

**LAPORAN
PEMROGRAMAN BERBASIS OBJEK**



Nama : Rizqi Ananda Jalil
Stambuk : 13020230244
Kelas : TI_PBO-14 (B4)
Dosen : Mardiyah Hasnawi, S.Kom., M.T., MTA

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR
2025**

Tugas 2 – Pemrograman Berbasis Objek

1. Output:

```
D:\Tugas PB0\Tugas 2>javac BacaString.java
D:\Tugas PB0\Tugas 2>java BacaString
Baca string dan Integer:
masukkan sebuah string: r
String yang dibaca : r
```

Penjelasan :

Program BacaString ini membaca inputan yang kita masukkan menggunakan library BufferedReader. Library IOException untuk menangani kesalahan input/output serta library InputStreamReader untuk mengonversi data input menjadi format yang bisa dibaca oleh program. Didalam program ini mendeklarasikan variabel str bertipe data String untuk menyimpan input. Objek BufferedReader datAIn dibuat untuk membaca input dari keyboard. Setelah mencetak pesan "Baca string dan integer: ", program ini akan meminta kita untuk memasukkan sebuah string. Disini saya menginputkan string "r" dan inputan tersebut dibaca menggunakan datAIn.readLine() dan disimpan dalam str. Terakhir, program akan mencetak kembali string yang kita masukkan sebelumnya dalam hal ini yang dicetak adalah r yang telah saya inputkan sebelumnya.

2. Output:

```
D:\Tugas PB0\Tugas 2>javac ForEver.java
D:\Tugas PB0\Tugas 2>java ForEver
Program akan looping, akhiri dengan ^c
Print satu baris ...
```

Penjelasan:

Program ForEver ini merupakan program sederhana yang menggunakan perulangan while(true) untuk mencetak teks ke layar. Di awal program, terdapat pernyataan System.out.println("Program akan looping, akhiri dengan ^c"); yang mencetak pesan bahwa program akan berjalan terus menerus kecuali dihentikan secara paksa. Namun, di dalam perulangan while(true), terdapat perintah System.out.print("Print satu baris ...\n"); yang mencetak teks ke layar, diikuti dengan perintah break;, yang langsung menghentikan perulangan. Dengan adanya break, program sebenarnya tidak berjalan terus-menerus seperti yang dikatakan di awal, tetapi hanya mencetak satu baris teks lalu berhenti.

3. Output:

- Nilai Positif

```
D:\Tugas PB0\Tugas 2>javac If1.java
D:\Tugas PB0\Tugas 2>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : 7
Nilai a positif 7
```

- Tidak ada output tambahan

```
D:\Tugas PB0\Tugas 2>javac If1.java
D:\Tugas PB0\Tugas 2>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : -3
```

Penjelasan:

Program If1 ini sebagai contoh penggunaan percabangan if dalam Java untuk menentukan apakah nilai yang diinputkan merupakan bilangan positif atau tidak. Didalam program ini menggunakan Scanner untuk membaca inputan yang kita masukkan. Program ini akan mencetak pesan "Contoh IF satu kasus" dan meminta kita sebagai pengguna untuk memasukkan sebuah bilangan integer. Nilai yang iinputkan akan disimpan dalam variabel a, Kemudian, program akan melakukan pengecekan menggunakan `if(a >= 0)`, yang berarti jika a lebih besar atau sama dengan nol maka program akan mencerak "Nilai a positif: diikuti dengan nilai a. Jika nilai a bernilai negatif, maka tidak ada output tambahan karena tidak ada perintah didalam source code untuk menangani bilangan negatif.

4. Output:

- Nilai Positif

```
D:\Tugas PB0\Tugas 2>javac If2.java  
  
D:\Tugas PB0\Tugas 2>java If2  
Contoh IF dua kasus  
  
Ketikkan suatu nilai integer : 3  
Nilai a positif 3
```

- Nilai Negatif

```
D:\Tugas PB0\Tugas 2>javac If2.java  
  
D:\Tugas PB0\Tugas 2>java If2  
Contoh IF dua kasus  
  
Ketikkan suatu nilai integer : -1  
Nilai a negatif -1
```

Penjelasan:

Program If2 ini mengimplementasikan penggunaan struktur percabangan if-else dalam Java untuk menentukan apakah bilangan yang diinputkan bernilai positif atau negatif. Sama dengan program If1, program If2 juga menggunakan Scanner untuk membaca inputan dari kita. Pertama, program ini akan mencetak pesan "Contoh IF dua kasus" dan meminta kita sebagai pengguna menginputkan sebuah bilangan integer. Nilai yang dimasukkan juga disimpan ke dalam variabel a. Kemudian akan dilakukan pengecekan jika `(a >= 0)` maka akan mencetak "Nilai a positif" diikuti dengan nilai a. Jika kondisi tersebut tidak terpenuhi maka perintah dalam blok else akan dieksekusi, mencetak "Nilai a negatif" diikuti dengan nilai a.

