

Kelas : A2

### Output Code program 1:

```
D:\SEMESTER 4\PBO\Tugas3>java No1

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

- **Penjelasan:**

Program di atas adalah contoh sederhana untuk membaca sebuah string dari input pengguna menggunakan kelas `BufferedReader` dalam Java. Program ini meminta pengguna untuk memasukkan sebuah string melalui konsol, kemudian menggunakan `BufferedReader` untuk membaca string tersebut. Setelah itu, string yang dibaca akan dicetak kembali ke konsol sebagai output. Program ini menunjukkan cara menggunakan `BufferedReader` untuk membaca input string dari pengguna melalui konsol dalam bahasa pemrograman Java.

### Output Kode program 2:

[illegible]

- **Penjelasan:**

Program "ForEver" adalah program Java yang menggunakan struktur perulangan tak terbatas (infinite loop) untuk terus mencetak satu baris teks ke konsol tanpa henti. Setelah mencetak satu baris teks, program akan kembali ke awal loop dan mencetak lagi. Program ini akan terus berjalan hingga dihentikan secara paksa oleh pengguna dengan menekan kombinasi tombol "Ctrl + c". Ini adalah contoh sederhana dari bagaimana menggunakan loop tak terbatas dalam pemrograman Java, dan digunakan di sini untuk tujuan demonstrasi atau debugging.

#### Output Kode program 3:

```
D:\SEMESTER 4\PBO\Tugas3>java No3
Contoh IF satu kasus
Ketikkan suatu nilai integer : 4

Nilai a positif 4
D:\SEMESTER 4\PBO\Tugas3>
```

- **Penjelasan:**

Program di atas adalah contoh penggunaan struktur kontrol if dalam bahasa pemrograman Java. Program ini meminta pengguna untuk memasukkan suatu nilai integer melalui input dari keyboard menggunakan kelas Scanner. Setelah itu, program akan memeriksa apakah nilai yang dimasukkan adalah positif atau tidak. Jika nilai tersebut positif (atau nol), maka program akan mencetak pesan yang menyatakan bahwa nilai tersebut positif bersama dengan nilainya. Jika nilai tersebut negatif, program tidak melakukan apa-apa. Hal ini mengilustrasikan bagaimana penggunaan if dapat digunakan untuk membuat keputusan berdasarkan kondisi tertentu dalam program Java.

#### Output Kode Program

```
D:\SEMESTER 4\PBO\Tugas3>java No4
Contoh IF dua kasus
Ketikkan suatu nilai integer :2
Nilai a positif 2

D:\SEMESTER 4\PBO\Tugas3>
```

- **Penjelasan:**

Program di atas adalah contoh penggunaan struktur kontrol IF dua kasus komplementer dalam bahasa pemrograman Java. Program ini meminta pengguna untuk memasukkan sebuah nilai integer dan kemudian memeriksa apakah nilai tersebut positif atau negatif. Jika nilai tersebut lebih besar atau sama dengan nol, program akan mencetak pesan "Nilai a positif" diikuti dengan nilai yang dimasukkan pengguna. Namun, jika nilai tersebut kurang dari nol, program akan mencetak pesan "Nilai a negatif" diikuti dengan nilai yang dimasukkan pengguna. Ini mengilustrasikan cara penggunaan struktur IF dua kasus untuk menangani dua kondisi yang saling berlawanan.

#### Output kode program 5:

```
D:\SEMESTER 4\PBO\Tugas3>java No5
Contoh IF tiga kasus
Ketikkan suatu nilai integer :3
Nilai a positif 3
```

- **Penjelasan:**

Program di atas merupakan contoh implementasi struktur kontrol if-else untuk menentukan tiga kasus berbeda berdasarkan nilai yang dimasukkan pengguna. Pertama, program meminta pengguna untuk memasukkan sebuah nilai integer. Selanjutnya, nilai tersebut diperiksa: jika nilai tersebut lebih besar dari 0, program akan mencetak bahwa nilai tersebut positif bersama dengan nilainya; jika nilai tersebut sama dengan 0, program akan mencetak bahwa nilainya nol; dan jika nilai tersebut kurang dari 0, program akan mencetak bahwa nilainya negatif bersama dengan nilainya. Dengan ini, program memberikan respons yang sesuai terhadap tiga kasus yang mungkin terjadi tergantung pada nilai yang dimasukkan pengguna.

**Output Kode Program 6:**

