

LAPORAN PRATIKUM ALGORITMA PEMROGRAMAN

STRING PADA JAVA

Disusun Oleh:

Endy Pardilian 2511531017

Dosen Pengampu:

Wahyudi. Dr., S.T,M.T

Asisten Pratikum:

Aufan Taufiqurrahman



DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

PADANG

2025

KATA PENGANTAR

Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban kegiatan praktikum algoritma pemrograman yang membahas mengenai String dalam bahasa pemrograman java. Melalui laporan ini, penulis dapat memahami materi pratikum secara mendalam. Penulisan laporan ini juga dapat melatih ketelitian, keteraturan, serta kemampuan menulis sesuai kaidah akademik. Dengan demikian, laporan pratikum ini dapat berfungsi sebagai sarana belajar, dokumentasi kegiatan, dan referensi untuk praktikum atau pembelajaran jenjang berikutnya.

Penulis menyadari bahwa laporan ini masih memiliki kekurangan, baik dari isi maupun penyajiannya. Oleh sebab itu, saran dan kritik sangat penulis harapkan untuk laporan berikutnya.

Padang, 2025

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan.....	2
1.3 Manfaat.....	2
BAB II PEMBAHASAN	3
2.1 Program Bilangan Prima	3
2.2 Program Biodata Mahasiswa.....	4
2.3 Program String1	7
2.4 Program String2	8
BAB III KESIMPULAN	10
3.1 Kesimpulan.....	10
3.2 Saran.....	10
DAFTAR PUSTAKA.....	11

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam pemrograman Java string merupakan aspek penting, karena dapat mempelajari mengenai class dan objek melalui penggunaan string. String sebenarnya merupakan class yang terdapat dalam library Java. Java String merupakan salah satu kelas dasar yang disediakan oleh Java untuk memanipulasi karakter. Java mendefinisikan class String dalam package `java.lang.String`, sehingga tidak perlu melakukan impor secara eksplisit. Java String digunakan untuk mendefinisikan string yang konstant (tidak bisa berubah). String terdiri dari data array yang bertipe `char`, dan kita juga dapat membuat objek String dengan menggunakan keyword `new` yang biasa digunakan untuk membuat objek dari class.

Untuk membandingkan dua objek String, kita dapat menggunakan fungsi sebagai berikut :

- `equals(String s)` Dengan fungsi ini, maka objek string yang bersangkutan akan dibandingkan dengan objek string `s`, pada parameter fungsi ini, dengan membedakan antara huruf besar dengan huruf kecil.
- `equalsIgnoreCase(String s)` Dengan fungsi ini, maka objek string yang bersangkutan akan dibandingkan dengan objek string `s`, pada parameter fungsi ini, dengan tanpa memperdulikan perbedaan antara huruf besar dengan huruf kecil.

Kedua fungsi diatas akan menghasilkan nilai boolean `true` apabila benar dan `false` apabila salah. Untuk menentukan awal dan akhir String, kita dapat menggunakan dua fungsi utama, yaitu :

- `startsWith(String s)` Dengan fungsi ini, maka objek String yang bersangkutan akan diperiksa, apakah diawali oleh objek String `s`, pada parameter fungsi ini.

- `endsWith(String s)` Dengan fungsi ini, maka objek string yang bersangkutan akan diperiksa, apakah diakhiri oleh objek string `s`, pada parameter fungsi ini.

Fungsi diatas akan menghasilkan nilai boolean `true` bila benar dan `false` bila salah. Kita dapat memperoleh panjang string dengan menggunakan method `length()`; Kita dapat memperoleh panjang string dengan menggunakan method `length()`; Ada dua method yang dapat digunakan untuk mencari posisi karakter dari string dan dua method untuk mendapatkan posisi `subString` dari string.

1. Method untuk mencari posisi karakter pada String:

- `indexOf(char karakter)`
Memerlukan argumen berupa karakter dan akan mengembalikan nilai posisi indeks dari karakter yang dicari. Posisi yang dikembalikan adalah posisi pertama dari karakter yang ditemukan. Bila karakter tidak ditemukan, maka akan mengembalikan nilai `-1`.
- `indexOf(char karakter, int indeks)`
Sama dengan sebelumnya, tetapi memerlukan argumen tambahan, yaitu indeks posisi awal pencarian dalam integer.

2. Method untuk mencari posisi `subString` pada String

- `indexOf(String Str)`
Penggunaan dan fungsi sama dengan method untuk char.
- `indexOf(String str, int indeks)`
- Penggunaan dan fungsi sama dengan method untuk char.

