

**PRAKTIKUM  
PEMROGRAMAN BERBASIS OBJEK**



Nama : Nabilah Tika Mushlihah Thahir

Stambuk : 13020230045

Dosen : Mardiyah Hasnawi, S.Kom.,M.T.

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

## Tugas 2 Pemrograman Berbasis Objek

Nama : Nabilah Tika Mushlihah Thahir

NIM 13020230045

Kelas : B1

### Program 1 :

```
D:\ILLA's\GitHub\Tugas-PB0\Tugas 2\Source code>javac BacaString.java
D:\ILLA's\GitHub\Tugas-PB0\Tugas 2\Source code>java BacaString

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

#### Keterangan :

Program ini menggunakan `BufferedReader` untuk membaca input dari pengguna. Variabel `str` bertipe `String` dideklarasikan untuk menyimpan input. Program mencetak pesan "Baca string dan Integer:" lalu meminta pengguna memasukkan sebuah string. Nilai yang diketikkan kemudian dibaca menggunakan `readLine()` dan ditampilkan kembali dengan teks "String yang dibaca : " diikuti oleh input pengguna.

### Program 2 :

```
Print satu baris ....
Print satu baris ....
Print satu baris ....
^C
D:\ILLA's\GitHub\Tugas-PB0\Tugas 2\Source code>
```

#### Keterangan :

Program ini mendefinisikan kelas `ForEver` yang menjalankan perulangan tak terbatas menggunakan `while (true)`. Program mencetak pesan "Program akan looping, akhiri dengan ^c" sekali, lalu terus mencetak "Print satu baris ...." tanpa henti. Program hanya bisa dihentikan secara paksa, misalnya dengan menekan `Ctrl + C`.

### Program 3 :

```
D:\ILLA's\GitHub\Tugas-PB0\Tugas 2\Source code>javac If1.java
D:\ILLA's\GitHub\Tugas-PB0\Tugas 2\Source code>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : 20

Nilai a positif 20
```

Keterangan :

Program ini menggunakan Scanner untuk membaca input dari pengguna. Variabel a bertipe integer dideklarasikan untuk menyimpan input tersebut. Program meminta pengguna mengetikkan suatu bilangan bulat, lalu membaca nilainya menggunakan nextInt(). Jika nilai a lebih besar atau sama dengan 0, program mencetak "Nilai a positif" diikuti dengan nilai a. Jika nilainya negatif, tidak ada output tambahan karena tidak ada kondisi lain yang diatur.

#### Program 4 :

```
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java If2
Contoh IF dua kasus
Ketikkan suatu nilai integer :2
Nilai a positif 2
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>|
```

Keterangan :

Program ini menggunakan Scanner untuk membaca input integer dari pengguna dan menentukan apakah nilainya positif atau negatif. Variabel a bertipe integer menyimpan input tersebut. Program meminta pengguna mengetikkan bilangan bulat, lalu memeriksa nilainya menggunakan if-else. Jika  $a \geq 0$ , program mencetak "Nilai a positif" diikuti dengan nilai a. Jika  $a < 0$ , program mencetak "Nilai a negatif" diikuti dengan nilai a. Dengan demikian, program mencakup dua kasus komplementer untuk memeriksa tanda bilangan.

#### Program 5 :

```
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer :-4
Nilai a negatif -4
```

Keterangan :

Program ini menggunakan Scanner untuk membaca input integer dari pengguna dan menentukan apakah nilainya positif, nol, atau negatif. Variabel a bertipe integer menyimpan input tersebut. Program meminta pengguna memasukkan bilangan bulat, lalu memeriksa nilainya menggunakan if-else if-else. Jika  $a > 0$ , program mencetak "Nilai a positif" diikuti dengan nilai a. Jika  $a == 0$ , program mencetak "Nilai Nol" diikuti dengan nilai a. Jika  $a < 0$ , program mencetak "Nilai a negatif" diikuti dengan nilai a. Dengan demikian, program mencakup tiga kasus untuk memeriksa tanda bilangan.

#### Program 6 :

```
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>javac KasusBoolean.java
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java KasusBoolean
true
benar
```

Keterangan :

Program ini mendeklarasikan variabel bool bertipe boolean dan menginisialisasinya dengan nilai true. Program kemudian memeriksa nilai bool menggunakan if-else. Jika bool true, program mencetak "true", jika false, mencetak "false" (meskipun tidak akan pernah terjadi karena bool diatur ke true). Selanjutnya, program memeriksa negasi bool dengan if(!bool). Karena !bool berarti false, program mencetak "benar". Dengan demikian, program menunjukkan penggunaan tipe data boolean dan operator logika dasar

#### Program 7 :

```
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
N RETURN
Yang anda ketik adalah huruf mati

D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
a
Yang anda ketik adalah a
```

Keterangan :

Program ini menggunakan Scanner untuk membaca satu karakter dari pengguna dan menentukan apakah karakter tersebut adalah huruf vokal atau bukan. Variabel cc bertipe char digunakan untuk menyimpan input. Program memeriksa nilai cc menggunakan switch-case. Jika pengguna mengetikkan 'a', 'u', 'e', 'i', atau 'o', program mencetak huruf vokal yang diketik. Jika karakter lain diketik, default case akan dijalankan, mencetak "Yang anda ketik adalah huruf mati". Program ini menunjukkan cara kerja switch-case untuk menangani beberapa kondisi secara efisien.

#### Program 8 :

