

LAPORAN PRATIKUM
ALGORITMA DAN PEMROGRAMAN
METHOD STRING DALAM BAHASA JAVA

Disusun Oleh:

Khairun Nisa

2511532015

Dosen Pengampu:

DR. WAHYUDI, S.T, M.T

Asisten Pratikum:

Aufan Taufiqurrahman



FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN INFORMATIKA
UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas terselesaikannya laporan praktikum Algoritma dan Pemrograman ini. Laporan ini disusun sebagai dokumentasi hasil kegiatan praktikum yang telah dilaksanakan pada 30 Oktober 2025, dengan pembahasan method String dalam bahasa Java.

Penulis mengucapkan terima kasih kepada Bapak Dosen serta asisten laboratorium yang telah membimbing selama praktikum berlangsung. Semoga laporan ini dapat bermanfaat bagi pembaca dan menjadi referensi untuk perkembangan ilmu pengetahuan di bidang pemrograman.

Padang, 14 November 2025

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Pratikum	1
1.3 Manfaat Pratikum	2
BAB II PEMBAHASAN	3
2.1 Bilangan Prima.....	3
2.2 Mahasiswa	5
2.3 Panggil Mahasiswa	7
2.4 Panggil Mahasiwa 2	8
2.5 String 1	10
2.6 String 2	12
BAB III KESIMPULAN	15
3.1 Kesimpulan	15
DAFTAR PUSTAKA	16

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam dunia pemrogramana, data tidak selalu berupa angka. Sering kali program harus menangani data berupa teks atau karakter, yang dalam istilah pemrograman dikenal sebagai String. String merupakan tipe data yang sangat umum digunakan dalam berbagai aplikasi, mulai dari pengolahan teks sederhana hingga sistem pencarian, validasi data, hingga pengembangan antarmuka pengguna.

String bukanlah tipe data primitif namun merupakan tipe data class. Untuk membuat variabel yang menyimpan sebuah string, kita membuat sebuah object dari class String [1]. Class String berisi string yang tetap (*immutable string*). Artinya sekali instance String dibuat maka isinya tidak bisa diubah. Kelas String memiliki 13 konstruktor yang memungkinkan kita membuat obyek String dan menginisialisasi nilainya dengan menggunakan berbagai macam sumber data yang berbeda [2].

Bahasa pemrograman modern, termasuk bahasa Java menyediakan berbagai method String (metode operasi pada string) yang memungkinkan programmer untuk memanipulasi, menganalisis, dan memformat teks dengan cara yang efisien dan terstruktur. Beberapa operasi umum meliputi mengubah huruf kapital/kecil (*uppercase/lowercase*), memotong bagian tertentu (*substring*), mencari posisi karakter atau kata (*search*), mengganti bagian dari teks (*replace*), serta memisahkan atau menggabungkan string (*split dan join*).

1.2 Tujuan Praktikum

1. Memahami konsep dasar data string dan perannya dalam pemrograman.
2. Mengetahui berbagai method string yang tersedia dalam bahasa pemrograman yang digunakan, yaitu bahasa Java.
3. Mampu mengimplementasikan method string untuk memanipulasi, menganalisis, dan memformat teks dalam program.
4. Melatih kemampuan dalam merancang algoritma sederhana yang melibatkan operasi teks menggunakan method string.

5. Meningkatkan keterampilan problem solving melalui studi kasus yang berbasis pemrosesan string.

1.3 Manfaat Praktikum

1. Memberikan pemahaman praktis tentang penggunaan string sebagai struktur data teks dalam pemrograman.
2. Membantu mahasiswa menulis kode yang lebih efisien dan mudah dibaca saat menangani data berbasis teks.
3. Menjadi dasar bagi pengembangan aplikasi yang melibatkan input pengguna, validasi data, parsing teks, atau pengolahan dokumen.
4. Memperkuat fondasi algoritma dan logika pemrograman melalui penerapan operasi string dalam berbagai skenario nyata.
5. Memfasilitasi kesiapan mahasiswa dalam menghadapi tugas atau proyek pemrograman lanjutan yang memerlukan manipulasi teks.

