

LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN  
PENGUNAAN METHOD STRING DAN METHOD



Oleh: Afif Naufal Zahran

NIM: 2511533009

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

FAKULTAS TEKNOLOGI INFORMASI

DEPARTEMEN INFORMATIKA

UNIVERSITAS ANDALAS

## **Kata Pengantar**

Pedoman ini disusun sebagai rujukan resmi bagi mahasiswa Departemen Informatika dalam penyusunan laporan praktikum pada mata kuliah *Pemrograman Dasar dengan Java*. Dokumen ini tidak hanya memberikan gambaran umum mengenai format penulisan, tetapi juga menguraikan secara rinci sistematika laporan, tata cara penyajian isi, serta contoh penulisan kode program yang dilengkapi dengan referensi ilmiah. Melalui panduan ini, mahasiswa diharapkan mampu menyusun laporan yang tidak sekadar memenuhi aspek administratif, tetapi juga mencerminkan ketelitian, keteraturan, dan penerapan kaidah penulisan akademik pada tingkat dasar. Dengan demikian, laporan praktikum yang dihasilkan dapat berfungsi sebagai media pembelajaran, dokumentasi kegiatan, sekaligus sarana untuk melatih keterampilan menulis ilmiah yang akan bermanfaat dalam jenjang studi selanjutnya.

Padang, 2025

Tim Penyusun

## DAFTAR ISI

BAB I .....	
1.1 Latar Belakang.....	
1.2 Tujuan .....	
1.3 Manfaat .....	
BAB II .....	
2,1 Teori.....	
2,2 Program.....	
BAB III.....	
3,1 Kesimpulan.....	
3,2 Saran .....	

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Pada bahasa pemrograman manapun terdapat tipe data yang merepresentasikan teks, yaitu serangkaian karakter seperti huruf, angka, simbol, dan spasi. Dalam kode, string selalu diapit oleh tanda kutip, seperti tanda kutip ganda (“!@# teks123”) atau tunggal (‘ ’). Walau begitu, string sendiri tidak termasuk ke dalam tipe data primitif. Melainkan sebuah *class* yang disediakan oleh *java* dan memiliki method untuk memanipulasi value dari *string*.

### **1.2 Tujuan**

Tujuan dari laporan praktikum ini agar pembaca dan penulis dapat mendalami dan memahami tentang *String Method*, dan *Method* beserta implementasinya di bahasa pemrograman *Java*.

### **1.3 Manfaat**

Manfaat dari praktikum tentang *string* dan *method* agar pembaca mendapatkan pengetahuan tentang *string* dan *method*. Manfaat untuk penulis agar dapat memahami *String*, *Method*, *Setter*, dan *Getter* pada bahasa pemrograman *Java*.

## BAB II

### PEMBAHASAN

#### 2,1 Teori

*String* adalah tipe data dalam bahasa pemrograman yang digunakan untuk merepresentasikan teks, yakni serangkaian karakter seperti huruf, angka, simbol, dan spasi. *String* sendiri merupakan urutan karakter, bisa berupa kata, kalimat atau bahkan urutan karakter acak seperti “8923 asd9asd9298 \*@#”. Di java sendiri *String* tidak termasuk ke tipe data primitif, melainkan sebuah *class* yang disediakan oleh *Java* yang dapat dideklarasikan dengan (*String* s = “halo dunia”). *String Class* sendiri menyediakan *method* yang dapat digunakan untuk memanipulasi isi dari *string*.

Tidak hanya *string* pada pertemuan ke-7 kali ini kita juga mempelajari sedikit dasar OOP(*Object-Oriented Programming*) dimana kita membuat suatu *class* yang memiliki *attribute*, *getter method*, dan *setter method*.

#### 2,2 Program

Pada pertemuan pekan ke-7 terdapat 6 program yang dibuat yakni  
BilanganPrima\_2511533009.java, Mahasiswa\_\_2511533009.java,  
PanggilMahasiswa\_2511533009.java, PanggilMahasiswa2\_2511533009.java,  
String1\_2511533009.java, String2\_2511533009.java.

a. BilanganPrima\_2511533009.java

```
1 package pekan7_2511533009;
2 import java.util.Scanner;
3
4 public class BilanganPrima_2511533009 {
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        // TODO Auto-generated method stub
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 }
```

Gambar 2.1

Pada gambar 2.1 program ini membuat sebuah method dimana method tersebut me-*return value boolean*. Method ini berfungsi untuk melakukan pengecekan apakah suatu nilai dari parameter *method isPrime* adalah bilangan prima. Bilangan tersebut akan dicek apakah bilangan tersebut habis dibagi dengan 1 dan dirinya sendiri.

b. Mahasiswa\_2511533009.java

```
3 public class Mahasiswa_2511533009 {
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
18    //membuat accessor (getter)
19    public int getNim() {
20        return nim;
21    }
22    public String getNim2() {
23        return nim2;
24    }
25    public String getNama() {
26        return nama;
27    }
28    //metode lain
29    public void Cetak() {
30        System.out.println("Nim : "+nim);
31        System.out.println("Nama : "+nama);
32    }
33    public void Cetak2() {
34        System.out.println("Nim : "+nim2);
35        System.out.println("Nama : "+nama);
36    }
```

Gambar 2.2

Pada gambar 2.2 program ini menerapkan konsep sederhana OOP(*Object-Oriented Programming*) dimana kita membuat suatu *class* yang memiliki *attributes* seperti nim, nama, dan nim2. Setelah itu, pada *class* ini terdapat beberapa *method* *getter*, dan *setter* yang berfungsi untuk meangambil dan mengubah *attributes* dari *class* tersebut.

