

**PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

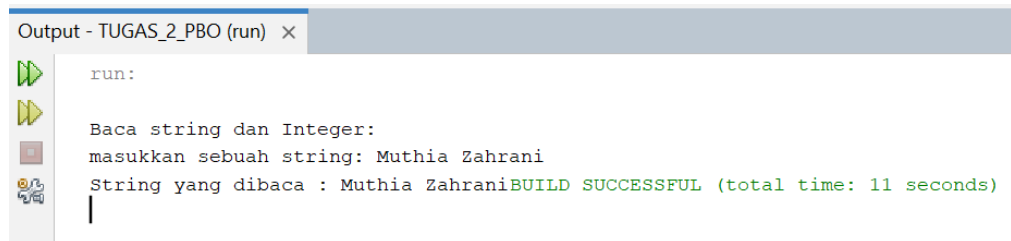


Nama : Muthia Zahrani
Stambuk : 13020230216
Frekuensi : TI_PBO-13 (B3)
Dosen : Mardiyah Hasnawi, S.Kom., M.T

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

Tugas 2 PBO

1. Baca String

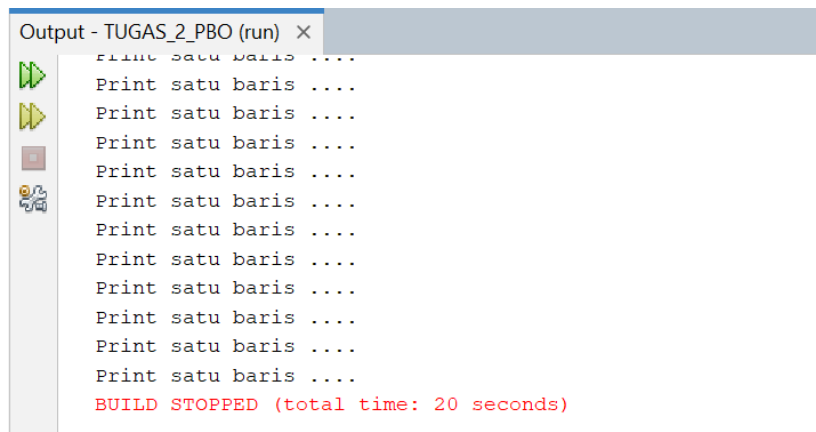


```
run:
Baca string dan Integer:
masukkan sebuah string: Muthia Zahrani
String yang dibaca : Muthia Zahrani
BUILD SUCCESSFUL (total time: 11 seconds)
```

Penjelasan kode program:

Program Java ini membaca input string dari pengguna dengan menggunakan `BufferedReader`. Program ini akan meminta pengguna untuk memasukkan teks, lalu membacanya dengan `readLine()`, dan mencetak kembali input tersebut. Penggunaan `BufferedReader` bersama `InputStreamReader(System.in)` adalah untuk membaca input dari keyboard, throws `IOException` di gunakan di metode `main()` untuk menghindari error saat membaca input. Saat dijalankan, program akan menampilkan pesan untuk memasukkan string, misalnya ketika memasukkan sebuah string yaitu "Muthia Zahrani", kemudian program akan mencetak kembali string yang telah dimasukkan dengan output "String yang dibaca: Muthia Zahrani".

2. ForEver



```
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
BUILD STOPPED (total time: 20 seconds)
```

Penjelasan kode program:

Program Java ini merupakan penggunaan loop tanpa henti dengan menggunakan `while (true)`. Program ini akan mencetak pesan "Print satu baris" secara terus-menerus tanpa batas hingga pengguna menghentikannya secara manual pada output.

