

## TUGAS 2

## PEMROGRAMAN BERORIENTASI OBJEK

Sri Ulfiani Rifai

13020230350

### B1/Teknik Informatika

```
Command Prompt
Microsoft Windows [Version 10.0.18363.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Fadli>java -version
java version "23.0.2" 2025-01-21
Java(TM) SE Runtime Environment (build 23.0.2+7-58)
Java HotSpot(TM) 64-Bit Server VM (build 23.0.2+7-58, mixed mode, sharing)

C:\Users\Fadli>javac -version
javac 23.0.2

C:\Users\Fadli>D:

D:\>cd PBO

D:\PBO>cd Tugas

D:\PBO\Tugas>cd Tugas 2
```

Pertama masuk ke Command Prompt, lalu masuk ke **java -version** untuk menampilkan versi java yang terinstall di sistem. Lalu, ditampilkan outputnya menunjukkan laptop saya menggunakan **Java 23.0.2** yang rilis pada 21 Januari 2025. Selanjutnya, masuk ke **javac -version** untuk menampilkan versi dari **Java Compiler (javac)** yang digunakan untuk mengompilasi kode Java. Masuk ke **D:**, artinya berpindah ke drive **D:** dari drive **C:** jika folder yang kamu buat tersimpan di drive **D:**. Kemudian masuk ke **cd PBO**, untuk masuk ke folder **PBO** di drive **D:**. Lalu masuk ke **cd Tugas**, dan terakhir masuk ke **cd Tugas 2** untuk masuk ke folder tersebut yang berada dalam folder **PBO**.

### 1. Class BacaString

```
Command Prompt
Microsoft Windows [Version 10.0.18363.1139]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Fadli>java -version
java version "23.0.2" 2025-01-21
Java(TM) SE Runtime Environment (build 23.0.2+7-58)
Java HotSpot(TM) 64-Bit Server VM (build 23.0.2+7-58, mixed mode, sharing)

C:\Users\Fadli>javac -version
javac 23.0.2

C:\Users\Fadli>D:

D:\>cd PBO

D:\PBO>cd Tugas

D:\PBO\Tugas>cd Tugas 2

D:\PBO\Tugas\Tugas 2>dir
Volume in drive D is NAVAL09
Volume Serial Number is 928E-D3C9

Directory of D:\PBO\Tugas\Tugas 2

11/03/2025  20:58    <DIR>          .
11/03/2025  20:58    <DIR>          ..
11/03/2025  20:58                590 BacaString.java
                1 File(s)              590 bytes
                2 Dir(s)  11.742.228.480 bytes free

D:\PBO\Tugas\Tugas 2>javac BacaString.java

D:\PBO\Tugas\Tugas 2>java BacaString.java
Baca string dan Integer:
masukkan sebuah string: Halo Sri
String yang dibaca : Halo Sri

D:\PBO\Tugas\Tugas 2>dir
Volume in drive D is NAVAL09
Volume Serial Number is 928E-D3C9

Directory of D:\PBO\Tugas\Tugas 2

11/03/2025  21:01    <DIR>          .
11/03/2025  21:01    <DIR>          ..
11/03/2025  21:01                1.255 BacaString.class
11/03/2025  20:58                590 BacaString.java
                2 File(s)              1.845 bytes
                2 Dir(s)  11.742.224.384 bytes free
```

Pada kode, membaca input dari pengguna menggunakan **BufferedReader** dan mencetak kembali input yang dimasukkan. Program meminta pengguna untuk memasukkan sebuah string, kemudian membaca input tersebut menggunakan **readLine()**, dan akhirnya mencetak string yang telah dibaca. Output yang dihasilkan menunjukkan proses ini, di mana saya diminta memasukkan sebuah string, lalu saya memasukkan "Halo Sri", dan program mencetak "String yang dibaca : Halo Sri".

```

C:\Command Prompt>

Directory of D:\PBO\Tugas\Tugas 2

11/03/2025    21:02         <DIR>          .
11/03/2025    21:02         <DIR>          ..
11/03/2025    21:01        1.255 BacaString.class
11/03/2025    20:58           590 BacaString.java
11/03/2025    21:02           274 ForEver.java
                3 File(s)            2.119 bytes
                2 Dir(s)       11.742.224.384 bytes free


D:\PBO\Tugas\Tugas 2>javac ForEver.java

D:\PBO\Tugas\Tugas 2>java ForEver.java
Program akan looping, akhiri dengan ^c
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
^C

D:\PBO\Tugas\Tugas 2>dir
Volume in drive D is HAWAL09
Volume Serial Number is 928E-D3C9

Directory of D:\PBO\Tugas\Tugas 2

11/03/2025    21:02         <DIR>          .
11/03/2025    21:02         <DIR>          ..
11/03/2025    21:01        1.255 BacaString.class
11/03/2025    20:58           590 BacaString.java
11/03/2025    21:02           527 ForEver.class
11/03/2025    21:02           274 ForEver.java
                4 File(s)            2.646 bytes
                2 Dir(s)       11.742.224.384 bytes free
```

