

LAPORAN PRATIKUM
ALGORITMA PEMROGRAMAN DAN PEMROGRAMAN
STRING PADA JAVA

DISUSUN OLEH:

Rayya Syaquinah Putri Hasibuan

2511532007

DOSEN PENGAMPU:

Dr. Wahyudi, S.T. M.T

ASISTEN PRATIKUM:

Aufan Taufiqurrahman



PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga laporan praktikum ini dapat diselesaikan dengan baik. Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban pelaksanaan praktikum Algoritma Pemrograman dengan judul String pada Java. Laporan ini bertujuan untuk mendokumentasikan prosedur, hasil, serta pembahasan dari kegiatan praktikum, sekaligus sebagai sarana pembelajaran bagi penulis untuk lebih memahami materi terkait.

Penulis menyadari bahwa laporan ini masih memiliki kekurangan. Oleh karena itu, saran dan kritik yang membangun sangat diharapkan demi penyempurnaan laporan di masa mendatang. Akhir kata, penulis mengucapkan terima kasih kepada dosen pengampu, asisten pratikum, serta semua pihak yang telah membantu sehingga laporan ini dapat terselesaikan.

Padang, 11 November 2025

Rayya Syaqqinah Putri Hasibuan

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI	iii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Pratikum.....	1
1.3 Manfaat Pratikum.....	1
BAB II PEMBAHASAN	2
2.1 Deskripsi Materi Pratikum	2
2.2 Bilangan Prima.....	3
2.3 Panggil Mahasiswa	4
2.4 Panggil Mahasiswa 2	5
2.5 String.....	6
2.6 String 2	7
2.7 Mahasiswa.....	8
BAB III KESIMPULAN DAN SARAN	9
3.1 KESIMPULAN	9
3.2 SARAN	9
DAFTAR PUSTAKA.....	10

BAB I

PENDAHULUAN

1.1 Latar Belakang

String merupakan salah satu tipe data penting dalam Java yang digunakan untuk menyimpan dan memanipulasi teks. Pemahaman tentang cara kerja String sangat diperlukan karena banyak aspek pemrograman, seperti input-output, pemrosesan data, dan komunikasi antar program, melibatkan pengolahan teks. Oleh karena itu, praktikum ini dilaksanakan untuk memberikan pemahaman dasar mengenai konsep dan penggunaan String dalam pemrograman Java.

1.2 Tujuan Pratikum

Tujuan praktikum ini adalah agar dapat memahami cara mendeklarasikan dan menggunakan variabel String, menggabungkan dua string dengan operator + maupun metode concat(), serta mengenal cara menulis tanda kutip di dalam teks. Praktikum ini juga bertujuan untuk melatih membedakan antara operasi aritmetika dan penggabungan string dalam program Java.

1.3 Manfaat Pratikum

Melalui praktikum ini, kita memperoleh kemampuan untuk mengolah data teks secara efisien dan memahami pentingnya pemilihan metode yang tepat dalam penggunaan String. Selain itu, praktikum ini bermanfaat untuk meningkatkan keterampilan pemrograman, memperkuat pemahaman konsep dasar Java, serta mempersiapkan menghadapi permasalahan pemrograman yang lebih kompleks di masa mendatang.

BAB II PEMBAHASAN

2.1 Deskripsi Materi Pratikum

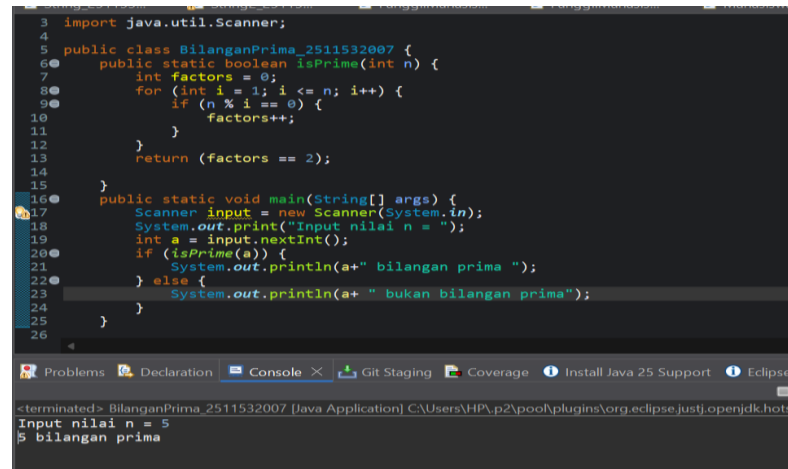
String berguna untuk menyimpan data yang dapat direpresentasikan dalam bentuk teks. Beberapa operasi yang paling sering digunakan pada string adalah memeriksa keberadaannya *length*, membangun dan menggabungkannya menggunakan operator string `+` dan `+=`, memeriksa keberadaan atau lokasi substring dengan `indexOf()` metode, atau mengekstrak substring dengan `substring()` metode tersebut. Penting untuk diketahui bahwa dalam JavaScript, string bersifat immutable. Artinya, setelah string dibuat, isinya tidak dapat diubah.

