

LAPORAN PRATIUM ALGORITMA PEMROGRAMAN

STRING PADA JAVA

DI SUSUN OLEH :

DIGO YUANDRA

NIM 2511533017

DOSEN PENGAMPU : Dr.WAHYUDI, S.T, M.T

ASISTEN LABORATORIUM: JOVANTRI IMMANUEL GULO



DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

PADANG

2025

KATA PENGANTAR

Sebelumnya saya Panjatkan Puji syukur atas kehadiran Tuhan Yang Maha Esa, berkat rahmat dan izinnya juga laporan praktikum “String Pada Java” ini dapat diselesaikan dengan baik. Saya ucapkan terima kasih yang sebesar besarnya kepada Dr. Wahyudi, S.T, M.T selaku dosen pengampu yang telah membimbing Mata kuliah Algoritma dan Pemrograman. Dan tidak lupa saya ucapkan terima kasih kepada Uda AUFAN Tafiqurrahman selaku asisten labor yang telah membimbing praktikum String Pada Java

Laporan ini disusun sebagai Bentuk hasil kegiatan praktikum yang bertujuan untuk memahami konsep dasar String Pada java dalam pemrograman. Penulis menyadari bahwa pemahaman mendalam terhadap kedua topik ini merupakan fondasi essential bagi pengembangan keterampilan pemrograman yang lebih advanced.

Saya menyadari bahwa penulisan laporan praktikum ini masih jauh dari kata sempurna baik dari segi pembahasan dan penulisannya. namun dengan demikian saya telah berupaya dengan segala kemampuan dan pengetahuan yang dimiliki supaya laporan ini dapat selesai dengan baik dan oleh karenanya saya dengan rendah hati menerima saran, masukan dan kritikan guna penyempurnaan laporan ini.

Padang, 11 November 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....	2
DAFTAR ISI	3
BAB I PENDAHULUAN	3
1.1 Latar Belakang.....	3
1.2 Tujuan.....	3
1.3 Manfaat.....	3
BAB II PEMBAHASAN	4
2.1 Langkah pengerjaan	4
BAB III KESIMPULAN	13
3.1 Kesimpulan.....	13
3.2 Saran.....	13
DAFTAR PUSTAKA	13

BAB I PENDAHULUAN

1.1 Latar Belakang

String di Java bukanlah tipe data primitif seperti int atau char, melainkan sebuah class yang berada di dalam paket java.lang. Keunikan String sebagai objek, namun dengan dukungan sintaksis yang mirip tipe data primitif (seperti penggunaan operator + untuk konkatenasi), menjadikannya topik yang krusial untuk dipahami. Pemahaman yang mendalam tentang bagaimana String bekerja, termasuk perbedaan antara String Literal dan String Object (melalui kata kunci new), serta konsep immutability (ketidakmampuan untuk diubah), sangatlah penting.

Kesalahan dalam memahami konsep String seringkali menyebabkan inefisiensi dalam penggunaan memori (memory leak) dan penurunan performa program, terutama ketika melakukan operasi manipulasi String dalam skala besar. Oleh karena itu, praktikum ini dilakukan untuk memberikan pengalaman langsung dalam mengimplementasikan berbagai operasi String, sehingga mahasiswa dapat membedakan cara kerja dan penggunaan String yang tepat dalam membangun aplikasi Java yang efisien dan robust.

1.2 Tujuan

1. Untuk memahami konsep dasar class String dan perbedaannya dengan tipe data primitif.
2. Untuk menganalisis perbedaan antara pembuatan String menggunakan String Literal dan dengan kata kunci new beserta dampaknya terhadap memori.
3. Untuk mengimplementasikan berbagai metode yang tersedia dalam class String (seperti length(), concat(), substring(), equals(), compareTo(), dll) untuk memanipulasi data teks.
4. Untuk memahami konsep immutability pada String dan solusinya menggunakan StringBuilder atau StringBuffer untuk operasi manipulasi String yang kompleks.
5. Untuk membedakan penggunaan metode equals() dan operator == dalam membandingkan String.

