LAPORAN PRATIKUM
PEMROGRAMAN ALGORITMA PEMROGRAMAN

LAPORAN PRAKTIKUM PEMROGRAMAN DASAR JAVA: STRING PADA JAVA

disusun oleh:

Khairunnisa M.

NIM 2511532005

Dosen Pengampu:

DR. Wahyudi, S. T., M. T.

Asisten Pratikum:

Aufan Taufiqurrahman

DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur penulis penjatkan ke hadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan praktikum ini dengan baik. Laporan ini disusununtuk memenuhi salah satu tugas pada mata kuliah Pemrograman Dasar Java dengan topik pembahasan String.

Melalui praktikum ini, penulis belajar memahami konsep dan penerapan *string* pada pemrograman Java, operasi dasar pada string, dan penggunaan metode bawaan Java yang berhubungan dengan pengolahan teks. Selain itu, praktikum ini juga membantu penulis dalam melatiih logika berpikir secara terstruktur, efisien, dan sistematis dalam menyelesaikan permasalahan pemrograman.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Akhir kata, penulis mengucapkan terima kasih kepada dosen pengampu, asistensi praktikum, serta rekan-rekan yang telah membantu dalam proses pelaksanaan praktikum ini. Semoga laporan ini dapat memberikan manfaat bagi pembaca dan menjadi referensi untuk memahami konsep string.

Padang, 11 November 2025

Khairunnisa M.

DAFTAR ISI

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam pemrograman, *string* merupakan salah satu tipe data penting yang digunakan untuk menyimpan dan memanipulasi teks. Dalam bahasa pemrograman Java, *string* bukan hanya sekedar kumpulan karakter, tetapi juga merupakan objek dari kelas *string* yang memiliki banyak metode bawaan untuk mempermudah pengolahan data teks.

Pemahaman terhadap konsep *string* sangat diperlukan karena hampir semua aplikasi memerlukan interaksi dengan teks, seperti menampilkan pesan, menerima input pengguna, menyimpan nama, atau mengolah data yang berbentuk kalimat. Melalui praktikum ini, mahasiswa dapat memahami cara mendeklarasikan, menggabungkan, membandingkan, serta menggunakan berbagai metode string dalam Java.

Selain itu, praktikum ini juga memperkenalkan penerapan *string* pada program berorientasi objek (OOP), seperti penggunaan ***string*** dalam atribut kelas dan pemanggilan metode. Dengan demikian, mahasiswa dapat memahami tidak hanya aspek sintaks, tetapi juga konsep pemrograman yang lebih terstruktur.

1.2 Tujuan Praktikum

1. Memahami konsep dasar tipe data *string* dalam Java.
2. Mengetahui cara membuat, menyimpan, dan menampilkan data bertipe *string*.
3. Menggunakan berbagai method bawaan dari kelas *string* seperti *length(), toUpperCase() , toLowerCase() , indexOf() , concat() , startsWith() , dan contains()*.
4. Meningkatkan pemahaman logika pemrograman melalui pengolahan input *string* dari pengguna.

1.3 Manfaat Praktikum

1. Mahasiswa dapat memahami dan menerapkan konsep *String* dalam pembuatan program.
2. Terampil dalam menggunakan berbagai fungsi manipulasi *string* untuk mengolah teks.
3. Mampu mengintegrasikan penggunaan *string* dalam konsep OOP, seperti pada atribut dan *method* dalam *class*.

4. Dapat menerapkan *string* dalam kasus nyata, seperti menampilkan identitas mahasiswa, memproses data input, dan membuat tampilan *output* yang informatif.
5. Memperkuat dasar logika pemrograman dan keterampilan berpikir komputasional dalam menyelesaikan masalah berbasis teks.

BAB II

PEMBAHASAN

2.1. BilanganPrima_2511532005

Progam ini menentukan apakah sebuah bilangan merupakan bilangan prima atau bukan, menggunakan perulangan dan fungsi *Boolean isPrime()*.

```
package Pekan7_2511532005;

import java.util.Scanner;

public class BilanganPrima_2511532005 {
    public static boolean isPrime(int n) {
        int factors = 0;
        for (int i = 1; i <= n; i++) {
            if (n % i == 0) {
                factors++;
            }
        }
        return (factors == 2);

    }
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Input nilai n = ");
        int a = input.nextInt();
        if (isPrime(a)) {
            System.out.println(a+" bilangan prima ");
        } else {
            System.out.println(a+ "bukan bilangan prima");
        }
    }


}
```

Gambar 2.1. Kode Pemrograman BilanganPrima_2511532005

Langkah kerja :

1. Mulai program dengan mendeklarasikan kelas BilanganPrima_2511532005.
2. *Import library Scanner* dan *java.util* untuk membaca input dari pengguna.
3. Buat *method* utama *main(String[] args)* sebagai titik awal pemrograman.
4. Di dalam *method main,* buat objek *Scanner* untuk menerima *input* angka dari pengguna.

