

LAPORAN PRAKTIKUM
PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN
“STRING PADA JAVA”



disusun Oleh:
NAIRA RAMADHANI HALIL
2511533027
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 penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya, laporan praktikum dengan judul “STRING PADA JAVA” dapat diselesaikan tepat waktu.

Laporan ini disusun untuk memenuhi salah satu tugas mata kuliah Praktikum Algoritma dan Pemrograman, sekaligus sebagai sarana pembelajaran dalam memahami konsep dasar pemrograman khususnya materi string pada Bahasa Java.

Pada kesempatan ini, penulis menyampaikan terimakasih kepada:

1. Bapak Dr. Wahyudi, S.T., M.T. selaku dosen pengampu mata kuliah Algoritma dan pemrograman yang telah memberikan bimbingan dan arahan.
2. Bang Jovantri Immanuel Gulo selaku asisten praktikum kelas A yang telah membantu pelaksanaan praktikum.
3. Teman-teman mahasiswa yang telah membantu dan memberi dukungan serta berdiskusi bersama dalam menyelesaikan laporan ini.

Penulis menyadari bahwa laporan ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi penyempurnaan laporan di masa mendatang.

Akhir kata, semoga laporan ini dapat memberikan manfaat bagi pembaca dan menambah wawasan mengenai pemrograman Java.

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Tujuan Praktikum.....	1
1.3 Manfaat Praktikum.....	1
BAB II PE,BAHASAN.....	3
2.1 Langkah Kerja Praktikum	3
2.2 Analisis Hasil dan Pembahasan	10
BAB III PENUTUP.....	14
3.1 Kesimpulan	14
DAFTAR PUSTAKA	

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam pemrograman Java, *string* merupakan salah satu tipe data yang paling sering digunakan dalam pengolahan teks. Java menyediakan kelas **String**, **StringBuilder**, dan **StringBuffer** untuk mempermudah manipulasi data berupa karakter. Pemahaman mengenai string sangat penting karena hampir semua aplikasi membutuhkan pengolahan teks, seperti membaca input, menampilkan output, memformat data, serta melakukan pencarian atau penggantian kata.

Pada materi pekan ke-7, fokus pembelajaran diarahkan pada cara menggunakan objek String, mengenali sifat *immutable* pada String, serta menerapkan berbagai metode penting seperti length(), charAt(), substring(), toUpperCase(), toLowerCase(), equals(), dan fungsi manipulasi lainnya. Melalui praktikum ini, mahasiswa diharapkan mampu memahami konsep dasar pengolahan string sekaligus memanfaatkannya dalam menyelesaikan masalah pemrograman sederhana.

1.2 Tujuan Praktikum

1. Memahami konsep dasar string dalam bahasa pemrograman Java.
2. Mengetahui perbedaan antara **String**, **StringBuilder**, dan **StringBuffer**.
3. Menggunakan berbagai metode pada kelas String untuk melakukan manipulasi teks.
4. Melatih kemampuan dalam membaca, memproses, dan mengolah input berupa teks.
5. Menerapkan string dalam program sederhana sebagai pemecahan masalah.
6. Meningkatkan keterampilan mahasiswa dalam menulis kode yang efisien dan mudah dipahami ketika berhubungan dengan data teks.

1.3 Manfaat Praktikum

1. Mahasiswa mampu memahami cara kerja string dan perilakunya yang *immutable* dalam Java.
2. Mahasiswa dapat menggunakan berbagai metode string untuk kebutuhan pemrosesan teks.
3. Meningkatkan kemampuan dalam membuat program yang memerlukan manipulasi teks seperti pencarian, pemisahan, penggabungan, dan modifikasi kata.

4. Menjadi bahan penting untuk materi pemrograman selanjutnya seperti pengolahan file, database, dan GUI yang juga menggunakan string.
5. Membantu mahasiswa dalam menyusun program yang lebih interaktif dan dinamis melalui pengelolaan input-output berbasis teks.

