

TEORI PEMROGRAMAN BERBASIS OBJEK



Nama : Nur Aqidah Safanikiah
Stambuk : 13020230121
Frekuensi : TI_PBO-12 (B2)
Dosen : Mardiyah Hasnawi, S.Kom.,M.T.,MTA

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

1. Output Program BacaString

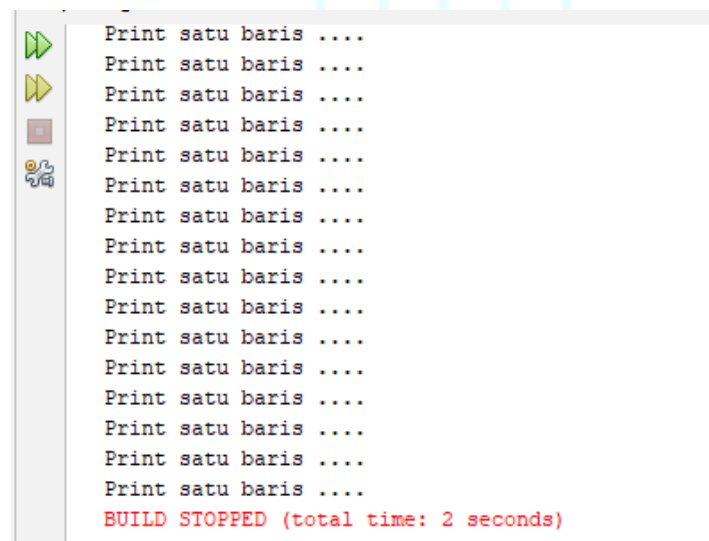
```
run:

Baca string dan Integer:
masukkan sebuah string: Nur Aqidah Safanikiah
String yang dibaca : Nur Aqidah Safanikiah
BUILD SUCCESSFUL (total time: 31 seconds)
```

Penjelasan :

Program **BacaString** adalah aplikasi Java sederhana yang membaca input string dari pengguna melalui konsol. Program ini mengimpor kelas dari paket **java.io** untuk menangani input/output. Di dalam metode **main**, program mendeklarasikan variabel **String** untuk menyimpan input dan menggunakan **BufferedReader** untuk membaca data dari konsol. Setelah menampilkan pesan untuk meminta pengguna memasukkan string, program membaca input tersebut dan menyimpannya dalam variabel. Akhirnya, program mencetak string yang dibaca ke konsol, memberikan umpan balik kepada pengguna tentang input yang mereka masukkan. Program ini menunjukkan cara dasar berinteraksi dengan pengguna di Java.

2. Output Program ForEver

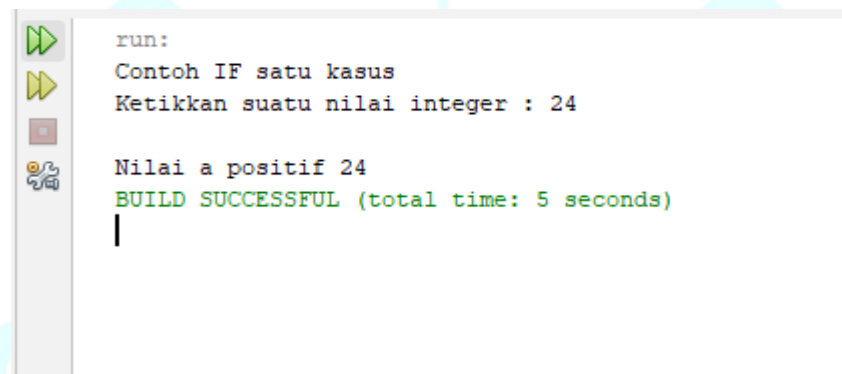


```
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
BUILD STOPPED (total time: 2 seconds)
```

Penjelasan:

Program 'ForEver' adalah aplikasi Java yang menjalankan loop tak terbatas. Di dalam metode 'main', program pertama-tama mencetak pesan yang memberi tahu pengguna bahwa program akan terus berulang dan dapat dihentikan dengan menekan 'Ctrl + C'. Setelah itu, program memasuki loop 'while (true)', yang berarti kondisi loop selalu benar, sehingga program akan terus mencetak "Print satu baris" ke konsol tanpa henti. Program ini dirancang untuk menunjukkan bagaimana loop tak terbatas bekerja, dan pengguna harus menghentikannya secara manual.

