

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN  
METHOD DAN STRING JAVA**



**Disusun oleh:**

Ghinada Fathanawafa Algma  
2511533008

**Dosen Pengampu:**  
Dr. Wahyudi S.T. M.T

**Asisten Praktikum:**  
Rahmad Dwirizki Olders

**DEPARTEMEN INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS ANDALAS**

**2025**

## KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan ini yang berjudul “Pengenalan Method dan String dalam Bahasa Pemrograman Java” dengan baik dan tepat waktu.

Laporan ini disusun sebagai bagian dari pembelajaran algoritma dan pemrograman, khususnya untuk memahami konsep dasar dan penerapan method serta string dalam pengembangan perangkat lunak. Kedua topik ini merupakan fondasi penting dalam pemrograman berorientasi objek, di mana *method* berfungsi sebagai blok kode yang dapat digunakan kembali untuk menjalankan tugas tertentu, dan *string* sebagai tipe data yang digunakan untuk memanipulasi teks.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat dan menjadi referensi yang berguna bagi pembaca dalam memahami dasar-dasar Java secara lebih mendalam.

Akhir kata, penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan dan bantuan dalam penyusunan laporan ini.

Padang, 10 November 2025

Penulis

## DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI .....	ii
BAB I PENDAHULUAN .....	1
1.1    Latar Belakang .....	1
1.2    Tujuan.....	1
1.3    Manfaat .....	2
BAB II PEMBAHASAN .....	3
2.1    Dasar Teori .....	3
2.1.1 Method .....	3
2.1.2 String.....	3
2.2    Langkah Kerja.....	4
2.2.1 Membuat <i>packsge</i> dan <i>class</i> .....	4
2.2.2 Program 1 : Bilangan Prima.....	6
2.2.3 Program 2 : Mahasiswa.....	7
2.2.4 Program 3 : Panggil Mahasiswa.....	8
2.2.5 Program 4 : Panggil Mahasiswa 2.....	9
2.2.6 Program 5 : String 1 .....	10
2.2.7 Program 6 : String 2 .....	11
BAB III PENUTUP .....	13
3.1 Kesimpulan .....	13
DAFTAR PUSTAKA.....	14

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Perkembangan teknologi informasi yang pesat telah menjadikan pemrograman sebagai keterampilan fundamental dalam berbagai bidang, mulai dari pendidikan, bisnis, hingga industri kreatif. Salah satu bahasa pemrograman yang banyak digunakan secara global adalah Java, yang dikenal karena sifatnya yang *object-oriented*, portabel, dan aman.

Dalam pemrograman Java, *method* dan *string* merupakan dua komponen dasar yang sangat penting. *Method* memungkinkan pengembang untuk mengorganisasi kode secara modular dan efisien, memudahkan pemeliharaan serta pengembangan program yang kompleks. Sementara itu, *string* digunakan untuk memanipulasi data teks, yang sangat umum dalam interaksi pengguna, pengolahan data, dan komunikasi antar sistem.

Pemahaman yang mendalam tentang cara kerja *method* dan *string* tidak hanya membantu dalam menulis kode yang bersih dan terstruktur, tetapi juga menjadi fondasi dalam membangun aplikasi yang responsif dan fungsional. Oleh karena itu, pembahasan mengenai kedua topik ini menjadi sangat relevan dalam proses pembelajaran pemrograman Java, khususnya bagi pelajar dan mahasiswa yang sedang mendalami dunia rekayasa perangkat lunak.

Makalah ini disusun untuk memberikan penjelasan teoritis dan praktis mengenai konsep *method* dan *string* dalam Java, serta menunjukkan penerapannya melalui contoh kode dan studi kasus sederhana.

### **1.2 Tujuan**

1. Menjelaskan konsep dasar *method* dalam Java, termasuk struktur, jenis, dan cara penggunaannya dalam program.

2. Menguraikan fungsi dan manipulasi string sebagai tipe data penting dalam pengolahan teks di Java.
3. Memberikan contoh implementasi method dan string dalam bentuk kode program sederhana dan aplikatif.
4. Meningkatkan pemahaman mahasiswa terhadap prinsip pemrograman berorientasi objek melalui studi kasus yang relevan.
5. Mendorong kemampuan analisis dan problem solving dalam menyusun program Java yang efisien dan terstruktur.

