

**LAPORAN
PEMROGRAMAN BERBASIS OBJEK**



Nama : Andi Nureani Bintang
Stambuk : 13020230265
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:\Aini\Tugas PBO\Tugas 2>javac BacaString.java
D:\Aini\Tugas PBO\Tugas 2>java BacaString

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

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 "aini", program membacanya menggunakan datAIn.readLine() dan menyimpannya dalam variabel str. Terakhir, program mencetak kembali string yang telah dimasukkan sebelumnya, dalam hal ini "aini".

2. Output:

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

Penjelasan:

Program ForEver adalah program dasar yang menggunakan perulangan while(true) untuk mencetak teks ke layar. Di awal eksekusi, program menampilkan pesan "Program akan looping, akhiri dengan ^c" sebagai pemberitahuan bahwa program seharusnya berjalan terus-menerus 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 segera menghentikan perulangan. Akibatnya, meskipun awalnya tampak seperti program akan berjalan tanpa henti, sebenarnya program hanya mencetak satu baris teks dan langsung berhenti.

3. Output:

```
D:\Aini\Tugas PBO\Tugas 2>javac If1.java
D:\Aini\Tugas PBO\Tugas 2>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : 56

Nilai a positif 56
D:\Aini\Tugas PBO\Tugas 2>javac If1.java
D:\Aini\Tugas PBO\Tugas 2>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : -56
```

Penjelasan:

Program If1 menunjukkan penggunaan percabangan if dalam Java untuk menentukan apakah suatu bilangan bersifat 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. Kemudian, program mengevaluasi nilai tersebut dengan kondisi if(a >= 0). Jika bilangan yang

dimasukkan lebih dari atau sama dengan nol, program mencetak "Nilai a positif:" diikuti dengan nilai a. Misalnya, ketika saya memasukkan 56, program menampilkan output tersebut. Namun, jika bilangan yang dimasukkan negatif, tidak ada output tambahan karena program tidak memiliki instruksi untuk menangani bilangan negatif. Sebagai contoh, saat saya memasukkan -56, program tidak mencetak apa pun setelahnya.

4. Output:

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

Ketikkan suatu nilai integer : 45
Nilai a positif 45

D:\Aini\Tugas PBO\Tugas 2>-54
'-54' is not recognized as an internal or external command,
operable program or batch file.

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

Ketikkan suatu nilai integer : -54
Nilai a negatif -54

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

Penjelasan:

Program If2 menggunakan percabangan if-else dalam Java untuk menentukan apakah suatu bilangan positif atau negatif. Dengan Scanner, program membaca input dari pengguna setelah menampilkan pesan "Contoh IF dua kasus". Bilangan yang dimasukkan disimpan dalam variabel a dan dievaluasi dengan if ($a \geq 0$). Jika a bernilai nol atau lebih, program mencetak "Nilai a positif: 45". Jika a negatif, blok else mencetak "Nilai a negatif: -54".

5. Output:

```
D:\Aini\Tugas PBO\Tugas 2>javac If3.java
D:\Aini\Tugas PBO\Tugas 2>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer : 1
Nilai a positif 1

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

D:\Aini\Tugas PBO\Tugas 2>javac If3.java
D:\Aini\Tugas PBO\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 1, 0, atau -2. Seperti pada program If2, program ini memanfaatkan Scanner untuk membaca input dari pengguna setelah menampilkan pesan "Contoh IF tiga kasus". Nilai yang dimasukkan disimpan dalam variabel a, lalu dievaluasi dengan tiga kondisi: jika $a > 0$, program mencetak "Nilai a: 1" beserta angka yang dimasukkan. Jika $a == 0$, program menampilkan "Nilai a: 0". Jika $a < 0$, blok else mencetak "Nilai a: -2" diikuti oleh nilai a. Sebagai contoh, saat saya memasukkan 123,

program menampilkan "Nilai a: 1, 123". Jika memasukkan -25, outputnya "Nilai a: -2, -25", sedangkan untuk 0, program mencetak "Nilai a: 0, 0".

6. Output:

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

Penjelasan :

Program KasusBoolean memanfaatkan tipe data Boolean dalam struktur if-else di Java. Di awal program, variabel bool bertipe Boolean dideklarasikan dan diinisialisasi dengan nilai true. Program kemudian melakukan dua evaluasi. Pada kondisi pertama, jika if(bool) dijalankan, program mencetak "true". Jika bool bernilai false, program masuk ke blok else dan mencetak "false". Pada evaluasi kedua, kondisi if(!bool) menggunakan operator negasi (!) 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:\Aini\Tugas PBO\Tugas 2>java KasusSwitch  
Ketikkan sebuah huruf, akhiri dengan RETURN  
a  
Yang anda ketik adalah a  
  
D:\Aini\Tugas PBO\Tugas 2>javac KasusSwitch.java  
  
D:\Aini\Tugas PBO\Tugas 2>java KasusSwitch  
Ketikkan sebuah huruf, akhiri dengan RETURN  
n  
Yang anda ketik adalah huruf mati
```

