

Nama & NPM	Topik:	Tanggal:
Marchellino Madani Putra G1F024064	Tipe Data	5 September 2024

[No. 1] Identifikasi Masalah:

- 1) Uraikan permasalahan dan variabel

```
public class KelasKu {
    private static void main(String[] args) {
        System.out.println("Halo Mahasiswa UNIB")
    }
}
```

Luaran:

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
String literal is not properly closed by a double-quote
Syntax error, insert ";" to complete BlockStatements
```

Latihan 1:

1. Evaluasi penyebab kesalahan terjadi dan perbaiki agar program dapat berjalan!
2. Ubah teks yang ditampilkan program menjadi nama lengkap Anda.
3. Tambahkan baris `System.out.println("");` untuk diisi dengan data alamat, dan jenis kelamin.

[No.1] Analisis dan Argumentasi

- 1) Method di deklarasikan menjadi Public static void agar bias di jalankan oleh Java.
- 2) Menutup dengan tanda kutip di akhir string ("Halo Mahasiswa Unib")

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Mulai
 - (b) Buka Java Compiler
 - (c) Buat project
 - (d) Deklarasi public static
 - (e) Print data
 - (f) Tekan run
 - (g) Selesai
- 2) Kode program dan luaran

The screenshot shows a Java IDE with a file named 'Main.java'. The code defines a class 'KelasKu' with a 'main' method. The 'main' method contains three 'System.out.println' statements that print the name, address, and gender. The output window on the right shows the execution of the program, displaying the same three lines of text. The output is preceded by the command 'java -cp /tmp/7aYrfP2cjg/KelasKu' and followed by '=== Code Execution Successful ==='.

```
Main.java
1- public class KelasKu {
2-     public static void main(String[] args) { // Mengubah main string menjadi
        public static void
3         System.out.println("Nama:Marchellino Madani Putra");
4         System.out.println("alamat:Sebakul"); // Tanda kutip ditutup dengan benar
5         System.out.println("jenis kelamin:laki laki");
6     }
7 }
```

Output

```
java -cp /tmp/7aYrfP2cjg/KelasKu
Nama:Marchellino Madani Putra
alamat:Sebakul
jenis kelamin:laki laki
=== Code Execution Successful ===
```

Teks yang muncul adalah Nama:Marchellino Madani Putra, alamat:Sebakul jenis kelamin:laki laki. Ini adalah string yang ditentukan dalam pernyataan `System.out.println()`. Keluaran ini menunjukkan bahwa program berjalan dengan baik dan berhasil mengeksekusi perintah untuk mencetak output.

[No.1] Kesimpulan

Analisa

Analisa ini menunjukkan bahwa proses eksekusi kode Java ini mencakup banyak konsep dasar yang penting. Dan terdapat beberapa permasalahan yang ada seperti tanda kutip tidak di tutup kembali dengan tanda kutip dan mengubah main string menjadi public static void yang dapat menyebabkan eror pada program, program ini juga menggambarkan konsep dasarpemrograman.

[No. 2] Identifikasi Masalah:

Java memiliki 8 tipe data primitif dan non-primitif

Apabila diketahui data berikut

1. C 5
2. 'L'
3. "mobil"
4. 5.0
5. 5.0f
6. -5

Latihan 2:

- 1.) Rekomendasikan tipe data yang tepat dari data Contoh 2.
- 2.) Simpulkan karakteristik penggunaan setiap tipe data!

[No.2] Analisis dan Argumentasi

- 1.) 5 Tipe data(int)
Angka ini adalah bilangan bulat tanpa desimal, int adalah tempat untuk menyimpan bilangan bulat
- 2.) 'L' Tipe data(char)
Tipe data `char` digunakan untuk menyimpan satu karakter, seperti huruf, angka, atau simbol.
- 3.) "mobil" Tipe data(String)
"mobil" adalah urutan karakter, tipe data string digunakan untuk menyimpan teks, seperti kata atau kalimat.
- 4.) 5.0 Tipe data(double)
Tipe data double cocok untuk menyimpan nilai desimal dengan presisi tinggi, seperti angka yang membutuhkan banyak tempat desimal.
- 5.) 5.0f Tipe data(float)
Tipe data float digunakan ketika pengguna ingin menghemat memori dan tidak memerlukan banyak tempat desimal. Atau menyimpan bilangan desimal yang lebih rendah
- 6.) -5 Tipe data(int)
Ini adalah bilangan bulat negatif. Sama seperti yang pertama, `int` adalah tipe data untuk menyimpan nilai bilangan bulat baik positif maupun negatif.

[No.2] Kesimpulan**Analisa**

Java memiliki 8 tipe data primitif yang digunakan untuk menyimpan nilai. Pemilihan tipe data yang tepat sangat penting untuk menghemat memori, selain itu pemilihan tipe data yang tepat juga dapat membantu memperjelas kapan dan mengapa harus menggunakan tipe data tersebut. Pemilihan tipe data juga dapat membantu pengguna dalam meminimalisir kesalahan dan mempercepat kinerja aplikasi dan kinerja program yang digunakan.

[No. 3] Identifikasi Masalah:

1.) Uraikan permasalahan dan variable

