

Nama : Ria Fitriani Palallo
Kelas : B2
NIM : 13020200179

Laporan Tugas 2

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

1) Program 1

Output :

```
run:
f : 20.0
f11: 10.0
BUILD SUCCESSFUL (total time: 1 second)
```

Penjelasan :

Program tersebut bertujuan untuk menampilkan atau mencetak isi/nilai dari variable f dan f11.

- **public class Asgdll** → dibuat sebuah class dengan nama Asgdll yang menggunakan access specifier public yang mengindikasikan bahwa class tsb. memiliki hak akses bebas ke class lain.
- **public static void main(String[] args)** → merupakan method utama dalam class Asgdll.
- **float f= 20.0f;** → dideklarasikan sebuah variable f bertipe data float(menampung bilangan pecahan) yang telah diinisialisasi/diberikan nilai. Untuk tipe data float, setelah nilai perlu ditambahkan notasi 'f' atau 'F'.
- **double f11;** → deklarasi sebuah variable bertipe data double yang diberi nama f11.
- **f11=10.0f;** → merupakan sebuah statement assignment yang dimana variable f11 diinisialisasi nilainya menjadi 10.0
- **System.out.println("f : "+f + "\nf11: "+f11);** → Baris kode tsb. merupakan sebuah perintah dalam java untuk menampilkan atau mencetak isi dari variabel-variabel yang telah dideklarasikan yaitu variabel f dan f11.

2) Program 2

Output :

```
run:
hello
Ini nilai i :5
BUILD SUCCESSFUL (total time: 0 seconds)
```

Penjelasan :

Program tersebut bertujuan untuk menampilkan teks "hello" dan juga menampilkan teks "Ini nilai i : 5" yang diikuti dengan pemanggilan isi dari variabel i yaitu 5.

- **public class Asign** → dibuat sebuah class dengan nama Asgdll yang menggunakan access specifier public yang mengindikasikan bahwa class tsb. memiliki hak akses bebas ke class lain.
- **public static void main(String[] args)** → merupakan method utama dalam class Asign.

- **int i;** → deklarasi sebuah variabel i dengan tipe data integer yang dimana tipe data tsb. dapat menampung bilangan bulat.
- **System.out.print ("hello\n");** → Perintah dalam java untuk mencetak string “hello”.
- **i = 5;** → Statement atau pernyataan assignment yang dimana variable i diisi atau diberi nilai 5.
- **System.out.println ("Ini nilai i : " + i);** → Perintah output untuk mencetak string “Ini nilai i : ” yang diikuti oleh pemanggilan isi variabel i yang dimana isi dari variabel i yaitu 5 juga akan ikut dicetak.
- Perintah **\n** pada **println** merupakan perintah untuk pindah baris.

3) Program 3

Output :

```
run:
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
BUILD SUCCESSFUL (total time: 0 seconds)
```

Penjelasan :

Program tsb. bertujuan untuk mendefinisikan variabel-variabel bertipe data int, short int, long int, dan char yang nilainya telah diinisialisasi serta mencetak teks dan isi/nilai dari variabel-variabel yang dipanggil.

- Tipe data **short int**, **int**, dan **long int** merupakan tipe data numerik yang menampung bilangan bulat. Perbedaan dari ketiga tipe data ini terletak pada range atau jangkauannya. Adapun tipe data **char** dimana tipe data tsb. dapat menampung karakter tunggal yang biasanya didefinisikan dengan tanda petik (‘) di awal dan di akhir karakter serta bisa juga didefinisikan menggunakan nilai ASCII dari suatu karakter.
- **char c = 65;** → merupakan inisialisasi karakter dengan integer. Bilangan 65 merupakan nilai ASCII dari karakter ‘A’ sehingga saat program dijalankan maka yang dicetak adalah hasil dari konversi nilai ASCII ke suatu karakter yaitu ‘A’.
- Perintah **\t** pada **System.out.println ("\t(int) = " + ki);** merupakan escape sequence yaitu karakter yang tidak dapat diketik melalui keyboard dan dianggap karakter tunggal. Output yang dihasilkan berupa sebuah spasi seperti saat menekan tombol tab pada keyboard.

4) Program 4

Output :

```
run:
Contoh membaca dan menulis, ketik nilai integer:
10
11
Nilai yang dibaca : 10BUILD SUCCESSFUL (total time: 13 seconds)
```

Penjelasan :

Program tsb. bertujuan untuk membaca masukan berupa integer dengan menggunakan Class Scanner kemudian mencetaknya.

- **import java.util.Scanner;** → Import kelas Scanner ke dalam java. Di dalam paket **java** terdapat sub-paket **util**. Di dalam **util** terdapat file class **Scanner** yang didalamnya terdapat beberapa fungsi yang digunakan untuk membaca masukan baik itu masukan yang bertipe data primitif maupun yang bertipe data class.
- **int a;** → Deklarasi variabel **a** dengan tipe integer yang akan menyimpan nilai masukan
- **Scanner masukan;** → Deklarasi variabel **masukan** dengan tipe data class Scanner.
- **masukan = new Scanner(System.in);** → Pembuatan objek dari class Scanner.
- **a = masukan.nextInt();** → Objek **masukan** digunakan untuk mengakses fungsi **nextInt()** yang dimana fungsi ini digunakan sebagai perintah membaca input angka yang bertipe integer. Lalu masukan tersebut akan disimpan di variabel **a**.
- Saat menambah perintah **masukan.nextInt()** pada program, hasil outputnya meminta user untuk menginputkan bilangan bulat. Namun, inputan tersebut tidak disimpan dalam sebuah variabel karena tidak menyertakan statement assignment sehingga inputan user tidak dapat dicetak kembali.

5) Program 5

Output :

```
run:
hello
baca 1 karakter : R
baca 1 bilangan : 1
R
1
bye
BUILD SUCCESSFUL (total time: 20 seconds)
```

Penjelasan :

Program tsb. bertujuan untuk membaca masukan berupa integer dan char dengan menggunakan metode **readline()** kemudian mencetaknya kembali.

- **import java.io.BufferedReader;** → import kelas **BufferedReader** dari paket **io**
- **import java.io.IOException;** → import kelas **IOException** dari paket **io**
- **import java.io.InputStreamReader;** → import kelas **InputStreamReader** dari paket **io**
- **throws IOException** → mekanisme penanganan error pada kelas **BufferedReader** yang bisa terjadi karena data tidak ditemukan.
- **BufferedReader dataIn = new BufferedReader(new InputStreamReader(System.in));** → perintah membuat objek **dataIn** dari kelas **BufferedReader**

- **cc =dataIn.readLine().charAt(0);** → Perintah untuk memanggil fungsi readLine() agar dapat membaca masukan karakter yang kemudian disimpan ke dalam variabel cc
- **bil =Integer.parseInt(dataIn.readLine());** → Perintah untuk membaca masukan bilangan. Pada kelas BufferedReader, fungsi tidak bisa membaca masukan berupa bilangan. Maka dari itu, masukan bilangan harus lebih dulu disimpan dalam bentuk string kemudian dikonversi menjadi bilangan agar masukannya dapat terbaca.

