

## **PEMROGRAMAN BERORIENTASI OBJEK**



Nama : Andika Pratama

Nim 13020220009

Kelas : A1

Dosen : Mardiyah Hasnawati, S.Kom., M.T.

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

        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);

    }
}
```

### 1. Import Statement:

```
import java.io.BufferedReader;  
  
import java.io.IOException;  
  
import java.io.InputStreamReader;  
  
import javax.swing.*;
```

#### PENJELASAN KODE :

Kode diawali dengan beberapa pernyataan impor. Ini digunakan untuk mengimpor kelas-kelas yang diperlukan untuk operasi input-output dan penggunaan antarmuka grafis (Swing).

### 2. Deklarasi Kelas:

```
public class BacaString {
```

#### PENJELASAN KODE :

Ini adalah deklarasi kelas utama yang disebut "BacaString".

### 3. Metode Utama:

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

#### PENJELASAN KODE :

Ini adalah metode utama yang akan dieksekusi saat program dimulai. Metode ini melemparkan IOException, yang berarti itu dapat melemparkan kesalahan saat terjadi masalah dengan operasi input/output.

### 4. Variabel Lokal:

```
String str; BufferedReader dataIn = new BufferedReader(new InputStreamReader(System.in));
```

#### PENJELASAN KODE :

Sebuah variabel String str dideklarasikan untuk menyimpan string yang akan dibaca dari input pengguna. Objek dataIn dari kelas BufferedReader digunakan untuk membaca input dari pengguna menggunakan InputStreamReader yang membaca dari System.in (input standar).

## 5. Mengambil Input:

```
System.out.print ("\nBaca string dan Integer: \n"); System.out.print("masukkan sebuah string: ");  
str= datAIn.readLine();
```

### PENJELASAN KODE :

Pesan-pesan ini mencetak petunjuk untuk pengguna dan meminta pengguna untuk memasukkan sebuah string. Kemudian, string yang dimasukkan oleh pengguna dibaca menggunakan metode `readLine()` dari objek `BufferedReader` dan disimpan dalam variabel `str`.

## 6. Menampilkan Output:

```
System.out.print ("String yang dibaca : "+ str);
```

### PENJELASAN KODE :

Setelah membaca string dari pengguna, program mencetak kembali string tersebut ke konsol.

## 7. Penutup Kelas:

```
}
```

### PENJELASAN KODE :

Penutup dari metode `main` dan kelas `BacaString`.

## .2 Kode Program

```
public class ForEver {  
  
    /**  
     * @param args  
     */  
    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");  
        }  
    }  
}
```

PENJELASAN KODE :

1. Deklarasi Kelas:

```
public class ForEver {
```

Ini adalah deklarasi kelas utama yang disebut "ForEver".

2. Metode Utama:

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

Ini adalah metode utama yang akan dieksekusi saat program dimulai.

3. Pesan Print:

```
System.out.println("Program akan looping, akhiri dengan ^c");
```

Pesan ini mencetak petunjuk kepada pengguna bahwa program akan terus melakukan loop tanpa henti, dan dapat dihentikan dengan menggunakan kombinasi tombol "Ctrl + C".

4. Loop While:

```
while (true) { System.out.print ("Print satu baris \n"); }
```

Ini adalah loop while yang tidak pernah berakhir karena kondisinya selalu benar (true). Di dalam loop, program mencetak satu baris teks "Print satu baris" ke konsol. Karena tidak ada pernyataan atau kondisi yang dapat mengubah nilai kebenaran dari kondisi loop (true), maka program akan terus mencetak baris teks ini tanpa henti.

5. Penutup Metode dan Kelas:

```
}
```

Penutup dari metode main dan kelas ForEver.

### 3. Kode Program

```
/* contoh pemakaian IF satu kasus */
/* membaca nilai integer, menuliskan nilainya jika positif */

import java.util.Scanner;

public class If1 {

    /**
     * @param args
     */
    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);

    }
}
```

#### PENJELASAN KODE :

##### 1. Import Pernyataan:

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

Ini adalah pernyataan impor yang diperlukan untuk menggunakan kelas Scanner, yang digunakan untuk mendapatkan input dari pengguna.

##### 2. Deklarasi Kelas:

```
public class If1 {
```

Ini adalah deklarasi kelas utama yang disebut "If1".

##### 3. Metode Utama:

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

Ini adalah metode utama yang akan dieksekusi saat program dimulai.

#### 4. Variabel Lokal:

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

Sebuah objek masukan dari kelas Scanner dideklarasikan untuk mendapatkan input dari pengguna. Selain itu, variabel a dideklarasikan untuk menyimpan nilai integer yang dimasukkan oleh pengguna.

#### 5. Menampilkan Pesan:

```
System.out.print ("Contoh IF satu kasus \n"); System.out.print ("Ketikkan suatu nilai integer : ");
```

Pesan-pesan ini mencetak petunjuk kepada pengguna untuk memasukkan suatu nilai integer.

#### 6. Menerima Input:

```
a = masukan.nextInt();
```

Program menggunakan objek masukan untuk memanggil metode nextInt() yang akan membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya dalam variabel a.

#### 7. Struktur Kondisional IF:

```
if (a >= 0) System.out.print ("\nNilai a positif "+ a);
```

Ini adalah struktur kondisional IF yang mengecek apakah nilai variabel a lebih besar dari atau sama dengan nol. Jika kondisi ini terpenuhi (nilai a positif), maka program akan mencetak pesan yang menyatakan bahwa nilai a adalah positif bersama dengan nilainya.

#### 8. Penutup Metode dan Kelas:

```
}
```

Penutup dari metode main dan kelas If1.

#### 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 {

    /**
     * @param args
     */
    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);

        }

    }

}
```

#### PENJELASAN KODE :

Kode di atas adalah contoh penggunaan struktur kondisional IF dengan dua kasus komplementer dalam bahasa pemrograman Java. Program ini membaca sebuah nilai integer dari pengguna dan menampilkan pesan yang sesuai tergantung pada apakah nilai tersebut positif atau negatif. Berikut adalah penjelasan lebih rinci:

##### 1. Import Pernyataan:

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



Ini adalah pernyataan impor yang diperlukan untuk menggunakan kelas Scanner, yang digunakan untuk mendapatkan input dari pengguna.

