

Makassar, 23 maret 2022

LAPORAN TUGAS 2
PEMROGRAMAN BERORIENTASI OBJEK



Nama : Avin Safitri
Stambuk : 13020200025
Kelas : B1

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS MUSLIM INDONESIA
MAKASSAR
2021

1. Tugas Praktek : Praktek Program Java : Variabel dan tipe Data

1. Kode Program

```
E:\>javac Asgd11.java  
  
E:\>java Asgd11  
f : 20.0  
f11: 10.0  
  
E:\>_
```

Program 1 mempunyai tujuan untuk mencontohkan tampilan tipe data float dan double pada pemrograman java. Nilai float yang diinput pada kamus adalah 20.0f, sementara nilai double(f11) yang di set adalah 10.0f, kemudian untuk mencetak digunakan perintah :

System.out.println ("f : "+f + "\nfil:"+f11);

Dan hasilnya dapat dilihat pada gambar output diatas. **Float** & **Double** digunakan untuk menampung angka pecahan, perbedaan antara keduanya terletak pada jangkauan angka serta tingkat ketelitian.

2. Kode Program

```
E:\>javac Asign.java  
  
E:\>java Asign  
hello  
Ini nilai i :5  
  
E:\>_
```

Program 2 mempunyai fungsi untuk menampilkan nilai yang telah di assigned, pada kasus program ini, nilai yang di input pada int i adalah 5. Untuk mencetak, digunakan syntax:

System.out.print("hello\n"); i=5;

System.out.println(" ini nilai i : "+ i);

3. Kode Program

```
E:\>javac ASIGNi.java  
  
E:\>java ASIGNi  
Karakter = A  
Karakter = Z  
Karakter = A  
Karakter = Z  
Bilangan integer (short) = 1  
          (int) = 1  
          (long)= 10000  
Bilangan Real x = 50.20000076293945  
Bilangan Real y = 50.2  
  
E:\>
```

Program 3 memiliki tujuan untuk memberikan & menampilkan contoh sederhana untuk mendefinisikan variable-variabel bilangan bulat, karakter, dan bilangan riil, yang termasuk kedalam tipe data primitive. Yang ditampilkan pada program ini diantaranya:

1. Short, yang merupakan tipe data integer(bilangan bulat) dengan rentang yang cukup pendek.
2. int, yang merupakan tipe data integer(bilangan bulat) yang paling sering digunakan
3. long, yang merupakan tipe data integer(bilangan bulat). dengan rentang nilai yang paling besar diantara tipe data integer.
4. char, yang merupakan tipe data untuk menyatakan sebuah karakter, bisa di inisialisasi dengan memasukkan bilangan, ataupun dengan memakai ‘’.
5. Float dan double, yang digunakan untuk menampung pecahan.

4. Kode Program

```
E:\>javac BacaData.java
E:\>java BacaData
Contoh membaca dan menulis, ketik nilai integer:
19
Nilai yang dibaca : 19
E:\>
```

Program 4 memiliki tujuan untuk memberikan gambaran tentang penggunaan Class Scanner untuk membaca integer.

Import java.util.scanner

digunakan untuk mengimpor/memasukkan paket scanner kedalam code.

Scanner masukan

Digunakan untuk mendefinisikan scanner

masukan=new Scanner(System.in); a=masukan.nextInt();

dan syntax diatas digunakan untuk membuat sebuah perintah untuk memasukkan nilai.

5. Kode Program

```

E:\>javac Bacakar.java
E:\>java Bacakar
hello
baca 1 karakter : A
baca 1 bilangan : 6
A
6
bye
E:\>

```

Program 5 mempunyai tujuan untuk memberikan contoh penggunaan `InputStreamReader` dan `BufferedReader` dalam pembacaan dan penginputan data. Pada program ini dideklarasikan dua variabel yaitu `cc` yang bertipe data `char` dan `bil` yang bertipe data `int`.

`InputStreamReader isr= new InputStreamReader(System.in);`

Adalah syntax yang digunakan untuk mendefinisikan `InputStreamReader` pada program ini.