Kode menjalankan perulangan tak terbatas (while (true)) yang terus mencetak **"Print satu baris ...."** tanpa henti. Program hanya bisa dihentikan secara paksa dengan

kombinasi tombol **Ctrl + C** (^C di terminal), yang menghentikan eksekusi program secara manual. Sebab itulah outputnya terus berulang hingga kita menghentikannya.

### 3. Class If1

```
D:\PBO\Tugas\Tugas 2>javac If1.java
D:\PBO\Tugas\Tugas 2>java If1.java
Contoh IF satu kasus
Ketikkan suatu nilai integer : 4

Nilai a positif 4
D:\PBO\Tugas\Tugas 2>java If1.java
Contoh IF satu kasus
Ketikkan suatu nilai integer : -4

D:\PBO\Tugas\Tugas 2>dir
Volume in drive D is NAVAL09
Volume Serial Number is 928E-D3C9

Directory of D:\PBO\Tugas\Tugas 2

11/03/2025  21:12    <DIR>          .
11/03/2025  21:12    <DIR>          ..
11/03/2025  21:01             1.255 BacaString.class
11/03/2025  20:58             590 BacaString.java
11/03/2025  21:02             527 ForEver.class
11/03/2025  21:02             274 ForEver.java
11/03/2025  21:12             1.120 If1.class
11/03/2025  21:12             515 If1.java
               6 File(s)              4.281 bytes
               2 Dir(s)  11.742.212.096 bytes free
```

Perintah **javac If1.java** digunakan untuk mengompilasi file **If1.java**, kemudian menghasilkan file **If1.class**. Setelah itu, program dijalankan dengan perintah **java If1.java**, yang mengeksekusi kode dalam file tersebut. Setelah eksekusi selesai, perintah **dir** dijalankan kembali untuk melihat perubahan dalam folder.

Dalam kode terdapat struktur **IF satu kasus** untuk mengecek apakah nilai integer yang dimasukkan adalah positif. Jika **a >= 0**, maka program akan mencetak "**Nilai a positif**" diikuti dengan nilai a. Pada output pertama, saya menginputkan angka **4**, yang merupakan bilangan positif, sehingga program mencetak: "nilai a positif 4". Dan pada output yang kedua, saya menginputkan angka **-4**, yang merupakan bilangan negatif. Karena kondisi if (**a >= 0**) tidak terpenuhi, maka program tidak mencetak apa pun setelah meminta input.

### 4. Class If2

```
D:\PBO\Tugas\Tugas 2>javac If2.java
D:\PBO\Tugas\Tugas 2>java If2.java
Contoh IF dua kasus
Ketikkan suatu nilai integer :10
Nilai a positif 10

D:\PBO\Tugas\Tugas 2>java If2.java
Contoh IF dua kasus
Ketikkan suatu nilai integer :0
Nilai a positif 0

D:\PBO\Tugas\Tugas 2>java If2.java
Contoh IF dua kasus
Ketikkan suatu nilai integer :-8
Nilai a negatif -8

D:\PBO\Tugas\Tugas 2>dir
Volume in drive D is NAVAL09
Volume Serial Number is 928E-D3C9

Directory of D:\PBO\Tugas\Tugas 2

11/03/2025  21:14    <DIR>          .
11/03/2025  21:14    <DIR>          ..
11/03/2025  21:01             1.255 BacaString.class
11/03/2025  20:58             590 BacaString.java
11/03/2025  21:02             527 ForEver.class
11/03/2025  21:02             274 ForEver.java
11/03/2025  21:12             1.120 If1.class
11/03/2025  21:12             515 If1.java
11/03/2025  21:14             1.191 If2.class
11/03/2025  21:14             663 If2.java
               8 File(s)              6.135 bytes
               2 Dir(s)  11.742.208.000 bytes free
```

