

Nama : Ikram
Nim : 13020220053
Kelas : A2
Matkul : Pemrograman Berorientasi Objek

Tugas 3

Output Kode Program 1

```
D:\SEMESTER 4\TUGAS PBO\tugas 3>javac BacaString.java
D:\SEMESTER 4\TUGAS PBO\tugas 3>java BacaString

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

Penjelasan Mengenai Program :

1. Import paket-paket yang diperlukan: `java.io.BufferedReader`, `java.io.IOException`, `java.io.InputStreamReader`, `javax.swing.JOptionPane`.
2. Deklarasi kelas `BacaString` dengan metode `main`.
3. Membuat objek `BufferedReader` untuk membaca input dari `System.in`.
4. Menampilkan pesan untuk meminta pengguna memasukkan sebuah string menggunakan `System.out.print`.
5. Membaca baris string yang dimasukkan oleh pengguna dari konsol menggunakan `dataIn.readLine()` dan menyimpannya dalam variabel `str`.
6. Menampilkan kembali string yang dimasukkan oleh pengguna menggunakan `System.out.print`

Output Kode Program 2

```
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 ....
```

Penjelasan Mengenai Program :

1. Deklarasi kelas **Forever** yang berisi metode **main**.
2. Di dalam metode **main**, program mencetak pesan "Program akan looping, akhiri dengan ^c" menggunakan **System.out.println**.
3. Program memasuki loop **while (true)** yang akan terus berjalan selama kondisinya adalah **true**. Ini berarti bahwa loop ini akan terus berjalan tanpa henti.
4. Di dalam loop, program mencetak pesan "Print satu baris " menggunakan **System.out.print**.

4. Karena tidak ada kode di dalam loop yang mengubah nilai kondisi **true**, maka loop akan berjalan tanpa henti, mencetak pesan yang sama berulang kali ke konsol.
5. Program akan terus berjalan sampai dihentikan secara paksa oleh pengguna dengan menekan
6. kombinasi tombol **^c** (Ctrl + C) di terminal (di lingkungan UNIX/Linux) atau dengan menekan tombol "Terminate" di IDE atau lingkungan pengembangan yang digunakan.

Output Kode Program 3

```
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>javac If1.java  
  
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>java If1  
Contoh IF satu kasus  
Ketikkan suatu nilai integer : 4  
  
Nilai a positif 4
```

Penjelasan Mengenai Program :

1. Import paket **java.util.Scanner** yang diperlukan untuk membaca input dari pengguna.
2. Deklarasi kelas **If1** yang berisi metode **main**.
3. Di dalam metode **main**, sebuah objek **Scanner** bernama **masukan** dibuat untuk membaca input dari **System.in**.
4. Sebuah variabel **a** dengan tipe data integer dideklarasikan untuk menyimpan nilai yang dimasukkan oleh pengguna.
5. Program mencetak pesan "Contoh IF satu kasus" dan "Ketikkan suatu nilai integer : " menggunakan **System.out.print**.
6. Program menggunakan **masukan.nextInt()** untuk membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya ke dalam variabel **a**.
7. Program memeriksa apakah nilai **a** lebih besar dari atau sama dengan 0 dengan menggunakan pernyataan kondisional **if (a >= 0)**.
8. Jika kondisi tersebut benar (nilai **a** positif atau nol), program mencetak pesan "Nilai a positif" diikuti dengan nilai **a** menggunakan **System.out.print**.
9. Jika kondisi tersebut salah (nilai **a** negatif), tidak ada yang dicetak

Output Kode Program 4

```
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>javac If2.java  
  
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>java If2  
Contoh IF dua kasus  
Ketikkan suatu nilai integer :12  
  
