

**LAPORAN PRAKTIKUM
AGORITMA PEMROGRAMAN
“STRING PADA JAVA”**

disusun Oleh:

SOFIAN ARBA’I

2511533029

Dosen Pengampu:

Dr. WAHYUDI, S.T, M.T

Asisten Praktikum:

JOVANTRI IMMANUEL GULO



**DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
2025**

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya, laporan praktikum Algoritma dan Pemrograman pekan ke-7 dengan judul *String pada Java* ini dapat diselesaikan dengan baik dan tepat waktu. Laporan ini disusun sebagai bentuk dokumentasi dan refleksi dari kegiatan praktikum yang bertujuan untuk memahami konsep dasar dan penerapan manipulasi string dalam bahasa pemrograman Java. Materi string merupakan bagian penting dalam pemrograman karena banyak digunakan dalam pengolahan data teks, validasi input, dan pengembangan aplikasi yang interaktif.

Dalam laporan ini, penulis membahas berbagai metode dan fungsi yang tersedia dalam kelas String di Java, seperti length (), charAt (), substring (), equals (), dan toUpperCase (), serta penerapannya dalam studi kasus sederhana. Penulis berharap laporan ini dapat memberikan gambaran yang jelas mengenai pengolahan string dalam Java dan menjadi referensi yang bermanfaat bagi pembaca.

Penulis menyadari bahwa laporan ini masih memiliki kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Akhir kata, penulis mengucapkan terima kasih kepada dosen pembimbing, asisten praktikum, serta semua pihak yang telah membantu dalam penyusunan laporan ini.

Padang, 11 November 2025

Sofian Arba'i

DAFTAR ISI

KATA PENGANTAR.....	i
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 Fungsi String	3
2.3 Aplikasi Penggunaan String.....	4
2.3.1 Kode program Penggunaan String.....	4
2.3.2 Kode program Penggunaan String.....	6
2.3.3 Kode program Penggunaan String.....	8
2.3.4 Kode program Penggunaan String.....	10
2.3.5 Kode program Penggunaan String.....	11
BAB III PENUTUP	14
3.1 Kesimpulan	14
3.2 Saran	14
DAFTAR PUSTAKA.....	15

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam bahasa pemrograman Java, string merupakan salah satu tipe data yang paling sering digunakan karena berfungsi untuk menyimpan dan memproses data dalam bentuk teks. Hampir setiap aplikasi Java, mulai dari program sederhana hingga sistem yang kompleks, memerlukan pengolahan string untuk menampilkan pesan, membaca input pengguna, atau memanipulasi data teks. Oleh karena itu, pemahaman yang baik mengenai konsep dan karakteristik string menjadi hal yang sangat penting bagi seorang programmer.

Selain memahami konsep dasarnya, mahasiswa juga perlu menguasai berbagai *method* bawaan dari kelas String yang disediakan oleh Java, seperti `length()`, `charAt()`, `substring()`, dan `equals()`. Melalui praktikum ini, mahasiswa diharapkan dapat memahami cara menggunakan method-method tersebut untuk melakukan operasi pada string secara efektif, serta mampu menerapkannya dalam pembuatan program yang melibatkan manipulasi teks secara dinamis dan efisien. Oleh karena itu, praktikum ini dilakukan untuk memperdalam pemahaman mahasiswa terhadap konsep dan implementasi string dalam bahasa pemrograman Java.

1.2 Tujuan

Tujuan yang diharapkan setelah praktikum sebagai berikut:

1. Mengetahui pengertian dan karakteristik tipe data String dalam Java.
2. Mempelajari berbagai *method* bawaan kelas String seperti `length()`, `charAt()`, `substring()`, `toUpperCase()`, dan `equals()`.
3. Melatih kemampuan dalam mengolah dan memanipulasi data teks menggunakan operasi string.
4. Meningkatkan keterampilan pemrograman dalam penerapan konsep string pada program Java sederhana.

