

Laporan Pengerjaan Praktikum Teknik Pemrograman Pertemuan 1

Oleh:

Nama : Muhammad Nabil Syauqi Rasyiq

NIM : 241524018

Kelas : 1A



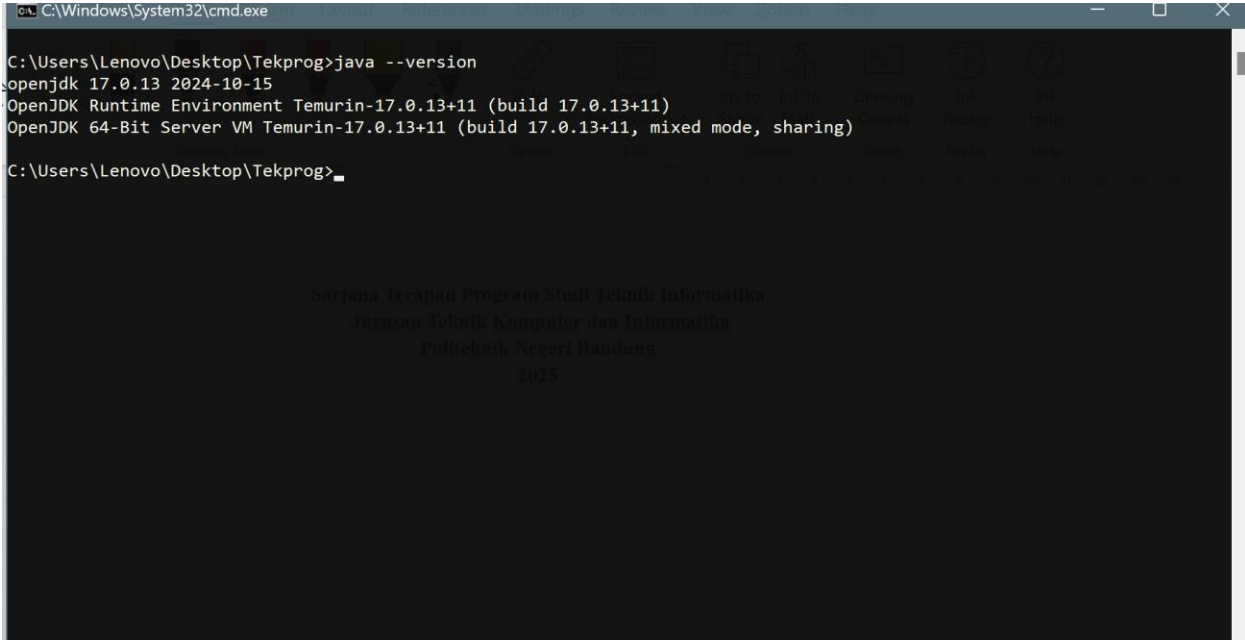
**Sarjana Terapan Program Studi Teknik Informatika
Jurusan Teknik Komputer dan Informatika
Politeknik Negeri Bandung
2025**

DAFTAR ISI

DAFTAR ISI	2
1. Set up java environment.....	3
Hasil Screenshoot Akhir	3
Kendala	3
2. Run with notepad dan CMD	3
Hasil Screenshoot.....	3
Proses pengerjaan dan kendalanya.....	5
3. Run menggunakan IntelliJ IDEA	5
Hasil Screenshoot.....	5
Proses pengerjaan.....	6
4. Soal Analisis 1.....	6
Hasil Screenshoot.....	6
Solusi soal 1	7
5. Soal Analisis 2.....	7
Solusi soal 2	7
6. Lesson learn dan Kesimpulan	8

1. Set up java environment

Hasil Screenshoot Akhir



```
C:\Windows\System32\cmd.exe
C:\Users\Lenovo\Desktop\Tekprog>java --version
openjdk 17.0.13 2024-10-15
OpenJDK Runtime Environment Temurin-17.0.13+11 (build 17.0.13+11)
OpenJDK 64-Bit Server VM Temurin-17.0.13+11 (build 17.0.13+11, mixed mode, sharing)
C:\Users\Lenovo\Desktop\Tekprog>
```

Untuk mengsetup java environment kita dapat menginstall terlebih dahulu di internet seperti pada link berikut :

<https://www.oracle.com/id/java/technologies/downloads/>

Kendala

Kendala pengerjaan set up ini adalah adanya perbedaan jenis installer, ada yang perlu di tambahkan pathnya secara manual sampa perbedaan mana yang open source dan mana yang bukan.

