

**LAPORAN PRAKTIKUM ALGORITMA
PEMROGRAMAN**

**“METHOD DAN STRING DALAM PEMROGRAMAN
JAVA”**

DISUSUN OLEH:

NAYSHA APRILIA RIZKI

2511531014

DOSEN PENGAMPU: Dr. WAHYUDI, S.T,M.T

**ASISTEN PRAKTIKUM: MUHAMMAD ZAKI AL
HAFIZ**



**DEPARTEMEN INFORMATIKA FAKULTAS
TEKNOLOGI INFORMASI PROGRAM STUDI
INFORMATIKA UNIVERSITAS ANDALAS**

2025

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya, Laporan praktikum ini dapat diselesaikan dengan baik. Laporan praktikum ini disusun sebagai salah satu tugas dalam rangka memenuhi tugas yang diberikan oleh dosen dan juga asisten praktikum.

Padang, 16 November 2025

Naysha Aprilia Rizki

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi yang pesat dalam beberapa dekade terakhir telah memberikan dampak besar terhadap berbagai aspek kehidupan manusia, termasuk di bidang pendidikan, industri, dan komunikasi. Salah satu pilar utama dalam perkembangan teknologi tersebut adalah **pemrograman komputer**. Di antara banyak bahasa pemrograman yang ada, **Java** merupakan salah satu bahasa yang sangat populer dan banyak digunakan di berbagai belahan dunia.

Java adalah bahasa pemrograman tingkat tinggi yang bersifat **object-oriented**, dikembangkan oleh Sun Microsystems pada tahun 1995, dan kini dikelola oleh Oracle. Keunggulan utama Java terletak pada kemampuannya untuk berjalan di berbagai platform (platform-independent) berkat konsep **Write Once, Run Anywhere (WORA)** yang dimilikinya. Hal ini memungkinkan program yang dibuat dalam Java untuk dijalankan di berbagai sistem operasi tanpa perlu diubah ulang.

Selain itu, Java memiliki sintaks yang relatif mudah dipahami bagi pemula, dokumentasi yang lengkap, dan komunitas pengguna yang besar, sehingga sangat mendukung proses belajar dan pengembangan perangkat lunak. Java juga menjadi bahasa utama dalam pengembangan berbagai aplikasi, seperti aplikasi desktop, mobile (Android), web, hingga sistem enterprise berskala besar.

Dengan melihat pentingnya Java dalam dunia pemrograman dan aplikasinya yang luas, sangatlah relevan untuk mempelajari dasar-dasar bahasa pemrograman Java, baik dari segi teori maupun praktik. Oleh karena itu, laporan ini disusun untuk memberikan pemahaman yang lebih mendalam tentang bahasa Java, struktur dasarnya, serta penerapan konsep pemrograman berorientasi objek dalam bahasa tersebut

1.2 Tujuan

1. Membantu mahasiswa memahami konsep dasar pemrograman Java melalui penerapan langsung.
2. Melatih kemampuan menulis, mengompilasi, dan mengeksekusi program dengan mengikuti aturan sintaksis Java.
3. Meningkatkan keterampilan dalam memecahkan masalah (problem solving) dengan pendekatan algoritmik

1.3 Manfaat Praktikum

1. Mahasiswa dapat memahami konsep dasar pemrograman secara praktis, bukan hanya teori.
2. Melatih keterampilan logika dan analisis dalam menyusun serta mengeksekusi kode.
3. Memberikan pengalaman langsung dalam menggunakan bahasa Java sebagai salah satu bahasa pemrograman populer.
4. Menjadi bekal awal untuk memahami materi pemrograman yang lebih kompleks pada praktikum berikutnya.
5. Membiasakan mahasiswa dengan proses debugging dan pemecahan masalah dalam pemrograman.