## 2. Deklarasi Kelas:

```
public class If2 {
```

Ini adalah deklarasi kelas utama yang disebut "If2".

## 3. Metode Utama:

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

Ini adalah metode utama yang akan dieksekusi saat program dimulai.

## 4. Variabel Lokal:

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

Sebuah variabel a dideklarasikan untuk menyimpan nilai integer yang dimasukkan oleh pengguna. Objek masukan dari kelas Scanner dideklarasikan untuk mendapatkan input dari pengguna.

## 5. Menampilkan Pesan:

```
System.out.print ("Contoh IF dua kasus \n"); System.out.print ("Ketikkan suatu nilai integer :");
```

Pesan-pesan ini mencetak petunjuk kepada pengguna untuk memasukkan suatu nilai integer.

## 6. Menerima Input:

```
a = masukan.nextInt();
```

Program menggunakan objek masukan untuk memanggil metode nextInt() yang akan membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya dalam variabel a.

#### 7. Struktur Kondisional IF dengan Dua Kasus:

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

Ini adalah struktur kondisional IF dengan dua kasus. Jika nilai a lebih besar dari atau sama dengan nol, maka pesan "Nilai a positif" akan dicetak bersama dengan nilai a. Jika tidak, maka pesan "Nilai a negatif" akan dicetak bersama dengan nilai a.

#### 8. Penutup Metode dan Kelas:

```
}
```

Penutup dari metode main dan kelas If2.

### 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 {  
  
    /**  
     * @param args  
     */  
    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);
    }
}
}
}

```

#### PENJELASAN KODE :

Kode di atas adalah contoh program sederhana dalam bahasa pemrograman Java yang menggunakan struktur kontrol percabangan if-else-if untuk menentukan tiga kasus berdasarkan nilai yang dimasukkan pengguna. Berikut penjelasan maksud dari kode tersebut:

1. `import java.util.Scanner;;` Mendeklarasikan bahwa program akan menggunakan kelas Scanner dari paket java.util untuk membaca input dari pengguna.
2. `public class If3 {:` Mendefinisikan kelas utama program yang disebut If3.
3. `public static void main(String[] args) {:` Mendefinisikan metode utama program yang akan dieksekusi pertama kali saat program dijalankan.
4. `int a;;` Mendeklarasikan variabel a sebagai integer yang akan digunakan untuk menyimpan nilai yang dimasukkan pengguna.
5. `Scanner masukan=new Scanner(System.in);:` Membuat objek Scanner bernama masukan untuk membaca input dari pengguna melalui konsol.
6. `System.out.print ("Contoh IF tiga kasus \n");:` Menampilkan pesan "Contoh IF tiga kasus" di konsol.
7. `System.out.print ("Ketikkan suatu nilai integer :");:` Menampilkan pesan "Ketikkan suatu nilai integer :" di konsol untuk meminta pengguna memasukkan nilai.
8. `a=masukan.nextInt();:` Membaca input yang dimasukkan pengguna dan menyimpannya ke dalam variabel a.

9. `if (a > 0){`: Memulai blok percabangan `if`, jika nilai `a` lebih besar dari 0, maka blok kode dalam `if` akan dieksekusi.
10. `System.out.println ("Nilai a positif "+ a);`: Menampilkan pesan "Nilai a positif" diikuti dengan nilai `a` jika nilai `a` lebih besar dari 0.
11. `else if (a == 0){`: Jika nilai `a` tidak lebih besar dari 0, maka dilakukan pengecekan apakah nilai `a` sama dengan 0.
12. `System.out.println ("Nilai Nol "+ a);`: Jika nilai `a` sama dengan 0, maka pesan "Nilai Nol" diikuti dengan nilai `a` akan ditampilkan.
13. `else /* a > 0 */ {`: Jika nilai `a` tidak lebih besar dari 0 dan tidak sama dengan 0, maka blok kode ini akan dieksekusi.
14. `System.out.println ("Nilai a negatif "+ a);`: Menampilkan pesan "Nilai a negatif" diikuti dengan nilai `a` jika nilai `a` kurang dari 0.
15. `}`: Penutup dari blok percabangan `if-else`.
16. `}`: Penutup dari metode `main`.
17. `}`: Penutup dari kelas `If3`.

## 6. Kode Program

```

/* Eksrpesi kondisional dengan boolean */
public class KasusBoolean {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /* Kamus */
        boolean bool;
        /*
        Algori
        tma */
        bool=
        true;
        if(boo

```

```

        1) {
System.out.print("true\n");

        } else
            System.out.print("false\n");

        if(!bool) {
            System.out.print("salah\n");
        } else
            System.out.print("benar\n");

    }

}

```

#### PENJELASAN KODE :

Kode di atas adalah contoh program sederhana dalam bahasa pemrograman Java yang menggunakan ekspresi kondisional dengan tipe data boolean. Berikut penjelasan singkat dari kode tersebut:

1. `public class KasusBoolean {`: Mendefinisikan kelas utama program yang disebut KasusBoolean.
2. `public static void main(String[] args) {`: Mendefinisikan metode utama program yang akan dieksekusi pertama kali saat program dijalankan.
3. `boolean bool;;`: Mendeklarasikan variabel `bool` sebagai boolean.
4. `bool= true;;`: Memberikan nilai `true` kepada variabel `bool`.
5. `if(bool) {`: Memulai blok percabangan `if`, jika nilai `bool` adalah `true`, maka blok kode dalam `if` akan dieksekusi.
6. `System.out.print("true\n");`: Jika nilai `bool` adalah `true`, maka akan menampilkan "true" di konsol.
7. `} else:` Bagian `else` dari percabangan `if-else`, akan dieksekusi jika nilai `bool` bukan `true`
8. `System.out.print("false\n");`: Jika nilai `bool` bukan `true`, maka akan menampilkan "false" di konsol.

9. `if(!bool) {`: Memulai blok percabangan if, tetapi dalam ekspresi kondisional ini menggunakan operator "!" yang artinya "not". Jadi, jika nilai bool adalah false, maka blok kode dalam if akan dieksekusi.
10. `System.out.print("salah\n");`: Jika nilai bool adalah false, maka akan menampilkan "salah" di konsol.
11. `} else:` Bagian else dari percabangan if-else, akan dieksekusi jika nilai bool adalah true.
12. `System.out.print("benar\n");`: Jika nilai bool adalah true, maka akan menampilkan "benar" di konsol.
13. `}`: Penutup dari metode main.
14. `}`: Penutup dari kelas KasusBoolean.

## 7. Kode Program

