

**LAPORAN
PEMROGRAMAN BEORIENTASI OBJEK**



Nama : Narendra Awangga
Nim : 13020220022
Dosen : Mardiyah Hasnawi, S.Kom.,M.T.
Kelas : A1

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS MUSLIM INDONESIA

MAKASSAR

2024

1. Baca String

- **Output**

```
C:\Windows\System32\cmd.e × + v
Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\SEMESTER 4\PB0\TUGAS 3\Sourcecode_13020220022>java BacaString

Baca string dan Integer:
masukkan sebuah string: Kamu
String yang dibaca : Kamu
C:\SEMESTER 4\PB0\TUGAS 3\Sourcecode_13020220022>
```

- **Penjelasan program**

Program ini adalah contoh sederhana dalam bahasa pemrograman Java yang bertujuan untuk membaca sebuah string dari input pengguna dan menampilkannya kembali ke layar. Berikut adalah penjelasan setiap bagian dari program tersebut:

1. Tujuan Program

Tujuan utama program ini adalah untuk mendemonstrasikan cara menggunakan kelas `BufferedReader` untuk membaca input string dari pengguna menggunakan keyboard, kemudian menampilkan string yang dibaca kembali ke layar.

2. Keyword, Tipe Data, dan Kelas yang Digunakan

- **`import java.io.BufferedReader;`**: Menggunakan keyword **`import`** untuk mengimpor kelas `BufferedReader` dari paket **`java.io`**, yang digunakan untuk membaca input karakter dari aliran input.
- **`import java.io.IOException;`**: Menggunakan keyword **`import`** untuk mengimpor `IOException` dari paket **`java.io`**, yang akan menangani kesalahan masukan/keluaran saat membaca input.
- **`import java.io.InputStreamReader;`**: Menggunakan keyword **`import`** untuk mengimpor kelas `InputStreamReader` dari paket **`java.io`**, yang digunakan untuk mengonversi aliran input byte menjadi aliran input karakter.
- **`import javax.swing.*;`**: Menggunakan keyword **`import`** untuk mengimpor seluruh kelas dari paket **`javax.swing`**, yang mungkin akan digunakan untuk pengembangan antarmuka grafis, meskipun tidak digunakan secara langsung dalam program ini.
- **`public class BacaString {`**: Deklarasi kelas utama dengan nama **`BacaString`**.

- **public static void main(String[] args) throws IOException {**: Deklarasi metode **main()** sebagai titik masuk program. Menggunakan tipe data **void** untuk menunjukkan bahwa metode **main()** tidak mengembalikan nilai apa pun. Menggunakan parameter **String[] args** untuk menerima argumen baris perintah. Dilemparkan **IOException** yang mungkin terjadi selama eksekusi program.
- **String str;**: Deklarasi variabel **str** dengan tipe data **String** untuk menyimpan string yang akan dibaca dari input pengguna.
- **BufferedReader datAIn = new BufferedReader(new InputStreamReader(System.in));**: Membuat objek **BufferedReader** bernama **datAIn** yang akan digunakan untuk membaca input dari pengguna. Objek ini menggunakan **InputStreamReader** untuk mengonversi aliran input byte (**System.in**) menjadi aliran input karakter yang dapat dibaca oleh **BufferedReader**.
- **System.out.print("\nBaca string dan Integer: \n");**: Menggunakan **System.out.print()** untuk menampilkan pesan ke layar yang meminta pengguna memasukkan sebuah string.
- **str = datAIn.readLine();**: Menggunakan metode **readLine()** dari objek **BufferedReader** untuk membaca sebuah baris string dari input pengguna dan menyimpannya ke dalam variabel **str**.
- **System.out.print("String yang dibaca : " + str);**: Menampilkan string yang dibaca dari input pengguna ke layar.

3. Komponen Utama:

- Penggunaan **BufferedReader** untuk membaca input dari pengguna.
- Penggunaan **System.out.print()** untuk menampilkan pesan ke layar.
- Membaca string dari input pengguna dan menampilkannya kembali ke layar.

Tujuan utama dari program ini adalah untuk memberikan pemahaman dasar tentang bagaimana membaca dan menampilkan string dari input pengguna dalam bahasa pemrograman Java.

2. Forever

- **Output**

[illegible]

- Penjelasan program

Program di atas adalah contoh sederhana dalam bahasa pemrograman Java yang menggunakan loop `while` untuk membuat sebuah program yang berjalan terus-menerus (forever). Berikut penjelasan setiap bagian dari program tersebut:

1. Tujuan Program:

Tujuan utama program ini adalah untuk mendemonstrasikan cara membuat sebuah program Java yang berjalan terus-menerus tanpa berhenti, kecuali jika program tersebut dihentikan secara paksa oleh pengguna menggunakan kombinasi tombol `^c` (Ctrl+C).

2. Keyword, Tipe Data, dan Kelas yang Digunakan:

- **`public class ForEver`** {: Deklarasi kelas utama dengan nama **`ForEver`**.
- **`public static void main(String[] args)`** {: Deklarasi metode **`main()`** sebagai titik masuk program. Menggunakan tipe data **`void`** untuk menunjukkan bahwa metode **`main()`** tidak mengembalikan nilai apa pun. Menggunakan parameter **`String[] args`** untuk menerima argumen baris perintah.
- **`System.out.println("Program akan looping, akhiri dengan ^c");`** {: Menggunakan **`System.out.println()`** untuk menampilkan pesan ke layar yang memberi tahu pengguna bahwa program akan terus melakukan looping dan dapat dihentikan dengan menekan kombinasi tombol `^c`.
- **`while (true)`** {: Menggunakan loop **`while`** dengan kondisi **`true`**, sehingga loop akan terus berjalan selama kondisinya benar (selalu benar dalam hal ini).
- **`System.out.print("Print satu baris \n");`** {: Menampilkan pesan ke layar dalam loop **`while`**, menandakan bahwa program sedang melakukan iterasi.

3. Komponen Utama:

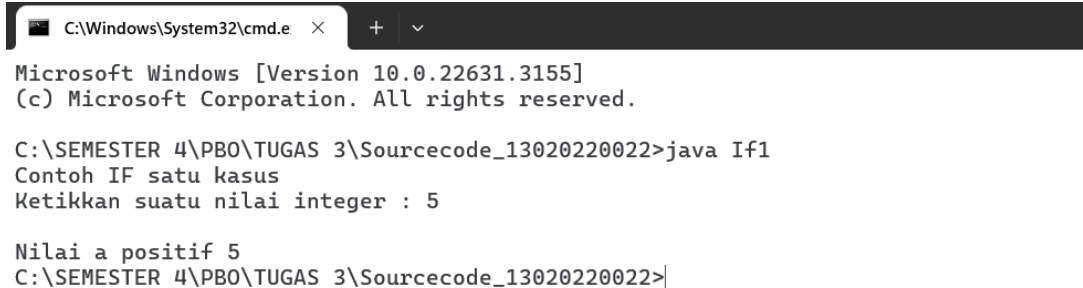
- Penggunaan **`System.out.println()`** untuk menampilkan pesan ke layar.
- Penggunaan loop **`while`** dengan kondisi **`true`** untuk membuat program berjalan terus-menerus.
- Menampilkan pesan di dalam loop **`while`** untuk menandakan bahwa program sedang berjalan.

Tujuan utama dari program ini adalah untuk memberikan pemahaman tentang cara membuat program Java yang berjalan terus-menerus menggunakan loop **`while(true)`**, dan menunjukkan cara menghentikan

program tersebut dengan menekan kombinasi tombol [^]c (Ctrl+C). Program ini dapat digunakan untuk tujuan debug, tes, atau demonstrasi.

3. If 1

- **Output**



```
C:\Windows\System32\cmd.e × + -
Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : 5

Nilai a positif 5
C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>|
```

- **Penjelasan program**

Program di atas adalah contoh sederhana dalam bahasa pemrograman Java yang menggunakan struktur kontrol if untuk mengecek apakah suatu nilai integer yang dimasukkan oleh pengguna merupakan bilangan positif. Berikut penjelasan setiap bagian dari program tersebut:

1. Tujuan Program

Tujuan utama program ini adalah untuk mendemonstrasikan penggunaan struktur kontrol if dalam Java untuk mengevaluasi suatu kondisi tertentu, yaitu apakah suatu nilai integer yang dimasukkan oleh pengguna adalah positif atau tidak.

2. Keyword, Tipe Data, dan Kelas yang Digunakan

- **import java.util.Scanner;**: Menggunakan keyword **import** untuk mengimpor kelas Scanner dari paket java.util. Kelas Scanner digunakan untuk membaca input dari pengguna.
- **public class If1 {**: Deklarasi kelas utama dengan nama **If1**.
- **public static void main(String[] args) {**: Deklarasi metode **main()** sebagai titik masuk program. Menggunakan tipe data **void** untuk menunjukkan bahwa metode **main()** tidak mengembalikan nilai apa pun. Menggunakan parameter **String[] args** untuk menerima argumen baris perintah.
- **Scanner masukan = new Scanner(System.in);**: Membuat objek **Scanner** dengan nama **masukan** untuk membaca input dari pengguna melalui keyboard.
- **int a;**: Mendeklarasikan variabel **a** dengan tipe data **int** untuk menyimpan nilai integer yang dimasukkan oleh pengguna.