1.3 Manfaat

Manfaat yang didapatkan setelah praktikum sebagai berikut:

1. Mampu memahami konsep dasar dan cara kerja string dalam Java
2. Dapat mengimplementasikan berbagai operasi string untuk menyelesaikan permasalahan yang berkaitan dengan teks.
3. Meningkatkan kemampuan logika dan efisiensi penulisan kode dalam pemrograman Java.
4. Menjadi dasar dalam memahami pemrosesan data teks yang lebih kompleks, seperti manipulasi file atau data input-output pada aplikasi Java.

BAB II

PEMBAHASAN

2.1 Pengertian String

Dalam bahasa pemrograman, string dapat didefinisikan sebagai kumpulan karakter yang membentuk urutan tertentu. Karakter dalam string bisa berupa huruf, angka, simbol, dan spasi. Dalam hampir semua bahasa pemrograman, string diapit oleh tanda kutip (tanda kutip tunggal atau ganda) untuk menandai awal dan akhir string. Pemrograman yang sering digunakan dalam string adalah untuk merepresentasikan teks, pesan, alamat, atau data teks lainnya. Misalnya, dalam development aplikasi web, string digunakan untuk menyimpan teks pada halaman web, menerima input dari pengguna, atau menampilkan pesan kepada pengguna. Ada beberapa jenis string yang umum digunakan dalam pemrograman. Dua jenis yang umum string adalah:

1. String Literal

String Literal adalah string yang ditulis secara langsung di dalam kode program. Contohnya: “Halo, dunia!”. String literal biasanya digunakan untuk menginisialisasi variabel atau memberikan nilai langsung kepada sebuah string.

2. String Objek

Beberapa bahasa pemrograman mendukung objek string yang memiliki metode dan fungsi bawaan. Objek string memungkinkan operasi seperti penggabungan, pemotongan, penggantian, dan lainnya. Misalnya, dalam bahasa Java, kegunaan objek String adalah untuk melakukan manipulasi pada string.

2.2 Fungsi String

Fungsi string adalah beragam fungsi yang memungkinkan pemrogram untuk melakukan manipulasi, pengolahan, dan pemeriksaan terhadap string. Fungsi-fungsi tersebut dapat digunakan untuk memanipulasi struktur, mengubah format, mencari, dan melakukan berbagai operasi pada string.

Berikut ini adalah beberapa fungsi string umum yang digunakan dalam pemrograman:

1. Concat, Untuk menggabungkan beberapa kata atau Kalimat pada Java, disebut dengan concatenation.
2. Substring, Substring merupakan fungsi yang digunakan untuk mengambil beberapa karakter dari sebuah Kalimat atau kata. Fungsi ini memiliki 2 parameter, yaitu index awalan dan juga index akhir. Indexnya diambil sesuai dengan urutan karakter yang ada di Kalimat tersebut.
3. Contains, Untuk mengecek Apakah sebuah Kalimat mengandung karakter yang kita tentukan atau tidak, kita bisa memakan fungsi contains. Fungsi ini akan mengembalikan atau menghasilkan boolean true jika Kalimat tersebut mengandung karakter yang kita inputkan. Sebaliknya, fungsi contains akan menghasilkan boolean false jika Kalimat tersebut tidak mengandung Kalimat yang kita inputkan.
4. ToLowerCase, ToLowerCase adalah fungsi yang digunakan untuk merubah Kalimat atau karakter menjadi huruf kecil.
5. ToUpperCase, Kebalikan dari fungsi ToLowerCase, fungsi ToUpperCase adalah fungsi yang digunakan untuk merubah Kalimat atau karakter menjadi huruf kapital.
6. Length, Length adalah fungsi yang digunakan untuk menghitung berapa banyak jumlah karakter yang ada di dalam sebuah String.

2.3 Aplikasi Penggunaan String

Pada pertemuan praktikum pekan 7 mahasiswa diinstruksikan untuk menyalin kode program tentang penggunaan string pada java yang telah dibuat oleh dosen. Kode program tersebut diharapkan sebagai latihan dan contoh untuk kode program string. Berikut adalah kode program yang diberikan oleh dosen untuk praktikum pekan 7.