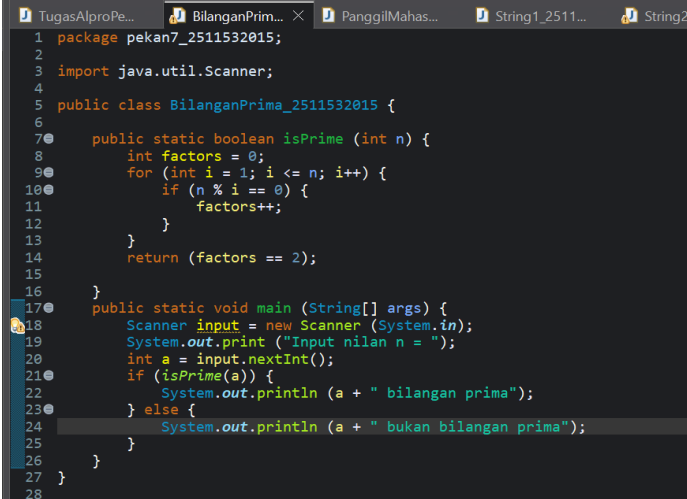
BAB II

PEMBAHASAN

2.1 Bilangan Prima

Praktikum ini memperkenalkan konsep dasar pembuatan method sendiri (user-defined-method) dalam bahasa pemrograman Java, method ini memungkinkan programmer untuk memisahkan bagian logika tertentu dari program utama, sehingga kode menjadi lebih terstruktur, mudah dibaca, dan dapat digunakan berulang. Sebagai contoh penerapan, dalam praktikum ini dibuat sebuah program yang bertujuan untuk mengecek apakah suatu bilangan bulat positif merupakan bilangan prima atau bukan berdasarkan input pengguna. Program ini menggunakan method khusus bernama isPrime() yang dirancang untuk menghitung jumlah faktor dari suatu bilangan dan mengembalikan nilai true jika bilangan tersebut hanya memiliki dua faktor yaitu (1 dan dirinya sendiri) yang merupakan ciri khas bilangan prima. Berikut langkah-langkah dalam pembuatan program Bilangan Prima:

1. Buat package baru dengan cara yang sama seperti praktikum minggu-minggu sebelumnya, dengan nama 'package7_2511532015'
2. Kemudian buat class baru di package7_2511532015 dengan nama 'BilanganPrima_2511532015'
3. Ketikkan kode program:



```
1 package pekan7_2511532015;
2
3 import java.util.Scanner;
4
5 public class BilanganPrima_2511532015 {
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
```

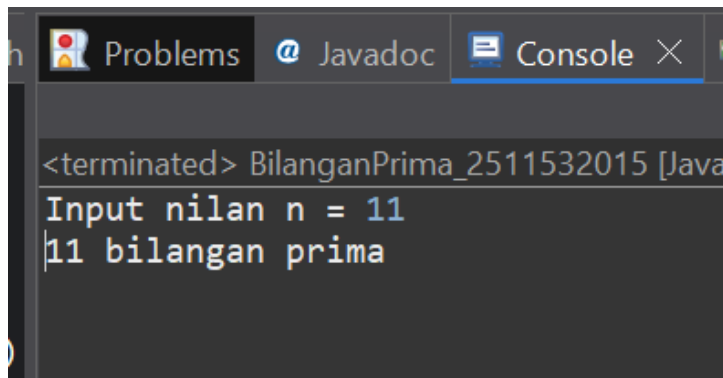
Kode Program 2.1: Kode program 'BilanganPrima'

Blok 7-14 mengecek apakah bilangan n adalah bilangan prima atau bukan menggunakan perulangan for. Program menghitung jumlah faktor

dari n menggunakan loop dari $i=1$ sampai $i=n$, jika $n\%i==0$ artinya faktor dari n maka variabel factors ditambah 1 ($factors++$). Setelah loop selesai, jika $factors==2$, maka n adalah bilangan prima (true) jika tidak maka false.

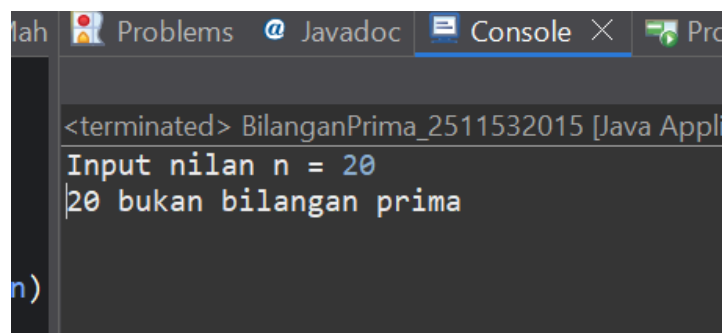
Selanjutnya blok 17-26, program membuat objek class scanner yang digunakan untuk meminta pengguna memasukkan sebuah angka. Setelah nilai tersimpan, program memanggil method `isPrime(a)` sebagai kondisi dalam pernyataan percabangan if-else. Method `isPrime` akan mengembalikan nilai true jika a merupakan bilangan prima, atau false jika bukan.

