

**LAPORAN PRATIKUM**  
**PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN**  
**“BILANGAN PRIMA, METHOD DAN MANIPULASI STRING”**

**disusun Oleh:**

**AFIF RAHMAN SALEH**

**NIM 251153105**

**Dosen Pengampu: Dr. WAHYUDI, S.T, M.T**

**Asisten Pratikum: AUFAN TAUFIQURRAHAMAN**



**DEPARTEMEN INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

**TAHUN**

**2025**

## DAFTAR ISI

DAFTAR ISI .....	i
KATA PENGANTAR .....	ii
BAB I .....	1
PENDAHULUAN .....	1
1.2 Tujuan Praktikum .....	1
1.3 Manfaat Praktikum .....	2
BAB II .....	3
PEMBAHASAN .....	3
2.1 Program 1: BilanganPrima_2511531005.java .....	3
2.2 Program 2 & 3: Mahasiswa_2511531005.java dan PanggilMahasiswa_2511531005.java .....	4
2.3 Program 4: PangggilMahasiswa2_2511531005.java .....	6
2.4 Program 5: String1_2511531005.java .....	7
2.5 Program 6: String2_2511531005.java .....	9
BAB III .....	11
KESIMPULAN .....	11
3.1 Kesimpulan .....	11
DAFTAR PUSTAKA .....	12

## **KATA PENGANTAR**

Puji syukur kami panjatkan atas kehadiran Tuhan Yang Maha Esa atas rahmat dan karunia-Nya laporan praktikum ini dapat diselesaikan dengan baik. Laporan ini disusun sebagai bentuk penugasan dalam kegiatan praktikum mata kuliah Algoritma dan Pemograman khususnya pada materi “ Bilangan Prima, Method, dan Manipulasi String “.

Dalam penyusunan laporan ini, penulis berusaha untuk menjelaskan secara sistematis mulai dari latar belakang, tujuan, dan manfaat praktikum, hingga pembahasan kode program, analisis hasil, serta kesimpulan dan saran untuk pengembangan lebih lanjut. Materi yang dibahas diharapkan dapat menjadi referensi bagi mahasiswa atau praktikan dalam memahami dan mengimplementasikan algoritma perulangan secara efektif.

Padang, 2025

Penulis

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Java merupakan salah satu bahasa pemrograman berorientasi objek yang banyak digunakan dalam pengembangan perangkat lunak. Dalam pembelajaran ini sangat penting untuk memahami konsep dasar seperti penggunaan metode (method), manipulasi string, dan logika pemrograman seperti penentuan bilangan prima. Melalui praktikum pekan 7 ini, mahasiswa dapat mempraktikkan dan menganalisis konsep-konsep tersebut secara langsung dengan membuat program sederhana yang telah dipraktikkan pada hari senin kemarin.

Pemahaman terhadap bilangan prima sangat berguna dalam bidang kriptografi dan keamanan data. Selain itu, manipulasi string dan penggunaan metode adalah dasar dari pembuatan program yang terstruktur. Oleh karena itu, praktikum ini menjadi landasan penting dalam membangun kemampuan pemrograman berorientasi objek.

### **1.2 Tujuan Praktikum**

Praktikum ini bertujuan untuk :

1. Mampu menganalisis dan memahami logika untuk menentukan bilangan prima.
2. Memahami cara membuat dan menggunakan metode dalam Java.
3. Memahami konsep dasar manipulasi string di Java, seperti panjang string, perubahan huruf besar/kecil, pencarian indeks, dan penggabungan string.
4. Menerapkan konsep enkapsulasi dalam pembuatan kelas dan objek sederhana.

5. Menggunakan metode bawaan dari kelas String dan Scanner untuk interaksi dengan pengguna.

### **1.3 Manfaat Praktikum**

Praktikum ini bermanfaat untuk :

1. Melatih kemampuan dalam merancang logika program yang efisien .
2. Meningkatkan pemahaman terhadap struktur program berbasis objek di Java.
3. Memberikan pengalaman praktis dalam pengolahan data berupa angka dan teks.
4. Menjadi fondasi untuk pengembangan aplikasi yang lebih kompleks untuk kedepannya.

