

Tugas II

Makassar, 13 Maret 2025

TEORI
PEMROGRAMAN BERORIENTASI OBJEK



Nama : Alexandria Kayla Kazya Putri Nur
Nim : 13020230001
Dosen : Mardiyah Hasnawi, S.Kom., M.T.

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

Tugas Program

Program 1

Output :

```
D:\Tugas 2\source code>javac BacaString.java
```

```
D:\Tugas 2\source code>java BacaString
```

```
Baca string dan Integer:
masukkan sebuah string: alexa
String yang dibaca : alexa
```

```
D:\Tugas 2\source code>javac BacaString.java
```

```
D:\Tugas 2\source code>java BacaString
```

```
Baca string dan Integer:
masukkan sebuah string: 2
String yang dibaca : 2
```

Keterangan :

Pada output pertama, saya memasukkan "alexa", sehingga program mencetak String yang dibaca "alexa". Pada output kedua, saya memasukkan "2", yang meskipun merupakan angka, tetap dianggap sebagai string karena tidak ada proses konversi ke tipe data numerik. Oleh karena itu, program tetap mencetak String yang dibaca "2".

Program 2

Output :

[illegible]

Keterangan :

Output menunjukkan perintah mencetak "Print satu baris" berulang kali, Ini akibat dari looping yang terus menjalankan perintah cetak dan akan berakhir jika menekan ctrl + c.

Program 3

Output :

```
D:\Tugas 2\source code>javac If1.java

D:\Tugas 2\source code>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : 1

Nilai a positif 1
```

Keterangan :

Program ini menggunakan struktur control if untuk menangani satu kondisi. saya diminta memasukkan nilai integer, dan jika nilai tersebut positif, program mencetak bahwa nilai tersebut positif beserta angkanya. Pada eksekusi ini, saya memasukkan angka 1, sehingga program menampilkan output "Nilai a positif 1".

Program 4

Output :

```
D:\Tugas 2\source code>javac If2.java

D:\Tugas 2\source code>java If2
Contoh IF dua kasus
Ketikkan suatu nilai integer :2
Nilai a positif 2

D:\Tugas 2\source code>javac If2.java

D:\Tugas 2\source code>java If2
Contoh IF dua kasus
Ketikkan suatu nilai integer :-2
Nilai a negatif -2
```

Keterangan:

Program ini menggunakan struktur if else untuk menangani dua kondisi. Jika saya memasukkan bilangan positif, program mencetak bahwa nilainya positif, dan jika memasukkan bilangan negatif, program mencetak bahwa nilainya negatif. Pada output pertama, input 2 menghasilkan output "Nilai a positif 2", sedangkan pada output kedua, input -2 menghasilkan output "Nilai a negatif -2".

Program 5

Output :

```
D:\Tugas 2\source code>javac If3.java

D:\Tugas 2\source code>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer :3
Nilai a positif 3
```

```
D:\Tugas 2\source code>javac If3.java
```

```
D:\Tugas 2\source code>java If3  
Contoh IF tiga kasus  
Ketikkan suatu nilai integer :-3  
Nilai a negatif -3
```

```
D:\Tugas 2\source code>javac If3.java
```

```
D:\Tugas 2\source code>java If3  
Contoh IF tiga kasus  
Ketikkan suatu nilai integer :0  
Nilai Nol 0
```

Keterangan :

Program ini menggunakan struktur if else – if else untuk menangani tiga kondisi bilangan positif, negatif, dan nol. Jika saya memasukkan angka positif, program mencetak bahwa nilainya positif jika negatif, program mencetak bahwa nilainya negative dan jika nol, program mencetak "Nilai Nol". Pada output pertama dengan input 3, output menunjukkan "Nilai a positif 3", pada output kedua dengan -3, output "Nilai a negatif -3", dan pada output ketiga dengan 0 , output "Nilai Nol 0".

