

# **LAPORAN PRAKTIKUM PBO**



**Disusun Oleh:**  
**Muhamad Haikal Hariyanto (231511081)**  
**Jurusan Teknik Komputer dan Informatika**

**Program Studi D-3 Teknik Informatika**  
**Politeknik Negeri Bandung**

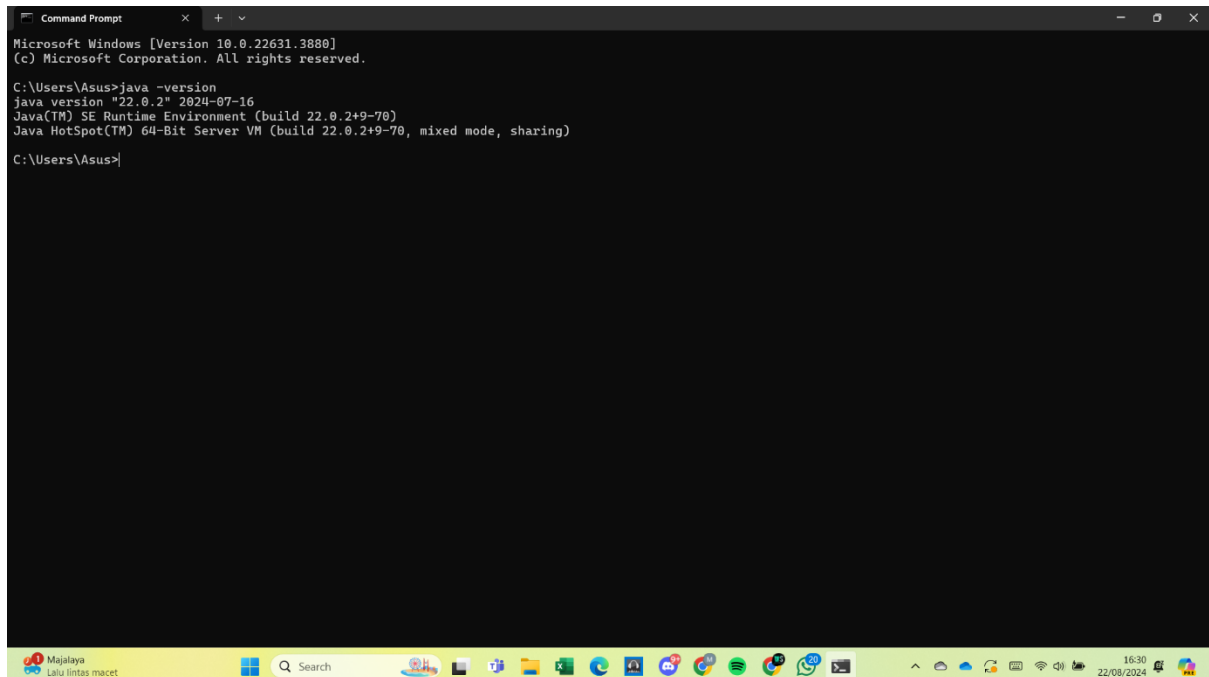
**22 Agustus 2024**

# Judul : Pengenalan Mata Kuliah Pemrograman Berorientasi Object

Hari : Kamis, 22 Agustus 2024

Persoalan :

## 1. Setup Software Environment



```
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Asus>java -version
java version "22.0.2" 2024-07-16
Java(TM) SE Runtime Environment (build 22.0.2+9-70)
Java HotSpot(TM) 64-Bit Server VM (build 22.0.2+9-70, mixed mode, sharing)

C:\Users\Asus>
```

Dalam persoalan ini saya tidak mengalami kendala sama sekali

## 2. Using Notepad & Command Line Tools

```
1 public class Welcome {
2     public static void main(String[] args)
3     {
4         String greeting = "Welcome to Core Java!";
5         System.out.println(greeting);
6         for (int i = 0; i < greeting.length(); i++)
7             System.out.print("=");
8         System.out.println();
9     }
10 }
11 }
```

