

**LAPORAN PRATIKUM**  
**PEMROGRAMAN ALGORITMA PEMROGRAMAN**  
**METHOD DAN STRING PADA JAVA**



**Disusun Oleh:**

Gina Ramadhani

Nim: 2511533014

**Dosen Pengampu:** DR. Wahyudi, S.T, M.T

**Asisten Pratikum:** Rahmad Dwi Rizki Olders

**DAPERTEMEN INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS ANDALAS**

## KATA PENGANTAR

Laporan praktikum ini disusun sebagai bentuk tanggung jawab penulis atas pelaksanaan kegiatan praktikum pada mata kuliah Algoritma dan Pemrograman, khususnya mengenai materi Method dan String dalam Bahasa Pemrograman Java. Penyusunan laporan ini bertujuan untuk membantu penulis memperdalam pemahaman terhadap konsep-konsep dasar pemrograman, terutama dalam penerapan method dan pengelolaan string sebagai bagian penting dari pembuatan program yang efisien dan terstruktur.

Selain berfungsi sebagai dokumentasi resmi kegiatan praktikum, laporan ini juga menjadi sarana pembelajaran tambahan bagi penulis. Melalui penyusunan laporan ini, penulis berkesempatan untuk melatih kemampuan menulis secara sistematis, mencatat hasil percobaan dengan rapi, serta menyajikan informasi teknis secara jelas dan terorganisir. Hal ini tidak hanya bertujuan untuk memenuhi persyaratan praktikum, tetapi juga untuk mengembangkan kemampuan analisis dan komunikasi ilmiah yang bermanfaat dalam proses pembelajaran selanjutnya.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Beberapa kekurangan mungkin terdapat dalam penulisan, penjelasan konsep, maupun penyajian hasil percobaan. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang bersifat membangun dari dosen pembimbing maupun pembaca agar laporan-laporan berikutnya dapat menjadi lebih baik, jelas, dan mudah dipahami. Dengan demikian, laporan ini diharapkan tidak hanya menjadi bukti pelaksanaan praktikum, tetapi juga menjadi sarana untuk meningkatkan kualitas pembelajaran dan kemampuan akademik penulis secara menyeluruh.

Padang, 2025

Penulis

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>i</b>
<b>DAFTAR ISI.....</b>	<b>ii</b>
<b>BAB I.....</b>	<b>1</b>
<b>PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Tujuan Praktikum .....	1
1.3 Manfaat Praktikum .....	2
<b>BAB II .....</b>	<b>3</b>
<b>PEMBAHASAN .....</b>	<b>3</b>
2.1 Praktikum “BilanganPrima” .....	3
2.2 Praktikum “Mahasiswa_2511533014” .....	6
2.3 Praktikum “PanggilMahasiswa_2511533014” .....	7
2.4 Praktikum “PanggilMahasiswa2_2511533014” .....	10
2.5 Praktikum “PanggilMahasiswa3_2511533014” .....	12
2.6 Praktikum “String1” .....	15
2.7 Praktikum “String2” .....	17
<b>BAB III.....</b>	<b>20</b>
<b>PENUTUP.....</b>	<b>20</b>
3.1 Kesimpulan.....	20
3.2 Saran .....	20
<b>DAFTAR PUSTAKA .....</b>	<b>21</b>

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Bahasa pemrograman Java merupakan salah satu bahasa yang banyak digunakan dalam dunia pemrograman karena bersifat berorientasi objek, portable, dan memiliki struktur yang mudah dipahami. Dalam proses pembelajaran pemrograman Java, terdapat dua materi penting yang perlu dipahami dengan baik, yaitu Method dan String. Kedua konsep ini berperan penting dalam membangun program yang efisien, terstruktur, dan mudah dikembangkan.

Method digunakan untuk menampung sekumpulan perintah yang memiliki fungsi tertentu agar kode program menjadi lebih terorganisir dan mudah digunakan kembali. Dengan memanfaatkan method, programmer dapat menghindari penulisan kode yang berulang serta meningkatkan keterbacaan dan efisiensi program.

Sementara itu, string merupakan tipe data yang digunakan untuk menyimpan dan mengolah kumpulan karakter. Dalam pemrograman, string sering digunakan untuk menangani data teks seperti nama, pesan, atau input dari pengguna. Java menyediakan berbagai method bawaan dalam kelas String yang memudahkan programmer dalam melakukan operasi seperti penggabungan, pemisahan, pencarian, dan pengubahan isi string.