```
public class TipeData {  
  
    public static void main(String args[]) {  
        int a = 5555555555;  
        byte b = 4444444444;  
        float c = 12.345678910f;  
        double d = 12.345678910d;  
        char e = 'abc';  
  
        System.out.println(a);  
        System.out.println(b);  
        System.out.println(c);  
        System.out.println(d);  
        System.out.println(e);  
    }  
}
```

Luaran:

```
/TipeData.java:3: error: integer number too large  
    int a = 5555555555;  
/TipeData.java:4: error: integer number too large  
    byte b = 4444444444;  
/TipeData.java:7: error: unclosed character literal  
    char e = abc';  
3 errors
```

Latihan 3:

1. Evaluasi penyebab kesalahan pada Contoh 3!
2. Rekomendasikan tipe data yang sesuai untuk data tersebut!

[No.3] Analisis dan Argumentasi

- 1.) Kesalahan pertama terdapat pada pemilihan tipe data yang tidak tepat, nilai 5555555555 terlalu besar untuk menggunakan tipe data int, dari hasil evaluasi tipe data yang cocok di gunakan untuk nilai tersebut adalah tipe data long. Karena tipe data ini di gunakan untuk menyimpan bilangan bulat yang nilainya lebih besar dan tidak dapat di simpan dalam tipe data integer.
- 2.) Kesalahan ke dua sama dengan kesalahan yang pertama karna pemilihan tipe data yang tidak tepat, nilai 4444444444 terlalu besar jika menggunakan tipe data byte. Tipe data yang cocok untuk nilai tersebut adalah tipe data integer karna memiliki ruang penyimpanan bilangan yang masih cukup untuk nilai tersebut.
- 3.) Kesalahan selanjutnya di sebabkan oleh kesalahan dalam memilih tipe data, tipe data char hanya dapat menyimpan satu karakter. Rubah tipe data menjadi string karna string dapat menyimpan lebih dari satu karakter, atau kurangi menjadi satu karakter pada tipe data char.

[No.3] Penyusunan Algoritma dan Kode Program

- 1.) Algoritma
 - (a) Mulai
 - (b) Buka Java Compiler
 - (c) Buat project
 - (d) Mulai
 - (e) Deklarasi variable
 - (f) Menampilkan nilai
 - (g) Print data
 - (h) Eksekusi
 - (i) Selesai
- 2.) Kode program dan luaran

	Output	Generated Files
<pre>1 public class TipeData { 2 3 public static void main(String args[]) { 4 long a = 5555555555L; 5 int b = 4444444444; 6 float c = 12.345678910f; 7 double d = 12.345678910; 8 String e = "abc"; 9 10 System.out.println(a); 11 System.out.println(b); 12 System.out.println(c); 13 System.out.println(d); 14 System.out.println(e); 15 } 16 }</pre>	<pre>5555555555 4444444444 12.345679 12.34567891 abc </pre>	

Hasil output yang di keluarkan setelah melakukan evaluasi dan perbaikan adalah 5555555555 nilai ini adalah variable a yang disimpan menggunakan tipe data long, (L) di belakang nilai 5555555555 bertujuan memberitau bahwa nilai tersebut menggunakan tipe data long. Lalu 4444444444 Ini adalah nilai dari variable b yang disimpan sebagai integer, sedangkan 12.345679 ini adalah nilai dari variable c, yang disimpan sebagai float. 12.3456789 ini adalah nilai dari variable d, yang disimpan sebagai double, dan memiliki kapasitas yang lebih tinggi, abc adalah nilai dari variabel e, disimpan sebagai stirng yang bisa menampilkan ebih dari satu karakter.

[No3] Kesimpulan

Analisa

Output yang dihasilkan menunjukkan bagaimana berbagai tipe data berfungsi dengan baik dalam Java, dan bagaimana tipe data tersebut menangani penyimpanan dan pengambilan nilai. Dengan beberapa kesalahan dalam pemilihan tipe data dapat membuat kita lebih teliti lagi dalam memahami berbagai macam tipe data yang ada pada java, memahami macam macam tipe data serta kemampuan yang ada pada tipe data tersebut dapat memudahkan pengguna dalam menghadapi masalah yang serupa.

[No. 4] Identifikasi Masalah:

- 1.) Uraikan permasalahan dan variabel

```
public class KonversiTipeData {  
    public static void main(String args[]) {  
        byte x;  
        int a = 270;  
        double b = 128.128;  
        System.out.println("int dikonversi ke byte");  
        x = (byte) a;  
        System.out.println("a dan x " + a + " " + x);  
        System.out.println("double dikonversi ke int");  
        a = (int) b;  
        System.out.println("b dan a " + b + " " + a);  
        System.out.println("double dikonversi ke byte");  
        x = (byte)b;  
        System.out.println("b dan x " + b + " " + x);  
    }  
}
```

Luaran:

```
int dikonversi ke byte  
a dan x 270 14  
double dikonversi ke int  
b dan a 128.128 128  
double dikonversi ke byte  
b dan x 128.128 -128
```

Latihan 4:

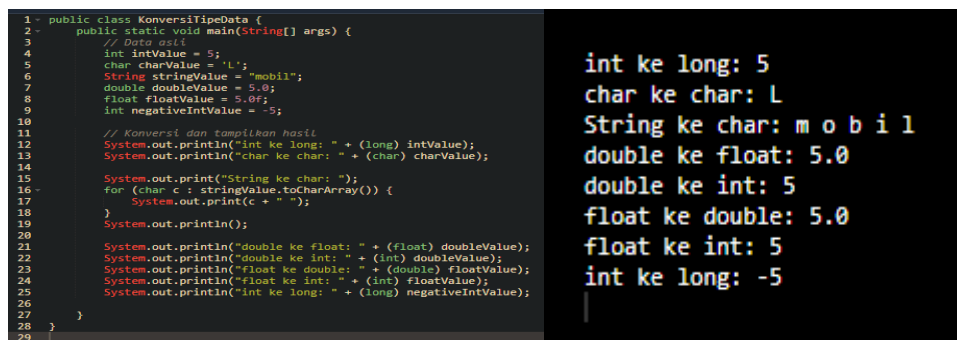
1. Rekomendasikan konversi tipe data pada Latihan 2 ke bentuk tipe data lain yang kompatibel.
2. Simpulkan alasan jenis konversi tipe data tersebut!

[No.4] Analisis dan Argumentasi

1. Proses konversi tipe data pada latihan 2 pada setiap masing masing data yang ada, lakukan deklarasi variable, lalu konversikan masing masing tipe data yang telah di buat di latihan 2 ke yang lebih kompetibel int ke long, char ke char, String ke char, double ke float, double ke int, float ke double, float ke int, int ke long.
 - a. Mengonversi int menjadi long karena dapat menyimpan angka yang lebih besar dibandingkan int.
 - b. Konversi ini tidak mengubah nilai, karena hasilnya tetap karakter 'L'. jika di konversikan ke int maka 'L' akan berubah menjadi angka.
 - c. Mengonversi string menjadi array karakter char[]. Menggunakan to chararray() program mencetak setiap karakter dari string "mobil" satu per satu.
 - d. Mengonversi double ke float dan int. Mengonversi ke int akan menghilangkan bagian desimal.
 - e. Mengonversi float ke double.
 - f. Konversi bilangan bulat negative menjadi.
2. Kode ini memberikan gambaran yang jelas tentang konversi tipe data dalam Java, proses pengonversian data ini dilakukan untuk menyesuaikan tipe data sesuai dengan kapasitasnya agar program tidak berat untuk di jalankan.

[No.4] Penyusunan Algoritma dan Kode Program

- 3) Algoritma
 - (a) Mulai
 - (b) Buka Java Compiler
 - (c) Buat project
 - (d) Deklarasi kelas
 - (e) Metode main
 - (f) Deklarasi variable
 - (g) Proses konversi
 - (h) Print data
 - (i) Selesai
- 4) Kode program dan keluaran



```
1 public class KonversiTipeData {
2     public static void main(String[] args) {
3         // Data asli
4         int intValue = 5;
5         char charValue = 'l';
6         String stringValue = "mobil";
7         double doubleValue = 5.0;
8         float floatValue = 5.0f;
9         int negativeIntValue = -5;
10
11         // Konversi dan tampilkan hasil
12         System.out.println("int ke long: " + (long) intValue);
13         System.out.println("char ke char: " + (char) charValue);
14
15         System.out.print("String ke char: ");
16         for (char c : stringValue.toCharArray()) {
17             System.out.print(c + " ");
18         }
19         System.out.println();
20
21         System.out.println("double ke float: " + (float) doubleValue);
22         System.out.println("double ke int: " + (int) doubleValue);
23         System.out.println("float ke double: " + (double) floatValue);
24         System.out.println("float ke int: " + (int) floatValue);
25         System.out.println("int ke long: " + (long) negativeIntValue);
26     }
27 }
28
29
```

int ke long: 5
char ke char: l
String ke char: m o b i l
double ke float: 5.0
double ke int: 5
float ke double: 5.0
float ke int: 5
int ke long: -5

Jadi kode di atas adalah proses konversi tipe data di dalam java yang hasilnya di tampilkan menggunakan System.out.println(). Proses konversi ini bertujuan supaya data yang kita masukkan sesuai dengan kapasitas di tipe data, jadi pemilihan tipe data perlu di perhatikan saat menjalankan program. keluaran tersebut adalah hasil dari konversi tipe data yang sudah di jalankan seperti penjelasan sebelumnya.

[No.4] Kesimpulan

Analisa

Secara umum, java mendukung dan membolehkan konversi tipe data selama konversi tersebut mengikuti aturan plebarann atau pengecilan tipe data, dari yang kecil ke besar atau besar ke kecil. Untuk dapat menjalankan proses pemrograman yang laçar dan lebih efisien, oleh karena itu mengetahui tipe data itu penting untuk di ketahui. Proses konversi ini juga menghindari terjadinya Overflow.