```
import java.util.Scanner;

public class KasusSwitch {

    /**
     * @param args
     */
    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");
    }
}
}
}

```

#### PENJELASAN KODE :

Kode di atas adalah contoh program dalam bahasa pemrograman Java yang menggunakan struktur kontrol switch-case untuk menentukan aksi yang akan diambil berdasarkan nilai yang dimasukkan pengguna. Berikut adalah penjelasan singkat dari kode tersebut:

1. `import java.util.Scanner;`: Mendeklarasikan bahwa program akan menggunakan kelas Scanner dari paket `java.util` untuk membaca input dari pengguna.
2. `public class KasusSwitch {`: Mendefinisikan kelas utama program yang disebut KasusSwitch.
3. `public static void main(String[] args) {`: Mendefinisikan metode utama program yang akan dieksekusi pertama kali saat program dijalankan.
4. `char cc;`: Mendeklarasikan variabel `cc` sebagai karakter yang akan digunakan untuk menyimpan nilai yang dimasukkan pengguna.
5. `Scanner masukan=new Scanner(System.in);`: Membuat objek Scanner bernama `masukan` untuk membaca input dari pengguna melalui konsol.
6. `System.out.print ("Ketikkan sebuah huruf, akhiri dengan RETURN \n");`: Menampilkan pesan "Ketikkan sebuah huruf, akhiri dengan RETURN" di konsol untuk meminta pengguna memasukkan sebuah huruf.
7. `cc=masukan.next().charAt(0);`: Membaca input yang dimasukkan pengguna dan menyimpan karakter pertama dari input tersebut ke dalam variabel `cc`.

8. `switch (cc) {`: Memulai struktur kontrol switch-case dengan ekspresi yang dievaluasi adalah nilai variabel `cc`.
9. `case 'a': { System.out.print (" Yang anda ketik adalah a\n"); break;}`: Jika nilai `cc` adalah 'a', maka blok kode dalam case ini akan dieksekusi, yaitu menampilkan pesan "Yang anda ketik adalah a". Kemudian, `break` digunakan untuk keluar dari switch-case.
10. `case 'u': { System.out.print (" Yang anda ketik adalah u\n"); break;}`: Jika nilai `cc` adalah 'u', maka blok kode dalam case ini akan dieksekusi, yaitu menampilkan pesan "Yang anda ketik adalah u". Kemudian, `break` digunakan untuk keluar dari switch-case.
11. `case 'e': { System.out.print (" Yang anda ketik adalah e\n"); break;}`: Kasus yang serupa dengan penjelasan sebelumnya, namun kali ini untuk huruf 'e'.
12. `case 'i': { System.out.print (" Yang anda ketik adalah i\n"); break;}`: Kasus yang serupa dengan penjelasan sebelumnya, namun kali ini untuk huruf 'i'.
13. `case 'o': { System.out.print (" Yang anda ketik adalah o\n"); break;}`: Kasus yang serupa dengan penjelasan sebelumnya, namun kali ini untuk huruf 'o'.
14. `default: System.out.print (" Yang anda ketik adalah huruf mati\n");`: Bagian ini akan dieksekusi jika tidak ada case yang sesuai dengan nilai `cc` yang dimasukkan pengguna. Pesan "Yang anda ketik adalah huruf mati" akan ditampilkan di konsol.
15. `}`: Penutup dari struktur kontrol switch-case.
16. `}`: Penutup dari metode `main`.
17. `}`: Penutup dari kelas `KasusSwitch`.



#### 8. Kode Program

```
import java.util.Scanner;

/* Membaca jari-jari, menghitung luas lingkaran */
/* latihan pemakaian konstanta */
public class Konstant {

    /**
    * @param args

    */
    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");

    }
}
```

#### PENJELASAN KODE :

Kode di atas adalah contoh program dalam bahasa pemrograman Java yang digunakan untuk menghitung luas lingkaran berdasarkan jari-jarinya. Berikut penjelasan singkat dari kode tersebut:

1. `import java.util.Scanner;;` Mendeklarasikan bahwa program akan menggunakan kelas Scanner dari paket java.util untuk membaca input dari pengguna.
2. `public class Konstant {:` Mendefinisikan kelas utama program yang disebut Konstant.
3. `public static void main(String[] args) {:` Mendefinisikan metode utama program yang akan dieksekusi pertama kali saat program dijalankan.

4. `final float PHI = 3.1415f;` Mendeclare konstanta PHI dengan nilai 3.1415. Keyword `final` digunakan agar nilai konstanta tidak dapat diubah setelah diberikan nilai pertama kali.
5. `float r;` Mendeclare variabel `r` sebagai bilangan floating point yang akan digunakan untuk menyimpan jari-jari lingkaran.
6. `Scanner masukan=new Scanner(System.in);` Membuat objek `Scanner` bernama `masukan` untuk membaca input dari pengguna melalui konsol.
7. Program mencetak pesan ke layar untuk meminta pengguna memasukkan jari-jari lingkaran.
8. `r = masukan.nextFloat();` Membaca input yang dimasukkan pengguna dan menyimpannya dalam variabel `r`.
9. `System.out.print ("Luas lingkaran = "+ (PHI * r * r)+"\n");` Menghitung luas lingkaran dengan rumus  $\pi * r^2$ , kemudian menampilkan hasilnya di layar. Variabel `PHI` digunakan untuk merepresentasikan nilai  $\pi$  yang merupakan konstanta.
10. `System.out.print ("Akhir program \n");` Menampilkan pesan "Akhir program" di layar.
11. `};` Penutup dari metode `main`.
12. `};` Penutup dari kelas `Konstant`.

## 9. Kode Program

