

JobSheet - Week 1 - Fundamental Programming Structures in Java

Nama: Iyantona Ariesta Putra

NIM: 251524106

Kelas: 1D

Repo GitHub: <https://github.com/Ikanapa/TP>

Instruksi Pengerjaan:

1. Kerjakan 5 soal di bawah ini dengan melengkapi setiap kolom jawaban yang disediakan pada jobsheet ini.
2. Jawaban setiap soal mencakup source code, screenshot hasil dari program yang ditampilkan full screen termasuk taskbar (tambahkan beberapa screenshot jika diperlukan), penjelasan permasalahan dan solusi yang dihadapi, nama teman yang membantu memecahkan masalah (opsional).
3. Dikumpulkan pada Assignment Classroom sesuai dengan deadline yang tertera pada assignment tersebut.
4. Format penamaan file jobsheet: W1_P_<Kelas 1X>_<3 Digit_NIM_Terakhir>.docx/pdf. Contoh: W1_P_1B_001.docx/pdf.
5. Buatlah satu file java yang mengandung jawaban dalam bentuk source untuk satu jawaban yang dapat langsung dieksekusi. Contoh penamaan: 1-Datatypes.java, 2-Variables.java, 3-Arithmetic.java, 4-TypeCasting.java, dan 5-Operator.java.
6. Submit semua jawaban dalam bentuk file java pada repository GitHub masing-masing.

No. 1 Data Types

Soal Praktikum

Java memiliki 8 tipe data primitif: char, boolean, byte, short, int, long, float, dan double.

Untuk praktikum ini, kita akan Latihan dengan tipe data primitif yang digunakan untuk menyimpan nilai bilangan bulat, yaitu byte, short, int, dan long.

- A byte is an 8-bit signed integer.
- A short is a 16-bit signed integer.
- An int is a 32-bit signed integer.
- A long is a 64-bit signed integer.

Dengan diberikan sebuah bilangan bulat masukan, Anda harus menentukan tipe data primitif mana yang mampu menyimpan masukan tersebut dengan benar.

Input Format

Baris pertama berisi bilangan bulat, T, yang menunjukkan jumlah kasus uji. Setiap kasus uji, T, terdiri dari satu baris dengan bilangan bulat, n, yang nilainya bisa sangat besar atau sangat kecil.

Output Format

Untuk setiap variabel masukan n dan tipe data primitif yang sesuai, Anda harus menentukan apakah tipe data primitif yang diberikan mampu menyimpannya. Jika ya, maka cetak:

```
N can be fitted in:  
* datatype
```

Jika terdapat lebih dari satu tipe data yang sesuai, cetak masing-masing pada barisnya sendiri dan urutkan berdasarkan ukurannya (misalnya: **byte < short < int < long**).

Jika angka tersebut tidak dapat disimpan dalam salah satu dari empat tipe data primitif yang disebutkan di atas, cetak baris berikut:

```
N can't be fitted anywhere
```

Sample Input:

```
5  
-150  
150000  
1500000000  
213333333333333333333333333333333333  
-1000000000000000
```

Sample Output:

```
-150 can be fitted in:  
* short  
* int  
* long  
150000 can be fitted in:  
* int  
* long  
1500000000 can be fitted in:  
* int  
* long  
21333333333333333333333333333333333333 can't be fitted anywhere.
```

Explanation:

Angka 150 dapat disimpan dalam tipe data short, int, atau long. Angka 21333333333333333333333333333333333333 sangat besar dan berada di luar rentang nilai yang diizinkan untuk tipe data primitif yang dibahas dalam masalah ini.

Source Code

[TP/DataType.java at master · Ikanapa/TP](#)

