

LAPORAN
PEMROGRAMAN BERORIENTASI OBJEK



Nama : Andi Shavira Andriani.S
Stambuk : 13020230029
Kelas : B1
Dosen : Mardiyah Hasnawi, S.Kom., M.T.

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

1. Output:

```
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>javac BacaString.java
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java BacaString

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

Penjelasan:

Program BacaString dalam Java membaca input string dari pengguna menggunakan `BufferedReader`. Program dimulai dengan mengimpor library yang diperlukan, lalu mendeklarasikan kelas `BacaString` dan metode `main()`. Variabel `str` digunakan untuk menyimpan input, sementara `BufferedReader` membaca string dari keyboard dengan `readLine()`. Setelah pengguna memasukkan string, program menampilkannya kembali di layar. Program ini tidak membaca integer karena hanya menggunakan `BufferedReader.readLine()`, yang selalu membaca input sebagai string. Tidak ada konversi dari string ke integer menggunakan `Integer.parseInt()` atau metode lain yang memungkinkan pembacaan angka.

2. Output:

[illegible]

Penjelasan:

Program ForEver ini adalah program infinite loop (perulangan tak terbatas) karena menggunakan while (true), yang selalu bernilai true sehingga tidak akan pernah berhenti kecuali dihentikan secara paksa. Program ini akan terus mencetak teks tanpa henti kecuali dihentikan secara paksa (Ctrl + C di terminal atau menghentikan proses program).

3. Output:

```
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>javac If1.java

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : 96

Nilai a positif 96
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : -96
```

Penjelasan:

Program If1 dalam Java ini menunjukkan contoh penggunaan struktur IF satu kasus, yaitu hanya menjalankan suatu perintah jika kondisi tertentu terpenuhi. Program ini hanya menampilkan angka jika nilainya positif atau nol. Jika negatif, program tidak memberikan output tambahan.

4. Output:

```
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>javac If2.java

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java If2
Contoh IF dua kasus
Ketikkan suatu nilai integer :23
Nilai a positif 23

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java If2
Contoh IF dua kasus
Ketikkan suatu nilai integer :-23
Nilai a negatif -23
```

Penjelasan:

Program If2 dalam Java ini adalah contoh penggunaan struktur IF dua kasus komplementer, yaitu menangani dua kemungkinan: ketika nilai positif atau nol ($a \geq 0$) dan ketika nilai negatif ($a < 0$). Ketika nilainya yang diinput merupakan nilai positif maka output programnya adalah "Nilai a positif (angka yang diinput)" sebaliknya jika yang diinput angka negatif maka output programnya adalah "Nilai a negatif (angka yang diinput)"

5. Output:

```
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer :17
Nilai a positif 17

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer :-17
Nilai a negatif -17

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer :0
Nilai Nol 0
```

Penjelasan:

Program If3 adalah contoh Penggunaan IF pada 3 kasus, yaitu menangani tiga kemungkinan nilai integer: positif ($a > 0$), nol ($a == 0$), dan negatif ($a < 0$). Sehingga ketika diinputkan nilai positif maka output programnya “Nilai a positif (nilai yang diinputkan)”, ketika diinputkan nilai 0 maka output programnya “Nilai Nol 0”, sedangkan ketika diinputkan nilai negatif maka output programnya “Nilai a negatif (nilai yang diinputkan)”.

6. Output:

```
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>javac KasusBoolean.java

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java KasusBoolean
true
benar
```

Penjelasan:

Program KasusBoolean ini menunjukkan penggunaan ekspresi kondisional dengan tipe data boolean dalam Java. Program ini mendemonstrasikan bagaimana nilai boolean digunakan dalam struktur kontrol IF-ELSE. Pada percabangan pertama, karena bool bernilai true, maka output program adalah "true". Pada percabangan kedua, kondisi !bool (negasi dari true) bernilai false, sehingga ketika else dijalankan maka output dari program tersebut adalah "benar".

7. Output:

```
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>javac KasusSwitch.java  
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java KasusSwitch  
Ketikkan sebuah huruf, akhiri dengan ENTER: a  
Yang anda ketik adalah a  
  
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java KasusSwitch  
Ketikkan sebuah huruf, akhiri dengan ENTER: m  
Yang anda ketik adalah huruf mati (konsonan)
```

Penjelasan:

Program ini menggunakan switch-case untuk mengevaluasi sebuah karakter (char) yang dimasukkan oleh pengguna. Program menentukan apakah karakter tersebut merupakan huruf vokal (a, i, u, e, o) atau huruf konsonan (huruf mati).

8. Output:

```
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>javac Konstant.java  
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java Konstant  
Jari-jari lingkaran =5  
Luas lingkaran = 78.537506  
Akhir program
```

Penjelasan:

Program ini membaca jari-jari lingkaran yang dimasukkan oleh pengguna, kemudian menghitung dan menampilkan luas lingkaran dengan menggunakan konstanta PHI. Dengan menggunakan scanner untuk membaca nilai yang diinputkan.

9. Output:

```
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>javac Max2.java  
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java Max2  
Maksimum dua bilangan:  
Ketikkan dua bilangan, pisahkan dengan ENTER: 5  
17  
Ke dua bilangan: a = 5 b = 17  
Nilai maksimum: 17
```

Penjelasan:

Program ini membandingkan dua bilangan yang diinputkan dan menampilkan output bilangan maksimum diantara keduanya.

10. Output:

```
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>javac PriFor.java

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java PriFor
Baca N, print 1 s/d N N = 17
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
Akhir program
```

Penjelasan:

Program ini melakukan looping dari angka 1 hingga angka n (angka yang diinputkan) sehingga ketika angka n sudah tercapai maka looping akan berhenti dan program akan berakhir.

11. Output:

```
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintIterasi
Nilai N >0 = 7
Print i dengan ITERATE :
1
2
3
4
5
6
7

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintIterasi
Nilai N >0 = -7
Print i dengan ITERATE :
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

Penjelasan:

Program ini membaca nilai N dari pengguna dan mencetak angka dari 1 hingga N menggunakan perulangan tak terbatas (for(;;)), yang hanya berhenti jika kondisi $i == N$ terpenuhi. Variabel i dimulai dari 1 dan bertambah setiap iterasi. Jika $N > 0$, program berjalan normal dan mencetak angka dari 1 hingga N. Namun, jika $N < 1$, kondisi $i == N$ tidak akan pernah terpenuhi karena i dimulai dari 1 dan terus bertambah, sedangkan N lebih kecil dari 1, sehingga perulangan berjalan tanpa henti (infinite loop).

12. Output:

```
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>javac PrintRepeat.java

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintRepeat
Nilai N >0 = 5
Print i dengan REPEAT:
1
2
3
4
5

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintRepeat
Nilai N >0 = 0
Print i dengan REPEAT:
1
```

Penjelasan:

Program ini membaca nilai N dari pengguna dan mencetak angka dari 1 hingga N menggunakan perulangan do-while. Variabel i dimulai dari 1 dan selalu dijalankan minimal sekali sebelum kondisi diperiksa ($i \leq N$). Jika $N > 0$, program mencetak angka dari 1 hingga N. Namun, jika $N \leq 0$, perulangan tetap berjalan minimal satu kali, mencetak angka 1, lalu berhenti karena kondisi $i \leq N$ tidak terpenuhi. Hal ini terjadi karena perulangan do-while selalu menjalankan blok kode setidaknya sekali sebelum ia mengevaluasi kondisinya.

13. Output:

```
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>javac PrintWhile.java

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintWhile
Nilai N >0 = 8
Print i dengan WHILE:
1
2
3
4
5
6
7
8

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintWhile
Nilai N >0 = -2
Print i dengan WHILE:
```

Penjelasan:

Program ini membaca nilai N dari pengguna dan mencetak angka dari 1 hingga N menggunakan perulangan while. Jika $N > 0$, program berjalan normal dan mencetak angka 1 hingga N. Namun, jika $N \leq 0$, kondisi $i \leq N$ langsung salah sejak awal, sehingga perulangan tidak akan berjalan sama sekali, dan program tidak mencetak angka apa pun.

14. Output:

```
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>javac PrintWhile1.java

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintWhile1
Nilai N >0 = 7
Print i dengan WHILE (ringkas):
1
2
3
4
5
6
7

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintWhile1
Nilai N >0 = 0
Print i dengan WHILE (ringkas):
```

Penjelasan:

Program ini membaca nilai N dari pengguna dan mencetak angka dari 1 hingga N menggunakan perulangan while dalam bentuk yang lebih ringkas. Pada setiap iterasi, program mencetak i, kemudian langsung menambah nilai i ($i++$) dalam perintah cetak. Jika $N > 0$, angka 1 hingga N akan dicetak. Jika $N \leq 0$, perulangan tidak berjalan karena kondisi awal $i \leq N$ langsung salah, sehingga tidak ada output angka.