- **System.out.print("Contoh IF satu kasus \n");**: Menampilkan pesan ke layar yang memberi informasi tentang program.
- **System.out.print("Ketikkan suatu nilai integer : ");**: Menampilkan pesan ke layar yang meminta pengguna untuk memasukkan suatu nilai integer.
- **a = masukan.nextInt();**: Membaca input integer dari pengguna dan menyimpannya ke dalam variabel **a** menggunakan metode **nextInt()** dari objek **masukan**.
- **if (a >= 0)**: Menggunakan struktur kontrol **if** untuk mengevaluasi apakah nilai **a** lebih besar dari atau sama dengan 0.
- **System.out.print("\nNilai a positif " + a);**: Menampilkan pesan ke layar jika kondisi dalam **if** terpenuhi, yaitu jika nilai **a** adalah bilangan positif.

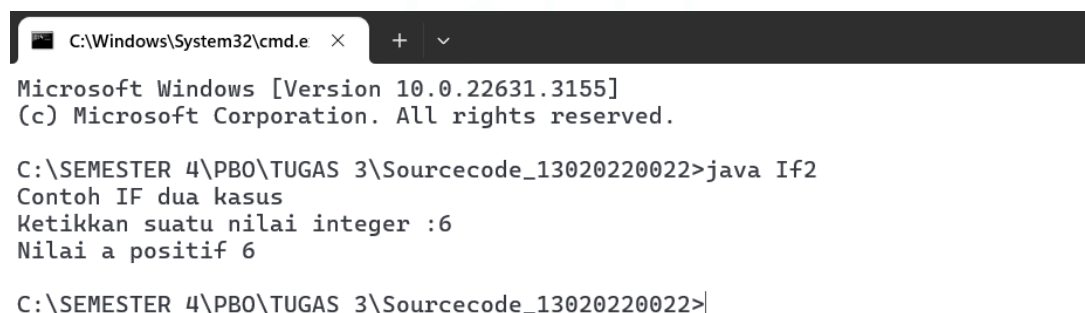
3. Komponen Utama:

- Penggunaan **Scanner** untuk membaca input dari pengguna.
- Penggunaan struktur kontrol **if** untuk mengevaluasi kondisi tertentu.

Tujuan utama dari program ini adalah untuk memberikan pemahaman tentang penggunaan struktur kontrol **if** dalam Java untuk mengevaluasi kondisi tertentu dan menampilkan hasilnya kepada pengguna. Program ini menunjukkan cara menggunakan **Scanner** untuk membaca input dari pengguna serta penggunaan **if** untuk mengevaluasi apakah nilai yang dimasukkan adalah bilangan positif.

4. If 2

• Output



```

C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>java If2
Contoh IF dua kasus
Ketikkan suatu nilai integer :6
Nilai a positif 6

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>

```

• Penjelasan program

Program di atas adalah contoh sederhana dalam bahasa pemrograman Java yang menggunakan struktur kontrol **if-else** untuk mengecek apakah

suatu nilai integer yang dimasukkan oleh pengguna merupakan bilangan positif atau negatif. Berikut penjelasan setiap bagian dari program tersebut:

1. Tujuan Program

Tujuan utama program ini adalah untuk mendemonstrasikan penggunaan struktur kontrol if-else dalam Java untuk mengevaluasi suatu kondisi tertentu, yaitu apakah suatu nilai integer yang dimasukkan oleh pengguna adalah positif atau negatif.

2. Keyword, Tipe Data, dan Kelas yang Digunakan

- **import java.util.Scanner;**: Menggunakan keyword **import** untuk mengimpor kelas **Scanner** dari paket **java.util**. Kelas **Scanner** digunakan untuk membaca input dari pengguna.
- **public class If2 {**: Deklarasi kelas utama dengan nama **If2**.
- **public static void main(String[] args) {**: Deklarasi metode **main()** sebagai titik masuk program. Menggunakan tipe data **void** untuk menunjukkan bahwa metode **main()** tidak mengembalikan nilai apa pun. Menggunakan parameter **String[] args** untuk menerima argumen baris perintah.
- **int a;**: Mendeklarasikan variabel **a** dengan tipe data **int** untuk menyimpan nilai integer yang dimasukkan oleh pengguna.
- **Scanner masukan = new Scanner(System.in);**: Membuat objek **Scanner** dengan nama **masukan** untuk membaca input dari pengguna melalui keyboard.
- **System.out.print("Contoh IF dua kasus \n");**: Menampilkan pesan ke layar yang memberi informasi tentang program.
- **System.out.print("Ketikkan suatu nilai integer :");**: Menampilkan pesan ke layar yang meminta pengguna untuk memasukkan suatu nilai integer.
- **a = masukan.nextInt();**: Membaca input integer dari pengguna dan menyimpannya ke dalam variabel **a** menggunakan metode **nextInt()** dari objek **masukan**.
- **if (a >= 0) {**: Menggunakan struktur kontrol **if** untuk mengevaluasi apakah nilai **a** lebih besar dari atau sama dengan 0.
- **System.out.println("Nilai a positif " + a);**: Menampilkan pesan ke layar jika kondisi dalam **if** terpenuhi, yaitu jika nilai **a** adalah bilangan positif.
- **else {**: Bagian **else** akan dieksekusi jika kondisi dalam **if** tidak terpenuhi.
- **System.out.println("Nilai a negatif " + a);**: Menampilkan pesan ke layar jika kondisi dalam **if** tidak terpenuhi, yaitu jika nilai **a** adalah bilangan negatif.

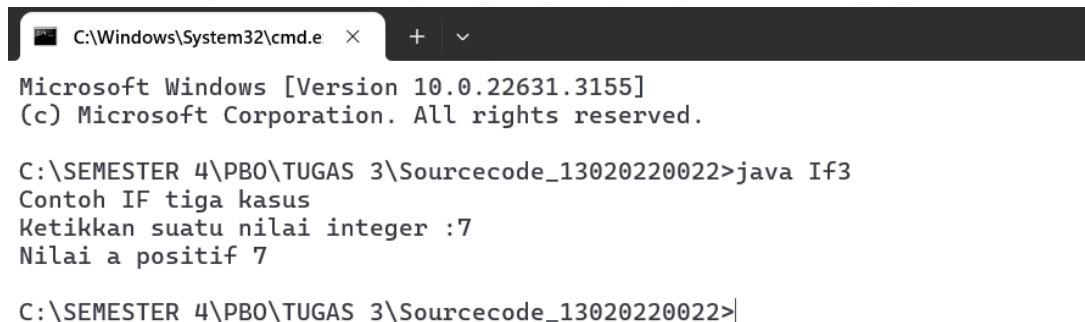
3. Komponen Utama:

- Penggunaan **Scanner** untuk membaca input dari pengguna.
- Penggunaan struktur kontrol **if-else** untuk mengevaluasi kondisi tertentu dan menentukan tindakan yang akan diambil berdasarkan hasil evaluasi kondisi.
- Menampilkan pesan ke layar sesuai dengan hasil evaluasi kondisi.

Tujuan utama dari program ini adalah untuk memberikan pemahaman tentang penggunaan struktur kontrol if-else dalam Java untuk mengevaluasi kondisi tertentu dan menentukan tindakan yang sesuai berdasarkan hasil evaluasi kondisi. Program ini menunjukkan cara menggunakan Scanner untuk membaca input dari pengguna serta penggunaan if- else untuk menentukan apakah nilai yang dimasukkan adalah bilangan positif atau negatif.

5. If 3

- **Output**



```

C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer :7
Nilai a positif 7

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>

```

- **Penjelasan program**

Program di atas adalah contoh sederhana dalam bahasa pemrograman Java yang menggunakan struktur kontrol if-else-if untuk mengecek apakah suatu nilai integer yang dimasukkan oleh pengguna merupakan bilangan positif, nol, atau negatif. Berikut penjelasan setiap bagian dari program tersebut:

1. Tujuan Program

Tujuan utama program ini adalah untuk mendemonstrasikan penggunaan struktur kontrol if-else-if dalam Java untuk mengevaluasi suatu kondisi tertentu, yaitu apakah suatu nilai integer yang dimasukkan oleh pengguna adalah positif, nol, atau negatif.

2. Keyword, Tipe Data, dan Kelas yang Digunakan

- **import java.util.Scanner;**: Menggunakan keyword **import** untuk mengimpor kelas **Scanner** dari paket **java.util**. Kelas **Scanner** digunakan untuk membaca input dari pengguna.
- **public class If3 {**: Deklarasi kelas utama dengan nama **If3**.
- **public static void main(String[] args) {**: Deklarasi metode **main()** sebagai titik masuk program. Menggunakan tipe data **void** untuk menunjukkan bahwa metode **main()** tidak mengembalikan nilai apa pun. Menggunakan parameter **String[] args** untuk menerima argumen baris perintah.
- **int a;**: Mendeklarasikan variabel **a** dengan tipe data **int** untuk menyimpan nilai integer yang dimasukkan oleh pengguna.
- **Scanner masukan = new Scanner(System.in);**: Membuat objek **Scanner** dengan nama **masukan** untuk membaca input dari pengguna melalui keyboard.
- **System.out.print("Contoh IF tiga kasus \n");**: Menampilkan pesan ke layar yang memberi informasi tentang program.
- **System.out.print("Ketikkan suatu nilai integer :");**: Menampilkan pesan ke layar yang meminta pengguna untuk memasukkan suatu nilai integer.
- **a = masukan.nextInt();**: Membaca input integer dari pengguna dan menyimpannya ke dalam variabel **a** menggunakan metode **nextInt()** dari objek **masukan**.
- **if (a > 0) {**: Menggunakan struktur kontrol **if** untuk mengevaluasi apakah nilai **a** lebih besar dari 0.
- **System.out.println("Nilai a positif " + a);**: Menampilkan pesan ke layar jika kondisi dalam **if** terpenuhi, yaitu jika nilai **a** adalah bilangan positif.
- **else if (a == 0) {**: Bagian **else if** akan dieksekusi jika kondisi dalam **if** tidak terpenuhi dan kondisi dalam **else if** terpenuhi.
- **System.out.println("Nilai Nol " + a);**: Menampilkan pesan ke layar jika kondisi dalam **else if** terpenuhi, yaitu jika nilai **a** adalah nol.
- **else {**: Bagian **else** akan dieksekusi jika kondisi dalam **if** dan **else if** tidak terpenuhi.
- **System.out.println("Nilai a negatif " + a);**: Menampilkan pesan ke layar jika kondisi dalam **else** terpenuhi, yaitu jika nilai **a** adalah bilangan negatif.