1.3 Manfaat

1. Memahami konsep dan operasi String secara mendalam
2. Meningkatkan skill pemecahan masalah pemrograman.

BAB II PEMBAHASAN

2.1 Langkah pengerjaan

1. Program BilanganPrima_2511533017

a) Inisialisasi Scanner dan Variabel

- Objek Scanner dibuat untuk membaca input dari user
- Variabel a dideklarasikan untuk menyimpan bilangan yang akan diperiksa

b) Input dari User

- Program mencetak prompt "Input nilai n =" ke konsol
- Membaca input bilangan dari user dan menyimpannya ke variabel a

c) Pemeriksaan Bilangan Prima

- Program memanggil method isPrime(a) dengan parameter bilangan yang diinput
- Method isPrime melakukan pemeriksaan dengan menghitung jumlah faktor pembagi

d) Proses Pengecekan di Method isPrime

- Variabel factors diinisialisasi dengan nilai 0 untuk menghitung faktor
- Loop for mengecek semua bilangan dari 1 sampai n
- Setiap kali n habis dibagi i ($n \% i == 0$), nilai factors bertambah 1
- Method mengembalikan true jika factors == 2 (bilangan prima), false jika tidak

e) Output Hasil

- Jika isPrime(a) mengembalikan true, program menampilkan "[bilangan] bilangan prima"
- Jika isPrime(a) mengembalikan false, program menampilkan "[bilangan] bukan bilangan prima"

Kode program

```
BilanganPrima_2511533017.java X
1 package Pekan7_2511533017;
2
3 import java.util.Scanner;
4
5 public class BilanganPrima_2511533017 {
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 }
```

Gambar 2.1

Output yang dihasilkan

```
<terminated> BilanganPrima_2511533017
input nilai n = 7
7 bilangan prima
```

Gambar 2.2

2. Program Mahasiswa_2511533017
 - a) Deklarasi Class dan Variabel

- Class Mahasiswa_2511533017 dideklarasikan sebagai template objek mahasiswa
- Terdapat 3 variabel instance:
 - nim (int) - untuk menyimpan NIM dalam bentuk numerik
 - nim2 (String) - untuk menyimpan NIM dalam bentuk string
 - nama (String) - untuk menyimpan nama mahasiswa

b) Method Setter (Mutator)

- setNim(int nim): Mengatur nilai NIM numerik dengan parameter integer
- Menggunakan this.nim untuk membedakan variabel instance dari parameter
- setNim2(String nim2): Mengatur nilai NIM string dengan parameter string
- Menggunakan this.nim2 untuk mengassign nilai ke variabel instance
- setName(String nama): Mengatur nilai nama mahasiswa
- Menggunakan this.nama untuk mengassign nilai ke variabel instance

