|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **MEIDA DINAFANI**  **G1F024058** | **Tipe Data** | **30 Agustus 2024** |

**[No. 1] Identifikasi Masalah:**

Salin dan tempel potongan kode ini ke Eclipse atau Jdoodle.

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!

Penyebab kesalahan terjadi karena ada beberapa hal yang perlu ditambahkan

Supaya program dapat berjalan. Seperti menambahkan tanda petik dua, dan

menambahkan tanda titik koma pada setiap akhirannya.

1. Ubah teks yang ditampilkan program menjadi nama lengkap Anda.

Pertama ubah program dari ("Halo Mahasiswa UNIB)menjadi ("Nama: Meida Dinafani"); Supaya program dapat berjalan.

1. Tambahkan baris System.out.println(“”}; untuk diisi dengan data alamat, dan jenis kelamin.

System.***out***.println("Nama: Meida Dinafani");

System.***out***.println("Alamat: Pagar Batu");

System.***out***.println("Jenis Kelamin: Perempuan");

**[No.1] Analisis dan Argumentasi**

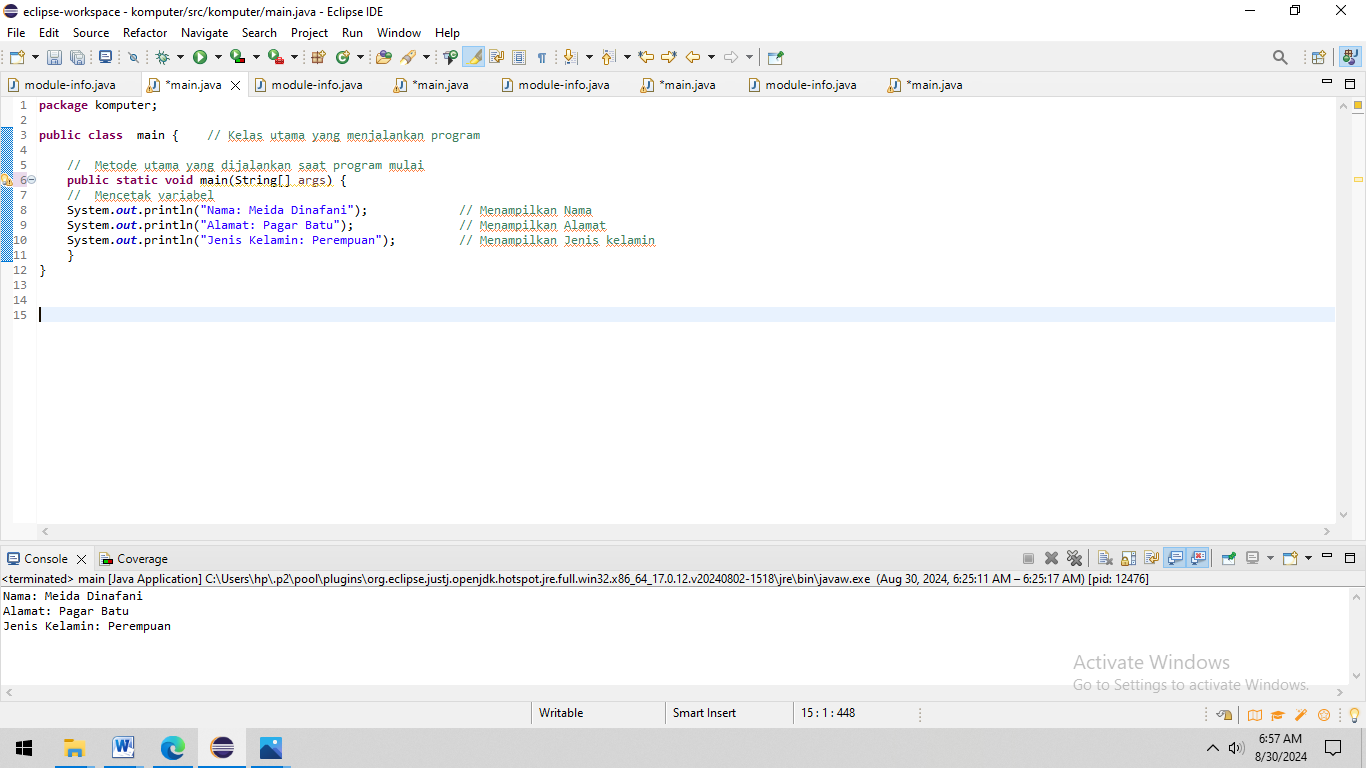
1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara Penambahan beberapa program yang kurang supaya program dapat berjalan.
2. Alasan solusi ini karena supaya program dapat berjalan dan tidak error setelah dicoba.
3. Perbaikan kode program dengan cara menambahkan program yang kurang seperti