3. Output Program If1



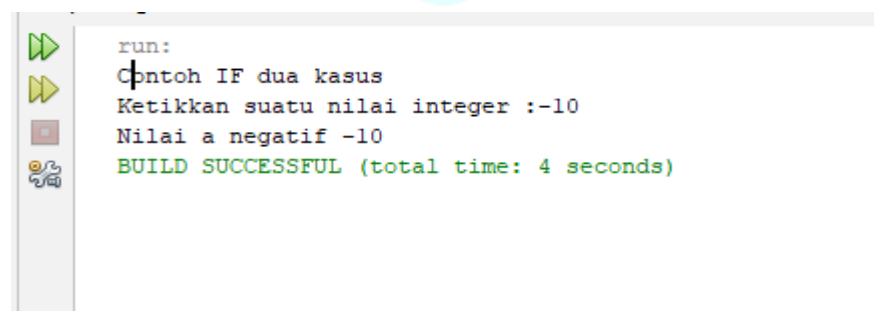
```
run:
Contoh IF satu kasus
Ketikkan suatu nilai integer : 24

Nilai a positif 24
BUILD SUCCESSFUL (total time: 5 seconds)
```

Penjelasan:

Program 'If1' adalah aplikasi Java yang memeriksa apakah sebuah angka yang dimasukkan oleh pengguna adalah positif. Program ini menggunakan 'Scanner' untuk membaca input dari pengguna. Setelah meminta pengguna untuk memasukkan angka, program menyimpan angka tersebut dalam variabel 'a'. Jika angka tersebut lebih besar atau sama dengan nol, program akan mencetak pesan yang menyatakan bahwa angka tersebut positif.

4. Output Program If2



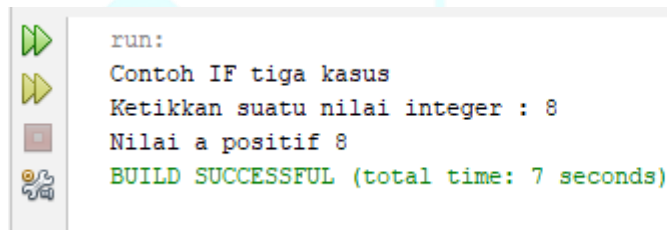
```
run:
Contoh IF dua kasus
Ketikkan suatu nilai integer :-10

Nilai a negatif -10
BUILD SUCCESSFUL (total time: 4 seconds)
```

Penjelasan :

Program 'If2' adalah aplikasi Java yang memeriksa apakah angka yang dimasukkan oleh pengguna adalah positif atau negatif. Program ini menggunakan 'Scanner' untuk membaca input. Setelah pengguna memasukkan angka, program menyimpannya dalam variabel 'a'. Jika angka tersebut lebih besar atau sama dengan nol, program akan mencetak bahwa angka itu positif. Jika angka tersebut kurang dari nol, program akan mencetak bahwa angka itu negatif.

5. Output Program If3

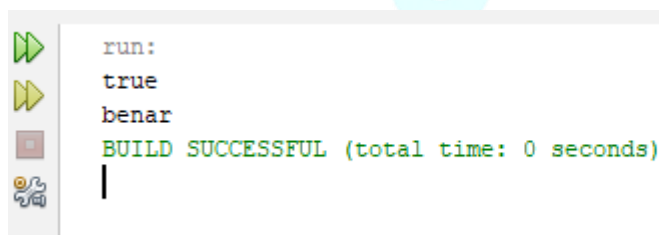


```
run:
Contoh IF tiga kasus
Ketikkan suatu nilai integer : 8
Nilai a positif 8
BUILD SUCCESSFUL (total time: 7 seconds)
```

Penjelasan :

Program Java yang diberikan adalah contoh penggunaan struktur kontrol 'if' untuk menangani tiga kondisi berdasarkan nilai integer yang dimasukkan oleh pengguna. Program ini dimulai dengan mengimpor kelas 'Scanner' untuk menerima input dari konsol. Setelah mendeklarasikan variabel 'a', program meminta pengguna untuk memasukkan nilai integer. Kemudian, menggunakan struktur 'if', program memeriksa apakah nilai tersebut positif, nol, atau negatif, dan mencetak hasil yang sesuai. Jika nilai lebih besar dari 0, program mencetak "Nilai a positif"; jika sama dengan 0, mencetak "Nilai Nol"; dan jika kurang dari 0, mencetak "Nilai a negatif".