4. Setelah kode program selesai ditulis lalu dijalankan, akan menghasilkan output:



```
<terminated> BilanganPrima_2511532015 [Java  
Input nilan n = 11  
11 bilangan prima
```

Gambar 2.1: Hasil Output BilanganPrima (true)



```
<terminated> BilanganPrima_2511532015 [Java Appl  
Input nilan n = 20  
20 bukan bilangan prima
```

Gambar 2.2: Hasil Output BilanganPrima (false)

Setelah program dijalankan, sistem akan menampilkan pesan “Input nilai n= “ di konsol, lalu pengguna memasukkan sebuah angka. Setelah pengguna mengetikkan bilangan bulat dan menekan tombol enter, program segera memproses input tersebut dan memanggil metode `isPrime()`. Metode ini memeriksa setiap angka dari 1 hingga nilai input untuk menghitung berapa kali bilangan tersebut habis dibagi (jumlah

faktornya). Jika jumlah faktornya terdapat dua, program akan mencetak pesan bahwa bilangan tersebut adalah bilangan prima, jika tidak akan muncul 'bilangan ini bukan prima'. Dengan demikian, program berhasil memberikan respons langsung berdasarkan logika matematis sederhana yang telah dikodekan sebelumnya.

2.2 Mahasiswa

Program ini adalah sebuah kelas java yang dirancang untuk merepresentasikan data seorang mahasiswa, dengan informasi seperti NIM dan nama. Kelas ini menggunakan prinsip-prinsip dasar dalam pemrograman, seperti enkapsulasi dalam pemrograman berorientasi objek (OOP), yaitu diwujudkan melalui penggunaan setter dan getter sebagai satu-satunya cara untuk mengakses dan memodifikasi data internal objek.

Enkapsulasi adalah suatu cara untuk menyembunyikan implementasi detail dari class untuk mencegah akses data yang ilegal. Ada dua hal yang mendasar dari enkapsulasi yaitu information hiding dan interface to access data. Information hiding adalah menyembunyikan informasi dari suatu class agar tidak bisa diakses dari luar class, caranya hanya dengan memberikan private pada variabel yang inginkita hiding. Selanjutnya mengenai interface to access data, ialah cara kita untuk mengubah nilai pada suatu variabel yang telah dilakukan information hiding. Cara yang dilakukan adalah melalui method, lewat method ini kita bisa merubah nilai suatu variabel yang telah mengalami information hiding [3].

Berikut langkah-langkah dalam pembuatan program Mahasiswa:

1. Sebelum mengetikkan kode, terlebih dahulu membuat class baru di package `7_2511532015`, dan diberi nama 'Mahasiswa_2511532015'.
2. Ketikkan kode program seperti berikut:


```

1 package pekan7_2511532015;
2
3 public class Mahasiswa_2511532015 {
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    //method 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 }
37

```

Kode program 2.2: Kode program Mahasiswa

Program ini sebenarnya hanya berupa kerangka awal (template atau blueprint) dari sebuah kelas yang dirancang untuk digunakan oleh pengguna lain di kemudian hari. Dengan kata lain, program Mahasiswa_2511532015 ini tidak dimaksudkan untuk dijalankan secara mandiri, melainkan sebagai komponen pendukung yang akan dipanggil atau diinstansikan oleh program utama untuk mengolah data mahasiswa. Di dalamnya sudah disediakan struktur dasar berupa variabel privat (nim, nim2, dan nama) serta metode-metode akses seperti setter dan getter yang memungkinkan program lain mengisi dan mengambil data secara terkendali.

Setter adalah sebuah aksi saat kita memasukkan nilai/values kedalam suatu variabel/object, sedangkan getter adalah sebuah aksi saat kita mengambil sebuah nilai/values dari suatu variabel/object [3].