6) Program 6

Output :

```
run:
5.0
6.0
2
3.2000000047683716
53
53.0
53.0
3
3.14
BUILD SUCCESSFUL (total time: 2 seconds)
```

Penjelasan :

Program tersebut bertujuan untuk mengubah nilai dari variabel dengan tipe data primitif ke dalam tipe data yang lain.

- Untuk mengubah nilainya, harus dideklarasikan terlebih dahulu variabel bertipe data yang nilainya telah diinisialisasi contohnya **int a=5, char g='5'**. Misalnya ingin mengubah nilai variabel a dengan tipe data int ke dalam tipe data float, maka cukup ketikkan perintah **System.out.println((float)a);** dan nilai variabel a akan dikonversi ke dalam tipe data float sehingga pada output nilai a mencetak 5.0. Begitupun dengan nilai variabel g yang bertipe data char jika ingin diubah nilainya ke dalam tipe data float maka perintah yang digunakan yaitu **System.out.println((float)g);**
- Format untuk mencetak casting dengan menggunakan tipe data primitif :
System.out.println((tipe data baru)variabel yang akan diubah nilainya);

7) Program 7

Output :

```
run:
a : 67
k : 45.0
d : 100.0
n : 9
m : 5
l : 3.2
k : 67.0
c : 9.0
l : 3.2
BUILD SUCCESSFUL (total time: 2 seconds)
```

Penjelasan :

Program tsb. bertujuan untuk mengubah nilai dari variabel dengan tipe data Class ke dalam tipe data yang lain dan begitupun sebaliknya.

- Format statement untuk konversi nilai bertipe data Class ke tipe data primitif

Contoh :

a = Integer.parseInt(n); → konversi String ke Integer

(output yang tercetak saat variabel a dipanggil adalah 67 yang dimana nilai tsb. merupakan hasil dari nilai variabel n bertipe string yang telah diubah nilainya ke dalam tipe data integer)

ket :

a = variabel untuk menyimpan nilai hasil casting

Integer.parseInt = metode parseInt dalam kelas Integer

n = variabel yang nilainya ingin diubah ke tipe data lain.

- Format statement untuk konversi nilai bertipe data primitif ke tipe data Class

Contoh :

n = String.valueOf(b); → konversi Integer ke String

(output yang tercetak saat variabel n dipanggil adalah 9 yang dimana nilai tsb. merupakan hasil dari nilai variabel b yang bertipe data integer yang telah diubah nilainya ke dalam tipe data string)

ket:

n = variabel untuk menyimpan nilai hasil casting

String.valueOf(b) = ekspresi untuk menkonversi dari suatu tipe data ke dalam tipe data string

b = variabel yang nilainya ingin diubah ke tipe data lain.

8) Program 8

Output :

```
run:
x = 1
y = 2
hasil ekspresi = (x<y)?x:y = 1BUILD SUCCESSFUL (total time: 0 seconds)
```

Penjelasan :

Program tsb. bertujuan untuk mencetak hasil ekspresi yang menggunakan operator kondisional.

- **((x < y) ? x : y)** → merupakan operator kondisional yang akan mengecek apakah nilai x lebih kecil dari nilai y dan akan mencetak statement sesuai dengan hasil kondisi tersebut (true [mencetak nilai x] atau false [mencetak nilai y]). Dalam hal ini nilai x=1 dan nilai y=2, kemudian dicek apakah 1 < 2? Jawabannya true maka akan mencetak nilai dari variabel x sehingga output yang dicetak yaitu 1.

9) Program 9

Output :

```
run:
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) = 3BUILD SUCCESSFUL (total time: 2 seconds)
```

Penjelasan :

Program tsb. bertujuan untuk menampilkan hasil pembagian integer dan juga menampilkan hasil pembagian casting.

- **System.out.print ("x/y (format integer) = "+ x/y);**
System.out.print ("\nx/y (format float) = "+ x/y);
→ Saat menjalankan perintah diatas maka hasil x/y yang tercetak pada output adalah nilai 0. Sedangkan saat dilakukan perhitungan secara manual maka nilai dari $x/y = \frac{1}{2} = 0.5$. Output mencetak nilai 0 karena tipe data dari kedua variabel tsb adalah integer yang dimana hanya dapat menampung bilangan bulat saja dan tidak dapat menampung bilangan desimal. Itulah mengapa nilai 0.5 ini tidak tercetak pada output.
- Agar hasil output tidak nol, maka harus membuat variabel-variabel yang baru dengan tipe data float kemudian menyalin nilai dari variabel x dan y kedalam variabel baru tsb. (fx dan fy). Setelah itu, deklarasikan operasi pembagian dengan menggunakan variabel fx dan fy.
- Untuk menampilkan hasil pembagian casting maka digunakan perintah :
System.out.print ("\nfloat(x)/float(y) (format integer) = "+(float)x/(float)y);
System.out.print ("\nfloat(x)/float(y) (format float) = "+(float)x/(float)y);

10) Program 10

Output :

```
run:
Hello
Hello World
Welcome
BUILD SUCCESSFUL (total time: 2 seconds)
```

Penjelasan :

Program tsb. bertujuan untuk mencetak teks dan mencetak perintah enter atau ganti baris.

- **System.out.print("Hello");** → Perintah mencetak teks "Hello"
- **System.out.print("\nHello ");** → Untuk mencetak perintah enter atau ingin mengganti baris sebelum teks dicetak maka menambahkan "\n" pada awal teks. Jadi output pertama dan output kedua berada di baris yang berbeda.
- **System.out.println("World");** → Perintah ganti baris juga dapat digunakan dengan menambahkan "\n" setelah **print**. Perintah ini digunakan saat ingin mengganti baris setelah output dicetak. Jadi, output teks World dan Welcome berbeda baris sedangkan untuk output teks World dan Hello berada di baris yang sama.

11) Program 11

Output :

```
run:
Nilai i : 5
Nilai j : 3
BUILD SUCCESSFUL (total time: 2 seconds)
```

Penjelasan :

Program tersebut bertujuan untuk menampilkan efek perubahan dari penggunaan operator increment (++).

- Dideklarasikan variabel $i = 3$ dan variabel $j = i++$
- Karena perintah untuk mencetak hanya 1 maka pengoperasian dari $(++i)$ dan (j) dilakukan bersamaan. Output mencetak nilai $j = 3$ karena sebelumnya dideklarasikan bahwa isi dari variabel j adalah operator post-increment $(i++)$ yang dimana variabel j akan mengambil terlebih dahulu nilai i lalu variabel i ditambah 1 sehingga nilai j yang dicetak adalah 3. Sekarang i bernilai 4 karena telah dilakukan post-increment.
- Output mencetak nilai $i = 5$ karena perintah yang dipanggil yaitu pre-increment $(++i)$ yang dimana nilai i akan ditambah 1 lebih dulu sehingga nilai $i = 4 + 1 = 5$. Jadi, output akan mencetak nilai $i : 5$

12) Program 12

Output :

```
run:
n = 10
x = 1
y = 2
n & 8 = 8
x & ~ 8 = 1
y << 2 = 8
y >> 3 = 0
BUILD SUCCESSFUL (total time: 7 seconds)
```

Penjelasan :

Program tersebut bertujuan agar memudahkan user dalam menyelesaikan operasi yang menggunakan operator bitwise.