2.3.1 Kode program Penggunaan String

Kode program untuk penggunaan string yang pertama adalah untuk menentukan apakah suatu bilangan termasuk bilangan prima atau bukan. Berikut adalah kode programnya.

```

1 package pekan7_2511533029;
2 import java.util.Scanner;
3 public class BilanganPrima_2511533029 {
4     public static Boolean isPrime(int n) {
5         int factors =0;
6         for (int i=1; i<=n; i++) {
7             if (n % i ==0) {
8                 factors++;
9             }
10        }
11        return (factors == 2);
12    }
13
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        input.close();
20        if (isPrime(a)) {
21            System.out.println(a+ " merupakan bilangan prima");
22        } else {
23            System.out.println(a+ " bukan merupakan bilangan prima");
24        }
25    }
26}
27
28 }
```

Kode Program 2.1

Langkah pembuatan kode program

1. Membuat project dan class baru.
2. Menambahkan import.java.util.random.
3. Buat method isPrime(int n) yang akan memeriksa apakah bilangan n adalah bilangan prima dengan menghitung jumlah pembaginya menggunakan perulangan for.
4. Membuat objek scanner.
5. Simpan input pengguna ke dalam variabel a.
6. Gunakan struktur if-else untuk menampilkan hasil apakah bilangan tersebut prima atau bukan.
7. Jalankan program dan cek kembali jika ada kesalahan.

Program Java tersebut berfungsi untuk menentukan apakah suatu bilangan termasuk bilangan prima atau bukan. Program diawali dengan mengimpor library java.util.Scanner untuk membaca input dari pengguna melalui keyboard. Di dalam kelas BilanganPrima_25115330

29, terdapat dua bagian utama, yaitu method `isPrime()` dan method `main()`.

Method `isPrime(int n)` digunakan untuk memeriksa keprimaan suatu bilangan dengan menghitung jumlah pembaginya. Program melakukan perulangan dari 1 hingga n , dan setiap kali n habis dibagi i , nilai `factors` akan bertambah satu. Setelah perulangan selesai, jika jumlah pembaginya tepat dua (yaitu 1 dan dirinya sendiri), maka method akan mengembalikan nilai `true`, menandakan bahwa bilangan tersebut adalah bilangan prima.

Pada method `main()`, program meminta pengguna untuk memasukkan sebuah bilangan. Nilai yang dimasukkan disimpan dalam variabel `a`, lalu diperiksa menggunakan method `isPrime(a)`. Jika hasilnya `true`, program menampilkan pesan bahwa bilangan tersebut merupakan bilangan prima, sedangkan jika `false`, program menampilkan bahwa bilangan tersebut bukan bilangan prima.

2.3.2 Kode program Penggunaan String

Kode program selanjutnya yaitu sebuah kode program Program Java di atas berfungsi untuk mendemonstrasikan konsep *enkapsulasi* dalam pemrograman berorientasi objek (OOP). Enkapsulasi adalah proses menyembunyikan data (variabel) agar tidak dapat diakses langsung dari luar kelas, melainkan harus melalui metode khusus seperti *getter* dan *setter*. Berikut adalah kode programnya.

```

1 package pekan7_2511533029;
2
3 public class Mahasiswa_2511533029 {
4     //variable global
5     private int nim;
6     private String nama,nim2;
7     //membuat mutator (setter)
8
9     public void setNim (int nim) {
10         this.nim=nim;
11     }
12     public void setNim2 (String nim2) {
13         this.nim2=nim2;
14     }
15     public void setNama(String nama) {
16         this.nama=nama;
17     }
18     //membuat accessor
19     public int getNim() {
20         return nim;
21     }
22     public String getNim2() {
23         return nim2;
24     }
25
26     public String getNama() {
27         return nama;
28     }
29     //metode lain
30     public void Cetak() {
31         System.out.println("Nim : "+ nim);
32         System.out.println("Nama : "+ nama);
33     }
34     public void Cetak2() {
35         System.out.println("Nim : "+ nim2);
36         System.out.println("Nama : "+ nama);
37     }
38 }
39 }
```

Kode Program 2.2

Langkah pembuatan kode program

1. Membuat project dan class baru.
2. Buat variabel nim, nama, dan nim2 dengan akses private agar hanya bisa diakses dari dalam kelas.
3. Tambahkan metode setter (mutator): buat tiga metode setNim(int nim), setNim2(String nim2), dan setNama(String nama) untuk memberikan nilai pada variabel-variabel tersebut..
4. Tambahkan metode getter (accessor): buat tiga metode getNim(), getNim2(), dan getNama() yang mengembalikan nilai dari variabel yang sesuai.

