

**LAPORAN**  
**PEMROGRAMAN BERORIENTASI OBJEK**



Nama : Putri Habiba Syari  
Stambuk : 13020230333  
Kelas : B1  
Dosen : Mardiyah Hasnawi, S.Kom., M.T.

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

## 1. BacaString.java

Program ini membaca input string dari pengguna menggunakan **BufferedReader** dan menampilkannya kembali. **BufferedReader** digunakan untuk membaca input baris demi baris, sementara throws **IOException** diperlukan untuk menangani kemungkinan kesalahan saat membaca data. Import `javax.swing.*`; tetap ada, meskipun tidak digunakan dalam kode ini, mungkin untuk pengembangan lebih lanjut yang melibatkan GUI. Saat dijalankan, program meminta pengguna memasukkan teks, menyimpannya dalam variabel `str`, lalu menampilkannya di layar sesuai dengan input yang diberikan.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>javac BacaString.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java BacaString

Baca string dan Integer:
Masukkan sebuah string: Indonesia
String yang dibaca: Indonesia
```

```

D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>javac If1.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer: 13

Nilai a positif 13
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer: -13

D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>

```

#### 4. If2.java

Program ini membaca sebuah bilangan integer dari pengguna dan mengecek apakah nilainya positif atau negatif. Jika angka yang dimasukkan lebih besar atau sama dengan nol, program akan mencetak bahwa angka tersebut positif. Sebaliknya, jika angka kurang dari nol, program akan mencetak bahwa angka tersebut negatif. Program ini menggunakan **Scanner** untuk menerima input dan **struktur percabangan IF-ELSE** untuk menangani kedua kemungkinan nilai.

Output:

```

D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>javac If2.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java If2
Contoh IF dua kasus
Ketikkan suatu nilai integer: 29
Nilai a positif 29

D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java If2
Contoh IF dua kasus
Ketikkan suatu nilai integer: -29
Nilai a negatif -29

```

#### 5. If3.java

Program ini membaca sebuah bilangan integer dari pengguna dan mengecek apakah nilainya positif, nol, atau negatif. Jika angka yang dimasukkan lebih besar dari nol, program mencetak bahwa angka tersebut positif. Jika angka sama dengan nol, program mencetak bahwa nilai tersebut adalah nol. Sebaliknya, jika angka kurang dari nol, program mencetak bahwa angka tersebut negatif. Program ini menggunakan **Scanner** untuk menerima input dan **struktur percabangan IF-ELSE IF-ELSE** untuk menangani tiga kemungkinan nilai.

Output:

```

D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>javac If3.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer: 11
Nilai a positif 11

D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer: 0
Nilai a Nol 0

D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer: -11
Nilai a negatif -11

```

#### 6. KasusBoolean.java

Program ini mendemonstrasikan penggunaan **ekspresi kondisional dengan boolean**. Variabel bool diinisialisasi dengan nilai true. Struktur percabangan **IF-ELSE** pertama akan mencetak "**true**" jika bool bernilai **true**, jika tidak maka mencetak "**false**". Kemudian, percabangan **IF-ELSE** kedua menggunakan negasi (!bool), sehingga jika bool bernilai **false**, akan mencetak "**salah**", dan jika **true**, akan mencetak "**benar**".

Output:

```

D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>javac KasusBoolean.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java KasusBoolean
true
benar

```

#### 7. KasusSwitch.java

Program ini membaca sebuah karakter dari pengguna dan mengecek apakah karakter tersebut merupakan huruf vokal (a, u, e, i, o). Jika karakter yang dimasukkan adalah salah satu dari huruf vokal, program akan mencetak pesan yang sesuai. Jika karakter yang dimasukkan bukan huruf vokal, program akan mencetak bahwa yang diketik adalah **huruf mati**. Program ini menggunakan **struktur percabangan switch-case**, yang lebih efisien daripada **if-else** dalam kasus seperti ini.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>javac KasusSwitch.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN:
a
Yang anda ketik adalah a
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN:
u
Yang anda ketik adalah u
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN:
e
Yang anda ketik adalah e
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN:
i
Yang anda ketik adalah i
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN:
o
Yang anda ketik adalah o
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN:
p
Yang anda ketik adalah huruf mati
```