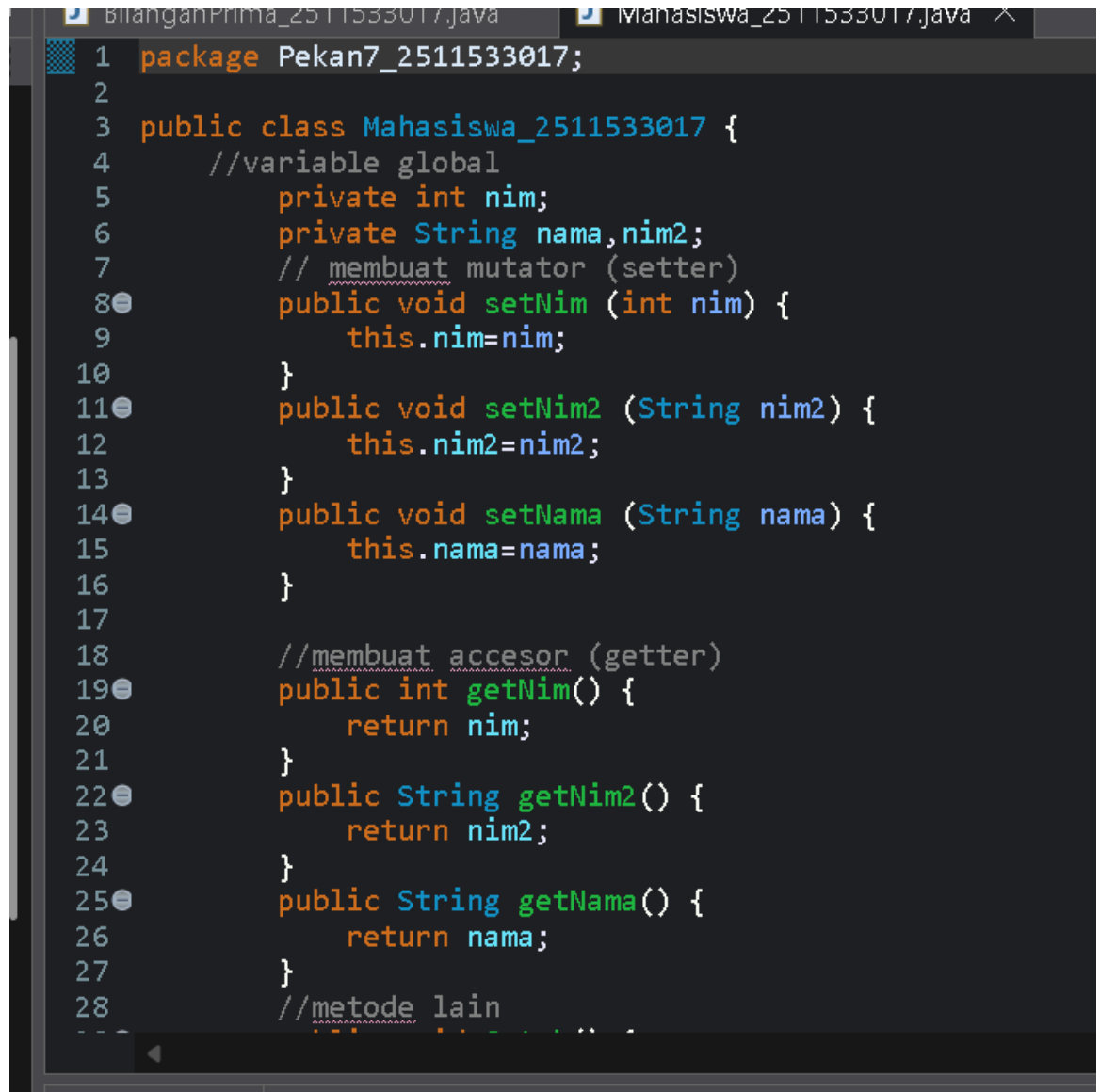
c) Method Getter (Accessor)

- getNim(): Mengembalikan nilai NIM numerik (tipe int)
- Menggunakan return nim untuk mengembalikan nilai variabel instance
- getNim2(): Mengembalikan nilai NIM string (tipe String)
- Menggunakan return nim2 untuk mengembalikan nilai variabel instance
- getName(): Mengembalikan nilai nama mahasiswa (tipe String)
- Menggunakan return nama untuk mengembalikan nilai variabel instance

d) Method Cetak Data

- cetak(): Menampilkan data mahasiswa dengan NIM numerik
- Mencetak "Nim :" diikuti nilai variabel nim
- Mencetak "Nama :" diikuti nilai variabel nama
- cetak2(): Menampilkan data mahasiswa dengan NIM string
- Mencetak "Nim2 :" diikuti nilai variabel nim2
- Mencetak "Nama :" diikuti nilai variabel nama

Kode program:



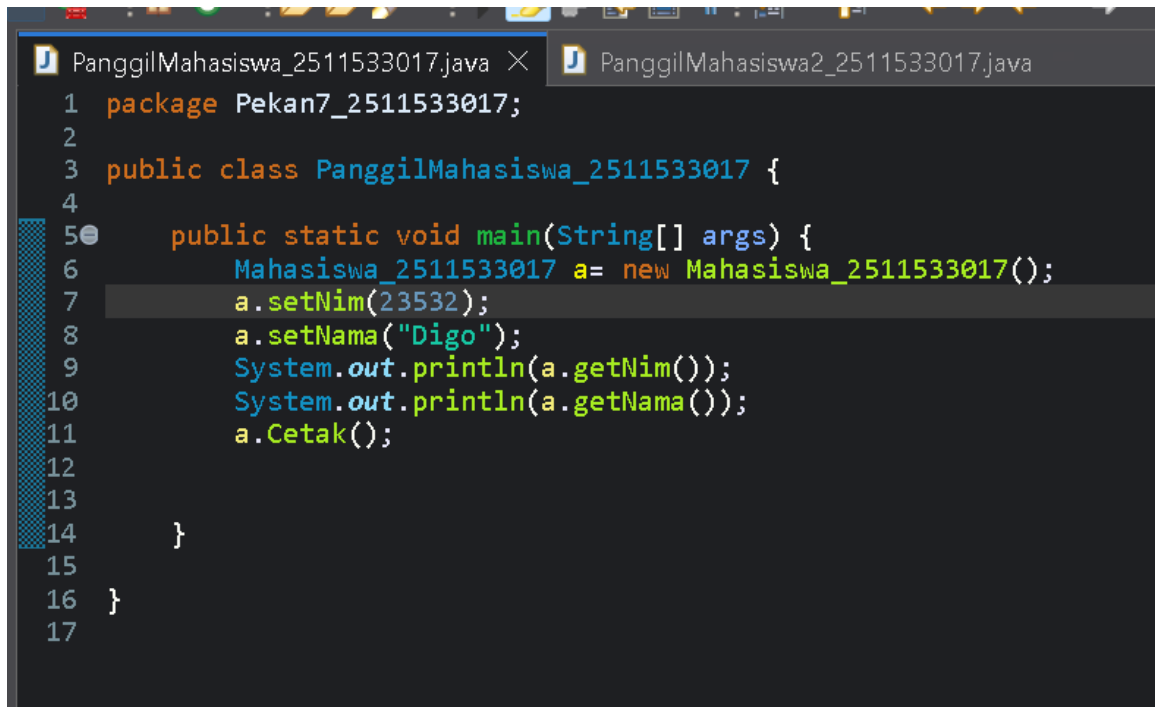
```
1 package Pekan7_2511533017;
2
3 public class Mahasiswa_2511533017 {
4     //variable global
5     private int nim;
6     private String nama,nim2;
7     // membuat mutator (setter)
8     public void setNim (int nim) {
9         this.nim=nim;
10    }
11    public void setNim2 (String nim2) {
12        this.nim2=nim2;
13    }
14    public void setName (String nama) {
15        this.nama=nama;
16    }
17
18    //membuat accesor (getter)
19    public int getNim() {
20        return nim;
21    }
22    public String getNim2() {
23        return nim2;
24    }
25    public String getName() {
26        return nama;
27    }
28    //metode lain
29    ...
30 }
```

Gambar 2.3

3. Program PanggilMahasiswa_2511533017

- a) Package Declaration package pekan7_2511533017 - mengelompokkan class dalam package tertentu
- b) Class Declaration public class PanggilMahasiswa_2511533017 - class utama yang dapat diakses dari mana saja
- c) Main Method
 - public static void main(String[] args) - method utama sebagai entry point program
 - Modifier static - method dapat dipanggil tanpa membuat instance class
 - Return type void - tidak mengembalikan nilai

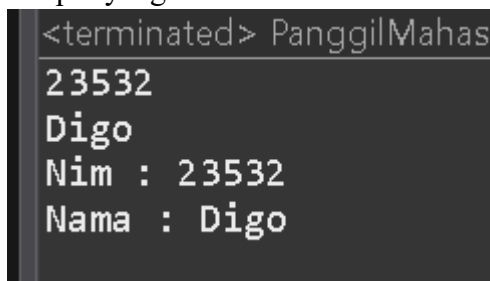
Kode program:



```
1 package Pekan7_2511533017;
2
3 public class PanggilMahasiswa_2511533017 {
4
5     public static void main(String[] args) {
6         Mahasiswa_2511533017 a= new Mahasiswa_2511533017();
7         a.setNim(23532);
8         a.setNama("Digo");
9         System.out.println(a.getNim());
10        System.out.println(a.getNama());
11        a.Cetak();
12
13    }
14
15 }
16
17 }
```

Gambar 2.5

Output yang dihasilkan:



```
<terminated> PanggilMahas
23532
Digo
Nim : 23532
Nama : Digo
```

