

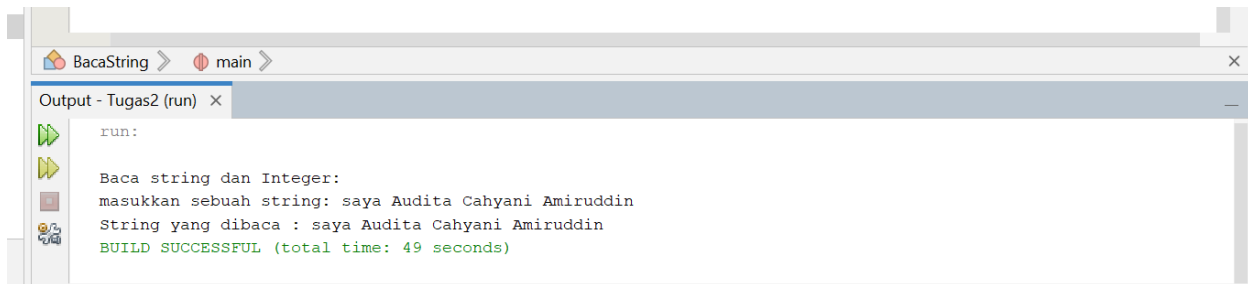
**PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**



Nama : Audita Cahyani Amiruddin
Stambuk : 13020230101
Frekuensi : TI_SD-12 (B2)
Dosen : Mardiyah Hasnawi, S.Kom.,M.T.

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

1. BacaString



```
BacaString > main >
Output - Tugas2 (run) X
run:
Baca string dan Integer:
masukkan sebuah string: saya Audita Cahyani Amiruddin
String yang dibaca : saya Audita Cahyani Amiruddin
BUILD SUCCESSFUL (total time: 49 seconds)
```

Penjelasan : Program Java ini membaca input string dari pengguna menggunakan `BufferedReader`. Pertama, objek `BufferedReader` dibuat untuk menangani input dari konsol. Program meminta pengguna memasukkan sebuah string, lalu membacanya menggunakan metode `readLine()` dan mencetak kembali string yang telah dimasukkan. Untuk meningkatkan keandalan, program sebaiknya menggunakan `try-with-resources` agar `BufferedReader` tertutup secara otomatis dan menangani `IOException` dengan `try-catch` guna mencegah error saat membaca input.

2. ForEver



```
Output - Tugas2 (run) #4 X
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 ....
```

Penjelasan : Program Java ini menjalankan perulangan tak terbatas menggunakan `while (true)`, yang terus mencetak "Print satu baris" tanpa henti. Pesan awal "Program akan looping, akhiri dengan ^C" memberi tahu pengguna bahwa program hanya bisa dihentikan secara manual, misalnya dengan menekan `Ctrl + C` di terminal. Karena tidak ada kondisi penghentian, program dapat menyebabkan penggunaan CPU yang tinggi, sehingga lebih baik menambahkan mekanisme penghentian, seperti menunggu input pengguna atau menggunakan `break` dalam kondisi tertentu.

3. If1



```
tugas2.If1 > main > if (a >= 0) >
Output - Tugas2 (run) #4 X
run:
Contoh IF satu kasus
Ketikkan suatu nilai integer : 11

Nilai a positif 11
BUILD SUCCESSFUL (total time: 5 seconds)
```

Penjelasan: Program ini menggunakan struktur `IF` satu kasus untuk mengecek apakah suatu bilangan integer yang dimasukkan pengguna bernilai positif. Dengan bantuan `Scanner`, program membaca input

dari pengguna, lalu memeriksa apakah nilai tersebut ≥ 0 . Jika kondisi terpenuhi, program mencetak "Nilai a positif [nilai]", namun jika negatif, tidak ada output karena tidak ada blok else.

4. If2



```
tugas2.If2 > main > if (a >= 0) else >
Output - Tugas2 (run) #4 X
run:
Contoh IF dua kasus
Ketikkan suatu nilai integer :-11
Nilai a negatif -11
BUILD SUCCESSFUL (total time: 3 seconds)
```

Penjelasan : Program ini menggunakan struktur IF dua kasus komplementer untuk menentukan apakah bilangan integer yang dimasukkan pengguna bernilai positif atau negatif. Setelah membaca input menggunakan Scanner, program mengevaluasi nilai tersebut. Jika $a \geq 0$, program mencetak "Nilai a positif [nilai]", sedangkan jika $a < 0$, program mencetak "Nilai a negatif [nilai]". Dengan adanya blok else, program selalu memberikan output terlepas dari apakah nilai positif atau negatif.

