

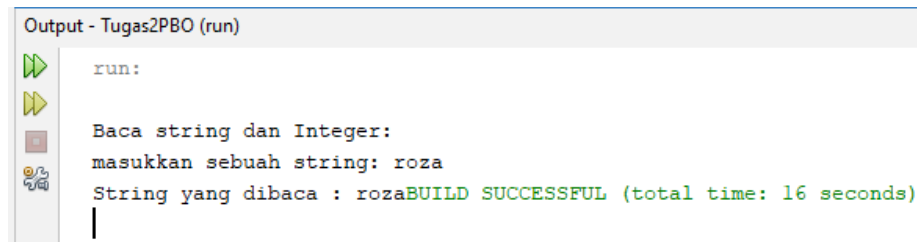
**PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**



Nama : Roza Dhea Aldama Imran
Stambuk : 13020230205
Frekuensi : TI_PBO-13 (B3)
Dosen : Mardiyah Hasnawi, S.Kom., M.T., MTA

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

1. BacaString

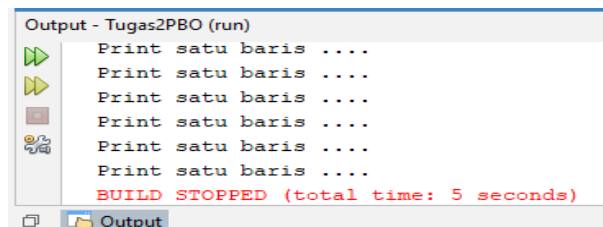


```
Output - Tugas2PBO (run)
run:
Baca string dan Integer:
masukkan sebuah string: roza
String yang dibaca : roza
BUILD SUCCESSFUL (total time: 16 seconds)
```

Penjelasan Kode Program :

Program **BacaString** dalam Java berfungsi untuk membaca input string dari pengguna melalui keyboard menggunakan `BufferedReader`. Program ini pertama-tama meminta pengguna untuk memasukkan sebuah string, kemudian membacanya dengan metode `readLine()`, dan akhirnya mencetak kembali string yang telah dimasukkan. Pada kode ini, **`BufferedReader`** digunakan bersama `InputStreamReader(System.in)` untuk membaca input dari keyboard. Selain itu, program menangani kemungkinan `IOException` dengan menggunakan `throws IOException` pada metode `main()`. Saat dijalankan, program akan menampilkan pesan untuk memasukkan string, dan saya memasukkan string “roza” kemudian program mencetak kembali “roza”.

2. ForEver

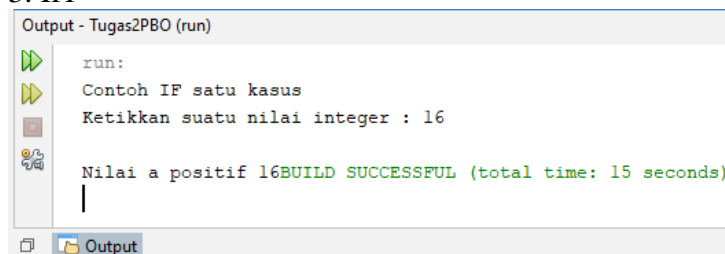


```
Output - Tugas2PBO (run)
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
BUILD STOPPED (total time: 5 seconds)
```

Penjelasan Kode Program :

Program **ForEver** dalam Java menggunakan looping tanpa henti (infinite loop) dengan `while (true)`. Program mencetak pesan "Print satu baris" secara berulang tanpa batas hingga dihentikan secara oleh pengguna.

3. If1



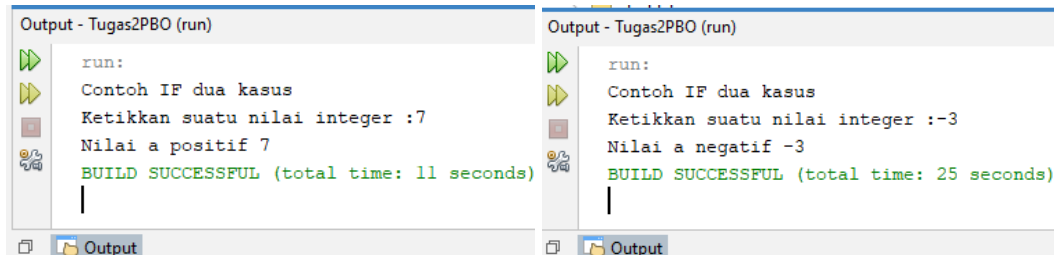
```
Output - Tugas2PBO (run)
run:
Contoh IF satu kasus
Ketikkan suatu nilai integer : 16
Nilai a positif 16
BUILD SUCCESSFUL (total time: 15 seconds)
```