5. Tampilkan pesan ke layar, untuk meminta pengguna memasukkan sebuah bilangan.
6. Simpan input pengguna ke dalam variabel int angka = *input.nextInt();*
7. Panggil *method isPrime(int n)* untuk memeriksa apakah angka tersebut merupakan bilangan prima.
8. Masukkan logika nya.
9. Lalu Kembali ke *method main,* gunakan struktur percabangan *if-else.*
10. Tampilkan hasil ke layar apakah bilangan tersebut prima atau bukan.
11. Program selesai.

Program ini menunjukkan penerapan logika perulangan dan kondisi untuk menentukan bilangan prima. Dengan membuat fungsi terpisah *(isPrime),* kode menjadi lebih modular dan mudah dipahami. Ini juga memperkuat konsep pemecahan masalah dengan metode *(function)* dalam Java.

2.2. Mahasiswa_2511532005

Program ini merupkan program sederhana yang digunakan untuk mendefinisiskan sebuah kelas bernama Mahasiswa yang memiliki atribut seperti nama, nim, dan kelas. Progam ini menunjukkan penerapan konsep dasar OOP *(Object Oriented Programming)* dalam Java, yaitu enkapsulasi, dengan cara menyembunyikan data menggunakan *modifier private* dan mengaksesnya melalui *method setter* dan *getter*.

```
package Pekan7_2511532005;

public class Mahasiswa_2511532005 {
    // variabel global
    private int nim;
    private String nama, nim2;
    // membuat mutator (setter)
    public void setNim (int nim) {
        this.nim=nim;
    }
    public void setNim2 (String nim2) {
        this.nim2=nim2;
    }
    public void setNama (String nama) {
        this.nama=nama;
    }
    // membuat accessor (getter)
    public int getNim() {
        return nim;
    }
    public String getNim2 () {
        return nim2;
    }
    public String getNama() {
        return nama;
    }
    // metode lain
    public void Cetak() {
        System.out.println("Nim : "+nim);
        System.out.println("Nama : "+nama);
    }
    public void Cetak2()    {
        System.out.println("Nim : "+nim2);
        System.out.println("Nama : "+nama);
    }

}
```

Gambar 2.2. Kode program Mahasiswa_2511532005.

Langkah kerja:

1. Membuat file program Mahasiswa_2511532005.java
2. Mendeklarasikan kelas Mahasiswa_2511532005 untuk mempresentasikan data mahasiswa.
3. Menambahkan atribut seperti nama, nim, dan kelas.
4. Menetapkan *modifier private* untuk menjaga keamanan data.
5. Membuat *method setter* untuk mengisi nilai atribut (setNama, setNim, setKelas).
6. Membuat *method getter* untuk menampilkan atau mengambil nilai atribut (getNama, getNim, getKelas).
7. Kelas ini nantinya akan dipanggil oleh kelas lain (panggilMahasiswa_2511532005) untuk menampilkan data mahasiswa.

Program ini menunjukkan penerapan prinsip enkapulasi dengan baik melalui penggunaan *method setter* dan *getter.* Dengan pendekatan ini, data dalam kelas tidak bisa diakses langsung dari luar, melainkan melalui method khusus yang sudah ditentukan. Hal ini membantu menjaga keamanan data dan membuat struktur program lebih rapi serta mudah dikembangkan. Selain itu, konsep ini juga menjadi dasar penting dalam pembuatan program berbasis objek di Java.

2.3. panggilMahasiswa_2511532005

Program panggilMahasiswa_2511532005 berfungsi sebagai kelas utama *(main class)* yang digunakan untuk menjalankan dan menguji kelas Mahasiswa_2511532005. Program ini membuat objek dari kelas Mahasiswa_2511532005, kemudian memanggil *method setter* untuk mengisi data mahasiswa seperti nama, NIM, dan kelas, serta menggunakan *method getter* untuk menampilkan data tersebut ke layar melalui *System.out.println()*.

```java
package Pekan7_2511532005;

public class panggilMahasiswa_2511532005 {
    public static void main(String[] args) {
        Mahasiswa_2511532005 a = new Mahasiswa_2511532005();
        a.setNim(23532);
        a.setNama("Rahmat");
        System.out.println(a.getNim());
        System.out.println(a.getNama());
        a.Cetak();

    }

}
```

Gambar 2.3. Program Java dengan panggilMahasiswa_2511532005.

