

Tugas 2

Makassar, 28 Februari 2025

**PRAKTIKUM  
PEMROGRAMAN BERORIENTASI OBJEK**



Nama : Hikmalia  
Stambuk : 13020230233  
Dosen : Mardiyah Hasnawi, S.Kom., MT,MTA

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

1. Program Java merupakan Contoh sederhana dari contoh penggunaan Metode `BufferedReader` dimana program ini berguna untuk membaca dan menampilkan kembali teks yang dimasukkan oleh pengguna. Saat dijalankan, program pertama-tama akan menampilkan pesan di layar yang meminta pengguna untuk mengetikkan sebuah string (teks). Setelah pengguna memasukkan teks, program akan membaca input tersebut dan langsung mencetaknya kembali dengan format "String yang dibaca: ". Meskipun dalam pesan awal terdapat kata "Integer", program ini sebenarnya hanya membaca input sebagai teks, sehingga meskipun pengguna mengetik angka, angka tersebut tetap dianggap sebagai teks dan tidak bisa digunakan dalam perhitungan matematika. Seperti pada contoh ketika mengetikkan kata seperti *Hikmalia*, program akan menampilkan hasil yang sama sesuai dengan input yang dimasukkan, maka akan menampilkan output seperti pada gambar di bawah ini.

**OUTPUT:**

```
D:\Tugas-PB0\Tugas2>javac BacaString.java

D:\Tugas-PB0\Tugas2>java BacaString

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

2. Program ini program sederhana dalam bahasa Java yang menggunakan Loop dengan perintah `while` (true) biasanya digunakan untuk membuat perulangan tanpa henti (infinite loop). Namun, pada program berikut ini hanya mencetak satu baris teks lalu langsung berhenti. Hal ini terjadi karena di dalam perulangan terdapat perintah ``break;``, yang langsung menghentikan proses setelah satu kali perulangan. Akibatnya, output yang muncul di layar hanya terdiri dari dua baris, yaitu pesan awal dan satu baris teks tambahan. Sehingga akan menampilkan output seperti pada gambar di bawah ini.

**OUTPUT:**

```
D:\Tugas-PB0\Tugas2>javac ForEver.java

D:\Tugas-PB0\Tugas2>java ForEver.java
Program akan looping, akhiri dengan ^c
Print satu baris
```

3. Program ini adalah contoh sederhana penggunaan struktur percabangan `if` dalam Java untuk mengecek apakah suatu angka yang dimasukkan oleh pengguna adalah bilangan positif. Pertama, program menampilkan pesan awal di layar dan meminta pengguna untuk mengetikkan sebuah angka. Setelah

pengguna memasukkan angka, program membacanya menggunakan objek Scanner dan menyimpannya dalam variabel. Selanjutnya, program memeriksa apakah angka tersebut lebih besar atau sama dengan nol. Jika ya, program akan mencetak bahwa angka tersebut adalah bilangan positif. Misalnya, jika kita mengetik angka 12, maka output yang muncul adalah "Nilai a positif 12". Namun, jika memasukkan angka negatif, program tidak akan menampilkan apa pun karena tidak ada perintah untuk menangani kondisi tersebut.

#### **OUTPUT:**

```
D:\Tugas-PBO\Tugas2>javac if1.java

D:\Tugas-PBO\Tugas2>java if1.java
Contoh IF satu kasus
Ketikkan suatu nilai integer : 12

Nilai a positif 12
```

4. Program ini merupakan contoh penggunaan percabangan if-else dalam Java untuk menentukan apakah sebuah bilangan yang dimasukkan oleh pengguna bersifat positif atau negatif. Pertama, program menampilkan pesan awal di layar dan meminta pengguna untuk memasukkan sebuah angka. Setelah mengetikkan angka, program membacanya menggunakan objek Scanner dan menyimpannya dalam variabel. Selanjutnya, program mengecek nilai angka tersebut menggunakan percabangan if-else. Jika angka lebih besar atau sama dengan nol, program akan mencetak bahwa angka tersebut adalah bilangan positif beserta nilainya. Sebaliknya, jika angka yang dimasukkan bernilai negatif, program akan mencetak bahwa angka tersebut adalah bilangan negatif. Sebagai contoh, jika memasukkan angka 20, program akan menampilkan "Nilai a positif 20", sedangkan jika angka yang dimasukkan adalah -20, program akan mencetak "Nilai a negatif -20".

#### **OUTPUT:**

