

LAPORAN PRATIKUM  
ALGORITMA DAN PEMROGRAMAN  
PENERAPAN STRING DAN METHOD DALAM JAVA



disusun Oleh:  
Annisa Layli Ramadhani  
2511532024  
Informatika Kelas B

Dosen Pengampu: Wahyudi. Dr.. S.T.M.T  
Asisten Pratikum: Rahmad Dwirizki Olders

DEPARTEMEN INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS ANDALAS  
TAHUN 2025

## **KATA PENGANTAR**

Puji syukur penulis panjatkan ke hadirat Allah SWT karena dengan rahmat-Nya laporan praktikum Algoritma dan Pemrograman tanggal 10 November 2025 dapat diselesaikan. Laporan ini membahas materi tentang Penerapan String dan Method dalam Java sebagai dasar penting dalam memahami pemrograman. Penulis mengucapkan terima kasih kepada dosen, asisten, dan rekan-rekan yang telah membantu selama praktikum. Penulis menyadari laporan ini masih jauh dari sempurna, sehingga kritik dan saran sangat diharapkan. Semoga laporan ini bermanfaat bagi pembaca.

Padang, 10 November 2025

Penulis

## DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB I PENDAHULUAN.....	4
1.1 Latar Belakang .....	4
1.2 Tujuan .....	5
1.3 Manfaat .....	6
BAB II PEMBAHASAN.....	7
2.1 Dasar Teori .....	7
2.1.1 String .....	7
2.1.2 Method .....	9
2.2 Langkah Kerja.....	10
2.2.1 Pembuatan <i>Package</i> dan <i>Class</i> Pekan 7 .....	10
2.2.2 Program Pertama: BilanganPrima_2511532024 .....	12
2.2.3 Program Kedua: Mahasiswa_2511532024.....	12
2.2.4 Program Ketiga: PanggilMahasiswa_2511532024.....	13
2.2.5 Program Keempat: PanggilMahasiswa2_2511532024.....	13
2.2.6 Program Kelima: String1_2511532024 .....	14
2.2.7 Program Keenam: String2_2511532024 .....	14
2.2.8 <i>Commit</i> dan <i>Push</i> ke GitHub.....	15
BAB III PENUTUP.....	17
3.1 Kesimpulan .....	17
3.2 Saran.....	17
DAFTAR PUSTAKA .....	18

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Bahasa pemrograman java merupakan salah satu bahasa yang banyak digunakan dalam pengembangan perangkat lunak karena sifatnya yang berorientasi objek, portabel, dan memiliki pustaka yang lengkap. Dalam proses pemrograman menggunakan java, salah satu konsep dasar yang sering diterapkan adalah penggunaan string dan method. Kedua konsep ini memiliki peran penting dalam membangun program yang efisien, terstruktur, dan mudah dikelola.

String dalam java digunakan untuk mempresentasikan data berupa teks atau kumpulan karakter. Hampir semua aplikasi modern memerlukan pengolahan data teks, baik untuk menampilkan pesan, menyimpan input pengguna, maupun memproses informasi dari file dan database. Oleh karena itu, pemahaman tentang cara kerja string sangat dibutuhkan agar programmer dapat melakukan berbagai operasi seperti penggabungan teks, pemisahan kata, pencarian karakter tertentu, atau pengubahan format tulisan dengan mudah. Kelas string di java menyediakan berbagai method bawaan yang membantu proses manipulasi teks agar lebih cepat dan efisien.

Sementara itu, method merupakan bagian dari konsep dasar pemrograman berorientasi objek yang berfungsi untuk mengelompokkan perintah dalam sebuah blok kode tertentu. Method memudahkan programmer dalam mengorganisasi logika program agar lebih mudah dibaca dan dikelola. Dengan adanya method, suatu proses dapat dipanggil berkali-kali tanpa harus menulis ulang kode yang sama. Hal ini mendukung prinsip *code reusability* dan *modularity*, yang menjadi dasar dalam pengembangan perangkat lunak modern.

