

**LAPORAN
PEMROGRAMAN BERBASIS OBJEK**



Nama : Nazwa Jihan Masika
Stambuk : 13020230267
Kelas : TI_PBO-14 (B4)
Dosen : Mardiyah Hasnawi, S.Kom., M.T., MTA

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

Tugas 2 Pemrograman Berbasis Objek

1. Output:

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

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

Penjelasan :

Program BacaString memanfaatkan library `BufferedReader` untuk menerima input dari pengguna. Library `IOException` digunakan untuk menangani kesalahan input/output yang mungkin terjadi, sementara `InputStreamReader` berfungsi mengonversi data masukan ke format yang dapat diproses oleh program.

Dalam program ini, variabel `str` dengan tipe `String` digunakan untuk menyimpan input. Objek `BufferedReader` bernama `datAIn` dibuat untuk membaca masukan dari konsol. Setelah menampilkan pesan "Baca string dan numbers:", program meminta pengguna memasukkan sebuah string.

Misalnya, jika saya memasukkan string "Nazwa", program akan membacanya menggunakan `datAIn.readLine()` dan menyimpannya dalam variabel `str`. Terakhir, program mencetak kembali string yang telah dimasukkan sebelumnya, yaitu "Nazwa".

2. Output:

```
D:\Nazwa\Tugas PBO\Tugas 2>javac ForEver.java
D:\Nazwa\Tugas PBO\Tugas 2>java ForEver
Program akan looping, akhiri dengan ^c
Print satu baris ...
```

Penjelasan:

Program `ForEver` menggunakan perulangan `while(true)`, yang biasanya dirancang untuk berjalan tanpa henti hingga secara eksplisit dihentikan. Saat dijalankan, program menampilkan pesan "Program akan looping, akhiri dengan ^c", memberi tahu pengguna bahwa program ini seharusnya berjalan terus-menerus hingga dihentikan secara manual.

Namun, di dalam perulangan `while(true)`, terdapat perintah `System.out.print("Print satu baris ...\n");` yang mencetak teks ke layar, diikuti oleh perintah `break;` yang segera menghentikan perulangan. Akibatnya, meskipun sekilas tampak bahwa program akan berjalan selamanya, sebenarnya program hanya mencetak satu baris teks dan kemudian berhenti.

Penggunaan perintah `break` dalam perulangan memungkinkan program keluar dari perulangan sebelum kondisi akhirnya tercapai. Dalam konteks ini, meskipun kondisi perulangan `while(true)` selalu benar, perintah `break` menyebabkan perulangan berhenti setelah satu iterasi.

3. Output:

```

D:\Nazwa\Tugas PBO\Tugas 2>javac If1.java

D:\Nazwa\Tugas PBO\Tugas 2>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : -1

D:\Nazwa\Tugas PBO\Tugas 2>javac If1.java

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

Nilai a positif 1

```

Penjelasan:

Program If1 dalam Java menggunakan `if` untuk mengecek apakah bilangan yang dimasukkan pengguna positif atau tidak. Jika bilangan ≥ 0 , program mencetak "Nilai a positif:" diikuti angka tersebut. Namun, untuk bilangan negatif, tidak ada output karena tidak ada blok `else`. Misalnya, input 1 menghasilkan "Nilai a positif: 1", sedangkan -1 tidak mencetak apa pun. Untuk menangani bilangan negatif, bisa ditambahkan blok `else` agar program menampilkan pesan sesuai dengan kondisi angka yang dimasukkan.

4. Output:

```

D:\Nazwa\Tugas PBO\Tugas 2>javac If2.java

D:\Nazwa\Tugas PBO\Tugas 2>java If2
Contoh IF dua kasus

Ketikkan suatu nilai integer : 67
Nilai a positif 67

D:\Nazwa\Tugas PBO\Tugas 2>javac If2.java

D:\Nazwa\Tugas PBO\Tugas 2>java If2
Contoh IF dua kasus

Ketikkan suatu nilai integer : -67
Nilai a negatif -67

```

Penjelasan:

Program If2 menggunakan percabangan if-else dalam Java untuk mengevaluasi apakah suatu bilangan bersifat positif atau negatif. Program ini memanfaatkan Scanner untuk menerima input dari pengguna setelah menampilkan pesan "Contoh IF dua kasus".