BAB II

PEMBAHASAN

2.1 Langkah Kerja Praktikum

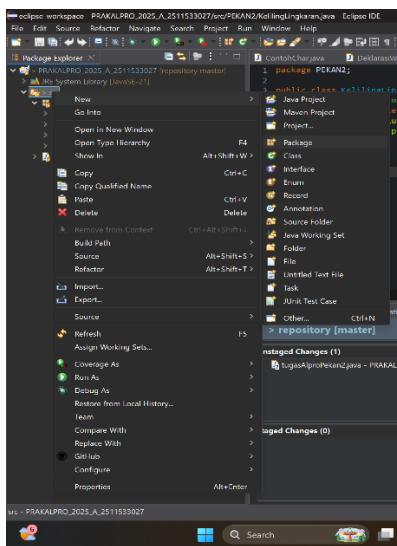
1. Persiapan Awal

- Buka aplikasi Eclipse IDE di laptop atau di komputer.

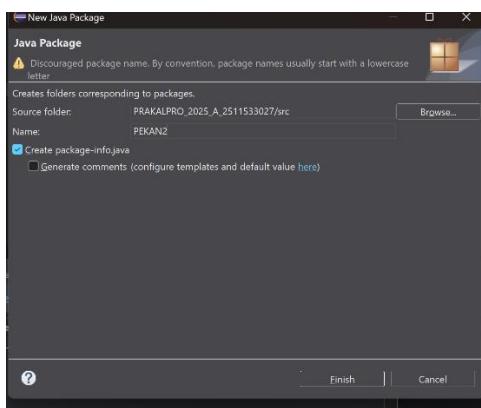


2. Membuat Package

- Klik kana pada **src**, lalu pilih **New**, terakhir klik **Package**.



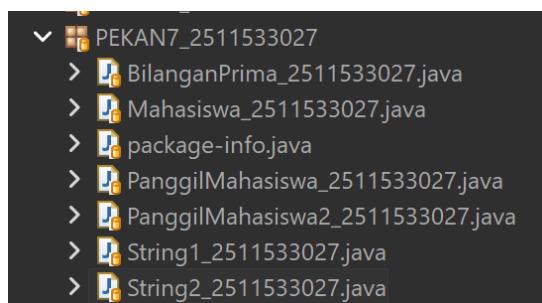
- Beri nama package **PEKAN7_2511533027**.



3. Membuat Class Baru untuk Setiap Percobaan

- Klik kanan pada package **PEKAN7**, lalu pilih **New**, terakhir klik **Class**.
- Buat beberapa class sesuai percobaan.

- BilanganPrima_2511533027.java
- Mahasiswa_2511533027.java
- PanggilMahasiswa_2511533027.java
- PanggilMahasiswa2_2511533027.java
- String1_2511533027.java
- String2_2511533027.java



- Saat membuat class, centang opsi **public static void main(String[] args)** agar otomatis ada fungsi **main**.

4. Menulis Kode Program

a) BilanganPrima_2511533027.java

- Buat method isPrime(int n) untuk menentukan apakah angka prima.
- Pada main, buat objek Scanner untuk menerima input angka.
- Baca nilai integer dari pengguna.
- Panggil method isPrime(a) untuk mengecek ke-prima-an.
- Tampilkan apakah bilangan tersebut prima atau bukan.
- Jalankan program dan amati output.

```

1 package PEKAN7_2511533027;
2
3 import java.util.Scanner;
4
5 public class BilanganPrima_2511533027 {
6
7     public static boolean isPrime (int n) {
8         int factors = 0;
9         for (int i=1; i<=n ; i++) {
10            if (n%i==0) {
11                factors++;
12            }
13        }
14        return (factors==2);
15    }
16
17    public static void main(String[] args) {
18        Scanner input = new Scanner(System.in);
19        System.out.print("Input nilai n = ");
20        int a = input.nextInt();
21        if (isPrime(a)) {
22            System.out.println(a+" bilangan prima");
23        }else{
24            System.out.println(a+" bukan bilangan prima");
25        }
26    }
27
28 }

```

- Output:

The screenshot shows a Java application running in an IDE. The console tab is active, displaying the following text:
<terminated> BilanganPrima_2511533027 [Java Application] C:\
Input nilai n = 12
12 bukan bilangan prima

b) Mahasiswa_2511533027.java

- Buat method setter untuk mengisi nilai variabel.
- Buat method getter untuk mengambil nilai variabel.
- Buat method Cetak() dan Cetak2() untuk menampilkan data.
- Simpan class sebagai objek pendukung untuk percobaan berikutnya.

```

1 package PEKAN7_2511533027;
2
3 public class Mahasiswa_2511533027 {
4     // variabel global
5     private int nim;
6     private String nama, nim2;
7
8     // membuat mutator (setter)
9     public void setNim(int nim) {
10         this.nim = nim;
11     }
12
13     public void setNim2(String nim2) {
14         this.nim2 = nim2;
15     }
16
17     public void setNama(String nama) {
18         this.nama = nama;
19     }
20
21     // membuat accessor (getter)
22     public int getNim() {
23         return nim;
24     }
25
26     public String getNim2() {
27         return nim2;
28     }
29
30     public String getName() {
31         return nama;
32     }
33
34     // metode lain
35     public void Cetak() {
36         System.out.println("Nim : " + nim);
37         System.out.println("Nama : " + nama);
38     }
39
40     public void Cetak2() {
41         System.out.println("Nim : " + nim2);
42         System.out.println("Nama : " + nama);
43     }
44 }
```

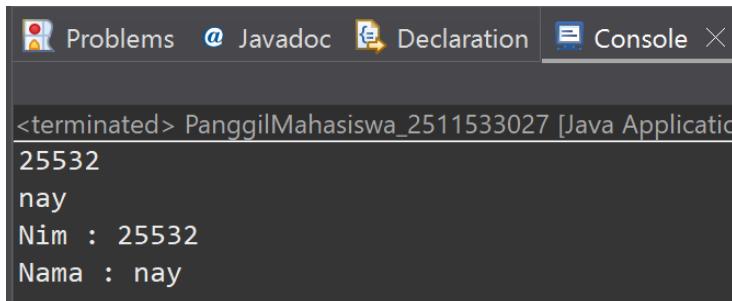
c) PanggilMahasiswa_2511533027.java

- Import class Mahasiswa_2511533027.
- Buat objek baru:
 - ❖ Mahasiswa_2511533027 a = new
 - ❖ Mahasiswa_2511533027();
- Isi nilai NIM dan Nama menggunakan setter.
- Ambil nilai menggunakan getter untuk ditampilkan.
- Panggil metode Cetak() untuk menampilkan data.
- Jalankan program.

```

1 package PEKAN7_2511533027;
2
3 public class PanggilMahasiswa_2511533027 {
4
5     public static void main(String[] args) {
6         Mahasiswa_2511533027 a = new Mahasiswa_2511533027();
7         a.setNim (25532);
8         a.setNama ("nay");
9         System.out.println(a.getNim());
10        System.out.println(a.getNama());
11        a.Cetak();
12    }
13
14 }
```

- Output:



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the following text:

```

<terminated> PanggilMahasiswa_2511533027 [Java Application]
25532
nay
Nim : 25532
Nama : nay
```

d) PanggilMahasiswa2_2511533027.java

- Import Scanner.
- Minta pengguna memasukkan:
 - ❖ NIM (string)
 - ❖ Nama
- Buat objek Mahasiswa.
- Simpan input ke variabel nim2 dan nama menggunakan setter.
- Gunakan fungsi string:
 - ❖ .startsWith("25") → cek angkatan 2025
 - ❖ .contains("1153") → cek prodi Informatika
- Cetak data menggunakan Cetak2().
- Jalankan program dan amati output.

```

1 package PEKAN7_2511533027
2 ;
3 import java.util.Scanner;
4
5 public class PanggilMahasiswa2_2511533027 {
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_2511533027 a = new Mahasiswa_2511533027();
14       a.setNim2(x);
15       a.setNama(y);
16       if (a.getNim2().startsWith("25")) {
17           System.out.println(a.getNama()+" anda angkatan 2025");
18       }
19       if (a.getNim2().contains("1153")) {
20           System.out.println("Anda merupakan mahasiswa Informatika");
21
22       }
23       a.Cetak2();
24       input.close();
25   }
26 }
27 }
```

- Output:

The screenshot shows a Java application window with tabs for Problems, Javadoc, Declaration, and Console. The Console tab is active, displaying the following text:

```

<terminated> PanggilMahasiswa2_2511533027 [Java Application]
NIM: 2511533027
Nama: Naira
Naira anda angkatan 2025
Anda merupakan mahasiswa Informatika
Nim : 2511533027
Nama : Naira

```

e) String1_2511533027.java

- Buat variabel string: salam = "Assalamualaikum";
- Cetak panjang string menggunakan `length()`.
- Cetak string dalam huruf besar menggunakan `toUpperCase()`.
- Cetak dalam huruf kecil menggunakan `toLowerCase()`.
- Cari posisi kata "salam" dengan `indexOf("salam")`.
- Jalankan program.

```

1 package PEKAN7_2511533027;
2
3 public class String1_2511533027 {
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()); //outputsnya ga kapital
10       System.out.println(salam.indexOf("salam")); //outputs 2
11   }
12 }
13 }
```

- Output:

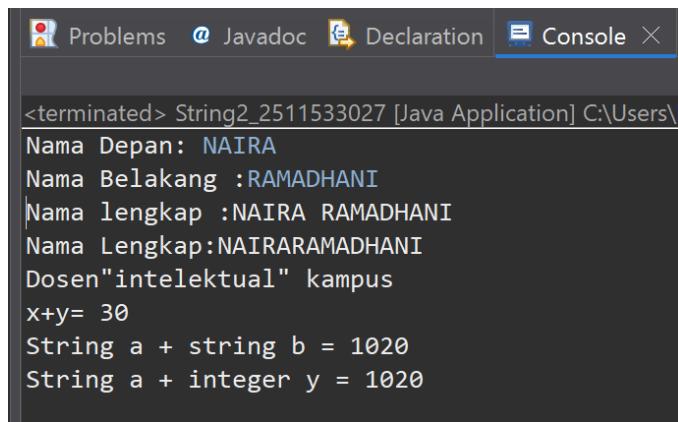
```
<terminated> String1_2511533027 [Java Application] C:\Users\panjang salam adalah : 15
ASSALAMUALAIKUM
assalamualaikum
2
```

f) String2_2511533027.java

- Minta input nama depan dan nama belakang.
- Gabungkan keduanya menggunakan operator +.
- Gabungkan menggunakan concat().
- Tampilkan kalimat teks dengan tanda kutip di dalam string.
- Lakukan operasi:
 - ❖ Penjumlahan integer
 - ❖ Penggabungan string angka
 - ❖ Penggabungan string + integer
- Tampilkan seluruh hasil dan jalankan program.

```
1 package PEKANT_2511533027;
2
3 import java.util.Scanner;
4
5 public class String2_2511533027 {
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       input.close();
28    }
29
30 }
```

- Output:

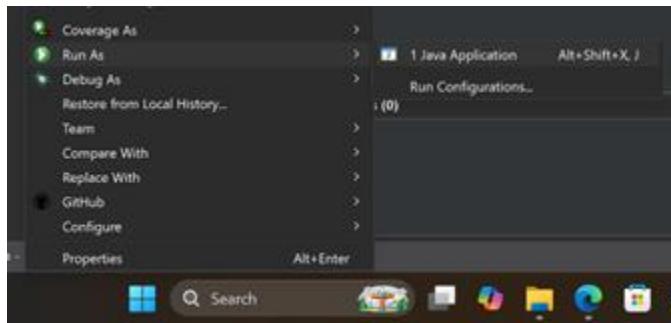


The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the following text:

```
<terminated> String2_2511533027 [Java Application] C:\Users\H
Nama Depan: NAIRA
Nama Belakang :RAMADHANI
Nama lengkap :NAIRA RAMADHANI
Nama Lengkap:NAIRARAMADHANI
Dosen"intelektual" kampus
x+y= 30
String a + string b = 1020
String a + integer y = 1020
```

5. Kompilasi dan Menjalankan Program

- Klik kanan pada file program, pilih **Run As**, terakhir klik **Java Application**.
- Lihat hasil output pada tab Console.