Penjelasan:

Program KasusSwitch menggunakan struktur switch-case dalam Java untuk mengenali karakter yang dimasukkan oleh pengguna. Variabel cc bertipe char dideklarasikan, dan input dibaca menggunakan Scanner. Setelah menampilkan pesan "Silakan masukkan sebuah huruf, kemudian tekan RETURN", program mengambil karakter pertama dari input dengan masukan.next().charAt(0). Selanjutnya, switch-case digunakan untuk memeriksa apakah huruf yang dimasukkan adalah vokal kecil ('a', 'u', 'e', 'i', atau 'o'). Jika karakter cocok, program mencetak pesan sesuai dengan huruf tersebut. Sebagai contoh, ketika saya memasukkan 'a', program menampilkan pesan yang sesuai. Namun, saat saya memasukkan 'n', program masuk ke blok default, mencetak "Yang Anda ketik adalah huruf mati", menandakan bahwa karakter tersebut adalah konsonan.

8. Output:

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

D:\Aini\Tugas PBO\Tugas 2>java Konstant
Jari-jari lingkaran = 57
Luas lingkaran = 10206.733
Akhir program
```

Penjelasan:

Program ini menghitung luas lingkaran berdasarkan jari-jari yang diberikan oleh pengguna. Konstanta PHI dideklarasikan sebagai 3.1415f menggunakan final, sehingga nilainya tidak berubah selama program berjalan. Variabel r bertipe float digunakan untuk menyimpan nilai jari-jari. Program meminta pengguna memasukkan jari-jari dengan menampilkan pesan "Jari-jari lingkaran =". Misalnya, jika pengguna memasukkan 57, nilai tersebut dibaca menggunakan `masukan.nextFloat()` dan disimpan dalam r. Selanjutnya, program menghitung luas lingkaran dengan rumus $\text{PHI} \times r \times r$ dan mencetak hasilnya, yaitu 10207.0145. Sebagai penutup, program menampilkan pesan "Akhir program", menandakan bahwa eksekusi telah selesai.

9. Output:

```
D:\Aini\Tugas PBO\Tugas 2>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dengan RETURN :
4 7
Ke dua bilangan : a = 4 b = 7
Nilai b yang maksimum: 7

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

D:\Aini\Tugas PBO\Tugas 2>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dengan RETURN :
9 2
Ke dua bilangan : a = 9 b = 2
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 nilai 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, yang kemudian dibaca menggunakan `masukan.nextInt()` dan disimpan dalam a dan b. Kemudian, program membandingkan kedua bilangan dengan 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 4 untuk a dan 7 untuk b, program menampilkan "Nilai b yang maksimum: 7". Sebaliknya, jika pengguna memasukkan 9 untuk a dan 2 untuk b, program mencetak "Nilai a yang maksimum: 9".