Dalam penerapannya, penggunaan string sering kali tidak terlepas dari method. Banyak operasi terhadap string yang dilakukan melalui pemanggilan method seperti `length()`, `substring()`, `toUpperCase()`,

toLowerCase(), equals(), dan sebagainya. Melalui method-method tersebut, Java memungkinkan pengolahan data teks secara fleksibel dan efisien. Selain menggunakan method bawaan, programmer juga dapat membuat method sendiri untuk memproses string sesuai kebutuhan program, misalnya method untuk memeriksa format teks, menghitung jumlah huruf tertentu, atau membalik urutan karakter.

Penerapan string dan method tidak hanya penting dari sisi teknis, tetapi juga dari segi pembelajaran konsep dasar logika pemrograman. Pemahaman mengenai cara kerja keduanya dapat membantu mahasiswa dan pengembang memahami bagaimana data diproses, disimpan, dan dimanipulasi dalam sebuah program. Dengan menguasai penerapan string dan method, seseorang dapat menulis kode yang lebih efisien, mudah dipelihara, serta dapat dikembangkan untuk kebutuhan aplikasi yang lebih kompleks di masa depan.

Dengan demikian, pembahasan tentang penerapan string dan method dalam bahasa Java menjadi hal yang penting untuk dipelajari. Keduanya merupakan komponen fundamental yang tidak dapat dipisahkan dalam pengembangan program, terutama dalam membangun aplikasi yang memerlukan pengolahan data teks dan pemanggilan fungsi secara terstruktur.

## **1.2 Tujuan**

Tujuan dari pelaksanaan praktikum antara lain sebagai berikut:

- 1.2.1 Memahami konsep dasar string sebagai tipe data yang digunakan untuk menyimpan dan memanipulasi teks dalam bahasa java.
- 1.2.2 Menulis dan menerapkan method sendiri guna membagi logika program menjadi bagian-bagian kecil yang terorganisir.
- 1.2.3 Membangun program yang mudah dibaca, diuji, dan dikembangkan dengan memanfaatkan kombinasi string dan method secara efektif.

### **1.3 Manfaat**

Manfaat dari pelaksanaan praktikum antara lain sebagai berikut:

- 1.3.1 Mempermudah pengolahan data teks, seperti penggabungan kata, pencarian karakter, serta validasi format input pengguna.
- 1.3.2 Meningkatkan efisiensi dan kerapian program karena penggunaan method membantu menghindari penulisan kode yang berulang.
- 1.3.3 Memperkuat pemahaman terhadap struktur dan alur logika program, sehingga memudahkan dalam proses debugging dan pengembangan lebih lanjut.

## **BAB II**

### **PEMBAHASAN**

#### **2.1 Dasar Teori**

Dalam bahasa pemrograman Java, string dan method merupakan dua konsep dasar yang penting dalam pembuatan program. Keduanya digunakan hampir di setiap aplikasi, baik untuk menyimpan maupun memproses data. String berfungsi untuk menampung teks atau kumpulan karakter, sedangkan method digunakan untuk menjalankan perintah atau logika tertentu secara berulang dan terstruktur.

##### **2.1.1 String**

Dalam bahasa pemrograman Java, string merupakan salah satu komponen penting yang digunakan untuk menyimpan dan memproses data berupa teks atau kumpulan karakter. Berbeda dengan tipe data primitif seperti int, char, atau boolean, string termasuk dalam kategori objek karena merupakan bagian dari kelas string yang terdapat di dalam paket java.lang. Oleh karena itu, string memiliki berbagai method bawaan yang dapat digunakan untuk melakukan berbagai operasi manipulasi teks, seperti penggabungan kata, pengambilan sebagian teks, perbandingan antar string, hingga pengubahan format huruf menjadi kapital atau huruf kecil.

String sangat sering digunakan dalam hampir setiap program karena hampir semua aplikasi membutuhkan pengolahan data teks, misalnya untuk menampilkan pesan ke pengguna, menyimpan input dari keyboard, membaca file, atau menampilkan hasil dari proses tertentu. Pembuatan string di java cukup sederhana, yaitu dengan menuliskan teks di antara tanda kutip ganda, misalkan string nama = "Pemrograman Java";. Nilai string tersebut dapat diakses dan dimanipulasi menggunakan berbagai method bawaan yang sudah disediakan

oleh Java, sehingga programmer tidak perlu membuat fungsi tambahan untuk operasi dasar terhadap teks.