- Penyelesaian operasi bitwise ini dilakukan pada level bit
- $n \& 8 = 8 \rightarrow$ menggunakan operator bitwise AND yang membandingkan 2 bit dan menghasilkan nilai 1 hanya jika kedua bit bernilai 1. ($1010 \text{ AND } 1000 = 1000 = 8$)
- $x \& \sim 8 = 1 \rightarrow$ menggunakan operator bitwise AND, bit dari bilangan bulat 8 diinvert 1000 menjadi 0111. ($1 \text{ AND } 0111 = 0001 = 1$)
- $y \<\< 2 = 8 \rightarrow$ menggunakan operator bitwise shift left yang memindahkan susunan bit ke kiri dan membuang bit paling kiri serta memberikan nilai 0 pada bit paling kanan. (10 digeser ke kiri sebanyak 2 bit sehingga menjadi 1000 = 8)
- $y \>\> 3 = 0 \rightarrow$ menggunakan operator bitwise shift right yang memindahkan susunan bit ke kanan dan membuang bit paling kanan serta memberikan nilai 0 pada bit paling kiri. (10 digeser ke kanan sebanyak 3 bit sehingga menjadi 0000 = 0)

13) Program 13

Output :

```
run:
i = 3
j = □
i & j = 0
i | j = 7
i ^ j = 7
81.0
~i = -4
BUILD SUCCESSFUL
```

Penjelasan :

Program tersebut bertujuan agar memudahkan user dalam menyelesaikan operasi yang memakai beberapa operator relational dan bitwise.

- $i = 3 \rightarrow$ menampilkan nilai 3 (integer) karena isi variabel i yang bertipe data char dipanggil kembali menggunakan metode casting yang dimana akan mengubah nilai atau isi dari variabel i ke dalam tipe data Integer.
- $j = \square$
 \rightarrow menampilkan simbol yang tidak diketahui karena variabel j bertipe data char yang hanya dapat menampung data karakter dimana nilai karakter harus diapit dengan tanda petik tunggal ('') sedangkan pada kode program, inisialisasi nilai dari variabel tsb. tidak menggunakan tanda petik tunggal sehingga saat variabel j dipanggil (tanpa casting) maka output akan mencetak simbol yang tidak diketahui.
- $i \& j = 0 \rightarrow$ menggunakan operator bitwise AND
 $i | j = 7 \rightarrow$ menggunakan operator bitwise inclusive OR
 $i \wedge j = 7 \rightarrow$ menggunakan operator bitwise exclusive OR
- $81.0 \rightarrow$ Hasil dari perintah $\text{Math.pow}(i, j) \rightarrow$ Class Math memiliki method $\text{pow}(a,b)$ untuk pemangkatan. Jadi $3^4=81.0$
- $\sim i = -4 \rightarrow$ Hasil invert biner dari 00000011 menjadi 11111100 = -4

14) Program 14

Output :

```
run:
true
false
true
true
true
BUILD SUCCESSFUL (total time: 0 seconds)
```

Penjelasan :

Program tersebut bertujuan untuk mencetak hasil dari operasi menggunakan operator bitwise dan operator conditional.

- **`if (true && true){ System.out.println(true && true);}`**
 \rightarrow Perintah untuk mencetak hasil dari kondisi pada statement jika kondisi if bernilai true. Operator $\&\&$ termasuk ke dalam operator logical dan merupakan short-circuit operator. Short circuit operator disini maksudnya adalah jika operator yang kiri bernilai false, maka operator yang kanan tidak akan di jalankan, karena sudah pasti

hasilnya akan false. Kondisi if bernilai true sehingga dijalankan statement di dalamnya yang menghasilkan output bernilai true.

- **if (true & true) { System.out.println(true & false); }**
→ Perintah untuk mencetak hasil dari kondisi dalam statement jika kondisi if bernilai true. Operator & bisa berfungsi sebagai conditional operator atau bisa juga berfungsi sebagai operator bitwise pada bilangan. Jika berfungsi sebagai conditional operator, maka tidak berlaku fungsi short-circuit, akibatnya operator kiri dan kanan pasti akan di evaluasi apapun kondisinya. Kondisi if bernilai true sehingga dijalankan statement di dalamnya yang menghasilkan output false.
- **if (true) { System.out.println(true); }**
→ Perintah mencetak hasil true karena kondisi if selalu bernilai true.
- **if (true || true){ System.out.println(true); }**
→ Perintah untuk mencetak hasil dari kondisi dalam statement jika kondisi if bernilai true. Operator || termasuk ke dalam operator logical dan merupakan short-circuit operator. Short-circuit operator disini maksudnya adalah jika kondisi yang kiri bernilai true, maka operasi yang kanan tidak dijalankan, sebab sudah pasti hasilnya true. Kondisi if bernilai true, maka dieksekusi statement di dalamnya dan menghasilkan output bernilai true.
- **if (true|false) { System.out.println(true|false); }**
→ Perintah untuk mencetak hasil dari kondisi dalam statement jika kondisi if bernilai true. Operator | berfungsi sebagai conditional operator atau bisa juga berfungsi sebagai bit pada bilangan. Jika berfungsi sebagai conditional operator, maka tidak berlaku fungsi short-circuit, akibatnya operator kiri dan kanan pasti akan di evaluasi apapun kondisinya. Kondisi if bernilai true, maka dieksekusi statement di dalamnya dan menghasilkan output bernilai true.

15) Program 15

Output :

```
run:
Nilai e = 10
Nilai k = 0
Nilai k = 4BUILD SUCCESSFUL (total time: 2 seconds)
```

Penjelasan :

Program tsb. bertujuan untuk mencetak hasil dari operasi yang menggunakan operator ternary.

- **Nilai e = 10** → Isi variabel e didapat dari operasi yang melibatkan operator ternary. **((int)c > (int)d) ? c: d** artinya apakah nilai variabel c yaitu 8 lebih besar dari nilai variabel d yaitu 10? Jawabannya false maka nilai yang diisikan ke variabel e adalah nilai dari variabel d yaitu 10.
- **Nilai k = 0** → Isi variabel k didapat dari operasi yang melibatkan operator ternary. **((i>j) ? i: j)** artinya apakah nilai variabel i yaitu 0 lebih besar dari nilai variabel j yaitu 0? Jawabannya false maka nilai yang diisikan ke variabel k adalah nilai dari variabel j yaitu 0.
- **Nilai k = 4** → Isi variabel k didapat dari operasi yang melibatkan operator ternary. **((i++>j++) ? i: j)** artinya apakah hasil dari post-increment i (nilai i dilakukan evaluasi lebih dulu sebelum ditambah 1) yaitu 2 lebih besar dari post-increment j (nilai j dilakukan evaluasi sebelum ditambah 1) yaitu 3? Jawabannya false maka nilai yang

diisikan ke variabel k adalah nilai dari variabel j yang sekarang (nilai ditambah 1 karena terdapat post-increment) yaitu 3+1=4. Jadi output yang dicetak adalah nilai 4.

