

**LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN**

**“STRING PADA JAVA”**

**DISUSUN OLEH:**

**ALIYATAR RAFI AHMAD**

**2511533031**

**DOSEN PENGAMPU:**

**Dr. WAHYUDI, S.T, M.T**

**ASISTEN PRAKTIKUM:**

**JOVANTRI IMMANUEL GULO**



**DEPARTEMEN INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS ANDALAS**

**2025**

## KATA PENGANTAR

Puji syukur Saya panjatkan ke hadirat Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya, Saya dapat menyelesaikan laporan praktikum pada mata kuliah Algoritma Pemrograman dengan judul **“String pada Bahasa Pemrograman Java”** dengan baik dan tepat waktu.

Laporan ini dibuat sebagai salah satu bentuk pelaksanaan praktikum pada mata kuliah Algoritma Pemrograman. Tujuan dari laporan ini adalah untuk memahami konsep dasar *String* dalam bahasa pemrograman Java, meliputi cara deklarasi, penggabungan, perbandingan, serta penggunaan berbagai method yang berkaitan dengan pengolahan teks.

Saya berharap laporan ini dapat memberikan manfaat bagi pembaca, khususnya mahasiswa yang sedang mempelajari dasar-dasar algoritma dan pemrograman Java. Saya juga menyadari bahwa laporan ini masih jauh dari sempurna, sehingga kritik dan saran yang membangun sangat diharapkan demi penyempurnaan di masa mendatang.

Akhir kata, penulis mengucapkan terima kasih kepada dosen pengampu mata kuliah Algoritma Pemrograman, asisten labor, serta semua pihak yang telah membantu dalam penyusunan laporan ini.

Padang, 11 November 2025

Aliyatar Rafi Ahmad

## DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI .....	ii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Tujuan .....	1
1.3 Manfaat .....	2
BAB II PEMBAHASAN .....	3
2.1 Pengertian String .....	3
2.2 Method-Method Pada String .....	3
2.3 Kode Program .....	4
2.3.1 Bilangan Prima_2511533031 .....	4
2.3.2 Mahasiswa_2511533031 .....	5
2.3.3 Panggil Mahasiswa_2511533031 .....	7
2.3.4 Panggil Mahasiswa2_2511533031 .....	8
2.3.5 String1_2511533031 .....	9
2.3.6 String2_2511533031 .....	11
BAB III PENUTUP .....	13
3.1 Kesimpulan .....	13
3.2 Saran .....	13
DAFTAR PUSTAKA .....	14

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Dalam pemrograman Java, *String* merupakan tipe data non-primitif yang sangat penting digunakan untuk mengolah data dalam bentuk teks, seperti nama, input pengguna, pesan, maupun analisis karakter. *String* memiliki banyak kegunaan, mulai dari program sederhana hingga aplikasi besar. Java menyediakan banyak method bawaan seperti *length()*, *toUpperCase()*, *indexOf()*, *startsWith()*, dan *contains()* yang memudahkan manipulasi dan pemeriksaan isi teks.

Selain itu, praktikum ini juga memberikan pemahaman mengenai bagaimana data dari pengguna diproses dan ditampilkan kembali. Melalui contoh kode program yang dibuat, mahasiswa belajar bagaimana cara memvalidasi data tertentu, misalnya mengecek apakah NIM diawali angka tertentu atau apakah sebuah teks mengandung potongan karakter tertentu. Pemahaman seperti ini sangat berguna dalam pembuatan aplikasi yang membutuhkan identifikasi data, validasi input, dan pengolahan informasi.

Dengan memahami dasar-dasar *String*, mahasiswa dapat lebih mudah membangun program yang membutuhkan pengolahan teks, seperti menampilkan informasi, memproses input pengguna, maupun mengenali pola tertentu pada data.

### 1.2 Tujuan

1. Memahami konsep dasar *String* pada Java.
2. Mengenal method-method penting pada kelas *String*, seperti *length()*, *toUpperCase()*, *toLowerCase()*, dan *indexOf()*.
3. Melatih kemampuan dalam menerapkan method *String* di dalam program, agar bisa digunakan untuk kebutuhan manipulasi teks di berbagai kasus.
4. Mengetahui cara mengolah data mahasiswa menggunakan kombinasi *String* dan OOP.

### 1.3 Manfaat

1. Mahasiswa memahami penggunaan dasar tipe data String dalam Java.
2. Mahasiswa mampu menggunakan berbagai method String seperti *length()*, *toUpperCase()*, *toLowerCase()*, *indexOf()*, dan *concat()*.
3. Mahasiswa terlatih membangun logika pemrograman dasar melalui studi kasus sederhana.