5.If3



```
tugas2.If3 > main >
Output - Tugas2 (run) #4 X
run:
Contoh IF tiga kasus
Ketikkan suatu nilai integer :0
Nilai Nol 0
BUILD SUCCESSFUL (total time: 5 seconds)
```

Penjelasan : Program ini menggunakan struktur IF tiga kasus untuk mengevaluasi apakah bilangan integer yang dimasukkan pengguna bernilai positif, nol, atau negatif. Jika nilai lebih dari nol, program mencetak "Nilai a positif [nilai]"; jika sama dengan nol, mencetak "Nilai Nol [nilai]"; dan jika kurang dari nol, mencetak "Nilai a negatif [nilai]". Dengan pendekatan ini, program dapat menangani semua kemungkinan nilai integer.

6.KasusBoolean

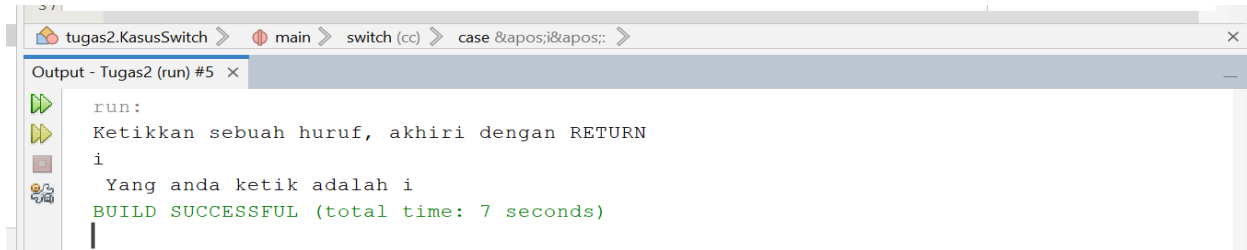


```
Output - Tugas2 (run) #4 X
run:
true
benar
BUILD SUCCESSFUL (total time: 0 seconds)
```

Penjelasan : Program ini mendemonstrasikan penggunaan kondisional dengan tipe data boolean di Java. Variabel bool diinisialisasi dengan true, lalu diperiksa dalam dua kondisi if. Jika bool bernilai true, program mencetak "true", jika false, mencetak "false". Selanjutnya, menggunakan operator negasi (!bool), jika bool false, program mencetak "salah", jika true, mencetak "benar". Program ini

menunjukkan bagaimana boolean dapat digunakan untuk mengontrol alur program dengan kondisi if-else.

7. KasusSwitch



```
tugas2.KasusSwitch > main > switch (cc) > case &apos;i&apos;; >  
Output - Tugas2 (run) #5 <  
run:  
Ketikkan sebuah huruf, akhiri dengan RETURN  
i  
Yang anda ketik adalah i  
BUILD SUCCESSFUL (total time: 7 seconds)
```

Penjelasan : Program ini menggunakan struktur switch-case untuk mengevaluasi karakter yang dimasukkan pengguna. Setelah membaca satu karakter dengan Scanner, program memeriksa apakah karakter tersebut adalah huruf vokal (a, u, e, i, o) dan mencetak pesan yang sesuai. Jika karakter yang dimasukkan bukan vokal, maka akan masuk ke blok default dan mencetak "Yang anda ketik adalah huruf mati".

8. Konstant



```
tugas2.Konstant >  
Output - Tugas2 (run) #4 <  
run:  
Jari-jari lingkaran =4  
Luas lingkaran = 50.264  
Akhir program  
BUILD SUCCESSFUL (total time: 3 seconds)
```

Penjelasan : Program ini menghitung luas lingkaran dengan menggunakan konstanta $\text{PHI} = 3.1415f$. Setelah membaca jari-jari (r) yang dimasukkan pengguna menggunakan Scanner, program menghitung luas lingkaran dengan rumus $\pi \times r^2$ dan mencetak hasilnya. Penggunaan final float memastikan bahwa nilai PHI tidak dapat diubah.

