

# Laporan Tugas 3(Praaktek)

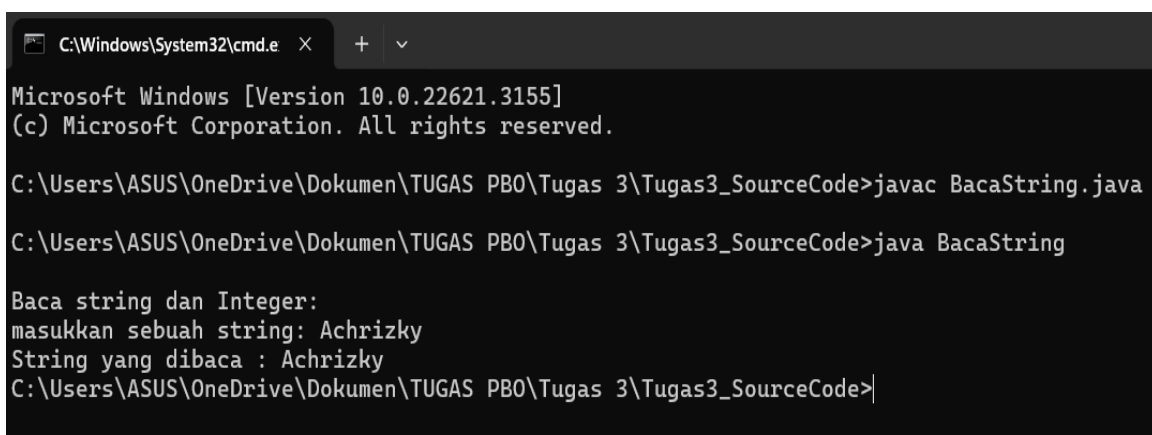
**Nama : Achmad Rizky Maulana**

**Nim : 13020220120**

**Kelas : A4 – Teknik Informatika**

**MatKul : PBO (Pemrograman Berorientasi Objek)**

## **1. Program BacaString**



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

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>javac BacaString.java
C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>java BacaString

Baca string dan Integer:
masukkan sebuah string: Achrizky
String yang dibaca : Achrizky
C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>
```

Penjelasan:

Tujuan dari program ini adalah contoh dari sebuah program yang membaca input string dari pengguna menggunakan kelas `BufferedReader` untuk membaca dari input stream standar

Berikut penjelasan singkat tentang setiap bagian programnya:

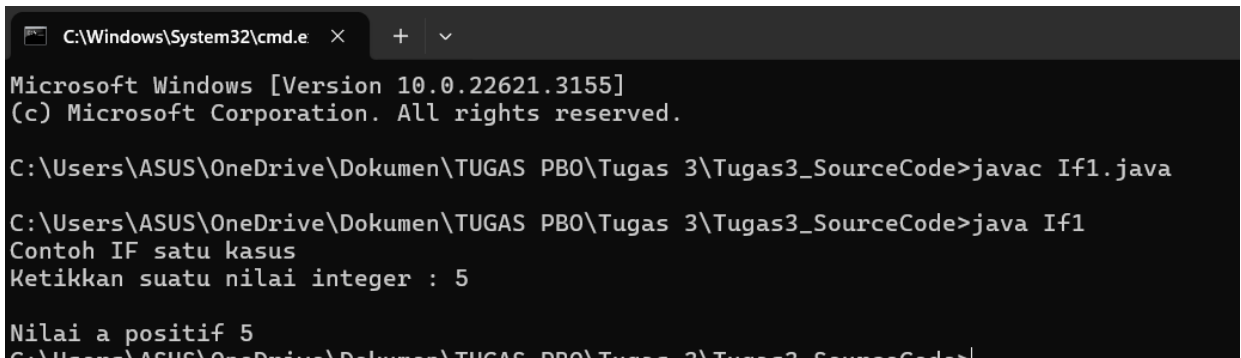
1. `BufferedReader;` : Baris ini mendeklarasikan beberapa kelas yang akan digunakan dalam program, termasuk `BufferedReader` untuk membaca input, `IOException` untuk menangani kesalahan input/output, `InputStreamReader` untuk membaca input stream, dan `JOptionPane` untuk menampilkan dialog pesan.
2. `public class BacaString {` : Deklarasi kelas `BacaString`.
3. `public static void main(String[] args) throws IOException {` : merupakan titik masuk eksekusi program. `IOException` dilemparkan karena kita menggunakan `BufferedReader` yang dapat menimbulkan pengecualian input/output.

- ## 2. Program ForEver

Penjelasan :  
Program yang diberikan adalah contoh dari sebuah program yang menggunakan sebuah loop tak terbatas (infinite loop) dalam bahasa pemrograman Java. Loop tak terbatas adalah sebuah struktur kontrol di mana blok kode di dalamnya akan terus dieksekusi selama kondisi yang diberikan selalu bernilai benar.

1. `public static void main(String[] args) {` : Metode main yang merupakan titik masuk eksekusi program.
2. `System.out.println("Program akan looping, akhiri dengan ^c");` : Pesan permintaan untuk memberi tahu pengguna bahwa program akan terus melakukan loop tak terbatas sampai dihentikan secara paksa dengan menggunakan kombinasi tombol `^c` (biasanya digunakan di terminal/command prompt untuk menghentikan eksekusi program).
3. `while (true) { System.out.print ("Print satu baris ....\n"); }` : Loop while yang tak pernah berakhir (`while (true)`) akan terus mencetak pesan "Print satu baris ...." ke layar. Karena kondisinya selalu benar (`true`), loop ini akan berjalan tanpa henti sampai program dihentikan secara paksa.

### 3. Program IF1