16) Program 16

Output :

```
OPERASI LOGICAL BOOL
Bool1 && Bool2 = false
Bool1 || Bool2 = true
!Bool1 = false
Bool1 ^ Bool2 = true

OPERASI NUMERIK
i + j = 7
i - j = 3
i / j = 2
i * j = 10
i / j (Pembagian bulat) = 2
i % j (Sisa modulo) = 1

OPERASI NUMERIK
x + y = 10.0
x - y = 0.0
x / y = 1.0
x * y = 25.0

OPERASI RELATIONAL NUMERIK
(i == j) = false
(i != j) = true
(i < j) = false
(i > j) = true
(i <= j) = false
(i >= j) = true

OPERASI RELATIONAL NUMERIK
(x != y) = false
(x < y) = false
(x > y) = false
(x <= y) = true
(x >= y) = true
BUILD SUCCESSFUL (total time: 1 second)
```

Penjelasan :

Program tersebut bertujuan untuk mencetak hasil dari pengoperasian variabel-variabel bertipe dasar.

- **Bool1 && Bool2 = false** → menggunakan operator logical AND, true && false menghasilkan false karena salah satu operandnya bernilai false
- Bool1 || Bool2 = true** → menggunakan operator logical OR, true || false menghasilkan true karena salah satu operandnya bernilai true
- ! Bool1 = false** → menggunakan operator logical negation, ! true menghasilkan nilai false karena operator ini akan memberikan nilai kebalikan dari operandnya.
- Bool1 ^ Bool2 = true** → menggunakan operator logical XOR, true^false menghasilkan true karena operator ini akan memberikan nilai true hanya jika kedua operand berbeda nilainya

- **i / j (Pembagian bulat) = 2** → Menggunakan operator pembagian. Menghasilkan nilai 2 karena 5 dibagi 2 menghasilkan nilai 2. Sebenarnya hasilnya berupa bilangan desimal 2.5 namun variabel hsl bertipe integer yang hanya dapat menyimpan bilangan bulat.
- **i % j (Sisa modulo) = 1** → Menggunakan operator modulus yang menyimpan sisa hasil bagi. Menghasilkan nilai 1 karena 1 adalah sisa hasil bagi dari 5/2
- **(i == j) = false** → Menggunakan operator relational. (i==j) bernilai false karena 5 tidak sama dengan 2. Variabel penampung hasil dari operasi bertipe data Boolean sehingga hanya mencetak output false atau true.
 - (i != j) = true** → (i!=j) bernilai true karena 5 tidak sama dengan 2.
 - (i < j) = false** → (i<j) bernilai false karena 5 (nilai i) tidak kurang dari 2 (nilai j)
 - (i > j) = true** → (i>j) bernilai true karena 5 (nilai i) lebih besar dari 2 (nilai j)
 - (i <= j) = false** → (i<=j) bernilai false karena 5 (nilai i) tidak kurang ataupun tidak sama dengan 2 (nilai j).
 - (i >= j) = true** → (i>=j) bernilai true karena 5 (nilai i) lebih besar atau sama dengan 2 (nilai j).

2. Tugas Praktek : Praktek Program Java : Standar IO dan Struktur Kontrol

1) Program 1

Output :

```
run:
Baca string dan Integer:
masukkan sebuah string: Ria Fitriani Palallo
String yang dibaca : Ria Fitriani PalalloBUILD SUCCESS
```

Penjelasan :

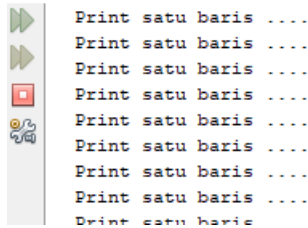
Program tersebut bertujuan untuk membaca masukan berupa string dengan menggunakan metode `readLine()` kemudian mencetaknya kembali.

- **import java.io.BufferedReader;** → import kelas `BufferedReader` dari paket `io`
- **import java.io.IOException;** → import kelas `IOException` dari paket `io`
- **import java.io.InputStreamReader;** → import kelas `InputStreamReader` dari paket `io`
- **import javax.swing.*;** → import semua kelas dari paket `swing`
- **throws IOException** → mekanisme penanganan error pada kelas `BufferedReader` yang bisa terjadi karena data tidak ditemukan.
- **BufferedReader datAIn = new BufferedReader(new InputStreamReader(System.in));** → perintah membuat objek `datAIn` dari kelas `BufferedReader`
- **str =datAIn.readLine();** → Perintah untuk memanggil fungsi `readLine()` agar dapat membaca masukan string yang kemudian disimpan ke dalam variabel `str`.

2) Program 2

Output :

Output - tugas2no2 (run)



```
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 ....
```

Penjelasan :

Program tersebut bertujuan untuk mencetak statement di dalam loop WHILE yang dilakukan secara berulang-ulang tanpa henti dikarenakan kondisi perulangan bernilai selalu true. Suatu statement dari loop akan dijalankan secara terus menerus selama kondisi bernilai true. Looping dapat dihentikan jika perintah break ditambahkan di akhir statement atau dapat juga dihentikan dengan mengubah ekspresi dari kondisi perulangannya.

3) Program 3

Output :

```
run:
Contoh IF satu kasus
Ketikkan suatu nilai integer : 20

Nilai a positif 20BUILD SUCCESSFUL (total time: 40 seconds)
```

Penjelasan :

Program tersebut bertujuan untuk membaca masukan berupa integer dengan menggunakan Class Scanner kemudian mencetak nilai masukannya jika kondisi if terpenuhi

- **import java.util.Scanner;** → Import kelas Scanner ke dalam java. Di dalam paket **java** terdapat sub-paket **util**. Di dalam **util** terdapat file class **Scanner** yang didalamnya terdapat beberapa fungsi yang digunakan untuk membaca masukan baik itu masukan yang bertipe data primitif maupun yang bertipe data class.
- **Scanner masukan = new Scanner(System.in);** → Pembuatan objek dari class Scanner.
- **a = masukan.nextInt();** → Objek **masukan** digunakan untuk mengakses fungsi **nextInt()** yang dimana fungsi ini digunakan sebagai perintah membaca input angka yang bertipe integer. Lalu masukan tersebut akan disimpan di variabel **a**.
- **if (a >= 0)**
 System.out.print ("\nNilai a positif "+ a);
→ Terdapat pengkondisian if yang dimana jika ekspresi terpenuhi, maka akan menjalankan statementnya. Jika nilai masukan yang disimpan dalam variabel **a** bernilai lebih dari atau sama dengan 0, maka masukan tsb. akan dicetak.