Program 6

Output :

```
D:\Tugas 2\source code>javac KasusBoolean.java  
  
D:\Tugas 2\source code>java KasusBoolean  
true  
benar
```

Keterangan :

Program ini menunjukkan penggunaan tipe data Boolean dalam Java. Program mencetak nilai true, yang kemungkinan berasal dari sebuah variabel boolean atau hasil dari suatu kondisi logika. Setelah itu, program mencetak "benar", yang mungkin merupakan representasi dari nilai boolean dalam bentuk teks menggunakan struktur kontrol seperti if.

Program 7

Output :

```
D:\Tugas 2\source code>javac KasusSwitch.java  
  
D:\Tugas 2\source code>java KasusSwitch  
Ketikkan sebuah huruf, akhiri dengan RETURN  
b  
Yang anda ketik adalah huruf mati  
  
D:\Tugas 2\source code>javac KasusSwitch.java  
  
D:\Tugas 2\source code>java KasusSwitch  
Ketikkan sebuah huruf, akhiri dengan RETURN  
o  
Yang anda ketik adalah o
```

Keterangan :

Program ini menggunakan struktur switch dalam Java untuk menangani berbagai kemungkinan input huruf. Jika saya memasukkan huruf tertentu, program mencetak kategori atau deskripsi terkait. Dalam eksekusi pertama, input b dikategorikan sebagai "huruf mati", sedangkan pada output kedua, input o hanya ditampilkan kembali tanpa kategori khusus karena o ada dalam source code nya.

Program 8

Output :

```
D:\Tugas 2\source code>javac Konstant.java

D:\Tugas 2\source code>java Konstant
Jari-jari lingkaran =11
Luas lingkaran = 380.1215
Akhir program
```

Keterangan :

Program ini menghitung luas lingkaran berdasarkan jari-jari yang diberikan, yaitu 11. Dengan menggunakan rumus luas lingkaran $\pi \times r^2$, program kemungkinan menggunakan nilai π (pi) sebagai konstanta, menghasilkan output luas lingkaran sebesar 380.1215. Setelah menampilkan hasil perhitungan, program mencetak "Akhir program" sebagai tanda selesai.

Program 9

Output :

```
D:\Tugas 2\source code>javac Max2.java

D:\Tugas 2\source code>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
10
11
Ke dua bilangan : a = 10 b = 11
Nilai b yang maksimum: 11
```

Keterangan :

Program ini membandingkan dua bilangan yang dimasukkan, yaitu 10 dan 11. Program kemudian menentukan dan menampilkan bilangan terbesar di antara keduanya, dalam hal ini angka 11.

Program 10

Output :

```

D:\Tugas 2\source code>javac PriFor.java

D:\Tugas 2\source code>java PriFor
Baca N, print 1 s/d N N = 10
1
2
3
4
5
6
7
8
9
10
Akhir program

```

Keterangan :

Program ini menggunakan perulangan for untuk mencetak bilangan dari 1 hingga N, di mana N = 10 sesuai dengan input pengguna. Program membaca nilai N, mencetak angka secara berurutan, lalu mengakhiri eksekusi dengan menampilkan pesan "Akhir program".

Program 11

Output :

```

D:\Tugas 2\source code>javac PrintIterasi.java

D:\Tugas 2\source code>java PrintIterasi
Nilai N >0 = 2
Print i dengan ITERATE :
1
2

```

```

D:\Tugas 2\source code>java PrintIterasi
Nilai N >0 = 0
Print i dengan ITERATE :
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21

```

Keterangan :

Program ini membaca nilai N yang lebih dari 0, dalam hal ini N = 2. Kemudian, program mencetak angka dari 1 hingga N menggunakan iterasi dengan perulangan for. Hasilnya, angka 1 dan 2 ditampilkan secara berurutan. Output kedua menunjukkan iterasi mencetak angka dari 1 sampai di hentikan dengan menekan ctrl + c meskipun nilai N diberikan sebagai 0. Ini menunjukkan adanya kemungkinan

kesalahan logika dalam program, di mana perulangan tetap berjalan meskipun N tidak lebih dari 0. Program mungkin menggunakan loop tak terbatas atau tidak memiliki pengecekan kondisi yang benar untuk N = 0.

