

**LAPORAN
PEMROGRAMAN BERBASIS OBJEK**



Nama : Audiawaliya Pratiwi Utami
Stambuk : 13020230258
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 PBO

1. Output:

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

Baca string dan Integer:
masukkan sebuah string: audi
String yang dibaca : audi
```

Penjelasan :

Program BacaString ini membaca inputan yang kita masukkan menggunakan library BufferedReader. Library IOException untuk menangani kesalahan input/output serta library InputStreamReader untuk mengonversi information input menjadi arrange yang bisa dibaca oleh program. Didalam program ini mendeklarasikan variabel str bertipe information String untuk menyimpan input. Objek BufferedReader datAIn dibuat untuk membaca input dari console. Setelah mencetak pesan "Baca string dan numbers: ", program ini akan meminta kita untuk memasukkan sebuah string. Disini saya menginputkan string "audi" 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 andi yang telah saya inputkan sebelumnya.

2. Output:

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

Penjelasan:

Program ForEver merupakan program dasar yang menggunakan loop while(true) untuk mencetak teks ke layar. Pada awalnya, program menampilkan pesan "Program akan looping, akhiri dengan ^c" sebagai informasi bahwa program seharusnya berjalan terus-menerus hingga dihentikan secara manual. Namun, di dalam loop while(true), terdapat perintah System.out.print("Print satu baris ...\n"); yang mencetak teks ke layar, diikuti dengan break;, yang langsung menghentikan perulangan. Dengan adanya break, program sebenarnya tidak berjalan tanpa henti seperti yang dikatakan sebelumnya, melainkan hanya mencetak satu baris teks lalu berhenti.

3. Output:

```
D:\Audiawaliya\Tugas PBO\Tugas 2>javac If1.java
D:\Audiawaliya\Tugas PBO\Tugas 2>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : 58

Nilai a positif 58
D:\Audiawaliya\Tugas PBO\Tugas 2>javac If1.java
D:\Audiawaliya\Tugas PBO\Tugas 2>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : -58
```

Penjelasan:

Program If1 mendemonstrasikan penggunaan percabangan if dalam Java untuk menentukan apakah suatu bilangan bernilai positif atau tidak. Program ini menggunakan Scanner untuk menerima input dari pengguna. Setelah menampilkan pesan "Contoh IF satu kasus", program meminta pengguna memasukkan sebuah bilangan bulat. Selanjutnya, program memeriksa nilai tersebut dengan kondisi if(a >= 0). Jika bilangan yang dimasukkan lebih dari atau sama dengan nol, program akan mencetak "Nilai a

positif:" diikuti dengan nilai a disini saya memasukkan 58. Namun, jika bilangan yang dimasukkan negatif, tidak ada output tambahan karena program tidak memiliki instruksi untuk menangani bilangan negatif disini saya memasukkan nilai -58 dan tidak ada output tambahan setelahnya.

4. Output:

```
D:\Audiawaliya\Tugas PBO\Tugas 2>javac If2.java
D:\Audiawaliya\Tugas PBO\Tugas 2>java If2
Contoh IF dua kasus

Ketikkan suatu nilai integer : 34
Nilai a positif 34

D:\Audiawaliya\Tugas PBO\Tugas 2>javac If2.java
D:\Audiawaliya\Tugas PBO\Tugas 2>java If2
Contoh IF dua kasus

Ketikkan suatu nilai integer : -34
Nilai a negatif -34
```

Penjelasan:

Program If2 menggunakan struktur percabangan if-else dalam Java untuk menentukan apakah angka yang dimasukkan bernilai positif atau negatif. Seperti pada program If1, program ini juga menggunakan Scanner untuk membaca input dari pengguna. Setelah menampilkan pesan "Contoh IF dua kasus", pengguna diminta memasukkan sebuah bilangan bulat, yang kemudian disimpan dalam variabel a. Program kemudian mengevaluasi nilai a menggunakan kondisi if ($a \geq 0$). Jika angka yang dimasukkan lebih dari atau sama dengan nol, program mencetak "Nilai a positif" diikuti dengan nilai a. Sebaliknya, jika angka tersebut negatif, blok else akan dieksekusi dan mencetak "Nilai a negatif" diikuti dengan nilai a. Saat saya memasukkan 34, program menampilkan "Nilai a positif: 34", sedangkan untuk -34, program mencetak "Nilai a negatif: -34".

5. Output:

```
D:\Audiawaliya\Tugas PBO\Tugas 2>javac If3.java
D:\Audiawaliya\Tugas PBO\Tugas 2>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer : 123
Nilai a positif 123

D:\Audiawaliya\Tugas PBO\Tugas 2>javac If3.java
D:\Audiawaliya\Tugas PBO\Tugas 2>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer : -25
Nilai a negatif -25

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

Penjelasan:

Program If3 menerapkan struktur percabangan if-else if-else dalam Java untuk menentukan apakah angka yang dimasukkan bernilai positif, nol, atau negatif. Seperti pada program If2, program ini menggunakan Scanner untuk membaca input dari pengguna. Setelah menampilkan pesan "Contoh IF tiga kasus", pengguna diminta memasukkan sebuah bilangan bulat yang disimpan dalam variabel a.

Program kemudian mengevaluasi nilai *a* dengan tiga kondisi: jika $a > 0$, program mencetak "Nilai *a* positif" diikuti dengan angka yang dimasukkan. Jika $a == 0$, program menampilkan "Nilai Nol" beserta angka tersebut. Jika $a < 0$, blok *else* dijalankan untuk mencetak "Nilai *a* negatif" diikuti oleh nilai *a*.

Saat saya memasukkan 123, program menampilkan "Nilai *a* positif: 123". Untuk -25, program mencetak "Nilai *a* negatif: -25", sedangkan jika saya memasukkan 0, program menampilkan "Nilai Nol: 0".

6. Output:

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

Penjelasan :

Program *KasusBoolean* menerapkan tipe data Boolean dalam struktur *if-else* di Java. Pada awalnya, program mendeklarasikan variabel *bool* bertipe Boolean dan menginisialisasinya dengan *true*.

Program kemudian melakukan dua evaluasi. Pada kondisi pertama, jika *if(bool)* dieksekusi, program mencetak "true", sementara jika *bool* bernilai *false*, program akan mencetak "false" dalam blok *else*. Pada evaluasi kedua, kondisi *if(!bool)* menggunakan operator negasi (!) yang membalikkan nilai *bool*, sehingga jika *bool* awalnya *true*, *!bool* menjadi *false*, dan program melewati blok *if*, lalu menjalankan *else* yang mencetak "benar". Jika *bool* awalnya *false*, program akan menghasilkan output "salah".

7. Output:

```
D:\Audiawaliya\Tugas PBO\Tugas 2>javac KasusSwitch.java
D:\Audiawaliya\Tugas PBO\Tugas 2>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
a
Yang anda ketik adalah a
D:\Audiawaliya\Tugas PBO\Tugas 2>javac KasusSwitch.java
D:\Audiawaliya\Tugas PBO\Tugas 2>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
i
Yang anda ketik adalah i
D:\Audiawaliya\Tugas PBO\Tugas 2>javac KasusSwitch.java
D:\Audiawaliya\Tugas PBO\Tugas 2>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
u
Yang anda ketik adalah u
D:\Audiawaliya\Tugas PBO\Tugas 2>javac KasusSwitch.java
D:\Audiawaliya\Tugas PBO\Tugas 2>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
RETURN
Yang anda ketik adalah huruf mati
```

Penjelasan:

Program *KasusSwitch* menerapkan struktur *switch-case* dalam Java untuk mengidentifikasi karakter yang diinput pengguna. Program mendeklarasikan variabel *cc* bertipe *char* dan menggunakan *Scanner* untuk membaca input.