5. Output:

- Nilai Positif

```
D:\Tugas PB0\Tugas 2>javac If3.java  
  
D:\Tugas PB0\Tugas 2>java If3  
Contoh IF tiga kasus  
Ketikkan suatu nilai integer : 2  
Nilai a positif 2
```

- Nilai Negatif

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

D:\Tugas PBO\Tugas 2>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer : -4
Nilai a negatif -4
```

- Nilai Nol

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

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

Penjelasan:

Program If3 ini mengimplementasikan penggunaan struktur percabangan if-else if-else dalam Java untuk menentukan apakah bilangan yang diinputkan bernilai positif, nol, atau negatif. Sama seperti program If2, program ini juga menggunakan Scanner untuk membaca input dari pengguna. Pertama, program mencetak pesan "Contoh IF tiga kasus" dan meminta pengguna menginputkan sebuah bilangan integer. Nilai yang dimasukkan disimpan dalam variabel a. Kemudian, program melakukan pengecekan: jika (a > 0), maka akan mencetak "Nilai a positif" diikuti dengan nilai a. Jika (a == 0), maka akan mencetak "Nilai Nol" diikuti dengan nilai a. Jika kondisi tersebut tidak terpenuhi (a < 0), maka perintah dalam blok else akan dieksekusi, mencetak "Nilai a negatif" diikuti dengan nilai a.

6. Output:

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

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

Penjelasan :

Program KasusBoolean ini mengimplementasikan penggunaan tipe data Boolean dalam struktur percabangan if-else di Java. Pertama, program mendeklarasikan variabel bool bertipe Boolean dan memberinya nilai true. Kemudian, program melakukan dua kali pengecekan yaitu pertama if(bool) maka akan mencetak nilai "true". Jika bernilai false, maka program akan mencetak "false: dalam blok else. Kedua, if(!bool) terdapat operator negasi !(NOT) membalik nilai bool, sehingga !bool menjadi false. Karena kondisi ini tidak terpenuhi, maka program melewati perintah dalam blok if dan mengeksekusi else sehingga mencetak "benar". Jika bool awalnya false, maka akan mencetak "salah".

7. Output:

- Huruf vokal

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

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

- Huruf mati

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

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

Penjelasan:

Program KasusSwitch ini mengimplementasikan penggunaan struktur percabangan switch-case dalam Java untuk mengevaluasi sebuah karakter yang kita inputkan sebagai pengguna. Program mendeklarasikan variabel cc bertipe char dan menggunakan Scanner untuk membaca inputan. Setelah mencetak pesan "Ketikkan sebuah huruf, akhiri dengan RETURN", program membaca satu karakter menggunakan masukan.next().charAt(0), yang mengambil karakter pertama dari inputan. Selanjutnya, program melakukan pengecekan menggunakan switch-case. Jika kita melakukan penginputan dengan mengetikkan huruf vokal kecil ('a', 'u', 'e', 'i', atau 'o'), program akan mencetak pesan sesuai dengan huruf yang diketikkan. Jika huruf yang diketik bukan vokal, maka blok default akan dijalankan, mencetak "Yang anda ketik adalah huruf mati", yang berarti huruf konsonan. Program ini menunjukkan cara menggunakan switch-case untuk menangani berbagai kemungkinan input tanpa harus menggunakan banyak pernyataan if-else.

8. Output:

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

D:\Tugas PBO\Tugas 2>java Konstant
Jari-jari lingkaran = 21
Luas lingkaran = 1385.4014
Akhir program
```

Penjelasan:

Program Konstant ini mengimplementasikan perhitungan luas lingkaran berdasarkan jari-jari yang diinputkan. Dalam program ini, yang pertama adalah kita mendeklarasikan konstanta PHI dengan nilai 3.1415f menggunakan kata kunci final, yang berarti nilainya tidak dapat diubah selama program berjalan. Kemudian mendeklarasikan variabel r yang bertipe data float untuk digunakan menyimpan nilai jari-jari yang nilainya akan diinputkan oleh pengguna. Setelah itu dibuat juga objek Scanner dengan nama masukan untuk membaca inputan lalu akan mencetak pesan "Jari-jari lingkaran = ", program nya akan meminta kita memasukkan nilai jari-jari. Dalam hal ini saya menginputkan jari-jari bernilai 21 kemudian nilai tersebut dibaca menggunakan masukan.nextFloat() dan disimpan didalam variabel r. Selanjutnya program akan menghitung luas lingkaran menggunakan rumus $\text{PHI} \times r \times r$ dan mencetak hasilnya yaitu 1385.44014. Lalu program akan menampilkan pesan "Akhir program" sebelum selesai dieksekusi.

9. Output:

- Nilai a yang maksimum

```
D:\Tugas PBO\Tugas 2>javac Max2.java

D:\Tugas PBO\Tugas 2>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dengan RETURN :
9 4
Ke dua bilangan : a = 9 b = 4
Nilai a yang maksimum: 9
```

- Nilai b yang maksimum

```
D:\Tugas PBO\Tugas 2>javac Max2.java

D:\Tugas PBO\Tugas 2>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dengan RETURN :
2 5
Ke dua bilangan : a = 2 b = 5
Nilai b yang maksimum: 5
```

Penjelasan :

Program Max2 ini mengimplementasikan bagaimana menentukan bilangan terbesar dari dua bilangan yang diinputkan. Didalam program ini, dideklarasikan dua variabel yaitu a dan b yang bertipe data integer untuk menyimpan dua bilangan bulat dan objek Scanner dengan nama masukan yang dibuat untuk membaca inputan yang kita ketikkan sebagai pengguna. Setelah mencetak pesan “Maksimum dua bilangan”, program akan meminta pengguna untuk memasukkan dua bilangan secara terpisah dengan menekan tombol ENTER setelah setiap langkah yang dilakukan. Nilai tersebut kemudian dibaca menggunakan `masukan.nextInt()` dan disimpan dalam variabel a dan b. Program kemudian membandingkan kedua bilangan menggunakan struktur percabangan if-else yaitu jika a lebih besar atau sama dengan b, maka program mencetak “Nilai a yang maksimum: a” dan jika b lebih besar maka program mencetak “Nilai b yang maksimum: b”. Dari situlah program bisa menentukan dan mencetak bilangan terbesar diantara dua angka yang dimasukkan. Dalam hal ini saya menginputkan angka 9=a dan 4=b sehingga didapatkan variabel a yang menjadi nilai maksimum dan saya mencoba kembali menginputkan angka 2=a dan 5=b sehingga variabel b yang menjadi nilai maksimum.

10. Output:

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

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

Penjelasan:

Program PriFor mengimplementasikan pencetakan bilangan dari 1 hingga N berdasarkan yang diinputkan. Dalam program ini, dideklarasikan terlebih dahulu variabel i yang digunakan sebagai penghitung dalam perulangan (for) sedangkan variabel N menyimpan batas angka yang akan dicetak dan dideklarasikan juga objek Scanner dengan nama masukan untuk membaca inputan. Setelah itu, dicetak pesan “Baca N, print 1 s/d N”, program meminta kita memasukkan nilai N yang dibaca menggunakan `masukan.nextInt()` lalu dijalankan perulangan

for dimana nilai awal $i = 1$, perulangan berjalan dengan batasan $i \leq N$, dalam hal ini saya menginputkan N nya sebanyak 3. Lalu setiap iterasi mencetak nilai i menggunakan `System.out.println(i)` hingga angka 3. Setelah perulangan selesai, program mencetak "Akhir program" yang menandakan bahwa eksekusinya telah berakhir.

