

Latihan 1

Nama & NPM	Topik:	Tanggal:
Dini Ramadona G1F024050	Tipe Data	26 agustus 2024

[No.1] Identifikasi Masalah

1) Permasalahan

- 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.

Contoh 1: 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")  
    }  
}
```

Pada soal diatas masih ada kode-kode yang kurang dan tidak tepat kita akan melengkapi tipe data yang kurang dan belum tepat

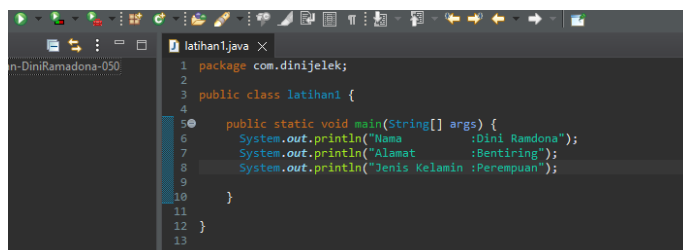
[No.1] Analisis dan Argumentasi

- 1) Saya mengusulkan permasalahan ini dapat diatasi dengan cara **String Literal** ubah "Halo Mahasiswa UNIB)" menjadi "Halo Mahasiswa UNIB)" untuk memastikan string literal terbuka dengan benar menggunakan tanda kutip ganda (").
Sintaks Error kurangnya Semicolon tambahkan semicolon (;) di akhir method main.
tanda Kurung Kurawal yang Tidak Sesuai pastikan tanda kurung kurawal {} sesuai dengan sintaks Java.
- 2) Alasan solusi ini karena dengan perbaikan ini, program akan berjalan tanpa kesalahan dan menampilkan teks yang diinginkan serta kita akan lebih mengenal kode-kode dijava.
- 3) Perbaiki kode program dengan cara ubah tanda kutip tunggal ke tanda kutip ganda ubah `Halo Mahasiswa UNIB)` menjadi `Halo Mahasiswa UNIB")` tambahkan tanda kurung kurawal `{}` yang sesuai serta tambahkan semicolon (;) di akhir method main .

[No.1] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - (a) Membuka aplikasi eclipse
 - (b) Salin kode yang ada di E-learning lalu paste ke eclipse
 - (c) Mencari Penyebab Kesalahan kode
 - (d) Mencari tutorial di youtube Rumah Raflesia
 - (e) Mencari tipe data yang cocok lalu perbaiki kesalahan
 - (f) Menambahkan tanda kutip ganda (")
 - (g) Menambahkan tanda kurung `{}`
 - (h) Menambahkan semicolon (;)
 - (i) Periksa hasil output untuk memastikan bahwa data ditampilkan dengan benar sesuai dengan tipe data yang digunakan
 - (j) Selesai

2) Kode program dan luaran



