

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

DISUSUN OLEH:

NOFRI ILHAM

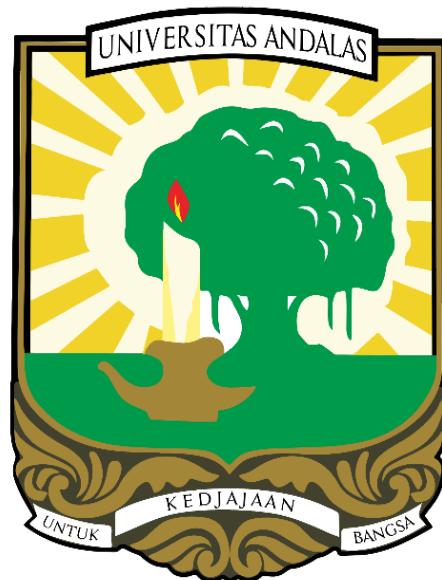
2511531013

DOSEN PENGAMPU:

Dr. WAHYUDI, S.T, M.T

ASISTEN PRAKTIKUM:

MUHAMMAD ZAKI AL HAFIZ



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

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya, laporan praktikum dengan materi *String pada Java* ini dapat diselesaikan dengan baik dan tepat waktu. Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban dan pemahaman penulis terhadap materi praktikum pada mata kuliah Pemrograman Dasar.

Dalam praktikum ini, penulis mempelajari berbagai konsep dasar mengenai manipulasi String, seperti penggunaan metode `length()`, `charAt()`, `substring()`, serta pengolahan teks lainnya yang menjadi bagian penting dalam pemrograman Java. Diharapkan laporan ini dapat memberikan gambaran yang jelas mengenai langkah kerja, hasil percobaan, serta analisis yang penulis lakukan.

Penulis mengucapkan terima kasih kepada dosen pengampu dan asisten praktikum yang telah memberikan bimbingan selama pelaksanaan praktikum. Penulis menyadari bahwa laporan ini masih jauh dari sempurna; oleh karena itu, kritik dan saran sangat diharapkan demi perbaikan di masa mendatang.

Akhir kata, semoga laporan praktikum ini bermanfaat bagi pembaca dan dapat menjadi referensi dalam memahami materi String pada Java.

Padang, 15 November 2025

Nofri Ilham

DAFTAR ISI

Contents

KATA PENGANTAR	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 Jenis-jenis String	3
2.3 Fungsi String	3
2.4 Aplikasi penggunaan String pada Java	5
2.4.1 Kode Program Bilangan Prima	5
2.4.2 Kode Program Mahasiswa	7
2.4.3 Kode Program Panggil Mahasiswa 1	9
2.4.4 Kode Program Panggil Mahasiswa 2	10
2.4.5 Kode Program Panggil Mahasiswa 3	12
2.4.6 Kode Program String 1	13
2.4.7 Kode program String 2	14
BAB III PENUTUP	16
3.1 Kesimpulan	16
3.2 Saran	16
DAFTAR PUSTAKA	17

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam pembelajaran pemrograman Java, memahami tipe data String merupakan hal yang sangat penting karena hampir seluruh proses input, output, dan pengolahan teks bergantung pada penggunaan String. Berbeda dengan tipe data primitif, String bersifat immutable, sehingga setiap perubahan nilai sebenarnya menghasilkan objek baru. Karakteristik ini membuat mahasiswa harus memahami bagaimana Java menangani teks serta bagaimana metode-metode pada kelas String seperti `length()`, `charAt()`, `substring()`, dan `equals()` bekerja untuk mempermudah proses manipulasi karakter. Pemahaman konsep ini menjadi dasar penting sebelum mempelajari materi lanjutan seperti array, file, maupun struktur data.

Melalui praktikum ini, mahasiswa diharapkan dapat menerapkan teori mengenai String dalam bentuk program yang nyata. Dengan melakukan percobaan langsung, mahasiswa dapat mengembangkan kemampuan logika, memahami alur pemrosesan teks dalam Java, serta menyelesaikan permasalahan yang berkaitan dengan pengolahan data berbentuk huruf, kata, maupun kalimat. Praktikum ini juga memberikan pengalaman langsung dalam mengimplementasikan metode String sehingga mahasiswa memiliki fondasi yang kuat dalam membuat program yang lebih kompleks di tahap berikutnya.