`nputStreamReader` merupakan sebuah Variabel yang terdapat Pada sebuah Aplikasi yang memiliki basis data Java yang di gunakan untuk sebuah variabel dalam memasukan Inputan ke dalam sebuah program. Dari nama Input sudah pasti memiliki Fungsi Input (yang di gunakan dalam memasukan sebuah data pada sebuah program yang kita buat) dan data tersebut dalam bentuk Variabel.

`BufferedReader dataIn= new BufferedReader(isr);`

Adalah syntax yang digunakan untuk mendefinisikan `BufferedReader` pada program ini

Sedangkan dari `BufferedReader` merupakan Sebuah Variabel dalam Aplikasi Java dengan Fungsi Membaca Sebuah Variabel yang akan kita Input pada sebuah Program. Jadi Fungsi `BufferedReader` Adalah Membaca Sebuah Bilangan Character sehingga bilangan atau karakter tersebut dapat dimanfaatkan untuk Sebuah Program.

Pada program ini terdapat syntax:

`cc=dataIn.readLine().charAt(0);`

yang merupakan method dari class `BufferedReader`. Method ini digunakan untuk mengembalikan baris lengkap dari representasi String dengan napa yang telah dimasukkan user pada console, sedangkan `charAt` digunakan untuk mengambil karakter pada sebuah string sesuai index yang diinginkan.

Dan:

`bil=Integer.parseInt(dataIn.readLine`

yang didalamnya terdapat `parseInt`. `parseInt` digunakan untuk mengembalikan nilai integer yang telah diinput sesuai data yang diinput ke konsol.

6. Kode Program

```
E:\>javac Casting1.java

E:\>java Casting1
5.0
6.0
2
53
53.0
53.0
3
3.14
E:\>
```

Program 6 mempunyai tujuan untuk mencontohkan penggunaan Casting pada tipe data primitive. Casting secara sederhana adalah pengkonversian tipe data ke tipe data yang lain. Ada 4 tipe data yang akan dikonversi pada program ini, yaitu int, float, char, dan double. Pertama-tama, data-data akan dideklarasikan terlebih dahulu lalu dikonversi dengan menggunakan perintah print yang dimodifikasi untuk casting, contohnya pada syntax dibawah ini:

System.out.println((float)a); // int <-- float

Yang digunakan untuk mengkonversi tipe data int menjadi float.

7. Kode Program

```
E:\>javac Casting2.java

E:\>java Casting2
a : 67
k : 45.0
d : 100.0
n : 9
m : 5
l : 3.2
k : 67.0
c : 9.0
l : 3.2
E:\>
```

Tujuan dari program 7 adalah untuk menunjukkan penggunaan casting pada tipe data class. Pada program ini, terdapat beberapa method seperti parseInt, parseDouble, parseFloat, yang digunakan untuk mengembalikan nilai sesuai tipe data yang disebutkan. valueOf memiliki fungsi yang mirip dengan parse, yaitu mengembalikan nilai/objek bilangan tertentu yang memegang nilai argument. Seperti biasa, data dideklarasikan terlebih dahulu lalu dikonversi dengan casting.

Tujuan dari program 8 adalah untuk menunjukkan pemakaian operator kondisional, khususnya operator ‘?’ atau operator ternary. Pada program ini, operator tersebut digunakan pada statement:

x < y ? x : y

(x<y) adalah kondisi yang diperiksa, jika bernilai true, maka x yang akan ditampilkan, sebaliknya jika bernilai false, maka y yang akan ditampilkan. Karena

pada program ini kondisi tersebut bernilai true, maka yang ditampilkan adalah nilai x yaitu 1.

8. Kode Program

```
E:\>javac Ekspresi.java

E:\>java Ekspresi
x = 1
y = 2
hasil ekspresi = (x<y)?x:y = 1
E:\>
```

9. Kode Program

```
E:\>javac Ekspresi1.java

E:\>java Ekspresi1
x/y (format integer) = 0
x/y (format float) = 0
x/y (format integer) = 0.5
x/y (format float) = 0.5
float(x)/float(y) (format integer) = 0.5
float(x)/float(y) (format float) = 0.5
x/y (format integer) = 3
x/y (format float) = 3
E:\>
```

Program 9 bertujuan untuk menunjukkan contoh pembagian atau penggunaan operator

pembagian pada tipe data integer sekaligus casting dari integer ke bentuk float. Pada program ini digunakan operator aritmatika '/' untuk melakukan operasi pembagian, salah satunya:

System.out.print ("x/y (format integer) = "+ x/y);

Selain itu, juga terdapat beberapa syntax untuk casting pada program ini, salah satunya:

Yang mengonversi integer menjadi float.