10. Output:

```

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

D:\Aini\Tugas PBO\Tugas 2>java PriFor
Baca N, print 1 s/d N = 8
1
2
3
4
5
6
7
8
Akhir program

```

Penjelasan:

Program PriFor mencetak bilangan dari 1 hingga N sesuai dengan input yang diberikan oleh pengguna. Variabel *i* digunakan sebagai penghitung dalam perulangan for, sementara N menyimpan batas angka yang akan dicetak. Program ini juga memanfaatkan Scanner untuk membaca input. Setelah menampilkan pesan "Baca N, print 1 s/d N", program meminta pengguna memasukkan nilai N, yang kemudian dibaca menggunakan `masukan.nextInt()`. Selanjutnya, perulangan for dijalankan, dimulai dari *i* = 1 dan berlanjut hingga *i* ≤ N. Sebagai contoh, jika pengguna memasukkan N = 8, program mencetak angka 1, 2, 3, 4, 5, 6, 7, dan 8 dengan `System.out.println(i)` di setiap iterasi. Setelah perulangan selesai, program menampilkan pesan "Akhir program" sebagai tanda bahwa eksekusi telah berakhir.

11. Output:

```

D:\Aini\Tugas PBO\Tugas 2>javac PrintIterasi.java

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

```

Penjelasan:

Program PrintIterasi mencetak bilangan dari 1 hingga N berdasarkan input yang diberikan pengguna. Variabel *i* berfungsi sebagai penghitung dalam perulangan, sementara N menentukan batas angka yang akan dicetak. Program juga menggunakan Scanner untuk membaca input. Setelah menampilkan pesan "Nilai N > 0 =", program meminta pengguna memasukkan N, yang dibaca menggunakan `masukan.nextInt()`. Kemudian, program menginisialisasi *i* = 1 dan menjalankan perulangan for tanpa kondisi eksplisit (infinite loop). Di dalam perulangan, setiap iterasi mencetak nilai *i* dengan `System.out.println(i)`. Jika *i* sudah mencapai N, perulangan dihentikan menggunakan `break`. Jika belum, *i* bertambah satu (*i*++) dan iterasi berlanjut. Sebagai contoh, jika pengguna memasukkan N = 5, program akan mencetak angka 1, 2, 3, 4, dan 5, lalu perulangan berhenti.

12. Output:

```
D:\Aini\Tugas PBO\Tugas 2>javac PrintRepeat.java
D:\Aini\Tugas PBO\Tugas 2>java PrintRepeat
Nilai N > 0 = 6
Print i dengan REPEAT:
1
2
3
4
5
6
```

Penjelasan:

Program PrintRepeat mencetak bilangan dari 1 hingga N menggunakan perulangan do-while, berdasarkan input dari pengguna. Variabel i digunakan sebagai penghitung, sedangkan N menentukan batas angka yang akan dicetak. Program juga memanfaatkan 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, memastikan bahwa blok do dieksekusi setidaknya sekali sebelum memeriksa kondisi ($i \leq N$). Di dalam perulangan, program mencetak i dengan `System.out.print(i + "\n")`, lalu menaikkan i dengan `i++`. Perulangan berlangsung hingga i melebihi N, setelah itu program berhenti. Sebagai contoh, jika pengguna memasukkan N = 6, program akan mencetak angka 1, 2, 3, 4, 5, dan 6, lalu perulangan berakhir.

13. Output:

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

Penjelasan:

Program PrintWhile mencetak bilangan dari 1 hingga N menggunakan perulangan while, berdasarkan input pengguna. Variabel i digunakan sebagai penghitung, sedangkan N menentukan batas angka yang akan ditampilkan. Program ini juga memanfaatkan 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. Pada setiap iterasi, program mencetak i menggunakan `System.out.println(i)`, lalu menaikkan nilainya dengan `i++`. Perulangan berakhir ketika i melebihi N. Sebagai contoh, jika pengguna memasukkan N = 7, program akan mencetak angka 1, 2, 3, 4, 5, 6, dan 7, kemudian berhenti.