1.2 Tujuan

Tujuan dari praktikum pada pekan ke-7 ini adalah:

1. Memahami konsep dasar String dalam bahasa pemrograman Java.
2. Mempelajari sifat String yang bersifat immutable.
3. Menggunakan berbagai metode bawaan String untuk memanipulasi teks.
4. Melatih kemampuan analisis dalam menerapkan String pada studi kasus sederhana.
5. Mengembangkan keterampilan coding mahasiswa dalam mengelola data berbasis teks.

1.3 Manfaat

Manfaat dari praktikum pada pekan ke-7 ini adalah:

1. Mahasiswa mampu mengolah dan memanipulasi teks secara efektif dalam program Java.
2. Mahasiswa dapat memahami metode-metode dasar String sehingga dapat digunakan dalam berbagai proyek pemrograman.
3. Memberikan dasar yang kuat untuk mempelajari materi lanjutan seperti array, file handling, dan struktur data.
4. Membantu mahasiswa membangun logika pemrograman yang lebih terstruktur melalui pemahaman manipulasi teks.
5. Meningkatkan kemampuan mahasiswa dalam menyelesaikan permasalahan yang melibatkan data berbentuk teks.

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.

2.2 Jenis-jenis String

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.

2.3 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. Length(), metode length() mengembalikan jumlah karakter dalam string, termasuk spasi dan karakter khusus.
2. charAt() , metode charAt() mengembalikan karakter yang terletak pada indeks tertentu dalam string. Indeksnya berbasis nol, artinya karakter pertama berada pada indeks 0.
3. isEmpty(), Metode isEmpty() memeriksa apakah suatu string kosong (yaitu, memiliki panjang 0). Metode ini mengembalikan nilai true jika string kosong dan false sebaliknya.
4. equals(), metode equals() digunakan untuk membandingkan dua string untuk mengetahui kesamaan konten. Metode ini akan mengembalikan nilai true jika kedua string berisi urutan karakter yang sama, dan false sebaliknya.
5. Substring(), metode substring() digunakan untuk mengekstrak bagian dari suatu string. Metode ini mengembalikan string baru yang dimulai pada indeks yang ditentukan dan berlanjut hingga akhir string atau indeks lain yang ditentukan.
6. concat(), metode concat() digunakan untuk menggabungkan dua string. Metode ini menambahkan string yang ditentukan di akhir string asli.
7. indexOf(), Metode in indexOf() mengembalikan indeks kemunculan pertama karakter atau sub-string tertentu dalam string. Jika karakter atau sub-string tidak ditemukan, metode ini akan mengembalikan -1.
8. Metode Konversi Khusus

Ada dua metode yang umum digunakan untuk mengubah huruf besar/kecil pada string: toLowerCase() dan toUpperCase(). Metode-metode ini mengembalikan string baru dengan semua karakter diubah menjadi huruf kecil atau huruf besar.

- a. toLowerCase(), Metode ini mengubah semua karakter dalam string menjadi huruf kecil.

- b. `toUpperCase()`, Metode ini mengubah semua karakter dalam string menjadi huruf besar.

2.4 Aplikasi penggunaan String pada Java

Pada praktikum pekan ke-7, mahasiswa diberikan instruksi untuk menyalin dan menjalankan kode program String pada java yang telah disiapkan oleh dosen. Kegiatan ini bertujuan sebagai latihan sekaligus contoh kode String pada Java. Berikut merupakan kode program yang digunakan pada praktikum pekan ini.

2.4.1 Kode Program Bilangan Prima

```

1 package pekan7_2511531013;
2 import java.util.Scanner;
3 public class BilanganPrima_2511531013 {
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        return (factors==2);
11    }
12
13
14    public static void main(String[] args) {
15        Scanner input = new Scanner(System.in);
16        System.out.print("Input nilai n = ");
17        int a=input.nextInt();
18        if ( isPrime(a)) {
19            System.out.println(a + " Bilangan Prima ");
20        }else {
21            System.out.println(a + " bukan bilangan prima");
22        }
23
24    }
25}
26
27 }
```

Kode Program 2.1

Langkah” Program

1. Import Scanner untuk membaca input dari user.
2. Buat method `isPrime(int n)` untuk ngecek bilangan prima.

3. Di dalam method, buat variabel factors untuk menghitung jumlah pembagi.
4. Loop dari 1 sampai n untuk tes semua kemungkinan pembagi.
5. Jika $n \% i == 0$, tambahkan nilai factors.
6. Return true jika factors == 2, artinya bilangan prima.
7. Di main(), baca input angka dari user.
8. Panggil isPrime(a) untuk cek hasil.
9. Tampilkan apakah angka tersebut prima atau bukan.

