Contoh Jawaban:

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Ariansyah Putra Ramadhan**  **G1F024080** | **Deklarasi dasar Java** | **30 Agustus 2024** |

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

Diketahui dari soal :

1.1  Evaluasi penyebab kesalahan terjadi dan perbaiki agar program dapat berjalan!  
1.2. Ubah teks yang ditampilkan program menjadi nama lengkap Anda.  
1.3. Tambahkan baris System.out.println(“”}; untuk diisi dengan data alamat, dan jenis kelamin.

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

1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara:

1. **Modifikasi Akses** private **pada Metode** main

2. **Kurangnya Tanda Titik Koma pada pernyataan** System.out.println("Halo Mahasiswa UNIB")

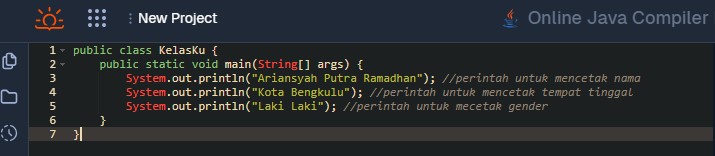
1. Alasan solusi ini karena
2. Metode private hanya dapat diakses dari dalam kelas itu sendiri, sehingga metode main tidak akan dapat diakses oleh JVM.
3. Tanda titik koma menandakan akhir dari suatu pernyataan dalam bahasa Java. Tanpa tanda titik koma, compiler akan menganggap pernyataan tersebut belum selesai.
4. Perbaikan kode program dengan cara
5. mengubah private menjadi public.
6. menambahkan tanda titik koma setelah System.out.println.

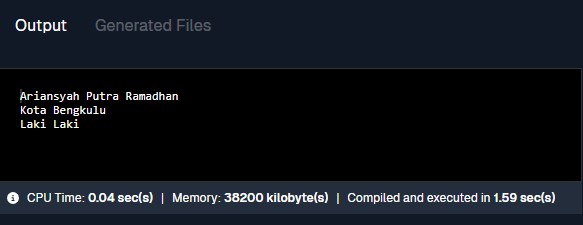
**[No.2 ] Penyusunan Algoritma dan Kode Program**

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

1. Mulai program.
2. Tampilkan teks "Ariansyah Putra Ramadhan" di layar.
3. Tampilkan teks "Kota Bengkulu" di layar.
4. Tampilkan teks "Laki Laki" di layar.
5. Akhiri program.
6. Kode program dan luaran
7. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot



1. Analisa luaran yang dihasilkan   
   Contoh:

Ketika kode ini dijalankan, program akan mencetak tiga baris teks ke konsol:

Ariansyah Putra Ramadhan

Kota Bengkulu

Laki Laki

**[No.1] Kesimpulan**

1. **Analisa kesalahan**
2. **Kesalahan Utama:**

Kesalahan utama dalam kode di atas terletak pada penggunaan kata kunci private pada metode main.

1. **Penjelasan Kesalahan:**

Metode main harus bersifat public: Metode main adalah titik masuk utama sebuah program Java. JVM (Java Virtual Machine) mencari dan menjalankan metode main secara otomatis ketika program dijalankan. Agar JVM dapat menemukan dan menjalankan metode main, metode ini harus memiliki akses public.

1. **Alasan public untuk main:**

Visibilitas: Kata kunci public membuat metode main dapat diakses dari mana saja, termasuk oleh JVM.

Titik Masuk: Metode main adalah pintu gerbang untuk memulai eksekusi program. Jika main bersifat private, JVM tidak akan dapat mengaksesnya.

1. Alasan Pengambilan Keputusan untuk Perbaikan:

Standar Java: Penggunaan public untuk metode main adalah konvensi yang sudah ditetapkan

dalam bahasa Java.

Fungsionalitas: Dengan membuat main bersifat public, kita memastikan bahwa program dapat

dijalankan dengan benar.