1.2 Tujuan

1. Memahami konsep dasar kelas String dalam bahasa pemrograman java
2. Mengetahui cara membuat dan menginisialisasi objek String
3. Mempelajari berbagai method

1.3 Manfaat

1. Meningkatkan pemahaman tentang cara kerja String
2. Membantu pembuatan program yang membutuhkan pengolahan dan analisis teks
3. Menambah keterampilan menggunakan method method untuk mengolah teks

BAB II

PEMBAHASAN

2.1 Program Bilangan Prima

```
1 package pekan7_2511531017;
2
3 import java.util.Scanner;
4
5 public class BilanganPrima_2511531017 {
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     public static void main(String[] args) {
16         Scanner input = new Scanner(System.in);
17         System.out.print("Input nilai n= ");
18         int a= input.nextInt();
19         if (isPrime(a)) {
20             System.out.println(a+" bilangan prima");
21         }else {
22             System.out.println(a+" bukan bilangan prima");
23         }
24     }
25 }
26
27
```

Code 2.1

Nama file: BilanganPrima_2511531017

1. Variable *factors* untuk menghitung jumlah pembagi variable *n*
2. *For (int i=1;i<=n;i++)*
 - Untuk menentukan nilai *i* dari 1 hingga *n*
 - *if (n%i ==0)*, saat *i* habis membagi *n* maka *i* adalah faktor *n*
 - *factors* akan bertambah setiap ditemukan factor
3. Bilangan prima hanya memiliki 2 factor
 - Jika *factors == 2*, adalah bilangan prima
 - Jika *factors != 2*, bukan bilangan prima
4. *System.out.print("Input nilai n= ");*
int a= input.nextInt();
 - Menampilkan perintah untuk memasukan nilai *n*
 - Nilai tersebut disinpan di variable *a*
5. *isPrime(a)* untuk melihat apakah nilai yang diinput adalah bil prima

- Jika prima akan menampilkan pesan “bilangan prima”
- Jika tidak akan menampilkan pesan “bukan bilangan prima”

Output:

Input nilai n= 3 3 bilangan prima	Input nilai n= 4 4 bukan bilangan prima
--------------------------------------	--

Gambar 2.1

1. 3 merupakan bilangan prima karena hanya memiliki 2 faktor
2. 4 bukan bilangan prima karena memiliki lebih dari 2 faktor

2.2 Program Biodata Mahasiswa

```

1 package pekan7_2511531017;
2
3 public class Mahasiswa_2511531017 {
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 setnama (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 getNama() {
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 }

```

Code 2.2

Nama file: Mahasiswa_2511531017

1. *Privat* bertujuan untuk melindungi data agar hanya dapat diakses melalui method tertentu.
2. *this* digunakan untuk membedakan variabel instance (*this.nim*) dan parameter method (*nim*). *setNim* akan menyimpan nilai ke variabel *nim*.
3. Method *getNim* akan mengembalikan nilai *nim*.
4. *Cetak()* akan menampilkan nilai *nim* bertipe integer dan *nama*.

```
1 package pekan7_2511531017;
2
3 public class PanggilMahasiswa_2511531017 {
4
5     public static void main(String[] args) {
6         Mahasiswa_2511531017 a= new Mahasiswa_2511531017();
7         a.setNim(23532);
8         a.setnama("Endy");
9         System.out.println(a.getNim());
10        System.out.println(a.getNama());
11        a.Cetak();
12    }
13 }
```

Code 2.3

Nama file : PanggilMahasiswa_2511531017

1. Objek dari file Mahasiswa_2511531017 akan digunakan, kita bisa mengakses semua method public yang ada di file sebelumnya.
2. *setNim(23532)* menyimpan nilai 23532 ke variabel *nim* di dalam variabel *a*, *setnama("Endy")* menyimpan nama "Endy" ke variabel *nama*
3. *getNim()* mengambil nilai dari variabel *nim* di dalam variabel *a*. *getNama()* mengambil nilai dari variabel *nama*.
4. *a.Cetak();* memanggil method *Cetak()* dari file Mahasiswa_2511531017 yang menampilkan Nim dan Nama

Output:

```
23532
Endy
Nim : 23532
Nama :Endy
```

Gambar 2.2


```

1 package pekan7_2511531017;
2
3 import java.util.Scanner;
4
5 public class PanggilMahasiswa2_2511531017 {
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_2511531017 a= new Mahasiswa_2511531017();
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 }

```

Code 2.4

Nama file: PanggilMahasiswa2_2511531017

1. Program meminta pengguna untuk memasukkan:
 - *NIM* (disimpan dalam variabel *x*)
 - *Nama* (disimpan dalam variabel *y*)
2. Karena menggunakan `nextLine()`, program bisa membaca seluruh teks, termasuk spasi.
3. Membuat objek *a* dari class `Mahasiswa_2511531017`. Melalui objek ini, kita bisa memanggil semua method.
4. `x.startsWith("25")` mengecek apakah NIM dimulai dengan angka “25”.
 - Jika ya, maka dianggap angkatan tahun 2025.
 - Program mencetak "anda angkatan 2025".
5. `x.contains("1153")` mengecek apakah NIM mengandung angka “1153” idi dalamnya, program menampilkan "Anda Mahasiswa Informatika".
6. `a.Cetak()`; memanggil method `Cetak()` dari file `Mahasiswa_2511531017` yang menampilkan Nim dan Nama

Output:

```
NIM: 2511531017
Nama: Endy Pardilian
Anda angkatan 2025
Anda Mahasiswa Informatika
Nim :2511531017
Nama :Endy Pardilian
```

