

**LAPORAN  
PEMROGRAMAN BERBASIS OBJEK**



Nama : Andi Mutiara Ayu Larasati  
Stambuk : 13020230261  
Kelas : TI\_PBO-14 (B4)  
Dosen : Mardiyah Hasnawi, S.Kom., M.T., MTA

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

## Tugas 2 Pemrograman Berbasis Objek

### 1. Output:

```
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac BacaString.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java BacaString

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

Penjelasan :

Program BacaString ini membaca inputan yang kita masukkan menggunakan library `BufferedReader`. Library `IOException` untuk menangani kesalahan input/output serta library `InputStreamReader` untuk mengonversi information input menjadi arrange yang bisa dibaca oleh program. Didalam program ini mendeklarasikan variabel `str` bertipe information `String` untuk menyimpan input. Objek `BufferedReader` `datAIn` dibuat untuk membaca input dari console. Setelah mencetak pesan "Baca string dan numbers: ", program ini akan meminta kita untuk memasukkan sebuah string. Disini saya menginputkan string "andi" dan inputan tersebut dibaca menggunakan `datAIn.readLine()` dan disimpan dalam `str`. Terakhir, program akan mencetak kembali string yang kita masukkan sebelumnya dalam hal ini yang dicetak adalah andi yang telah saya inputkan sebelumnya.

### 2. Output:

```
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac ForEver.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java ForEver
Program akan looping, akhiri dengan ^c
Print satu baris ...
```

Penjelasan:

Program `ForEver` ini adalah program dasar yang memanfaatkan loop `while(true)` untuk menampilkan teks di layar. Pada awal program, terdapat instruksi `System.out.println("Program akan looping, akhiri dengan ^c");` yang menginformasikan bahwa program ini akan terus berjalan sampai dihentikan secara manual. Namun, di dalam loop `while(true)`, ada perintah `System.out.print("Print satu baris ...\n");` yang akan menampilkan teks ke layar, diikuti dengan perintah `break;`, yang langsung menghentikan loop. Dengan adanya `break`, program sebenarnya tidak berlanjut tanpa henti seperti yang dinyatakan sebelumnya, melainkan hanya menampilkan satu baris teks dan kemudian menghentikannya.

### 3. Output:

```
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac If1.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : 8

Nilai a positif 8
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac If1.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : -8
```

Penjelasan:

Program `If1` ini mengilustrasikan pemakaian percabangan `if` dalam Java untuk menilai apakah nilai yang diberikan bilangan positif atau tidak. Dalam program ini, menggunakan `Scanner` digunakan untuk mengambil input yang kita berikan. Program ini akan menampilkan pesan "Contoh IF satu kasus" dan meminta pengguna untuk memasukkan sebuah bilangan bulat.

Selanjutnya, program akan melakukan pemeriksaan dengan `if(a >= 0)`, yang berarti jika `a` lebih dari atau sama dengan nol maka program akan mencetak "Nilai `a` positif: diikuti dengan nilai `a`". Jika nilai `a` adalah negatif, tidak akan ada output tambahan karena tidak ada instruksi didalam source code untuk menangani bilangan negatif.

#### 4. Output:

```
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac If2.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java If2
Contoh IF dua kasus
Ketikkan suatu nilai integer : 61
Nilai a positif 61

D:\Andi Mutiara\Tugas PBO\Tugas 2>javac If2.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java If2
Contoh IF dua kasus
Ketikkan suatu nilai integer : -61
Nilai a negatif -61
```

#### Penjelasan:

Program `If2` ini menerapkan struktur percabangan `if-else` di dalam Java untuk mengevaluasi apakah angka yang dimasukkan memiliki nilai positif atau negatif. Seperti pada program `If1`, program `If2` juga memanfaatkan `Scanner` untuk mengambil inputan dari pengguna. Pertama, program ini akan menampilkan pesan "Contoh IF dua kasus" dan meminta kita sebagai pengguna untuk memasukkan sebuah bilangan bulat. Angka yang dimasukkan juga disimpan dalam variabel `a`. Selanjutnya, akan dilakukan pengecekan jika `(a >= 0)` maka program akan mencetak "Nilai `a` positif" disertai dengan nilai `a`. Apabila kondisi tersebut tidak terpenuhi, maka perintah dalam blok `else` akan dilaksanakan, menampilkan "Nilai `a` negatif" diikuti dengan nilai `a`.

#### 5. Output:

```
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac If3.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer : 95
Nilai a positif 95

D:\Andi Mutiara\Tugas PBO\Tugas 2>javac If3.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer : -95
Nilai a negatif -95

D:\Andi Mutiara\Tugas PBO\Tugas 2>javac If3.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer : 0
Nilai Nol 0
```

#### Penjelasan:

Program `If3` ini mengaplikasikan struktur percabangan `if-else if-else` dalam bahasa pemrograman Java untuk menilai apakah angka yang dimasukkan merupakan angka positif, nol, atau negatif. Seperti pada program `If2`, program ini juga memanfaatkan `Scanner` untuk mengambil input dari pengguna. Pertama-tama, program menampilkan pesan "Contoh IF tiga kasus" dan meminta pengguna untuk memasukkan sebuah bilangan bulat. Angka yang

diterima disimpan ke dalam variabel a. Selanjutnya, program melakukan pemeriksaan: jika (a > 0), maka akan mencetak "Nilai positif" diikuti dengan isi dari a. Jika (a == 0), maka akan menampilkan "Nilai Nol" disertai dengan nilai a. Apabila kondisi-kondisi tersebut tidak tercapai (a < 0), maka instruksi dalam blok else akan dijalankan, mencetak "Nilai a negatif" beserta nilai a.

#### 6. Output:

```
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac KasusBoolean.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java KasusBoolean
true
benar
```

Penjelasan :

Program KasusBoolean ini mengimplementasikan tipe data Boolean dalam struktur pengambilan keputusan if-else di Java. Pada tahap awal, program mendefinisikan sebuah variabel bool yang bertipe Boolean dan menginisiasinya dengan nilai true. Selanjutnya, program melakukan dua evaluasi. Pertama, ketika if(bool) dieksekusi, program akan mengoutput "true". Apabila nilainya false, maka program akan mengoutput "false" pada bagian else. Kedua, pada if(!bool), terdapat operator negasi !(NOT) yang membalikkan nilai bool, sehingga !bool akan menjadi false. Karena kondisi tersebut tidak terpenuhi, program akan melewati instruksi di dalam blok if dan menjalankan else yang menghasilkan output "benar". Jika nilai awal bool adalah false, maka output yang dihasilkan adalah "salah".

#### 7. Output:

```
C:\WINDOWS\system32\cmd. x + -
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac KasusSwitch.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
i
Yang anda ketik adalah i
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac KasusSwitch.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
a
Yang anda ketik adalah a
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac KasusSwitch.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
u
Yang anda ketik adalah u
```

```
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac KasusSwitch.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
e
Yang anda ketik adalah e
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac KasusSwitch.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
o
Yang anda ketik adalah o
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac KasusSwitch.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java KasusSwitch
Ketikkan sebuah huruf, akhiri dengan RETURN
t
Yang anda ketik adalah huruf mati
```

Penjelasan:

Program KasusSwitch ini menjalankan penerapan struktur cabang switch-case dalam Java untuk menilai sebuah karakter yang diinput oleh pengguna. Program ini mendefinisikan variabel cc dengan tipe char dan memanfaatkan Scanner untuk mengambil input. Setelah menampilkan pesan "Silakan masukkan sebuah huruf, kemudian tekan RETURN", program membaca satu karakter melalui masukan.next().charAt(0), yang mengambil karakter pertama dari input. Selanjutnya, program melakukan pemeriksaan dengan menggunakan switch-case. Jika pengguna menginput huruf vokal kecil ('a', 'u', 'e', 'i', atau 'o'), program akan menampilkan pesan yang sesuai dengan huruf yang dimasukkan. Namun, jika karakter yang dimasukkan bukan vokal, blok default akan dieksekusi dan program akan menampilkan "Yang anda ketik adalah huruf mati", yang menunjukkan bahwa huruf tersebut adalah konsonan. Program ini menggambarkan cara menggunakan switch-case untuk mengelola berbagai kemungkinan input tanpa memerlukan banyak pernyataan if-else.

#### 8. Output:

```
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac Konstant.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java Konstant
Jari-jari lingkaran = 87
Luas lingkaran = 23778.012
Akhir program
```

Penjelasan:

Program ini menghitung luas lingkaran berdasarkan jari-jari yang dimasukkan oleh pengguna. Diawali dengan mendeklarasikan konstanta PHI bernilai 3.1415f menggunakan kata kunci final, sehingga nilainya tidak dapat diubah selama program berjalan. Kemudian, variabel r bertipe float dideklarasikan untuk menyimpan nilai jari-jari yang akan diinputkan.

Selanjutnya, dibuat objek Scanner bernama masukan untuk membaca input pengguna. Program menampilkan pesan "Jari-jari lingkaran =" sebagai instruksi agar pengguna memasukkan nilai jari-jari. Misalnya, jika pengguna memasukkan 87, nilai tersebut dibaca menggunakan masukan.nextFloat() dan disimpan dalam variabel r.

Setelah itu, program menghitung luas lingkaran dengan rumus  $\text{PHI} \times r \times r$ , kemudian mencetak hasilnya, yaitu 23778.012. Sebelum eksekusi berakhir, program menampilkan pesan "Akhir program" sebagai tanda bahwa proses telah selesai.

#### 9. Output:

```
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac Max2.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dengan RETURN :
10 6
Ke dua bilangan : a = 10 b = 6
Nilai a yang maksimum: 10

D:\Andi Mutiara\Tugas PBO\Tugas 2>javac Max2.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dengan RETURN :
6 10
Ke dua bilangan : a = 6 b = 10
Nilai b yang maksimum: 10
```

Penjelasan :

Program Max2 menentukan bilangan terbesar dari dua angka yang dimasukkan oleh pengguna. Di dalam program ini, terdapat dua variabel, yaitu a dan b, yang bertipe integer

untuk menyimpan bilangan bulat. Selain itu, dibuat juga objek Scanner bernama masukan untuk membaca input dari pengguna.

Program diawali dengan mencetak pesan "Maksimum dua bilangan", lalu meminta pengguna memasukkan dua bilangan secara terpisah dengan menekan ENTER setelah setiap angka. Nilai yang dimasukkan dibaca menggunakan `masukan.nextInt()` dan disimpan ke dalam variabel `a` dan `b`.

Selanjutnya, program membandingkan kedua bilangan menggunakan struktur `if-else`. Jika `a` lebih besar atau sama dengan `b`, program akan mencetak "Nilai `a` yang maksimum: `a`". Namun, jika `b` lebih besar, maka program mencetak "Nilai `b` yang maksimum: `b`". Dengan cara ini, program dapat menentukan dan menampilkan bilangan terbesar dari dua angka yang diinputkan.

Misalnya ketika saya memasukkan 10 untuk `a` dan 6 untuk `b`, maka program menentukan bahwa `a` adalah bilangan terbesar. Sebaliknya, jika pengguna memasukkan 6 untuk `a` dan 10 untuk `b`, maka `b` yang menjadi nilai maksimum.

#### 10. Output:

```
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac PriFor.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java PriFor
Baca N, print 1 s/d N N = 7
1
2
3
4
5
6
7
Akhir program
```

Penjelasan:

Program `PriFor` mencetak bilangan dari 1 hingga `N` sesuai dengan nilai yang dimasukkan oleh pengguna. Di awal program, dideklarasikan variabel `i` sebagai penghitung dalam perulangan `for`, sedangkan variabel `N` digunakan untuk menyimpan batas angka yang akan dicetak. Selain itu, dibuat objek Scanner bernama `masukan` untuk membaca input pengguna.

Program menampilkan pesan "Baca N, print 1 s/d N", lalu meminta pengguna memasukkan nilai `N`, yang kemudian dibaca menggunakan `masukan.nextInt()`. Setelah itu, program menjalankan perulangan `for`, di mana `i` dimulai dari 1 dan terus meningkat hingga mencapai `N`, dengan kondisi `i <= N`.

Misalnya ketika saya memasukkan `N = 7`, maka program mencetak angka 1, 2, 3, 4, 5, 6, dan 7 menggunakan `System.out.println(i)` pada setiap iterasi. Setelah perulangan selesai, program menampilkan pesan "Akhir program" sebagai tanda bahwa eksekusi telah berakhir.

#### 11. Output:

```
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac PrintIterasi.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java PrintIterasi
Nilai N > 0 = 6
Print i dengan ITERATE :
1
2
3
4
5
6
```

Penjelasan:

Program PrintIterasi mencetak bilangan dari 1 hingga N sesuai dengan input yang diberikan oleh pengguna. Program ini mendeklarasikan variabel *i* sebagai penghitung dalam perulangan (for) dan variabel *N* untuk menyimpan batas angka yang akan dicetak. Selain itu, dibuat objek Scanner bernama masukan untuk membaca input pengguna.

Setelah mencetak pesan "Nilai N > 0 =", program meminta pengguna memasukkan nilai *N*, yang dibaca menggunakan `masukan.nextInt()`. Kemudian, program menetapkan *i* = 1 dan menjalankan perulangan `for(;;)`, yang merupakan infinite loop tanpa kondisi eksplisit.

Dalam perulangan ini, setiap iterasi mencetak nilai *i* menggunakan `System.out.println(i)`. Jika *i* sudah mencapai *N*, perulangan dihentikan dengan `break`. Jika belum, nilai *i* bertambah satu (`i++`).

Misalnya ketika saya memasukkan *N* = 6, maka program akan mencetak angka 1, 2, 3, 4, 5, dan 6 lalu perulangan berhenti.

## 12. Output:

```
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac PrintRepeat.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java PrintRepeat
Nilai N > 0 = 4
Print i dengan REPEAT:
1
2
3
4
```

Penjelasan:

Program PrintRepeat mencetak bilangan dari 1 hingga *N* menggunakan perulangan do-while, berdasarkan nilai *N* yang dimasukkan oleh pengguna. Program ini mendeklarasikan variabel *i* sebagai penghitung dalam perulangan, sementara variabel *N* menyimpan batas angka yang akan dicetak. Selain itu, dibuat objek Scanner bernama masukan untuk membaca input dari pengguna.

Setelah mencetak pesan "Nilai N > 0 =", program meminta pengguna memasukkan nilai *N*, yang kemudian dibaca menggunakan `masukan.nextInt()`. Selanjutnya, variabel *i* diinisialisasi dengan nilai 1, lalu program menjalankan perulangan do-while. Perulangan ini memastikan bahwa blok kode dalam do dieksekusi setidaknya satu kali sebelum mengevaluasi kondisi (`i ≤ N`).

Pada setiap iterasi, program mencetak nilai *i* menggunakan `System.out.print(i + "\n")`, lalu menaikkan *i* dengan `i++`. Perulangan terus berlanjut hingga nilai *i* melebihi *N*, setelah itu program berhenti.

Misalnya ketika saya memasukkan  $N = 4$ , maka program akan mencetak angka 1, 2, 3, dan 4 lalu perulangan berhenti.

### 13. Output:

```
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac PrintWhile.java

D:\Andi Mutiara\Tugas PBO\Tugas 2>java PrintWhile
Nilai N > 0 = 10
Print i dengan WHILE:
1
2
3
4
5
6
7
8
9
10
```

Penjelasan:

Program PrintWhile mencetak bilangan dari 1 hingga N menggunakan perulangan while, berdasarkan nilai N yang dimasukkan oleh pengguna. Variabel i digunakan sebagai penghitung dalam perulangan, sementara variabel N menyimpan batas angka yang akan dicetak. Selain itu, dibuat objek Scanner bernama masukan untuk membaca input dari pengguna.

Setelah menampilkan pesan "Nilai N > 0 =", program meminta pengguna memasukkan nilai N, yang dibaca menggunakan masukan.nextInt(). Selanjutnya, variabel i diinisialisasi dengan nilai 1, lalu program menjalankan perulangan while, yang terus berjalan selama kondisi ( $i \leq N$ ) bernilai true.

Dalam setiap iterasi, program mencetak nilai i menggunakan System.out.println(i), lalu menaikkan nilai i dengan i++. Perulangan berhenti ketika nilai i melebihi N.

Misalnya ketika saya memasukkan  $N = 10$ , program akan mencetak angka 1, 2, 3, 4, 5, 6, 7, 8, 9, dan 10 lalu perulangan berhenti.

### 14. Output:

```
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac PrintWhile1.java

D:\Andi Mutiara\Tugas PBO\Tugas 2>java PrintWhile1
Nilai N > 0 = 5
Print i dengan WHILE (ringkas):
1
2
3
4
5
```

Penjelasan:

Program PrintWhile1 mencetak bilangan dari 1 hingga N menggunakan perulangan while dengan cara yang lebih ringkas. Variabel i digunakan sebagai penghitung dalam perulangan dan langsung diinisialisasi dengan 1, sementara variabel N menyimpan batas angka yang akan dicetak. Objek Scanner bernama masukan digunakan untuk membaca input dari pengguna.

Setelah mencetak pesan "Nilai N > 0 =", program meminta pengguna memasukkan nilai N, yang kemudian dibaca menggunakan masukan.nextInt(). Selanjutnya, program menjalankan perulangan while, yang terus berjalan selama kondisi ( $i \leq N$ ) bernilai true.



Berbeda dari program sebelumnya, dalam perulangan ini digunakan ekspresi `System.out.println(i++)`, yang mencetak nilai `i` dan langsung menaikkan nilainya dalam satu perintah. Perulangan berhenti saat `i` melebihi `N`.

Misalnya ketika saya memasukkan `N = 6`, program akan mencetak angka 1, 2, 3, 4, 5, dan 6, lalu berhenti.

#### 15. Output:

```
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac PrintXinterasi.java

D:\Andi Mutiara\Tugas PBO\Tugas 2>java PrintXinterasi
Masukan nilai x (int), akhiri dg 999: 261
Masukan nilai x (int), akhiri dg 999: 3
Masukan nilai x (int), akhiri dg 999: 4
Masukan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 268

D:\Andi Mutiara\Tugas PBO\Tugas 2>javac PrintXinterasi.java

D:\Andi Mutiara\Tugas PBO\Tugas 2>java PrintXinterasi
Masukan nilai x (int), akhiri dg 999: 999
Kasus kosong
Hasil penjumlahan = 0
```

Penjelasan:

Program `PrintXinterasi` menjumlahkan angka yang dimasukkan hingga pengguna memasukkan 999 sebagai tanda berhenti. Variabel `x` menyimpan input angka, sementara `Sum` menyimpan hasil penjumlahan.

Setelah mencetak "Masukkan nilai x (int), akhiri dengan 999:", program meminta input. Jika angka pertama 999, program mencetak "Kasus kosong". Jika tidak, angka pertama disimpan ke `Sum`, lalu perulangan `for(;;)` berjalan, menambahkan setiap angka ke `Sum` hingga 999 dimasukkan, yang menghentikan perulangan.

Contoh:

- Input 261, 3, 4, 999 output "Hasil penjumlahan = 268".
- Input langsung 999, output "Kasus kosong", "Hasil penjumlahan = 0".

#### 16. Output:

```
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac PrintXRepeat.java

D:\Andi Mutiara\Tugas PBO\Tugas 2>java PrintXRepeat
Masukan nilai x (int), akhiri dg 999: 98
Masukan nilai x (int), akhiri dg 999: 76
Masukan nilai x (int), akhiri dg 999: 24
Masukan nilai x (int), akhiri dg 999: 42
Masukan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 240

D:\Andi Mutiara\Tugas PBO\Tugas 2>javac PrintXRepeat.java

D:\Andi Mutiara\Tugas PBO\Tugas 2>java PrintXRepeat
Masukan nilai x (int), akhiri dg 999: 999
Kasus kosong
```

Penjelasan :

Program `PrintXRepeat` menjumlahkan angka yang dimasukkan hingga pengguna memasukkan 999 sebagai tanda berhenti. Variabel `x` menyimpan input angka, sedangkan `Sum` menyimpan hasil penjumlahan. Objek `Scanner` digunakan untuk membaca input pengguna.

Setelah mencetak "Masukkan nilai x (int), akhiri dengan 999:", program meminta input. Jika angka pertama 999, program mencetak "Kasus kosong". Jika tidak, Sum diinisialisasi ke 0, lalu perulangan do-while dijalankan.

Setiap angka yang dimasukkan ditambahkan ke Sum, lalu program kembali meminta input. Perulangan berjalan hingga pengguna memasukkan 999, yang menghentikan proses (while (x != 999)). Setelah selesai, program mencetak "Hasil penjumlahan = Sum".

Contoh:

- Input 08, 76, 24, 42, 999, output "Hasil penjumlahan = 240".
- Input langsung 999, output "Kasus kosong".

#### 17. Output:

```
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac PrintXRepeat.java

D:\Andi Mutiara\Tugas PBO\Tugas 2>java PrintXRepeat
Masukan nilai x (int), akhiri dg 999: 999
Kasus kosong

D:\Andi Mutiara\Tugas PBO\Tugas 2>javac PrintXWhile.java

D:\Andi Mutiara\Tugas PBO\Tugas 2>java PrintXWhile
Masukan nilai x (int), akhiri dg 999: 5
Masukan nilai x (int), akhiri dg 999: 3
Masukan nilai x (int), akhiri dg 999: 4
Masukan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 12

D:\Andi Mutiara\Tugas PBO\Tugas 2>javac PrintXWhile.java

D:\Andi Mutiara\Tugas PBO\Tugas 2>java PrintXWhile
Masukan nilai x (int), akhiri dg 999: 999
Hasil penjumlahan = 0
```

Penjelasan:

Program PrintXWhile menjumlahkan angka yang dimasukkan hingga pengguna memasukkan 999 sebagai tanda berhenti. Variabel x menyimpan input angka, sedangkan Sum menyimpan hasil penjumlahan. Scanner digunakan untuk membaca input.

Setelah mencetak "Masukkan nilai x (int), akhiri dengan 999:", program meminta input. Jika angka bukan 999, program menambahkannya ke Sum dan terus meminta input baru. Perulangan while (x != 999) berjalan hingga pengguna memasukkan 999, lalu program mencetak "Hasil penjumlahan = Sum".

Contoh:

- Input 5, 3, 4, 999, output "Hasil penjumlahan = 12".
- Input langsung 999, output "Hasil penjumlahan = 0".

#### 18. Output:

```
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac SubProgram.java

D:\Andi Mutiara\Tugas PBO\Tugas 2>java SubProgram
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
8 4
Ke dua bilangan : a = 8 b : 4
Maksimum = 8
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 4 b = 8
```

Penjelasan:

Program SubProgram mencari nilai maksimum dari dua bilangan dan mencoba menukarnya. Variabel a dan b menyimpan input dari pengguna, sedangkan objek Scanner membaca input.

Setelah pengguna memasukkan dua bilangan, program mencetak nilainya dan menentukan bilangan terbesar. Fungsi tukar(a, b) digunakan untuk menukar nilai, tetapi karena Java menggunakan passing by value untuk tipe data primitif, perubahan dalam fungsi tidak mempengaruhi nilai asli di main().

Contoh:

- Input a = 2, b = 7, output "Maksimum = 7", dan pertukaran dalam fungsi tidak mengubah nilai di main().

#### 19. Output:

```
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac Tempair.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 26
Wujud air cair
26
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac Tempair.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>-1
'-1' is not recognized as an internal or external command,
operable program or batch file.
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac Tempair.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = -1
Wujud air beku
-1
D:\Andi Mutiara\Tugas PBO\Tugas 2>javac Tempair.java
D:\Andi Mutiara\Tugas PBO\Tugas 2>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 120
Wujud air uap/gas
120
```

Penjelasan :

Program Tempair menentukan wujud air berdasarkan temperatur yang diinputkan pengguna menggunakan percabangan if-else if-else. Variabel T bertipe integer digunakan untuk menyimpan suhu dalam derajat Celcius, sementara Scanner membaca input dari pengguna. Jika nilai T kurang dari 0, program mencetak "Wujud Air Beku". Jika T berada dalam rentang 0 hingga 100 derajat Celcius, program mencetak "Wujud Air Cair". Jika T lebih dari 100 derajat Celcius, program mencetak "Wujud Air Uap/Gas". Misalnya, jika pengguna menginput -1, output yang dihasilkan adalah "Wujud Air Beku". Jika memasukkan 26, outputnya "Wujud Air Cair", dan jika 120, outputnya "Wujud Air Uap/Gas".