

Tugas II

Makassar, 07 Maret 2025

**PRAKTIKUM
PEMROGRAMAN BERIOIENTASI OBJEK**



Nama : Firli Anastasya Hafid
Stambuk : 13020230241
Dosen : Mardiyah Hasnawi, S.Kom., M.T.

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

```
C:\Tugas-PBO\Tugas 2>java BacaString

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

Penjelasan:

Pada program java ini berfungsi untuk membaca string dari pengguna melalui input keyboard menggunakan BufferedReader.

- `java.io.BufferedReader` dan `java.io.InputStreamReader` digunakan untuk membaca input dari keyboard.
- `java.io.IOException` untuk menangani kemungkinan kesalahan saat membaca input.
- `javax.swing.*` diimpor tetapi tidak digunakan dalam program.
- Program meminta pengguna memasukkan sebuah string.
- Menggunakan `dataIn.readLine()` untuk membaca input dan menyimpannya dalam variabel `str`.
- Program mencetak kembali string yang telah dimasukkan dengan format String yang dibaca : `<input pengguna>`

2. Program 2

[illegible]

```
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
^CPrint satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
```

[illegible]

Penjelasan:

Program **ForEver** ini adalah contoh **infinite loop** (perulangan tak terbatas) menggunakan perulangan while (true).

1. Menampilkan Pesan Awal

- `System.out.println("Program akan looping, akhiri dengan ^c");`
- Memberi tahu pengguna bahwa program akan berjalan terus-menerus, dan hanya bisa dihentikan dengan **Ctrl + C** (di terminal/command prompt) atau menghentikan proses secara paksa.

2. Perulangan Tanpa Henti (while (true))

- `while (true) { }` membuat program terus mengulang tanpa kondisi berhenti.
- `System.out.print("Print satu baris\n");` akan terus mencetak teks "**Print satu baris**" tanpa henti.

3. Cara Menghentikan Program

Karena tidak ada kondisi untuk keluar dari loop, program hanya bisa dihentikan secara manual dengan, Menekan Ctrl + C di terminal (untuk menghentikan paksa).

3. Program 3

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

Nilai a positif 1
```

Penjelasan:

- `import java.util.Scanner;` digunakan untuk memungkinkan **input dari pengguna** melalui keyboard.
- `Scanner masukan = new Scanner(System.in);` membuat objek Scanner untuk membaca input.
- `int a;` mendeklarasikan variabel a untuk menyimpan angka yang dimasukkan pengguna.
- `System.out.print("Ketikkan suatu nilai integer : ");` meminta pengguna memasukkan **bilangan bulat**.
- `a = masukan.nextInt();` membaca input tersebut dan menyimpannya dalam variabel a.
- Jika nilai a **lebih besar atau sama dengan nol**, maka program mencetak bahwa angka tersebut adalah **bilangan positif**.

4. Program 4

```
C:\Tugas-PB0\Tugas 2>java If2
Contoh IF dua kasus
Ketikkan suatu nilai integer :-2
Nilai a negatif -2

C:\Tugas-PB0\Tugas 2>java If2
Contoh IF dua kasus
Ketikkan suatu nilai integer :1
Nilai a positif 1
```

Penjelasan:

1. Mengimpor Library Scanner

- `import java.util.Scanner;` memungkinkan program membaca input dari pengguna melalui keyboard.

2. Deklarasi Scanner dan Variabel

- `Scanner masukan = new Scanner(System.in);` membuat objek Scanner untuk membaca input.
- `int a;` mendeklarasikan variabel a untuk menyimpan bilangan yang dimasukkan oleh pengguna.

3. Meminta Input dari Pengguna

- `System.out.print("Ketikkan suatu nilai integer : ");` menampilkan prompt agar pengguna memasukkan sebuah bilangan bulat.
- `a = masukan.nextInt();` membaca nilai integer dari input dan menyimpannya dalam variabel a.

4. Percabangan IF-ELSE

- `if (a >= 0):` Jika nilai a **lebih besar atau sama dengan nol**, program mencetak bahwa angka tersebut **positif**.
- `else:` Jika kondisi if tidak terpenuhi (artinya `a < 0`), maka program mencetak bahwa angka tersebut **negatif**.

5. Program 5

```
C:\Tugas-PB0\Tugas 2>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer :9
Nilai a positif 9

C:\Tugas-PB0\Tugas 2>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer :0
Nilai Nol 0

C:\Tugas-PB0\Tugas 2>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer :-6
Nilai a negatif -6
```

Penjelasan:

- Program ini **membedakan angka positif, nol, dan negatif** menggunakan **percabangan if-else if-else**.
- **Lebih lengkap dibanding If1 dan If2**, karena menangani tiga kemungkinan nilai.