System.out.println("Halo Mahasiswa UNIB) ini menambahkan tanda petik dua setelah kata

UNIB dan tanda titik koma pada akhirannya menjadi System.out.println("Halo Mahasiswa UNIB”);

**[No.1 ] Penyusunan Algoritma dan Kode Program**

1. Algoritma
2. Mulai
3. Deklerasikan Variabel
4. Masukkan nilai ke variabel
5. Cetak nilai dari variabel
6. Selesai
7. Kode program dan luaran
8. Screenshot Kode program, luaran, dan komentar



1. Analisa luaran yang dihasilkan

Luaran yang dihasilkan sudah sesuai dengan program yang disusun.   
Serta tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

Dengan hasil luaran :

Nama: Meida Dinafani

Alamat: Pagar Batu

Jenis Kelamin: Perempuan

**[No.1] Kesimpulan**

1. **Analisa**

Pada program ini, saya menggunakan bentuk kelas public karena Kelas dengan akses bentuk public diperlukan untuk program Java. Berdasarkan permasalahan, algoritma, dan kode program yang saya gunakan ada terdapat Program yang harus ditambahkan seperti penggunaan tanda petik dua dan pada setiap akhirannya ditambahkan tanda titik koma, agar program tidak terjadinya error. Program yang diberikan ini bertujuan untuk menampilkan beberapa informasi tentang Pengguna. Seperti : Nama, Alamat, dan Jenis kelamin. Selain itu, menambahkan spasi setelah tanda titik dua dalam output agar memberikan keterbacaan yang lebih baik dan konsistensi dalam format output.

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **MEIDA DINAFANI**  **G1F024058** | **Tipe Data** | **30 Agustus 2024** |

**[No. 2] Identifikasi Masalah:**

Java memiliki 8 tipe data primitif dan non-primitif. Apabila diketahui data berikut:

1. 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.

Tipe data yang direkomendasikan untuk setiap nilai dalam Contoh 2 adalah sebagai berikut:

1. 5 : tipe data integer
2. ‘L’ : tipe data char

3. "mobil" : tipe data String

4. 5.0 : tipe data double

5. 5.0f : tipe data float

6. -5 : tipe data integer negatif

1. Simpulkan karakteristik penggunaan setiap tipe data!
2. 5 : int

Bilangan bulat dalam rentang nilai (-2.147.483.648 sampai 2.147.483.647).

1. ‘L’: char

Karakter yang menyimpan tipe data dalam panjang 16 bit.

1. "mobil": String

Karakter yang menyimpan tipe data dalam bentuk teks atau kalmiat.

1. 5.0 : double

Tipe data dalam bentuk desimal yang menyimpan data dalam panjang 64 bit.

1. 5.0f : float

Tipe data dalam bentuk desimal notasi float yang menyimpan data dalam panjang

32 bit.

1. -5: int

Bilangan bulat dalam bentuk negatif dalam rentang nilai (-2.147.483.648 sampai 2.147.483.647).

**[No.2] Analisis dan Argumentasi**

1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara menentukan tipe data yang

Sesuai dengan program data yang ditentukan.

1. Alasan solusi ini karena supaya program dapat berjalan dan tidak error setelah dicoba dan sesuai dengan tipe data.
2. Perbaikan kode program dengan cara menentukan tipe data yang sesuai yaitu :

Integer = 5

Char = 'L'

String = "mobil"