```
D:\Tugas-PBO\Tugas2>javac if2.java

D:\Tugas-PBO\Tugas2>java if2.java
Contoh IF dua kasus
Ketikkan suatu nilai integer :20
Nilai a positif 20

D:\Tugas-PBO\Tugas2>javac if2.java

D:\Tugas-PBO\Tugas2>java if2.java
Contoh IF dua kasus
Ketikkan suatu nilai integer :-20
Nilai a negatif -20
```

5. Program ini merupakan contoh penggunaan percabangan if-else if-else dalam Java untuk mengklasifikasikan suatu bilangan sebagai bilangan positif, nol, atau negatif. Pertama, program menampilkan pesan awal yang memberi tahu

bahwa ini adalah contoh "IF tiga kasus". Selanjutnya, program meminta untuk memasukkan sebuah angka integer melalui keyboard. Setelah mengetikkan angka, program membacanya menggunakan objek Scanner dan menyimpannya dalam variabel a. Setelah angka tersimpan, program akan melakukan pengecekan menggunakan struktur if-else if-else. Jika angka yang dimasukkan lebih besar dari nol, program akan mencetak bahwa angka tersebut adalah bilangan positif, misalnya jika mengetik 29, maka program akan menampilkan "Nilai a positif 29". Jika angka yang dimasukkan adalah nol, program akan mencetak "Nilai Nol 0". Namun, jika angka yang dimasukkan lebih kecil dari nol, program akan mencetak bahwa angka tersebut adalah bilangan negatif, seperti jika mengetik -29, maka output yang muncul adalah "Nilai a negatif -29".

#### OUTPUT:

```
D:\Tugas-PBO\Tugas2>javac if3.java

D:\Tugas-PBO\Tugas2>java if3.java
Contoh IF tiga kasus
Ketikkan suatu nilai integer :0
Nilai Nol 0

D:\Tugas-PBO\Tugas2>javac if3.java

D:\Tugas-PBO\Tugas2>java if3.java
Contoh IF tiga kasus
Ketikkan suatu nilai integer :29
Nilai a positif 29

D:\Tugas-PBO\Tugas2>javac if3.java

D:\Tugas-PBO\Tugas2>java if3.java
Contoh IF tiga kasus
Ketikkan suatu nilai integer :-29
Nilai a negatif -29
```

6. Program ini adalah contoh sederhana dalam Java yang menggunakan tipe data boolean untuk mengevaluasi kondisi true atau false. Program diawali dengan mendeklarasikan variabel bool bertipe boolean, lalu variabel tersebut diberi nilai true. Setelah itu, program mengecek nilai bool menggunakan pernyataan if-else. Pada kondisi pertama, program memeriksa apakah bool bernilai true. Karena bool sudah diatur menjadi true, maka kondisi if(bool) akan terpenuhi dan program mencetak "true". Jika bool bernilai false, seharusnya yang dicetak adalah "false", tetapi dalam kasus ini, kondisi if terpenuhi sehingga hanya "true" yang muncul. Kemudian, pada kondisi kedua, program memeriksa nilai !bool, yang berarti "tidak bool" atau kebalikan dari nilai bool. Karena bool bernilai true, maka !bool menjadi false. Karena kondisi if(!bool) tidak terpenuhi, maka program masuk ke bagian else dan mencetak "benar".

#### OUTPUT:

```
D:\Tugas-PBO\Tugas2>javac KasusBoolean.java

D:\Tugas-PBO\Tugas2>java KasusBoolean.java
true
benar
```

7. Program ini dibuat menggunakan bahasa Java dan menerapkan struktur switch untuk menentukan jenis huruf yang diketik oleh pengguna. Program diawali dengan mengimpor pustaka Scanner agar dapat membaca input dari pengguna. Dalam fungsi main, program mendeklarasikan variabel cc dengan tipe char untuk menyimpan satu karakter yang dimasukkan. Saat dijalankan, diminta untuk mengetikkan satu huruf dan menekan Enter. Input yang diberikan akan diproses menggunakan `masukan.next().charAt(0)`, sehingga hanya karakter pertama yang digunakan. Selanjutnya, program mengevaluasi huruf tersebut menggunakan switch. Jika huruf yang diketik adalah salah satu huruf vokal kecil (a, u, e, i, o), program akan menampilkan pesan yang menunjukkan huruf yang diketik. Jika huruf yang dimasukkan merupakan konsonan, program akan masuk ke bagian default dan mencetak pesan bahwa huruf tersebut adalah "huruf mati" (konsonan). Misalnya, jika pengguna mengetik huruf 'a', program akan menampilkan "Yang anda ketik adalah a", sedangkan jika pengguna mengetik huruf 'g', yang merupakan konsonan, program akan menampilkan "Yang anda ketik adalah huruf mati".

**OUTPUT:**

```
D:\Tugas-PB0\Tugas2>javac KasusSwitch.java

D:\Tugas-PB0\Tugas2>java KasusSwitch.java
Ketikkan sebuah huruf, akhiri dengan RETURN
a
Yang anda ketik adalah a

D:\Tugas-PB0\Tugas2>javac KasusSwitch.java

D:\Tugas-PB0\Tugas2>java KasusSwitch.java
Ketikkan sebuah huruf, akhiri dengan RETURN
g
Yang anda ketik adalah huruf mati
```

8. Program ini menghitung luas lingkaran berdasarkan jari-jari yang dimasukkan. Pertama, program menetapkan nilai  $\pi$  (pi) sebagai 3.1415. Kemudian, memasukkan jari-jari lingkaran, yang kemudian digunakan untuk menghitung luas dengan rumus  $\pi \times r \times r$ . Hasil perhitungan akan muncul, diikuti dengan pesan "Akhir program" sebagai tanda bahwa eksekusi telah selesai. Misalnya, jika memasukkan jari-jari 21, program akan menampilkan luas lingkaran sebesar 1385.4014.

**OUTPUT:**

```
D:\Tugas-PB0\Tugas2>javac Konstant.java

D:\Tugas-PB0\Tugas2>java Konstant.java
Jari-jari lingkaran = 21
Luas lingkaran = 1385.4014
Akhir program
```

9. Program Java ini berfungsi untuk menentukan bilangan terbesar dari dua bilangan yang dimasukkan. Pertama, program mendeklarasikan dua variabel a dan b bertipe int untuk menyimpan bilangan yang dimasukkan. Program kemudian menggunakan objek Scanner untuk membaca input dari pengguna. Pengguna diminta untuk mengetikkan dua bilangan secara berurutan dan menekan tombol "Enter" setelah masing-masing bilangan. Setelah menerima input, program mencetak kembali kedua bilangan tersebut untuk memastikan input yang diterima. Selanjutnya, program membandingkan nilai a dan b menggunakan pernyataan if-else. Jika a lebih besar atau sama dengan b, maka program mencetak bahwa nilai maksimum adalah a. Jika tidak, maka program mencetak bahwa nilai maksimum adalah b. Misalnya, jika pengguna memasukkan angka 3 dan 2, program akan mencetak bahwa a = 3 dan b = 2, lalu menampilkan hasil bahwa nilai maksimum adalah 3.

**OUTPUT:**