Melalui praktikum ini, mahasiswa diharapkan dapat memahami dan menerapkan konsep method serta string dengan benar dalam pembuatan program Java. Pemahaman ini menjadi dasar penting untuk mengembangkan kemampuan logika dan keterampilan pemrograman yang lebih kompleks di tingkat selanjutnya.

#### **1.2 Tujuan Praktikum**

1. Memahami konsep dasar method dan perannya dalam menyusun program yang terstruktur dan efisien.

2. Mempelajari cara membuat, memanggil, dan menggunakan method dengan parameter serta nilai kembalian.
3. Memahami konsep string sebagai tipe data untuk menyimpan dan memanipulasi teks.
4. Mengetahui serta mempraktikkan berbagai method bawaan pada kelas String untuk melakukan operasi pengolahan teks, seperti penggabungan, pemisahan, pencarian, dan pengubahan isi string.
5. Meningkatkan kemampuan logika dan keterampilan pemrograman melalui penerapan method dan string dalam program sederhana.

### **1.3 Manfaat Praktikum**

1. Meningkatkan pemahaman tentang konsep method dan string dalam bahasa pemrograman Java.
2. Melatih kemampuan menulis program yang terstruktur, efisien, dan mudah dipahami.
3. Mengembangkan kemampuan berpikir logis dan sistematis dalam menyelesaikan masalah pemrograman.
4. Memberikan pengalaman langsung dalam menerapkan teori pemrograman pada praktik nyata.
5. Menjadi dasar yang kuat untuk mempelajari materi pemrograman Java yang lebih kompleks dan berorientasi objek.

## BAB II

### PEMBAHASAN

#### 2.1 Praktikum “BilanganPrima”

```
1 package pekan7_2511533014;
2
3 import java.util.Scanner;
4 public class BilanganPrima_2511533014 {
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("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 }
26 }
```

**Gambar 2.1** Kode program praktikum BilanganPrima

Program Java ini berfungsi untuk mengecek apakah sebuah bilangan merupakan bilangan prima atau bukan. Program memiliki method `isPrime` yang menghitung jumlah faktor sebuah angka dengan cara memeriksa pembagi dari 1 hingga angka tersebut. Jika jumlah faktor tepat dua, method mengembalikan nilai `true`, menandakan bilangan prima. Di bagian `main`, program meminta input angka dari pengguna, kemudian memanggil method `isPrime` untuk menentukan dan menampilkan apakah angka tersebut bilangan prima atau bukan.

Berikut penjelasan langkah-langkah kerjanya:

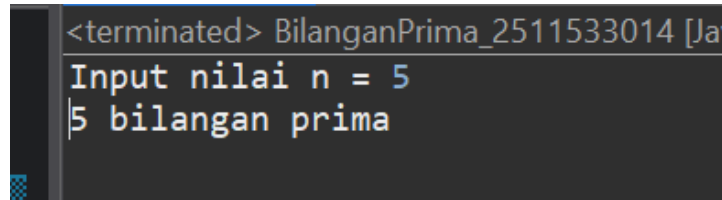
1. Import Scanner

Program mengimpor `java.util.Scanner` untuk membaca input dari pengguna.

2. Deklarasi kelas BilanganPrima  
Seluruh fungsi dan method berada di dalam kelas ini.
3. Method *isPrime(int n)* dibuat  
Method ini bertujuan untuk mengecek apakah suatu bilangan adalah bilangan prima atau bukan.
4. Inisialisasi variabel factors dengan nilai 0  
Variabel ini digunakan untuk menghitung jumlah faktor pembagi bilangan n.
5. Perulangan for dari 1 sampai n  
Program mengecek semua angka dari 1 sampai n.
6. Pengecekan if ( $n \% i == 0$ )  
Jika i adalah pembagi n (sisa pembagian 0), maka factors ditambah 1.
7. Return hasil pengecekan  
return (factors == 2);  
→ Bilangan prima hanya punya 2 faktor (1 dan dirinya sendiri).  
→ Jika benar 2, maka true (prima). Jika tidak, false.
8. Method main() dimulai  
Tempat eksekusi utama program.
9. Membuat objek Scanner  
Scanner input = new Scanner(System.in);  
Untuk membaca input angka dari pengguna.
10. Menampilkan pesan untuk memasukkan nilai n  
Program meminta user memasukkan bilangan.
11. Membaca input dan menyimpannya ke variabel a
12. Memanggil fungsi isPrime(a)  
Program mengecek apakah bilangan a adalah prima.
13. Menampilkan hasil
  - Jika true: mencetak "a bilangan prima"

- Jika false: mencetak "a bukan bilangan prima"

Dari Langkah-langkah diatas kita akan mendapatkan output:



```
<terminated> BilanganPrima_2511533014 [Java...]  
Input nilai n = 5  
5 bilangan prima
```