Dalam penggabungan string dalam JavaScript, penggabungan string menggabungkan dua atau lebih string menjadi satu string beserta variabelnya. Hal tersebut dapat dilakukann menggunakan operator `+`. User dapat menambahkan string kosong (" ") di antaranya untuk menciptakan spasi di antara kedua string. Cara lain untuk menggabungkan string dalam JavaScript adalah dengan menggunakan *concat()* yaitu metode yang tersedia pada setiap objek string.

Dalam JavaScript, user dapat mengonversi string menjadi huruf besar atau huruf kecil menggunakan metode *toLowerCase* pada string di JavaScript akan mengembalikan salinan string dengan huruf-hurufnya diubah menjadi huruf kecil. Angka, simbol, dan karakter lainnya tidak terpengaruh sedangkan metode *toUpperCase* pada string mengembalikan salinan string dengan huruf kapital. Angka, simbol, dan karakter lainnya tidak terpengaruh dan dapat dilihat contohnya pada sub 2.5

Method *setter* dan *getter* harus diberikan modifier *public*, karena method ini akan diakses dari luar *class*. Perbedaan method setter dengan getter terletak pada nilai kembalian, parameter, dan isi method-nya. Method setter tidak memiliki nilai kembalian void (kosong). Karena tugasnya hanya untuk mengisi data ke dalam atribut. Sedangkan method *getter* memiliki nilai kembalian sesuai dengan tipe data yang akan diambil.

2.2 Bilangan Prima



```

3 import java.util.Scanner;
4
5 public class BilanganPrima_2511532007 {
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 }

```

Console Output:

```

<terminated> BilanganPrima_2511532007 [Java Application] C:\Users\HP\p2\pools\plugins\org.eclipse.justi.openjdk.hotspot
Input nilai n = 5
5 bilangan prima

```

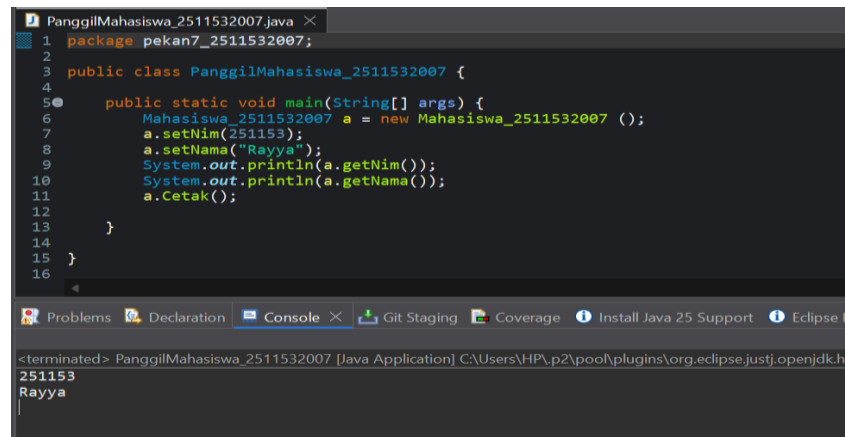
Gambar 2.2 kode program dan output "BilanganPrima"

Langkah pengerjaan:

1. Membuat *package* pekan7 dan *new class* untuk pemrograman Bilangan Prima dengan nama "BilanganPrima_2511532007".
2. Eksekusi program dimulai dari method *public static boolean isPrime(int n){* yang berfungsi akan mengembalikan *true* jika n adalah bilangan prima, dan *false* jika bukan.
3. Menginput kode `or (int i = 1; i <= n; i++) { if (n % i == 0) { factors++; return (factors == 2);` untuk mengecek jumlah faktor dari suatu bilangan n.
4. *Running* program.
5. Masukkan contoh angka, misal kita input angka 5 seperti pada gambar, maka program akan mendeteksi angka tsb bilangan prima atau tidak.

Dari output dapat kita lihat bahwa fungsi *isPrime* dapat bekerja dengan benar karena saat program dijalankan dan user memasukkan angka 5, muncul bilangan "5 bilangan prima". Program berhasil menghitung jumlah faktor dan menentukan apakah suatu bilangan termasuk prima atau tidak.