3. Komponen Utama:

- Penggunaan **Scanner** untuk membaca input dari pengguna.
- Penggunaan struktur kontrol **if-else-if** untuk mengevaluasi kondisi tertentu dan menentukan tindakan yang akan diambil berdasarkan hasil evaluasi kondisi.
- Menampilkan pesan ke layar sesuai dengan hasil evaluasi kondisi.

Tujuan utama dari program ini adalah untuk memberikan pemahaman tentang penggunaan struktur kontrol if-else-if dalam Java untuk mengevaluasi kondisi tertentu dan menentukan tindakan yang sesuai berdasarkan hasil evaluasi kondisi. Program ini menunjukkan cara menggunakan Scanner untuk membaca input dari pengguna serta penggunaan if-else-if untuk menentukan apakah nilai yang dimasukkan adalah bilangan positif, nol, atau negatif.

6. Kasus Boolean

- Output

```
C:\Windows\System32\cmd.e  ×  +  v
Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>java KasusBoolean
true
benar

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>|
```

- **Penjelasan program**

Program di atas adalah contoh sederhana dalam bahasa pemrograman Java yang menggunakan tipe data boolean untuk mendemonstrasikan penggunaan kondisi boolean dalam struktur kontrol if. Berikut penjelasan setiap bagian dari program tersebut:

1. Tujuan Program

Tujuan utama program ini adalah untuk mendemonstrasikan penggunaan tipe data boolean dalam Java serta penggunaan kondisi boolean dalam struktur kontrol if untuk mengevaluasi kebenaran atau kebenaran negasi dari suatu ekspresi boolean.

2. Keyword, Tipe Data, dan Kelas yang Digunakan

- **public class KasusBoolean {**: Deklarasi kelas utama dengan nama **KasusBoolean**.
- **public static void main(String[] args) {**: Deklarasi metode **main()** sebagai titik masuk program. Menggunakan tipe data **void** untuk menunjukkan bahwa metode **main()** tidak mengembalikan nilai apa pun. Menggunakan parameter **String[] args** untuk menerima argumen baris perintah.
- **boolean bool;**: Mendeklarasikan variabel **bool** dengan tipe data **boolean** untuk menyimpan nilai boolean.

- **bool = true;**: Menginisialisasi variabel **bool** dengan nilai boolean **true**.
- **if (bool) {**: Menggunakan struktur kontrol **if** untuk mengevaluasi nilai boolean **bool**. Jika nilainya **true**, maka blok kode di dalam **if** akan dieksekusi.
- **System.out.print("true\n");**: Menampilkan pesan ke layar jika kondisi dalam **if** terpenuhi, yaitu jika nilai boolean **bool** adalah **true**.
- **else**: Bagian **else** akan dieksekusi jika kondisi dalam **if** tidak terpenuhi.
- **if (!bool) {**: Menggunakan struktur kontrol **if** untuk mengevaluasi negasi dari nilai boolean **bool**. Jika nilainya **false**, maka blok kode di dalam **if** akan dieksekusi.
- **System.out.print("salah\n");**: Menampilkan pesan ke layar jika kondisi dalam **if** terpenuhi, yaitu jika nilai boolean **bool** adalah **false**.
- **else**: Bagian **else** akan dieksekusi jika kondisi dalam **if** tidak terpenuhi.
- **System.out.print("benar\n");**: Menampilkan pesan ke layar jika kondisi dalam **else** terpenuhi, yaitu jika nilai boolean **bool** adalah **true**.

3. Komponen Utama

- Penggunaan tipe data **boolean** untuk menyimpan nilai kebenaran.
- Penggunaan struktur kontrol **if** untuk mengevaluasi nilai boolean dan negasinya.
- Menampilkan pesan ke layar sesuai dengan hasil evaluasi kondisi.

Tujuan utama dari program ini adalah untuk memberikan pemahaman tentang penggunaan tipe data boolean dalam Java serta penggunaan kondisi boolean dalam struktur kontrol **if** untuk mengevaluasi kebenaran atau kebenaran negasi dari suatu ekspresi boolean. Program ini menunjukkan cara menginisialisasi, mengevaluasi, dan menggunakan nilai boolean dalam program Java.

7. Kasus Switch

- Output

```
C:\Windows\System32\cmd.e  X  +  v

Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
e
Yang anda ketik adalah e

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>|
```

• Penjelasan program

Program di atas adalah contoh sederhana dalam bahasa pemrograman Java yang menggunakan struktur kontrol switch-case untuk mengecek huruf yang dimasukkan oleh pengguna. Berikut penjelasan setiap bagian dari program tersebut:

1. Tujuan Program

Tujuan utama program ini adalah untuk mendemonstrasikan penggunaan struktur kontrol switch-case dalam Java untuk mengevaluasi nilai dari suatu variabel dan menjalankan kode yang sesuai dengan nilai tersebut.

2. Keyword, Tipe Data, dan Kelas yang Digunakan

- **import java.util.Scanner;**: Menggunakan keyword **import** untuk mengimpor kelas **Scanner** dari paket **java.util**. Kelas **Scanner** digunakan untuk membaca input dari pengguna.
- **public class KasusSwitch** {: Deklarasi kelas utama dengan nama **KasusSwitch**.
- **public static void main(String[] args)** {: Deklarasi metode **main()** sebagai titik masuk program. Menggunakan tipe data **void** untuk menunjukkan bahwa metode **main()** tidak mengembalikan nilai apa pun. Menggunakan parameter **String[] args** untuk menerima argumen baris perintah.
- **char cc;**: Mendeklarasikan variabel **cc** dengan tipe data **char** untuk menyimpan huruf yang dimasukkan oleh pengguna.
- **Scanner masukan = new Scanner(System.in);**: Membuat objek **Scanner** dengan nama **masukan** untuk membaca input dari pengguna melalui keyboard.
- **System.out.print("Ketikkan sebuah huruf, akhiri dengan RETURN \n");**: Menampilkan pesan ke layar yang meminta pengguna untuk memasukkan sebuah huruf.
- **cc = masukan.next().charAt(0);**: Membaca input dari pengguna dan menyimpan karakter pertama dari input tersebut ke dalam variabel **cc**.
- **switch (cc)** {: Menggunakan struktur kontrol **switch** untuk mengevaluasi nilai dari variabel **cc**.
- **case 'a':** {: Kasus pertama untuk nilai **cc** sama dengan **'a'**. Jika kondisi ini terpenuhi, maka blok kode di dalam **case 'a'** akan dieksekusi.
- **System.out.print(" Yang anda ketik adalah a\n");**: Menampilkan pesan ke layar jika kondisi dalam **case 'a'** terpenuhi.
- **break;**: Keyword **break** digunakan untuk menghentikan eksekusi switch-case dan keluar dari struktur kontrol switch.

- **default::** Bagian **default** akan dieksekusi jika tidak ada kasus yang cocok dengan nilai **cc**.
- **System.out.print(" Yang anda ketik adalah huruf mati\n");::** Menampilkan pesan ke layar dalam **default** jika tidak ada kasus yang cocok dengan nilai **cc**.

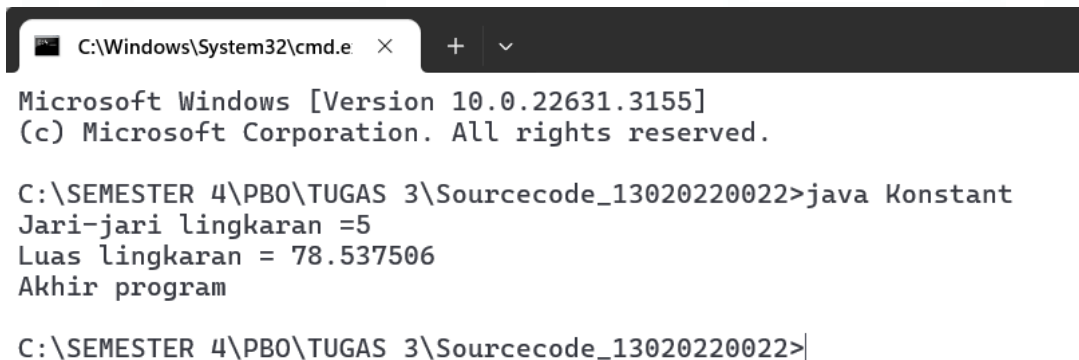
3. Komponen Utama:

- Penggunaan tipe data **char** untuk menyimpan nilai huruf.
- Penggunaan struktur kontrol **switch-case** untuk mengevaluasi nilai dari variabel **cc** dan menjalankan kode yang sesuai dengan nilai tersebut.
- Menampilkan pesan ke layar sesuai dengan hasil evaluasi nilai **cc**.