Gambar 2.6

4. Program PanggilMahasiswa2_2511533017:
 - a) Setup Input
 - Inisialisasi Scanner input untuk baca input keyboard
 - b) Input Data User
 - Minta input NIM → disimpan di variabel x
 - Minta input Nama → disimpan di variabel y
 - c) Pembuatan Objek Mahasiswa
 - Instansiasi objek a dari class Mahasiswa_2511533017
 - Set data menggunakan setNim2(x) dan setNama(y)

d) Validasi Data

- Cek Angkatan: if (x.startsWith("25"))
- Tampilkan "Anda angkatan 2025"
- Cek Program Studi: if (a.getNim2().contains("1153"))
- Tampilkan "Anda Mahasiswa Informatika"

e) Output & Cleanup

- Panggil a.cetak2() untuk tampilkan data
- Tutup scanner dengan input.close()

Kode program:

```
1 package Pekan7_2511533017;
2
3 import java.util.Scanner;
4
5 public class PanggilMahasiswa2_2511533017 {
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_2511533017 a= new Mahasiswa_2511533017();
14        a.setNim2 (x);
15        a.setNama (y);
16        if(x.startsWith("25")) {
17            System.out.println(y + " anda angkatan 2025 ");
18        }
19        if(a.getNim2().contains ("1153")) {
20            System.out.println("Anda Mahasiswa Informatika ");
21        }
22    }
```

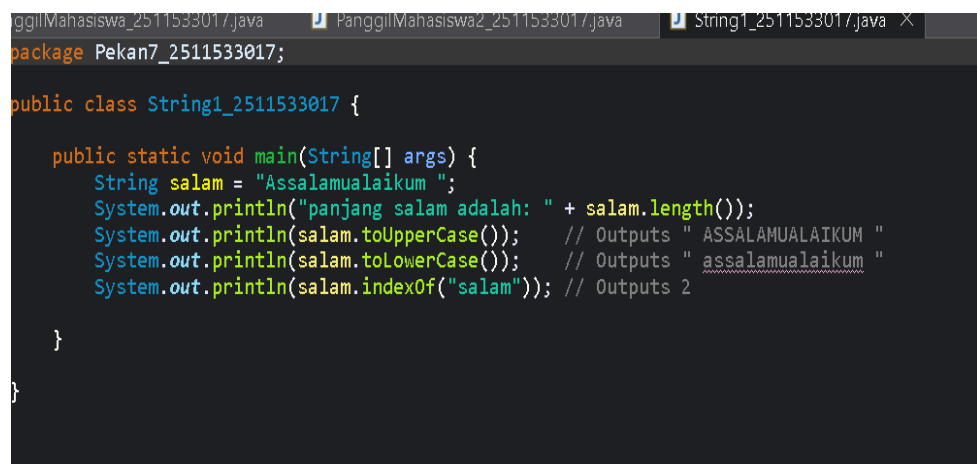
Gambar 2.7 Output
yang dihasilkan:

```
<terminated> PanggilMahasiswa2_2511533017 [Java Ap
NIM: 2511533017
Nama: DIGO YUANDRA
DIGO YUANDRA anda angkatan 2025
Anda Mahasiswa Informatika
Nim : 2511533017
Nama : DIGO YUANDRA
```

5. Program String1_2511533017

- a) Variabel String
 - String salam = "Assalamualaikum"
 - Nilai: "Assalamualaikum" (15 karakter)
- b) Operasi String yang Diuji
 - Panjang String
 - Method: salam.length()
 - Fungsi: Menghitung total karakter
 - Output: "panjang salam adalah :15"
 - Huruf Kapital
 - Method: salam.toUpperCase()
 - Fungsi: Konversi ke huruf besar semua
 - Output: "ASSALAMUALAIKUM"
 - Huruf Kecil
 - Method: salam.toLowerCase()
 - Fungsi: Konversi ke huruf kecil semua
 - Output: "assalamualaikum"
 - Pencarian Substring
 - Method: salam.indexOf("salam")
 - Fungsi: Mencari posisi substring "salam"
 - Output: "3" (index mulai substring)

Kode program:



```
package Pekan7_2511533017;

public class String1_2511533017 {

    public static void main(String[] args) {
        String salam = "Assalamualaikum ";
        System.out.println("panjang salam adalah: " + salam.length());
        System.out.println(salam.toUpperCase()); // Outputs " ASSALAMUALAIKUM "
        System.out.println(salam.toLowerCase()); // Outputs " assalamualaikum "
        System.out.println(salam.indexOf("salam")); // Outputs 2
    }
}
```

