

Tugas II

Makassar, 06 Maret 2024

TEORI
PEMROGRAMAN BERORIENTASI OBJEK



Nama. : Rezki
Nim : 13020230139
Dosen : Mardiyah Hasnawi, S.Kom., M.T.

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

1. BacaString

```
[macbookpro@Macbooks-MacBook-Pro PBO % javac BacaString.java
[macbookpro@Macbooks-MacBook-Pro PBO % java BacaString

Baca string dan Integer:
masukkan sebuah string: A
String yang dibaca : A
macbookpro@Macbooks-MacBook-Pro PBO %
```

Program ini menunjukkan cara membaca input string dari pengguna. Program menggunakan `BufferedReader` untuk mengambil input dari keyboard. Pengguna diminta memasukkan string, kemudian program membaca input tersebut dengan method `readLine()` dan menampilkannya kembali di layar. Konsep ini fundamental karena interaksi dengan pengguna merupakan bagian penting dalam pemrograman.

2. ForEver

[illegible]

Program ini mendemonstrasikan konsep infinite loop (perulangan tak terhingga). Program menggunakan `while(true)` yang akan terus berjalan tanpa berhenti karena kondisinya selalu bernilai `true`. Setiap iterasi loop mencetak "Print satu baris....". Program hanya dapat dihentikan dengan menekan `Ctrl+C`. Konsep ini penting dipahami untuk menghindari bug dalam pemrograman.

3. If1

```
Last login: Thu Mar 13 22:02:43 on ttys000
macbookpro@Macbooks-MacBook-Pro PBO % javac If1.java
macbookpro@Macbooks-MacBook-Pro PBO % java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : 2

Nilai a positif 2%
macbookpro@Macbooks-MacBook-Pro PBO %
```

Program ini menunjukkan penggunaan struktur kondisional if sederhana. Program membaca nilai integer dari pengguna, kemudian memeriksa apakah nilai tersebut positif atau nol ($a \geq 0$). Jika kondisi terpenuhi, program menampilkan pesan bahwa

nilai tersebut positif beserta nilainya. Program ini menunjukkan dasar pengambilan keputusan dalam pemrograman.

4. If2

```
Last login: Thu Mar 13 22:03:47 on ttys000
[macbookpro@Macbooks-MacBook-Pro PBO % javac If2.java
[macbookpro@Macbooks-MacBook-Pro PBO % java If2
Contoh IF dua kasus
Ketikkan suatu nilai integer :23
Nilai a positif 23
macbookpro@Macbooks-MacBook-Pro PBO % █
```

Program ini memperluas konsep If1 dengan menambahkan struktur else. Program membaca nilai integer dan memeriksa apakah nilai tersebut positif/nol atau negatif. Program menampilkan pesan sesuai dengan kondisi yang terpenuhi. Program ini menunjukkan bagaimana menangani dua kemungkinan hasil yang saling eksklusif dalam pengambilan keputusan.

5. If3

```
Last login: Thu Mar 13 22:03:57 on ttys000
[macbookpro@Macbooks-MacBook-Pro PBO % javac If3.java
[macbookpro@Macbooks-MacBook-Pro PBO % java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer :3
Nilai a positif 3
macbookpro@Macbooks-MacBook-Pro PBO % █
```

Program ini mendemonstrasikan penggunaan struktur if-else if-else untuk menangani tiga kemungkinan kondisi. Program membaca nilai integer dan mengkategorikannya sebagai positif ($a > 0$), nol ($a == 0$), atau negatif ($a < 0$). Program menampilkan pesan sesuai dengan kategori yang terpenuhi. Struktur ini efisien untuk menangani multiple kondisi yang saling eksklusif.

6. KasusBoolean

```
[macbookpro@Macbooks-MacBook-Pro PBO % javac KasusBoolean.java
[macbookpro@Macbooks-MacBook-Pro PBO % java KasusBoolean
true
benar
macbookpro@Macbooks-MacBook-Pro PBO % █
```

Program ini menunjukkan penggunaan tipe data boolean dan operator logika. Program mendeklarasikan variabel boolean dengan nilai true, kemudian menggunakan struktur if-else untuk memeriksa nilai tersebut. Program juga mendemonstrasikan penggunaan operator negasi (!) untuk membalik nilai boolean. Pemahaman operator logika penting untuk membuat kondisi kompleks.

