

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

**Disusun Oleh:**

Dinda Amelia

(2511531020)

**Dosen Pengampu:**

Dr. Wahyudi, S.T, M.T

**Asisten Praktikum:**

Muhammad Zaki Al Hafiz



**DEPARTEMEN INFORMATIKA**

**FAKULTAS TEKNOLOGI INFORMASI**

**UNIVERSITAS ANDALAS**

**PADANG 2025**

## KATA PENGANTAR

Segala puji bagi Allah SWT yang telah memberikan kesempatan serta kemudahan kepada penulis untuk dapat menyelesaikan Laporan Praktikum pada mata kuliah Algoritma Pemrograman, sehingga Laporan Praktikum ini dapat dikumpulkan dengan tepat waktu. Atas rahmat dan karunianya Laporan Praktikum dapat terselesaikan dengan baik. Sholawat serta salam juga penulis sampaikan kepada baginda tercinta yaitu Nabi Muhammad SAW. yang kita nantikan syafa'atnya di akhirat nanti. Laporan Praktikum ini bertujuan untuk menambah wawasan para pembaca untuk lebih memperdalam ilmu yang ada pada makalah ini.

Laporan pratikum ini disusun sebagai bentuk penanggung jawaban atas pelaksanaan kegiatan pratikum mata kuliah algoritma pemograman yang membahas tentang Method dan string pada java. Melalui pemograman ini penulis dapat memperkuat pemahaman mengenai cara membuat method sendiri, memanggil method dari class lain, serta mengelola data teks menggunakan berbagai method bawaan seperti `length()`, `substring()`, `toUpperCase()`, dan lainnya.

Dalam penyusunan Laporan Praktikum ini, penulis mengalami banyak kesulitan dan penulis menyadari bahwa Laporan Praktikum ini masih jauh dari kesempurnaan. Untuk itu, penulis sangat mengharapkan kritik dan saran yang membangun demi kesempurnaan pada Laporan Praktikum ini.

Padang, 12 November 2025

Penulis

## DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI .....	ii
BAB I.....	1
PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Tujuan.....	1
1.3 Manfaat Praktikum .....	1
BAB 2 .....	2
PEMBAHASAN.....	2
2.1 Class pertama ‘Bilangan Prima_2511531020’ .....	3
2.1.1 Kode Program .....	3
2.1.2 Langkah Kerja.....	3
2.1.3 Analisis Hasil .....	4
2.2 Class kedua ‘Mahasiswa_2511531020’ .....	4
2.2.1 Kode Program .....	4
2.2.2 Langkah Kerja.....	5
2.2.3 Analisis Hasil .....	5
2.3 Class ketiga ‘PanggilMahasiswa_2511531020’ .....	6
2.3.1 Kode Program .....	6
2.3.2 Langkah Kerja.....	6
2.3.3 Analisis Hasil .....	6
2.4 lass keempat ‘PanggilMahasiswa2_2511531020’ .....	7
2.4.1 Kode Program .....	7
2.4.2 Langkah Kerja.....	7
2.4.3 Analisis Hasil .....	8
2.5 Class kelima ‘PanggilMahasiswa3_2511531020’ .....	8
2.5.1 Kode Program .....	8
2.5.2 Langkah Kerja.....	9
2.5.3 Analisis Hasil .....	10
2.6 Class keenam ‘String1_2511531020’ .....	10
2.6.1 Kode Program .....	10
2.6.2 Langkah Kerja.....	10
2.6.3 Analisis Hasil .....	10
2.7 Class ketujuh ‘String2_2511531020’ .....	11
2.7.1 Kode Program .....	11
2.7.2 Langkah Kerja.....	11