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Ariansyah Putra Ramadhan**  **G1F024080** | **Variabel dan tipe data** | **30 Agustus 2024** |

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

1. Diketahui dari soal :

2.1. Rekomendasikan tipe data yang tepat dari data Contoh 2.  
2.2. Simpulkan karakteristik penggunaan setiap tipe data!

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

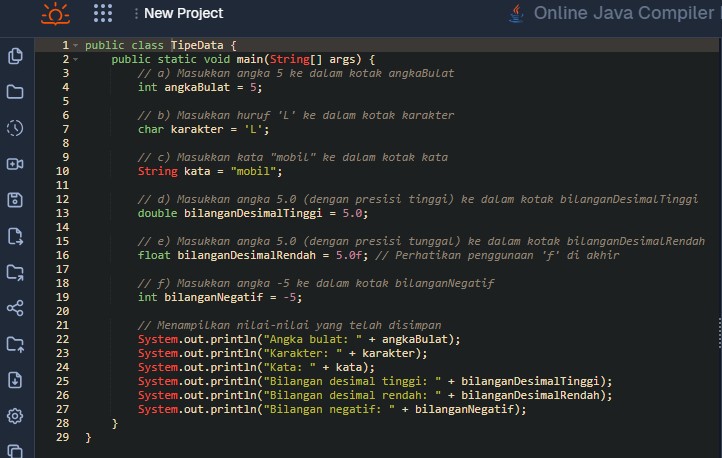
1. **5 adalah int** atau **Integer** Merupakan bilangan bulat. Tipe int adalah tipe primitif yang paling umum digunakan untuk bilangan bulat. Integer adalah wrapper class untuk int.
2. **‘L’ adalah char** atau **Character** Merupakan karakter tunggal. Tipe char digunakan untuk menyimpan karakter tunggal.
3. "mobil" adalah **String** Merupakan sekumpulan karakter (teks). Tipe String digunakan untuk menyimpan deretan karakter.
4. **5.0 adalah double** atau **Double** Merupakan bilangan desimal dengan presisi tinggi. Tipe double adalah tipe primitif yang umum digunakan untuk bilangan desimal.
5. **5.0f adalah float** atau **Float** Merupakan bilangan desimal dengan presisi tunggal. Suffix f menunjukkan bahwa nilai ini bertipe float.
6. **-5 adalah int** atau **Integer** Merupakan bilangan bulat negatif. Tipe int cocok untuk menyimpan bilangan bulat.

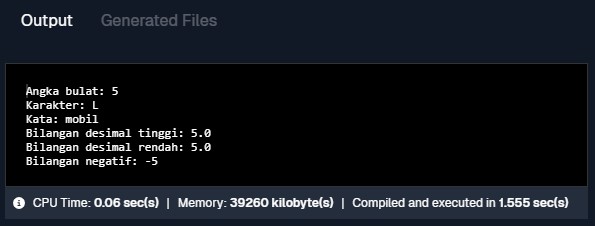
**[No.1 ] Penyusunan Algoritma dan Kode Program**

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

1. Masukkan angka 5 ke dalam kotak angkaBulat.
2. Masukkan huruf 'L' ke dalam kotak karakter.
3. Masukkan kata "mobil" ke dalam kotak kata.
4. Masukkan angka 5.0 (dengan presisi tinggi) ke dalam kotak bilanganDesimalTinggi.
5. Masukkan angka 5.0 (dengan presisi tunggal) ke dalam kotak bilanganDesimalRendah.
6. Masukkan angka -5 ke dalam kotak bilanganNegatif.
7. Kode program dan luaran
   1. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot



1. Analisa luaran yang dihasilkan

Kode di atas akan menghasilkan output berikut di konsol:

Angka bulat: 5

Karakter: L

Kata: mobil

Bilangan desimal tinggi: 5.0

Bilangan desimal rendah: 5.0

Bilangan negatif: -5

**[No.1] Kesimpulan**

* **Evaluasi**

1. Apa konsekuensi/dampak dari kode program yang dibuat?

Kode program yang saya berikan ini sangat sederhana dan tidak memiliki dampak yang signifikan. Namun, ini adalah dasar yang baik untuk memahami konsep pemrograman yang lebih kompleks. Dengan menambahkan fitur-fitur baru, kita dapat membuat program yang memiliki dampak yang lebih besar.