Salah satu hal penting yang perlu dipahami adalah bahwa objek string di java bersifat immutable, artinya isi dari string tidak dapat diubah setelah dibuat. Ketika dilakukan operasi seperti penggabungan atau pemotongan teks, java sebenarnya membuat objek string baru di memori dan tidak mengubah objek yang lama. Konsep ini membuat pengelolaan string di java menjadi lebih aman dan stabil, terutama saat digunakan dalam program berskala besar atau sistem yang berjalan secara bersamaan (multithreading).

Selain itu, java menyediakan berbagai method untuk mempermudah manipulasi string. Method `length()` digunakan untuk mengetahui jumlah karakter dalam string, `toUpperCase()` dan `toLowerCase()` digunakan untuk mengubah huruf menjadi kapital atau kecil. Sebagai contoh, perintah `String kata = "Java"; System.out.println(kata.toUpperCase());` akan menghasilkan output "JAVA". Melalui berbagai method tersebut, pengolahan data teks dapat dilakukan dengan cara yang lebih efisien dan fleksibel.

Dalam praktik pemrograman, string juga berperan penting dalam proses interaksi antara program dan pengguna. Misalnya, ketika pengguna memasukkan data melalui keyboard, data tersebut diterima dalam bentuk string sebelum diproses lebih lanjut. String juga digunakan dalam proses validasi input, seperti memastikan format email yang benar, memeriksa panjang kata sandi, atau menyesuaikan input dengan pola tertentu. Selain itu, dalam aplikasi berbasis web atau sistem informasi, string digunakan untuk mengolah data dari database dan menampilkannya dalam bentuk yang mudah dibaca oleh pengguna.



### 2.1.2 Method

Dalam bahasa pemrograman Java, *method* merupakan blok kode atau fungsi yang berisi sekumpulan perintah yang digunakan untuk menjalankan suatu tugas tertentu. Method berperan penting dalam membentuk struktur program agar lebih terorganisasi, efisien, dan mudah dipelihara. Dengan menggunakan method, programmer dapat memecah program yang besar menjadi bagian-bagian kecil yang memiliki fungsi khusus sehingga mempermudah proses pengembangan maupun perbaikan program. Sebuah method biasanya memiliki nama, tipe pengembalian nilai (*return type*), parameter (opsional), serta isi perintah yang diletakkan di dalam tanda kurung kurawal {}. Contoh sederhana penggunaan method adalah `public static void sapa() { System.out.println("Selamat datang di Java!"); }`, yang berfungsi untuk menampilkan pesan ke layar. Method juga dapat memiliki nilai kembalian, seperti pada contoh `public static int tambah(int a, int b) { return a + b; }`, di mana method tersebut akan mengembalikan hasil penjumlahan dari dua angka.

Dalam paradigma pemrograman berorientasi objek (Object-Oriented Programming/OOP), method juga berperan dalam mendefinisikan perilaku (*behavior*) dari suatu objek. Setiap kelas (class) di Java dapat memiliki satu atau lebih method yang berfungsi untuk menggambarkan tindakan yang dapat dilakukan oleh objek dari kelas tersebut. Sebagai contoh, dalam kelas Mahasiswa, dapat dibuat method seperti `hitungNilaiRata()` untuk menghitung nilai rata-rata mahasiswa atau `tampilkanData()` untuk menampilkan informasi mahasiswa ke layar. Dengan konsep ini, method menjadi bagian integral dari objek yang memungkinkan terjadinya interaksi antarobjek dalam program.

Selain itu, method di Java mendukung prinsip modularitas, reusabilitas, dan enkapsulasi. Modularitas berarti program dibagi ke dalam bagian-bagian kecil yang lebih mudah dikelola.