Nilai a positif 12
```

Penjelasan Mengenai Program :

1. Deklarasi kelas **If2** yang berisi metode **main**.

2. Di dalam metode **main**, sebuah objek **Scanner** bernama **masukan** dibuat untuk membaca input dari **System.in**.
3. Sebuah variabel **a** dengan tipe data integer dideklarasikan untuk menyimpan nilai yang dimasukkan oleh pengguna.
4. Program mencetak pesan "Contoh IF dua kasus" dan "Ketikkan suatu nilai integer :" menggunakan **System.out.print**.
5. Program menggunakan **masukan.nextInt()** untuk membaca nilai integer yang dimasukkan oleh pengguna dan menyimpannya ke dalam variabel **a**.
6. Program memeriksa apakah nilai **a** lebih besar dari atau sama dengan 0 dengan menggunakan pernyataan kondisional **if (a >= 0)**.
7. Jika kondisi tersebut benar (nilai **a** positif atau nol), program mencetak pesan "Nilai a positif" diikuti dengan nilai **a** menggunakan **System.out.println**.
8. Jika kondisi tersebut salah (nilai **a** negatif), program menggunakan blok **else** dan mencetak pesan "Nilai a negatif" diikuti dengan nilai **a** menggunakan **System.out.println**.

Output Kode Program 5

```
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>javac If3.java
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer :18 5 1
Nilai a positif 18
```

Penjelasan Mengenai Program :

1. Deklarasi kelas **If3** dengan metode **main**.
2. Membuat objek **Scanner** untuk membaca input dari pengguna.
3. Mendeklarasikan variabel **a** untuk menyimpan nilai yang dimasukkan oleh pengguna.
4. Mencetak pesan permintaan input dan membaca nilai integer dari pengguna.
5. Memeriksa tiga kasus nilai **a** menggunakan pernyataan kondisional **if-else if-else**.
6. Menampilkan hasil sesuai dengan kasus nilai **a**.

Output Kode Program 6

```
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>javac KasusBoolean.java
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>java KasusBoolean
true
benar
```

Penjelasan Mengenai Program :

1. Pertama, di dalam metode **main**, sebuah variabel **bool** dengan tipe data boolean diperkenalkan untuk menyimpan nilai kebenaran.
2. Variabel **bool** tersebut kemudian diinisialisasi dengan nilai **true**.
3. Program menggunakan pernyataan **if** untuk memeriksa nilai dari variabel **bool**. Jika nilai **bool** adalah **true**, program akan mencetak "**true**" menggunakan **System.out.print**.

4. Namun, jika nilai bool adalah false, pernyataan di dalam blok else akan dijalankan, dan program akan mencetak "false".
5. Selanjutnya, program menggunakan pernyataan if dengan operator logika negasi ! untuk memeriksa kebalikan dari nilai variabel bool. Jika nilai bool adalah false, maka pernyataan di dalam blok if akan dijalankan, dan program mencetak "salah".
6. Di sisi lain, jika nilai bool adalah true, maka pernyataan di dalam blok else akan dieksekusi, dan program mencetak "benar".

Output Kode Program 7

```
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>javac KasusSwitch.java

D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
A
Yang anda ketik adalah huruf mati
```

Penjelasan Mengenai Program :

1. Deklarasi kelas **KasusSwitch** dengan metode **main**.
2. Membuat objek **Scanner** untuk membaca input dari pengguna.
3. Mendeklarasikan variabel **cc** untuk menyimpan karakter yang dimasukkan oleh pengguna.
4. Mencetak pesan permintaan input karakter.
5. Membaca satu karakter dari pengguna dan menyimpannya dalam variabel **cc**.
6. Menggunakan pernyataan **switch** untuk memeriksa nilai karakter yang dimasukkan oleh pengguna.
7. 7. Setiap **case** pada **switch** mencocokkan karakter yang dimasukkan oleh pengguna dengan karakter tertentu, dan mencetak pesan sesuai.
8. Jika tidak ada kasus yang cocok, akan dieksekusi bagian **default**, mencetak pesan bahwa karakter yang dimasukkan adalah "huruf mati".

Output Kode Program 8