2.2 Analisis Hasil dan Pembahasan

- BilanganPrima_2511533027.java**
 - Hasil: Program meminta pengguna memasukkan sebuah nilai integer. Setelah itu, program menentukan apakah angka tersebut merupakan bilangan prima atau bukan. Jika faktor pembagi angka hanya dua (1 dan dirinya sendiri), program menampilkan “bilangan prima”; jika tidak, program menampilkan “bukan bilangan prima”.
 - Analisis: Program menggunakan metode `isPrime(int n)` untuk menghitung jumlah faktor dari bilangan yang dimasukkan. Variabel `factors` akan bertambah setiap kali `n % i == 0`. Kondisi bilangan prima dipenuhi jika `factors == 2`. Pada fungsi `main`, input dibaca melalui `Scanner`, kemudian hasil pemeriksaan dari method `isPrime()` digunakan untuk menentukan output. Logika yang digunakan sederhana namun efektif untuk angka kecil.
 - Teori Pendukung: Menurut Deitel & Deitel (2019) dalam *Java How to Program*, bilangan prima dapat ditentukan dengan memeriksa banyaknya pembagi sebuah

bilangan. Penggunaan perulangan for dan operator modulus % merupakan teknik dasar untuk memeriksa faktor bilangan dalam pemrograman.

2. Mahasiswa_2511533027.java

- a) Hasil: Program mendefinisikan sebuah class Mahasiswa dengan atribut nim, nim2 (string), dan nama. Program menyediakan method setter, getter, serta dua metode cetak (Cetak() dan Cetak2()) untuk menampilkan data mahasiswa.
- b) Analisis: Class ini menerapkan prinsip OOP berupa encapsulation, di mana variabel diberi modifier private sehingga hanya bisa diakses melalui method setter dan getter. Metode Cetak() menampilkan atribut nim dan nama, sedangkan metode Cetak2() menampilkan nim2 dan nama. Class ini menjadi dasar bagi program lain yang memanggil objek Mahasiswa.
- c) Teori Pendukung: Menurut Barnes & Kölking (2017) dalam *Objects First with Java*, encapsulation adalah prinsip OOP yang menyembunyikan detail data agar hanya bisa diakses melalui method tertentu. Hal ini menjaga keamanan data dan memungkinkan pengembangan kode yang lebih terstruktur.

3. PanggilMahasiswa_2511533027.java

- a) Hasil: Program membuat objek Mahasiswa, mengisi nilai nim dan nama, lalu menampilkan keduanya menggunakan getter dan method Cetak(). Output berupa data mahasiswa yang telah diset sebelumnya.
- b) Analisis: Percobaan ini menunjukkan cara instansiasi objek dalam Java. Metode setter setNim() dan setNama() digunakan untuk memberi nilai pada objek, sedangkan getter dan method Cetak() digunakan untuk menampilkan informasi. Program berjalan satu arah tanpa memerlukan input dari pengguna, sehingga menghasilkan output yang konsisten.
- c) Teori Pendukung: Menurut Oracle (2022), objek dalam Java merupakan instance dari class dan dibuat menggunakan keyword new. Konsep ini memungkinkan reuse code dan modularitas. Penggunaan getter dan setter juga merupakan praktik standar dalam pengelolaan atribut objek.

4. PanggilMahasiswa2_2511533027.java

- a) Hasil: Program meminta input NIM dan Nama dari pengguna. Berdasarkan NIM yang dimasukkan, program mengevaluasi apakah pengguna merupakan angkatan 2025 (menggunakan startsWith("25")) dan apakah termasuk mahasiswa Informatika (menggunakan contains("1153")). Akhirnya, program menampilkan data lengkap mahasiswa.

- b) Analisis: Program memanfaatkan method bawaan kelas String seperti .startsWith() dan .contains() untuk melakukan validasi pola data pada NIM. Data yang dimasukkan disimpan ke dalam objek Mahasiswa dan ditampilkan menggunakan Cetak2(). Percobaan ini menggabungkan konsep OOP dengan manipulasi string untuk analisis data sederhana.
- c) Teori Pendukung: Menurut Schildt (2021) dalam *Java: The Complete Reference*, kelas String menyediakan banyak method untuk pencocokan pola dan pencarian substring. Method seperti .contains() dan .startsWith() sangat berguna dalam validasi data teks.