Tujuan utama dari program ini adalah untuk memberikan pemahaman tentang penggunaan struktur kontrol switch-case dalam Java untuk mengevaluasi nilai dari suatu variabel dan menjalankan kode yang sesuai dengan nilai tersebut. Program ini menunjukkan cara menggunakan switch-case untuk mengecek huruf yang dimasukkan oleh pengguna dan menampilkan pesan yang sesuai.

8. Konstant

- Output



```

C:\Windows\System32\cmd.e  x  +  v
Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>java Konstant
Jari-jari lingkaran =5
Luas lingkaran = 78.537506
Akhir program

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>|

```

- Penjelasan program

Program di atas adalah contoh sederhana dalam bahasa pemrograman Java yang menghitung luas lingkaran berdasarkan jari-jari yang dimasukkan oleh pengguna. Program ini menggunakan konstanta untuk nilai phi (π) dan variabel untuk menyimpan jari-jari lingkaran. Berikut penjelasan setiap bagian dari program tersebut:

1. Tujuan Program

Tujuan utama program ini adalah untuk mendemonstrasikan penggunaan konstanta dalam Java untuk menyimpan nilai yang tidak berubah (seperti phi

dalam perhitungan lingkaran) dan penggunaan variabel untuk menyimpan nilai yang dapat berubah (seperti jari-jari lingkaran).

2. Keyword, Tipe Data, dan Kelas yang Digunakan

- **import java.util.Scanner;**: Menggunakan keyword **import** untuk mengimpor kelas **Scanner** dari paket **java.util**. Kelas **Scanner** digunakan untuk membaca input dari pengguna.
- **public class Konstant** {: Deklarasi kelas utama dengan nama **Konstant**.
- **public static void main(String[] args)** {: Deklarasi metode **main()** sebagai titik masuk program. Menggunakan tipe data **void** untuk menunjukkan bahwa metode **main()** tidak mengembalikan nilai apa pun. Menggunakan parameter **String[] args** untuk menerima argumen baris perintah.
- **final float PHI = 3.1415f;**: Deklarasi konstanta **PHI** dengan tipe data **float** menggunakan keyword **final**. Konstanta ini menyimpan nilai phi (π) dengan presisi 3.1415.
- **float r;**: Mendeklarasikan variabel **r** dengan tipe data **float** untuk menyimpan jari-jari lingkaran yang dimasukkan oleh pengguna.
- **Scanner masukan = new Scanner(System.in);**: Membuat objek **Scanner** dengan nama **masukan** untuk membaca input dari pengguna melalui keyboard.
- **r = masukan.nextFloat();**: Membaca input float dari pengguna dan menyimpannya ke dalam variabel **r** menggunakan metode **nextFloat()** dari objek **masukan**.
- **System.out.print("Luas lingkaran = " + (PHI * r * r) + "\n");**: Menghitung dan menampilkan luas lingkaran ke layar menggunakan rumus πr^2 , di mana nilai phi diambil dari konstanta **PHI**.
- **System.out.print("Akhir program \n");**: Menampilkan pesan ke layar yang menandakan akhir program.

3. Komponen Utama

- Penggunaan konstanta **final** untuk menyimpan nilai phi (π) yang tidak berubah.
- Penggunaan variabel untuk menyimpan jari-jari lingkaran yang dimasukkan oleh pengguna.
- Penggunaan kelas **Scanner** untuk membaca input dari pengguna.
- Menggunakan rumus πr^2 untuk menghitung luas lingkaran dan menampilkannya ke layar.
- Menampilkan pesan ke layar untuk menandakan akhir program.

Tujuan utama dari program ini adalah untuk memberikan pemahaman tentang penggunaan konstanta dalam Java untuk menyimpan nilai yang tidak berubah dan penggunaan variabel untuk menyimpan nilai yang dapat

berubah. Program ini juga menunjukkan cara menghitung luas lingkaran berdasarkan jari-jari yang dimasukkan oleh pengguna menggunakan rumus matematika yang tepat.

9. Max

- Output

```
C:\Windows\System32\cmd.e  X  +  v

Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
3 5
Ke dua bilangan : a = 3 b = 5
Nilai b yang maksimum: 5

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>
```

- Penjelasan program

Program di atas adalah contoh sederhana dalam bahasa pemrograman Java yang mengambil dua bilangan dari pengguna dan menampilkan bilangan yang lebih besar di antara keduanya. Berikut penjelasan setiap bagian dari program tersebut:

1. Tujuan Program

Tujuan utama program ini adalah untuk mendemonstrasikan cara menggunakan struktur kontrol if-else dalam Java untuk menemukan bilangan maksimum dari dua bilangan yang dimasukkan oleh pengguna.

2. Keyword, Tipe Data, dan Kelas yang Digunakan

- **import java.util.Scanner;**: Menggunakan keyword **import** untuk mengimpor kelas Scanner dari paket java.util. Kelas Scanner digunakan untuk membaca input dari pengguna.
- **public class Max2 {**: Deklarasi kelas utama dengan nama **Max2**.
- **public static void main(String[] args) {**: Deklarasi metode **main()** sebagai titik masuk program. Menggunakan tipe data **void** untuk menunjukkan bahwa metode **main()** tidak mengembalikan nilai apa pun. Menggunakan parameter **String[] args** untuk menerima argumen baris perintah.
- **int a, b;**: Mendeklarasikan variabel **a** dan **b** dengan tipe data **int** untuk menyimpan dua bilangan yang dimasukkan oleh pengguna.

- **Scanner masukan = new Scanner(System.in);**: Membuat objek **Scanner** dengan nama **masukan** untuk membaca input dari pengguna melalui keyboard.
- **System.out.print("Maksimum dua bilangan : \n");**: Menampilkan pesan ke layar yang memberi informasi tentang tujuan program.
- **System.out.print("Ketikkan dua bilangan, pisahkan dg RETURN :\n");**: Menampilkan pesan ke layar yang meminta pengguna untuk memasukkan dua bilangan, dipisahkan dengan tombol Enter.
- **a = masukan.nextInt();**: Membaca input bilangan pertama dari pengguna dan menyimpannya dalam variabel **a** menggunakan metode **nextInt()** dari objek **masukan**.
- **b = masukan.nextInt();**: Membaca input bilangan kedua dari pengguna dan menyimpannya dalam variabel **b** menggunakan metode **nextInt()** dari objek **masukan**.
- **System.out.println("Ke dua bilangan : a = " + a + " b = " + b);**: Menampilkan bilangan yang dimasukkan oleh pengguna ke layar.
- **if (a >= b) {**: Menggunakan struktur kontrol **if** untuk mengevaluasi apakah nilai **a** lebih besar dari atau sama dengan nilai **b**.
- **System.out.println("Nilai a yang maksimum " + a);**: Menampilkan pesan ke layar jika kondisi dalam **if** terpenuhi, yaitu jika nilai **a** lebih besar dari atau sama dengan nilai **b**.
- **else {**: Bagian **else** akan dieksekusi jika kondisi dalam **if** tidak terpenuhi.
- **System.out.println("Nilai b yang maksimum: " + b);**: Menampilkan pesan ke layar jika kondisi dalam **if** tidak terpenuhi, yaitu jika nilai **b** lebih besar dari nilai **a**.

3. Komponen Utama

- Penggunaan kelas **Scanner** untuk membaca input dari pengguna.
- Penggunaan struktur kontrol **if-else** untuk menemukan bilangan maksimum dari dua bilangan yang dimasukkan.
- Menampilkan pesan ke layar yang memberikan informasi tentang bilangan yang dimasukkan oleh pengguna dan bilangan maksimum di antara keduanya.

Tujuan utama dari program ini adalah untuk memberikan pemahaman tentang cara menggunakan struktur kontrol **if-else** dalam Java untuk menemukan bilangan maksimum dari dua bilangan yang dimasukkan oleh pengguna. Program ini menunjukkan cara menggunakan **Scanner** untuk membaca input dari pengguna dan menggunakan struktur kontrol **if-else** untuk mengevaluasi dan menampilkan hasil yang sesuai.

10. PriFor

- **Output**

```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>java PriFor
Baca N, print 1 s/d N N = 4
1
2
3
4
Akhir program

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>
```

- **Penjelasan program**

Program di atas adalah contoh sederhana dalam bahasa pemrograman Java yang menggunakan perulangan **for** untuk mencetak angka dari 1 hingga N, di mana N adalah bilangan yang dimasukkan oleh pengguna. Berikut adalah penjelasan setiap bagian dari program tersebut:

1. Tujuan Program

Tujuan utama program ini adalah untuk mendemonstrasikan penggunaan perulangan **for** dalam Java untuk mencetak angka dari 1 hingga N, di mana N adalah bilangan yang dimasukkan oleh pengguna.

2. Keyword, Tipe Data, dan Kelas yang Digunakan

- **import java.util.Scanner;**: Menggunakan keyword **import** untuk mengimpor kelas **Scanner** dari paket **java.util**. Kelas **Scanner** digunakan untuk membaca input dari pengguna.
- **public class PriFor {**: Deklarasi kelas utama dengan nama **PriFor**.
- **public static void main(String[] args) {**: Deklarasi metode **main()** sebagai titik masuk program. Menggunakan tipe data **void** untuk menunjukkan bahwa metode **main()** tidak mengembalikan nilai apa pun. Menggunakan parameter **String[] args** untuk menerima argumen baris perintah.
- **int i, N;**: Mendeklarasikan dua variabel **i** dan **N** dengan tipe data **int**. Variabel **N** digunakan untuk menyimpan bilangan yang dimasukkan oleh pengguna, sedangkan variabel **i** digunakan sebagai variabel iterasi dalam perulangan **for**.

