

**LAPORAN  
PEMROGRAMAN BERBASIS OBJEK**



Nama : Elva Ruhustina  
Stambuk : 13020230288  
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:\Eva\Tugas PBO\Tugas 2>javac BacaString.java
D:\Eva\Tugas PBO\Tugas 2>java BacaString
Baca string dan Integer:
masukkan sebuah string: Elva
String yang dibaca : Elva
```

Penjelasan :

Program BacaString menggunakan library BufferedReader untuk membaca input dari pengguna. Library IOException digunakan untuk menangani kesalahan input/output, sedangkan InputStreamReader berfungsi mengonversi data masukan ke format yang dapat diproses oleh program.

Dalam program ini, variabel str bertipe String dideklarasikan untuk menyimpan input. Objek BufferedReader bernama datAIn dibuat untuk membaca masukan dari konsol. Setelah menampilkan pesan "Baca string dan numbers:", program meminta pengguna memasukkan sebuah string.

Sebagai contoh, ketika saya memasukkan string "Elva", program membacanya menggunakan datAIn.readLine() dan menyimpannya dalam variabel str. Terakhir, program mencetak kembali string yang telah dimasukkan sebelumnya, dalam hal ini "Elva".

### 2. Output:

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

Penjelasan:

Program ForEver menggunakan perulangan while(true) yang seharusnya berjalan tanpa henti. Saat dijalankan, program terlebih dahulu menampilkan pesan "Program akan looping, akhiri dengan ^c", memberi tahu pengguna bahwa program dirancang untuk terus berjalan hingga dihentikan secara manual.

Namun, di dalam perulangan while(true), terdapat perintah System.out.print("Print satu baris ...\n"); yang mencetak teks ke layar, diikuti oleh break; yang langsung menghentikan perulangan. Akibatnya, meskipun sekilas terlihat seperti program akan berjalan selamanya, sebenarnya program hanya mencetak satu baris teks dan kemudian berhenti.

### 3. Output:

```
D:\Eva\Tugas PBO\Tugas 2>javac If1.java
D:\Eva\Tugas PBO\Tugas 2>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : 98
Nilai a positif 98
D:\Eva\Tugas PBO\Tugas 2>javac If1.java
D:\Eva\Tugas PBO\Tugas 2>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : -98
```

Penjelasan:

Program If1 mendemonstrasikan penggunaan percabangan if dalam Java untuk menentukan apakah suatu bilangan bersifat positif atau tidak. Program ini memanfaatkan Scanner untuk membaca input dari pengguna.

Setelah menampilkan pesan "Contoh IF satu kasus", program meminta pengguna memasukkan sebuah bilangan bulat. Kemudian, program mengevaluasi nilai tersebut dengan kondisi  $\text{if}(a \geq 0)$ . Jika bilangan yang dimasukkan lebih dari atau sama dengan nol, program mencetak "Nilai a positif:" diikuti dengan nilai a.

Sebagai contoh, ketika saya memasukkan 98, program akan menampilkan output tersebut. Namun, jika bilangan yang dimasukkan negatif, program tidak memberikan output tambahan karena tidak ada instruksi untuk menangani bilangan negatif. Misalnya, saat saya memasukkan -98, program tidak mencetak apa pun setelahnya.

#### 4. Output:

```
D:\Eva\Tugas PB0\Tugas 2>javac If2.java
D:\Eva\Tugas PB0\Tugas 2>java If2
Contoh IF dua kasus

Ketikkan suatu nilai integer : 67
Nilai a positif 67

D:\Eva\Tugas PB0\Tugas 2>javac If2.java
D:\Eva\Tugas PB0\Tugas 2>java If2
Contoh IF dua kasus

Ketikkan suatu nilai integer : -67
Nilai a negatif -67
```

Penjelasan:

Program If2 menerapkan percabangan if-else dalam Java untuk menentukan apakah suatu bilangan bersifat positif atau negatif. Program ini menggunakan Scanner untuk membaca input dari pengguna setelah menampilkan pesan "Contoh IF dua kasus".

Bilangan yang dimasukkan disimpan dalam variabel a dan dievaluasi menggunakan  $\text{if}(a \geq 0)$ . Jika nilai a nol atau lebih, program mencetak "Nilai a positif: 67". Sebaliknya, jika a bernilai negatif, blok else akan mencetak "Nilai a negatif: -67".

#### 5. Output:

```
D:\Eva\Tugas PB0\Tugas 2>javac If3.java
D:\Eva\Tugas PB0\Tugas 2>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer : 69
Nilai a positif 69

D:\Eva\Tugas PB0\Tugas 2>javac If3.java
D:\Eva\Tugas PB0\Tugas 2>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer : -69
Nilai a negatif -69

D:\Eva\Tugas PB0\Tugas 2>javac If3.java
D:\Eva\Tugas PB0\Tugas 2>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer : 0
Nilai Nol 0
```

Penjelasan:

Program If3 menggunakan percabangan if-else if-else dalam Java untuk menentukan apakah angka yang dimasukkan bernilai positif, nol, atau negatif. Program ini menggunakan Scanner untuk membaca input dari pengguna setelah menampilkan pesan "Contoh IF tiga kasus". Nilai yang dimasukkan disimpan dalam variabel a, kemudian dievaluasi berdasarkan tiga kondisi utama. Jika angka lebih dari nol, program mencetak "Nilai a: 69". Jika angka sama dengan nol, program akan menampilkan "Nilai a: 0". Sedangkan jika angka kurang dari nol, program mencetak "Nilai a: -69". Sebagai contoh, jika pengguna memasukkan 69, maka output yang ditampilkan adalah "Nilai a: 69". Jika pengguna memasukkan -69, maka program mencetak "Nilai a: -69". Sementara itu, jika pengguna memasukkan 0, program akan menampilkan "Nilai a: 0".

6. Output:

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

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

Penjelasan :

Program KasusBoolean menggunakan tipe data Boolean dalam struktur if-else di Java. Di awal program, variabel bool bertipe Boolean dideklarasikan dan diberikan nilai true. Program kemudian melakukan dua pengecekan kondisi. Pada evaluasi pertama, jika if(bool) dieksekusi, program mencetak "true". Namun, jika bool bernilai false, program akan masuk ke blok else dan mencetak "false". Pada evaluasi kedua, program menggunakan operator negasi (!) dalam kondisi if(!bool) untuk membalikkan nilai bool. Jika bool awalnya true, maka !bool menjadi false, sehingga program melewati blok if dan menjalankan else, yang mencetak "benar". Sebaliknya, jika bool bernilai false, program akan mencetak "salah".

7. Output:

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

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

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

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

Penjelasan:

Program KasusSwitch menerapkan struktur switch-case dalam Java untuk mengidentifikasi karakter yang dimasukkan oleh pengguna. Variabel cc bertipe char digunakan untuk menyimpan input, sementara Scanner digunakan untuk membaca data dari pengguna. Setelah menampilkan pesan "Silakan masukkan sebuah huruf, kemudian tekan RETURN", program mengambil karakter pertama dari input menggunakan masukan.next().charAt(0). Kemudian, switch-case memeriksa apakah karakter tersebut merupakan huruf vokal kecil ('a', 'u', 'e', 'i', atau 'o'). Jika cocok, program mencetak pesan yang sesuai dengan huruf tersebut.

Sebagai contoh, ketika pengguna memasukkan huruf 'i', program mengenali bahwa itu adalah vokal dan menampilkan pesan yang sesuai. Namun, jika karakter yang dimasukkan bukan

vokal, seperti 't', program masuk ke blok default dan mencetak "Yang Anda ketik adalah huruf mati", menandakan bahwa karakter tersebut merupakan konsonan.

8. Output:

```
D:\Eva\Tugas PB0\Tugas 2>javac Konstant.java

D:\Eva\Tugas PB0\Tugas 2>java Konstant
Jari-jari lingkaran = 53
Luas lingkaran = 8824.474
Akhir program
```

Penjelasan:

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

Program menampilkan pesan "Jari-jari lingkaran =" untuk meminta input dari pengguna. Jika pengguna memasukkan 53, nilai tersebut dibaca menggunakan `masukan.nextFloat()` dan disimpan dalam r. Setelah itu, program menghitung luas lingkaran menggunakan rumus  $\text{PHI} \times r \times r$  dan menampilkan hasilnya, yaitu 8815.1845. Sebagai penutup, program mencetak pesan "Akhir program", yang menandakan bahwa eksekusi telah selesai.

9. Output:

```
D:\Eva\Tugas PB0\Tugas 2>javac Max2.java

D:\Eva\Tugas PB0\Tugas 2>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dengan RETURN :
2 9
Ke dua bilangan : a = 2 b = 9
Nilai b yang maksimum: 9

D:\Eva\Tugas PB0\Tugas 2>javac Max2.java

D:\Eva\Tugas PB0\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 digunakan untuk menentukan bilangan terbesar dari dua angka yang dimasukkan oleh pengguna. Program ini mendeklarasikan dua variabel, a dan b, bertipe integer untuk menyimpan bilangan bulat, serta menggunakan objek Scanner bernama masukan untuk membaca input dari pengguna.

Setelah menampilkan pesan "Maksimum dua bilangan", program meminta pengguna memasukkan dua angka secara terpisah. Nilai tersebut kemudian dibaca menggunakan `masukan.nextInt()` dan disimpan dalam variabel a dan b.

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

Sebagai contoh, jika pengguna memasukkan a = 2 dan b = 9, program akan menampilkan "Nilai b yang maksimum: 9". Sebaliknya, jika pengguna memasukkan a = 9 dan b = 3, program mencetak "Nilai a yang maksimum: 9".

10. Output:

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

D:\Eva\Tugas PBO\Tugas 2>java PriFor
Baca N, print 1 s/d N N = 7
1
2
3
4
5
6
7
Akhir program
```

Penjelasan:

Program PriFor mencetak bilangan dari 1 hingga N berdasarkan input yang diberikan oleh pengguna. Variabel *i* digunakan sebagai penghitung dalam perulangan for, sedangkan N menyimpan batas angka yang akan dicetak. Program ini 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 menggunakan `masukan.nextInt()` dan disimpan dalam variabel N.

Selanjutnya, perulangan for dijalankan dengan inisialisasi *i* = 1, kemudian terus berjalan hingga *i* <= N, dengan *i* bertambah satu setiap iterasi. Di setiap langkah, `System.out.println(i)` digunakan untuk mencetak angka ke layar.

Sebagai contoh, jika pengguna memasukkan N = 7, program mencetak angka 1, 2, 3, 4, 5, 6, dan 7 secara berurutan. Setelah perulangan selesai, program menampilkan pesan "Akhir program", menandakan bahwa eksekusi telah berakhir.

11. Output:

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

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

Penjelasan:

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

Setelah menampilkan pesan "Nilai N > 0 =", program meminta pengguna memasukkan N, yang kemudian dibaca menggunakan `masukan.nextInt()` dan disimpan dalam variabel N.

Program menginisialisasi *i* = 1 dan menjalankan perulangan for tanpa kondisi eksplisit (perulangan tak terbatas). Dalam setiap iterasi, program mencetak nilai *i* menggunakan `System.out.println(i)`. Jika *i* sudah mencapai N, perulangan dihentikan menggunakan `break`. Jika belum, nilai *i* bertambah satu (*i*++) dan iterasi berlanjut.

Sebagai contoh, jika pengguna memasukkan  $N = 4$ , program mencetak angka 1, 2, 3, dan 4, lalu perulangan berhenti.

#### 12. Output:

```
D:\Eva\Tugas PBO\Tugas 2>javac PrintRepeat.java

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

Penjelasan:

Program PrintRepeat menampilkan bilangan dari 1 hingga  $N$  menggunakan perulangan do-while, berdasarkan input yang diberikan oleh pengguna. Variabel  $i$  digunakan sebagai penghitung, sedangkan  $N$  menentukan batas angka yang akan dicetak. Program ini juga menggunakan Scanner untuk membaca input.

Setelah menampilkan pesan "Nilai  $N > 0 =$ ", program meminta pengguna memasukkan  $N$ , yang kemudian dibaca menggunakan `masukan.nextInt()`. Variabel  $i$  diinisialisasi dengan 1, lalu perulangan do-while dijalankan. Perulangan ini memastikan bahwa blok do selalu dieksekusi setidaknya sekali sebelum memeriksa kondisi ( $i \leq N$ ).