**Gambar 2.2** Kode program praktikum BilanganPrima

### 2.1.1 Analisis

- Program berjalan dengan baik tanpa error dan menghasilkan output sesuai dengan proses pengecekan bilangan prima.
- Saat fungsi `isPrime()` dijalankan, program melakukan perulangan dari 1 sampai n untuk menghitung jumlah angka yang menjadi faktor pembagi dari nilai yang dimasukkan.
- Setiap kali perulangan berjalan, program mengecek apakah i dapat membagi n tanpa sisa; jika iya maka nilai factors bertambah satu.
- Setelah perulangan selesai, program memeriksa apakah jumlah faktor sama dengan 2, yang berarti bilangan tersebut merupakan bilangan prima.
- Pada bagian `main()`, program meminta input dari pengguna lalu memanggil fungsi `isPrime()` untuk menentukan apakah bilangan tersebut prima atau bukan.
- Jika `isPrime()` mengembalikan nilai `true`, program menampilkan “n bilangan prima”.
- Jika `isPrime()` mengembalikan nilai `false`, program menampilkan “n bukan bilangan prima”.
- Pada contoh input 5, jumlah faktor yang ditemukan adalah 2 sehingga program menampilkan “5 bilangan prima” yang sesuai dengan hasil yang benar.



## 2.2 Praktikum “Mahasiswa\_2511533014”

```

1 package pekan7_2511533014;
2
3 public class Mahasiswa_2511533014 {
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 }

```

**Gambar 2.3** Kode program praktikum Mahasiswa\_2511533014

Kode program di atas merupakan sebuah kelas Java bernama Mahasiswa\_2511533014 yang digunakan untuk menyimpan dan mengelola data mahasiswa. Di dalam kelas ini terdapat tiga variabel privat, yaitu nim, nama, dan nim2, yang hanya dapat diakses melalui metode khusus dalam kelas tersebut. Program ini menyediakan metode *setter* untuk mengubah nilai setiap variabel, serta metode *getter* untuk mengambil nilai variabel yang dibutuhkan. Selain itu, terdapat dua metode, yaitu Cetak dan Cetak2, yang berfungsi menampilkan data mahasiswa ke layar. Metode Cetak menampilkan nilai nim dan nama, sedangkan Cetak2 menampilkan nim2 dan nama. Secara keseluruhan, kelas ini menggambarkan konsep enkapsulasi dalam pemrograman berorientasi objek dengan membatasi akses langsung terhadap variabel dan menggunakan metode untuk mengelola data.

Berikut penjelasan langkah-langkah kerjanya:

1. Program mendeklarasikan kelas Mahasiswa\_2511533014 sebagai wadah untuk menyimpan data mahasiswa.
2. Di dalam kelas, tiga variabel privat dibuat: nim, nama, dan nim2 agar tidak bisa diakses langsung dari luar kelas.
3. Program menyediakan metode *setter* (setNim, setNim2, setNama) untuk memasukkan atau mengubah nilai variabel.
4. Program menyediakan metode *getter* (getNim, getNim2, getNama) untuk mengambil dan mengembalikan nilai variabel ketika dibutuhkan.
5. Metode Cetak dijalankan untuk menampilkan nilai nim dan nama ke layar.
6. Metode Cetak2 dijalankan untuk menampilkan nilai nim2 dan nama ke layar.
7. Kelas ini menerapkan konsep enkapsulasi dengan mengelola data melalui getter dan setter, bukan akses langsung ke variabel.

Program ini secara tidak langsung tidak akan menghasilkan output, Jadi kelasnya siap digunakan, tetapi belum digunakan, sehingga tidak akan menampilkan apa pun sampai ada pemanggilan dari method main().

### 2.3 Praktikum “PanggilMahasiswa\_2511533014”

```

1 package pekan7_2511533014;
2
3 public class PanggilMahasiswa_2511533014 {
4     public static void main(String[] args) {
5         Mahasiswa_2511533014 a= new Mahasiswa_2511533014();
6         a.setNim(23532);
7         a.setNama ("Gina");
8         System.out.println(a.getNim());
9         System.out.println(a.getNama());
10        a.Cetak();
11    }
12 }
13
14 }
15