- **Scanner masukan = new Scanner(System.in);**: Membuat objek **Scanner** dengan nama **masukan** untuk membaca input dari pengguna melalui keyboard.
- **System.out.print("Baca N, print 1 s/d N ");**: Menampilkan pesan ke layar yang memberi informasi tentang tujuan program.
- **System.out.print("N = ");**: Menampilkan pesan ke layar yang meminta pengguna untuk memasukkan nilai N.
- **N = masukan.nextInt();**: Membaca input bilangan N dari pengguna dan menyimpannya dalam variabel **N** menggunakan metode **nextInt()** dari objek **masukan**.
- **for (i = 1; i <= N; i++) {**: Memulai perulangan **for** dengan menginisialisasi variabel **i** dengan nilai 1. Perulangan akan terus berjalan selama nilai **i** kurang dari atau sama dengan nilai **N**, dan nilai **i** akan bertambah satu setiap iterasi.
- **System.out.println(i);**: Mencetak nilai **i** ke layar pada setiap iterasi perulangan, sehingga mencetak angka dari 1 hingga N.
- **System.out.println("Akhir program \n");**: Menampilkan pesan ke layar yang menandakan akhir program.

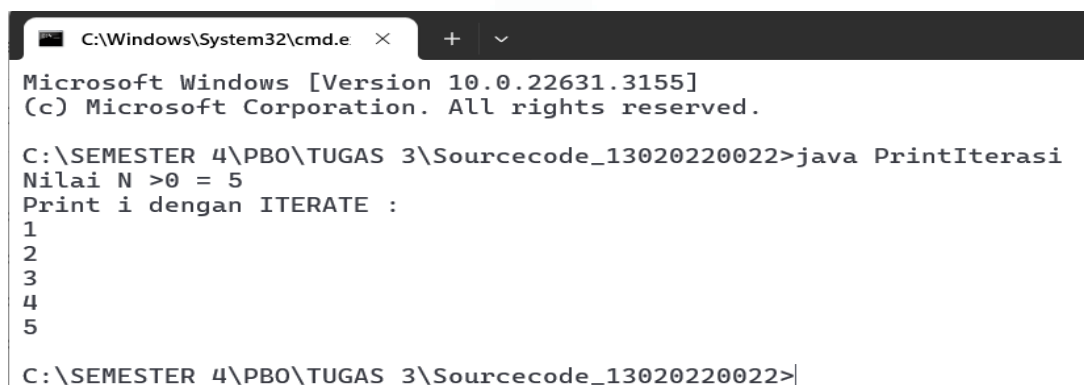
3. Komponen Utama

- Penggunaan perulangan **for** untuk mencetak angka dari 1 hingga N.
- Penggunaan kelas **Scanner** untuk membaca input dari pengguna.
- Menampilkan pesan ke layar yang memberikan informasi tentang tujuan program.
- Menampilkan pesan ke layar yang menandakan akhir program.

Tujuan utama dari program ini adalah untuk memberikan pemahaman tentang cara menggunakan perulangan **for** dalam Java untuk mencetak angka dari 1 hingga N, di mana N adalah bilangan yang dimasukkan oleh pengguna. Program ini menunjukkan cara menggunakan perulangan **for** untuk melakukan iterasi dan mencetak nilai dari suatu rentang tertentu.

11. Print Iterasi

- **Output**



```

C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>java PrintIterasi
Nilai N >0 = 5
Print i dengan ITERATE :
1
2
3
4
5

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>

```

- **Penjelasan program**

Program di atas adalah contoh sederhana dalam bahasa pemrograman Java yang menggunakan perulangan **for** untuk mencetak angka dari 1 hingga N, di mana N adalah bilangan yang dimasukkan oleh pengguna. Berikut adalah penjelasan setiap bagian dari program tersebut:

1. **Tujuan Program**

Tujuan utama program ini adalah untuk mendemonstrasikan penggunaan perulangan **for** dalam Java untuk mencetak angka dari 1 hingga N, di mana N adalah bilangan yang dimasukkan oleh pengguna.

2. **Keyword, Tipe Data, dan Kelas yang Digunakan**

- **import java.util.Scanner;**: Menggunakan keyword **import** untuk mengimpor kelas Scanner dari paket java.util. Kelas Scanner digunakan untuk membaca input dari pengguna.
- **public class PrintIterasi {**: Deklarasi kelas utama dengan nama **PrintIterasi**.
- **public static void main(String[] args) {**: Deklarasi metode **main()** sebagai titik masuk program. Menggunakan tipe data **void** untuk menunjukkan bahwa metode **main()** tidak mengembalikan nilai apa pun. Menggunakan parameter **String[] args** untuk menerima argumen baris perintah.
- **int N;**: Mendeklarasikan variabel **N** dengan tipe data **int** untuk menyimpan bilangan yang dimasukkan oleh pengguna.
- **int i;**: Mendeklarasikan variabel **i** dengan tipe data **int** untuk digunakan sebagai variabel iterasi dalam perulangan.
- **Scanner masukan = new Scanner(System.in);**: Membuat objek **Scanner** dengan nama **masukan** untuk membaca input dari pengguna melalui keyboard.
- **System.out.print("Nilai N >0 = ");**: Menampilkan pesan ke layar yang meminta pengguna untuk memasukkan nilai N.
- **N = masukan.nextInt();**: Membaca input bilangan N dari pengguna dan menyimpannya dalam variabel **N** menggunakan metode **nextInt()** dari objek **masukan**.
- **i = 1;**: Menginisialisasi variabel **i** dengan nilai 1.
- **System.out.print("Print i dengan ITERATE : \n");**: Menampilkan pesan ke layar yang memberi informasi tentang metode pencetakan menggunakan iterasi.
- **for (;;) {**: Memulai perulangan **for** tanpa kondisi awal, tanpa kondisi penghentian, dan tanpa ekspresi iterasi. Ini menciptakan perulangan tak terbatas.
- **System.out.println(i);**: Mencetak nilai **i** ke layar pada setiap iterasi perulangan.

- **if (i == N) break;**: Menggunakan pernyataan **break** untuk keluar dari perulangan jika nilai **i** sama dengan nilai **N**.
- **else { i++; }**: Jika nilai **i** belum mencapai nilai **N**, maka nilai **i** akan ditambah satu dan perulangan akan terus berlanjut.

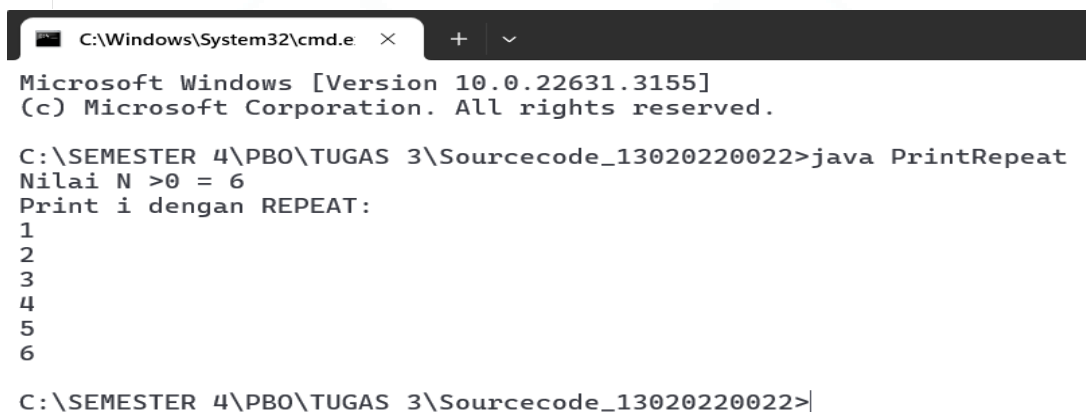
3. Komponen Utama:

- Penggunaan perulangan **for** tanpa kondisi awal, tanpa kondisi penghentian, dan tanpa ekspresi iterasi untuk membuat perulangan tak terbatas.
- Penggunaan kelas Scanner untuk membaca input dari pengguna.
- Menampilkan pesan ke layar yang memberikan informasi tentang metode pencetakan menggunakan iterasi.
- Menampilkan nilai variabel **i** ke layar pada setiap iterasi perulangan.
- Penggunaan pernyataan **break** untuk keluar dari perulangan jika kondisi tertentu terpenuhi.

Tujuan utama dari program ini adalah untuk memberikan pemahaman tentang cara menggunakan perulangan **for** dalam Java untuk mencetak angka dari 1 hingga N, di mana N adalah bilangan yang dimasukkan oleh pengguna. Program ini juga menunjukkan cara membuat perulangan tak terbatas dan menggunakan pernyataan **break** untuk keluar dari perulangan saat kondisi tertentu terpenuhi.