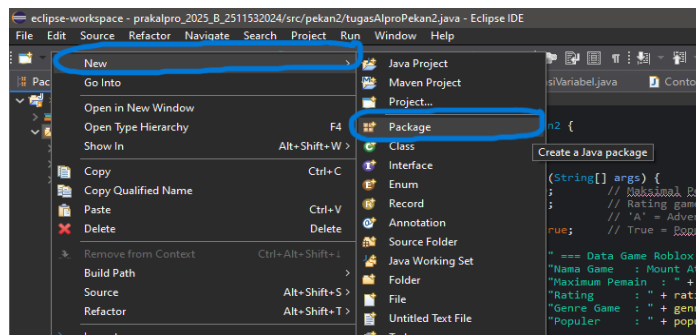
Reusabilitas memungkinkan method yang sama digunakan kembali di berbagai bagian program tanpa harus menulis ulang kode yang sama, sehingga menghemat waktu dan mengurangi potensi kesalahan. Enkapsulasi berarti detail implementasi suatu method dapat disembunyikan dari bagian program lain sehingga hanya hasil atau fungsinya saja yang digunakan. Dengan ketiga prinsip ini, penggunaan method tidak hanya membuat program lebih efisien, tetapi juga menjadikannya lebih aman dan mudah dikembangkan di masa mendatang.

Secara keseluruhan, method memiliki peran yang sangat penting dalam Java karena menjadi pondasi utama dalam pengaturan alur logika program. Dengan memahami cara kerja method, programmer dapat menulis kode yang lebih terstruktur, mudah dipahami, serta sesuai dengan prinsip-prinsip pemrograman modern. Penggunaan method juga mempermudah proses debugging, pengujian, dan pengembangan aplikasi, sehingga menjadi salah satu aspek fundamental yang harus dikuasai oleh setiap pengembang Java.

## 2.2 Langkah Kerja

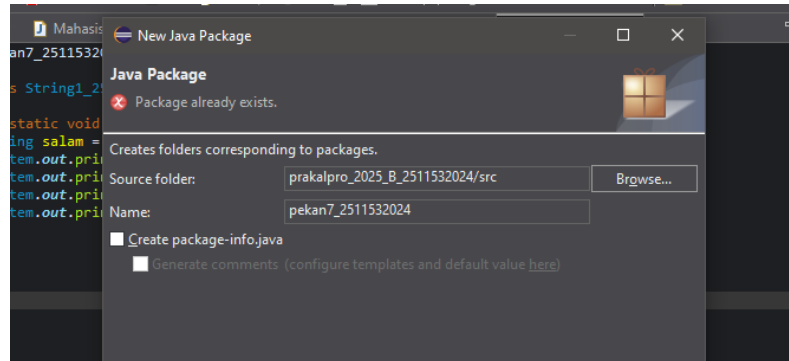
### 2.2.1 Pembuatan *Package* dan *Class* Pekan 7

1. Buka aplikasi eclipse, lalu buat package baru pada src klik kanan, lalu pilih “New” dan klik tulisan Package.



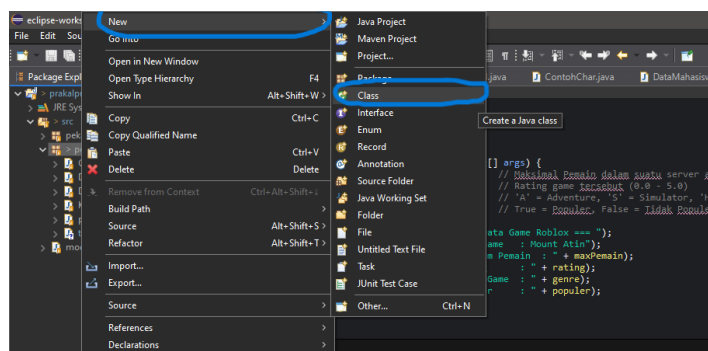
2.2.1 pembuatan package

2. Kemudian buat nama packagenya tanpa spasi, huruf kapital, karakter khusus. Beri nama “pekan7\_2511532024”. Lalu klik finish.



