

**LAPORAN PRAKTIKUM FUNDAMENTAL PROGRAMING  
STRUCTURES IN JAVA**



Disusun oleh :

ALDA PUJAMA

241511066

**PROGRAM STUDI TEKNIK INFORMATIKA D3  
JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA  
POLITEKNIK NEGERI BANDUNG  
2024**

## DAFTAR ISI

DAFTAR ISI .....	i
BAB I .....	2
HASIL PRAKTIKUM .....	2
A. Link Github Pengerjaan Praktikum .....	2
1. Link Github.....	2
B. Pengerjaan Soal .....	2
1. Welcome java .....	2
2. Soal 1 .....	3
3. Soal 1 Data Type .....	4
4. Soal 2 variable .....	7
5. Soal 3 operator .....	8
6. Soal 4 operator .....	9
7. Soal 5 string .....	10

# BAB I

## HASIL PRAKTIKUM

### A. Link Github Pengerjaan Praktikum

#### 1. Link Github

<https://github.com/Arud4Alda/Praktikum-1-fundamental>

### B. Pengerjaan Soal

#### 1. Welcome java

//This program displays a greeting for the reader.

//@version 1.30 2014-02-27

//@author Cay Horstmann

public class welcome

{

    public static void main(String[] args)

    {

        String greeting = "Welcome to Core Java!";

        System.out.println(greeting);

        for (int i=0; i < greeting.length(); i++)

            System.out.print("=");

        System.out.println();

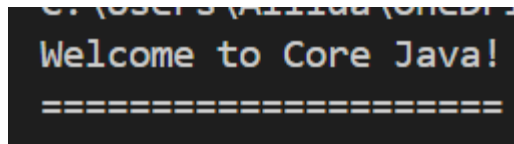
    }

}

Jawaban :

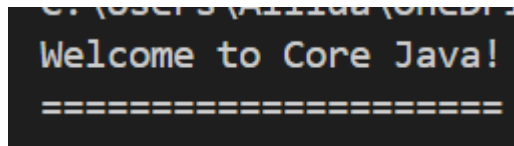
Screenshoot hasil akhir program





```
C:\Users\Alinda\OneDr...  
Welcome to Core Java!  
=====
```

Screenshoot jawaban soal



```
C:\Users\Alinda\OneDr...  
Welcome to Core Java!  
=====
```

Permasalahan yang dihadapi

Dalam pengerjaan welcome java ini alhamdulillah saya lancar dalam pengerjaannya. Jika permasalahan mungkin karena program java pertama saya jadi saya kebingungan membuat file pertama lalu cara mengcompile program javanya.

Solusi permasalahan yang dihadapi

Berdasarkan permasalahan diatas ternyata cukup langsung membuat file.java saja langsung dan mengcompilanya di terminal dengan cara memasukan perintah `javac namafile.java` <enter> lalu `java namafile`. Setelah itu muncullah hasil akhir program kita.

Teman yang membantu

Fatimah dan Rahma.

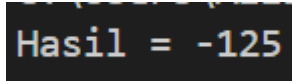
## 2. Soal 1

```
public class Soal1 {  
    public static void main(String[] args)    {  
        byte angka = 125;  
        byte angka2 = 6;  
        byte hasil = (byte) (angka+angka2) ;  
        System.out.println("Hasil = "+hasil);  
    }  
}
```

Berapa output yang keluar ? Tuliskan alasan dan referensinya

Jawaban:

Screenshoot hasil akhir program



```
Hasil = -125
```

Screenshoot jawaban soal

Output yang keluar itu -125 karena memang untuk tipe data byte Cuma bisa dari -128 sampai 127. Jadi jika  $125 + 6 = 131$  tapi karena byte setelah 127 kembali lagi ke -128 karenanya  $125 + 6 = -125$ .

Permasalahan yang dihadapi

Dalam pengerjaan welcome java ini alhamdulillah saya lancar dalam pengerjaanya.

### 3. Soal 1 Data Type

Java has 8 primitive data types; char, boolean, byte, short, int, long, float, and double. For this exercise, we'll work with the primitives used to hold integer values (byte, short, int, and long):

- A byte is an 8-bit signed integer.
- A short is a 16-bit signed integer.
- An int is a 32-bit signed integer.
- A long is a 64-bit signed integer

Given an input integer, you must determine which primitive data types are capable of properly storing that input.

Input Format

The first line contains an integer,  $T$ , denoting the number of test cases. Each test case,  $T$ , is comprised of a single line with an integer,  $n$ , which can be arbitrarily large or small.

Output Format

For each input variable `n` and appropriate primitive datatype, you must determine if the given primitives are capable of storing it. If yes, then print:

N can be fitted in:

\* datatype

If there is more than one appropriate data type, print each one on its own line and order them by size (i.e.: byte < short < int < long).

If the number cannot be stored in one of the four aforementioned primitives, print the line:

## N can't be fitted anywhere

Sample Input:

5

-150

150000

1500000000

21333

-10000000000000000

Sample Output:

-150 can be fitted in:

\* short

\* int

\* long

150000 can be fitted in:

\* int

\* long

1500000000 can be fitted in:

\* int

\* long

21333333333333333333333333333333333333

can't be fitted anywhere.