6. Output program kasusBoolean

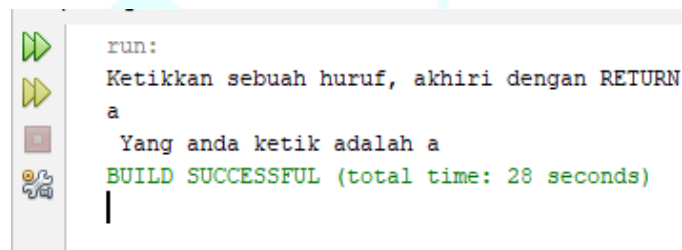


```
run:
true
benar
BUILD SUCCESSFUL (total time: 0 seconds)
```

Penjelasan:

Program Java ini menunjukkan penggunaan ekspresi kondisional dengan tipe data boolean. Di dalam metode `main`, variabel boolean `bool` diinisialisasi dengan nilai `true`. Program pertama-tama memeriksa nilai `bool` menggunakan pernyataan `if`. Karena `bool` bernilai `true`, program mencetak "true" ke konsol. Selanjutnya, program memeriksa negasi dari `bool` dengan `!bool`. Karena `bool` adalah `true`, negasinya menjadi `false`, sehingga program mencetak "benar". Dengan demikian, output dari program ini adalah "true" diikuti oleh "benar".

7. Output Program KasusSwitch

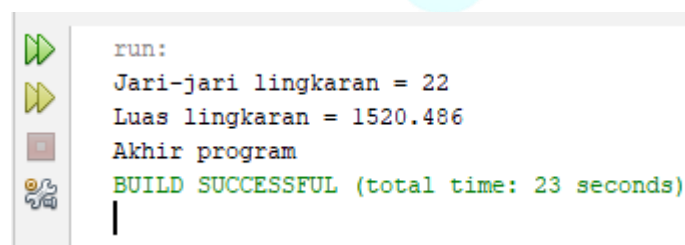


```
run:
Ketikkan sebuah huruf, akhiri dengan RETURN
a
Yang anda ketik adalah a
BUILD SUCCESSFUL (total time: 28 seconds)
```

Penjelasan:

Program Java ini menggunakan struktur kontrol `switch` untuk menentukan jenis huruf yang dimasukkan oleh pengguna. Di dalam metode `main`, program meminta pengguna untuk mengetikkan sebuah huruf dan menyimpannya dalam variabel `cc`. Setelah itu, program menggunakan pernyataan `switch` untuk memeriksa nilai dari `cc`. Jika huruf yang dimasukkan adalah salah satu dari vokal ('a', 'u', 'e', 'i', 'o'), program akan mencetak pesan yang sesuai, seperti "Yang anda ketik adalah a". Jika huruf yang dimasukkan bukan salah satu dari vokal tersebut, program akan mencetak "Yang anda ketik adalah huruf mati".