```
import java.util.Scanner;

/* Maksimum dua bilangan yang dibaca */
public class Max2 {

    /**
     * @param args
     */
    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);
        }
    }
}
```

### PENJELASAN KODE :

1. Kode di atas adalah sebuah program sederhana dalam bahasa pemrograman Java yang bertujuan untuk menentukan bilangan maksimum dari dua bilangan yang dimasukkan pengguna. Berikut adalah penjelasan singkat dari kode tersebut:
2. `import java.util.Scanner;;` Mendeklarasikan bahwa program akan menggunakan kelas Scanner dari paket java.util untuk membaca input dari pengguna.
3. `public class Max2 {`: Mendefinisikan kelas utama program yang disebut Max2.

4. `public static void main(String[] args) {`: Mendefinisikan metode utama program yang akan dieksekusi pertama kali saat program dijalankan.
5. Mendeklarasikan dua variabel integer, a dan b, yang akan digunakan untuk menyimpan dua bilangan yang dimasukkan oleh pengguna.
6. `Scanner masukan=new Scanner(System.in);`: Membuat objek Scanner bernama masukan untuk membaca input dari pengguna melalui konsol.
7. Program mencetak pesan "Maksimum dua bilangan" dan "Ketikkan dua bilangan, pisahkan dg RETURN" di layar untuk memberi petunjuk kepada pengguna.
8. `a=masukan.nextInt(); b=masukan.nextInt();`: Membaca dua bilangan yang dimasukkan oleh pengguna dan menyimpannya dalam variabel a dan b.
9. `System.out.println ("Ke dua bilangan : a = "+ a+" b = "+b);`: Menampilkan kembali kedua bilangan yang dimasukkan pengguna.
10. `if (a >= b){`: Memulai blok percabangan if, jika nilai a lebih besar atau sama dengan nilai b, maka blok kode dalam if akan dieksekusi.
11. `System.out.println ("Nilai a yang maksimum "+ a);`: Jika nilai a lebih besar atau sama dengan b, maka program akan menampilkan pesan "Nilai a yang maksimum" diikuti dengan nilai a.
12. `else /* a > b */{`: Bagian else dari percabangan if-else, akan dieksekusi jika nilai a tidak lebih besar atau sama dengan nilai b.
13. `System.out.println ("Nilai b yang maksimum: "+ b);`: Jika nilai a tidak lebih besar atau sama dengan b, maka program akan menampilkan pesan "Nilai b yang maksimum" diikuti dengan nilai b.
14. `}`: Penutup dari blok percabangan if-else.
15. `}`: Penutup dari metode main.
16. `}`: Penutup dari kelas Max2.

#### 10. Kode Program

```
import java.util.Scanner;

/* Baca N, Print 1 s/d N dengan FOR */
public class PriFor {

    /**
     * @param
     */
    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");
    }
}
```

#### PENJELASAN KODE :

Kode di atas adalah contoh program dalam bahasa pemrograman Java yang bertujuan untuk membaca sebuah bilangan bulat N dari pengguna dan kemudian menampilkan angka dari 1 hingga N menggunakan perulangan for. Berikut adalah penjelasan singkat dari kode tersebut:

1. `import java.util.Scanner;`: Mendeklarasikan bahwa program akan menggunakan kelas Scanner dari paket java.util untuk membaca input dari pengguna.
2. `public class PriFor {`: Mendefinisikan kelas utama program yang disebut PriFor.
3. `public static void main(String[] args) {`: Mendefinisikan metode utama program yang akan dieksekusi pertama kali saat program dijalankan.

4. Mendeklarasikan dua variabel integer, yaitu i dan N, yang akan digunakan dalam program.
5. `Scanner masukan=new Scanner(System.in);`: Membuat objek Scanner bernama masukan untuk membaca input dari pengguna melalui konsol.
6. Program mencetak pesan "Baca N, print 1 s/d N" di layar, memberikan petunjuk kepada pengguna.
7. `System.out.print ("N = "); N=masukan.nextInt();`: Program meminta pengguna memasukkan sebuah bilangan bulat N dan menyimpannya dalam variabel N.
8. `for (i = 1; i <= N; i++){`: Memulai perulangan for yang akan berjalan dari 1 hingga N. Setiap iterasi, nilai variabel i akan bertambah satu.
9. `System.out.println (i); }`: Pada setiap iterasi, program akan mencetak nilai variabel i di layar, sehingga mencetak angka dari 1 hingga N secara berurutan.
10. `System.out.println ("Akhir program \n");`: Setelah perulangan selesai, program akan menampilkan pesan "Akhir program" di layar.
11. `};`: Penutup dari metode main.
12. `};`: Penutup dari kelas PriFor.

## 11. KODE PROGRAM

```
import java.util.Scanner;
/* Baca N, */
/* Print i = 1 s/d N dengan ITERATE */

