

LAPORAN
PEMROGRAMAN BERORIENTASI OBJEK



Nama : Nifal
Stambuk : 13020220039
Kelas : A1
Dosen : Mardiyah Hasnawi, S.Kom., M.T.,MTA

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
        /* 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);

    }
}
```

- **Penjelasan program**

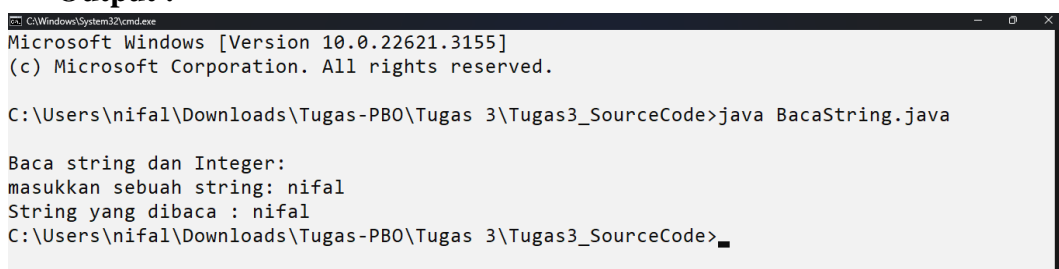
Kode program untuk membaca input string dari pengguna menggunakan **BufferedReader** dan mencetaknya ke layar.

Berikut adalah penjelasan langkah demi langkah dari kode tersebut:

1. **Import Paket:** Baris-baris pertama program mengimpor kelas-kelas yang diperlukan dari paket **java.io** dan **javax.swing**. Ini termasuk **BufferedReader** untuk membaca input dari pengguna dan **JOptionPane** untuk menampilkan dialog pesan (meskipun dalam kode ini belum digunakan).
2. **Deklarasi Kelas:** Program mengandung kelas tunggal bernama **BacaString**.
3. **Metode `main`:** Ini adalah titik masuk program Java. Di dalamnya, instruksi-instruksi yang akan dieksekusi pada saat program dijalankan ditempatkan.
4. **Deklarasi Variabel:** Variabel lokal **str** bertipe String dideklarasikan untuk menyimpan input string dari pengguna.
5. **Membuat Objek `BufferedReader`:** Baris ini menciptakan objek **BufferedReader** yang disebut **datAIn**, yang digunakan untuk membaca input dari **System.in** (input standar, biasanya dari keyboard).
6. **Meminta Input:** Program mencetak pesan "masukkan sebuah string: " ke konsol, memberi tahu pengguna untuk memasukkan string.
7. **Membaca Input:** Program menggunakan metode **readLine()** dari objek **BufferedReader** untuk membaca satu baris input dari pengguna dan menyimpannya dalam variabel **str**.
8. **Menampilkan Output:** Program mencetak pesan yang berisi string yang dibaca dari input pengguna ke konsol.
9. **Penanganan Error:** Bagian **throws IOException** dalam deklarasi metode **main** menunjukkan bahwa program dapat melempar pengecualian **IOException** jika ada kesalahan saat membaca input.
10. **Penutup:** Kurung kurawal menandai akhir dari blok kode dalam metode **main** dan kelas **BacaString**.

Itulah langkah-langkah utama dalam kode tersebut. Program ini sederhana namun penting untuk memahami bagaimana membaca input dari pengguna dan menampilkan output dalam bahasa pemrograman Java.

- **Output :**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nifal\Downloads\Tugas-PB0\Tugas 3\Tugas3_SourceCode>java BacaString.java

Baca string dan Integer:
masukkan sebuah string: nifal
String yang dibaca : nifal
C:\Users\nifal\Downloads\Tugas-PB0\Tugas 3\Tugas3_SourceCode>
```

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 program**

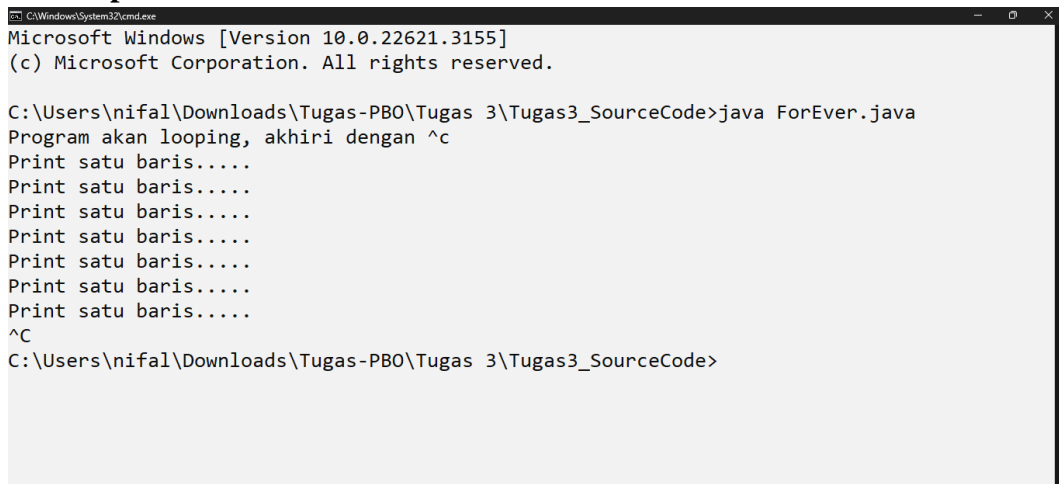
Program yang menggunakan loop **while** untuk mencetak baris teks terus-menerus sampai program dihentikan secara paksa (biasanya dengan menekan Ctrl + C).

Berikut adalah penjelasan langkah-langkah dari kode tersebut:

1. Deklarasi Kelas: Program mengandung kelas tunggal bernama **Forever**.
2. Metode **main**: Ini adalah titik masuk program Java. Di dalamnya, instruksi-instruksi yang akan dieksekusi pada saat program dijalankan ditempatkan.
3. Pesan Awal: Program mencetak pesan "Program akan looping, akhiri dengan ^c" ke konsol. Pesan ini memberi tahu pengguna bahwa program akan terus berjalan dalam loop.
4. Loop **while(true)**: Program memulai sebuah loop **while** dengan kondisi **true**, yang berarti loop akan terus berjalan selama kondisi tersebut benar.
5. Isi Loop: Di dalam loop, program mencetak pesan "Print satu baris....." ke konsol. Ini adalah baris yang akan dicetak berulang kali dalam loop.
6. Penanganan Kesalahan: Program tidak menyertakan penanganan kesalahan atau cara keluar dari loop. Oleh karena itu, satu-satunya cara untuk menghentikan program adalah dengan menekan kombinasi tombol Ctrl + C untuk menghentikan eksekusi secara paksa.
7. Penutup: Kurung kurawal menandai akhir dari blok kode dalam metode **main** dan kelas **Forever**.

Program ini adalah contoh dari loop tak terbatas yang akan terus berjalan tanpa henti sampai dihentikan secara paksa oleh pengguna. Ini bukanlah praktik yang baik dalam pengembangan perangkat lunak, kecuali jika ada alasan khusus untuk melakukan hal tersebut, seperti pengembangan perangkat lunak untuk tujuan pendidikan atau demonstrasi.