8. Output program Konstant

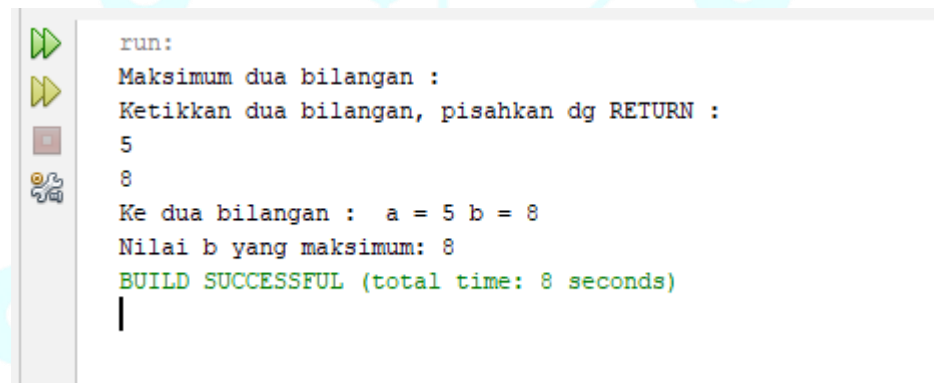


```
run:
Jari-jari lingkaran = 22
Luas lingkaran = 1520.486
Akhir program
BUILD SUCCESSFUL (total time: 23 seconds)
```

Penjelasan:

Program Java ini menghitung luas lingkaran berdasarkan jari-jari yang dimasukkan oleh pengguna. Di dalam metode `main`, program mendeklarasikan sebuah konstanta `PHI` yang menyimpan nilai π (pi) dengan presisi 3.1415. Program kemudian meminta pengguna untuk memasukkan nilai jari-jari lingkaran dan menyimpannya dalam variabel `r`. Setelah mendapatkan input, program menghitung luas lingkaran menggunakan rumus $\text{Luas} = \pi \times r^2$ dan mencetak hasilnya ke konsol. Terakhir, program menampilkan pesan "Akhir program" untuk menandakan bahwa proses telah selesai.

9. Output Program Max2

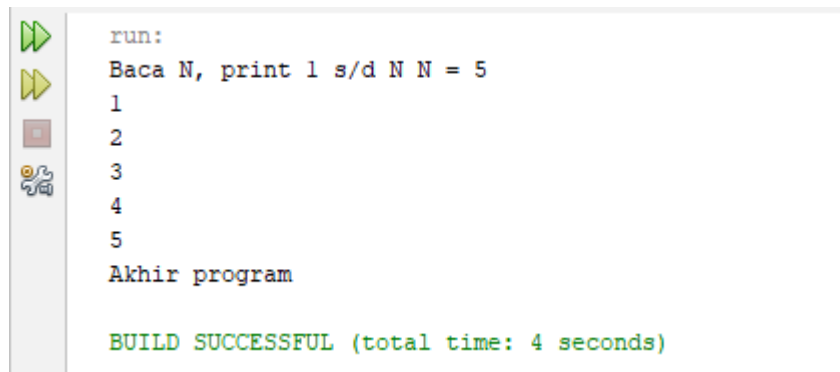


```
run:
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
5
8
Ke dua bilangan : a = 5 b = 8
Nilai b yang maksimum: 8
BUILD SUCCESSFUL (total time: 8 seconds)
```

Penjelasan:

Program Java ini dirancang untuk menemukan nilai maksimum dari dua bilangan yang dimasukkan oleh pengguna. Di dalam metode `main`, program pertama-tama mendeklarasikan dua variabel integer, `a` dan `b`, dan menggunakan objek `Scanner` untuk membaca input dari konsol. Program meminta pengguna untuk memasukkan dua bilangan, yang kemudian disimpan dalam variabel `a` dan `b`. Setelah itu, program mencetak kedua bilangan tersebut. Dengan menggunakan pernyataan `if`, program membandingkan nilai `a` dan `b`. Jika `a` lebih besar atau sama dengan `b`, program mencetak bahwa nilai maksimum adalah `a`; jika tidak, program mencetak bahwa nilai maksimum adalah `b`.

10. Output Program PriFor



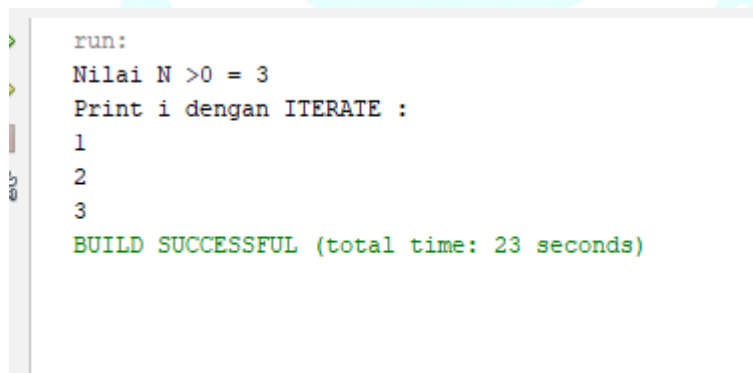
```
run:
Baca N, print 1 s/d N N = 5
1
2
3
4
5
Akhir program

BUILD SUCCESSFUL (total time: 4 seconds)
```