5. Tambahkan metode cetak (Cetak() dan Cetak2()) untuk menampilkan nilai variabel ke layar menggunakan System.out.println().
6. Jalankan program dan cek kembali jika ada kesalahan.

Program Java di atas berfungsi untuk mendemonstrasikan konsep *enkapsulasi* dalam pemrograman berorientasi objek (OOP). Enkapsulasi adalah proses menyembunyikan data (variabel) agar tidak dapat diakses langsung dari luar kelas, melainkan harus melalui metode khusus seperti *getter* dan *setter*. Dalam program ini, dibuat sebuah kelas bernama Mahasiswa_2511533029 yang memiliki tiga variabel data, yaitu nim, nama, dan nim2. Ketiganya diberi modifier private, sehingga hanya bisa diakses di dalam kelas itu sendiri.

Program menyediakan beberapa metode *mutator* (*setter*) seperti setNim(), setNim2(), dan setNama() yang berfungsi untuk memberikan nilai ke variabel nim, nim2, dan nama. Selain itu, terdapat metode *accessor* (*getter*) yaitu getNim(), getNim2(), dan getNama() untuk mengambil atau menampilkan nilai dari masing-masing variabel. Dua metode tambahan, Cetak() dan Cetak2(), digunakan untuk menampilkan nilai nim, nim2, dan nama ke layar. Dengan demikian, program ini memperlihatkan bagaimana sebuah objek dapat menyimpan data mahasiswa dengan aman dan hanya dimodifikasi melalui metode yang disediakan.

Konsep ini sangat penting dalam pemrograman berorientasi objek karena membantu menjaga keamanan data, menghindari perubahan langsung pada variabel, dan membuat program lebih mudah dikelola

2.3.3 Kode program Penggunaan String

Kode program selanjutnya yaitu sebuah untuk memanggil dan menggunakan kelas Mahasiswa_2511533029 yang telah dibuat sebelumnya. Kelas ini dinamakan PanggilMahasiswa_2511533029 dan

berisi metode utama main() yang menjadi titik awal eksekusi program. Berikut adalah kode programnya.

```

1 package pekan7_2511533029;
2
3 public class PanggilMahasiswa_2511533029 {
4
5     public static void main(String[] args) {
6         Mahasiswa_2511533029 a= new Mahasiswa_2511533029();
7         a.setNim(3029);
8         a.setNama("Sofian arba'i");
9         System.out.println(a.getNim());
10        System.out.println(a.getNama());
11        a.Cetak();
12    }
13
14
15 }
```

Kode Program 2.3

Langkah pembuatan kode program

1. Membuat project dan class baru.
2. Gunakan metode setter untuk mengisi nilai nim dan nama pada objek tersebut.
3. Panggil metode getter (getNim() dan getNama()) untuk menampilkan nilai yang telah disimpan.
4. Panggil metode Cetak() untuk menampilkan data mahasiswa secara lengkap.
5. Jalankan program dan cek kembali jika ada kesalahan.

Program di atas berfungsi untuk memanggil dan menggunakan kelas Mahasiswa_2511533029 yang telah dibuat sebelumnya. Kelas ini dinamakan PanggilMahasiswa_2511533029 dan berisi metode utama main() yang menjadi titik awal eksekusi program. Di dalamnya, dibuat sebuah objek baru dari kelas Mahasiswa_2511533029 dengan perintah Mahasiswa_2511533029 a = new Mahasiswa_2511533029();. Objek ini digunakan untuk mengakses metode-metode yang ada di kelas Mahasiswa_2511533029, seperti *setter*, *getter*, dan *cetak*.