### **1.3 Manfaat**

1. Menambah pemahaman tentang konsep dasar method dan string pada Java.
2. Meningkatkan kemampuan dalam menyusun program yang efisien dan terstruktur.

## **BAB II**

### **PEMBAHASAN**

#### **2.1 Dasar Teori**

##### **2.1.1 Method**

Method adalah blok kode yang dirancang untuk menjalankan tugas tertentu dan dapat dipanggil berulang kali dalam program. Method membantu modularisasi program, meningkatkan keterbacaan, dan memudahkan pemeliharaan.

Struktur Method terdiri dari tipe pengambilan (*return type*), nama method, parameter (jika ada), dan isi kode.

Jenis method:

1. Void method: tidak mengembalikan nilai.
2. Return method: mengembalikan nilai tertentu.
3. Static mothod: dapat dipanggil tanpa membuat objek.
4. Instance method: harus dipanggil melalui objek.

##### **2.1.2 String**

String adalah tipe data yang digunakan untuk menyimpan dan memanipulasi teks. Di Java, string merupakan objek dari kelas `java.lang.String`.

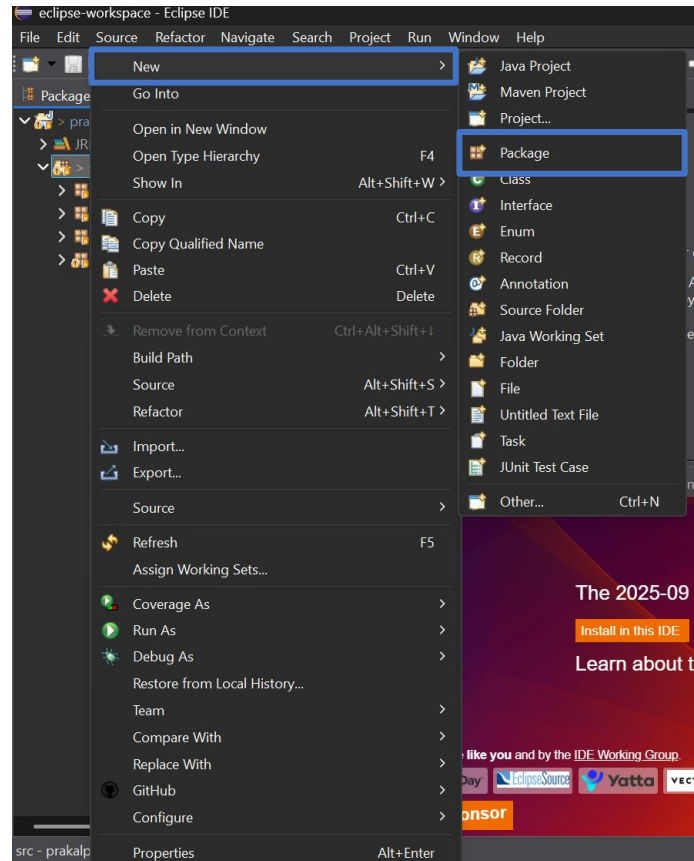
Metode umum dalam string:

1. `length()`: menghitung Panjang string.
2. `charAt(int index)`: mengambil karakter pada indeks tertentu.
3. `substring(int start, int end)`: mengambil bagian dari string.
4. `equals(String another)`: membandingkan dua string.
5. `toUpperCase()`, `toLowerCase()`: mengubah huruf menjadi kapital atau kecil.