```

**Gambar 2.4** Kode program praktikum PanggilMahasiswa\_2511533014

Kode program di atas merupakan kelas `PanggilMahasiswa_2511533014` yang berfungsi untuk menjalankan dan menguji kelas `Mahasiswa_2511533014`. Di dalam method `main`, program membuat sebuah objek bernama `a` dari kelas `Mahasiswa_2511533014`. Setelah objek dibuat, program mengisi data mahasiswa dengan memanggil metode setter, yaitu `setNim(23532)` untuk memberikan nilai pada `nim` dan `setNama("Gina")` untuk memberikan nilai pada `nama`. Selanjutnya, program menampilkan nilai `nim` dan `nama` ke layar dengan memanggil metode getter `getNim()` dan `getNama()`. Setelah itu, metode `Cetak()` juga dipanggil untuk menampilkan data mahasiswa sesuai cara cetak yang telah didefinisikan di dalam kelas `Mahasiswa_2511533014`. Dengan demikian, kode ini berfungsi sebagai pemanggil yang mengeksekusi dan menampilkan output dari objek mahasiswa.

Berikut penjelasan langkah-langkah kerjanya:

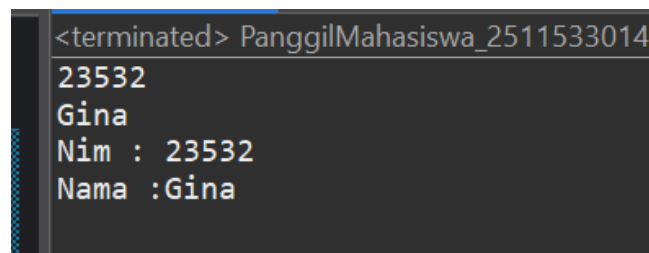
1. Menggunakan method `main()`  
 Karena program Java membutuhkan titik awal eksekusi. Tanpa `main()`, kode tidak bisa dijalankan.
2. Membuat objek dengan sintaks `new Mahasiswa_2511533014()`  
 Ini digunakan untuk membuat *instance* dari kelas `Mahasiswa_2511533014` agar program dapat mengakses dan menyimpan data di dalam objek tersebut. Tanpa objek, method setter, getter, dan `Cetak` tidak bisa digunakan.
3. Menggunakan sintaks `a.setNim(23532)` dan `a.setNama("Gina")`  
 Setter digunakan karena variabel `nim` dan `nama` bersifat `private`. Artinya, variabel tersebut tidak boleh diakses langsung dari luar kelas, sehingga harus menggunakan metode resmi (setter) untuk mengisi nilainya. Ini adalah penerapan enkapsulasi.
4. Menggunakan `System.out.println(a.getNim())` dan `a.getNama()`  
 Getter digunakan untuk mengambil nilai dari variabel yang bersifat `private`. Ini memberi akses secara aman dan terkontrol agar data dapat dibaca namun tetap terlindungi.
5. Menggunakan `a.Cetak()`  
 Method ini dipakai untuk menampilkan data mahasiswa dalam format yang sudah ditentukan di dalam kelas `Mahasiswa_2511533014`. Dengan cara ini,

proses menampilkan data lebih rapi, tidak perlu menulis ulang perintah `println` di class `main`.

6. Menggunakan sintaks pemanggilan metode melalui objek (`a.method`)

Karena semua `method` dalam kelas `Mahasiswa_2511533014` bukan `static`, sehingga harus dipanggil melalui objek. Ini sesuai konsep OOP bahwa perilaku objek hanya bisa digunakan setelah objek dibuat.

Dari penjelasan kode program di atas kita menghasilkan output:



```
<terminated> PanggilMahasiswa_2511533014
23532
Gina
Nim : 23532
Nama :Gina
```

**Gambar 2.5** Output kode program `PanggilMahasiswa_2511533014`

### 2.1.3 Analisis

- Program berjalan dengan baik tanpa error dan berhasil menampilkan output sesuai data yang diberikan.
- Program menampilkan nilai `nim` dan nama pertama kali dengan menggunakan metode *getter*, yaitu `getNim()` dan `getNama()`.
- Setelah itu, program memanggil metode `Cetak()` untuk menampilkan kembali data `nim` dan nama menggunakan format cetak yang telah dibuat di dalam kelas `Mahasiswa_2511533014`.
- Pemanggilan *getter* dan metode `Cetak` menunjukkan bahwa data dalam objek dapat diakses dan ditampilkan dengan dua cara berbeda namun tetap menghasilkan informasi yang sama.
- Program menunjukkan penerapan enkapsulasi karena variabel `nim` dan nama bersifat `private` dan hanya bisa diakses melalui metode *setter* dan *getter*.