10. Kode Program

```
E:\>javac Hello.java

E:\>java Hello
Hello
Hello World
Welcome
E:\>
```

Program 10 mempunyai tujuan untuk mencontohkan penggunaan dasar dari

System.out.print(""); dan

System.out.println("");

Serta \n

Print digunakan untuk mencetak data, sedangkan println digunakan untuk mencetak data serta membuat garis baru pada output. \n memiliki fungsi yang mirip dengan println, yaitu membuat garis baru.

11. Kode Program

```
E:\>javac Incr.java
E:\>java Incr
Nilai i : 5
Nilai j : 3
E:\>
```

Program ini bertujuan untuk menunjukkan contoh cara pemakaian beberapa operator seperti +, &, <<, dan >> dalam java. + digunakan untuk menjumlahkan 2 buah variable, & merupakan operator bitwise yang digunakan untuk operasi dalam bentuk bit dengan logika AND(Nilai kebenaran operator AND adalah benar ketika kedua operand bernilai benar). << dan >> melambangkan pergeseran bit ke kiri dan ke kanan, respectively.

12. Kode Program

```
E:\>javac Oper1.java
E:\>java Oper1
n = 10
x = 1
y = 2
n & 8 = 8
x & ~ 8 = 1
y << 2 = 8
y >> 3 = 0
E:\>
```

Program ini bertujuan untuk menunjukkan pemakaian beberapa operator bitwise seperti &, |, ^, dan ~ . & merupakan operator bitwise yang digunakan

untuk operasi dalam bentuk bit dengan logika AND(Nilai kebenaran operator AND adalah benar ketika kedua operand bernilai benar), | merupakan operator bitwise yang digunakan untuk operasi dalam bentuk bit dengan logika OR(bernilai benar jika kedua variable yang diinput bernilai salah), ^ digunakan untuk operasi dalam bentuk dengan logika XOR(bernilai true jika jumlah variable yang diinput adalah ganjil), dan ~ digunakan untuk operasi dengan bentuk logika NOT(Negasi dari variable).

13. Kode Program

```
E:\>java Oper2
i = 3
j = 4
i & j = 0
i | j = 7
i ^ j = 7
81.0
~i = -4
E:\>
```

Program ini bertujuan untuk menunjukkan pemakaian beberapa operator bitwise seperti &, |, ^, dan ~. & merupakan operator bitwise yang digunakan untuk operasi dalam bentuk bit dengan logika AND(Nilai kebenaran operator AND adalah benar ketika kedua operand bernilai benar), | merupakan operator bitwise yang digunakan untuk operasi dalam bentuk bit dengan logika OR(bernilai benar jika kedua variable yang diinput bernilai salah), ^ digunakan untuk operasi dalam bentuk dengan logika XOR(bernilai true jika jumlah variable yang diinput adalah ganjil), dan ~ digunakan untuk operasi dengan bentuk logika NOT(Negasi dari variable).

14. Kode Program

```
E:\>javac Oper3.java
E:\>java Oper3
true
false
true
true
true
E:\>
```

Programan 14 memiliki fungsi yang hampir mirip dengan program 13, hanya saja kali ini langsung memakai true sebagai acuan operasi. Ada terdapat salah satu operator logika pada program ini, yaitu ||, yang memakai logika OR. Syntax:

if (true || true){ System.out.println(true); } //true = true or true

15. Kode Program


```
E:\>javac Oper4.java

E:\>java Oper4
Nilai e = 10
Nilai k = 0
Nilai k = 4
E:\>
```

Program 15 mempunyai fungsi menunjukkan penggunaan beberapa operator yang berbeda tipe seperti operator ternary(?) dan increment(++), pada salah satu syntax, kita bisa melihat penggunaan kedua operator tersebut secara sekaligus:

```
k = ((i++>j++) ? i: j) ;
```

```
System.out.print ("\nNilai k = "+ k);
```