3. If1

```
Output - TUGAS_2_PBO (run) x
run:
Contoh IF satu kasus
Ketikkan suatu nilai integer : 29

Nilai a positif 29BUILD SUCCESSFUL (total time: 8 seconds)
```

Penjelasan kode program:

Program ini menggunakan struktur IF satu kasus untuk mengecek apakah bilangan yang dimasukkan pengguna bernilai positif atau nol. Program ini menggunakan scanner untuk membaca input integer. Jika angka yang dimasukkan ≥ 0 , maka program akan mencetak bahwa bilangan tersebut positif. Namun, jika angka negatif dimasukkan, maka program tidak akan menampilkan apa pun karena tidak ada perintah untuk menangani angka negatif.

4. If2

```
Output - TUGAS_2_PBO (run) x
run:
Contoh IF dua kasus
Ketikkan suatu nilai integer : -6
Nilai a negatif -6
BUILD SUCCESSFUL (total time: 5 seconds)
```

```
Output - TUGAS_2_PBO (run) x
run:
Contoh IF dua kasus
Ketikkan suatu nilai integer : 6
Nilai a positif 6
BUILD SUCCESSFUL (total time: 13 seconds)
```

Penjelasan kode program:

Program Java ini mengecek apakah angka yang dimasukkan pengguna positif atau negative. Program ini menggunakan Scanner untuk membaca input bilangan integer, lalu menggunakan struktur IF dua kasus untuk menentukan nilainya. Jika angka ≥ 0 , program akan menampilkan bahwa angka tersebut positif. Jika angka < 0 , program akan menampilkan bahwa angka tersebut negatif. Misalnya, jika memasukkan 6, program menampilkan "Nilai a positif 6", dan jika memasukkan -6, program menampilkan "Nilai a negatif -6".

5. If3

```
Output - TUGAS_2_PBO (run) x
run:
Contoh IF tiga kasus
Ketikkan suatu nilai integer : 2
Nilai a positif 2
BUILD SUCCESSFUL (total time: 27 seconds)
```

```
Output - TUGAS_2_PBO (run) x
run:
Contoh IF tiga kasus
Ketikkan suatu nilai integer : -2
Nilai a negatif -2
BUILD SUCCESSFUL (total time: 4 seconds)
```

```
Output - TUGAS_2_PBO (run) #4 x
run:
Contoh IF tiga kasus
Ketikkan suatu nilai integer : 0
Nilai Nol 0
BUILD SUCCESSFUL (total time: 8 seconds)
```

Penjelasan kode program:

Program Java ini mengecek apakah angka yang dimasukkan pengguna positif, nol, atau negatif. Program tersebut menggunakan scanner untuk membaca input bilangan integer, lalu penggunaan struktur IF tiga kasus untuk menentukan nilainya. Jika angka > 0 , program akan menampilkan bahwa angka tersebut positif. Jika angka $= 0$, program akan menampilkan "Nilai Nol", dan jika angka < 0 , program akan menampilkan "Nilai a negatif". Misalnya, jika memasukkan 2, program menampilkan "Nilai a positif 2", jika memasukkan 0, program menampilkan "Nilai Nol 0", dan jika memasukkan -2, program menampilkan "Nilai a negatif -2".

6. Kasus Boolean

```

Output - TUGAS_2_PBO (run) ×
run:
true
benar
BUILD SUCCESSFUL (total time: 0 seconds)

```

Penjelasan kode program:

Program Java ini menggunakan ekspresi kondisional dengan boolean untuk mengevaluasi true atau false. Mendeklarasikan variabel bool dengan nilai true, lalu mengecek nilainya menggunakan struktur IF. Jika `bool == true`, program akan mencetak "true". Selanjutnya, program mengecek `!bool` (negasi dari bool). Karena `!bool` berarti false, maka bagian else akan dijalankan dan mencetak "benar".