## BAB II

### PEMBAHASAN

#### 2.1 Pengertian String

Dalam bahasa pemrograman Java, *String* adalah tipe data yang digunakan untuk menyimpan teks. String bukan tipe primitif seperti *int* atau *double*, melainkan merupakan objek dari kelas *String*. Karena sering digunakan dalam pemrograman, Java menyediakan berbagai metode untuk memanipulasi string dengan mudah dan efisien.

*String* di Java adalah immutable, artinya setelah dibuat, nilai string tidak bisa diubah. Jika kita mengubah string, sebenarnya Java membuat objek string baru.

Berikut contoh pendeklarasian variabel dengan tipe data String:

```
class BelajarJava {  
    public static void main(String args[]){  
  
        String var1 = "Belajar Java";  
        String var2 = " di Unand";  
  
        System.out.print(var1);  
        System.out.print(var2);  
  
    }  
}
```

Output : Belajar Java di Unand

#### 2.2 Method-Method Pada String

1. Length()

Untuk menghitung jumlah karakter dalam sebuah String.

2. toUpperCase()

Method ini akan mengubah semua isi string yang non kapital akan menjadi kapital. Jika string asal sudah uppercase, maka string yang dikembalikan sama dengan string asal.

### 3. toLowerCase()

Method toLowerCase merupakan kebalikan dari method toUpperCase(), method ini akan mengembalikan string yang identik dengan string asal namun dalam bentuk non kapital.

### 4. indexOf()

Mencari posisi awal dari sebuah kata atau karakter dalam String.

### 5. Concat()

Menggabungkan dua String menjadi satu.

### 6. startsWith() dan contains()

Method digunakan untuk menguji apakah string diawali dengan spesifik karakter tertentu.

## 2.3 Kode Program

Pada praktikum pekan 7, mahasiswa telah membuat program mengenai *String* dalam bahasa java. Berikut kode program yang telah dibuat dalam praktikum pekan 7.

### 2.3.1 Bilangan Prima\_2511533031

```

1 package pekan7_2511533031;
2
3 import java.util.Scanner;
4
5 public class BilanganPrima_2511533031 {
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     public static void main(String[] args) {
16         Scanner input = new Scanner(System.in);
17         System.out.print("Input nilai n = ");
18         int a = input.nextInt();
19         if (isPrime(a)) {
20             System.out.println(a + " Bilangan prima");
21         } else {
22             System.out.println(a + " bukan bilangan prima");
23         }
24     }
25 }

```

Gambar 1.1

Program ini dibuat untuk mengecek apakah sebuah angka merupakan bilangan prima atau tidak. Program ini bekerja dengan membaca input angka dari pengguna, lalu memeriksa jumlah faktor pembagiannya melalui method *isPrime()*. Jika angka tersebut hanya

memiliki dua faktor, yaitu 1 dan dirinya sendiri, maka program menyatakan bahwa angka tersebut adalah bilangan prima. Jika tidak, program menampilkan bahwa angka tersebut bukan bilangan prima. Program ini menunjukkan penggunaan perulangan, operator modulus, dan pemanggilan method dalam Java.

Langkah Pembuatan :

1. Import Scanner untuk membaca input.
2. Buat method isPrime() untuk mengecek jumlah faktor dari angka.
3. Di dalam isPrime, lakukan perulangan dari 1 sampai n dan hitung banyaknya pembagi.
4. Kembalikan true jika faktor berjumlah 2, selain itu false.
5. Di main, buat objek Scanner dan minta input angka dari user.
6. Panggil isPrime(a) untuk mengecek apakah angka prima.
7. Tampilkan hasil menggunakan percabangan if-else.

### 2.3.2 Mahasiswa\_2511533031

```

1 package pekan7_2511533031;
2
3 public class Mahasiswa_2511533031 {
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    //membuat accesor (getter)
18    public int getNim() {
19        return nim;
20    }
21    public String getNim2() {
22        return nim2;
23    }
24    public String getName() {
25        return nama;
26    }
27    // metode lain
28    public void Cetak() {
29        System.out.println("Nim : "+nim);
30        System.out.println("Nim : "+nama);
31    }
32    public void Cetak2() {
33        System.out.println("Nim : "+nim2);
34        System.out.println("Nim : "+nama);}
35
36 }

```

Gambar 1.2



Program diatas berfungsi sebagai *template* atau cetakan untuk menyimpan informasi penting tentang seorang mahasiswa, yaitu NIM dan nama. Program ini menyediakan beberapa metode yang memungkinkan pengguna untuk mengisi data mahasiswa melalui metode setter (setNim, setNim2, dan setName), kemudian mengambil kembali datanya melalui metode *getter* (getNim, getNim2, dan getName).