Perintah **javac If2.java** digunakan untuk mengompilasi file **If2.java**, menghasilkan file **If2.class**. Setelah itu, program dijalankan dengan perintah **java If2.java**, yang

memungkinkan pengguna untuk memasukkan nilai saat eksekusi. Setelah program selesai dijalankan, perintah dir menunjukkan bahwa file **If2.class** telah ditambahkan. Kode program pada **If2.java** menggunakan struktur IF dua kasus komplementer (**if-else**), yang memastikan bahwa setiap nilai input selalu menghasilkan output. Jika nilai yang dimasukkan lebih besar atau sama dengan nol (**a >= 0**), maka program akan mencetak "**Nilai a positif**" diikuti dengan nilai tersebut. Sebaliknya, jika nilai yang dimasukkan kurang dari nol (**a < 0**), program akan mencetak "**Nilai a negatif**" beserta nilainya. Output yang dihasilkan selalu ada, berbeda dengan program sebelumnya (**If1.java**) yang hanya menggunakan satu kondisi if tanpa else. Dalam **If1.java**, jika nilai yang dimasukkan negatif, program tidak akan mencetak apa pun setelah input. Sedangkan, dalam **If2.java**, karena ada blok else, program tetap memberikan output meskipun nilai yang dimasukkan negatif. Oleh karena itu, hasil eksekusi program ini lebih lengkap dan mencakup semua kemungkinan input, baik positif maupun negatif.

## 5. Class If3

```
Command Prompt
D:\PBO\Tugas\Tugas 2>javac If3.java

D:\PBO\Tugas\Tugas 2>java If3.java
Contoh IF tiga kasus
Ketikkan suatu nilai integer :3
Nilai a positif 3

D:\PBO\Tugas\Tugas 2>java If3.java
Contoh IF tiga kasus
Ketikkan suatu nilai integer :0
Nilai Nol 0

D:\PBO\Tugas\Tugas 2>java If3.java
Contoh IF tiga kasus
Ketikkan suatu nilai integer :-3
Nilai a negatif -3
```

Kode program **If3.java** menggunakan struktur IF tiga kasus (**if-else if-else**), yang mencakup semua kemungkinan nilai integer yang diinputkan. Program akan membaca sebuah bilangan dan menentukan apakah bilangan tersebut positif, nol, atau negatif. Jika bilangan lebih besar dari nol (**a > 0**), program akan mencetak "**Nilai a positif**" diikuti dengan nilainya. Jika bilangan sama dengan nol (**a == 0**), program akan mencetak "**Nilai Nol**" diikuti dengan angka nol. Jika bilangan kurang dari nol (**a < 0**), program akan mencetak "**Nilai a negatif**" diikuti dengan nilainya.

Perbedaan utama antara **If3.java** dengan kode sebelumnya (**If2.java**) terletak pada perlakuan terhadap angka nol. Pada **If2.java**, angka nol diperlakukan sebagai bilangan positif karena kondisi yang diperiksa adalah **a >= 0**. Namun, dalam **If3.java**, angka nol memiliki kondisi tersendiri (**a == 0**), sehingga lebih jelas dalam membedakan tiga kemungkinan nilai yang dapat diinputkan. Dengan demikian, program ini lebih lengkap dalam menangani berbagai input angka dibandingkan dengan struktur IF satu atau dua kasus.

## 6. Class KasusBoolean

```
D:\PBO\Tugas\Tugas 2>javac KasusBoolean.java

D:\PBO\Tugas\Tugas 2>java KasusBoolean.java
true
benar
```

Kode program **KasusBoolean.java** menunjukkan penggunaan ekspresi kondisional dengan tipe data **boolean**. Program mendeklarasikan variabel bool dengan nilai **true** dan menggunakannya dalam pernyataan **if-else** untuk menentukan output.

Pada pengecekan pertama, jika bool bernilai **true**, maka akan mencetak "**true**", jika tidak, akan mencetak "**false**". Karena nilai awal bool adalah **true**, maka program mencetak "**true**".

Pada pengecekan kedua, ekspresi **!bool** digunakan, yang berarti kebalikan dari nilai bool. Karena bool adalah **true**, maka **!bool** akan menjadi **false**, sehingga bagian **if(!bool)** tidak dijalankan dan program mengeksekusi bagian **else**, yang mencetak "**benar**".