## 2.4 Praktikum “PanggilMahasiswa2\_2511533014”

```

1 package pekan7_2511533014;
2
3 import java.util.Scanner;
4
5 public class PanggilMahasiswa2_2511533014 {
6     public static void main(String[] args) {
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_2511533014 a= new Mahasiswa_2511533014();
13        a.setNim2(x);
14        a.setNama(y);
15        if(x.startsWith("25")) {
16            System.out.println(a.getNama()+ " anda angkatan 2025");
17        }
18        if(a.getNim2().contains("1153")) {
19            System.out.println(y+" Anda Mahasiswa Informatika");
20        }
21        a.Cetak2();
22        input.close();
23
24    }
25 }
26
27 }
28

```

**Gambar 2.6** Kode program praktikum PanggilMahasiswa2\_2511533014

Kode program di atas merupakan sebuah kelas bernama PanggilMahasiswa2\_2511533014 yang berfungsi untuk menerima input dari pengguna dan memproses data mahasiswa menggunakan kelas Mahasiswa\_2511533014. Program dimulai dengan membuat objek Scanner untuk membaca input NIM dan nama dari keyboard. Nilai yang dimasukkan pengguna kemudian disimpan ke dalam variabel x dan y. Setelah itu, program membuat objek baru dari kelas Mahasiswa\_2511533014 dan mengisi data objek tersebut menggunakan metode setNim2(x) dan setNama(y). Program kemudian melakukan pengecekan kondisi: jika NIM yang dimasukkan diawali angka “25”, maka program menampilkan pesan bahwa mahasiswa tersebut merupakan angkatan 2025. Selanjutnya, jika NIM mengandung angka “1153”, program menampilkan bahwa mahasiswa tersebut berasal dari program studi Informatika. Di akhir proses, metode Cetak2 dipanggil untuk menampilkan data mahasiswa, dan Scanner ditutup untuk mengakhiri proses input. Program ini menunjukkan penggunaan input, pengecekan kondisi, serta pemanggilan metode dalam pemrograman berorientasi objek.

Berikut penjelasan langkah-langkah kerjanya:

1. Program mulai berjalan dari method main, lalu langsung menyiapkan Scanner untuk membaca input dari keyboard.
2. Pengguna diminta memasukkan NIM, dan hasilnya disimpan sebagai teks di variabel x.
3. Setelah itu pengguna diminta memasukkan nama, lalu nilai tersebut disimpan di variabel y.
4. Program kemudian membuat sebuah objek dari kelas Mahasiswa\_2511533014 untuk menampung data yang baru saja dimasukkan.
5. Nilai NIM yang dimasukkan tadi dimasukkan ke objek lewat setNim2(x), dan nama dimasukkan lewat setNama(y).
6. Program mengecek apakah NIM diawali angka “25”. Jika iya, program menampilkan pesan bahwa mahasiswa tersebut merupakan angkatan 2025.
7. Program juga mengecek apakah di dalam NIM terdapat “1153”. Jika ada, program menampilkan bahwa mahasiswa tersebut termasuk mahasiswa Informatika.
8. Setelah itu program memanggil Cetak2() untuk menampilkan kembali data yang sudah tersimpan dalam objek.
9. Pada bagian akhir, Scanner ditutup karena proses input sudah selesai.

Dari penjelasan kode program kode di atas kita menghasilkan output:

```
<terminated> PanggilMahasiswa2_2511533014 [
NIM: 2511533014
Nama: gina
gina anda angkatan 2025
gina Anda Mahasiswa Informatika
Nim : 2511533014
Nama : gina
```

**Gambar 2.7** Output kode program PanggilMahasiswa2\_2511533014

### 2.1.4 Analisis

- Program berjalan normal tanpa error karena tidak ada pesan kesalahan saat dijalankan.
- Output pertama menampilkan NIM dan Nama sesuai data yang sudah disimpan.
- Program kemudian menampilkan informasi angkatan dengan menggabungkan nama dan kalimat tambahan.
- Setelah itu muncul keterangan jurusan karena program menampilkan teks penjelasan yang masih menggunakan nama yang sama.
- NIM dan Nama muncul kembali karena ada bagian program yang memanggil ulang data tersebut.
- Semua output muncul berurutan sesuai urutan perintah yang dieksekusi.

### 2.5 Praktikum “PanggilMahasiswa3\_2511533014”

```

1 package pekan7_2511533014;
2
3 import java.util.Scanner;
4
5 public class PanggilMahasiswa3_2511533014 {
6     public static void main (String[] args) {
7         Mahasiswa_2511533014 a= new Mahasiswa_2511533014();
8         Scanner input= new Scanner(System.in);
9         System.out.print("NIM: ");
10        a.setNim2(input.nextLine());
11        System.out.print("Nama: ");
12        a.setNama(input.nextLine());
13        if(a.getNim2().startsWith("25")) {
14            System.out.println(a.getNama()+" anda angkatan 2025");
15        }
16        if(a.getNim2().contains("1153")) {
17            System.out.println(a.getNama()+" Anda Mahasiswa Informatika");
18        }
19        a.Cetak2();
20        input.close();
21    }
22 }
23
24 }
25
26

```

**Gambar 2.8** Kode program praktikum PanggilMahasiswa3\_2511533014

Program tersebut bekerja dengan cara membaca input NIM dan nama mahasiswa menggunakan *Scanner*, kemudian menyimpannya ke dalam objek *Mahasiswa\_2511533014*. Setelah data dimasukkan, program melakukan dua pengecekan: pertama, jika NIM dimulai dengan angka “25”, maka program

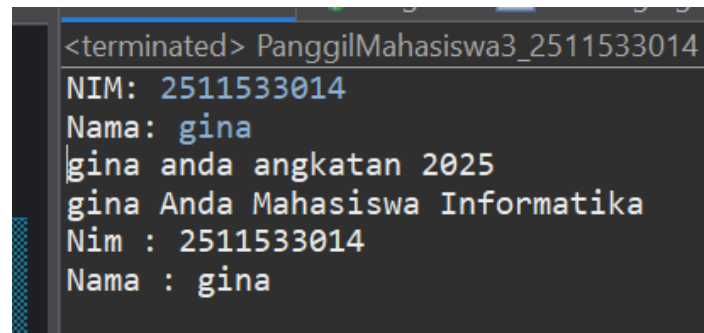
menampilkan bahwa mahasiswa tersebut adalah angkatan 2025; kedua, jika NIM mengandung angka “1153”, program menampilkan bahwa mahasiswa tersebut adalah mahasiswa Informatika. Setelah itu, program memanggil *method* Cetak2() untuk menampilkan informasi tambahan dari objek mahasiswa, lalu menutup input Scanner sebelum program selesai dijalankan.

Berikut penjelasan langkah-langkah kerjanya:

1. Program memanggil package `pekan7_2511533014` dan mengimpor *Scanner* untuk input.
2. Di dalam main, program membuat objek Mahasiswa\_2511533014 bernama a.
3. Program membuat objek *Scanner* untuk membaca input dari keyboard.
4. Program meminta pengguna memasukkan NIM, lalu menyimpannya dengan `a.setNim2(...)`.
5. Program meminta pengguna memasukkan Nama, lalu menyimpannya dengan `a.setNama(...)`.
6. Program melakukan pengecekan pertama:
  - Jika NIM mulai dengan "25", maka mencetak:  
nama anda angkatan 2025.
7. Program melakukan pengecekan kedua:
  - Jika NIM mengandung "1153", maka mencetak:  
nama Anda Mahasiswa Informatika.
8. Program memanggil *method* `a.Cetak2()` untuk menampilkan data tambahan (sesuai isi *method* di kelas Mahasiswa).
9. *Scanner* ditutup dengan `input.close()`.



Dari penjelasan langkah-langkah kerja kode program diatas kita mendapatkan output seperti ini:



**Gambar 2.9** Output kode program PanggilMahasiswa3\_2511533014

### 2.1.5 Analisis

- Program berjalan tanpa error karena tidak ada pesan kesalahan yang muncul saat dijalankan.
- Output pertama menampilkan NIM dan Nama sesuai data yang diinput, yaitu “2511533014” dan “gina”.
- Program menampilkan informasi angkatan karena NIM dimulai dengan “25”, sehingga muncul kalimat “gina anda angkatan 2025”.
- Program menampilkan keterangan jurusan karena NIM mengandung “1153”, sehingga muncul teks “gina Anda Mahasiswa Informatika”.
- Method Cetak2() menjalankan perintah yang mencetak ulang NIM dan Nama, sehingga data tampil kembali pada output.
- Seluruh hasil output muncul runtut sesuai urutan perintah dalam program.
- Output dari program ini sama dengan program PanggilMahasiswa2

## 2.6 Praktikum “String1”



```

1 package pekan7_2511533014;
2
3 public class String1_2511533014 {
4     public static void main(String[] args) {
5         String salam = "Assalamualiakum";
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")); // Outputs 2
10    }
11 }
12
13 }
14

```

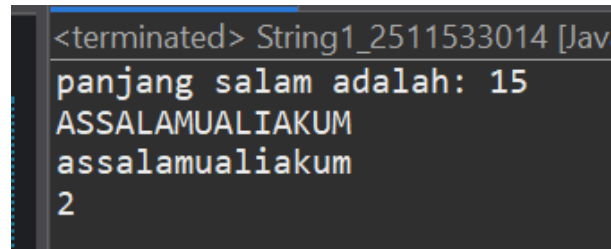
**Gambar 2.10** Kode program praktikum String1

Program ini mendefinisikan sebuah string bernama salam berisi teks “Assalamualaikum”, kemudian menampilkan panjang karakter dari *string* tersebut. Setelah itu program mencetak versi huruf besar dan huruf kecil dari teks yang sama. Terakhir, program mencari posisi kata “salam” di dalam *string* dan menampilkannya sebagai output. Program berjalan secara berurutan sesuai perintah yang ditulis.