Penjelasan Kode Program :

Program **If1** dalam Java menggunakan struktur IF satu kasus untuk mengecek apakah bilangan yang dimasukkan pengguna bernilai positif atau nol. Program ini menggunakan `Scanner` untuk membaca input integer dari pengguna. Jika angka yang dimasukkan lebih besar atau sama dengan nol, program akan mencetak nilai tersebut sebagai bilangan positif. Namun, jika angka negatif

dimasukkan, program tidak akan menampilkan apa pun karena tidak ada perintah untuk menangani angka negatif.

4. If2



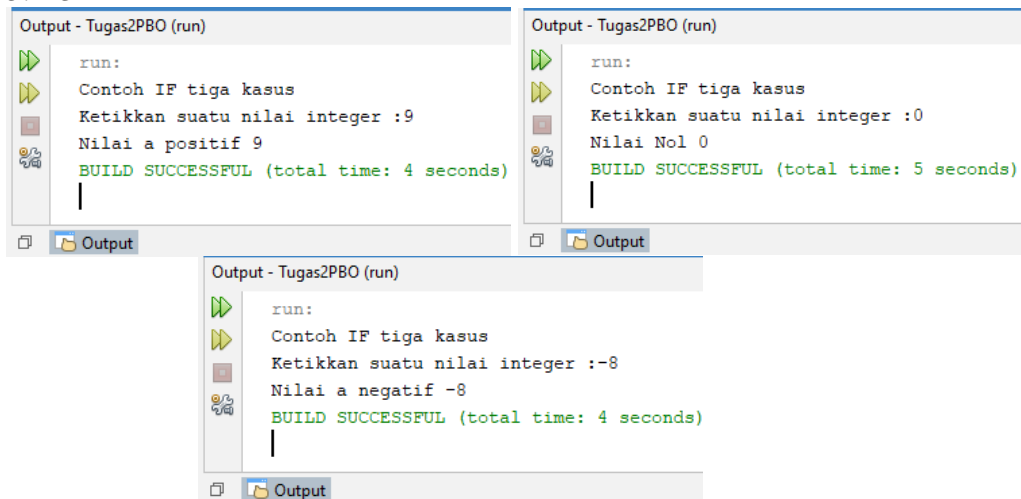
```
run:
Contoh IF dua kasus
Ketikkan suatu nilai integer :7
Nilai a positif 7
BUILD SUCCESSFUL (total time: 11 seconds)

run:
Contoh IF dua kasus
Ketikkan suatu nilai integer :-3
Nilai a negatif -3
BUILD SUCCESSFUL (total time: 25 seconds)
```

Penjelasan Kode Program :

Program **If2** dalam Java digunakan untuk menentukan apakah angka yang dimasukkan oleh pengguna adalah positif atau negatif. Program ini meminta pengguna memasukkan sebuah bilangan integer menggunakan Scanner, lalu mengecek nilainya dengan struktur IF dua kasus. Jika angka yang dimasukkan lebih besar atau sama dengan nol, program akan menampilkan pesan bahwa angka tersebut positif. Sebaliknya, jika angka kurang dari nol, program akan menampilkan pesan bahwa angka tersebut negatif. Saat saya memasukkan angka **7**, program menampilkan "Nilai a positif 7", sedangkan ketika saya memasukkan angka **-7**, program menampilkan "Nilai a negatif -7".

