

**LAPORAN PRATIKUM PEKAN
ALGORITMA PEMROGRAMAN**

“STRING PADA JAVA”

DISUSUN OLEH:

NABIL FIKRI

2511533011

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 kepada Allah SWT yang telah memberikan kelimpahan rahmat dan hidayahnya yaitu kesehatan dan kekuatan ,atas rahmat tersebut penulis dapat mengerjakan laporan pratikum pekan ke-7 ini dengan tepat waktu. Laporan ini disusun sebagai tugas mingguan mata kuliah Pratikum Algoritma Pemrograman yaitu dengan topik “Penggunaan String” pada Java.

Melalui laporan ini, penulis bertujuan untuk mempelajari serta memahami konsep pada String dalam bahasa Java, khususnya penggunaan struktur while dan do-while. Kedua jenis perulangan tersebut memiliki peranan penting dalam pengembangan program yang efisien, terstruktur, dan mudah dikelola. Praktikum ini juga diharapkan dapat membantu mahasiswa dalam mengasah kemampuan logika serta pemahaman terhadap alur eksekusi program.

Penulis menyadari laporan ini masih jauh dari sempurna, oleh karena itu kritik dan saran sangat diharapkan demi perbaikan ke depannya. Semoga laporan ini dapat bermanfaat bagi penulis maupun pembaca.

Padang, 13 November 2025

Nabil Fikri

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Pratikum	2
1.3 Manfaat Pratikum	2
BAB II PEMBAHASAN	3
2.1 Pengertian String	3
2.2 Jenis-jenis String	3
2.3 Fungsi String	3
2.4 Kode Program	5
2.4.1 Bilangan Prima_2511533011	5
2.4.2 Mahasiswa_2511533011	6
2.4.3 PanggilMahasiswa_2511533011	7
2.4.4 PanggilMahasiswa2_2511533011	8
2.4.5 String1_2511533011	9
2.4.6 String2_2511533011	10
BAB III PENUTUP	12
3.1 Kesimpulan	12
3.2 Saran	12
DAFTAR PUSTAKA	13

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dalam pemrograman Java, String merupakan salah satu tipe data yang paling sering digunakan karena berfungsi untuk menyimpan dan memanipulasi teks atau kumpulan karakter. Hampir semua program Java, baik yang sederhana maupun kompleks, membutuhkan pengolahan data berupa teks, seperti nama pengguna, pesan, input, maupun output. Oleh karena itu, pemahaman tentang penggunaan String menjadi hal yang sangat penting bagi setiap programmer.

Java memerlukan String sebagai objek dari kelas String, bukan sekadar tipe data primitif. Hal ini memungkinkan berbagai operasi seperti penggabungan (concatenation), pemotongan (substring), pengubahan huruf besar dan kecil, serta pencarian karakter tertentu dilakukan dengan mudah menggunakan metode bawaan seperti `length()`, `charAt()`, `toUpperCase()`, dan sebagainya.

Selain itu, Java juga menyediakan kelas tambahan seperti `StringBuilder` dan `StringBuffer` untuk memudahkan manipulasi teks yang bersifat dinamis atau membutuhkan efisiensi memori yang lebih baik. Dengan fitur-fitur ini, penggunaan String dalam Java menjadi fleksibel, efisien, dan sangat mendukung pengembangan aplikasi modern.

Secara keseluruhan, pemahaman tentang konsep dan penggunaan String dalam Java sangat penting untuk membangun program yang efektif, terutama dalam pengolahan data berbasis teks yang menjadi bagian vital dalam hampir semua aplikasi komputer.

1.2 Tujuan Pratikum

1. Memahami konsep dasar dalam bahasa pemrograman Java.
2. Mampu menggunakan metode bawaan kelas String seperti `length()`, `charAt()`, `toUpperCase`, `toLowerCase`, dan lainnya.
3. Mengetahui perbedaan antara penggunaan operator `+` (concatenation) dan metode manipulasi String lainnya.
4. Mampu membedakan penggunaan `set` (mengatur) dan `get` (mengambil).
5. Dapat mengimplementasikan penggunaan String dalam program Java sederhana yang melibatkan input, output, dan pengolahan teks.

1.3 Manfaat Pratikum

1. Mahasiswa memahami dan mampu mengelola data teks dengan lebih efektif dalam pemrograman Java.
2. Dapat menulis program yang lebih efisien dan mudah dibaca dengan memanfaatkan metode-metode String.
3. Meningkatkan kemampuan logika dan pemahaman terhadap konsep objek dan metode di Java.
4. Menjadi dasar untuk mempelajari pemrosesan data yang lebih kompleks, seperti pengolahan file, input pengguna, atau aplikasi berbasis teks.
5. Membantu mahasiswa terbiasa dengan penggunaan objek dan konsep OOP (Object Oriented Programming) yang menjadi inti dari bahasa 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.