1. Evaluasi input program, proses perhitungan, dan luaran yang dihasilkan! (jika ada)

### Input Program :

### **Tidak ada input eksplisit:** Program ini tidak meminta pengguna untuk memasukkan data apa pun. Semua nilai variabel sudah ditentukan langsung dalam kode.

### Proses Perhitungan

**Tidak ada perhitungan:** Program ini tidak melakukan perhitungan aritmatika atau logika yang kompleks. Fungsi utamanya adalah mendeklarasikan variabel dan mencetak nilai variabel tersebut.

### Luaran Program :

**Output ke konsol:** Program akan mencetak nilai dari setiap variabel ke layar (konsol). Output yang dihasilkan adalah:

Angka bulat: 5

Karakter: L

Kata: mobil

Bilangan desimal tinggi: 5.0

Bilangan desimal rendah: 5.0

Bilangan negatif: -5

Contoh Jawaban:

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Ariansyah Putra Ramadhan**  **G1F024080** | **Variabel dan tipe data** | **30 Agustus 2024** |

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

1. Uraikan permasalahan dan variabel

Diketahui dari soal :

3.1. Evaluasi penyebab kesalahan pada Contoh 3!  
3.2. Rekomendasikan tipe data yang sesuai untuk data tersebut!

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

1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara :

Gunakan tipe data yang lebih besar: Untuk mengatasi overflow pada int dan byte, gunakan tipe data yang lebih besar seperti long atau int untuk a dan byte.

Gunakan char untuk karakter tunggal: Gunakan char untuk menyimpan karakter tunggal, seperti 'a', 'b', atau 'c'. Jika ingin menyimpan string yang terdiri dari beberapa karakter, gunakan tipe data String.

1. Alasan solusi ini karena :

Overflow pada int: Nilai 55555555555 terlalu besar untuk tipe data int. Ini akan menyebabkan overflow dan menghasilkan nilai yang tidak diharapkan.

Overflow pada byte: Nilai 4444444444 juga terlalu besar untuk tipe data byte.

Karakter tunggal pada char: Tipe data char hanya dapat menyimpan satu karakter tunggal, sedangkan 'abc' adalah string yang terdiri dari tiga karakter.

1. Perbaikan kode program dengan cara :

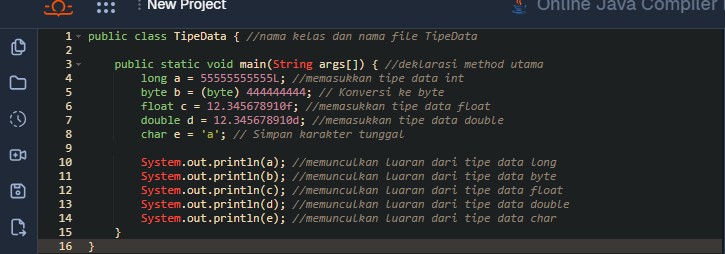
Gunakan tipe data yang lebih besar: Untuk mengatasi overflow pada int dan byte, gunakan tipe data yang lebih besar seperti long atau int untuk a dan byte.

Gunakan char untuk karakter tunggal: Gunakan char untuk menyimpan karakter tunggal, seperti 'a', 'b', atau 'c'. Jika Anda ingin menyimpan string yang terdiri dari beberapa karakter, gunakan tipe data String.

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

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

1. Cetak nilai variabel a ke konsol menggunakan System.out.println(a).
2. Cetak nilai variabel b ke konsol menggunakan System.out.println(b).
3. Cetak nilai variabel c ke konsol menggunakan System.out.println(c).
4. Cetak nilai variabel d ke konsol menggunakan System.out.println(d).
5. Cetak nilai variabel e ke konsol menggunakan System.out.println(e).
6. Kode program dan luaran
7. Screenshot/ Capture potongan kode dan hasil luaran Beri komentar pada kode yang di Screenshot



1. Analisa luaran yang dihasilkan

long a = 55555555555L;:

Tipe data long digunakan untuk menyimpan bilangan bulat yang sangat besar.