16. Kode Program

```
E:\>javac Oprator.java

E:\>java Oprator
Silahkan baca teksnya dan tambahkan perintah untuk menampilkan output
Operasi Logika
==AND==
true && false = false
==OR==
true || false = true
==NEGASI==
! true = false
==XOR==
true ^false = true
Operasi Numerik
5 + 2 = 7
5 - 2 = 3
5 * 2 = 10
5 / 2 = 2.5
5 / 2 = 2
5 % 2 = 1
Operasi Numerik
5.0 + 5.0 = 10.0
5.0 - 5.0 = 0.0
5.0 / 5.0 = 1.0
5.0 * 5.0 = 25.0
Operasi Relasional Numerik
5 == 2 : false
5 != 2 : true
5 < 2 : false
5 > 2 : true
5 <= 2 : false
5 >= 2 : true
Operasi Relasional Numerik
5.0 == 5.0 : true
5.0 != 5.0 : false
5.0 < 5.0 : false
5.0 > 5.0 : false
5 <= 5.0 : true
5.0 >= 5.0 : true
```

Program 16 bertujuan untuk menunjukkan berbagai macam operasi yang dapat dilakukan oleh setiap operator yang terdapat dalam java. Sederhananya, program ini merupakan gabungan dari program 11-15.

2. Silahkan kerjakan tugas praktek di kelas, kumpul setelah waktu kuliah selesai berikut:

1. Kode Program

```
D:\>javac BacaString.java
D:\>java BacaString.java

Baca string dan Integer:
masukkan sebuah string: Afin2312
String yang dibaca : Afin2312
D:\>
```

Program 1 bertujuan untuk memberikan contoh sederhana pembacaan tipe data string dengan menggunakan `InputStreamReader` dan `BufferedReader`. Terdapat juga 2 class import baru yaitu:

```
import java.io.IOException;
```

```
import javax.swing.*;
```

javax.swing.* mengimpor kelas-kelas pada java yang bisa digunakan untuk mengembangkan aplikasi berbasis GUI. Sementara java.io.IOException mengimpor sebuah method yang dapat membuat program membaca input Data String.

2. Kode program

[illegible]

Program 2 memiliki tujuan untuk menunjukkan contoh perulangan while. Pada program ini, selama kondisinya bernilai true, maka program akan terus menerus diulang hingga pernyataan nya bernilai false, tetapi dikarenakan tidak terdapat kondisi yang bernilai false, maka program terus menerus berulang tanpa akhir.

3. Kode Program

```

E:\>javac If1.java

E:\>java If1
Contoh IF satu kasus
Ketikkan suatu nilai integer : 2

Nilai a positif 2
E:\>

```

Programan 3 memiliki tujuan untuk menunjukkan contoh pemakaian percabangan if. Di program ini terdapat pemakaian scanner untuk mempermudah penginputan data. Pada program ini, jika nilai yang diinput pada a bernilai ≥ 0 , maka di program akan dicetak Nilai a positif.

```
if (a >= 0)
```

```
System.out.print ("\nNilai a positif "+ a
```

4. Kode Program

```

E:\>javac If2.java

E:\>java

E:\>java If2
Contoh IF dua kasus
Ketikkan suatu nilai integer :6
Nilai a positif 6
E:\>

```

Program 4 memiliki tujuan menunjukkan contoh pemakaian IF dua kasus (if, else). Pada program ini juga digunakan scanner untuk mempermudah input. Jika nilai a ≥ 0 , maka akan dicetak Nilai a positif, tetapi apabila nilai a tidak memenuhi kondisi tersebut, maka akan tertius Nilai a negative.

```
if (a >= 0){
```

```
System.out.println ("Nilai a positif "+ a);
```

```
} else{
```

```
System.out.println ("Nilai a negatif "+ a);
```

5. Kode Program

```

E:\>javac If3.java

E:\>java If3
Contoh IF tiga kasus
Ketikkan suatu nilai integer :3
Nilai a positif 3
E:\>

```

Fungsi dari program ini adalah menunjukkan contoh pemakaian dari IF 3 kasus (if, else if, else). Program ini juga memakai scanner. Jika nilai a lebih dari 0, maka akan tertulis Nilai a positif, tetapi jika nilai a sama dengan 0, maka akan tertulis nilai nol, dan jika nilai a tidak memenuhi kedua kondisi tersebut, maka akan tertulis Nilai a negative.

