

LAPORAN PRAKTIKUM FUNDAMENTAL JAVA PROGRAMMING

*Laporan praktikum ini disusun untuk memenuhi Tugas Mata Kuliah Pemrograman
Berorientasi Objek Praktek*



Disusun oleh:

Wildan Setya Nugraha

211511032

**PROGRAM STUDI D3 TEKNIK INFORMATIKA
JURUSAN TEKNIK KOMPUTER DAN
INFORMATIKA
POLITEKNIK NEGERI BANDUNG**

2022

DAFTAR ISI

DAFTAR ISI	2
SOAL 1 TIPE DATA	1
SOAL 2 VARIABEL	3
SOAL 3 OPERATOR	5
SOAL 4 OPERATOR I	6
SOAL 5 STRING	8

SOAL 1 TIPE DATA

1. Screenshot Program

```
1 package tipeData;
2
3 import java.util.Scanner;
4
5 public class DataType {
6
7     public static void main(String[] args) {
8         Scanner inputUser = new Scanner(System.in);
9         double data = inputUser.nextDouble();
10
11         if((data >= Long.MIN_VALUE && data <= Long.MAX_VALUE)) {
12             System.out.println((long)data + " Can be fitted in: ");
13             if (data >= Byte.MIN_VALUE && data <= Byte.MAX_VALUE) {
14                 System.out.println("Byte");
15             }
16             if (data >= Short.MIN_VALUE && data <= Short.MAX_VALUE) {
17                 System.out.println("Short");
18             }
19             if (data >= Integer.MIN_VALUE && data <= Integer.MAX_VALUE) {
20                 System.out.println("Integer");
21             }
22             System.out.println("Long");
23         } else {
24             System.out.println(data + " can't be fitted anywhere");
25         }
26     }
27 }
```

Penjelasan Algoritma / Program.

Program ini menginputkan nilai atau value dengan tipe data double, karena batas maksimalnya tipe data double sangat besar, sehingga ketika menginputkan bilangan yang lebih dari maksimal/minimal dari tipe data long tidak akan error.

Program ini akan menunjukkan bilangan yang kita inputkan itu termasuk ke tipe data apa saja. Tentunya untuk menentukan hal tersebut kita perlu tahu **minimal/maksimal value** dari **setiap tipe data**.

Pertama kali, program akan mengecek apakah bilangan yang kita inputkan masih termasuk ke dalam min sampai maks tipe data long. Mengapa inputan yang kita bandingkan dimulai dari **tipe data long**, karena **tipe data byte, short, integer** itu merupakan termasuk **subset (bagian)** dari tipe data long, jadi ketika kondisi pertama terpenuhi maka otomatis bilangan yang kita inputkan itu memungkinkan akan termasuk ke tipe data byte, short, integer, dan long. Tinggal bagaimana bilangan tersebut saat di pengkondisian berikutnya. Maka dari itu, kita bandingkan terlebih dahulu dengan tipe data long, jadi ketika inputan lebih dari maks dan kurang dari min tipe data long, maka bilangan inputan itu tidak akan masuk ke dalam tipe data byte, short, integer, maupun long.

2. Screenshot Jawaban

[illegible]

3. Permasalahan yang dihadapi
 - Perbedaan syntax inputan pada Java dengan bahasa C.
4. Solusi dari permasalahan yang dihadapi

Dengan membaca dokumentasi syntax-syntax Java terutama syntax inputan. Hasil dari bacaan tersebut, dijelaskan bahwasannya untuk user menginput sendiri itu harus terlebih dahulu **mengimport scanner** dari **java.util.scanner**. Karena Scanner itu merupakan sebuah class, maka kita perlu menginstansiasi terlebih dahulu dengan cara berikut.

Scanner_variabel_name = new Scanner (parameter)

Jika menginginkan inputan maka parameter diisikan oleh **System.in** yang artinya system akan menerima inputan.
5. Nama teman yang membantu memecahkan permasalahan di persoalan ini.

SOAL 2 VARIABEL

1. Screenshot Program

Constants

```
1 package variables;
2
3 public class Constants {
4     public static void main(String[] args) {
5         final double CM_PER_INCH = 2.54;
6         double paperWidth = 8.5;
7         double paperHeight = 11;
8         System.out.println("Paper size in centimeters: " + paperWidth * CM_PER_INCH + " by " + paperHeight * CM_PER_INCH);
9     }
10 }
```

Constants2

```
1 package variables;
2
3 public class Constants2 {
4     public static final double CM_PER_INCH = 2.54;
5
6     public static void main(String[] args) {
7         double paperWidth = 8.5;
8         double paperHeight = 11;
9         System.out.println("Paper size in centimeters: " + paperWidth * CM_PER_INCH + " by " + paperHeight * CM_PER_INCH);
10    }
11 }
```

2. Screenshot Jawaban

Constants

```
Paper size in centimeters: 21.59 by 27.94
```

Constants2

```
Paper size in centimeters: 21.59 by 27.94
```

Dilihat dari output yang dikeluarkan oleh class Constants dan class Constants2 itu hasilnya tak ada yang berbeda, keduanya memiliki output yang sama. Perbedaan dari masing-masing class tersebut hanya pada penulisan syntax variabel konstanta yang pada class Constants penulisannya **di dalam public static void main** dengan **final double**. Namun, pada class Constants2 penulisannya **di luar public static void main** dengan penulisan **public static final double**.

Meskipun hasil yang dikeluarkan itu sama, tetapi itu merupakan perbedaan yang cukup signifikan. Perbedaanya adalah ketika kita mendeklarasikan variabel konstanta dengan penulisan **final tipe_data**, maka variabel konstanta tersebut hanya dapat dipanggil di dalam fungsi maupun prosedur yang memanggilnya atau yang biasa dikenal dengan **variabel lokal**. Berbeda halnya dengan kita menuliskan **static** di awalnya, maka variabel konstanta tersebut dapat digunakan oleh procedure maupun

function lain, hal itu biasa dikenal dengan sebutan **variabel global**. Jika ditambah **public** di depannya, maka variabel tersebut **dapat dipanggil di luar class** Constants.

3. Permasalahan yang Dihadapi
Tidak ada permasalahan pada contoh kasus ini.
4. Solusi dari Permasalahan yang Dihadapi
Karena tidak ada permasalahan yang dihadapi maka tidak memiliki solusi dari permasalahan.
5. Nama teman yang membantu memecahkan permasalahan di persoalan ini.
-

SOAL 3 OPERATOR

1. Screenshot Program

```
1 package operators;
2
3 public class FloatingPoint {
4
5     public static void main(String[] args) {
6         double x = 92.98;
7         int nx = (int) Math.round(x);
8
9         System.out.println("Hasil dari " + nx);
10    }
11
12 }
```

2. Screenshot Jawaban

```
Hasil dari nx: 93
```

Pada kasus tersebut nilai dari nx berupa integer atau bilangan bulat yang dibulatkan ke bilangan terdekatnya, sehingga nilai yang berawal 92.98 dibulatkan menjadi 93. Hal tersebut terjadi karena kita menggunakan method dari **class Math** yaitu **Math.round()**.

Method Math.round() ini akan mengembalikan nilai berupa bilangan integer atau bilangan bulat yang telah dibulatkan ke bilangan terdekatnya. Misalkan angka 92.3, nah dari kasus tersebut jika dibelakang koma itu 4 ke bawah maka akan dibulatkan menjadi 92, sehingga **Math.round()** tersebut akan membulatkan ke bawah. Bagi dibelakang koma itu mulai dari 5 ke atas, maka akan dibulatkan ke atas atau jika misal angka 92.6 akan dibulatkan menjadi 93. Begitulah method **Math.round()** berfungsi.

Karena nilai awalnya berupa bilangan koma dan kita perlu menjadikan bilangan tersebut menjadi bilangan bulat, maka kita perlu mencasting bilangan koma (double) tersebut ke bilangan bulat (int) dengan ditampung pada variabel bertipe data integer pula.

3. Permasalahan yang Dihadapi

Tidak ada permasalahan pada contoh kasus ini.

4. Solusi dari Permasalahan yang Dihadapi

Karena tidak ada permasalahan yang dihadapi maka tidak memiliki solusi dari permasalahan.

5. Nama teman yang membantu memecahkan permasalahan di persoalan ini.

-

SOAL 4 OPERATOR I

1. Screenshot Program

```
1 package operators;
2
3 public class ConvertDataType {
4
5     static short methodOne(long l) {
6         int i = (int) l;
7         return (short) i;
8     }
9     public static void main(String[] args) {
10         double d = 10.25;
11         float f = (float) d;
12         byte b = (byte) methodOne((long) f);
13         System.out.println("Hasil dari b: " + b);
14     }
15 }
16
17 }
```

2. Screenshot Jawaban

```
Hasil dari b: 10
```

Output dari nilai variabel b ini merupakan hasil dari berbagai casting atau convert dari tipe data sebelumnya dan dari methodOne.

- Pertama dari variabel d bertipe data double dengan value 10.250000000000000 (15 digit dibelakang koma).
- Variabel d tersebut di casting ke dalam tipe data float sehingga menjadi 10.250000 (6 digit dibelakang koma).
- Variabel f di casting kembali ke dalam tipe data long, sehingga yang tadinya merupakan bilangan desimal berubah menjadi bilangan bulat dengan dibulatkan kebawah menjadi 10.
- Masuk ke dalam function **methodOne** dengan membawa nilai dari variabel f yang telah di casting.
- Di dalam function, variabel f tersebut akan di casting kembali ke tipe data integer dan di return kan dengan tipe data short, karena nilai atau value dari variabel f yang telah di casting long itu 10 dan masih termasuk ke tipe data integer, short, dan byte. Maka nilai dari variabel b itu tetap menjadi 10, sehingga nilai atau output yang dikeluarkan adalah 10.

3. Permasalahan yang Dihadapi

Permasalahan yang dihadapi ketika memahami contoh kasus ini adalah memahami setiap nilai yang di casting atau di convert, karena banyak sekali tipe data yang di casting pada kasus kali ini.

4. Solusi dari Permasalahan yang Dihadapi
Solusi yang dilakukan adalah dengan menyusuri satu per satu hasil castingan setiap tipe data.
5. Nama teman yang membantu memecahkan permasalahan di persoalan ini.
-

SOAL 5 STRING

1. Screenshot Program

```
1 package strings;
2
3 public class formatString {
4
5     public static void main(String[] args) {
6         String A = "hello";
7         String B = "java";
8
9         // Jumlah panjang dari variabel A dan B
10        int jumlahKarakter = A.length() + B.length();
11        System.out.println(jumlahKarakter);
12
13        // Membandingkan string dengan konsep lexicographically
14        if(A.compareTo(B) < 0) {
15            System.out.println("No");
16        } else if(A.compareTo(B) == 0) {
17            System.out.println("Same");
18        } else {
19            System.out.println("Yes");
20        }
21
22        // Mengubah setiap huruf pertama menjadi uppercase dan menggabungkan variabel A & B
23        String firstCapitalA = A.substring(0,1).toUpperCase() + A.substring(1);
24        String firstCapitalB = B.substring(0,1).toUpperCase() + B.substring(1);
25        String joinKata = firstCapitalA + " " + firstCapitalB;
26        System.out.println(joinKata);
27
28    }
29
30 }
```

2. Screenshot Jawaban

```
9
No
Hello Java
```

Penjelasan Jawaban

1. Angka 9 itu didapat dari jumlah panjang karakter yang terdapat pada variabel A & variabel B. Pada kasus tersebut variabel A berjumlah karakter sebanyak 5 karakter dan variabel B berjumlah karakter sebanyak 4 karakter.
2. Membandingkan string dengan konsep lexicographically adalah membandingkan urutan dari setiap karakter dimulai dari karakter pertama yang dibandingkan. Membandingkannya ini kita hanya perlu menggunakan method **compareTo()**, kita dapat menggunakan method tersebut karena

variabel String merupakan sebuah class yang didalamnya terdapat method tersebut.

Maksud dari method tersebut adalah, kita bandingkan variabel A dan variabel B dimulai dari karakter pertama dengan membandingkannya sesuai urutan alphabet.

3. Dalam membuat karakter pertama saja yang berhuruf kapital maka kita perlu menggunakan sebuah method yaitu **substring()**. Method substring ini artinya kita ambil karakter dengan indeks 0 sampai indeks 1 yang mana itu adalah karakter pertama. Setelah itu, kita gunakan method **upperCase()** untuk membuat karakter tersebut menjadi huruf kapital. Lalu untuk karakter-karakter selanjutnya maka tinggal tambahkan juga substring yang dimulai dari indeks 1 atau artinya karakter-karakter selanjutnya.

3. Permasalahan yang Dihadapi

Permasalahan yang dihadapi adalah memahami konsep lexicographically dan bagaimana atau method apa yang harus digunakan untuk membandingkan karakter dengan konsep lexicographically. Permasalahan selanjutnya yaitu cara mengambil 1 karakter dan menjadikannya huruf kapital.

4. Solusi dari Permasalahan yang Dihadapi

Solusi untuk menghadapi permasalahan yang dialami adalah dengan cara membaca dokumentasi-dokumentasi di internet tentang lexicographically, cara mendapatkan solusi ini, saya lakukan juga pada permasalahan kedua saya. Berikut link yang menurut saya cukup untuk menemukan solusi permasalahan.

Link konsep Lexicographically.

<https://coderolls.com/compare-two-strings-lexicographically-in-java/>

Link mengubah karakter pertama menjadi huruf kapital.

<https://attacomsian.com/blog/capitalize-first-letter-of-string-java>

5. Nama teman yang membantu memecahkan permasalahan di persoalan ini.

-