```

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

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>javac If1.java

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : 5

Nilai a positif 5

```

Penjelasan :

Program Java di atas adalah contoh penggunaan struktur kontrol IF dalam satu kasus. Berikut adalah penjelasan singkat tentang setiap bagian programnya.

Berikut adalah penjelasan singkat tentang setiap bagian programnya:

1. `import java.util.Scanner;` : Baris-baris ini adalah komentar yang memberikan deskripsi tentang program.
2. `public static void main(String[] args) {` : Metode main yang merupakan titik masuk eksekusi program.
3. `Scanner masukan=new Scanner(System.in);` : Membuat objek Scanner untuk membaca input dari pengguna. Variabel **a** dideklarasikan untuk menyimpan nilai integer yang akan dimasukkan.
4. `System.out.print ("Ketikkan suatu nilai integer : ");` : Pesan permintaan untuk pengguna untuk memasukkan nilai integer ditampilkan. Nilai tersebut kemudian dibaca menggunakan `nextInt()` dari objek Scanner dan disimpan dalam variabel **a**.
5. `if (a >= 0) System.out.print ("\nNilai a positif "+ a);` : Kondisi **if** digunakan untuk memeriksa apakah nilai **a** lebih besar dari atau sama dengan 0. Jika kondisi ini terpenuhi, maka pesan "Nilai a positif" disertai dengan nilai **a** akan dicetak ke layar.

## 4. Program IF2

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

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>javac If2.java

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>java If2
Contoh IF dua kasus
Ketikkan suatu nilai integer :14
Nilai a positif 14

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>|
```

Penjelasan :

Program tersebut adalah contoh penggunaan struktur kontrol IF dalam dua kasus komplementer. Program ini meminta pengguna untuk memasukkan sebuah nilai integer, lalu mencetak pesan sesuai dengan nilai yang dimasukkan.

1. Impor kelas Scanner dari paket java.util untuk membaca input dari pengguna.
2. Deklarasikan kelas If2.
3. Metode main() adalah titik masuk eksekusi program.
4. Deklarasikan variabel integer 'a' untuk menyimpan nilai yang dimasukkan oleh pengguna. Buat objek Scanner untuk membaca input dari pengguna.
5. Program mencetak pesan permintaan untuk pengguna untuk memasukkan nilai integer.
6. Nilai yang dimasukkan oleh pengguna dibaca menggunakan metode nextInt() dari objek Scanner dan disimpan dalam variabel 'a'.
7. Program melakukan pemeriksaan dengan menggunakan struktur kontrol IF. Jika nilai 'a' lebih besar dari atau sama dengan 0, maka program mencetak pesan "Nilai a positif" disertai dengan nilai 'a'. Jika nilai 'a' kurang dari 0, maka program mencetak pesan "Nilai a negatif" disertai dengan nilai 'a'.
8. Program selesai.