6. Kode Program

```
E:\>javac KasusBoolean.java  
  
E:\>java KasusBoolean  
true  
benar  
  
E:\>
```

Tujuan dari program 6 adalah untuk menunjukkan sebuah kasus IF 2 kasus (if, else) dengan menggunakan tipe data Boolean. Kondisi pertama adalah jika bool bernilai true (yang sudah di definisikan dengan bool=true;), maka akan tertulis true pada output program, tetapi jika tidak memenuhi kondisi tersebut, maka akan tertulis false pada output. Selain itu, ada juga kondisi !bool, yang berarti negasi/kebalikan dari bool, yang berarti bool disini memiliki nilai false. Jika bool bernilai true, maka yang tertulis di output adalah salah, akan tetapi jika bool bernilai false, maka akan tertulis benar di output.

7. Kode Program

Program 7 bertujuan untuk menunjukkan sebuah contoh penggunaan percabangan switch case pada java. Pada program ini digunakan scanner untuk mempermudah input. Ada 5 case di program ini, jika anda mengetik salah satu dari huruf a,i,u,e,o maka akan tercetak “yang anda ketik adalah (input)”, tetapi jika anda mengetik selain dari 5 huruf tersebut, maka akan tercetak yang anda ketik adalah huruf mati. **Switch(nama_variabel)** adalah inisialisasi dari percabangan switch case, **Case ‘value1...n’** adalah kode program yang dijalankan apabila nama variable sesuai dengan value. **Default** adalah kode program yang dijalankan apabila tidak ada kondisi yang memenuhi. Terdapat juga **Break**, yang berfungsi untuk menghentikan paksa perulangan yang berlangsung

```
E:\>javac KasusSwitch.java  
  
E:\>java KasusSwitch  
Ketikkan sebuah huruf, akhiri dengan RETURN  
a  
Yang anda ketik adalah a  
  
E:\>
```

8. Kode Program

```

E:\>javac Konstant.java

E:\>java Konstant
Dari-jari lingkaran =7
Luas lingkaran = 153.9335
Akhir program

E:\>

```

Program 8 bertujuan untuk menunjukkan contoh dari pemakaian konstanta pada penghitungan luas lingkaran. Disini terdapat keyword **Final**. Final digunakan untuk pendeklarasian yang absolute atau mutlak. Disini juga digunakan scanner untuk mempermudah penginputan. Dan terdapat syntax yang berbunyi:

System.out.print ("Luas lingkaran = "+ (PHI * r * r)+"\n"

Yang berfungsi untuk menghitung luas lingkaran dengan rumus $\pi \times r \times r$.

9. Kode Program

```

E:\>javac Max2.java

E:\>java Max2
Maksimum dua bilangan :
Ketikkan dua bilangan, pisahkan dg RETURN :
11
5
Ke dua bilangan : a = 11 b = 5
Nilai a yang maksimum 11

E:\>

```

Program 9 bertujuan untuk mencari maksimum dari dua bilangan yang dibaca dengan bantuan IF dua kondisi (if, else). Scanner dipakai untuk mempermudah penginputan. Kondisinya adalah jika nilai $a \geq b$, maka tercetak nilai a yang maksimum. Tetapi jika kondisi tersebut tidak terpenuhi (alias b yang lebih besar), maka tercetak nilai b yang maksimum.

10. Kode Program

```

E:\>javac PriFor.java

E:\>java PriFor
Baca N, print 1 s/d N N = 12
1
2
3
4
5
6
7
8
9
10
11
12
Akhir program

```

Program 10 bertujuan untuk mencontohkan perulangan dengan menggunakan FOR(kondisi). Scanner digunakan untuk mempermudah penginputan data. Terdapat juga beberapa operator seperti =, <=, dan increment(++) yang dipakai dalam kondisi FOR. Kondisi pada program ini yaitu:

```
i = 1 i <= N i++
```

Program ini akan terus mencetak nilai yang dimulai dari 1 sampai dengan bilangan yang anda input.