\* long

Jawaban :

### Screenshoot hasil akhir program

[illegible]

## Screenshoot jawaban soal

```

1 import java.util.Scanner;
2 public class Soaldatatype
3 {
4     public static void main(String[] args)
5     {
6         Scanner sc = new Scanner(System.in);
7         System.out.print("===== Uji coba tipe data =====\n\n");
8         System.out.print("Masukkan jumlah uji coba: ");
9         int t = sc.nextInt();
10        for(int i=0;i<t;i++)
11        {
12            System.out.print("\nMasukkan bilangan: ");
13            String m = sc.nextLine();
14            try {
15                long n = Long.parseLong(m); // coba parsing ke long
16                System.out.println(n + " can be fitted in:");
17
18                if (n >= Byte.MIN_VALUE && n <= Byte.MAX_VALUE) {
19                    System.out.println("  byte");
20                }
21
22                if (n >= Short.MIN_VALUE && n <= Short.MAX_VALUE) {
23                    System.out.println("  short");
24                }
25
26                if (n >= Integer.MIN_VALUE && n <= Integer.MAX_VALUE) {
27                    System.out.println("  int");
28                }
29
30                // pasti muat di long karena berhasil parse
31                System.out.println("  long");
32
33            } catch (NumberFormatException e) {
34                // kalau angka terlalu besar untuk long
35                System.out.println(m + " can't be fitted anywhere.");
36            }
37        }
38        sc.close();
39    }
40 }

```

Permasalahan yang dihadapi

Dalam pengerjaan soal data type ini awalnya saya menghadapi permasalahan untuk menentukan tipe data yang menampung value n karena jika menggunakan long untuk mengecek value yang tidak bisa masuk ke tipe data long jadi susah.

Solusi permasalahan yang dihadapi

Berdasarkan permasalahan diatas dibuatlah tipe data string yang menampung value n pertama lalu untuk pengecekan dimasukan ke dalam tipe data long juga jika tidak bisa dimasukan ke dalam tipe data long berarti akan menampilkan inputan tersebut can't be fitted anywhere.

#### 4. Soal 2 variable

Perhatikan baris program dibawah ini:

```
public class Constants {  
    public static void main(String[] args) {  
        final double CM_PER_INCH = 2.54;  
        double paperWidth = 8.5;  
        double paperHeight = 11;  
        System.out.println("Paper size in centimeters: " + paperWidth *  
            CM_PER_INCH + " by " + paperHeight * CM_PER_INCH);  
    }  
}
```

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

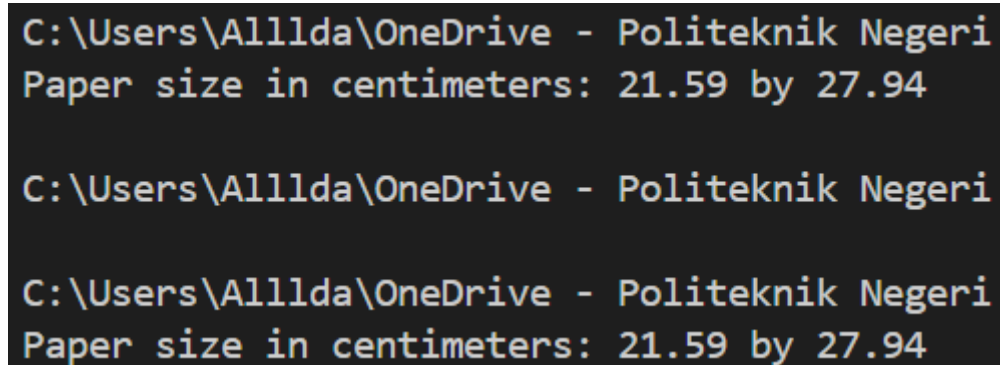


Dari 2 contoh baris program diatas, jawablah pertanyaan dibawah ini:

1. Bagaimana output dari masing masing class Constants dan Constants2?
2. Apa perbedaan penggunaan final double dengan public static final double?

Jawaban :

Screenshoot hasil akhir program



```
C:\Users\Alllda\OneDrive - Politeknik Negeri
Paper size in centimeters: 21.59 by 27.94

C:\Users\Alllda\OneDrive - Politeknik Negeri

C:\Users\Alllda\OneDrive - Politeknik Negeri
Paper size in centimeters: 21.59 by 27.94
```

Screenshoot jawaban soal

1. Output dari kedua program hasilnya sama dengan constanta2 :  
Paper size in centimeters: 21.59 by 27.94
2. final double = konstanta lokal yang hanya bisa diakses di dalam method tempatnya didefinisikan.  
public static final double = konstanta class-level yang bersifat global dan dapat diakses dari mana saja menggunakan  
NamaClass.NAMA\_KONSTANTA.