```
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>javac Konstant.java

D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>java Konstant
Jari-jari lingkaran =12
Luas lingkaran = 452.37598
Akhir program
```

Penjelasan Mengenai Program :

1. Deklarasi kelas **Konstant** yang berisi metode **main**.
2. Di dalam metode **main**, sebuah konstanta **PHI** dideklarasikan dengan nilai 3.1415 menggunakan kata kunci **final**. Konstanta ini akan digunakan untuk menyimpan nilai Phi (π).
3. Sebuah variabel **r** dengan tipe data **float** dideklarasikan untuk menyimpan nilai jari-jari lingkaran yang dimasukkan oleh pengguna.

4. Program membuat objek **Scanner** bernama **masukan** untuk membaca input dari **System.in**.
5. Program mencetak pesan "Jari-jari lingkaran =" menggunakan **System.out.print**.
6. Program menggunakan **masukan.nextFloat()** untuk membaca nilai jari-jari lingkaran yang dimasukkan oleh pengguna dan menyimpannya dalam variabel **r**.
7. Program menghitung luas lingkaran dengan rumus $\pi * r * r$, dan mencetak hasilnya menggunakan **System.out.print**.
8. Program mencetak pesan "Akhir program" menggunakan **System.out.print**.

Output Kode Program 9

```
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>javac Max2.java
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
1
9
Ke dua bilangan : a = 1 b = 9
Nilai b yang maksimum: 9
```

Penjelasan Mengenai Program :

1. Deklarasi kelas **Max2** dengan metode **main**.
2. Membuat dua variabel **a** dan **b** untuk menyimpan dua bilangan yang akan dibaca.
3. Membuat objek **Scanner** untuk membaca input dari pengguna.
4. Mencetak pesan permintaan untuk memasukkan dua bilangan.
5. Membaca dua bilangan yang dimasukkan oleh pengguna.
6. Menampilkan kembali kedua bilangan yang dimasukkan.
7. Menggunakan pernyataan **if** untuk memeriksa bilangan mana yang lebih besar.
8. Menampilkan bilangan yang lebih besar.

Output Kode Program 10

```
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>javac PriFor.java
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>java PriFor
Baca N, print 1 s/d N N = 1
1
Akhir program
```

Penjelasan Mengenai Program :

1. Program mendeklarasikan kelas "PriFor" dengan metode "main".
2. Program membuat objek Scanner untuk membaca input.
3. Pengguna diminta untuk memasukkan nilai N.
4. Nilai N dibaca dan disimpan.
5. Program menggunakan perulangan for untuk mencetak angka dari 1 hingga N.
6. Setelah mencetak angka, program mencetak pesan "Akhir program".

Output Kode Program 11

```
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>javac PrintIterasi.java

D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>java PrintIterasi
Nilai N >0 = 10
Print i dengan ITERATE :
1
2
3
4
5
6
7
8
9
10
```

Penjelasan Mengenai Program :

1. Deklarasi kelas **PrintIterasi** dengan metode **main**.
2. Membuat dua variabel **N** dan **i** untuk menyimpan nilai yang akan dibaca dan nilai iterasi.
3. Membuat objek **Scanner** untuk membaca input dari pengguna.
4. Mencetak pesan permintaan untuk memasukkan nilai **N**.
5. Membaca nilai **N** yang dimasukkan oleh pengguna.
6. Menginisialisasi nilai iterasi **i** dengan 1.
7. Mencetak pesan sebelum melakukan iterasi.
8. Menggunakan loop **for** tanpa kondisi awal, kondisi, atau pernyataan iterasi.
9. Di dalam loop, mencetak nilai **i**.
10. Menggunakan **if** untuk memeriksa apakah nilai **i** sudah sama dengan nilai **N**.
11. Jika sudah sama, menggunakan **break** untuk keluar dari loop.
12. Jika belum sama, menambahkan nilai **i** dengan 1 untuk mencetak nilai berikutnya.

Output Kode Program 12