Penjelasan:

Program Java ini bertujuan untuk membaca sebuah angka bulat 'N' dari pengguna dan mencetak semua angka dari 1 hingga 'N' menggunakan loop 'for'. Di dalam metode 'main', program pertama-tama mendeklarasikan dua variabel, 'i' dan 'N', serta menggunakan objek 'Scanner' untuk menerima input dari konsol. Setelah meminta pengguna untuk memasukkan nilai 'N', program menyimpan input tersebut. Kemudian, dengan menggunakan loop 'for', program mengiterasi dari 1 hingga 'N', mencetak setiap angka pada baris baru. Setelah selesai mencetak, program menampilkan pesan "Akhir program" untuk menandakan bahwa proses telah selesai.

11. Output Program PrintIterasi



```
run:
Nilai N >0 = 3
Print i dengan ITERATE :
1
2
3

BUILD SUCCESSFUL (total time: 23 seconds)
```

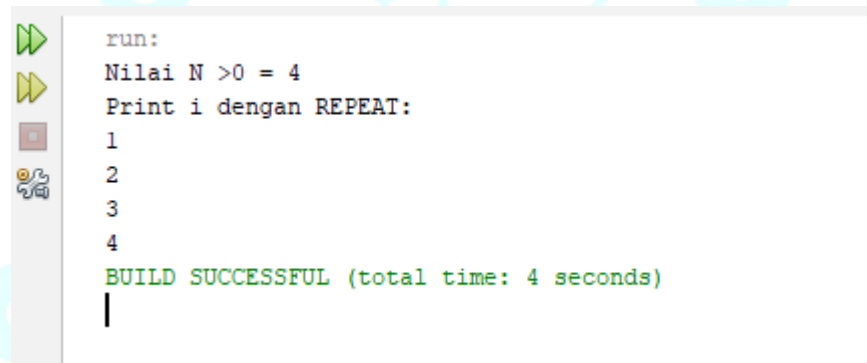
Penjelasan:

Program Java ini dirancang untuk membaca sebuah angka bulat positif 'N' dari pengguna dan mencetak semua angka dari 1 hingga 'N' menggunakan pendekatan iterasi. Di dalam metode 'main', program pertama-tama mendeklarasikan dua variabel, 'N' dan 'i', serta

menggunakan objek `Scanner` untuk menerima input dari konsol. Setelah meminta pengguna untuk memasukkan nilai `N`, program menyimpan input tersebut dan menginisialisasi `i` dengan nilai 1.

Program kemudian memasuki loop tak terbatas (`for (;;)`), di mana ia mencetak nilai `i`. Setelah mencetak, program memeriksa apakah `i` sudah mencapai nilai `N`. Jika ya, program akan menghentikan loop dengan pernyataan `break`. Jika tidak, program akan meningkatkan nilai `i` sebesar 1 untuk melanjutkan ke angka berikutnya. Dengan cara ini, program mencetak semua angka dari 1 hingga `N` secara berurutan.