## 7. Class KasusSwitch

```
Command Prompt

D:\PBO\Tugas\Tugas 2>javac KasusSwitch.java

D:\PBO\Tugas\Tugas 2>java KasusSwitch.java
Ketikkan sebuah huruf, akhiri dengan RETURN
a
Yang anda ketik adalah a

D:\PBO\Tugas\Tugas 2>java KasusSwitch.java
Ketikkan sebuah huruf, akhiri dengan RETURN
u
Yang anda ketik adalah u

D:\PBO\Tugas\Tugas 2>java KasusSwitch.java
Ketikkan sebuah huruf, akhiri dengan RETURN
s
Yang anda ketik adalah huruf mati
```

Program **KasusSwitch.java** menggunakan **switch-case** untuk mengecek apakah karakter yang dimasukkan merupakan huruf vokal atau bukan. Pengguna diminta mengetikkan sebuah huruf, lalu program membandingkannya dengan beberapa kemungkinan, yaitu '**a**', '**u**', '**e**', '**i**', dan '**o**'. Jika huruf yang diinputkan cocok dengan salah satu vokal tersebut, program akan mencetak pesan bahwa huruf yang diketik adalah vokal yang bersangkutan. Namun, keika huruf yang diinputkan tidak sesuai dengan salah satu vocal, maka program mencetak pesan “Yang anda ketik adalah huruf mati”.

Berdasarkan output, saya menginputkan 'a', dan program menampilkan "**Yang anda ketik adalah a**". Jika saya inputkan 'u', program menampilkan "**Yang anda ketik adalah u**". Namun, ketika saya menginputkan huruf selain vokal, misalnya 's', program akan mencetak "**Yang anda ketik adalah huruf mati**".

## 8. Class Konstant

```
D:\PBO\Tugas\Tugas 2>javac Konstant.java

D:\PBO\Tugas\Tugas 2>java Konstant.java
Jari-jari lingkaran =25
Luas lingkaran = 1963.4375
Akhir program

D:\PBO\Tugas\Tugas 2>java Konstant.java
Jari-jari lingkaran =10
Luas lingkaran = 314.15002
Akhir program
```

Program **Konstant.java** digunakan untuk menghitung luas lingkaran dengan rumus  $L = \pi \times r^2$ , di mana  $\pi$  (PHI) memiliki nilai 3.1415. Pengguna diminta untuk menginputkan jari-jari lingkaran, lalu program akan menghitung luasnya dan menampilkan hasilnya.

Berdasarkan output, saya menginputkan dengan jari-jari **25**, dan program menampilkan "**Luas lingkaran = 1963.4375**". Jika jari-jari yang saya inputkan adalah **10**, program akan menampilkan "**Luas lingkaran = 314.15002**". Setelah hasilnya ditampilkan, program mencetak pesan "**Akhir program**", yang menandakan eksekusi telah selesai.

## 9. Class Max2

```
D:\PBO\Tugas\Tugas 2>javac Max2.java
D:\PBO\Tugas\Tugas 2>java Max2.java
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
12
6
Ke dua bilangan : a = 12 b = 6
Nilai a yang maksimum 12

D:\PBO\Tugas\Tugas 2>java Max2.java
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
3
9
Ke dua bilangan : a = 3 b = 9
Nilai b yang maksimum: 9

D:\PBO\Tugas\Tugas 2>java Max2.java
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
8
8
Ke dua bilangan : a = 8 b = 8
Nilai a yang maksimum 8
```

Program **Max2.java** bertujuan untuk membaca dua bilangan integer yang diinputkan dan menentukan bilangan yang lebih besar. Program menggunakan struktur **if-else** untuk membandingkan kedua bilangan dan menampilkan hasilnya.

Ketika program dijalankan, pengguna diminta untuk menginputkan dua bilangan dengan menekan **ENTER** setelah setiap bilangan. Program kemudian mencetak kedua bilangan tersebut dan menentukan mana yang lebih besar. Jika bilangan pertama lebih besar atau sama dengan bilangan kedua, program mencetak bilangan pertama sebagai yang terbesar. Jika tidak, program mencetak bilangan kedua sebagai yang terbesar.

Dari hasil eksekusi, menginputkan bilangan **12 dan 6**, program menampilkan bahwa nilai maksimum adalah **12**. Dan ketika saya menginputkan **3 dan 9**, program menampilkan **9** sebagai nilai maksimum. Jika kedua bilangan sama, seperti **8 dan 8**, program tetap mencetak salah satunya sebagai nilai maksimum karena kondisi **a >= b** tetap terpenuhi.