2.2 Jenis-jenis String

Ada beberapa jenis String yang umum digunakan dalam pemrograman. Dua jenis diantaranya adalah yang paling umum yaitu :

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

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.

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. Penggabungan (Concatenation)

Fungsi penggabungan digunakan untuk menggabungkan dua atau lebih string menjadi satu string yang lebih panjang. Hal ini dilakukan dengan menggunakan operator “+” atau fungsi concat (tergantung pada bahasa pemrograman yang digunakan).

2. Pemotongan (Substring)

Fungsi pemotongan digunakan untuk mengambil sebagian string yang diinginkan berdasarkan posisi karakter atau indeks. Dalam umumnya, fungsi ini menerima dua parameter: indeks awal dan indeks akhir. Hasilnya adalah potongan string yang diambil.

3. Pencarian dan penggantian (Search and Replace)

Fungsi-fungsi seperti ‘find’ atau ‘replace’ digunakan untuk mencari kemunculan suatu substring dalam string dan menggantinya dengan substring baru. Fungsi find mengembalikan indeks pertama kemunculan substring, sedangkan fungsi replace menggantikan semua kemunculan substring dengan substring baru.

4. Pemisahan (Splitting)

Fungsi split digunakan untuk memecah string menjadi bagian-bagian kecil berdasarkan pemisah tertentu, seperti spasi atau tanda baca. Hasilnya adalah sebuah koleksi (misalnya array atau list) yang berisi potongan-potongan string.

5. Manipulasi Karakter

Fungsi-fungsi seperti 'upper', 'lower', dan 'strip' digunakan untuk melakukan manipulasi pada karakter-karakter dalam string. Fungsi upper mengubah semua karakter dalam string menjadi huruf kapital, 'lower' mengubah semua karakter menjadi huruf kecil, dan 'strip' menghapus spasi atau karakter tambahan di awal atau akhir string.

2.4 Kode Program

2.4.1 Bilangan Prima_2511533011

```

1 package pekan7_2511533011;
2 import java.util.Scanner;
3
4 public class BilanganPrima_2511533011 {
5
6     public static boolean isPrime(int n) {
7         int factors = 0;
8         for (int i = 1; i <= n; i++) {
9             if (n % i == 0) {
10                 factors++;
11             }
12         }
13         return (factors == 2);
14     }
15
16     public static void main (String[] args) {
17         Scanner input = new Scanner(System.in);
18         System.out.print("Input nilai n = ");
19         int a = input.nextInt();
20         if (isPrime(a)) {
21             System.out.println(a+" bilangan prima");
22         } else {
23             System.out.println(a+" bukan bilangan prima");
24         }
25     }
26 }
27 }
28

```

Kode Program 2.4.1

Program ini merupakan program mengecek sebuah bilangan apakah termasuk bilangan prima atau bukan. Scanner berfungsi untuk bisa membaca

input dari keyboard. Code isPrime() digunakan untuk mengecek apakah suatu bilangan prima atau tidak.

Langkah pengerjaan :

1. Program mulai dari main().
2. Masukkan angka (n).
3. Nilai yang dimasukkan akan disimpan ke variabel a.
4. Program ini memanggil fungsi isPrime(a) untuk mengecek a apakah prima :
 - Inisialisasi factors = 0.
 - Loop dari 1 sampai n.
 - Setiap kali n habis dibagi i, factors bertambah.
 - Setelah loop selesai, jika factors == 2 maka hasilnya true, jika tidak false.
5. Program menampilkan hasil :
 - Jika true maka akan menampilkan “a merupakan bilangan prima”
 - Jika false maka akan menampilkan “a bukan bilangan prima”

2.4.2 Mahasiswa_2511533011

```

1 package pekan7_2511533011;
2
3 public class Mahasiswa_2511533011 {
4     // variabel 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    //membuat accessor (getter)
18    public int getNim() {
19        return nim;
20    }
21    public String getNim2() {
22        return nim2;
23    }
24    public String getName() {
25        return nama;
26    }
27    //metode lain
28    public void cetak() {
29        System.out.println("Nim : "+nim);
30        System.out.println("Nama : "+nama);
31    }
32    public void cetak2() {
33        System.out.println("Nim : "+nim2);
34        System.out.println("Nama : "+nama);
35    }
36
37 }

```

Kode Program 2.4.2

Program ini membahas tentang konsep enkapsulasi dalam pemrograman berorientasi objek (OOP). Fungsi utamanya buat menyimpan dan mengatur data mahasiswa yaitu nim dan nama menggunakan setter, getter, dan metode cetak. Setter digunakan untuk mengisi atau mengubah nilai atribut. Kata *this* di sini berarti “variabel yang ada di dalam kelas ini”. Sementara itu, getter digunakan untuk memanggil nilai yang telah diset. Karena variabelnya *private*, maka satu-satunya cara untuk “melihat” nilainya adalah lewat getter ini.