12. Output Program PrintRepeat

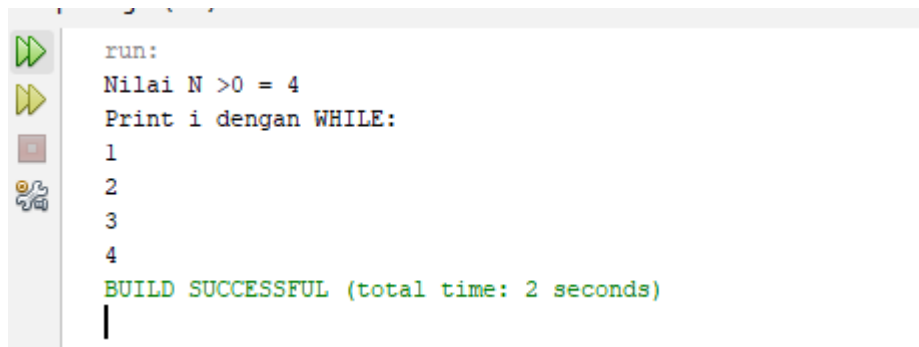


```
run:
Nilai N >0 = 4
Print i dengan REPEAT:
1
2
3
4
BUILD SUCCESSFUL (total time: 4 seconds)
```

Penjelasan:

Program Java ini bertujuan untuk membaca sebuah angka bulat positif `N` dari pengguna dan mencetak semua angka dari 1 hingga `N` menggunakan struktur kontrol `do-while`. Di dalam metode `main`, program pertama-tama mendeklarasikan dua variabel, `N` dan `i`, serta menggunakan objek `Scanner` untuk menerima input dari konsol. Setelah meminta pengguna untuk memasukkan nilai `N`, program menyimpan input tersebut dan menginisialisasi `i` dengan nilai 1. Program kemudian memasuki loop `do-while`, di mana ia mencetak nilai `i` pada setiap iterasi. Setelah mencetak, program meningkatkan nilai `i` sebesar 1 untuk melanjutkan ke angka berikutnya. Loop akan terus berlanjut hingga nilai `i` lebih besar dari `N`. Dengan demikian, program ini mencetak semua angka dari 1 hingga `N` secara berurutan.

13. Output Program PrintWhile

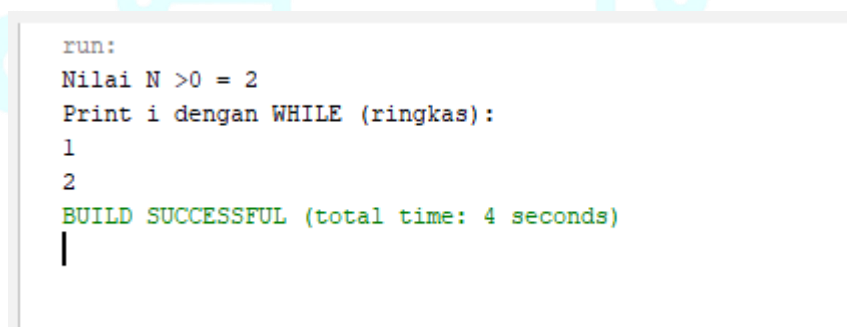


```
run:
Nilai N >0 = 4
Print i dengan WHILE:
1
2
3
4
BUILD SUCCESSFUL (total time: 2 seconds)
```

Penjelasan:

Program Java ini membaca angka bulat positif 'N' dari pengguna dan mencetak angka dari 1 hingga 'N' menggunakan loop 'while'. Setelah menginisialisasi variabel 'i' dengan 1, program mencetak nilai 'i' selama 'i' kurang dari atau sama dengan 'N', kemudian meningkatkan 'i' hingga mencapai batas tersebut.

14. Output Program PrintWhile1



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

Penjelasan :

Program Java ini membaca angka bulat positif 'N' dari pengguna dan mencetak angka dari 1 hingga 'N' menggunakan loop 'while' dengan cara yang ringkas. Setelah menginisialisasi variabel 'i' dengan 1, program meminta pengguna untuk memasukkan nilai 'N'. Dalam loop 'while', program mencetak nilai 'i' dan sekaligus meningkatkan 'i' dengan menggunakan operator post-increment ('i++'). Loop akan terus berjalan hingga 'i' lebih besar dari 'N'.

15. Output Program PrintXinterasi

```

run:
Masukkan nilai x (int), akhiri dg 999 : 5
Masukkan nilai x (int), akhiri dg 999 : 3
Masukkan nilai x (int), akhiri dg 999 : 8
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 16
BUILD SUCCESSFUL (total time: 13 seconds)
|

```

Penjelasan:

Program Java ini membaca nilai bulat `x` dari pengguna dan menjumlahkannya hingga pengguna memasukkan angka 999. Jika pengguna langsung memasukkan 999, program akan menampilkan pesan bahwa tidak ada data yang dijumlahkan. Jika ada nilai yang dimasukkan, program akan menginisialisasi total penjumlahan dengan nilai pertama dan terus meminta input hingga 999 dimasukkan. Setelah itu, program mencetak hasil penjumlahan total.

