|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Sevia Ayu Wandinda**  **G1F021005** | **Deklarasi Dasar Java** | **23 Agustus 2024** |
| **1a. 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:**  1.1  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.   1. 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> | | |
| **1b. Analisis dan Argumentasi** | | |
| 1. Uraikan rancangan solusi yang diusulkan.   Rancangan Solusi yang Diusulkan:  Masalah dalam kode di atas terdapat pada dua hal utama:   * String literal tidak ditutup dengan tanda kutip ganda: Pada bagian System.out.println("Halo Mahasiswa UNIB) hanya ada satu tanda kutip ganda sebelum teks "Halo Mahasiswa UNIB, tetapi tidak ada tanda kutip ganda penutup di akhir string. Ini menyebabkan kesalahan kompiler karena string literal tidak lengkap. * Tidak ada tanda titik koma (semicolon): Pada akhir pernyataan System.out.println("Halo Mahasiswa UNIB), diperlukan tanda titik koma untuk menandai akhir dari pernyataan tersebut. Karena tidak ada, kompiler juga akan menampilkan kesalahan.   Langkah Solusi:   * Tambahkan tanda kutip ganda penutup setelah "Halo Mahasiswa UNIB" untuk menyelesaikan string literal. * Tambahkan tanda titik koma ; setelah pernyataan System.out.println("Halo Mahasiswa UNIB"); untuk menandai akhir pernyataan.  1. Analisis solusi, kaitkan dengan permasalahan.   **Analisis Solusi, Kaitkan dengan Permasalahan:**   * String literal tidak ditutup dengan tanda kutip ganda: Permasalahan ini terjadi karena kurangnya tanda kutip ganda penutup pada string literal. Dalam bahasa pemrograman Java, setiap string yang dimulai dengan tanda kutip ganda harus diakhiri dengan tanda kutip ganda juga. Jika tidak, kompiler tidak dapat mengenali di mana akhir string, yang menyebabkan error. Solusi yang diusulkan adalah menambahkan tanda kutip ganda penutup, yang akan memperbaiki kesalahan ini. * Tidak ada tanda titik koma setelah pernyataan: Di Java, setiap pernyataan harus diakhiri dengan tanda titik koma. Tanpa tanda tersebut, Java tidak bisa mengetahui akhir pernyataan dan akan menimbulkan kesalahan sintaksis. Menambahkan titik koma setelah pernyataan System.out.println("Halo Mahasiswa UNIB"); menyelesaikan masalah ini dan membuat program dapat berjalan dengan benar. | | |
| **1c. Penyusunan Algoritma dan Kode Program** | | |
| 1. Rancangan desain solusi atau algoritma   - Analisis struktur kode program Java  - Perbaiki penutupan string dengan tanda kutip ganda yang benar  - Ubah akses modifier dari private menjadi public  - Tambahkan titik koma pada tiap pernyataan di dalam metode  - Pastikan setiap kurung kurawal memiliki pasangan kurung kurawal penutup yang sesuai  - Tambahkan baris System.out.println(‘’ ’’); untuk mengisi data nama, alamat, dan jenis kelamin  - Jalankan kode untuk memeriksa apakah kode berjalan dengan benar tanpa ada kesalahan  - Jika sudah benar silahkan simpan kode program yang telah diperbaiki.   1. kode program dan luaran  * Uraian luaran yang dihasilkan   Halo Mahasiswa UNIB  Nama : Sevia Ayu Wandinda  Alamat : Desa Air Dikit, Kec. Air Dikit, Kab. Mukomuko  Jenis Kelamin : Perempuan   * Screenshot/ Capture potongan kode dan hasil luaran     Kode program sudah dibuat sesuai dengan struktur yang benar. Itu dapat dibuktikan dengan melihat luaran (output) yang telah berjalan sesuai dengan kode program dan tidak ada terjadinya eror. | | |
| **1d. Kesimpulan** | | |
