

LAPORAN PRAKTIKUM

PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN

“PENERAPAN KONSEP OOP (*OBJECT-ORIENTED PROGRAMMING*)

DALAM BAHASA PEMROGRAMAN JAVA”

disusun oleh

Hafizh Habibullah

2511531002

Dosen Pengampu: Dr. Wahyudi, S.T., M.T.

Asisten Praktikum: Muhammad Zaki Al Hafiz



DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

TAHUN 2025

## **KATA PENGANTAR**

Puji syukur kita panjatkan ke hadirat Allah SWT. karena atas rahmat dan karunia-Nya laporan praktikum Algoritma dan Pemrograman ini dapat diselesaikan dengan baik.

Laporan ini disusun untuk memenuhi tugas praktikum pada mata kuliah Algoritma dan Pemrograman dengan topik Penerapan Konsep OOP (*Object-Oriented Programming*) Dalam Bahasa Pemrograman Java. Melalui laporan ini, penulis berusaha menjelaskan hasil praktikum yang meliputi kode program, penjelasan tentang cara kerja program yang dibuat, serta hasil output dari program.

Penulis sangat mengharapkan kritik dan saran yang membangun agar dapat menjadi bahan perbaikan di kemudian hari. Semoga laporan ini dapat memberikan manfaat bagi penulis maupun pembaca.

Padang, 14 November 2025

Hafizh Habibullah

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>i</b>
<b>DAFTAR ISI.....</b>	<b>ii</b>
<b>DAFTAR LAMPIRAN .....</b>	<b>iii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1    Latar Belakang.....	1
1.2    Tujuan.....	1
1.3    Manfaat.....	1
<b>BAB 2 PEMBAHASAN.....</b>	<b>2</b>
2.1    Menentukan Bilangan Prima sesuai Input pengguna.....	2
2.2    Penerapan enkapsulasi pada Konsep OOP ( <i>Object-Oriented Programming</i> ) ...	4
2.3    Kode Program PanggilMahasiswa_2511531002.....	5
2.4    Kode Program PanggilMahasiswa2_2511531002.....	6
2.5    Manipulasi Dasar String dalam Java .....	7
2.6    Penggabungan String serta Perbedaan Operasi String dan Integer.....	8
<b>BAB III KESIMPULAN.....</b>	<b>10</b>
3.1    Kesimpulan.....	10
3.2    Saran Pengembangan.....	10
<b>DAFTAR PUSTAKA .....</b>	<b>11</b>

## DAFTAR LAMPIRAN

Gambar 2. 1 - Kode Program BilanganPrima_2511531002 .....	2
Gambar 2. 2 - Kode Program Mahasiswa_2511531002 .....	4
Gambar 2. 3 - Kode Program PanggilMahasiswa_2511531002 .....	5
Gambar 2. 4 - Kode Program PanggilMahasiswa2_2511531002 .....	6
Gambar 2. 5 - Kode Program String1_2511531002 .....	7
Gambar 2. 6 - Kode Program String2_2511531002 .....	8

Tabel 2. 1 - Hasil Output dari BilanganPrima_2511531002 .....	3
Tabel 2. 2 - Hasil Output dari PanggilMahasiswa_2511531002 .....	6
Tabel 2. 3 - Hasil Output dari PanggilMahasiswa2_2511531002 .....	7
Tabel 2. 4 - Hasil Output dari String1_2511531002 .....	8
Tabel 2. 5 - Hasil Output dari String2_2511531002 .....	9

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

*Object-Oriented Programming (OOP)* adalah salah satu paradigma pemrograman yang sering digunakan untuk memudahkan pengembangan dalam membuat aplikasi yang efisien dan terstruktur. Dengan konsep ini, pengembangan program dapat dipecah menjadi objek-objek yang saling berinteraksi, sehingga kode menjadi lebih mudah dipahami, dikembangkan, dan dipelihara.

Selain konsep OOP, manipulasi *String* juga menjadi aspek penting dalam pemrograman. Hampir setiap aplikasi memerlukan pengelolaan data berbentuk teks, mulai dari input pengguna hingga proses pengolahan data. Dengan memahami cara kerja dan metodenya, program dapat dibuat menjadi lebih interaktif dan fungsional.

