

Nama & NPM	Topik:	Tanggal:
TIARA YEMELDA G1F021008	Tipe Data	10 oktober 2024

JAWABAN NO 1 :

[No. 1.1] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel:

- Soal: Pada soal masih ada pesan kesalahan "unclosed string literal".
- Diketahui dari soal: Variabel `System.out.println("Halo Mahasiswa UNIB`` tidak memiliki tanda kutip penutup dan metode `main`` bersifat private.

[No. 1.1] Analisis dan Argumentasi:

1) menurut saya permasalahan ini dapat diatasi dengan cara:

- Menambahkan tanda kutip penutup pada string literal dan mengganti akses modifier metode `main`` dari `private`` menjadi `public``.

2) Alasan solusi ini karena:

- String literal harus ditutup dengan tanda kutip ganda untuk menandai akhir dari teks yang akan dicetak. Selain itu, metode `main`` harus bersifat `public`` karena merupakan entry point yang akan dijalankan oleh JVM saat program dijalankan.

3) Perbaiki kode program dengan cara:

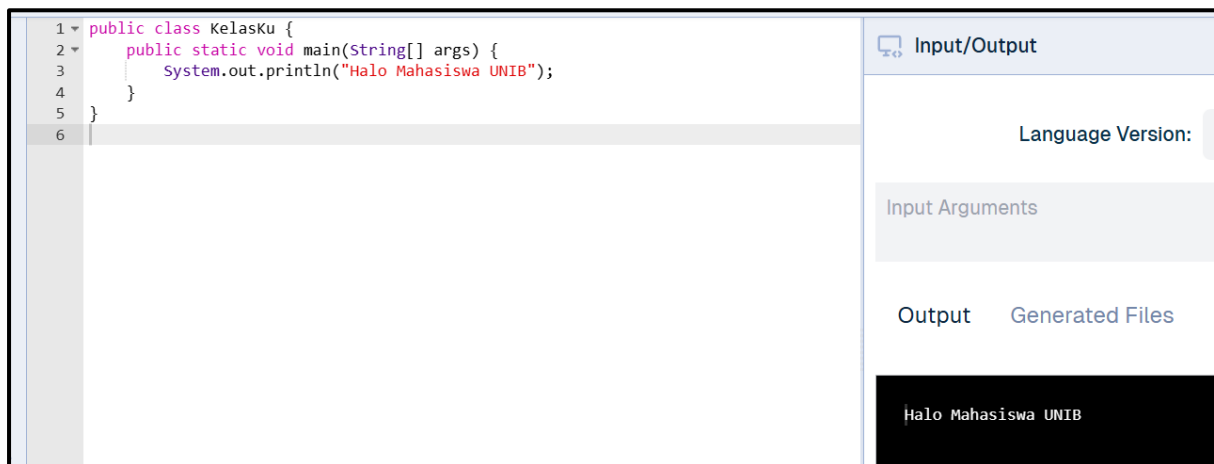
- Menambahkan tanda kutip penutup pada baris `System.out.println("Halo Mahasiswa UNIB");`` dan mengganti akses metode `main`` menjadi `public``.

[No. 1.1] Penyusunan Algoritma dan Kode Program:

1) Algoritma:

- (a) Deklarasikan kelas `KelasKu``.
- (b) Buat metode `main`` dengan akses modifier `public`` yang akan dijalankan oleh JVM.
- (c) Cetak teks "Halo Mahasiswa UNIB" menggunakan perintah `System.out.println``.
- (d) Tampilkan hasil.

2) Kode program dan luaran:



The screenshot shows a Java IDE with a code editor on the left and a right-hand panel titled "Input/Output". The code editor contains the following Java code:

```
1 public class Kelasku {  
2     public static void main(String[] args) {  
3         System.out.println("Halo Mahasiswa UNIB");  
4     }  
5 }  
6
```

The right-hand panel has tabs for "Input/Output", "Language Version:", "Input Arguments", "Output", and "Generated Files". The "Output" tab is selected, showing the text "Halo Mahasiswa UNIB" in a black box.

b) Analisa luaran yang dihasilkan:

- Luaran sudah sesuai dengan program yang disusun. Program mencetak string "Halo Mahasiswa UNIB" ke layar tanpa kesalahan. Kode dan tipe data yang digunakan telah sesuai dengan kebutuhan.

