

**LAPORAN**  
**FUNDAMENTAL PROGRAMMING STRUCTURES IN JAVA**  
Diajukan untuk memenuhi satu tugas Mata Kuliah Pemograman Berbasis Objek



Disusun oleh :  
Seruni Libertina Islami (241511064)

**Jurusan Teknik Komputer dan Informatika**  
**Program Studi D-3 Teknik Informatika**  
**Politeknik Negeri Bandung**  
**2025**

Instalasi, Setting Environment Variable, Compile Welcome.java using CMD, run Welcome.java using CMD :

```
Microsoft Windows [Version 10.0.26100.4061]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Seruni>java -version
java version "24.0.2" 2025-07-15
Java(TM) SE Runtime Environment (build 24.0.2+12-54)
Java HotSpot(TM) 64-Bit Server VM (build 24.0.2+12-54, mixed mode, sharing)
```

```
D:\SEMESTER 3\PEMOGRAMAN BERORIENTASI OBJEK\W1_064_Seruni Libertia I
slami>javac Welcome.java

D:\SEMESTER 3\PEMOGRAMAN BERORIENTASI OBJEK\W1_064_Seruni Libertia I
slami>java Welcome
Welcome to Core Java!
=====
```

```
D:\SEMESTER 3\PEMOGRAMAN BERORIENTASI OBJEK\W1_064_Seruni Libertia I
slami>javac ContohSoal1.java

D:\SEMESTER 3\PEMOGRAMAN BERORIENTASI OBJEK\W1_064_Seruni Libertia I
slami>java ContohSoal1
Hasil 1 -125

D:\SEMESTER 3\PEMOGRAMAN BERORIENTASI OBJEK\W1_064_Seruni Libertia I
slami>
```

## Soal 1 Data Types

```

1  Soal1.java > ...
2  /*
3  Soal 1 Data Types
4  Nama : Seruni Libertina Islami
5  Kelas : 2B - D3
6  NIM : 241511064
7  */
8  import java.util.*;          // Scanner untuk input
9  import java.math.BigInteger; // Untuk angka besar
10
11 public class Soal1 {
12     Run | Debug
13     public static void main(String[] args) {
14         Scanner sc = new Scanner(System.in); // Scanner untuk membaca input
15
16         int T = sc.nextInt(); // Jumlah test case yang akan diuji
17
18         // Definisi batas tiap tipe data
19         final BigInteger BYTE_MIN = BigInteger.valueOf(Byte.MIN_VALUE);
20         final BigInteger BYTE_MAX = BigInteger.valueOf(Byte.MAX_VALUE);
21         final BigInteger SHORT_MIN = BigInteger.valueOf(Short.MIN_VALUE);
22         final BigInteger SHORT_MAX = BigInteger.valueOf(Short.MAX_VALUE);
23         final BigInteger INT_MIN = BigInteger.valueOf(Integer.MIN_VALUE);
24         final BigInteger INT_MAX = BigInteger.valueOf(Integer.MAX_VALUE);
25         final BigInteger LONG_MIN = BigInteger.valueOf(Long.MIN_VALUE);
26         final BigInteger LONG_MAX = BigInteger.valueOf(Long.MAX_VALUE);
27
28         while (T-- > 0) {
29             String s = sc.next(); // Baca input sebagai string biar bisa menampung angka besar sekali
30
31             try {
32                 BigInteger n = new BigInteger(s); // Konversi string ke BigInteger
33
34                 // Cek apakah angka muat di tipe long
35                 if (n.compareTo(LONG_MIN) >= 0 && n.compareTo(LONG_MAX) <= 0) {
36                     System.out.println(s + " can be fitted in:");
37
38                     // Cek apakah muat dalam byte
39                     if (n.compareTo(BYTE_MIN) >= 0 && n.compareTo(BYTE_MAX) <= 0) {
40                         System.out.println(x:" byte");
41                     }
42
43                     // Cek apakah muat dalam short
44                     if (n.compareTo(SHORT_MIN) >= 0 && n.compareTo(SHORT_MAX) <= 0) {
45                         System.out.println(x:" short");
46                     }
47
48                     // Cek apakah muat dalam int
49                     if (n.compareTo(INT_MIN) >= 0 && n.compareTo(INT_MAX) <= 0) {
50                         System.out.println(x:" int");
51                     }
52
53                     // Karena sudah dipastikan muat di long, otomatis bisa dicetak
54                     System.out.println(x:" long");
55                 } else {
56                     // Jika lebih besar dari long (tidak muat di tipe data manapun)
57                     System.out.println(s + " can't be fitted anywhere.");
58                 }
59             } catch (NumberFormatException e) {
60                 // Jika input bukan angka valid
61                 System.out.println(s + " can't be fitted anywhere.");
62             }
63         }
64         sc.close();
65     }
66 }