## 5. Program If3

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

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>javac If3.java

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer :100
Nilai a positif 100

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>|
```

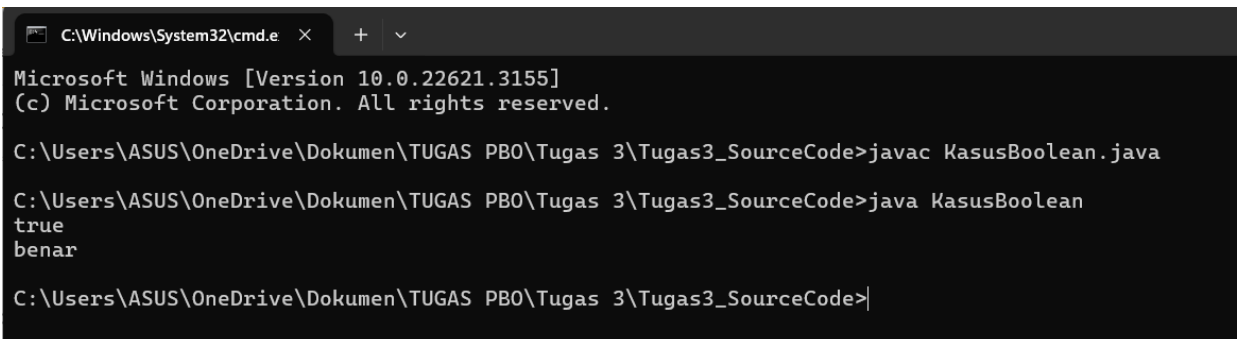
Penjelasan :

Program berikut adalah contoh penggunaan struktur kontrol IF dalam tiga kasus. Program ini meminta pengguna untuk memasukkan sebuah nilai integer, lalu mencetak pesan sesuai dengan nilai yang dimasukkan.

Berikut adalah penjelasan dari setiap bagian programnya :

1. Baris 1-2: Import kelas Scanner dari paket java.util untuk membaca input dari pengguna.
2. Komentar di baris 3-6 memberikan deskripsi tentang program dan kelas If3.
3. Metode main() (baris 9-26) adalah titik masuk eksekusi program.
4. Deklarasikan variabel integer 'a' untuk menyimpan nilai yang dimasukkan oleh pengguna. Objek Scanner dibuat untuk membaca input dari pengguna.
5. Program mencetak pesan permintaan untuk pengguna untuk memasukkan nilai integer.
6. Nilai yang dimasukkan oleh pengguna dibaca menggunakan metode nextInt() dari objek Scanner dan disimpan dalam variabel 'a'.
7. Program melakukan pemeriksaan dengan menggunakan struktur kontrol IF dengan tiga kasus. Jika nilai 'a' lebih besar dari 0, maka program mencetak pesan "Nilai a positif" disertai dengan nilai 'a'. Jika nilai 'a' sama dengan 0, maka program mencetak pesan "Nilai Nol" disertai dengan nilai 'a'. Jika kedua kondisi tersebut tidak terpenuhi, maka program mencetak pesan "Nilai a negatif" disertai dengan nilai 'a'.

## 6. Program KasusBoolean



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

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PB0\Tugas 3\Tugas3_SourceCode>javac KasusBoolean.java

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PB0\Tugas 3\Tugas3_SourceCode>java KasusBoolean
true
benar

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PB0\Tugas 3\Tugas3_SourceCode>|
```

Penjelasan :

penggunaan ekspresi kondisional dengan tipe data boolean. Program ini menciptakan variabel boolean 'bool' dan menetapkan nilainya menjadi true. Kemudian, program menggunakan struktur kontrol IF untuk membuat keputusan berdasarkan nilai boolean 'bool'.

Berikut penjelasan singkat dari setiap bagian program:

1. Variabel boolean 'bool' dideklarasikan untuk menyimpan nilai boolean.
2. Variabel 'bool' diinisialisasi dengan nilai true.
3. Program menggunakan struktur kontrol IF untuk melakukan percabangan berdasarkan nilai boolean 'bool':
  - Jika nilai 'bool' adalah true, maka pesan "true" akan dicetak ke layar.
  - Jika nilai 'bool' adalah false, maka pesan "false" akan dicetak ke layar.
4. Program kemudian menggunakan struktur kontrol IF lagi untuk memeriksa negasi dari nilai 'bool':
  - Jika nilai 'bool' adalah false, maka pesan "salah" akan dicetak ke layar karena negasi dari false adalah true.
  - Jika nilai 'bool' adalah true, maka pesan "benar" akan dicetak ke layar karena negasi dari true adalah false.

## 7. Program KasusSwitch

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

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>javac KasusSwitch.java

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
b
Yang anda ketik adalah huruf mati

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>|
```

Penjelas :

Program Java di atas adalah contoh penggunaan struktur kontrol switch-case untuk mengevaluasi nilai dari suatu variabel dan menjalankan blok kode sesuai dengan nilai tersebut. Program ini meminta pengguna untuk memasukkan sebuah huruf, kemudian mencetak pesan yang sesuai dengan huruf yang dimasukkan.

Berikut adalah penjelasan singkat tentang setiap bagian programnya:

1. Deklarasi kelas **KasusSwitch** dan metode **main** sebagai titik masuk eksekusi program.
2. Variabel **cc** dideklarasikan sebagai tipe data **char** untuk menyimpan huruf yang dimasukkan oleh pengguna. Objek **Scanner** dibuat untuk membaca input dari pengguna.
3. Program mencetak pesan permintaan untuk pengguna untuk memasukkan sebuah huruf.
4. Nilai yang dimasukkan oleh pengguna dibaca menggunakan metode **next().charAt(0)** dari objek Scanner. Metode **next()** digunakan untuk membaca sebuah token (sebuah kata atau huruf), kemudian **charAt(0)** digunakan untuk mengambil karakter pertama dari token tersebut, karena kita hanya ingin membaca satu huruf. Karakter tersebut disimpan dalam variabel **cc**.
5. Struktur kontrol switch-case digunakan untuk mengevaluasi nilai dari variabel **cc**:
  - Jika nilai **cc** adalah 'a', maka program mencetak pesan "Yang anda ketik adalah a".
  - Jika nilai **cc** adalah 'u', maka program mencetak pesan "Yang anda ketik adalah u".
  - Jika nilai **cc** adalah 'e', maka program mencetak pesan "Yang anda ketik adalah e".
  - Jika nilai **cc** adalah 'i', maka program mencetak pesan "Yang anda ketik adalah i".
  - Jika nilai **cc** adalah 'o', maka program mencetak pesan "Yang anda ketik adalah o".
  - Jika nilai **cc** tidak cocok dengan kasus-kasus di atas, maka blok default akan dijalankan, dan program mencetak pesan "Yang anda ketik adalah huruf mati".
6. Program selesai.

## 8. Program Konstant

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

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>javac Konstant.java

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>java Konstant
Jari-jari lingkaran =20
Luas lingkaran = 1256.6001
Akhir program

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>|
```

Penjelasan :

Program Java di atas adalah contoh penggunaan konstanta dalam perhitungan luas lingkaran. Konstanta **PHI** digunakan untuk menyimpan nilai  $\pi$  (pi) yang merupakan angka konstan dalam matematika. Program ini meminta pengguna untuk memasukkan jari-jari lingkaran, kemudian menghitung dan mencetak luas lingkaran berdasarkan nilai jari-jari yang dimasukkan.