5. If3



```
run:
Contoh IF tiga kasus
Ketikkan suatu nilai integer :9
Nilai a positif 9
BUILD SUCCESSFUL (total time: 4 seconds)

run:
Contoh IF tiga kasus
Ketikkan suatu nilai integer :0
Nilai Nol 0
BUILD SUCCESSFUL (total time: 5 seconds)

run:
Contoh IF tiga kasus
Ketikkan suatu nilai integer :-8
Nilai a negatif -8
BUILD SUCCESSFUL (total time: 4 seconds)
```

Penjelasan Kode Program :

Program **If3** dalam Java menentukan apakah angka yang dimasukkan pengguna positif, nol, atau negatif menggunakan struktur IF tiga kasus. Jika angka lebih dari **0**, program menampilkan bahwa angka tersebut positif. Jika angka **0**, program menampilkan "Nilai Nol", dan jika kurang dari **0**, program menampilkan "Nilai a negatif". Saat saya memasukkan angka "9", program menampilkan "Nilai a positif 9". Jika saya memasukkan **0**, program menampilkan "Nilai Nol 0", dan jika saya memasukkan **-8**, program menampilkan "Nilai a negatif -8".

6. KasusBoolean

```
Output - Tugas2PBO (run)

run:
true
benar
BUILD SUCCESSFUL (total time: 1 second)
```

Penjelasan Kode Program :

Program **KasusBoolean** dalam Java menggunakan ekspresi kondisional dengan boolean untuk mengevaluasi nilai true atau false. Program mendeklarasikan variabel bool dengan nilai true, lalu mengeceknya menggunakan struktur IF. Pada pengecekan pertama, karena bool bernilai true, maka program mencetak "true". Selanjutnya, program mengecek kondisi !bool (negasi dari bool). Karena !bool berarti false, maka bagian else akan dijalankan dan mencetak "benar".

7. KasusSwitch

```
Output - Tugas2PBO (run)

run:
Ketikkan sebuah huruf, akhiri dengan RETURN
a
Yang anda ketik adalah a
BUILD SUCCESSFUL (total time: 2 seconds)

Output - Tugas2PBO (run)

run:
Ketikkan sebuah huruf, akhiri dengan RETURN
F
Yang anda ketik adalah huruf mati
BUILD SUCCESSFUL (total time: 7 seconds)
```

Penjelasan Kode Program :

Program **KasusSwitch** menggunakan struktur switch-case untuk mengevaluasi huruf yang dimasukkan pengguna. Program meminta pengguna mengetikkan satu huruf, kemudian memeriksa apakah huruf tersebut merupakan huruf vokal, yaitu 'a', 'u', 'e', 'i', atau 'o'. Jika huruf yang dimasukkan termasuk dalam daftar tersebut, program akan menampilkan pesan sesuai dengan huruf yang diketik. Namun, jika huruf yang dimasukkan bukan huruf vokal, program akan mencetak bahwa huruf tersebut adalah huruf mati. Saat saya mengetik huruf 'i', program menampilkan "Yang anda ketik adalah i", sedangkan jika saya mengetik huruf 'b', program menampilkan "Yang anda ketik adalah huruf mati".

8. Konstant

```
Output - Tugas2PBO (run)

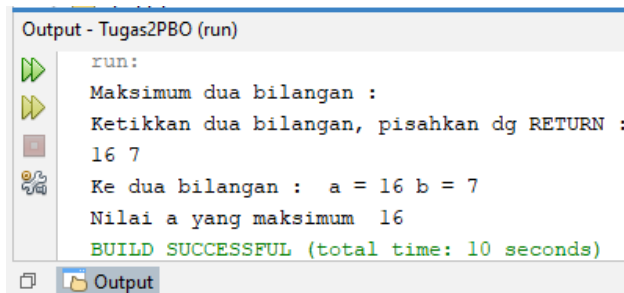
run:
Jari-jari lingkaran =16
Luas lingkaran = 804.224
Akhir program
BUILD SUCCESSFUL (total time: 7 seconds)
```

Penjelasan Kode Program :

Program **Konstant** digunakan untuk menghitung luas lingkaran

berdasarkan jari-jari yang dimasukkan oleh pengguna. Program ini menggunakan konstanta $\text{PHI} = 3.1415f$ dan meminta pengguna memasukkan nilai jari-jari (r) melalui Scanner. Setelah itu, luas lingkaran dihitung dengan rumus $\text{PHI} \times r \times r$, lalu hasilnya ditampilkan. Saat saya memasukkan $r = 16$, program menghitung luasnya sebagai $3.1415 \times 16 \times 16 = 804.224$. Maka, output yang ditampilkan adalah "Luas lingkaran = 804.224", diikuti dengan pesan "Akhir program" sebagai tanda bahwa eksekusi telah selesai.