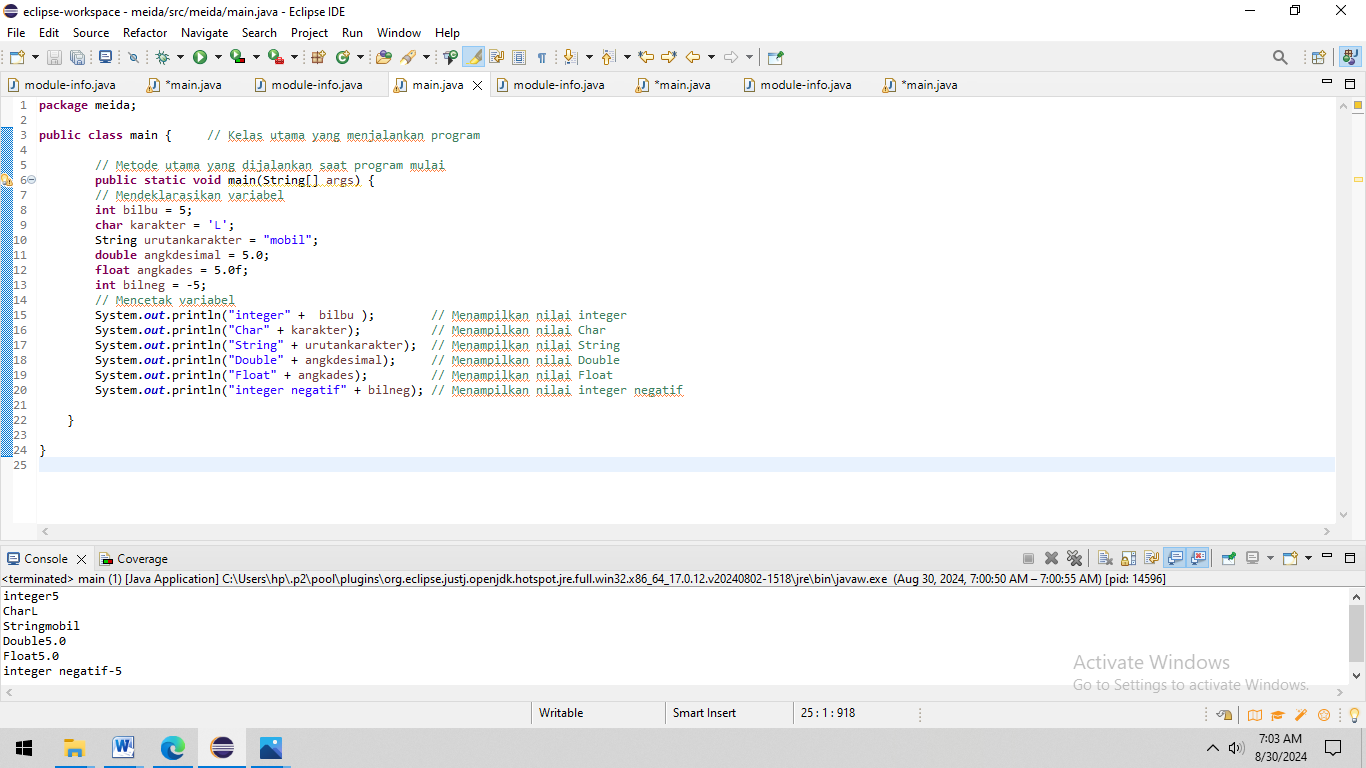
Double = 5.0

Float = 5.0f

Integer negatif = -5

**[No.2 ] Penyusunan Algoritma dan Kode Program**

1. Algoritma
2. Mulai
3. Deklarasi variabel dengan nilai
4. Cetak nilai dari variabel
5. Selesai
6. Kode program dan luaran
7. Screenshot Kode program, luaran, dan Komentar



1. Analisa luaran yang dihasilkan

Luaran yang dihasilkan sudah sesuai dengan program yang disusun.   
Serta tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

Dengan hasil luaran :

integer5

CharL

Stringmobil

Double5.0

Float5.0

integer negatif-5

**[No.2] Kesimpulan**

* 1. **Analisa**

Pada program ini, saya menggunakan bentuk kelas public karena Kelas dengan akses bentuk public diperlukan untuk program Java. Berdasarkan permasalahan, algoritma, dan kode program yang saya gunakan ada beberapa tipe data yang harus ditentukan terlebih dahulu termasuk ke tipe data yang mana seperti :

1. 5: tipe data integer
2. ‘L’: tipe data char

3. "mobil": tipe data String

4. 5.0: tipe data double

5. 5.0f: tipe data float

6. -5: tipe data integer negatif

Pada program ini, Saya mengambil alasan dengan solusi ini karena supaya program dapat berjalan dan tidak error setelah dicoba dan sesuai dengan tipe data.

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **MEIDA DINAFANI**  **G1F024058** | **Tipe Data** | **30 Agustus 2024** |

**[No. 3] Identifikasi Masalah:**

Salin dan tempel potongan kode ini ke Eclipse atau Jdoodle

public class TipeData {

public static void main(String args[]) {  
int a = 55555555555;  
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 = 55555555555;

/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!

1. int a = 55555555555;

Tipe data int hanya dapat menyimpan dalam rentang nilai

(-2.147.483.648 sampai 2.147.483.647). Nilai 55555555555 ini melebihi batas maksimum untuk int.

2. byte b = 4444444444;

Tipe data byte hanya dapat menyimpan dalam rentang nilai dari -128 hingga 127. Nilai 4444444444 melebihi rentang ini.

1. float c = 12.345678910f;

Tipe data float ini sudah benar, dan dapat menyimpan dalam panjang 32 bit.