```

[illegible]

### Kesimpulan:

Setiap angka harus dibandingkan dengan batas tipe data primitif. Tipe data lebih besar bersifat inklusif. Urutan pengecekan harus dari terkecil ke terbesar (byte → short → int → long) agar output teratur dan sesuai soal.

## Soal 2

### a. Constants

```
J Soal2Constants.java > Soal2Constants
1  /*
2  Soal 2 Constants
3  Nama : Seruni Libertina Islami
4  Kelas : 2B - D3
5  NIM : 241511064
6  */
7
8  public class Soal2Constants {
9      Run | Debug
10     public static void main(String[] args) {
11         // Deklarasi konstanta lokal
12         // 1 inch = 2.54 cm
13         final double CM_PER_INCH = 2.54;
14
15         // Ukuran kertas dalam satuan inch
16         double paperWidth = 8.5;
17         double paperHeight = 11;
18         // Cetak ukuran kertas dalam centimeter
19         // Lebar = 8.5 inch x 2.54
20         // Tinggi = 11 inch x 2.54
21         System.out.println("Paper size in centimeters: "
22             + paperWidth * CM_PER_INCH + " by " + paperHeight * CM_PER_INCH);
23     }
24 }
25
```

```
PS D:\SEMESTER 3\PEMOGRAMAN BERORIENTASI OBJEK\W1_064_Seruni Libertina Islami> javac Soal2Constants.java
PS D:\SEMESTER 3\PEMOGRAMAN BERORIENTASI OBJEK\W1_064_Seruni Libertina Islami> java Soal2Constants
Paper size in centimeters: 21.59 by 27.94
PS D:\SEMESTER 3\PEMOGRAMAN BERORIENTASI OBJEK\W1_064_Seruni Libertina Islami>
```

### b. Constants2

```
J Soal2Constants2.java > Soal2Constants2
1  /*
2  Soal 2 Constants 2
3  Nama : Seruni Libertina Islami
4  Kelas : 2B - D3
5  NIM : 241511064
6  */
7
8
9  public class Soal2Constants2 {
10
11     public static final double CM_PER_INCH = 2.54;
12
13     Run | Debug
14     public static void main(String[] args) {
15         // Ukuran kertas dalam inch
16         double paperWidth = 8.5;
17         double paperHeight = 11;
18
19         // Konversi: inch x 2.54
20         System.out.println("Paper size in centimeters: "
21             + paperWidth * CM_PER_INCH + " by " + paperHeight * CM_PER_INCH);
22     }
23 }
```

```
PS D:\SEMESTER 3\PEMOGRAMAN BERORIENTASI OBJEK\W1_064_Seruni Libertina Islami> javac Soal2Constants2.java
PS D:\SEMESTER 3\PEMOGRAMAN BERORIENTASI OBJEK\W1_064_Seruni Libertina Islami> java Soal2Constants2
Paper size in centimeters: 21.59 by 27.94
PS D:\SEMESTER 3\PEMOGRAMAN BERORIENTASI OBJEK\W1_064_Seruni Libertina Islami>
```

1. Bagaimana output dari masing masing class Constants dan Constants2?

Lebar kertas:  $8.5 * 2.54 = 21.59$

Tinggi kertas:  $11 * 2.54 = 27.94$

Constants dan Constants2 menghasilkan output yang sama, karena perhitungannya identic

2. Apa perbedaan penggunaan final double dengan public static final double?

a. final double (di class Constants)

- Dideklarasikan di dalam method (main).
- Bersifat lokal, hanya bisa dipakai di method itu saja.
- final artinya nilainya konstan (tidak bisa diubah setelah diberi nilai).
- Tidak bisa diakses dari method lain atau kelas lain.

b. public static final double (di class Constants2)

Dideklarasikan di level kelas (bukan hanya di method).