BAB II

PEMBAHASAN

2.1 Pengertian method dan string

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:

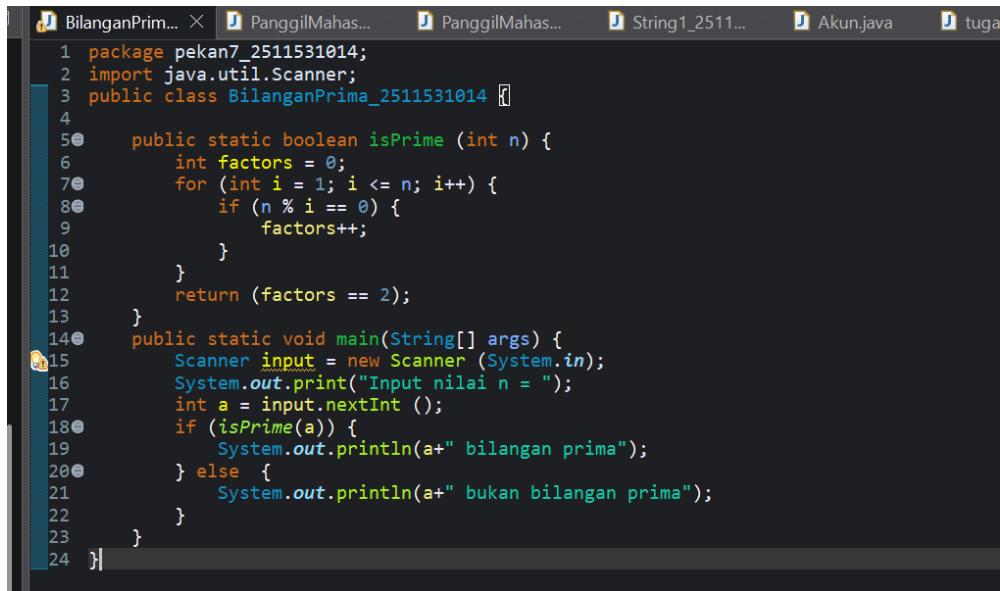
1. Void method: tidak mengembalikan nilai
2. Return method: mengembalikan nilai tertentu.
3. Static method: dapat dipanggil tanpa membuat objek.
4. Instance method: harus dipanggil melalui objek.

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.

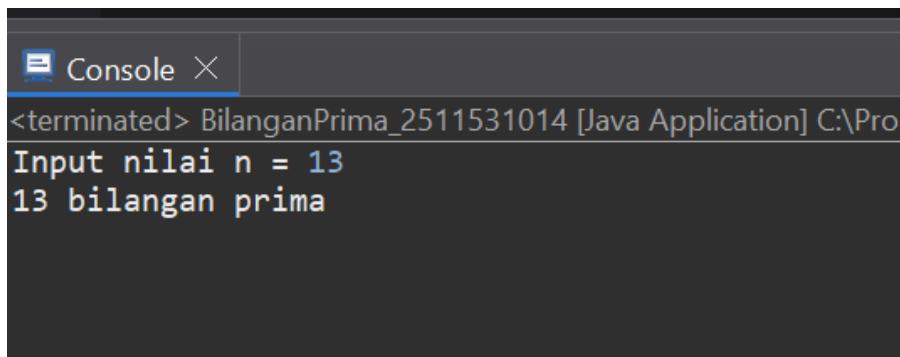
2,2 Praktikum Pekan 7

2.1.1 Bilangan Prima_2511531014



```
1 package pekan7_2511531014;
2 import java.util.Scanner;
3 public class BilanganPrima_2511531014 {
4
5     public static boolean isPrime (int n) {
6         int factors = 0;
7         for (int i = 1; i <= n; i++) {
8             if (n % i == 0) {
9                 factors++;
10            }
11        }
12        return (factors == 2);
13    }
14    public static void main(String[] args) {
15        Scanner input = new Scanner (System.in);
16        System.out.print("Input nilai n = ");
17        int a = input.nextInt ();
18        if (isPrime(a)) {
19            System.out.println(a+" bilangan prima");
20        } else {
21            System.out.println(a+" bukan bilangan prima");
22        }
23    }
24 }
```