## 10. Class PriFor

```
D:\PBO\Tugas\Tugas 2>javac PriFor.java
D:\PBO\Tugas\Tugas 2>java PriFor.java
Baca N, print 1 s/d N N = 6
1
2
3
4
5
6
Akhir program
```

Kode program **PriFor.java** digunakan untuk membaca sebuah bilangan integer **N** dari input pengguna dan mencetak angka dari **1 hingga N** menggunakan perulangan **FOR**. Saat program dijalankan, pengguna diminta untuk menginputkan nilai **N**. Program kemudian menggunakan perulangan **for** yang dimulai dari **i = 1** hingga **i <= N**,

mencetak setiap nilai **i** dalam satu baris. Setelah selesai, program menampilkan pesan "**Akhir program**".

Berdasarkan output, saya memasukkan **N = 6**, sehingga program akan mencetak angka **1 hingga 6** secara berurutan, diikuti dengan pesan **Akhir program**.

## 11. Class PrintIterasi

```
D:\PBO\Tugas\Tugas 2>javac PrintIterasi.java
D:\PBO\Tugas\Tugas 2>java PrintIterasi.java
Nilai N >0 = 4
Print i dengan ITERATE :
1
2
3
4
```

Program **PrintIterasi.java** digunakan untuk mencetak angka dari 1 hingga N menggunakan perulangan **for** tanpa batasan eksplisit atau **infinite loop**. Program ini meminta untuk memasukkan nilai N yang lebih dari nol, kemudian mencetak angka mulai dari 1 hingga N. Perulangan ini terus berjalan tanpa batas, tetapi dihentikan secara manual menggunakan perintah **break** saat nilai **i** mencapai N.

Perbedaannya dengan program **PriFor.java** terletak pada cara perulangan dilakukan. Pada **PriFor.java**, perulangan **for** memiliki kondisi yang jelas, yaitu  $i \leq N$ , sehingga iterasi berhenti secara otomatis setelah mencapai batas. Sementara itu, **PrintIterasi.java** menggunakan perulangan tanpa batas yang dikontrol secara manual dengan **break**, sehingga lebih fleksibel tetapi juga lebih berisiko jika tidak dikelola dengan baik.

Berdasarkan pada output, saat program dijalankan dan saya sebagai pengguna memasukkan nilai **N = 4**, output yang dihasilkan adalah angka **1 hingga 4** yang dicetak secara berurutan. Setelah mencapai angka 4, program berhenti karena kondisi **if (i == N) break;** terpenuhi.

## 12. Class PrintRepeat

```
Command Prompt
D:\PBO\Tugas\Tugas 2>javac PrintRepeat.java
D:\PBO\Tugas\Tugas 2>java PrintRepeat.java
Nilai N >0 = 5
Print i dengan REPEAT:
1
2
3
4
5
D:\PBO\Tugas\Tugas 2>java PrintRepeat.java
Nilai N >0 = 0
Print i dengan REPEAT:
1
```

Program **PrintRepeat.java** digunakan untuk mencetak angka dari 1 hingga N menggunakan perulangan **do-while**. Program meminta untuk memasukkan nilai N yang lebih besar dari nol. Setelah itu, angka **i** akan dicetak mulai dari 1, lalu bertambah satu per iterasi hingga mencapai N.

Berbeda dengan program **PriFor.java** yang menggunakan perulangan **for** dengan kondisi eksplisit, serta **PrintIterasi.java** yang menggunakan **for** tanpa kondisi eksplisit (infinite loop dengan **break**), program ini menggunakan **do-while**.

Perbedaannya terletak pada mekanisme eksekusi, **do-while** akan menjalankan blok perulangan setidaknya sekali, karena kondisi diperiksa setelah proses dijalankan.

Pada saat program dijalankan dengan menginputkan  $N = 5$ , output yang dihasilkan adalah angka **1 hingga 5** yang dicetak berurutan. Namun, jika saya menginputkan  $N = 0$ , program tetap mencetak **1**, meskipun kondisi **while** ( $i \leq N$ ) seharusnya tidak terpenuhi. Hal ini terjadi karena **do-while** menjalankan satu iterasi terlebih dahulu sebelum memeriksa kondisi, sehingga angka **1** tetap tercetak.