12. Print Repeat

• Output



```

C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>java PrintRepeat
Nilai N >0 = 6
Print i dengan REPEAT:
1
2
3
4
5
6

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>

```

• Penjelasan program

Program di atas adalah contoh sederhana dalam bahasa pemrograman Java yang menggunakan perulangan **do-while** untuk mencetak angka dari 1 hingga N, di mana N adalah bilangan yang dimasukkan oleh pengguna. Berikut adalah penjelasan setiap bagian dari program tersebut:

1. Tujuan Program:

Tujuan utama program ini adalah untuk mendemonstrasikan penggunaan perulangan **do-while** dalam Java untuk mencetak angka dari 1 hingga N, di mana N adalah bilangan yang dimasukkan oleh pengguna.

2. Keyword, Tipe Data, dan Kelas yang Digunakan

- **import java.util.Scanner;**: Menggunakan keyword **import** untuk mengimpor kelas Scanner dari paket java.util. Kelas Scanner digunakan untuk membaca input dari pengguna.
- **public class PrintRepeat {**: Deklarasi kelas utama dengan nama **PrintRepeat**.
- **public static void main(String[] args) {**: Deklarasi metode **main()** sebagai titik masuk program. Menggunakan tipe data **void** untuk menunjukkan bahwa metode **main()** tidak mengembalikan nilai apa pun. Menggunakan parameter **String[] args** untuk menerima argumen baris perintah.
- **int N;**: Mendeklarasikan variabel **N** dengan tipe data **int** untuk menyimpan bilangan yang dimasukkan oleh pengguna.
- **int i;**: Mendeklarasikan variabel **i** dengan tipe data **int** untuk digunakan sebagai variabel iterasi dalam perulangan.
- **Scanner masukan = new Scanner(System.in);**: Membuat objek **Scanner** dengan nama **masukan** untuk membaca input dari pengguna melalui keyboard.
- **System.out.print("Nilai N >0 = ");**: Menampilkan pesan ke layar yang meminta pengguna untuk memasukkan nilai N.
- **N = masukan.nextInt();**: Membaca input bilangan N dari pengguna dan menyimpannya dalam variabel **N** menggunakan metode **nextInt()** dari objek **masukan**.
- **i = 1;**: Menginisialisasi variabel **i** dengan nilai 1.
- **System.out.print("Print i dengan REPEAT: \n");**: Menampilkan pesan ke layar yang memberikan informasi tentang metode pencetakan menggunakan perulangan **do-while**.
- **do { ... } while (i <= N);**: Memulai perulangan **do-while** yang akan terus berjalan selama nilai **i** kurang dari atau sama dengan nilai **N**. Setiap iterasi, nilai **i** akan dicetak ke layar dan kemudian variabel **i** akan bertambah satu.

3. Komponen Utama

- Penggunaan perulangan **do-while** untuk mencetak angka dari 1 hingga N.
- Penggunaan kelas Scanner untuk membaca input dari pengguna.
- Menampilkan pesan ke layar yang memberikan informasi tentang metode pencetakan menggunakan perulangan **do-while**.

- Menampilkan nilai variabel **i** ke layar pada setiap iterasi perulangan.
- Perulangan **do-while** akan terus berjalan selama kondisi **i <= N** terpenuhi.

Tujuan utama dari program ini adalah untuk memberikan pemahaman tentang cara menggunakan perulangan **do-while** dalam Java untuk mencetak angka dari 1 hingga N, di mana N adalah bilangan yang dimasukkan oleh pengguna. Program ini menunjukkan cara menggunakan perulangan **do-while** untuk melakukan iterasi dan mencetak nilai dari suatu rentang tertentu.

13. Print while

• Output

```
C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\SEMASTER 4\PBO\TUGAS 3\Sourcecode_13020220022>java PrintWhile
Nilai N >0 = 5
Print i dengan WHILE:
1
2
3
4
5

C:\SEMASTER 4\PBO\TUGAS 3\Sourcecode_13020220022>|
```

• Penjelasan program

Program di atas adalah contoh sederhana dalam bahasa pemrograman Java yang menggunakan perulangan **while** untuk mencetak angka dari 1 hingga N, di mana N adalah bilangan yang dimasukkan oleh pengguna. Berikut adalah penjelasan setiap bagian dari program tersebut:

1. Tujuan Program

Tujuan utama program ini adalah untuk mendemonstrasikan penggunaan perulangan **while** dalam Java untuk mencetak angka dari 1 hingga N, di mana N adalah bilangan yang dimasukkan oleh pengguna.

2. Keyword, Tipe Data, dan Kelas yang Digunakan

- **import java.util.Scanner;**: Menggunakan keyword **import** untuk mengimpor kelas Scanner dari paket java.util. Kelas Scanner digunakan untuk membaca input dari pengguna.
- **public class PrintWhile {**: Deklarasi kelas utama dengan nama **PrintWhile**.

- **public static void main(String[] args) {**: Deklarasi metode **main()** sebagai titik masuk program. Menggunakan tipe data **void** untuk menunjukkan bahwa metode **main()** tidak mengembalikan nilai apa pun. Menggunakan parameter **String[] args** untuk menerima argumen baris perintah.
- **int N;**: Mendeklarasikan variabel **N** dengan tipe data **int** untuk menyimpan bilangan yang dimasukkan oleh pengguna.
- **int i;**: Mendeklarasikan variabel **i** dengan tipe data **int** untuk digunakan sebagai variabel iterasi dalam perulangan.
- **Scanner masukan = new Scanner(System.in);**: Membuat objek **Scanner** dengan nama **masukan** untuk membaca input dari pengguna melalui keyboard.
- **System.out.print("Nilai N >0 = ");**: Menampilkan pesan ke layar yang meminta pengguna untuk memasukkan nilai **N**.
- **N = masukan.nextInt();**: Membaca input bilangan **N** dari pengguna dan menyimpannya dalam variabel **N** menggunakan metode **nextInt()** dari objek **masukan**.
- **i = 1;**: Menginisialisasi variabel **i** dengan nilai 1.
- **System.out.print("Print i dengan WHILE: \n");**: Menampilkan pesan ke layar yang memberikan informasi tentang metode pencetakan menggunakan perulangan **while**.
- **while (i <= N) { ... }**: Memulai perulangan **while** yang akan terus berjalan selama nilai **i** kurang dari atau sama dengan nilai **N**. Setiap iterasi, nilai **i** akan dicetak ke layar dan kemudian variabel **i** akan bertambah satu.

3. Komponen Utama

- Penggunaan perulangan **while** untuk mencetak angka dari 1 hingga **N**.
- Penggunaan kelas **Scanner** untuk membaca input dari pengguna.
- Menampilkan pesan ke layar yang memberikan informasi tentang metode pencetakan menggunakan perulangan **while**.
- Menampilkan nilai variabel **i** ke layar pada setiap iterasi perulangan.
- Perulangan **while** akan terus berjalan selama kondisi **i <= N** terpenuhi.

Tujuan utama dari program ini adalah untuk memberikan pemahaman tentang cara menggunakan perulangan **while** dalam Java untuk mencetak angka dari 1 hingga **N**, di mana **N** adalah bilangan yang dimasukkan oleh pengguna. Program ini menunjukkan cara menggunakan perulangan **while** untuk melakukan iterasi dan mencetak nilai dari suatu rentang tertentu.

14. Print While 1

- **Output**

```
C:\Windows\System32\cmd.e X + v
Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>java PrintWhile1
Nilai N >0 = 5
Print i dengan WHILE (ringkas):
1
2
3
4
5

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>
```

- **Penjelasan program**

Program di atas adalah contoh sederhana dalam bahasa pemrograman Java yang menggunakan perulangan **while** untuk mencetak angka dari 1 hingga N, di mana N adalah bilangan yang dimasukkan oleh pengguna. Berikut adalah penjelasan setiap bagian dari program tersebut:

1. Tujuan Program:

Tujuan utama program ini adalah untuk mendemonstrasikan penggunaan perulangan **while** dalam Java untuk mencetak angka dari 1 hingga N, di mana N adalah bilangan yang dimasukkan oleh pengguna.

2. Keyword, Tipe Data, dan Kelas yang Digunakan

- **import java.util.Scanner;**: Menggunakan keyword **import** untuk mengimpor kelas **Scanner** dari paket **java.util**. Kelas **Scanner** digunakan untuk membaca input dari pengguna.
- **public class PrintWhile1 {**: Deklarasi kelas utama dengan nama **PrintWhile1**.
- **public static void main(String[] args) {**: Deklarasi metode **main()** sebagai titik masuk program. Menggunakan tipe data **void** untuk menunjukkan bahwa metode **main()** tidak mengembalikan nilai apa pun. Menggunakan parameter **String[] args** untuk menerima argumen baris perintah.
- **int N;**: Mendeklarasikan variabel **N** dengan tipe data **int** untuk menyimpan bilangan yang dimasukkan oleh pengguna.
- **int i = 1;**: Mendeklarasikan variabel **i** dengan tipe data **int** dan langsung menginisialisasinya dengan nilai 1. Variabel **i** akan digunakan sebagai variabel iterasi dalam perulangan.

- **Scanner masukan = new Scanner(System.in);**: Membuat objek **Scanner** dengan nama **masukan** untuk membaca input dari pengguna melalui keyboard.
- **System.out.print("Nilai N >0 = ");**: Menampilkan pesan ke layar yang meminta pengguna untuk memasukkan nilai N.
- **N = masukan.nextInt();**: Membaca input bilangan N dari pengguna dan menyimpannya dalam variabel **N** menggunakan metode **nextInt()** dari objek **masukan**.
- **System.out.print("Print i dengan WHILE (ringkas): \n");**: Menampilkan pesan ke layar yang memberikan informasi tentang metode pencetakan menggunakan perulangan **while**.
- **while (i <= N) { ... }**: Memulai perulangan **while** yang akan terus berjalan selama nilai **i** kurang dari atau sama dengan nilai **N**. Setiap iterasi, nilai **i** akan dicetak ke layar dan kemudian variabel **i** akan bertambah satu menggunakan operator **++**.