16. Output Program PrintXRepeat

```

run:
Masukkan nilai x (int), akhiri dg 999 : 4
Masukkan nilai x (int), akhiri dg 999 : 6
Masukkan nilai x (int), akhiri dg 999 : 99
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 109
BUILD SUCCESSFUL (total time: 8 seconds)

```

Penjelasan:

Program Java ini membaca nilai bulat `x` dari pengguna dan menjumlahkannya hingga pengguna memasukkan angka 999. Jika pengguna langsung memasukkan 999, program akan menampilkan pesan bahwa tidak ada data yang dijumlahkan. Jika ada nilai yang dimasukkan, program menginisialisasi total penjumlahan `Sum` dengan 0 dan menggunakan loop `do-while` untuk terus meminta input nilai `x` dan

menambahkannya ke `Sum` hingga 999 dimasukkan. Setelah loop berakhir, program mencetak hasil penjumlahan total

17. Output Program PrintXWhile

```
run:
Masukkan nilai x (int), akhiri dg 999 : 34
Masukkan nilai x (int), akhiri dg 999 : 67
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 101
BUILD SUCCESSFUL (total time: 9 seconds)
```

Penjelasan:

Program Java ini membaca nilai bulat `x` dari pengguna dan menjumlahkannya hingga pengguna memasukkan angka 999. Program dimulai dengan menginisialisasi variabel `Sum` dengan 0. Setelah meminta pengguna untuk memasukkan nilai `x`, program menggunakan loop `while` untuk terus menambahkan nilai `x` ke `Sum` selama `x` tidak sama dengan 999. Setelah loop berakhir, program mencetak hasil penjumlahan total.

18. Output Program SubProgram

```
run:
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
4
6
Ke dua bilangan : a = 4 b = 6
Maksimum = 6
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 6 b = 4
BUILD SUCCESSFUL (total time: 4 seconds)
```

Penjelasan:

Program Java ini mendemonstrasikan penggunaan prosedur dan fungsi. Berikut adalah penjelasan singkat tentang fungsionalitasnya:

- Fungsi `'maxab'`: Fungsi ini menerima dua parameter integer `'a'` dan `'b'`, dan mengembalikan nilai maksimum di antara keduanya menggunakan operator ternary.
- Prosedur `'tukar'`: Prosedur ini bertugas untuk menukar dua bilangan bulat. Meskipun nilai `'a'` dan `'b'` ditukar di dalam prosedur, perubahan ini tidak akan terlihat di luar prosedur karena Java menggunakan passing by value untuk parameter primitif.
- Program Utama :Di dalam metode `'main'`, program meminta pengguna untuk memasukkan dua bilangan bulat. Setelah itu, program mencetak kedua bilangan tersebut, memanggil fungsi `'maxab'` untuk menampilkan nilai maksimum, dan kemudian memanggil prosedur `'tukar'` untuk menampilkan hasil pertukaran bilangan.

19. Output Program Tempair

```
run:
Contoh IF tiga kasus
Temperatur (der. C) = 5
Wujud air cair : 5
BUILD SUCCESSFUL (total time: 4 seconds)
```

Penjelasan:

Program Java ini meminta pengguna untuk memasukkan temperatur dalam derajat Celsius dan menentukan wujud air berdasarkan nilai yang dimasukkan. Setelah pengguna memasukkan nilai, program menggunakan struktur kontrol `'if'` untuk mengevaluasi temperatur tersebut. Jika temperatur kurang dari 0, program mencetak bahwa wujud air adalah **beku**. Jika temperatur berada di antara 0 dan 100, program menyatakan bahwa wujud air adalah **cair**. Terakhir, jika temperatur lebih dari 100, program menginformasikan bahwa wujud air adalah **uap/gas**.