Screenshot Hasil

```

1 import java.util.Scanner;
2 import java.math.BigInteger;
3
4 public class Datatype {
5     public static void main(String[] args) {
6         // input baca
7         Scanner pembaca = new Scanner(System.in);
8
9         // baca jumlah kasus uji
10        int jumlahkasusujji = pembaca.nextInt();
11
12        // Loop untuk setiap kasus uji
13        for (int indeks = 0; indeks < jumlahkasusujji; indeks++) {
14            // Membaca n sebagai string, lalu konversi ke BigInteger karena n bisa sangat besar
15            String nString = pembaca.next();
16            BigInteger bilangan = new BigInteger(nString);
17
18            // Mengcek apakah bilangan bisa disimpan dalam short, int, atau long
19            boolean dapatDisimpanDiShort = bilangan.compareTo(BigInteger.valueOf(Short.MIN_VALUE)) >= 0 &&
20                bilangan.compareTo(BigInteger.valueOf(Short.MAX_VALUE)) <= 0;
21            boolean dapatDisimpanDiInt = bilangan.compareTo(BigInteger.valueOf(Integer.MIN_VALUE)) >= 0 &&
22                bilangan.compareTo(BigInteger.valueOf(Integer.MAX_VALUE)) <= 0;
23            boolean dapatDisimpanDiLong = bilangan.compareTo(BigInteger.valueOf(Long.MIN_VALUE)) >= 0 &&
24                bilangan.compareTo(BigInteger.valueOf(Long.MAX_VALUE)) <= 0;
25
26            // jika bilangan bisa disimpan dalam salah satu tipe data, cetak pesan dan tipe data yang sesuai
27            if (dapatDisimpanDiShort || dapatDisimpanDiInt || dapatDisimpanDiLong) {
28                System.out.println(nString + " can be fitted in:");
29                if (dapatDisimpanDiShort) {
30                    System.out.println(x: " short");
31                }
32                if (dapatDisimpanDiInt) {
33                    System.out.println(x: " int");
34                }
35                if (dapatDisimpanDiLong) {
36                    System.out.println(x: " long");
37                }
38            }
39
40        }
41
42        // Menutup Scanner
43        pembaca.close();
44    }
45
46 }
47

```

Penjelasan Permasalahan dan Solusi

PERMASALAHAN

Bilangan 'n' bisa lebih besar dari 'Long.MAX_VALUE' atau lebih kecil dari 'Long.MIN_VALUE', sehingga kita tidak bisa menggunakan tipe data primitif langsung. Kita perlu menggunakan BigInteger untuk menangani nilai besar.

SOLUSI

1. Membaca Input: Gunakan Scanner untuk membaca T terlebih dahulu. Kemudian, untuk setiap kasus uji, baca n sebagai string (karena bisa sangat besar), lalu konversi ke BigInteger untuk perbandingan yang aman.
2. Pengecekan Range: Bandingkan n dengan batas minimum dan maksimum dari masing-masing tipe data

Nama Teman Hal yang Dibantu (Opsional)

No. 2 Variables

Soal Praktikum

Perhatikan dua bagian program di bawah ini.

Bagian 1:

```
public class Constants {  
    public static void main(String[] args) {  
        final double CM_PER_INCH = 2.54; double paperWidth = 8.5;  
        double paperHeight = 11;  
        System.out.println("Paper size in centimeters: " + paperWidth  
* CM_PER_INCH + " by " + paperHeight * CM_PER_INCH);  
    }  
}
```

Bagian 2:

```
public class Constants2 {  
    public static final double CM_PER_INCH = 2.54; public static void  
main(String[] args) {  
    double paperWidth = 8.5; double paperHeight = 11;  
    System.out.println("Paper size in centimeters: " + paperWidth  
* CM_PER_INCH + " by " + paperHeight * CM_PER_INCH);  
}
```

Dari 2 contoh baris program diatas, jawablah pertanyaan dibawah ini:

1. Bagaimana output dari masing masing class Constants dan Constants2?
2. Apa perbedaan penggunaan final double dengan public static final double?

Source Code

[TP/Constants2.java at master · Ikanapa/TP](#)

[TP/Constants.java at master · Ikanapa/TP](#)

Screenshot Hasil

The screenshot shows the VS Code interface with the following details:

- Explorer:** Shows files in the project: TP PR, .vscode, 3-Arithmetic.java, 4-typecasting.java, 5-Operator.java, Constants.class, Constants.java, Constants2.class, Constants2.java, DataType.class, DataType.java, and W1_P_1X_XXX_v1.1.docx.
- Terminal:** Displays the command 'java -version' and its output: 'java version "17.0.2"'. It also shows the path 'C:\Users\USER\Documents\JTK SEM 2\TP PR>'.
- Status Bar:** Shows the date '12/02/2026' and battery level '64%'. Other status icons are also present.