## BAB II

### PEMBAHASAN

#### 2.1 Program 1: BilanganPrima\_2511531005.java

```
package pekan7_2511531005;

import java.util.Scanner;

public class BilanganPrima_2511531005 {
    public static boolean isPrime (int n) {
        int factor = 0;
        for (int i=1 ; i<= n ; i++) {
            if (n % i == 0) {
                factor++;
            }
        }
        return (factor == 2);
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("input nilai n = ");
        int a = input.nextInt();
        if (isPrime(a)) {
            System.out.println(a+ " bilangan prima ");
        }else {
            System.out.println(a+ " bukan bilangan prima ");
        }
    }
}
```

##### Kode Program 2.1

Menentukan apakah suatu bilangan bulat yang dimasukkan pengguna merupakan bilangan prima atau bukan.

Langkah langkah :

1. Masukkan input n berupa bilangan bulat melalui Scanner.
2. Panggil metode isPrime(n).
3. Di dalam isPrime(), variabel factor diinisialisasi dengan nilai 0.
4. Loop for dari 1 hingga n:
  - Jika  $n \% i == 0$ , maka i adalah faktor  $\rightarrow$  factor++.
5. Setelah loop selesai, jika factor == 2, kembalikan true (prima); jika tidak, kembalikan false.

Output :

```
input nilai n = 37
37 bilangan prima
```

```
input nilai n = 48
48 bukan bilangan prima
```

## Output 2.1

Analisis hasil :

Menurut definisi matematika, bilangan prima hanya memiliki dua faktor positif: 1 dan dirinya sendiri (Hardy & Wright, 1979). Logika program ini benar secara konseptual. Namun, pendekatannya kurang efisien karena mengecek semua angka dari 1 hingga  $n$ . Secara teori, cukup mengecek hingga  $\sqrt{n}$  karena faktor selalu berpasangan. Misalnya, jika  $n = 25$ , cukup cek hingga 5. jika tidak ada faktor hingga  $\sqrt{25}$ , maka  $n$  pasti prima.

## 2.2 Program 2 & 3: Mahasiswa\_2511531005.java dan PanggilMahasiswa\_2511531005.java

```
package pekan7_2511531005;

public class Mahasiswa_2511531005 {
    private int nim;
    private String nama,nim2;

    • public void setNim (int nim) {
        this.nim=nim;
    }
    • public void setNim2 (String nim2) {
        this.nim2=nim2;
    }
    • public void setName (String nama) {
        this.nama=nama;
    }
    • public int getNim() {
        return nim;
    }
    • public String getNim2() {
        return nim2;
    }
    • public String getName() {
        return nama;
    }

    • public void Cetak1 () {
        System.out.println("nim : "+ nim);
        System.out.println("nama : "+ nama);
    }
    • public void Cetak2 () {
        System.out.println("nim : "+ nim2);
        System.out.println("nama : "+ nama);
    }
}
```

Kode Program 2.2

```
package pekan7_2511531005;

public class PanggilMahasiswa_2511531005 {

    • public static void main(String[] args) {
        Mahasiswa_2511531005 a = new Mahasiswa_2511531005();
        a.setNim (23532);
        a.setName("rahmat");
        System.out.println(a.getNim());
        System.out.println(a.getName());
        a.Cetak();
    }
}
```

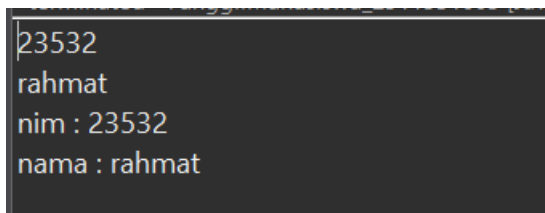
Kode Program 2.3

Menerapkan konsep enkapsulasi dengan membuat kelas Mahasiswa yang menyimpan data NIM dan nama, lalu mengaksesnya melalui objek.

Langkah langkah :

1. Buat kelas Mahasiswa dengan mendefinisikan atribut privat: nim (int), nama, dan nim2 (String).
2. Buat method setter untuk mengisi data dan getter untuk mengambil data.
3. Buat objek a di program PanggilMahasiswa dari kelas Mahasiswa.
4. Isi data dengan a.setNim(23532) dan a.setNama("rahmat").
5. Tampilkan data melalui getter dan method Cetak().