Setelah menampilkan pesan "Silakan masukkan sebuah huruf, kemudian tekan RETURN", program mengambil karakter pertama dari input dengan *masukan.next().charAt(0)*.

Kemudian, switch-case digunakan untuk memeriksa apakah karakter yang dimasukkan merupakan huruf vokal kecil ('a', 'u', 'e', 'i', atau 'o'). Jika cocok, program mencetak pesan sesuai huruf yang dimasukkan. Jika bukan vokal, blok default akan dijalankan, menampilkan "Yang anda ketik adalah huruf mati", menunjukkan bahwa karakter tersebut adalah konsonan. Program ini menunjukkan bagaimana switch-case dapat menangani berbagai kemungkinan input dengan lebih efisien dibandingkan if-else.

8. Output:

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

D:\Audiawaliya\Tugas PBO\Tugas 2>java Konstant
Jari-jari lingkaran = 145
Luas lingkaran = 66050.04
Akhir program
```

Penjelasan:

Program ini menghitung luas lingkaran berdasarkan jari-jari yang dimasukkan oleh pengguna. Konstanta PHI dideklarasikan dengan nilai 3.1415f menggunakan final, sehingga nilainya tetap selama program berjalan. Variabel r bertipe float digunakan untuk menyimpan nilai jari-jari.

Program kemudian meminta pengguna memasukkan jari-jari dengan menampilkan pesan "Jari-jari lingkaran = ". Misalnya, jika pengguna memasukkan 145, nilai tersebut dibaca menggunakan `masukan.nextFloat()` dan disimpan dalam r.

Setelah itu, program menghitung luas lingkaran dengan rumus $\text{PHI} \times r \times r$ dan mencetak hasilnya, yaitu 66051.0125. Sebagai penutup, program menampilkan pesan "Akhir program", menandakan proses telah selesai.

9. Output:

```
D:\Audiawaliya\Tugas PBO\Tugas 2>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dengan RETURN :
6 8
Ke dua bilangan : a = 6 b = 8
Nilai b yang maksimum: 8

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

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

Penjelasan :

Program Max2 menentukan bilangan terbesar dari dua angka yang dimasukkan oleh pengguna. Di dalam program ini, terdapat dua variabel, yaitu a dan b, yang bertipe integer untuk menyimpan bilangan bulat. Selain itu, dibuat juga objek Scanner bernama masukan untuk membaca input dari pengguna.

Program diawali dengan mencetak pesan "Maksimum dua bilangan", lalu meminta pengguna memasukkan dua bilangan secara terpisah dengan menekan ENTER setelah setiap angka. Nilai yang dimasukkan dibaca menggunakan `masukan.nextInt()` dan disimpan ke dalam variabel a dan b.

Selanjutnya, program membandingkan kedua bilangan menggunakan struktur if-else. Jika a lebih besar atau sama dengan b, program akan mencetak "Nilai a yang

maksimum: a". Namun, jika b lebih besar, maka program mencetak "Nilai b yang maksimum: b". Dengan cara ini, program dapat menentukan dan menampilkan bilangan terbesar dari dua angka yang diinputkan.

Misalnya ketika saya memasukkan 9 untuk a dan 3 untuk b, maka program menentukan bahwa a adalah bilangan terbesar. Sebaliknya, jika pengguna memasukkan 6 untuk a dan 8 untuk b, maka b yang menjadi nilai maksimum.

10. Output:

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

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

Penjelasan:

Program PriFor mencetak bilangan dari 1 hingga N berdasarkan input yang diberikan pengguna. Variabel i digunakan sebagai penghitung dalam perulangan for, sedangkan N menyimpan batas angka yang akan dicetak. Program juga menggunakan Scanner untuk membaca input dari pengguna.

Setelah menampilkan pesan "Baca N, print 1 s/d N", program meminta pengguna memasukkan nilai N, yang kemudian dibaca dengan `masukan.nextInt()`. Selanjutnya, perulangan for dijalankan, dimulai dari $i = 1$ dan terus bertambah hingga $i \leq N$.