Berikut adalah penjelasan singkat tentang setiap bagian programnya:

1. Variabel **PHI** dideklarasikan sebagai konstanta dengan kata kunci **final**, yang berarti nilai konstan ini tidak dapat diubah selama program berjalan. Nilai **PHI** diinisialisasi dengan nilai  $\pi$  (pi) yang didefinisikan sebagai 3.1415.
2. Variabel **r** dideklarasikan sebagai tipe data **float** untuk menyimpan jari-jari lingkaran. Objek **Scanner** dibuat untuk membaca input dari pengguna.
3. Program mencetak pesan permintaan untuk pengguna untuk memasukkan nilai jari-jari lingkaran.
4. Nilai yang dimasukkan oleh pengguna dibaca menggunakan metode **nextFloat()** dari objek **Scanner** dan disimpan dalam variabel **r**.
5. Program menghitung luas lingkaran dengan rumus  $\pi * r^2$  dan mencetak hasilnya.
6. Program mencetak pesan "Akhir program".

## 9. Program Max2

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

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>javac Max2.java

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
15
16
Ke dua bilangan : a = 15 b = 16
Nilai b yang maksimum: 16

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>|
```

Penjelasan :

Program Java di atas adalah contoh implementasi untuk menentukan nilai maksimum dari dua bilangan yang dimasukkan oleh pengguna.

Berikut adalah penjelasan singkat tentang setiap bagian programnya:

1. Deklarasi kelas **Max2** dan metode **main** sebagai titik masuk eksekusi program.
2. Variabel **a** dan **b** dideklarasikan untuk menyimpan dua bilangan yang dimasukkan oleh pengguna. Objek **Scanner** dibuat untuk membaca input dari pengguna.
3. Program mencetak pesan permintaan kepada pengguna untuk memasukkan dua bilangan, dipisahkan oleh RETURN.
4. Nilai bilangan pertama yang dimasukkan oleh pengguna dibaca menggunakan metode **nextInt()** dari objek **Scanner** dan disimpan dalam variabel **a**. Kemudian, nilai bilangan kedua dibaca dan disimpan dalam variabel **b**.
5. Program mencetak kedua bilangan yang dimasukkan oleh pengguna.
6. Struktur kontrol IF digunakan untuk membandingkan nilai **a** dan **b**. Jika nilai **a** lebih besar dari atau sama dengan nilai **b**, maka program mencetak pesan "Nilai a yang maksimum" beserta nilai **a**. Jika nilai **b** lebih besar dari **a**, maka program mencetak

pesan "Nilai b yang maksimum" beserta nilai **b**.

## 10. Program PriFor

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

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>javac PriFor.java

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>java PriFor
Baca N, print 1 s/d N N = 11
1
2
3
4
5
6
7
8
9
10
11
Akhir program

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>|
```

Penjelasan :

Program di atas adalah contoh implementasi untuk membaca sebuah nilai N dari pengguna dan mencetak angka dari 1 hingga N menggunakan perulangan for.

Berikut adalah penjelasan singkat tentang setiap bagian programnya:

1. Deklarasi kelas **PriFor** dan metode **main** sebagai titik masuk eksekusi program.
2. Variabel **i** dan **N** dideklarasikan untuk digunakan dalam program. Objek **Scanner** dibuat untuk membaca input dari pengguna.
3. Program mencetak pesan permintaan kepada pengguna untuk memasukkan nilai N.
4. Nilai N yang dimasukkan oleh pengguna dibaca menggunakan metode **nextInt()** dari objek **Scanner** dan disimpan dalam variabel **N**.
5. Program memulai perulangan **for**, dimulai dari nilai **i** sama dengan 1 hingga **N**. Setiap iterasi, nilai **i** akan bertambah satu.
6. Di dalam perulangan, program mencetak nilai **i** ke layar menggunakan metode **println()**.
7. Setelah perulangan selesai, program mencetak pesan "Akhir program" sebagai penanda akhir program.