Langkah pengerjaan :

1. Program mendefinisikan kelas Mahasiswa_2511533011.
2. Di dalam kelas, ada atribut nim, nama, dan nim2 yang bersifat *private*.
3. Tiga setter dibuat untuk memberi nilai pada atribut.
4. Tiga getter dibuat untuk mengambil nilai atribut.
5. Dua metode (Cetak dan Cetak2) dibuat untuk menampilkan hasil.
6. Ketika nanti dibuat main class yang memanggil kelas ini:
 - Objek mahasiswa dibuat.
 - Nilai nim dan nama diisi dengan setter.
 - Nilai ditampilkan lewat getter atau metode Cetak().

2.4.3 PanggilMahasiswa_2511533011

```

1 package pekan7_2511533011;
2
3 public class PanggilMahasiswa_2511533011 {
4
5     public static void main(String[] args) {
6         Mahasiswa_2511533011 a = new Mahasiswa_2511533011();
7         a.setNim(23532);
8         a.setNama("Rahmat");
9         System.out.println(a.getNim());
10        System.out.println(a.getNama());
11        a.Cetak();
12    }
13
14 }
```

Kode Program 2.4.3

Program ini merupakan lanjutan dari program 2.5.1 sebagai kelas pemanggil yang berfungsi untuk menguji dan menjalankannya. Pada baris ke 6 yaitu untuk membuat objek baru dari kelas Mahasiswa_2511533011

bernama a, maksudnya yaitu seperti menghidupkan blueprint yang telah dibuat sebelumnya.

Langkah pengerjaan :

1. Program mulai dari main().
2. Objek a dari kelas Mahasiswa_2511533011 dibuat.
3. Objek a diberi nilai nim dan nama menggunakan setNim() dan setNama().
4. Nilai yang sudah diset diambil kembali dengan getNim() dan getNama(), lalu ditampilkan.
5. Method Cetak() dipanggil untuk menampilkan data mahasiswa dalam format rapi.
6. Program selesai.

2.4.4 PanggilMahasiswa2_2511533011

```

1 package pekan7_2511533011;
2
3 import java.util.Scanner;
4
5 public class PanggilMahasiswa2_2511533011 {
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_2511533011 a = new Mahasiswa_2511533011();
14        a.setNim2(x);
15        a.setNama(y);
16        if (x.startsWith("25")) {
17            System.out.println(y + " anda angkatan 2025");
18        }
19        if (x.contains("1153")) {
20            System.out.println("Anda Mahasiswa Informatika");
21        }
22        a.Cetak2();
23        input.close();
24    }
25 }
26
27 }

```

Kode Program 2.4.4

Program ini juga lanjutan dari program 2.5.3 dan 2.5.4 dengan tambahan input dari pengguna (scanner) dan ada logika tambahan buat indentifikasi angkatan & jurusan. Terdapat 3M (membuat, meminta, dan menyimpan) inputan dari pengguna. Kemudian data dimasukkan ke objek menggunakan

setter dan getter. Fungsi `startsWith("25")` memeriksa apakah NIM diawali dengan angka 25, Kalau iya, berarti mahasiswa tersebut angkatan 2025. Sementara itu, Fungsi `contains("1153")` memeriksa apakah NIM mengandung angka “1153”. Terakhir yaitu `a.Cetak2()` untuk menampilkan hasil NIM dan Nama menggunakan method dari kelas `Mahasiswa_2511533011`, lalu ditutup dengan `input.close()`.

Langkah Pengerjaan :

1. Program mulai dari `main()`.
2. Program minta user mengetik NIM dan Nama.
3. Data disimpan ke variabel `x` dan `y`.
4. Objek mahasiswa (`a`) dibuat.
5. Nilai NIM dan Nama dimasukkan ke objek.
6. Program melakukan dua pengecekan:
 - Jika NIM diawali “25” → tampilkan pesan “angkatan 2025”.
 - Jika NIM berisi “1153” → tampilkan pesan “mahasiswa Informatika”.
7. Program menampilkan data lengkap lewat `a.Cetak2()`.
8. Program selesai.