Misalnya, jika pengguna memasukkan $N = 5$, maka program mencetak angka 1, 2, 3, 4, dan 5 menggunakan `System.out.println(i)` di setiap iterasi. Setelah perulangan selesai, program menampilkan pesan "Akhir program" sebagai tanda bahwa eksekusi telah selesai.

11. Output:

```
D:\Audiawaliya\Tugas PBO\Tugas 2>javac PrintIterasi.java

D:\Audiawaliya\Tugas PBO\Tugas 2>java PrintIterasi
Nilai N > 0 = 9
Print i dengan ITERATE :
1
2
3
4
5
6
7
8
9
```

Penjelasan:

Program PrintIterasi mencetak bilangan dari 1 hingga N berdasarkan input yang diberikan oleh pengguna. Variabel i digunakan sebagai penghitung dalam perulangan, sedangkan N menyimpan batas angka yang akan dicetak. Program juga menggunakan Scanner untuk membaca input dari pengguna.

Setelah menampilkan pesan "Nilai N > 0 =", program meminta pengguna memasukkan nilai N, yang kemudian dibaca menggunakan `masukan.nextInt()`. Selanjutnya, program menginisialisasi `i = 1` dan menjalankan perulangan `for` tanpa kondisi eksplisit (infinite loop).

Di dalam perulangan, setiap iterasi mencetak nilai `i` menggunakan `System.out.println(i)`. Jika nilai `i` telah mencapai N, perulangan dihentikan dengan `break`. Jika belum, `i` bertambah satu (`i++`) dan iterasi berlanjut.

Sebagai contoh, jika pengguna memasukkan N = 9, program akan mencetak angka 1, 2, 3, 4, 5, 6, 7, 8, dan 9, lalu perulangan berhenti.

12. Output:

```
D:\Audiawaliya\Tugas PBO\Tugas 2>javac PrintRepeat.java
D:\Audiawaliya\Tugas PBO\Tugas 2>java PrintRepeat
Nilai N > 0 = 12
Print i dengan REPEAT:
1
2
3
4
5
6
7
8
9
10
11
12
```

Penjelasan:

Program `PrintRepeat` mencetak bilangan dari 1 hingga N menggunakan perulangan `do-while`, berdasarkan input pengguna. Program ini mendeklarasikan variabel `i` sebagai penghitung dan N sebagai batas angka yang akan dicetak, dengan `Scanner` digunakan untuk membaca input. Setelah menampilkan pesan "Nilai N > 0 =", program meminta pengguna memasukkan N, yang dibaca menggunakan `masukan.nextInt()`. Variabel `i` diinisialisasi dengan 1, lalu perulangan `do-while` dijalankan, memastikan blok `do` dieksekusi setidaknya sekali sebelum mengevaluasi kondisi (`i ≤ N`). Dalam perulangan, program mencetak `i` menggunakan `System.out.print(i + "\n")`, lalu menaikkan `i` dengan `i++`. Perulangan terus berjalan hingga `i` melebihi N, setelah itu program berhenti. Jika pengguna memasukkan N = 12, program akan mencetak angka 1 hingga 12, lalu perulangan berhenti.

13. Output:

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

Penjelasan:

Program `PrintWhile` menampilkan bilangan dari 1 hingga N menggunakan perulangan `while`, berdasarkan input pengguna. Variabel `i` berfungsi sebagai penghitung, sedangkan N menyimpan batas angka yang akan dicetak. `Scanner` digunakan untuk membaca input dari pengguna. Setelah menampilkan pesan "Nilai N > 0 =", program meminta pengguna memasukkan N, yang dibaca melalui `masukan.nextInt()`. Variabel `i` diinisialisasi dengan 1, lalu perulangan `while` dijalankan selama kondisi (`i ≤ N`) bernilai `true`. Setiap iterasi

mencetak *i* menggunakan `System.out.println(i)`, lalu nilai *i* bertambah dengan `i++`. Perulangan berakhir saat *i* melebihi *N*. Jika pengguna memasukkan *N* = 4, program akan mencetak angka 1, 2, 3, dan 4, kemudian berhenti.