Output :



```
23532
rahmat
nim : 23532
nama : rahmat
```

Output 2.3

Analisis hasil :

Program ini menerapkan prinsip enkapsulasi dalam OOP yang dimana data disembunyikan (private) dan hanya dapat diakses melalui metode publik (Sommerville, 2016). Ini mencegah modifikasi data secara langsung dari luar kelas, sehingga menjaga integritas objek. Penggunaan dua tipe NIM (int dan String) menunjukkan fleksibilitas dalam representasi data.



### 2.3 Program 4: PanggilMahasiswa2\_2511531005.java

```
package pekan7_2511531005;

import java.util.Scanner;

public class PanggilMahasiswa2_2511531005 {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("nim : ");
        String x = input.nextLine();
        System.out.println("nama :");
        String y = input.nextLine();
        Mahasiswa_2511531005 a = new Mahasiswa_2511531005();
        a.setNim2(x);
        a.setNama(y);
        if (x.startsWith("25")) {
            System.out.println(y+ " anda angkatan 2025");
        }
        if (x.contains("1153")) {
            System.out.println("anda mahasiswa informatika");
        }
        a.Cetak2();
        input.close();
    }
}
```

#### Kode Program 2.4

Memproses input NIM sebagai string dan menganalisisnya menggunakan metode manipulasi string untuk menentukan angkatan dan program studi.

Langkah langkah :

1. Input NIM dan nama sebagai string.
2. Baut objek Mahasiswa, dan set data menggunakan setNim2() dan setNama().
3. Cek program:
  - Jika NIM dimulai dengan "25" → cetak "anda angkatan 2025".
  - Jika NIM mengandung "1153" → cetak "anda mahasiswa informatika".
4. Tampilkan data dengan Cetak2().

Output :

```
nim : 2511531005
nama :
afif rahman
afif rahman anda angkatan 2025
anda mahasiswa informatika
nim : 2511531005
nama : afif rahman
```

#### Output 2.4

Analisis hasil :

Program ini memanfaatkan dua metode penting dari kelas String:

1. `startsWith(String prefix)`: mengembalikan true jika string dimulai dengan teks tertentu.
2. `contains(CharSequence s)`: mengembalikan true jika substring ada di dalam string.

Menurut dokumentasi resmi Java (Oracle, 2025), kedua metode ini case-sensitive dan sangat berguna dalam validasi format data. Penggunaan NIM sebagai String memungkinkan analisis pola teks, yang tidak mungkin jika NIM berupa int (karena angka tidak memiliki metode string).

#### 2.4 Program 5: String1\_2511531005.java

```
package pekan7_2511531005;

public class String1_2511531005 {

    public static void main(String[] args) {
        String salam = "assalamualaikum";
        System.out.println("panjang salam adalah : " + salam.length());
        System.out.println(salam.toUpperCase());
        System.out.println(salam.toLowerCase());
        System.out.println(salam.indexOf("salam"));
    }
}
```

#### Kode Program 2.5

Mengenalkan metode dasar manipulasi string di Java.

Langkah langkah :

1. Deklarasikan string `salam = "assalamualaikum"`.
2. Tampilkan :
  - Panjang string (`length()` → 15)
  - Versi huruf kapital (`toUpperCase()` → "ASSALAMUALAIKUM")
  - Versi huruf kecil (`toLowerCase()` → "assalamualaikum")
  - Indeks kemunculan substring "salam" (`indexOf()` → 3)

Output :

```
panjang salam adalah : 15
ASSALAMUALAIKUM
assalamualaikum
2
```

Output 2.5

Analisis hasil :

1. `length()`: menghitung jumlah karakter.
2. `toUpperCase()/toLowerCase()`: berguna untuk normalisasi input pengguna.
3. `indexOf()`: mengembalikan indeks pertama kemunculan substring. Jika tidak ditemukan, nilai -1 dikembalikan.

Dalam Java, string bersifat immutable, artinya metode ini tidak mengubah objek asli, melainkan mengembalikan objek baru (Eckel, 2006). Ini penting untuk memahami efisiensi memori dalam aplikasi besar.