The screenshot shows the VS Code interface with the following details:

- EXPLORER** sidebar: Shows files like 3-Arithmetic.java, 4-TypeCasting.java, 5-Operator.java, Constants.class, Constants2.class, Constants2.java, Datatype.class, and Datatype.java.
- CODE** tab: Contains the Java code for Constants2.java.
- TERMINAL** tab: Displays the command-line output of the Java program.
- SUGGESTED ACTIONS**: Includes "Run: Constants2.java" and "Run: Constants2.java".
- AI Assistant**: A sidebar with the message "Build with Agent" and notes about AI responses being inaccurate, generating agent instructions, and AI on the codebase.

```

1 public class Constants2 {
2     public static final double CM_PER_INCH = 2.54; public static void
3         Run [Debug]
4         main(String[] args) {
5             double paperWidth = 8.5; double paperHeight = 11;
6             System.out.println("Paper size in centimeters: " + paperWidth
7
8             * CM_PER_INCH + " by " + paperHeight * CM_PER_INCH);
9         }
10    }

```

```

Microsoft Windows [Version 10.0.26200.7705]
(c) Microsoft Corporation. All rights reserved.

C:\Users\USER\Documents\JTK SEM 2\TP PR> cmd /c "<C:\Program Files\Java\jdk-25.0.2\bin\java.exe" --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp "C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\b1f8cd7eff0748ef9c551764c2aaefd\redhat\java\dt_wsl\TP PR_242dec9\bin" Constants2 "
Paper size in centimeters: 21.59 by 27.94

C:\Users\USER\Documents\JTK SEM 2\TP PR>

```

Penjelasan Permasalahan dan Solusi

1. Output dari masing masing class :

- Class Constants outputnya adalah paper size in centimeters: 21.59 by 27.94
- Class Constants2 outputnya adalah paper size in centimeters: 21.59 by 27.94

Penjelasan : Kode program "Constants" menghitung lebar kertas ($8.5 \text{ inci} \times 2.54 \text{ cm/inch} = 21.59 \text{ cm}$) dan tinggi ($11 \text{ inci} \times 2.54 \text{ cm/inch} = 27.94 \text{ cm}$), lalu mencetak hasilnya.

2. Perbedaan penggunaan final double dengan public static final double :

- Final double : Mendeklarasikan variabel lokal yang tidak bisa diubah setelah inisialisasi dan tidak bisa diakses dari luar metode atau kelas lain.
- Public static final double : Mendeklarasikan konstanta kelas yang bersifat publik yang bisa diakses tanpa membuat objek kelas (statis) dan final, namun bisa diakses dari mana saja dalam program.

PERMASALAHAN

Pada kode program Constants, 'CM_PER_INCH' dideklarasikan sebagai variabel lokal di dalam metode 'main'. Yang berarti ini tidak bisa digunakan diluar metode tsb, apabila kode diperluas dan misal ingin menambahkan metode lain dalam kelas atau kelas lain. Kita harus mendeklarasikan ulang, yang menyebabkan duplikasi kode yang tidak efektif.

Pada kedua program juga memiliki masalah umum, yakni tidak adanya validasi input atau penanganan error yang bisa menjadi masalah jika nilai seperti 'paperWidth' atau 'paperHeight' berasal dari input pengguna.

SOLUSI

1. Gunakan public static final untuk konstanta global
2. Buat validasi

Nama Teman dan Hal yang Dibantu (Opsiional)

No. 3 Arithmetic - Math Class

Soal Praktikum

Perhatikan bagian program di bawah ini.

```

Class FloatingPoint {
    public static void main(String[] args) {
        double x = 92.98;
        int nx = (int) Math.round(x);
    }
}

```

}

Math Class berisi bermacam-macam fungsi matematika seperti pada contoh diatas pada penggunaan `round(x)`, terdapat beberapa pertanyaan yang perlu untuk dijelaskan:

1. Pada kasus berikut jelaskan nilai nx setelah digunakan **Math.round(x)!**
2. Kenapa dibutuhkan cast (int) dalam penggunaan **Math.round(x)!**

Source Code

[TP/3-Arithmetic.java at master · Ikanapa/TP](#)