9. Max2



```

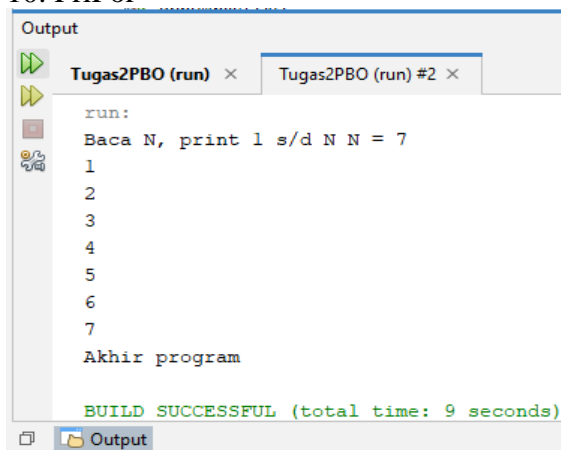
Output - Tugas2PBO (run)
run:
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
16 7
Ke dua bilangan : a = 16 b = 7
Nilai a yang maksimum 16
BUILD SUCCESSFUL (total time: 10 seconds)

```

Penjelasan Kode Program :

Program **Max2** digunakan untuk menentukan bilangan terbesar dari dua angka yang dimasukkan oleh pengguna. Program meminta pengguna memasukkan dua bilangan integer, kemudian membandingkannya menggunakan struktur if-else. Jika bilangan pertama (a) lebih besar atau sama dengan bilangan kedua (b), maka program akan mencetak bahwa a adalah nilai maksimum. Sebaliknya, jika b lebih besar dari a , maka program akan mencetak bahwa b adalah nilai maksimum. Saat saya memasukkan $a = 16$ dan $b = 7$, program menampilkan bahwa nilai a yang maksimum adalah 16, karena 16 lebih besar dari 7.

10. PriFor



```

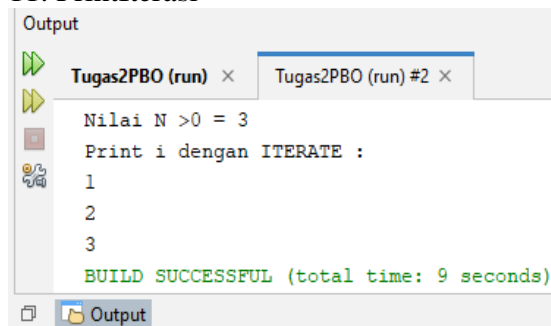
Output
Tugas2PBO (run) x Tugas2PBO (run) #2 x
run:
Baca N, print 1 s/d N N = 7
1
2
3
4
5
6
7
Akhir program
BUILD SUCCESSFUL (total time: 9 seconds)

```

Penjelasan Kode Program :

Program **PriFor** mencetak bilangan dari 1 hingga N menggunakan perulangan for. Setelah pengguna memasukkan N , program akan mencetak angka secara berurutan mulai dari 1 hingga N . Saat saya memasukkan $N = 7$, program mencetak 1, 2, 3, 4, 5, 6, dan 7, masing-masing pada baris baru.

11. PrintIterasi

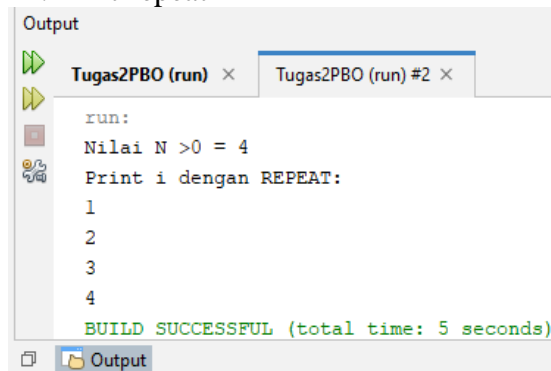


```
run:
Nilai N >0 = 3
Print i dengan ITERATE :
1
2
3
BUILD SUCCESSFUL (total time: 9 seconds)
```

Penjelasan Kode Program :

Program **PrintIterasi** mencetak angka dari 1 hingga N menggunakan perulangan for tanpa kondisi eksplisit dan berhenti dengan break saat $i == N$. Ketika saya memasukkan $N = 3$, program mencetak 1, 2, dan 3, lalu berhenti. Pendekatan ini memungkinkan perulangan berjalan tanpa batas hingga mencapai kondisi berhenti yang ditentukan di dalamnya.