[No. 1.1] Kesimpulan:

1) Analisa:

- Pada program ini saya menggunakan bentuk kelas `public` karena kelas `Kelasku` harus dapat diakses oleh JVM saat program dijalankan. Perbaikan program dilakukan dengan menambahkan tanda kutip penutup pada string literal dan mengubah akses modifier metode `main` menjadi `public` karena struktur Java mengharuskan metode entry point dapat diakses secara publik.

[No. 1.2] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel:

- Diketahui dari soal: Menyusun kode program Java yang menampilkan data pribadi Anda dengan minimal 5 informasi data diri.

[No. 1.2] Analisis dan Argumentasi:

1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara:

- Membuat program Java yang mencetak informasi data pribadi seperti nama, umur, alamat, jurusan, dan hobi.

2) Alasan solusi ini karena:

- Untuk menampilkan data diri, kita membutuhkan variabel dengan tipe data yang sesuai, seperti `String` untuk nama dan alamat, `int` untuk umur, dan sebagainya.

3) Perbaiki kode program dengan cara:


- Menyusun kode program menggunakan metode `main` yang akan mencetak lima informasi data pribadi.

[No. 1.2] Penyusunan Algoritma dan Kode Program:

1) Algoritma:

- (a) Deklarasikan kelas `DataPribadi`.
- (b) Buat metode `main`.
- (c) Deklarasikan variabel yang menyimpan informasi pribadi.
- (d) Cetak variabel tersebut menggunakan `System.out.println`.
- (e) Tampilkan hasil.

2) Kode program dan luaran:



The screenshot shows a Java IDE with a code editor on the left and an 'Input/Output' panel on the right. The code defines a class `DataPribadi` with a `main` method that declares variables for name, age, address, major, and hobby, and prints them. The output panel shows the execution results.

```
1 public class DataPribadi {  
2     public static void main(String[] args) {  
3         String nama = "Tiara Yemelda";  
4         int umur = 20;  
5         String alamat = "Bengkulu";  
6         String jurusan = "Sistem Informasi";  
7         String hobi = "Membaca buku";  
8  
9         System.out.println("Nama: " + nama);  
10        System.out.println("Umur: " + umur + " tahun");  
11        System.out.println("Alamat: " + alamat);  
12        System.out.println("Jurusan: " + jurusan);  
13        System.out.println("Hobi: " + hobi);  
14    }  
15 }  
16 }
```

Input/Output

Language Version: JDK 2

Input Arguments

Output Generated Files

```
Nama: Tiara Yemelda  
Umur: 20 tahun  
Alamat: Bengkulu  
Jurusan: Sistem Informasi  
Hobi: Membaca buku
```

Compiled and executed in 1.899 sec(s)

b) Analisa luaran yang dihasilkan:

Luaran sudah sesuai dengan program yang disusun. Program mencetak lima informasi data pribadi: nama, umur, alamat, jurusan, dan hobi. Tidak ada kesalahan sintaks atau runtime.

[No. 1.2] Kesimpulan:

1) Analisa:

- Pada program ini, saya menggunakan tipe data `String` untuk menyimpan nama, alamat, jurusan, dan hobi, serta `int` untuk menyimpan umur. Perbaikan dilakukan dengan menyusun kode sesuai dengan aturan Java untuk menampilkan informasi tersebut.

JAWABAN NO 2

[No. 2.1] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel:

- Pada soal masih ada pesan kesalahan terkait:
 - Nilai integer terlalu besar untuk tipe data `int`.
 - Nilai byte terlalu besar untuk tipe data `byte`.
 - Karakter literal `char` tidak valid karena berisi lebih dari satu karakter.
- Diketahui dari soal: variabel `a` bertipe `int`, `b` bertipe `byte`, `c` bertipe `float`, `d` bertipe `double`, dan `e` bertipe `char`.