14. Output:

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

Penjelasan:

Program `PrintWhile1` mencetak bilangan dari 1 hingga *N* menggunakan perulangan `while` dengan cara yang lebih ringkas. Variabel *i* berfungsi sebagai penghitung dan langsung diinisialisasi dengan 1, sementara *N* menyimpan batas angka yang akan dicetak. Objek `Scanner` digunakan untuk membaca input dari pengguna. Setelah menampilkan pesan "Nilai *N* > 0 =", program meminta pengguna memasukkan *N*, yang kemudian dibaca melalui `masukan.nextInt()`. Perulangan `while` dijalankan selama kondisi (*i* ≤ *N*) bernilai `true`. Berbeda dari versi sebelumnya, perulangan ini menggunakan `System.out.println(i++)`, yang mencetak nilai *i* dan langsung menambahkannya dalam satu perintah. Perulangan berakhir ketika *i* melebihi *N*. Jika pengguna memasukkan *N* = 4, program akan mencetak angka 1, 2, 3, dan 4, lalu berhenti.

15. Output:

```
D:\Audiawaliya\Tugas PBO\Tugas 2>javac PrintXinterasi.java
D:\Audiawaliya\Tugas PBO\Tugas 2>java PrintXinterasi
Masukan nilai x (int), akhiri dg 999: 98
Masukan nilai x (int), akhiri dg 999: 12
Masukan nilai x (int), akhiri dg 999: 34
Masukan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 144

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

Penjelasan:

Program `PrintXinterasi` menjumlahkan angka yang dimasukkan oleh pengguna hingga 999 dimasukkan sebagai tanda berhenti. Variabel *x* digunakan untuk menyimpan input angka, sementara variabel `Sum` digunakan untuk menyimpan hasil penjumlahan. Setelah mencetak pesan "Masukkan nilai *x* (int), akhiri dengan 999:", program meminta pengguna untuk memasukkan angka. Jika angka pertama yang dimasukkan adalah 999, program langsung mencetak "Kasus kosong" dan menampilkan "Hasil penjumlahan = 0". Namun, jika angka pertama bukan 999, nilainya akan disimpan dalam `Sum`, lalu perulangan `for(;;)` berjalan, menambahkan setiap angka ke `Sum` hingga 999 dimasukkan, yang kemudian menghentikan perulangan.

Jika pengguna memasukkan 98, 12, 34, 999, program menjumlahkan angka-angka tersebut dan mencetak "Hasil penjumlahan = 144". Jika pengguna langsung memasukkan 999, program mencetak "Kasus kosong" dan "Hasil penjumlahan = 0".

16. Output:


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

D:\Audiawaliya\Tugas PBO\Tugas 2>java PrintXRepeat
Masukan nilai x (int), akhiri dg 999: 2
Masukan nilai x (int), akhiri dg 999: 3
Masukan nilai x (int), akhiri dg 999: 4
Masukan nilai x (int), akhiri dg 999: 12
Masukan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 21
```

Penjelasan :

Program PrintXRepeat menjumlahkan angka yang dimasukkan oleh pengguna hingga 999 dimasukkan sebagai tanda berhenti. Variabel x menyimpan angka yang diinput, sementara Sum digunakan untuk menyimpan hasil penjumlahan. Program menggunakan objek Scanner untuk membaca input dari pengguna.

Setelah mencetak pesan "Masukkan nilai x (int), akhiri dengan 999:", program meminta pengguna memasukkan angka. Jika angka pertama yang dimasukkan adalah 999, program mencetak "Kasus kosong". Jika tidak, program menginisialisasi Sum ke 0, lalu menjalankan perulangan do-while yang terus menambahkan angka ke Sum hingga pengguna memasukkan 999, yang menghentikan perulangan (while (x != 999)). Setelah perulangan selesai, program mencetak "Hasil penjumlahan = Sum".