Screenshot Hasil

```

File Edit Selection View Go Run Terminal Help ⏎ →
EXPLORER 3-Arithmetic.java ×
TP PR
vscode
-WRL2284.tmp
3-Arithmetic.java
4-TypeCasting.java
5-Operator.java
Constants.class
Constants.java
Constants2.class
Constants2.java
DataType.class
DataType.java
W1_P_IX_XXX_y1.tdock

3-Arithmetic.java > FloatingPoint > main(String[])
1 class FloatingPoint {
2     Run | Debug
3     public static void main(String[] args) {
4         double x = 92.98;
5         int nx = (int) Math.round(x);
6         System.out.println(nx);
7     }
8 }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Microsoft Windows [Version 10.0.26200.7705]
(c) Microsoft Corporation. All rights reserved.

C:\Users\USER\Documents\JTK SEM 2\TP PR> cmd /C ""C:\Program Files\Java\jdk-25.0.2\bin\java.exe" --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp "C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\b1f8cd77eff6748ef9c551764c2aaefd\redhat.java\jdt_ws\TP PR_242dec9\bin" FloatingPoint"
93

C:\Users\USER\Documents\JTK SEM 2\TP PR>

SUGGESTED ACTIONS
Build Workspace
Show Config
+ 3-Arithmetic.java
Describe what to build ne
Ln 5, Col 24 Spaces: 4 UTF-8 CRLF ⓘ Java ⓘ Go Live ⓘ Prettier ⓘ 16.00 12/02/2026
100% 80% 12/02/2026

```

Penjelasan Permasalahan dan Solusi

1. Nilai nx setelah digunakan Math.round(x) :

- Dalam kode 'double x = 92.98; int nx = (int) Math.round(x);', nilai nx adalah '93'

Penjelasan : metode 'Math.round(double x)' membulatkan nilai 'x' ke bilangan bulat terdekat. Karena $x=92.98$, bagian desimalnya 0,98 sehingga dibulatkan ke atas menjadi 93.0 (dalam bentuk 'long').

Kemudian, cast '(int)' mengubahnya menjadi 'int' tanpa kehilangan nilai.

2. Kenapa dibutuhkan cast (int) dalam penggunaan Math.round (x) :

Karena metode 'Math.round(double x)' mengembalikan nilai bertipe 'long', bukan 'int'. Dan Cast '(int)' dibutuhkan untuk mengubah 'long' menjadi 'int', karena variabel 'nx' dideklarasikan sebagai 'int'. Tanpa cast, akan terjadi error kompilasi karena ketidakcocokan tipe data. Namun, cast ini aman hanya jika nilai 'long' berada dalam rentang 'int'.

PERMASALAHAN

1. Risiko overflow apabila x sangat besar yang bisa menghasilkan long yang melebihi rentang 'int'
2. Tidak ada validasi

SOLUSI

1. Periksa rentang sebelum cast atau pastikan nilai long dari Math.round() dalam rentang int untuk menghindari overflow

Nama Teman dan Hal yang Dibantu (Opsional)

No. 4 Type Casting/ Data Type Conversion

Soal Praktikum

Perhatikan baris program dibawah ini:

```
class ConvertDataType {  
    static short methodOne(long l) {  
        int i = (int) l; return (short)i;  
    }  
  
    public static void main(String[] args) {  
        double d = 10.25; float f = (float) d;  
        byte b = (byte) methodOne((long) f); System.out.println(b);  
    }  
}
```