2.3 Panggil Mahasiswa



The screenshot shows the Eclipse IDE with a Java file named `PanggilMahasiswa_2511532007.java`. The code defines a package `pekan7_2511532007` and a public class `PanggilMahasiswa_2511532007`. Inside the class, there is a `main` method that creates a `Mahasiswa_2511532007` object, sets its NIM to 251153 and name to Rayya, prints the NIM and name, and calls a `Cetak` method. The console output at the bottom shows the program terminated successfully, displaying the NIM `251153` and the name `Rayya`.

```

1 package pekan7_2511532007;
2
3 public class PanggilMahasiswa_2511532007 {
4
5     public static void main(String[] args) {
6         Mahasiswa_2511532007 a = new Mahasiswa_2511532007 ();
7         a.setNim(251153);
8         a.setNama("Rayya");
9         System.out.println(a.getNim());
10        System.out.println(a.getNama());
11        a.Cetak();
12    }
13 }
14
15 }
16

```

Console Output:

```

<terminated> PanggilMahasiswa_2511532007 [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justi.openjdk.hc
251153
Rayya

```

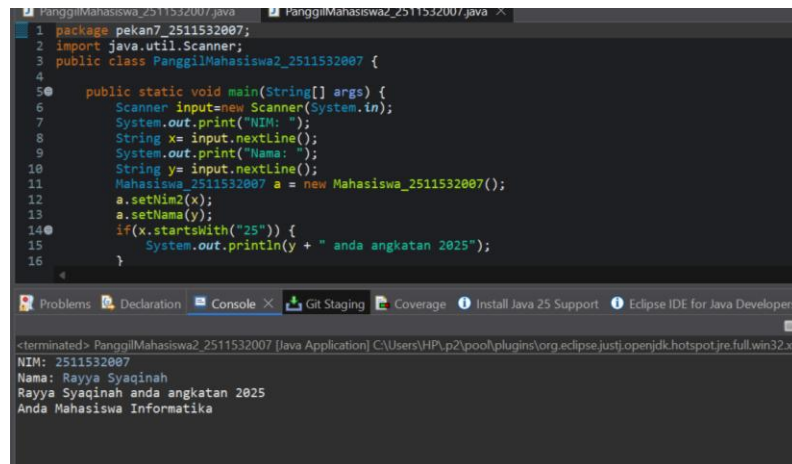
Gambar 2.3 kode program dan output "PanggilMahasiswa".

Langkah pengerjaan:

1. Membuat *package* pekan7 dan *new class* untuk pemrograman Mahasiswa dengan nama "PanggilMahasiswa_2511532007".
2. Eksekusi program dimulai dari method *public static void main(String[] args)*.
3. Menggunakan method *setter (set)* pada program untuk memberikan nilai ke variabel yang bersifat privat dengan kode program detail pada sub 2.6 yaitu "Mahasiswa".
4. Menggunakan method *getter (get)* untuk mengambil nilai dari variabel tersebut agar bisa ditampilkan diluar *class*.
5. *Run* program
6. Dari hasil kode program ini ketika di *run* akan menghasilkan output dengan data mahasiswa yang telah diinput seperti contoh pada gambar 2.3.

Untuk analisis hasil, saat program ini dijalankan, output menampilkan data mahasiswa berupa NIM dan Nama yang sebelumnya dimasukkan melalui method *set*. Dan data tersebut diambil kembali dengan method *get* dan ditampilkan melalui method *cetak()*. Hasil ini menunjukkan bahwa kode program dengan fungsi *set* dan *get* berjalan sesuai konsep yang melindungi data dari akses dan pembacaan nilai secara aman melalui method khusus.

2.4 Panggil Mahasiswa 2



```

1 package pekan7_2511532007;
2 import java.util.Scanner;
3 public class PanggilMahasiswa2_2511532007 {
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_2511532007 a = new Mahasiswa_2511532007();
12        a.setNim2(x);
13        a.setNama(y);
14        if(x.startsWith("25")) {
15            System.out.println(y + " anda angkatan 2025");
16        }
17    }
18 }

```

Problems Declaration Console Git Staging Coverage Install Java 25 Support Eclipse IDE for Java Developers

```

<terminated> PanggilMahasiswa2_2511532007 [Java Application] C:\Users\HP\AppData\Local\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.jre\bin\java.exe
NIM: 2511532007
Nama: Rayya Syaqinah
Rayya Syaqinah anda angkatan 2025
Anda Mahasiswa Informatika

