

Tugas II

Makassar, 06 Maret 2024

PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK



Nama : Rezky Nur Amalia. M
Nim : 13020230327
Dosen : Mardiyah Hasnawi, S.Kom., M.T.

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR

2025

1. Output program **BacaString**

```
D:\semester 4\PBO\Tugas 2>javac BacaString.java
D:\semester 4\PBO\Tugas 2>java BacaString.java

Baca string dan Integer:
masukkan sebuah string: Hello World
String yang dibaca : Hello World
```

Keterangan :

Program ini membaca input berupa string menggunakan `BufferedReader` dan menampilkannya kembali ke layar. Setelah pengguna memasukkan sebuah string, program akan menyimpannya dalam variabel `str` lalu mencetaknya dengan format "String yang dibaca : [input pengguna]". Penggunaan `BufferedReader` memungkinkan program untuk membaca input lebih efisien, terutama dalam skenario yang memerlukan pengolahan data dalam jumlah besar.

2. Output program **ForEver**

[illegible]

Keterangan :

Program ini adalah contoh infinite loop (perulangan tak terbatas) menggunakan while (true). Setelah dijalankan, program akan mencetak "Print satu baris" secara berulang tanpa henti. Pengguna hanya dapat menghentikan eksekusi dengan "Ctrl + C" atau dengan menghentikan proses secara manual. Meskipun loop ini bisa digunakan untuk simulasi atau debugging, penggunaan infinite loop tanpa kondisi penghentian dapat menyebabkan CPU bekerja terus-menerus, yang dapat mengakibatkan beban sistem tinggi.

3. Output program **If1**

```
D:\semester 4\PBO\Tugas 2>javac If1.java
D:\semester 4\PBO\Tugas 2>java If1.java
Contoh IF satu kasus
Ketikkan suatu nilai integer : 1

Nilai a positif 1
D:\semester 4\PBO\Tugas 2>java If1.java
Contoh IF satu kasus
Ketikkan suatu nilai integer : -1
```

Keterangan :

Program ini merupakan implementasi dari IF satu kasus. Program meminta pengguna untuk memasukkan bilangan integer melalui Scanner. Jika bilangan yang dimasukkan lebih dari atau sama dengan nol (≥ 0), maka program akan mencetak bahwa nilai tersebut positif. Namun, jika bilangan tersebut negatif, program tidak akan memberikan output apa pun

karena tidak ada blok else yang menangani kondisi tersebut. Program ini cocok untuk mendeteksi bilangan positif atau nol, tetapi tidak memberikan respons jika inputnya negatif.

4. Output program **If2**

```
D:\semester 4\PBO\Tugas 2>javac If2.java

D:\semester 4\PBO\Tugas 2>java If2.java
Contoh IF dua kasus
Ketikkan suatu nilai integer :9
Nilai a positif 9

D:\semester 4\PBO\Tugas 2>java If2.java
Contoh IF dua kasus
Ketikkan suatu nilai integer :-2
Nilai a negatif -2
```

Keterangan :

Program ini adalah contoh struktur IF dua kasus komplementer. Program meminta pengguna memasukkan bilangan integer menggunakan Scanner, lalu mengevaluasi nilainya dengan dua kemungkinan:

- Jika bilangan lebih dari atau sama dengan nol ($a \geq 0$) → Program mencetak bahwa bilangan tersebut positif.
- Jika bilangan kurang dari nol ($a < 0$) → Program mencetak bahwa bilangan tersebut negatif.

Karena menggunakan if-else, program dapat menangani dua kondisi yang saling berlawanan (komplementer), sehingga setiap nilai yang dimasukkan akan mendapatkan output yang sesuai.

5. Output program **If3**

```
D:\semester 4\PBO\Tugas 2>javac If3.java

D:\semester 4\PBO\Tugas 2>java If3.java
Contoh IF tiga kasus
Ketikkan suatu nilai integer :12
Nilai a positif 12

D:\semester 4\PBO\Tugas 2>java If3.java
Contoh IF tiga kasus
Ketikkan suatu nilai integer :0
Nilai Nol 0

D:\semester 4\PBO\Tugas 2>java If3.java
Contoh IF tiga kasus
Ketikkan suatu nilai integer :-6
Nilai a negatif -6
```

Keterangan :

Program ini adalah contoh IF tiga kasus. Program meminta pengguna memasukkan bilangan integer menggunakan Scanner, lalu mengevaluasi nilainya dengan tiga kemungkinan:

- Jika bilangan lebih dari nol ($a > 0$) → Program mencetak bahwa bilangan tersebut positif.
- Jika bilangan sama dengan nol ($a == 0$) → Program mencetak bahwa bilangan tersebut nol.
- Jika bilangan kurang dari nol ($a < 0$) → Program mencetak bahwa bilangan tersebut negatif.