### **1.2 Tujuan**

Adapun tujuan dari praktikum ini adalah:

1. Mengetahui struktur dasar program Java berbasis objek.
2. Memahami cara kerja *class* dan *object* dalam penerapan OOP.
3. Menggunakan berbagai metode dalam kelas *String* untuk melakukan manipulasi teks.
4. Mengembangkan logika pemrograman yang lebih terstruktur dan efisien.

### **1.3 Manfaat**

Manfaat dari Praktikum ini adalah:

1. Menambah pemahaman tentang konsep dasar pemrograman berbasis objek.
2. Melatih kemampuan dalam menulis dan menganalisis kode Java secara sistematis.
3. Memperluas pengetahuan tentang fungsi dan metode manipulasi *String* yang sering digunakan dalam pemrograman.
4. Membiasakan diri dengan praktik penulisan kode yang baik dan terstruktur.

## BAB 2

### PEMBAHASAN

#### 2.1 Menentukan Bilangan Prima sesuai Input pengguna

```
1 package pekan7_2511531002;
2
3 import java.util.Scanner;
4
5 public class BilanganPrima_2511531002 {
6     public static boolean isPrime(int n) {
7         int factors = 0;
8         for (int i = 1; i <= n; i++) {
9             if (n % i == 0) {
10                 factors++;
11             }
12         }
13         return (factors == 2);
14     }
15
16     public static void main(String[] args) {
17         Scanner input = new Scanner (System.in);
18         System.out.print("Input nilai n = ");
19         int a = input.nextInt();
20         input.close();
21         if (isPrime(a)) System.out.print(a + " bilangan prima");
22         else System.out.println(a + " bukan bilangan prima");
23     }
24 }
```

Gambar 2. 1 - Kode Program BilanganPrima\_2511531002

Dalam kode program ini, dibuat kode program simpel yang menentukan apakah bilangan yang dimasukkan oleh pengguna merupakan bilangan Prima atau tidak. Program dimulai dengan penggunaan *method* **boolean**. *Method* ini nantinya akan mengembalikan nilai logika **true** atau **false**. Input dari pengguna disini disimpan di variabel **n**.

Pada *method* **boolean**, digunakan perulangan *for-loop* yang akan menjalankan kode dari variabel **i = 1**, dan perulangan akan berjalan selama **i** lebih kecil daripada **n**. Diakhir perulangan, nilai **i** akan bertambah sampai kondisi sebelumnya tidak lagi sesuai. Program akan memeriksa apakah **n** habis dibagi oleh **i** dengan kondisi **if (n % i == 0)**. Jika iya, maka variabel **factors** akan bertambah satu. Setelah perulangan selesai, hasilnya akan digunakan untuk menentukan apakah bilangan tersebut merupakan bilangan prima (memiliki tepat dua faktor).

Setelah proses pengecekan selesai pada *method isPrime(int n)*, hasil pengujian tersebut akan digunakan di dalam *method main*. *Method main* berfungsi sebagai titik awal program yang akan dijalankan. Program meminta pengguna untuk memasukkan nilai **n** dengan menggunakan *utility Scanner*, kemudian menyimpannya ke variabel **a**.

Selanjutnya, nilai **a** dikirim sebagai argumen ke *method isPrime(a)* untuk diperiksa apakah bilangan tersebut prima atau tidak. Jika *method* mengembalikan nilai **true**, program akan menampilkan pesan bahwa bilangan tersebut adalah bilangan prima. Sebaliknya, jika nilai yang dikembalikan adalah **false**, maka program akan menampilkan pesan bahwa bilangan tersebut bukan bilangan prima.

