

Tugas 2

Makassar, 07 Maret 2025

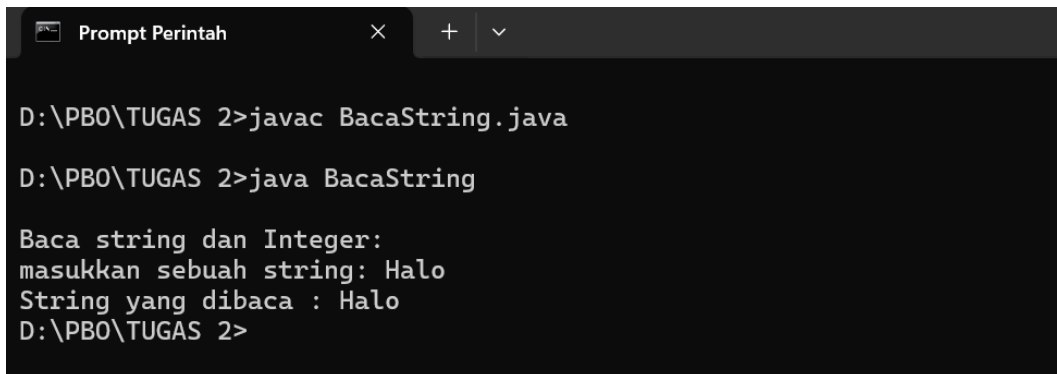
**PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**



Nama : Marsha Selma Rahim
Stambuk : 13020230169
Kelas : B2
Dosen : Mardiyah Hasnawati, S.Kom., M.T.

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR
2024**

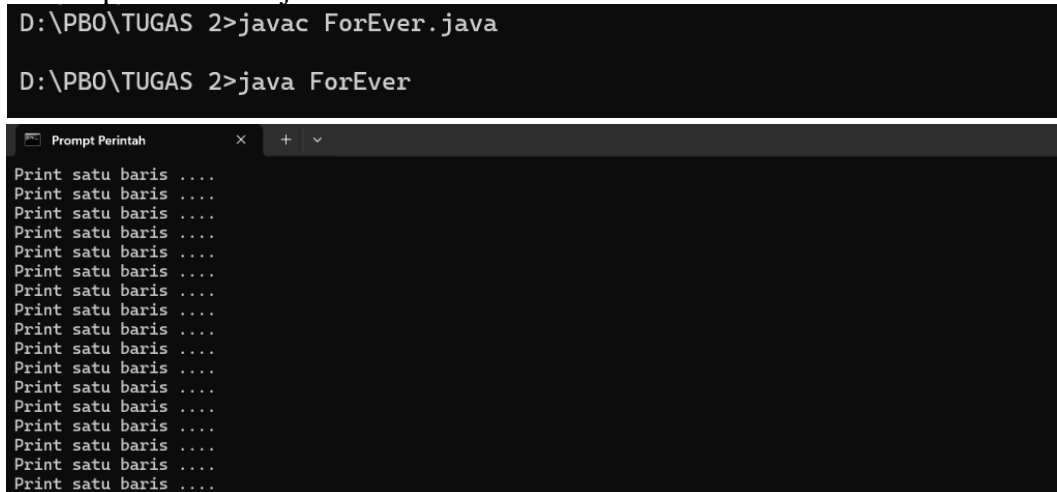
1. BacaString.java adalah nama file kode sumber Java yang akan dikompilasi. Setelah kompilasi berhasil, ini adalah perintah untuk menjalankan program Java yang telah dikompilasi.masukkan sebuah string : Halo Ini adalah keluaran dari program Java. Program meminta pengguna untuk memasukkan sebuah string.Pengguna memasukkan



```
Prompt Perintah
D:\PBO\TUGAS 2>javac BacaString.java
D:\PBO\TUGAS 2>java BacaString
Baca string dan Integer:
masukkan sebuah string: Halo
String yang dibaca : Halo
D:\PBO\TUGAS 2>
```