- **Output**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nifal\Downloads\Tugas-PB0\Tugas 3\Tugas3_SourceCode>java ForEver.java
Program akan looping, akhiri dengan ^c
Print satu baris.....
Print satu baris.....
Print satu baris.....
Print satu baris.....
Print satu baris.....
Print satu baris.....
Print satu baris.....
^C
C:\Users\nifal\Downloads\Tugas-PB0\Tugas 3\Tugas3_SourceCode>
```

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 program**

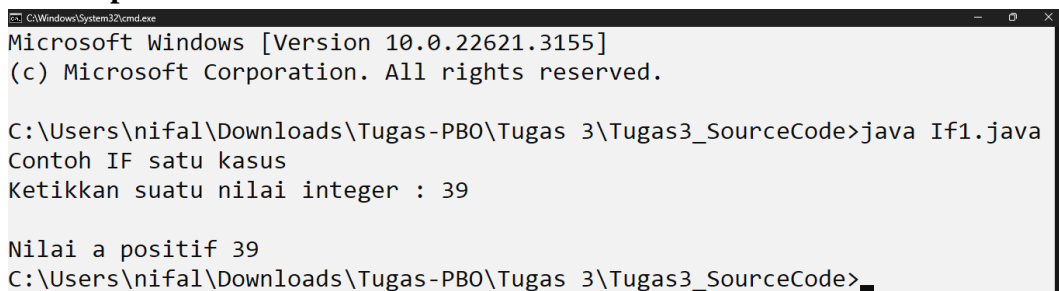
Program yang menggunakan struktur kontrol **if** untuk memeriksa apakah sebuah nilai integer positif atau tidak.

Berikut adalah penjelasan langkah-langkah dari kode tersebut:

1. Deklarasi Kelas: Program mengandung kelas tunggal bernama **If1**.
2. Metode **main**: Ini adalah titik masuk program Java. Di dalamnya, instruksi-instruksi yang akan dieksekusi pada saat program dijalankan ditempatkan.
3. Import Paket: Program mengimpor kelas **Scanner** dari paket **java.util** untuk membaca input dari pengguna.
4. Deklarasi Variabel: Variabel lokal **masukan** dideklarasikan sebagai objek **Scanner** untuk membaca input dari pengguna, dan variabel lokal **a** dideklarasikan sebagai bilangan bulat untuk menyimpan nilai yang dimasukkan.
5. Pesan Awal: Program mencetak pesan "Contoh IF satu kasus" dan "Ketikkan suatu nilai integer : " ke konsol. Pesan-pesan ini memberi petunjuk kepada pengguna untuk memasukkan nilai integer.
6. Membaca Input: Program menggunakan metode **nextInt()** dari objek **Scanner** untuk membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya dalam variabel **a**.
7. Struktur Kontrol **if**: Program menggunakan struktur kontrol **if** untuk memeriksa apakah nilai yang dimasukkan (**a**) adalah positif atau tidak. Jika nilai **a** lebih besar dari atau sama dengan 0, maka pesan "Nilai a positif" diikuti dengan nilai **a** dicetak ke konsol.
8. Penutup: Kurung kurawal menandai akhir dari blok kode dalam metode **main** dan kelas **If1**.

Program ini sederhana namun penting untuk memahami penggunaan struktur kontrol **if** dalam Java. Ini menunjukkan bagaimana kita dapat mengontrol alur program berdasarkan kondisi tertentu.

- **Output**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>java If1.java
Contoh IF satu kasus
Ketikkan suatu nilai integer : 39

Nilai a positif 39
C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>
```

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 program**

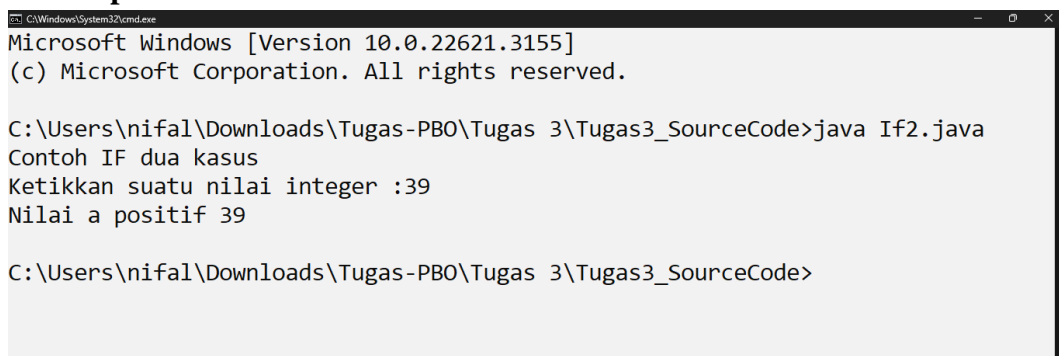
Program yang menggunakan struktur kontrol **if-else** untuk memeriksa apakah sebuah nilai integer positif atau negatif.

Berikut adalah penjelasan langkah-langkah dari kode tersebut:

1. Deklarasi Kelas: Program mengandung kelas tunggal bernama **If2**.
2. Metode **main**: Ini adalah titik masuk program Java. Di dalamnya, instruksi-instruksi yang akan dieksekusi pada saat program dijalankan ditempatkan.
3. Import Paket: Program mengimpor kelas **Scanner** dari paket **java.util** untuk membaca input dari pengguna.
4. Deklarasi Variabel: Variabel lokal **masukan** dideklarasikan sebagai objek **Scanner** untuk membaca input dari pengguna, dan variabel lokal **a** dideklarasikan sebagai bilangan bulat untuk menyimpan nilai yang dimasukkan.
5. Pesan Awal: Program mencetak pesan "Contoh IF dua kasus" dan "Ketikkan suatu nilai integer :" ke konsol. Pesan-pesan ini memberi petunjuk kepada pengguna untuk memasukkan nilai integer.
6. Membaca Input: Program menggunakan metode **nextInt()** dari objek **Scanner** untuk membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya dalam variabel **a**.
7. Struktur Kontrol **if-else**: Program menggunakan struktur kontrol **if-else** untuk memeriksa apakah nilai yang dimasukkan (**a**) adalah positif atau negatif. Jika nilai **a** lebih besar dari atau sama dengan 0, maka pesan "Nilai a positif" diikuti dengan nilai **a** dicetak ke konsol. Jika tidak, pesan "Nilai a negatif" diikuti dengan nilai **a** dicetak ke konsol.
8. Penutup: Kurung kurawal menandai akhir dari blok kode dalam metode **main** dan kelas **If2**.

Program ini menunjukkan bagaimana kita dapat menggunakan struktur kontrol **if-else** untuk memeriksa dua kasus yang saling komplementer dan melakukan tindakan yang sesuai berdasarkan kondisi yang diberikan.

- **Output**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>java If2.java
Contoh IF dua kasus
Ketikkan suatu nilai integer :39
Nilai a positif 39

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>
```

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 program**

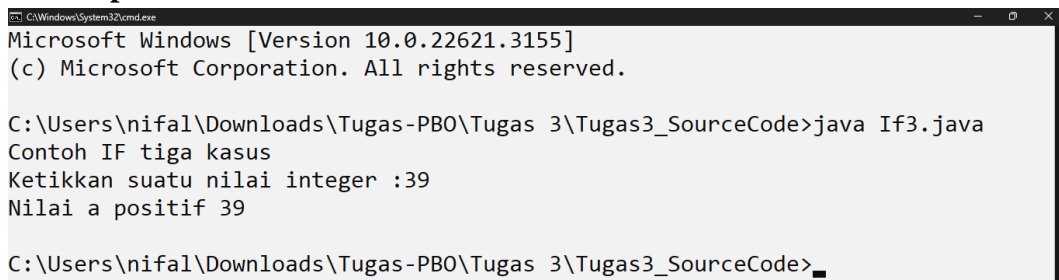
Program yang menggunakan struktur kontrol **if-else if-else** untuk memeriksa apakah sebuah nilai integer positif, nol, atau negatif.

Berikut adalah penjelasan langkah-langkah dari kode tersebut:

1. Deklarasi Kelas: Program mengandung kelas tunggal bernama **If3**.
2. Metode **main**: Ini adalah titik masuk program Java. Di dalamnya, instruksi-instruksi yang akan dieksekusi pada saat program dijalankan ditempatkan.
3. Import Paket: Program mengimpor kelas **Scanner** dari paket **java.util** untuk membaca input dari pengguna.
4. Deklarasi Variabel: Variabel lokal **masukan** dideklarasikan sebagai objek **Scanner** untuk membaca input dari pengguna, dan variabel lokal **a** dideklarasikan sebagai bilangan bulat untuk menyimpan nilai yang dimasukkan.
5. Pesan Awal: Program mencetak pesan "Contoh IF tiga kasus" dan "Ketikkan suatu nilai integer :" ke konsol. Pesan-pesan ini memberi petunjuk kepada pengguna untuk memasukkan nilai integer.
6. Membaca Input: Program menggunakan metode **nextInt()** dari objek **Scanner** untuk membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya dalam variabel **a**.
7. Struktur Kontrol **if-else if-else**: Program menggunakan struktur kontrol **if-else if-else** untuk memeriksa tiga kasus yang mungkin terjadi pada nilai **a**. Pertama, program memeriksa apakah **a** lebih besar dari 0. Jika benar, program mencetak pesan "Nilai a positif". Jika tidak, program memeriksa apakah **a** sama dengan 0. Jika benar, program mencetak pesan "Nilai Nol". Jika kedua kondisi sebelumnya salah, maka program mencetak pesan "Nilai a negatif".
8. Penutup: Kurung kurawal menandai akhir dari blok kode dalam metode **main** dan kelas **If3**.

Program ini menunjukkan bagaimana kita dapat menggunakan struktur kontrol **if-else if-else** untuk memeriksa beberapa kasus dan melakukan tindakan yang sesuai berdasarkan kondisi yang diberikan.

- **Output**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nifal\Downloads\Tugas-PB0\Tugas 3\Tugas3_SourceCode>java If3.java
Contoh IF tiga kasus
Ketikkan suatu nilai integer :39
Nilai a positif 39

C:\Users\nifal\Downloads\Tugas-PB0\Tugas 3\Tugas3_SourceCode>_
```