## 2.2 Langkah Kerja

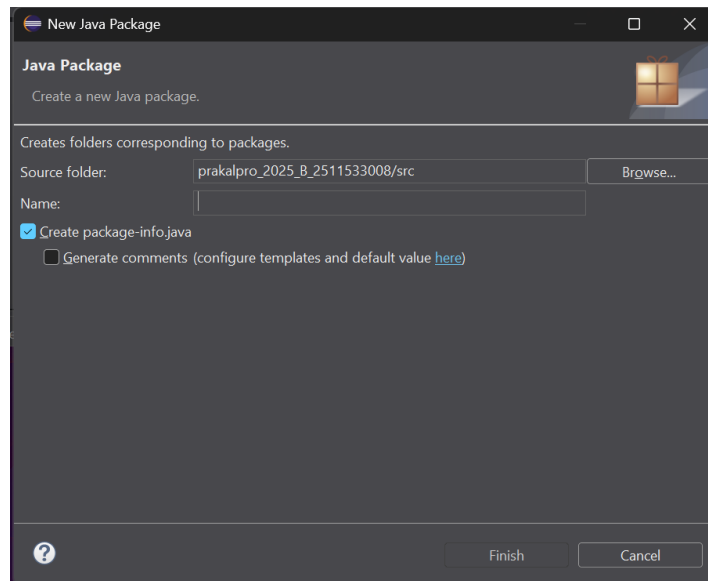
### 2.2.1 Membuat *packsge* dan *class*

1. Sebelum membuat atau menuliskan kode program pertama tam akita harus membuat *package*. Dengan cara klik kanan pada *src*, lalu pilih *New* dan pilih *package*.



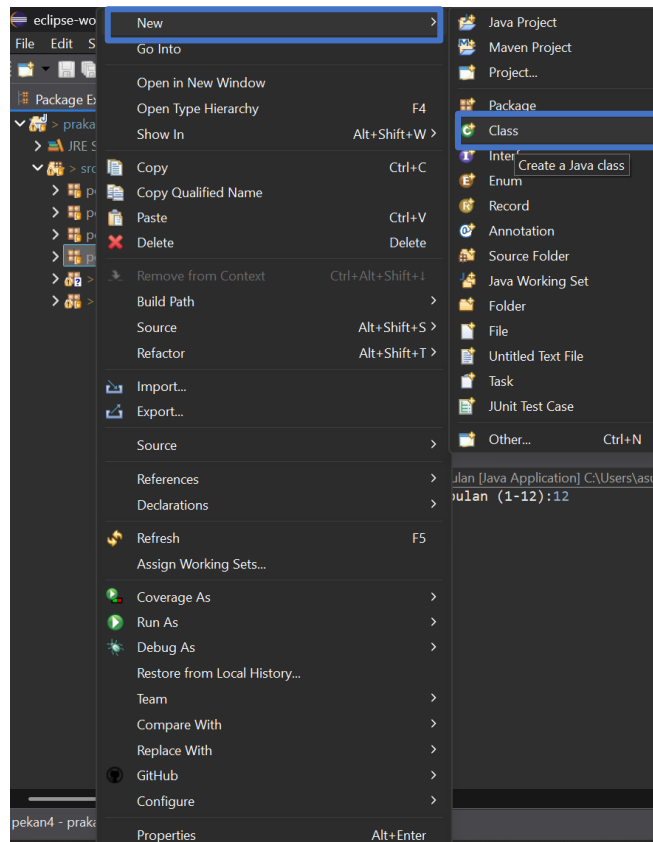
Gambar 2.1 membuat package

2. Setelah mengklik *package* kita akan diminta untuk mengisi nama *package* pada menu “Java Package”.



Gambar 2.2 membuat nama package

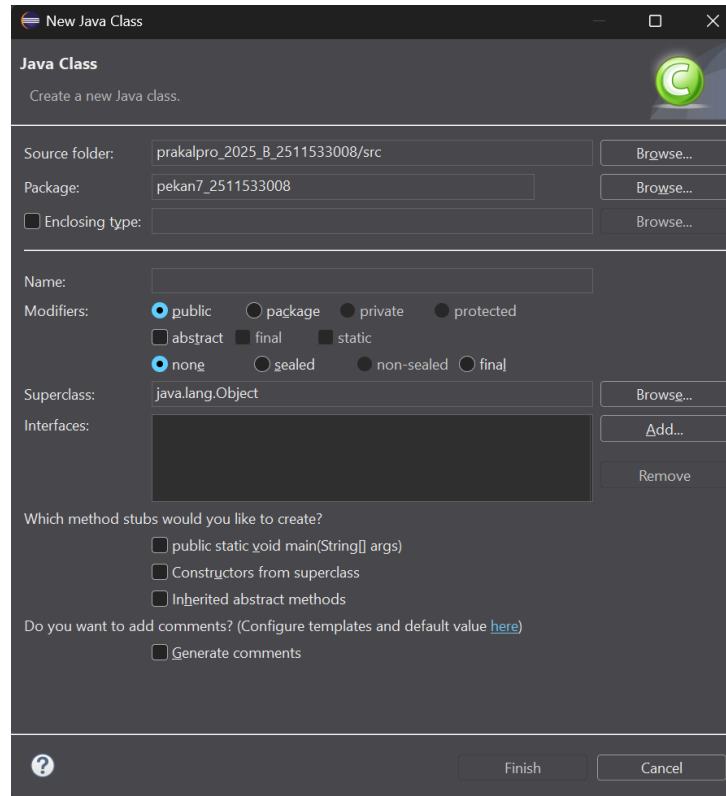
- Setelah selesai membuat *package* klik kanan pada *package* yang telah dibuat lalu pilih *New* dan pilih *class*.