Program diatas digunakan untuk mengecek apakah suatu angka merupakan bilangan prima atau bukan. Pemeriksaan dilakukan melalui method isPrime(), yang bekerja dengan menghitung berapa banyak faktor pembagi dari angka tersebut. Sebuah bilangan disebut prima jika hanya memiliki dua faktor, yaitu 1 dan dirinya sendiri, sehingga kondisi yang dicek adalah factors == 2.

Pada bagian main(), program meminta input angka dari user kemudian memanggil method isPrime() untuk menentukan status bilangan tersebut. Jika hasilnya true, program menampilkan bahwa angka tersebut adalah bilangan prima. Jika false, program menyatakan bahwa angka tersebut bukan bilangan prima. Program ini menjadi contoh dasar bagaimana fungsi, looping, dan logika digunakan bersama untuk melakukan pengecekan sifat bilangan.

2.4.2 Kode Program Mahasiswa

```
1 package pekan7_2511531013;
2
3 public class Mahasiswa_2511531013 {
4
5     //variable global
6     private int nim;
7     private String nama,nim2;
8     //membuat mutator (setter)
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 accestor(getter)
19     public int getNim() {
20         return nim;
21     }
22     public String getNim2() {
23         return nim2;
24     }
25     public String getNama() {
26         return nama;
27     }
28     //metode lain
29     public void Cetak() {
30         System.out.println("Nim : " + nim);
31         System.out.println("Nama : "+ nama);
32     }
33     public void Cetak2() {
34         System.out.println("Nim : " + nim2);
35         System.out.println("Nama : "+ nama);
36     }
37
38
39
40 }
```

Kode Program 2.2

Langkah-langkah

1. Deklarasikan class Mahasiswa_2511531013.
2. Buat atribut private: nim, nama, dan nim2.
3. Buat method setter (setNim, setNim2, setNama) untuk mengisi nilai atribut dari luar.
4. Buat method getter (getNim, getNim2, getNama) untuk mengambil nilai atribut.
5. Buat method Cetak() untuk menampilkan NIM (int) dan nama.
6. Buat method Cetak2() untuk menampilkan NIM (String) dan nama.
7. Kelas ini siap dipakai oleh kelas lain melalui objek mahasiswa.

Kelas Mahasiswa_2511531013 ini dibuat untuk merepresentasikan data seorang mahasiswa menggunakan konsep class dan object pada Java. Di dalamnya terdapat tiga variabel global (nim, nama, dan nim2) yang disimpan sebagai atribut objek. Kelas ini juga menerapkan konsep enkapsulasi, di mana setiap atribut dibuat private agar tidak bisa diakses langsung dari luar kelas. Untuk memanipulasi dan membaca data tersebut, digunakan metode setter (mutator) dan getter (accessor) yang memungkinkan pengaturan nilai atribut secara aman dan terkontrol.

Selain setter dan getter, kelas ini juga menyediakan dua metode tambahan yaitu Cetak() dan Cetak2(). Metode ini berfungsi untuk menampilkan data mahasiswa ke layar, dengan perbedaan pada penggunaan variabel NIM: metode Cetak() menampilkan NIM bertipe int, sedangkan Cetak2() menampilkan NIM bertipe String.

2.4.3 Kode Program Panggil Mahasiswa 1

```

1 package pekan7_2511531013;
2
3 public class PanggilMahasiswa_2511531013 {
4     public static void main(String[] args) {
5         Mahasiswa_2511531013 a = new Mahasiswa_2511531013();
6         a.setNim(511531013);
7         a.setNama("Nofri Ilham");
8         System.out.println(a.getNim());
9         System.out.println(a.getNama());
10        a.Cetak();
11    }
12
13 }
```

Kode Program 2.3

Langkah-langkah

1. Import atau pastikan class Mahasiswa_2511531013 ada dalam paket yang sama.
2. Buat objek a dari class Mahasiswa_2511531013.
3. Isi atribut NIM dan nama menggunakan setNim() dan setNama().
4. Tampilkan nilai NIM dan nama menggunakan getNim() dan getNama().
5. Panggil method Cetak() untuk menampilkan data mahasiswa dalam format bawaan class.
6. Program selesai dieksekusi.