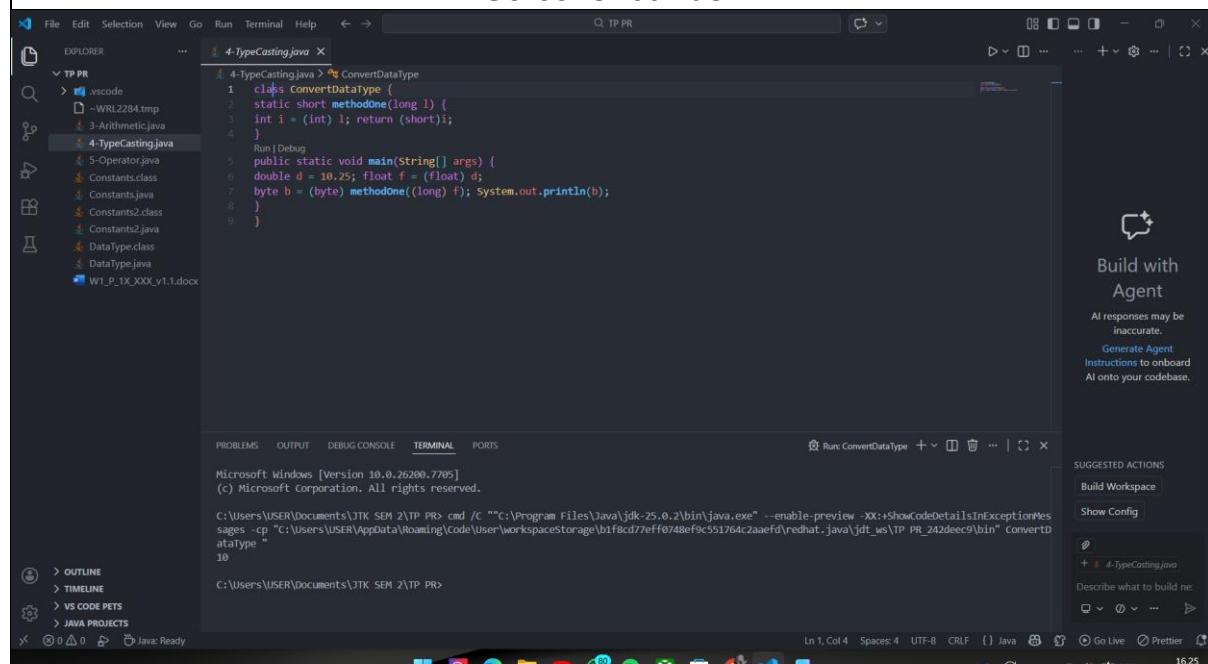
Program berikut melakukan convert tipe data yang berukuran besar ke kecil (long -> int -> short) dan (double -> float -> byte).

1. Jelaskan output nilai dari variable b.
2. Jelaskan apa yang berubah dari variable d menjadi variable b setelah dilakukan cast?

Source Code

[TP/4-TypeCasting.java at master · Ikanapa/TP](#)

Screenshot Hasil



```
File Edi Selection View Go Run Terminal Help ← → Q TP PR  
EXPLORER ... 4-TypeCasting.java ConvertDataType  
1 class ConvertDataType {  
2     static short methodOne(long l) {  
3         int i = (int) l; return (short)i;  
4     }  
5     public static void main(String[] args) {  
6         double d = 10.25; float f = (float) d;  
7         byte b = (byte) methodOne((long) f); System.out.println(b);  
8     }  
9 }
```

Run | Debug

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.26200.7705]
(c) Microsoft Corporation. All rights reserved.

C:\Users\USER\Documents\JTK SEM 2\TP PR> cmd /C "<C:\Program Files\Java\jdk-25.0.2\bin\java.exe" --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp "C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\b1f8cd77eff0748ef9c551764czaaefd\redhat.java\jdt_ws\TP_PR_242dec9\bin" ConvertDataType
10

C:\Users\USER\Documents\JTK SEM 2\TP PR>

OUTLINE TIMELINE VS CODE PETS JAVA PROJECTS Java: Ready

Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.

SUGGESTED ACTIONS Build Workspace Show Config

+ 4-TypeCasting.java
Describe what to build next
Go Live Prettier

16.25 88% 12/02/2026

Penjelasan Permasalahan dan Solusi

1. Output nilai variabel b :

Nilai berubah dari 10.25 (desimal) menjadi 10 (integer bulat) karena casting dari float ke long memotong bagian desimal

Nilai awal : 'd = 10.25'

Konversi ke float : 'f = (float) d' jadi 'f' lalu menjadi '10.25f'

Konversi ke long : '(long) f' jadi Casting 'float' ke 'long' memotong bagian desimal, sehingga menjadi '10L'
Panggilan methodOne(10L) :

'int i = (int) 10L' jadi 'i = 10'

'return (short) i' jadi 'short' (nilai 10 masih dalam rentang, jadi kembali 10)

Konversi ke byte : 'byte b = (byte) 10' jadi byte (nilai 10 masih valid , jadi b = 10)

2. Perubahan dari variable d ke variable b :

Nilai berubah dari 10.25 (desimal) menjadi 10 (integer bulat) karena casting dari float ke long memotong bagian desimal.