12. PrintRepeat

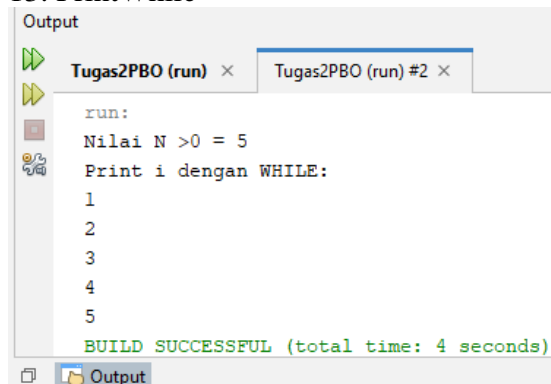


```
run:
Nilai N >0 = 4
Print i dengan REPEAT:
1
2
3
4
BUILD SUCCESSFUL (total time: 5 seconds)
```

Penjelasan Kode Program :

Program **PrintRepeat** mencetak angka dari 1 hingga N menggunakan perulangan do-while. Perulangan ini dijalankan setidaknya sekali sebelum memeriksa kondisi berhenti. Saat saya memasukkan $N = 4$, program mencetak 1, 2, 3, dan 4, lalu berhenti. Pendekatan ini memastikan bahwa minimal satu iterasi tetap dijalankan, bahkan jika nilai N kecil.

13. PrintWhile

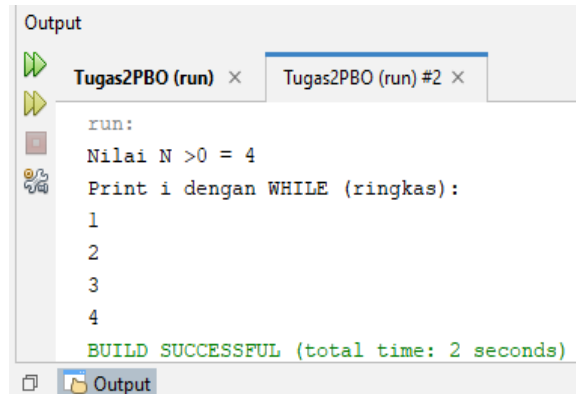


```
run:
Nilai N >0 = 5
Print i dengan WHILE:
1
2
3
4
5
BUILD SUCCESSFUL (total time: 4 seconds)
```

Penjelasan Kode Program :

Program **PrintWhile** mencetak angka dari 1 hingga N menggunakan perulangan while. Perulangan ini akan terus berjalan selama kondisi $i \leq N$ terpenuhi. Saat saya memasukkan $N = 5$, program mencetak 1, 2, 3, 4, dan 5, lalu berhenti. Pendekatan ini memastikan bahwa perulangan hanya dieksekusi jika kondisi awal sudah memenuhi syarat.

14. PrintWhile1

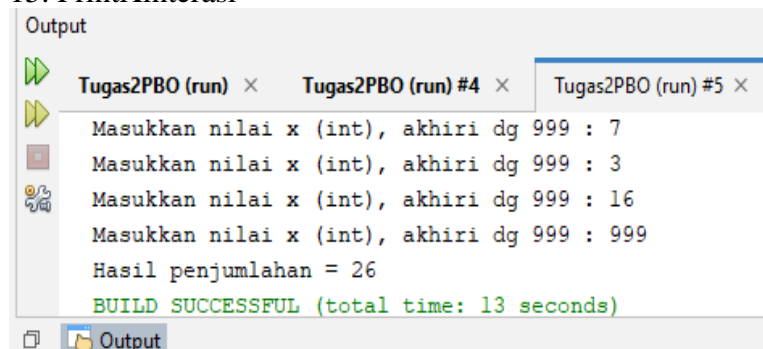


```
run:
Nilai N >0 = 4
Print i dengan WHILE (ringkas):
1
2
3
4
BUILD SUCCESSFUL (total time: 2 seconds)
```

Penjelasan Kode Program :

Program **PrintWhile1** mencetak angka dari 1 hingga N menggunakan perulangan while dalam bentuk ringkas. Variabel i dimulai dari 1, lalu ditingkatkan langsung di dalam perintah cetak ($i++$). Saat saya memasukkan $N = 4$, program mencetak 1, 2, 3, dan 4, lalu berhenti. Struktur ini lebih sederhana dibandingkan versi sebelumnya karena menggabungkan proses cetak dan peningkatan nilai dalam satu baris.