Di dalam perulangan, program mencetak  $i$  menggunakan `System.out.print(i + "\n")`, kemudian menaikkan nilai  $i$  dengan  $i++$ . Perulangan akan terus berjalan hingga  $i$  melebihi  $N$ , setelah itu program berhenti.

Sebagai contoh, jika pengguna memasukkan  $N = 3$ , program mencetak angka 1, 2, dan 3, lalu perulangan berakhir.

#### 13. Output:

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

D:\Eva\Tugas PBO\Tugas 2>java PrintWhile
Nilai N > 0 = 8
Print i dengan WHILE:
1
2
3
4
5
6
7
8
```

Penjelasan:

Program PrintWhile digunakan untuk mencetak bilangan dari 1 hingga  $N$  menggunakan perulangan while, berdasarkan input dari pengguna. Variabel  $i$  berfungsi sebagai penghitung, sedangkan  $N$  menentukan batas angka yang akan ditampilkan. Program ini juga menggunakan Scanner untuk membaca input.

Setelah menampilkan pesan "Nilai  $N > 0 =$ ", program meminta pengguna memasukkan  $N$ , yang kemudian dibaca menggunakan `masukan.nextInt()`. Variabel  $i$  diinisialisasi dengan 1, lalu perulangan while berjalan selama kondisi ( $i \leq N$ ) masih terpenuhi.

Di dalam perulangan, program mencetak nilai  $i$  menggunakan `System.out.println(i)`, kemudian menaikkan  $i$  dengan  $i++$ . Perulangan akan berhenti ketika nilai  $i$  melebihi  $N$ .

Sebagai contoh, jika pengguna memasukkan  $N = 8$ , program akan mencetak angka 1, 2, 3, 4, 5, 6, 7, dan 8, lalu program selesai dijalankan.

#### 14. Output:

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

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

Penjelasan:

Program PrintWhile1 mencetak bilangan dari 1 hingga N menggunakan perulangan while dengan sintaks yang lebih ringkas. Variabel i berfungsi sebagai penghitung dan langsung diinisialisasi dengan 1, sedangkan N menentukan batas angka yang akan dicetak. Program ini juga menggunakan Scanner untuk menerima input dari pengguna.

Setelah menampilkan pesan "Nilai N > 0 =", program meminta pengguna memasukkan N, yang kemudian dibaca menggunakan masukan.nextInt(). Perulangan while berjalan selama kondisi ( $i \leq N$ ) masih terpenuhi.

Tidak seperti versi sebelumnya, perulangan ini menggunakan System.out.println(i++), yang mencetak nilai i lalu langsung menaikkannya dalam satu perintah. Perulangan akan berhenti saat i melebihi N.

Sebagai contoh, jika pengguna memasukkan  $N = 5$ , program akan mencetak angka 1, 2, 3, 4, dan 5, lalu program berakhir.

#### 15. Output:

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

D:\Eva\Tugas PBO\Tugas 2>java PrintXinterasi
Masukan nilai x (int), akhiri dg 999: 1
Masukan nilai x (int), akhiri dg 999: 3
Masukan nilai x (int), akhiri dg 999: 5
Masukan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 9
```

Penjelasan:

Program PrintXinterasi menghitung jumlah dari angka yang dimasukkan oleh pengguna hingga angka 999 digunakan sebagai penanda berhenti. Variabel x digunakan untuk menyimpan input angka, sementara Sum digunakan untuk menampung hasil penjumlahan.

Saat dijalankan, program menampilkan pesan "Masukkan nilai x (int), akhiri dengan 999:", kemudian meminta pengguna memasukkan angka. Jika angka pertama yang dimasukkan adalah 999, program langsung mencetak "Kasus kosong" dan menampilkan "Hasil penjumlahan = 0". Namun, jika angka yang dimasukkan bukan 999, nilai tersebut disimpan ke dalam Sum, lalu program memasuki perulangan for(;;) yang terus menjumlahkan angka hingga pengguna memasukkan 999, yang akan menghentikan perulangan.

Sebagai contoh, jika pengguna memasukkan 1, 3, 5, 999, program akan menjumlahkan angka-angka tersebut dan menampilkan "Hasil penjumlahan = 9". Jika pengguna langsung memasukkan 999, program akan mencetak "Kasus kosong" dan "Hasil penjumlahan = 0".