```
D:\Tugas-PB0\Tugas2>javac Max2.java

D:\Tugas-PB0\Tugas2>java Max2.java
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
3
2
Ke dua bilangan : a = 3 b = 2
Nilai a yang maksimum 3
```

10. Program Java ini digunakan untuk mencetak angka dari 1 hingga N, di mana N adalah bilangan yang dimasukkan oleh pengguna. Pertama, program mendeklarasikan variabel i sebagai penghitung dalam perulangan dan N sebagai batas akhir angka yang akan dicetak. Program kemudian menggunakan objek Scanner untuk membaca input dari pengguna dan meminta mereka memasukkan nilai N. Setelah menerima nilai N, program menggunakan perulangan for untuk mencetak angka dari 1 hingga N. Jika N bernilai positif, program akan mencetak angka dari 1 hingga N secara berurutan. Namun, jika N bernilai negatif atau nol, program tidak akan mencetak angka apa pun selain pesan "Akhir program". Sebagai contoh, jika pengguna memasukkan N = 5, program akan mencetak angka 1, 2, 3, 4, dan 5. Namun, jika pengguna memasukkan N = -3, tidak ada angka yang dicetak karena kondisi perulangan for tidak terpenuhi.

**OUTPUT:**

```
D:\Tugas-PB0\Tugas2>javac PriFor.java

D:\Tugas-PB0\Tugas2>java PriFor.java
Baca N, print 1 s/d N N = -3
Akhir program

D:\Tugas-PB0\Tugas2>javac PriFor.java

D:\Tugas-PB0\Tugas2>java PriFor.java
Baca N, print 1 s/d N N = 5
1
2
3
4
5
Akhir program
```

11. Program Java ini digunakan untuk mencetak angka dari 1 hingga N, di mana N adalah bilangan yang dimasukkan. Pertama, program mendeklarasikan variabel N untuk menyimpan angka yang dimasukkan pengguna dan variabel i sebagai penghitung. Program kemudian menggunakan objek Scanner untuk membaca input dan meminta pengguna memasukkan nilai N. Setelah menerima input, program menginisialisasi i dengan nilai 1 dan mencetak pesan "Print i dengan ITERATE :". Selanjutnya, program menggunakan perulangan for(;;), yang merupakan bentuk perulangan tanpa kondisi eksplisit (perulangan tak terbatas). Di dalam perulangan ini, program mencetak nilai i, lalu memeriksa apakah i telah mencapai N. Jika ya, perulangan dihentikan dengan perintah break. Jika tidak, nilai i ditambah 1 dan perulangan berlanjut. Sebagai contoh, jika pengguna memasukkan N = 5, program akan mencetak angka 1, 2, 3, 4, dan 5 secara berurutan, lalu berhenti.

**OUTPUT:**

```
D:\Tugas-PBO\Tugas2>javac PrintIterasi.java

D:\Tugas-PBO\Tugas2>java PrintIterasi.java
Nilai N >0 = 5
Print i dengan ITERATE :
1
2
3
4
5
```

12. Program Java ini berfungsi untuk mencetak angka dari 1 hingga N menggunakan perulangan do-while. Pertama, program mendeklarasikan variabel N untuk menyimpan angka yang dimasukkan pengguna dan variabel i sebagai penghitung. Program kemudian menggunakan objek Scanner untuk meminta pengguna memasukkan nilai N. Setelah menerima input, variabel i diinisialisasi dengan nilai 1, dan program mencetak pesan "Print i dengan REPEAT:". Perulangan do-while memastikan bahwa blok kode di dalamnya akan dijalankan setidaknya satu kali, bahkan jika kondisi perulangan tidak terpenuhi sejak awal. Dalam setiap iterasi, nilai i akan dicetak, kemudian i akan bertambah satu. Perulangan akan terus berjalan hingga i lebih besar dari N. Jika pengguna memasukkan N = 5, program akan mencetak angka 1, 2, 3, 4, dan 5 secara berurutan. Namun, jika pengguna memasukkan N = -5, program tetap mencetak angka 1 sekali, karena perulangan do-while selalu mengeksekusi setidaknya satu kali sebelum memeriksa kondisi.

**OUTPUT:**