Struktur if-else memastikan bahwa setiap kemungkinan nilai integer dapat ditangani dengan benar, sehingga program memberikan output yang sesuai untuk semua kasus.

6. Output program **KasusBoolean**

```
D:\semester 4\PB0\Tugas 2>javac KasusBoolean.java

D:\semester 4\PB0\Tugas 2>java KasusBoolean.java
true
benar
```

Keterangan :

Program ini menggunakan ekspresi kondisional dengan tipe data boolean untuk menentukan output berdasarkan nilai bool. Variabel bool diinisialisasi dengan true, lalu diperiksa dalam dua kondisi if-else. Kondisi pertama mencetak "true" karena bool bernilai true. Kondisi kedua menggunakan negasi !bool, yang berarti jika bool false, maka akan mencetak "salah", tetapi karena bool bernilai true, program mencetak "benar"

7. Output program **KasusSwitch**

```
D:\semester 4\PB0\Tugas 2>javac KasusSwitch.java

D:\semester 4\PB0\Tugas 2>java KasusSwitch.java
Ketikkan sebuah huruf, akhiri dengan RETURN
a
Yang anda ketik adalah a

D:\semester 4\PB0\Tugas 2>java KasusSwitch.java
Ketikkan sebuah huruf, akhiri dengan RETURN
C
Yang anda ketik adalah huruf mati
```

Keterangan :

Program ini menggunakan percabangan switch-case untuk menentukan jenis huruf yang diketik oleh pengguna. Pengguna diminta memasukkan satu karakter melalui Scanner, lalu karakter tersebut diperiksa dalam switch. Jika karakter yang dimasukkan adalah salah satu dari 'a', 'i', 'u', 'e', 'o', program akan mencetak bahwa huruf tersebut adalah vokal sesuai dengan input. Jika karakter yang dimasukkan bukan huruf vokal, maka akan masuk ke default dan mencetak "Yang anda ketik adalah huruf mati". Program ini berguna untuk mengenali huruf vokal dan konsonan dalam input pengguna.

8. Output program **Kontsant**

```
D:\semester 4\PB0\Tugas 2>javac Konstant.java

D:\semester 4\PB0\Tugas 2>java Konstant.java
Jari-jari lingkaran =7
Luas lingkaran = 153.9335
Akhir program
```

Keterangan :

Program ini menghitung luas lingkaran dengan memanfaatkan konstanta PHI. Pengguna diminta memasukkan jari-jari lingkaran, lalu program menggunakan rumus $\pi \times r^2$ untuk menghitung luasnya. Nilai π (PHI) ditetapkan sebagai 3.1415 menggunakan kata kunci final, yang berarti nilainya tidak bisa diubah. Setelah perhitungan, program mencetak hasil luas lingkaran dan menampilkan pesan akhir. Program ini merupakan contoh penggunaan konstanta dalam perhitungan matematika di Java.

9. Output program **Max2**

```
D:\semester 4\PB0\Tugas 2>javac Max2.java

D:\semester 4\PB0\Tugas 2>java Max2.java
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
14
7
Ke dua bilangan : a = 14 b = 7
Nilai a yang maksimum 14
```

Keterangan :

Program ini menentukan bilangan maksimum dari dua angka yang dimasukkan oleh pengguna. Setelah meminta input dua bilangan integer, program membandingkan kedua nilai tersebut menggunakan struktur if-else. Jika a lebih besar atau sama dengan b, maka a adalah bilangan maksimum, sedangkan jika b lebih besar dari a, maka b adalah bilangan maksimum. Program ini menunjukkan cara membandingkan dua angka dan mencetak hasilnya, yang berguna dalam berbagai aplikasi pemrograman yang membutuhkan perbandingan nilai numerik.

10. Output program **PriFor**

```
D:\semester 4\PB0\Tugas 2>javac PriFor.java

D:\semester 4\PB0\Tugas 2>java PriFor.java
Baca N, print 1 s/d N N = 5
1
2
3
4
5
Akhir program
```

Keterangan :

Program ini mencetak bilangan dari 1 hingga N menggunakan perulangan for. Setelah pengguna memasukkan nilai N, program akan menjalankan perulangan mulai dari 1 hingga N, mencetak setiap angka secara berurutan. Proses ini dilakukan dengan increment (i++) pada setiap iterasi. Program ini merupakan contoh dasar penggunaan looping dalam Java, yang sering digunakan untuk mencetak deret angka, menghitung jumlah, atau mengolah data dalam jumlah tertentu.