```

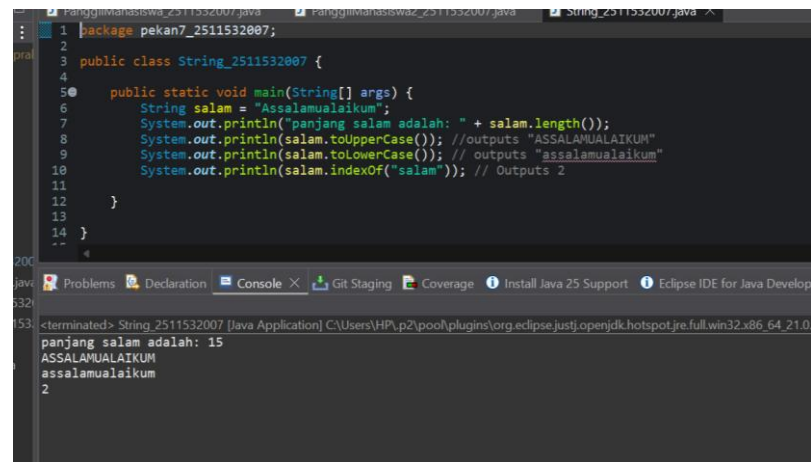
Gambar 2.4 kode program dan output "PanggilMahasiswa2".

Langkah pengerjaan:

1. Membuat *package* pekan7 dan dua *class* untuk pemrograman dengan nama "PanggilMahasiswa2_2511532007" dan "Mahasiswa_2511532007".
2. Eksekusi program dimulai dari method *public static void main(String[] args)*.
3. Menginput *import java.util.Random*;
4. Pada class "Mahasiswa_2511532007" membuat atribut nim dan nama dengan method program 2.7 "Mahasiswa_2511532007" .
5. Membuat method *main()* untuk meminta input NIM dan Nama dari user menggunakan *Scanner*.
6. Menambahkan logika kondisi *if(x.startsWith("25"))* untuk mengecek apakah NIM diawali dengan 25, yang berarti angkatan 2025. Dan gunakan *if(x.contains("1153"))* untuk mengecek apakah NIM mengandung angka 1153, yang menunjukkan Mahasiswa Informatika.
7. Run program

Program ini berhasil menjalankan kode program sesuai dengan fungsinya yaitu *setter* dan *getter*, dengan atribut *nim*, *nim2*, dan *nama* yang dilindungi dengan modifier *private* pada kode program. Method *cetak2()* juga berhasil berfungsi menampilkan hasil akhir berupa NIM dan Nama yang telah diinput.

2.5 String



```

1 package pekan7_2511532007;
2
3 public class String_2511532007 {
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()); // outputs "ASSALAMUALAIKUM"
9         System.out.println(salam.toLowerCase()); // outputs "assalamualaikum"
10        System.out.println(salam.indexOf("salam")); // Outputs 2
11    }
12 }
13
14 }
15
16 --

```

<terminated> String_2511532007 [Java Application] C:\Users\HP\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_21.0.
 panjang salam adalah: 15
 ASSALAMUALAIKUM
 assalamualaikum
 2

Gambar 2.5 kode program dan output "String_2511532007"

Langkah pengerjaan:

1. Membuat *package* "pekan7_2511532007" dan *new class* dengan nama "String_2515532007".
2. Eksekusi program dimulai dari method *public static void main(String[] args)*.
3. Mendeklarasikan variable string bernama salam dengan isi teks yang dapat dilihat pada Gambar 2.5
4. Menggunakan method bawaan string yaitu *length()* yang berfungsi menghitung panjang teks, *toUpperCase()* untuk mengubah huruf menjadi kapital, *toLowerCase()* mengubah semua huruf menjadi huruf kecil, dan *indexOf* untuk mencari posisi awal substring dalam teks.
5. *Run* program untuk melihat hasil output.

Program ini juga berhasil berfungsi sesuai dengan kode yang diinput, terkhusus untuk method bawaan string yang menghasilkan ouput yang sesuai dengan kode nya masing masing.

2.6 String 2

```

1 package pekan7_2511532007;
2
3 public class String2_2511532007 {
4
5     public static void main(String[] args) {
6         String firstName = "Syifa";
7         String lastName = "Muhassanah";
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 = integer y =" + v);
22    }
23 }
24
25
26

```

```

<terminated> String2_2511532007 [Java Application]
Nama Lengkap: Syifa Muhassanah
Nama Lengkap: SyifaMuhassanah
txt1
x + y = 30
String a + string b =1020
String a = integer y =1020

```

Gambar 2.6 adalah kode program dan hasil output dari program "String 2".