Nilai 55555555555L akan dicetak secara utuh karena long memiliki rentang nilai yang cukup luas.

byte b = (byte) 444444444;:

Tipe data byte hanya dapat menyimpan bilangan bulat kecil dalam rentang tertentu.

Karena nilai 444444444 terlalu besar untuk byte, akan terjadi overflow. Hasil yang sebenarnya akan tergantung pada implementasi Java.

float c = 12.345678910f;:

Tipe data float digunakan untuk menyimpan bilangan desimal dengan presisi tunggal.

Nilai 12.345678910f akan dicetak, tetapi mungkin tidak semua digit di belakang koma akan akurat karena keterbatasan presisi dari float.

double d = 12.345678910d;:

Tipe data double digunakan untuk menyimpan bilangan desimal dengan presisi ganda.

Nilai 12.345678910d akan dicetak dengan presisi yang lebih tinggi dibandingkan float.

char e = 'a';:

Tipe data char digunakan untuk menyimpan karakter tunggal.

Karakter 'a' akan dicetak secara langsung.

**[No.1] Kesimpulan**

1. **Evaluasi**
   1. **Apa konsekuensi/dampak dari kode program yang dibuat?**

### 1. **Overflow pada Tipe Data** byte

* **Penyebab:** Nilai 444444444 terlalu besar untuk dapat ditampung oleh tipe data byte yang memiliki rentang nilai yang sangat terbatas, yaitu -128 hingga 127.
* **Konsekuensi:** Ketika terjadi overflow, nilai yang sebenarnya akan dipotong dan hasilnya akan menjadi nilai negatif yang merupakan sisa dari pembagian dengan 256. Ini berarti nilai yang ditampilkan untuk variabel b tidak akan sesuai dengan nilai aslinya.

### 2. **Presisi pada Tipe Data** float **dan** double

* **Penyebab:** Tipe data float dan double memiliki batasan dalam representasi bilangan desimal.
* **Konsekuensi:** Ketika menyimpan bilangan desimal dengan presisi tinggi, beberapa digit di belakang koma mungkin hilang atau dibulatkan. Hal ini bisa menjadi masalah dalam perhitungan yang membutuhkan akurasi tinggi.

### 3. **Penggunaan Tipe Data yang Tepat**

* **Pentingnya pemilihan tipe data:** Memilih tipe data yang tepat sangat penting untuk menghindari kesalahan perhitungan dan penggunaan memori yang tidak efisien.
  1. **Evaluasi input program, proses perhitungan, dan luaran yang dihasilkan! (jika ada)**

### Input Program

* **Tidak ada input:** Program ini tidak menerima input dari pengguna. Semua nilai variabel sudah ditentukan langsung dalam kode.

### Proses Perhitungan

* **Tidak ada perhitungan kompleks:** Program ini tidak melakukan operasi matematika atau logika yang rumit. Fungsi utamanya adalah mendeklarasikan variabel dengan tipe data tertentu dan mencetak nilai variabel tersebut.
* **Konversi tipe data:** Terdapat satu konversi tipe data, yaitu dari int ke byte. Konversi ini dilakukan secara eksplisit menggunakan casting (byte). Namun, karena nilai 444444444 terlalu besar untuk tipe data byte, akan terjadi overflow.

### Luaran Program

* **Output ke konsol:** Program akan mencetak nilai dari setiap variabel ke layar.
  + **Variabel** a**:** Akan mencetak nilai 55555555555 karena tipe data long dapat menampung nilai sebesar itu.
  + **Variabel** b**:** Akan mencetak nilai yang dihasilkan dari overflow. Nilai ini akan tergantung pada implementasi Java.
  + **Variabel** c**:** Akan mencetak nilai 12.345679 atau nilai yang serupa. Karena keterbatasan presisi dari tipe data float, beberapa digit di belakang koma mungkin tidak akurat.
  + **Variabel** d**:** Akan mencetak nilai 12.34567891 atau nilai yang serupa. Tipe data double memiliki presisi yang lebih tinggi dibandingkan float, sehingga nilai akan lebih akurat.
  + **Variabel** e**:** Akan mencetak karakter a.