Bilangan yang dimasukkan disimpan dalam variabel a dan diperiksa menggunakan if ($a \geq 0$). Jika nilai a nol atau lebih, program akan menampilkan "Nilai a positif: 67". Sebaliknya, jika a bernilai negatif, blok else akan mencetak "Nilai a negatif: -67".

5. Output:

```

D:\Nazwa\Tugas PBO\Tugas 2>javac If3.java

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

D:\Nazwa\Tugas PBO\Tugas 2>javac If3.java

D:\Nazwa\Tugas PBO\Tugas 2>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer : -12
Nilai a negatif -12

D:\Nazwa\Tugas PBO\Tugas 2>javac If3.java

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

```

Penjelasan:

Program If3 menerapkan percabangan if-else if-else dalam Java untuk mengidentifikasi apakah angka yang dimasukkan bernilai positif, nol, atau negatif. Program ini menggunakan Scanner untuk menerima input dari pengguna setelah menampilkan pesan "Contoh IF tiga kasus".

Nilai yang dimasukkan disimpan dalam variabel a dan dievaluasi berdasarkan tiga kondisi utama. Jika angka lebih dari nol, program akan mencetak "Nilai a: 12". Jika angka sama dengan nol, program menampilkan "Nilai a: 0". Sedangkan jika angka kurang dari nol, program mencetak "Nilai a: -12".

Sebagai contoh, jika pengguna memasukkan 12, maka output yang ditampilkan adalah "Nilai a: 12". Jika pengguna memasukkan -12, program akan mencetak "Nilai a: -12". Sementara itu, jika pengguna memasukkan 0, program menampilkan "Nilai a: 0".

6. Output:

```
D:\Nazwa\Tugas PBO\Tugas 2>javac KasusBoolean.java
D:\Nazwa\Tugas PBO\Tugas 2>java KasusBoolean
true
benar
```

Penjelasan :

Program KasusBoolean menggunakan tipe data Boolean dalam struktur if-else di Java. Pada awal program, variabel bool dengan tipe Boolean dideklarasikan dan diberi nilai true.

Program kemudian melakukan dua kali pengecekan kondisi. Pada evaluasi pertama, jika if(bool) dieksekusi, maka program akan mencetak "true". Namun, jika bool bernilai false, program masuk ke blok else dan mencetak "false".

Pada evaluasi kedua, program menerapkan operator negasi (!) dalam kondisi if(!bool) untuk membalikkan nilai bool. Jika bool awalnya true, maka !bool menjadi false, sehingga program melewati blok if dan menjalankan else, yang mencetak "benar". Sebaliknya, jika bool bernilai false, program akan mencetak "salah".

7. Output:

```
D:\Nazwa\Tugas PBO\Tugas 2>javac KasusSwitch.java
D:\Nazwa\Tugas PBO\Tugas 2>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
e
Yang anda ketik adalah e
D:\Nazwa\Tugas PBO\Tugas 2>v
'v' is not recognized as an internal or external command,
operable program or batch file.
D:\Nazwa\Tugas PBO\Tugas 2>javac KasusSwitch.java
D:\Nazwa\Tugas PBO\Tugas 2>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
v
Yang anda ketik adalah huruf mati
```

Penjelasan:

Program KasusSwitch menggunakan struktur switch-case dalam Java untuk mengenali karakter yang dimasukkan oleh pengguna. Variabel cc bertipe char digunakan untuk menyimpan input, sementara Scanner berfungsi membaca data dari pengguna.

Setelah menampilkan pesan "Silakan masukkan sebuah huruf, kemudian tekan RETURN", program mengambil karakter pertama dari input menggunakan `masukan.next().charAt(0)`. Selanjutnya, switch-case mengevaluasi apakah karakter tersebut termasuk huruf vokal kecil ('a', 'e', 'i', 'o', atau 'u'). Jika sesuai, program menampilkan pesan yang sesuai dengan huruf tersebut.