Gambar 2.3 membuat class



- Setelah itu akan muncul menu “Java Class” di sana kita diminta untuk mengisi nama *class*.



Gambar 2.4 membuat nama class

### 2.2.2 Program 1 : Bilangan Prima

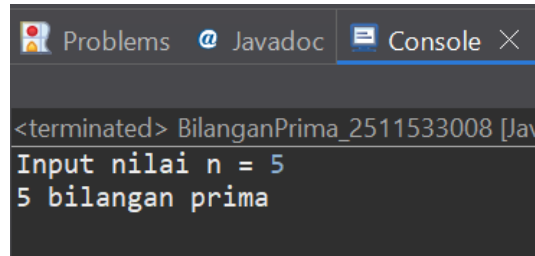
- Setelah membuat *class* kita akan menuliskan kode program berikut ini

```
1 package pekan7_2511533008;
2
3 import java.util.Scanner;
4
5 public class BilanganPrima_2511533008 {
6
7     public static boolean isPrime(int n) {
8         int factors = 0;
9         for (int i = 1; i <= n; i++) {
10             if (n % i == 0) {
11                 factors++;
12             }
13         }
14         return (factors == 2);
15     }
16
17     public static void main(String[] args) {
18         Scanner input = new Scanner(System.in);
19         System.out.print("Input nilai n = ");
20         int a = input.nextInt();
21         if (isPrime(a)) {
22             System.out.println(a + " bilangan prima");
23         } else {
24             System.out.println(a + " bukan bilangan prima");
25         }
26     }
27 }
28
29 }
```

Kode program 2.1 program bilangan prima

Kode program ini digunakan untuk menentukan apakah nilai yang diinputkan merupakan bilangan prima atau tidak.

2. Setelah memasukkan kode program diatas klik run untuk menjalankan program tersebut dan output akan tampil pada menu console.



```
<terminated> BilanganPrima_2511533008 [Java...]  
Input nilai n = 5  
5 bilangan prima
```

*Gambar 2.5 output bilangan prima*

Pada saat mengklik tombol run pada menu console kita akan diminta untuk menginputkan nilai yang akan diuji, disini saya menggunakan 5 sebagai nilai uji dan hasilnya adalah 5 merupakan bilangan prima.

### **2.2.3 Program 2 : Mahasiswa**

Program yang diberi nama mahasiswa ini adalah program method. Dimana program ini berisi method – method yang bisa dipanggil pada program program selanjutnya di *class* yang berbeda. Berikut merupakan kode program method yang di tulis pada program kedua ini.

```

1 package pekan7_2511533008;
2
3 public class Mahasiswa_2511533008 {
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

```

*Kode program 2.2 mahasiswa*

Dalam program kedua ini tidak ada output yang dihasilkan karena hanya berisi method-method saja.

### 2.2.4 Program 3 : Panggil Mahasiswa

Pada program 3 ini kita akan memanggil atau menggunakan method yang telah kita buat pada program 2 sebelumnya.