7. Kasus Switch

```

Output - TUGAS_2_PBO (run) ×
run:
Ketikkan sebuah huruf, akhiri dengan RETURN
a
Yang anda ketik adalah a
BUILD SUCCESSFUL (total time: 3 seconds)

Output - TUGAS_2_PBO (run) ×
run:
Ketikkan sebuah huruf, akhiri dengan RETURN
m
Yang anda ketik adalah huruf mati
BUILD SUCCESSFUL (total time: 3 seconds)

```

Penjelasan kode program:

Program Java ini menggunakan switch-case untuk mengevaluasi huruf yang dimasukkan pengguna. Program akan meminta pengguna mengetik satu huruf, lalu mengecek apakah huruf tersebut termasuk vokal ('a', 'u', 'e', 'i', atau 'o'). Jika huruf yang dimasukkan adalah vokal, program akan menampilkan pesan sesuai dengan huruf yang diketik. Jika bukan huruf vokal, program akan mencetak bahwa huruf tersebut adalah huruf mati. Misalnya, jika menginput huruf 'a', program akan menampilkan "Yang anda ketik adalah a", sedangkan jika menginput huruf 'm', program akan menampilkan "Yang anda ketik adalah huruf mati".

8. Konstant

```
Output - TUGAS_2_PBO (run) X
run:
Jari-jari lingkaran = 15
Luas lingkaran = 706.8375
Akhir program
BUILD SUCCESSFUL (total time: 6 seconds)
```

Penjelasan kode program:

Program Java ini menghitung luas lingkaran berdasarkan jari-jari yang dimasukkan pengguna. Menggunakan konstanta $\text{PHI} = 3.1415f$ dan Scanner untuk membaca input jari-jari (r). Lalu menghitung luas lingkaran dengan rumus $\text{PHI} \times r \times r$, lalu menampilkan hasilnya. Misalnya, jika memasukkan jari-jari lingkaran = 15, maka program akan menghitung $3.1415 \times 15 \times 15 = 706.8375$. Maka, output yang ditampilkan adalah "Luas lingkaran = 706.8375", diikuti dengan pesan "Akhir program" sebagai tanda bahwa eksekusi telah selesai.

9. Max

```
Output - TUGAS_2_PBO (run) X
run:
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
2 9
Ke dua bilangan : a = 2 b = 9
Nilai b yang maksimum: 9
BUILD SUCCESSFUL (total time: 18 seconds)
```

Penjelasan kode program:

Program Java ini menentukan bilangan terbesar dari dua angka yang dimasukkan pengguna. Menggunakan scanner untuk membaca input dua bilangan integer, lalu membandingkan nilainya dengan if-else. Jika $a \geq b$, program akan mencetak bahwa a adalah nilai maksimum. Sebaliknya, jika $b > a$, program akan mencetak bahwa b adalah nilai maksimum. Misalnya, jika memasukkan $a = 2$ dan $b = 9$, program akan menampilkan bahwa nilai b yang maksimum adalah 9, karena 9 lebih besar dari 2.

10. Prifor

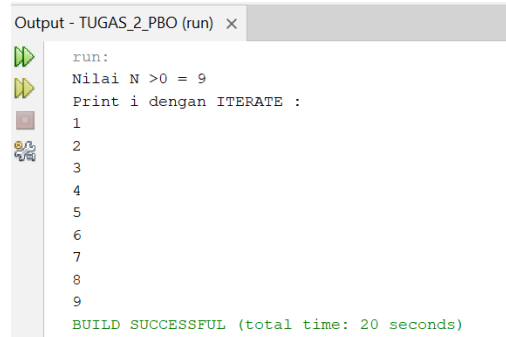
```
Output - TUGAS_2_PBO (run) X
run:
Baca N, print 1 s/d N N = 7
1
2
3
4
5
6
7
Akhir program
BUILD SUCCESSFUL (total time: 17 seconds)
```

Penjelasan kode program:

Program Java ini mencetak bilangan dari 1 hingga N menggunakan perulangan for. Setelah pengguna memasukkan N, program akan mencetak angka dari 1

sampai N secara berurutan. Misalnya, jika memasukkan N = 7, program akan mencetak 1, 2, 3, 4, 5, 6, dan 7, masing-masing pada baris baru.