Program 12

Output :

```
D:\Tugas 2\source code>javac PrintRepeat.java

D:\Tugas 2\source code>java PrintRepeat
Nilai N >0 = 10
Print i dengan REPEAT:
1
2
3
4
5
6
7
8
9
10
```

```
D:\Tugas 2\source code>javac PrintRepeat.java

D:\Tugas 2\source code>java PrintRepeat
Nilai N >0 = 0
Print i dengan REPEAT:
1
```

Keterangan :

program ini menunjukkan penggunaan perulangan untuk mencetak angka dari 1 hingga N. Pada eksekusi pertama, N = 10, sehingga program mencetak angka 1 hingga 10. Pada eksekusi kedua, N = 0, tetapi program tetap mencetak 1. Hal ini mengindikasikan bahwa perulangan yang digunakan adalah repeat (do-while loop), yang mengeksekusi setidaknya sekali sebelum mengecek kondisi, sehingga angka 1 tetap dicetak meskipun N = 0.

Program 13

Output :

```
D:\Tugas 2\source code>javac PrintWhile.java

D:\Tugas 2\source code>java PrintWhile
Nilai N >0 = 5
Print i dengan WHILE:
1
2
3
4
5
```

```
D:\Tugas 2\source code>javac PrintWhile.java

D:\Tugas 2\source code>java PrintWhile
Nilai N >0 = 0
Print i dengan WHILE:
```

Keterangan :

program ini menunjukkan penggunaan perulangan while untuk mencetak angka dari 1 hingga N. Pada eksekusi pertama, $N = 5$, sehingga program mencetak angka 1 hingga 5. Pada eksekusi kedua $N = 0$, sehingga tidak ada angka yang dicetak setelah pesan "Print i dengan WHILE:". Hal ini terjadi karena perulangan while hanya akan dieksekusi jika kondisi awalnya terpenuhi ($N > 0$), sehingga jika $N = 0$, perulangan tidak berjalan sama sekali.

Program 14

Output :

```
D:\Tugas 2\source code>java PrintWhile1
Nilai N >0 = 7
Print i dengan WHILE (ringkas):
1
2
3
4
5
6
7

D:\Tugas 2\source code>javac PrintWhile1.java

D:\Tugas 2\source code>java PrintWhile1
Nilai N >0 = 0
Print i dengan WHILE (ringkas):
```

Keterangan :

Program ini membaca input N dari pengguna dan mencetak angka dari 1 hingga N menggunakan perulangan while. Variabel i diinisialisasi dengan 1 dan langsung diinkrementasi dalam perintah `System.out.println(i++)`, sehingga nilai i dicetak terlebih dahulu sebelum bertambah. Jika $N = 7$, program mencetak angka 1 hingga 7. Jika $N = 0$, perulangan tidak berjalan, sehingga tidak ada angka yang ditampilkan setelah teks "Print i dengan WHILE (ringkas):".

Program 15

Output :

```
D:\Tugas 2\source code>javac PrintXiterasi.java

D:\Tugas 2\source code>java PrintXiterasi
Masukkan nilai x (int), akhiri dg 999: 5
Masukkan nilai x (int), akhir dg 999 : 2
Masukkan nilai x (int), akhir dg 999 : 3
Masukkan nilai x (int), akhir dg 999 : 2
Masukkan nilai x (int), akhir dg 999 : 1
Masukkan nilai x (int), akhir dg 999 : 999
Hasil penjumlahan = 13
```



```
D:\Tugas 2\source code>javac PrintXiterasi.java

D:\Tugas 2\source code>java PrintXiterasi
Masukkan nilai x (int), akhiri dg 999: 999
Kasus kosong
Hasil penjumlahan = 0
```

Keterangan :

Program ini menghitung jumlah dari serangkaian bilangan bulat yang dimasukkan oleh pengguna, dengan 999 sebagai penanda akhir input. Jika pengguna langsung memasukkan 999, program menampilkan "Kasus kosong" dan hasil penjumlahan 0. Jika pengguna memasukkan angka lain sebelum 999, program menjumlahkan angka-angka tersebut hingga 999 dimasukkan, lalu mencetak hasilnya. Misalnya, jika pengguna memasukkan 5, 2, 3, 2, 1 lalu 999, hasil penjumlahan adalah 13.