3. Komponen Utama

- Penggunaan perulangan **while** untuk mencetak angka dari 1 hingga N.
- Penggunaan kelas **Scanner** untuk membaca input dari pengguna.
- Menampilkan pesan ke layar yang memberikan informasi tentang metode pencetakan menggunakan perulangan **while**.
- Menampilkan nilai variabel **i** ke layar pada setiap iterasi perulangan, dan pada saat yang sama, menambahkan nilai **i** dengan 1 menggunakan operator **++**.

Tujuan utama dari program ini adalah untuk memberikan pemahaman tentang cara menggunakan perulangan **while** dalam Java untuk mencetak angka dari 1 hingga N secara ringkas. Dengan menggunakan operator **++** pada saat pencetakan, program dapat mencetak nilai **i** dan kemudian menambahkannya secara bersamaan.

15. Print X iterasi

• Output

```
C:\Windows\System32\cmd.e  ×  +  v

Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>java PrintXiterasi
Masukkan nilai x (int), akhiri dg 999: 4
Masukkan nilai x (int),akhiri dg 999 : 3
Masukkan nilai x (int),akhiri dg 999 : 999
Hasil penjumlahan = 7

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>|
```

• Penjelasan program

Program di atas adalah contoh program dalam bahasa pemrograman Java yang meminta pengguna untuk memasukkan nilai-nilai bilangan bulat (integer) berulang kali, dan kemudian menghitung jumlah semua nilai yang dimasukkan kecuali nilai 999. Program akan terus meminta pengguna untuk memasukkan nilai hingga nilai yang dimasukkan adalah 999. Berikut adalah penjelasan setiap bagian dari program tersebut:

1. Tujuan Program

Tujuan utama program ini adalah untuk mendemonstrasikan cara menggunakan perulangan **for** dan struktur pengendali **break** dalam Java untuk meminta pengguna memasukkan nilai-nilai bilangan bulat berulang kali dan menghitung jumlahnya.

2. Keyword, Tipe Data, dan Kelas yang Digunakan

- **import java.util.Scanner;**: Menggunakan keyword **import** untuk mengimpor kelas **Scanner** dari paket **java.util**. Kelas **Scanner** digunakan untuk membaca input dari pengguna.
- **public class PrintXinterasi {**: Deklarasi kelas utama dengan nama **PrintXinterasi**.
- **public static void main(String[] args) {**: Deklarasi metode **main()** sebagai titik masuk program. Menggunakan parameter **String[] args** untuk menerima argumen baris perintah.

3. Variabel dan Objek

- **int Sum = 0;**: Mendeklarasikan variabel **Sum** yang bertipe data **int** dan menginisialisasinya dengan nilai 0. Variabel ini digunakan untuk menyimpan jumlah dari semua nilai yang dimasukkan pengguna.
- **int x;**: Mendeklarasikan variabel **x** yang bertipe data **int**. Variabel ini digunakan untuk menyimpan nilai yang dimasukkan pengguna.

4. Membaca Input dari Pengguna

- **Scanner masukan = new Scanner(System.in);**: Membuat objek **Scanner** dengan nama **masukan** untuk membaca input dari pengguna melalui keyboard.
- **x = masukan.nextInt();**: Meminta pengguna untuk memasukkan nilai **x** (integer) dan menyimpannya dalam variabel **x**.

5. Pengulangan dan Pengendalian Aliran

- **if (x == 999) { ... }**: Memeriksa apakah nilai yang dimasukkan pengguna adalah 999. Jika iya, maka program akan menampilkan

pesan "Kasus kosong" dan tidak akan melakukan penghitungan lebih lanjut.

- **else { ... }:** Jika nilai yang dimasukkan pengguna bukan 999, program akan menginisialisasi variabel **Sum** dengan nilai **x**, dan kemudian memulai perulangan tak terbatas (**for (;;) { ... }**) untuk meminta pengguna memasukkan nilai baru.
- **for (;;) { ... }:** Perulangan tak terbatas untuk meminta pengguna memasukkan nilai **x** baru hingga pengguna memasukkan nilai 999.
- **if (x == 999) break;** Jika nilai **x** yang dimasukkan adalah 999, program akan keluar dari perulangan menggunakan pernyataan **break**.
- **else { Sum = Sum + x; }:** Jika nilai **x** yang dimasukkan tidak sama dengan 999, nilai **x** akan ditambahkan ke variabel **Sum**.

6. Menampilkan Output

- **System.out.println("Hasil penjumlahan = " + Sum);:** Menampilkan hasil penjumlahan dari semua nilai yang dimasukkan pengguna (kecuali nilai 999) ke layar.

Tujuan utama dari program ini adalah untuk memberikan pemahaman tentang cara menggunakan perulangan for, struktur pengendali break, dan penggunaan kelas Scanner untuk membaca input dari pengguna dalam Java. Program ini memungkinkan pengguna memasukkan nilai-nilai berulang kali hingga pengguna memasukkan nilai 999, dan kemudian menghitung dan menampilkan jumlah dari semua nilai yang dimasukkan kecuali nilai 999.

16. Print X Repeat

• Output

```
C:\Windows\System32\cmd.e × + v
Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>java PrintXRepeat
Masukkan nilai x (int), akhiri dgn 999 : 1
Masukkan nilai x (int), akhiri dgn 999 : 2
Masukkan nilai x (int), akhiri dgn 999 : 999
Hasil penjumlahan = 3

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>
```

• Penjelasan program

Program di atas adalah contoh program dalam bahasa pemrograman Java yang meminta pengguna untuk memasukkan nilai-nilai bilangan bulat

(integer) berulang kali, dan kemudian menghitung jumlah semua nilai yang dimasukkan kecuali nilai 999. Program akan terus meminta pengguna untuk memasukkan nilai hingga nilai yang dimasukkan adalah 999. Berikut adalah penjelasan setiap bagian dari program tersebut:

1. Tujuan Program

Tujuan utama program ini adalah untuk mendemonstrasikan cara menggunakan perulangan **do-while** dalam Java untuk meminta pengguna memasukkan nilai-nilai bilangan bulat berulang kali dan menghitung jumlahnya.

2. Keyword, Tipe Data, dan Kelas yang Digunakan

- **import java.util.Scanner;**: Menggunakan keyword **import** untuk mengimpor kelas **Scanner** dari paket **java.util**. Kelas **Scanner** digunakan untuk membaca input dari pengguna.
- **public class PrintXRepeat {**: Deklarasi kelas utama dengan nama **PrintXRepeat**.
- **public static void main(String[] args) {**: Deklarasi metode **main()** sebagai titik masuk program. Menggunakan parameter **String[] args** untuk menerima argumen baris perintah.

3. Variabel dan Objek

- **int Sum;**: Mendeklarasikan variabel **Sum** yang bertipe data **int**. Variabel ini digunakan untuk menyimpan jumlah dari semua nilai yang dimasukkan pengguna (kecuali nilai 999).
- **int x;**: Mendeklarasikan variabel **x** yang bertipe data **int**. Variabel ini digunakan untuk menyimpan nilai yang dimasukkan pengguna.

4. Membaca Input dari Pengguna

- **Scanner masukan = new Scanner(System.in);**: Membuat objek **Scanner** dengan nama **masukan** untuk membaca input dari pengguna melalui keyboard.
- **x = masukan.nextInt();**: Meminta pengguna untuk memasukkan nilai **x** (integer) dan menyimpannya dalam variabel **x**.

5. Pengulangan dan Pengendalian Aliran

- **if (x == 999) { ... }**: Memeriksa apakah nilai yang dimasukkan pengguna adalah 999. Jika iya, maka program akan menampilkan pesan "Kasus kosong" dan tidak akan melakukan penghitungan lebih lanjut.

- **else { ... }**: Jika nilai yang dimasukkan pengguna bukan 999, program akan menginisialisasi variabel **Sum** dengan nilai 0, dan kemudian memulai perulangan **do-while**.
- **do { ... } while (x != 999);**: Perulangan **do-while** akan terus berjalan selama nilai **x** tidak sama dengan 999. Di dalam perulangan, nilai **x** akan ditambahkan ke variabel **Sum**, dan kemudian pengguna akan diminta untuk memasukkan nilai baru.
- **System.out.println("Hasil penjumlahan = " + Sum);**: Setelah pengguna memasukkan nilai 999 dan keluar dari perulangan, program akan menampilkan hasil penjumlahan dari semua nilai yang dimasukkan pengguna (kecuali nilai 999).

Tujuan utama dari program ini adalah untuk memberikan pemahaman tentang cara menggunakan perulangan **do-while** dalam Java untuk meminta pengguna memasukkan nilai-nilai berulang kali hingga pengguna memasukkan nilai 999, dan kemudian menghitung dan menampilkan jumlah dari semua nilai yang dimasukkan kecuali nilai 999. Program ini juga menggunakan kelas **Scanner** untuk membaca input dari pengguna.