11. Output program **PrintIterasi**

```
D:\semester 4\PB0\Tugas 2>java PrintIterasi.java
Nilai N >0 = 4
Print i dengan ITERATE :
1
2
3
4

D:\semester 4\PB0\Tugas 2>java PrintIterasi.java
Nilai N >0 = 6
Print i dengan ITERATE :
1
2
3
4
5
6
```

Keterangan :

Program PrintIterasi membaca input N dari pengguna, lalu mencetak angka dari 1 hingga N menggunakan perulangan tak terbatas (for(;;)) yang dikendalikan secara manual. Variabel i diinisialisasi dengan 1 dan setiap iterasi mencetak nilai i, kemudian diperiksa

apakah $i == N$. Jika iya, perulangan dihentikan dengan break, sedangkan jika tidak, i ditambah satu ($i++$) untuk iterasi berikutnya. Pendekatan ini memungkinkan kontrol manual terhadap perulangan tanpa batas, meskipun dalam praktiknya bisa digantikan dengan for atau while yang lebih eksplisit.

12. Output program **PrintRepeat**

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

D:\semester 4\PBO\Tugas 2>java PrintRepeat.java
Nilai N >0 = 1
Print i dengan REPEAT:
1

D:\semester 4\PBO\Tugas 2>java PrintRepeat.java
Nilai N >0 = 0
Print i dengan REPEAT:
1
```

Keterangan :

Program PrintRepeat membaca input N dari pengguna dan mencetak angka dari 1 hingga N menggunakan perulangan do-while. Variabel i diinisialisasi dengan 1 sebagai elemen pertama, kemudian dalam blok do, nilai i dicetak dan ditingkatkan ($i++$). Perulangan akan terus berjalan selama kondisi $i \leq N$ terpenuhi. Pendekatan ini memastikan bahwa setidaknya satu iterasi terjadi, bahkan jika nilai N yang dimasukkan kurang dari atau sama dengan nol.

13. Output program **PrintWhile**

```
D:\semester 4\PBO\Tugas 2>javac PrintWhile.java

D:\semester 4\PBO\Tugas 2>java PrintWhile.java
Nilai N >0 = 4
Print i dengan WHILE:
1
2
3
4

D:\semester 4\PBO\Tugas 2>java PrintWhile.java
Nilai N >0 = 1
Print i dengan WHILE:
1

D:\semester 4\PBO\Tugas 2>java PrintWhile.java
Nilai N >0 = 0
Print i dengan WHILE:
```

Keterangan :

Program PrintWhile membaca input N dari pengguna dan mencetak angka dari 1 hingga N menggunakan perulangan while. Variabel i diinisialisasi dengan 1 sebagai elemen pertama, kemudian selama kondisi $i \leq N$ terpenuhi, program mencetak nilai i dan meningkatkannya ($i++$). Perulangan berhenti ketika nilai i melebihi N, sehingga memastikan semua angka dari 1 hingga N dicetak secara berurutan.

14. Output program **PrintWhile1**

```
D:\semester 4\PBO\Tugas 2>javac PrintWhile1.java

D:\semester 4\PBO\Tugas 2>java PrintWhile1.java
Nilai N >0 = 4
Print i dengan WHILE (ringkas):
1
2
3
4

D:\semester 4\PBO\Tugas 2>java PrintWhile1.java
Nilai N >0 = 1
Print i dengan WHILE (ringkas):
1

D:\semester 4\PBO\Tugas 2>java PrintWhile1.java
Nilai N >0 = 0
Print i dengan WHILE (ringkas):
```

Keterangan :

Program PrintWhile1 membaca input N dari pengguna dan mencetak angka dari 1 hingga N menggunakan perulangan while dalam format yang lebih ringkas. Variabel i diinisialisasi dengan 1, kemudian selama $i \leq N$, program mencetak nilai i dan langsung meningkatkannya dalam satu perintah (`System.out.println(i++)`). Struktur ini mengurangi jumlah baris kode tanpa mengubah logika program, memastikan bahwa semua angka dari 1 hingga N tetap dicetak secara berurutan.

15. Output program **PrintXinterasi**

```
D:\semester 4\PBO\Tugas 2>java PrintXinterasi.java
Masukkan nilai x (int), akhiri dg 999 : 999
Kasus kosong
Hasil penjumlahan = 0

D:\semester 4\PBO\Tugas 2>java PrintXinterasi.java
Masukkan nilai x (int), akhiri dg 999 : 12
Masukkan nilai x (int), akhiri dg 999 : 19
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 31
```

Keterangan :

Program PrintXinterasi membaca sejumlah bilangan bulat dari pengguna dan menjumlahkannya hingga pengguna memasukkan angka 999 sebagai tanda berhenti. Program dimulai dengan membaca nilai pertama x, jika x langsung 999, maka akan mencetak "Kasus kosong" dan berhenti. Jika tidak, nilai pertama tersebut dijadikan nilai awal Sum. Selanjutnya, program memasuki perulangan `for (;) { }`, yang berjalan tanpa batas hingga 999 dimasukkan. Setiap angka yang dimasukkan akan ditambahkan ke Sum, dan setelah 999 dimasukkan, hasil penjumlahan akan ditampilkan.

16. Output program **PrintRepeat**