7. KasusSwitch

```
Last login: Fri Mar 14 18:27:50 on console
macbookpro@Macbooks-MacBook-Pro PBO % javac KasusSwitch.java
macbookpro@Macbooks-MacBook-Pro PBO % java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
a
Yang anda ketik adalah a
macbookpro@Macbooks-MacBook-Pro PBO % █
```

Program ini mendemonstrasikan penggunaan struktur switch-case. Program membaca sebuah karakter dari pengguna dan menggunakan switch untuk memeriksa apakah karakter tersebut vokal (a, e, i, o, u) atau huruf mati. Program menampilkan pesan yang sesuai dengan kasus yang terpenuhi. Switch-case merupakan alternatif yang lebih efisien dari multiple if-else untuk kondisi yang eksak.

8. Konstant

```
macbookpro@Macbooks-MacBook-Pro PBO % javac Konstant.java
macbookpro@Macbooks-MacBook-Pro PBO % java Konstant
Jari-jari lingkaran =15
Luas lingkaran = 706.8375
Akhir program
macbookpro@Macbooks-MacBook-Pro PBO % █
```

Program ini menunjukkan penggunaan konstanta dalam pemrograman. Program mendefinisikan konstanta PHI dengan nilai 3.1415f menggunakan keyword final. Program membaca jari-jari lingkaran dari pengguna, menghitung luas lingkaran, dan menampilkan hasilnya. Penggunaan konstanta menjamin konsistensi nilai dan meningkatkan keterbacaan kode.

9. Max2

```
macbookpro@Macbooks-MacBook-Pro PBO % javac Max2.java
macbookpro@Macbooks-MacBook-Pro PBO % 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
macbookpro@Macbooks-MacBook-Pro PBO % █
```

Program ini menunjukkan aplikasi perbandingan nilai. Program membaca dua nilai integer, membandingkannya, dan menampilkan nilai yang lebih besar. Program menggunakan struktur if-else untuk membuat keputusan berdasarkan hasil perbandingan. Konsep ini fundamental dalam algoritma perbandingan dan pengurutan.

10.PriFor

```
macbookpro@Macbooks-MacBook-Pro PBO % javac PriFor.java
macbookpro@Macbooks-MacBook-Pro PBO % java PriFor
Baca N, print 1 s/d N N = 2
1
2
Akhir program

macbookpro@Macbooks-MacBook-Pro PBO % █
```

Program ini mendemonstrasikan penggunaan perulangan for standar. Program membaca nilai N dan menggunakan for loop untuk mencetak angka dari 1 hingga N. Struktur for (inisialisasi; kondisi; increment) merupakan pola standar untuk perulangan dengan jumlah iterasi yang diketahui sebelumnya.

11.PrintIterasi

```
macbookpro@Macbooks-MacBook-Pro PBO % javac PrintIterasi.java
macbookpro@Macbooks-MacBook-Pro PBO % java PrintIterasi
Nilai N >0 = 5
Print i dengan ITERATE :
1
2
3
4
5
macbookpro@Macbooks-MacBook-Pro PBO % █
```

Program ini menunjukkan penggunaan perulangan for dengan pendekatan berbeda. Program menggunakan for(;;) untuk membuat infinite loop, kemudian menggunakan break untuk keluar dari loop saat kondisi tertentu terpenuhi. Pendekatan ini memberikan fleksibilitas lebih dalam mengendalikan alur perulangan.

12.PrintRepeat

```
macbookpro@Macbooks-MacBook-Pro PBO % javac PrintRepeat.java
macbookpro@Macbooks-MacBook-Pro PBO % java PrintRepeat
Nilai N >0 = 3
Print i dengan REPEAT:
1
2
3
macbookpro@Macbooks-MacBook-Pro PBO % █
```

Program ini menunjukkan penggunaan perulangan do-while. Program membaca nilai N dan menggunakan do-while untuk mencetak angka dari 1 hingga N. Ciri khas do-while adalah minimal sekali eksekusi karena kondisi diperiksa di akhir loop. Struktur ini berguna ketika kode harus dijalankan setidaknya sekali.

13.PrintWhile

```
macbookpro@Macbooks-MacBook-Pro PBO % javac PrintWhile.java
macbookpro@Macbooks-MacBook-Pro PBO % java PrintWhile
Nilai N >0 = 2
Print i dengan WHILE:
1
2
macbookpro@Macbooks-MacBook-Pro PBO % █
```

Program ini mendemonstrasikan penggunaan perulangan while. Program membaca nilai N dan menggunakan while untuk mencetak angka dari 1 hingga N. Berbeda dengan do-while, kondisi while diperiksa di awal sehingga kode di dalam loop mungkin tidak dieksekusi sama sekali jika kondisi awal tidak terpenuhi.