2. D:\PBO\TUGAS 2>javac ForEver.java

Ini adalah perintah yang dimasukkan ke dalam Command Prompt (cmd) untuk mengkompilasi file kode sumber Java bernama "ForEver.java".javac adalah perintah untuk menjalankan *Java Compiler*, yang mengubah kode sumber Java menjadi *bytecode* yang dapat dieksekusi oleh Java Virtual Machine (JVM). D:\PBO\TUGAS 2> menunjukkan lokasi direktori tempat perintah ini dijalankan.



```
Prompt Perintah
D:\PBO\TUGAS 2>javac ForEver.java
D:\PBO\TUGAS 2>java 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 ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
Print satu baris ....
D:\PBO\TUGAS 2>
```

3. Ini adalah perintah untuk mengkompilasi file kode sumber Java bernama "If1.java" menggunakan *Java Compiler* (javac). Setelah kompilasi berhasil, ini adalah perintah untuk menjalankan program Java yang telah dikompilasi. Ketikkan suatu nilai integer : 2 Ini adalah keluaran dari program Java, yang meminta pengguna untuk memasukkan nilai integer.Pengguna memasukkan nilai "2".Nilai a positif 2

Ini adalah keluaran dari program Java, yang menunjukkan bahwa nilai yang dimasukkan (2) adalah positif. Gambar ini menunjukkan eksekusi program Java yang menggunakan pernyataan if untuk memeriksa apakah suatu nilai integer positif. Program ini meminta pengguna untuk memasukkan nilai integer, dan kemudian menampilkan pesan yang menunjukkan apakah nilai tersebut positif. Dalam contoh ini, pengguna memasukkan nilai 2, dan program dengan benar mengidentifikasinya sebagai nilai positif.

```
D:\PBO\TUGAS 2>javac If1.java  
  
D:\PBO\TUGAS 2>java If1  
Contoh IF satu kasus  
Ketikkan suatu nilai integer : 2  
  
Nilai a positif 2
```

4. Gambar ini menunjukkan eksekusi program Java yang menggunakan pernyataan if untuk memeriksa kondisi suatu nilai integer. Program ini meminta pengguna untuk memasukkan nilai integer, dan kemudian menampilkan pesan berdasarkan kondisi yang diperiksa. Dalam contoh ini, pengguna memasukkan nilai 8, dan program mengidentifikasinya sebagai nilai positif. Kemungkinan besar, program "If2.java" memiliki struktur logika yang akan memberikan output yang berbeda jika nilai yang diinputkan bukan nilai positif, misalnya negatif, atau nol. Secara ringkas, program ini menggunakan percabangan if, untuk mengecek apakah sebuah nilai integer positif atau bukan.

```
D:\PBO\TUGAS 2>javac If2.java  
  
D:\PBO\TUGAS 2>java If2  
Contoh IF dua kasus  
Ketikkan suatu nilai integer :8  
Nilai a positif 8
```

5. Gambar ini menunjukkan eksekusi program Java yang menggunakan pernyataan if untuk memeriksa kondisi suatu nilai integer. Program ini meminta pengguna untuk memasukkan nilai integer, dan kemudian menampilkan pesan berdasarkan kondisi yang diperiksa. Dalam contoh ini, pengguna memasukkan nilai 5, dan program mengidentifikasinya sebagai nilai positif. Kemungkinan besar, program "If3.java" memiliki struktur logika yang akan memberikan output yang berbeda jika nilai yang diinputkan bukan nilai positif, misalnya negatif, atau nol, oleh karena tertulis di baris ke tiga, yaitu "Contoh IF tiga kasus". Secara singkat, program ini menggunakan percabangan if dengan 3 kemungkinan kondisi, untuk mengecek apakah sebuah nilai integer positif, negatif, atau nol.

```
D:\PBO\TUGAS 2>javac PrintIterasi.java
```

```
D:\PBO\TUGAS 2>java PrintIterasi
```

```
Nilai N >0 = 5
```

```
Print i dengan ITERATE :
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

6. Gambar ini menunjukkan eksekusi program Java yang berurusan dengan nilai boolean. Program "KasusBoolean.java" mengkompilasi dan berjalan, dan kemudian menghasilkan output "true" dan "benar". Ini menunjukkan bahwa program tersebut kemungkinan besar melakukan operasi yang mengevaluasi kondisi boolean, dan kondisi tersebut menghasilkan nilai benar (true).Kemungkinan besar, program "KasusBoolean.java" memiliki kode yang mencetak nilai boolean secara langsung, atau melakukan perbandingan yang menghasilkan nilai boolean true.Secara singkat, program ini mencetak nilai boolean benar (true) ke layar,yang diwakili oleh kata "true" dan "benar".