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Ariansyah Putra Ramadhan**  **G1F024080** | **Konversi Tipe Data** | **30 Agustus 2024** |

**[No. 1] Identifikasi Masalah:**  
Diketahui dari soal :

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

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

1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara mengkonversi tipe data yang terdapat pada Latihan dua ke tipe data double
2. Alasan solusi ini karena
3. **Ketika saya mengkonversi tipe data** int **ke** double **di Java, secara umum akan terjadi perluasan tipe data.** Artinya, nilai integer akan dipertahankan, tetapi representasinya akan berubah menjadi bilangan pecahan dengan bagian desimal yang bernilai nol.
4. **Ketika saya mengkonversi tipe data** byte **ke** double **di Java, secara umum akan terjadi perluasan (widening) tipe data.** Sama seperti ketika mengkonversi int ke double, nilai byte akan dipertahankan, namun representasinya akan berubah menjadi bilangan pecahan dengan bagian desimal yang bernilai nol.
5. Ketika saya mengkonversi tipe data float ke double hal yang akan terjadi adalah :

Nilai tetap: Nilai numerik dari variabel float akan tetap sama setelah dikonversi menjadi double.

Representasi berubah: Representasi internal dari nilai tersebut di memori akan berubah untuk mengakomodasi presisi yang lebih tinggi dari double.

1. Ketika saya mengkonversi tipe data double ke double hal yang akan terjadi adalah:

Tipe data yang sama: Baik sebelum maupun sesudah konversi, data tersebut tetap bertipe double. double adalah tipe data untuk merepresentasikan bilangan pecahan dengan presisi tinggi.

Tidak ada perluasan atau penyempitan: Karena kedua tipe data sama, tidak ada perluasan atau penyempitan rentang nilai.

Tidak ada perubahan nilai: Nilai numerik dari variabel double akan tetap sama setelah "dikonversi".

1. Ketika saya mengkonversi tipe data char ke double hal yang akan terjadi adalah:

Karakter ke Kode Numerik: Karakter yang saya konversi akan diubah menjadi kode numerik Unicode-nya.

Kode Numerik ke double: Kode numerik ini kemudian akan secara otomatis dipromosikan menjadi tipe data double. Artinya, nilai integer dari kode karakter tersebut akan menjadi bagian bilangan bulat dari bilangan desimal, sedangkan bagian desimalnya akan menjadi 0.

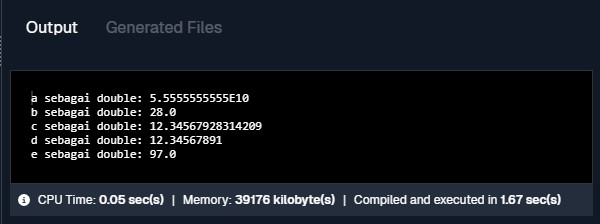
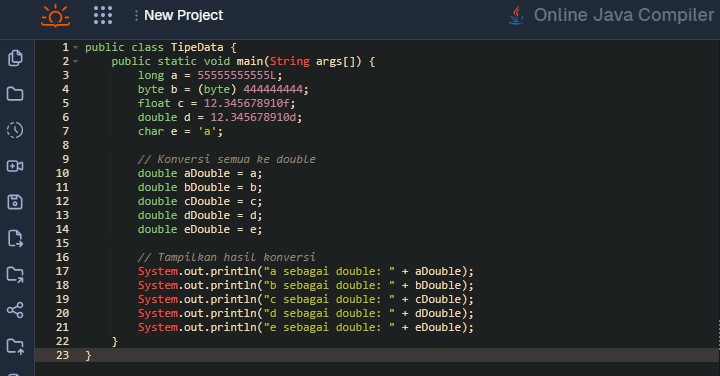
1. Perbaikan kode program dengan cara mengkonversi tiap tipe data yang terdapat pada latihan soal 2 menjadi tipe data double

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

1. Algoritma

Algoritma adalah langkah-langkah penyelesaian masalah.