#### 8. Konstant.java

Program ini membaca jari-jari lingkaran dari pengguna dan menghitung luas lingkaran menggunakan rumus  $\pi \times r^2$ . Konstanta **PHI** dideklarasikan dengan nilai **3.1415f** untuk mewakili nilai  $\pi$ . Input diambil menggunakan **Scanner**, lalu program menghitung luas dan menampilkannya di layar.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>javac Konstant.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java Konstant
Jari-jari lingkaran = 18
Luas lingkaran = 1017.846
Akhir program
```

#### 9. Max2.java

Program ini membaca dua bilangan integer dari pengguna dan menentukan bilangan yang lebih besar. Input diambil menggunakan **Scanner**, lalu program membandingkan kedua bilangan dengan **if-else**. Jika a lebih besar atau sama dengan b, program mencetak bahwa a adalah nilai maksimum. Jika tidak, b yang dianggap sebagai nilai maksimum.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>javac Max2.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java Max2
Menentukan nilai maksimum dari dua bilangan
Masukkan dua bilangan, pisahkan dengan ENTER:
13
07
Dua bilangan yang dimasukkan: a = 13, b = 7
Bilangan maksimum adalah: 13
```

#### 10. PriFor.java

Program ini meminta pengguna memasukkan nilai **N**, lalu mencetak angka dari **1 hingga N** menggunakan perulangan **FOR**. Variabel **i** digunakan sebagai penghitung yang dimulai dari 1 dan bertambah hingga mencapai **N**. Setelah semua angka dicetak, program menampilkan pesan **"Akhir program"**

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>javac PriFor.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java PriFor
Baca N, print 1 s/d N
N = 3
1
2
3
Akhir program
```

#### 11. PrintIterasi.java

Program ini membaca sebuah bilangan **N** dari pengguna, kemudian mencetak angka dari **1 hingga N** menggunakan perulangan **for (;)**, yang merupakan bentuk **infinite loop**. Perulangan hanya berhenti ketika **i** mencapai **N**, di mana **break** akan menghentikan proses. Setiap iterasi, nilai **i** akan bertambah **1** hingga mencapai batas yang ditentukan.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>javac PrintIterasi.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintIterasi
Nilai N > 0 = 7
Print i dengan ITERATE :
1
2
3
4
5
6
7
```

#### 12. PrintRepeat.java

Program ini meminta pengguna memasukkan **N**, lalu mencetak angka dari **1 hingga N** menggunakan perulangan **do-while**. Perulangan **do-while** memastikan bahwa kode di dalamnya akan dieksekusi minimal sekali sebelum memeriksa kondisi **i <= N**. Nilai **i** dimulai dari **1**, lalu bertambah **1** setiap iterasi hingga mencapai **N**.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintRepeat
Nilai N > 0 = 4
Print i dengan REPEAT:
1
2
3
4
```

#### 13. PrintWhile.java

Program ini meminta pengguna memasukkan **N**, kemudian mencetak angka dari **1 hingga N** menggunakan perulangan **while**. Perulangan ini berjalan selama kondisi **i <= N** terpenuhi. Nilai **i** dimulai dari **1** dan akan terus bertambah **1** setiap iterasi hingga melebihi **N**, sehingga program berhenti.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>javac PrintWhile.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintWhile
Nilai N > 0 = 6
Print i dengan WHILE:
1
2
3
4
5
6
```