- public : Bisa diakses dari luar kelas, misalnya Constants2.CM\_PER\_INCH.
- Static : Milik kelas, bukan milik objek; hanya ada satu salinan untuk semua objek.
- final : Nilainya konstan, tidak bisa diubah.

### Soal 3 Operators

```
J Soal3.java > Soal3
1  /*
2  Soal 2 Operators
3  Nama : Seruni Libertina Islami
4  Kelas : 2B - D3
5  NIM : 241511064
6  */
7
8  public class Soal3 {
9      Run | Debug
10     public static void main(String[] args) {
11         double x = 92.98;
12
13         // Math.round(double) mengembalikan long
14         long rounded = Math.round(x); // membulatkan angka double ke bilangan bulat, hasil
15         int nx = (int) rounded;      // cast ke int
16
17         System.out.println("x = " + x);
18         System.out.println("Math.round(x) = " + rounded); // menampilkan hasil pembulatan (1
19         System.out.println("nx = " + nx);
20     }
21 }
22
```

```

PS D:\SEMESTER 3\PEMOGRAMAN BERORIENTASI OBJEK\W1_064_Seruni Libertina Islami> javac Soal3.java
PS D:\SEMESTER 3\PEMOGRAMAN BERORIENTASI OBJEK\W1_064_Seruni Libertina Islami> java Soal3
x = 92.98
Math.round(x) = 93
nx = 93
PS D:\SEMESTER 3\PEMOGRAMAN BERORIENTASI OBJEK\W1_064_Seruni Libertina Islami>

```

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

Nilai nx setelah Math.round(x)

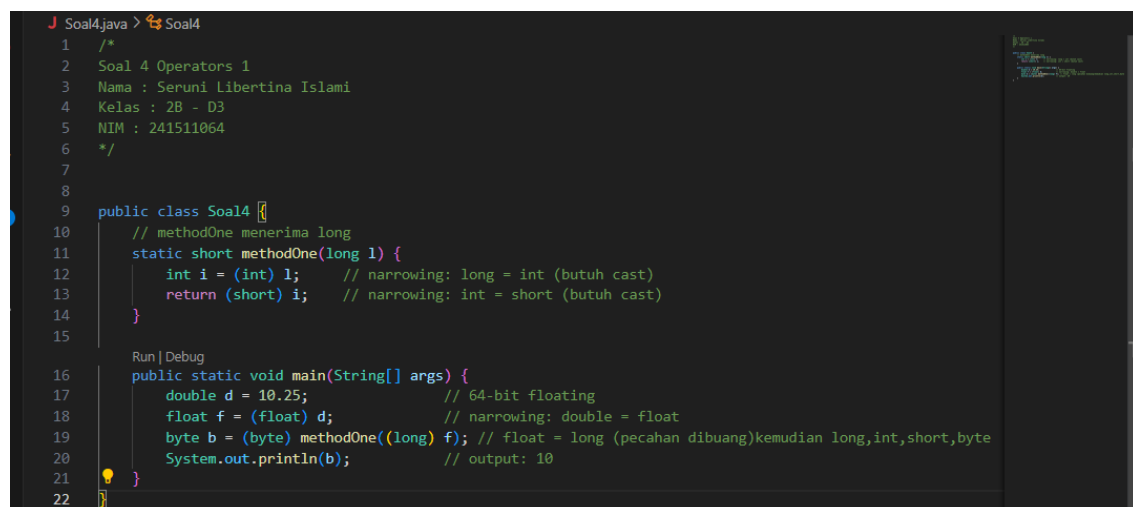
- x = 92.98
- Math.round(92.98) membulatkan ke bilangan bulat terdekat  $\Rightarrow$  93
- Setelah dicast ke int, nx = 93.