2.7.3 Analisis Hasil .....	11
BAB III .....	13
KESIMPULAN.....	13
DAFTAR PUSTAKA.....	14

## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

program Method dan String pada Java ini adalah karena dalam membuat sebuah program, kita perlu menulis kode yang teratur dan mudah dipahami. Dengan menggunakan *method*, kita bisa memecah program menjadi bagian-bagian kecil yang memiliki tugas masing-masing, sehingga program lebih rapi dan mudah diperbaiki jika ada kesalahan. Sementara itu, *string* digunakan untuk mengolah teks seperti nama, kata, atau kalimat yang sering muncul dalam berbagai aplikasi. Java menyediakan banyak fitur bawaan untuk memanipulasi string, seperti menghitung panjang teks, mengubah huruf besar-kecil, atau mengambil sebagian teks tertentu. Oleh karena itu, mempelajari method dan string sangat penting agar kita bisa membuat program yang efisien, mudah dipelihara, dan bermanfaat dalam kehidupan sehari-hari.

#### 1.2 Tujuan

- 1.2.1 Memahami cara membuat dan menggunakan *method* dalam program Java.
- 1.2.2 Mengetahui cara kerja *string* dan berbagai fungsi yang bisa digunakan untuk mengolah teks.
- 1.2.3 Melatih kemampuan menyusun program yang terstruktur, efisien, dan mudah dipahami.
- 1.2.4 Menerapkan konsep modularitas dan manipulasi data teks dalam pemrograman Java.

#### 1.3 Manfaat Praktikum

- 1.3.1 Membantu mahasiswa memahami konsep *method* dan *string* secara lebih mendalam.
- 1.3.2 Membiasakan diri menulis kode yang rapi dan terorganisir dengan membagi program menjadi beberapa bagian kecil.
- 1.3.3 Memudahkan dalam mengolah teks seperti nama, kalimat, atau data input dari pengguna.
- 1.3.4 Meningkatkan kemampuan berpikir logis dalam membuat program yang efektif dan mudah dikembangkan.

## BAB 2

### PEMBAHASAN

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.

## 2.1 Class pertama 'Bilangan Prima\_2511531020'

### 2.1.1 Kode Program

```
1 package pekan7_2511531020;
2
3 import java.util.Scanner;
4 public class BilanganPrima_2511531020 {
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 }
15 public static void main (String[] args ) {
16     Scanner input = new Scanner (System.in);
17     System.out.print("input5"
18         + " 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 }
```

### 2.1.2 Langkah Kerja

- 1) `int factors = 0;`  
Variabel untuk menghitung berapa banyak bilangan yang dapat membagi `n`.
- 2) `for (int i = 1; i <= n; i++)`  
Perulangan dari 1 sampai `n`.
- 3) `if (n % i == 0)`  
Mengecek apakah `n` habis dibagi `i`.
- 4) `factors++`  
Jika habis dibagi, jumlah faktor ditambah 1.
- 5) `return (factors == 2);`  
Bilangan prima hanya punya **2 faktor** → 1 dan dirinya sendiri.  
Jika faktor = 2 → **true** (bilangan prima), selain itu → **false**.
- 6) Membuat **Scanner** untuk membaca input dari keyboard.
- 7) Menampilkan teks agar user memasukkan angka.
- 8) `a = input.nextInt()` menyimpan angka yang dimasukkan user.

- 9) Memanggil method `isPrime(a)`
- 10) Jika hasilnya **true** → tampilkan "a bilangan prima".
- 11) Jika **false** → tampilkan "a bukan bilangan prima".

### 2.1.3 Analisis Hasil

Jalankan program dengan memilih menu 'Run' dan akan menghasilkan output sebagai berikut.

```
input5 nilai n = 5
5 bilangan prima
```

## 2.2 Class kedua 'Mahasiswa\_2511531020'

### 2.2.1 Kode Program

```
1 package pekan7_2511531020;
2
3 public class Mahasiswa_2511531020 {
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 setName (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    }
33    public void cetak2() {
```



```

34     System.out.println("Nim2 :"+nim2);
35     System.out.println("Nama :"+nama);
36
37
38 }
39
40 }
41

```

### 2.2.2 Langkah Kerja

#### 1) Deklarasi Class.

Menandakan bahwa ini adalah class Mahasiswa yang menyimpan data tentang mahasiswa

#### 2) Variabel Global (Field / Atribut)

private → hanya bisa diakses di dalam class.

nim → tipe int, untuk menyimpan NIM.

nama dan nim2 → tipe String, menyimpan nama mahasiswa dan NIM bentuk string.

