

# **LAPORAN PRAKTIKUM FUNDAMENTAL PROGRAMING STRUCTURES IN JAVA**

Laporan ini dibuat untuk memnuhi tugas mata kuliah Pemrograman Berorientasi Objek  
(PRAKTIKUM)



Disusun oleh :  
Arief Rahman Ahmadhusein (211511009)

PROGRAM STUDI DIPLOMA III TEKNIK INFORMATIKA  
JURUSAN TEKNIK KOMPUTER DAN INFORMATIKA  
POLITEKNIK NEGERI BANDUNG 2022

## DAFTAR ISI

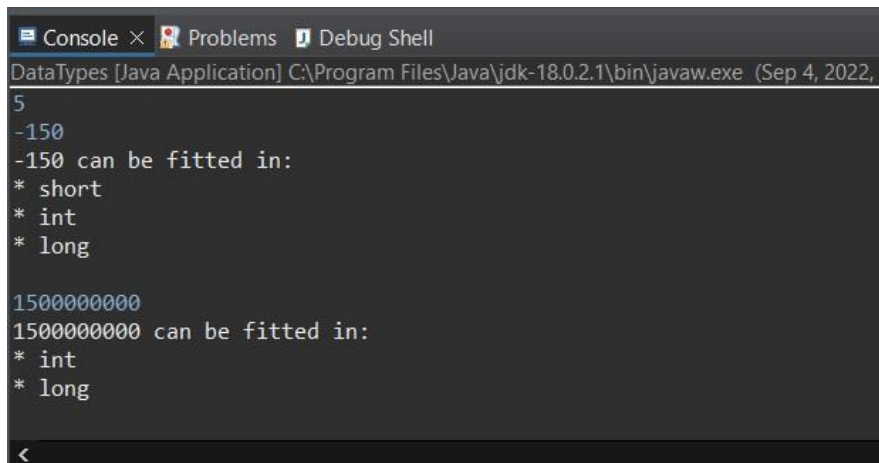
SOAL 1 .....	3
SOAL 2 .....	4
SOAL 3 .....	5
SOAL 4 .....	6
SOAL 5 .....	7

## SOAL 1

Source code yang digunakan

```
1 import java.util.*;
2
3
4
5
6 class DataTypes{
7     public static void main(String []argh)
8     {
9         Scanner sc = new Scanner(System.in);
10        int t=sc.nextInt();
11
12        for(int i=0;i<t;i++)
13        {
14            try
15            {
16                long x=sc.nextLong();
17                System.out.println(x+" can be fitted in: ");
18                if(x>=-128 && x<=127) {System.out.println("* byte");}
19                if(x>=Short.MIN_VALUE && x<=Short.MAX_VALUE){System.out.println("* short");}
20                if(x>=Integer.MIN_VALUE && x<=Integer.MAX_VALUE){System.out.println("* int");}
21                if(x>=Long.MIN_VALUE && x<=Long.MAX_VALUE){System.out.println("* long");}
22            }
23            catch(Exception e)
24            {
25                System.out.println(sc.next()+" can't be fitted anywhere.");
26            }
27        }
28    }
29 }
30 }
```

Hasil dari program terkait



```
Console x Problems Debug Shell
DataTypes [Java Application] C:\Program Files\Java\jdk-18.0.2.1\bin\javaw.exe (Sep 4, 2022, 7
5
-150
-150 can be fitted in:
* short
* int
* long

1500000000
1500000000 can be fitted in:
* int
* long
<
```

Kendala

- Terdapat bug kesalahan pada awal pembuatan program dimana program tidak dapat di run karena membuat java dalam notepad dengan mengubah extensi langsung dengan java sehingga file corrupt tidak bisa dibaca.

Solusi

- Membuat project baru didalam eclipse sehingga file tidak korup.