Program PanggilMahasiswa_2511531013 ini berfungsi sebagai kelas utama (main class) untuk menguji dan memanggil fitur yang sudah dibuat pada kelas Mahasiswa_2511531013. Di dalam method main(), program membuat sebuah objek bernama a dari class Mahasiswa. Melalui objek ini, program mengisi data mahasiswa menggunakan metode setter yaitu setNim() dan setNama(). Data yang telah dimasukkan kemudian diakses kembali menggunakan metode getter (getNim() dan getNama()), lalu ditampilkan ke layar.

Selain menampilkan data menggunakan getter, program juga memanggil metode Cetak() yang berada di dalam class Mahasiswa. Method ini secara otomatis mencetak ulang kedua data tersebut dalam format yang sudah disiapkan di dalam class. Dengan demikian, program ini menunjukkan cara memanfaatkan object, setter, getter, dan method internal dari sebuah class untuk mengelola dan menampilkan data mahasiswa.

2.4.4 Kode Program Panggil Mahasiswa 2

```

1 package pekan7_2511531013;
2 import java.util.Scanner;
3 public class PanggilMahasiswa2_2511531013 {
4
5     public static void main(String[] args) {
6         Scanner input=new Scanner(System.in);
7         System.out.print("NIM: ");
8         String x =input.nextLine();
9         System.out.print("Nama: ");
10        String y=input.nextLine();
11        Mahasiswa_2511531013 a=new Mahasiswa_2511531013();
12        a.setNim2(x);
13        a.setNama(y);
14        if(x.startsWith("25")) {
15            System.out.println(a.getNama()+" anda angkatan 2025");
16        }
17        if(a.getNim2().contains("1153")) {
18            System.out.println(y+ " Anda mahasiswa Informatika");
19        }
20    }
21    a.Cetak2();
22    input.close();
23}
24}
25}
26}

```

Kode Program 2.4

Langkah-langkah

1. Buat objek Scanner untuk membaca input dari user.
2. Minta user memasukkan NIM dan nama.
3. Buat objek Mahasiswa_2511531013.
4. Simpan NIM menggunakan setNim2() dan nama dengan setNama().

5. Cek apakah NIM diawali "25" → tampilkan pesan angkatan 2025.
6. Cek apakah NIM mengandung "1153" → tampilkan pesan mahasiswa Informatika.
7. Panggil Cetak2() untuk menampilkan data mahasiswa.
8. Tutup scanner.

Program diatas digunakan untuk menginput data mahasiswa secara langsung dari pengguna melalui keyboard. Program meminta user memasukkan NIM dan nama, kemudian menyimpannya ke dalam objek kelas Mahasiswa_2511531013 menggunakan metode setNim2() dan setNama(). Data NIM disimpan dalam bentuk string karena digunakan untuk pengecekan pola tertentu seperti angkatan dan prodi.

Setelah data tersimpan, program melakukan dua pengecekan: pertama, jika NIM dimulai dengan angka "25", maka program menampilkan bahwa mahasiswa tersebut adalah angkatan 2025. Kedua, jika NIM mengandung substring "1153", program mengidentifikasi bahwa mahasiswa tersebut berasal dari Program Studi Informatika. Pada bagian akhir, method Cetak2() dipanggil untuk menampilkan kembali data mahasiswa menggunakan format cetak yang disediakan dalam kelas Mahasiswa.