11. Code Program

```

E:\>javac PrintIterasi.java

E:\>java PrintIterasi
Nilai N >0 = 6
Print i dengan ITERATE :
1
2
3
4
5
6

E:\>

```

Program 11 memiliki fungsi yang mirip dengan program 10, yaitu mencontohkan pemakaian perulangan, akan tetapi, pada program 11, digunakan juga FOR(;;) atau bisa dibilang iterate. Kondisi if adalah jika i bernilai sama dengan N, maka perulangan akan dihentikan, namun jika tidak memenuhi kondisi tersebut, maka nilai I akan dicetak secara berurut dari nilai 1 sampai dengan nilai value yang anda inputkan.

12. Code Program

```
E:\>javac PrintRepeat.java

E:\>java PrintRepeat
Nilai N >0 = 8
Print i dengan REPEAT:
1
2
3
4
5
6
7
8
```

Program 12 memiliki fungsi yang sama dengan program 11, tetapi disini melainkan memakai for, kita memakai perulangan DO WHILE. Kode akan menuruh system untuk melakukan pencetakan nilai i sampai dengan value yang di inputkan selama kondisi WHILE terpenuhi ($i \leq N$);.

13. Code Program

```
E:\>javac PrintWhile.java

E:\>java PrintWhile
Nilai N >0 = 5
Print i dengan WHILE:
1
2
3
4
5

E:\>
```

Program 13 mempunyai tujuan untuk menunjukkan contoh cara perulangan dengan memakai WHILE. Selama kondisi di dalam while terpenuhi, maka system akan mencetak nilai dari 1 sampai dengan value yang kita input.

14. Code Program

```

E:\>javac PrintWhile1.java

E:\>java PrintWhile1
Nilai N >0 = 13
Print i dengan WHILE (ringkas):
1
2
3
4
5
6
7
8
9
10
11
12
13
E:\>

```

Program 14 mempunyai tujuan yang sama dengan 13, bahkan strukturnya mirip. Hanya saja, pada program 14 diringkas penggunaan code nya di bagian perulangan WHILE nya.

15. Code Program

```

E:\>javac PrintXinterasi.java

E:\>java PrintXinterasi
Masukkan nilai x (int), akhiri dg 999 : 8
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 8
E:\>

```

Program 15 memiliki tujuan untuk menjumlahkan nilai yang dibaca (x) dengan iterate atau penggunaan if else dan perulangan for. Jika nilai x yang dimasukkan pada awal adalah 999, maka akan tercetak kasus kosong. Jika tidak memenuhi perulangan tersebut, akan dilakukan operasi penjumlahan dengan pendeklarasian Sum=x;. kondisi perulangan for dipakai dan scanner dipakai untuk menginput nilai x untuk operasi penjumlahan. Jika nilai x=999 maka operasinya akan dihentikan secara paksa, akan tetapi jika itu tidak terpenuhi, maka operasi penjumlahan Sum=Sum+x; akan dilakukan.

16. Code Program

```

E:\>javac PrintXRepeat.java

E:\>java PrintXRepeat
Masukkan nilai x (int), akhiri dg 999 : 16
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 16

```

Program 16 memiliki fungsi yang mirip dengan 15, yaitu menjumlahkan nilai yang di input, akan tetapi disini dipakai if else dan perulangan do while. Jika

x memiliki nilai 999 maka akan tampil kasus kosong. Jika kondisi tersebut tidak terpenuhi, maka akan dilakukan operasi penjumlahan dengan inisialisasi Sum=0; Selanjutnya do akan mengeksekusi perintah penjumlahan dengan nilai yang di input dari scanner selama x tidak memiliki nilai yang sama dengan 999.

17. Code Program

```
E:\>javac PrintXWhile.java

E:\>java PrintXWhile
Masukkan nilai x (int), akhiri dg 999 : 3 999
Masukkan nilai x (int), akhiri dg 999 : Hasil penjumlahan = 3

E:\>
```

Program 17 memiliki fungsi yang mirip dengan 16, perbedaannya disini dipakai perulangan while daripada do while. Program penjumlahan akan dieksekusi selama nilai x yang diinput scanner tidak sama dengan 999.