4) Program 4

Output :

```
run:
Contoh IF dua kasus
Ketikkan suatu nilai integer :4
Nilai a positif 4
BUILD SUCCESSFUL (total time: 7 seconds)

run:
Contoh IF dua kasus
Ketikkan suatu nilai integer :-4
Nilai a negatif -4
BUILD SUCCESSFUL (total time: 7 seconds)
```

Penjelasan :

Program tersebut bertujuan untuk membaca masukan berupa nilai integer dengan menggunakan Class Scanner lalu mencetak nilai a positif ataupun nilai a negatif tergantung dari kondisi yang terpenuhi.

- **Scanner masukan = new Scanner(System.in);** → Pembuatan objek dari class Scanner.
- **a = masukan.nextInt();** → Objek **masukan** digunakan untuk mengakses fungsi **nextInt()** yang dimana fungsi ini digunakan sebagai perintah membaca input angka yang bertipe integer. Lalu masukan tersebut akan disimpan di variabel **a**.
- **if (a >= 0){**
 System.out.println ("Nilai a positif "+ a);
 }
 else{
 System.out.println ("Nilai a negatif "+ a);
 }
→ Terdapat statement if-else. Nilai masukan pada variabel a akan dicek terlebih dahulu apakah nilai a >=0. Jika benar maka statement if akan dijalankan yaitu mencetak 'Nilai a positif [nilai a]'. Namun jika bernilai salah atau dengan kata lain nilai a < 0 maka statement else akan dijalankan yaitu dengan mencetak 'Nilai a negative [nilai a]'.

5) Program 5

Output :

```
run:
Contoh IF tiga kasus
Ketikkan suatu nilai integer :5
Nilai a positif 5

run:
Contoh IF tiga kasus
Ketikkan suatu nilai integer :0
Nilai Nol 0

run:
Contoh IF tiga kasus
Ketikkan suatu nilai integer :-5
Nilai a negatif -5
```

Penjelasan :

Program tersebut bertujuan untuk membaca masukan berupa nilai integer dengan menggunakan Class Scanner lalu mencetak isi dari statement tergantung dari kondisi yang terpenuhi.

- **Scanner masukan = new Scanner(System.in);** → Pembuatan objek dari class Scanner.
- **a = masukan.nextInt();** → Objek **masukan** digunakan untuk mengakses fungsi **nextInt()** yang dimana fungsi ini digunakan sebagai perintah membaca input angka yang bertipe integer. Lalu masukan tersebut akan disimpan di variabel **a**.
- Program tsb. menggunakan IF dengan 3 kasus atau kondisi. Kondisi pertama yaitu jika nilai a > 0 maka akan mencetak 'Nilai a positif [nilai a]'. Kondisi kedua yaitu jika a = 0

maka akan mencetak 'Nilai Nol [nilai a]'. Kondisi ketiga yaitu jika nilai a tidak lebih dari 0 dan nilai a tidak sama dengan 0 atau dengan kata lain jika nilai $a < 0$ maka akan mencetak 'Nilai a negatif [nilai a]'.

6) Program 6

Output :

```
run:
true
benar
BUILD SUCCESSFUL (total time: 0 seconds)
```

Penjelasan :

Program tersebut bertujuan untuk mencetak statement berdasarkan kondisi (ekspresi kondisional dengan boolean) yang terpenuhi.

- Isi dari variabel bool yaitu nilai true.
- Terdapat 2 pengkondisian If-Else sehingga ada 2 output yang tercetak. Output pertama mencetak 'true' karena nilai dari bool itu true sehingga kondisi `if(bool)` terpenuhi dan langsung menjalankan statementnya. Output kedua mencetak 'benar' karena saat dicek kondisi `if(!bool)` akan menghasilkan nilai false yang dimana jika kondisinya false maka statement di dalamnya tidak akan dijalankan dan otomatis akan langsung menjalankan perintah `else` sehingga tercetaklah output 'benar'.

7) Program 7

Output :

<pre>run: Ketikkan sebuah huruf, akhiri dengan RETURN a Yang anda ketik adalah a BUILD SUCCESSFUL (total time: 15 seconds)</pre>	<pre>run: Ketikkan sebuah huruf, akhiri dengan RETURN r Yang anda ketik adalah huruf mati BUILD SUCCESSFUL (total time: 18 seconds)</pre>
--	---

Penjelasan :

Program tersebut bertujuan untuk membaca masukan berupa karakter dengan menggunakan Class Scanner lalu mencetak statement jika nilai case sama dengan nilai masukan.

- **Scanner masukan = new Scanner(System.in);** → Pembuatan objek dari class Scanner.
- **cc = masukan.next().charAt(0);** → Objek **masukan** digunakan untuk mengakses fungsi **next().charAt(0)** yang dimana fungsi ini digunakan sebagai perintah membaca inputan karakter yang bertipe char. Lalu isi masukan tersebut akan disimpan di variabel **cc**
- Terdapat switch case yang ekspresinya merupakan isi dari variabel cc. Ekspresi akan dibandingkan dengan nilai-nilai pada setiap case. Jika ditemukan ada nilai yang sama maka statementnya dan seluruh statement dibawahnya akan dijalankan. Namun, jika di akhir statement dideklarasikan perintah **break;** maka yang dieksekusi hanya statement pada case yang nilai casenya sama dengan nilai ekspresi. Adapun jika nilai ekspresi tidak ada yang sama dengan nilai case, maka statement default akan dieksekusi.
- Misalnya isi dari variabel cc yang saya masukkan yaitu 'a'. Maka yang dieksekusi adalah statement dari case pertama sehingga menghasilkan output 'Yang anda ketik adalah a'.

8) Program 8

Output :

```
run:
Jari-jari lingkaran = 2
Luas lingkaran = 12.566
Akhir program
BUILD SUCCESSFUL (total time: 20 secon
```

Penjelasan :

Program tersebut bertujuan untuk membaca masukan berupa bilangan desimal dengan menggunakan Class Scanner yang disimpan pada variabel `r` dan juga menghitung luas lingkaran.

- **final float PHI = 3.1415f;** → Deklarasi konstanta di Java dengan menggunakan keyword `final`
- **Scanner masukan = new Scanner(System.in);** → Pembuatan objek dari class `Scanner`.
- **r = masukan.nextFloat();** → Objek `masukan` digunakan untuk mengakses fungsi `nextFloat()` yang dimana fungsi ini digunakan sebagai perintah membaca inputan berupa bilangan pecahan atau desimal yang bertipe `float`. Lalu isi masukan tersebut akan disimpan di variabel `r`
- **System.out.print ("Luas lingkaran = "+ (PHI * r * r)+"\n");**
→ Perintah untuk mencetak 'Luas Lingkaran = ' dan hasil dari operasi aritmetika untuk menghitung luas lingkaran

9) Program 9

Output :

<pre>run: Maksimum dua bilangan : Ketikkan dua bilangan, pisahkan dg RETURN : 10 1 Ke dua bilangan : a = 10 b = 1 Nilai a yang maksimum 10 BUILD SUCCESSFUL (total time: 22 seconds)</pre>	<pre>run: Maksimum dua bilangan : Ketikkan dua bilangan, pisahkan dg RETURN : 1 10 Ke dua bilangan : a = 1 b = 10 Nilai b yang maksimum: 10 BUILD SUCCESSFUL (total time: 7 seconds)</pre>
--	--

Penjelasan :

Program tersebut bertujuan untuk membaca 2 masukan berupa bilangan integer dengan menggunakan Class Scanner lalu menentukan salah satu dari kedua nilai masukan tsb. yang merupakan nilai maksimum berdasarkan seleksi dari pengkondisian `if-else`.