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;

        /* 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");

    }

}
```

- **Penjelasan program**

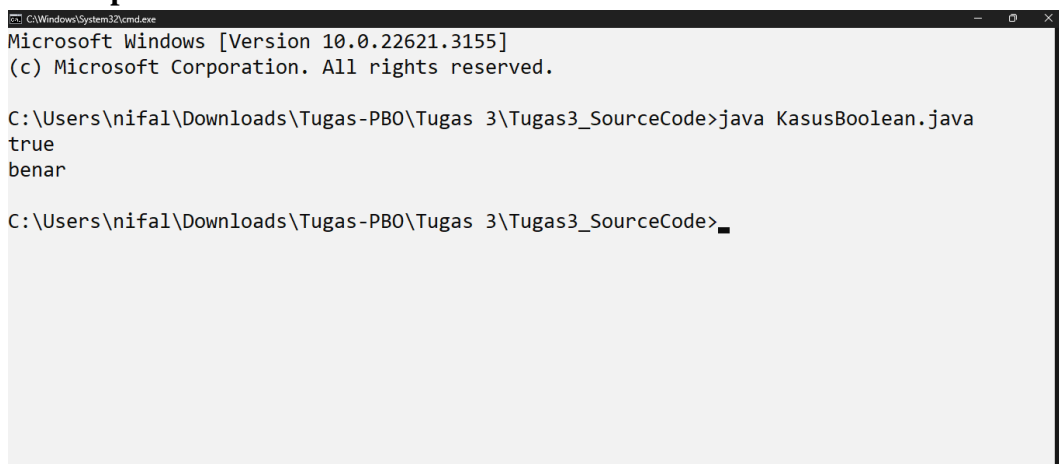
Program yang Anda berikan adalah contoh penggunaan ekspresi kondisional dengan tipe data boolean dalam bahasa pemrograman Java.

Berikut adalah penjelasan langkah-langkah dari kode tersebut:

1. Deklarasi Kelas: Program mengandung kelas tunggal bernama **KasusBoolean**.
2. Metode **main**: Ini adalah titik masuk program Java. Di dalamnya, instruksi-instruksi yang akan dieksekusi pada saat program dijalankan ditempatkan.
3. Deklarasi Variabel: Variabel lokal **bool** dideklarasikan sebagai tipe data boolean untuk menyimpan nilai kebenaran.
4. Inisialisasi Variabel: Variabel **bool** diinisialisasi dengan nilai **true**.
5. Ekspresi Kondisional: Program menggunakan struktur kontrol **if** untuk mengevaluasi nilai boolean **bool**. Jika nilai **bool** adalah **true**, maka pesan "true" dicetak ke konsol. Jika nilai **bool** adalah **false**, maka pesan "false" dicetak ke konsol.
6. Ekspresi Kondisional Negatif: Program menggunakan struktur kontrol **if** untuk mengevaluasi ekspresi negatif **!bool**, yang berarti memeriksa apakah nilai **bool** adalah **false**. Jika nilai **bool** adalah **false**, maka pesan "salah" dicetak ke konsol. Jika nilai **bool** adalah **true**, maka pesan "benar" dicetak ke konsol.
7. Penutup: Kurung kurawal menandai akhir dari blok kode dalam metode **main** dan kelas **KasusBoolean**.

Program ini menunjukkan cara menggunakan tipe data boolean dan ekspresi kondisional untuk mengontrol alur program berdasarkan nilai kebenaran.

- **Output**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nifal\Downloads\Tugas-PB0\Tugas 3\Tugas3_SourceCode>java KasusBoolean.java
true
benar

C:\Users\nifal\Downloads\Tugas-PB0\Tugas 3\Tugas3_SourceCode>_
```

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 program**

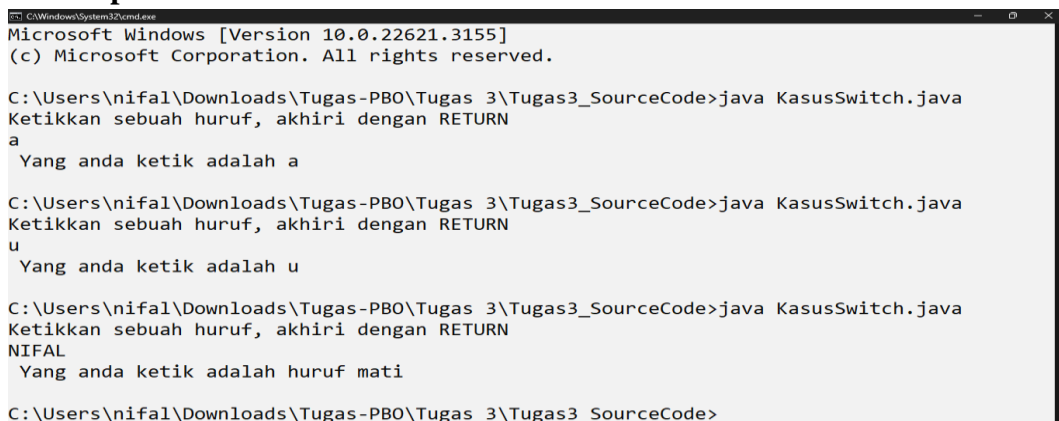
Program yang Anda berikan adalah contoh penggunaan struktur kontrol **switch-case** dalam bahasa pemrograman Java untuk memeriksa input pengguna terhadap beberapa kemungkinan nilai karakter (huruf).

Berikut adalah penjelasan langkah-langkah dari kode tersebut:

1. Deklarasi Kelas: Program mengandung kelas tunggal bernama **KasusSwitch**.
2. Metode **main**: Ini adalah titik masuk program Java. Di dalamnya, instruksi-instruksi yang akan dieksekusi pada saat program dijalankan ditempatkan.
3. Import Paket: Program tidak mengimpor paket tertentu dalam contoh ini.
4. Deklarasi Variabel: Variabel lokal **masukan** dideklarasikan sebagai objek **Scanner** untuk membaca input dari pengguna, dan variabel lokal **cc** dideklarasikan sebagai karakter untuk menyimpan karakter yang dimasukkan.
5. Pesan Awal: Program mencetak pesan "Ketikkan sebuah huruf, akhiri dengan RETURN" ke konsol. Pesan ini memberi petunjuk kepada pengguna untuk memasukkan sebuah huruf.
6. Membaca Input: Program menggunakan metode **next().charAt(0)** dari objek **Scanner** untuk membaca karakter yang dimasukkan oleh pengguna dan menyimpannya dalam variabel **cc**.
7. Struktur Kontrol **switch-case**: Program menggunakan struktur kontrol **switch-case** untuk memeriksa nilai karakter yang dimasukkan oleh pengguna (**cc**). Jika karakter yang dimasukkan adalah 'a', 'u', 'e', 'i', atau 'o', maka pesan yang sesuai dicetak ke konsol. Jika tidak, maka pesan "Yang anda ketik adalah huruf mati" dicetak ke konsol.
8. Penutup: Kurung kurawal menandai akhir dari blok kode dalam metode **main** dan kelas **KasusSwitch**.

Program ini menunjukkan bagaimana kita dapat menggunakan struktur kontrol **switch-case** untuk memeriksa beberapa kemungkinan nilai dari suatu ekspresi dan melakukan tindakan yang sesuai berdasarkan nilai yang diberikan. Dalam contoh ini, program memeriksa input pengguna terhadap beberapa huruf vokal dan memberikan tanggapan sesuai dengan karakter yang dimasukkan.