Langkah yang dilakukan untuk menginstall java adalah sebagai berikut:

1. Download installer java
2. Install java
3. Pastikan path bin telah ditambahkan di environment
4. Tes version java untuk memastikan bahwa JVM siap digunakan

2. Run with notepad dan CMD

Hasil Screenshoot

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenovo\Desktop\Tekprog>dir
Volume in drive C is Windows-SSD
Volume Serial Number is A8DF-9455

Directory of C:\Users\Lenovo\Desktop\Tekprog

Date modified          Type          Size
02/13/2025  10:28 AM    <DIR>         .
02/13/2025  08:21 AM    <DIR>         ..
02/13/2025  09:40 AM    <FILE> 0 Laporan Praktikum Tekprog 1.docx  0 KB
02/13/2025  09:20 AM    <DIR>         tes
02/13/2025  10:26 AM    <FILE> 256 Welcome.java  1 KB
                2 File(s)          256 bytes
                3 Dir(s)  249,808,670 bytes free

C:\Users\Lenovo\Desktop\Tekprog>javac Welcome.java

C:\Users\Lenovo\Desktop\Tekprog>dir
Volume in drive C is Windows-SSD
Volume Serial Number is A8DF-9455

Directory of C:\Users\Lenovo\Desktop\Tekprog

Date modified          Type          Size
02/13/2025  10:28 AM    <DIR>         .
02/13/2025  08:21 AM    <DIR>         ..
02/13/2025  09:40 AM    <FILE> 0 Laporan Praktikum Tekprog 1.docx  0 KB
02/13/2025  09:20 AM    <DIR>         tes
02/13/2025  10:28 AM    <FILE> 597 Welcome.class  1 KB
02/13/2025  10:26 AM    <FILE> 256 Welcome.java  1 KB
                3 File(s)          853 bytes
                3 Dir(s)  249,808,318,464 bytes free

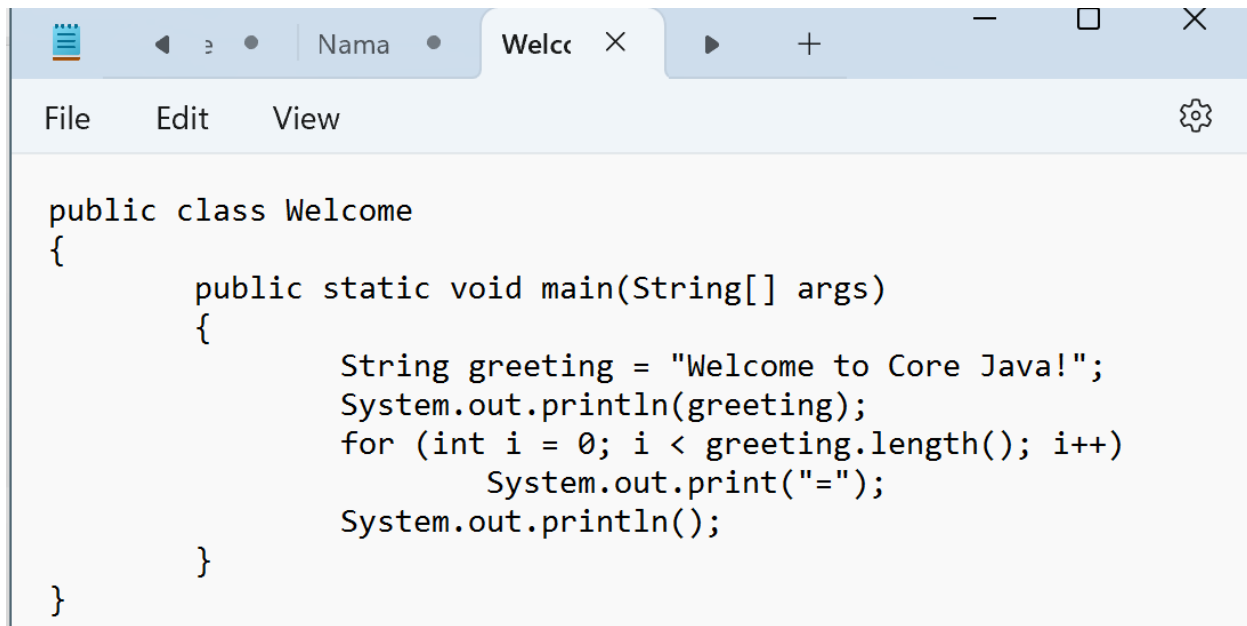
C:\Users\Lenovo\Desktop\Tekprog>java Welcome
Welcome to Core Java!
=====