11. Output:

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

Penjelasan:

Program `PrintIterasi` mengimplementasikan pencetakan bilangan dari 1 hingga N berdasarkan input yang dimasukkan. Dalam program ini, dideklarasikan variabel i digunakan sebagai penghitung dalam perulangan (`for`), sedangkan variabel N menyimpan batas angka yang akan dicetak. Dideklarasikan juga objek `Scanner` dengan nama `masukan` untuk membaca input dari pengguna. Setelah itu, dicetak pesan "Nilai $N > 0 =$ ", program meminta kita memasukkan nilai N , yang dibaca menggunakan `masukan.nextInt()`. Setelah itu, nilai awal $i = 1$, lalu program menjalankan perulangan `for(;;)`, yang merupakan infinite loop tanpa kondisi eksplisit. Dalam perulangan ini, setiap iterasi mencetak nilai i menggunakan `System.out.println(i)`. Jika i sudah sama dengan N , perulangan dihentikan dengan `break`. Jika belum, nilai i bertambah satu ($i++$). Dalam hal ini, saya menginputkan $N = 4$, program akan mencetak angka 1, 2, 3, dan 4 kemudian berhenti.

12. Output:

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

Penjelasan:

Program `PrintRepeat` mengimplementasikan pencetakan bilangan dari 1 hingga N menggunakan perulangan `do-while` berdasarkan input yang dimasukkan. Dalam program ini, dideklarasikan variabel i digunakan sebagai penghitung dalam perulangan, sedangkan variabel N menyimpan batas angka yang akan dicetak. Dideklarasikan juga objek `Scanner` dengan nama `masukan` untuk membaca input dari pengguna. Setelah mencetak pesan "Nilai $N > 0 =$ ", program meminta kita sebagai pengguna memasukkan nilai N , yang dibaca menggunakan `masukan.nextInt()`. Setelah itu, nilai awal $i = 1$, lalu program menjalankan perulangan `do-while`, yang memastikan bahwa perintah dalam blok `do` dieksekusi minimal satu kali sebelum mengevaluasi kondisi ($i \leq N$). Dalam perulangan ini, setiap iterasi mencetak nilai i menggunakan `System.out.print(i + "\n")`, kemudian menaikkan nilai i dengan $i++$. Perulangan terus berjalan hingga nilai i melebihi N , lalu program berhenti. Dalam hal ini, saya memasukkan $N = 5$, program akan mencetak angka 1, 2, 3, 4, dan 5, kemudian berhenti.

13. Output:

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

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

Penjelasan:

Program PrintWhile mengimplementasikan pencetakan bilangan dari 1 hingga N menggunakan perulangan while berdasarkan input yang dimasukkan. Dalam program ini, variabel i digunakan sebagai penghitung dalam perulangan, sedangkan variabel N menyimpan batas angka yang akan dicetak. Dideklarasikan juga objek Scanner dengan nama masukan untuk membaca input yang kita inputkan. Setelah mencetak pesan "Nilai N > 0 =", program meminta pengguna memasukkan nilai N, yang dibaca menggunakan masukan.nextInt(). Setelah itu, nilai awal i = 1, lalu program menjalankan perulangan while, yang akan terus berjalan selama kondisi ($i \leq N$) bernilai benar. Dalam perulangan ini, setiap iterasi mencetak nilai i menggunakan System.out.println(i), kemudian menaikkan nilai i dengan i++. Perulangan akan berhenti ketika nilai i melebihi N. Dalam hal ini, saya memasukkan N = 3, program akan mencetak angka 1, 2, dan 3, kemudian berhenti.

14. Output:

```
D:\Tugas PBO\Tugas 2>javac PrintWhile1.java

D:\Tugas PBO\Tugas 2>java PrintWhile1
Nilai N > 0 = 6
Print i dengan WHILE (ringkas):
1
2
3
4
5
6
```

Penjelasan:

Program PrintWhile1 mengimplementasikan pencetakan bilangan dari 1 hingga N menggunakan perulangan while dengan cara yang lebih ringkas. Dalam program ini, variabel i digunakan sebagai penghitung dalam perulangan dan langsung diinisialisasi dengan 1, sedangkan variabel N menyimpan batas angka yang akan dicetak. Objek Scanner dengan nama masukan digunakan untuk membaca input dari pengguna. Setelah mencetak pesan "Nilai N > 0 =", program meminta kita sebagai pengguna untuk memasukkan nilai N, yang dibaca menggunakan masukan.nextInt(). Program kemudian menjalankan perulangan while, yang akan terus berjalan selama kondisi ($i \leq N$) bernilai benar. Berbeda dengan program sebelumnya, dalam perulangan ini, ekspresi System.out.println(i++) digunakan untuk mencetak i dan langsung menaikkan nilainya dalam satu perintah. Perulangan akan berhenti ketika nilai i melebihi N. Dalam hal ini, saya memasukkan N = 6, program akan mencetak angka 1, 2, 3, 4, 5, dan 6 kemudian berhenti.