### 13. Class PrintWhile

```
Command Prompt
D:\PBO\Tugas\Tugas 2>javac PrintWhile.java
D:\PBO\Tugas\Tugas 2>java PrintWhile.java
Nilai N >0 = 3
Print i dengan WHILE:
1
2
3
D:\PBO\Tugas\Tugas 2>java PrintWhile.java
Nilai N >0 = 0
Print i dengan WHILE:
```

Program **PrintWhile.java** digunakan untuk membaca bilangan  $N$  dan mencetak angka dari 1 hingga  $N$  menggunakan perulangan **while**. Program meminta untuk menginputkan nilai  $N$  yang lebih besar dari nol. Variabel  $i$  diinisialisasi dengan nilai 1, kemudian program mencetak angka dari 1 sampai  $N$  dengan menaikkan nilai  $i$  setiap iterasi hingga mencapai  $N$ .

Berbeda dengan **PrintRepeat.java** yang menggunakan **do-while**, perulangan **while** dalam program ini mengevaluasi kondisi sebelum menjalankan blok kode. Jika kondisi tidak terpenuhi sejak awal (misalnya, jika  $N = 0$ ), maka perulangan **tidak akan dijalankan sama sekali**, sehingga tidak ada output angka.

Contohnya pada output, ketika diinputkan dengan  $N = 3$ , program mencetak angka **1, 2, 3** secara berurutan. Namun, jika  $N = 0$ , program hanya mencetak teks "**Print i dengan WHILE:**" tanpa mencetak angka apa pun, karena kondisi **while** ( $i \leq N$ ) langsung **false** sejak awal.

### 14. Class PrintWhile1

```
Command Prompt
D:\PBO\Tugas\Tugas 2>javac PrintWhile1.java
D:\PBO\Tugas\Tugas 2>java PrintWhile1.java
Nilai N >0 = 5
Print i dengan WHILE (ringkas):
1
2
3
4
5
```

Program **PrintWhile1.java** berfungsi untuk membaca bilangan  $N$  dan mencetak angka dari **1 hingga N** menggunakan perulangan **while** dalam bentuk yang lebih ringkas. Variabel  $i$  diinisialisasi dengan nilai **1**, lalu program mencetak angka dari **1 hingga N** menggunakan perintah **System.out.println(i++)**, yang sekaligus mencetak nilai  $i$  dan menaikkannya dalam satu baris kode. Berbeda dengan **PrintWhile.java**, program ini lebih ringkas karena deklarasi variabel  $i$  langsung dilakukan saat



inisialisasi, dan proses peningkatan nilai **i** dilakukan langsung dalam pernyataan **System.out.println(i++)**, menghilangkan kebutuhan untuk menulis **i++** secara terpisah.

Saya menginputkan dengan **N = 5**. Saat dijalankan, program akan mencetak angka **1, 2, 3, 4, dan 5** secara berurutan. Namun, ketika saya inputkan **N = 0**, tidak ada angka yang dicetak karena kondisi **while (i <= N)** tidak terpenuhi sejak awal.

## 15. Class PrintXinterasi

```
D:\PBO\Tugas\Tugas 2>javac PrintXinterasi.java

D:\PBO\Tugas\Tugas 2>java PrintXinterasi.java
Masukkan nilai x (int), akhiri dg 999: 999
Kasus kosong
Hasil penjumlahan = 0

D:\PBO\Tugas\Tugas 2>java PrintXinterasi.java
Masukkan nilai x (int), akhiri dg 999: 5
Masukkan nilai x (int), akhiri dg 999 : 1
Masukkan nilai x (int), akhiri dg 999 : 7
Masukkan nilai x (int), akhiri dg 999 : 2
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 15
```

Program **PrintXinterasi.java** digunakan untuk membaca sekumpulan bilangan bulat yang diinputkan, kemudian menjumlahkannya hingga diinputkan angka **999** sebagai tanda berhenti.

Pertama, program meminta untuk menginputkan nilai pertama. Jika nilai tersebut **999**, maka program akan mencetak "Kasus kosong" dan menampilkan hasil penjumlahan **0** karena tidak ada angka yang diproses. Namun, jika yang diinputkan adalah angka selain **999**, angka tersebut langsung menjadi nilai awal **Sum**.

Selanjutnya, program memasuki perulangan **for (;;) (loop tak terbatas)** untuk terus meminta penginputan angka baru. Jika yang diinputkan **999**, loop akan berhenti (**break**). Jika bukan **999**, angka tersebut akan ditambahkan ke **Sum**. Setelah loop selesai, program mencetak hasil penjumlahan.