Gambar 2.9

Hasil output

```
panjang salam adalah: 16
ASSALAMUALAIKUM
assalamualaikum
2
```

Gambar 2.10

6. Program String2_2511533017

a) Inisialisasi Scanner dan Input Data

- Objek Scanner dibuat untuk membaca input dari user
- Program meminta input "Nama depan:" dan menyimpan di variabel firstName
- Program meminta input "Nama Belakang:" dan menyimpan di variabel lastName

b) Deklarasi dan Manipulasi String

- Variabel txt1 diinisialisasi dengan nilai: "Dosen\"intelektual\"kampus" · Escape character \" digunakan untuk menampilkan tanda kutip dalam string

c) Output Nama Lengkap

- Cara 1: Concatenation dengan operator +
 - firstName + " " + lastName
- Cara 2: Method concat()
 - firstName.concat(lastName)

d) Operasi Aritmatika vs String Concatenation

- Operasi Integer: $x + y = 30$ (penjumlahan numerik)
- Concatenation String: "10" + "20" = "1020" (penggabungan teks) ·
- Mixed Operation: "10" + 20 = "1020" (integer dikonversi otomatis ke string)

e) Cleanup Resources

- Method input.close() dipanggil untuk menutup Scanner

Kode program:

```

package Pekan7_2511533017;

import java.util.Scanner;

public class String2_2511533017 {

    public static void main(String[] args) {
        Scanner input=new Scanner(System.in);
        System.out.print("Nama Depan: ");
        String firstName= input.nextLine();
        System.out.print("Nama Belakang: ");
        String lastName= input.nextLine();
        String txt1 = "Dosen\"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";
    }
}

```

Gambar 2.9

Hasil output

```

Nama Depan: DIGO
Nama Belakang: YUANDRA
Nama Lengkap: DIGO YUANDRA
Nama Lengkap: DIGOYUANDRA
Dosen"intelektual" kampus
x+ y= 30
String a _ string b =1020
String a + integer v =1020

```

Gambar 2.10

BAB III KESIMPULAN

3.1 Kesimpulan

Berdasarkan praktikum yang telah dilakukan, dapat disimpulkan bahwa String dalam Java merupakan object dengan karakteristik khusus berupa immutability (tidak dapat diubah). Berbagai metode built-in seperti `length()`, `toUpperCase()`, `toLowerCase()`, `indexOf()`, `startsWith()`, dan `contains()` terbukti efektif untuk manipulasi dan validasi teks. Metode `equals()` berfungsi untuk membandingkan nilai konten String, sedangkan operator `==` membandingkan reference object dalam memory.

Dalam implementasinya, String dapat diintegrasikan dengan konsep encapsulation melalui property private yang diakses via getter dan setter. Scanner class berperan penting dalam menerima input user yang kemudian dapat diproses menggunakan berbagai metode String. Java juga mampu melakukan conversion otomatis antara tipe data primitif (seperti `int`) dan String ketika menggunakan operator `+`, sehingga memudahkan dalam pembentukan output. Kombinasi seluruh fitur ini memungkinkan pembuatan program yang efektif untuk manipulasi teks, seperti yang terlihat dalam validasi NIM mahasiswa dan operasi String lainnya.

3.2 Saran

1. Akan lebih baik bila dosen menyelenggarakan sesi pra-praktikum di kelas agar mahasiswa dapat memperoleh pemahaman awal yang lebih memadai, sehingga dapat menghindari kepanikan atau kesalahan saat praktikum
2. Sebaiknya dosen membagikan materi praktikum terlebih dahulu melalui iLearn agar mahasiswa bisa mempersiapkan diri sebelum pelaksanaan praktikum

DAFTAR PUSTAKA

- [1] Oracle America, Inc., "Class String (Java Platform SE 8)," Oracle Documentation, 2014. [Online]. Available: <https://docs.oracle.com/javase/8/docs/api/java/lang/String.html> [Accessed: 13-Nov-2025].
- [2] Oracle Corporation, "The Java™ Tutorials – Strings," Oracle, 2019. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/data/strings.html> [Accessed: 13-Nov-2025].