## 2.5 Program 6: String2\_2511531005.java

```
package pekan7_2511531005;

import java.util.Scanner;

public class String2_2511531005 {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("nama depan : ");
        String firstName = input.nextLine();
        System.out.println("nama belakang : ");
        String lastName = input.nextLine();
        String txt1 = "desen \" intelektual \" kampus";
        System.out.println("nama lengkap : " + firstName + " " + lastName);
        System.out.println("nama lengkap : " + firstName.concat(lastName));
        System.out.println(txt1);
        int x = 10;
        int y = 20;
        int z = x + y;
        System.out.println("x+y= " + z);
        String a = "10";
        String b = "20";
        String c = a + b;
        System.out.println("String a + String b = " + c);
        String v = a + y;
        System.out.println("String a + integer y = " + v);
    }
}
```

### Kode Program 2.6

Menunjukkan perbedaan antara operasi aritmatika dan konkatenasi string, serta penggunaan escape character.

Langkah langkah :

1. Input nama depan dan belakang.
2. Tampilkan nama lengkap dengan operator + (otomatis tambah spasi).
3. Tampilkan hasil concat() (tanpa spasi otomatis).
4. Tunjukkan string dengan tanda kutip menggunakan \".
5. Bandingkan :
  - Penjumlahan integer:  $10 + 20 = 30$
  - Konkatenasi string:  $"10" + "20" = "1020"$
  - Gabungan string dan integer:  $"10" + 20 = "1020"$

Output :

```
nama depan :  
afif  
nama belakang:  
rahman  
nama lengkap : afif rahman  
nama lengkap : afifrahman  
desen " intelektual " kampus  
x+y= 30  
String a + String b = 1020  
String a + integer y = 1020
```

## Output 2.6

### Analisis hasil :

Java menerapkan string conversion saat operator + digunakan dengan satu operand berupa string. Semua operand lain akan diubah menjadi string. Ini menjelaskan mengapa "10" + 20 menghasilkan "1020", bukan 30. Metode concat() hanya menerima satu string dan tidak menambahkan spasi. Berbeda dengan operator + yang lebih fleksibel. Penggunaan \" menunjukkan escape sequence, yaitu cara menyisipkan karakter khusus dalam literal string.

## **BAB III**

### **KESIMPULAN**

#### **3.1 Kesimpulan**

Berdasarkan pelaksanaan praktikum yang telah dilakukan, dapat disimpulkan bahwa penggunaan metode, enkapsulasi, dan manipulasi string dalam Java telah berhasil dipahami dan diterapkan melalui serangkaian program sederhana. Program untuk mengecek bilangan prima menunjukkan bagaimana logika dapat dikemas dalam sebuah metode untuk meningkatkan keterbacaan kode. Penerapan enkapsulasi pada kelas Mahasiswa membuktikan prinsip dasar pemrograman berorientasi objek, di mana data privat hanya dapat diakses melalui antarmuka yang terkontrol, sehingga menjaga integritas informasi. Selain itu, berbagai operasi string seperti pengubahan huruf, pencarian substring, pengecekan awalan, serta perbedaan antara operasi aritmatika dan konkatenasi memberikan pemahaman mendalam tentang karakteristik dan perilaku kelas String dalam Java. Secara keseluruhan, praktikum ini telah memenuhi tujuannya dalam memperkenalkan konsep konsep fundamental pemrograman Java yang menjadi fondasi penting bagi pengembangan aplikasi yang lebih kompleks.

## DAFTAR PUSTAKA

- [1] Oracle Corporation, Java™ Platform, Standard Edition Documentation. 2025. [Online]. Available: <https://docs.oracle.com/en/java/javase/>
- [2] B. Eckel, Thinking in Java, 4th ed. Upper Saddle River, NJ, USA: Prentice Hall, 2006.
- [3] I. Sommerville, Software Engineering, 10th ed. Harlow, United Kingdom: Pearson Education, 2016.
- [4] G. H. Hardy and E. M. Wright, An Introduction to the Theory of Numbers, 5th ed. Oxford, UK: Oxford University Press, 1979.
- [5] Oracle Corporation, The Java® Language Specification, Java SE 17 Edition. 2023. [Online]. Available: <https://docs.oracle.com/javase/specs/>