1. double d = 12.345678910d;

Tipe data double ini sudah benar, dan dapat menyimpan dalam panjang 64 bit.

1. char e = 'abc';  
   Tipe data char hanya dapat menyimpan karakter tunggal, dan Cara penulisannya menggunakan tanda kutip tunggal ‘...’ serta hanya boleh berisi satu karakter.
2. Rekomendasikan tipe data yang sesuai untuk data tersebut!
3. int a = 55555555555;

Tipe data long untuk angka besar long a = 55555555555L

1. byte b = 4444444444;

Tipe data long untuk angka besar long b = 444444444L

1. float c = 12.345678910f;

Tipe data float untuk angka desimal dengan presisi lebih rendah float c = 12.3456789f;

1. double d = 12.345678910d;

Tipe data double untuk angka desimal dengan presisi lebih tinggi double d = 12.345678910;

1. char e = 'abc';

Tipe data char untuk satu karakter char e = 'a'

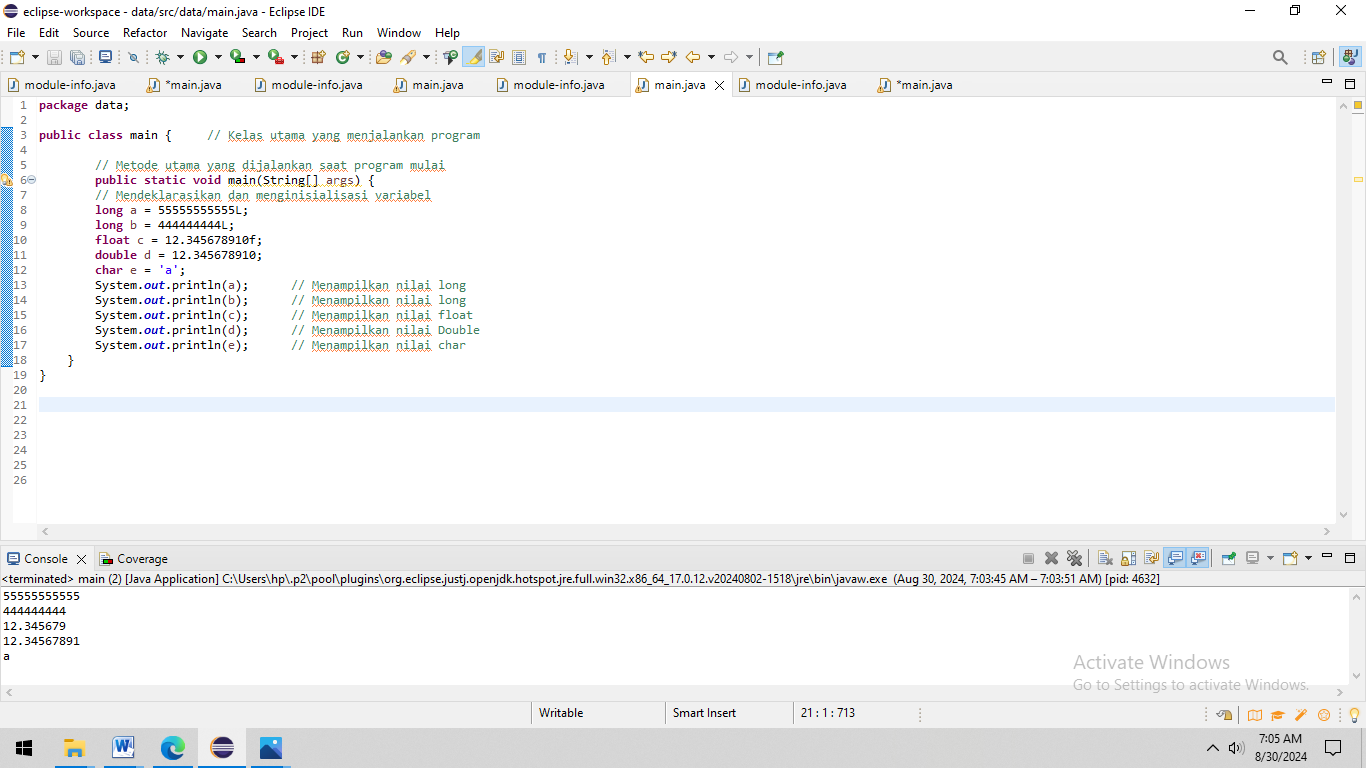
**[No.3] Analisis dan Argumentasi**

1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara menentukan tipe data yang sesuai untuk program tersebut.
2. Alasan solusi ini karena agar dapat mengatasi masalah terkait tipe data dan memastikan bahwa program dapat berfungsi dan tidak terjadi error.
3. Perbaikan kode program dengan cara :
4. menggunakan long untuk menyimpan nilai yang lebih besar dari batas kapasitas int, dan menambahkan sufiks L di akhir angka, seperti 55555555555L.
5. menggunakan long untuk menyimpan nilai yang lebih besar dari batas kapasitas byte, dan menambahkan sufiks L di akhir angka, seperti 4444444444L.
6. menggunakan float untuk angka desimal dengan presisi lebih rendah, serta menambahkan sufiks f di akhir angka , seperti 12.345678910f. tapi float hanya dapat menyimpan sekitar 7 digit desimal dengan akurasi yang memadai. Jadi, saat menggunakan float, nilai ini mungkin akan dibulatkan atau tidak sepenuhnya akurat.
7. menggunakan double untuk angka desimal dengan presisi lebih tinggi. serta tidak wajib menambahkan sufiks d di akhir angka , sehingga hanya 12.345678910.
8. Menggunakan char hanya untuk satu karakter, serta gunakan String untuk beberapa karakter.

**[No.3 ] Penyusunan Algoritma dan Kode Program**

* + - 1. Algoritma
         1. Mulai
         2. Deklarasi dan Inisialisasi Variabel
         3. Cetak Nilai Variabel
         4. Selesai
      2. Kode program dan luaran

Screenshot Kode program, hasil luaran, dan Komentar



Analisa luaran yang dihasilkan   
Luaran yang dihasilkan sudah sesuai dengan program yang disusun.   
Serta tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.

Dengan hasil luaran :

55555555555

444444444

12.345679