```
D:\PBO\TUGAS 2>javac KasusBoolean.java
```

```
D:\PBO\TUGAS 2>java KasusBoolean
```

```
true
```

```
benar
```

7. Gambar ini menunjukkan eksekusi program Java yang menggunakan pernyataan switch. Program "KasusSwitch.java" meminta pengguna untuk memasukkan sebuah huruf, dan kemudian menampilkan pesan yang menunjukkan huruf yang dimasukkan. Dalam contoh ini, pengguna memasukkan huruf "a", dan program dengan benar mengkonfirmasi.Kemungkinan besar, program "KasusSwitch.java" memiliki struktur logika yang akan memberikan output yang berbeda untuk huruf-huruf yang berbeda.Secara singkat, program ini menggunakan struktur switch case, untuk menampilkan huruf apa yang telah dimasukan oleh user.

```
D:\PBO\TUGAS 2>javac KasusSwitch.java
```

```
D:\PBO\TUGAS 2>java KasusSwitch
```

```
Ketikkan sebuah huruf, akhiri dengan RETURN
```

```
a
```

```
Yang anda ketik adalah a
```

8. Gambar ini menunjukkan eksekusi program Java yang menghitung luas lingkaran. Program "Konstant.java" menggunakan nilai jari-jari yang

telah ditentukan (kemungkinan sebagai konstanta), dan kemudian menghitung dan menampilkan luas lingkaran. Kemungkinan besar, program "Konstant.java" memiliki kode yang mendefinisikan nilai jari-jari sebagai konstanta, dan kemudian menggunakan rumus luas lingkaran (πr^2) untuk menghitung luasnya. Secara singkat, program java ini menghitung luas lingkaran berdasarkan nilai jari-jari lingkaran.

```
D:\PBO\TUGAS 2>javac Konstant.java  
  
D:\PBO\TUGAS 2>java Konstant  
Jari-jari lingkaran =8  
Luas lingkaran = 201.056  
Akhir program
```

9. Gambar ini menunjukkan eksekusi program Java yang mencari nilai maksimum dari dua bilangan. Program "Max2.java" meminta pengguna untuk memasukkan dua bilangan, dan kemudian menampilkan nilai maksimum dari kedua bilangan tersebut. Dalam contoh ini, pengguna memasukkan bilangan 2 dan 4, dan program dengan benar mengidentifikasi 4 sebagai nilai maksimum. Kemungkinan besar, program "Max2.java" memiliki kode yang membandingkan dua bilangan dan menampilkan nilai yang lebih besar. Secara singkat, program ini membandingkan 2 nilai integer dan menampilkan nilai yang terbesar.

```
D:\PBO\TUGAS 2>javac Max2.java  
  
D:\PBO\TUGAS 2>java Max2  
Maksimum dua bilangan :  
Ketikkan dua bilangan, pisahkan dg RETURN :  
2 4  
Ke dua bilangan : a = 2 b = 4  
Nilai b yang maksimum: 4
```

10. Gambar ini menunjukkan eksekusi program Java yang mencari nilai maksimum dari dua bilangan. Program "Max2.java" meminta pengguna untuk memasukkan dua bilangan, dan kemudian menampilkan nilai maksimum dari kedua bilangan tersebut. Dalam contoh ini, pengguna memasukkan bilangan 2 dan 4, dan program dengan benar mengidentifikasi 4 sebagai nilai maksimum. Kemungkinan besar, program "Max2.java" memiliki kode yang membandingkan dua bilangan dan menampilkan nilai yang lebih besar. Secara singkat, program ini membandingkan 2 nilai integer dan menampilkan nilai yang terbesar.