2.4.5 Kode Program Panggil Mahasiswa 3

```

1 package pekan7_2511531013;
2 import java.util.Scanner;
3 public class PanggilMahasiswa3_2511531013 {
4
5     public static void main(String[] args) {
6         Mahasiswa_2511531013 a= new Mahasiswa_2511531013();
7         Scanner input =new Scanner(System.in);
8         System.out.print("NIM: ");
9         a.setNim2(input.nextLine());
10        System.out.print("Nama: ");
11        a.setNama(input.nextLine());
12        if(a.getNim2().startsWith("25")) {
13            System.out.println(a.getNama()+" Anda angkatan 2025");
14        }
15    }
16    if(a.getNim2().contains("1153")) {
17        System.out.println(a.getNama()+" Anda Mahasiswa Informatika");
18    }
19    a.Cetak2();
20    input.close();
21 }
22
23 }
24 }
```

Kode Program 2.5

Langkah-langkah

1. Buat objek Mahasiswa_2511531013.
2. Buat objek Scanner untuk membaca input.
3. Minta user memasukkan NIM → set ke setNim2().
4. Minta user memasukkan nama → set ke setNama().
5. Cek NIM dengan startsWith("25") → tampilkan pesan angkatan 2025.
6. Cek NIM dengan contains("1153") → tampilkan pesan mahasiswa Informatika.
7. Panggil Cetak2() untuk menampilkan data mahasiswa.
8. Tutup scanner.

Program PanggilMahasiswa3_2511531013 berfungsi sebagai kelas utama untuk menerima input data mahasiswa dan melakukan pengecekan sederhana berdasarkan NIM. Program mulai dengan membuat objek dari kelas Mahasiswa_2511531013, kemudian

membaca NIM dan nama dari pengguna menggunakan Scanner. Nilai yang dimasukkan langsung disimpan ke dalam objek dengan menggunakan metode setter (`setNim2()` dan `setNama()`). Data yang sudah disimpan kemudian diproses agar program dapat mengenali pola tertentu pada NIM.

Setelah data diterima, program melakukan dua pengecekan menggunakan method `startsWith()` dan `contains()`. Jika NIM diawali dengan "25", program menampilkan bahwa mahasiswa tersebut merupakan angkatan 2025. Jika NIM mengandung substring "1153", program mengidentifikasi mahasiswa sebagai mahasiswa Informatika. Setelah seluruh kondisi diperiksa, program memanggil method `Cetak2()` untuk menampilkan kembali informasi mahasiswa secara lengkap sesuai format pada kelas Mahasiswa.

2.4.6 Kode Program String 1

Kode Program 2.6

Langkah-langkah

1. Buat variabel String bernama salam berisi "Assalamualaikum".
2. Cetak panjang string menggunakan `salam.length()`.
3. Cetak string dalam huruf besar dengan `salam.toUpperCase()`.
4. Cetak string dalam huruf kecil dengan `salam.toLowerCase()`.
5. Cari indeks substring "salam" menggunakan `salam.indexOf("salam")`.
6. Jalankan program dan tampilkan semua output di console.

Program `String1_2511531013` menunjukkan penggunaan beberapa method dasar pada kelas String di Java. Program dimulai dengan mendeklarasikan sebuah variabel bertipe String bernama `salam` yang berisi teks "Assalamualaikum". Setelah itu, program menampilkan panjang string menggunakan method `length()`, yang menghitung jumlah karakter di dalam string. Program juga

menampilkan versi huruf besar dan huruf kecil dari string tersebut menggunakan method toUpperCase() dan toLowerCase().

Selain itu, program memanfaatkan method indexOf() untuk mencari posisi awal dari substring "salam" di dalam variabel salam. Hasil yang ditampilkan berupa angka indeks yang menunjukkan posisi substring tersebut ditemukan.

2.4.7 Kode program String 2

```

1 package pekan7_2511531013;
2 import java.util.Scanner;
3 public class String2_2511531013 {
4
5     public static void main(String[] args) {
6         Scanner input=new Scanner(System.in);
7         System.out.print("Nama depan : ");
8         String firstName=input.nextLine();
9         System.out.print("Nama Belakang : ");
10        String lastName=input.nextLine();
11        String txt1="Dosen\"Intelektual\" Kampus";
12        System.out.println("Nama Lengkap : "+ firstName + " " + lastName );
13        System.out.println("Nama Lengkap : "+ firstName.concat(lastName) );
14        System.out.println(txt1);
15        int x=10;
16        int y=20;
17        int z= x+y;
18        System.out.println("x + y = "+z);
19        String a="10";
20        String b= "20";
21        String c= a+b;
22        System.out.println("String a+ String b =" +c);
23        String v=a+y;
24        System.out.println("String a+ integer y =" +v);
25
26        input.close();
27    }
28 }

```

Kode Program 2.7

Langkah-langkah

1. Buat objek Scanner untuk membaca input dari user.
2. Minta user memasukkan nama depan → simpan ke firstName.
3. Minta user memasukkan nama belakang → simpan ke lastName.
4. Cetak nama lengkap menggunakan operator +.