15. Output:

- PrintXinterasi


```
D:\Tugas PB0\Tugas 2>javac PrintXinterasi.java

D:\Tugas PB0\Tugas 2>java PrintXinterasi
Masukan nilai x (int), akhiri dg 999: 5
Masukan nilai x (int), akhiri dg 999: 145
Masukan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 150
```

- Kasus Kosong

```
D:\Tugas PB0\Tugas 2>javac PrintXinterasi.java

D:\Tugas PB0\Tugas 2>java PrintXinterasi
Masukan nilai x (int), akhiri dg 999: 999
Kasus kosong
Hasil penjumlahan = 0
```

Penjelasan:

Program PrintXinterasi mengimplementasikan proses penjumlahan angka yang dimasukkan pengguna hingga input 999 sebagai tanda berhenti. Dalam program ini, dideklarasikan variabel `x` untuk menyimpan angka yang dimasukkan, sementara variabel `Sum` digunakan untuk menyimpan hasil penjumlahan. Dideklarasikan juga objek `Scanner` dengan nama `masukan` untuk membaca input dari pengguna. Setelah mencetak pesan "Masukan nilai x (int), akhiri dg 999:", program meminta kita sebagai pengguna untuk memasukkan angka. Jika angka yang dimasukkan adalah 999, program langsung mencetak "Kasus kosong", yang berarti tidak ada angka yang dijumlahkan. Namun, jika angka pertama bukan 999, program akan menginisialisasi `Sum` dengan nilai angka pertama yang dimasukkan dan menjalankan perulangan `for(;;)`. Dalam perulangan ini, setiap angka yang dimasukkan akan dijumlahkan ke dalam `Sum`, lalu program kembali meminta input angka. Perulangan akan terus berjalan hingga pengguna memasukkan 999, yang menjadi syarat berhenti (`if (x == 999) break`). Setelah perulangan selesai, program mencetak hasil penjumlahan menggunakan "Hasil penjumlahan = `Sum`". Dalam hal ini saya memasukkan angka 5, 145, lalu memasukkan 999, maka program akan menjumlahkan angka selain 999 dan mencetak "Hasil penjumlahan = 150" sebelum berakhir lalu saya juga mencoba menginputkan langsung "999" sehingga output yang dihasilkan adalah "Status Kosong" dan "Hasil Penjumlahan = 0".

16. Output:

- PrintXRepeat

```
D:\Tugas PB0\Tugas 2>javac PrintXRepeat.java

D:\Tugas PB0\Tugas 2>java PrintXRepeat
Masukan nilai x (int), akhiri dg 999: 12
Masukan nilai x (int), akhiri dg 999: 34
Masukan nilai x (int), akhiri dg 999: 56
Masukan nilai x (int), akhiri dg 999: 99
Masukan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 201
```

- Kasus Kosong

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

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

Penjelasan :

Program PrintXRepeat mengimplementasikan proses penjumlahan angka yang dimasukkan pengguna hingga input 999 sebagai tanda berhenti. Dalam program ini, dideklarasikan variabel *x* untuk menyimpan angka yang dimasukkan, sementara variabel *Sum* digunakan untuk menyimpan hasil penjumlahan. Dideklarasikan juga objek *Scanner* dengan nama *masukan* untuk membaca input dari pengguna. Setelah mencetak pesan "Masukan nilai x (int), akhiri dg 999:", program meminta kita sebagai pengguna untuk memasukkan angka. Jika angka yang dimasukkan adalah 999, program langsung mencetak "Kasus kosong", yang berarti tidak ada angka yang dijumlahkan. Namun, jika angka pertama bukan 999, program akan menginisialisasi *Sum* = 0 dan menjalankan perulangan *do-while*. Dalam perulangan ini, setiap angka yang dimasukkan akan dijumlahkan ke dalam *Sum*, lalu program kembali meminta input angka. Perulangan akan terus berjalan hingga pengguna memasukkan 999, yang menjadi syarat berhenti (*while* (*x* != 999)). Setelah perulangan selesai, program mencetak hasil penjumlahan menggunakan "Hasil penjumlahan = *Sum*". Dalam hal ini saya memasukkan angka 12, 34, 56, 99 lalu memasukkan 999, maka program akan menjumlahkan angka selain dari 999 dan mencetak "Hasil penjumlahan = 201" sebelum berakhir. Saya juga mencoba menginputkan langsung "999" sehingga output yang dihasilkan adalah "Status Kosong".