Langkah kerja:

1. Mendeklarasikan kelas panggilMahasiswa_2511532005.
2. Membuat *method main(String[] args)* sebagai titik awal eksekusi program.
3. Membuat objek baru dari kelas Mahasiswa_2511532005 dengan sintaks Mahasiswa_2511532005 *mhs= new Mahasiswa_2511532005();*.
4. Mengisi nilai atribut nama, nim, dan kelas menggunakan *method setter* (setNama, setNim, setKelas).
5. Menampilkan data mahasiswa ke layar menggunakan *method getter* (getNama, getNim, getKelas) melalui perintah *System.out.println()*.
6. Program menampilkan hasil berupa informasi lengkap tentang mahasiswa yang sudah diinput.

Program ini memperlihatkan cara mengimplementasikan objek dari sebuah kelas lain di Java. Dengan memanfaatkan *method setter* dan *getter,* data dapat diatur dan ditampilkan denga naman tanpa mengakses variabel secara langsung. Konsep ini memperkuat pemahaman tentang instansiasi objek, pemanggilan *method,* dan hubungan antar kelas dalam paradigma pemrograman berorientasi objek. Program ini juga menjadi dasar penting untuk memahami bagaimana data diolah dan ditampilkan melalui objek di Java

2.4. panggilMahasiswa2_2511532005

Program ini merupakan pengembangan dari program sebelumnya yang hanya menampilkan satu data mahasiswa. Pada program ini, kita bisa menampilkan beberapa data mahasiswa sekaligus. Program ini juga menunjukkan bagaiman cara membuat lebih dari satu objek dari kelas Mahasiswa_2511532005, lalu menampilkan datanya satu per satu. Jadi, intinya program ini memperlihatkan cara kerja objek ganda dalam Java.

```
package Pekan7_2511532005;
import java.util.Scanner;

public class panggilMahasiswa2_2511532005 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("NIM: ");
        String x = input.nextLine();
        System.out.print("NAMA: ");
        String y = input.nextLine();
        Mahasiswa_2511532005 a = new Mahasiswa_2511532005();
        a.setNim2(x);
        a.setNama(y);
        if (x.startsWith("25")) {
            System.out.println(y + " Anda Angkatan 2025");
        }
        if (x.contains("1153")) {
            System.out.println("Anda Mahasiswa Informatika");

        }
        a.Cetak2();
        input.close();

    }

}
```

Gambar 2.4. Program Java dengan panggilMahasiswa2_2511532005.

Langkah kerja:

1. Membuat *class* baru dengan nama panggilMahasiswa2_2511532005.
2. Mendeklarasikan *method main()* sebagai bagian utama program.
3. Membuat beberapa objek dari kelas Mahasiswa_2511532005, misalnya mhs1, mhs2, dan mhs3.
4. Mengisi data setiap mahasiswa (nama, NIM, dan kelas) dengan *method setter*.

5. Menampilkan data tersebut menggunakan *method getter* dan *System.out.println()*.
6. Program akan menampilkan seluruh data mahasiswa secara berurutan di layar.

Dari program ini, kita bisa memahami bahwa satu kelas dapat digunakan untuk membuat banyak objek yang masing-masing menyimpan data sendiri. Konsep ini penting dalam pemrograman berorientasi objek, karena membantu kita mengelola banyak data dengan struktur yang rapi dan efisien. Selain itu, program ini juga memperlihatkan bagaimana Java memungkinkan kita untuk mengelola data yang berbeda tanpa harus menulid ulang kode yang sama berkali-kali.