1. Pada *class* Panggil Mahasiswa buatlah kode program berikut ini

```

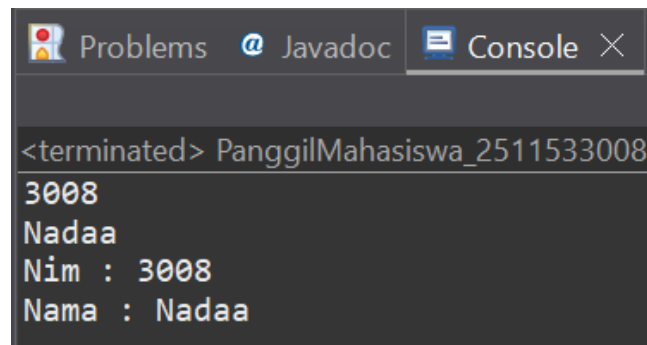
1 package pekan7_2511533008;
2
3 public class PanggilMahasiswa_2511533008 {
4
5     public static void main(String[] args) {
6         Mahasiswa_2511533008 a = new Mahasiswa_2511533008();
7         a.setNim(3008);
8         a.setNama("Nadaa");
9         System.out.println(a.getNim());
10        System.out.println(a.getNama());
11        a.Cetak();
12    }
13 }
14
15 }
16

```

*Kode program 2.3 panggil mahasiswa*

Pada kode program terlihat bahwa kita memasukkan method yang telah kita buat pada *class* sebelumnya. Nama *class* sebelumnya juga di tulis yaitu Mahasiswa\_2511533008 untuk memanggil method yang ada pada *class* tersebut.

2. Untuk menjalankan program ini klik run dan akan muncul output dari program berikut pada menu console.



```

<terminated> PanggilMahasiswa_2511533008
3008
Nadaa
Nim : 3008
Nama : Nadaa

```

*Gambar 2.6 output panggil mahasiswa*

### 2.2.5 Program 4 : Panggil Mahasiswa 2

Selain menggunakan cara pada program 3 kita bisa menggunakan cara berikut ini untuk memanggil method yang telah dibuat pada *class* sebelumnya.

1. Buat *class* baru dan tuliskan kode program berikut pada *class* tersebut.

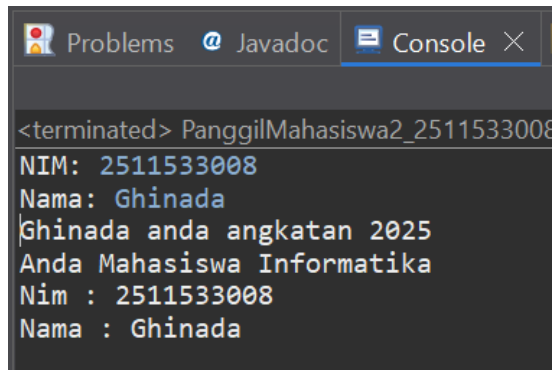
```

1 package pekan7_2511533008;
2
3 import java.util.Scanner;
4
5 public class PanggilMahasiswa2_2511533008 {
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_2511533008 a = new Mahasiswa_2511533008();
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 }
28

```

*Kode program 2.4 panggil mahasiswa 2*

2. Untuk menjalankan kode program berikut klik run dan akan muncul output pada menu console.



```

<terminated> PanggilMahasiswa2_2511533008
NIM: 2511533008
Nama: Ghinada
Ghinada anda angkatan 2025
Anda Mahasiswa Informatika
Nim : 2511533008
Nama : Ghinada

```

*Gambar 2.7 output panggil mahasiswa 2*

Pada output program ini pengguna akan diminta untuk menginputkan nim serta nama dan barulah hasil akhirnya di cetak.

## 2.2.6 Program 5 : String 1

1. Pada program ini buat *class* baru dan tulis kode program berikut.

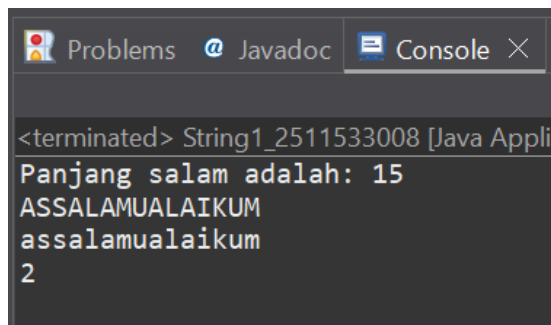
```

1 package pekan7_2511533008;
2
3 public class String1_2511533008 {
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"));
11    }
12 }
13
14 }
15

```

*Kode program 2.5 string1*

2. Untuk menjalankan kode program klik run dan akan muncul output pada menu console.



```

<terminated> String1_2511533008 [Java Appli
Panjang salam adalah: 15
ASSALAMUALAIKUM
assalamualaikum
2