- **Output**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>java KasusSwitch.java
Ketikkan sebuah huruf, akhiri dengan RETURN
a
Yang anda ketik adalah a

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>java KasusSwitch.java
Ketikkan sebuah huruf, akhiri dengan RETURN
u
Yang anda ketik adalah u

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>java KasusSwitch.java
Ketikkan sebuah huruf, akhiri dengan RETURN
NIFAL
Yang anda ketik adalah huruf mati

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>
```

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 program**

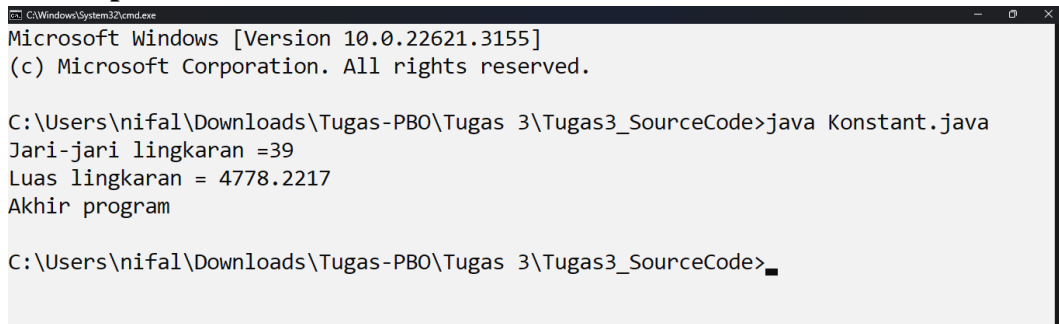
Program yang menggunakan konstanta untuk menghitung luas lingkaran berdasarkan jari-jari yang dimasukkan pengguna.

Berikut adalah penjelasan langkah-langkah dari kode tersebut:

1. Deklarasi Kelas: Program mengandung kelas tunggal bernama **Konstant**.
2. Metode **main**: Ini adalah titik masuk program Java. Di dalamnya, instruksi-instruksi yang akan dieksekusi pada saat program dijalankan ditempatkan.
3. Import Paket: Program mengimpor kelas **Scanner** dari paket **java.util** untuk membaca input dari pengguna.
4. Deklarasi Konstanta: Program mendeklarasikan konstanta **PHI** dengan nilai 3.1415, yang merupakan nilai estimasi dari phi (π).
5. Deklarasi Variabel: Variabel lokal **r** dideklarasikan sebagai tipe data float untuk menyimpan jari-jari lingkaran yang dimasukkan oleh pengguna.
6. Pesan Awal: Program mencetak pesan "Jari-jari lingkaran =" ke konsol. Pesan ini memberi petunjuk kepada pengguna untuk memasukkan jari-jari lingkaran.
7. Membaca Input: Program menggunakan metode **nextFloat()** dari objek **Scanner** untuk membaca nilai float yang dimasukkan oleh pengguna dan menyimpannya dalam variabel **r**.
8. Menghitung Luas Lingkaran: Program menghitung luas lingkaran dengan rumus $\pi * r * r$, di mana π adalah konstanta **PHI** dan **r** adalah jari-jari lingkaran yang dimasukkan oleh pengguna.
9. Menampilkan Output: Program mencetak pesan yang berisi luas lingkaran yang telah dihitung ke konsol.
10. Pesan Penutup: Program mencetak pesan "Akhir program" ke konsol sebagai penanda bahwa program telah selesai dijalankan.

Program ini menunjukkan bagaimana kita dapat menggunakan konstanta untuk menyimpan nilai yang tidak berubah dan menggunakannya dalam perhitungan atau operasi lainnya dalam program Java. Dalam contoh ini, konstanta **PHI** digunakan untuk menyimpan nilai phi (π) yang digunakan dalam perhitungan luas lingkaran.

- **Output**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>java Konstant.java
Jari-jari lingkaran =39
Luas lingkaran = 4778.2217
Akhir program

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>_
```

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 program**

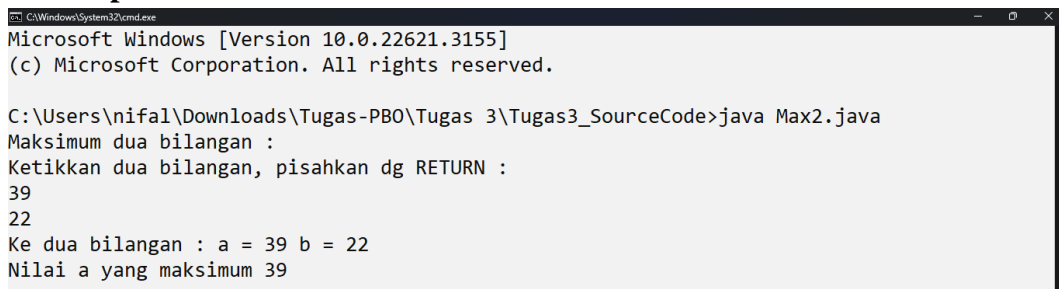
Program yang mengambil dua bilangan dari pengguna dan mencetak bilangan yang lebih besar di antara keduanya.

Berikut adalah penjelasan langkah-langkah dari kode tersebut:

1. Deklarasi Kelas: Program mengandung kelas tunggal bernama **Max2**.
2. Metode **main**: Ini adalah titik masuk program Java. Di dalamnya, instruksi-instruksi yang akan dieksekusi pada saat program dijalankan ditempatkan.
3. Import Paket: Program mengimpor kelas **Scanner** dari paket **java.util** untuk membaca input dari pengguna.
4. Deklarasi Variabel: Variabel lokal **a** dan **b** dideklarasikan sebagai bilangan bulat untuk menyimpan dua bilangan yang dimasukkan oleh pengguna.
5. Pesan Awal: Program mencetak pesan "Maksimum dua bilangan :" ke konsol, memberikan petunjuk kepada pengguna bahwa program akan mencari bilangan maksimum dari dua bilangan yang dimasukkan.
6. Meminta Input: Program mencetak pesan "Ketikkan dua bilangan, pisahkan dg RETURN :" ke konsol, memberikan petunjuk kepada pengguna untuk memasukkan dua bilangan secara berurutan, dipisahkan dengan menekan tombol "Enter" setelah memasukkan masing-masing bilangan. Nilai-nilai tersebut kemudian dimasukkan menggunakan metode **nextInt()** dari objek **Scanner** dan disimpan dalam variabel **a** dan **b** secara berurutan.
7. Membandingkan Bilangan: Program menggunakan struktur kontrol **if** untuk membandingkan nilai **a** dan **b**. Jika nilai **a** lebih besar dari atau sama dengan nilai **b**, maka pesan "Nilai a yang maksimum" diikuti dengan nilai **a** dicetak ke konsol. Jika tidak, pesan "Nilai b yang maksimum" diikuti dengan nilai **b** dicetak ke konsol.
8. Penutup: Kurung kurawal menandai akhir dari blok kode dalam metode **main** dan kelas **Max2**.

Program ini sederhana namun penting untuk memahami cara membandingkan nilai dan menggunakan struktur kontrol **if** dalam Java. Ini menunjukkan bagaimana kita dapat menulis kode untuk menemukan nilai maksimum dari dua bilangan yang dimasukkan oleh pengguna.

- **Output**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nifal\Downloads\Tugas-PB0\Tugas 3\Tugas3_SourceCode>java Max2.java
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
39
22
Ke dua bilangan : a = 39 b = 22
Nilai a yang maksimum 39
```

10. Kode Program

```
import java.util.Scanner;

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

public class PriFor {

    /**
     * @param args
     */

    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 program**

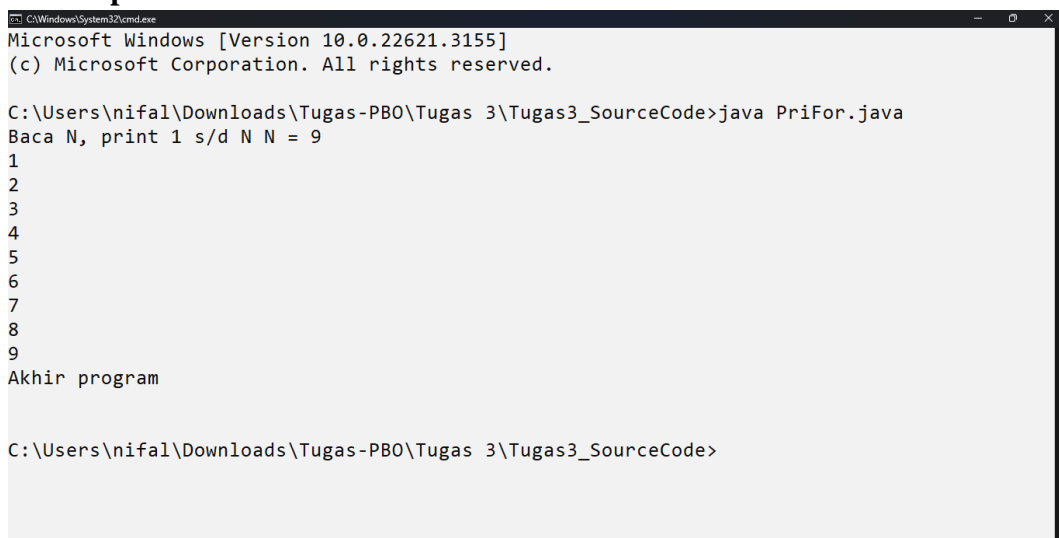
Program yang membaca nilai N dari pengguna dan mencetak angka dari 1 hingga N menggunakan loop **for**.