Berikut penjelasan langkah-langkah kerjanya:

1. Program mendeklarasikan variabel salam dengan nilai *string* "Assalamualaikum".
2. Program menampilkan panjang teks pada variabel salam menggunakan `salam.length()`.
3. Program menampilkan teks salam dalam huruf kapital penuh menggunakan `salam.toUpperCase()`.
4. Program menampilkan teks salam dalam huruf kecil penuh menggunakan `salam.toLowerCase()`.
5. Program mencari posisi awal kata "salam" di dalam *string* utama menggunakan `salam.indexOf("salam")` dan menampilkannya.

Dari penjelasan langkah-langkah kerja kode program diatas menghasilkan output seperti ini:



```
<terminated> String1_2511533014 [Jav  
panjang salam adalah: 15  
ASSALAMUALIAKUM  
assalamualiakum  
2
```

**Gambar 2.11** Output kode program String1

#### 2.1.6 Analisis

1. Program berjalan dengan normal tanpa error, terbukti karena tidak ada pesan kesalahan saat dijalankan.
2. Output pertama menampilkan panjang *string* “Assalamualaikum”, yaitu 15 karakter, sesuai hasil dari `salam.length()`.
3. Baris berikutnya menampilkan *string* dalam huruf kapital penuh menggunakan `salam.toUpperCase()`, sehingga menghasilkan “ASSALAMUALIAKUM”.
4. Setelah itu program menampilkan versi huruf kecil dari *string* yang sama melalui `salam.toLowerCase()`, sehingga muncul “assalamualaikum”.
5. Output terakhir menunjukkan indeks awal kemunculan kata “salam” di dalam *string*, yaitu angka 2, hasil dari `salam.indexOf("salam")`.
6. Seluruh output muncul secara berurutan sesuai alur perintah yang dieksekusi dalam program.

## 2.7 Praktikum “String2”

```

1 package pekan7_2511533014;
2
3 import java.util.Scanner;
4 public class String2_2511533014 {
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
31 }
32

```

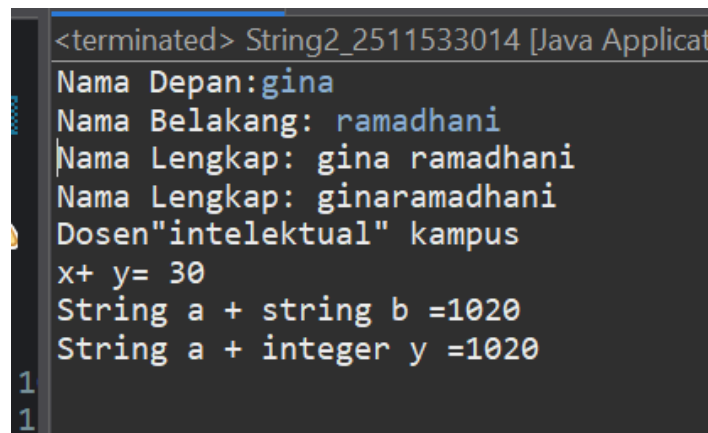
**Gambar 2.12** Kode program praktikum String2

Program ini dimulai dengan meminta input berupa nama depan dan nama belakang dari pengguna, lalu menampilkan nama lengkap dengan dua metode penggabungan *string*. Program juga menampilkan sebuah teks tambahan yang disimpan dalam variabel *txt1*. Selanjutnya, program melakukan operasi matematika dengan menjumlahkan dua bilangan bulat dan menampilkan hasilnya. Setelah itu, program menunjukkan perbedaan antara penggabungan *string* dan penjumlahan angka dengan menggabungkan dua *string* angka serta menggabungkan *string* dengan *integer*. Semua proses dijalankan secara berurutan sesuai instruksi dalam program.

1. Program memulai dengan membuat objek *Scanner* untuk membaca input dari keyboard.
2. Program meminta pengguna memasukkan Nama Depan, lalu menyimpannya ke variabel *firstname*.
3. Program meminta pengguna memasukkan Nama Belakang, lalu menyimpannya ke variabel *lastname*.