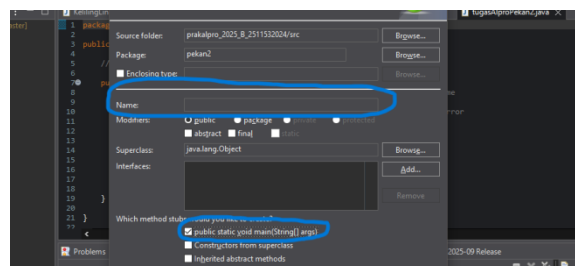
2.2.2 pembuatan package

3. Ketika package sudah jadi. Klik kanan pada bagian “pekan7\_2511532024”. Lalu klik “new” dan pilih bagian “class” untuk memulai membuat program.



2.2.3 pembuatan class

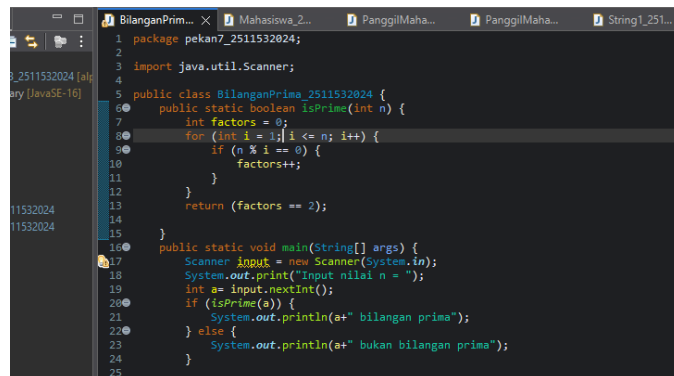
4. Buat nama class program yang akan di buat pada bagian nama spasi dan menggunakan huruf kapital pada awal kata, lalu klik bagian “public static void main(string[] args)”. Kemudian klik “finish”.



2.2.4 pembuatan class

### 2.2.2 Program Pertama: BilanganPrima\_2511532024

1. Masukkan sintaks berikut.



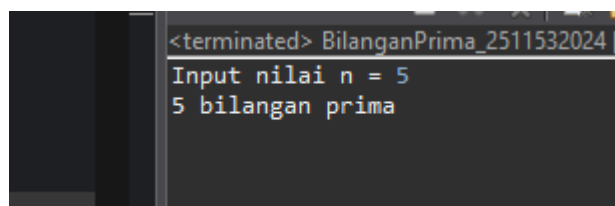
```

1 package pekan7_2511532024;
2
3 import java.util.Scanner;
4
5 public class BilanganPrima_2511532024 {
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 }

```

2.2.5 sintaks bilangan prima

2. Lalu klik bagian “run” untuk melihat hasil sintaks yang ada di inputkan. Lalu akan menghasilkan output seperti berikut.



```

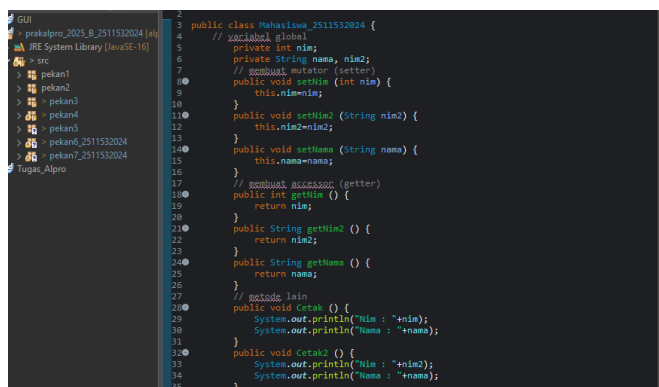
<terminated> BilanganPrima_2511532024 [
Input nilai n = 5
5 bilangan prima