9.Max2



```
tugas2.Max2 > main >  
Output - Tugas2 (run) #4 <  
run:  
Maksimum dua bilangan :  
Ketikkan dua bilangan, pisahkan dg RETURN :  
21  
4  
Ke dua bilangan : a = 21 b = 4  
Nilai a yang maksimum 21  
BUILD SUCCESSFUL (total time: 15 seconds)
```

Penjelasan : Program ini menentukan bilangan maksimum dari dua bilangan integer yang dimasukkan pengguna. Setelah membaca dua bilangan dengan Scanner, program membandingkan keduanya

menggunakan struktur IF. Jika $a \geq b$, program mencetak a sebagai nilai maksimum, sedangkan jika $a < b$, program mencetak b sebagai nilai maksimum

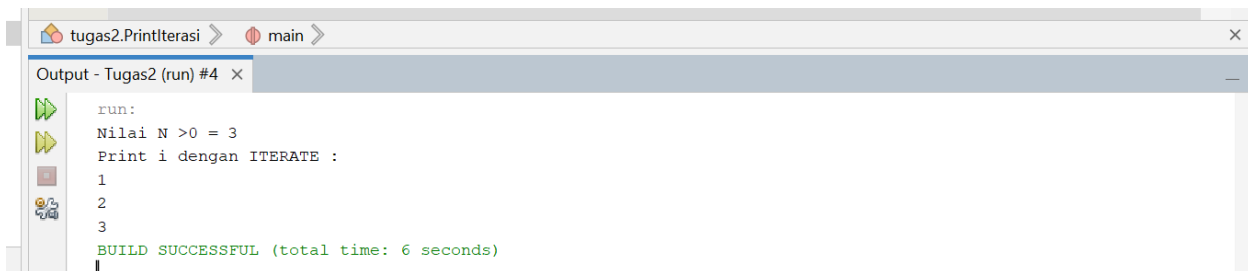
10. PriFor



```
run:
Baca N, print 1 s/d N N = 4
1
2
3
4
Akhir program
BUILD SUCCESSFUL (total time: 3 seconds)
```

Penjelasan : Program ini membaca sebuah bilangan integer N dan mencetak angka dari 1 hingga N menggunakan perulangan FOR. Setelah membaca nilai N dari pengguna menggunakan Scanner, program menjalankan loop for ($i = 1; i \leq N; i++$), yang mencetak nilai i dari 1 hingga N secara berurutan.

11. PrintIterasi



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

Penjelasan : Program ini membaca bilangan N dan mencetak angka dari 1 hingga N menggunakan perulangan tak terbatas (for(;;)) yang dikendalikan dengan break. Setelah membaca nilai N dan menginisialisasi $i = 1$, program mencetak i, lalu memeriksa apakah $i == N$. Jika iya, perulangan berhenti dengan break, jika tidak, nilai i bertambah. Penggunaan for(;;) bisa diganti dengan while ($i \leq N$) agar lebih jelas.

12. PrintRepeat



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

Penjelasan : Program ini menggunakan perulangan do-while untuk mencetak angka dari 1 hingga N. Setelah membaca nilai N, program menginisialisasi $i = 1$, lalu mencetak i sebelum menambahkannya.

Perulangan akan terus berjalan selama $i \leq N$, karena do-while memastikan blok kode dijalankan minimal sekali sebelum mengecek kondisi.

13. PrintWhile



```
tugas2.PrintWhile > main >
Output - Tugas2 (run) #4 x
run:
Nilai N >0 = 5
Print i dengan WHILE:
1
2
3
4
5
BUILD SUCCESSFUL (total time: 6 seconds)
```

Penjelasan : Program ini menggunakan perulangan while untuk mencetak angka dari 1 hingga N. Setelah membaca nilai N dari pengguna, program menginisialisasi $i = 1$, lalu menjalankan perulangan while ($i \leq N$). Setiap iterasi mencetak i , kemudian menambahkannya satu per satu hingga $i > N$, yang menyebabkan perulangan berhenti.

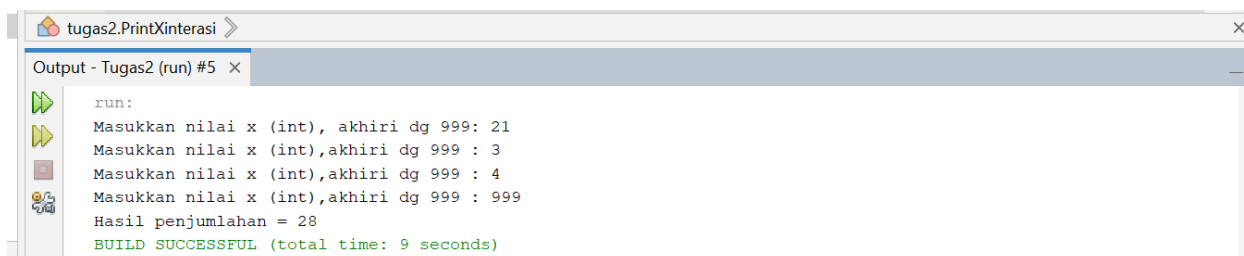
14. PrintWhile1



```
tugas2.PrintWhile1 > main >
Output - Tugas2 (run) #4 x
run:
Nilai N >0 = 2
Print i dengan WHILE (ringkas):
1
2
BUILD SUCCESSFUL (total time: 2 seconds)
```