#### 14. PrintWhile1.java

Program ini membaca input **N** dari pengguna, lalu mencetak angka dari **1 hingga N** menggunakan perulangan **while**. Variabel **i** diinisialisasi dengan **1**, dan dalam setiap iterasi, nilai **i** langsung ditampilkan sebelum ditingkatkan (**i++**). Perulangan berhenti ketika **i** melebihi **N**.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>javac PrintWhile1.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintWhile1
Nilai N > 0 = 10
Print i dengan WHILE (ringkas):
1
2
3
4
5
6
7
8
9
10
```

#### 15. PrintXIterasi.java

Program ini membaca serangkaian bilangan integer dari pengguna dan menjumlahkannya hingga pengguna memasukkan **999** sebagai tanda berhenti. Jika angka pertama yang dimasukkan adalah **999**, program akan menampilkan pesan "**Kasus kosong**", menandakan bahwa tidak ada angka yang dijumlahkan. Jika tidak, angka pertama langsung menjadi nilai awal penjumlahan. Kemudian, dalam perulangan **for(;;)** (infinite loop), program terus meminta input hingga angka **999** dimasukkan, lalu mencetak hasil penjumlahan.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>javac PrintXIterasi.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintXIterasi
Masukkan nilai x (int), akhiri dengan 999: 11
Masukkan nilai x (int), akhiri dengan 999: 13
Masukkan nilai x (int), akhiri dengan 999: 29
Masukkan nilai x (int), akhiri dengan 999: 18
Masukkan nilai x (int), akhiri dengan 999: 3
Masukkan nilai x (int), akhiri dengan 999: 999
Hasil penjumlahan = 74

D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintXIterasi
Masukkan nilai x (int), akhiri dengan 999: 999
Kasus kosong
Hasil penjumlahan = 0

D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintXIterasi
Masukkan nilai x (int), akhiri dengan 999: 1
Masukkan nilai x (int), akhiri dengan 999: 999
Hasil penjumlahan = 1
```

#### 16. PrintXRepeat.java

Program ini meminta pengguna memasukkan nilai integer dan menjumlahkan semua nilai yang dimasukkan hingga pengguna mengetik **999** sebagai tanda berhenti. Jika nilai pertama yang dimasukkan adalah **999**, program akan mencetak "**Kasus kosong**" dan langsung berhenti. Jika tidak, program mulai menjumlahkan nilai dengan **sum = 0** dan menggunakan perulangan **do-while** untuk terus meminta input baru serta menambahkan setiap nilai ke dalam **sum**. Perulangan ini akan berhenti saat pengguna memasukkan **999**, lalu program mencetak hasil penjumlahan dari semua angka yang telah dimasukkan.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>javac PrintXRepeat.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintXRepeat
Masukkan nilai x (int), akhiri dengan 999: 999
Kasus kosong

D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintXRepeat
Masukkan nilai x (int), akhiri dengan 999: 22
Masukkan nilai x (int), akhiri dengan 999: 2003
Masukkan nilai x (int), akhiri dengan 999: 999
Hasil penjumlahan = 2025
```

#### 17. PrintXWhile.java

Program ini membaca beberapa nilai integer dari pengguna dan menjumlahkannya hingga pengguna memasukkan **999** sebagai tanda berhenti. Variabel **Sum** diinisialisasi dengan **0** untuk menyimpan total penjumlahan. Program meminta input pertama sebelum memasuki perulangan **while**. Selama nilai yang dimasukkan bukan **999**, nilai tersebut ditambahkan ke dalam **Sum**, dan program kembali meminta input baru. Ketika pengguna memasukkan **999**, perulangan berhenti dan hasil penjumlahan ditampilkan di layar.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>javac PrintXWhile.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java PrintXWhile
Masukkan nilai x (int), akhiri dg 999: 11
Masukkan nilai x (int), akhiri dg 999: 13
Masukkan nilai x (int), akhiri dg 999: 29
Masukkan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 53
```

#### 18. SubProgram.java

Program ini menerima dua bilangan integer, menampilkan bilangan terbesar dengan fungsi `maxab()`, dan mencoba menukar kedua bilangan dengan prosedur `tukar()`. Namun, karena Java menggunakan `pass-by-value` untuk tipe primitif, `tukar()` hanya menukar salinan `a` dan `b`, bukan nilai aslinya di `main()`.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>javac SubProgram.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java SubProgram
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN:
13
29
Ke dua bilangan: a = 13 b = 29
Maksimum = 29
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 29 b = 13
```

#### 19. Tempair.java

Program ini membaca suhu dalam derajat Celsius dari pengguna, lalu menentukan wujud air menggunakan struktur percabangan `if-else if`. Jika suhu kurang dari  $0^{\circ}\text{C}$ , air dalam keadaan beku; jika antara  $0^{\circ}\text{C}$  hingga  $100^{\circ}\text{C}$ , air dalam keadaan cair; dan jika lebih dari  $100^{\circ}\text{C}$ , air dalam keadaan uap atau gas.

Output:

```
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>javac Tempair.java
D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = -11
Wujud air beku
-11

D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 13
Wujud air cair
13

D:\Putt\smstr 4\Pemrograman Berorientasi Objek\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 2918
Wujud air uap/gas
2918
```