#### 3) Mutator (Setter) → Setter digunakan untuk mengubah / memberi nilai ke variabel.

setNim() → Memberi nilai ke variabel nim.

this.nim membedakan variabel class dan parameter.

setNim2() → Mengisi nilai untuk variabel nim2.

setNama() → Mengisi nilai nama mahasiswa.

#### 4) Accessor (Getter) → Getter digunakan untuk mengambil / membaca nilai dari variabel private. Getter sangat penting agar data tetap aman tetapi masih bisa dibaca oleh program lain.

getNim(), getNim2(), getNama().

#### 5) Metode Lain (Method Output)

Fungsinya:

cetak() → menampilkan nim (int)

cetak2() → menampilkan nim2 (String)

### 2.2.3 Analisis Hasil

Tidak menghasilkan output karena hanya menghasilkan method-method saja.

## 2.3 Class ketiga ‘PanggilMahasiswa\_2511531020’

### 2.3.1 Kode Program

```
1 package pekan7_2511531020;  
2  
3 public class PanggilMahasiswa_2511531020 {  
4     public static void main (String [] args) {  
5         Mahasiswa_2511531020 a= new Mahasiswa_2511531020();  
6         a.setNim(1020);  
7         a.setNama("Dinda");  
8         System.out.println(a.getNim());  
9         System.out.println(a.getNama());  
10        a.cetak();  
11    }  
12 }  
13  
14 }  
15
```

### 2.3.2 Langkah Kerja

- 1) Deklarasi Class→ Class ini berfungsi sebagai program utama (main class) untuk memanggil class Mahasiswa\_2511531020.
- 2) Method main()→ Method yang pertama kali dijalankan ketika program dieksekusi.
- 3) Membuat Objek→Membuat object a dari class Mahasiswa\_2511531020. Dengan object ini, program bisa mengakses method setter, getter, dan cetak. Objek = “wadah” untuk menyimpan data mahasiswa.
- 4) Mengisi Data ke Objek (Setter) →Mengubah nilai nama menjadi "Dinda".
- 5) Mengambil dan Menampilkan Data (Getter) → Menampilkan nama mahasiswa dari object.
- 6) Memanggil Method cetak() → Menampilkan data mahasiswa menggunakan method cetak() pada class Mahasiswa.
- 7) Output yang ditampilkan dari cetak():

### 2.3.3 Analisis Hasil

Jalankan program dengan memilih menu ‘Run’ dan akan menghasilkan output sebagai berikut.

```
1020
Dinda
Nim :1020
Nama :Dinda
```

## 2.4 lass keempat ‘PanggilMahasiswa2\_2511531020’

### 2.4.1 Kode Program

```
1 package pekan7_2511531020;
2
3 import java.util.Scanner;
4
5 public class PanggilMahasiswa2_2511531020 {
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_2511531020 a = new Mahasiswa_2511531020 ();
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
```

### 2.4.2 Langkah Kerja

- 1) Package→ Menunjukkan bahwa file ini berada dalam package bernama pekan7\_2511531020. Package berfungsi mengelompokkan file-file Java agar lebih terstruktur
- 2) Deklarasi Class→ Class ini berfungsi sebagai program utama (main class) untuk memanggil class Mahasiswa\_2511531020.
- 3) Import Scanner→ Mengimpor kelas Scanner yang digunakan untuk input dari keyboard.
- 4) Deklarsi KelasMethod mainMendefinisikan kelas utama yang memiliki fungsi main.
- 5) Method main→ Titik awal eksekusi program Java.
- 6) Membuat objek Scanner→ Membuat objek Scanner bernama input untuk membaca data dari user.