Berikut adalah penjelasan langkah-langkah dari kode tersebut:

1. Deklarasi Kelas: Program mengandung kelas tunggal bernama **PriFor**.
2. Metode **main**: Ini adalah titik masuk program Java. Di dalamnya, instruksi-instruksi yang akan dieksekusi pada saat program dijalankan ditempatkan.
3. Import Paket: Program mengimpor kelas **Scanner** dari paket **java.util** untuk membaca input dari pengguna.
4. Deklarasi Variabel: Variabel lokal **i** dan **N** dideklarasikan sebagai bilangan bulat. Variabel **masukan** dideklarasikan sebagai objek **Scanner** untuk membaca input dari pengguna.
5. Meminta Input: Program mencetak pesan "Baca N, print 1 s/d N" dan "N = " ke konsol, memberikan petunjuk kepada pengguna untuk memasukkan nilai N. Nilai N tersebut kemudian dimasukkan menggunakan metode **nextInt()** dari objek **Scanner** dan disimpan dalam variabel **N**.
6. Loop **for**: Program menggunakan loop **for** untuk mencetak angka dari 1 hingga N. Pada setiap iterasi, nilai **i** ditingkatkan, dan angka tersebut dicetak ke konsol.
7. Pesan Penutup: Program mencetak pesan "Akhir program" ke konsol sebagai penanda bahwa program telah selesai dijalankan.

Program ini sederhana namun efektif untuk memahami penggunaan loop **for** dalam Java. Ini menunjukkan bagaimana kita dapat menggunakan loop untuk mengulang tugas tertentu sejumlah N kali, dalam hal ini mencetak angka dari 1 hingga N.

- **Output**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nifal\Downloads\Tugas-PB0\Tugas 3\Tugas3_SourceCode>java PriFor.java
Baca N, print 1 s/d N N = 9
1
2
3
4
5
6
7
8
9
Akhir program

C:\Users\nifal\Downloads\Tugas-PB0\Tugas 3\Tugas3_SourceCode>
```

11. Kode Program

```
import java.util.Scanner;

/* Baca N, */
/* Print i = 1 s/d N dengan ITERATE */
public class PrintIterasi {

    /**
     * @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 ITERATE : \n");
        for (;){

            System.out.println(i); /* Proses */

            if (i == N)

                /* Kondisi Berhenti */ break;

            else {

                i++; /* Next Elmt */

            }

        } /* (i == N) */

    }

}
```

- **Penjelasan program**

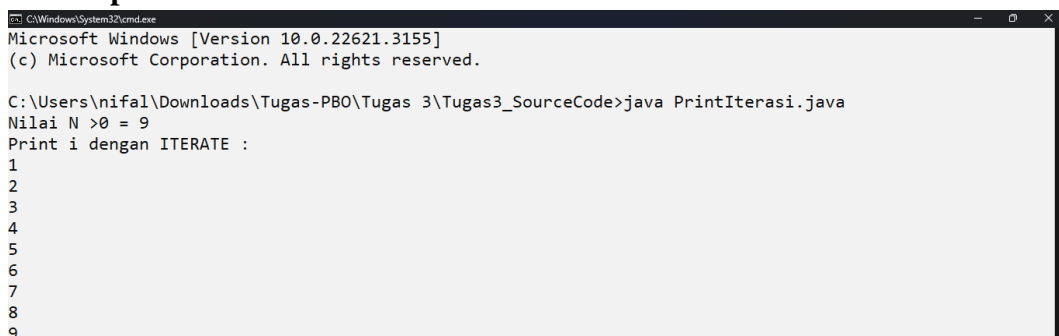
Program yang membaca nilai N dari pengguna dan mencetak nilai i dari 1 hingga N menggunakan iterasi dengan loop **for**.

Berikut adalah penjelasan langkah-langkah dari kode tersebut:

1. Deklarasi Kelas: Program mengandung kelas tunggal bernama **PrintIterasi**.
2. Metode **main**: Ini adalah titik masuk program Java. Di dalamnya, instruksi-instruksi yang akan dieksekusi pada saat program dijalankan ditempatkan.
3. Import Paket: Program mengimpor kelas **Scanner** dari paket **java.util** untuk membaca input dari pengguna.
4. Deklarasi Variabel: Variabel lokal **N** dan **i** dideklarasikan sebagai bilangan bulat. Variabel **masukan** dideklarasikan sebagai objek **Scanner** untuk membaca input dari pengguna.
5. Meminta Input: Program mencetak pesan "Nilai N >0 = " ke konsol, memberikan petunjuk kepada pengguna untuk memasukkan nilai N. Nilai N tersebut kemudian dimasukkan menggunakan metode **nextInt()** dari objek **Scanner** dan disimpan dalam variabel **N**.
6. Inisialisasi Iterasi: Variabel **i** diinisialisasi dengan nilai 1, yang merupakan nilai awal dari iterasi.
7. Loop **for**: Program menggunakan loop **for** tanpa kondisi awal, kondisi, atau pernyataan peningkatan (**for (;;)**) yang akan terus berjalan hingga ada pernyataan **break** di dalamnya.
8. Proses dan Pengecekan: Pada setiap iterasi, nilai **i** dicetak ke konsol. Kemudian, program memeriksa apakah nilai **i** sama dengan nilai **N**. Jika ya, maka loop dihentikan dengan menggunakan pernyataan **break**. Jika tidak, nilai **i** ditingkatkan dengan satu (**i++**) untuk memproses iterasi selanjutnya.

Program ini menunjukkan bagaimana kita dapat menggunakan loop **for** untuk melakukan iterasi tanpa kondisi awal atau akhir yang jelas. Dalam hal ini, loop akan terus berjalan sampai kondisi tertentu terpenuhi, yaitu ketika nilai **i** sama dengan nilai **N**.

- **Output**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>java PrintIterasi.java
Nilai N >0 = 9
Print i dengan ITERATE :
1
2
3
4
5
6
7
8
9
```

12. Kode Program

```
import java.util.Scanner;

public class PrintRepeat {

    /**
     * @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 program**

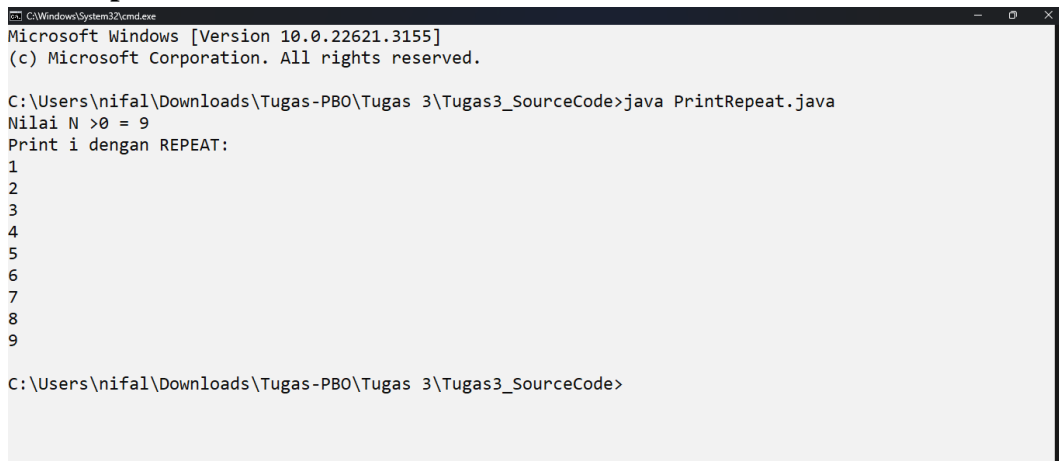
Program yang membaca nilai N dari pengguna dan mencetak nilai i dari 1 hingga N menggunakan iterasi dengan loop **do-while**.

Berikut adalah penjelasan langkah-langkah dari kode tersebut:

1. Deklarasi Kelas: Program mengandung kelas tunggal bernama **PrintRepeat**.
2. Metode **main**: Ini adalah titik masuk program Java. Di dalamnya, instruksi-instruksi yang akan dieksekusi pada saat program dijalankan ditempatkan.
3. Import Paket: Program tidak mengimpor paket tertentu dalam contoh ini.
4. Deklarasi Variabel: Variabel lokal **N** dan **i** dideklarasikan sebagai bilangan bulat. Variabel **masukan** dideklarasikan sebagai objek **Scanner** untuk membaca input dari pengguna.
5. Meminta Input: Program mencetak pesan "Nilai N >0 = " ke konsol, memberikan petunjuk kepada pengguna untuk memasukkan nilai N. Nilai N tersebut kemudian dimasukkan menggunakan metode **nextInt()** dari objek **Scanner** dan disimpan dalam variabel **N**.
6. Inisialisasi Iterasi: Variabel **i** diinisialisasi dengan nilai 1, yang merupakan nilai awal dari iterasi.
7. Loop **do-while**: Program menggunakan loop **do-while** untuk mencetak nilai i dari 1 hingga N. Pada setiap iterasi, nilai **i** dicetak ke konsol, kemudian nilai **i** ditingkatkan dengan satu (**i++**). Pengulangan dilanjutkan selama nilai **i** kurang dari atau sama dengan nilai **N**.

Program ini menunjukkan bagaimana kita dapat menggunakan loop **do-while** untuk melakukan iterasi minimal satu kali dan kemudian melakukan pengulangan selama kondisi tertentu terpenuhi. Dalam hal ini, program mencetak nilai i dari 1 hingga N sampai nilai i mencapai atau melampaui nilai N.

- **Output**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>java PrintRepeat.java
Nilai N >0 = 9
Print i dengan REPEAT:
1
2
3
4
5
6
7
8
9

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>
```