11. Print Iterasi

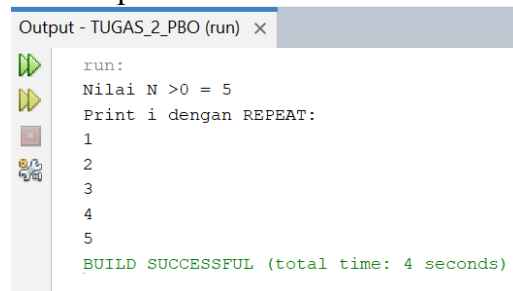


```
Output - TUGAS_2_PBO (run) x
run:
Nilai N >0 = 9
Print i dengan ITERATE :
1
2
3
4
5
6
7
8
9
BUILD SUCCESSFUL (total time: 20 seconds)
```

Penjelasan kode program:

Program Java ini mencetak angka dari 1 hingga N menggunakan perulangan for tanpa kondisi eksplisit. Perulangan akan berjalan terus hingga mencapai $i == N$, lalu berhenti dengan break. Misalnya, jika memasukkan N = 5, program akan mencetak 1, 2, 3, 4, dan 5, lalu berhenti. Pendekatan ini memungkinkan perulangan berjalan tanpa batas sampai mencapai kondisi berhenti yang ditentukan di dalamnya.

12. Print Repeat

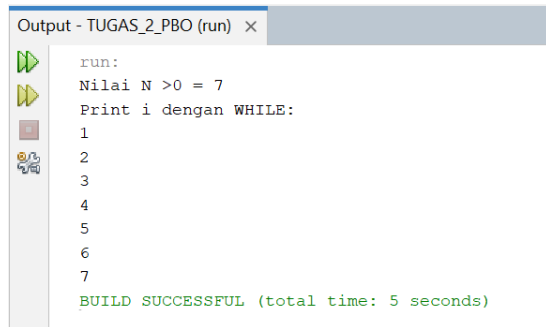


```
Output - TUGAS_2_PBO (run) x
run:
Nilai N >0 = 5
Print i dengan REPEAT:
1
2
3
4
5
BUILD SUCCESSFUL (total time: 4 seconds)
```

Penjelasan kode program:

Dalam program tersebut terdapat dua variabel bertipe integer yang dideklarasikan, yaitu N untuk menyimpan batas akhir perulangan dan i sebagai variabel penghitung. Selanjutnya, objek Scanner dibuat untuk menerima input dari pengguna melalui keyboard. Program kemudian meminta pengguna memasukkan nilai N, yang harus lebih besar dari nol. Setelah itu, variabel i diinisialisasi dengan nilai 1 sebagai elemen pertama yang akan dicetak. Proses perulangan **do-while** dimulai dengan mencetak nilai i, kemudian nilai i bertambah satu di setiap iterasi. Perulangan ini terus berjalan hingga nilai i lebih besar dari N. Perulangan ini akan selalu berjalan setidaknya sekali sebelum memeriksa kondisi berhenti. Misalnya, jika memasukkan N = 5, program akan mencetak 1, 2, 3, 4, dan 5, lalu berhenti.

13. Print While

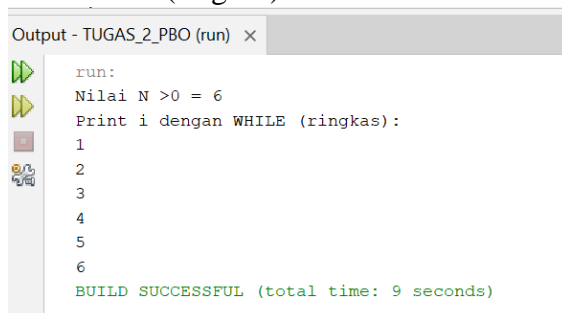


```
run:
  Nilai N >0 = 7
  Print i dengan WHILE:
  1
  2
  3
  4
  5
  6
  7
  BUILD SUCCESSFUL (total time: 5 seconds)
```