- **Scanner masukan = new Scanner(System.in);** → Pembuatan objek dari class `Scanner`.
- **a=masukan.nextInt();** → Objek `masukan` mengakses fungsi `nextInt()` yang berfungsi untuk membaca masukan berupa bilangan bulat atau integer. Lalu isi masukan akan disimpan di variabel `a`
b=masukan.nextInt(); → Objek `masukan` mengakses fungsi `nextInt()` yang berfungsi untuk membaca masukan berupa bilangan bulat atau integer. Lalu isi masukan akan disimpan di variabel `b`
- Dalam program ini, Jika saya menginputkan nilai `a = 10` dan nilai `b = 1` maka untuk menentukan nilai maksimum, kedua nilai masukan akan diseleksi di kondisi `if(a>=b)`.

Kondisi bernilai benar karena 10 lebih besar dari 1 sehingga statement 'Nilai a yang maksimum 10' dicetak.

- Saat saya menginputkan nilai a = 1 dan nilai b=10 maka kedua masukan akan diseleksi di kondisi if(a>=b). Kondisi bernilai salah karena 1 lebih kecil dari 10 sehingga langsung mengeksekusi statement else yaitu mencetak 'Nilai b yang maksimum 10'.

10) Program 10

Output :

```
run:
Baca N, print 1 s/d N N = 4
1
2
3
4
Akhir program
```

Penjelasan :

Program tersebut bertujuan untuk membaca masukan berupa bilangan integer dengan menggunakan Class Scanner yang disimpan di dalam variabel N kemudian mencetak 1 s/d N dengan perulangan FOR.

- **N=masukan.nextInt();** → Objek masukan mengakses fungsi **nextInt()** yang berfungsi untuk membaca masukan berupa bilangan bulat atau integer. Lalu isi masukan akan disimpan di variabel N
- **for (i = 1; i <= N; i++){**
 System.out.println (i); }
→ Nilai i=1, lalu dibandingkan nilai i=1 dengan N=4 apakah 1<=4? Jawabannya true sehingga akan dijalankan perintah untuk mencetak nilai variabel i yaitu 1. Setelah itu, nilai i dinaikkan 1 menjadi i=2. Lalu, dicek lagi apakah 2<=4? Jawabannya true sehingga dijalankan kembali perintah mencetak nilai variabel i yaitu 2. Kemudian nilai i dinaikkan 1, begitu seterusnya sampai kondisi i<=N sudah tidak terpenuhi.

11) Program 11

Objek :

```
run:
Nilai N >0 = 3
Print i dengan ITERATE :
1
2
3
```

Penjelasan :

Program tersebut bertujuan untuk membaca masukan berupa bilangan bulat atau integer yang disimpan di dalam variabel N kemudian mencetak nilai i yaitu 1 s/d N dengan iterate atau iterasi.

- **Scanner masukan = new Scanner(System.in);** → Pembuatan objek dari class Scanner.

- **N=masukan.nextInt();** → Objek masukan mengakses fungsi **nextInt()** yang berfungsi untuk membaca masukan berupa bilangan bulat atau integer. Lalu isi masukan akan disimpan di variabel **N**
- **i = 1;** → Inisialisasi nilai **i**
- **for (;;){** → Masuk ke dalam perulangan for tanpa ekspresi
 - System.out.println(i);** → Mencetak nilai **i**
 - if (i == N)** → Dicek apakah nilai **i** sama dengan **N**? Jika true maka iterasi akan berhenti, namun jika false maka akan mengeksekusi statement else
 - break;**
 - else {** → Akan dieksekusi jika kondisi **i** tidak sama dengan **N**
 - i++;** → Nilai **i** dinaikkan 1
 - }**
- Program akan terus mencetak nilai **i** yang akan terus dinaikkan nilainya sebesar 1 sampai kondisi (**i==N**) bernilai true.

12) Program 12

Output :

```
run:
Nilai N >0 = 3
Print i dengan REPEAT:
1
2
3
```

Penjelasan :

Program tersebut bertujuan untuk membaca masukan berupa bilangan bulat atau integer yang disimpan di dalam variabel **N** kemudian mencetak nilai **i** yaitu 1 s/d **N** dengan menggunakan perulangan do-while.

- **N=masukan.nextInt();** → Objek masukan mengakses fungsi **nextInt()** yang berfungsi untuk membaca masukan berupa bilangan bulat atau integer. Lalu isi masukan akan disimpan di variabel **N**
- **i = 1;** → Inisialisasi nilai **i**
- **do{**
 - System.out.print (i+"\n");** /* Proses */
 - i++;** /* Next Elmt */
- }**
- while (i <= N);**
 - Perulangan diatas merupakan perulangan do-while yang akan menjalankan statement do-while terlebih dahulu sebelum ekspresi atau kondisi dicek. Pada program tsb. perulangan dijalankan sebanyak 3 iterasi karena setelah iterasi ketiga, ketika ekspresi atau kondisi diperiksa akan menghasilkan nilai false karena nilai **i=4** lebih besar dari nilai **N=3** sehingga perulangan yang mencetak nilai **i** dihentikan.

13) Program 13

Output :

```
run:
Nilai N >0 = 2
Print i dengan WHILE:
1
2
```

Penjelasan :

Program tersebut bertujuan untuk membaca masukan berupa bilangan bulat atau integer yang disimpan di dalam variabel N kemudian mencetak nilai i yaitu 1 s/d N dengan menggunakan perulangan while.

- **i = 1;** → Inisialisasi nilai i
- **while (i <= N){** // Kondisi Perulangan
 System.out.println (i);
 i++;
}

→ Perulangan diatas merupakan perulangan while yang menjalankan statement hanya jika kondisi perulangan bernilai true. Pada program tsb. perulangan dijalankan sebanyak 2 iterasi karena setelah iterasi kedua, ketika ekspresi atau kondisi kembali diperiksa maka akan menghasilkan nilai false karena nilai i=3 lebih besar dari nilai N=2 sehingga perulangan yang mencetak nilai i dihentikan.