## 11. Program PrintIterasi

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

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>javac PrintIterasi.java

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>java PrintIterasi
Nilai N >0 = 5
Print i dengan ITERATE :
1
2
3
4
5

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>
```

Penjelasan :

Program Java di atas adalah contoh implementasi untuk membaca sebuah nilai N dari pengguna dan mencetak angka dari 1 hingga N menggunakan perulangan iteratif dengan menggunakan instruksi break untuk menghentikan perulangan.

Berikut adalah penjelasan singkat tentang setiap bagian programnya:

1. Deklarasi kelas **PrintIterasi** dan metode **main** sebagai titik masuk eksekusi program.
2. Variabel **N** dan **i** dideklarasikan untuk digunakan dalam program. Objek **Scanner** dibuat untuk membaca input dari pengguna.
3. Program mencetak pesan permintaan kepada pengguna untuk memasukkan nilai N.
4. Nilai N yang dimasukkan oleh pengguna dibaca menggunakan metode **nextInt()** dari objek **Scanner** dan disimpan dalam variabel N.
5. Variabel **i** diinisialisasi dengan nilai 1 sebagai nilai awal.
6. Program mencetak pesan "Print i dengan ITERATE :".
7. Perulangan **for** dibuat tanpa kondisi awal, kondisi akhir, atau ekspresi iterasi. Ini membuat perulangan tidak terbatas dan akan terus berlanjut hingga instruksi **break** dieksekusi.
8. Di dalam perulangan, program mencetak nilai **i** ke layar.
9. Kemudian, program memeriksa apakah nilai **i** sama dengan nilai N. Jika kondisi ini terpenuhi, maka perulangan dihentikan dengan menggunakan instruksi **break**.
10. Jika kondisi **i == N** tidak terpenuhi, maka nilai **i** akan ditambahkan satu untuk mendapatkan nilai berikutnya.
11. Program selesai.

## 12. Program PrintRepeat

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

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>javac PrintRepeat.java

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>java PrintRepeat
Nilai N >0 = 6
Print i dengan REPEAT:
1
2
3
4
5
6

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>|
```

Penjelasan :

Program Java di atas adalah contoh implementasi untuk membaca sebuah nilai N dari pengguna dan mencetak angka dari 1 hingga N menggunakan perulangan do-while.

Berikut adalah penjelasan singkat tentang setiap bagian programnya:

1. Deklarasi kelas **PrintRepeat** dan metode **main** sebagai titik masuk eksekusi program.
2. Variabel **N** dan **i** dideklarasikan untuk digunakan dalam program. Objek **Scanner** dibuat untuk membaca input dari pengguna.
3. Program mencetak pesan permintaan kepada pengguna untuk memasukkan nilai N.
4. Nilai N yang dimasukkan oleh pengguna dibaca menggunakan metode **nextInt()** dari objek **Scanner** dan disimpan dalam variabel **N**.
5. Variabel **i** diinisialisasi dengan nilai 1 sebagai nilai awal.
6. Program mencetak pesan "Print i dengan REPEAT :".
7. Perulangan **do-while** dibuat dengan melakukan pencetakan nilai **i** ke layar, kemudian nilai **i** ditambahkan satu. Perulangan ini akan terus berlanjut selama nilai **i** kurang dari atau sama dengan nilai **N**.
8. Jika nilai **i** melebihi nilai **N**, maka perulangan akan dihentikan.
9. Program selesai.

## 13. Program PrintWhile

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

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>javac PrintWhile.java

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>java PrintWhile
Nilai N >0 = 7
Print i dengan WHILE:
1
2
3
4
5
6
7

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>|
```

Penjelasan :

Program Java di atas adalah contoh implementasi untuk membaca sebuah nilai N dari pengguna dan mencetak angka dari 1 hingga N menggunakan perulangan while.

Berikut adalah penjelasan singkat tentang setiap bagian programnya:

1. Deklarasi kelas **PrintWhile** dan metode **main** sebagai titik masuk eksekusi program.
2. Variabel **N** dan **i** dideklarasikan untuk digunakan dalam program. Objek **Scanner** dibuat untuk membaca input dari pengguna.
3. Program mencetak pesan permintaan kepada pengguna untuk memasukkan nilai N.
4. Nilai N yang dimasukkan oleh pengguna dibaca menggunakan metode **nextInt()** dari objek **Scanner** dan disimpan dalam variabel **N**.
5. Variabel **i** diinisialisasi dengan nilai 1 sebagai nilai awal.
6. Program mencetak pesan "Print i dengan WHILE :".
7. Perulangan **while** dibuat dengan kondisi bahwa nilai **i** kurang dari atau sama dengan nilai **N**. Perulangan ini akan terus berlanjut selama kondisi terpenuhi.
8. Di dalam perulangan, program mencetak nilai **i** ke layar menggunakan metode **println()**, kemudian nilai **i** ditambahkan satu sebagai langkah iterasi berikutnya.
9. Perulangan berakhir ketika nilai **i** melebihi nilai **N**.
10. Program selesai.

## 14. Program PrintWhile

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

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>javac PrintWhile.java

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>java PrintWhile
Nilai N >0 = 9
Print i dengan WHILE:
1
2
3
4
5
6
7
8
9

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>|
```

Penjelasan :

Program di atas adalah implementasi ringkas untuk membaca sebuah nilai N dari pengguna dan mencetak angka dari 1 hingga N menggunakan perulangan while.