```
D:\semester 4\PBO\Tugas 2>java PrintXRepeat.java
Masukkan nilai x (int), akhiri dg 999 : 999
Kasus kosong

D:\semester 4\PBO\Tugas 2>java PrintXRepeat.java
Masukkan nilai x (int), akhiri dg 999 : 10
Masukkan nilai x (int), akhiri dg 999 : 12
Masukkan nilai x (int), akhiri dg 999 : 13
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 35
```

Keterangan :

Program PrintXRepeat membaca serangkaian bilangan bulat dari pengguna dan menjumlahkannya menggunakan perulangan do-while hingga pengguna memasukkan 999 sebagai tanda berhenti. Jika nilai pertama yang dimasukkan adalah 999, program akan

langsung mencetak "Kasus kosong" dan berhenti. Jika tidak, program akan menginisialisasi $\text{Sum} = 0$ dan memasuki perulangan do-while, yang terus menjumlahkan nilai x ke dalam Sum hingga pengguna memasukkan 999. Setelah perulangan berakhir, hasil penjumlahan akan ditampilkan ke layar.

17. Output program **PrintXWhile**

```
D:\semester 4\PBO\Tugas 2>javac PrintXWhile.java

D:\semester 4\PBO\Tugas 2>java PrintXWhile.java
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 0

D:\semester 4\PBO\Tugas 2>java PrintXWhile.java
Masukkan nilai x (int), akhiri dg 999 : 10
Masukkan nilai x (int), akhiri dg 999 : 20
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 30
```

Keterangan :

Program **PrintXWhile** membaca serangkaian bilangan bulat dari pengguna dan menjumlahkannya menggunakan perulangan while hingga pengguna memasukkan 999 sebagai tanda berhenti. Program dimulai dengan menginisialisasi $\text{Sum} = 0$, lalu meminta pengguna memasukkan bilangan pertama. Jika bilangan yang dimasukkan bukan 999, program akan masuk ke dalam perulangan while, menjumlahkan setiap angka yang dimasukkan ke dalam Sum , dan terus meminta input baru hingga pengguna memasukkan 999. Setelah perulangan berhenti, program mencetak hasil penjumlahan ke layar.

18. Output program **SubProgram**

```
D:\semester 4\PBO\Tugas 2>javac SubProgram.java

D:\semester 4\PBO\Tugas 2>java SubProgram.java
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
5
6
Ke dua bilangan : a = 5 b = 6
Maksimum = 6
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 6 b = 5
```

Keterangan :

Program **SubProgram** mendemonstrasikan penggunaan prosedur dan fungsi dalam Java untuk mengolah dua bilangan bulat. Program ini memiliki fungsi `maxab(int a, int b)` yang mengembalikan nilai maksimum dari dua bilangan menggunakan operator ternary, serta prosedur `tukar(int a, int b)` yang mencoba menukar nilai dua bilangan tetapi tidak mempengaruhi nilai di luar prosedur karena Java menggunakan pass by value. Dalam main, pengguna diminta memasukkan dua bilangan, lalu program mencetak bilangan maksimum menggunakan fungsi `maxab`, kemudian mencoba menukar nilai menggunakan prosedur `tukar`, namun perubahan tidak terjadi di luar prosedur karena nilai hanya ditukar dalam cakupan lokal prosedur tersebut.

19. Output program **Tempair**


```
D:\semester 4\PBO\Tugas 2>javac Tempair.java
D:\semester 4\PBO\Tugas 2>java Tempair.java
Contoh IF tiga kasus
Temperatur (der. C) = -5
Wujud air beku
-5
D:\semester 4\PBO\Tugas 2>java Tempair.java
Contoh IF tiga kasus
Temperatur (der. C) = 37
Wujud air cair
37
D:\semester 4\PBO\Tugas 2>java Tempair.java
Contoh IF tiga kasus
Temperatur (der. C) = 110
Wujud air uap/gas
110
```

Keterangan :

Program Tempair menentukan wujud air berdasarkan suhu yang dimasukkan oleh pengguna menggunakan struktur if-else dengan tiga kondisi utama. Pengguna diminta memasukkan temperatur dalam derajat Celsius, lalu program mengevaluasi kondisi: jika suhu kurang dari 0°C, air berada dalam bentuk beku; jika suhu antara 0°C dan 100°C (termasuk 0 dan 100), air berada dalam bentuk cair; dan jika suhu lebih dari 100°C, air berada dalam bentuk uap/gas. Hasilnya kemudian dicetak ke layar sesuai dengan kondisi yang terpenuhi.