

Laporan Praktikum 1

Mata Kuliah: Teknik Pemrograman



POLBAN

Oleh:

Nama : Arkan Ramadhan Nugraha

NIM : 241524033

Kelas : 1B-TI4

Daftar Isi

Laporan Praktikum 1.....	
Daftar Isi.....	i
BAB I TUJUAN.....	1
BAB II Pengerjaan Soal.....	2
2.1. Setup Software Environment.....	2
2.2. Using Notepad and Command Line Tools.....	2
2.3. Using an Integrated Development Environment.....	3
2.4. Soal Analisis 1.....	3
2.5. Soal Analisis 2.....	5
BAB III TEMUAN (LESSON LEARNED).....	7

BAB I TUJUAN

Tujuan dari latihan ini yaitu:

1. Mahasiswa dapat Mengenal mata kuliah Pemrograman Berorientasi Obyek.
2. Mahasiswa dapat mengenali dan menyebutkan Teknologi saat ini yang menggunakan penerapan Pemrograman Berorientasi Object.
3. Mahasiswa dapat melakukan setup Software environment.
4. Mahasiswa dapat menganalisis input proses dan output yang dihasilkan potongan source code kasus yang dikaitkan dengan konsep pada mata kuliah DDP.

BAB II Pengerjaan Soal

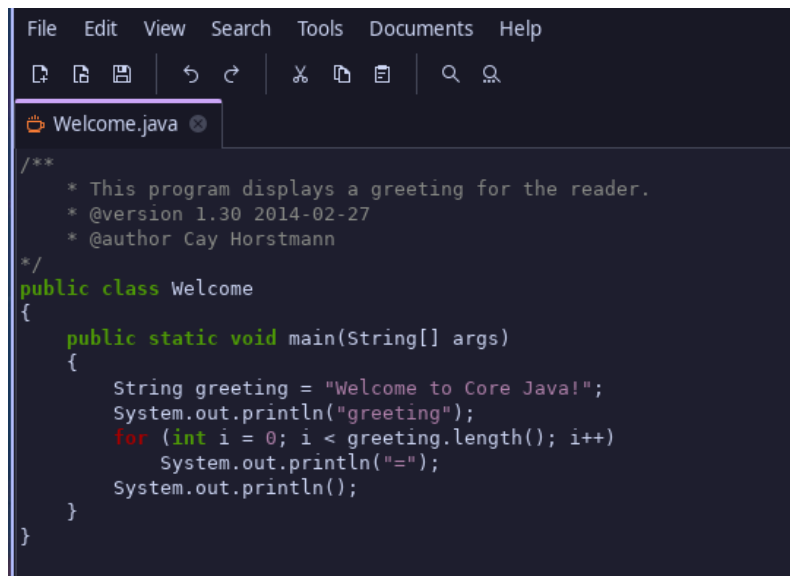
2.1. Setup Software Environment

1. Tugas
 1. Install Java Development Kit (JDK)
 2. Setting up JDK
 1. Ikuti petunjuk pada 2.1.2 Setting up JDK
 3. Test pada command prompt *Java -version*
2. Screenshot Hasil Akhir Program

```
arney1@arney1-IdeaPad-3-14ALC6:~$ java -version
openjdk version "21.0.6" 2025-01-21
OpenJDK Runtime Environment (build 21.0.6+7-Ubuntu-124.04.1)
OpenJDK 64-Bit Server VM (build 21.0.6+7-Ubuntu-124.04.1, mixed mode, sharing)
arney1@arney1-IdeaPad-3-14ALC6:~$
```