Berikut adalah penjelasan singkat tentang setiap bagian programnya:

1. Deklarasi kelas **PrintWhile1** dan metode **main** sebagai titik masuk eksekusi program. Variabel **N** dideklarasikan untuk menyimpan nilai yang akan dimasukkan oleh pengguna. Variabel **i** diinisialisasi langsung dengan nilai 1.
2. Objek **Scanner** dibuat untuk membaca input dari pengguna.
3. Program mencetak pesan permintaan kepada pengguna untuk memasukkan nilai N.
4. Nilai N yang dimasukkan oleh pengguna dibaca menggunakan metode **nextInt()** dari objek **Scanner** dan disimpan dalam variabel **N**.
5. Program mencetak pesan "Print i dengan WHILE (ringkas):".
6. Perulangan **while** dibuat dengan kondisi bahwa nilai **i** kurang dari atau sama dengan nilai **N**. Perulangan ini akan terus berjalan selama kondisi terpenuhi.
7. Di dalam perulangan, program mencetak nilai **i** ke layar menggunakan metode **println()**, dan nilai **i** ditambahkan satu secara bersamaan menggunakan operator **++**.
8. Perulangan berakhir ketika nilai **i** melebihi nilai **N**.
9. Program selesai.

## 15. Program PrintWhile1

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

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>javac PrintWhile1.java

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>java PrintWhile1
Nilai N >0 = 3
Print i dengan WHILE (ringkas):
1
2
3

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>|
```

Penjelasan :

Program di atas adalah program sederhana yang ditulis dalam bahasa pemrograman Java untuk mencetak angka dari 1 hingga N menggunakan struktur perulangan while. Berikut penjelasan lebih detailnya:

1. **import java.util.Scanner;**: Mendeklarasikan penggunaan paket atau class Scanner yang berada dalam paket java.util. Ini memungkinkan program untuk membaca input dari pengguna.
2. **public class PrintWhile1 { ... }**: Mendefinisikan sebuah class dengan nama PrintWhile1.
3. **public static void main(String[] args) { ... }**: Metode utama (main method) dari program Java. Eksekusi program dimulai dari sini.
4. **int N;**: Deklarasi variabel N dengan tipe data integer yang akan digunakan untuk menyimpan nilai yang dimasukkan oleh pengguna.
5. **int i = 1;**: Inisialisasi variabel i dengan nilai 1. Variabel i akan digunakan untuk melakukan iterasi dari 1 hingga N.
6. **Scanner masukan=new Scanner(System.in);**: Membuat objek Scanner bernama masukan untuk menerima input dari pengguna.
7. **System.out.print ("Nilai N >0 = ");**: Mencetak pesan untuk meminta pengguna memasukkan nilai N.
8. **N = masukan.nextInt();**: Membaca nilai N yang dimasukkan oleh pengguna dan menyimpannya dalam variabel N.
9. **System.out.print ("Print i dengan WHILE (ringkas): \n");**: Mencetak pesan sebelum memulai proses pencetakan nilai i.
10. **while (i <= N){ ... }**: Memulai perulangan while dengan kondisi bahwa nilai i kurang dari atau sama dengan N.
11. **System.out.println (i++);**: Mencetak nilai i ke layar dan kemudian meningkatkan nilainya dengan 1 (i++). Ini akan mencetak nilai i dari 1 hingga N satu per satu di setiap iterasi.
12. **}**: Menutup blok while loop.

## 16. Program PrintXrpeat

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

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PB0\Tugas 3\Tugas3_SourceCode>javac PrintXRepeat.java

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PB0\Tugas 3\Tugas3_SourceCode>java PrintXRepeat
Masukkan nilai x (int), akhiri dg 999 : 7
Masukkan nilai x (int), akhiri dg 999 : 6
Masukkan nilai x (int), akhiri dg 999 : 5
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 18

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PB0\Tugas 3\Tugas3_SourceCode>|
```

Penjelasan :

Program tersebut merupakan contoh program dalam bahasa pemrograman Java yang membaca nilai-nilai x dari pengguna dan menjumlahkannya menggunakan perulangan do-while.

Program akan terus meminta pengguna memasukkan nilai x sampai pengguna memasukkan nilai 999 sebagai penanda akhir. Setelah itu, program akan mencetak hasil penjumlahan dari semua nilai x yang dimasukkan pengguna.

Berikut adalah penjelasan singkat mengenai program tersebut:

1. **import java.util.Scanner;**: Mendeklarasikan penggunaan paket atau class Scanner yang berada dalam paket java.util. Ini memungkinkan program untuk membaca input dari pengguna.
2. **public class PrintXRepeat { ... }**: Mendefinisikan sebuah class dengan nama PrintXRepeat.
3. **public static void main(String[] args) { ... }**: Metode utama (main method) dari program Java. Eksekusi program dimulai dari sini.
4. **int Sum;**: Deklarasi variabel Sum dengan tipe data integer yang akan digunakan untuk menyimpan hasil penjumlahan nilai-nilai x.
5. **int x;**: Deklarasi variabel x dengan tipe data integer yang akan digunakan untuk menyimpan nilai yang dimasukkan oleh pengguna.
6. **Scanner masukan=new Scanner(System.in);**: Membuat objek Scanner bernama masukan untuk menerima input dari pengguna.
7. **System.out.print ("Masukkan nilai x (int), akhiri dg 999 : ");**: Mencetak pesan untuk meminta pengguna memasukkan nilai x. Pengguna diminta untuk mengakhiri masukan dengan memasukkan nilai 999.
8. **x = masukan.nextInt();**: Membaca nilai x yang dimasukkan oleh pengguna dan menyimpannya dalam variabel x. Ini adalah elemen pertama yang akan dimasukkan ke dalam penjumlahan.
9. **if (x == 999){ ... }else { ... }**: Melakukan pengecekan apakah pengguna langsung memasukkan nilai 999. Jika iya, maka program akan mencetak pesan bahwa kasusnya kosong. Jika tidak, program akan melanjutkan untuk menjumlahkan nilai-nilai x yang dimasukkan oleh pengguna.
10. **Sum = 0;**: Menginisialisasi variabel Sum dengan nilai 0. Ini dilakukan sebelum memulai proses penjumlahan.
11. **do { ... } while (x != 999);**: Memulai perulangan do-while. Program akan terus melakukan proses penjumlahan dan meminta pengguna memasukkan nilai x hingga pengguna memasukkan nilai 999.
12. **Sum = Sum + x;**: Menambahkan nilai x yang dimasukkan pengguna ke dalam nilai Sum.
13. **System.out.println ("Hasil penjumlahan = "+Sum);**: Mencetak hasil penjumlahan dari semua nilai x yang dimasukkan pengguna.

## 17. Program PrintXWhile

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

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>javac PrintXWhile.java

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>java PrintXWhile
Masukkan nilai x (int), akhiri dg 999 : 5
Masukkan nilai x (int), akhiri dg 999 : 6
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 11

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>|
```

Penjelasan :

Program di atas adalah contoh program dalam bahasa pemrograman Java yang membaca nilai-nilai x dari pengguna dan menjumlahkannya menggunakan perulangan while.

Program akan terus meminta pengguna memasukkan nilai x sampai pengguna memasukkan nilai 999 sebagai penanda akhir.

Berikut adalah penjelasan singkat mengenai program tersebut:

1. **import java.util.Scanner;**: Mendeklarasikan penggunaan paket atau class Scanner yang berada dalam paket java.util. Ini memungkinkan program untuk membaca input dari pengguna.
2. **public class PrintXWhile { ... }**: Mendefinisikan sebuah class dengan nama PrintXWhile.
3. **public static void main(String[] args) { ... }**: Metode utama (main method) dari program Java. Eksekusi program dimulai dari sini.
4. **int Sum;**: Deklarasi variabel Sum dengan tipe data integer yang akan digunakan untuk menyimpan hasil penjumlahan nilai-nilai x.
5. **int x;**: Deklarasi variabel x dengan tipe data integer yang akan digunakan untuk menyimpan nilai yang dimasukkan oleh pengguna.
6. **Scanner masukan=new Scanner(System.in);**: Membuat objek Scanner bernama masukan untuk menerima input dari pengguna.
7. **Sum = 0;**: Menginisialisasi variabel Sum dengan nilai 0. Ini dilakukan sebelum memulai proses penjumlahan.
8. **System.out.print ("Masukkan nilai x (int), akhiri dg 999 : ");**: Mencetak pesan untuk meminta pengguna memasukkan nilai x. Pengguna diminta untuk mengakhiri masukan dengan memasukkan nilai 999.
9. **x = masukan.nextInt();**: Membaca nilai x yang dimasukkan oleh pengguna dan menyimpannya dalam variabel x. Ini adalah elemen pertama yang akan dimasukkan ke dalam penjumlahan.
10. **while (x != 999) { ... }**: Memulai perulangan while. Program akan terus melakukan proses penjumlahan dan meminta pengguna memasukkan nilai x hingga pengguna memasukkan nilai 999.
11. **Sum = Sum + x;**: Menambahkan nilai x yang dimasukkan pengguna ke dalam nilai Sum.
12. **System.out.print("Masukkan nilai x (int), akhiri dg 999 : ");**: Mencetak pesan untuk meminta pengguna memasukkan nilai x lagi setelah satu iterasi telah selesai.
13. **System.out.println("Hasil penjumlahan = "+ Sum);**: Mencetak hasil penjumlahan dari semua nilai x yang dimasukkan pengguna setelah pengguna memasukkan nilai 999.

## 18. Program SubProgram

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

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>javac SubProgram.java

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>java SubProgram
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
2
4
Ke dua bilangan : a = 2 b = 4
Maksimum = 4
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 4 b = 2

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PBO\Tugas 3\Tugas3_SourceCode>|
```

Penjelasan :

Program di atas adalah contoh program dalam bahasa pemrograman Java yang mengandung prosedur dan fungsi. Program ini memiliki dua subprogram: **maxab** dan **tukar**.