```

2.2.6 running kode dam output sintaks

### 2.2.3 Program Kedua: Mahasiswa\_2511532024

1. Ulangi pembuatan class seperti program sebelumnya dan tersebut “Mahasiswa\_2511532024”. Lalu masukkan sintaks seperti berikut.



```

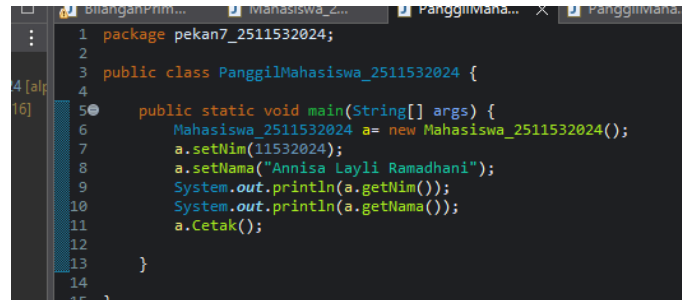
1 package pekan7_2511532024;
2
3 import java.util.Scanner;
4
5 public class Mahasiswa_2511532024 {
6     // atribut global
7     private int nim;
8     private String nama;
9     // constructor
10    public void setnim(int nim) {
11        this.nim = nim;
12    }
13    public void setnama(String nama) {
14        this.nama = nama;
15    }
16    // method accessor (getter)
17    public int getnim() {
18        return nim;
19    }
20    public String getnim2() {
21        return nim;
22    }
23    public String getnama() {
24        return nama;
25    }
26    // method lain
27    public void cetak() {
28        System.out.println("nim : "+nim);
29        System.out.println("nama : "+nama);
30    }
31    public void cetak2() {
32        System.out.println("nim : "+nim);
33        System.out.println("nama : "+nama);
34    }
35 }

```

2.2.7 sintaks mahasiswa

### 2.2.4 Program Ketiga: PanggilMahasiswa\_2511532024

1. Ulangi pembuatan class seperti program sebelumnya dan beri nama class tersebut “PanggilMahasiswa\_2511532024”. Lalu masukkan sintaks seperti berikut.



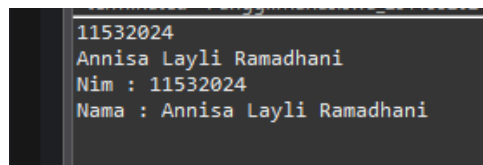
```

1 package pekan7_2511532024;
2
3 public class PanggilMahasiswa_2511532024 {
4
5     public static void main(String[] args) {
6         Mahasiswa_2511532024 a= new Mahasiswa_2511532024();
7         a.setNim(11532024);
8         a.setNama("Annisa Layli Ramadhani");
9         System.out.println(a.getNim());
10        System.out.println(a.getNama());
11        a.Cetak();
12    }
13 }
14
15

```

2.2.8 sintaks panggil mahasiswa

2. Klik “run” untuk melihat hasil sintaks yang sudah dibuat.



```

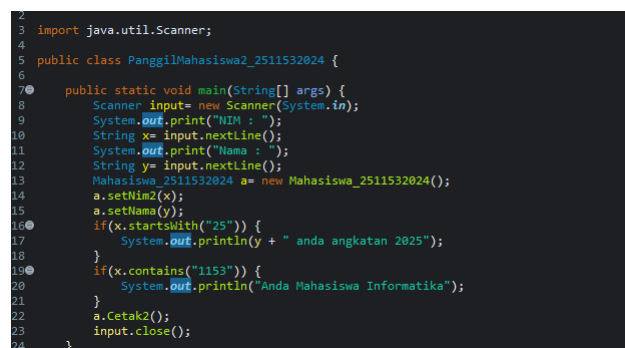
11532024
Annisa Layli Ramadhani
Nim : 11532024
Nama : Annisa Layli Ramadhani

```

2.2.9 running kode dan output sintaks

### 2.2.5 Program Keempat: PanggilMahasiswa2\_2511532024

1. Ulangi pembuatan class seperti program sebelumnya dan beri nama class tersebut “PanggilMahasiswa2\_2511532024”. Lalu masukkan sintaks seperti berikut.



```

2
3 import java.util.Scanner;
4
5 public class PanggilMahasiswa2_2511532024 {
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_2511532024 a= new Mahasiswa_2511532024();
14        a.setNim(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 }

```

2.2.10 sintaks panggil mahasiswa 2

2. Klik “run” untuk melihat hasil sintaks yang sudah dibuat.

```
<terminated> PanggilMahasiswa2_2511532024 [Java Application]
NIM : 2511532024
Nama : Annisa Layli Ramadhani
Annisa Layli Ramadhani anda angkatan 2025
Anda Mahasiswa Informatika
Nim : 2511532024
Nama : Annisa Layli Ramadhani
```