Pada output yang sudah dijalankan, jika yang diinputkan adalah angka **5, 1, 7, 2**, lalu **999**, maka program akan menjumlahkan angka tersebut dan mencetak **Hasil penjumlahan = 15**. Namun, jika langsung menginputkan **999**, maka program akan menampilkan **Kasus kosong** dengan hasil **0**.

## 16. Class PrintXRepeat

```
Command Prompt

D:\PBO\Tugas\Tugas 2>javac PrintXRepeat.java

D:\PBO\Tugas\Tugas 2>java PrintXRepeat.java
Masukkan nilai x (int), akhiri dg 999 : 999
Kasus kosong

D:\PBO\Tugas\Tugas 2>java PrintXRepeat.java
Masukkan nilai x (int), akhiri dg 999 : 8
Masukkan nilai x (int), akhiri dg 999 : 2
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 10
```

Program **PrintXRepeat.java** digunakan untuk membaca sekumpulan bilangan bulat dari yang diinputkan dan menjumlahkannya yang diinputkan adalah angka **999** sebagai tanda berhenti.

Pertama, program meminta untuk menginputkan nilai **x**. Jika langsung menginputkan **999**, maka program akan mencetak "**Kasus kosong**" dan berhenti tanpa melakukan

proses penjumlahan. Namun, jika nilai awal bukan **999**, program akan memasuki perulangan **do-while** untuk menjumlahkan angka yang diinputkan.

Di dalam loop **do-while**, nilai **x** langsung ditambahkan ke **Sum**, kemudian program kembali meminta input berikutnya. Perulangan akan terus berjalan hingga pengguna menginputkan **999**, yang menjadi kondisi berhenti. Setelah itu, program mencetak hasil penjumlahan angka yang telah dimasukkan.

Pada output yang sudah dijalankan, saya menginputkan angka **8, 2**, lalu **999**, maka program akan menjumlahkan angka tersebut dan menampilkan **Hasil penjumlahan = 10**. Namun, ketika saya langsung menginputkan **999**, maka program hanya mencetak "**Kasus kosong**" tanpa hasil penjumlahan.

## 17. Class PrintXWhile

```
Command Prompt

D:\PBO\Tugas\Tugas 2>javac PrintXWhile.java

D:\PBO\Tugas\Tugas 2>java PrintXWhile.java
Masukkan nilai x (int), akhiri dg 999 : 5
Masukkan nilai x (int), akhiri dg 999 : 2
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 7

D:\PBO\Tugas\Tugas 2>java PrintXWhile.java
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 0
```

Program **PrintXWhile.java** berfungsi untuk membaca sekumpulan bilangan bulat yang diinputkan dan menjumlahkannya hingga yang diinputkan adalah angka **999** sebagai tanda berhenti. Dalam program ini, variabel **Sum** diinisialisasi dengan nilai **0**. Program pertama-tama meminta untuk menginputkan nilai **x** sebelum memasuki perulangan **while**. Jika yang diinputkan langsung **999**, program tidak akan memasuki perulangan dan akan langsung mencetak hasil penjumlahan sebagai **0**. Namun, jika nilai yang diinputkan bukan **999**, maka perulangan **while** akan terus berjalan dan menjumlahkan angka yang dimasukkan ke dalam **Sum** hingga pengguna akhirnya memasukkan **999**, yang menyebabkan loop berhenti dan hasil penjumlahan ditampilkan. Pada output dari program yang sudah dijalankan, saya menginputkan angka **5**, lalu **2**, dan kemudian **999**, maka program akan menampilkan **Hasil penjumlahan = 7**. Sebaliknya, ketika saya langsung menginputkan **999**, hasil yang ditampilkan adalah **0**.

Perbedaan utama antara **PrintXWhile.java** dan **PrintXRepeat.java** terletak pada jenis perulangan yang digunakan. **PrintXWhile.java** menggunakan perulangan **while**, di mana kondisi diperiksa terlebih dahulu sebelum masuk ke dalam perulangan. Jika angka pertama yang diinput adalah **999**, maka perulangan tidak akan dijalankan sama sekali. Sementara itu, **PrintXRepeat.java** menggunakan perulangan **do-while**, yang memastikan bahwa perulangan akan dieksekusi setidaknya satu kali sebelum memeriksa kondisi **999**. Akibatnya, dalam **PrintXRepeat.java**, jika yang diinputkan angka selain **999**, nilai tersebut langsung dijumlahkan sebelum pengecekan kondisi berhenti dilakukan.