Untuk bilangan positif. Jadi  $92.98 \rightarrow 93$ . Untuk bilangan negatif, misal  $-1.5 \rightarrow -1$ .

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

Karena Math.round(double) mengembalikan tipe long, sedangkan variabel nx dideklarasikan int. Mengubah long ke int adalah narrowing conversion (mengecilkan tipe).

#### Soal 4 Operators (1)



```

1  /*
2  Soal 4 Operators 1
3  Nama : Seruni Libertina Islami
4  Kelas : 2B - D3
5  NIM : 241511064
6  */
7
8
9  public class Soal4 {
10     // methodOne menerima long
11     static short methodOne(long l) {
12         int i = (int) l; // narrowing: long = int (butuh cast)
13         return (short) i; // narrowing: int = short (butuh cast)
14     }
15
16     Run | Debug
17     public static void main(String[] args) {
18         double d = 10.25; // 64-bit floating
19         float f = (float) d; // narrowing: double = float
20         byte b = (byte) methodOne((long) f); // float = long (pecahan dibuang) kemudian long, int, short, byte
21         System.out.println(b); // output: 10
22     }
23 }

```

```

PS D:\SEMESTER 3\PEMOGRAMAN BERORIENTASI OBJEK\W1_064_Seruni Libertina Islami> javac Soal3.java
PS D:\SEMESTER 3\PEMOGRAMAN BERORIENTASI OBJEK\W1_064_Seruni Libertina Islami> java Soal3
x = 92.98
Math.round(x) = 93
nx = 93
PS D:\SEMESTER 3\PEMOGRAMAN BERORIENTASI OBJEK\W1_064_Seruni Libertina Islami>