4. Program menampilkan nama lengkap dengan dua cara: menggunakan operasi penggabungan (`firstname + " " + lastname`) dan menggunakan *method concat*.
5. Program membuat variabel teks *txt1* berisi kalimat “Dosen ‘intelektual’ kampus”, kemudian menampilkannya.
6. Program mendefinisikan dua variabel *integer* `x = 10` dan `y = 20`, lalu menjumlahkannya menjadi `z = x + y`, dan menampilkan hasilnya.
7. Program membuat variabel *string* `a = "10"` dan `b = "20"`, lalu menggabungkannya menjadi `c = a + b`, kemudian mencetak hasil gabungan *string* tersebut.
8. Program menjumlahkan *string* `a` dengan *integer* `y` dalam operasi `v = a + y`, sehingga membentuk penggabungan teks dan angka, dan mencetak hasilnya.

Dari penjelasan kode program diatas sehingga menghasilkan output seperti ini:



```

<terminated> String2_2511533014 [Java Applicat
Nama Depan: gina
Nama Belakang: ramadhani
Nama Lengkap: gina ramadhani
Nama Lengkap: ginaramadhani
Dosen"intelektual" kampus
x+ y= 30
String a + string b =1020
String a + integer y =1020
1
1

```

**Gambar 2.13** Output kode program String2

### 2.1.7 Analisis

- Program berjalan normal tanpa menampilkan error.
- Output menampilkan nama depan dan nama belakang sesuai nilai variabel.
- Program menampilkan nama lengkap dalam dua versi: dengan spasi dan tanpa spasi.

- Teks yang mengandung tanda kutip (“intelektual”) tampil dengan benar.
- Operasi aritmatika pada variabel numerik menghasilkan output  $x + y = 30$ .
- Penggabungan *string* dengan *string* menghasilkan "1020".
- Penggabungan *string* dengan *integer* juga menghasilkan "1020" karena integer otomatis dikonversi menjadi *string*.
- Semua output muncul secara berurutan sesuai urutan perintah program.

## BAB III

### PENUTUP

#### 3.1 Kesimpulan

Pada praktikum mengenai *method* dan *string* dalam bahasa pemrograman Java, dapat disimpulkan bahwa kedua konsep ini memiliki peranan penting dalam membangun program yang terstruktur dan fleksibel. Penggunaan *method* memungkinkan pembagian program menjadi bagian-bagian kecil yang memiliki fungsi spesifik sehingga kode lebih teratur, mudah dipahami, dan mudah dikelola. Penggunaan *method* juga membantu mengurangi duplikasi kode serta mempermudah proses *debugging*.

Sementara itu, pengolahan *string* menjadi salah satu kemampuan dasar yang sangat penting karena banyak program memerlukan manipulasi teks, seperti penggabungan, pencarian karakter, penghitungan panjang *string*, hingga perubahan bentuk huruf. Melalui berbagai percobaan di praktikum, terlihat bahwa Java menyediakan banyak *method* bawaan untuk mempermudah manipulasi *string* sehingga programmer dapat mengolah data secara lebih efektif dan efisien.

#### 3.2 Saran

Untuk praktikum selanjutnya, alangkah baiknya jika diberikan lebih banyak contoh dan latihan soal yang bervariasi supaya kami bisa lebih memahami penggunaan *method* dan *string* dalam berbagai situasi. Selain itu, akan sangat membantu jika penjelasan teori disertai demonstrasi langsung saat praktikum, sehingga mahasiswa bisa langsung melihat penerapan konsepnya dalam program. Dengan begitu, proses belajar akan terasa lebih mudah, lebih jelas, dan lebih menarik.

## DAFTAR PUSTAKA

- [1] J. Gosling, B. Joy, G. Steele, dan G. Bracha, *The Java Language Specification*, 3rd ed. Addison-Wesley, 2005.
- [2] Oracle, “The Java Tutorials – Learning the Java Language,” 2024. [Daring]. Tersedia pada: <https://docs.oracle.com/javase/tutorial/> (Diakses: 15-Nov-2025).
- [3] H. Schildt, *Java: The Complete Reference*, 11th ed. New York: McGraw-Hill, 2019.
- [4] K. N. King, *Java Programming: From the Ground Up*. New York: McGraw-Hill, 2012.
- [5] Oracle, “String Class Documentation,” 2024. [Daring]. Tersedia pada: <https://docs.oracle.com/javase/8/docs/api/java/lang/String.html> (Diakses: 15-Nov-2025).