b)`: Mendefinisikan prosedur tukar yang mengambil dua parameter a dan b yang merepresentasikan dua bilangan bulat.
  - Proses ini bertujuan untuk menukar nilai kedua parameter. Namun, perlu diingat bahwa dalam Java, parameter fungsi merupakan nilai yang disalin, sehingga perubahan nilai parameter dalam prosedur tidak akan mempengaruhi nilai variabel asli di luar prosedur.
  - Prosedur ini mencetak nilai kedua parameter setelah pertukaran dilakukan.

### 4. Metode Utama:

- Metode utama dalam Java adalah `main(String[] args)`.

### 5. Membaca Input dari Pengguna:

- Program akan menampilkan pesan untuk memasukkan dua bilangan bulat dan kemudian membaca input dari pengguna menggunakan objek `Scanner`.
- Input dari pengguna disimpan dalam variabel a dan b.

### 6. Memanggil Fungsi dan Prosedur:

- Program kemudian memanggil fungsi `maxab(a, b)` untuk mencari nilai maksimum dari kedua bilangan yang dimasukkan.
- Selanjutnya, program memanggil prosedur `tukar(a, b)` untuk menukar nilai dari kedua bilangan.

### Tujuan Program:

- Tujuan dari program ini adalah untuk memberikan contoh penggunaan prosedur dan fungsi dalam bahasa pemrograman Java. Program ini membantu pengguna untuk memahami perbedaan antara prosedur dan fungsi, serta cara menggunakan keduanya dalam program Java.

### 19. Kode Program :

```
import java.util.Scanner;

/* contoh pemakaian IF tiga kasus : wujud air */

public class Tempair {
    public static void main(String[] args) {
```

```
// TODO Auto-generated method stub
/* Kamus : */
int T;
Scanner masukan=new Scanner(System.in);

/* Program */
System.out.print ("Contoh IF tiga kasus \n");
System.out.print ("Temperatur (der. C) = ");
T=masukan.nextInt();
if (T < 0) {
    System.out.print ("Wujud air beku \n"+ T);
}else if ((0 <= T) && (T <= 100)){
    System.out.print ("Wujud air cair \n"+ T);
}else if (T > 100){
    System.out.print ("Wujud air uap/gas \n"+ T);
};
}
}
```

- Output :

```
D:\Tugas PBO\Tugas 3>javac Tempair.java
D:\Tugas PBO\Tugas 3>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 18
Wujud air cair
18
```

- Penjelasan :

1. Deklarasi Kelas:

- Kelas utama dalam program ini diberi nama Tempair.

2. Metode Utama:

- Metode utama dalam Java adalah main(String[] args).

3. Deklarasi Variabel:

- int T;; Mendeklarasikan variabel T yang akan digunakan untuk menyimpan suhu air yang dimasukkan oleh pengguna.

- `Scanner masukan = new Scanner(System.in);` : Membuat objek Scanner untuk membaca input dari pengguna melalui keyboard.

#### 4. Meminta Input dari Pengguna:

- Program akan menampilkan pesan "Temperatur (der. C) = " untuk meminta pengguna memasukkan suhu air.
- Program kemudian membaca input suhu dari pengguna menggunakan `masukan.nextInt()` dan menyimpannya dalam variabel `T`.

#### 5. Penggunaan Struktur Pemilihan if-else if-else:

- Program menggunakan struktur pemilihan if-else if-else untuk menentukan wujud air berdasarkan suhu yang dimasukkan pengguna.
- Jika suhu (`T`) kurang dari 0, program akan mencetak "Wujud air beku" berserta nilai suhu.
- Jika suhu (`T`) di antara 0 dan 100, program akan mencetak "Wujud air cair" berserta nilai suhu.
- Jika suhu (`T`) lebih dari 100, program akan mencetak "Wujud air uap/gas" berserta nilai suhu.

#### Tujuan Program:

- Tujuan dari program ini adalah untuk memberikan contoh penggunaan struktur pemilihan if-else if-else dalam Java untuk menentukan kondisi berdasarkan rentang nilai suatu variabel. Program ini membantu pengguna untuk memahami cara menggunakan struktur pemilihan untuk mengambil keputusan berdasarkan kondisi tertentu.