```
D:\Tugas-PBO\Tugas2>javac PrintRepeat.java

D:\Tugas-PBO\Tugas2>java PrintRepeat
Nilai N >0 = -5
Print i dengan REPEAT:
1

D:\Tugas-PBO\Tugas2>javac PrintRepeat.java

D:\Tugas-PBO\Tugas2>java PrintRepeat
Nilai N >0 = 5
Print i dengan REPEAT:
1
2
3
4
5
```

13. Program Java ini digunakan untuk mencetak angka dari 1 hingga N menggunakan perulangan while. Program dimulai dengan mendeklarasikan variabel N untuk menyimpan input dari pengguna dan variabel i sebagai penghitung. Kemudian, program menggunakan Scanner untuk meminta pengguna memasukkan nilai N. Setelah menerima input, variabel i diinisialisasi dengan nilai 1, dan program mencetak pesan "Print i dengan WHILE:". Perulangan while akan terus berjalan selama nilai i masih lebih kecil atau sama dengan N. Dalam setiap iterasi, program mencetak nilai i dan kemudian menaikkan nilainya sebesar 1. Jika N bernilai lebih besar dari 0, program akan mencetak angka dari 1 hingga N. Sebagai contoh, jika pengguna memasukkan N = 3, program akan mencetak angka 1, 2, dan 3. Namun, jika pengguna memasukkan N = 0, perulangan tidak akan dieksekusi sama sekali karena kondisi  $i \leq N$  langsung bernilai salah

**OUTPUT:**

```
D:\Tugas-PBO\Tugas2>javac PrintWhile.java
D:\Tugas-PBO\Tugas2>java PrintWhile.java
Nilai N >0 = 0
Print i dengan WHILE:
D:\Tugas-PBO\Tugas2>javac PrintWhile.java
D:\Tugas-PBO\Tugas2>java PrintWhile.java
Nilai N >0 = 3
Print i dengan WHILE:
1
2
3
```

14. Program ini bertujuan untuk mencetak angka dari 1 hingga N menggunakan perulangan while. Program diawali dengan mendeklarasikan variabel N untuk menyimpan input dari pengguna dan variabel i yang diinisialisasi dengan nilai 1. Setelah pengguna memasukkan nilai N, program mencetak pesan "Print i dengan WHILE (ringkas):". Kemudian, perulangan while akan terus berjalan selama nilai i masih lebih kecil atau sama dengan N. Di dalam perulangan, program mencetak nilai i dan langsung menambah nilainya (i++). Jika pengguna memasukkan angka  $N > 0$ , program akan mencetak angka mulai dari 1 hingga N. Misalnya, jika  $N = 7$ , outputnya adalah 1, 2, 3, 4, 5, 6, 7. Namun, jika pengguna memasukkan nilai negatif atau nol, program hanya akan menampilkan pesan awal tanpa mencetak angka, karena kondisi while ( $i \leq N$ ) tidak terpenuhi.

**OUTPUT:**

```
D:\Tugas-PBO\Tugas2>javac PrintWhile1.java
D:\Tugas-PBO\Tugas2>java PrintWhile1.java
Nilai N >0 = -3
Print i dengan WHILE (ringkas):
D:\Tugas-PBO\Tugas2>javac PrintWhile1.java
D:\Tugas-PBO\Tugas2>java PrintWhile1.java
Nilai N >0 = 7
Print i dengan WHILE (ringkas):
1
2
3
4
5
6
7
```



15. Program ini berfungsi untuk membaca sejumlah bilangan bulat yang dimasukkan oleh pengguna, kemudian menjumlahkan nilai-nilai tersebut hingga pengguna memasukkan angka 999 sebagai tanda berhenti. Alur eksekusi program dimulai dengan mendeklarasikan variabel Sum yang berfungsi untuk menyimpan hasil penjumlahan dengan nilai awal 0, serta variabel x untuk menyimpan input dari pengguna. Program ini pertama-tama meminta pengguna memasukkan angka. Jika angka pertama bukan 999, maka nilai tersebut disimpan dalam variabel Sum. Kemudian, program memasuki perulangan for(;;), yang merupakan perulangan tak terbatas sampai pengguna memasukkan 999. Setiap angka yang dimasukkan akan dijumlahkan ke dalam variabel Sum, dan ketika 999 dimasukkan, program keluar dari perulangan dan mencetak total jumlah angka yang telah dimasukkan.

**OUTPUT:**