Contoh Hasil Output
Input nilai n = 5 5 bilangan prima
Input nilai n = 9 9 bukan bilangan prima

Tabel 2. 1 - Hasil Output dari BilanganPrima\_2511531002

## 2.2 Penerapan enkapsulasi pada Konsep OOP (*Object-Oriented Programming*)

```
1 package pekan7_2511531002;
2
3 public class Mahasiswa_2511531002 {
4     //variabel global
5     private int nim;
6     private String nama, nim2;
7     //membuat mutator (setter)
8     public void setNim(int nim) {
9         this.nim=nim;
10    }
11    public void setNim2(String nim2) {
12        this.nim2=nim2;
13    }
14    public void setName(String nama) {
15        this.nama=nama;
16    }
17
18    public static void main(String[] args) {
19    }
20
21    public int getNim() {
22        return nim;
23    }
24    public String getNim2() {
25        return nim2;
26    }
27    public String getName() {
28        return nama;
29    }
30
31    //metode lain
32    public void Cetak() {
33        System.out.println("Nim : " + nim);
34        System.out.println("Nama : " + nama);
35    }
36    public void Cetak2() {
37        System.out.println("Nim : " + nim2);
38        System.out.println("Nama : " + nama);
39    }
40 }
```

Gambar 2. 2 - Kode Program Mahasiswa\_2511531002

Program diatas merupakan contoh penerapan **enkapsulasi** dalam konsep **OOP** (*Object-Oriented Programming*) menggunakan Bahasa Pemrograman Java. Terdapat beberapa variabel dengan akses **private**, yaitu **nim**, **nama**, dan **nim2**. Variabel ini disembunyikan dari luar kelas supaya tidak bisa diakses langsung, dan



hanya bisa diubah melalui *method setter* (`setNim`, `setNim2`, `setNama`) serta dibaca menggunakan *method getter* (`getNim`, `getNim2`, `getNama`).

Bagian **public static void main(String[] args)** berfungsi sebagai titik awal eksekusi program, meskipun pada kode ini belum terdapat isi di dalamnya. Biasanya objek dari kelas **Mahasiswa** akan dibuat di dalam **main** untuk menguji metode-metode yang sebelumnya sudah ditulis.

Selain itu, terdapat dua metode lain bernama **Cetak()** dan **Cetak2()** yang digunakan untuk menampilkan data mahasiswa ke layar. *Method Cetak()* menampilkan nilai dari **nim** dan **nama**, sedangkan *Method Cetak2()* menampilkan nilai dari **nim2** dan **nama**. Perbedaan ini menunjukkan bahwa program memisahkan dua data identitas mahasiswa dengan variabel yang berbeda, yang di mana **nim** menggunakan variabel **int**, dan **nim2** menggunakan variabel **String**.

### 2.3 Kode Program PanggilMahasiswa\_2511531002

```
1 package pekan7_2511531002;
2
3 public class PanggilMahasiswa_2511531002 {
4     public static void main(String[] args) {
5         Mahasiswa_2511531002 a = new Mahasiswa_2511531002();
6         a.setNim(23532);
7         a.setNama("Hafizh");
8         System.out.println(a.getNim());
9         System.out.println(a.getNama());
10        a.Cetak();
11    }
12 }
```

Gambar 2. 3 - Kode Program PanggilMahasiswa\_2511531002

Kode diatas merupakan lanjutan dari *class* sebelumnya, yaitu **Mahasiswa\_2511531002**. Kode Program ini berfungsi untuk menguji class tersebut. Di dalam *method main*, objek **a** dari kelas **Mahasiswa\_2511531002** dibuat dengan perintah **Mahasiswa\_2511531002 a = new Mahasiswa\_2511531002();**. Setelah objek dibuat, program mengisi data dengan memanggil metode *setter* **setNim(23532)** untuk mengisi NIM dan **setNama("Hafizh")** untuk mengisi nama.

Nilai yang sudah disimpan ditampilkan kembali ke layar dengan metode *getter* `getNim()` dan `getNama()`. Baris `a.Cetak()` di akhir akan memanggil metode `cetak()` dari *class* sebelumnya untuk menampilkan data Mahasiswa secara lengkap.