2.2. Using Notepad and Command Line Tools

1. Screenshot Hasil Akhir Program



```
File Edit View Search Tools Documents Help
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.println("=");
        System.out.println();
    }
}
```

```

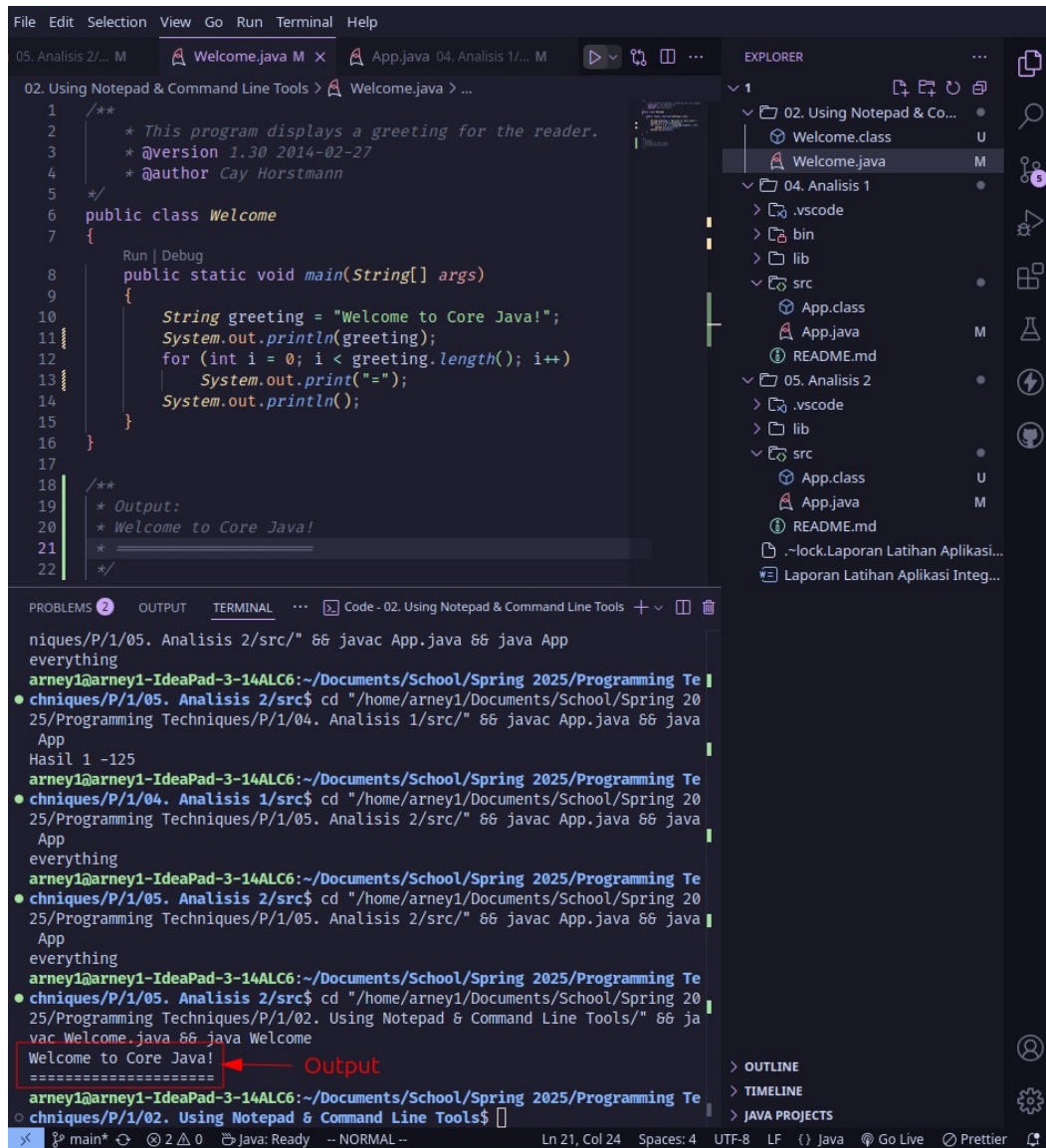
arney1@arney1-IdeaPad-3-14ALC6:~/Documents/School/Spring 2025/Programming Techniques/P/1/02. U
sing Notepad & Command Line Tools$ javac Welcome.java
arney1@arney1-IdeaPad-3-14ALC6:~/Documents/School/Spring 2025/Programming Techniques/P/1/02. U
sing Notepad & Command Line Tools$ java Welcome
Welcome to Core Java!

arney1@arney1-IdeaPad-3-14ALC6:~/Documents/School/Spring 2025/Programming Techniques/P/1/02. U
sing Notepad & Command Line Tools$ 

