|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Michelia Erza Annadhira**  **G1F024035** | **Deklarasi dasar Java** | **26 Agustus 2024** |
| **[1] Identifikasi Masalah:** | | |
| 1. Uraikan permasalahan dan variable   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.   Di soal masih terdapat kesalahan :   1. Exception in thread "main" java.lang.Error:  terjadi karena ‘main’ harus bersifat publik dan statik, serta memiliki return type ‘void’, tetapi di sini tidak ada modifier ‘public’ 2. String literal is not properly closed by a double-quote terjadi karena tidak adanya penutup tanda kutip ganda (“) pada string literal dalam metode ‘System.out.println’ 3. Menunjukkan bahwa kode tidak lengkap, karena terdapat kesalahan pada sintaksis yang menyebabkan blok kode tidak tertutup dengan benar 4. Rincikan sumber informasi yang relevan (buku / webpage)   [Video Bagian 1 – Pengenalan dan Struktur Program Java](http://youtu.be/OjY8ByJE_bA)    <https://youtu.be/OjY8ByJE_bA> | | |
| **[2] Analisis dan Argumentasi** | | |
| 1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara memperbaiki kesalahan pada penulisan string literal dan menambahkan tanda kurung kurawal serta titik koma. 2. Alasan solusi ini karena kode yang tidak lengkap dan kesalahan penulisan string literal sehingga menyebabkan compiler tidak dapat menjalankan program dengan benar. 3. Perbaikan kode program dengan cara menutup string literal dengan tanda kutip ganda dan memastikan semua blok kode ditutup dengan benar, serta menambahkan baris yang diperlukan untuk data tambahan. Seperti :  * Mengubah method main dari private menjadi public   private static void main(String[] args) {  Menjadi  public static void main(String[] args) {   * Menambahkan tanda kutip (”) dan titik koma (;) pada baris    System.out.println("Halo Mahasiswa UNIB)  Menjadi   System.out.println("Halo Mahasiswa UNIB”); | | |
| **[3] Penyusunan Algoritma dan Kode Program** | | |
| * Algoritma  1. Baca dan analisis kesalahan yang dihasilkan struktur kode program 2. Perbaiki kesalahan method main ‘private’ menjadi ‘public’   private static void main(String[] args) { Menjadi  public static void main(String[] args) { 3. Perbaiki kesalahan yang disebabkan oleh literal string yang tidak ditutup dengan tanda kutip ganda (“)  System.out.println("Halo Mahasiswa UNIB) Menjadi  System.out.println("Halo Mahasiswa UNIB”) 4. Tambahkan tanda titik koma pada akhir string untuk menutupnya  System.out.println("Halo Mahasiswa UNIB”) Menjadi  System.out.println("Halo Mahasiswa UNIB”); 5. Pastikan tanda kurung kurawal terpasang dengan benar 6. Mengubah teks yang ditampilkan program menjadi nama lengkap Anda System.out.println("Halo Mahasiswa UNIB”); Menjadi System.out.println("Nama : Michelia Erza Annadhira”); 7. Tambahkan baris System.out.println(“”}; untuk diisi dengan alamat data alamat, dan jenis kelamin 8. Jalankan kode untuk memeriksa apakah program bias dijalankan dengan baik dan tidak ada kesalahan 9. Simpan kode program yang telah diperbaiki dan catat perubahan yang telah dilakukan   WhatsApp Image 2024-08-28 at 22.50.20  Luaran (output) :  WhatsApp Image 2024-08-28 at 22.50.51  Kode program sudah disusun dengan struktur yang benar. Hal ini dapat dilihat dari luaran (output) yang sesuai dengan kode program dan tidak terjadi eror lagi | | |
| **[4] Kesimpulan** | | |
| 1. Pada program itu saya menggunakan bentuk kelas ‘public’ karena kelas utama dalam Java harus dideklarasikan sebagai ‘public’ agar dapat diakses dari luar dan untuk memastikan metode ‘main’ dapat diakses oleh JVM saat eksekusi program. 2. Perbaikan program dengan menambahkan tanda kutip ganda di akhir string karena struktur Java mengharuskan setiap string literal ditutup dengan tanda kutip ganda. Selain itu, saya juga menambahkan titik koma (‘;’) untuk menyelesaikan pernyataan ‘System.out.printlnl’ dan mengubah akses modifier metode ‘main’ dari ‘private’ menjadi ‘public’ agar program dapat dijalankan dengan benar. | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Michelia Erza Annadhira**  **G1F024035** | **Variable dan tipe data** | **26 Agustus 2024** |