- 7) Input NIM→ print digunakan untuk menampilkan teks tanpa ganti baris. `nextLine()` membaca satu baris input sebagai String.
- 8) Input Nama→ Sama seperti sebelumnya, tetapi untuk nama mahasiswa.
- 9) Membuat Objek→ Membuat objek dari class `Mahasiswa_2511531020`. Objek a digunakan untuk memanggil method-method pada class tersebut.
- 10) Mengisi Nilai ke Atribut dalam Kelas→ Memanggil setter untuk mengisi atribut `nim` dan `nama`. `x` dan `y` adalah input dari user.
- 11) Mengecek Awal NIM→ `startsWith("25")` mengecek apakah NIM dimulai dengan angka 25. Jika benar, tampil: "nama anda dan Angkatan 2025"
- 12) Mengecek Jurusan Berdasarkan NIM→ `contains("1153")` mengecek apakah NIM mengandung angka 1153. Jika ya, maka mahasiswa dinyatakan dari program studi Informatika.
- 13) Memanggil Method `cetak2()`→ Memanggil method `cetak2()` dalam kelas `Mahasiswa_2511531020`. Biasanya method ini berfungsi menampilkan data mahasiswa (NIM & Nama).
- 14) Menutup Scanner→ Menutup scanner untuk menghindari memory leak.

#### 2.4.3 Analisis Hasil

Jalankan program dengan memilih menu 'Run' dan akan menghasilkan output sebagai berikut.

```
<terminated> PanggilMahasiswa2_2511531020
NIM: 2511531020
Nama: Dinda
Dinda anda angkatan 2025
Anda Mahasiswa Informatika
Nim2 :2511531020
Nama :Dinda
```

### 2.5 Class kelima 'PanggilMahasiswa3\_2511531020'

#### 2.5.1 Kode Program

```

1 package pekan7_2511531020;
2
3 import java.util.Scanner;
4 public class PanggilMahasiswa3_2511531020 {
5     public static void main (String [] args) {
6         Mahasiswa_2511531020 a= new Mahasiswa_2511531020();
7         Scanner input= new Scanner(System.in);
8         System.out.print("NIM: ");
9         a.setNim2(input.nextLine());
10        System.out.print("Nama: ");
11        a.setNama(input.nextLine());
12        if(a.getNim2().startsWith("25")) {
13            System.out.println(a.getNama()+ " anda angkatan 2025");
14        }
15        if(a.getNim2().contains("1153")) {
16            System.out.println(a.getNama()+ " Anda Mahasiswa Informatika");
17        }
18        a.cetak2();
19        input.close ();
20    }
21 }
22
23

```

### 2.5.2 Langkah Kerja

- 1) Membuat objek baru bernama a dari kelas Mahasiswa\_2511531020. Ini adalah instansiasi kelas.
- 2) Membuat objek Scanner untuk mengambil input dari pengguna melalui konsol.
- 3) Membaca input baris teks (NIM) dari pengguna dan menyimpannya ke dalam objek a menggunakan setNim2 (sebuah setter method).
- 4) Membaca input baris teks (Nama) dari pengguna dan menyimpannya ke dalam objek a menggunakan setNama (sebuah setter method).
- 5) Mengecek apakah nilai NIM dari objek a diawali dengan string "25". Jika benar, mencetak bahwa mahasiswa tersebut angkatan 2025.
- 6) Mengecek apakah nilai NIM dari objek a mengandung (memiliki di dalamnya) string "1153". Jika benar, mencetak bahwa mahasiswa tersebut Mahasiswa Informatika.
- 7) Memanggil method cetak2 pada objek a, yang kemungkinan besar digunakan untuk menampilkan data mahasiswa yang sudah disimpan.
- 8) Menutup objek Scanner untuk membebaskan sumber daya sistem.

kode ini berfungsi untuk mengambil input NIM dan Nama dari pengguna, menyimpan data tersebut ke dalam objek Mahasiswa, lalu melakukan pengecekan kondisi berdasarkan nilai NIM untuk menentukan angkatan dan program studi, dan terakhir mencetak data atau hasil pengecekan.