C:\Users\Lenovo\Desktop\Tekprog>
```

ebra.txt | README.txt | please • | Nama • | Welcc X

File Edit View

```
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();
    }
}
```

A screenshot of a code editor window. The title bar shows a tab named 'Welcc' with a close button. The menu bar includes 'File', 'Edit', and 'View'. The code area contains a Java class named 'Welcome' with a 'main' method. The code prints 'Welcome to Core Java!' followed by a series of equals signs. The code is as follows:

```
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();
    }
}
```

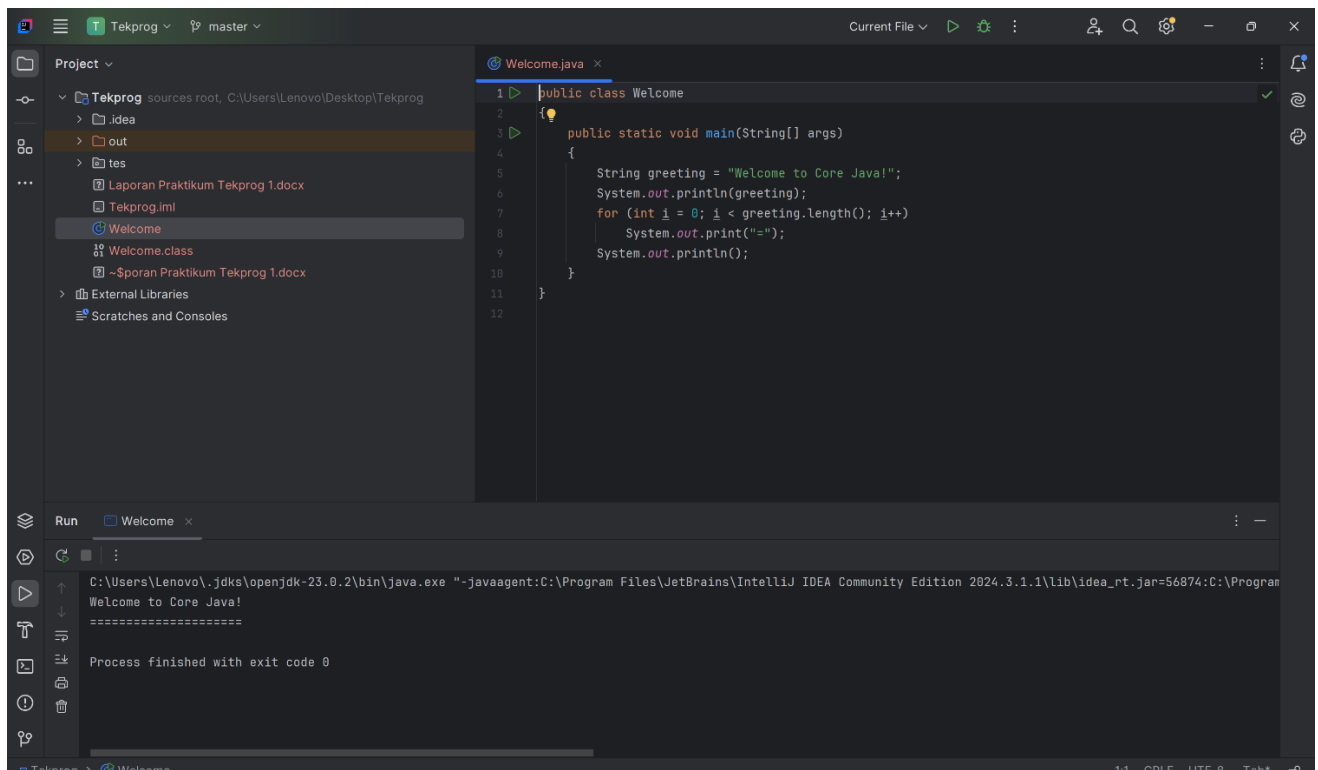
Proses pengerjaan dan kendalanya

Proses pengerjaannya dilakukan dengan terlebih dahulu membuat file dengan format java masukan sintaks sederhana dan berhasil. Ilustrasi proses mengcompile dapat di jabarkan sebagai berikut:

Dengan menggunakan sintaks “javac Welcome.java” JVM akan mengubah file Welcome.java menjadi Welcome.class. lalu dengan interpreter java kita dapat menjalankan Welcome.class sehingga hasil/ output dapat di lihat

3. Run menggunakan IntelliJ IDEA

Hasil Screenshoot



Proses pengerjaan

Proses merun program menggunakan code editor lebih midah untuk dilakukan karena kita cukup menekan tombol run dan program berhasil dijalankan. Proses menginstall Code editor dapat dilihat pada tutorial berikut : <https://youtu.be/-09UiNKwpok?si=8LhM1GEsbr6aQ8jY>

4. Soal Analisis 1

Hasil Screenshoot