6. Program 6

```
C:\Tugas-PBO\Tugas 2>java KasusBoolean
true
benar
```

Penjelasan:

Program KasusBoolean ini menunjukkan penggunaan tipe data boolean dan bagaimana nilai true atau false digunakan dalam percabangan if-else di Java.

7. Program 7

```
C:\Tugas-PBO\Tugas 2>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
E
Yang anda ketik adalah huruf mati

C:\Tugas-PBO\Tugas 2>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
L
Yang anda ketik adalah huruf mati
```

Penejasan:

- Scanner masukan = new Scanner(System.in); digunakan untuk membaca input.
- char cc; digunakan untuk menyimpan **karakter** yang diketik oleh pengguna.
- masukan.next() membaca satu kata sebagai string.
- .charAt(0) mengambil **karakter pertama** dari string tersebut.
- Program akan mengecek apakah cc adalah salah satu huruf vokal (**a, i, u, e, o**).
- Jika iya, maka akan mencetak pesan sesuai dengan huruf yang diketik.
- Jika bukan, maka akan masuk ke bagian default, yang mencetak "Yang anda ketik adalah huruf mati".

8. Program 8

```
C:\Tugas-PBO\Tugas 2>java Konstant
Jari-jari lingkaran =12
Luas lingkaran = 452.37598
Akhir program
```

Penjelasan:

- final digunakan untuk mendeklarasikan **konstanta** yang nilainya **tidak bisa diubah**.
- PHI adalah nilai **π (pi)** yang digunakan dalam perhitungan luas lingkaran.
- Tipe data float digunakan, dan ditambahkan **huruf f** agar dikenali sebagai tipe float.

9. Program 9

```
C:\Tugas-PBO\Tugas 2>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
10
6
Ke dua bilangan : a = 10 b = 6
Nilai a yang maksimum 10

C:\Tugas-PBO\Tugas 2>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
1
5
Ke dua bilangan : a = 1 b = 5
Nilai b yang maksimum: 5
```

Penjelasan:

- Program meminta pengguna memasukkan **dua bilangan**.
- `masukan.nextInt()` membaca angka pertama dan menyimpannya ke `a`.
- `masukan.nextInt()` membaca angka kedua dan menyimpannya ke `b`.
- Jika `a` lebih besar atau sama dengan `b`, maka `a` adalah nilai maksimum.
- Jika tidak, berarti `b` lebih besar dan menjadi nilai maksimum.

10. Program 10

```
C:\Tugas-PBO\Tugas 2>java PriFor
Baca N, print 1 s/d N N = 5
1
2
3
4
5
Akhir program
```

Penjelasan:

- Scanner masukan digunakan untuk membaca input pengguna.
- `int i, N;` mendeklarasikan variabel:
 - a) `i` sebagai penghitung dalam perulangan.
 - b) `N` sebagai batas akhir perulangan (input dari pengguna).
- Program ini mencetak angka **dari 1 hingga N** menggunakan perulangan `for`.
- Jika `N = 0` atau **negatif**, program sebaiknya tidak menjalankan perulangan.

11. Program 11

```
C:\Tugas-PBO\Tugas 2>java PrintIterasi
Nilai N >0 = 6
Print i dengan ITERATE :
1
2
3
4
5
6
```

Penjelasan:

- Scanner masukan digunakan untuk membaca input pengguna.
- int N; menyimpan nilai batas akhir perulangan.
- int i; sebagai penghitung perulangan.
- for (;;) adalah bentuk **infinite loop (perulangan tanpa batas)**.
- **Kondisi berhenti ditentukan dengan if (i == N) break;**, yang akan menghentikan perulangan jika i mencapai N.
- Jika i belum mencapai N, nilai i akan terus bertambah (i++).

12. Program 12

```
C:\Tugas-PBO\Tugas 2>java PrintRepeat
Nilai N >0 = 3
Print i dengan REPEAT:
1
2
3

C:\Tugas-PBO\Tugas 2>java PrintRepeat
Nilai N >0 = 0
Print i dengan REPEAT:
1
```

Penjelasan:

- a) Scanner masukan digunakan untuk membaca input pengguna.
- b) int N, i; mendeklarasikan variabel:
 - N sebagai batas akhir angka yang dicetak.
 - i sebagai penghitung dalam perulangan.
- c) **do-while** akan menjalankan blok kode minimal satu kali, terlepas dari kondisi awal.
- d) i = 1 adalah inisialisasi penghitung.
- e) Setiap iterasi:
 - Nilai i dicetak.
 - i bertambah satu (i++).
- f) Perulangan berhenti jika i > N, perulangan tetap **berjalan minimal sekali**, meskipun N = 0.