[No. 2.1] Analisis dan Argumentasi:

1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara:

- Mengubah tipe data variabel `a` menjadi `long` agar dapat menyimpan bilangan yang lebih besar.
- Mengubah tipe data variabel `b` menjadi `int` agar dapat menyimpan nilai yang lebih besar dari batas `byte`.
- Mengubah karakter literal `char` `e` menjadi satu karakter tunggal.

2) Alasan solusi ini karena:

- Tipe data `long` memiliki kapasitas yang lebih besar dari `int`.
- Tipe data `int` memiliki kapasitas yang lebih besar dari `byte`.
- Tipe data `char` hanya bisa menyimpan satu karakter.

3) Perbaikan kode program dengan cara:

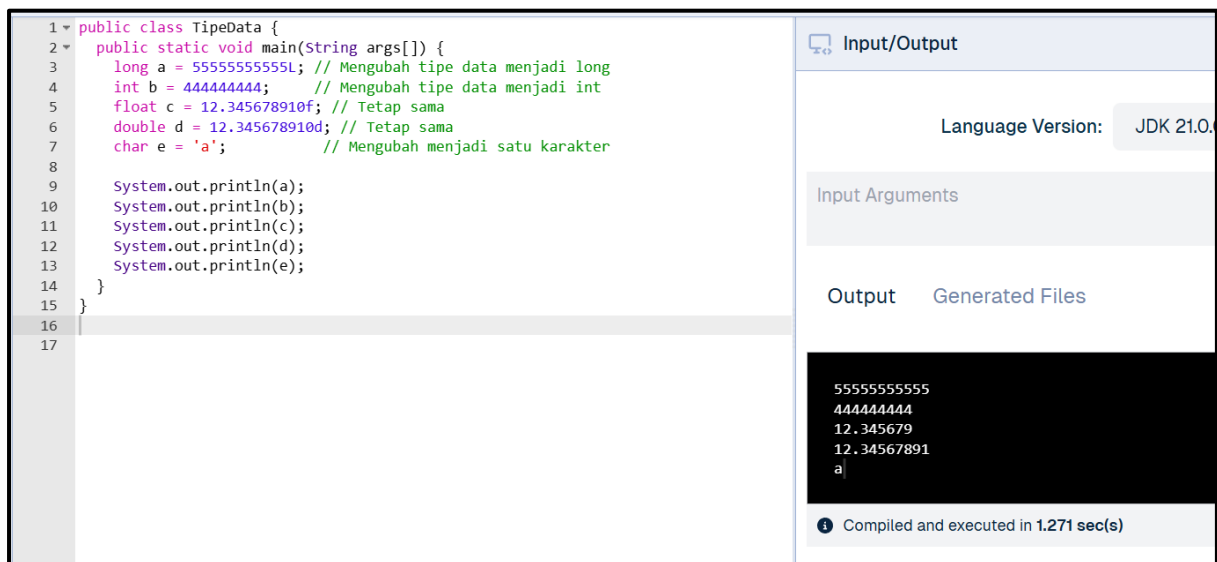
- Mengganti tipe data variabel `a` menjadi `long`, `b` menjadi `int`, dan mengubah karakter literal `e` menjadi satu karakter.

[No. 2.1] Penyusunan Algoritma dan Kode Program:

1) Algoritma:

- (a) Deklarasikan variabel `a` sebagai `long`.
- (b) Deklarasikan variabel `b` sebagai `int`.
- (c) Deklarasikan variabel `c` sebagai `float` dan variabel `d` sebagai `double`.
- (d) Deklarasikan variabel `e` sebagai `char` dengan satu karakter.
- (e) Cetak nilai semua variabel.
- (f) Jalankan program untuk menampilkan hasil.