Sebagai contoh, jika pengguna memasukkan 'e', program akan mencetak pesan yang sesuai untuk huruf vokal. Namun, jika pengguna memasukkan 'v', yang bukan huruf vokal, program tidak akan mencocokkannya dengan kasus yang ada dalam switch, sehingga tidak menampilkan pesan untuk huruf vokal.

8. Output:

```
D:\Nazwa\Tugas PBO\Tugas 2>javac Konstant.java
D:\Nazwa\Tugas PBO\Tugas 2>java Konstant
Jari-jari lingkaran = 35
Luas lingkaran = 3848.3374
Akhir program
```

Penjelasan:

Program ini menghitung luas lingkaran berdasarkan jari-jari yang dimasukkan oleh pengguna. Konstanta PHI ditetapkan sebagai 3.1415f menggunakan final, sehingga nilainya tidak berubah selama program berjalan. Variabel r dengan tipe float digunakan untuk menyimpan jari-jari yang diberikan.

Program menampilkan pesan "Jari-jari lingkaran =" untuk meminta input dari pengguna. Jika pengguna memasukkan 35, nilai tersebut dibaca menggunakan `masukan.nextFloat()` dan disimpan dalam r. Selanjutnya, program menghitung luas lingkaran menggunakan rumus $\text{PHI} \times r \times r$ dan menampilkan hasilnya, yaitu 3848.4775. Sebagai penutup, program mencetak pesan "Akhir program", menandakan bahwa eksekusi telah selesai.

9. Output:

```
D:\Nazwa\Tugas PBO\Tugas 2>javac Max2.java
D:\Nazwa\Tugas PBO\Tugas 2>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dengan RETURN :
1 6
Ke dua bilangan : a = 1 b = 6
Nilai b yang maksimum: 6

D:\Nazwa\Tugas PBO\Tugas 2>javac Max2.java
D:\Nazwa\Tugas PBO\Tugas 2>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dengan RETURN :
7 2
Ke dua bilangan : a = 7 b = 2
Nilai a yang maksimum: 7
```

Penjelasan :

Program Max2 menentukan bilangan terbesar dari dua angka yang dimasukkan pengguna. Variabel a dan b bertipe integer digunakan untuk menyimpan input, sementara Scanner membaca data dari pengguna. Setelah menampilkan pesan "Maksimum dua bilangan", program meminta dua angka, membacanya dengan `masukan.nextInt()`, lalu membandingkannya menggunakan if-else. Jika a lebih besar atau sama dengan b, program mencetak "Nilai maksimum adalah: a", sedangkan jika b lebih besar, program

menampilkan "Nilai maksimum adalah: b". Contohnya, jika $a = 1$ dan $b = 6$, program mencetak "Nilai maksimum adalah: 6", sedangkan jika $a = 7$ dan $b = 2$, outputnya "Nilai maksimum adalah: 7".

10. Output:

```
D:\Nazwa\Tugas PBO\Tugas 2>javac PriFor.java

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

Penjelasan:

Program PriFor mencetak bilangan dari 1 hingga N berdasarkan input pengguna. Variabel i digunakan sebagai penghitung dalam perulangan for, sedangkan N menyimpan batas angka yang dimasukkan melalui Scanner. Setelah menampilkan pesan "Baca N, print 1 s/d N", program membaca nilai N dengan `masukan.nextInt()`. Perulangan for dimulai dari $i = 1$ hingga $i \leq N$, mencetak setiap angka menggunakan `System.out.println(i)`. Jika pengguna memasukkan $N = 9$, program mencetak angka 1 hingga 9 secara berurutan. Setelah selesai, ditampilkan pesan "Akhir program" sebagai penutup.

11. Output:

```
D:\Nazwa\Tugas PBO\Tugas 2>javac PrintIterasi.java

D:\Nazwa\Tugas PBO\Tugas 2>java PrintIterasi
Nilai N > 0 = 3
Print i dengan ITERATE :
1
2
3
```

Penjelasan:

Program PrintIterasi mencetak bilangan dari 1 hingga N berdasarkan input pengguna. Variabel i digunakan sebagai penghitung, sementara N menentukan batas angka yang dimasukkan melalui Scanner. Setelah menampilkan pesan "Nilai N > 0 =", program membaca N dengan `masukan.nextInt()`. Perulangan for tanpa kondisi eksplisit dijalankan, mencetak nilai i menggunakan `System.out.println(i)`. Jika i mencapai N, perulangan dihentikan dengan `break`, jika belum, i bertambah satu. Jika pengguna memasukkan $N = 3$, program mencetak 1, 2, 3, lalu berhenti.

12. Output:

```
D:\Nazwa\Tugas PBO\Tugas 2>javac PrintRepeat.java

D:\Nazwa\Tugas PBO\Tugas 2>java PrintRepeat
Nilai N > 0 = 5
Print i dengan REPEAT:
1
2
3
4
5
```

Penjelasan:

Program PrintRepeat mencetak bilangan dari 1 hingga N menggunakan perulangan do-while, berdasarkan input pengguna. Variabel i digunakan sebagai penghitung, sedangkan N menyimpan batas angka yang dimasukkan melalui Scanner. Setelah menampilkan pesan "Nilai N > 0 =", program membaca N dengan masukan.nextInt(), lalu menginisialisasi i = 1. Perulangan do-while memastikan angka dicetak minimal sekali sebelum memeriksa kondisi $i \leq N$. Setiap iterasi mencetak i dengan System.out.print(i + "\n"), lalu menaikkan i. Jika N = 5, program mencetak 1, 2, 3, 4, 5, lalu berhenti.

13. Output:

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

D:\Nazwa\Tugas PBO\Tugas 2>java PrintWhile
Nilai N > 0 = 8
Print i dengan WHILE:
1
2
3
4
5
6
7
8
```

Penjelasan:

Program PrintWhile mencetak bilangan dari 1 hingga N menggunakan perulangan while, berdasarkan input pengguna. Variabel i digunakan sebagai penghitung, sementara N menentukan batas angka yang dibaca melalui Scanner. Setelah menampilkan "Nilai N > 0 =", program membaca N dengan masukan.nextInt(), lalu menginisialisasi i = 1. Perulangan while berjalan selama $i \leq N$, mencetak i dengan System.out.println(i), lalu menaikkan i. Jika N = 8, program mencetak 1 hingga 8, lalu berhenti.

14. Output:

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

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

Penjelasan:

Program PrintWhile1 mencetak bilangan dari 1 hingga N menggunakan perulangan while dengan sintaks lebih ringkas. Variabel i diinisialisasi dengan 1, sementara N ditentukan melalui Scanner. Setelah menampilkan "Nilai N > 0 =", program membaca N dengan masukan.nextInt(). Perulangan while berjalan selama $i \leq N$, mencetak i dengan System.out.println(i++), yang sekaligus menaikkan nilainya. Jika N = 5, program mencetak 1 hingga 5, lalu berhenti.

15. Output:

```

D:\Nazwa\Tugas PBO\Tugas 2>javac PrintXinterasi.java

D:\Nazwa\Tugas PBO\Tugas 2>java PrintXinterasi
Masukan nilai x (int), akhiri dg 999: 1
Masukan nilai x (int), akhiri dg 999: 5
Masukan nilai x (int), akhiri dg 999: 9
Masukan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 15

D:\Nazwa\Tugas PBO\Tugas 2>javac PrintXinterasi.java

D:\Nazwa\Tugas PBO\Tugas 2>java PrintXinterasi
Masukan nilai x (int), akhiri dg 999: 999
Kasus kosong
Hasil penjumlahan = 0

```

Penjelasan:

Program PrintXinterasi menjumlahkan angka yang dimasukkan pengguna hingga 999 sebagai penanda berhenti. Variabel x menyimpan input, sementara Sum menampung hasil penjumlahan. Program menampilkan "Masukkan nilai x (int), akhiri dengan 999:", lalu membaca angka pertama. Jika 999 dimasukkan langsung, program mencetak "Kasus kosong" dan "Hasil penjumlahan = 0". Jika tidak, angka dijumlahkan dalam perulangan for(;;) hingga 999 dimasukkan. Jika pengguna memasukkan 1, 5, 9, 999, program mencetak "Hasil penjumlahan = 15".