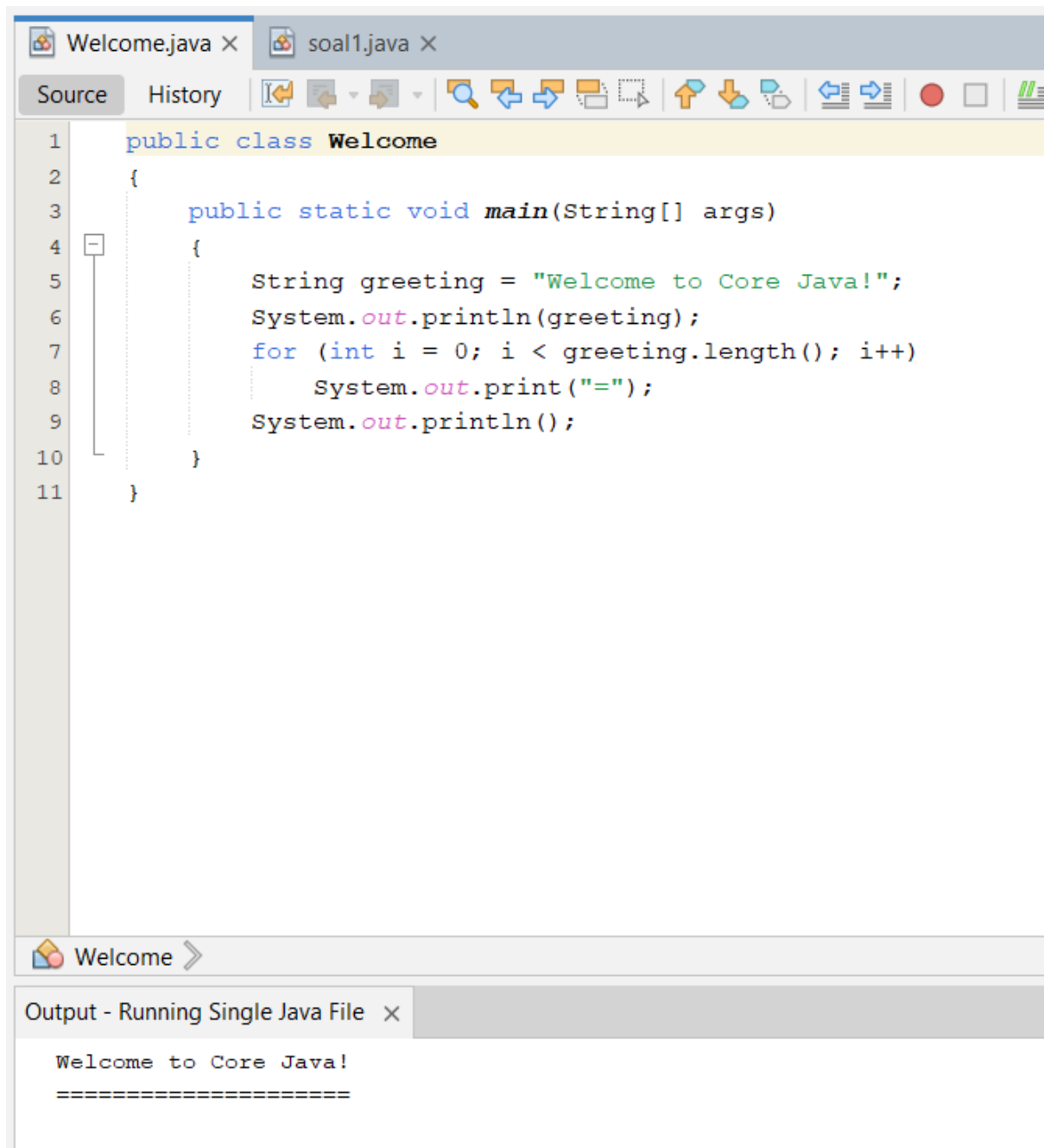
Specify a module path where to find annotation processors  
--processor-path <path>, -processorpath <path>  
Specify where to find annotation processors  
-profile <profile>  
Check that API used is available in the specified profile.  
This option is deprecated and may be removed in a future release.  
--release <release>  
Compile for the specified Java SE release.  
Supported releases:  
8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22  
-s <directory> Specify where to place generated source files  
--source <release>, -source <release>  
Provide source compatibility with the specified Java SE release.  
Supported releases:  
8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22  
--source-path <path>, -sourcepath <path>  
Specify where to find input source files  
--system <jdk>|none Override location of system modules  
--target <release>, -target <release>  
Generate class files suitable for the specified Java SE release.  
Supported releases:  
8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22  
--upgrade-module-path <path>  
Override location of upgradeable modules  
-verbose Output messages about what the compiler is doing  
--version, -version Version information  
-Werror Terminate compilation if warnings occur

PS C:\Users\Asus\OneDrive - Politeknik Negeri Bandung\semester 3\PBO> java Welcome  
Error: Could not find or load main class Welcome  
Caused by: java.lang.ClassNotFoundException: Welcome  
PS C:\Users\Asus\OneDrive - Politeknik Negeri Bandung\semester 3\PBO> cd "c:\Users\Asus\OneDrive - Politeknik Negeri Bandung\semester 3\PBO\" ; if (\$?) { javac welcome.java } ;  
if (\$?) { java welcome }  
Error: Could not find or load main class welcome  
Caused by: java.lang.NoClassDefFoundError: welcome (wrong name: Welcome)  
PS C:\Users\Asus\OneDrive - Politeknik Negeri Bandung\semester 3\PBO> java Welcome  
Welcome to Core Java!

```
1 /**
2  * This program displays a greeting for the reader.
3  * @version 1.30 2014-02-27
4  * @author Cay Horstmann
5  */
6 public class Welcome
7 {
8     public static void main(String[] args)
9     {
10         String greeting = "Welcome to Core Java!";
11         System.out.println(greeting);
12         for (int i = 0; i < greeting.length(); i++)
13             System.out.print("=");
14         System.out.println();
15     }
16 }
```