17. Print X While

• Output

```

C:\Windows\System32\cmd.e  X  +  v
Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>java PrintXWhile
Masukkan nilai x (int), akhiri dg 999 : 2
Masukkan nilai x (int), akhiri dg 999 : 2
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 4

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>

```

• Penjelasan program

Program di atas adalah contoh program dalam bahasa pemrograman Java yang meminta pengguna untuk memasukkan nilai-nilai bilangan bulat (integer) berulang kali, dan kemudian menghitung jumlah semua nilai yang dimasukkan kecuali nilai 999. Program akan terus meminta pengguna untuk memasukkan nilai hingga nilai yang dimasukkan adalah 999. Berikut adalah penjelasan setiap bagian dari program tersebut:

1. Tujuan Program

Tujuan utama program ini adalah untuk mendemonstrasikan cara menggunakan perulangan **while** dalam Java untuk meminta pengguna

memasukkan nilai-nilai bilangan bulat berulang kali dan menghitung jumlahnya.

2. Keyword, Tipe Data, dan Kelas yang Digunakan

- **import java.util.Scanner;**: Menggunakan keyword **import** untuk mengimpor kelas **Scanner** dari paket **java.util**. Kelas **Scanner** digunakan untuk membaca input dari pengguna.
- **public class PrintXWhile {}**: Deklarasi kelas utama dengan nama **PrintXWhile**.
- **public static void main(String[] args) {}**: Deklarasi metode **main()** sebagai titik masuk program. Menggunakan parameter **String[] args** untuk menerima argumen baris perintah.

3. Variabel dan Objek

- **int Sum;**: Mendeklarasikan variabel **Sum** yang bertipe data **int**. Variabel ini digunakan untuk menyimpan jumlah dari semua nilai yang dimasukkan pengguna (kecuali nilai 999).
- **int x;**: Mendeklarasikan variabel **x** yang bertipe data **int**. Variabel ini digunakan untuk menyimpan nilai yang dimasukkan pengguna.

4. Membaca Input dari Pengguna

- **Scanner masukan = new Scanner(System.in);**: Membuat objek **Scanner** dengan nama **masukan** untuk membaca input dari pengguna melalui keyboard.
- **x = masukan.nextInt();**: Meminta pengguna untuk memasukkan nilai **x** (integer) dan menyimpannya dalam variabel **x**.

5. Pengulangan dan Pengendalian Aliran

- **while (x != 999) { ... }**: Perulangan **while** akan terus berjalan selama nilai **x** tidak sama dengan 999. Di dalam perulangan, nilai **x** akan ditambahkan ke variabel **Sum**, dan kemudian pengguna akan diminta untuk memasukkan nilai baru.
- **System.out.println("Hasil penjumlahan = " + Sum);**: Setelah pengguna memasukkan nilai 999 dan keluar dari perulangan, program akan menampilkan hasil penjumlahan dari semua nilai yang dimasukkan pengguna (kecuali nilai 999).

Tujuan utama dari program ini adalah untuk memberikan pemahaman tentang cara menggunakan perulangan **while** dalam Java untuk meminta pengguna memasukkan nilai-nilai berulang kali hingga pengguna memasukkan nilai 999, dan kemudian menghitung dan menampilkan jumlah

dari semua nilai yang dimasukkan kecuali nilai 999. Program ini juga menggunakan kelas Scanner untuk membaca input dari pengguna.

18. Sub Program

- **Output**

```
C:\Windows\System32\cmd.e  X  +  v
Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>java SubProgram
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
11 44
Ke dua bilangan : a = 11 b = 44
Maksimum = 44
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 44 b = 11

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>|
```

- **Penjelasan program**

Program di atas adalah contoh program dalam bahasa pemrograman Java yang menunjukkan penggunaan subprogram atau fungsi. Program ini memiliki dua subprogram yaitu **maxab** yang mengembalikan nilai maksimum dari dua bilangan yang diberikan, dan **tukar** yang menukar nilai dua variabel yang diberikan. Berikut adalah penjelasan setiap bagian dari program tersebut:

1. Tujuan Program

Tujuan utama program ini adalah untuk mendemonstrasikan penggunaan subprogram atau fungsi dalam bahasa pemrograman Java.

2. Keyword, Tipe Data, dan Kelas yang Digunakan

- **import java.util.Scanner;**: Menggunakan keyword **import** untuk mengimpor kelas Scanner dari paket java.util. Kelas Scanner digunakan untuk membaca input dari pengguna.
- **public class SubProgram {}**: Deklarasi kelas utama dengan nama **SubProgram**.
- **public static int maxab(int a, int b) { ... }**: Mendeklarasikan subprogram **maxab** yang mengembalikan nilai maksimum dari dua bilangan yang diberikan.
- **public static void tukar(int a, int b) { ... }**: Mendeklarasikan subprogram **tukar** yang menukar nilai dua variabel yang diberikan.

3. Subprogram atau Fungsi

- **public static int maxab(int a, int b) { ... }**: Subprogram **maxab** menerima dua parameter **a** dan **b** yang merupakan bilangan bulat

(integer). Subprogram ini mengembalikan nilai maksimum dari kedua bilangan menggunakan operator ternary.

- **public static void tukar(int a, int b) { ... }:** Subprogram **tukar** menerima dua parameter **a** dan **b** yang merupakan bilangan bulat (integer). Subprogram ini menukar nilai kedua variabel menggunakan variabel sementara **temp**, dan menampilkan nilai kedua variabel setelah pertukaran.

4. Membaca Input dari Pengguna

- **Scanner masukan = new Scanner(System.in);**: Membuat objek **Scanner** dengan nama **masukan** untuk membaca input dari pengguna melalui keyboard.
- **a = masukan.nextInt();**: Meminta pengguna untuk memasukkan nilai **a** (integer) dan menyimpannya dalam variabel **a**.
- **b = masukan.nextInt();**: Meminta pengguna untuk memasukkan nilai **b** (integer) dan menyimpannya dalam variabel **b**.

5. Panggilan Subprogram

- **System.out.println("Maksimum = " + (maxab(a, b)));**: Memanggil subprogram **maxab** untuk menampilkan nilai maksimum dari dua bilangan **a** dan **b**.
- **tukar(a, b);**: Memanggil subprogram **tukar** untuk menukar nilai kedua variabel **a** dan **b**.

Tujuan utama dari program ini adalah untuk memberikan pemahaman tentang cara menggunakan subprogram atau fungsi dalam bahasa pemrograman Java. Subprogram memungkinkan kita untuk memisahkan logika tertentu dalam program ke dalam unit-unit terpisah, sehingga meningkatkan keterbacaan dan modularitas kode.

19. Tempair

- **Output**

```
C:\Windows\System32\cmd.e  X  +  v
Microsoft Windows [Version 10.0.22631.3155]
(c) Microsoft Corporation. All rights reserved.

C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 4
Wujud air cair
4
C:\SEMESTER 4\PBO\TUGAS 3\Sourcecode_13020220022>
```


• Penjelasan program

Program di atas adalah contoh program dalam bahasa pemrograman Java yang menentukan wujud air berdasarkan suhu yang dimasukkan pengguna. Program ini menggunakan struktur pengkondisian **if-else if-else** untuk menentukan wujud air berdasarkan suhu yang dimasukkan pengguna. Berikut adalah penjelasan setiap bagian dari program tersebut:

1. Tujuan Program

Tujuan utama program ini adalah untuk mendemonstrasikan penggunaan struktur pengkondisian **if-else if-else** dalam Java untuk menentukan wujud air berdasarkan suhu yang dimasukkan pengguna.

2. Keyword, Tipe Data, dan Kelas yang Digunakan

- **import java.util.Scanner;**: Menggunakan keyword **import** untuk mengimpor kelas `Scanner` dari paket `java.util`. Kelas `Scanner` digunakan untuk membaca input dari pengguna.
- **public class Tempair {**: Deklarasi kelas utama dengan nama **Tempair**.
- **public static void main(String[] args) {**: Deklarasi metode **main()** sebagai titik masuk program. Menggunakan parameter **String[] args** untuk menerima argumen baris perintah.

3. Variabel dan Objek

- **int T;**: Mendeklarasikan variabel **T** yang bertipe data **int** untuk menyimpan suhu air yang dimasukkan pengguna.
- **Scanner masukan = new Scanner(System.in);**: Membuat objek **Scanner** dengan nama **masukan** untuk membaca input dari pengguna melalui keyboard.

4. Membaca Input dari Pengguna

- **T = masukan.nextInt();**: Meminta pengguna untuk memasukkan suhu air dalam derajat Celsius dan menyimpannya dalam variabel **T**.

5. Pengkondisian

- **if (T < 0) { ... }**: Memeriksa apakah suhu air kurang dari 0. Jika ya, maka program akan menampilkan pesan "Wujud air beku" dan nilai suhu.
- **else if ((0 <= T) && (T <= 100)) { ... }**: Memeriksa apakah suhu air antara 0 hingga 100. Jika ya, maka program akan menampilkan pesan "Wujud air cair" dan nilai suhu.

- **else if (T > 100) { ... }**: Memeriksa apakah suhu air lebih dari 100. Jika ya, maka program akan menampilkan pesan "Wujud air uap/gas" dan nilai suhu.
- **;**: Pernyataan kosong setelah blok **if-else if-else**.

Tujuan utama dari program ini adalah untuk memberikan pemahaman tentang cara menggunakan struktur pengkondisian **if-else if-else** dalam Java untuk menentukan wujud air berdasarkan suhu yang dimasukkan pengguna. Program ini membaca suhu dari pengguna, dan berdasarkan suhu tersebut, menampilkan pesan yang sesuai dengan wujud air pada suhu tersebut.