16. Output:

```

D:\Nazwa\Tugas PBO\Tugas 2>javac PrintXRepeat.java

D:\Nazwa\Tugas PBO\Tugas 2>java PrintXRepeat
Masukan nilai x (int), akhiri dg 999: 2
Masukan nilai x (int), akhiri dg 999: 4
Masukan nilai x (int), akhiri dg 999: 6
Masukan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 12

D:\Nazwa\Tugas PBO\Tugas 2>javac PrintXRepeat.java

D:\Nazwa\Tugas PBO\Tugas 2>java PrintXRepeat
Masukan nilai x (int), akhiri dg 999: 999
Kasus kosong

```

Penjelasan :

Program PrintXRepeat menjumlahkan angka yang dimasukkan pengguna hingga 999 sebagai tanda berhenti. Variabel x menyimpan input, sementara Sum menampung hasil penjumlahan. Program menampilkan "Masukkan nilai x (int), akhiri dengan 999:" dan membaca angka pertama. Jika 999 dimasukkan langsung, program mencetak "Kasus kosong". Jika tidak, Sum diinisialisasi ke 0, lalu perulangan do-while menjumlahkan angka hingga 999 dimasukkan. Setelah itu, program mencetak "Hasil penjumlahan = Sum". Jika pengguna memasukkan 2, 4, 6, 999, program menampilkan "Hasil penjumlahan = 12".

17. Output:

```

D:\Nazwa\Tugas PBO\Tugas 2>javac PrintXWhile.java

D:\Nazwa\Tugas PBO\Tugas 2>java PrintXWhile
Masukan nilai x (int), akhiri dg 999: 9
Masukan nilai x (int), akhiri dg 999: 8
Masukan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 17

```

Penjelasan:

Program PrintXWhile menjumlahkan angka yang dimasukkan pengguna hingga 999 sebagai tanda berhenti. Variabel x menyimpan input, sementara Sum menampung hasil penjumlahan. Jika angka pertama 999, program mencetak "Hasil penjumlahan = 0" dan berhenti. Jika tidak,

angka dijumlahkan dalam perulangan while ($x \neq 999$) hingga 999 dimasukkan. Jika pengguna memasukkan 9, 8, 999, program mencetak "Hasil penjumlahan = 17".

18. Output:

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

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

Penjelasan:

Program SubProgram menentukan bilangan terbesar dari dua angka yang dimasukkan pengguna dan mencoba menukarnya. Variabel a dan b menyimpan input, lalu program mencetak nilai awal dan menentukan nilai maksimum. Jika $a = 5$ dan $b = 4$, program menampilkan "Maksimum = 5". Kemudian, fungsi tukar(a, b) dipanggil, tetapi karena Java menggunakan passing by value, perubahan hanya terjadi dalam fungsi tanpa memengaruhi nilai asli, sehingga a tetap 5 dan b tetap 4 setelah eksekusi.

19. Output:

```
D:\Nazwa\Tugas PBO\Tugas 2>javac Tempair.java

D:\Nazwa\Tugas PBO\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 7
Wujud air cair
7
D:\Nazwa\Tugas PBO\Tugas 2>javac Tempair.java

D:\Nazwa\Tugas PBO\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 120
Wujud air uap/gas
120
D:\Nazwa\Tugas PBO\Tugas 2>javac Tempair.java

D:\Nazwa\Tugas PBO\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = -2
Wujud air beku
-2
```

Penjelasan :

ka $0 \leq T \leq 100$, outputnya "Wujud Air Cair". Jika $T > 100$, program menampilkan "Wujud Air Uap/Gas".

Contoh: untuk $T = 7$, outputnya "Wujud Air Cair"; untuk $T = 120$, hasilnya "Wujud Air Uap/Gas"; dan untuk $T = -2$, program mencetak "Wujud Air Beku".