Penjelasan kode program:

Program ini membaca input N dari pengguna dan mencetak angka dari 1 hingga N menggunakan perulangan while. Setelah pengguna memasukkan nilai N, program menginisialisasi variabel i dengan nilai 1. Perulangan while kemudian berjalan selama nilai i masih kurang dari atau sama dengan N, mencetak angka i pada setiap iterasi, lalu menambah nilai i satu per satu hingga mencapai N. Setelah perulangan selesai, program akan menampilkan output sesuai dengan inputan dari pengguna. Misalnya memasukkan nilai N=7, maka program akan menampilkan “print I dengan while” lalu mencetak angka 1 hingga 7 secara berurutan dalam baris baru.

14. Print While (ringkas)

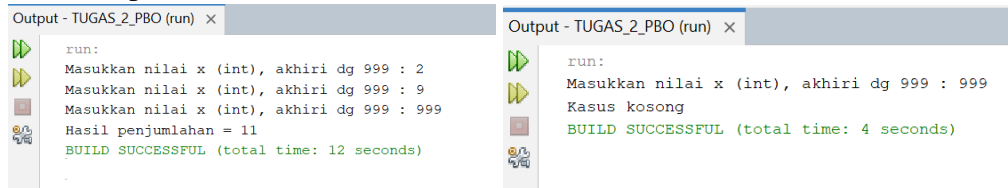


```
run:
  Nilai N >0 = 6
  Print i dengan WHILE (ringkas):
  1
  2
  3
  4
  5
  6
  BUILD SUCCESSFUL (total time: 9 seconds)
```

Penjelasan kode program:

Program ini membaca input N dari pengguna dan mencetak angka dari 1 hingga N menggunakan perulangan while dengan format yang lebih ringkas. Variabel penghitungan i langsung diinisialisasi dengan 1 saat dideklarasikan. Setelah pengguna memasukkan nilai N, perulangan while berjalan selama $i \leq N$, mencetak nilai i, lalu langsung menaikkan nilainya dalam satu baris menggunakan `i++`. Misalnya, jika memasukkan $N = 6$, program akan mencetak 1, 2, 3, 4, 5, dan 6, lalu berhenti. Struktur ini lebih sederhana dibandingkan versi sebelumnya karena penggunaan ekspresi `i++` langsung dalam perintah `System.out.println(i++)`.

15. PrintXRepeat



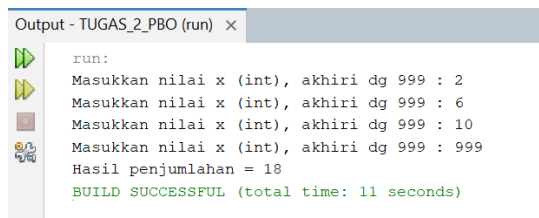
```
run:
Masukkan nilai x (int), akhiri dg 999 : 2
Masukkan nilai x (int), akhiri dg 999 : 9
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 11
BUILD SUCCESSFUL (total time: 12 seconds)

run:
Masukkan nilai x (int), akhiri dg 999 : 999
Kasus kosong
BUILD SUCCESSFUL (total time: 4 seconds)
```

Penjelasan kode program:

Program ini membaca serangkaian angka yang dimasukkan pengguna dan menjumlahkannya hingga angka 999 dimasukkan sebagai tanda berhenti. Jika angka pertama yang dimasukkan adalah 999, program menampilkan "Kasus kosong" dan berhenti. Jika tidak, program mulai menjumlahkan angka dengan perulangan do-while, memastikan angka pertama selalu dihitung. Saat saya memasukkan 2, 9, 999, program menjumlahkan $2 + 9 = 11$. Ketika 999 dimasukkan, program berhenti dan menampilkan "Hasil penjumlahan = 11".