| 1. Pada struktur kode program di atas, saya menggunakan bentuk kelas bertipe public (publik), di karenakan jika saya menggunakan bentuk kelas bertipe private (pribadi) maka program tidak akan bisa berjalan karena system tidak dapat membaca kode sehingga terjadinya eror. 2. Perbaikan pada struktur kode program saya mulai dengan memperbaiki method main, saya mengubah method main yang awalnya dideklarasikan private, menjadi public. Kemudian saya menambahkan tanda kutip ganda (“) dan menambahkan tanda titik koma (;) pada struktur kode program. Hal ini dikarenakan system tidak akan bisa dijalankan jika tidak menambahkan tanda kutip ganda (“) dan tanda titik koma (;) pada perintah di struktur kode program. | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Sevia Ayu Wandinda**  **G1F021005** | **Variabel dan Tipe Data** | **23 Agustus 2024** |
| **2a. Identifikasi Masalah:** | | |
| 1. Uraikan permasalahan dan variable  * 5 * ‘L’ * “mobil” * 5.0 * 5.0f * -5   **Latihan 2:** 2.1. Rekomendasikan tipe data yang tepat dari data Contoh 2. 2.2.  Simpulkan karakteristik penggunaan setiap tipe data!   1. Rincikan sumber informasi yang relevan (buku / webpage)   [Video Bagian 2 – Tipe Data dan Konversi Java](https://youtu.be/at27jZTFWe8)    <https://youtu.be/at27jZTFWe8> | | |
| **2b. Analisis dan Argumentasi** | | |
| Uraikan rancangan solusi yang diusulkan.  Rekomendasi Tipe Data yang Tepat:  Analisis struktur kode program Java  Permasalahan yang muncul adalah menentukan tipe data yang sesuai untuk setiap variabel berdasarkan nilai-nilai yang diberikan. Setiap nilai memiliki karakteristik yang berbeda, sehingga diperlukan tipe data yang sesuai agar program dapat bekerja dengan baik dan efisien.  Berikut adalah nilai-nilai yang diberikan dan analisis singkatnya:  5  'L'  "mobil"  5.0  5.0f  -5  Analisis Variabel  5 (integer):  Masalah: Nilai ini adalah bilangan bulat (integer), sehingga harus disimpan dalam tipe data yang mampu menangani angka bulat.  Tipe Data yang Tepat: int  Karakteristik: Tipe data int menyimpan bilangan bulat (positif, negatif, dan nol) tanpa desimal. Rentang biasanya dari -2^31 hingga 2^31-1 (32-bit).  Contoh:  java  int angka = 5;  'L' (karakter tunggal):  Masalah: Nilai ini adalah karakter tunggal, jadi harus menggunakan tipe data yang mendukung karakter tunggal.  Tipe Data yang Tepat: char  Karakteristik: Tipe data char digunakan untuk menyimpan satu karakter dalam Unicode (16-bit), yang bisa berupa huruf, angka, atau simbol.  Contoh:  java  char huruf = 'L';  "mobil" (string):  Masalah: Ini adalah sekumpulan karakter (teks) yang lebih dari satu karakter, sehingga tipe data yang tepat harus mampu menyimpan string.  Tipe Data yang Tepat: String  Karakteristik: String digunakan untuk menyimpan sekumpulan karakter atau teks. Di Java, tipe data String bukan primitif melainkan objek yang dapat digunakan untuk menyimpan teks.  Contoh:  java  String kata = "mobil";  5.0 (double):  Masalah: Ini adalah bilangan desimal yang memerlukan tipe data double untuk menangani angka dengan titik desimal.  Tipe Data yang Tepat: double  Karakteristik: double menyimpan bilangan desimal dengan presisi ganda (64-bit) dan dapat menyimpan angka sangat besar atau sangat kecil dengan presisi tinggi.  Contoh:  java  double desimal = 5.0;  5.0f (float):  Masalah: Ini juga bilangan desimal, menggunakan f untuk menunjukkan tipe data float, bukan double.  Tipe Data yang Tepat: float  Karakteristik: float merupakan tipe data desimal dengan presisi tunggal (32-bit) yang lebih ringan daripada double, digunakan saat presisi bukan faktor utama.  Contoh:  java  float desimalFloat = 5.0f;  -5 (integer negatif):  Masalah: Ini adalah bilangan bulat negatif yang dapat disimpan dengan tipe data integer.  