```
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>javac Konstant.java

D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java Konstant
Jari-jari lingkaran =20
Luas lingkaran = 1256.6001
Akhir program
```

Keterangan :

Program ini menggunakan Scanner untuk membaca jari-jari lingkaran dari pengguna dan menghitung luas lingkaran. Variabel r bertipe float digunakan untuk menyimpan jari-jari, sedangkan PHI adalah konstanta bertipe float dengan nilai 3.1415f. Program meminta pengguna memasukkan jari-jari, lalu menghitung luas lingkaran menggunakan rumus  $\text{PHI} * r * r$ . Hasil perhitungan ditampilkan, diikuti dengan pesan "Akhir program". Program ini menunjukkan penggunaan konstanta dengan final dan perhitungan matematis sederhana.

#### Program 9 :

```
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN:
2 3
Ke dua bilangan: a = 2 b = 3
Nilai b yang maksimum: 3
```

#### Keterangan :

Program ini menggunakan **Scanner** untuk membaca dua bilangan integer dari pengguna dan menentukan bilangan yang lebih besar. Variabel a dan b digunakan untuk menyimpan bilangan tersebut. Setelah pengguna memasukkan kedua bilangan, program mencetak nilainya. Menggunakan **if-else**, program membandingkan a dan b: jika a lebih besar atau sama dengan b, program mencetak "Nilai a yang maksimum"; jika tidak, program mencetak "Nilai b yang maksimum". Program ini menunjukkan cara membandingkan dua bilangan dan menampilkan nilai maksimum di antara keduanya.

#### Program 10 :

```
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>javac PriFor.java
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java PriFor
Baca N, print 1 s/d N N = 2
1
2
Akhir program
```

#### Keterangan :

Program ini menggunakan Scanner untuk membaca sebuah bilangan integer N dari pengguna, lalu mencetak bilangan dari 1 hingga N menggunakan perulangan for. Variabel i digunakan sebagai penghitung (counter) yang dimulai dari 1 dan bertambah satu setiap iterasi, hingga mencapai nilai N. Setelah perulangan selesai, program mencetak "Akhir program". Program ini menunjukkan cara kerja for loop untuk mencetak deret bilangan secara berurutan sesuai input pengguna.

#### Program 11 :

```
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>javac PrintIterasi.java
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java PrintIterasi
Nilai N >0 = 5
Print i dengan ITERATE :
1
2
3
4
5
```

#### Keterangan :

Program ini menggunakan **Scanner** untuk membaca bilangan integer N dari pengguna, lalu mencetak bilangan dari 1 hingga N menggunakan **perulangan for tanpa kondisi eksplisit**. Variabel i diinisialisasi dengan 1 dan bertambah satu setiap iterasi. Perulangan akan terus berjalan hingga kondisi **if (i == N)** terpenuhi, di mana program akan keluar dari perulangan menggunakan **break**. Jika kondisi belum terpenuhi, nilai i akan bertambah. Program ini menunjukkan cara menggunakan **for loop tanpa syarat** dan menghentikan proses dengan **break**.

#### Program 12 :

```
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java PrintRepeat
Nilai N >0 = 2
Print i dengan REPEAT:
1
2
```

#### Keterangan :

Program ini menggunakan Scanner untuk membaca bilangan integer N dari pengguna, lalu mencetak bilangan dari 1 hingga N menggunakan perulangan do-while. Variabel i diinisialisasi dengan 1 sebagai elemen pertama. do-while memastikan proses mencetak nilai i dilakukan setidaknya satu kali, kemudian i ditambah 1 di setiap iterasi. Perulangan akan terus berlanjut selama i <= N. Program ini menunjukkan cara menggunakan do-while loop untuk melakukan perulangan yang selalu dieksekusi minimal sekali, meskipun kondisinya salah sejak awal.

#### Program 13 :

```
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java PrintWhile
Nilai N >0 = 5
Print i dengan WHILE:
1
2
3
4
5
```

#### Keterangan :

Program ini menggunakan Scanner untuk membaca bilangan integer N dari pengguna, lalu mencetak bilangan dari 1 hingga N menggunakan perulangan while. Variabel i diinisialisasi dengan 1 sebagai elemen pertama. Selama kondisi **i <= N** terpenuhi, program akan mencetak nilai i, lalu menambah i satu per satu. Perulangan berhenti ketika nilai i melebihi N. Program ini menunjukkan cara menggunakan while loop untuk melakukan perulangan yang bergantung pada kondisi tertentu dan akan terus berjalan selama syaratnya masih benar (true).

#### Program 14 :