16. PrintXWhile

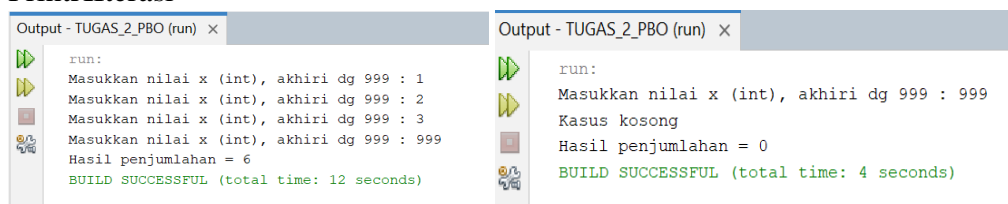


```
run:
Masukkan nilai x (int), akhiri dg 999 : 2
Masukkan nilai x (int), akhiri dg 999 : 6
Masukkan nilai x (int), akhiri dg 999 : 10
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 18
BUILD SUCCESSFUL (total time: 11 seconds)
```

Penjelasan kode program:

Program ini membaca serangkaian bilangan bulat dari pengguna dan menjumlahkannya menggunakan perulangan while, dengan proses berhenti ketika pengguna memasukkan angka 999. Program dimulai dengan menginisialisasi variabel Sum sebagai 0 untuk menyimpan total penjumlahan. Pengguna kemudian diminta memasukkan angka pertama, yang disimpan dalam variabel x. Perulangan while berjalan selama nilai x tidak sama dengan 999, di mana setiap angka yang dimasukkan akan ditambahkan ke Sum. Setelah setiap iterasi, program meminta pengguna memasukkan angka berikutnya. Jika pengguna memasukkan 999, perulangan berhenti, dan program mencetak hasil penjumlahan semua angka yang telah dimasukkan sebelum 999.

17. PrintXIterasi



```
run:
Masukkan nilai x (int), akhiri dg 999 : 1
Masukkan nilai x (int), akhiri dg 999 : 2
Masukkan nilai x (int), akhiri dg 999 : 3
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 6
BUILD SUCCESSFUL (total time: 12 seconds)

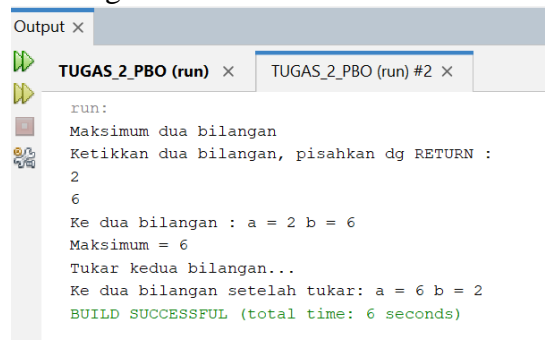
run:
Masukkan nilai x (int), akhiri dg 999 : 999
Kasus kosong
Hasil penjumlahan = 0
BUILD SUCCESSFUL (total time: 4 seconds)
```

Penjelasan kode program:

Program ini membaca serangkaian bilangan bulat dari pengguna dan menjumlahkannya menggunakan perulangan for tanpa batas (for (;;)), yang

dihentikan menggunakan break saat pengguna memasukkan 999. Program pertama-tama meminta pengguna memasukkan angka. Jika angka yang dimasukkan adalah 999, program mencetak "Kasus kosong" lalu langsung berhenti tanpa melakukan perhitungan, dan menampilkan hasil penjumlahan = 0. Jika angka pertama bukan 999, angka tersebut dijadikan nilai awal Sum, lalu program terus meminta input angka baru. Setiap angka yang dimasukkan ditambahkan ke Sum hingga pengguna memasukkan 999, yang menghentikan perulangan dan mencetak hasil penjumlahan. Seperti pada output di atas saya memasukkan 1, 2, 3, 999, program menetapkan 1 sebagai nilai awal. Lalu, angka 2 ditambahkan ke total, menghasilkan 3. Selanjutnya, angka 3 ditambahkan lagi, sehingga total menjadi 6. Saat 999 dimasukkan, program berhenti dan menampilkan "Hasil penjumlahan = 6".