17. Output:

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

D:\Tugas PBO\Tugas 2>java PrintXWhile
Masukan nilai x (int), akhiri dg 999: 8
Masukan nilai x (int), akhiri dg 999: 5
Masukan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 13

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

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

Penjelasan:

Program PrintXWhile mengimplementasikan proses penjumlahan angka yang dimasukkan pengguna hingga input 999 sebagai tanda berhenti. Dalam program ini, dideklarasikan variabel *x* untuk menyimpan angka yang dimasukkan, sementara variabel *Sum* digunakan untuk menyimpan hasil penjumlahan. Objek *Scanner* dengan nama *masukan* digunakan untuk membaca inputan. Setelah mencetak pesan "Masukan nilai x (int), akhiri dg 999:", program meminta pengguna untuk memasukkan angka. Jika angka yang dimasukkan bukan 999, program akan menambahkan angka tersebut ke dalam *Sum* dan kembali meminta input angka. Perulangan *while* (*x* != 999) akan terus berjalan selama input bukan 999. Jika pengguna memasukkan 999, perulangan berhenti, dan program mencetak hasil penjumlahan

menggunakan "Hasil penjumlahan = Sum". Dalam hal ini saya memasukkan angka 8, 5, lalu memasukkan 999, maka program akan menjumlahkan angka selain 999 dan mencetak "Hasil penjumlahan = 13" sebelum berakhir. Saya juga mencoba menginputkan langsung "999" sehingga output yang dihasilkan adalah "Hasil Penjumlahan: 0".

18. Output:

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

D:\Tugas PBO\Tugas 2>java SubProgram
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
2 7
Ke dua bilangan : a = 2 b : 7
Maksimum = 7
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 7 b = 2
```

Penjelasan:

Program SubProgram mengimplementasikan pencarian nilai maksimum dari dua bilangan serta pertukaran nilainya. Didalam program dilakukan pendeklarasian dua variabel yaitu a dan b untuk menyimpan inputan dari pengguna, sementara objek Scanner dengan nama masukan digunakan untuk membaca input. Setelah kita menginputkn dua bilangan, maka program akan mencetak nilai keduanya dan menentukan bilangan terbesar menggunakan fungsi tukar(a, b), yang bertujuan untuk menukar kedua bilangan tersebut. Namun, karena Java menggunakan passing by value untuk data primitif, perubahan dalam fungsi tukar() tidak mempengaruhi nilai asli a dan b dalam main(). Dalam hal ini saya memasukkan angka a=2 dan b=7, maka program akan mencetak "Maksimum = 8", dan setelah pemanggilan fungsi tukar() bertujuan untuk menukar kedua bilangan.

19. Output:

- Wujud Air Beku

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

D:\Tugas PBO\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = -78
Wujud air beku
-78
```

- Wujud Air Cair

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

D:\Tugas PBO\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 75
Wujud air cair
75
```

- Wujud Air Uap/Gas

```
D:\Tugas PB0\Tugas 2>javac Tempair.java
```

```
D:\Tugas PB0\Tugas 2>java Tempair
```

```
Contoh IF tiga kasus
```

```
Temperatur (der. C) = 104
```

```
Wujud air uap/gas
```

```
104
```

Penjelasan :

Program Tempair mengimplementasikan percabangan if-else if-else untuk menentukan wujud air berdasarkan temperatur yang diinputkan pengguna. Dalam program ini mendeklarasikan variabel T dengan tipe data integer untuk menyimpan nilai temperatur dalam derajat Celcius dan objek Scanner untuk membaca inputan. Jika T kurang dari 0, program mencetak bahwa air berada dalam wujud beku. Jika T berada dalam rentang 0 hingga 100 derajat Celcius, air berada dalam wujud cair. Jika T lebih dari 100 derajat Celcius, air berada dalam wujud uap/gas. Jika T lebih dari 100 derajat Celcius, air berada dalam wujud uap/gas. Dalam hal ini saya menginputkan -78, programnya mencetak "Wujud Air Beku"; 75, programnya mencetak "Wujud Air Cair"; 104, programnya mencetak "Wujud Air Uap/Gas".