Pada pemrograman kali ini digunakan untuk memastikan apakah bilangan tersebut merupakan bilangan prima atau bukan, pada pemograman ini menggunakan method, pada Method *isPrime(int n)* untuk melihat apakah n adalah bilangan prima atau bukan dan method *main()* sebagai Static method yang memanggil tanpa membuat objek, Output dibawah ini



```
Console ×
<terminated> BilanganPrima_2511531014 [Java Application] C:\Projek\BilanganPrima_2511531014
Input nilai n = 13
13 bilangan prima
```

```
<terminated> BilanganPrima_2511531014 [Java]
Input nilai n = 8
8 bukan bilangan prima
```

2.1.2 Mahasiswa_2511531014

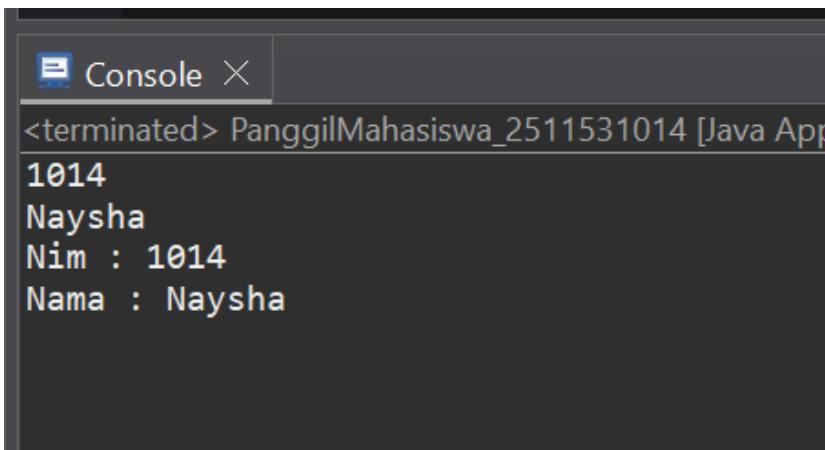
```
1 package pekan7_2511531014;
2
3 public class Mahasiswa_2511531014 {
4     //variabel global
5     private int nim;
6     private String nama,nim2;
7     //membuat mutator (setter)
8     public void setNim (int nim) {
9         this.nim=nim;
10    }
11    public void setNim2 (String nim2){
12        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 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}
```

Pada pemrograman kali ini juga merupakan contoh dari pemograman yang menggunakan gabungan antara method dan string, terdapat string dibagian ‘nim, nim2,nama’ sedangkan method pda bagian ‘getNim(), getNim2(), getName()’. Kode diatas hanya untuk menciptakan class utama yang menyimpan data mahasiswa, dan tidak memiliki output.

2.1.3 Panggil Mahasiswa 1 _2511531014

```
1 package pekan7_2511531014;
2
3 public class PanggilMahasiswa_2511531014 {
4
5     public static void main(String[] args) {
6         Mahasiswa_2511531014 a= new Mahasiswa_2511531014();
7         a.setNim(1014);
8         a.setNama("Naysha");
9         System.out.println(a.getNim());
10        System.out.println(a.getNama());
11        a.Cetak();
12
13    }
14
15 }
16
```

Pada pemrograman kali ini, merupakan kelanjutan dari pemograman Mahasiswa_2511531014, disini data yang kita simpan di class utama akan dipanggil kembali yang menghasilkan output seperti dibawah



```
Console <terminated> PanggilMahasiswa_2511531014 [Java App]
1014
Naysha
Nim : 1014
Nama : Naysha
```

2.1.4 Panggil Mahasiswa 2 2511531014

```
1 package pekan7_2511531014;
2 import java.util.Scanner;
3 public class PanggilMahasiswa2_2511531014 {
4     public static void main(String []args) {
5         Scanner input = new Scanner(System.in);
6         System.out.print("NIM: ");
7         String x = input.nextLine();
8         System.out.print("Nama: ");
9         String y = input.nextLine();
10        Mahasiswa_2511531014 a = new Mahasiswa_2511531014();
11        a.setNim2(x);
12        a.setNama(y);
13        if (x.startsWith("25")) {
14            System.out.println(y + " anda angkatan 2025");
15        }
16        if (x.contains("1153")) {
17            System.out.println("Anda Mahasiswa Informatika");
18        }
19        a.Cetak2();
20        input.close();
21    }
22 }
23
```