5. Cetak nama lengkap menggunakan method .concat().
6. Buat string txt1 berisi teks dengan escape character \".
7. Tampilkan isi txt1.
8. Lakukan penjumlahan integer ($x + y$) dan tampilkan hasilnya.
9. Lakukan concatenation string ($a + b$) dan tampilkan hasilnya.
10. Gabungkan string dan integer ($a + y$) lalu tampilkan hasilnya.
11. Tutup scanner.

Program diatas menunjukkan penggunaan dasar dari operasi string, input teks dari pengguna, string concatenation, penggunaan escape character, serta perbedaan antara penjumlahan numerik dan penggabungan string. Program dimulai dengan meminta pengguna memasukkan nama depan dan nama belakang, kemudian menggabungkannya menggunakan dua cara: operator + dan method concat(). Program juga menampilkan sebuah string yang berisi tanda kutip ganda menggunakan escape character ("") pada variabel txt1, sehingga string tersebut dapat dicetak dengan benar.

Selain itu, program membandingkan operasi penjumlahan antara tipe data integer dan string. Tiga variabel integer (x , y , dan z) digunakan untuk menunjukkan hasil penjumlahan numerik biasa. Lalu variabel string (a dan b) digunakan untuk menunjukkan bahwa penggabungan string akan menghasilkan concatenation, bukan penjumlahan. Program juga menampilkan hasil penggabungan string dengan integer ($a + y$), yang secara otomatis mengonversi integer menjadi string. Hal ini memperlihatkan bagaimana Java menangani operasi string dengan tipe data lain dalam satu ekspresi.

BAB III

PENUTUP

3.1 Kesimpulan

Berdasarkan praktikum minggu ke-7 dengan materi *String pada Java*, dapat disimpulkan bahwa:

1. Praktikum minggu ke-7 membantu mahasiswa memahami konsep dasar String dalam Java yang merupakan bagian penting dalam pengolahan teks.
2. Mahasiswa mempelajari sifat String yang bersifat immutable sehingga setiap perubahan akan menghasilkan objek baru.
3. Berbagai metode penting pada String seperti length(), charAt(), substring(), equals(), toUpperCase(), dan indexOf() berhasil dipahami dan diterapkan dalam program.
4. Praktikum juga melatih penggunaan konsep OOP melalui pembuatan class Mahasiswa dengan penerapan enkapsulasi menggunakan setter dan getter.
5. Program-program yang diuji seperti pengecekan bilangan prima dan pemanggilan data mahasiswa memperkuat pemahaman terhadap logika pemrograman dan penggunaan objek dalam Java.
6. Secara keseluruhan, praktikum ini meningkatkan kemampuan mahasiswa dalam memanipulasi data teks serta membangun dasar yang kuat untuk materi pemrograman lanjutan.

3.2 Saran

1. Mahasiswa disarankan mempelajari lebih banyak metode String lainnya seperti replace(), split(), dan trim() agar pemahaman manipulasi teks menjadi lebih lengkap.
2. Penulisan kode sebaiknya menggunakan komentar yang jelas untuk memudahkan proses evaluasi dan pemahaman ulang.
3. Latihan tambahan dengan studi kasus yang lebih beragam perlu dilakukan agar kemampuan logika dan pengolahan String semakin meningkat.
4. Pengembangan class Mahasiswa dapat dilanjutkan dengan menambahkan constructor, validasi input, atau method tambahan agar konsep OOP lebih dikuasai.

DAFTAR PUSTAKA

[1] Coding Studio, “String Adalah? Pengertian, Fungsi, dan Contohnya,” [Daring]. Tersedia pada: <https://codingstudio.id/blog/string-adalah/> [Diakses: 15-Nov-2025].

[2] Java Handbook, “Overview of String Methods dalam Java,” [Daring]. Tersedia pada:

<https://www.javahandbook.com/java-strings/overview-of-string-methods/> [Diakses: 15-Nov-2025].