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 program**

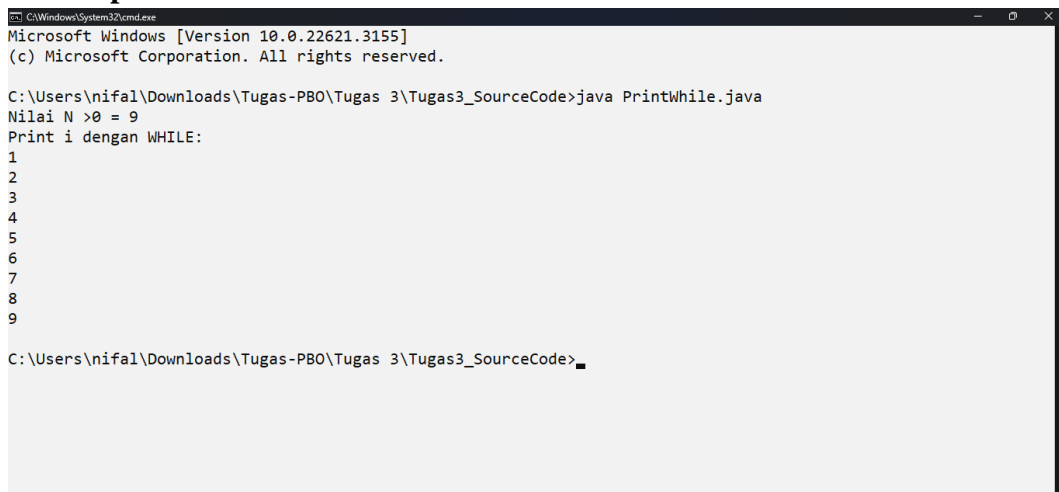
Program yang membaca nilai N dari pengguna dan mencetak nilai i dari 1 hingga N menggunakan iterasi dengan loop **while**.

Berikut adalah penjelasan langkah-langkah dari kode tersebut:

1. Deklarasi Kelas: Program mengandung kelas tunggal bernama **PrintWhile**.
2. Metode **main**: Ini adalah titik masuk program Java. Di dalamnya, instruksi-instruksi yang akan dieksekusi pada saat program dijalankan ditempatkan.
3. Import Paket: Program mengimpor kelas **Scanner** dari paket **java.util** untuk membaca input dari pengguna.
4. Deklarasi Variabel: Variabel lokal **N** dan **i** dideklarasikan sebagai bilangan bulat. Variabel **masukan** dideklarasikan sebagai objek **Scanner** untuk membaca input dari pengguna.
5. Meminta Input: Program mencetak pesan "Nilai N >0 = " ke konsol, memberikan petunjuk kepada pengguna untuk memasukkan nilai N. Nilai N tersebut kemudian dimasukkan menggunakan metode **nextInt()** dari objek **Scanner** dan disimpan dalam variabel **N**.
6. Inisialisasi Iterasi: Variabel **i** diinisialisasi dengan nilai 1, yang merupakan nilai awal dari iterasi.
7. Loop **while**: Program menggunakan loop **while** untuk mencetak nilai i dari 1 hingga N. Pada setiap iterasi, nilai **i** dicetak ke konsol, kemudian nilai **i** ditingkatkan dengan satu (**i++**). Pengulangan dilanjutkan selama nilai **i** kurang dari atau sama dengan nilai **N**.

Program ini menunjukkan bagaimana kita dapat menggunakan loop **while** untuk melakukan iterasi selama kondisi tertentu terpenuhi. Dalam hal ini, program mencetak nilai i dari 1 hingga N sampai nilai i mencapai atau melampaui nilai N.

- **Output**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>java PrintWhile.java
Nilai N >0 = 9
Print i dengan WHILE:
1
2
3
4
5
6
7
8
9

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>
```

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 program**

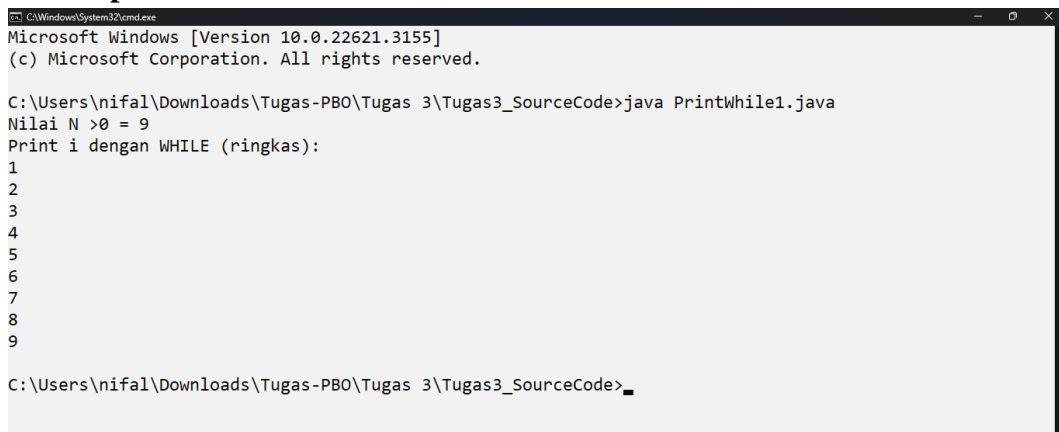
Program yang membaca nilai N dari pengguna dan mencetak nilai i dari 1 hingga N menggunakan iterasi dengan loop **while**.

Berikut adalah penjelasan langkah-langkah dari kode tersebut:

1. Deklarasi Kelas: Program mengandung kelas tunggal bernama **PrintWhile1**.
2. Metode **main**: Ini adalah titik masuk program Java. Di dalamnya, instruksi-instruksi yang akan dieksekusi pada saat program dijalankan ditempatkan.
3. Import Paket: Program mengimpor kelas **Scanner** dari paket **java.util** untuk membaca input dari pengguna.
4. Deklarasi Variabel: Variabel lokal **N** dideklarasikan sebagai bilangan bulat untuk menyimpan nilai yang dimasukkan oleh pengguna. Variabel **i** juga dideklarasikan sebagai bilangan bulat dan diinisialisasi dengan nilai 1.
5. Meminta Input: Program mencetak pesan "Nilai N >0 = " ke konsol, memberikan petunjuk kepada pengguna untuk memasukkan nilai N. Nilai N tersebut kemudian dimasukkan menggunakan metode **nextInt()** dari objek **Scanner** dan disimpan dalam variabel **N**.
6. Loop **while**: Program menggunakan loop **while** untuk mencetak nilai i dari 1 hingga N. Pada setiap iterasi, nilai **i** dicetak ke konsol, kemudian nilai **i** ditingkatkan dengan satu menggunakan operator peningkatan post-increment (**i++**). Pengulangan dilanjutkan selama nilai **i** kurang dari atau sama dengan nilai **N**.

Program ini menunjukkan cara yang lebih ringkas untuk melakukan iterasi dengan loop **while** dan mencetak nilai i dari 1 hingga N sampai nilai i mencapai atau melampaui nilai N. Dengan menggunakan operator peningkatan post-increment (**i++**) dalam instruksi cetak, pengurangan langkah dilakukan dalam loop tanpa memerlukan pernyataan terpisah.

- **Output**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>java PrintWhile1.java
Nilai N >0 = 9
Print i dengan WHILE (ringkas):
1
2
3
4
5
6
7
8
9

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>
```

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
        /* 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 */