Dalam persoalan ini saya mengalami kendala karena tidak mengerti intruksi nya. Lalu saya menanyakan kepastian nya si tugas yang disuruh itu apa. Teman yang membantu saya “**Muhammad Fachry Khairiansyah**”

### 3. Using an Integrate Development Environment



The screenshot shows an IDE with two tabs: 'Welcome.java' and 'soal1.java'. The 'Source' tab is active, displaying the following Java code:

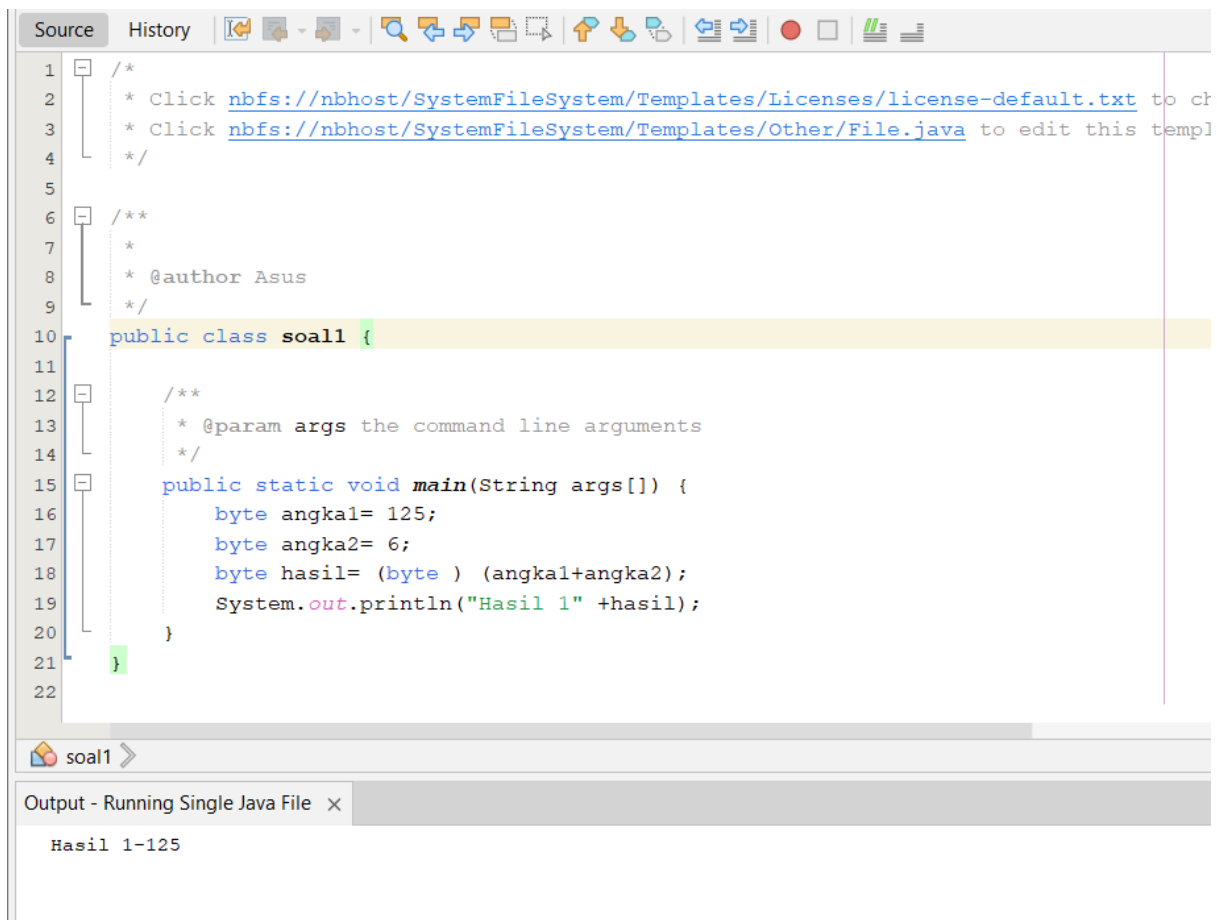
```
1 public class Welcome
2 {
3     public static void main(String[] args)
4     {
5         String greeting = "Welcome to Core Java!";
6         System.out.println(greeting);
7         for (int i = 0; i < greeting.length(); i++)
8             System.out.print("=");
9         System.out.println();
10    }
11 }
```