2.2.11 running kode dan output sintaks

## 2.2.6 Program Kelima: String1\_2511532024

1. Ulangi pembuatan class seperti program sebelumnya dan beri nama class tersebut “String1\_2511532024”. Lalu masukkan sintaks seperti berikut.

```
1 package pekan7_2511532024;
2
3 public class String1_2511532024 {
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 }
```

2.2.12 sintaks string 1

2. Klik “run” untuk melihat hasil sintaks yang sudah dibuat.

```
<terminated> String1_2511532024 [Java Application]
panjang salam adalah: 15
ASSALAMUALAIKUM
assalamualaikum
2
```

2.2.13 running kode dan output

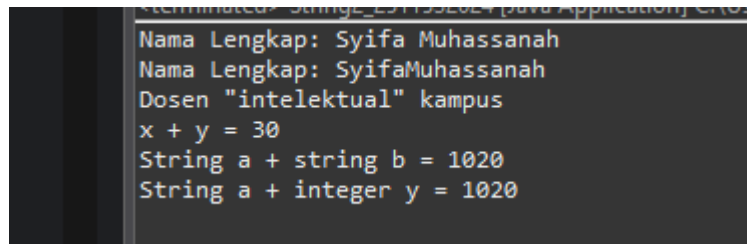
## 2.2.7 Program Keenam: String2\_2511532024

1. Ulangi pembuatan class seperti program sebelumnya dan beri nama class tersebut “String2\_2511532024”. Lalu masukkan sintaks seperti berikut.

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

2.2.14 sintaks string 2

2. Klik “run” untuk melihat hasil sintaks yang sudah dibuat.



```

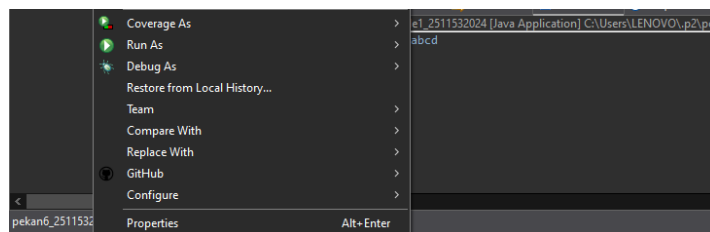
Nama Lengkap: Syifa Muhassanah
Nama Lengkap: SyifaMuhassanah
Dosen "intelektual" kampus
x + y = 30
String a + string b = 1020
String a + integer y = 1020