18. SubProgram

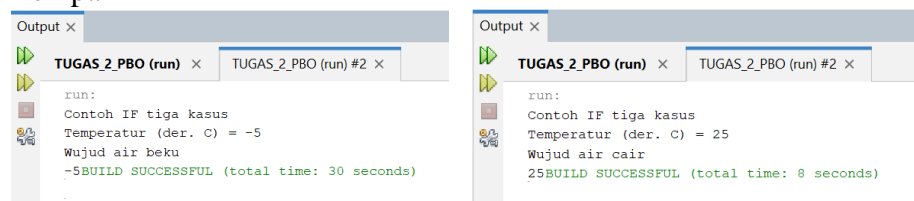


```
run:
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
2
6
Ke dua bilangan : a = 2 b = 6
Maksimum = 6
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 6 b = 2
BUILD SUCCESSFUL (total time: 6 seconds)
```

Penjelasan kode program:

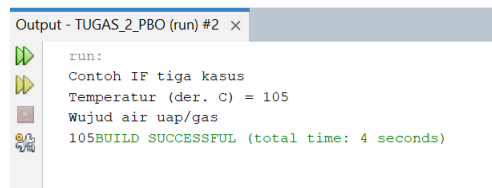
Program ini menggunakan prosedur dan fungsi untuk mengolah dua bilangan bulat yang dimasukkan pengguna. Saat saya memasukkan 6 dan 2, program pertama-tama mencetak kedua angka tersebut. Kemudian, fungsi `maxab(int a, int b)` digunakan untuk menentukan nilai terbesar dari dua bilangan dengan menggunakan operator ternary., yaitu 6. Setelah itu, prosedur `tukar()` dijalankan untuk menukar nilai kedua bilangan, tetapi karena Java menggunakan pass-by-value, perubahan hanya terjadi di dalam prosedur tanpa mempengaruhi nilai asli di `main()`. Akibatnya, nilai `a` dan `b` di `main()` tetap sama, sementara dalam prosedur `tukar()`, nilainya sudah tertukar.

19. Tempair



```
run:
Contoh IF tiga kasus
Temperatur (der. C) = -5
Wujud air beku
-5BUILD SUCCESSFUL (total time: 30 seconds)

run:
Contoh IF tiga kasus
Temperatur (der. C) = 25
Wujud air cair
25BUILD SUCCESSFUL (total time: 8 seconds)
```

A screenshot of a terminal window titled "Output - TUGAS_2_PBO (run) #2". The window contains the following text: "run:", "Contoh IF tiga kasus", "Temperatur (der. C) = 105", "Wujud air uap/gas", and "105BUILD SUCCESSFUL (total time: 4 seconds)". To the left of the text are four small icons: a green play button, a yellow play button, a red square, and a blue gear.

```
run:
Contoh IF tiga kasus
Temperatur (der. C) = 105
Wujud air uap/gas
105BUILD SUCCESSFUL (total time: 4 seconds)
```

Penjelasan kode program:

Pada program Tempair ini digunakan untuk menentukan wujud air berdasarkan suhu yang dimasukkan oleh pengguna. Saat saya memasukkan suhu -5 derajat Celsius, program mendeteksi bahwa air berada dalam keadaan beku dan menampilkan "Wujud air beku -5". Jika saya memasukkan suhu 25 derajat Celsius, program mengenali bahwa air dalam kondisi cair dan mencetak "Wujud air cair 25". Sementara itu, saat saya memasukkan suhu 105 derajat Celsius, program mengidentifikasi bahwa air sudah berubah menjadi uap atau gas, sehingga menampilkan "Wujud air uap/gas 105". Program ini menggunakan percabangan if-else untuk menentukan wujud air sesuai dengan batas suhu yang telah ditentukan.