18. Code Program

```
E:\>javac SubProgram.java

E:\>java SubProgram
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
11
8
Ke dua bilangan : a = 11 b = 8
Maksimum = 11
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 8 b = 11

E:\>
```

Program 18 memiliki tujuan untuk mencontohkan sebuah program yang mengandung sebuah prosedur dan fungsi, selain itu, program ini bertujuan juga untuk mendemonstrasikan penukaran (swap) nilai yang di input. Program ini mempunyai beberapa subprogram/function seperti maxab yang bertipe int dan tukar yang bertipe void. Maxab bertujuan untuk mencari nilai maksimum dari a dan b dengan penggunaan operator ternary (?), jika $A \geq b$ maka a adalah nilai maksimum, sedangkan jika false, maka b adalah nilai maksimum.

Lalu pada void tukar terdapat operasi swap atau penukaran nilai a dengan b, syntax:

int temp; temp=a; a=b; b=temp;

Setelah itu, pada fungsi utama, digunakan scanner untuk mempermudah penginputan data. Disini akan di input dua bilangan yang bertipe data integer. Selanjutnya, maxab akan dijalankan untuk mencari nilai maksimum dari kedua bilangan, dan terakhir void tukar akan dijalankan untuk menukar nilai atau value dari a dan b.

19. Code Program

```
E:\>javac Tempair.java

E:\>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 9
Wujud air cair
9
E:\>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = -18
Wujud air beku
-18
E:\>java Tempair
Contoh IF tiga kasus
Temperatur (der. C) = 110
Wujud air uap/gas
110
E:\>
```

Program 19 bertujuan untuk mencontohkan penggunaan IF tiga kasus (Program pengolahan wujud air). Scanner digunakan untuk mempermudah penginputan data. Kondisi IF yaitu jika nilai T yang diinput kurang dari 0, maka akan tercetak wujud air beku, selanjutnya jika Nilai t yang di input tidak sama dengan nol dan kurang dari atau sama dengan 100, maka system mencetak wujud air cair, terakhir jika nilai T yang diinput melebihi 100, maka akan dicetak wujud air gas.

3. Kasus : Buat Flowchart dan Program menggunakan bahasa java untuk Konversi Waktu (Jam:Menit:Detik) dari masukan detik!

Menampilkan Waktu dalam format jam:menit:detik.

Code Program

```
E:\>javac Konversiwaktu.java

E:\>java Konversiwaktu
Masukkan Angka =
3991
konversi dari nilai : 3991 adalah :
31Detik = 6Menit = 1Jam = 1JamSekarang
E:\>_
```

Program ini saya buat agar bisa mengkonversi nilai yang di-input menjadi satuan time/waktu yang terorganisir. Pertama-tama saya memasukkan paket scanner dengan `import java.util.Scanner;` gunanya adalah untuk mempermudah penulisan. Kemudian saya buat class public dengan nama Program. Maksud dari public berarti bisa diakses dimana saja.

Setelah membuat class, saya langsung membuat main program dengan: `public static void main(String [] args){` tanpa elemen tersebut Program yang dibuat tidak akan berjalan dengan baik.

Selanjutnya saya deklarasikan beberapa variable yang berkaitan dengan waktu dengan tipe data int.

Setelah itu saya membuat scanner baru dengan nama input. Scanner merupakan class yang menyediakan fungsi-fungsi untuk mengambil input dari keyboard. Kemudian saya mendeklarasikan tipe data untuk beberapa objek seperti jam, menit, detik, dan Conversion dengan tipe data int.

Lalu saya membuat menu untuk menginput data dengan:

`System.out.print("Masukkan Nilai yang ingin Dikonversi ke waktu : ");`
System.out.print digunakan untuk mencetak data yang dimasukkan ke dalam (), untuk varian System.out.println, mencetak datanya dipisah dengan baris baru/ new line.

Selanjutnya, saya membuat perintah untuk memasukkan nilai dengan syntax:

`totalDetik = input.nextInt();`

Dengan tipe data integer untuk data yang akan diinput. Saat di run, maka kita akan diperintahkan untuk memasukkan nilai pertama.

Selanjutnya adalah operasi penghitungan serta konversi dari nilai yang di input ke bentuk waktu dengan syntax:

`detikSekarang=totalDetik%60;`

`totalMenit=totalDetik/60;`

`menitSekarang=totalMenit%60;`

`totalJam=totalMenit/60;`

`jamSekarang = totalJam%24;`

Terakhir adalah menampilkan output dengan perintah system.out.println. output akan menampilkan hasil konversi dari nilai yang di input ke dalam bentuk waktu.

Flowchart:

