

Laporan Pengerjaan Praktikum Teknik Pemrograman Pertemuan 2

Oleh:

Nama : Muhammad Nabil Syauqi Rasyiq

NIM : 241524018

Kelas : 1A



**Sarjana Terapan Program Studi Teknik Informatika
Jurusan Teknik Komputer dan Informatika
Politeknik Negeri Bandung
2025**

DAFTAR ISI

DAFTAR ISI	2
SOAL 1	3
Hasil screenshot akhir	3
Proses pengerjaan, kendala, dan solusinya	3
Source Code	4
SOAL 2	5
Hasil screenshoot	5
Soal dan jawaban pertanyaan.....	5
Proses pengerjaan, kendala, dan solusinya	5
Source code	6
SOAL 3	7
Hasil screenshoot	7
Soal dan Jawaban	7
Proses pengerjaan, kendala, dan solusinya	7
Source code	8
SOAL 4	8
Hasil screenshoot	8
Soal dan Jawaban	8
Proses Pengerjaan, kendala, dan solusinya	9
Source code	9
SOAL 5	10
Hasil screenshoot	10
Proses pengerjaan, Kendala dan solusinya	10
Source code	11
LESSON LEARN	13
REFERENSI	14

SOAL 1

Hasil screenshot akhir

```
C:\Users\Lenovo\Desktop\Tekprog\Pertemuan 2\tugas Praktikum\Soal 1\bin>java Datatype
5 dapat ditampung pada tipe data:
==> Byte
==> Short
==> Integer
==> Long
-150 dapat ditampung pada tipe data:
==> Short
==> Integer
==> Long
150000 dapat ditampung pada tipe data:
==> Integer
==> Long
1500000000 dapat ditampung pada tipe data:
==> Integer
==> Long
21333333333333333333333333333333 can't be fitted anywhere          SOAL 1
-1000000000000000 dapat ditampung pada tipe data:
==> Long

C:\Users\Lenovo\Desktop\Tekprog\Pertemuan 2\tugas Praktikum\Soal 1\bin>
```

Proses pengerjaan, kendala, dan solusinya

- ❖ Sebelum melakukan implementasi program saya terlebih dahulu mengidentifikasi Panjang dari Setiap tipe data (batasan). Lalu setelah melakukan identifikasi, saya terlebih dahulu membuat kerangka program. Setelah kerangka program sudah dirasa tepat dan cukup saya mulai mengimplementasikan kerangka program menjadi sebuah Program java.
- ❖ Pada soal kali ini saya mengalami beberapa kendala seperti bagaimana caranya untuk menampung nilai (angka) yang akan di cek. Hal ini menjadi kendala bagi saya karena ketika mencoba menyimpan semua nilai yang akan di uji pada type data long, masih ada angka/nilai yang tidak bisa ditampung karena melebihi kapasitas tipe data long.
- ❖ Untuk menyelesaikan kendala yang saya alami saya coba mencarinya di internet. Setelah melakukan pencarian saya menemukan sebuah class BigInteger yang dapat menampung angka tanpa batasan. BigInteger dapat kita gunakan dengan mengimport java.math. setelah mengimplementasikan solusi ini saya dapat menyelesaikan soal nomor 1.
- ❖ Pada soal pertama ini saya melakukan sebuah pengembangan dengan mengimport java.util.* agar dapat menggunakan Scanner yang saya gunakan untuk meminta input user.
- ❖ Pada soal pertama ini saya mengerjakannya secara mandiri tanpa bantuan dari teman. Saya berhasil mempelajarinya melalui Internet.

Source Code

[illegible]

SOAL 2

Hasil screenshoot

```
C:\Users\Lenovo\Desktop\Tekprog\Pertemuan 2\Tugas Praktikum\Soal 2\bin>java Constants
Paper size in centimeters: 21.59 by 27.94

C:\Users\Lenovo\Desktop\Tekprog\Pertemuan 2\Tugas Praktikum\Soal 2\bin>java Constants2
Paper size in centimeters: 21.59 by 27.94

C:\Users\Lenovo\Desktop\Tekprog\Pertemuan 2\Tugas Praktikum\Soal 2\bin>_
```

SOAL 2

Hasil screenshoot

Soal dan jawaban pertanyaan

1. Bagaimana output dari masing masing class Constants dan Constants2?
Output dari kedua program sama yaitu :
Paper size in centimeters: 21.59 by 27.94
2. Apa perbedaan penggunaan final double dengan public static final double?

Pada contoh program yang di berikan kita tidak bisa melihat secara langsung letak perbedaan dari penggunaan final double dan public static final double. Namun kenyataannya kedua hal tersebut jelas berbeda. Salah satu letak perbedaan diantara keduanya adalah pada batasan pengaksesannya. Final double membuat variabel tidak bisa di ubah sama sekali. Penempatannya di dalam method membuat ia tidak bisa di akses di luar public static *void* main(String[] args){}. Sedangkan pada penggunaan public static final double, variabel juga tidak dapat di ubah namun dapat di akses di semua method pada class yang sama.

Kesimpulannya perbedaan antara kedua itu adalah bila final double hanya dapat di akses di dalam method sebagai variabel local sedangkan public static final double dapat di akses secara global pada class yang sama.

Proses pengerjaan, kendala, dan solusinya

- ❖ Untuk mengerjakan soal ini saya tidak memerlukan banyak eksplorasi. Sebagian besar telah saya ketahui sebelumnya jadi tidak ada kendala yang besar. Namun mungkin saya perlu mencari tahu apakah yang saya ketahui sudah lengkap apa belum
- ❖ Untuk menyelesaikan kendala saya melakukan eksplorasi ulang dan mendapati penjelasan yang mungkin cukup untuk menjawab.
- ❖ Soal ini juga saya selesaikan secara mandiri karena saya merasa masih mampu untuk mempelajarinya.

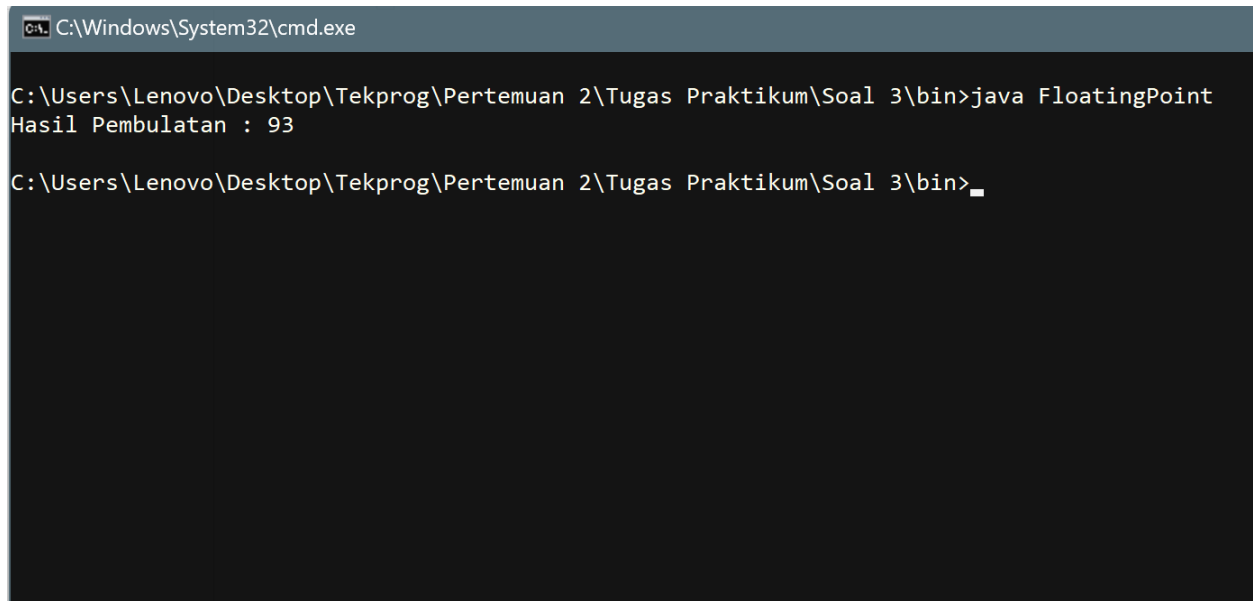
Source code

```
1 public class Constants {
2     public static void main(String[] args)
3     {
4         final double CM_PER_INCH = 2.54;
5         double paperWidth = 8.5;
6         double paperHeight = 11;
7         System.out.println("Paper size in centimeters: " + paperWidth * CM_PER_INCH
8         + " by " + paperHeight * CM_PER_INCH);
9     }
10 }
11 }
12 }
```

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

SOAL 3

Hasil screenshoot



```
C:\Windows\System32\cmd.exe

C:\Users\Lenovo\Desktop\Tekprog\Pertemuan 2\Tugas Praktikum\Soal 3\bin>java FloatingPoint
Hasil Pembulatan : 93

C:\Users\Lenovo\Desktop\Tekprog\Pertemuan 2\Tugas Praktikum\Soal 3\bin>
```

Soal dan Jawaban

Math Class berisi bermacam-macam fungsi matematika seperti pada contoh diatas pada penggunaan round(x), terdapat beberapa pertanyaan yang perlu untuk dijelaskan:

1. Pada kasus berikut jelaskan nilai **nx** setelah digunakan **Math.round(x); !**

Setelah menggunakan Math.round(x) maka nilai nx akan berubah menjadi 93. Hal ini terjadi sebab Math.round akan membulatkan nilai 92.98 dengan ketentuan :

1. bila nilai koma lebih dari sama dengan 5 maka akan di bulatkan ke atas
2. bila nilai koma kurang dari 5 maka akan di bulatkan ke bawah

dengan ketenteuan tersebut maka nilai nx akan berubah menjadi 93

2. Kenapa dibutuhkan cast (int) dalam penggunaan **Math.round(x) ?**

Cast perlu digunakan sebab Math.round hanya menmbulatkan nilai tidak mengubah tipe data. Sedangkan nilai pembulatan perlu di ubah menjadi tipe data integer agar nilai hasil pembulatan dapat di simpan di variabel nx yang bertipe data integer.

Proses pengerjaan, kendala, dan solusinya

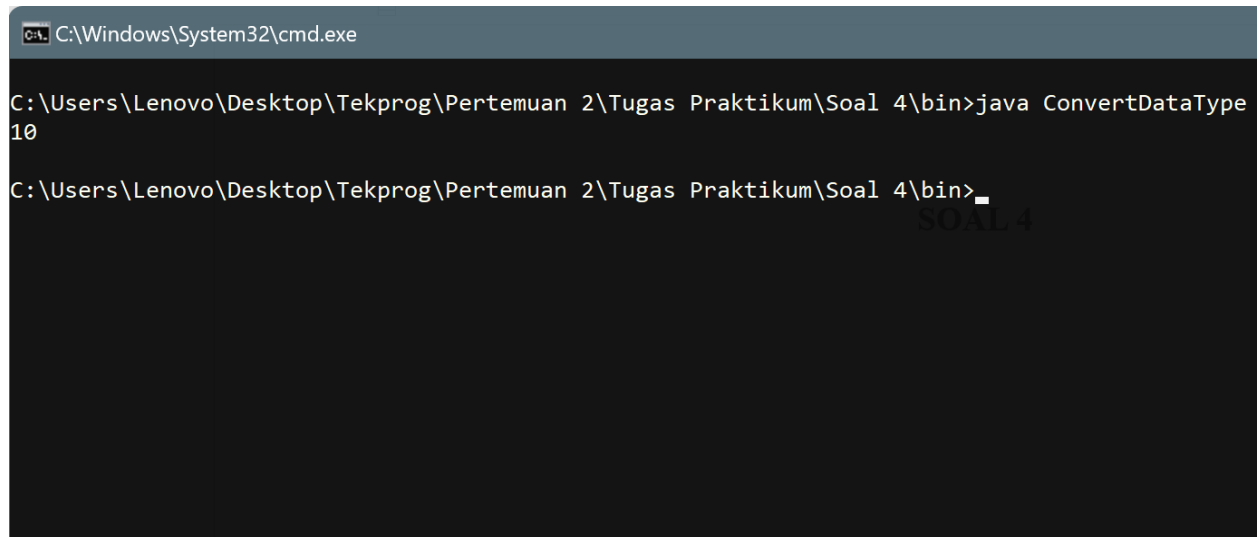
- ❖ Soal ini saya kerjakan dengan mengeksplorasi terkait kegunaan method math.round()
- ❖ Tidak ada kendala pada pengerjaan soal ini

Source code

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

SOAL 4

Hasil screenshoot



```
C:\Windows\System32\cmd.exe

C:\Users\Lenovo\Desktop\Tekprog\Pertemuan 2\Tugas Praktikum\Soal 4\bin>java ConvertDataType
10

C:\Users\Lenovo\Desktop\Tekprog\Pertemuan 2\Tugas Praktikum\Soal 4\bin>
```

Soal dan Jawaban

Program berikut melakukan convert tipe data yang berukuran besar ke kecil (long -> int -> short) dan (double -> float -> byte).

1. Jelaskan output nilai dari variable b.

Output yang dihasilkan dari variabel **b** adalah 10. **B** dapat bernilai 10 karena hasil dari casting 10.25 yang asalnya bertipe data double menjadi byte berikut alurnya:

- a. Nilai 10.25 yang terdapat pada variabel **d** di casting dari tipe data double menjadi float lalu disimpan di variabel **f**
 - b. Variabel **f** di cast menjadi tipe data long menjadi 10L lalu di parsing sebagai suatu parameter ke **methodone**
 - c. Pada **methodone** nilai mengalami perubahan dari long menjadi integer lalu menjadi short dan di return kembali ke program utama.
 - d. Pada program utama hasil return dari **methodone** yang bernilai 10 dengan tipe data short kembali di cast menjadi tipe data byte dan disimpan pada variabel **b** yang bertipe byte.
2. Jelaskan apa yang berubah dari variable **d** menjadi variable **b** setelah dilakukan cast? Setelah melakukan berbagai proses casting perubahan yang paling utama adalah perubahan tipe data. Hal ini akan sangat terlihat perbedaannya sebab tipe data double yang menyimpan nilai berbentuk decimal diubah menjadi byte yang menyimpan nilai bilangan bulat. Nilai dari variabel **d** juga berubah yang tadinya bernilai 10 lebih 0,25 menjadi nilai 10 bilangan bulat.

Note:

Alasan mengapa casting diatas dapat terjadi dengan baik adalah karena nilai 10.25 masuk atau dapat di tampung di semua tipe data yang ada

Proses Pengerjaan, kendala, dan solusinya

- ❖ Proses pengerjaan soal ini adalah saya perlu mentracing kode program. Hal ini diperlukan ketelitian. Kita perlu memastikan bahwa proses casting tipe data sesuai dengan ketentuan yang ada. Kita juga perlu mempertimbang kondisi perubahan decimal menjadi integer.
- ❖ Tidak ada kendala pada proses pengerjaan ini. Namun agar dapat menjawab dengan benar kita perlu benar benar memperhatikan perubahan tipe data juga perubahan nilainya. Yang pada kasus ini ternyata nilai berubah dari 10,25 menjadi 10.
- ❖ Solusi yang saya dapatkan dari pengerjaan soal ini adalah agar dapat mengelola kode dengan lebih baik kita perlu melakukan banyak Latihan. Latihan yang dapat kita lakukan adalah seperti memperbaiki kode yang error, banyak melakukan trasing pada suatu algoritma, dan cara – cara lainnya.

Source code

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

SOAL 5

Hasil screenshoot

```
C:\Windows\System32\cmd.exe

C:\Users\Lenovo\Desktop\Tekprog\Pertemuan 2\Tugas Praktikum\Soal 5\bin>java CompareStrings
Banyak kata : 9
Apakah termasuk lexicographically? : No
Hello Java

C:\Users\Lenovo\Desktop\Tekprog\Pertemuan 2\Tugas Praktikum\Soal 5\bin>_
```

Proses pengerjaan, Kendala dan solusinya

- ❖ Soal ini saya kerjakan dengan terlebih dahulu mengeksplorasi apa itu lexicographically?, bagaimana operasi string di Bahasa pemrograman java. Dari eksplorasi ini saya mendapatkan beberapa point utama yaitu:

- **Lexicographically** adalah istilah yang digunakan dalam pemrograman dan ilmu komputer untuk merujuk pada cara membandingkan string berdasarkan urutan leksikografis, yaitu urutan yang mirip dengan kamus (dictionary order).
- Pada prosesnya lexicographically akan membandingkan huruf pertama yang beda.
- Kita dapat menggunakan substring(parameter indeks) untuk mengambil kata yang ada pada suatu string prosesnya adalah dengan memparssing parameter berupa indeks awal string yang mau diambil dan sampai indeks mana akan diambilnya.
- Untuk mengubah suatu string menjadi kapital kita dapat menggunakan method toUpperCase()
- Untuk membandingkan besar/ nilai dari string kita dapat menggunakan method compareTo() yang akan mereturn nilai selisih antar dua string yang dibandingkan yang mana jika bernilai negative berarti yang dibandingkan lebih kecil dari pembanding.
- ❖ Setelah mengeksplorasi saya mulai memetakan alur program yang akan dibuat. Lalu saya menimplementasikan alur program tersebut menjadi sebuah program dalam Bahasa java.
- ❖ Kendala dari pengerjaan soal nomor 5 ini terletak bagaimana cara kita dapat mengeksplorasi secara efisien. waktu yang saya gunakan untuk mengeksplorasi cukup banyak sehingga waktu pengerjaan menjadi lebih lama.
- ❖ Solusi dari kendala tersebut adalah saya harus lebih banyak membaca dokumentasi Bahasa pemrograman java di luar waktu praktikum.
- ❖ Walau soal nomor 5 ini juga saya selesaikan secara mandiri. Saya juga terinspirasi dengan hasil presentasi Darrel ketika menjelaskan soal ini.

Source code



```
1 public class CompareStrings {
2     public static void main(String[] args) {
3
4         String A = "hello";
5         String B = "java";
6
7         int panjang = A.length() + B.length();
8
9         System.out.println("Banyak kata : " + panjang);
10
11         boolean Cek = (A.compareTo(B) > 0);
12
13         A = A.substring(0,1).toUpperCase() + A.substring(1);
14         B = B.substring(0,1).toUpperCase() + B.substring(1);
15
16
17         System.out.println("Apakah termasuk lexicographically? : " + (Cek? "Yes" : "No") );
18
19         String Combine = A + " " + B;
20
21         System.out.println(Combine);
22
23     }
24 }
```

LESSON LEARN

Sangat banyak sekali yang dapat saya pelajari pada pertemuan kali ini. Mulai dari mengeksplorasi sintaks dalam Bahasa java yang saya belum familiar dengannya. Lalu Bahasa pemrograman java yang menerapkan konsep OOP membuat saya merasa lebih mudah dalam memetakan sebuah alur program. Saya juga belajar bahwa dengan berani presentasi akan membuat pemahaman saya lebih kuat dari biasanya.

LINK GITHUB : <https://github.com/Rasyiq603011/Tekprog-Pertemuan-2->

REFERENSI

<https://www.w3schools.com/java/default.asp>

<http://geeksforgeeks.org/java/>

Text book : Core Java Volume I--Fundamentals, 10th Edition

<https://chatgpt.com/>