```
Welcome.java  Soal1.java x
1 package tes;
2
3 public class Soal1 { new *
4     public static void main(String[] args){ new *
5         byte angka1 = 125;
6         byte angka2 = 6;
7         byte hasil = (byte) (angka1+angka2);
8
9         System.out.println("Hasil 1 "+hasil);
10    }
11 }
12
```

Solusi soal 1

Program akan memberikan output -125. Hal ini terjadi karena tipe data byte memiliki nilai minimum -128 dan nilai maksimum 127. Sehingga ketika 125 ditambah 6 dia akan kereset ulang kenilai minimum yaitu -125. Berikut ilustrasinya:

125 → 126 → 127 → -128 → -127 → -126 → -125

bisa dilihat pada ilustrasi diatas ketika 125 ditambah dengan 6 maka ia akan sampai di -125

5. Soal Analisis 2

```
Welcome.java  Soal1.java  Soal2.java x
1 package tes;
2
3 public class Soal2 { new *
4     public static void main(String[] args){ new *
5         int i = 42;
6         String s = (i<40)?"life":(i>50)?"universe":"everything";
7         System.out.println(s);
8     }
9 }
10
```

Solusi soal 2

Program akan mengeluarkan output “everything” perhatikan ilustrasi berikut;

(condition)? True : false

Diatas kita dapat melihat pada kondisi pertama yaitu ($i < 40$) itu bernilai false maka program tidak mengambil “Life” tetapi lanjut ke condition kedua ($i > 50$). Ternyata pada kondisi kedua juga itu bernilai false sehingga ia akan mengambil output “everything”

6. Lesson learn dan Kesimpulan

Pada praktikum ini saya mempelajari bagaimana suatu compiler java berjalan. Saya sadar saya harus lebih berhati-hati lagi karena java sangat sensitif pada kesalahan penggunaan huruf. Proses praktikum ini juga saya berdiskusi dengan Naufal Khairian dan Praditama Ajmal. Hasil praktikum kali ini saya abadikan di github saya.

Link github : <https://github.com/Nabilpnyk/Tegprog-perytemuan-1>