13. Program 13

```
C:\Tugas-PBO\Tugas 2>java PrintWhile
Nilai N >0 = 3
Print i dengan WHILE:
1
2
3
```

Penjelasan:

- Scanner masukan digunakan untuk membaca input pengguna.
- int N; menyimpan nilai batas akhir perulangan.
- int i = 1; sebagai penghitung perulangan, dimulai dari 1.
- Program meminta pengguna memasukkan nilai N, yaitu batas akhir angka yang akan dicetak.
- while (i <= N) berarti program akan mencetak angka selama nilai i masih kurang dari atau sama dengan N.
- System.out.println(i++); akan mencetak i, kemudian menambah nilai i setelahnya
- Jika pengguna memasukkan N = 0 atau N < 0, tidak ada angka yang dicetak, karena kondisi awal i = 1 langsung gagal memenuhi i <= N.

14. Program 14

```
C:\Tugas-PBO\Tugas 2>java PrintWhile1
Nilai N >0 = 3
Print i dengan WHILE (ringkas):
1
2
3
```

Penjelasan:

- Scanner masukan digunakan untuk membaca input dari pengguna.
- int N; menyimpan nilai batas akhir perulangan.
- int i = 1; sebagai penghitung perulangan, dimulai dari 1.
- Kondisi while (i <= N) memastikan program mencetak angka dari 1 hingga N.
- System.out.println(i++); mencetak nilai i, lalu menambah nilai i setelahnya.

15. Program 15

```
C:\Tugas-PBO\Tugas 2>java PrintXinterasi
Masukkan nilai x (int), akhiri dg 999: 1
Masukkan nilai x (int),akhiri dg 999 : 3
Masukkan nilai x (int),akhiri dg 999 : 8
Masukkan nilai x (int),akhiri dg 999 : 999
Hasil penjumlahan = 12
```

Penjelasan:

- Jika input pertama **bukan 999**, nilai tersebut langsung dimasukkan ke Sum.
- Program memasuki **loop tanpa batas (for (;))**, yang akan berjalan terus hingga pengguna memasukkan **999**.
- Setiap angka yang dimasukkan pengguna akan ditambahkan ke Sum.
- Jika pengguna memasukkan **999**, perulangan berhenti dengan break.

16. Program 16

```
C:\Tugas-PBO\Tugas 2>java PrintXRepeat
Masukkan nilai x (int), akhiri dg 999 : 2
Masukkan nilai x (int), akhiri dg 999 : 1
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 3
```


Penejelasan:

- **Perulangan do-while** memastikan **setidaknya satu angka** dijumlahkan sebelum mengecek kondisi $x \neq 999$.
- Setiap angka yang dimasukkan pengguna akan ditambahkan ke Sum.
- Jika pengguna memasukkan **999**, perulangan berhenti.

17. Program 17

```
C:\Tugas-PBO\Tugas 2>java PrintXWhile
Masukkan nilai x (int), akhiri dg 999 : 1
Masukkan nilai x (int), akhiri dg 999 : 8
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 9
```

Penjelasan:

- a) Inisialisasi: $\text{Sum} = 0$ untuk menyimpan hasil penjumlahan.
- b) Input pertama (x) dimasukkan sebelum perulangan dimulai.
- c) Perulangan while berjalan selama $x \neq 999$:
 - Setiap angka ditambahkan ke Sum.
 - Meminta input berikutnya.
- d) Ketika $x == 999$, perulangan berhenti dan hasil penjumlahan ditampilkan.

18. Program 18

Penjelasan:

- a) Metode `maxab(int a, int b)`
 - Mengembalikan nilai terbesar antara a dan b menggunakan operator ternary ($? :$).
- b) Metode `tukar(int a, int b)`
 - Menukar nilai a dan b menggunakan variabel sementara temp.

- Namun, karena Java menggunakan pass-by-value, perubahan nilai dalam metode ini tidak berpengaruh ke variabel di main().

c) Metode main(String[] args)

- Meminta input dua bilangan dari pengguna.
- Menampilkan kedua bilangan dan nilai maksimum menggunakan maxab().
- Memanggil tukar(a, b), tetapi nilai asli di main() tidak berubah karena Java tidak mendukung pass-by-reference untuk tipe primitif.

19. Program 19

```
C:\Tugas-PBO\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = -9
Wujud air beku
-9
C:\Tugas-PBO\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 12
Wujud air cair
12
```

Penjelasan:

1. Input suhu (T) dari pengguna.
2. Percabangan if-else menentukan wujud air:
 - Jika $T < 0 \rightarrow$ Air dalam keadaan beku.
 - Jika $0 \leq T \leq 100 \rightarrow$ Air dalam keadaan cair.
 - Jika $T > 100 \rightarrow$ Air dalam keadaan uap/gas.