```
D:\PBO\TUGAS 2>javac PriFor.java

D:\PBO\TUGAS 2>java PriFor
Baca N, print 1 s/d N N = N
Exception in thread "main" java.util.InputMismatchException
    at java.base/java.util.Scanner.throwFor(Scanner.java:964)
    at java.base/java.util.Scanner.next(Scanner.java:1619)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2284)
    at java.base/java.util.Scanner.nextInt(Scanner.java:2238)
    at PriFor.main(PriFor.java:17)
```

11. Gambar ini menunjukkan eksekusi program Java yang melakukan iterasi (perulangan) dan mencetak nilai variabel iterasi. Program "PrintIterasi.java" meminta pengguna untuk menentukan nilai N (jumlah iterasi), dan kemudian mencetak nilai variabel iterasi dari 1 hingga N. Dalam contoh ini, pengguna memasukkan nilai 5, dan program mencetak nilai 1 hingga 5. Kemungkinan besar, program "PrintIterasi.java" menggunakan loop (seperti for atau while) untuk melakukan iterasi dan mencetak nilai variabel iterasi. Secara singkat, program ini melakukan perulangan dari 1 hingga nilai N dan mencetak nilai perulangannya.

```
D:\PBO\TUGAS 2>javac PrintIterasi.java

D:\PBO\TUGAS 2>java PrintIterasi
Nilai N >0 = 5
Print i dengan ITERATE :
1
2
3
4
5
```

12. Gambar ini menunjukkan eksekusi program Java yang melakukan pengulangan (loop) dan mencetak nilai variabel pengulangan. Program "PrintRepeat.java" meminta pengguna untuk menentukan nilai N (jumlah pengulangan), dan kemudian mencetak nilai variabel pengulangan dari 1 hingga N. Dalam contoh ini, pengguna memasukkan nilai 8, dan program mencetak nilai 1 hingga 8. Kemungkinan besar, program "PrintRepeat.java" menggunakan loop (seperti for atau while) untuk melakukan pengulangan dan mencetak nilai variabel pengulangan. Secara singkat, program ini melakukan perulangan dari 1 hingga nilai N dan mencetak nilai perulangannya.

```
D:\PBO\TUGAS 2>javac PrintRepeat.java
```

```
D:\PBO\TUGAS 2>java PrintRepeat
```

```
Nilai N >0 = 8
```

```
Print i dengan REPEAT:
```

```
1  
2  
3  
4  
5  
6  
7  
8
```

13. Gambar ini menunjukkan eksekusi program Java yang melakukan pengulangan menggunakan loop while dan mencetak nilai variabel pengulangan. Program "PrintWhile.java" menggunakan nilai N (jumlah pengulangan), dan kemudian mencetak nilai variabel pengulangan dari 1 hingga N. Dalam contoh ini, pengguna memasukkan nilai 4, dan program mencetak nilai 1 hingga 4. Kemungkinan besar, program "PrintWhile.java" menggunakan loop `while` untuk melakukan pengulangan dan mencetak nilai variabel pengulangan. Secara singkat, program ini melakukan pengulangan dari 1 hingga nilai N menggunakan loop `while` dan mencetak nilai perulangannya.

```
D:\PBO\TUGAS 2>javac PrintWhile.java
```

```
D:\PBO\TUGAS 2>java PrintWhile
```

```
Nilai N >0 = 4
```

```
Print i dengan WHILE:
```

```
1  
2  
3  
4
```

14. Gambar ini menunjukkan eksekusi program Java yang melakukan pengulangan menggunakan loop while dan mencetak nilai variabel pengulangan. Program "PrintWhile1.java" menggunakan nilai N (jumlah pengulangan), dan kemudian mencetak nilai variabel pengulangan dari 1 hingga N. Dalam contoh ini, pengguna memasukkan nilai 3, dan program mencetak nilai 1 hingga 3. Kemungkinan besar, program "PrintWhile1.java" menggunakan loop `while` dengan kode yang lebih ringkas dibandingkan contoh sebelumnya, namun tetap melakukan fungsi yang sama, yaitu mencetak nilai variabel pengulangan. Secara singkat, program ini melakukan pengulangan dari 1 hingga nilai N menggunakan loop `while` dengan kode yang lebih ringkas dan mencetak nilai perulangannya.