    }
}
```

- **Penjelasan program**

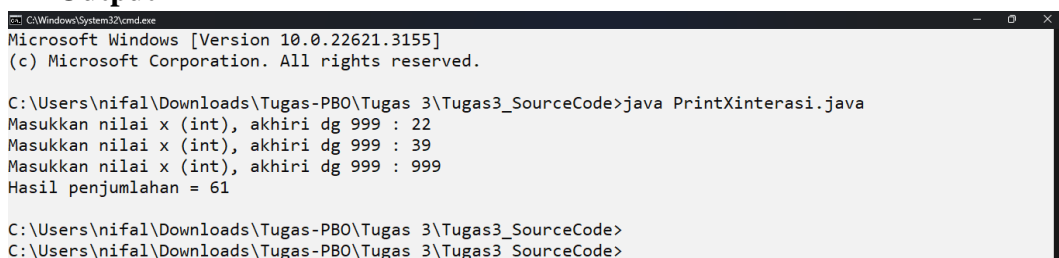
Program yang membaca nilai *x* dari pengguna dan menjumlahkan nilai yang dibaca hingga pengguna memasukkan nilai 999 sebagai penanda akhir.

Berikut adalah penjelasan langkah-langkah dari kode tersebut:

1. Deklarasi Kelas: Program mengandung kelas tunggal bernama **PrintXinterasi**.
2. Metode **main**: Ini adalah titik masuk program Java. Di dalamnya, instruksi-instruksi yang akan dieksekusi pada saat program dijalankan ditempatkan.
3. Import Paket: Program mengimpor kelas **Scanner** dari paket **java.util** untuk membaca input dari pengguna.
4. Deklarasi Variabel: Variabel lokal **Sum** dideklarasikan sebagai bilangan bulat untuk menyimpan hasil penjumlahan. Variabel *x* juga dideklarasikan sebagai bilangan bulat untuk menyimpan nilai yang dimasukkan oleh pengguna.
5. Meminta Input: Program mencetak pesan "Masukkan nilai *x* (int), akhiri dg 999 : " ke konsol, memberikan petunjuk kepada pengguna untuk memasukkan nilai *x*. Nilai *x* tersebut kemudian dimasukkan menggunakan metode **nextInt()** dari objek **Scanner** dan disimpan dalam variabel *x*.
6. Inisialisasi: Program memeriksa apakah nilai *x* yang dimasukkan oleh pengguna adalah 999. Jika iya, maka program mencetak pesan "Kasus kosong" karena tidak ada nilai yang dimasukkan. Jika tidak, maka nilai *x* dijadikan sebagai nilai awal dari variabel **Sum**.
7. Loop **for**: Program menggunakan loop **for** tanpa kondisi awal, kondisi, atau pernyataan peningkatan (**for (;;)**), yang akan terus berjalan hingga ada pernyataan **break** di dalamnya.
8. Pengecekan dan Penjumlahan: Pada setiap iterasi, program meminta pengguna untuk memasukkan nilai *x* baru. Jika nilai yang dimasukkan adalah 999, maka loop dihentikan dengan menggunakan pernyataan **break**. Jika tidak, nilai tersebut ditambahkan ke dalam variabel **Sum**.

Program ini menunjukkan bagaimana kita dapat menggunakan loop **for** untuk melakukan iterasi hingga kondisi tertentu terpenuhi, dalam hal ini hingga pengguna memasukkan nilai 999 sebagai penanda akhir. Dalam proses ini, program menjumlahkan nilai-nilai yang dimasukkan oleh pengguna dan mencetak hasilnya setelah pengguna selesai memasukkan data.

- **Output**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>java PrintXinterasi.java
Masukkan nilai x (int), akhiri dg 999 : 22
Masukkan nilai x (int), akhiri dg 999 : 39
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 61

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>
C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>
```

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 program**

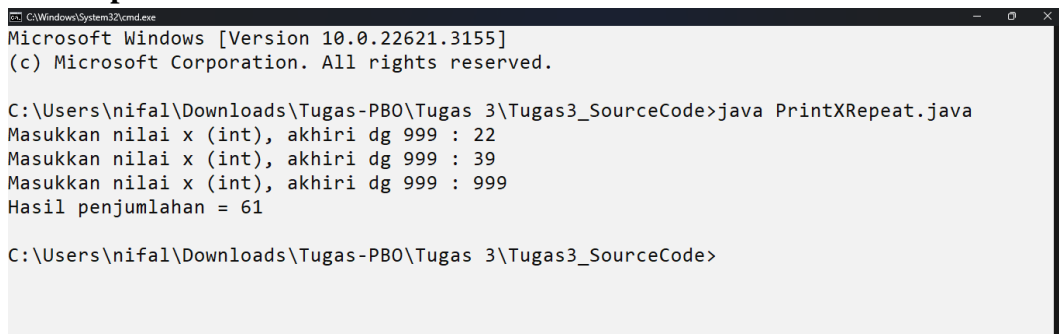
Program yang membaca nilai x dari pengguna dan menjumlahkan nilai yang dibaca hingga pengguna memasukkan nilai 999 sebagai penanda akhir.

Berikut adalah penjelasan langkah-langkah dari kode tersebut:

1. Deklarasi Kelas: Program mengandung kelas tunggal bernama **PrintXRepeat**.
2. Metode **main**: Ini adalah titik masuk program Java. Di dalamnya, instruksi-instruksi yang akan dieksekusi pada saat program dijalankan ditempatkan.
3. Import Paket: Program mengimpor kelas **Scanner** dari paket **java.util** untuk membaca input dari pengguna.
4. Deklarasi Variabel: Variabel lokal **Sum** dideklarasikan sebagai bilangan bulat untuk menyimpan hasil penjumlahan. Variabel x juga dideklarasikan sebagai bilangan bulat untuk menyimpan nilai yang dimasukkan oleh pengguna.
5. Meminta Input: Program mencetak pesan "Masukkan nilai x (int), akhiri dg 999 : " ke konsol, memberikan petunjuk kepada pengguna untuk memasukkan nilai x . Nilai x tersebut kemudian dimasukkan menggunakan metode **nextInt()** dari objek **Scanner** dan disimpan dalam variabel x .
6. Inisialisasi: Program memeriksa apakah nilai x yang dimasukkan oleh pengguna adalah 999. Jika iya, maka program mencetak pesan "Kasus kosong" karena tidak ada nilai yang dimasukkan. Jika tidak, variabel **Sum** diinisialisasi dengan nilai 0 sebagai nilai awal.
7. Loop **do-while**: Program menggunakan loop **do-while** untuk menjumlahkan nilai x hingga pengguna memasukkan nilai 999 sebagai penanda akhir. Pada setiap iterasi, nilai x ditambahkan ke dalam variabel **Sum**, kemudian program meminta pengguna untuk memasukkan nilai x baru. Pengulangan dilanjutkan selama nilai x yang dimasukkan tidak sama dengan 999.

Program ini menunjukkan bagaimana kita dapat menggunakan loop **do-while** untuk melakukan iterasi minimal satu kali dan kemudian melakukan pengulangan selama kondisi tertentu terpenuhi. Dalam hal ini, program menjumlahkan nilai-nilai yang dimasukkan oleh pengguna dan mencetak hasilnya setelah pengguna selesai memasukkan data.

- **Output**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nifal\Downloads\Tugas-PB0\Tugas 3\Tugas3_SourceCode>java PrintXRepeat.java
Masukkan nilai x (int), akhiri dg 999 : 22
Masukkan nilai x (int), akhiri dg 999 : 39
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 61