2.4.5 String1_2511533011

```

1 package pekan7_2511533011;
2
3 public class String1_2511533011 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         String salam = "Assalamualaikum";
8         System.out.println("panjang salam adalah : " + salam.length());
9         System.out.println(salam.toUpperCase()); // Outputs "ASSALAMUALAIKUM"
10        System.out.println(salam.toLowerCase()); //Outputs "assalamualaikum"
11        System.out.println(salam.indexOf("salam")); //Outputs 2
12    }
13
14 }
15

```

Kode Program 2..5

Program ini merupakan program sederhana penggunaan kelas `String` seperti `length()` untuk menghitung jumlah karakter dalam string, `toUpperCase()` untuk mengubah semua huruf menjadi kapital, `toLowerCase()` untuk mengubah semua huruf menjadi huruf kecil, dan `indexOf()` untuk Mencari posisi awal substring "salam" di dalam "Assalamualaikum". Dalam string “Assalamualaikum”, tidak ditemukan substring “salam” yang persis sama

(karena huruf 'S' besar di awal kata), sehingga method tersebut menghasilkan nilai -1 yang menandakan bahwa substring tidak ditemukan.

Langkah Pengerjaan :

1. Membuat class baru.
2. Menulis method main.
3. Deklarasi variabel String.
4. Menghitung Panjang variabel.
5. Mengubah huruf menjadi kapital dan kecil.
6. Mencari Posisi substring di dalam String.
7. Menjalankan Program.

2.4.6 String2_2511533011

```

1 package pekan7_2511533011;
2
3 import java.util.Scanner;
4
5 public class String2_2511533011 {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         Scanner input = new Scanner(System.in);
10        System.out.print("Nama Depan :");
11        String firstName = input.nextLine();
12        System.out.print("Nama Belakang :");
13        String lastName = input.nextLine();
14        String txt1 = "Dosen\intelektual\ kampus";
15        System.out.println("Nama Lengkap : "+firstName + " " + lastName);
16        System.out.println("Nama Lengkap : "+firstName.concat(lastName));
17        System.out.println(txt1);
18        int x = 10;
19        int y = 20;
20        int z = x + y;
21        System.out.println("x + y = "+z);
22        String a = "10";
23        String b = "20";
24        String c = a + b;
25        System.out.println("String a + string b = "+c);
26        String v = a + y;
27        System.out.println("String a + integer y = "+v);
28    }
29 }
30
31 }
32

```

Kode Program 2.4.6

Program ini merupakan program penggabungan (concatenation) string dengan menggunakan kode program .concat() untuk menggabungkan string tanpa spasi dan Escape character untuk menampilkan tanda kutip ganda di dalam teks.

Langkah pengerjaan :

1. Buat class baru.
2. Gunakan Scanner.
3. Dekalrasi Scanner.
4. Input Nama Depan dan Nama Belakang.
5. Tambahkan teks dan gabungkan string.
6. Tambahkan Operasi Aritmatika (integer).
7. Tambahkan Operasi gabungan.
8. Gabungkan string dan integer.
9. Jalankan program.

BAB III

PENUTUP

3.1 Kesimpulan

Berdasarkan hasil praktikum yang telah dilakukan, dapat disimpulkan bahwa String dalam bahasa pemrograman Java memiliki peran yang sangat penting dalam pengolahan data berbasis teks. Melalui berbagai metode seperti `length()`, `toUpperCase()`, `toLowerCase()`, `concat()`, dan penggunaan operator `+`, programmer dapat dengan mudah memanipulasi dan mengelola data teks sesuai kebutuhan program.

Selain itu, penggunaan objek `Scanner` juga membantu dalam menerima input teks dari pengguna secara dinamis. Praktikum ini memperlihatkan bahwa Java memperlakukan String sebagai objek dari kelas `String`, bukan sebagai tipe data primitif, sehingga dapat menggunakan berbagai metode bawaan untuk manipulasi teks.

Dengan memahami konsep dan penerapan String ini, mahasiswa mampu membuat program yang lebih interaktif, efisien, serta mudah dibaca dan dikembangkan di kemudian hari.

3.2 Saran

Dalam pelaksanaan praktikum ini, disarankan agar mahasiswa lebih sering berlatih menulis dan menjalankan kode program secara mandiri untuk memperkuat pemahaman tentang konsep String. Selain itu, pemahaman terhadap perbedaan antara operasi aritmatika dan penggabungan string juga perlu diperhatikan agar tidak terjadi kesalahan logika dalam program.

Bagi praktikum berikutnya, disarankan untuk mempelajari kelas pendukung seperti `StringBuilder` dan `StringBuffer` guna memahami cara kerja manipulasi teks yang lebih efisien dalam penggunaan memori dan kecepatan eksekusi. Dengan latihan dan eksplorasi yang berkelanjutan, kemampuan pemrograman mahasiswa dalam Java akan semakin berkembang.

DAFTAR PUSTAKA

- [1] By Team, "String Adalah: Pengertian, Jenis, Fungsi, dan Contohnya," codingstudio, 30 Juli 2023. [Online]. Available: <https://codingstudio.id/blog/string-adalah/>. [Accessed 14 November 2025].