15. PrintXinterasi

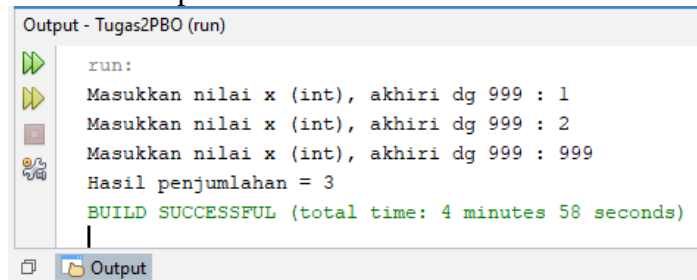


```
Masukkan nilai x (int), akhiri dg 999 : 7
Masukkan nilai x (int), akhiri dg 999 : 3
Masukkan nilai x (int), akhiri dg 999 : 16
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 26
BUILD SUCCESSFUL (total time: 13 seconds)
```

Penjelasan Kode Program :

Program **PrintXinterasi** ini membaca serangkaian angka yang dimasukkan oleh pengguna dan menjumlahkannya hingga pengguna memasukkan angka 999 sebagai tanda berhenti. Saat saya memasukkan 7, 3, 16, 999, program pertama-tama membaca angka 7 dan menetapkan sebagai nilai awal. Kemudian, angka 3 ditambahkan ke jumlah sebelumnya, menghasilkan 10. Selanjutnya, angka 16 ditambahkan, sehingga total menjadi 26. Ketika angka 999 dimasukkan, program menghentikan proses perhitungan dan menampilkan hasil akhir, yaitu 26.

16. PrintXRepeat

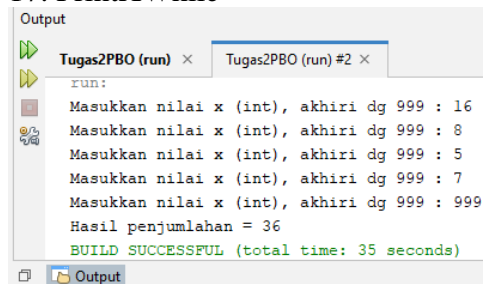


```
run:
Masukkan nilai x (int), akhiri dg 999 : 1
Masukkan nilai x (int), akhiri dg 999 : 2
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 3
BUILD SUCCESSFUL (total time: 4 minutes 58 seconds)
```

Penjelasan Kode Program :

Program **PrintXRepeat** menjumlahkan angka yang dimasukkan hingga saya memasukkan 999 sebagai tanda berhenti. Saat saya memasukkan **1, 2, 999**, program menjumlahkan $1 + 2 = 3$. Setelah memasukkan 999, program berhenti dan menampilkan "Hasil penjumlahan = 3".

17. PrintXWhile

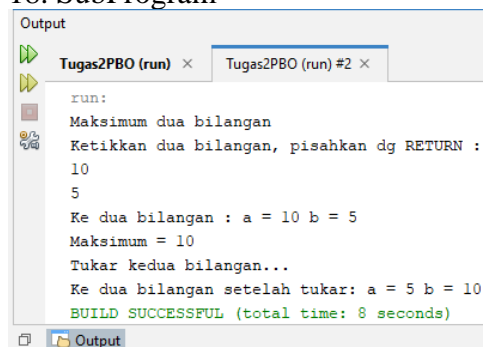


```
run:
Masukkan nilai x (int), akhiri dg 999 : 16
Masukkan nilai x (int), akhiri dg 999 : 8
Masukkan nilai x (int), akhiri dg 999 : 5
Masukkan nilai x (int), akhiri dg 999 : 7
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 36
BUILD SUCCESSFUL (total time: 35 seconds)
```

Penjelasan Kode Program :

Program **PrintXWhile** berfungsi untuk menjumlahkan serangkaian angka yang dimasukkan oleh pengguna hingga menemui angka 999 sebagai tanda berhenti. Saat saya memasukkan angka 16, 8, 5, 7, 999, program pertama-tama membaca angka 16 dan menambahkannya ke dalam total jumlah. Kemudian, angka 8, 5, dan 7 juga ditambahkan satu per satu ke dalam jumlah total. Ketika saya memasukkan 999, program menghentikan proses perulangan dan mencetak hasil penjumlahan dari semua angka sebelumnya. Dengan demikian, hasil akhirnya adalah 36, yang diperoleh dari perhitungan $16 + 8 + 5 + 7$.

