

**PRAKTIKUM
PEMROGRAMAN BERBASIS OBJEK**



Nama : Fanisa Violena
Stambuk : 13020230120
Frekuensi : TI_PBO-12 (B2)
Dosen : Mardiyah Hasnawi, S.Kom.,M.T.,MTA

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

1. BacaString

```
run:
Baca string dan Integer:
masukkan sebuah string: Fanisa Violena
String yang dibaca : Fanisa Violena
BUILD SUCCESSFUL (total time: 26 seconds)
```

➤ Penjelasan:

- **Mengimpor Library yang dibutuhkan**

java.io. untuk menangani operasi input-output.

BufferedReader bersama dengan InputStreamReader(System.in) untuk membaca input dari keyboard. yang memungkinkan pengguna untuk memasukkan teks yang akan disimpan dalam variabel str.

- **Penggunaan**

Program mendeklarasikan kelas BacaString dan memiliki fungsi main(), yang merupakan titik awal eksekusi program. Variabel str bertipe String dideklarasikan untuk menyimpan input yang dimasukkan oleh pengguna. Program akan menampilkan pesan untuk meminta pengguna memasukkan sebuah string, lalu membaca input yang diberikan. Setelah pengguna memasukkan teks dan menekan Enter, program menampilkan kembali teks yang telah dimasukkan dengan menambahkan pesan "String yang dibaca :" di depannya.

2. ForEver

```
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
BUILD STOPPED (total time: 3 seconds)
```

➤ Penjelasan

Buatlah Class dengan nama "ForEver", lalu fungsi main() adalah titik awal program. Untuk mencetak pesan awal yaitu "("Program akan looping, akhiri dengan ^c)" menggunakan **System.out.println**. dan gunakan **while(true) { System.out.print("Print satu baris\n");** untuk melakukan perulangan yang tak berhenti.

3. If1

```
run:
Contoh IF satu kasus
Ketikkan suatu nilai integer : 27

Nilai a positif 27
BUILD SUCCESSFUL (total time: 7 seconds)
```

➤ **Penjelasan:**

Import Library **java.util.Scanner**; yang berfungsi untuk membaca input dari pengguna melalui keyboard. Dan imulai dengan mendeklarasikan kelas If1 dan metode main() sebagai titik awal eksekusi. Lalu, Membuat objek Scanner untuk membaca input pengguna dan buat variable a dengan type integer untuk menyimpan nilai input. Kemudian, melakukan perkondisian IF untuk menentukan bilangan positif ($a \geq 0$) dan bilangan negative (< 0).

4. If2

```
run:
Contoh IF dua kasus
Ketikkan suatu nilai integer :90
Nilai a positif 90
BUILD SUCCESSFUL (total time: 15 seconds)
```

➤ **Penjelasan:**

Import Library **java.util.Scanner**; yang berfungsi untuk membaca input dari pengguna melalui keyboard. Dan imulai dengan mendeklarasikan kelas If2 dan metode main() sebagai titik awal eksekusi. Lalu, Membuat objek Scanner untuk membaca input pengguna dan buat variable a dengan type integer untuk menyimpan nilai input. Kemudian, melakukan perkondisian IF untuk menentukan bilangan positif ($a \geq 0$) dan else jika bilangan negative (< 0).

5. If3

```
run:
Contoh IF tiga kasus
Ketikkan suatu nilai integer :-1
Nilai a negatif -1
BUILD SUCCESSFUL (total time: 7 seconds)
|
```

➤ **Penjelasan:**

Import Library **java.util.Scanner**; yang berfungsi untuk membaca input dari pengguna melalui keyboard. Dan imulai dengan mendeklarasikan kelas If2 dan metode main() sebagai titik awal eksekusi. Lalu, Membuat objek Scanner untuk membaca input pengguna dan buat variable a dengan type integer untuk menyimpan nilai input. Kemudian, melakukan perkondisian IF untuk menentukan bilangan positif ($a > 0$), else if jika bilangan negative (< 0) dan else jika bilangan Nol ($= 0$).

6. KasusBoolean

```
run:
true
benar
BUILD SUCCESSFUL (total time: 0 seconds)
```

➤ **Penjelasan:**

Buatlah class dengan nama “KasusBoolean” Dimana Boolean hanya bernilai true dan false. Dan pada program ini Boolean melakukan dalam percabangan If-Else. Jika boolean bernilai **true**, maka blok kode dalam if akan dieksekusi. Jika boolean bernilai **false**, maka blok kode dalam else akan dieksekusi. Operator ! digunakan untuk membalik nilai boolean (true menjadi false, dan sebaliknya).