```

*Gambar 2.8 output string1*

### 2.2.7 Program 6 : String 2

1. Buat *class* baru dan masukkan kode program berikut.

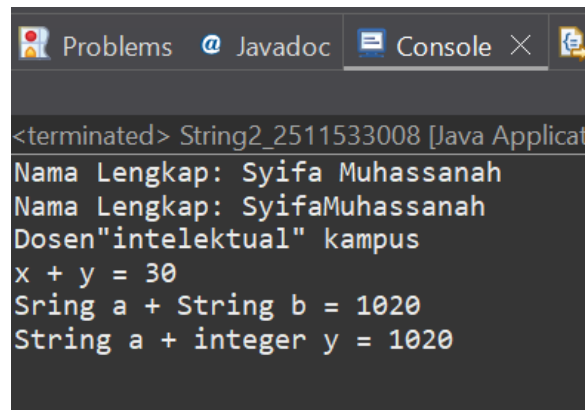
```

1 package pekan7_2511533008;
2
3 public class String2_2511533008 {
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

```

*Kode program 2.6 string2*

2. Untuk menjalankan kode program tersebut klik run dan output akan muncul pada menu console.

A screenshot of an IDE's console window. The window has a title bar with tabs for 'Problems', 'Javadoc', and 'Console'. The 'Console' tab is active. The output text is as follows:

```
<terminated> String2_2511533008 [Java Applicat  
Nama Lengkap: Syifa Muhassanah  
Nama Lengkap: SyifaMuhassanah  
Dosen"intelektual" kampus  
x + y = 30  
String a + String b = 1020  
String a + integer y = 1020
```

*Gambar 2.9 output string2*

## **BAB III**

### **PENUTUP**

#### **3.1 Kesimpulan**

Dari praktikum ini dapat disimpulkan bahwa pemahaman tentang method sangat meningkatkan modularitas dan keterbacaan program karena memungkinkan pemisahan tugas ke dalam blok kode yang dapat dipanggil ulang; pemilihan antara *static* dan *instance* method menentukan cara pemanggilan dan pengelolaan state objek, sedangkan perbedaan *void* dan *return type* mempengaruhi cara data diproses dan diuji. Penggunaan parameter dan overloading menambah fleksibilitas desain API, dan pemahaman bahwa String adalah objek dengan banyak metode bawaan (seperti pengambilan substring, perbandingan isi dengan equals, serta perubahan kapitalisasi) penting untuk menghindari bug logika, terutama perbedaan antara perbandingan referensi (==) dan perbandingan isi. Secara keseluruhan, penerapan method dan manipulasi String yang benar berkontribusi pada pembuatan program yang lebih efisien, mudah dipelihara, dan lebih tangguh terhadap kesalahan input teks; oleh karena itu disarankan melanjutkan latihan dengan studi kasus yang lebih kompleks untuk memperdalam kemampuan praktis dan debugging.



## DAFTAR PUSTAKA

- [1] H. Schildt, Java: A Beginner's Guide, 7th ed. New York, NY, USA: McGraw-Hill Education, 2018.
- [2] J. Bloch, Effective Java, 3rd ed. Boston, MA, USA: Addison-Wesley, 2018.
- [3] H. Schildt, Java: The Complete Reference, 11th ed. New York, NY, USA: McGraw-Hill Education, 2018.
- [4] K. Sierra and B. Bates, Head First Java, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2005.
- [5] B. Eckel, Thinking in Java, 4th ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2006.
- [6] Oracle, "The Java™ Tutorials — Methods," Oracle. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/javaOO/methods.html>. . [Accessed: 10-Nov-2025].
- [7] Oracle, "String (Java Platform SE 11)," Oracle. [Online]. Available: <https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/String.html>. . [Accessed: 10-Nov-2025].
- [8] A. Gosling, B. Joy, and G. Steele, The Java Language Specification, Java SE 11 Edition. Boston, MA, USA: Addison-Wesley, 2019.
- [9] GeeksforGeeks, "Java String Class," GeeksforGeeks. [Online]. Available: <https://www.geeksforgeeks.org/java-lang-string-class-java/>. . [Accessed: 10-Nov-2025].