```
D:\Tugas-PBO\Tugas2>javac PrintXinterasi.java

D:\Tugas-PBO\Tugas2>java PrintXinterasi.java
Masukkan nilai x (int), akhiri dg 999: 5
Masukkan nilai x (int), akhiri dg 999 : 7
Masukkan nilai x (int), akhiri dg 999 : 15
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 27
```

16. Program PrintXRepeat menggunakan perulangan do-while, yang memastikan bahwa setidaknya satu iterasi dieksekusi sebelum memeriksa kondisi berhenti. Program ini bertujuan untuk menjumlahkan serangkaian bilangan bulat yang dimasukkan pengguna hingga angka 999 diberikan sebagai tanda berhenti. Alur eksekusinya dimulai dengan mendeklarasikan variabel Sum untuk menyimpan hasil penjumlahan dan x untuk menyimpan input pengguna. Jika input pertama adalah 999, program langsung mencetak "Kasus kosong" dan berhenti tanpa melakukan perhitungan. Namun, jika angka pertama bukan 999, program menginisialisasi Sum dengan 0 dan masuk ke dalam perulangan do-while, di mana setiap angka yang dimasukkan akan dijumlahkan ke dalam Sum sebelum meminta input berikutnya. Proses ini terus berulang hingga pengguna memasukkan 999, yang menyebabkan perulangan berhenti, lalu program mencetak hasil penjumlahan. Sebagai contoh, jika pengguna memasukkan 7, 29, dan 4, kemudian memasukkan 999, program akan menjumlahkan angka tersebut ( $7 + 29 + 4 = 40$ ) dan menampilkan "Hasil penjumlahan = 40".

**OUTPUT:**

```
D:\Tugas-PBO\Tugas2>java PrintXRepeat.java
Masukkan nilai x (int), akhiri dg 999 : 999
Kasus kosong

D:\Tugas-PBO\Tugas2>javac PrintXRepeat.java

D:\Tugas-PBO\Tugas2>java PrintXRepeat.java
Masukkan nilai x (int), akhiri dg 999 : 7
Masukkan nilai x (int), akhiri dg 999 : 29
Masukkan nilai x (int), akhiri dg 999 : 4
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 40
```

17. Program PrintXWhile menggunakan perulangan while untuk menjumlahkan angka yang dimasukkan oleh pengguna hingga angka 999 diberikan sebagai tanda berhenti. Variabel Sum diinisialisasi dengan 0, lalu program meminta

input awal. Jika input pertama adalah 999, program langsung mencetak "Hasil penjumlahan = 0" dan berhenti tanpa menampilkan "Kasus kosong". Namun, jika angka pertama bukan 999, program masuk ke dalam perulangan while, di mana setiap angka yang dimasukkan langsung ditambahkan ke dalam Sum, kemudian meminta input berikutnya. Proses ini terus berulang hingga pengguna memasukkan 999, yang menyebabkan perulangan berhenti dan program mencetak total hasil penjumlahan. Sebagai contoh, jika pengguna memasukkan 16, 23, -5, dan 8, lalu menghentikan dengan 999, program akan menjumlahkan angka tersebut ( $16 + 23 - 5 + 8 = 42$ ) dan menampilkan "Hasil penjumlahan = 42".

#### OUTPUT:

```
D:\Tugas-PBO\Tugas2>javac PrintXWhile.java

D:\Tugas-PBO\Tugas2>java PrintXWhile.java
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 0