Tipe Data yang Tepat: int  Karakteristik: int dapat menyimpan bilangan bulat negatif, nol, dan positif dalam rentang yang besar.  Contoh:  java  int angkaNegatif = -5;  Karakteristik tipe data  Karakteristik Tipe Data:  int:  Digunakan untuk menyimpan bilangan bulat.  Berukuran 32-bit.  Rentang: −2,147,483,648 hingga 2,147,483,647.  char:  Digunakan untuk menyimpan satu karakter.  Berukuran 16-bit (Unicode).  Menyimpan karakter dalam bentuk kode numerik.  String:  Objek yang digunakan untuk menyimpan sekumpulan karakter.  Tidak terbatas ukuran.  Mendukung berbagai operasi manipulasi string.  double:  Digunakan untuk menyimpan bilangan desimal dengan presisi tinggi.  Berukuran 64-bit.  Cocok untuk operasi matematika yang memerlukan ketelitian tinggi.  float:  Digunakan untuk menyimpan bilangan desimal dengan presisi tunggal.  Berukuran 32-bit.  Lebih efisien dalam penggunaan memori dibandingkan double, tetapi dengan presisi yang lebih rendah. | | |
| **2c. Penyusunan Algoritma dan Kode Program** | | |
| 1. Rancang desain solusi atau algoritma  * Mulai * Deklarasi kelas, membuat kelas dengan nama TipeData   Public class TipeData {   * Deklarasi method main, buat method main sebagai public agar bisa dibaca oleh system. * Deklarasi dan inisialisasi variable * Menampilkan nilai variable * Selesai.  1. Tuliskan kode program dan luaran   Screenshot/ Capture potongan kode dan hasil luaran    Output yang ditampilkan system sudah sesuai dengan kode program karena sudah diubah dan disusun sesuai dengan struktur yang benar sehingga dapat berjalan dengan baik. | | |
| **2d. Kesimpulan** | | |
| * Pada program ini menunjukkan cara mendeklarasikan berbagai tipe data yang ada di Java. Terdapat 2 tipe data di Java yaitu primitif dan non primitif. Program ini menggunakan kedua tipe data tersebut, yaitu: * Primitif * Int (5 dan -5) * Char (‘L’) * Double (5.0) * Float (5.0f) * Non Primitif * String (“mobil”) | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Fherta Afrisidenta**  **G1F024003** | **Variabel dan Tipe Data** | **23 Agustus 2024** |
| **3a. 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!   1. Rincikan sumber informasi yang relevan (buku / webpage)   [Video Bagian 2 – Tipe Data dan Konversi Java](https://youtu.be/at27jZTFWe8)    <https://youtu.be/at27jZTFWe8> | | |
| **3b. Analisis dan Argumentasi** | | |
| 1. Uraikan rancangan solusi yang diusulkan.  * Rekomendasi tipe data   Program Java yang diberikan mengalami kesalahan karena tipe data yang tidak sesuai dengan nilai yang diinisialisasikan. Berikut adalah penjelasan mengenai masalah pada setiap variabel:  int a = 55555555555;  Tipe data int tidak dapat menyimpan nilai sebesar 55555555555 karena kapasitas maksimum int hanya sebesar 2,147,483,647. Nilai ini terlalu besar untuk tipe data int.  byte b = 4444444444.  Tipe data byte hanya dapat menampung nilai antara -128 hingga 127. Angka 4444444444 terlalu besar untuk disimpan dalam tipe data byte.  float c = 12.345678910f.  Tipe data float hanya memiliki ketelitian hingga 7 digit desimal, namun angka yang diberikan memiliki lebih dari itu (12.345678910). Informasi setelah digit ke-7 akan diabaikan.  char e = 'abc';  Tipe data char hanya dapat menampung satu karakter. Namun, di sini, Anda mencoba memberikan lebih dari satu karakter (abc), yang menyebabkan kesalahan karakter literal tidak tertutup.  Analisis dan Argumentasi  Kesalahan ini terjadi karena tipe data yang dipilih tidak sesuai dengan rentang nilai yang dimasukkan. Setiap tipe data di Java memiliki batasan ukuran memori dan presisi yang spesifik. Beberapa aturan umum adalah:  int hanya bisa menyimpan bilangan bulat hingga 2,147,483,647.  