1. Mulai: Program dimulai.
2. Deklarasi Variabel: Variabel-variabel dideklarasikan dan diinisialisasi.
3. Konversi ke Double: Variabel-variabel dikonversi ke double.
4. Tampilkan Hasil: Hasil konversi dicetak ke layar.
5. Selesai: Program berakhir.
6. Kode program dan luaran
7. Screenshot/ Capture potongan kode dan hasil luaran

Beri komentar pada kode yang di Screenshot

Analisa luaran yang dihasilkan   
a sebagai double: 5.55555555555E10:

* Nilai a (55555555555L) dikonversi menjadi double.
* Karena nilai ini terlalu besar untuk diwakili dalam notasi desimal biasa, Java menggunakan notasi ilmiah (5.55555555555E10) untuk representasinya.

b sebagai double: -120.0:

* Nilai b (444444444) dikonversi menjadi double.
* Namun, karena byte memiliki rentang nilai yang terbatas, nilai ini mengalami overflow (melebihi batas maksimum).
* Hasilnya adalah nilai negatif yang dihitung berdasarkan modulo (sisa pembagian) dari rentang nilai byte.

c sebagai double: 12.34567891:

* Nilai c (12.345678910f) dikonversi menjadi double.
* Karena double memiliki presisi yang lebih tinggi daripada float, nilai ini dikonversi tanpa kehilangan presisi.

d sebagai double: 12.34567891:

* Nilai d (12.345678910d) sudah bertipe double, sehingga tidak terjadi konversi.

e sebagai double: 97.0:

* Karakter 'a' dikonversi menjadi nilai numerik Unicode-nya, yang adalah 97.
* Nilai ini kemudian dikonversi menjadi double, menghasilkan 97.0.

**[No.1] Kesimpulan**

1. **Evaluasi**
2. Apa konsekuensi/dampak dari kode program yang dibuat?
3. Overflow pada byte:

* Ketika mencoba menyimpan nilai 444444444 dalam byte, terjadi overflow karena nilai ini melebihi rentang maksimum byte.
* Hasilnya adalah nilai negatif (-120) yang dihitung berdasarkan modulo (sisa pembagian) dari rentang nilai byte.
* Ini dapat menyebabkan kesalahan logika dalam program jika nilai yang diharapkan adalah positif.

1. Kehilangan Presisi pada float:

* Meskipun float dikonversi ke double, ada kemungkinan kehilangan presisi jika nilai float memiliki terlalu banyak digit di belakang koma.
* Namun, dalam contoh ini, nilai c cukup kecil sehingga tidak terjadi kehilangan presisi yang signifikan.

1. Tidak Ada Dampak pada long, double, dan char:

* Konversi dari long ke double tidak menimbulkan masalah karena double memiliki rentang nilai yang lebih luas.
* Konversi dari double ke double tidak mengubah nilai.
* Konversi dari char ke double menghasilkan nilai numerik Unicode tanpa masalah.

1. Evaluasi input program, proses perhitungan, dan luaran yang dihasilkan! (jika ada)
2. **Evaluasi:**

Input:

* a: Nilai long yang besar (55555555555L)
* b: Nilai byte yang melebihi rentang (444444444)
* c: Nilai float dengan beberapa digit desimal (12.345678910f)
* d: Nilai double dengan beberapa digit desimal (12.345678910d)
* e: Karakter 'a'

1. **Proses Perhitungan:**

Konversi ke double:

* a dikonversi ke double tanpa masalah karena double memiliki rentang nilai yang lebih luas.
* b dikonversi ke double dengan overflow karena nilai melebihi rentang byte.
* c dikonversi ke double dengan kemungkinan kehilangan presisi kecil karena float memiliki presisi yang lebih rendah.
* d tidak perlu dikonversi karena sudah bertipe double.
* e dikonversi ke double untuk mendapatkan nilai numerik Unicode.

1. **Luaran:**

a sebagai double: 5.55555555555E10

b sebagai double: -120.0

c sebagai double: 12.34567891

d sebagai double: 12.34567891

e sebagai double: 97.0