2.5. String1_2511532005

Program string1_2511532005 menunjukkan penggunaan beberapa *method* bawaan dari *class String* di Java. Program ini berfokus pada cara memanipulasi teks seperti menghitung panjang string, mengubah huruf menjadi besar atau kecil, serta mencari posisi suatu kata di dalam *string.* Program ini membantu kita memahami bagaimana teks bisa diolah dengan mudah menggunakan *method* yang sudah disediakan Java.

```
package Pekan7_2511532005;

public class String1_2511532005 {
    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 2.5. Program Java dengan String1_2511532005.

Langkah kerja:

1. Membuat *class* baru dengan nama String1_2511532005.java .
2. Mendeklarasikan variabel salam dengan nilai “Assalamualaikum”.
3. Menampilkan panjang *string* menggunakan *length()* .
4. Mengubah semua huruf menjadi kapital dengan *toUpperCase()* .
5. Mengubah semua huruf menjadi huruf kecil dengan *toLoweCase()* .

6. Mencari posisi kata “salam” di dalam string menggunakan *indexOf()* .
7. Menampilkan semua hasil tersebut di layar menggunakan *System.out.println()* .

Dari program ini kita bisa belajar bahwa objek *String* di Java memiliki banyak *method* bawaan yang memudahkan manipulasi teks. *Method* seperti *toUpperCase(), toLowerCase()* , dan *indexOf()* sangat berguna dalam pengolahan data berbasis teks, misalnya untuk validasi input atau pencarian kata. Program ini juga memperlihatkan bahwa Java menangani *string* sebagai objek yang fleksibel dan kuat, bukan hanya sekedar kumpulan karakter.

2.6. String2_2511532005

Program String2_2511532005 dibuat untuk memperlihatkan bagaimana cara menggabungkan teks *(string)* dan angka di Java, serta bagaimanan input dari pengguna bisa digunakan dalam operasi *string*. program ini juga menunjukkan penggunan beberapa *method string* seperti *concat()* dan contoh penggunaan karakter *escape* \” di dalam teks.

```java
package Pekan7_2511532005;
import java.util.Scanner;

public class String2_2511532005 {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Nama Depan: ");
        String firstName = input.nextLine();
        System.out.print("Nama Belakang: ");
        String lastName = input.nextLine();
        String txt1 = "Dosen\"intelektual\" kampus";
        System.out.println ("Nama Lnegkap: " +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";
        String c = a + b;
        System.out.println("String a + String b ="+c);
        String v = a + y;
        System.out.println("String a + integer y= "+v);

    }

}
```

Gambar 2.6. Program Java dengan String2_2511532005.

Langkah kerja:

1. Membuat file *String2_2511532005* .
2. Program meminta input dari pengguna berupa nama depan dan nama belakang menggunakan [1] [2] [3].
3. Menyimpan input ke dalam variabel firstName dan lastName.
4. Program menampilkan contoh string dengan tanda kutip ganda menggunakan karakter \" .
5. Setelah itu, program melakukan penjumlahan angka (x + y) dan penggabungan string (a + b).
6. Hasil dari semua operasi ditampilkan di layar untuk menunjukkan perbedaan antara penggabungan teks dan perhitungan angka.

Program ini menunjukkan perbedaan antara penjumlahan angka dan penggabungan teks di Java. Jika dua nilai bertipe int dijumlahkan, hasilnya angka; sedangkan dua String akan digabung jadi satu kalimat. Program juga mengenalkan cara pakai concat() dan karakter khusus (\") untuk menampilkan tanda kutip di dalam teks. Secara singkat, program ini membantu memahami dasar manipulasi string dan tipe data di Java.

BAB III
KESIMPULAN

Berdasarkan hasil praktikum yang telah dilaukan pada materi *String* pada Java, dapat disimpulkan bahwa tipe data *String* memiliki peran yang sangat penting dalam pemrograman karena digunakan untuk menyimpan dan memanipulasi data berbentuk teks. Melalui praktikum ini, mahasiswa telah memahami cara membuat variable *string*, melakukan operasi penggabungan teks, serta menggunakan berbagai *method* bawaan kelas *String* untuk mengolah teks sesuai kebutuhan.

Selain itu, praktikum ini juga memperlihatkan bagaimana *string* dapat digunakan dalam pemrograman berorientasi objek (OOP), yang menggunakan atribut bertipe *string* untuk menyimpan data nama dan NIM, serta memanggil metode setter, getter, dan metode lain untuk menampilkan informasi.

Sebagai saran, mahasiswa disarankan untuk lebih sering berlatih membuat variasi program *string* dengan kondisi dan objek yang lebih kompleks agar pemahaman terhadap konsep ini semakin kuat. Secara keseluruhan, praktikum ini berhasil meningkatkan pemahaman mahasiswa terhadap penggunaan *string* dalam Java serta memperkuat dasar logika dan struktur pemrograman.

DAFTAR PUSTAKA

Bibliography

[1] oracle corporation , "Class String," [Online]. Available: https://docs.oracle.com/javase/8/docs/api/java/lang/String.html. [Accessed 13 November 2025].

[2] Programiz, "Java String (With Examples)," [Online]. Available: https://www.programiz.com/java-programming/string. [Accessed 13 November 2025].

[3] J. Squirrels, "String Java," codegym, 14 Februari 2023. [Online]. Available: https://codegym.cc/id/groups/posts/id.849.string-java. [Accessed 13 November 2025].