7. KasusSwitch

```
run:
Ketikkan sebuah huruf, akhiri dengan RETURN
f
Yang anda ketik adalah huruf mati
BUILD SUCCESSFUL (total time: 6 seconds)
|
```

➤ **Penjelasan:**

Import Library **java.util.Scanner**; yang berfungsi untuk membaca input dari pengguna melalui keyboard. Deklarasikan variable cc dengan type char. Disini menggunakan struktur Switch-case untuk membandingkan karakter input dengan beberapa kemungkinan (a, I, u, e, o). Jika input sesuai dengan salah satu case, maka program mencetak huruf tersebut. jika input tidak sesuai dengan case yang ada, program akan menjalankan bagian **default**, yang menyatakan bahwa huruf tersebut adalah konsonan (huruf mati).

8. Konstant

```
run:
Jari-jari lingkaran =12
Luas lingkaran = 452.37598
Akhir program
BUILD SUCCESSFUL (total time: 10 seconds)
```

➤ **Penjelasan:**

Import Library **java.util.Scanner**; yang berfungsi untuk membaca input dari pengguna melalui keyboard. Mendeklarasikan Phi dengan type data float berisi nilai 3.1415 f(tanda kalua type float). Menggunakan **Scanner** untuk menerima input dari pengguna dan **menampilkan hasil perhitungan** dengan metode System.out.print().

9. Max2

```
run:
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
56
23
Ke dua bilangan : a = 56 b = 23
Nilai a yang maksimum 56
BUILD SUCCESSFUL (total time: 8 seconds)
|
```

➤ Penjelasan:

Import Library **java.util.Scanner**; yang berfungsi untuk membaca input dari pengguna melalui keyboard. Mendeklarasikan nilai a dan b dengan type data integer. Dengan menggunakan if-else yang Dimana jika a lebih besar atau sama dengan b, maka a adalah bilangan maksimum. Jika b lebih besar dari a, maka b adalah bilangan maksimum. Program ini berguna untuk **menentukan bilangan terbesar dari dua angka** dengan logika sederhana dalam Java.

10. Prifor

```
run:
Baca N, print 1 s/d N
N = 9
1
2
3
4
5
6
```

➤ Penjelasan:

Import Library **java.util.Scanner**; yang berfungsi untuk membaca input dari pengguna melalui keyboard. Mendefinisikan class “Prifor” dengan mendeklarasikan variable i dan N dengan type data integer Menggunakan **perulangan for** untuk mencetak angka **1 hingga N**. for (i = 1; i <= N; i++) memastikan bahwa angka dicetak **berurutan dari 1 sampai N**.

11. PrintIterasi

```
run:
Nilai N >0 = 2
Print i dengan ITERATE :
1
2
BUILD SUCCESSFUL (total time: 8 seconds)
```

➤ **Penjelasan:**

Import Library **java.util.Scanner**; yang berfungsi untuk membaca input dari pengguna melalui keyboard. Mendefinisikan class “PrintIterasi” dengan mendeklarasikan variable i dan N dengan type data integer Program ini **membaca sebuah bilangan N** dari pengguna. Menggunakan **perulangan for tanpa kondisi eksplisit** dan **menggunakan break untuk berhenti** ketika i == N. Perulangan tetap berjalan **tanpa batas** sampai kondisi if (i == N) terpenuhi.

12. PrintRepeat

```
Nilai N >0 = 6
Print i dengan REPEAT:
1
2
3
4
5
6
BUILD SUCCESSFUL (total time: 2 seconds)
```

➤ **Penjelasan:**

Import Library **java.util.Scanner**; yang berfungsi untuk membaca input dari pengguna melalui keyboard. Mendefinisikan class “**PrintRepeat**” dengan mendeklarasikan variable i dan N dengan type data integer. Program ini **membaca sebuah bilangan N** dari pengguna. Menggunakan **perulangan do-while** yang memastikan bahwa kode **dieksekusi minimal satu kali**, meskipun N = 0. Perulangan akan berhenti ketika i > N.

13. PrintWhile

```
run:
Nilai N >0 = 3
Print i dengan WHILE:
1
2
3
BUILD SUCCESSFUL (total time: 3 minutes 4 seconds)
```

➤ **Penjelasan:**

Import Library **java.util.Scanner**; yang berfungsi untuk membaca input dari pengguna melalui keyboard. Mendefinisikan class “**PrintWhile**” dengan mendeklarasikan variable i dan N dengan type data integer. Program ini **membaca sebuah bilangan N** dari pengguna. Menggunakan **perulangan while** yang terus berjalan selama kondisi **i <= N** masih terpenuhi. Perulangan akan berhenti ketika nilai i sudah lebih besar dari N.