```
1 package com.dinijelek;  
2  
3 public class latihani {  
4  
5     public static void main(String[] args) {  
6         System.out.println("Nama :Dini Ramdona");  
7         System.out.println("Alamat :Bentiring");  
8         System.out.println("Jenis Kelamin :Perempuan");  
9     }  
10 }  
11  
12  
13
```

- a) dari soal yang telah ada kita diminta untuk mengganti "Halo Mahasiswa UNIB" menjadi nama lengkap, alamat, dan jenis kelamin sebelumnya menggunakan tanda kutip tunggal (') yang tidak sesuai dengan sintaks Java. Perubahan ini memastikan bahwa string literal terbuka dengan benar menggunakan tanda kutip ganda (") teks nama lengkap `System.out.println("Nama: Dini Ramadona");` menampilkan nama lengkap anda baris ini menambahkan informasi nama lengkap Anda ke output program. `System.out.println("Alamat: Bentiring");` dan `System.out.println("Jenis Kelamin: Perempuan");` menampilkan data alamat dan jenis kelamin baris-baris ini menambahkan informasi alamat dan jenis kelamin ke output program dengan begitu kita akan memahami kode-kode yang ada di java serta mengetahui kesalahannya.
- b) Luaran sudah sesuai dengan program yang disusun. Program telah menampilkan teks yang diinginkan dengan benar, teks yang ditampilkan termasuk nama lengkap, alamat, dan jenis kelamin, tipe data yang ditampilkan telah sesuai dengan kebutuhan dan permintaan, data program telah menampilkan data alamat dan jenis kelamin secara terpisah data yang ditampilkan sudah sesuai dengan permintaan dan kebutuhan dengan demikian, program telah berjalan dengan benar dan menampilkan teks yang diinginkan.

[No.1] Kesimpulan

- 1) Analisa
 - a) Permasalahan

Dalam potongan kode yang diberikan, terdapat beberapa kesalahan sintaksis yang menyebabkan program tidak dapat berjalan.

Kesalahan tersebut meliputi:

 - Tanda kutip ganda pada string literal tidak ditutup dengan benar.
 - Tidak adanya semicolon (;) di akhir method main.
 - Tanda kurung kurawal ({}) tidak sesuai dengan sintaks Java.
 - b) Algoritma

Untuk memperbaiki kesalahan-kesalahan tersebut, langkah-langkah yang diambil adalah:

 - Memastikan semua string literal ditutup dengan tanda kutip ganda yang benar.
 - Menambahkan semicolon (;) di akhir setiap pernyataan yang memerlukannya.
 - Memastikan semua blok kode diakhiri dengan tanda kurung kurawal yang sesuai.
 - c) Dasar Alasan Pengambilan Keputusan

Keputusan untuk memperbaiki kesalahan sintaksis didasarkan pada pemahaman bahwa setiap bahasa pemrograman memiliki aturan dan sintaks yang harus diikuti agar kode dapat dijalankan dengan benar. Memperbaiki kesalahan ini penting untuk memastikan bahwa program dapat dikompilasi dan dieksekusi tanpa error, dalam konteks pemrograman, penting untuk menggunakan tipe data yang sesuai untuk menyimpan informasi., dalam kode yang diperbaiki, penggunaan string untuk menampilkan informasi seperti nama, alamat, dan jenis kelamin adalah tepat, karena string adalah tipe data yang dirancang untuk menyimpan teks, Memperbaiki kesalahan sintaksis dan memastikan bahwa kode ditulis dengan jelas dan terstruktur memastikan bahwa program berjalan dengan baik dan menampilkan informasi yang diinginkan.

Refleksi

Pengalaman belajar pada minggu ini telah sangat berharga karena membantu saya memahami lebih dalam tentang sintaks Java dan penggunaan tipe data yang tepat. Dengan menghadapi tantangan-tantangan yang ada, saya telah meningkatkan kemampuan saya dalam mengidentifikasi dan memperbaiki kesalahan sintaks serta mengadaptasi pengetahuan baru ke dalam konteks yang berbeda-beda. Insyaallah saya siap untuk menghadapi tantangan-tantangan lainnya dalam bidang pemrograman Java.

Latihan 2

Nama & NPM	Topik:	Tanggal:
Dini Ramadona G1F024050	Tipe Data	27 Agustus 2024
[No.2] Identifikasi Masalah:		
1) Permasalahan 2.1. Rekomendasikan tipe data yang tepat dari data Contoh 2. 2.2. Simpulkan karakteristik penggunaan setiap tipe data! Java memiliki 8 tipe data primitif dan non-primitif (baca slide atau video materi). Contoh 2: Apabila diketahui data berikut 1. 5 2. 'L' 3. "mobil" 4. 5.0 5. 5.0f 6. -5 Pada soal kali ini kita akan mencari kode-kode yang menghasilkan data diatas, java memiliki 8 tipe data primitif dan non-primitif dengan memahami tipe data primitif dan non-primitif di Java, kita dapat memilih tipe data yang tepat untuk setiap jenis informasi yang ingin kita simpan dalam program.		
[No.2] Analisis dan Argumentasi		
1) Uraikan rancangan solusi yang diusulkan. Rancangan solusi untuk permasalahan yang saya hadapi ada beberapa langkah sebagai berikut: Langkah 1: Identifikasi Tipe Data yang Diperlukan Setiap data yang diberikan dalam contoh dan tentukan tipe data yang paling sesuai berdasarkan karakteristik data tersebut, pastikan untuk membedakan antara tipe data primitif (seperti int, char, float, dan double) dan tipe data non-primitif (seperti String). Langkah 2: Rekomendasi Tipe Data Untuk bilangan bulat, gunakan int atau long jika nilai tersebut besar. Untuk karakter tunggal, gunakan char. Untuk urutan karakter, gunakan String. Untuk angka desimal, gunakan float untuk presisi tunggal dan double untuk presisi ganda. Langkah 3: Implementasi dalam Kode Setelah menentukan tipe data yang tepat, implementasikan dalam kode Java dengan mendeklarasikan variabel dengan tipe data yang sesuai, tulis contoh kode yang menampilkan penggunaan tipe data yang telah direkomendasikan. Langkah 4: Uji Kode		

Jalankan kode untuk memastikan bahwa semua tipe data berfungsi dengan baik dan tidak ada kesalahan kompilasi.

2) Analisis solusi, kaitkan dengan permasalahan.

Solusi yang diusulkan berfokus pada pemilihan dan penggunaan tipe data yang tepat dalam pemrograman Java. Dengan mengikuti langkah-langkah yang telah diuraikan, kita dapat menghindari kesalahan yang sering terjadi, seperti Menggunakan tipe data yang tidak sesuai dapat menyebabkan error saat kompilasi atau runtime, seperti integer number too large atau unclosed character literal.

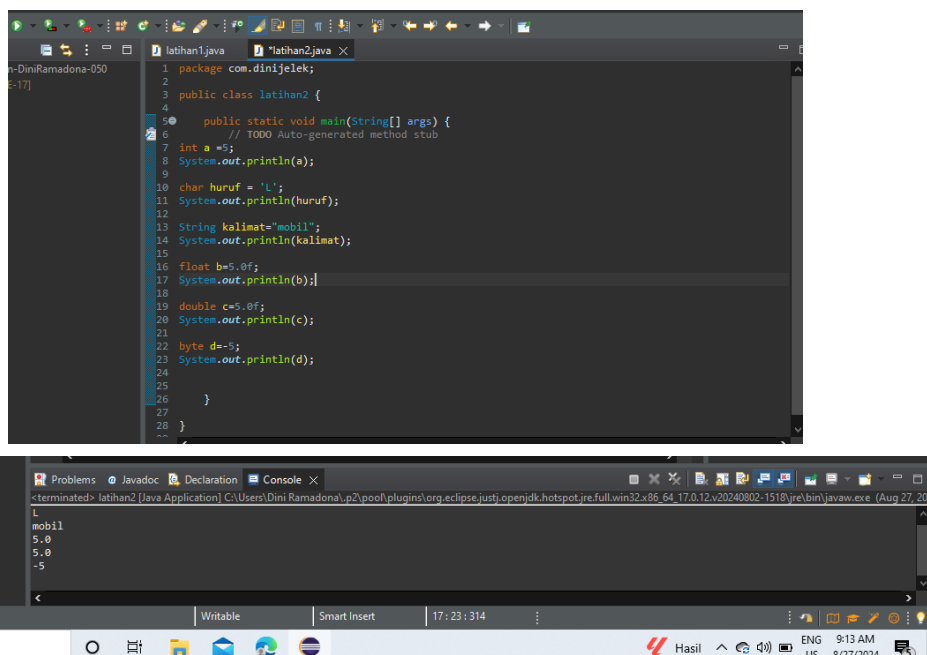
Permasalahan yang dihadapi dalam contoh sebelumnya mencakup kesalahan dalam penggunaan tipe data, seperti menggunakan int untuk angka yang terlalu besar, yang menyebabkan kesalahan kompilasi menggunakan char untuk lebih dari satu karakter, yang juga menyebabkan kesalahan menggunakan float untuk angka desimal yang memerlukan presisi lebih tinggi, yang dapat mengakibatkan kehilangan data, dengan menerapkan solusi yang diusulkan kita dapat mencegah Kesalahan

[No.2] Penyusunan Algoritma dan Kode Program

1) Algoritma

- (a) Identifikasi setiap data yang ada di E-learning yang akan digunakan dalam program.
- (b) Mencari tutorial di youtube dan website
- (c) Tentukan jenis informasi yang akan disimpan oleh setiap data (bilangan bulat, karakter tunggal, urutan karakter, atau angka desimal).
- (d) Membuka aplikasi eclipse
- (e) Mencari tipe data yang cocok
- (f) Gunakan int untuk bilangan bulat yang tidak terlalu besar
- (g) Gunakan long untuk bilangan bulat yang lebih besar
- (h) Gunakan char untuk menyimpan satu karakter
- (i) Gunakan String untuk menyimpan urutan karakter (teks)
- (j) Gunakan float untuk angka desimal dengan presisi tunggal
- (k) Gunakan double untuk angka desimal dengan presisi ganda
- (l) Periksa hasil output untuk memastikan bahwa data ditampilkan dengan benar sesuai dengan tipe data yang digunakan
- (m) Selesai

2) Tuliskan kode program dan luaran



The screenshot displays the Eclipse IDE interface. The top editor window shows a Java file named 'latihan2.java' with the following code:

```
1 package com.dinijelek;
2
3 public class latihan2 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int a = 5;
8         System.out.println(a);
9
10        char huruf = 'L';
11        System.out.println(huruf);
12
13        String kalimat = "mobil";
14        System.out.println(kalimat);
15
16        float b = 5.0f;
17        System.out.println(b);
18
19        double c = 5.0;
20        System.out.println(c);
21
22        byte d = -5;
23        System.out.println(d);
24
25    }
26
27 }
```

The bottom console window shows the output of the program:

```
L
mobil
5.0
5.0
-5
```

The status bar at the bottom indicates the file is writable, the time is 17:23:314, and the system language is English (US) on 8/27/2024.

- a) Dari soal yang ada kita disuruh untuk
Menampilkan nilai variabel a yang memiliki tipe data int, yaitu 5
Menampilkan nilai variabel b yang memiliki tipe data char, yaitu 'L'
Menampilkan nilai variabel c yang memiliki tipe data String, yaitu "mobil"
Menampilkan nilai variabel d yang memiliki tipe data float, yaitu 5.0
Menampilkan nilai variabel e yang memiliki tipe data double, yaitu 5.0
Dengan demikian, program telah berjalan dengan benar dan menampilkan teks yang diinginkan dengan menggunakan tipe data yang tepat untuk setiap jenis informasi yang disimpan.
- b) Setiap variabel dideklarasikan dengan tipe data yang sesuai untuk jenis informasi yang akan disimpan setiap deklarasi memberikan penjelasan singkat tentang tipe data yang digunakan dan nilai yang disimpan setiap pernyataan `System.out.println()` menjelaskan bahwa pernyataan tersebut digunakan untuk menampilkan nilai dari variabel yang telah dideklarasikan sebelumnya.

[No.2] Kesimpulan

- 1) Evaluasi
- a) Konsekuensi dari Skenario Pemrograman
- Menggunakan tipe data yang tepat dan mematuhi sintaks Java akan menghindari kesalahan kompilasi dan runtime.
Contoh: Menggunakan int untuk bilangan bulat, char untuk karakter tunggal, String untuk teks, float untuk angka desimal dengan presisi tunggal, dan double untuk angka desimal dengan presisi ganda.
Menggunakan tipe data yang sesuai akan mengoptimalkan penggunaan memori.
Contoh: Menggunakan float untuk angka desimal dengan presisi tunggal dapat menghemat memori dibandingkan menggunakan double.
Kemudahan Pemahaman dan Penggunaan:
Menggunakan tipe data yang tepat akan membuat kode lebih mudah dipahami dan digunakan oleh programmer lain.
Contoh: Deklarasi variabel dengan tipe data yang jelas akan membantu programmer lain memahami fungsi variabel tersebut.
Pengalaman Pengguna yang Baik:
Menggunakan tipe data yang tepat akan memberikan pengalaman pengguna yang lebih baik karena program akan berjalan dengan benar dan efisien.
Contoh: Program yang menggunakan tipe data yang tepat akan menampilkan informasi yang diinginkan dengan benar tanpa kesalahan.
- b) Evaluasi Input, Proses, dan Luaran
- Input yang diberikan dalam contoh adalah bilangan bulat (5), karakter tunggal ('L'), teks ("mobil"), angka desimal dengan presisi tunggal (5.0f), dan angka desimal dengan presisi ganda (5.0). Setiap input dideklarasikan dengan tipe data yang tepat, seperti int untuk bilangan bulat, char untuk karakter tunggal, String untuk teks, float untuk angka desimal dengan presisi tunggal, dan double untuk angka desimal dengan presisi ganda. Proses yang digunakan adalah deklarasi variabel dengan tipe data yang sesuai dan output hasilnya menggunakan `System.out.println`.

Refleksi

Pengalaman belajar pada minggu ini telah sangat berharga karena membantu saya memahami lebih dalam tentang sintaks Java dan penggunaan tipe data yang tepat. Dengan menghadapi tantangan-tantangan yang ada, saya telah meningkatkan kemampuan saya dalam mengidentifikasi dan memperbaiki kesalahan sintaks serta mengadaptasi pengetahuan baru ke dalam konteks yang berbeda-beda. Insyaallah saya siap untuk menghadapi tantangan-tantangan lainnya dalam bidang pemrograman Java.

Latihan 3

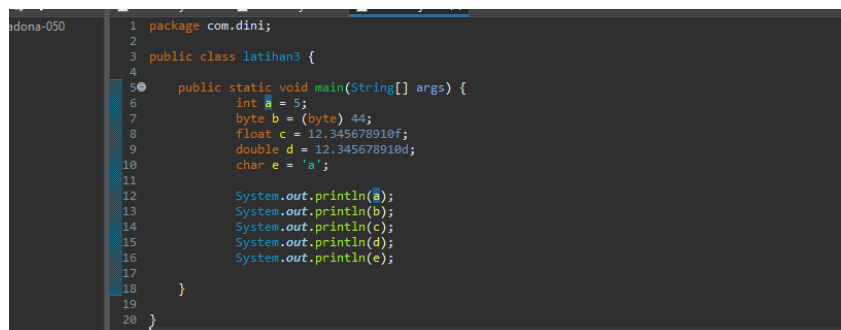
Nama & NPM	Topik:	Tanggal:
Dini Ramadona G1F024050	Tipe Data	27 Agustus 2024
[No.3] Identifikasi Masalah:		
<p>1) Permasalahan</p> <p>3.1. Evaluasi penyebab kesalahan pada Contoh 3!</p> <p>3.2. Rekomendasikan tipe data yang sesuai untuk data tersebut!</p> <p>Contoh 3: Salin dan tempel potongan kode ini ke Eclipse atau Jdoodle</p> <pre>public class TipeData { public static void main(String args[]) { int a = 5555555555; 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); } }</pre> <p>Pada soal ini dalam potongan kode yang diberikan terdapat beberapa kesalahan sintaksis yang menyebabkan program tidak dapat berjalan dengan sesuai analisis tipe data yang tepat untuk setiap jenis informasi yang disimpan agar program berjalan dengan sesuai.</p>		
[No.3] Analisis dan Argumentasi		
<p>a) Rancangan solusi saya untuk permasalahan yang dihadapi dalam potongan kode Java yang diberikan melibatkan beberapa langkah yang sistematis untuk memperbaiki kesalahan dan memastikan bahwa program berjalan dengan benar. Berikut adalah langkah-langkah rancangan solusi:</p> <p>Langkah 1: Identifikasi Tipe Data yang Tepat</p> <p>Pertama, identifikasi setiap data yang diberikan dalam contoh dan tentukan tipe data yang paling sesuai berdasarkan karakteristik data tersebut.</p> <p>5555555555 harus disimpan dalam tipe data long.</p> <p>4444444444 harus disimpan dalam tipe data int.</p> <p>12.345678910f sebaiknya diubah menjadi tipe double untuk presisi yang lebih tinggi.</p> <p>12.345678910d tetap sebagai double.</p> <p>Karakter 'abc' harus diubah menjadi karakter tunggal, misalnya 'a' untuk menggunakan tipe char.</p> <p>Langkah 2: Perbaiki Kode</p> <p>Setelah menentukan tipe data yang tepat, implementasikan dalam kode Java dengan mendeklarasikan variabel dengan tipe data yang sesuai. Berikut adalah contoh kode yang telah diperbaiki</p> <p>Langkah 3: Uji Kode</p> <p>Jalankan kode untuk memastikan bahwa semua tipe data berfungsi dengan baik dan tidak ada kesalahan kompilasi</p> <p>b) Solusi yang saya usulkan berfokus pada pemilihan dan penggunaan tipe data yang tepat dalam pemrograman Java. Dengan mengikuti langkah-langkah di atas, kita dapat menghindari kesalahan yang sering terjadi, seperti :</p> <p>Menggunakan tipe data yang tidak sesuai dapat menyebabkan error saat kompilasi atau runtime, seperti integer number too large atau unclosed character literal.</p>		

Memilih tipe data yang tepat membantu dalam pengelolaan memori yang lebih efisien, terutama dalam aplikasi yang memerlukan banyak data.

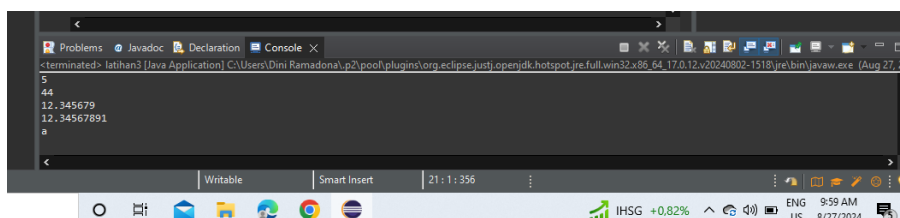
[No.3] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - a) Membuka aplikasi eclipse
 - b) Salin kode yang ada di E-learning lalu paste ke eclipse
 - c) Mencari Penyebab Kesalahan kode
 - d) Mencari tutorial di youtube Rumah Raflesia dan website lainnya
 - e) Tentukan tipe data yang tepat untuk memperbaiki kesalahan
 - f) `int a = 55555555555;` - Nilai ini terlalu besar untuk tipe data `int`.
 - g) `byte b = 44444444444;` - Nilai ini terlalu besar untuk tipe data `byte`.
 - h) `char e = 'abc';` - `char` hanya dapat menyimpan satu karakter, bukan lebih dari satu.
 - i) `float c = 12.345678910f;` - Nilai ini memiliki lebih banyak angka desimal daripada yang dapat disimpan oleh tipe data `float`
 - j) Jalankan kode untuk memastikan bahwa semua tipe data berfungsi dengan baik dan tidak ada kesalahan kompilasi
 - k) Selesai

- 2) Tuliskan kode program dan luaran



```
1 package com.dini;
2
3 public class latihan3 {
4
5     public static void main(String[] args) {
6         int a = 5;
7         byte b = (byte) 44;
8         float c = 12.345678910f;
9         double d = 12.345678910d;
10        char e = 'a';
11
12        System.out.println(a);
13        System.out.println(b);
14        System.out.println(c);
15        System.out.println(d);
16        System.out.println(e);
17    }
18 }
19
20 }
```



```
<terminated> latihan3 [Java Application] C:\Users\Dimi Ramadana\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.12.v20240802-1518\jre\bin\javaw.exe (Aug 27, 2024)
5
44
12.345679
12.34567891
a
```

[No.3] Kesimpulan

- 2) Kreasi
 - a) Dalam konteks pemrograman Java, ada beberapa pengetahuan baru yang saya dapatkan dan konsep yang relevan, menggunakan tipe data yang sesuai seperti `long` untuk nilai yang besar, `double` untuk angka desimal dengan presisi yang lebih tinggi, dan `char` untuk karakter tunggal. Ini membantu menghindari kesalahan kompilasi dan meningkatkan efisiensi memori.

Refleksi

Pengalaman belajar pada minggu ini telah sangat berharga karena membantu saya memahami lebih dalam tentang sintaks Java dan penggunaan tipe data yang tepat. Dengan menghadapi tantangan-tantangan yang ada, saya telah meningkatkan kemampuan saya dalam mengidentifikasi dan memperbaiki kesalahan sintaks serta mengadaptasi

pengetahuan baru ke dalam konteks yang berbeda-beda. Insyaallah saya siap untuk menghadapi tantangan-tantangan lainnya dalam bidang pemrograman Java

Latihan 4

Nama & NPM	Topik:	Tanggal:
Dini Ramadona G1F024050	Tipe Data	29 Agustus 2024

[No.4] Identifikasi Masalah:

- 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!

Contoh 4: 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);  
    }  
}
```

Pada soal ini kita disuruh konversi tipe data, kita akan konversi tipe data yang diatas serta melengkapi tipe data yang kurang dan yang belum tepat Dengan demikian, konversi tipe data harus dilakukan dengan hati-hati dan mempertimbangkan rentang nilai yang dapat diakomodasi oleh tipe data tujuan untuk menghindari kesalahan dan memastikan hasil yang akurat..

[No.4] Analisis dan Argumentasi

- a) Untuk menyelesaikan permasalahan dalam kode Java yang diberikan, kita perlu merancang solusi yang sistematis. Berikut adalah langkah-langkah untuk merancang solusi:

Langkah 1: Identifikasi Permasalahan

Identifikasi kesalahan yang ada dalam kode, termasuk kesalahan tipe data dan kesalahan sintaksis.

byte x; - Variabel ini tidak dideklarasikan dengan nilai awal.

int a = 270; - Nilai ini tidak melebihi batas maksimum tipe data int.

double b = 128.128; - Nilai ini tidak melebihi batas maksimum tipe data double.

Langkah 2: Perbaiki Kode

Ubah byte x; menjadi byte x = (byte) a; untuk mengkonversi nilai dari int ke byte.

Ubah int a = (int) b; menjadi int a = (int) b; untuk mengkonversi nilai dari double ke int.

Ubah byte x = (byte) b; menjadi byte x = (byte) b; untuk mengkonversi nilai dari double ke byte.

Langkah 3: Implementasi Kode

Tulis Ulang Kode dengan Tipe Data yang Tepat

b) Menggunakan byte untuk nilai yang melebihi batas Nilai 270 tidak dapat disimpan dalam tipe data byte karena melebihi batas maksimumnya. Menggunakan tipe data int untuk nilai yang tidak perlu dapat menghemat memori jika menggunakan tipe data yang lebih kecil seperti byte. Menggunakan tipe data double untuk nilai yang tidak perlu dapat menghemat memori jika menggunakan tipe data yang lebih kecil seperti int dengan menerapkan solusi yang diusulkan, kita dapat memastikan bahwa semua data dideklarasikan dengan tipe yang benar, sehingga menghindari kesalahan kompilasi mengoptimalkan penggunaan memori dan meningkatkan kinerja aplikasi dengan memilih tipe data yang sesuai berdasarkan kebutuhan spesifik data yang akan diproses.

[No.4] Penyusunan Algoritma dan Kode Program

- 1) Algoritma
 - a) Membuka aplikasi eclipse
 - b) Salin kode yang ada di E-learning lalu paste ke eclipse
 - c) Mencari tutorial di youtube Rumah Raflesia dan website lainnya
 - d) Identifikasi tipe data yang cocok untuk konversi tipe data
 - e) Mengkonversi tipe data
 - f) Mengkonversi int ke byte
 - g) Mengkonversi double ke byte
 - h) Mengkonversi double ke byte
 - i) Output hasil bahwa nilai yang dihasilkan adalah yang diharapkan
 - j) Selesai
- 2) Tuliskan kode program dan luaran

```
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);  
  
}  
}
```

```
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 x128.128 + x
```

[No.4] Kesimpulan

- 1) Analisa
 - a) **Permasalahan** yang dihadapi dalam contoh sebelumnya mencakup kesalahan dalam penggunaan tipe data, seperti:
PKode yang diberikan menghadapi beberapa masalah terkait konversi tipe data, seperti nilai int yang lebih besar dari batas maksimum byte tidak dapat dikonversi tanpa kehilangan informasi. Konversi double ke int yang mengakibatkan hilangnya bagian desimal. Potensi kesalahan pada konversi double ke byte jika nilai tidak berada dalam rentang yang valid.
Algoritma yang diusulkan mencakup langkah-langkah untuk:
Mengidentifikasi tipe data yang sesuai untuk setiap variabel.
Melakukan konversi tipe data dengan memeriksa rentang nilai sebelum konversi untuk menghindari kesalahan.
Mengoutput hasil konversi dengan jelas untuk memverifikasi bahwa nilai yang dihasilkan sesuai dengan yang diharapkan.
Kode program yang telah diperbaiki dan dilengkapi dengan komentar memberikan gambaran yang jelas tentang proses konversi tipe data yang dilakukan.

Kode tersebut juga mencakup pemeriksaan kondisi untuk memastikan bahwa konversi dilakukan hanya jika nilai berada dalam rentang yang valid, sehingga mengurangi risiko kesalahan.

- b) Mengikuti prinsip-prinsip pemrograman yang baik, seperti penggunaan tipe data yang tepat, penanganan kesalahan, dan dokumentasi yang jelas, adalah alasan utama dalam pengambilan keputusan untuk menyusun kode ini. Hal ini berkontribusi padapengembangan perangkat lunak yang lebih robust dan dapat diandalkan.

Refleksi

Pengalaman belajar pada minggu ini telah sangat berharga karena membantu saya memahami lebih dalam tentang sintaks Java dan penggunaan tipe data yang tepat. Dengan menghadapi tantangan-tantangan yang ada, saya telah meningkatkan kemampuan saya dalam mengidentifikasi dan memperbaiki kesalahan sintaks serta mengadaptasi pengetahuan baru ke dalam konteks yang berbeda-beda. Insyaallah saya siap untuk menghadapi tantangan-tantangan lainnya dalam bidang pemrograman Java