c. PanggilMahasiswa\_2511533009.java

```
1 package pekan7_2511533009;
2
3 public class PanggilMahasiswa_2511533009 {
4     public static void main(String[] args) {
5         Mahasiswa_2511533009 a= new Mahasiswa_2511533009();
6         a.setNim(23532);
7         a.setNama("Rahmat");
8         System.out.println(a.getNim());
9         System.out.println(a.getNama());
10        a.Cetak();
11    }
12
13 }
```

Gambar 2.3

Pada gambar 2.3 program ini menerapkan konsep sederhana OOP(*Object-Oriented Programming*) dimana kita membuat suatu *Object* dari *class* yang telah kita buat yakni Mahasiswa\_2511533009. Ketika kita membuat suatu *Object* kita harus menggunakan kata kunci *new* lalu nama class dan di akhiri dengan *new()*. Setelah itu, kita dapat memanggil *method* *setNim()* dan *setNama()* untuk mengubah *attribute* *nim* dan *nama* pada *object* yang kita buat. Lalu kita dapat memanggil *attribute* tersebut dengan menggunakan *method* *getNim()* dan *getNama()*.



d. PanggilMahasiswa2\_2511533009.java

```
1 package pekan7_2511533009;
2 import java.util.Scanner;
3 public class PanggilMahasiswa2_2511533009 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         Scanner input = new Scanner(System.in);
8         System.out.print("NIM: ");
9         String x= input.nextLine();
10        System.out.print("Nama: ");
11        String y= input.nextLine();
12        Mahasiswa_2511533009 a= new Mahasiswa_2511533009();
13        a.setNim2(x);
14        a.setNama(y);
15        if(x.startsWith("25")) {
16            System.out.println(y + " anda angkatan 2025");
17        }
18        if(x.contains("1153")) {
19            System.out.println("Anda Mahasiswa Informatika");
20        }
21        a.Cetak2();
22        input.close();
23    }
24
25 }
```

Gambar 2.4

Pada program kali ini memiliki kesamaan seperti program sebelumnya, namun value dari *attribute* diubah dengan *input* dari pengguna. Lalu pada percabang *if* dilakukan pengecekan apakah suatu string diawali dengan “25” dengan menggunakan method *startsWith()* dan pengecekan apakah suatu string memiliki “1153” pada value string tersebut dengan menggunakan method *contains()*. Setelah itu kita memanggil method *Cetak2()* untuk menampilkan *attributes* dari *nim2* dan *nama*.

e. String1\_2511533009.java

```
1 package pekan7_2511533009;
2
3 public class String1_2511533009 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         String salam = "Assalamualaikum";
8         System.out.println("panjang salam adalah: " + salam.length());
9         System.out.println(salam.toUpperCase()); // Outputs "ASSALAMUALAIKUM"
10        System.out.println(salam.toLowerCase()); // Outputs "assalamualaikum"
11        System.out.println(salam.indexOf("salam")); // Outputs 2
12    }
13
14 }
```

Gambar 2.5

Pada gambar 2.6 program menggunakan beberapa *method* pada *string* seperti *length()* berfungsi untuk menampilkan panjang dari suatu string, lalu *toUpperCase()* untuk mengubah seluruh huruf menjadi kapital lalu *toLowerCase()* kebalikan dari fungsi *toUpperCase()* dan *indexOf()* untuk menampilkan pada index beberapa suatu string

f. String2\_2511533009.java

```
1 package pekan7_2511533009;
2 import java.util.Scanner;
3 public class String2_2511533009 {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         Scanner input = new Scanner(System.in);
8         System.out.print("Nama Depan: ");
9         String firstName = input.nextLine();
10        System.out.print("Nama Belakang: ");
11        String lastName = input.nextLine();
12        String txt1 = "Dosen\intelektual\ kampus";
13        System.out.println("Nama Lengkap: "+firstName + " " + lastName);
14        System.out.println("Nama Lengkap: "+firstName.concat(lastName));
15        System.out.println(txt1);
16        int x = 10;
17        int y = 20;
18        int z = x + y;
19        System.out.println("x+ y= " + z);
20        String a = "10";
21        String b = "20";
22        String c = a + b;
23        System.out.println("String a + string b =" + c);
24        String v = a + y;
25        System.out.println("String a + string y = " + v);
26    }
27
28 }
```

Gambar 2.6

Pada gambar 2.6 program menggunakan fungsi *concat()* dimana kita dapat menggabungkan menggabungkan dua string menjadi satu. String juga dapat digabungkan tanpa menggunakan fungsi *concat()* yakni dengan menambahkan dua string menggunakan karakter "+". Jika suatu string hanya terdiri dari angka dan dijumlahkan maka hasilnya adalah gabungan dari string tersebut seperti "10" + "20" maka hasilnya adalah "1020".

## **BAB III**

### **PENUTUPAN**

#### **3,1 Kesimpulan**

String dan Object pada bahasa pemrograman *Java* sangat penting karena merupakan hal yang sangat fundamental pada bahasa pemrograman terutama bahasa pemrograman *Java*.

#### **3,2 Saran**

Laporan praktikum ini masih memiliki kekurangan. Karena itu, penulis sangat terbuka terhadap saran dan kritikan agar dapat meningkatkan kualitas laporan ini dan laporan-laporan selanjutnya.