| **[1] Identifikasi Masalah:** | | |
| 1. Uraikan permasalahan dan variable   Diketahui dari soal :   1. 5 2. ‘L’ 3. “mobil” 4. 5.0 5. 5.0f 6. -5   **Latihan 2:** 2.1. Rekomendasikan tipe data yang tepat dari data Contoh 2   1. 5 Tipe data ‘int’, alasannya karena angka 5 adalah bilangan bulat tanpa decimal yang berada dalam rentang tipe data ‘int’. 2. ‘L’ Tipe data ‘char’, alasannya karena ‘L’ adalah karakter tunggal. 3. “mobil” Tipe data ‘String’, alasannya karena “mobil” adalah urutan karakter yang direpresentasikan sebagai string dalam Java. 4. 5.0 Tipe data ‘double’, alasannya karena 5.0 adalah bilangan desimal yang direpresentasikan sebagai tipe data ‘double’ secara default dalam Java. 5. 5.0f Tipe data ‘float’, alasannya karena 5.0f adalah bilangan desimal yang diakhiri dengan ‘f’, menunjukkan bahwa nilai tersebut adalah tipe data ‘float’. 6. -5 Tipe data ‘int’, alasannya karena -5 adalah bilangan bulat negatif yang berada dalam rentang tipe data ‘int’   2.2.  Simpulkan karakteristik penggunaan setiap tipe data!   * ‘int’ Tipe data ‘int’ digunakan untuk menyimpan bilangan bulat (baik positif maupun negatif) tanpa desimal. Ukurannya adalah 32-bit. * ‘char’ Tipe data ‘char’ digunakan untuk menyimpan satu karakter Unicode. Ukurannya adalah 16-bit, dan dapat menyimpan karakter huruf, angka, atau simbol. * ‘String’ Tipe data ‘String’ digunakan untuk menyimpan urutan karakter atau teks. Meskipun bukan tipe data primitif, ‘String’ sangat sering digunakan untuk memanipulasi dan menyimpan teks. * ‘double’ Tipe data ‘double’ digunakan untuk menyimpan bilangan desimal dengan presisi ganda (64-bit). Ini adalah tipe data default untuk angka dalam Java. * ‘float’ Tipe data ‘float’ digunakan untuk menyimpan bilangan desimal dengan presisi (32-bit). Meskipun kurang presisi dibandingkan ‘double’, ‘float’ digunakan saat penggunaan memori lebih renda lebih diutamakan.      1. Rincikan sumber informasi yang relevan (buku / webpage)   [Video Bagian 2 – Tipe Data dan Konversi Java](https://youtu.be/at27jZTFWe8)    <https://youtu.be/at27jZTFWe8> | | |
| **[2] Analisis dan Argumentasi** | | |
| 1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara memilih tipe data yang sesuai berdasarkan karakteristik nilai yang akan disimpan. Setiap nilai harus dideklarasikan dengan tipe data yang paling cocok untuknya (misalnya, menggunakan ‘int’ untuk bilangan bulat, ‘char’ untuk karakter tunggal, ‘String’ untuk teks, ‘double’ untuk bilangan desimal, dan ‘float’ untuk bilangan desimal dengan penggunaan memori yang lebih efisien). 2. Alasan solusi ini karena setiap tipe data di java memiliki batasan dan karakteristik yang dirancang untuk mengelola jenis data tertentu dengan efisien. Memilih tipe data yang tepat tidak hanya memastikan program berfungsi dengan benar tetapi juga meningkatkan efisiensi penggunaan memori dan kinerja program. 3. Perbaikan kode program dengan cara menggunakan tipe data yang sesuai untuk setiap nilai dan memastikan variabel dideklarasikan dan diinisialisasi dengan benar dalam program Java. | | |
| **[3] Penyusunan Algoritma dan Kode Program** | | |