```

2.3. Using an Integrated Developoment Environment

1. Menggunakan VSCodeium (<https://vscodium.com/>)
2. Screenshot Hasil Akhir Program



2.4. Soal Analisis 1

1. Screenshot Hasil Akhir Program

The screenshot shows a VS Code editor with a Java file named `App.java`. The code defines a class `App` with a `main` method. It declares two `byte` variables, `angka1` (125) and `angka2` (6), and calculates their sum, storing it in `hasil`. The output is printed as `"Hasil 1 " + hasil`. Below the code, a multi-line comment explains the byte overflow: the sum 131 exceeds the upper limit of 127, causing the value to wrap around to -125. The terminal at the bottom shows the command `javac App.java` and `java App` being executed, resulting in the output `Hasil 1 -125`. The Explorer sidebar on the right shows the project structure, including folders for analysis and source files.

```
1 public class App {
2     public static void main(String[] args) {
3         byte angka1 = 125;
4         byte angka2 = 6;
5         byte hasil = (byte) (angka1 + angka2);
6
7         System.out.println("Hasil 1 " + hasil);
8     }
9 }
10
11 /*
12  * Berapa output yang keluar:
13  * 1 output saja
14  *
15  * Output:
16  * Hasil 1 -125
17  *
18  * Analisis:
19  * Byte adalah tipe data numerik yang menyimpan angka dari -128 s
20  * angka1 adalah variabel bertipe byte dengan nilai 125
21  * angka2 adalah variabel bertipe byte dengan nilai 6
22  * angka1+angka2 menghasilkan nilai 131, tapi ini melebihi batas
23  * ketika nilainya melebihi 127, nilai kembali ke batas bawah, da
24  * Jadi seperti ini
25  * 125 → 126 → 127 → -128 → -127 → -126 → -125
26  *
27  * Referensi:
28  * https://www.w3schools.com/java/java\_data\_types.asp
29  */
30
```

```
arney1@arney1-IdeaPad-3-14ALC6:~/Documents/School/Spring 2025/Programming Techniques/P/1/04. Analisis 1/src$ cd "/home/arney1/Documents/School/Spring 2025/Programming Techniques/P/1/04. Analisis 1/src/" && javac App.java && java App
Hasil 1 -125
arney1@arney1-IdeaPad-3-14ALC6:~/Documents/School/Spring 2025/Programming Techniques/P/1/04. Analisis 1/src$
```

2. Jawaban Soal

1. Berapa output yang keluar?

1. Satu output, yaitu “Hasil 1 -125”

2. Tuliskan alasan dan referensinya

1. Byte adalah tipe data numerik yang menyimpan angka dari -128 sampai 127. `angka1` adalah variabel bertipe byte dengan nilai 125. `angka2` adalah variabel bertipe byte dengan nilai 6. `angka1+angka2` menghasilkan nilai 131, tapi ini melebihi batas atas tipe data byte, sehingga terjadi overflow. ketika nilainya melebihi 127, nilai kembali ke batas bawah, dan operasi tetap berlanjut. Jadi seperti ini:

125 -> 126 -> 127 -> -128 -> -127 -> -126 -> -125

2. Referensi:

1. https://www.w3schools.com/java/java_data_types.asp

2.5. Soal Analisis 2

1. Screenshot Hasil Akhir Program

The screenshot shows a VS Code editor with a Java file named `App.java` in the `src` directory. The code defines a `main` method that uses a ternary operator to assign a value to `s` based on the value of `i`. The output of the program is displayed in the `OUTPUT` panel, showing the word `everything`. The `TERMINAL` panel shows the command `javac App.java` and the command `java App` being executed. The `EXPLORER` panel shows the project structure, including the `src` directory and the `App.class` file.

```
1 public class App {
2     public static void main(String[] args) {
3         String s = (i < 40) ? "life" : (i > 50) ? "universe" : "everything";
4         System.out.println(s);
5     }
6 }
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
```

Output:

```
* everything
*
* Teknik yang digunakan:
* Saya kurang mengerti disini "teknik" itu maksudnya apa, tapi a
* menggunakan Operasi Ternary
* Operasi ternary bekerja seperti if else
* syntax: variable = (condition) ? expressionTrue : expressionFalse
* jika condition = true, maka variable = expressionTrue
* jika condition = false, maka variable = expressionFalse
*/
```

arney1@arney1-IdeaPad-3-14ALC6:~/Documents/School/Spring 2025/Programming Techniques/P/1/04. Analisis 1/src\$ cd "/home/arney1/Documents/School/Spring 2025/Programming Techniques/P/1/05. Analisis 2/src/" && javac App.java && java App

everything

arney1@arney1-IdeaPad-3-14ALC6:~/Documents/School/Spring 2025/Programming Techniques/P/1/05. Analisis 2/src\$

2. Jawaban Soal

1. Bagaimana output setelah dijalankan

1. “everything”

2. Teknik yang digunakan

1. Algoritma ini menggunakan Operasi Ternary. Operasi ternary bekerja seperti if else. Syntax:

variable = (condition) ? expressionTrue : expressionFalse

Jika condition = true, maka variable = expressionTrue. Jika condition = false, maka variable = expressionFalse

BAB III TEMUAN (LESSON LEARNED)

1. Operasi Ternary, untuk mempersingkat algoritma if-else, tetapi menurut saya ini membuat kode susah dibaca.
2. Di Java, ketika nilai melampaui batas atas tipe data (overflow), maka nilai kembali ke batas bawah, dan operasi tetap berlanjut