Penjelasan : Program ini menggunakan perulangan while untuk mencetak angka dari 1 hingga N dengan cara yang lebih ringkas. Setelah membaca nilai N, program menginisialisasi $i = 1$, lalu menjalankan perulangan while ($i \leq N$). Dalam satu baris, program mencetak i dan langsung menaikkannya ($i++$), sehingga lebih efisien dibandingkan menulisnya dalam dua langkah terpisah.

15. PrintXinterasi

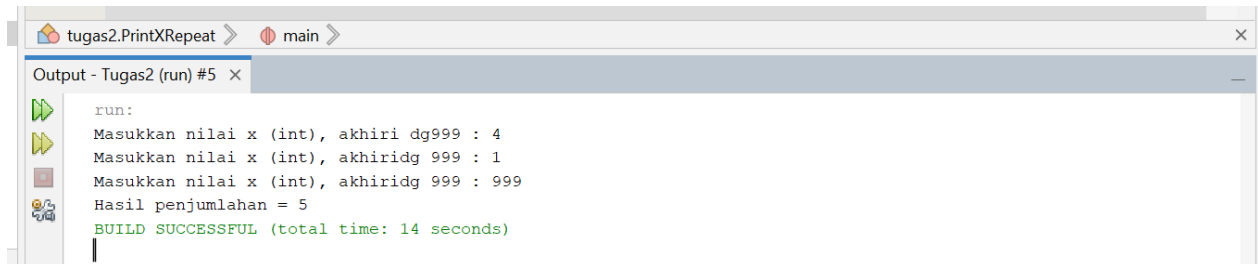


```
tugas2.PrintXinterasi >
Output - Tugas2 (run) #5 x
run:
Masukkan nilai x (int), akhiri dg 999: 21
Masukkan nilai x (int), akhiri dg 999 : 3
Masukkan nilai x (int), akhiri dg 999 : 4
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 28
BUILD SUCCESSFUL (total time: 9 seconds)
```

Penjelasan : Program ini membaca nilai x secara berulang dan menjumlahkannya menggunakan perulangan tak terbatas (`for(;;)`) hingga pengguna memasukkan 999 sebagai tanda berhenti. Jika nilai

pertama yang dimasukkan adalah 999, program mencetak "Kasus kosong" dan langsung berhenti. Jika tidak, nilai x pertama dijadikan inisialisasi Sum, lalu program terus meminta angka baru, menambahkannya ke Sum, hingga 999 dimasukkan.

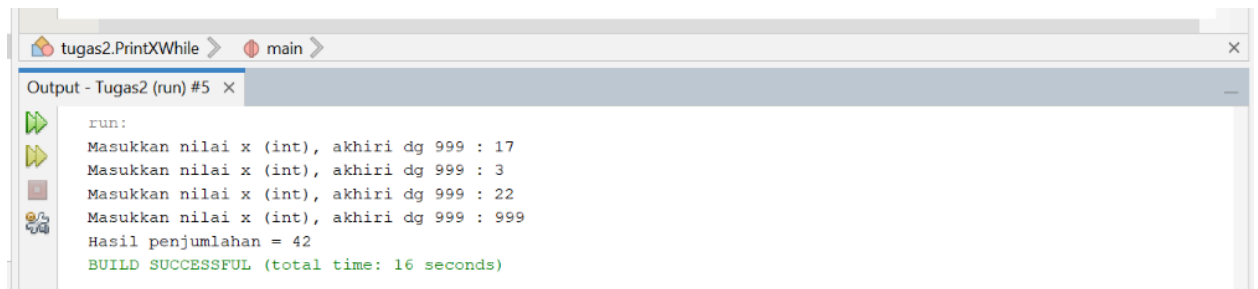
16.PrintXRepeat



```
tugas2.PrintXRepeat > main >
Output - Tugas2 (run) #5 x
run:
Masukkan nilai x (int), akhiri dg999 : 4
Masukkan nilai x (int), akhiridg 999 : 1
Masukkan nilai x (int), akhiridg 999 : 999
Hasil penjumlahan = 5
BUILD SUCCESSFUL (total time: 14 seconds)
```

Penjelasan : Program ini menggunakan perulangan do-while untuk membaca dan menjumlahkan nilai x yang dimasukkan oleh pengguna, hingga 999 dimasukkan sebagai tanda berhenti. Jika nilai pertama adalah 999, program mencetak "Kasus kosong" dan langsung berhenti. Jika tidak, program mulai menjumlahkan nilai x ke dalam Sum dan terus meminta input baru hingga 999 dimasukkan. Setelah perulangan selesai, hasil penjumlahan ditampilkan.