```

2.2.15 running kode dan output

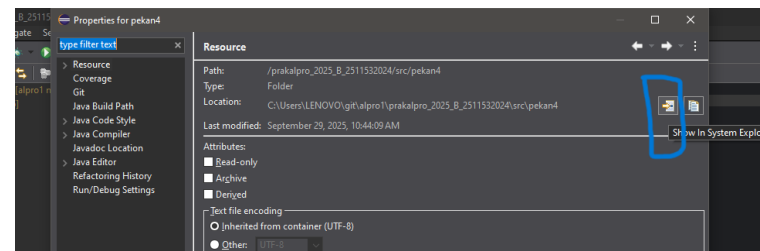
## 2.2.8 Commit dan Push ke GitHub

1. Klik kanan pada bagian “pekan6\_2511532024”. Kemudian pilih bagian “properties”.



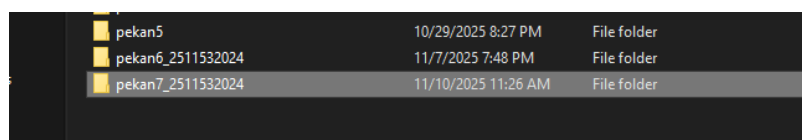
2.2.16 proses commit dan push ke GitHub

2. Setelah itu keluar opsi properties. Kemudian klik “Show In System Explorer”.



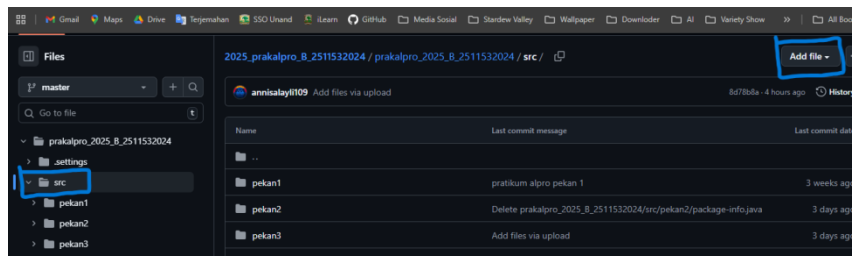
2.2.17 proses commit dan push ke github

3. Setelah itu akan terbuka folder file tersebut di file explorer.



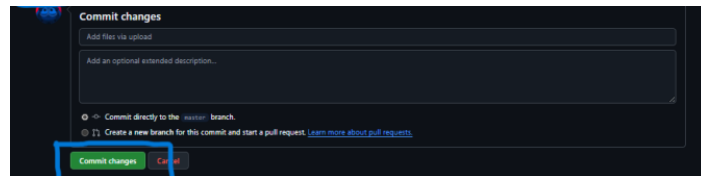
2.2.18 proses commit dan push ke GitHub

4. Kemudian buka akun GitHub dan buka pada bagian src. Dan klik pada bagian “add file”. Dan pilih “upload file”.



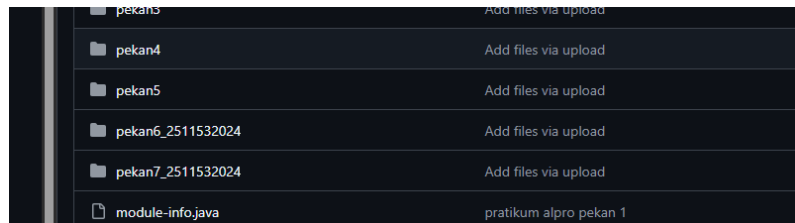
2.2.19 proses commit dan push ke GitHub

- Setelah itu tarik file “pekan7\_2511532024” dari file explorer ke bagian upload file dan kemudian tunggu loading file diunggah ke GitHub. Dan setelah itu klik “Commit changes”.



2.2.10 proses commit dan push ke GitHub

- Kemudian cek akun GitHub untuk melihat apakah file yang diupload sudah masuk atau belum. Jika sudah maka upload sudah selesai.



2.2.21 proses commit dan push ke GitHub



## **BAB III**

### **PENUTUP**

#### **3.1 Kesimpulan**

Berdasarkan pembahasan yang telah dijelaskan, dapat disimpulkan bahwa String dan Method merupakan dua komponen dasar yang memiliki peran penting dalam bahasa pemrograman Java. String berfungsi untuk menyimpan dan memanipulasi data berbentuk teks, sedangkan Method berperan sebagai wadah logika atau perintah yang digunakan untuk menjalankan tugas tertentu dalam program. Melalui penggunaan String, programmer dapat mengolah berbagai bentuk data teks dengan mudah, sementara penggunaan Method memungkinkan program ditulis secara lebih efisien, terstruktur, dan mudah dipelihara. Kedua konsep ini saling berkaitan, karena objek String di Java memiliki berbagai method bawaan yang dapat digunakan untuk memproses teks secara langsung. Dengan memahami dan menerapkan konsep String dan Method dengan baik, seorang programmer dapat menghasilkan program yang tidak hanya fungsional tetapi juga berkualitas tinggi serta sesuai dengan prinsip-prinsip pemrograman berorientasi objek.

#### **3.2 Saran**

Disarankan agar pemrogram terus memperdalam pemahaman tentang konsep String dan Method dalam Java melalui latihan langsung dan membaca dokumentasi resmi. Dengan sering berlatih dan menerapkan kedua konsep tersebut dalam program sederhana, kemampuan logika dan penulisan kode akan meningkat sehingga program yang dibuat menjadi lebih efisien dan terstruktur.

## DAFTAR PUSTAKA

- [1] Oracle, “*Class String (Java Platform SE 21)*,” Oracle Documentation, 2024. [Online]. Available: <https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/lang/String.html>
- [2] Oracle, “*Methods in Java Tutorial*,” Oracle Java Tutorials, 2024. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/javaOO/methods.html>