```
D:\SEMESTER 4\PBO\Tugas3>java No6
true
benar
```

- **Penjelasan:**

Program di atas adalah contoh implementasi ekspresi kondisional dengan tipe data boolean dalam bahasa pemrograman Java. Pada awalnya, sebuah variabel boolean bernama `bool` diinisialisasi dengan nilai `true`. Kemudian, program menggunakan struktur kontrol if untuk memeriksa nilai dari `bool`. Jika nilai `bool` adalah `true`, maka program akan mencetak "true". Jika nilai `bool` adalah `false`, program akan mencetak "false". Selanjutnya, program menggunakan operator negasi (`!`) untuk membalikkan nilai dari `bool`, dan mencetak pesan yang sesuai tergantung pada nilai tersebut. Dengan demikian, program menunjukkan cara penggunaan ekspresi kondisional dengan tipe data boolean untuk mengatur aliran program berdasarkan nilai boolean yang ditentukan.

**Output Kode program 7:**

```
D:\SEMESTER 4\PBO\Tugas3>java No7
Ketikkan sebuah huruf, akhiri dengan RETURN
i
Yang anda ketik adalah 'i'
```

- **Penjelasan:**

Program di atas merupakan contoh implementasi struktur kontrol switch-case dalam bahasa pemrograman Java. Program ini meminta pengguna untuk memasukkan sebuah huruf dan kemudian memeriksa huruf tersebut menggunakan struktur switch-case. Bergantung pada nilai yang dimasukkan pengguna, program akan mencetak pesan yang sesuai dengan huruf tersebut. Jika huruf yang dimasukkan adalah 'a', 'u', 'e', 'i', atau 'o', maka program akan mencetak pesan yang menyatakan huruf tersebut. Jika huruf yang dimasukkan bukan merupakan salah satu dari huruf tersebut, program akan mencetak pesan yang menyatakan bahwa huruf tersebut adalah "huruf mati". Dengan demikian, program memberikan respons yang berbeda tergantung pada huruf yang dimasukkan pengguna menggunakan struktur kontrol switch-case.

#### Output Kode Program 8:

```
D:\SEMESTER 4\PBO\Tugas3>java No8
Jari-jari lingkaran =4
Luas lingkaran = 50.264
Akhir program
```

- **Penjelasan:**

Program di atas merupakan contoh penggunaan konstanta dalam perhitungan luas lingkaran. Konstanta PHI didefinisikan dengan nilai 3.1415f sebagai pendekatan dari nilai  $\pi$  (pi). Program ini meminta pengguna untuk memasukkan nilai jari-jari lingkaran dan kemudian menghitung luas lingkaran berdasarkan nilai jari-jari yang dimasukkan. Setelah itu, program mencetak hasil perhitungan luas lingkaran dan pesan "Akhir program". Dengan menggunakan konstanta PHI, program memungkinkan perubahan nilai  $\pi$  dengan mudah jika diperlukan dalam perhitungan luas lingkaran, sehingga memudahkan pemeliharaan dan pembacaan kode.

#### Output Kode Program 9:

```
D:\SEMESTER 4\PBO\Tugas3>java No9
Maksimum dua bilangan:
Ketikkan dua bilangan, pisahkan dg RETURN:
23
12
Kedua bilangan: a = 23, b = 12
Nilai a yang maksimum: 23
```

- **Penjelasan:**

Program di atas adalah contoh implementasi untuk menentukan bilangan maksimum dari dua bilangan yang dimasukkan oleh pengguna. Program ini meminta pengguna untuk memasukkan dua bilangan, yang dipisahkan oleh tanda koma, dan kemudian membandingkan kedua bilangan tersebut. Setelah itu, program mencetak bilangan mana yang lebih besar sebagai bilangan maksimum. Jika bilangan pertama (`a`) lebih besar atau sama dengan bilangan kedua (`b`), maka bilangan pertama (`a`) akan dicetak sebagai bilangan maksimum. Jika tidak, bilangan kedua (`b`) akan dicetak sebagai bilangan maksimum. Dengan demikian, program memberikan respons yang sesuai tergantung pada nilai yang dimasukkan pengguna.

#### Output Kode Program 10:

```
D:\SEMESTER 4\PBO\Tugas3>java No10
Baca N, print 1 s/d N N = 4
1
2
3
4
Akhir program
```

- **Penjelasan:**

Program di atas adalah contoh implementasi penggunaan perulangan for dalam Java untuk mencetak angka dari 1 hingga N, dimana N merupakan angka yang dimasukkan oleh pengguna. Program ini meminta pengguna untuk memasukkan nilai N, kemudian menggunakan perulangan for untuk mencetak angka dari 1 hingga N secara berurutan. Setelah selesai mencetak, program mencetak pesan "Akhir program" untuk menandakan bahwa program telah selesai dijalankan. Dengan demikian, program memberikan respons yang sesuai tergantung pada nilai N yang dimasukkan pengguna, dengan mencetak angka dari 1 hingga N.

**Output Kode Program 11:**