Pertama, program memanggil metode setNim(3029) untuk memberikan nilai pada variabel nim, lalu memanggil setNama("Sofian arba'i") untuk mengisi nilai nama. Setelah itu, program menampilkan

nilai nim dan nama ke layar menggunakan *getter* getNim() dan getNama(). Terakhir, program menjalankan metode Cetak() yang juga menampilkan data mahasiswa ke layar. Dengan demikian, program ini memperlihatkan bagaimana cara membuat dan menggunakan objek dari kelas lain untuk mengakses data dan metode yang ada di dalamnya.

2.3.4 Kode program Penggunaan String

Kode program selanjutnya yaitu sebuah untuk mendemonstrasikan penggunaan berbagai metode bawaan dari kelas String di Java, yaitu length(), toUpperCase(), toLowerCase(), dan indexOf(). Berikut adalah kode programnya.

```

1 package pekan7_2511533029;
2
3 public class String1_2511533029 {
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()); // output ASSALAMUALATKUM
9         System.out.println(salam.toLowerCase()); // output assalamualaikum
10        System.out.println(salam.indexOf("salam"));
11    }
12 }
13
14 }
```

Kode Program 2.4

Langkah pembuatan kode program

1. Membuat project dan class baru.
2. Deklarasikan variabel String bernama salam dan beri nilai "Assalamualaikum".
3. Gunakan metode bawaan String:
 - a. length() untuk menghitung panjang teks.
 - b. toUpperCase() untuk mengubah huruf menjadi kapital semua.
 - c. toLowerCase() untuk mengubah huruf menjadi kecil semua.
 - d. indexOf("salam") untuk mencari posisi substring tertentu.
4. Cetak hasilnya ke layar menggunakan System.out.println().

5. Jalankan program dan cek kembali jika ada kesalahan.

Program di atas berfungsi untuk mendemonstrasikan penggunaan berbagai metode bawaan dari kelas String di Java, yaitu length(), toUpperCase(), toLowerCase(), dan indexOf(). Program diawali dengan mendeklarasikan variabel salam bertipe String yang berisi teks "Assalamualaikum". Setelah itu, program menggunakan beberapa metode untuk memanipulasi dan menampilkan informasi dari string tersebut.

Pertama, perintah salam.length() digunakan untuk menghitung dan menampilkan jumlah karakter dalam string "Assalamualaikum". Selanjutnya, salam.toUpperCase() mengubah seluruh huruf dalam string menjadi huruf besar (output: ASSALAMUALAIKUM), dan salam.toLowerCase() mengubah semua huruf menjadi huruf kecil (output: assalamualaikum). Terakhir, salam.indexOf("salam") digunakan untuk mencari posisi awal dari kata "salam" di dalam string, di mana hasilnya akan berupa indeks posisi karakter pertama yang ditemukan. Program ini menunjukkan bagaimana kelas String dapat digunakan untuk melakukan operasi sederhana namun penting dalam pengolahan teks di Java.

2.3.5 Kode program Penggunaan String

Kode program terkahir yaitu sebuah untuk menjelaskan penggunaan operasi string dan perbedaan antara penjumlahan angka dengan penggabungan teks (string concatenation) dalam bahasa pemrograman Java. Berikut adalah kode programnya.

```

1 package pekan7_2511533029;
2
3 public class String2_2511533029 {
4
5     public static void main(String[] args) {
6         String firstname = "Sofian";
7         String lastname = " Arba'i";
8         String txt1 = "dosen\"intelektual\" kampus";
9         System.out.println("Nama lengkap: " +firstname + lastname);
10        System.out.println("Nama lengkap: " +firstname.concat(lastname));
11        System.out.println(txt1);
12        int x=10;
13        int y=20;
14        int z=x+y;
15        System.out.println("x + y = "+z);
16        String a="10";
17        String b= "20";
18        String c= a+b;
19        System.out.println("String a + String b = " + c);
20        String v = a+y;
21        System.out.println("string a + string y = " +v);
22    }

```

Kode Program 2.5

1. Membuat project dan class baru.
2. Deklarasikan variabel string firstname, lastname, dan txt1 untuk menyimpan teks.
3. Gunakan operator “+” dan metode .concat() untuk menggabungkan dua string menjadi satu.
4. Tambahkan variabel integer x dan y, lakukan operasi x + y untuk menunjukkan hasil penjumlahan angka.
5. Tambahkan variabel string a dan b, lalu lakukan a + b untuk memperlihatkan hasil penggabungan teks.
6. Jalankan program dan cek Kembali jika ada kesalahan.