C:\Users\nifal\Downloads\Tugas-PB0\Tugas 3\Tugas3_SourceCode>
```

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 program**

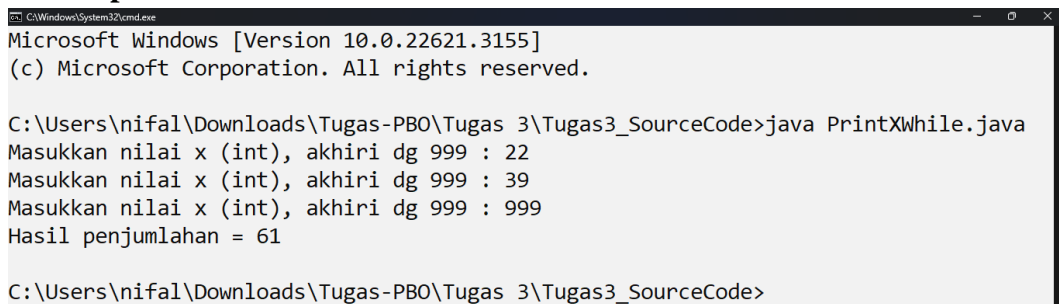
Program yang membaca nilai **x** dari pengguna dan menjumlahkan nilai yang dibaca hingga pengguna memasukkan nilai 999 sebagai penanda akhir.

Berikut adalah penjelasan langkah-langkah dari kode tersebut:

1. Deklarasi Kelas: Program mengandung kelas tunggal bernama **PrintXWhile**.
2. Metode **main**: Ini adalah titik masuk program Java. Di dalamnya, instruksi-instruksi yang akan dieksekusi pada saat program dijalankan ditempatkan.
3. Import Paket: Program mengimpor kelas **Scanner** dari paket **java.util** untuk membaca input dari pengguna.
4. Deklarasi Variabel: Variabel lokal **Sum** dideklarasikan sebagai bilangan bulat untuk menyimpan hasil penjumlahan. Variabel **x** juga dideklarasikan sebagai bilangan bulat untuk menyimpan nilai yang dimasukkan oleh pengguna.
5. Inisialisasi: Variabel **Sum** diinisialisasi dengan nilai 0 sebagai nilai awal.
6. Meminta Input: Program mencetak pesan "Masukkan nilai x (int), akhiri dg 999 : " ke konsol, memberikan petunjuk kepada pengguna untuk memasukkan nilai x. Nilai x tersebut kemudian dimasukkan menggunakan metode **nextInt()** dari objek **Scanner** dan disimpan dalam variabel **x**.
7. Loop **while**: Program menggunakan loop **while** untuk menjumlahkan nilai x hingga pengguna memasukkan nilai 999 sebagai penanda akhir. Pada setiap iterasi, nilai x ditambahkan ke dalam variabel **Sum**, kemudian program meminta pengguna untuk memasukkan nilai x baru. Pengulangan dilanjutkan selama nilai **x** yang dimasukkan tidak sama dengan 999.
8. Menampilkan Output: Setelah loop selesai, program mencetak hasil penjumlahan yang disimpan dalam variabel **Sum** ke konsol.

Program ini menunjukkan bagaimana kita dapat menggunakan loop **while** untuk melakukan iterasi selama kondisi tertentu terpenuhi. Dalam hal ini, program menjumlahkan nilai-nilai yang dimasukkan oleh pengguna dan mencetak hasilnya setelah pengguna selesai memasukkan data.

- **Output**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>java PrintXWhile.java
Masukkan nilai x (int), akhiri dg 999 : 22
Masukkan nilai x (int), akhiri dg 999 : 39
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 61

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>
```

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 */
        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);
    }
}
```

- **Penjelasan program**

Program yang mengandung prosedur dan fungsi.

Berikut adalah penjelasan langkah-langkah dari kode tersebut:

1. Deklarasi Kelas: Program mengandung kelas tunggal bernama **SubProgram**.
2. Fungsi **maxab**: Fungsi ini menerima dua parameter bilangan bulat **a** dan **b**, dan mengembalikan nilai maksimum dari kedua bilangan tersebut.
3. Prosedur **tukar**: Prosedur ini menerima dua parameter bilangan bulat **a** dan **b**, dan menukar nilai dari kedua parameter tersebut.
4. Metode **main**: Ini adalah titik masuk program Java. Di dalamnya, instruksi-instruksi yang akan dieksekusi pada saat program dijalankan ditempatkan.
5. Import Paket: Program mengimpor kelas **Scanner** dari paket **java.util** untuk membaca input dari pengguna.
6. Deklarasi Variabel: Variabel lokal **a** dan **b** dideklarasikan sebagai bilangan bulat untuk menyimpan nilai yang dimasukkan oleh pengguna.
7. Meminta Input: Program mencetak pesan untuk meminta pengguna memasukkan dua bilangan, yang dipisahkan oleh RETURN. Nilai-nilai tersebut kemudian dimasukkan menggunakan metode **nextInt()** dari objek **Scanner** dan disimpan dalam variabel **a** dan **b**.
8. Memanggil Fungsi: Program mencetak nilai maksimum dari dua bilangan yang dimasukkan oleh pengguna dengan memanggil fungsi **maxab(a, b)** dan mencetak hasilnya.
9. Memanggil Prosedur: Program mencetak pesan untuk menukar kedua bilangan dan memanggil prosedur **tukar(a, b)** untuk melakukan pertukaran nilai.

Program ini menunjukkan bagaimana kita dapat menggunakan fungsi untuk mengembalikan nilai berdasarkan perhitungan tertentu dan prosedur untuk melakukan tugas tertentu tanpa mengembalikan nilai. Dalam hal ini, fungsi **maxab** digunakan untuk menemukan nilai maksimum dari dua bilangan yang dimasukkan oleh pengguna, sedangkan prosedur **tukar** digunakan untuk menukar nilai dari dua bilangan.

- **Output**

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>java SubProgram.java
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
22
39
Ke dua bilangan : a = 22 b = 39
Maksimum = 39
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 39 b = 22
```

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 program**

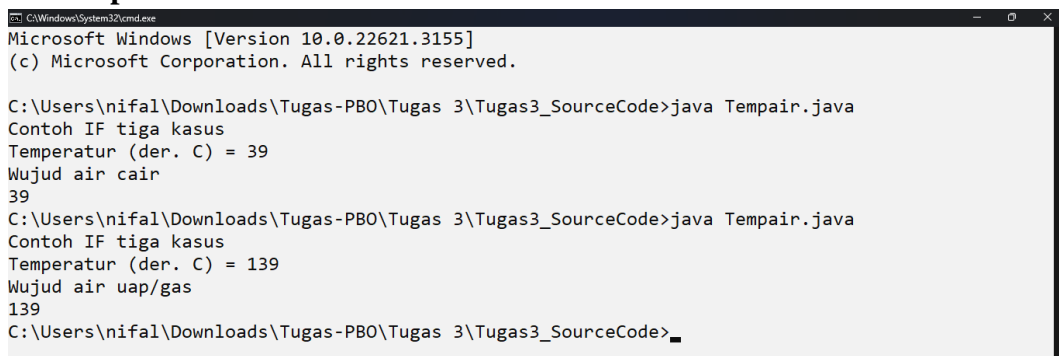
Program ini menggunakan struktur kontrol **if-else if** untuk menentukan wujud air berdasarkan suhu yang dimasukkan pengguna.

Berikut adalah penjelasan langkah-langkah dari kode tersebut:

1. Deklarasi Kelas: Program mengandung kelas tunggal bernama **Tempair**.
2. Metode **main**: Ini adalah titik masuk program Java. Di dalamnya, instruksi-instruksi yang akan dieksekusi pada saat program dijalankan ditempatkan.
3. Import Paket: Program tidak menggunakan paket tertentu.
4. Deklarasi Variabel: Variabel lokal **T** dideklarasikan sebagai bilangan bulat untuk menyimpan suhu yang dimasukkan oleh pengguna.
5. Meminta Input: Program mencetak pesan "Temperatur (der. C) = " ke konsol, memberikan petunjuk kepada pengguna untuk memasukkan suhu. Nilai suhu tersebut kemudian dimasukkan menggunakan metode **nextInt()** dari objek **Scanner** dan disimpan dalam variabel **T**.
6. Struktur **if-else if**: Program menggunakan struktur kontrol **if-else if** untuk menentukan wujud air berdasarkan suhu yang dimasukkan pengguna. Terdapat tiga kondisi yang diperiksa:
 - Jika suhu kurang dari 0, program mencetak pesan "Wujud air beku" berserta suhu yang dimasukkan.
 - Jika suhu di antara 0 dan 100 (inklusif), program mencetak pesan "Wujud air cair" berserta suhu yang dimasukkan.
 - Jika suhu lebih dari 100, program mencetak pesan "Wujud air uap/gas" berserta suhu yang dimasukkan.
7. Titik Penyimpulan: Program tidak memiliki pernyataan penutup yang jelas seperti **return** atau **break**. Ini tidak masalah dalam kasus ini karena program akan berakhir setelah eksekusi blok **main** selesai.

Program ini menunjukkan bagaimana kita dapat menggunakan struktur kontrol **if-else if** untuk mengevaluasi beberapa kondisi dan mengambil tindakan berdasarkan hasil evaluasi tersebut. Dalam hal ini, program menentukan wujud air berdasarkan suhu yang dimasukkan oleh pengguna.

- **Output**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>java Tempair.java
Contoh IF tiga kasus
Temperatur (der. C) = 39
Wujud air cair
39
C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>java Tempair.java
Contoh IF tiga kasus
Temperatur (der. C) = 139
Wujud air uap/gas
139
C:\Users\nifal\Downloads\Tugas-PBO\Tugas 3\Tugas3_SourceCode>_
```