```
D:\SEMESTER 4\PBO\Tugas3>java No11
Nilai N >0 = 3
Print i dengan ITERATE :
1
2
3
```

- **Penjelasan:**

Program di atas adalah contoh implementasi penggunaan iterasi menggunakan perulangan tanpa batas (infinite loop) dengan penggunaan kondisi berhenti menggunakan pernyataan `break`. Program ini meminta pengguna untuk memasukkan nilai N, kemudian menggunakan perulangan `for` tanpa inisialisasi, kondisi, atau perubahan langkah iterasi (iterasi dilakukan manual). Program akan terus mencetak nilai `i` yang dimulai dari 1 hingga nilai N yang dimasukkan pengguna. Pada setiap iterasi, program memeriksa apakah nilai `i` sudah sama dengan N. Jika i sudah sama dengan N, maka pernyataan `break` akan menjalankan perulangan, dan program akan keluar dari loop. Jika belum, nilai `i` akan ditambah 1 untuk melanjutkan iterasi berikutnya. Dengan demikian, program akan mencetak angka dari 1 hingga N, dan kemudian berhenti.

**Output Kode Program 12:**

```
D:\SEMESTER 4\PBO\Tugas3>java No12
Nilai N >0 = 2
Print i dengan REPEAT:
1
2
```

- **Penjelasan:**

Program di atas adalah contoh implementasi penggunaan perulangan do-while dalam bahasa pemrograman Java untuk mencetak angka dari 1 hingga N. Program ini meminta pengguna untuk memasukkan nilai N, kemudian menggunakan perulangan do-while untuk mencetak nilai i yang dimulai dari 1 hingga N. Setiap kali perulangan dilakukan, nilai i akan dicetak, kemudian nilai i akan ditambah 1 untuk iterasi berikutnya. Perulangan akan terus dilakukan selama nilai i

masih kurang dari atau sama dengan N. Dengan demikian, program akan mencetak angka dari 1 hingga N menggunakan perulangan do-while.

#### Output Kode Program 13:

```
D:\SEMESTER 4\PBO\Tugas3>java No13
Nilai N >0 = 3
Print i dengan WHILE:
1
2
3
```

- **Penjelasan:**

Program di atas adalah contoh implementasi penggunaan perulangan while dalam bahasa pemrograman Java untuk mencetak angka dari 1 hingga N. Program ini meminta pengguna untuk memasukkan nilai N, kemudian menggunakan perulangan while untuk mencetak nilai i yang dimulai dari 1 hingga N. Setiap kali perulangan dilakukan, nilai i akan dicetak, kemudian nilai i akan ditambah 1 untuk iterasi berikutnya. Perulangan akan terus dilakukan selama nilai i masih kurang dari atau sama dengan N. Dengan demikian, program akan mencetak angka dari 1 hingga N menggunakan perulangan while.

#### Output Kode Program 14:

```
D:\SEMESTER 4\PBO\Tugas3>java No14
Nilai N >0 = 3
Print i dengan WHILE (ringkas):
1
2
3
```

- **Penjelasan:**

Program di atas adalah implementasi ringkas dari penggunaan perulangan while dalam bahasa pemrograman Java untuk mencetak angka dari 1 hingga N. Program ini meminta pengguna untuk memasukkan nilai N, kemudian menggunakan perulangan while untuk mencetak nilai i yang dimulai dari 1 hingga N. Pada setiap iterasi, nilai i akan dicetak dan kemudian ditambah 1 menggunakan operator postfix increment (`i++`). Hal ini membuat program dapat mencetak nilai i sebelum penambahan, sehingga nilai i yang dicetak pada setiap iterasi adalah nilai i sebelum penambahan. Perulangan akan terus dilakukan selama nilai i masih kurang dari atau sama dengan N. Dengan demikian, program akan mencetak angka dari 1 hingga N menggunakan perulangan while dalam format yang lebih ringkas.

#### Output Kode Program 15:

```
D:\SEMESTER 4\PBO\Tugas3>java No15
Masukkan nilai x (int), akhiri dg 999: 4
Masukkan nilai x (int), akhiri dg 999: 5
Masukkan nilai x (int), akhiri dg 999: 1
Masukkan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 10
```

- **Penjelasan:**

Program di atas merupakan contoh implementasi dari penggunaan perulangan while dalam bahasa pemrograman Java untuk menghitung jumlah sejumlah bilangan yang dimasukkan oleh pengguna, dengan menghentikan proses input ketika pengguna memasukkan nilai 999. Program ini meminta pengguna untuk memasukkan nilai bilangan bulat x, kemudian menambahkannya ke dalam variabel sum. Proses ini terus berulang menggunakan perulangan while hingga pengguna memasukkan nilai 999. Pada setiap iterasi, program memeriksa apakah nilai yang dimasukkan pengguna adalah 999. Jika iya, program akan keluar dari perulangan. Jika tidak, nilai tersebut akan ditambahkan ke dalam variabel sum. Setelah perulangan selesai, program akan mencetak hasil penjumlahan dari semua bilangan yang dimasukkan pengguna kecuali nilai 999.

#### Output Kode Program 16:

```
D:\SEMESTER 4\PBO\Tugas3>java No16
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
```

- **Penjelasan:**

Program di atas merupakan contoh implementasi dari penggunaan perulangan do-while dalam bahasa pemrograman Java untuk menghitung jumlah sejumlah bilangan yang dimasukkan oleh pengguna, dengan menghentikan proses input ketika pengguna memasukkan nilai 999. Program ini meminta pengguna untuk memasukkan nilai bilangan bulat x, kemudian menambahkannya ke dalam variabel sum. Proses ini terus berulang menggunakan perulangan do-while hingga pengguna memasukkan nilai 999. Pada setiap iterasi, nilai tersebut akan ditambahkan ke dalam variabel sum. Setelah pengguna memasukkan nilai 999, program akan mencetak hasil penjumlahan dari semua bilangan yang dimasukkan pengguna kecuali nilai 999.

#### Output Kode Program 17:

```
D:\SEMESTER 4\PBO\Tugas3>java No17
Masukkan nilai x (int), akhiri dg 999 : 2
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 = 5
```

- **Penjelasan:**

Program di atas merupakan contoh implementasi dari penggunaan perulangan while dalam bahasa pemrograman Java untuk menghitung jumlah sejumlah bilangan yang dimasukkan oleh pengguna, dengan menghentikan proses input ketika pengguna memasukkan nilai 999. Program ini meminta pengguna untuk memasukkan nilai bilangan bulat x, kemudian menambahkannya ke dalam variabel sum. Proses ini terus berulang menggunakan perulangan while hingga pengguna memasukkan nilai 999. Pada setiap iterasi, nilai tersebut akan ditambahkan ke dalam variabel sum. Setelah pengguna memasukkan nilai 999, program akan mencetak hasil penjumlahan dari semua bilangan yang dimasukkan pengguna kecuali nilai 999.

#### Output Kode Program 18:

```
D:\SEMESTER 4\PBO\Tugas3>java No18
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dengan RETURN:
3
5
Dua bilangan: a = 3 b = 5
Maksimum = 5
Tukar kedua bilangan...
Kedua bilangan setelah tukar: a = 5 b = 3
```

- **Penjelasan:**

Program di atas adalah contoh implementasi fungsi-fungsi dalam bahasa pemrograman Java. Program ini memiliki dua fungsi, yaitu `maxab` untuk mencari nilai maksimum dari dua bilangan bulat dan `tukar` untuk menukar nilai dua bilangan bulat. Fungsi `maxab` mengembalikan nilai maksimum dari dua bilangan yang diberikan sebagai argumen. Fungsi ini menggunakan operator ternary untuk membandingkan dua bilangan dan mengembalikan nilai yang lebih besar di antara keduanya. Fungsi `tukar` digunakan untuk menukar nilai dua bilangan bulat yang diberikan sebagai argumen. Di dalamnya, nilai bilangan pertama disimpan dalam variabel temporer, kemudian nilai bilangan kedua disalin ke bilangan pertama, dan nilai temporer disalin ke bilangan kedua. Dalam fungsi `main`, program meminta pengguna untuk memasukkan dua bilangan bulat, kemudian mencetak bilangan tersebut, nilai maksimum dari dua bilangan, dan hasil pertukaran kedua bilangan menggunakan fungsi-fungsi yang telah didefinisikan.

#### Output Kode Program 19:

```
D:\SEMESTER 4\PBO\Tugas3>java No19
Contoh IF tiga kasus
Temperatur (der. C) = 9
Wujud air cair 9
```

- **Penjelasan:**

Program di atas merupakan contoh implementasi struktur kontrol if-else if-else dalam bahasa pemrograman Java untuk menentukan wujud air berdasarkan suhu yang dimasukkan oleh pengguna. Program ini meminta pengguna untuk memasukkan suhu dalam derajat Celsius, kemudian menggunakan serangkaian pernyataan if-else if-else untuk menentukan wujud air berdasarkan suhu tersebut. Jika suhu kurang dari 0, program akan mencetak bahwa air berada dalam keadaan beku. Jika suhu antara 0 hingga 100, program akan mencetak bahwa air berada dalam keadaan cair. Jika suhu lebih dari 100, program akan mencetak bahwa air berada dalam keadaan uap atau gas. Dengan demikian, program memberikan respons yang sesuai tergantung pada suhu yang dimasukkan pengguna.