Pada pemrograman kali ini juga merupakan pemanggilan data yang disimpan pada Mahasiswa_2511531014, yang menyatakan jika terdapat ‘25’ pada NIM maka menghasilkan output ‘Anda Angkatan 2025’ dan juga menyatakan ‘1153’ pada NIM maka menghasilkan output ‘Anda Mahasiswa Informatika’, Output seperti dibawah

```
<terminated> PanggilMahasiswa2_2511531014 [Java Application] C:\NIM: 2511531014Nama: naysha aprilia naysha aprilia anda angkatan 2025Anda Mahasiswa InformatikaNim : 2511531014Nama : naysha aprilia
```

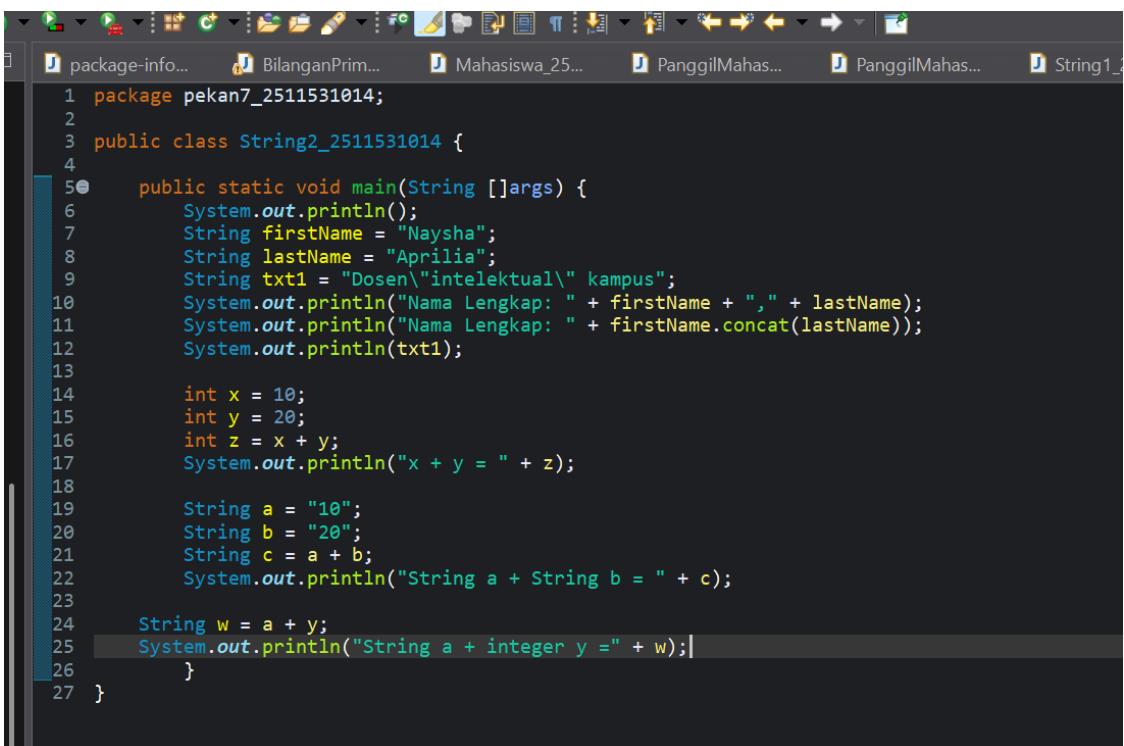
2.1.5 String1

```
1 package pekan7_2511531014;
2
3 public class String1_2511531014 {
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()); //output "ASSALAMUALAIKUM"
9         System.out.println(salam.toLowerCase()); //output "assalamualaikum"
10        System.out.println(salam.indexOf("salam")); //output 2
11    }
12
13
14 }
```

Pada pemrograman kali ini menggunakan String ‘toUpperCase()’ yang berguna untuk menghasilkan output dengan huruf besar, String ‘toLowerCase()’ yang berguna untuk menghasilkan output dengan huruf kecil, dan String ‘indexOf’ yang berguna untuk output kedua nya. Output sebagai berikut.

```
1 package pekan7_2511531014;
2
3 public class String2 {
4     public static void main(String []args) {
5         System.out.println();
6         String firstName = "Naysha";
7         String lastName = "Aprilia";
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
13        int x = 10;
14        int y = 20;
15        int z = x + y;
16        System.out.println("x + y = " + z);
17
18        String a = "10";
19        String b = "20";
20        String c = a + b;
21        System.out.println("String a + String b = " + c);
22
23        String w = a + y;
24        System.out.println("String a + integer y =" + w);
25    }
26 }
```

2.1.6 String2



```
1 package pekan7_2511531014;
2
3 public class String2_2511531014 {
4     public static void main(String []args) {
5         System.out.println();
6         String firstName = "Naysha";
7         String lastName = "Aprilia";
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
13        int x = 10;
14        int y = 20;
15        int z = x + y;
16        System.out.println("x + y = " + z);
17
18        String a = "10";
19        String b = "20";
20        String c = a + b;
21        System.out.println("String a + String b = " + c);
22
23        String w = a + y;
24        System.out.println("String a + integer y =" + w);
25    }
26 }
```

Pada pemrograman kali ini menggunakan String untuk keseluruhan mulai dari fisrt name, last name, txt1 dan menghasilkan output sebagai berikut

```
Console X
<terminated> String2_2511531014 [Java Application] C:\Program
Nama Lengkap: Naysha,Aprilia
Nama Lengkap: NayshaAprilia
Dosen"intelektual" kampus
x + y = 30
String a + String b = 1020
String a + integer y =1020
```

BAB III

PENUTUP

3.1. Kesimpulan

String adalah **tipe data non-primitive** yang digunakan untuk menyimpan teks atau kumpulan karakter. Di dalam kelas, String sering digunakan untuk atribut seperti nama, alamat, atau kode tertentu. Variabel bertipe String dapat diakses dan diubah melalui setter dan getter untuk menjaga prinsip *encapsulation*.

Method adalah **blok kode** yang berfungsi menjalankan suatu proses tertentu. Ada dua jenis method penting dalam OOP: **Setter (Mutator)** untuk mengubah atau mengisi nilai atribut. **Getter (Accessor)** untuk mengambil atau membaca nilai atribut. Method dapat mengembalikan nilai (seperti `getNama()`) atau tidak mengembalikan nilai (`void` seperti `setNama()`). Method membantu membuat kode lebih rapi, reusable, dan terstruktur.

Hubungan String dan Method, Method seperti `getNama()` dan `setNama()` bekerja dengan **mengambil atau mengubah data bertipe String**. Ini menunjukkan bagaimana method berfungsi sebagai “pengontrol” untuk mengakses variabel String yang bersifat private.

3.2. Saran

Pemahaman konsep dasar perulangan sangat penting bagi pemrogram Java pemula. Disarankan agar mahasiswa mempraktikkan berbagai contoh pemograman menggunakan method dan string untuk memahami perilakunya secara mendalam. Dan diharapkan agar mahasiswa lebih teliti dalam memahami syntax agar meminimalisir kesalahan penulisan kode. Terutama saat menggunakan Method dan String, pastikan kondisi pengontrol dan nilai variabel benar, agar program tidak mengalami Error.

DAFTAR PUSTAKA

Sumber Daring (website):

- [1] Deitel, P. J., & Deitel, H. M. (2017). *Java: How to Program* (10th Edition). Pearson Education.
- [2] A. Gosling, B. Joy, and G. Steele, The Java Language Specification, Java SE 11 Edition. Boston, MA, USA: Addison-Wesley, 2019.
- [3] Oracle, “The Java™ Tutorials — Methods,” Oracle. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/javaOO/methods.html>. . [Accessed: 10-Nov 2025].