```

1. Jelaskan output nilai dari variable b.

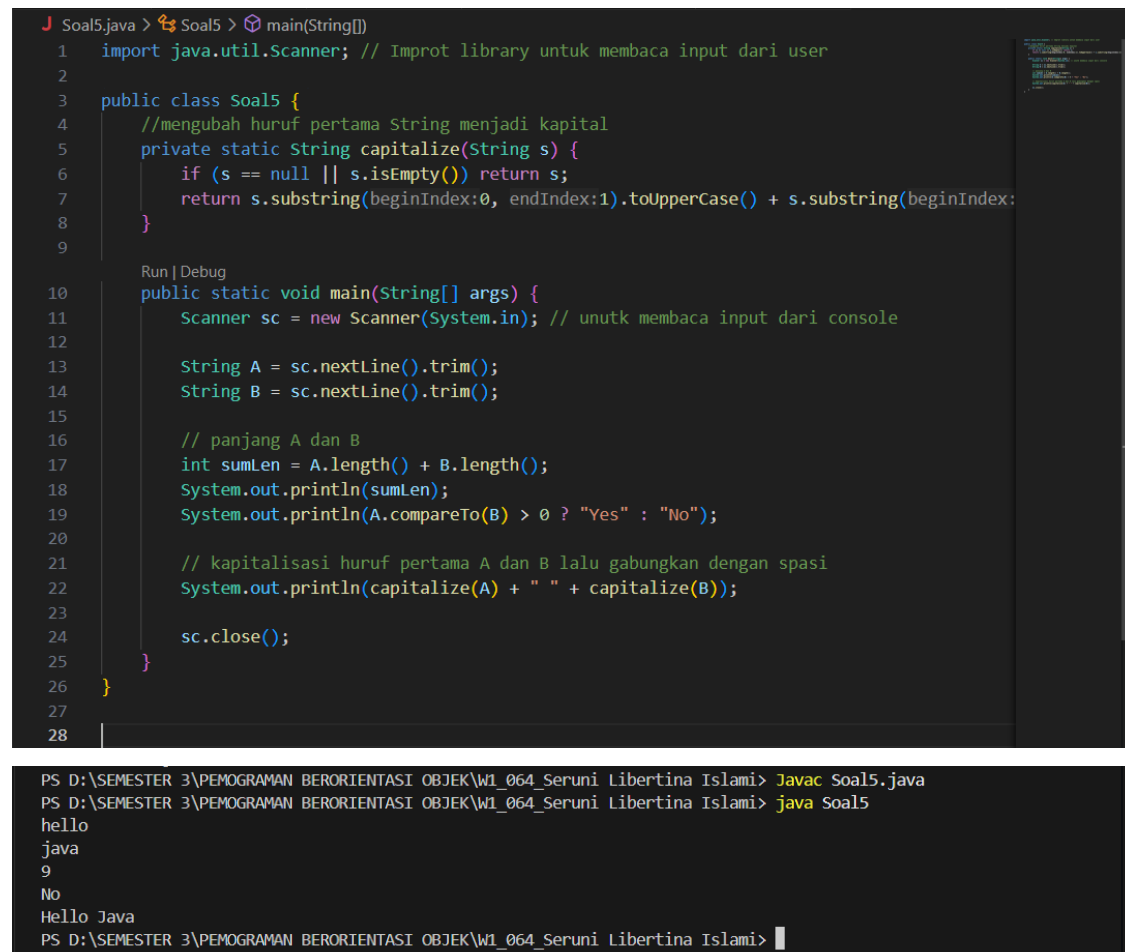
Output = 10

- d = 10.25
- f = (float) d  $\rightarrow$  tetap  $\approx$  10.25
- (long) f  $\rightarrow$  10 (bagian desimal dipotong tidak dibulatkan)
- methodOne mengembalikan short 10

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

Tipe data mengecil bertahap : Double → float → long → int → short → byte. Presisi hilang saat float ke long: digit desimal dihilangkan (10.25 jadi 10).

## Soal 5 Strings



```
J Soal5.java > Soal5 > main(String[])
1  import java.util.Scanner; // Improt library untuk membaca input dari user
2
3  public class Soal5 {
4      //mengubah huruf pertama String menjadi kapital
5      private static String capitalize(String s) {
6          if (s == null || s.isEmpty()) return s;
7          return s.substring(beginIndex:0, endIndex:1).toUpperCase() + s.substring(beginIndex:
8      }
9
10     Run | Debug
11     public static void main(String[] args) {
12         Scanner sc = new Scanner(System.in); // unutk membaca input dari console
13
14         String A = sc.nextLine().trim();
15         String B = sc.nextLine().trim();
16
17         // panjang A dan B
18         int sumLen = A.length() + B.length();
19         System.out.println(sumLen);
20         System.out.println(A.compareTo(B) > 0 ? "Yes" : "No");
21
22         // kapitalisasi huruf pertama A dan B lalu gabungkan dengan spasi
23         System.out.println(capitalize(A) + " " + capitalize(B));
24
25         sc.close();
26     }
27
28
```

```
PS D:\SEMESTER 3\PEMOGRAMAN BERORIENTASI OBJEK\W1_064_Seruni Libertina Islami> javac Soal5.java
PS D:\SEMESTER 3\PEMOGRAMAN BERORIENTASI OBJEK\W1_064_Seruni Libertina Islami> java Soal5
hello
java
9
No
Hello Java
PS D:\SEMESTER 3\PEMOGRAMAN BERORIENTASI OBJEK\W1_064_Seruni Libertina Islami>
```

1. Sum the lengths of A and B.

Contoh inputnya “hello” dan “java” : panjang  $5+4 = 9$ . Tulis “Yes” jika A leksikografis lebih besar dari B, selain itu “No”. Contoh hello vs java = No karena “hello” datang sebelum “java” di kamus

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

Perbandingan leksikografis artinya dibandingkan urutannya di kamus/alfabet. Kalo di java itu dilakukan dengan `A.compareTo(B)` :

- Jika hasilnya  $> 0$  berarti A lebih besar dari B.
- Jika hasilnya  $< 0$  berarti A lebih kecil dari B.

- Jika hasilnya = 0 berarti A dan B sama.

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

- Huruf pertama dari hello = Hello
- Huruf pertama dari java = Java
- Digabung dengan spasi = Hello Java