```
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>javac PrintRepeat.java

D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>java PrintRepeat
Nilai N >0 = 5
Print i dengan REPEAT:
1
2
3
4
5
```

Penjelasan Mengenai Program :

1. Deklarasi kelas **PrintRepeat** dengan metode **main**.
2. Membuat dua variabel **N** dan **i** untuk menyimpan nilai yang akan dibaca dan nilai iterasi.
3. Membuat objek **Scanner** untuk membaca input dari pengguna.
4. Mencetak pesan permintaan untuk memasukkan nilai **N**.

5. Membaca nilai N yang dimasukkan oleh pengguna.
6. Menginisialisasi nilai iterasi **i** dengan 1.
7. Mencetak pesan sebelum melakukan iterasi.
8. Menggunakan loop **do-while** untuk mencetak nilai **i** dari 1 hingga N.
9. Di dalam loop, mencetak nilai **i** dan menambahkan **i** dengan 1.
10. Kondisi untuk menjalankan loop adalah selama **i** kurang dari atau sama dengan N.

Output Kode Program 13

```
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>javac PrintWhile.java
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>java PrintWhile
Nilai N >0 = 11
Print i dengan WHILE:
1
2
3
4
5
6
7
8
9
10
11
```

Penjelasan Mengenai Program :

1. Program meminta pengguna untuk memasukkan nilai N.
2. Nilai N tersebut kemudian dibaca dan disimpan.
3. Variabel iterasi "i" diinisialisasi dengan nilai 1 sebagai elemen pertama.
4. Program mencetak pesan "Print i dengan WHILE:".
5. Selama nilai "i" kurang dari atau sama dengan nilai "N", program akan melakukan proses pencetakan nilai "i" ke layar.
6. Setiap iterasi, nilai "i" akan bertambah satu untuk mencetak nilai berikutnya.
7. Perulangan akan berhenti ketika nilai "i" melebihi nilai "N".
8. Setelah selesai mencetak nilai, program selesai dieksekusi.

Output Kode Program 14

```
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>javac PrintWhile1.java
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>java PrintWhile1
Nilai N >0 = 4
Print i dengan WHILE (ringkas):
1
2
3
4
```

Penjelasan Mengenai Program :

1. Deklarasi kelas **PrintWhile1** dengan metode **main**.
2. Membuat dua variabel N dan **i** untuk menyimpan nilai yang akan dibaca dan nilai iterasi. Variabel **i** diinisialisasi dengan nilai 1.
3. Membuat objek **Scanner** untuk membaca input dari pengguna.

4. Mencetak pesan permintaan untuk memasukkan nilai N.
5. Membaca nilai N yang dimasukkan oleh pengguna.
6. Mencetak pesan sebelum melakukan iterasi.
7. Menggunakan loop **while** untuk mencetak nilai **i** dari 1 hingga N dengan operator penambahan ++ untuk nilai **i**.
8. Kondisi untuk menjalankan loop adalah selama **i** kurang dari atau sama dengan N.

Output Kode Program 15

```
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>javac PrintXinterasi.java

D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>java PrintXinterasi
Masukkan nilai x (int), akhiri dg 999 : 2
Masukkan nilai x (int), akhiri dg 999 : 6
Masukkan nilai x (int), akhiri dg 999 : 4
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 12
```

Penjelasan Mengenai Program :

1. Deklarasi kelas **PrintXinterasi** dengan metode **main**.
2. Membuat dua variabel **Sum** dan **x** untuk menyimpan total nilai yang dibaca dan nilai yang dibaca saat ini.
3. Membuat objek **Scanner** untuk membaca input dari pengguna.
4. Mencetak pesan permintaan untuk memasukkan nilai **x**.
5. Membaca nilai **x** yang dimasukkan oleh pengguna.
6. Jika nilai **x** sama dengan 999, mencetak pesan "Kasus kosong".
7. Jika tidak, inisialisasi total **Sum** dengan nilai **x**.
8. Menggunakan loop **for** tanpa kondisi awal, kondisi, atau pernyataan iterasi.
9. Di dalam loop, mencetak pesan untuk memasukkan nilai **x** lagi.
10. Membaca nilai **x** berikutnya.
11. Jika nilai **x** sama dengan 999, keluar dari loop.
12. Jika tidak, menambahkan nilai **x** ke total **Sum**.
13. Setelah loop selesai, mencetak total **Sum**.
14. Program selesai.

Output Kode Program 16

```
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>javac PrintXRepeat.java