2) Kode program dan luaran:



The screenshot shows an IDE with a Java class named `TipeData`. The code declares variables `a` (long), `b` (int), `c` (float), `d` (double), and `e` (char), and prints their values. The output window on the right shows the results: `5555555555`, `444444444`, `12.345679`, `12.34567891`, and `a`. The IDE also shows the language version as JDK 21.0 and the execution time as 1.271 seconds.

```
1 public class TipeData {  
2     public static void main(String args[]) {  
3         long a = 5555555555L; // Mengubah tipe data menjadi long  
4         int b = 444444444; // Mengubah tipe data menjadi int  
5         float c = 12.345678910f; // Tetap sama  
6         double d = 12.345678910d; // Tetap sama  
7         char e = 'a'; // Mengubah menjadi satu karakter  
8  
9         System.out.println(a);  
10        System.out.println(b);  
11        System.out.println(c);  
12        System.out.println(d);  
13        System.out.println(e);  
14    }  
15 }  
16  
17
```

Input/Output

Language Version: JDK 21.0

Input Arguments

Output Generated Files

```
5555555555  
444444444  
12.345679  
12.34567891  
a
```

Compiled and executed in 1.271 sec(s)

b) Analisa luaran yang dihasilkan:

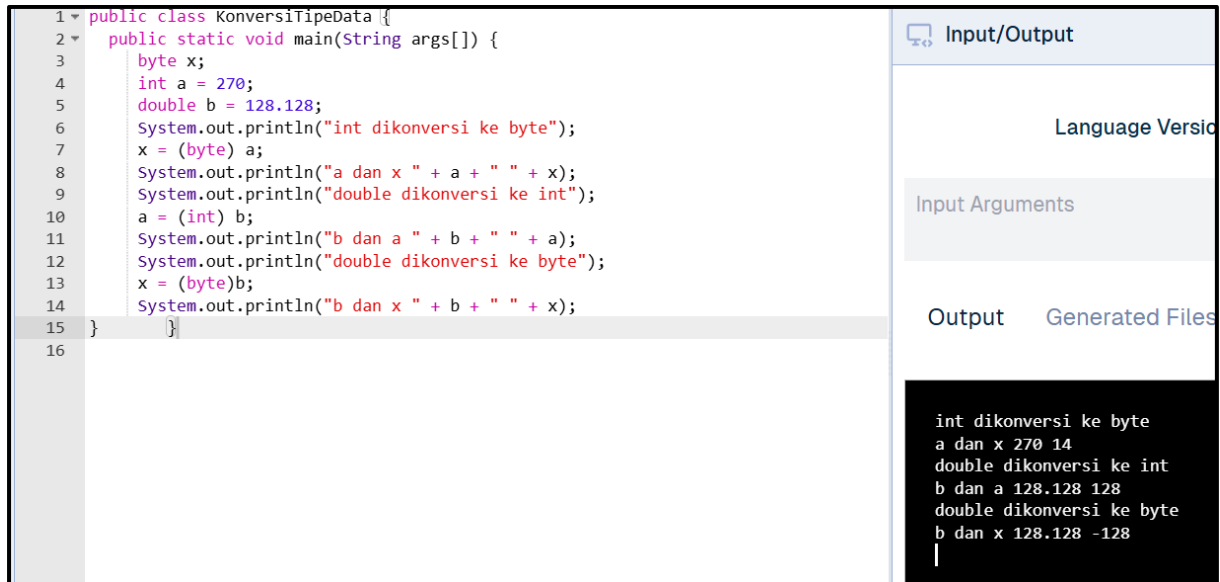
- Luaran sudah sesuai dengan program yang disusun. Tipe data yang ditampilkan telah sesuai dengan kapasitas masing-masing tipe data. Tidak ada kesalahan pada tipe data yang digunakan.

[No. 2.1] Kesimpulan:

1) Analisa:

- Pada program ini, saya menggunakan tipe data `long` untuk menangani nilai integer yang besar, dan tipe data `int` untuk menyimpan nilai yang sebelumnya terlalu besar untuk `byte`. Saya juga memperbaiki karakter literal `char` sehingga hanya terdiri dari satu karakter. Penyesuaian dilakukan karena Java memiliki batasan untuk setiap tipe data.