public class Program11 {

    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) */
}
}

```

#### PENJELASAN KODE

Kode di atas adalah contoh program dalam bahasa pemrograman Java yang bertujuan untuk membaca sebuah bilangan bulat N dari pengguna, kemudian mencetak angka dari 1 hingga N menggunakan perulangan iterate (iteratif). Berikut adalah penjelasan singkat dari kode tersebut:

1. `import java.util.Scanner;;` Mendeklarasikan bahwa program akan menggunakan kelas Scanner dari paket java.util untuk membaca input dari pengguna.
2. `public class Program11 {:` Mendefinisikan kelas utama program yang disebut Program11.
3. `public static void main(String[] args) {:` Mendefinisikan metode utama program yang akan dieksekusi pertama kali saat program dijalankan.
4. Mendeklarasikan dua variabel integer, yaitu N dan i, yang akan digunakan dalam program.
5. `Scanner masukan=new Scanner(System.in);:` Membuat objek Scanner bernama masukan untuk membaca input dari pengguna melalui konsol.
6. Program mencetak pesan "Nilai N >0 = " di layar, memberikan petunjuk kepada pengguna untuk memasukkan nilai N.

7. `N = masukan.nextInt();` Program membaca sebuah bilangan bulat N dari pengguna dan menyimpannya dalam variabel N.
8. `i = 1;` Inisialisasi nilai awal i menjadi 1. Ini akan digunakan sebagai angka pertama yang akan dicetak.
9. `System.out.print ("Print i dengan ITERATE : \n");` Menampilkan pesan "Print i dengan ITERATE :" di layar untuk memberikan informasi kepada pengguna.
10. `for (;){` Memulai perulangan for tanpa kondisi awal, kondisi perulangan, maupun pernyataan iterasi. Ini menandakan bahwa perulangan akan terus berjalan sampai ada pernyataan break yang menghentikan perulangan.
11. `System.out.println(i);` Mencetak nilai variabel i di layar pada setiap iterasi. Pada awalnya, nilai i adalah 1 dan akan terus bertambah satu pada setiap iterasi.
12. `if (i == N) break;` Program memeriksa apakah nilai i sudah sama dengan N. Jika i sudah sama dengan N, maka perulangan akan dihentikan dengan menggunakan pernyataan break.
13. `else { i++; }` Jika nilai i belum sama dengan N, maka nilai i akan ditambah satu untuk memproses iterasi selanjutnya.
14. `};` Penutup dari blok perulangan for.
15. `};` Penutup dari metode main.
16. `};` Penutup dari kelas Program11.



## 12. Kode Program

```
import java.util.Scanner;

/**
 * @param args
 */
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 */
}

}
```

### PENJELASAN KODE :

Kode program di atas adalah sebuah program dalam bahasa pemrograman Java yang bertujuan untuk membaca sebuah bilangan bulat N dari pengguna, kemudian mencetak angka dari 1 hingga N menggunakan perulangan dengan kata kunci do-while. Berikut penjelasan singkat dari kode tersebut:

1. `import java.util.Scanner;;` Mendeklarasikan bahwa program akan menggunakan kelas Scanner dari paket java.util untuk membaca input dari pengguna.
2. `public static void main(String[] args) {`: Mendefinisikan metode utama program yang akan dieksekusi pertama kali saat program dijalankan.
3. Mendeklarasikan dua variabel integer, yaitu N dan i, yang akan digunakan dalam program.

4. `Scanner masukan=new Scanner(System.in);`; Membuat objek Scanner bernama masukan untuk membaca input dari pengguna melalui konsol.
5. Program mencetak pesan "Nilai N >0 = " di layar, memberikan petunjuk kepada pengguna untuk memasukkan nilai N.
6. `N = masukan.nextInt();`; Program membaca sebuah bilangan bulat N dari pengguna dan menyimpannya dalam variabel N.
7. `i = 1;`; Inisialisasi nilai awal i menjadi 1. Ini akan digunakan sebagai angka pertama yang akan dicetak.
8. `System.out.print ("Print i dengan REPEAT: \n");`; Menampilkan pesan "Print i dengan REPEAT:" di layar untuk memberikan informasi kepada pengguna.
9. `do {`; Memulai perulangan do-while. Blok kode di dalam do akan dieksekusi setidaknya sekali, dan kemudian kondisi perulangan di cek.
10. `System.out.print (i+"\n");`; Mencetak nilai variabel i di layar pada setiap iterasi.
11. `i++;`; Menambahkan nilai variabel i dengan 1 setelah setiap iterasi. Ini digunakan untuk beralih ke nilai berikutnya.
12. `} while (i <= N);`; Memeriksa apakah nilai i masih kurang dari atau sama dengan N. Jika benar, maka perulangan akan terus berlanjut. Jika tidak, perulangan akan dihentikan.
13. `}`; Penutup dari metode main.

### 13. KODE PROGRAM

```
import java.util.Scanner;

/* Baca N, Print i = 1 s/d N dengan WHILE */
public class PrintWhile {

    /**
     * @param args
     */
    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) */

    }

}
```

#### PENJELASAN KODE :

1. import java.util.Scanner;; Mendeklarasikan bahwa program akan menggunakan kelas Scanner dari paket java.util untuk membaca input dari pengguna.
2. public class PrintWhile {}: Mendefinisikan kelas utama program yang disebut PrintWhile.
3. public static void main(String[] args) {}: Mendefinisikan metode utama program yang akan dieksekusi pertama kali saat program dijalankan.
4. Mendeklarasikan dua variabel integer, yaitu N dan i, yang akan digunakan dalam program.

5. `Scanner masukan=new Scanner(System.in);` Membuat objek Scanner bernama masukan untuk membaca input dari pengguna melalui konsol.
6. Program mencetak pesan "Nilai N >0 = " di layar, memberikan petunjuk kepada pengguna untuk memasukkan nilai N.
7. `N = masukan.nextInt();` Program membaca sebuah bilangan bulat N dari pengguna dan menyimpannya dalam variabel N.
8. `i = 1;` Inisialisasi nilai awal i menjadi 1. Ini akan digunakan sebagai angka pertama yang akan dicetak.
9. `System.out.print ("Print i dengan WHILE: \n");` Menampilkan pesan "Print i dengan WHILE:" di layar untuk memberikan informasi kepada pengguna.
10. `while (i <= N):` Memulai perulangan while. Perulangan akan terus berlangsung selama nilai i kurang dari atau sama dengan N.
11. `{` Memulai blok kode yang akan dieksekusi selama kondisi perulangan terpenuhi.
12. `System.out.println (i);` Mencetak nilai variabel i di layar pada setiap iterasi.
13. `i++;` Menambahkan nilai variabel i dengan 1 setelah setiap iterasi. Ini digunakan untuk beralih ke nilai berikutnya.
14. `};` Penutup dari blok perulangan while.
15. `}` Penutup dari metode main.

#### 14. Kode Program

```
import java.util.Scanner;

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

public class PrintWhile1 {

    /**
     * @param args
     */
    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)*/