D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>java PrintXRepeat
Masukkan nilai x (int), akhiri dg 999 : 8
Masukkan nilai x (int), akhiri dg 999 : 12
Masukkan nilai x (int), akhiri dg 999 : 6
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 26
```

Penjelasan Mengenai Program :

1. Program meminta pengguna untuk memasukkan nilai **x** (integer).
2. Nilai **x** yang dimasukkan oleh pengguna kemudian dibaca menggunakan objek **Scanner** dan disimpan dalam variabel **x**.

3. Program melakukan pemeriksaan kondisi apakah nilai **x** yang dimasukkan oleh pengguna adalah 999. Jika iya, program akan mencetak pesan "Kasus kosong".
4. Jika nilai **x** bukan 999, maka program akan menjalankan perulangan menggunakan do-while loop untuk memproses nilai **x**.
5. Di dalam perulangan, nilai **x** yang dimasukkan oleh pengguna akan ditambahkan ke dalam variabel **Sum** untuk dijumlahkan.
6. Proses 2-5 akan terus berulang sampai pengguna memasukkan nilai 999 sebagai penanda akhir input.
7. Setelah itu, program akan mencetak hasil penjumlahan dari semua nilai **x** yang dimasukkan oleh pengguna dengan pesan "Hasil penjumlahan = " diikuti dengan nilai **Sum**.

Output Kode Program 17

```
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>javac PrintXWhile.java

D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>java PrintXWhile
Masukkan nilai x (int), akhiri dg 999 : 2
Masukkan nilai x (int), akhiri dg 999 : 7
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 9
```

Penjelasan Mengenai Program :

1. Program membaca nilai-nilai dari pengguna dan menjumlahkannya menggunakan loop **while**.
2. Variabel **Sum** dan **x** digunakan untuk menyimpan total jumlah dan nilai yang dibaca.
3. Objek **Scanner** digunakan untuk membaca input dari pengguna.
4. Program mencetak pesan permintaan untuk memasukkan nilai **x**.
5. Loop **while** digunakan untuk membaca nilai **x** dan menambahkannya ke total **Sum** sampai pengguna memasukkan nilai 999.
6. Setelah loop selesai, program mencetak total jumlah nilai yang dimasukkan.

Output Kode Program 18

```
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>javac SubProgram.java

D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>java SubProgram
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
8
12
Ke dua bilangan : a = 8 b = 12
Maksimum = 12
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 12 b = 8
```

Penjelasan Mengenai Program :

1. Program memiliki dua fungsi: **maxab** untuk mencari nilai maksimum dari dua bilangan, dan **tukar** untuk menukar nilai dua bilangan.
2. Di dalam metode **main**, program membaca dua bilangan dari pengguna.
3. Program mencetak kedua bilangan yang dibaca.

4. Program mencetak nilai maksimum dari dua bilangan yang dibaca menggunakan fungsi **maxab**.
5. Program mencetak pesan dan menukar nilai kedua bilangan menggunakan prosedur **tukar**.
6. Nilai-nilai asli tidak berubah karena Java meneruskan argumen dengan nilainya sendiri ke dalam fungsi/prosedur.

Output Kode Program 19

```
D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>javac Tempair.java

D:\SEMESTER 4\PBO\Tugas 3\Tugas3_SourceCode>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 19
Wujud air cair
19
```

Penjelasan Mengenai Program :

1. Program mendeklarasikan kelas **Tempair** yang berisi metode **main**.
2. Di dalam metode **main**, program mendeklarasikan variabel **T** untuk menyimpan nilai suhu yang dimasukkan oleh pengguna.
3. Objek **Scanner** bernama **masukan** dibuat untuk membaca input dari **System.in**.
4. Program mencetak pesan permintaan untuk memasukkan suhu.
5. Program menggunakan **masukan.nextInt()** untuk membaca nilai suhu yang dimasukkan oleh pengguna dari konsol dan menyimpannya dalam variabel **T**.
6. Program menggunakan struktur kondisional **if** dengan tiga kasus untuk menentukan wujud air berdasarkan nilai suhu **T**