Below the code editor, there is a 'Welcome' tab and an 'Output - Running Single Java File' tab. The output shows the result of running the program:

```
Welcome to Core Java!
=====
```

Dalam persoalan ini saya mengalami kendala yaitu error pada proyek nya. Lalu solusi yang saya dapat adalah menghapus proyek nya lalu membuka file yang sudah saya buat dengan VScode jadi saya meng run nya yang bukan proyek tapi file single. Saya di bantu “**Muhammad Fachry Khairiansyah**”

## 4. Soal Analisis 1



```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to ch
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Other/File.java to edit this templ
4   */
5
6  /**
7   *
8   * @author Asus
9   */
10 public class soal1 {
11
12     /**
13      * @param args the command line arguments
14      */
15     public static void main(String args[]) {
16         byte angka1= 125;
17         byte angka2= 6;
18         byte hasil= (byte ) (angka1+angka2);
19         System.out.println("Hasil 1" +hasil);
20     }
21 }
22
```

soal1

Output - Running Single Java File x

Hasil 1-125

Dalam program tersebut, variabel `angka1` diset dengan nilai 125, dan `angka2` diset dengan nilai 6. Penjumlahan antara `angka1` dan `angka2` menghasilkan nilai 131, karena 125 ditambah 6 sama dengan 131. Namun, karena tipe data `byte` hanya bisa menampung nilai dalam rentang -128 hingga 127, nilai 131 berada di luar batas ini, menyebabkan terjadinya overflow. Saat overflow terjadi, Java secara otomatis melakukan wrapping nilai tersebut sesuai dengan rentang `byte`, sehingga nilai 131 dikonversi menjadi nilai yang berada dalam rentang `byte`, dalam hal ini menjadi -125. Perhitungan overflow dapat dijelaskan sebagai berikut: 131 dikurangi 128 menghasilkan 3 (karena rentang `byte` adalah -128 hingga 127), sehingga nilai 131 dikonversi menjadi -128 ditambah 3, yang sama dengan -125.

Sistem overflow ini saya ketahui dari yang di bahas di depan kelas dari seorang “Wildan Gumilang” tapi saya belum yakin lalu saya mengecek nya di chat GPT dan mendapat penjelasan yang saya pahami. Jadi penjelasan di atas adalah hasil pemahaman yang saya dapatkan dari penjelasan di depan dan Chat GPT.

## 5. Soal Analisis 2

```
10 public class soal2 {
11
12     /**
13      * @param args the command line arguments
14      */
15     public static void main(String args[]) {
16         int i = 42;
17         String s = (i < 40) ? "life" : (i > 50) ? "universe" : "everything";
18         System.out.println(s);
19     }
20 }
21
```

soal2 > main > i >

Output - Running Single Java File x

everything

Dalam soal ini saya pernah melihat yang sama di Matkul yang telah lalu yaitu membandingkan suatu nilai. Tapi saya di sini saya mencoba mencari menggunakan Chat GPT lalu saya mendapat istilah baru yaitu operator ternary.

Operator ternary memiliki format dasar:

kondisi ? nilaiJikaBenar : nilaiJikaSalah;

Program tersebut menggunakan operator ternary untuk menetapkan nilai variabel s berdasarkan kondisi nilai i. Variabel i diset dengan nilai 42. Operator ternary pertama mengecek apakah i kurang dari 40; jika ya, s diatur ke "life". Jika tidak, operator ternary kedua mengecek apakah i lebih besar dari 50; jika ya, s diatur ke "universe". Jika kedua kondisi tersebut tidak terpenuhi, seperti dalam kasus ini (karena 42 tidak kurang dari 40 dan tidak lebih dari 50), s diatur ke "everything". Jadi, program akan mencetak "everything".

Pembuktian Jika i di tetapkan menjadi kurang dari 40:

```
10 public class soal2 {
11
12     /**
13      * @param args the command line arguments
14      */
15     public static void main(String args[]) {
16         int i= 30;
17         String s = (i<40)?"life":(i>50)?"universe":"everything";
18         System.out.println(s);
19     }
20 }
21
```

soal2 > main > i >

Output - Running Single Java File x

life

Jika lebih dari 50:

```
9     */
10 public class soal2 {
11
12     /**
13      * @param args the command line arguments
14      */
15     public static void main(String args[]) {
16         int i= 60;
17         String s = (i<40)?"life":(i>50)?"universe":"everything";
18         System.out.println(s);
19     }
20 }
21
```

soal2 > main > i >

Output - Running Single Java File x

universe