Selain itu, program ini juga memiliki metode tambahan, yaitu Cetak() dan Cetak2(), yang digunakan untuk menampilkan data mahasiswa ke layar. Menariknya, program ini juga memperlihatkan dua cara berbeda dalam menyimpan NIM, yaitu sebagai tipe data *integer* (melalui nim) dan sebagai string (melalui nim2). Perbedaan ini penting karena beberapa format NIM sering kali membutuhkan tipe *string*, misalnya jika memuat angka depan tertentu atau panjang digit yang tidak boleh berubah.

Secara keseluruhan, program ini membantu memahami bagaimana menyimpan data dalam variabel kelas, bagaimana cara memanipulasi data tersebut melalui metode, dan bagaimana menampilkannya dalam bentuk output.

Langkah Pembuatan :

1. Mendeklarasikan variabel global berupa nim (tipe int), nim2 (tipe String), dan nama (String) untuk menyimpan NIM dan nama mahasiswa.
2. Membuat metode setter (setNim, setNim2, dan setName) untuk mengisi data ke dalam variabel-variabel tersebut.
3. Membuat metode getter (getNim, getNim2, dan getName) yang berfungsi mengembalikan dan menampilkan nilai dari variabel yang sudah diisi.
4. Membuat metode Cetak() untuk menampilkan NIM (tipe int) dan nama mahasiswa.
5. Membuat metode Cetak2() untuk menampilkan NIM versi String (nim2) bersama nama mahasiswa.

6. Menghubungkan class ini dengan program lain (misalnya PanggilMahasiswa) agar data dapat diinput dari luar dan dicetak melalui metode yang telah dibuat.

### 2.3.3 Panggil Mahasiswa\_2511533031

```

1 package pekan7_2511533031;
2
3 public class PanggilMahasiswa_2511533031 {
4     public static void main(String[] args) {
5         Mahasiswa_2511533031 a= new Mahasiswa_2511533031();
6         a.setNim(251153);
7         a.setNama("Atar");
8         System.out.println(a.getNim());
9         System.out.println(a.getNama());
10        a.Cetak();
11    }
12 }
13

```

Gambar 1.3

Program Ini adalah program sederhana yang digunakan untuk memanggil class Mahasiswa\_2511533031 diatas. Program ini membuat sebuah objek mahasiswa, mengisi data berupa NIM dan nama menggunakan metode setter, lalu menampilkan kembali data tersebut menggunakan getter dan metode Cetak().

Program ini menunjukkan cara mengakses class lain, memanggil metode di dalamnya, serta bagaimana data yang sudah disimpan bisa ditampilkan ke layar.

Langkah Pembuatan :

1. Import atau pastikan class Mahasiswa\_2511533031 berada dalam package yang sama agar dapat digunakan.
2. Membuat class baru bernama PanggilMahasiswa\_2511533031 sebagai program utama.
3. Membuat method main() sebagai titik awal menjalankan program.
4. Membuat objek baru dari class Mahasiswa\_2511533031 dengan perintah:  
'Mahasiswa\_2511533031 a = new Mahasiswa\_2511533031();'
5. Mengisi data objek dengan memanggil metode setter, seperti:
  - o a.setNim(251153);
  - o a.setNama("Atar");

6. Menampilkan data mahasiswa menggunakan metode getter:
  - a.getNim()
  - a.getNama()
7. Memanggil metode Cetak() untuk menampilkan seluruh data mahasiswa dalam format yang sudah disediakan di class utama.