JAWABAN NO 3



```
1 public class KonversiTipeData {
2     public static void main(String args[]) {
3         byte x;
4         int a = 270;
5         double b = 128.128;
6         System.out.println("int dikonversi ke byte");
7         x = (byte) a;
8         System.out.println("a dan x " + a + " " + x);
9         System.out.println("double dikonversi ke int");
10        a = (int) b;
11        System.out.println("b dan a " + b + " " + a);
12        System.out.println("double dikonversi ke byte");
13        x = (byte)b;
14        System.out.println("b dan x " + b + " " + x);
15    }
16 }
```

Input/Output

Language Version

Input Arguments

Output Generated Files

```
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
```

[No. 3.1] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel:

- Diketahui dari soal: Anda diminta mengidentifikasi data pribadi yang mencakup 8 tipe data primitif dan `String`. Contoh data: nama, umur, jenis kelamin, rata-rata NEM, tinggi badan, status menikah, dan sebagainya.

- Variabel yang digunakan:
- `String` untuk menyimpan nama dan alamat.
- `int` untuk umur dan NEM.
- `char` untuk jenis kelamin.
- `boolean` untuk status menikah.
- `float` dan `double` untuk tinggi dan berat badan.
- `byte`, `short`, dan `long` untuk keperluan tambahan yang sesuai

[No. 3.1] Analisis dan Argumentasi:

1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara:

- Menyusun program yang mencakup 8 tipe data primitif (`byte`, `short`, `int`, `long`, `float`, `double`, `char`, `boolean`) dan `String` untuk data pribadi.

2) Alasan solusi ini karena:

- Setiap tipe data digunakan untuk menyimpan informasi pribadi sesuai dengan batasan dan kebutuhan kapasitas memori.

3) Perbaiki kode program dengan cara:

- Menyusun variabel untuk menyimpan informasi pribadi dan mencetaknya ke layar menggunakan `System.out.println`.

[No. 3.2] Penyusunan Algoritma dan Kode Program:

1) Algoritma:

- (a) Deklarasikan variabel untuk data pribadi (nama, umur, jenis kelamin, tinggi, berat badan, dan lainnya).
- (b) Inisialisasi variabel dengan informasi pribadi.
- (c) Cetak variabel ke layar menggunakan `System.out.println`.
- (d) Jalankan program dan cek hasil keluaran.

2) Kode program dan luaran:

```
1 public class DataPribadi {
2     public static void main(String[] args) {
3         // 8 Tipe data primitif dan String
4         String nama = "Tiara Yemelda";
5         int umur = 20;
6         char jenisKelamin = 'P'; // 'L' untuk laki-laki, 'P' untuk perempuan
7         float tinggiBadan = 158;
8         double beratBadan = 50;
9         boolean sudahMenikah = false;
10        byte jumlahSaudara = 2;
11        short tahunLahir = 2004;
12        long nomorTelepon = 628123456789L;
13
14        // Output data pribadi
15        System.out.println("Nama: " + nama);
16        System.out.println("Umur: " + umur);
17        System.out.println("Jenis Kelamin: " + jenisKelamin);
18        System.out.println("Tinggi Badan: " + tinggiBadan + " cm");
19        System.out.println("Berat Badan: " + beratBadan + " kg");
20        System.out.println("Sudah Menikah: " + (sudahMenikah ? "Ya" : "Tidak"));
21        System.out.println("Jumlah Saudara: " + jumlahSaudara);
22        System.out.println("Tahun Lahir: " + tahunLahir);
23        System.out.println("Nomor Telepon: " + nomorTelepon);
24    }
25 }
26
```

Input/Output

Language Version: JDK 21.0.0

Input Arguments

Output Generated Files

```
Nama: Tiara Yemelda
Umur: 20
Jenis Kelamin: P
Tinggi Badan: 158.0 cm
Berat Badan: 50.0 kg
Sudah Menikah: Tidak
Jumlah Saudara: 2
Tahun Lahir: 2004
Nomor Telepon: 628123456789
|
```

Compiled and executed in 1.332 sec(s)

b) Analisa luaran yang dihasilkan:

- Program berhasil menampilkan informasi data pribadi dengan menggunakan tipe data yang sesuai. Tidak ada kesalahan sintaks atau runtime. Semua tipe data bekerja dengan baik untuk menyimpan data yang diinput.