Hasil Output
23532
Hafizh
Nim : 23532
Nama : Hafizh

Tabel 2. 2 - Hasil Output dari PanggilMahasiswa\_2511531002

#### 2.4 Kode Program PanggilMahasiswa2\_2511531002

```

1 package pekan7_2511531002;
2
3 import java.util.Scanner;
4
5 public class PanggilMahasiswa2_2511531002 {
6     public static void main(String[] args) {
7         Scanner input = new Scanner(System.in);
8         System.out.print("NIM: ");
9         String x = input.nextLine();
10        System.out.print("Nama: ");
11        String y = input.nextLine();
12        Mahasiswa_2511531002 a = new Mahasiswa_2511531002();
13        a.setNim2(x);
14        a.setNama(y);
15        if (x.startsWith("25")) {
16            System.out.println(y + " anda angkatan 2025");
17        }
18        if (x.contains("1153")) {
19            System.out.println("Anda Mahasiswa Informatika");
20        }
21        a.Cetak2();
22        input.close();
23    }
24 }

```

Gambar 2. 4 - Kode Program PanggilMahasiswa2\_2511531002

Program di atas hampir sama fungsinya seperti Program sebelumnya, yaitu menampilkan hasil data Mahasiswa ke layar. Hanya saja, data untuk Program ini berasal dari Input pengguna, yaitu **NIM** yang disimpan dengan **String x** dan **Nama** yang disimpan dengan **String y**. Setelah itu, program memanggil *setter* `setNim2` untuk mengisi NIM dan `setNama`.

Selanjutnya, program melakukan dua pengecekan, yaitu pengecekan angkatan pengguna dan program studinya yang diperiksa dari NIM yang dimasukkannya. Jika NIM diawali dengan “25”, maka program akan menampilkan pesan bahwa pengguna adalah angkatan 25. Selanjutnya, jika NIM memiliki “1153” didalamnya, maka program akan menampilkan pesan bahwa pengguna adalah mahasiswa Informatika.

Akhirnya, program memanggil metode **Cetak2** dengan menggunakan **a.Cetak2()**, yang dimana jika **Cetak2** dipanggil, maka program akan menampilkan hasil akhir yang menggunakan data dari **nim2** dan **nama**.

Hasil Output
<p>NIM: 2511531002</p> <p>Nama: Hafizh Habibullah</p> <p>Hafizh Habibullah anda angkatan 2025</p> <p>Anda Mahasiswa Informatika</p> <p>Nim : 2511531002</p> <p>Nama : Hafizh Habibullah</p>

Tabel 2. 3 - Hasil Output dari PanggilMahasiswa2\_2511531002

## 2.5 Manipulasi Dasar String dalam Java

```

1 package pekan7_2511531002;
2
3 public class String1_2511531002 {
4     public static void main(String[] args) {
5         String salam = "Assalamualaikum";
6         System.out.println("panjang salam adalah: " + salam.length());
7         System.out.println(salam.toUpperCase()); // Outputs "ASSALAMUALAIKUM"
8         System.out.println(salam.toLowerCase()); // Outputs "assalamualaikum"
9         System.out.println(salam.indexOf("salam")); // Outputs 2
10    }
11 }

```

Gambar 2. 5 - Kode Program String1\_2511531002

Program ini mendemonstrasikan penerapan Manipulasi String dalam Bahasa Java. Manipulasi String dilakukan untuk mengolah, memeriksa, memodifikasi data teks agar bisa digunakan sesuai kebutuhan program. Pada Program, digunakan empat jenis *method* bawaan, yaitu:

1. `length()` → untuk mengukur panjang teks (misalnya untuk validasi input).
2. `toUpperCase()` → untuk merubah teks menjadi huruf besar.

3. `toLowerCase()` → untuk merubah teks menjadi huruf kecil,
4. `indexOf()` → untuk mencari posisi suatu kata/karakter dalam teks.

Salah satu contoh penggunaan **`indexOf()`** yang bisa dilihat dari Kode Program diatas adalah Program diminta untuk mencari posisi kata “salam” pada “Assalamualaikum”. Hasilnya, program akan menampilkan output “2”.