| * Algoritma  1. Mulai 2. Identifikasi data yang diberikan dan karakteristiknya : 3. 5 adalah bilangan bulat 4. ‘L’ adalah karakter tunggal 5. “mobil” adalah urutan karakter (string) 6. 5.0 adalah bilangan decimal 7. 5.0f adalah bilangan desimal dengan akhiran ‘f’ yang menunjukkan tipe ‘float’ 8. -5 adalah bilangan bulat negatif 9. Tentukan tipe data yang sesuai : 10. Nilai 1 (‘5’) : tipe data ‘int’ karena bilangan bulat 11. Nilai 2 (‘‘L’‘) : tipe data ‘char’ karena merupakan karakter tunggal 12. Nilai 3 (‘“mobil”‘) : tipe daya ‘String’ karena urutan karakter (teks) 13. Nilai 4 (‘5.0’) : tipe data ‘double’ karena bilangan desimal 14. Nilai 5 (‘5.0f’) : tipe data ‘float’ karena bilangan desimal dengan suffix ‘f’, yang menunjukkan tipe data ‘float’ 15. Nilai 6 (‘-5’) : tipe data ‘int’ karena bilangan bulat negatif 16. Deklarasikan setiap variabel dengan tipe data yang sesuai berdasarkan nilai yang sudah ditentukan di langkah 3 : 17. int bilanganbulat = 5; 18. char karakter = ‘L’; 19. String teks = “mobil”; 20. double bilanganpecahan = 5.0; 21. float bilanganpecahanrendah = 5.0f; 22. int bilanganbulatnegatif = -5; 23. Gunakan perintah ‘System.out.println’ untuk mencetak nilai dari setiap variabel yang sudah dideklarasikan 24. Jalankan kode untuk memeriksa apakah program bias dijalankan dengan baik dan tidak ada kesalahan 25. Simpan kode program yang telah diperbaiki dan catat perubahan yang telah dilakukan   Screenshot (4)  Luaran (output) :  Screenshot (5)  Kode program sudah disusun dengan struktur yang benar. Hal ini dapat dilihat dari luaran (output) yang sesuai dengan kode program dan tidak terjadi eror lagi | | |
| **[4] Kesimpulan** | | |
| 1. Pada program itu saya menggunakan bentuk kelas ‘public’ karena kelas public diperlukan agar kode dapat diakses dari luar paket. Ini memastikan bahwa metode ‘main’ dapat diakses dan dieksekusi oleh JVM. 2. Perbaikan program dengan menambahkan tipe data yang seuai karena struktur Java mengharuskan bahwa setiap nilai disimpan dalam tipe data yang sesuai untuk memastikan operasi dilakukan dengan benar dan efisien. Java memerlukan deklarasi tipe data yang sesuai untuk menangani berbagai jenis data seperti bilangan bulat, karakter, dan teks. | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Michelia Erza Annadhira**  **G1F024035** | **Variabel dan tipe data** | **26 Agustus 2024** |
| **[1] Identifikasi Masalah:** | | |
| 1. Uraikan permasalahan dan variable   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:**  3.1. Evaluasi penyebab kesalahan pada Contoh 3! 3.2. Rekomendasikan tipe data yang sesuai untuk data tersebut!  Di soal masih terdapat kesalahan :   1. /TipeData.java:3: error: integer number too large   int a = 55555555555; Terjadi karena ‘int’ hanya mampu menyimpan nilai dalam rentang -2,147,483,648 hingga 2,147,647. Nilai ‘55555555555’ melebihi kapasitas tipe data ‘int’ 2. /TipeData.java:4: error: integer number too large    byte b = 4444444444; Terjadi karena ‘byte’ hanya mampu menyimpan nilai dalam rentang -128 hingga 127. Nilai ‘4444444444’ melebihi kapasitas tipe data ‘byte’ 3. /TipeData.java:7: error: unclosed character literal    char e = abc'; Terjadi karena ‘char’ hanya dapat menyimpan satu karakter tunggal, sedangakn “‘abc’” terdiri dari tiga karakter, sehingga menimbulkan eror 4. Rincikan sumber informasi yang relevan (buku / webpage)   [Video Bagian 2 – Tipe Data dan Konversi Java](https://youtu.be/at27jZTFWe8)    <https://youtu.be/at27jZTFWe8> | | |
| **[2] Analisis dan Argumentasi** | | |
| 1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara :  * Mengganti tipe data ‘int’ untuk variabel ‘a’ dengan ‘long’ * Mengganti tipe data ‘byte’ untuk variabel ‘b’ dengan ‘long’ * Mengganti nilai variebel ‘e’ menjadi 1 karakter tunggal  1. Alasan solusi ini karena :  * Tipe data ‘long’ mampu menyimpan bilangan bulat yang jauh lebih besar dibanding ‘int’ dan ‘byte’ * Tipe data ‘char’ hanya mampu menyimpan satu karakter  1. Perbaikan kode program dengan cara :  * Mengganti tipe data ‘int’ untuk variabel ‘a’ dengan ‘long’   int a = 55555555555; Menjadi long a = 55555555555L;   * Mengganti tipe data ‘byte’ untuk variabel ‘b’ dengan ‘long’ byte b = 4444444444; Menjadi long b = 4444444444L; * Mengganti nilai variebel ‘e’ menjadi 1 karakter tunggal char e = 'abc'; Menjadi char e = 'a'; | | |