PERMASALAHAN

1. Resiko overflow : Jika nilai 'd' lebih besar (misalnya 1000.25), casting ke byte bisa menghasilkan nilai negatif atau salah

2. Tidak ada pengecekan

SOLUSI

1. Pengecekan rentang sebelum casting

Nama Teman dan Hal yang Dibantu (Opsional)

No. 5 Operator

Soal Praktikum

Perhatikan bagian program di bawah ini.

```
class OperatorChallenge {
    public static void main(String[] args) {
        int a = 5;
        int b = 10;

        boolean result = (++a * 2 > b) && (b++ % 3 == 1);

        System.out.println("Hasil Boolean: " + result);
        System.out.println("Nilai a: " + a);
        System.out.println("Nilai b: " + b);
    }
}
```

Pertanyaan Analisis:

- Analisis Langkah Demi Langkah:** Jelaskan urutan eksekusi pada baris `boolean result = (++a * 2 > b) && (b++ % 3 == 1);`. Mana yang dijalankan lebih dulu antara `++a` dan perkalian `*`?
- Short-Circuit Logic:** Jika bagian pertama `(++a * 2 > b)` bernilai false, apakah bagian kedua `(b++ % 3 == 1)` akan tetap dieksekusi oleh Java? Jelaskan dampaknya pada nilai akhir variabel `b`.
- Output:** Berapakah nilai akhir dari `result`, `a`, dan `b`?

Source Code

[TP/5-Operator.java at master · Ikanapa/TP](#)

Screenshot Hasil

The screenshot shows the Visual Studio Code interface with the following details:

- Explorer:** Shows files in the workspace, including `5-Operator.java`, `Constants.class`, `Constants.java`, `Constants2.class`, `Constants2.java`, `DataTypes.class`, `DataTypes.java`, and `W1_P_1X_XXX_v1.1.docx`.
- Terminal:** Shows the command line output of the Java application.

```
Microsoft Windows [Version 10.0.26280.7785]
(c) Microsoft Corporation. All rights reserved.

C:\Users\USER\Documents\JTK SEM 2\TP PR> cmd /C "C:\Program Files\Java\jdk-25.0.2\bin\java.exe" --enable-preview -XX:+ShowCodeDetailsInExceptionMessages -cp "C:\Users\USER\AppData\Roaming\Code\User\workspaceStorage\b1f8cd7eff0748ef9c551764czaaefd\redhat\java\jdt_ws\TP_PR_242deec9\bin" OperatorChallenge
Hasil Boolean: true
Nilai a: 6
Nilai b: 11
```
- Suggested Actions:** A sidebar on the right suggests actions like "Build with Agent".

1. Urutan eksekusi boolean result :

Inisialisasi awal: a = 5, b = 10

Bagian kiri: $(++a * 2 > b)$: Nilai a ditingkatkan menjadi 6 lalu digunakan dalam ekspresi. Setelah $++a$, dilakukan perkalian $6*2=12$. Perbandingan $12 > 10 \rightarrow \text{true}$

Bagian kanan ($b++ \% 3 == 1$) (dieksekusi karena bagian kiri true : Nilai b (10) digunakan terlebih dahulu dalam ekspresi, lalu b ditingkatkan menjadi 11. Modulo $10 \% 3 = 1$. Perbandingan $1 == 1 = \text{true}$

Hasil ekspresi : $\text{true} \ \&\& \ \text{true} = \text{true}$.

Dalam bagian kiri, $++a$ dieksekusi lebih dulu (karena pre-increment memodifikasi nilai sebelum digunakan), diikuti oleh perkalian, lalu perbandingan.

2. Short-Circuit Logic: Dampak pada Bagian Kedua

Apakah bagian kedua dieksekusi jika bagian pertama false? Tidak. Operator ' $\&\&$ ' menggunakan short-circuit evaluation: Jika bagian kiri $(++a * 2 > b)$ bernilai false, Java langsung menghentikan evaluasi dan menetapkan result = false tanpa mengeksekusi bagian kanan ($b++ \% 3 == 1$). Dampaknya, Jika bagian kiri false (misalnya, jika a awalnya lebih kecil sehingga $++a * 2$ tidak $> b$), maka $b++$ tidak dieksekusi, sehingga b tetap 10 (tidak bertambah menjadi 11). Ini berarti nilai akhir b bergantung pada hasil bagian kiri

3. Output: Nilai Akhir dari result, a, dan b

- result: true
- a: 6
- b: 11

Nama Teman dan Hal yang Dibantu (Opsional)