[No. 3.2] Kesimpulan:

1) Analisa:

- Program ini menggunakan berbagai tipe data primitif (`byte`, `short`, `int`, `long`, `float`, `double`, `char`, `boolean`) dan `String` untuk menyimpan informasi pribadi. Pemilihan tipe data berdasarkan kebutuhan kapasitas dan jenis data.

JAWABAN NO 4

[No. 4.1] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel:

- Pada latihan ini, kita diminta untuk mengonstruksikan kode program konversi tipe data dari Latihan 3, menggunakan tipe data lain yang kompatibel.

- Diketahui: Variabel-variabel dengan berbagai tipe data (`byte`, `short`, `int`, `long`, `float`, `double`, `char`) akan dikonversi ke tipe data lain yang sesuai.

[No. 4.1] Analisis dan Argumentasi:

1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara:

- Melakukan konversi antara tipe data yang kompatibel, baik secara implisit (konversi otomatis oleh compiler) maupun eksplisit (konversi dengan casting).

2) Alasan solusi ini karena:

- Konversi tipe data diperlukan saat kita perlu mengubah tipe data dari satu bentuk ke bentuk lain untuk kebutuhan tertentu dalam program.

- Ada dua jenis konversi:

- Implisit : Konversi otomatis yang dilakukan oleh Java saat tipe yang lebih kecil dikonversi ke tipe yang lebih besar.

- Eksplisit : Konversi yang dilakukan secara manual oleh programmer menggunakan casting, terutama saat mengubah tipe yang lebih besar menjadi tipe yang lebih kecil.

3) Perbaiki kode program dengan cara:

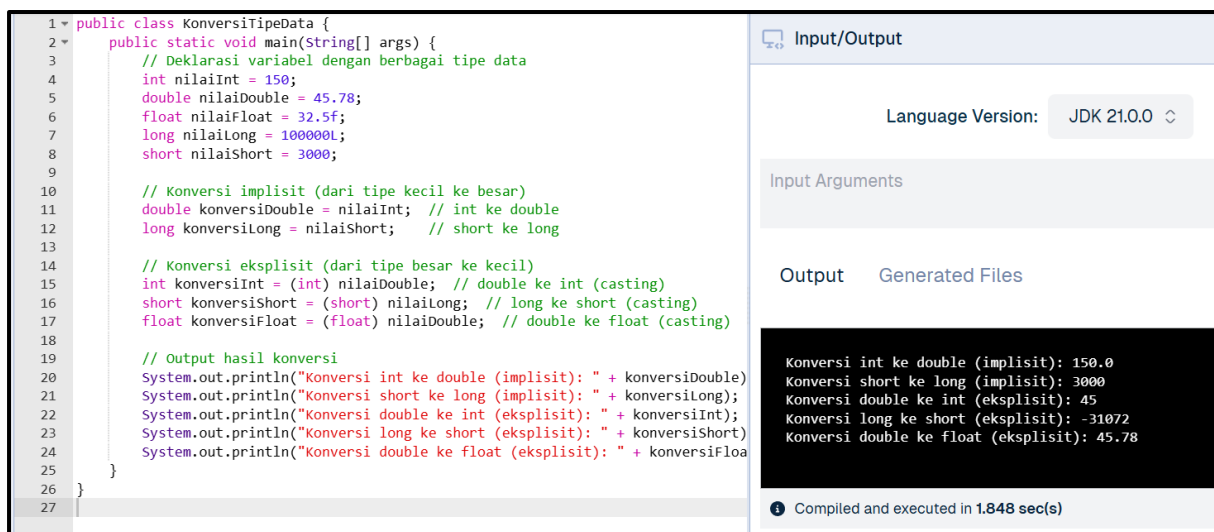
- Menyusun kode program yang menunjukkan beberapa contoh konversi dari satu tipe data ke tipe lain.

[No. 4.1] Penyusunan Algoritma dan Kode Program:

1) Algoritma:

- (a) Deklarasikan variabel dengan tipe data yang berbeda.
- (b) Lakukan konversi tipe data antara variabel-variabel tersebut, baik secara implisit maupun eksplisit.
- (c) Cetak hasil konversi ke layar.

2) Kode program dan luaran:



```
1 public class KonversiTipeData {
2     public static void main(String[] args) {
3         // Deklarasi variabel dengan berbagai tipe data
4         int nilaiInt = 150;
5         double nilaiDouble = 45.78;
6         float nilaiFloat = 32.5f;
7         long nilaiLong = 100000L;
8         short nilaiShort = 3000;
9
10        // Konversi implisit (dari tipe kecil ke besar)
11        double konversiDouble = nilaiInt; // int ke double
12        long konversiLong = nilaiShort; // short ke long
13
14        // Konversi eksplisit (dari tipe besar ke kecil)
15        int konversiInt = (int) nilaiDouble; // double ke int (casting)
16        short konversiShort = (short) nilaiLong; // long ke short (casting)
17        float konversiFloat = (float) nilaiDouble; // double ke float (casting)
18
19        // Output hasil konversi
20        System.out.println("Konversi int ke double (implisit): " + konversiDouble);
21        System.out.println("Konversi short ke long (implisit): " + konversiLong);
22        System.out.println("Konversi double ke int (eksplisit): " + konversiInt);
23        System.out.println("Konversi long ke short (eksplisit): " + konversiShort);
24        System.out.println("Konversi double ke float (eksplisit): " + konversiFloat);
25    }
26 }
27
```

Input/Output

Language Version: JDK 21.0.0

Input Arguments

Output Generated Files

```
Konversi int ke double (implisit): 150.0
Konversi short ke long (implisit): 3000
Konversi double ke int (eksplisit): 45
Konversi long ke short (eksplisit): -31072
Konversi double ke float (eksplisit): 45.78
```

Compiled and executed in 1.848 sec(s)

b) Analisa luaran yang dihasilkan:

- Program berhasil menunjukkan konversi antara tipe data yang berbeda, baik secara implisit maupun eksplisit. Tidak ada kesalahan dalam proses konversi.

[No. 4.2] Kesimpulan:

1) Analisa:

- Pada program ini, konversi implisit terjadi ketika tipe data yang lebih kecil (misalnya `int` atau `short`) secara otomatis dikonversi ke tipe data yang lebih besar (`double`, `long`).
- Konversi eksplisit terjadi saat kita mengubah tipe data yang lebih besar (misalnya `double` atau `long`) menjadi tipe data yang lebih kecil (`int`, `short`) dengan menggunakan casting. Casting ini diperlukan untuk menghindari hilangnya presisi atau overflow.

2) Evaluasi:

- Konversi implisit tidak membutuhkan casting karena aman dilakukan dari tipe data kecil ke besar.

- Konversi eksplisit membutuhkan casting manual, terutama untuk memastikan tidak terjadi overflow atau kehilangan presisi saat mengonversi dari tipe besar ke tipe kecil.

REFLEKSI :

Pada latihan ini, saya belajar dan mendalami konsep dasar Java, khususnya terkait deklarasi variabel, tipe data primitif, dan konversi tipe data. Melalui berbagai latihan, saya memahami bagaimana Java menangani kesalahan sintaksis seperti string literal yang tidak tertutup dan kesalahan dalam penulisan tipe data yang tidak kompatibel. Hal ini menambah wawasan saya tentang bagaimana Java melakukan validasi kode program dan pentingnya mengikuti aturan penulisan kode yang tepat.

Selain itu, saya juga belajar tentang konversi tipe data baik secara implisit maupun eksplisit. Konversi implisit mempermudah proses pengubahan tipe data yang lebih kecil menjadi lebih besar tanpa membutuhkan intervensi manual, sedangkan konversi eksplisit mengharuskan saya untuk lebih berhati-hati, terutama saat tipe data yang lebih besar dikonversi ke tipe yang lebih kecil.

Tantangan terbesar yang saya hadapi adalah memastikan setiap tipe data yang digunakan dalam program sesuai dengan kapasitas dan presisi yang dibutuhkan, terutama dalam kasus konversi eksplisit, di mana ada risiko kehilangan data atau presisi. Secara keseluruhan, pengalaman belajar ini memberikan pemahaman lebih dalam tentang aturan dan karakteristik Java, serta pentingnya struktur kode yang benar dalam pemrograman.