17.PrintXWhile



```
tugas2.PrintXWhile > main >
Output - Tugas2 (run) #5 x
run:
Masukkan nilai x (int), akhiri dg 999 : 17
Masukkan nilai x (int), akhiri dg 999 : 3
Masukkan nilai x (int), akhiri dg 999 : 22
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 42
BUILD SUCCESSFUL (total time: 16 seconds)
```

Penjelasan : Program di atas merupakan program Java yang menjumlahkan serangkaian bilangan yang dimasukkan oleh pengguna menggunakan perulangan while. Program dimulai dengan inisialisasi variabel Sum untuk menyimpan hasil penjumlahan dan menggunakan objek Scanner untuk membaca input pengguna. Program akan terus meminta pengguna memasukkan bilangan bulat, menambahkannya ke Sum, dan berhenti saat pengguna memasukkan angka 999 sebagai tanda akhir. Setelah perulangan selesai, program menampilkan hasil penjumlahan dari semua angka yang telah dimasukkan, kecuali 999.

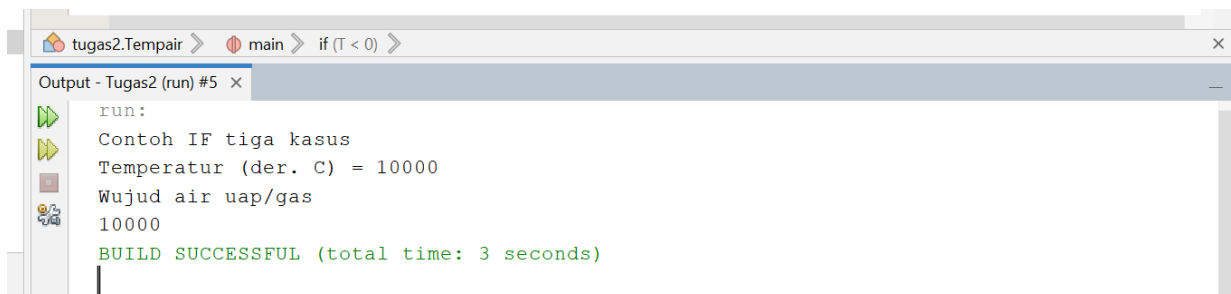
18. SubProgram



```
tugas2.SubProgram > main >
Output - Tugas2 (run) #5 x
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
21
3
Ke dua bilangan : a =21 b = 3
Maksimum = 21
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a =3 b = 21
BUILD SUCCESSFUL (total time: 8 seconds)
```

Penjelasan : Program di atas menggunakan fungsi dan prosedur dalam Java untuk mengolah dua bilangan bulat yang dimasukkan oleh pengguna. Fungsi `maxab(int a, int b)` mengembalikan nilai maksimum dari dua bilangan, sedangkan prosedur `tukar(int a, int b)` mencoba menukar kedua bilangan. Namun, karena Java menggunakan `pass by value` untuk tipe data primitif, prosedur `tukar()` hanya menukar nilai dalam lingkungannya sendiri tanpa mengubah variabel asli di `main()`. Program ini juga memperbaiki format output agar lebih jelas dan mudah dibaca oleh pengguna.

19. Tempair



```
tugas2.Tempair > main > if (T < 0) >
Output - Tugas2 (run) #5 x
run:
Contoh IF tiga kasus
Temperatur (der. C) = 10000
Wujud air uap/gas
10000
BUILD SUCCESSFUL (total time: 3 seconds)
```

Penjelasan : Program di atas menentukan wujud air berdasarkan suhu yang dimasukkan oleh pengguna menggunakan struktur `if-else`. Jika suhu (`T`) kurang dari 0°C , air berwujud beku; jika suhu antara 0°C hingga 100°C , air berwujud cair; dan jika suhu lebih dari 100°C , air berwujud uap/gas. Program ini membaca input pengguna menggunakan `Scanner`, kemudian mengevaluasi kondisi suhu untuk menentukan wujud air.