12.34567891

a

**[No.3] Kesimpulan**

* + 1. **Analisa**

Pada program ini, saya menggunakan bentuk kelas public karena Kelas dengan akses bentuk public diperlukan untuk program Java. Berdasarkan permasalahan, algoritma, dan kode program yang saya gunakan ditentukan berdasarkan nilai dan rentang data. menggunakan long untuk nilai besar dan menambahkan sufiks L di akhir angka .float untuk presisi lebih rendah, dan menambahkan sufiks f di akhir angka .double untuk presisi lebih tinggi, dan tidak wajib menambahkan sufiks d di akhir angka. menggunakan char hanya untuk satu karakter.

1. int a = 55555555555;

Tipe data long untuk angka besar long a = 55555555555L

1. byte b = 4444444444;

Tipe data long untuk angka besar long b = 444444444L

1. float c = 12.345678910f;

Tipe data float untuk angka desimal dengan presisi lebih rendah float c = 12.3456789f;

1. double d = 12.345678910d;

Tipe data double untuk angka desimal dengan presisi lebih tinggi double d = 12.345678910;

1. char e = 'abc';

Tipe data char untuk satu karakter char e = 'a'

Pada program ini, Saya mengambil alasan dengan solusi ini karena supaya program dapat berfungsi dengan benar, dan dengan tipe data yang tepat agar tidak terjadi error setelah dicoba.

Top of Form

Bottom of Form

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **MEIDA DINAFANI**  **G1F024058** | **Tipe Data** | **30 Agustus 2024** |

**[No. 4] Identifikasi Masalah:**

**Latihan 4:**

1. Rekomendasikan konversi tipe data pada Latihan 2 ke bentuk tipe data lain yang kompatibel.

untuk mengkonversi tipe data yang diberikan ke bentuk tipe data lain yang kompatibel, yakni perlu memeriksa tipe data, sumber, dan memastikan bahwa konversi tersebut valid. Rekomendasi konversi dari data yang diberikan sebagai berikut:

* 1. **Konversi dari** int **(5) ke byte**
  2. Konversi dari char (‘L’) ke char
  3. Konversi dari String ("mobil") ke string yang berisi angka ("123")
  4. Konversi dari double (5.0) ke double
  5. Konversi dari float (5.0f) ke float
  6. Konversi dari byte (-5) ke byte negatif

1. Simpulkan alasan jenis konversi tipe data tersebut!

Karena Konversi tipe data harus dilakukan dengan mempertimbangkan rentang nilai, presisi, dan potensi kehilangan informasi. Konversi otomatis umumnya aman untuk tipe data yang lebih besar ke tipe yang lebih kecil, sedangkan konversi sebaliknya memerlukan perhatian khusus untuk memastikan data tidak hilang atau terbuang. Selalu pertimbangkan casting eksplisit saat bekerja dengan tipe data yang berpotensi mengalami overflow atau pembulatan.