Berikut adalah penjelasan singkat mengenai program tersebut:

1. **public static int maxab (int a, int b) { ... }**: Mendefinisikan fungsi **maxab** yang mengambil dua parameter bertipe integer **a** dan **b**, dan mengembalikan nilai maksimum dari kedua bilangan tersebut.
2. **return ((a >= b) ? a : b);**: Dalam fungsi **maxab**, menggunakan operator ternary untuk menentukan nilai maksimum antara **a** dan **b**. Jika **a** lebih besar dari atau sama dengan **b**, maka fungsi mengembalikan nilai **a**, jika tidak, maka fungsi mengembalikan nilai **b**.
3. **public static void tukar (int a, int b) { ... }**: Mendefinisikan prosedur **tukar** yang mengambil dua parameter bertipe integer **a** dan **b**, dan bertujuan untuk menukar nilai dari kedua parameter tersebut.
4. **int temp; temp = a; a = b; b = temp;**: Dalam prosedur **tukar**, nilai dari **a** disimpan dalam variabel sementara **temp**, kemudian nilai **b** disimpan dalam variabel **a**, dan nilai **temp** (yang merupakan nilai awal dari **a**) disimpan dalam variabel **b**. Dengan demikian, nilai **a** dan **b** telah ditukar.
5. **public static void main(String[] args) { ... }**: Metode utama (main method) dari program Java. Eksekusi program dimulai dari sini.
6. **int a, b;**: Deklarasi variabel **a** dan **b** yang akan digunakan untuk menyimpan dua bilangan yang dimasukkan oleh pengguna.
7. **Scanner masukan=new Scanner(System.in);**: Membuat objek Scanner bernama **masukan** untuk menerima input dari pengguna.
8. **a = masukan.nextInt(); b = masukan.nextInt();**: Meminta pengguna untuk memasukkan dua bilangan secara berturut-turut dan menyimpannya dalam variabel **a** dan **b**.
9. **System.out.println ("Ke dua bilangan : a = " + a + " b = " + b);**: Mencetak nilai dari kedua bilangan yang dimasukkan oleh pengguna.
10. **System.out.println ("Maksimum = " + (maxab(a, b)));**: Memanggil fungsi **maxab** untuk mencari nilai maksimum dari kedua bilangan yang dimasukkan oleh pengguna, kemudian mencetak hasilnya.
11. **tukar (a, b);**: Memanggil prosedur **tukar** untuk menukar nilai dari kedua bilangan yang dimasukkan oleh pengguna, kemudian mencetak hasilnya.



## 19. Program Tempair

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

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PB0\Tugas 3\Tugas3_SourceCode>javac Tempair.java

C:\Users\ASUS\OneDrive\Dokumen\TUGAS PB0\Tugas 3\Tugas3_SourceCode>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 16
Wujud air cair
16
C:\Users\ASUS\OneDrive\Dokumen\TUGAS PB0\Tugas 3\Tugas3_SourceCode>|
```

Penjelasan :

Program di atas adalah contoh program dalam bahasa pemrograman Java yang menggunakan struktur IF dengan tiga kasus untuk menentukan wujud air berdasarkan suhu yang dimasukkan pengguna.

Berikut adalah penjelasan singkat mengenai program tersebut:

1. **import java.util.Scanner;**: Mendeklarasikan penggunaan paket atau class Scanner yang berada dalam paket java.util. Ini memungkinkan program untuk membaca input dari pengguna.
2. **public class Tempair { ... }**: Mendefinisikan sebuah class dengan nama Tempair.
3. **public static void main(String[] args) { ... }**: Metode utama (main method) dari program Java. Eksekusi program dimulai dari sini.
4. **int T;**: Deklarasi variabel T dengan tipe data integer yang akan digunakan untuk menyimpan suhu yang dimasukkan oleh pengguna.
5. **Scanner masukan=new Scanner(System.in);**: Membuat objek Scanner bernama masukan untuk menerima input dari pengguna.
6. **System.out.print ("Contoh IF tiga kasus \n");**: Mencetak pesan sebagai judul program.
7. **System.out.print ("Temperatur (der. C) = ");**: Mencetak pesan untuk meminta pengguna memasukkan suhu dalam derajat Celsius.
8. **T=masukan.nextInt();**: Membaca nilai suhu yang dimasukkan oleh pengguna dan menyimpannya dalam variabel T.
9. **if (T < 0) { ... } else if ((0 <= T) && (T <= 100)) { ... } else if (T > 100) { ... }:**  
Melakukan pengecekan kondisi suhu dengan tiga kasus:
  - Jika suhu kurang dari 0, program akan mencetak bahwa wujud air berupa beku.
  - Jika suhu antara 0 hingga 100, program akan mencetak bahwa wujud air berupa cair.
  - Jika suhu lebih dari 100, program akan mencetak bahwa wujud air berupa uap atau gas.
10. **System.out.print ("Wujud air beku \n"+ T);**: Mencetak pesan bahwa wujud air beku beserta suhu yang dimasukkan pengguna jika suhu kurang dari 0.
11. **System.out.print ("Wujud air cair \n"+ T);**: Mencetak pesan bahwa wujud air cair beserta suhu yang dimasukkan pengguna jika suhu antara 0 hingga 100.
12. **System.out.print ("Wujud air uap/gas \n"+ T);**: Mencetak pesan bahwa wujud air uap atau gas beserta suhu yang dimasukkan pengguna jika suhu lebih dari 100.