Program ini mendemonstrasikan bagaimana dua variabel string dapat digabungkan, cara menggunakan metode concat(), serta bagaimana Java memperlakukan operasi “+” ketika melibatkan tipe data berbeda, seperti angka dan string.

Program diawali dengan mendeklarasikan dua variabel string, yaitu firstname dan lastname, yang masing-masing berisi “Sofian” dan “Arba'i”. Kedua string ini kemudian digabung menggunakan dua cara: dengan operator + dan metode .concat(). Keduanya menghasilkan output “Nama lengkap: Sofian Arba'i”. Selanjutnya, variabel txt1 digunakan untuk menampilkan teks yang berisi tanda kutip ganda di

dalam string menggunakan karakter escape \" , sehingga hasilnya akan muncul sebagai dosen "intelektual" kampus.

Program kemudian menunjukkan perbedaan antara penjumlahan angka dan penggabungan string. Ketika x dan y bertipe int, operasi x + y menghasilkan nilai numerik 30. Namun, ketika a dan b bertipe String, operasi a + b tidak menjumlahkan, melainkan menggabungkan menjadi "1020". Hal yang sama terjadi saat String a digabung dengan int y, hasilnya tetap berupa string "1020" karena tipe data string memiliki prioritas lebih tinggi dalam operasi concatenation. Dengan demikian, program ini memperlihatkan bagaimana Java secara otomatis mengonversi angka menjadi string jika digabungkan dengan teks.

BAB III

PENUTUP

3.1 Kesimpulan

Praktikum pekan ke-7 mengenai *String pada Java* memberikan pemahaman yang mendalam tentang konsep dasar dan penerapan manipulasi string dalam pemrograman Java. Mahasiswa telah mempelajari berbagai metode penting dalam kelas String seperti length(), charAt(), substring(), equals(), toUpperCase(), dan toLowerCase() serta memahami fungsinya dalam pengolahan data teks. Melalui studi kasus dan implementasi program, mahasiswa juga diperkenalkan pada konsep enkapsulasi dan penggunaan objek dalam Java, yang memperkuat pemahaman terhadap prinsip pemrograman berorientasi objek. Praktikum ini berhasil menunjukkan bagaimana string digunakan dalam berbagai konteks, mulai dari validasi input hingga penggabungan teks, serta membedakan antara operasi numerik dan operasi string.

3.2 Saran

Agar pemahaman tentang string dalam Java semakin optimal, disarankan kepada mahasiswa untuk:

1. Melakukan eksplorasi lebih lanjut terhadap kelas String dan kelas pendukung lainnya seperti StringBuilder dan StringBuffer untuk memahami efisiensi manipulasi teks.
2. Aktif berdiskusi dan bertanya dalam sesi praktikum untuk memperjelas konsep yang belum dipahami, serta membiasakan diri membaca dokumentasi resmi Java.
3. Memperhatikan struktur kode dan dokumentasi program agar lebih mudah dipahami dan dikelola, terutama saat menggunakan metode getter dan setter.

DAFTAR PUSTAKA

- [1] codingstudio, “String Adalah: Pengertian, Jenis, Fungsi, dan Contohnya,”. [Daring]. Tersedia pada: [String Adalah? Pengertian, Jenis, Fungsi, Dan Contohnya](#) [Diakses: 13-Nov-2025].
- [2] Blog Rumah Coding, “Javanized : Cara Manipulasi String pada Java,”. [Daring]. Tersedia pada: Javanized : Cara Manipulasi String pada Java - Blog Rumah Coding [Diakses: 13-Nov-2025].