Program 16

Output :

```
D:\Tugas 2\source code>javac PrintXRepeat.java

D:\Tugas 2\source code>java PrintXRepeat
Masukkan nilai x (int), akhiri dg 999 : 10
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 = 15
```

```
D:\Tugas 2\source code>javac PrintXRepeat.java

D:\Tugas 2\source code>java PrintXRepeat
Masukkan nilai x (int), akhiri dg 999 : 999
Kasus kosong
```

Keterangan :

Program ini menjumlahkan serangkaian bilangan bulat yang dimasukkan oleh pengguna, dengan 999 sebagai tanda berhenti. Jika pengguna langsung memasukkan 999, program menampilkan "Kasus kosong" dan berhenti. Jika pengguna memasukkan angka lain, program menggunakan do-while loop untuk terus meminta input, menambahkan nilai ke Sum, hingga pengguna memasukkan 999. Misalnya, jika pengguna memasukkan 10, 2, 3 lalu 999, hasil penjumlahan adalah 15.

Program 17

Output :

```

D:\Tugas 2\source code>javac PrintXWhile.java

D:\Tugas 2\source code>java PrintXWhile
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 : 4
Masukkan nilai x (int), akhiri dg 999 : 5
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 15

D:\Tugas 2\source code>javac PrintXWhile.java

D:\Tugas 2\source code>java PrintXWhile
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 0

```

Keterangan :

Program ini menjumlahkan serangkaian bilangan yang dimasukkan pengguna hingga angka 999 dimasukkan sebagai tanda berhenti. Program menggunakan while loop untuk terus meminta input dan menambahkan nilai ke Sum selama input bukan 999. Jika pengguna langsung memasukkan 999, hasil penjumlahan adalah 0. Sebaliknya, jika pengguna memasukkan angka seperti 1, 2, 3, 4, 5 sebelum 999, hasil penjumlahan adalah 15.

Program 18

Output :

```

D:\Tugas 2\source code>javac SubProgram.java

D:\Tugas 2\source code>java SubProgram
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
5
8
Ke dua bilangan : a =5 b = 8
Maksimum = 8
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 8 b = 5

```

Keterangan :

Program ini meminta pengguna memasukkan dua bilangan, lalu menampilkan nilai maksimum dari keduanya menggunakan fungsi maxab(). Setelah itu, program mencoba menukar nilai kedua bilangan menggunakan fungsi tukar(). Namun, karena Java menggunakan pass by value, perubahan dalam fungsi tukar() tidak mempengaruhi nilai variabel asli di main(). Oleh karena itu, meskipun hasil pertukaran ditampilkan dalam fungsi tukar(), nilai asli di main() tetap tidak berubah.

Program 19

Output :

```
D:\Tugas 2\source code>javac Tempair.java

D:\Tugas 2\source code>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = -10
Wujud air beku
-10
D:\Tugas 2\source code>javac Tempair.java

D:\Tugas 2\source code>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 80
Wujud air cair
80
D:\Tugas 2\source code>javac Tempair.java

D:\Tugas 2\source code>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 150
Wujud air uap/gas
150
```

Keterangan :

Program ini menentukan wujud air berdasarkan suhu yang dimasukkan pengguna. Jika suhu di bawah 0°C, program menampilkan "Wujud air beku". Jika suhu antara 0°C hingga 100°C (inklusif), program menampilkan "Wujud air cair". Jika suhu lebih dari 100°C, program menampilkan "Wujud air uap/gas". Output yang dihasilkan sesuai dengan nilai yang dimasukkan, menunjukkan perubahan wujud air berdasarkan suhu yang diberikan.