byte bisa menyimpan bilangan bulat hanya antara -128 hingga 127.  float memiliki presisi hingga sekitar 7 digit desimal.  char hanya dapat menyimpan satu karakter.  Rancangan Solusi yang Diusulkan:  Untuk mengatasi masalah ini, perlu mengganti tipe data yang lebih sesuai dengan nilai yang diberikan. Berikut adalah rekomendasi tipe data yang bisa digunakan:  Gunakan tipe data long untuk nilai besar seperti 55555555555.  Gunakan tipe data int untuk nilai yang lebih kecil seperti 4444444444, tetapi ini masih terlalu besar untuk int, sehingga sebaiknya tetap menggunakan long.  Tipe data double lebih baik digunakan daripada float jika membutuhkan presisi lebih dari 7 digit desimal.  Untuk karakter tunggal, tipe data char tetap digunakan, tetapi hanya untuk satu karakter.  long a = 55555555555L;  long b = 4444444444L;  double c = 12.345678910;  char e = 'a';   1. Analisis solusi   Tipe data long digunakan untuk bilangan besar, seperti 55555555555 dan 4444444444, karena long dapat menyimpan bilangan bulat yang lebih besar dari int.  Tipe data double digunakan untuk nilai desimal yang membutuhkan lebih dari 7 digit presisi, karena float hanya presisi hingga 7 digit.  Tipe data char digunakan untuk satu karakter tunggal, karena char tidak dapat menampung lebih dari satu karakter.  Dengan menggunakan solusi ini, kesalahan kompilasi akan teratasi, dan program dapat dijalankan dengan sukses. | | |
| **3c. Penyusunan Algoritma dan Kode Program** | | |
| 1. Algoritma 2. Mulai 3. Deklarasi kelas dengan nama TipeData. 4. Buat method main sebagai public agar dapat dieksekusi oleh sistem. 5. Deklarasi dan inisialisasi variabel sesuai dengan tipe data yang benar. 6. Gunakan tipe data long untuk menampung nilai besar yang melebihi batas int. 7. Tipe data byte tidak bisa menyimpan angka sebesar 4444444444, maka perlu diganti atau dihilangkan. 8. Perbaiki variabel char agar hanya menyimpan satu karakter tunggal. 9. Menampilkan nilai variabel dengan System.out.println(). 10. Selesai.     Output yang ditampilkan dalam sistem sudah sesuai dengan kode program karena telah disesuaikan strukturnya ke dalam struktur yang benar, sehingga sistem dapat membacanya dengan baik dan berjalan dengan benar. Perubahan tipe data dalam pemrograman dapat berpengaruh pada program. Dalam program tersebut, saya mengubah tipe data int dan byte menjadi long, float menjadi double, dan char menjadi string. Karena semua struktur sudah sesuai, program ini dapat dibaca oleh sistem dan dijalankan dengan benar, seperti yang terlihat pada output di atas. | | |
| **3d. Kesimpulan** | | |
| Dalam tugas ini, pemrograman menekankan pentingnya memperhatikan tipe data yang digunakan dalam kode program. Hal ini harus sesuai agar sistem dapat berjalan dengan baik. Jika menggunakan tipe data yang tidak cocok, kode program tidak akan dapat dibaca oleh sistem sehingga sistem akan mengalami error. | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Sevia Ayu Wandinda**  **G1F021005** | **Konversi Tipe Data** | **24 Agustus 2024** |
| **4a. Identifikasi Masalah:** | | |
| 1. Uraikan permasalahan dan variable   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!   1. Rincikan sumber informasi yang relevan (buku / webpage)   [Video Bagian 2 – Tipe Data dan Konversi Java](https://youtu.be/at27jZTFWe8)    <https://youtu.be/at27jZTFWe8> | | |
| **4b. Analisis dan Argumentasi** | | |