16. Output:

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

D:\Eva\Tugas PBO\Tugas 2>java PrintXRepeat
Masukan nilai x (int), akhiri dg 999: 2
Masukan nilai x (int), akhiri dg 999: 4
Masukan nilai x (int), akhiri dg 999: 6
Masukan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 12
```

Penjelasan :

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

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

Sebagai contoh, jika pengguna memasukkan 2, 4, 6, 999, maka program akan menjumlahkan angka-angka tersebut dan menampilkan "Hasil penjumlahan = 12". Jika pengguna langsung memasukkan 999, program akan mencetak "Kasus kosong".

17. Output:

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

D:\Eva\Tugas PBO\Tugas 2>java PrintXWhile
Masukan nilai x (int), akhiri dg 999: 74
Masukan nilai x (int), akhiri dg 999: 90
Masukan nilai x (int), akhiri dg 999: 87
Masukan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 251
```

Penjelasan:

Program PrintXWhile menjumlahkan angka yang dimasukkan oleh pengguna hingga angka 999 digunakan sebagai tanda berhenti. Program ini menggunakan variabel x untuk menyimpan input angka dan Sum untuk menampung hasil penjumlahan, serta memanfaatkan objek Scanner untuk membaca input dari pengguna.

Setelah menampilkan pesan "Masukkan nilai x (int), akhiri dengan 999:", program mulai menerima angka. Jika angka pertama yang dimasukkan adalah 999, program langsung mencetak "Hasil penjumlahan = 0" dan berhenti. Namun, jika angka yang dimasukkan bukan 999, nilainya langsung ditambahkan ke dalam Sum, kemudian perulangan while (x != 999) berjalan, di mana setiap angka yang dimasukkan akan terus dijumlahkan hingga 999 digunakan untuk menghentikan perulangan. Setelah itu, program menampilkan hasil akhir dengan format "Hasil penjumlahan = Sum".

Sebagai contoh, jika pengguna memasukkan 74, 90, 87, 999, program akan menjumlahkan angka-angka tersebut dan mencetak "Hasil penjumlahan = 251". Jika pengguna langsung memasukkan 999, program akan menampilkan "Hasil penjumlahan = 0".

18. Output:

```

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

D:\Eva\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 bilangan terbesar dari dua angka yang dimasukkan oleh pengguna dan mencoba menukarnya. Variabel a dan b digunakan untuk menyimpan input, sementara Scanner digunakan untuk membaca data dari pengguna.

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

Selanjutnya, program memanggil fungsi tukar(a, b) untuk menukar nilai kedua variabel. Namun, karena Java menerapkan passing by value untuk tipe data primitif, perubahan hanya terjadi di dalam fungsi tersebut tanpa memengaruhi nilai asli dalam main(). Akibatnya, setelah fungsi selesai dijalankan, nilai a tetap 3 dan b tetap 8.

#### 19. Output:

```

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

D:\Eva\Tugas PBO\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 23
Wujud air cair
23
D:\Eva\Tugas PBO\Tugas 2>javac Tempair.java

D:\Eva\Tugas PBO\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = -9
Wujud air beku
-9
D:\Eva\Tugas PBO\Tugas 2>javac Tempair.java

D:\Eva\Tugas PBO\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 124
Wujud air uap/gas
124

```

Penjelasan :

Program Tempair menentukan wujud air berdasarkan suhu yang dimasukkan oleh pengguna menggunakan struktur if-else if-else. Variabel T bertipe integer digunakan untuk menyimpan suhu dalam derajat Celcius, sementara Scanner berfungsi untuk membaca input dari pengguna.

Jika  $T < 0$ , program menampilkan "Wujud Air Beku". Jika T berada dalam rentang 0 hingga 100 (termasuk batasnya), program mencetak "Wujud Air Cair". Jika  $T > 100$ , program akan menampilkan "Wujud Air Uap/Gas".

Sebagai contoh, jika pengguna memasukkan 23, program mencetak "Wujud Air Cair". Untuk suhu -9, program menampilkan "Wujud Air Beku", sedangkan dengan input 124, outputnya adalah "Wujud Air Uap/Gas".