Langkah pengerjaan:

1. Membuat *package* "pekan7_2511532007" dan *new class* dengan nama "String2_2511532007".
2. Eksekusi program dimulai dari method *public static void main(String[] args)*.
3. Mendeklarasikan variabel string yaitu *firstName* diisi dengan nilai "Syifa", *lastName* diisi dengan "Muhassanah", *txt1* diisi dengan teks yang berisi tanda kutip ganda.
4. Menginput hasil penggabungan string `System.out.println("Nama Lengkap: " + firstName + " " + lastName);` yang dimana menggunakan operator `+` dan `System.out.println("Nama Lengkap: " + firstName.concat(lastName))` menggunakan method `concat()`.
5. Selain itu menginput variabel dengan tipe berbeda yaitu `int x = 10`, `int y = 20` dan menjumlahkannya menjadi `z = x + y`.
6. *Run* program untuk melihat hasilnya yang dapat juga dilihat pada gambar 2.6.

Program ini memperlihatkan berbagai bentuk operasi pada string dan tipe data numerik, penggunaan operator `+` menambahkan spasi manual sedangkan `concat()` menyambung dua string tanpa spasi.

2.7 Mahasiswa

```
package pekan7_2511532007;

public class Mahasiswa_2511532007 {

    //variabel global
    private int nim;
    private String 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 : "+nim2);
        System.out.println("nama : "+nama);
    }
    public void cetak() {
    }
}
```

Gambar 2.7 adalah kode program dari program "Mahasiswa"

Langkah pengerjaanya:

1. Membuat *package* "pekan7_2511532007" dan *new class* dengan nama "Mahasiswa_2511532007".
2. Mendefinisikan atribut, seperti *private* int nim (menyimpan NIM dalam bentuk angka), *private* String nim2 (menyimpan NIM dalam bentuk teks (digunakan untuk input berbasis string), *Private* string nama (untuk menyimpan nama mahasiswa), atribut ini dibuat *private* agar tidak bisa diakses dari luar *class*.
3. Membuat method set dan get.
4. Membuat method cetak().
5. *Run* program lain seperti PanggilMahasiswa_2511532007 untuk menguji apakah data dapat diambil dan dicetak dengan benar.

Analisis hasil, kode program ini tidak bisa di jalankan karena kode ini berfungsi untuk menyimpan data dengan aman dan menampilkannya melalui method khusus pada program lain, program ini juga fleksibel karena dapat menerima NIM dalam bentuk angka maupun teks tergantung kebutuhan input.

BAB III

KESIMPULAN DAN SARAN

3.1 KESIMPULAN

Dari hasil praktikum pemrograman Java pada materi *String* dan *Enkapsulasi* (*Setter dan Getter*), dapat disimpulkan bahwa penggunaan tipe data *String* sangat penting dalam pengolahan teks, baik untuk manipulasi huruf, penggabungan kata, maupun pencarian karakter tertentu. Selain itu, penerapan konsep *setter* dan *getter* dalam *class* seperti *Mahasiswa_2511532007* menunjukkan pentingnya prinsip *enkapsulasi* dalam melindungi data agar tidak diakses secara langsung. Seluruh program yang dijalankan berhasil menunjukkan fungsi-fungsi dasar Java, baik dalam operasi string maupun pengelolaan data objek secara aman dan terstruktur.

3.2 SARAN

Dalam melakukan praktikum selanjutnya, disarankan untuk lebih teliti dalam menuliskan sintaks dan memahami alur eksekusi program agar tidak terjadi error saat kompilasi. Pemahaman konsep dasar seperti perbedaan antara operasi aritmatika dan penggabungan string juga perlu terus diperkuat..

DAFTAR PUSTAKA

- [1] “JavaScript Strings,” *W3Schools*, [Online]. Available: https://www.w3schools.com/js/js_strings.asp. [Accessed: 12-Nov-2025].
- [2] J. Olawanle, “What is a String in JS? The JavaScript String Variable Explained,” *freeCodeCamp.org*, Feb. 3, 2023. [Online]. Available: <https://www.freecodecamp.org/news/what-is-a-string-in-javascript/>. [Accessed: 12-Nov-2025].
- [3] A. Muhardian, “Belajar Java OOP: Menggunakan Method Setter dan Getter untuk Enkapsulasi,” *PetaniKode.com*, Mar. 22, 2018. [Online]. Available: <https://www.petanikode.com/java-oop-setter-getter/>. [Accessed: 12-Nov-2025].