5. String1_2511533027.java

- a) Hasil: Program menampilkan panjang string, versi huruf besar, versi huruf kecil, serta indeks kemunculan substring “salam” dari kata “Assalamualaikum”.
- b) Analisis: Program menggunakan beberapa fungsi dasar String:
- length() untuk menghitung jumlah karakter,
 - toUpperCase() dan toLowerCase() untuk mengubah format huruf,
 - indexOf() untuk mencari posisi substring.

Semua method bekerja sesuai teori bahwa string bersifat immutable sehingga operasi menghasilkan string baru tanpa mengubah nilai aslinya.

- c) Teori Pendukung: Menurut Deitel & Deitel (2019), kelas String menyediakan berbagai mekanisme manipulasi teks dan seluruh objek String bersifat immutable. Operasi seperti toUpperCase() selalu menghasilkan objek baru alih-alih mengubah objek asal.

6. String2_2511533027.java

- a) Hasil: Program meminta pengguna memasukkan nama depan dan belakang, lalu menampilkan hasil penggabungan string menggunakan operator + dan method concat(). Program juga menampilkan teks dengan tanda kutip, melakukan penjumlahan angka, serta memperlihatkan perbedaan antara penjumlahan integer dan penggabungan string.
- b) Analisis: Percobaan ini memperlihatkan berbagai perilaku string dan tipe data di Java. Operator + pada string menggabungkan teks, sedangkan untuk integer digunakan sebagai operator aritmatika. Ekspresi "10" + 20 menghasilkan "1020" karena Java melakukan konversi otomatis ke string (string concatenation).

Program juga menunjukkan penggunaan karakter escape untuk menampilkan tanda kutip.

- c) Teori Pendukung: Menurut Liang (2020) dalam *Introduction to Java Programming*, operator + bersifat overload dalam Java: berfungsi sebagai operator penjumlahan untuk angka dan operator concatenation untuk string. Selain itu, Java melakukan automatic type conversion ke string jika salah satu operand merupakan string.

BAB III

PENUTUP

3.1 Kesimpulan

Praktikum pekan ke-7 mengenai *String pada Java* memberikan pemahaman mendalam tentang cara kerja berbagai method bawaan kelas String, termasuk length(), toUpperCase(), toLowerCase(), substring(), charAt(), serta proses perbandingan string menggunakan equals() dan compareTo().

Melalui beberapa percobaan yang telah dilakukan, dapat disimpulkan bahwa:

1. **String bersifat immutable**, artinya setiap operasi manipulasi string akan menghasilkan objek baru.
2. Method–method string membantu mempermudah pemrosesan teks dalam berbagai kebutuhan pemrograman.
3. Program percobaan menunjukkan bagaimana Java menangani pembacaan input, manipulasi karakter, penentuan panjang string, pemotongan string, hingga pencarian substring.
4. Praktikum ini memperkuat dasar logika pemrograman, terutama dalam memahami struktur objek dan method bawaan pada Java.
5. Pemahaman tentang manipulasi string sangat penting karena hampir semua aplikasi membutuhkan pemrosesan teks, baik pada level input-output maupun pengelolaan data.

Secara keseluruhan, praktikum ini berhasil mencapai tujuan, yaitu memahami konsep dasar string dalam Java, menerapkan method-method penting, dan menghubungkannya dengan teori pemrograman berorientasi objek.

DAFTAR PUSTAKA

- [1] Wahyudi, *String pada Java*, presentasi kuliah, Program Studi Informatika, Universitas Andalas, 2025.
- [2] Schildt, H. (2021). *Java: The Complete Reference* (12th ed.). McGraw-Hill Education.
- [3] Oracle. (2024). *The Java™ Tutorials – Strings*. Oracle Documentation.
<https://docs.oracle.com/javase/tutorial/java/data/strings.html>
- [4] Liang, Y. D. (2022). *Introduction to Java Programming and Data Structures* (12th ed.). Pearson.
- [5] Deitel, P., & Deitel, H. (2020). *Java: How to Program* (11th ed.). Pearson.