## SOAL 2

Source code

```
20 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/1
5 package constants;
6
7 /**
8  *
9  * @author ASUS
10 */
11 public class Constants {
12     public static void main(String[] args) {
13         final double CM_PER_INCH = 2.54;
14         double paperWidth = 8.5;
15         double paperHeight = 11;
16         System.out.println("Paper size in centimeters: " +
17             paperWidth * CM_PER_INCH + " by " + paperHeight *
18             CM_PER_INCH);
19     }
20 }
21
```

```
20 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/1
5 package constants2;
6
7 /**
8  *
9  * @author ASUS
10 */
11 public class Constants2 {
12     public static final double CM_PER_INCH = 2.54;
13     public static void main(String[] args) {
14         double paperWidth = 8.5;
15         double paperHeight = 11;
16         System.out.println("Paper size in centimeters: " + pa
17             CM_PER_INCH + " by " + paperHeight * CM_PER_INCH)
18     }
19 }
20
```

Apa perbedaan antara penggunaan final double dengan public static final double

- Penggunaan final double hanya dapat digunakan pada class tempat dimana final double tersebut didefinisikan sedang kan public static bisa dipakai oleh class yang lain jadi tidak hanya pada class tempat dimana final double tersebut di define

## SOAL 3

### Source Code

```
1
2 /**
3  *
4  * @author MSI
5  */
6 public class FloatingPoint {
7     public static void main(String[] args) {
8         double x = 92.98;
9         int nx = (int) Math.round(x);
10        System.out.println("Output Nilai nx adalah : " + nx)
11    }}
12
13 |
```

### Hasil akhir program

```
Output Nilai nx adalah : 93
```

- Penggunaan `Math.round` untuk pembulatan bilangan decimal ke bilangan bulat. Dapat dilihat bahwa hasil menjadi 93 karena bilangan bulat terdekat dari 92.98 adalah 93 jadi dapat disimpulkan bahwa penggunaan fungsi tersebut bisa dibilang untuk membulatkan bilangan decimal ke bilangan terdekat.
- Penggunaan casting `(int)` bertujuan untuk mengubah data bertipe `double` menjadi data bertipe `integer` dikarenakan hasil output merupakan bilangan bulat dan bertipe `integer`

## SOAL 4

### Source Code

```
1  /**
2   *
3   * @author MSI
4   */
5  public class ConvertDataTypes {
6      static short methodOne(long l) {
7          int i = (int) l;
8          return (short) i;
9      }
10
11     public static void main(String[] args)
12     {
13         double d = 10.25;
14         float f = (float) d;
15         byte b = (byte) methodOne((long) f);
16         System.out.println(b);
17     }
18 }
```

Hasil akhir program

```
<terminated> ConvertDataTypes [Java Application] C:\Program
10
```

- Output nilai dari variable b adalah 10 seperti yang tertera pada hasil run program dimana nilai tersebut hasil daripada casting nilai variable sebelumnya
- Variable d bertipe double diubah tipe datanya menjadi float lalu dimasukkan kedalam variable f, variable f juga diganti tipe datanya menjadi byte dan akhirnya dimasukan lagi ke var b dengan tipe data byte. Pada pertengahan casting antara variable f dan b terdapat sebuah function yang digunakan untuk mengubah tipe data menjadi integer dan dimasukkan sementara kedalam variable i yang kemudian nilainya di kembalikandan di casting menjadi tipe data short

## SOAL 5

### Source Code

```
1 import java.util.*;
2
3
4 public class SOAL5 {
5     public static void main(String[] args) {
6         Scanner sc=new Scanner(System.in);
7         String A=sc.next();
8         String B=sc.next();
9
10        System.out.println(A.length()+B.length());
11        System.out.println( A.compareTo(B)>0 ? "Yes" : "No" );
12        System.out.println( A.substring(0,1).toUpperCase()+A.substring(1,A.length())+" "+B.substring(0,1).toUpperCase()+B.substring(1,B.length()));
13    }
14 }
15
```

### Hasil akhir program

```
<terminated> SOAL5 [Java Application]
hello

java
9
No
Hello Java
```