Jika pengguna memasukkan 2, 3, 4, 12, 999, maka program akan mencetak "Hasil penjumlahan = 21". Jika pengguna langsung memasukkan 999, program mencetak "Kasus kosong".

17. Output:

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

D:\Audiawaliya\Tugas PBO\Tugas 2>java PrintXWhile
Masukan nilai x (int), akhiri dg 999: 21
Masukan nilai x (int), akhiri dg 999: 23
Masukan nilai x (int), akhiri dg 999: 45
Masukan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 89
```

Penjelasan:

Jika pengguna memasukkan 21, 23, 45, program PrintXWhile akan menjumlahkan angka tersebut hingga pengguna memasukkan 999 sebagai tanda berhenti. Program menggunakan variabel x untuk menyimpan input angka dan Sum untuk menyimpan hasil penjumlahan. Scanner digunakan untuk membaca input dari pengguna.

Setelah mencetak pesan "Masukkan nilai x (int), akhiri dengan 999:", program mulai menerima angka. Selama input bukan 999, angka tersebut ditambahkan ke Sum, dan program terus meminta input baru. Perulangan while (x != 999) berjalan hingga pengguna memasukkan 999, lalu program mencetak "Hasil penjumlahan = Sum".

Jika pengguna memasukkan 21, 23, 45, 999, maka program akan mencetak "Hasil penjumlahan = 89". Jika pengguna langsung memasukkan 999, program akan mencetak "Hasil penjumlahan = 0".

18. Output:

```

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

D:\Audiawaliya\Tugas PBO\Tugas 2>java SubProgram
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
3 8
Ke dua bilangan : a = 3 b : 8
Maksimum = 8
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 8 b = 3

```

Penjelasan:

Program SubProgram menentukan nilai maksimum dari dua bilangan yang dimasukkan pengguna dan mencoba menukarnya. Variabel a dan b digunakan untuk menyimpan input, sementara Scanner membaca data dari pengguna.

Setelah menerima input, program mencetak nilai awal a dan b, lalu membandingkannya untuk menemukan bilangan terbesar. Dalam kasus ini, karena b (8) lebih besar dari a (3), program menampilkan "Maksimum = 8".

Kemudian, program memanggil fungsi tukar(a, b) untuk menukar nilai, tetapi karena Java menggunakan passing by value untuk tipe data primitif, perubahan hanya terjadi di dalam fungsi dan tidak memengaruhi nilai asli di dalam main(). Akibatnya, setelah fungsi selesai dieksekusi, a tetap 3 dan b tetap 8.

19. Output:

```

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

D:\Audiawaliya\Tugas PBO\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 12
Wujud air cair
12
D:\Audiawaliya\Tugas PBO\Tugas 2>javac Tempair.java

D:\Audiawaliya\Tugas PBO\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = -20
Wujud air beku
-20
D:\Audiawaliya\Tugas PBO\Tugas 2>javac Tempair.java

D:\Audiawaliya\Tugas PBO\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 145
Wujud air uap/gas
145

```

Penjelasan :

Program Tempair menentukan wujud air berdasarkan suhu yang dimasukkan pengguna menggunakan struktur if-else if-else. Variabel T bertipe integer menyimpan suhu dalam derajat Celcius, sementara Scanner membaca input pengguna. Jika $T < 0$, program menampilkan "Wujud Air Beku". Jika T berada di antara 0 dan 100 (termasuk 0 dan 100), program mencetak "Wujud Air Cair". Jika $T > 100$, program menampilkan "Wujud Air Uap/Gas". Dengan input 12, program menampilkan "Wujud Air Cair". Untuk -20, outputnya "Wujud Air Beku", dan untuk 145, program mencetak "Wujud Air Uap/Gas".