## 18. Class SubProgram

```
Command Prompt
D:\PBO\Tugas\Tugas 2>java SubProgram.java
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
2
1
Ke dua bilangan : a = 2 b = 1
Maksimum = 2
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 1 b = 2
```

Program **SubProgram.java** adalah contoh program yang menggunakan fungsi dan prosedur dalam Java. Program memiliki dua metode utama selain **main()**, yaitu **maxab()** sebagai fungsi dan **tukar()** sebagai prosedur.

Fungsi **maxab(int a, int b)** digunakan untuk mencari nilai maksimum dari dua bilangan bulat yang diinputkan. Fungsi ini menggunakan **operator ternary ((a >= b) ? a : b)** untuk mengembalikan nilai maksimum antara a dan b.

Prosedur **tukar(int a, int b)** berfungsi untuk menukar nilai dua bilangan yang diberikan. Proses pertukaran dilakukan dengan menggunakan variabel sementara **temp**. Setelah pertukaran, nilai yang baru akan dicetak di dalam prosedur tersebut. Dalam **main()**, program meminta untuk menginputkan dua bilangan integer. Setelah itu, program menampilkan nilai maksimum dari kedua bilangan tersebut dengan memanggil fungsi **maxab()**. Kemudian, program memanggil prosedur **tukar()** untuk menukar kedua bilangan.

Pada output program yang sudah saya jalankan, saya menginputkan angka **2** dan **1** sebagai dua bilangan yang akan diproses. Program pertama-tama menampilkan kedua bilangan tersebut, lalu mencari nilai maksimum di antara keduanya menggunakan fungsi **maxab**, yang menghasilkan output **2** sebagai bilangan terbesar. Setelah itu, program memanggil prosedur **tukar** untuk menukar nilai kedua bilangan, sehingga a yang awalnya **2** menjadi **1**, dan b yang awalnya **1** menjadi **2** di dalam prosedur. Namun, karena Java menggunakan *pass-by-value*, perubahan ini hanya berlaku di dalam prosedur **tukar**, sedangkan nilai a dan b di dalam **main()** tetap sama seperti input awal.

Namun, ada satu kelemahan dalam program ini, dimana nilai a dan b dalam metode **tukar()** tidak berubah di **main()** setelah pemanggilan prosedur, karena Java menggunakan *pass by value*. Artinya, hanya salinan dari nilai a dan b yang dikirim ke prosedur **tukar()**, sehingga perubahan di dalam prosedur tidak mempengaruhi nilai asli di **main()**.

## 19. Class Tempair

```
Command Prompt
D:\PBO\Tugas\Tugas 2>javac Tempair.java
D:\PBO\Tugas\Tugas 2>java Tempair.java
Contoh IF tiga kasus
Temperatur (der. C) = -5
Wujud air beku
-5
D:\PBO\Tugas\Tugas 2>java Tempair.java
Contoh IF tiga kasus
Temperatur (der. C) = 30
Wujud air cair
30
D:\PBO\Tugas\Tugas 2>java Tempair.java
Contoh IF tiga kasus
Temperatur (der. C) = 105
Wujud air uap/gas
105
```

Program **Tempair.java** digunakan untuk menentukan wujud air berdasarkan suhu yang diinputkan. Program ini menggunakan struktur *if-else* dengan tiga kondisi untuk mengklasifikasikan apakah air berada dalam bentuk **beku, cair, atau uap/gas**.

Pertama, program meminta untuk menginputkan suhu dalam derajat Celsius. Jika suhu yang diinputkan kurang dari **0°C**, program akan mencetak "*Wujud air beku*" karena air membeku pada suhu ini. Jika suhu berada dalam rentang **0°C hingga 100°C**, program akan mencetak "*Wujud air cair*", karena dalam kisaran ini air tetap dalam bentuk cair. Terakhir, jika suhu lebih dari **100°C**, program akan mencetak "*Wujud air uap/gas*", karena air mulai menguap pada suhu ini.

Berdasarkan pada putput dari program, saya menginputkan **-5**, maka program akan mengevaluasi bahwa **-5 < 0**, sehingga akan mencetak "*Wujud air beku -5*". Lalu saya menginputkan **30**, dan program akan mengecek bahwa **0 ≤ 30 ≤ 100**, sehingga mencetak "*Wujud air cair 30*". Kemudian saya menginputkan **105**, sehingga program akan melihat bahwa **105 > 100**, dan mencetak "*Wujud air uap/gas 105*".