    }

}
```

#### PENJELASAN KODE :

1. `import java.util.Scanner;`: Mendeklarasikan impor dari paket `java.util.Scanner`, yang diperlukan untuk memungkinkan penggunaan objek `Scanner` untuk membaca input dari pengguna.
2. `public class PrintWhile1 { ... }`: Mendefinisikan kelas Java utama dengan nama `PrintWhile1`.
3. `public static void main(String[] args) { ... }`: Mendefinisikan metode `main`, yang merupakan titik masuk program Java. Di dalam metode `main`, program akan mulai dieksekusi.

4. `int N;` Mendeklarasikan variabel `N` yang akan digunakan untuk menyimpan nilai yang dimasukkan oleh pengguna.
5. `int i = 1;` Mendeklarasikan dan menginisialisasi variabel `i` dengan nilai 1. Variabel `i` akan digunakan sebagai penghitung dalam perulangan `while` untuk mencetak nilai mulai dari 1 hingga  $\heartsuit N$ .
6. `Scanner masukan=new Scanner(System.in);` Membuat objek `Scanner` dengan nama `masukan`, yang digunakan untuk membaca input dari pengguna melalui keyboard.
7. `System.out.print ("Nilai N >0 = "); N = masukan.nextInt();` Mencetak pesan "Nilai N >0 = " dan kemudian menggunakan objek `Scanner` untuk membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya dalam variabel `N`.
8. `System.out.print ("Print i dengan WHILE (ringkas): \n");` Mencetak pesan "Print i dengan WHILE (ringkas):" untuk memberi tahu pengguna bahwa program akan mencetak nilai  $\heartsuit i$  menggunakan perulangan `while`.
9. `while (i <= N){ System.out.println (i++); }` Ini adalah perulangan `while` yang mencetak nilai  $\heartsuit i$  secara berurutan mulai dari 1 hingga  $\heartsuit N$ . Setiap iterasi, nilai  $\heartsuit i$  dicetak menggunakan `System.out.println`, kemudian nilai  $\heartsuit i$  ditingkatkan dengan operator `i++`. Perulangan akan berhenti ketika nilai  $\heartsuit i$  melebihi nilai  $\heartsuit N$ .
10. `};` Tutup dari blok perulangan `while`.
11. `};` Tutup dari metode `main`.
12. `};` Tutup dari kelas `PrintWhile1`.

## 15. KODE PROGRAM

```

import java.util.Scanner;

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

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        /*
         Kamu
         S :
         */
        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 */
    }
}

```

#### PENJELASAN KODE :

1. `import java.util.Scanner;;` Mendeklarasikan impor dari paket `java.util.Scanner`, yang diperlukan untuk memungkinkan penggunaan objek `Scanner` untuk membaca input dari pengguna.
2. `public class PrintXinterasi { ... }`: Mendefinisikan kelas Java utama dengan nama `PrintXinterasi`.
3. `public static void main(String[] args) { ... }`: Mendefinisikan metode `main`, yang merupakan titik masuk program Java. Di dalam metode `main`, program akan mulai dieksekusi.
4. `int Sum=0; int x;;` Mendeklarasikan dua variabel, `Sum` yang digunakan untuk menyimpan jumlah nilai-nilai yang dimasukkan oleh pengguna, dan `x` yang digunakan untuk menyimpan nilai yang dibaca dari pengguna.
5. `Scanner masukan=new Scanner(System.in);`: Membuat objek `Scanner` dengan nama `masukan`, yang digunakan untuk membaca input dari pengguna melalui keyboard.
6. `System.out.print ("Masukkan nilai x (int), akhiri dg 999 : ");`: Mencetak pesan untuk meminta pengguna memasukkan nilai `x` dan memberikan instruksi untuk mengakhiri masukan dengan angka 999.
7. `x = masukan.nextInt();`: Membaca nilai `x` yang dimasukkan oleh pengguna menggunakan objek `Scanner` dan menyimpannya dalam variabel `x`.
8. `if (x == 999){ System.out.print ("Kasus kosong \n"); }`: Mengecek apakah nilai `x` yang dimasukkan pengguna adalah 999. Jika ya, maka program akan mencetak pesan "Kasus kosong" yang menandakan bahwa tidak ada nilai yang dimasukkan.
9. `else { ... }`: Jika nilai `x` bukan 999, maka program akan masuk ke dalam blok ini.
10. `Sum = x;`: Menginisialisasi variabel `Sum` dengan nilai `x`, karena ini adalah nilai pertama yang dimasukkan oleh pengguna.



11. `for (;;) { ... };` Memulai perulangan tak terbatas. Hal ini dilakukan untuk terus meminta pengguna memasukkan nilai  $x$  hingga pengguna memasukkan nilai 999 untuk mengakhiri masukan.
12. `System.out.print ("Masukkan nilai x (int), akhiri dg 999 : ");` Mencetak pesan untuk meminta pengguna memasukkan nilai  $x$  lagi.
13. `x = masukan.nextInt();` Membaca nilai  $x$  yang dimasukkan oleh pengguna kembali menggunakan objek Scanner dan menyimpannya dalam variabel `x`.
14. `if(x==999) break;` Memeriksa apakah nilai  $x$  yang dimasukkan pengguna adalah 999. Jika ya, maka program akan keluar dari perulangan menggunakan pernyataan `break`.
15. `else { Sum = Sum + x; };` Jika nilai  $x$  bukan 999, maka nilai tersebut akan ditambahkan ke dalam variabel `Sum`.
16. `System.out.println("Hasil penjumlahan = "+ Sum);` Setelah perulangan selesai, program akan mencetak hasil penjumlahan dari semua nilai yang dimasukkan oleh pengguna.
17. `};` Tutup dari blok `else`.
18. `};` Tutup dari metode `main`.
19. `};` Tutup dari kelas `PrintXinterasi`.

## 16. KODE PROGRAM

```
import java.util.Scanner;

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

public class PrintXRepeat {

    /**
     * @param args
     */
    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 */