| **[3] Penyusunan Algoritma dan Kode Program** | | |
| * Algoritma  1. Baca dan analisis kesalahan yang dihasilkan struktur kode program   Periksa nilai untuk variabel ‘a’ yang diberikan, jika nilai terlalu besar untuk ‘int’, gunakan ‘long’. Tambahan huruf ‘L’ di akhir nilai yang diberikan pada variabel ‘long’ untuk memastikan nilai tersebut diperlakukan sebagai ‘long’ int a = 55555555555; Menjadi long a = 55555555555L;   1. Periksa nilai untuk variabel ‘b’ yang diberikan, jika nilai terlalu besar untuk ‘byte’, gunakan ‘long’. Tambahan huruf ‘L’ di akhir nilai yang diberikan pada variabel ‘long’ untuk memastikan nilai tersebut diperlakukan sebagai ‘long’ byte b = 4444444444; Menjadi long b = 4444444444L; 2. Periksa nilai untuk variabel ‘e’ yang diberikan, ganti nilai variebel ‘e’ menjadi 1 karakter tunggal char e = 'abc'; Menjadi char e = 'a'; 3. Jalankan kode untuk memeriksa apakah program bias dijalankan dengan baik dan tidak ada kesalahan 4. Simpan kode program yang telah diperbaiki dan catat perubahan yang telah dilakukan   Screenshot (17)  Luaran (output) :  Screenshot (18)  Kode program sudah disusun dengan struktur yang benar. Hal ini dapat dilihat dari luaran (output) yang sesuai dengan kode program dan tidak terjadi eror lagi | | |
| **[4] Kesimpulan** | | |
| 1. Pada program itu saya menggunakan tipe data ‘long’ untuk variabel ‘a’ dan ‘b’ karena nilai yang diberikan lebih besar dari batas maksimum yang bias ditampung tipe data ‘int’ dan ‘byte’ 2. Perbaikan program dengan menghapus nilai dari variabel ‘e’ menjadi 1 karakter tunggal saja karena ‘char’ hanya dapat menampung satu karakter. Struktur Java mengharuskan penggunaan ‘char’ untuk karakter yang memiliki nilai-nilai karakter tunggal | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Michelia Erza Annadhira**  **G1F024035** | **Konversi Tipe Data** | **26 Agustus 2024** |
| **[1] Identifikasi Masalah:** | | |
| 1. Uraikan permasalahan dan variable   Diketahui dari soal :  Salin dan tempel potongan kode ini ke Eclipse atau Jdoodle  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:** 4.1. Rekomendasikan konversi tipe data pada Latihan 2 ke bentuk tipe data lain yang kompatibel   1. 5 dari ‘byte’ dapat dikonversikan ke ‘int’, ‘long’, ‘float’ dan ‘double’ 2. ‘L’ dari ‘char’ dapat dikonversikan ke ‘int’, ‘long’, ‘float’ dan ‘double’ 3. “mobil” dari ‘String’ dapat dikonversikan ke ‘int’, ‘long’, ‘float’ dan ‘double’ dengan mengubah “mobil” menjadi “5” 4. 5.0 dari ‘double’ dapat dikonversikan ke ‘int’, ‘long’ dan ‘float’ 5. 5.0f dari ‘float’ dapat dikonversikan ke ‘int’, ‘long’ dan ‘double’ 6. -5 dari ‘int’ dapat dikonversikan ke ‘byte’, ‘long’, ‘float’ dan ‘double’   4.2. Simpulkan alasan jenis konversi tipe data tersebut!   1. Perluasan rentang konversi dari tipe yang lebih kecil (misalnya. ‘byte’) ke tipe data yang lebih besar (misalnya, ‘int’, ‘long’, ‘float’, ‘double’) dilakukan untuk memperluas rentang nilai yang dapat disimpan tanpa kehilangan informasi 2. Penghilangan desimal konversi dari tipe data floating-point (‘double’, ‘float’) ke tipe data integer (‘int’, ‘long’) menyebabkan hilangnya bagian desimal, hanya menyimpan bilangan bulat. Ini berguna ketika hanya bilangan bulat yang diperlukan 3. Parsing String ke numerik konversi dari ‘String’ ke tipe numerik memerlukan parsing karena mengungkapkan representasi Unicode dari karakter tersebut, yang merupakan kode numerik yang mendasari karakter. 4. Rincikan sumber informasi yang relevan (buku / webpage)   [Video Bagian 2 – Tipe Data dan Konversi Java](https://youtu.be/at27jZTFWe8)    <https://youtu.be/at27jZTFWe8> | | |
| **[2] Analisis dan Argumentasi** | | |