14. Output:

```
D:\Aini\Tugas PBO\Tugas 2>javac PrintWhile1.java
D:\Aini\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 cara yang lebih ringkas. Variabel i berfungsi sebagai penghitung dan langsung diinisialisasi dengan 1, sementara N menentukan batas angka yang akan ditampilkan. 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(). Perulangan while berjalan selama kondisi ($i \leq N$) masih terpenuhi. Berbeda dari versi sebelumnya, perulangan ini menggunakan System.out.println(i++), yang mencetak nilai i lalu langsung menambahkannya dalam satu perintah. Perulangan berhenti ketika i melebihi N. Sebagai contoh, jika pengguna memasukkan N = 5, program akan mencetak angka 1, 2, 3, 4, dan 5, lalu berhenti.

15. Output:

```
D:\Aini\Tugas PBO\Tugas 2>java PrintXinterasi
Masukan nilai x (int), akhiri dg 999: 23
Masukan nilai x (int), akhiri dg 999: 65
Masukan nilai x (int), akhiri dg 999: 34
Masukan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 122

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

D:\Aini\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 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 23, 65, 34, 999, program akan menjumlahkan angka-angka tersebut dan mencetak "Hasil penjumlahan = 122". Jika pengguna langsung memasukkan 999, program mencetak "Kasus kosong" dan "Hasil penjumlahan = 0".

16. Output:

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

D:\Aini\Tugas PBO\Tugas 2>java PrintXRepeat
Masukan nilai x (int), akhiri dg 999: 1
Masukan nilai x (int), akhiri dg 999: 2
Masukan nilai x (int), akhiri dg 999: 3
Masukan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 6
```

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

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 1, 2, 3, 999, maka program akan mencetak "Hasil penjumlahan = 6". Jika pengguna langsung memasukkan 999, program mencetak "Kasus kosong".

17. Output:

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

D:\Aini\Tugas PBO\Tugas 2>java PrintXWhile
Masukan nilai x (int), akhiri dg 999: 43
Masukan nilai x (int), akhiri dg 999: 23
Masukan nilai x (int), akhiri dg 999: 12
Masukan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 78
```

Penjelasan:

Program PrintXWhile menjumlahkan angka yang dimasukkan oleh pengguna hingga angka 999 dimasukkan sebagai tanda berhenti. Program ini menggunakan variabel x untuk menyimpan input angka dan Sum untuk menyimpan hasil penjumlahan, serta memanfaatkan objek Scanner untuk membaca input pengguna. Setelah mencetak 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 pertama bukan 999, nilainya langsung ditambahkan ke dalam Sum, lalu perulangan while (x != 999) berjalan, di mana setiap angka yang dimasukkan akan terus dijumlahkan hingga 999 dimasukkan untuk menghentikan perulangan. Setelah itu, program mencetak hasil akhir dengan format "Hasil penjumlahan = Sum". Misalnya, jika pengguna memasukkan 21, 23, 45, 999, program akan mencetak "Hasil penjumlahan = 89", sedangkan jika pengguna langsung memasukkan 999, program mencetak "Hasil penjumlahan = 0".

18. Output:

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

D:\Aini\Tugas PBO\Tugas 2>java SubProgram
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
1 7
Ke dua bilangan : a = 1 b : 7
Maksimum = 7
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 7 b = 1
```

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 digunakan untuk 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 (7) lebih besar dari a (1), program menampilkan "Maksimum = 7". Selanjutnya, 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 1 dan b tetap 7.

19. Output:

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

D:\Aini\Tugas PBO\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 26
Wujud air cair
26

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

D:\Aini\Tugas PBO\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = -5
Wujud air beku
-5

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

D:\Aini\Tugas PBO\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 150
Wujud air uap/gas
150
```

Penjelasan :

Program Tempair menentukan wujud air berdasarkan suhu yang dimasukkan pengguna menggunakan struktur if-else if-else. Variabel T bertipe integer digunakan untuk menyimpan suhu dalam derajat Celcius, sementara Scanner digunakan untuk 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 26, program menampilkan "Wujud Air Cair". Untuk -5, outputnya adalah "Wujud Air Beku", dan untuk 150, program mencetak "Wujud Air Uap/Gas".