}

}

```

PENJELASAN KODE :

1. import java.util.Scanner;; Mendeklarasikan impor dari paket java.util.Scanner, yang diperlukan untuk memungkinkan penggunaan objek Scanner untuk membaca input dari pengguna.
2. public class PrintXRepeat { ... }; Mendefinisikan kelas Java utama dengan nama PrintXRepeat.
3. public static void main(String[] args) { ... }; Mendefinisikan metode main, yang merupakan titik masuk program Java. Di dalam metode main, program akan mulai dieksekusi.

4. `int Sum; int x;`: Mendeklarasikan dua variabel, Sum yang digunakan untuk menyimpan jumlah nilai-nilai yang dimasukkan oleh pengguna, dan x yang digunakan untuk menyimpan nilai yang dibaca dari pengguna.
5. `Scanner masukan=new Scanner(System.in);`: Membuat objek Scanner dengan nama masukan, yang digunakan untuk membaca input dari pengguna melalui keyboard.
6. `System.out.print ("Masukkan nilai x (int), akhiri dg 999 : ");`: Mencetak pesan untuk meminta pengguna memasukkan nilai  $x$  dan memberikan instruksi untuk mengakhiri masukan dengan angka 999.
7. `x = masukan.nextInt();`: Membaca nilai  $x$  yang dimasukkan oleh pengguna menggunakan objek Scanner dan menyimpannya dalam variabel x.
8. `if (x == 999){ System.out.print("Kasus kosong \n"); }`: Mengecek apakah nilai  $x$  yang dimasukkan pengguna adalah 999. Jika ya, maka program akan mencetak pesan "Kasus kosong" yang menandakan bahwa tidak ada nilai yang dimasukkan.
9. `else { ... }`: Jika nilai  $x$  bukan 999, maka program akan masuk ke dalam blok ini.
10. `Sum = 0;`: Menginisialisasi variabel Sum dengan nilai 0 sebelum melakukan iterasi.
11. `do { ... } while (x != 999);`: Ini adalah struktur perulangan do-while, yang akan melakukan iterasi setidaknya satu kali, dan akan terus melakukan iterasi selama nilai  $x$  yang dimasukkan bukanlah 999.
12. `Sum = Sum + x;`: Mengakumulasi nilai  $x$  ke dalam variabel Sum.
13. `System.out.print ("Masukkan nilai x (int), akhiri dg 999 : ");`: Mencetak pesan untuk meminta pengguna memasukkan nilai  $x$  lagi.
14. `x = masukan.nextInt();`: Membaca nilai  $x$  yang dimasukkan oleh pengguna kembali menggunakan objek Scanner dan menyimpannya dalam variabel x.
15. `System.out.println ("Hasil penjumlahan = "+Sum);`: Setelah perulangan selesai, program akan mencetak hasil penjumlahan dari semua nilai yang dimasukkan oleh pengguna.
16. `}`: Tutup dari blok else.
17. `}`: Tutup dari metode main.
18. `}`: Tutup dari kelas PrintXRepeat.

## 17. KODE PROGRAM

```
import java.util.Scanner;

/* contoh baca nilai x, */
/* Jumlahkan nilai yang dibaca dengan WHILE */
public class PrintXWhile {

    /**
     * @param args
     */
    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 */
}
}

```

#### PENJELASAN KODE :

1. import java.util.Scanner;; Mendeklarasikan impor dari paket java.util.Scanner, yang diperlukan untuk memungkinkan penggunaan objek Scanner untuk membaca input dari pengguna.
2. public class PrintXWhile { ... }; Mendefinisikan kelas Java utama dengan nama PrintXWhile.
3. public static void main(String[] args) { ... }; Mendefinisikan metode main, yang merupakan titik masuk program Java. Di dalam metode main, program akan mulai dieksekusi.
4. int Sum; int x;; Mendeklarasikan dua variabel, Sum yang digunakan untuk menyimpan jumlah nilai-nilai yang dimasukkan oleh pengguna, dan x yang digunakan untuk menyimpan nilai yang dibaca dari pengguna.
5. Scanner masukan=new Scanner(System.in); Membuat objek Scanner dengan nama masukan, yang digunakan untuk membaca input dari pengguna melalui keyboard.
6. Sum = 0;; Menginisialisasi variabel Sum dengan nilai 0 sebelum melakukan iterasi.
7. System.out.print ("Masukkan nilai x (int), akhiri dg 999 : "); Mencetak pesan untuk meminta pengguna memasukkan nilai  $x$  dan memberikan instruksi untuk mengakhiri masukan dengan angka 999.

8. `x = masukan.nextInt();`: Membaca nilai  $x$  yang dimasukkan oleh pengguna menggunakan objek Scanner dan menyimpannya dalam variabel `x`.
9. `while (x != 999) { ... }`: Ini adalah struktur perulangan `while`, yang akan terus melakukan iterasi selama nilai  $x$  yang dimasukkan bukanlah 999.
10. `Sum = Sum + x;`: Mengakumulasi nilai  $x$  ke dalam variabel `Sum`.
11. `System.out.print("Masukkan nilai x (int), akhiri dg 999 : ");`: Mencetak pesan untuk meminta pengguna memasukkan nilai  $x$  lagi.
12. `x = masukan.nextInt();`: Membaca nilai  $x$  yang dimasukkan oleh pengguna kembali menggunakan objek Scanner dan menyimpannya dalam variabel `x`.
13. `System.out.println("Hasil penjumlahan = "+ Sum);`: Setelah perulangan selesai, program akan mencetak hasil penjumlahan dari semua nilai yang dimasukkan oleh pengguna.
14. `}`: Tutup dari blok perulangan `while`.
15. `}`: Tutup dari metode `main`.
16. `}`: Tutup dari kelas `PrintXWhile`.

## 18. KODE PROGRAM

```
import java.util.Scanner;

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

public class SubProgram {

    /**
     * @param args
     */
    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 */
        i
        n
        t
        t
        e
        m
        p
        ;
        t
        e
        m
        p
    }
```

```

        =
        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 dibacadg 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.nextI
nt();b =
        masukan.nextI
nt();
        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);