**[No.4] Analisis dan Argumentasi**

* 1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara menkonversi tipe data yang kompatibel.
  2. Alasan solusi ini karena agar dapat mengatasi masalah terkait tipe data dan memastikan bahwa program dapat berfungsi dan tidak terjadi error.
  3. Perbaikan kode program dengan cara:
     1. Konversi String:
* Mengganti "mobil" dengan string yang valid untuk konversi numerik ("123" dan "123.45").
* Menambahkan pengecekan jika stringValue tidak kosong sebelum konversi ke char.
  + 1. Konversi char:
* Menambahkan casting eksplisit ketika mengonversi dari char ke byte dan short untuk menghindari potensi kehilangan informasi.

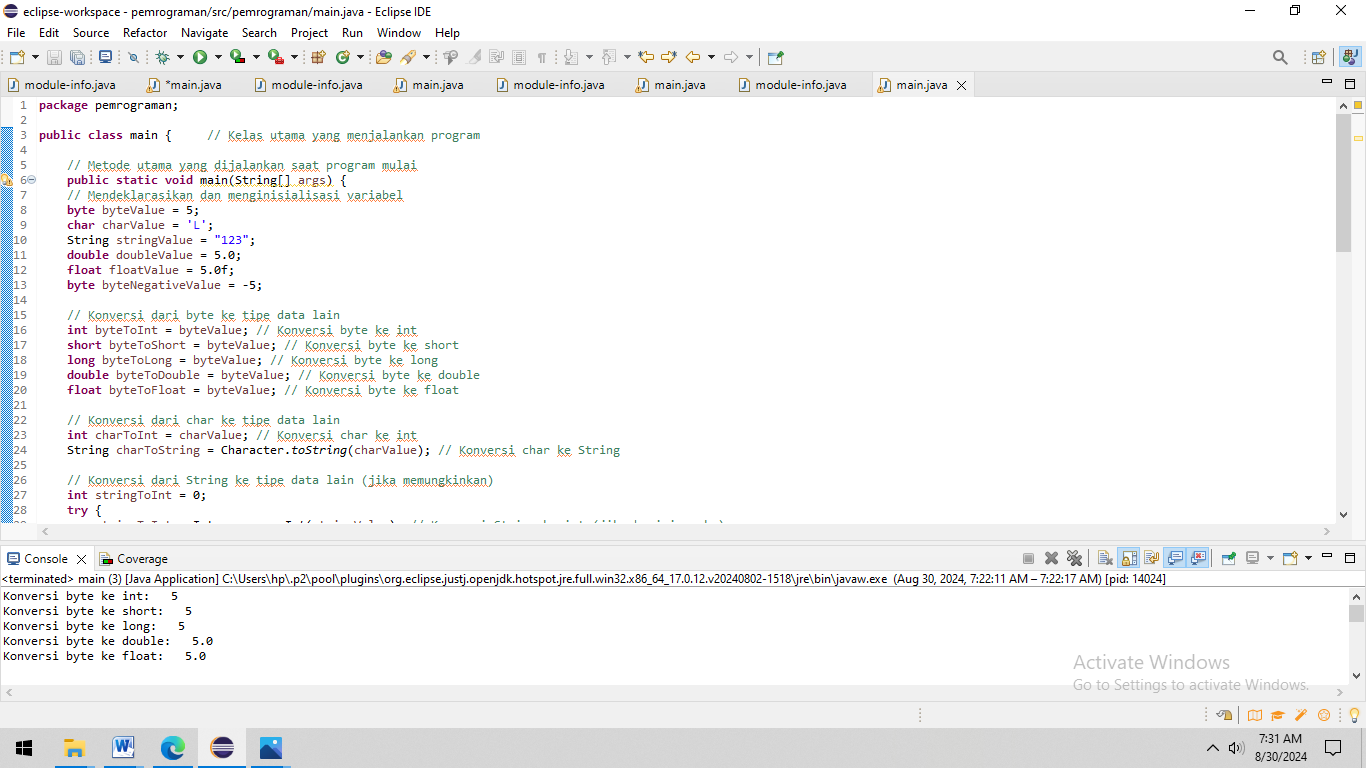
1. Casting Eksplisit:

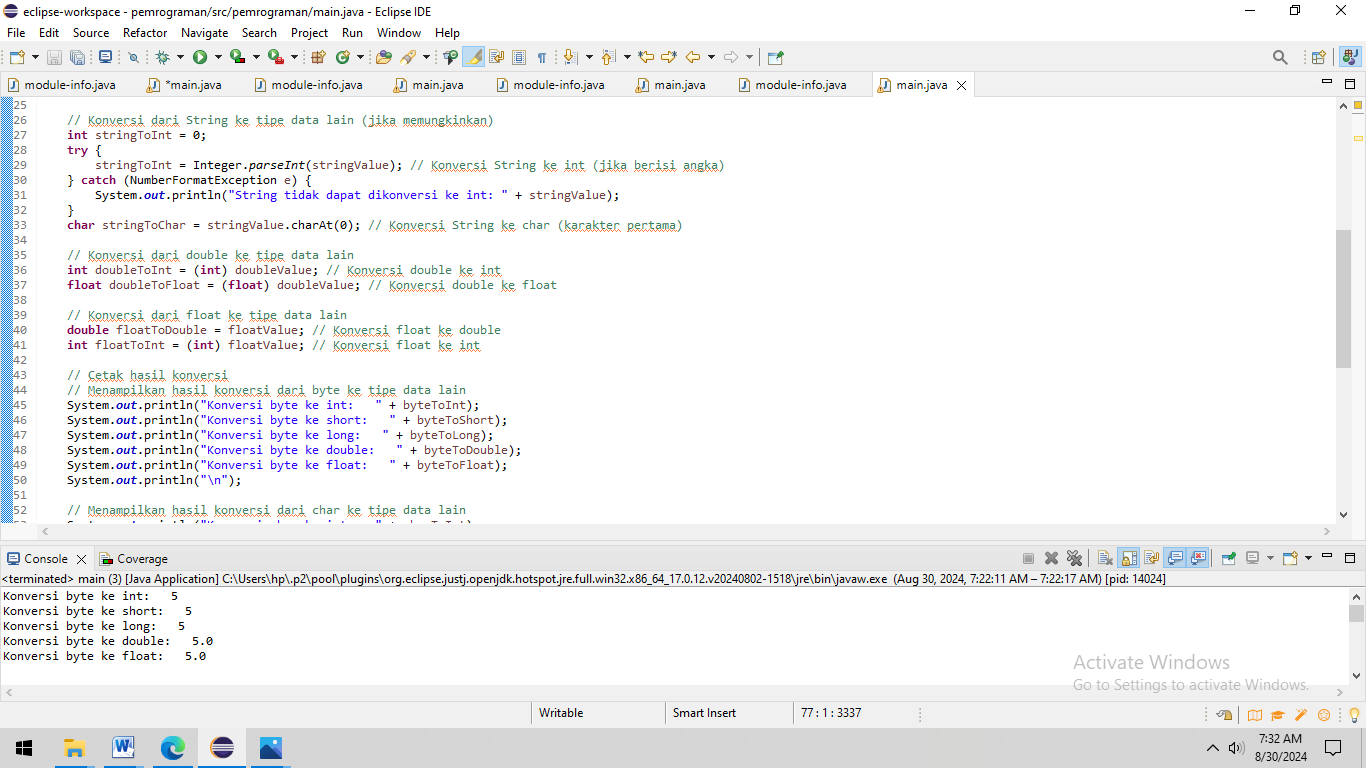
* Menggunakan casting eksplisit pada konversi yang dapat menyebabkan kehilangan informasi, seperti dari double ke byte atau short.