14.PrintWhile1

```
macbookpro@Macbooks-MacBook-Pro PBO % javac PrintWhile1.java
macbookpro@Macbooks-MacBook-Pro PBO % java PrintWhile1
Nilai N >0 = 4
Print i dengan WHILE (ringkas):
1
2
3
4
macbookpro@Macbooks-MacBook-Pro PBO % █
```

Program ini menunjukkan versi ringkas dari perulangan while. Program menggunakan operator post-increment (i++) dalam pemanggilan println, sehingga nilai i dicetak terlebih dahulu baru kemudian nilai i ditambah. Teknik ini menunjukkan optimasi kode untuk membuat program lebih ringkas dan efisien.

15.PrintXinterasi

```
macbookpro@Macbooks-MacBook-Pro PBO % javac PrintXinterasi.java
macbookpro@Macbooks-MacBook-Pro PBO % java PrintXinterasi
Masukkan nilai x (int), akhiri dg 999: 3
Masukkan nilai x (int),akhiri dg 999 : 4
Masukkan nilai x (int),akhiri dg 999 : 8
Masukkan nilai x (int),akhiri dg 999 : 999
Hasil penjumlahan = 15
macbookpro@Macbooks-MacBook-Pro PBO % █
```

Program ini menunjukkan perulangan dengan sentinal value. Program menjumlahkan bilangan-bilangan yang diinput pengguna sampai pengguna memasukkan nilai 999 (sentinal). Program menggunakan perulangan for(;;) dengan kondisi break internal. Pendekatan ini berguna untuk pemrosesan data yang jumlahnya tidak diketahui sebelumnya.

16.PrintXRepeat

```
macbookpro@Macbooks-MacBook-Pro PBO % javac PrintXRepeat.java
macbookpro@Macbooks-MacBook-Pro PBO % java PrintXRepeat
Masukkan nilai x (int), akhiri dg 999 : 7
Masukkan nilai x (int), akhiri dg 999 : 5
Masukkan nilai x (int), akhiri dg 999 : 99
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 111
macbookpro@Macbooks-MacBook-Pro PBO % █
```

Program ini menunjukkan implementasi sentinal value dengan do-while. Program menjumlahkan bilangan-bilangan yang diinput pengguna sampai pengguna memasukkan nilai 999. Struktur do-while menjamin proses penjumlahan minimal sekali dijalankan. Pendekatan ini memberikan alternatif implementasi untuk kasus yang sama.

17.PrintXWhile

```
macbookpro@Macbooks-MacBook-Pro PBO % javac PrintXWhile.java
macbookpro@Macbooks-MacBook-Pro PBO % java PrintXWhile
Masukkan nilai x (int), akhiri dg 999 : 2
Masukkan nilai x (int), akhiri dg 999 : 3
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 5
macbookpro@Macbooks-MacBook-Pro PBO % █
```

Program ini menunjukkan implementasi sentinal value dengan while. Program menjumlahkan bilangan-bilangan yang diinput pengguna sampai pengguna memasukkan nilai 999. Struktur while memeriksa kondisi di awal, sehingga memiliki perilaku berbeda dari do-while jika nilai pertama adalah 999.

18.SubProgram

```
macbookpro@Macbooks-MacBook-Pro PBO % javac SubProgram.java
macbookpro@Macbooks-MacBook-Pro PBO % java SubProgram
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
3
5
Ke dua bilangan : a = 3 b = 5
Maksimum = 5
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 5 b = 3
macbookpro@Macbooks-MacBook-Pro PBO % █
```

Program ini mendemonstrasikan penggunaan method dalam Java. Program mendefinisikan fungsi maxab() untuk mencari nilai maksimum dari dua bilangan, dan prosedur tukar() untuk menukar nilai dua bilangan. Program menunjukkan konsep modularitas dan reusabilitas kode yang penting dalam pengembangan software skala besar.

19.Tempair

```
macbookpro@Macbooks-MacBook-Pro PBO % javac Tempair.java
macbookpro@Macbooks-MacBook-Pro PBO % java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 21
Wujud air cair
21%
macbookpro@Macbooks-MacBook-Pro PBO % █
```

Program ini menunjukkan aplikasi praktis struktur if-else if-else. Program menentukan wujud air (beku, cair, atau gas) berdasarkan suhu yang diinput pengguna. Program menggunakan kondisi bersarang untuk mengklasifikasikan suhu ke dalam tiga kategori wujud air. Ini contoh bagaimana konsep pemrograman dapat diaplikasikan untuk mensimulasikan fenomena fisika.