```
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java PrintWhile1
Nilai N >0 = 5
Print i dengan WHILE (ringkas):
1
2
3
4
5
```

Keterangan :

Program ini menggunakan Scanner untuk membaca bilangan integer N dari pengguna, lalu mencetak bilangan dari 1 hingga N menggunakan perulangan while versi ringkas. Variabel i diinisialisasi dengan 1, dan kondisi while ( $i \leq N$ ) memastikan perulangan terus berjalan selama i kurang dari atau sama dengan N. Pada setiap iterasi, nilai i langsung dicetak dan dinaikkan sekaligus menggunakan  $i++$ . Program ini menunjukkan cara menggunakan while loop dengan penulisan singkat untuk melakukan perulangan dan memperbarui nilai penghitung dalam satu baris.

#### Program 15 :

```
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java PrintXinterasi
Masukkan nilai x (int), akhiri dg 999: 10
Masukkan nilai x (int), akhiri dg 999: 10
Masukkan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 20
```

Keterangan :

Program ini menggunakan Scanner untuk membaca bilangan integer x dari pengguna dan menjumlahkannya menggunakan perulangan tak terbatas (for (;;)) hingga pengguna memasukkan angka 999 sebagai tanda berhenti. Variabel Sum digunakan untuk menyimpan total penjumlahan, diinisialisasi dengan 0. Jika angka pertama yang dimasukkan adalah 999,

program mencetak "Kasus kosong". Jika tidak, nilai pertama langsung ditambahkan ke Sum, lalu program terus meminta input hingga 999 dimasukkan. Setiap bilangan baru ditambahkan ke Sum, dan saat perulangan berhenti, hasil penjumlahan ditampilkan. Program ini menunjukkan penggunaan infinite loop dan break untuk mengontrol proses penjumlahan data yang tidak diketahui jumlahnya.

#### Program 16 :

```
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java PrintXRepeat
Masukkan nilai x (int), akhiri dg 999: 5
Masukkan nilai x (int), akhiri dg 999: 5
Masukkan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 10

D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java PrintXRepeat
Masukkan nilai x (int), akhiri dg 999: 999
Kasus kosong
```

Keterangan :

Program ini menggunakan Scanner untuk membaca bilangan integer x dari pengguna dan menjumlahkannya menggunakan perulangan do-while hingga pengguna memasukkan angka 999 sebagai tanda berhenti. Jika angka pertama yang dimasukkan adalah 999, program langsung mencetak "Kasus kosong".

Jika tidak, variabel Sum diinisialisasi dengan 0, lalu perulangan **do-while** dijalankan: nilai x ditambahkan ke Sum, dan program kembali meminta input baru. Perulangan akan terus berjalan sampai pengguna memasukkan **999**. Setelah perulangan selesai, program mencetak total penjumlahan. Program ini memperlihatkan bagaimana **do-while loop** digunakan untuk memastikan proses berjalan setidaknya sekali, meskipun input awal mungkin langsung memenuhi kondisi berhenti.

#### Program 17 :

```
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>javac PrintXWhile.java

D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java PrintXWhile
Masukkan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 0

D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java PrintXWhile
Masukkan nilai x (int), akhiri dg 999: 1
Masukkan nilai x (int), akhiri dg 999: 1
Masukkan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 2
```

#### Keterangan :

Program ini menggunakan Scanner untuk membaca bilangan integer x dari pengguna dan menjumlahkannya menggunakan perulangan while hingga pengguna memasukkan angka 999 sebagai tanda berhenti. Variabel Sum diinisialisasi dengan 0 untuk menyimpan total penjumlahan. Program pertama-tama meminta input awal, lalu masuk ke perulangan while yang terus berjalan selama x tidak sama dengan 999.

Di dalam perulangan, nilai x ditambahkan ke Sum, kemudian pengguna diminta untuk memasukkan angka berikutnya. Ketika 999 dimasukkan, perulangan berhenti dan program mencetak hasil penjumlahan. Program ini menunjukkan bagaimana while loop digunakan untuk iterasi dengan kondisi berhenti yang dicek sebelum proses dilakukan.

#### Program 18 :

```
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>javac SubProgram.java

D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java SubProgram
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN: 2 5
Ke dua bilangan: a = 2 b = 5
Maksimum = 5
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 5 b = 2
```

#### Keterangan :

Program ini menggunakan fungsi maxab() untuk mengembalikan bilangan terbesar dari dua bilangan, dan prosedur tukar() untuk menukar nilai dua bilangan (meskipun tidak mempengaruhi nilai asli karena Java menggunakan pass by value). Program utama meminta dua bilangan dari pengguna, mencetak keduanya, menampilkan bilangan maksimum, lalu memanggil prosedur untuk menukar bilangan dan mencetak hasilnya.

#### Program 19 :



```
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = -5
Wujud air beku
-5
~
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 25
Wujud air cair
25
--
D:\ILLA's\GitHub\Tugas-PBO\Tugas 2\Source code>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 110
Wujud air uap/gas
110
```

Keterangan :

Program ini membaca suhu air (dalam °C) dari pengguna, lalu menggunakan IF tiga kasus untuk menentukan wujud air: beku jika suhu di bawah 0, cair jika antara 0–100, dan uap/gas jika di atas 100. Hasil wujud air dan suhu ditampilkan.