```
D:\PBO\TUGAS 2>javac PrintWhile1.java
```

```
D:\PBO\TUGAS 2>java PrintWhile1
```

```
Nilai N >0 = 3
```

```
Print i dengan WHILE (ringkas):
```

```
1  
2  
3
```

15. Gambar ini menunjukkan eksekusi program Java yang meminta pengguna untuk memasukkan serangkaian nilai integer. Program "PrintXinterasi.java" akan terus meminta input hingga pengguna memasukkan 999. Setelah input selesai, program akan menghitung dan menampilkan hasil penjumlahan dari nilai-nilai integer yang dimasukkan. Kemungkinan besar, program "PrintXinterasi.java" menggunakan loop (seperti while) untuk meminta input dan menyimpan nilai-nilai integer, serta menggunakan variabel untuk menghitung total penjumlahan. Secara singkat, program ini menjumlahkan serangkaian nilai integer yang dimasukkan oleh pengguna hingga pengguna memasukkan nilai 999.

```
D:\PBO\TUGAS 2>javac PrintXinterasi.java
```

```
D:\PBO\TUGAS 2>java PrintXinterasi
```

```
Masukkan nilai x (int), akhiri dg 999: 10
```

```
Masukkan nilai x (int), akhiri dg 999 : 5
```

```
Masukkan nilai x (int), akhiri dg 999 : 999
```

```
Hasil penjumlahan = 15
```

16. Masukkan nilai x (int), akhiri dg 999 : 20: Ini adalah output dari program Java yang meminta pengguna untuk memasukkan nilai integer (bilangan bulat). Pengguna memasukkan nilai 20.
Masukkan nilai x (int), akhiri dg 999 : 5: Program kembali meminta input, dan pengguna memasukkan nilai 5.
Masukkan nilai x (int), akhiri dg 999 : 999: Pengguna memasukkan nilai 999. Ini mengindikasikan bahwa pengguna telah selesai memasukkan angka, karena 999 dijadikan tanda untuk mengakhiri input.
Hasil penjumlahan = 25: Program menampilkan hasil penjumlahan dari nilai-nilai yang dimasukkan sebelumnya, yaitu $20 + 5 = 25$. Gambar tersebut menunjukkan proses kompilasi dan eksekusi program Java sederhana. Program ini meminta pengguna untuk memasukkan serangkaian bilangan bulat, dan akan berhenti meminta input ketika pengguna memasukkan angka 999. Setelah itu, program menghitung dan

menampilkan jumlah dari semua bilangan bulat yang telah dimasukkan. Program tersebut kemungkinan besar adalah contoh dasar dari penggunaan perulangan (loop) dan penanganan input pengguna dalam bahasa pemrograman Java. Angka 999 digunakan sebagai "sentinel value" atau nilai penjaga untuk menandakan akhir dari input pengguna.