### 2.5.3 Analisis Hasil

Jalankan program dengan memilih menu 'Run' dan akan menghasilkan output sebagai berikut.

```
<terminated> PanggilMahasiswa3_2511531020
NIM: 2511531020
Nama: Dinda
Dinda anda angkatan 2025
Dinda Anda Mahasiswa Informatika
Nim2 :2511531020
Nama :Dinda
```

## 2.6 Class keenam 'String1\_2511531020'

### 2.6.1 Kode Program

```
1 package pekan7_2511531020;
2
3 public class String1_2511531020 {
4     public static void main (String [] args) {
5         String salam = "Assalamualaikum";
6         System.out.println("panjang salam adalah: " + salam.length());
7         System.out.println(salam.toUpperCase()); //Outputs "ASSALAMUALAIKUM"
8         System.out.println(salam.toLowerCase()); //Outputs "assalamualaikum"
9         System.out.println(salam.indexOf ("salam"));
10    }
11 }
12 }
13 }
```

### 2.6.2 Langkah Kerja

kode ini menunjukkan beberapa method dasar yang sering digunakan untuk memanipulasi dan mendapatkan informasi dari objek String dalam Java, seperti mengetahui panjang, mengubah huruf kapital/kecil, dan mencari posisi substring.

### 2.6.3 Analisis Hasil

Jalankan program dengan memilih menu 'Run' dan akan menghasilkan output sebagai berikut.

```
<terminated> String1_2511531020
panjang salam adalah: 15
ASSALAMUALAIKUM
assalamualaikum
2
```

## 2.7 Class ketujuh 'String2\_2511531020'

### 2.7.1 Kode Program

```
1 package pekan7_2511531020;
2
3 import java.util.Scanner;
4 public class String2_2511531020 {
5     public static void main (String [] args) {
6         Scanner input= new Scanner(System.in);
7         System.out.print("Nama Depan: ");
8         String firstname= input.nextLine();
9         System.out.print("Nama Belakang: ");
10        String lastname= input.nextLine();
11        String txt1 = "Dosen\intelektual\ kampus";
12        System.out.println("Nama Lengkap: "+firstname + " " + lastname);
13        System.out.println("Nama Lengkap: "+firstname.concat(lastname));
14        System.out.println(txt1);
15        int x = 10;
16        int y =20;
17        int z =x + y;
18        System.out.println("x+ y= "+z);
19        String a = "10";
20        String b= "20";
21        String c = a + b;
22        System.out.println("String a + string b =" +c);
23        String v = a + y;
24        System.out.println("String a + integer y =" +v);
25
26    }
27 }
28
29
30 }
```

### 2.7.2 Langkah Kerja

Kode tersebut berisi program yang dirancang untuk berinteraksi dengan pengguna dan mendemonstrasikan operasi dasar Java pada tipe data String dan Integer. Program ini pertama-tama mengambil input dari pengguna untuk Nama Depan dan Nama Belakang menggunakan objek Scanner, lalu menggabungkannya (konkatenasi) menggunakan operator + dan method .concat() untuk membentuk Nama Lengkap. Selanjutnya, kode ini menunjukkan perbedaan mendasar pada operator +: ketika digunakan antara dua int ( $x + y$ ), operator ini melakukan penjumlahan aritmatika (menghasilkan 30), namun ketika digunakan antara dua String ( $a + b$ ) atau antara String dan int ( $a + y$ ), operator ini bertindak sebagai penggabung teks (konkatenasi) (menghasilkan "1020").

### 2.7.3 Analisis Hasil

Jalankan program dengan memilih menu 'Run' dan akan menghasilkan output sebagai berikut.

```
<terminated> String2_2511531020 [  
Nama Depan: Dinda  
Nama Belakang: Amelia  
Nama Lengkap: Dinda Amelia  
Nama Lengkap: DindaAmelia  
Dosen"intelektual" kampus  
x+ y= 30  
String a + string b =1020  
String a + integer y =1020
```



### BAB III

#### 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] A. Gosling, B. Joy, and G. Steele, The Java Language Specification, Java SE 11 Edition. Boston, MA, USA: Addison-Wesley, 2019.
- [5] Oracle, “The Java™ Tutorials — Methods,” Oracle. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/javaOO/methods.html>. . [Accessed: 10-Nov-2025].