14. PrintWhile1

```
run:
Nilai N >0 = 4
Print i dengan WHILE (ringkas):
1
2
3
4
BUILD SUCCESSFUL (total time: 8 seconds)
```

➤ Penjelasan:

Import Library **java.util.Scanner**; yang berfungsi untuk membaca input dari pengguna melalui keyboard. Mendefinisikan class “**PrintWhile1**” dengan mendeklarasikan variable i dan N dengan type data integer. Program ini membaca N, lalu mencetak angka **1 hingga N** menggunakan **perulangan while yang lebih ringkas**. **Ekspresi i++ langsung ditulis dalam println()**, sehingga kode lebih pendek dan efisien.

15. PrintXinterasi

```
run:
Masukkan nilai x (int), akhiri dg 999: 7
Masukkan nilai x (int),akhiri dg 999 : 4
Masukkan nilai x (int),akhiri dg 999 : 3
Masukkan nilai x (int),akhiri dg 999 : 999
Hasil penjumlahan = 14
BUILD SUCCESSFUL (total time: 1 minute 11 seconds)
```

➤ Penjelasan:

Import Library **java.util.Scanner**; yang berfungsi untuk membaca input dari pengguna melalui keyboard. Mendefinisikan class “**PrintXinterasi**”. Dengan mendeklarasikan variable sum yang bernilai 0 dan x yang sama-sama bertipe data integer. Program ini **membaca angka secara berulang**, menjumlahkan semuanya, dan berhenti ketika pengguna memasukkan **999**. Jika angka pertama yang dimasukkan adalah **999**, program langsung keluar dengan pesan “**Kasus kosong**”. Perulangan **tak terbatas (for(;;))** digunakan untuk membaca angka sampai kondisi **break** terpenuhi.

16. PrintXRepeat

```
run:
Masukkan nilai x (int), akhiri dg 999 : 67
Masukkan nilai x (int), akhiri dg 999 : 13
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 80
BUILD SUCCESSFUL (total time: 10 seconds)
```

➤ Penjelasan:

Import Library **java.util.Scanner**; yang berfungsi untuk membaca input dari pengguna melalui keyboard. Mendefinisikan class “**PrintXRepeat**”. Dengan mendeklarasikan variable sum dan x yang sama-sama bertipe data integer. Program ini membaca angka yang diinput pengguna dan menjumlahkannya sampai **999** dimasukkan. Jika angka pertama adalah **999**, program langsung keluar dengan pesan “**Kasus kosong**”. Perulangan **do-while** memastikan bahwa setidaknya satu angka diproses sebelum pengecekan kondisi berhenti.

17. PrintXWhile

```
run:
Masukkan nilai x (int), akhiri dg 999 : 7
Masukkan nilai x (int), akhiri dg 999 : 3
Masukkan nilai x (int), akhiri dg 999 : 10
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 20
BUILD SUCCESSFUL (total time: 9 seconds)
```

➤ Penjelasan:

Import Library **java.util.Scanner**; yang berfungsi untuk membaca input dari pengguna melalui keyboard. Mendefinisikan class “**PrintXWhile**”. Dengan mendeklarasikan variable sum dan x yang sama-sama bertipe data integer. Program ini membaca angka yang diinput pengguna dan menjumlahkannya sampai **999** dimasukkan. Perulangan **while** digunakan untuk memastikan program terus meminta input hingga kondisi berhenti tercapai.

18. SubProgram

```
run:
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
34
76
Ke dua bilangan : a = 34 b = 76
Maksimum = 76
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 76 b = 34
BUILD SUCCESSFUL (total time: 5 seconds)
```

➤ Penjelasan:

Import Library **java.util.Scanner**; yang berfungsi untuk membaca input dari pengguna melalui keyboard. Mendefinisikan class “**SubProgram**”. Membuat fungsi **maxab** dengan parameter a dan b bertipe data integer yang digunakan untuk mengembalikan nilai maksimum dari dua bilangan. Lalu, membuat prosedur dengan nama **Tukar** dengan parameter a dan b bertipe data integer, **Prosedur tukar** menukar dua bilangan, tetapi hanya berlaku dalam prosedur (karena Java menggunakan **pass by value**). **Variabel a dan b dalam main tidak berubah setelah pemanggilan tukar**, karena Java tidak mengubah nilai asli dalam metode pemanggil kecuali menggunakan array atau objek.

19. Tempair

```
run:
Contoh IF tiga kasus
Temperatur (der. C) = 120
Wujud air uap/gas
120BUILD SUCCESSFUL (total time: 17 seconds)
```

➤ Penjelasan:

Import Library **java.util.Scanner**; yang berfungsi untuk membaca input dari pengguna melalui keyboard. Mendefinisikan class “**SubProgram**”. Deklarasi variable T dengan type data integer. □ Program ini menentukan **wujud air berdasarkan suhu** menggunakan percabangan if-else. **if (T < 0)** menangani air dalam bentuk **es**. **else if ((0 <= T) && (T <= 100))** menangani air dalam bentuk **cair**. **else if (T > 100)** menangani air dalam bentuk **uap/gas**.