```
D:\PBO\TUGAS 2>javac PrintXRepeat.java

D:\PBO\TUGAS 2>java PrintXRepeat
Masukkan nilai x (int), akhiri dg 999 : 20
Masukkan nilai x (int), akhiri dg 999 : 5
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 25
```

17. Masukkan nilai x (int), akhiri dg 999 : 15: Ini adalah output dari program Java yang meminta pengguna untuk memasukkan nilai integer (bilangan bulat). Pengguna memasukkan nilai 15.
- Masukkan nilai x (int), akhiri dg 999 : 5: Program kembali meminta input, dan pengguna memasukkan nilai 5.
- Masukkan nilai x (int), akhiri dg 999 : 999: Pengguna memasukkan nilai 999. Ini mengindikasikan bahwa pengguna telah selesai memasukkan angka, karena 999 dijadikan tanda untuk mengakhiri input.
- Hasil penjumlahan = 20: Program menampilkan hasil penjumlahan dari nilai-nilai yang dimasukkan sebelumnya, yaitu $15 + 5 = 20$.
- Gambar tersebut menunjukkan proses kompilasi dan eksekusi program Java sederhana. Program ini meminta pengguna untuk memasukkan serangkaian bilangan bulat, dan akan berhenti meminta input ketika pengguna memasukkan angka 999. Setelah itu, program menghitung dan menampilkan jumlah dari semua bilangan bulat yang telah dimasukkan. Program tersebut kemungkinan besar adalah contoh dasar dari penggunaan perulangan (loop) while dan penanganan input pengguna dalam bahasa pemrograman Java. Angka 999 digunakan sebagai "sentinel value" atau nilai penjaga untuk menandakan akhir dari input pengguna.

```
D:\PBO\TUGAS 2>javac PrintXWhile.java

D:\PBO\TUGAS 2>java PrintXWhile
Masukkan nilai x (int), akhiri dg 999 : 15
Masukkan nilai x (int), akhiri dg 999 : 5
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 20
```

18. Maksimum dua bilangan: Program menampilkan pesan bahwa program ini akan menentukan nilai maksimum dari dua bilangan.

Ketikkan dua bilangan, pisahkan dg RETURN : 25 26: Program meminta pengguna untuk memasukkan dua bilangan bulat, yang dipisahkan oleh tombol Enter (RETURN). Pengguna memasukkan nilai 25 dan 26.

Ke dua bilangan : a = 25 b = 26: Program menampilkan nilai kedua bilangan yang dimasukkan, dengan variabel a bernilai 25 dan b bernilai 26.

Maksimum = 26: Program menampilkan nilai maksimum dari kedua bilangan yang dimasukkan, yaitu 26.

Tukar kedua bilangan...: Program memberikan indikasi bahwa akan terjadi pertukaran nilai dari kedua variabel.

Ke dua bilangan setelah tukar : a = 26 b = 25: Program menampilkan nilai kedua bilangan setelah ditukar, dengan variabel a bernilai 26 dan b bernilai 25.

Gambar tersebut menunjukkan proses kompilasi dan eksekusi program Java sederhana. Program ini melakukan beberapa operasi dasar pada dua bilangan bulat:

Mencari nilai maksimum: Program menentukan dan menampilkan nilai maksimum dari dua bilangan yang dimasukkan pengguna.

Menukar nilai variabel: Program menukar nilai dari kedua variabel, sehingga nilai a menjadi nilai b sebelumnya, dan sebaliknya.

```
D:\PBO\TUGAS 2>javac SubProgram.java

D:\PBO\TUGAS 2>java SubProgram
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
25 26
Ke dua bilangan : a =25 b = 26
Maksimum = 26
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 26 b = 25
```

19. D:\PBO\TUGAS 2>java Tempair: Perintah ini menjalankan program Java yang telah dikompilasi sebelumnya. Perintah `java` diikuti dengan nama kelas yang akan dijalankan, dalam hal ini "Tempair".if kasus Program menampilkan pesan bahwa ini adalah contoh penggunaan struktur if dengan tiga kasus (kemungkinan kondisi).Temperatur (der.c)=5: Program menampilkan pesan bahwa temperatur adalah 5 derajat Celcius.Wujud air Program menampilkan output bahwa wujud air pada temperatur 5 derajat Celcius adalah cair.5: Ada angka 5 yang ditampilkan lagi di baris berikutnya, kemungkinan ini adalah pengulangan dari nilai temperatur yang dimasukkan.

```
D:\PBO\TUGAS 2>javac Tempair.java

D:\PBO\TUGAS 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 5
Wujud air cair
5
D:\PBO\TUGAS 2>
```