Permasalahan yang dihadapi

Dalam pengerjaan welcome java ini alhamdulillah saya lancar dalam pengerjaanya.

5. Soal 3 operator

Perhatikan baris program dibawah ini:

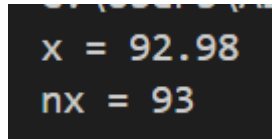
```
Class FloatingPoint{
    public static void main(String[] args) {
        double x = 92.98;
        int nx = (int) Math.round(x);
    }
}
```

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

Jawaban :

Screenshoot hasil akhir program



```
x = 92.98
nx = 93
```

Screenshoot jawaban soal

1. Nilai nx setelah menggunakan math round dari 92.98 dibulatkan menjadi 93
2. Penggunaan tipe data int karena hasil dari math round itu bilangan bulat (int) yang bukan desimal(float).

Permasalahan yang dihadapi

Dalam pengerjaan welcome java ini alhamdulillah saya lancar dalam pengerjaanya.

#### 6. Soal 4 operator

Perhatikan baris program dibawah ini:

```
class ConvertDataType {
    static short methodOne(long l) {
        int i = (int) l;
        return (short)i;
    }
    public static void main(String[] args) {
        double d = 10.25;
        float f = (float) d;
        byte b = (byte) methodOne((long) f);
        System.out.println(b);
    }
}
```

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.
2. Jelaskan apa yang berubah dari variable d menjadi variable b setelah dilakukan cast ?

Jawaban :

Screenshoot hasil akhir program

A small screenshot of a code editor or terminal window showing the number '10' in a light blue font on a dark background.

Screenshoot jawaban soal

1. Nilai b adalah 10
2. d = 10.25  
saat di-cast ke float = 10.25.  
lalu ke long = 10.  
lalu ke int = 10.  
lalu ke short = 10.

Permasalahan yang dihadapi

Dalam pengerjaan welcome java ini alhamdulillah saya lancar dalam pengerjaanya.

#### 7. Soal 5 string

This exercise is to test your understanding of Java Strings. A sample String declaration:

```
String myString = "Hello World!"
```

The elements of a String are called characters. The number of characters in a String is called the length, and it can be retrieved with the `String.length()` method.

Given two strings of lowercase English letters, A and B, perform the following operations:

1. Sum the lengths of A and B.
2. Determine if A is lexicographically larger than B (i.e: does B come before A in the dictionary?)

3. Capitalize the first letter in A and B and print them on a single line, separated by a space.

#### Input Format

The first line contains a string A. The second line contains another string B. The strings are comprised of only lowercase English letters.

#### Output Format

There are three lines of output:

For the first line, sum the lengths of A and B.

For the second line, write Yes if A is lexicographically greater than B otherwise print No instead.

For the third line, capitalize the first letter in both A and B and print them on a single line, separated by a space.

#### Sample Input 0

hello

java

#### Sample Output 0

9

No

Hello Java

Jawaban :

Screenshoot hasil akhir program

```
Masukan kata pertama dan kedua:
hello
java
Jumlah semua hurup :
9

Apakah kata java lebih banyak hurupnya?
No

Mengkapitalkan hurup pertama kata :
Hello Java
```

## Screenshoot jawaban soal



```
1 import java.util.Scanner;
2
3 public class testString
4 {
5     public static void main(String[] args)
6     {
7         //sistem input
8         Scanner sc = new Scanner(System.in);
9         System.out.println("Masukan kata pertama dan kedua:");
10        String A = sc.next();
11        String B = sc.next();
12
13        // 1. jumlah panjang
14        System.out.println("Jumlah semua hurup : ");
15        System.out.println(A.length() + B.length());
16
17        // 2. Lexicographical
18        System.out.println("\nApakah kata "+ B + " lebih banyak hurupnya? ");
19        System.out.println(A.compareTo(B) > 0 ? "Yes" : "No");
20
21        // 3. kapitalisasi
22        System.out.println("\nMengkapitalkan hurup pertama kata :");
23        String capA = A.substring(0, 1).toUpperCase() + A.substring(1);
24        String capB = B.substring(0, 1).toUpperCase() + B.substring(1);
25        System.out.println(capA + " " + capB);
26
27        sc.close();
28
29        /*
30        kata 1 = hello
31        kata 2 = java
32
33        output:
34        Jumlah semua hurup : 9
35        Apakah kata java lebih banyak hurupnya? No
36        Mengkapitalkan hurup pertama kata : Hello Java
37        */
38    }
39 }
```

## Permasalahan yang dihadapi

Untuk permasalahan itu lebih saya harus mengerti apa itu lexicographical itu apa? Lalu bagaimana caranya saya bisa mengkapitalkan hurup pertama setiap kata?

## Solusi permasalahan

Untuk permasalahan diatas Solusi yang saya itu pertama mencari tahu tentang lexicographical itu apa lalu juga belajar di w3scool terkait pemograman java untuk mengkapitalkan hurup pertama setiap kata dengan mengambil atau memisahkan hurup pertamanya lalu kapitalkan baru digabung dengan sisa katanya.