| 1. Saya mengusulkan permasalahan ini dapat diatasi dengan cara mengonversi setiap tipe data ke tipe yang kompatibel, yang memungkinkan nilai tersebut tetap digunakan tanpa kehilangan informasi atau presisi. 2. Alasan solusi ini karena menggunakan tipe data yang kompatibel memungkinkan operasi aritmatika dan manipulasi data dilakukan tanpa kesalahan. Java mendukung konversi antara tipe data numerik yang lebih kecil ke yang lebih besar, dan dari ‘char’ ke tipe numerik, karena ukuran dan representasi data memungkinkan konversi tanpa masalah. 3. Perbaikan kode program dengan cara menambahkan casting eksplisit atau metode parsing yang sesuai saat mengonversi antara tipe data. Contoh :  * Untuk mengonversi ‘char’ ke ‘int’, dapat menggunakan ‘(int) ‘L’;’ * Untuk mengonversi ‘String’ ke ‘int’, dapat menggunakan ‘Integer.parseInt(“5”);’ | | |
| **[3] Penyusunan Algoritma dan Kode Program** | | |
| * Algoritma  1. Mulai 2. Konversi tipe data ‘byte’ ke ‘int’, ‘long’, ‘float’ dan ‘double’ 3. Konversi tipe data ‘char’ ke ‘int’, ‘long’, ‘float’ dan ‘double’ 4. Konversi tipe data ‘String’ ke ‘int’, ‘long’, ‘float’ dan ‘double’ dengan mengubah “mobil” menjadi “5” 5. Konversi tipe data ‘double’ ke ‘int’, ‘long’ dan ‘float’ 6. Konversi tipe data ‘float’ ke ‘int’, ‘long’ dan ‘double’ 7. Konversi tipe data ‘int’ ke ‘byte’, ‘long’, ‘float’ dan ‘double’ 8. Jalankan kode untuk memeriksa apakah program bias dijalankan dengan baik dan tidak ada kesalahan 9. Simpan kode program yang telah diperbaiki dan catat perubahan yang telah dilakukan   Screenshot (14) Screenshot (15)  Screenshot (16)  Luaran (output) :  Screenshot (12) Screenshot (13)  Kode program sudah disusun dengan struktur yang benar. Hal ini dapat dilihat dari luaran (output) yang sesuai dengan kode program dan tidak terjadi eror lagi | | |
| **[4] Kesimpulan** | | |
| * Pada program itu saya menggunakan konversi eksplisit (casting) karena tipe data yang lebih besar atau lebih presisi seperti ‘double’ perlu diubah menjadi tipe data yang lebih kecil seperti ‘int’ atau ‘float’, yang memerlukan casting eksplisit untuk memastikan tidak ada kesalahan tipe data. * Perbaikan program dengan menambahkan casting eksplisit karena struktur Java mengharuskan konversi eksplisit ketika ada potensi kehilangan informasi atau ketika mengubah dati tipe data yang lebih besar ke yang lebih kecil. Ini memastikan bahwa pengembang secara sadar memahami risiko yang terkait dengan konversi tersebut. | | |
|  | | |

**Refleksi**

(Tuliskan singkat tentang pengalaman belajar, pemaknaan pengetahuan yang baru, tantangan yang dihadapi pada minggu tersebut. Ringkasan singkat dari semua soal, bukan per soal)

Selama minggu ini, saya mempelajari materi dasar tentang deklarasi dalam Java, variabel, tipe data, dan konversi tipe data. Pemahaman tentang konsep-konsep ini sangat penting untuk membangun pondasi yang kuat dalam pemprograman Java.

Saya cukup memahami bagaimana mendeklarasikan variabel tipe data yang sesuai, serta pentingnya memilih tipe data yang tepat untuk efisien dan akurasi program. Saya juga belajar tentang berbagai tipe data di Java seperti ‘int’, ‘double’, ‘char’ dan ‘String’, serta bagaimana masing-masing digunakan berdasarkan kebutuhan spesifik data.

Tantangan yang dihadapi adalah memahami kapan dan bagaimana konversi tipe data tanpa menyebabkan kehilangan informasi atau hasil yang tidak diinginkan.

Ringkasan singkat dari semua soal, deklarasi, tentang bagaimana variabel dideklarasikan dan digunakan di Java. Variabel dan tipe data, memahami tipe data yang berbeda dan bagaimana mereka mempengaruhi cara variabel menyimpan dan memproses informasi. Konversi tipe data, mengeskplorasi metode konversi antara tipe data dan pentingnya casting untuk memastikan program berfungsi dengan benar.