D:\Tugas-PBO\Tugas2>javac PrintXWhile.java

D:\Tugas-PBO\Tugas2>java PrintXWhile.java
Masukkan nilai x (int), akhiri dg 999 : 16
Masukkan nilai x (int), akhiri dg 999 : 23
Masukkan nilai x (int), akhiri dg 999 : -5
Masukkan nilai x (int), akhiri dg 999 : 8
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 42
```

18. Program SubProgram dibuat untuk melakukan dua operasi utama pada dua bilangan yang dimasukkan oleh pengguna: mencari nilai maksimum dan menukar nilai kedua bilangan. Program dimulai dengan mendeklarasikan dua fungsi: `maxab(int a, int b)` untuk menentukan bilangan terbesar menggunakan operator ternary, dan `tukar(int a, int b)` untuk menukar nilai kedua bilangan. Dalam fungsi `main()`, program meminta pengguna untuk memasukkan dua bilangan secara berurutan. Setelah itu, program menampilkan nilai kedua bilangan yang dimasukkan dan menghitung bilangan terbesar menggunakan fungsi `maxab()`, lalu mencetak hasilnya. Selanjutnya, program memanggil fungsi `tukar()` untuk menukar nilai kedua bilangan, tetapi karena Java menggunakan pass by value untuk tipe data primitif, perubahan nilai dalam fungsi tidak memengaruhi variabel di dalam `main()`. Akibatnya, meskipun di dalam fungsi `tukar()` nilai bilangan terlihat berubah, nilai aslinya di dalam `main()` tetap sama seperti semula. Sebagai contoh, jika pengguna memasukkan 7 dan 18, program akan menampilkan nilai awal  $a = 7$ ,  $b = 18$ , lalu mencetak Maksimum = 18. Setelah pemanggilan fungsi `tukar()`, output menampilkan  $a = 18$ ,  $b = 7$ , tetapi perubahan ini hanya terjadi di dalam fungsi `tukar()`, bukan di dalam `main()`.

#### OUTPUT:

```
D:\Tugas-PBO\Tugas2>javac SubProgram.java

D:\Tugas-PBO\Tugas2>java SubProgram.java
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
7
18
Ke dua bilangan : a = 7 b = 18
Maksimum = 18
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 18 b = 7
```

19. Program Tempair bertujuan untuk menentukan wujud air berdasarkan suhu yang dimasukkan oleh pengguna. Program ini menggunakan pernyataan if-else untuk menangani tiga kemungkinan kondisi suhu: beku, cair, dan uap/gas. Pertama, program meminta pengguna untuk memasukkan suhu dalam derajat Celsius (T). Jika suhu yang dimasukkan kurang dari 0, program mencetak "Wujud air beku", karena air membeku di bawah 0°C. Jika suhu berada dalam rentang 0 hingga 100°C, program mencetak "Wujud air cair", karena dalam rentang ini air tetap dalam bentuk cair. Jika suhu lebih dari 100°C, program mencetak "Wujud air uap/gas", karena air menguap pada suhu di atas 100°C. Misalnya, jika pengguna memasukkan -20, program akan menampilkan "Wujud air beku -20". Jika pengguna memasukkan 25, program akan mencetak "Wujud air cair 25". Jika pengguna memasukkan 130, program akan menampilkan "Wujud air uap/gas 130".

**OUTPUT:**

```
D:\Tugas-PBO\Tugas2>javac Tempair.java

D:\Tugas-PBO\Tugas2>java Tempair.java
Contoh IF tiga kasus
Temperatur (der. C) = -20
Wujud air beku
-20
D:\Tugas-PBO\Tugas2>javac Tempair.java

D:\Tugas-PBO\Tugas2>java Tempair.java
Contoh IF tiga kasus
Temperatur (der. C) = 25
Wujud air cair
25
D:\Tugas-PBO\Tugas2>javac Tempair.java

D:\Tugas-PBO\Tugas2>java Tempair.java
Contoh IF tiga kasus
Temperatur (der. C) = 130
Wujud air uap/gas
130
```