Hasil Output
panjang salam adalah: 15
ASSALAMUALAIKUM
assalamualaikum
2

Tabel 2. 4 - Hasil Output dari String1\_2511531002

## 2.6 Penggabungan String serta Perbedaan Operasi String dan Integer

```

1 package pekan7_2511531002;
2
3 public class String2_2511531002 {
4     public static void main(String[] args) {
5         String firstName = "Syifa";
6         String lastName = "Muhassanah";
7         String txt1 = "Dosen \"intelektual\" kampus";
8         System.out.println("Nama Lengkap: "+firstName + " " +lastName);
9         System.out.println("Nama Lengkap: "+firstName.concat(lastName));
10        System.out.println(txt1);
11        int x = 10;
12        int y = 20;
13        int z = x + y;
14        System.out.println("x + y = " +z);
15        String a = "10";
16        String b = "20";
17        String c = a + b;
18        System.out.println("String a + String b = " +c);
19        String v = a + y;
20        System.out.println("String a + integer y = " +v);
21    }
22 }

```

Gambar 2. 6 - Kode Program String2\_2511531002

Program ini adalah perkembangan dari Program sebelumnya, yaitu **String1\_2511531002**. Pada program ini, dijelaskan bagaimana Java melakukan **penggabungan string**, penggunaan *escaped characters*, serta perbedaan antara operasi angka dan operasi String.

Program mendefinisikan dua variabel String, yaitu **firstName** dan **lastName**, lalu menggabungkannya dengan dua cara:

1. Menggunakan operator “+”.
2. Menggunakan *method* **.concat()**, yang gunanya adalah menggabungkan dua string menjadi satu string baru.

Keduanya menghasilkan teks satu kesatuan yaitu **Nama Lengkap**.

Selain itu, terdapat contoh penggunaan *escaped character* (“\”) pada variabel **txt1** untuk menampilkan tanda kutip di dalam String, yang biasanya dianggap sebagai **pembatas teks**.

Program ini juga menunjukkan perbedaan antara **penjumlahan integer** dan **penggabungan string**:

- a. Ketika **x** dan **y** berupa angka, **x + y** akan menghasilkan hasil perhitungan aritmatika
- b. Namun ketika nilai seperti “**10**” dan “**20**” diperlukan sebagai string, operator **+** tidak menjumlahkan, tetapi **menggabungkan teks**, sehingga hasilnya menjadi “**1020**”.
- c. Contoh **a + y** menegaskan bahwa apabila salah satu operand adalah String, Java otomatis menganggap operasinya sebagai penggabungan String, bukan perhitungan.

Bagian ini pada intinya memperlihatkan bagaimana Java membedakan operasi berdasarkan tipe data, dan bagaimana String bisa diproses melalui penggabungan string dan *escaped characters*.

Hasil Output
Nama Lengkap: Syifa Muhassanah
Nama Lengkap: SyifaMuhassanah
Dosen “intelektual” kampus
<b>x + y = 30</b>
<b>String a + String b = 1020</b>
<b>String a + integer y = 1020</b>

Tabel 2. 5 - Hasil Output dari String2\_2511531002

## **BAB III**

### **KESIMPULAN**

#### **3.1 Kesimpulan**

Dari hasil praktikum, dapat disimpulkan bahwa program-program yang dibuat mencakup dasar pemrograman Java mulai dari logika sederhana, konsep OOP, hingga manipulasi String. Penjelasan di atas menunjukkan bagaimana Java memisahkan logika ke dalam *method*, mengamankan data melalui enkapsulasi, serta memproses teks menggunakan berbagai fungsi bawaan. Dengan memahami bagian-bagian dasar ini, proses pembuatan program menjadi lebih terstruktur, rapi, dan mudah dikembangkan.

#### **3.2 Saran Pengembangan**

Untuk pengembangan selanjutnya, pemahaman dasar ini dapat dilanjutkan ke penerapan fitur yang lebih kompleks seperti pewarisan, *polymorphism*, validasi input yang lebih ketat, atau manipulasi string lanjutan. Pengembangan ini akan membantu meningkatkan kualitas program sekaligus memperluas kemampuan dalam menerapkan konsep OOP dan pengolahan data di Java.

## DAFTAR PUSTAKA

- [1] Oracle, “The Java™ Tutorials,” Oracle Docs, 2024. Accessed: Oct. 20, 2024. [Online]. Available: <https://docs.oracle.com/javase/>
- [2] Oracle, “String (Java Platform SE Documentation),” Oracle Docs, 2024. [Online]. Available: <https://docs.oracle.com/javase/8/docs/api/java/lang/String.html>
- [3] Oracle, “Object-Oriented Programming Concepts,” Oracle Docs, 2024. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/concepts/>