Gambar 2.3

2.3 Program String1

```
1 package pekan7_2511531017;
2
3 public class String1_2511531017 {
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());
9         System.out.println(salam.toLowerCase());
10        System.out.println(salam.indexOf("salam"));
11    }
12 }
13
14 }
```

Code 2.5

Nama file: String_2511531017

1. Mendeklarasikan variabel *salam* bertipe *String* dan mengisinya dengan teks "Assalamualaikum".
2. Method *length()* mengembalikan jumlah karakter dalam string. Dalam "Assalamualaikum", jumlah hurufnya adalah 15 karakter.
3. Method *toUpperCase()* mengubah seluruh huruf menjadi huruf besar/kapital.
4. Method *toLowerCase()* mengubah seluruh huruf menjadi huruf kecil.
5. Method *indexOf("salam")* mencari posisi awal substring "salam" di dalam "Assalamualaikum". String "Assalamualaikum" hurufnya diindeks mulai dari 0. Substring "salam" dimulai pada huruf s kedua (indeks 2).

Output:

```
panjang salam adalah: 15
ASSALAMUALAIKUM
assalamualaikum
2
```

Gambar 2.4

2.4 Program String2

```
1 package pekan7_2511531017;
2
3 public class String2_2511531017 {
4
5     public static void main(String[] args) {
6         String firstName = "Endy";
7         String lastName = "Pardilian";
8         String txt1 = "Dosen\intelektual\"kampus";
9         System.out.println("Nama Lengkap: " + firstName + " " + lastName);
10        System.out.println("Nama Lengkap: " + firstName.concat(lastName));
11        System.out.println(txt1);
12        int x=10;
13        int y=20;
14        int z = x + y;
15        System.out.println("x+y= " + z);
16        String a= "10";
17        String b= "20";
18        String c = a + b;
19        System.out.println("String a + string b = " + c);
20        String v = a + y;
21        System.out.println("String a + string y = " + v);
22    }
23
24 }
```

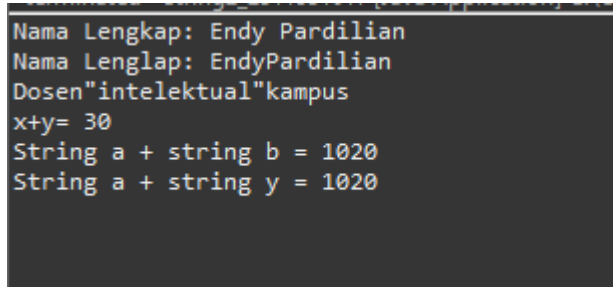
Code 2.

Nama file: String2_2511531017

1. *firstName* dan *lastName* menyimpan nama depan dan belakang.
2. *txt1* berisi teks dengan tanda kutip ganda di dalam *string*. Karakter (\") digunakan untuk menampilkan tanda kutip (")
3. "Nama Lengkap: " digabung dengan *firstName*, lalu ditambah spasi (" "), lalu *lastName*.
4. *concat()* adalah method bawaan dari *class String* untuk menggabungkan dua string. Hasilnya tidak menambahkan spasi otomatis.
5. *System.out.println(txt1);* Menampilkan isi *txt1*
6. *x* dan *y* adalah variabel bertipe *int*. $z = x + y$ menjumlahkan secara aritmetika ($10 + 20 = 30$).

7. a dan b adalah string, bukan angka. Operator “+” pada string akan menggabungkan teks. "10" + "20" menghasilkan "1020".

Output:



```
Nama Lengkap: Endy Pardilian
Nama Lenglap: EndyPardilian
Dosen"intelektual"kampus
x+y= 30
String a + string b = 1020
String a + string y = 1020
```

Gambar 2.5

BAB III

KESIMPULAN

3.1 Kesimpulan

Dari seluruh percobaan pada praktikum String dalam Java, dapat disimpulkan bahwa kelas String memiliki peranan penting dalam pemrograman Java untuk memanipulasi dan mengelola data berbentuk teks. Melalui berbagai method seperti `.length()`, `.toUpperCase()`, `.toLowerCase()`, `.indexOf()`, dan `.concat()`, programmer dapat dengan mudah melakukan operasi seperti menghitung panjang teks, mengubah huruf besar–kecil, mencari posisi substring, serta menggabungkan string. Selain itu, operator `+` juga dapat digunakan baik untuk melakukan penjumlahan bilangan maupun penggabungan string, tergantung pada tipe data yang digunakan. Dengan memahami konsep dasar dan fungsi dari class String, mahasiswa dapat lebih efektif dalam mengelola input, output, serta manipulasi teks pada program Java.

3.2 Saran

Sebagai saran, sebaiknya penjelasan materi saat praktikum bisa lebih detail dan perlahan, supaya mahasiswa yang belum terlalu paham coding bisa mengikuti dengan baik. Selain itu, dosen atau asisten praktikum sebaiknya memberikan kriteria pelaksanaan tugas yang lebih jelas, agar mahasiswa mengetahui apa saja yang harus dikerjakan dan bagaimana penilaiannya.

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].