14) Program 14

Output :

```
run:
Nilai N >0 = 2
Print i dengan WHILE (ringkas):
1
2
```

Penjelasan :

Program tersebut bertujuan untuk membaca masukan berupa bilangan bulat atau integer yang disimpan di dalam variabel N kemudian mencetak nilai i yaitu 1 s/d N dengan menggunakan perulangan while. Pendeklarasian perulangan while ini dibuat lebih ringkas yang dimana hanya terdiri dari 1 baris statement saja. Statement tersebut berisi perintah untuk mencetak nilai i sekaligus menaikkan atau menambah nilai i sebesar 1.

15) Program 15

Output :

```
run:
Masukkan nilai x (int), akhiri dg 999 : 1
Masukkan nilai x (int), akhiri dg 999 : 2
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 3
```

```
run:
Masukkan nilai x (int), akhiri dg 999 : 999
Kasus kosong
Hasil penjumlahan = 0
```

Penjelasan :

Program tersebut bertujuan untuk membaca nilai masukan berupa bilangan bulat atau integer yang disimpan di dalam variabel x kemudian menjumlahkan nilai-nilai masukan yang dibaca dengan loop FOR dan mencetak hasil penjumlahannya.

- **System.out.print ("Masukkan nilai x (int), akhiri dg 999 : ");**
x = masukan.nextInt();
→ Pertama, user diminta untuk memasukkan elemen pertama yaitu berupa bilangan bulat yang disimpan di variabel x.
- **if (x == 999){**
System.out.print ("Kasus kosong \n");
}
→ Setelah itu, terdapat if-else yang dimana jika nilai masukan sama dengan 999 maka akan mencetak 'Kasus kosong' (seperti pada screenshot output kedua). Dan program akan terhenti.
- **else {** → Jika kondisi if false maka statement else dijalankan.
Sum = x; → Pendefinisian nilai variabel sum = x
for (;){
System.out.print ("Masukkan nilai x (int), akhiri dg 999 : ");
x = masukan.nextInt(); → user diminta untuk menginput elemen kedua
if(x==999) → Jika nilai inputan x == 999 maka akan keluar dari looping
break; → keluar dari looping, user akan masuk kembali ke loop FOR dan diminta untuk menginputkan kembali elemen ketiga lalu diperiksa lagi di kondisi if-else dst.
else{ → Jika kondisi if bernilai false, maka statement else dijalankan
Sum = Sum + x; → Variabel sum yang baru diisi dari hasil sum + x
}
}
}
System.out.println("Hasil penjumlahan = "+ Sum); → Mencetak nilai variabel sum.

16) Program 16

Output :

```
run:
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 tersebut bertujuan untuk membaca nilai masukan berupa bilangan bulat atau integer yang disimpan di dalam variabel x kemudian menjumlahkan nilai-nilai masukan yang dibaca dengan loop do-while dan mencetak hasil penjumlahannya.

- **System.out.print ("Masukkan nilai x (int), akhiri dg 999 : ");**
x = masukan.nextInt();
→ Pertama, user diminta untuk memasukkan elemen pertama yaitu berupa bilangan bulat yang disimpan di variabel x.
- **if (x == 999){**
System.out.print ("Kasus kosong \n");
}
→ Setelah itu, terdapat if-else yang dimana jika nilai masukan sama dengan 999 maka akan mencetak 'Kasus kosong' lalu program akan terhenti.
- **else {** → Jika kondisi if false maka statement else dijalankan.
Sum = 0; → Pendefinisian nilai variabel sum = x
do{ → menjalankan statement di dalamnya, lalu mengecek kondisi looping
Sum = Sum + x; → Variabel sum yang baru diisi dari hasil sum + x
System.out.print ("Masukkan nilai x (int), akhiri dg 999 : ");
x = masukan.nextInt(); → user diminta untuk menginput elemen berikutnya
} while (x != 999); → memeriksa kondisi, jika true maka statement do kembali dijalankan. Namun jika bernilai false maka akan keluar dari perulangan
System.out.println ("Hasil penjumlahan = "+Sum); → Mencetak nilai variabel Sum.

17) Program 17

Output :

```
run:
Masukkan nilai x (int), akhiri dg 999 : 10
Masukkan nilai x (int), akhiri dg 999 : 10
Masukkan nilai x (int), akhiri dg 999 : 999
Hasil penjumlahan = 20
```

Penjelasan :

Program tersebut bertujuan untuk membaca nilai masukan berupa bilangan bulat atau integer yang disimpan di dalam variabel x kemudian menjumlahkan nilai-nilai masukan yang dibaca dengan loop while dan mencetak hasil penjumlahannya.

- **System.out.print ("Masukkan nilai x (int), akhiri dg 999 : ");**
x = masukan.nextInt();
 → Pertama, user diminta untuk memasukkan elemen pertama yaitu berupa bilangan bulat yang disimpan di variabel x.
- **while (x != 999){** → Kondisi berhenti. Statement dijalankan selama nilai x!=999
Sum = Sum + x; → Inisialisasi nilai sum yang baru yang diambil dari hasil penjumlahan sum + x
System.out.print("Masukkan nilai x (int), akhiri dg 999 : ");
x = masukan.nextInt(); → User diminta untuk menginputkan nilai x sebagai elemen pertama, dst.
} **System.out.println("Hasil penjumlahan = "+ Sum);** → Mencetak nilai variabel Sum.

18) Program 18

Output :

```
run:
Maksimum dua bilangan
Ketikkan dua bilangan, pisahkan dg RETURN :
2
4
Ke dua bilangan : a = 2 b = 4
Maksimum = 4
Tukar kedua bilangan...
Ke dua bilangan setelah tukar: a = 4 b = 2
```

Penjelasan :

Program tersebut bertujuan untuk membaca masukan berupa 2 bilangan integer, lalu menuliskan nilai maksimum dari dua bilangan yang dibaca dengan memanggil fungsi maxab serta menukar nilai dari kedua bilangan tsb. dengan memanggil fungsi tukar.

- **public static int maxab (int a, int b) {**
return ((a >= b) ? a : b);
}
 → Fungsi maxab ini bertipe data int yang dimana fungsi ini akan mengembalikan ekspresi/nilai ke pemanggil fungsi. Fungsi ini bertujuan untuk mencari nilai maksimum dari dua bilangan bulat.
- **public static void tukar (int a, int b) {**
int temp;
temp = a;
a = b;
b = temp;
System.out.println ("Ke dua bilangan setelah tukar: a = "+ a +" b = "+ b);
}
 → Fungsi tukar ini bertipe data void sehingga fungsi ini termasuk ke dalam fungsi yang tidak mengembalikan nilai. Fungsi ini hanya bertujuan untuk menukarkan nilai dari kedua bilangan.
- **System.out.println ("Maksimum = " + (maxab(a, b)));**
 → Perintah mencetak 'Maksimum = ' dan mencetak hasil pencarian nilai maksimum dari fungsi maxab.

- **tukar (a, b);** → Pemanggilan fungsi tukar sehingga fungsi tukar dieksekusi kemudian mencetak nilai kedua variabel berisi bilangan dari hasil penukaran kedua variabel itu sendiri.

19) Program 19

Output :