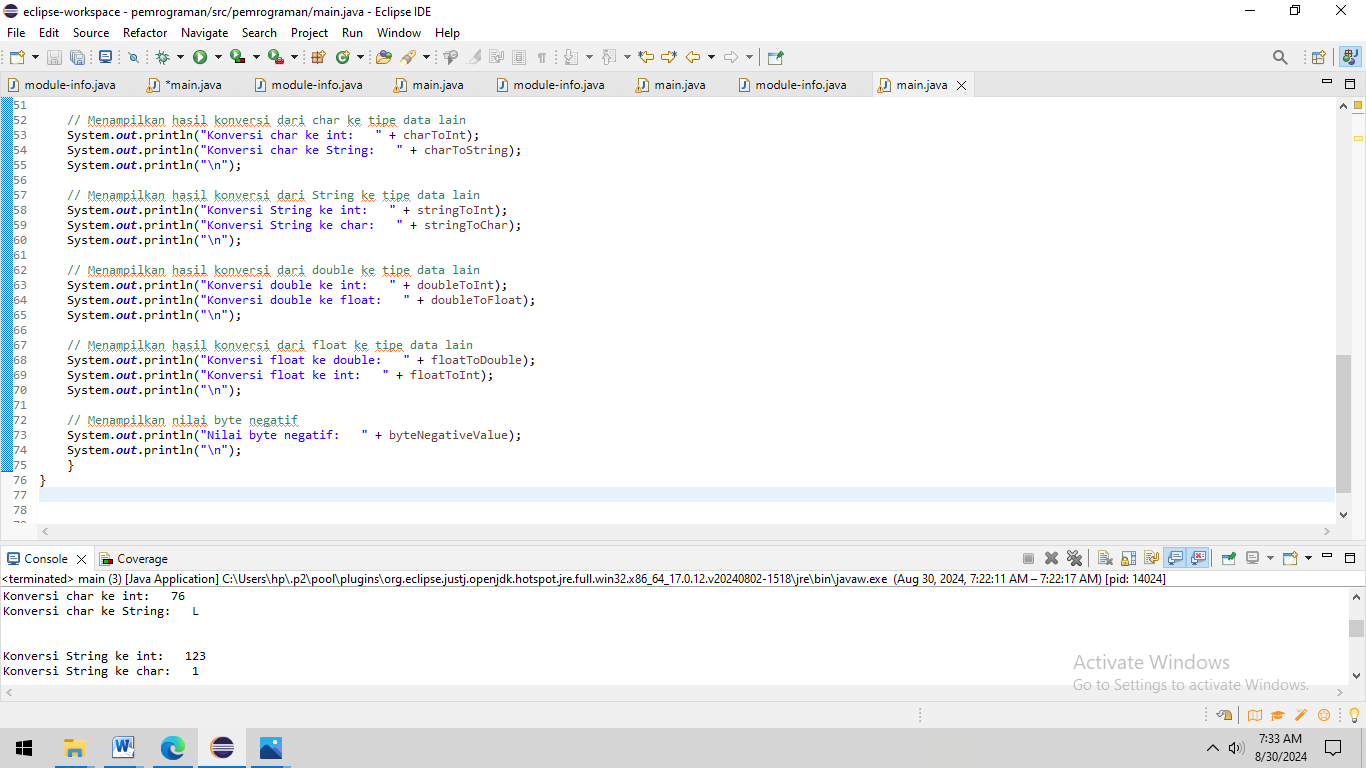
**[No.4 ] Penyusunan Algoritma dan Kode Program**

* + - 1. Algoritma
         1. Mulai
         2. Deklarasi dan Inisialisasi Variabel
         3. Konversikan ke tipe data
         4. Cetak hasil Konversi
         5. Selesai
      2. Kode program dan luaran

1. Screenshot Kode program, hasil luaran, dan Komentar







1. Analisa luaran yang dihasilkan

Luaran sudah sesuai dengan program yang disusun.   
Serta Tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan data.  
Dengan hasil luaran:

Konversi byte ke int: 5

Konversi byte ke short: 5

Konversi byte ke long: 5

Konversi byte ke double: 5.0

Konversi byte ke float: 5.0

Konversi char ke int: 76

Konversi char ke String: L

Konversi String ke int: 123

Konversi String ke char: 1

Konversi double ke int: 5

Konversi double ke float: 5.0

Konversi float ke double: 5.0

Konversi float ke int: 5

Nilai byte negatif: -5

**[No.4] Kesimpulan  
1. Analisa**

Pada program ini, saya menggunakan bentuk kelas public karena Kelas dengan akses bentuk public diperlukan untuk program Java. Berdasarkan permasalahan, algoritma, dan kode program yang saya gunakan ditentukan berdasarkan untuk konversi tipe data yang diberikan ke bentuk tipe data lain yang kompatibel, yakni perlu memeriksa tipe data, sumber, dan memastikan bahwa konversi tersebut valid nilai dan rentang data. Agar Pada program Konversi dapat berfungsi dengan benar, dan dengan tipe data yang tepat agar tidak terjadi error setelah dicoba.

Konversi dari tipe data:

* + 1. **Konversi dari** int **(5) ke byte**
    2. Konversi dari char (‘L’) ke char
    3. Konversi dari String ("mobil") ke string yang berisi angka ("123")
    4. Konversi dari double (5.0) ke double
    5. Konversi dari float (5.0f) ke float
    6. Konversi dari byte (-5) ke byte negative

**Refleksi**

Pengalaman belajar yang didapat yakni mendapatkan pelajaran dan pemahaman mengenai pemrograman tipe data yang ada di java meskipun ada yang susah dipahami. Pengetahuan yang di dapatkan yakni dapat mengetahui bagaimana program tipe data, bagaimana memperbaiki kode program, bagaimana membuat algoritma, dan menkonversi data yang ada di java. Serta mendapat pengetahuan juga mengenai Coding. Tantangan yang saya dihadapi yakni agak kesusahan dalam melakukan coding dan terkadang terjadi error, apalagi saya berasal dari sekolah Sma jadi belum tahu bagaimana Coding. Tapi agak sedikit membantu karena ada teman teman yang dari Smk yang sama sama mengajari juga.