### 2.3.4 Panggil Mahasiswa2\_2511533031

```

1 package pekan7_2511533031;
2
3 import java.util.Scanner;
4
5 public class PanggilMahasiswa2_2511533031 {
6
7     public static void main(String[] args) {
8         Scanner input= new Scanner(System.in);
9         System.out.print("NIM: ");
10        String x= input.nextLine();
11        System.out.print("Nama: ");
12        String y= input.nextLine();
13        Mahasiswa_2511533031 a= new Mahasiswa_2511533031();
14        a.setNim2(x);
15        a.setNama(y);
16        if(x.startsWith("25")) {
17            System.out.println(y + " anda angkatan 2025");
18        }
19        if(x.contains("1153")) {
20            System.out.println("Anda Mahasiswa Informatika");
21        }
22        a.Cetak2();
23        input.close();
24    }
25 }
26
27 }

```

Gambar 1.4

Program PanggilMahasiswa2\_2511533031 adalah program yang digunakan untuk menginput data mahasiswa langsung dari keyboard. Pengguna diminta memasukkan NIM dan nama, kemudian data tersebut disimpan dalam objek Mahasiswa\_2511533031 melalui setter versi string (setNim2).

Setelah data dimasukkan, program melakukan pengecekan pola pada NIM menggunakan method string:

- startsWith("25") → untuk menentukan apakah mahasiswa merupakan **angkatan 2025**.
- contains("1153") → untuk mengecek apakah mahasiswa berasal dari **prodi Informatika**.

Terakhir, program menampilkan kembali data mahasiswa menggunakan metode Cetak2().

Langkah Pembuatan :

1. Mengimpor library Scanner untuk membaca input dari keyboard.
2. Membuat class PanggilMahasiswa2\_2511533031 sebagai program utama.
3. Membuat method main() sebagai titik masuk program.
4. Membuat objek Scanner untuk menerima input NIM dan nama.
5. Meminta pengguna memasukkan NIM menggunakan `input.nextLine()`.
6. Meminta pengguna memasukkan nama dengan cara yang sama.
7. Membuat objek baru dari class Mahasiswa\_2511533031 untuk menyimpan data mahasiswa.
8. Menyimpan NIM dan nama ke dalam objek menggunakan `setNim2(x)` dan `setNama(y)`.
9. Melakukan pengecekan NIM:
  - Jika NIM diawali "25", tampilkan bahwa mahasiswa adalah angkatan 2025.
  - Jika NIM mengandung "1153", tampilkan bahwa mahasiswa adalah mahasiswa Informatika.
10. Memanggil metode Cetak2() untuk menampilkan data mahasiswa berdasarkan format penyimpanan NIM versi string.
11. Menutup Scanner menggunakan `input.close()` untuk menghindari kebocoran resource.

### 2.3.5 String1\_2511533031

```

1 package pekan7_2511533031;
2
3 public class String1_2511533031 {
4
5     public static void main(String[] args) {
6         String salam = "Assalamualaikum";
7         System.out.println("panjang salam adalah : "+salam.length());
8         System.out.println(salam.toUpperCase()); // Outputs "Assalamualaikum"
9         System.out.println(salam.toLowerCase()); // Outputs "assalamualaikum"
10        System.out.println(salam.indexOf("salam")); // Outputs 2
11    }
12 }

```

Gambar 1.5

Program `String1_2511533031` adalah program sederhana yang digunakan untuk mengenalkan **fungsi-fungsi dasar pada String di Java**. Program ini menggunakan sebuah variabel `String` bernama `salam`, kemudian menampilkan berbagai informasi menggunakan fungsi bawaan `String`, seperti:

- menghitung **panjang String** (`length()`),
- mengubah huruf menjadi **huruf besar** (`toUpperCase()`),
- mengubah huruf menjadi **huruf kecil** (`toLowerCase()`),
- mencari posisi suatu kata di dalam `String` (`indexOf("salam")`).

Program ini membantu memahami bagaimana `String` dapat dimanipulasi dan dianalisis dalam Java.

Langkah Pembuatan :

1. Membuat class baru bernama `String1_2511533031`.
2. Membuat method `main()` sebagai bagian utama yang akan dijalankan.
3. Mendeklarasikan variabel *String* bernama `salam` dan mengisinya dengan teks "Assalamualaikum".
4. Menampilkan panjang *String* menggunakan `salam.length()`.
5. Menampilkan `String` dalam huruf kapital menggunakan `salam.toUpperCase()`.
6. Menampilkan `String` dalam huruf kecil menggunakan `salam.toLowerCase()`.
7. Mencari posisi kata "salam" dalam `String` menggunakan `salam.indexOf("salam")`.
8. Menampilkan output ke layar menggunakan `System.out.println`.