    }

```

#### PENJELASAN KODE :

1. `import java.util.Scanner;;` Mendeklarasikan impor dari paket `java.util.Scanner`, yang diperlukan untuk memungkinkan penggunaan objek `Scanner` untuk membaca input dari pengguna.
2. `public class SubProgram { ... }` Mendefinisikan kelas Java utama dengan nama `SubProgram`.
3. `public static int maxab(int a, int b) { ... }` Mendefinisikan fungsi `maxab` yang mengambil dua parameter integer `a` dan `b` dan mengembalikan nilai maksimum dari dua bilangan tersebut.
  - `return ((a >= b) ? a : b);` Fungsi ini mengembalikan nilai `a` jika `a` lebih besar dari atau sama dengan `b`, dan mengembalikan nilai `b` jika `b` lebih besar dari `a`.
4. `public static void tukar(int a, int b) { ... }` Mendefinisikan prosedur `tukar` yang mengambil dua parameter integer `a` dan `b` dan bertujuan untuk menukar nilai kedua variabel tersebut.
5. `int temp; temp = a; a = b; b = temp;` Proses menukar nilai dilakukan dengan menggunakan variabel sementara `temp`. Nilai `a` disimpan di `temp`, nilai `b` disalin ke `a`, dan nilai `temp` (nilai semula `a`) disalin ke `b`.
6. `System.out.println("Ke dua bilangan setelah tukar: a = " + a + " b = " + b);` Menggunakan `System.out.println` untuk mencetak nilai `a` dan `b` setelah proses pertukaran.
7. `public static void main(String[] args) { ... }` Mendefinisikan metode `main`, yang merupakan titik masuk program Java.
8. `int a, b;` Deklarasi dua variabel integer `a` dan `b`.
9. `Scanner masukan = new Scanner(System.in);` Membuat objek `Scanner` untuk membaca input dari pengguna.
10. `System.out.print("Maksimum dua bilangan\n");` Mencetak pesan untuk memberi tahu pengguna bahwa program akan menemukan nilai maksimum dari dua bilangan yang dimasukkan.
11. `System.out.print("Ketikkan dua bilangan, pisahkan dg RETURN : \n");` Meminta pengguna untuk memasukkan dua bilangan, dipisahkan oleh tombol RETURN.
12. `a = masukan.nextInt(); b = masukan.nextInt();` Membaca dua bilangan yang dimasukkan oleh pengguna menggunakan objek `Scanner` dan menyimpannya dalam variabel `a` dan `b`.
13. `System.out.println("Ke dua bilangan : a = " + a + " b = " + b);` Mencetak nilai dari `a` dan `b` yang telah dimasukkan oleh pengguna.
14. `System.out.println("Maksimum = " + (maxab(a, b)));` Memanggil fungsi `maxab` untuk menemukan nilai maksimum dari `a` dan `b`, dan mencetak hasilnya.
15. `System.out.print("Tukar kedua bilangan...\n");` Mencetak pesan untuk memberi tahu pengguna bahwa nilai kedua bilangan akan ditukar.
16. `tukar(a, b);` Memanggil prosedur `tukar` untuk menukar nilai dari `a` dan `b`.
17. `}` Tutup dari metode `main`.
18. `}` Tutup dari kelas `SubProgram`.

## 19. KODE PROGRAM

```
import java.util.Scanner;

/* contoh pemakaian IF tiga kasus : wujud air */
public class Tempair {

    /**
     * @param args
     */
    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);
        };
    }
}
```

### PENJELASAN KODE :

1. `import java.util.Scanner;`: Mendeklarasikan impor dari paket `java.util.Scanner`, yang diperlukan untuk memungkinkan penggunaan objek `Scanner` untuk membaca input dari pengguna.
2. `public class Tempair { ... }`: Mendefinisikan kelas Java utama dengan nama `Tempair`.
3. `public static void main(String[] args) { ... }`: Mendefinisikan metode `main`, yang merupakan titik masuk program Java. Di dalam metode `main`, program akan mulai dieksekusi.
4. `int T;`: Mendeklarasikan variabel `T` yang akan digunakan untuk menyimpan suhu air.



5. `Scanner masukan = new Scanner(System.in);`:: Membuat objek Scanner dengan nama masukan, yang digunakan untuk membaca input dari pengguna melalui keyboard.
6. `System.out.print ("Contoh IF tiga kasus \n");`:: Mencetak pesan untuk memberitahu pengguna bahwa program akan menunjukkan contoh penggunaan struktur kontrol if dengan tiga kasus.
7. `System.out.print ("Temperatur (der. C) = "); T = masukan.nextInt();`:: Mencetak pesan untuk meminta pengguna memasukkan suhu air dalam derajat Celsius, kemudian membaca nilai yang dimasukkan oleh pengguna dan menyimpannya dalam variabel T.
8. `if (T < 0) { ... }`: Ini adalah blok if pertama, yang akan dieksekusi jika nilai T kurang dari 0.
9. `System.out.print ("Wujud air beku \n" + T);`:: Jika suhu T kurang dari 0, maka pesan "Wujud air beku" akan dicetak, bersama dengan nilai T.
10. `else if ((0 <= T) && (T <= 100)){ ... }`: Ini adalah blok else if kedua, yang akan dieksekusi jika blok if sebelumnya tidak terpenuhi dan nilai T berada dalam rentang antara 0 hingga 100.
11. `System.out.print ("Wujud air cair \n" + T);`:: Jika suhu T berada di antara 0 hingga 100, maka pesan "Wujud air cair" akan dicetak, bersama dengan nilai T.
12. `else if (T > 100){ ... }`: Ini adalah blok else if ketiga, yang akan dieksekusi jika blok if dan else if sebelumnya tidak terpenuhi dan nilai T lebih dari 100.
13. `System.out.print ("Wujud air uap/gas \n" + T);`:: Jika suhu T lebih dari 100, maka pesan "Wujud air uap/gas" akan dicetak, bersama dengan nilai T.
14. `}`: Tutup dari blok if-else if-else.
15. `}`: Tutup dari metode main.
16. `}`: Tutup dari kelas Tempair.