15. Output:

```
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>javac PrintXinterasi.java

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintXinterasi
Masukkan nilai x (int), akhiri dengan 999: 545
Masukkan nilai x (int), akhiri dengan 999: 45
Masukkan nilai x (int), akhiri dengan 999: 34
Masukkan nilai x (int), akhiri dengan 999: 67
Masukkan nilai x (int), akhiri dengan 999: 99
Masukkan nilai x (int), akhiri dengan 999: 143
Masukkan nilai x (int), akhiri dengan 999: 266
Masukkan nilai x (int), akhiri dengan 999: 432
Masukkan nilai x (int), akhiri dengan 999: 999
Hasil penjumlahan = 1631

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintXinterasi
Masukkan nilai x (int), akhiri dengan 999: 999
Kasus kosong
```

Penjelasan:

Program ini membaca sekumpulan bilangan bulat yang dimasukkan oleh pengguna, kemudian menjumlahkan semua nilai yang diberikan, hingga pengguna memasukkan angka 999 sebagai tanda berhenti. Jika input pertama adalah 999, program langsung mencetak "Kasus kosong" dan berhenti tanpa melakukan penjumlahan.

16. Output:

```
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>javac PrintXRepeat.java

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintXRepeat
Masukkan nilai x (int), akhiri dengan 999: 875
Masukkan nilai x (int), akhiri dengan 999: 353
Masukkan nilai x (int), akhiri dengan 999: 2234
Masukkan nilai x (int), akhiri dengan 999: 3434
Masukkan nilai x (int), akhiri dengan 999: 443
Masukkan nilai x (int), akhiri dengan 999: 999
Hasil penjumlahan = 7339

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintXRepeat
Masukkan nilai x (int), akhiri dengan 999: 999
Kasus kosong
```

Penjelasan:

Sama seperti sebelumnya, program ini akan melakukan penjumlahan angka bilangan bulat yang diinputkan hingga diinputkan angka 999 baru programnya akan berhenti dan menampilkan hasil penjumlahannya. Sedangkan, ketika diinputkan angka 999 sebagai inputan pertama maka program akan menampilkan "Kasus kosong".

17. Output:

```
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>javac PrintXWhile.java

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintXWhile
Masukkan nilai x (int), akhiri dengan 999: 976
Masukkan nilai x (int), akhiri dengan 999: 998
Masukkan nilai x (int), akhiri dengan 999: 78
Masukkan nilai x (int), akhiri dengan 999: 999
Hasil penjumlahan = 2052

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintXWhile
Masukkan nilai x (int), akhiri dengan 999: 999
Hasil penjumlahan = 0
```

Penjelasan:

Hampir sama dengan 2 program diatas yang melakukan penjumlahan bilangan bulat sampai angka yang diinputkan merupakan angka 999 maka program akan berakhir dan menampilkan jumlah. Bedanya, pada program ini ketika angka yang kita inputkan perta adalah 999 maka hasil penjumlahan yang akan tampil adalah 0.

18. Output:

```
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>javac SubProgram.java

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java SubProgram
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dengan ENTER: 17
26
Ke dua bilangan sebelum tukar: a = 17 b = 26
Maksimum = 26
Menukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 26 b = 17
```

Penjelasan:

Program ini membaca dua bilangan dari pengguna, menentukan bilangan yang lebih besar menggunakan fungsi `maxab(int a, int b)`, lalu mencoba menukar kedua bilangan menggunakan prosedur `tukar(int a, int b)`. Namun, karena Java menggunakan `pass-by-value` untuk tipe primitif, perubahan dalam prosedur `tukar()` tidak berpengaruh pada nilai asli di `main()`. Akibatnya, nilai bilangan tetap sama setelah pemanggilan prosedur.

19. Output:

```
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>javac Tempair.java

E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = -17
Wujud air beku
-17
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 17
Wujud air cair
17
E:\Shavdata semester 4\Pemrograman Berorientasi Objek\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 177
Wujud air uap/gas
177
```

Penjelasan:

Program ini menentukan wujud air berdasarkan suhu yang diinputkan. Jika suhu kurang dari 0°C maka masuk dalam kategori air beku, jika antara $0-100^{\circ}\text{C}$ maka masuk dalam kategori air cair dan jika lebih dari 100°C maka masuk dalam kategori air uap/gas. Program menggunakan `if-else if` untuk mengevaluasi kondisi dan mencetak hasilnya.