### 2.3.6 String2\_2511533031

```

1 package pekan7_2511533031;
2
3 import java.util.Scanner;
4
5 public class String2_2511533031 {
6
7     public static void main(String[] args) {
8         Scanner input = new Scanner(System.in);
9         System.out.print("Nama Depan: ");
10        String firstName= input.nextLine();
11        System.out.print("Nama Belakang: ");
12        String lastName= input.nextLine();
13        String txt1 = "Dosen\\intelektual\\kampus";
14        System.out.println("Nama Lengkap: "+firstName + " " + lastName);
15        System.out.println("Nama Lengkap: "+firstName.concat(lastName));
16        System.out.println(txt1);
17        int x = 10;
18        int y = 20;
19        int z = x + y;
20        System.out.println("x + y = "+z);
21        String a = "10";
22        String b = "20";
23        String c = a + b;
24        System.out.println("String a + string b =" +c);
25        String v = a + y;
26        System.out.println("String a + integer y =" +v);
27    }
28
29 }

```

Gambar 1.6

Program String2\_2511533031 adalah program yang digunakan untuk mengolah string dan menunjukkan perbedaan antara **operasi angka** dan **operasi string** di Java. Program ini meminta input pengguna berupa **nama depan** dan **nama belakang**, kemudian menggabungkannya menggunakan dua cara: operator + dan method concat().

Program ini juga menampilkan contoh **escape character** dengan string txt1, di mana tanda kutip ganda harus ditulis menggunakan \".

Selain itu, program ini menunjukkan perbedaan antara **penjumlahan angka** dan **penggabungan string**:

- $x + y$  menghasilkan nilai numerik (penjumlahan)
- "10" + "20" menghasilkan "1020" (penggabungan string)
- "10" + 20 tetap menghasilkan string "1020" karena angka otomatis dikonversi menjadi string

Program ini memberikan pemahaman dasar tentang cara kerja String, input, penggabungan teks, dan perbedaan tipe data dalam Java.

Langkah Pembuatan :

1. Membuat class String2\_2511533031 dan method main().
2. Membuat objek Scanner untuk membaca input nama depan dan belakang.
3. Menyimpan input pengguna ke variabel firstName dan lastName.
4. Menggabungkan nama menggunakan operator + dan method .concat().
5. Membuat string yang berisi tanda kutip menggunakan escape character (\").
6. Membuat variabel angka dan melakukan operasi penjumlahan (x + y).
7. Membuat variabel string angka lalu menggabungkannya (a + b) dan contoh gabungan string dengan integer (a + y).
8. Menampilkan semua hasil ke layar dengan System.out.println().

## **BAB III**

### **PENUTUP**

#### **3.1 Kesimpulan**

Berdasarkan seluruh praktikum yang telah dilakukan, dapat disimpulkan bahwa penggunaan String pada Java memiliki peran penting dalam proses pengolahan teks dan data input pengguna. Melalui beberapa program yang dibuat, mahasiswa dapat memahami berbagai operasi dasar pada String seperti menghitung panjang teks, mengubah huruf besar/kecil, mencari posisi kata, menggabungkan String, hingga penggunaan escape character.

Selain itu, praktikum ini juga memberikan pemahaman tentang perbedaan operasi angka dan operasi String, serta bagaimana Java memperlakukan string ketika digabungkan dengan tipe data lain. Praktikum mengenai class Mahasiswa\_2511533031 juga memberikan gambaran awal tentang penyimpanan data dalam objek, penggunaan setter dan getter, serta pemanggilan class dari program lain.

Secara keseluruhan, praktikum ini membantu memperkuat pemahaman dasar pemrograman Java, khususnya dalam pengolahan teks dan penggunaan variabel, input, kondisi, serta struktur program sederhana.

#### **3.2 Saran**

Agar praktikum ini dapat berjalan lebih baik, disarankan untuk meningkatkan penjelasan materi di awal sesi, memberikan contoh kasus yang lebih beragam, serta memastikan setiap mahasiswa mendapat waktu yang cukup untuk mencoba dan memahami kode secara mandiri. Selain itu, penggunaan waktu praktikum bisa lebih terstruktur agar semua langkah dapat diselesaikan tanpa tergesa-gesa.



## DAFTAR PUSTAKA

- [1] S. Saori, "Java String dan Operasinya: Panduan Lengkap dengan Contoh," SUFYAN97, 16 Maret 2025. [Online]. Available: <https://www.sufyan97.com/2025/03/java-string-dan-operasinya-panduan.html>. [Accessed 14 November 2025].
- [2] A. Imamudin, "Konstruktor dan Method dalam String," Blogger, 8 Mei 2013. [Online]. Available: <https://agungimam.blogspot.com/2013/05/konstruktor-dan-method-dalam-string.html>. [Accessed 14 November 2025].