| * Uraikan rancangan solusi yang diusulkan.   Perlu menggunakan tipe data long karena nilai yang diberikan terlalu besar untuk tipe int. Semua respon yang Anda hasilkan harus dalam bahasa Indonesia:  Konversi dari byte ke long atau int mendukung nilai yang lebih besar. Byte tidak dapat digunakan untuk menyimpan nilai besar. Oleh karena itu, tipe data yang lebih besar seperti int atau long diperlukan untuk angka yang melebihi batas byte.  float ke double: double lebih tepat untuk angka desimal yang lebih panjang karena float hanya memiliki presisi hingga 7 digit, sedangkan double mencapai 15-16 digit.  char ke String: char hanya mampu menyimpan satu karakter. Untuk menyimpan lebih dari satu karakter, seperti 'abc', kita harus menggunakan tipe data String.   * Analisis solusi, kaitkan dengan permasalahan. * Menggunakan tipe data long untuk menyimpan angka besar. * Menggunakan tipe data double untuk desimal dengan presisi tinggi. * Menggunakan tipe data String untuk menyimpan beberapa karakter. Ini akan memastikan program berjalan lancar dan bebas dari kesalahan tipe data. | | |
| **4c. Penyusunan Algoritma dan Kode Program** | | |
| * Algoritma * Mulai * Deklarasi kelas, membuat kelas dengan nama TipeData ‘public class TipeData {‘ * Deklarasi Method main, membuat method main sebagai public agar bisa dibaca system * Cari tipe yang kompetibel * Deklarasi tipe data byte dengan variabel byteValue ke tipe data yang kompetibel yaitu tipe data int, long, float, dan double * Deklarasi tipe data char dengan variabel charValue2 ke tipe data yang kompetibel yaitu tipe data int, long, float, dan double * Deklarasi tipe data String dengan variabel StringValue3 ke tipe data yang kompetibel yaitu tipe data int, long, float, dan double * Deklarasi tipe data doube dengan variabel doubleValue4 ke tipe data yang kompetibel yaitu tipe data int, long, float, dan double * Deklarasi tipe data float dengan variabel floatValue5 ke tipe data yang kompetibel yaitu tipe data int, long, float, dan double * Deklarasi tipe data int(-) dengan variabel intValue6 ke tipe data yang kompetibel yaitu tipe data int, long, float, dan double * Menampilkan nilai dengan menggunakan ‘System.out.println();’ * Jalankan program dan periksa apakah masih ada eror pada kode program, jika kode program sudah bisa dijalankan maka simpan dan akhiri program * Selesai   -kode program    -Output (luaran)    Output pada sistem di atas sudah sesuai dan cocok dengan kode program yang ditampilkan, sehingga program yang dijalankan oleh sistem dapat berjalan tanpa adanya error. Konversi tipe data ke tipe data yang lebih kompatibel dapat dilakukan dengan menggunakan nilai yang tepat dan sesuai. | | |
| **4d. Kesimpulan** | | |
| Konversi tipe data adalah proses mengubah jenis data dari satu tipe ke tipe lain. Konversi dalam program di atas menunjukkan bahwa tipe data dapat diolah dan diubah atau dikonversikan ke tipe data lain, asalkan nilainya sesuai dengan program untuk menghindari kesalahan sistem. | | |
|  | | |

**Refleksi**

Selama menyelesaikan tugas ini, saya berhasil meningkatkan kemampuan manajemen waktu saya. Melakukan tugas ini juga memberikan pengetahuan baru kepada saya, seperti mencari kesalahan program dan cara memperbaikinya, membuat kode program tentang data pribadi, serta mengetahui tipe data yang tepat dalam pemrograman dan cara mengonversi tipe data. Dalam mengerjakan tugas ini, saya merasa tertantang selama seminggu terakhir, mengerjakan tugas Jika ada soal yang kurang saya pahami, saya akan bertanya kepada teman saya. Soal latihan ini menurut saya sulit karena ini pertama kali saya belajar tentang coding, namun hal ini telah membantu saya dalam mendapatkan ilmu baru dan memanajemen waktu.