18. SubProgram

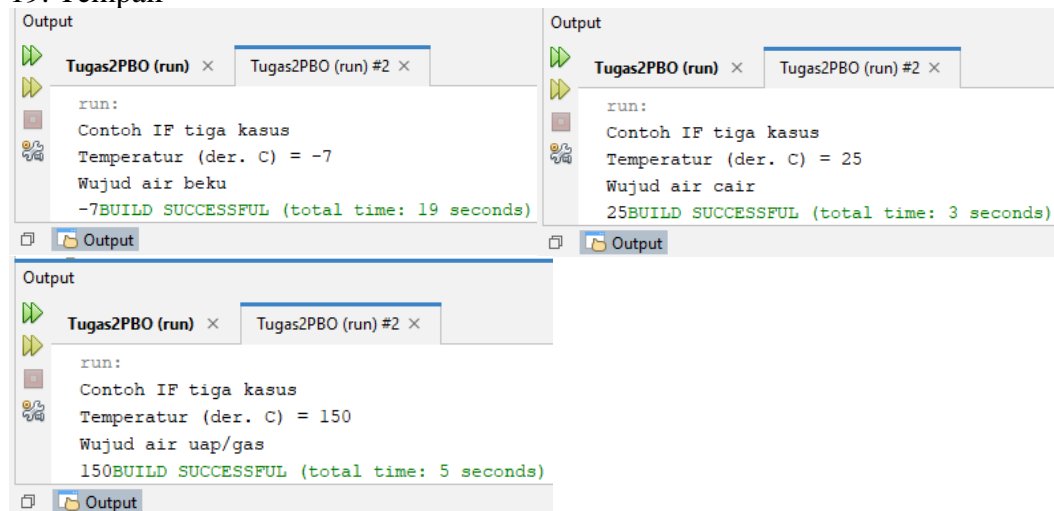


```
run:
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
10
5
Ke dua bilangan : a = 10 b = 5
Maksimum = 10
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 5 b = 10
BUILD SUCCESSFUL (total time: 8 seconds)
```


Penjelasan Kode Program :

Program **SubProgram** adalah contoh program yang menggunakan prosedur dan fungsi untuk mengolah dua bilangan bulat yang dimasukkan oleh pengguna. Saat saya memasukkan dua angka, misalnya 10 dan 5, program pertama-tama mencetak kedua bilangan tersebut. Kemudian, fungsi `maxab()` digunakan untuk menentukan bilangan yang lebih besar, dalam hal ini 10. Selanjutnya, prosedur `tukar()` dipanggil untuk menukar kedua bilangan, tetapi karena Java menggunakan `pass-by-value`, perubahan yang dilakukan dalam prosedur tidak mempengaruhi nilai asli dari variabel di dalam fungsi utama. Oleh karena itu, nilai `a` dan `b` di dalam `main()` tetap seperti semula, tetapi dalam prosedur `tukar()`, nilai yang ditampilkan sudah tertukar.

19. Tempair



Penjelasan Kode Program :

Program **Tempair** dirancang untuk menentukan wujud air berdasarkan suhu yang dimasukkan oleh pengguna. Saat saya memasukkan suhu -7 derajat Celsius, program mengidentifikasi bahwa air berada dalam keadaan beku dan menampilkan "Wujud air beku -7". Ketika saya memasukkan suhu 25 derajat Celsius, program menyatakan bahwa air berada dalam kondisi cair dengan output "Wujud air cair 25". Terakhir, saat saya memasukkan suhu 150 derajat Celsius, program mengenali bahwa air telah berubah menjadi uap atau gas, sehingga menampilkan "Wujud air uap/gas 150". Program ini menggunakan struktur percabangan `if-else` untuk menentukan wujud air berdasarkan rentang suhu yang telah ditentukan.