```
run:
Contoh IF tiga kasus
Temperatur (der. C) = -10
Wujud air beku
-10BUILD SUCCESSFUL (total time: 8 seconds)
```

→ kondisi ($T < 0$) bernilai true

```
run:
Contoh IF tiga kasus
Temperatur (der. C) = 10
Wujud air cair
10BUILD SUCCESSFUL (total time: 4 seconds)
```

→ kondisi ($(0 \leq T) \ \&\& \ (T \leq 100)$) bernilai true

```
run:
Contoh IF tiga kasus
Temperatur (der. C) = 101
Wujud air uap/gas
101BUILD SUCCESSFUL (total time: 4 seconds)
```

→ kondisi ($T > 100$) bernilai true

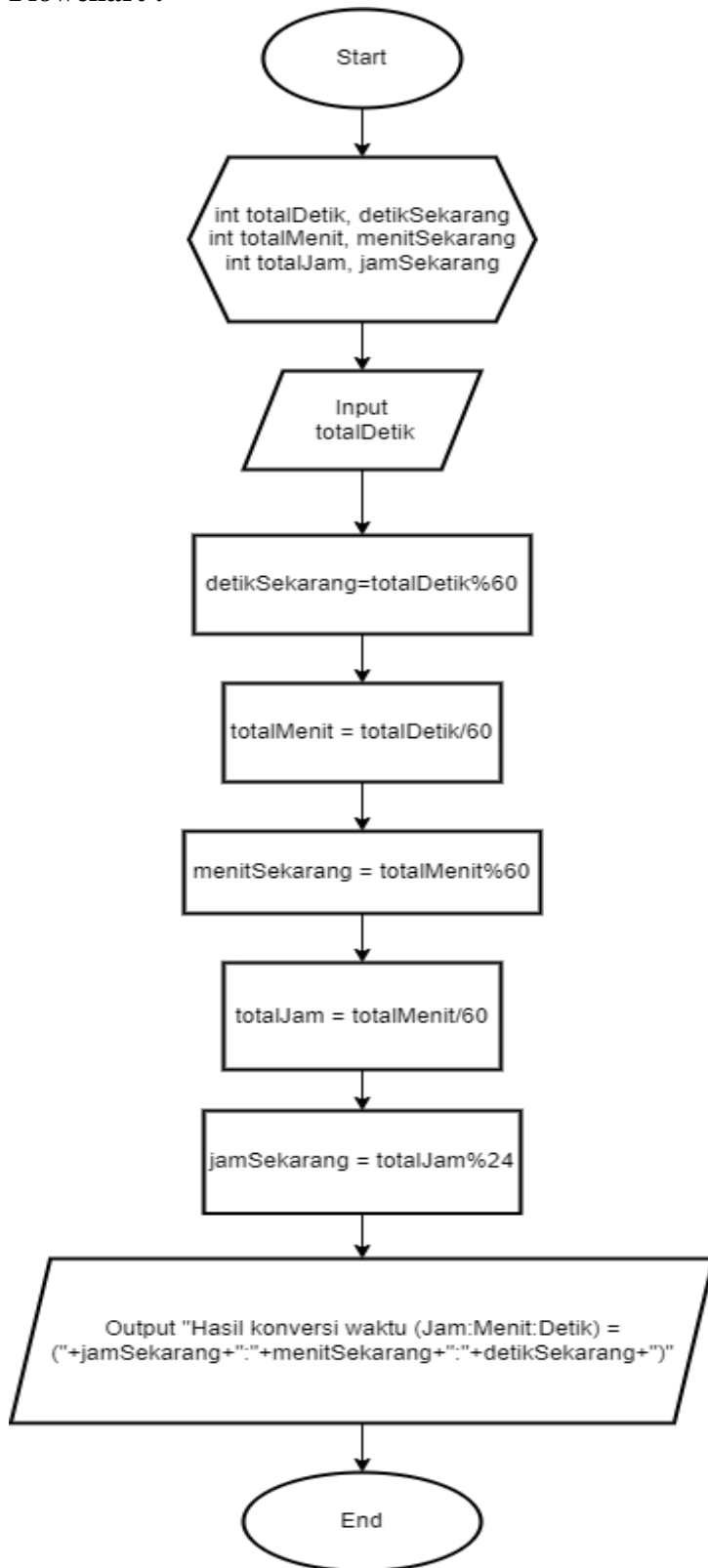
Penjelasan :

Program bertujuan untuk membaca masukan berupa bilangan integer yang disimpan ke dalam variabel T lalu nilai masukan tsb. diseleksi pada ekspresi-ekspresi if-else yang tersedia kemudian mencetak statement berdasarkan ekspresi if-else yang bernilai true.

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

Jawab :

Flowchart :



Output :

```
run:
Masukkan total detik : 10600
Hasil konversi waktu (Jam:Menit:Detik) = (2:56:40)
BUILD SUCCESSFUL (total time: 7 seconds)
```

Penjelasan :

Program tersebut bertujuan untuk menampilkan waktu dalam format (Jam:Menit:Detik) yang merupakan hasil konversi dari masukan totalDetik yang berupa bilangan integer dan dibaca menggunakan ClassScanner.

- **import java.util.Scanner;** → Import kelas Scanner ke dalam java. Di dalam paket **java** terdapat sub-paket **util**. Di dalam **util** terdapat file class **Scanner** yang didalamnya terdapat beberapa fungsi yang digunakan untuk membaca masukan.
- Semua variabel dalam program ini bertipe data integer
- **Scanner masukan = new Scanner(System.in);** → Pembuatan objek dari class Scanner.
- **System.out.print("Masukkan total detik : ");**
totalDetik = masukan.nextInt();
→ Objek masukan mengakses fungsi nextInt() yang berfungsi untuk membaca masukan berupa bilangan bulat atau integer. Lalu isi masukan akan disimpan di variabel totalDetik.
- **detikSekarang = totalDetik%60;** → mendapatkan nilai dari detikSekarang dengan menghitung operasi arithmetic modulus yaitu totalDetik%60 (misalnya 10600 detik % 60 = 40 detik)
- **totalMenit = totalDetik/60;** → mendapatkan nilai totalMenit dengan membagi totalDetik dengan 60 (misalnya 10600 detik/60 = 176 menit)
- **menitSekarang = totalMenit%60;** → mendapatkan nilai menitSekarang dari totalMenit%60 (misalnya 176 menit % 60 = 56 menit)
- **totalJam = totalMenit/60;** → mendapatkan nilai totalJam dengan membagi totalMenit dengan 60 (misalnya 176 menit / 60 = 2 jam)
- **jamSekarang = totalJam%24;** → mendapatkan nilai jamSekarang dari totalJam%24 (misalnya 2 jam % 24 = 2 jam)
- Dibuat perintah untuk menampilkan hasil konversi waktu dalam format (Jam:Menit:Detik) dengan memanggil nilai dari variabel jamSekarang, menitSekarang, dan detikSekarang.