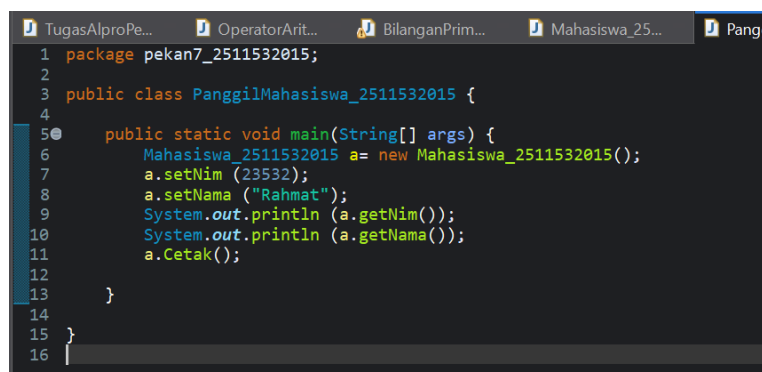
Saat objek dibuat, variabel-variabel tersebut masih kosong (atau bernilai default), lalu diisi melalui pemanggilan metode setter. Setelah data tersimpan, pengguna program bisa membacanya melalui getter atau langsung menampilkannya ke layar dengan metode Cetak() atau Cetak2(), yang secara internal mengakses variabel privat tersebut dan mencetaknya dalam format tertentu. Dengan demikian, peran utama kelas ini adalah

sebagai model data yang siap dipanggil, diinstansikan, dan diintegrasikan ke dalam program yang lebih besar dan fungsional.

2.3 Panggil Mahasiswa

Program ini adalah program pemanggil atau program utama yang bertugas menggunakan kelas Mahasiswa_2511532015 (program sebelumnya yang telah dibuat). Program ini tidak membuat data mahasiswa sendiri, tapi memanggil kelas Mahasiswa yang sudah dibuat sebelumnya, lalu mengisi datanya, membacanya, dan menampilkannya. Program ini hanya bisa berjalan jika kelas Mahasiswa_2511532015 sudah ada dan benar. Berikut langkah-langkah pembuatan programnya:

1. Sebelum mengetikkan kode program, terlebih dahulu membuat class baru di package7_2511532015 dengan nama 'PanggilMahasiswa_2511530215'.
2. Lalu ketikkan kode programnya:



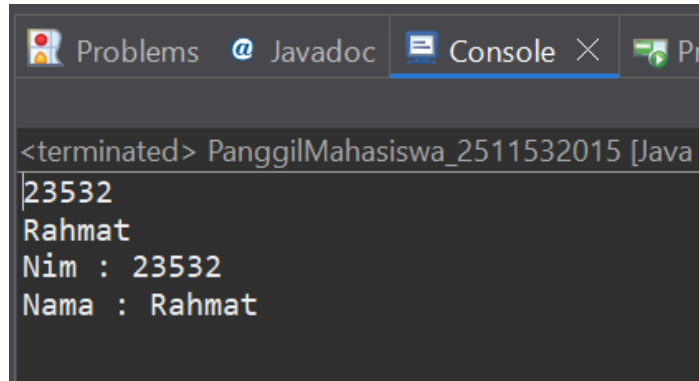
```
1 package pekan7_2511532015;
2
3 public class PanggilMahasiswa_2511532015 {
4
5     public static void main(String[] args) {
6         Mahasiswa_2511532015 a= new Mahasiswa_2511532015();
7         a.setNim (23532);
8         a.setName ("Rahmat");
9         System.out.println (a.getNim());
10        System.out.println (a.getName());
11        a.Cetak();
12    }
13 }
14
15 }
16 }
```

Kode program 2.3: Kode program PanggilMahasiswa

Pertama-tama, program membuat sebuah objek baru dari kelas Mahasiswa_2511532015 dengan perintah Mahasiswa_2511532015 a=new Mahasiswa_2511532015();. Artinya, komputer mengalokasikan memori untuk menyimpan data seseorang mahasiswa dan memberi nama sementara objek tersebut sebagai a. setelah objek a dibuat, program mengisi datanya dengan a.setNim(235323) dan a.setName("Rahmat") nilai ini disimpan ke variabel nim dan nama di kelas Mahasiswa_2511532015. Lalu, data ditampilkan lewat a.getNim() dan a.getName(), serta dicetak rapi dengan a.Cetak(). Semua ini bisa berjalan karena program ini hanya memanggil kelas Mahasiswa_2511532015 sebagai pabrik objek dimana ia tidak bekerja

sendiri tapi mengandalkan metode public dari kelas lain, sesuai prinsip OOP data disembunyikan akses lewat antarmuka yang aman.

3. Ketika dijalankan akan menghasilkan output:



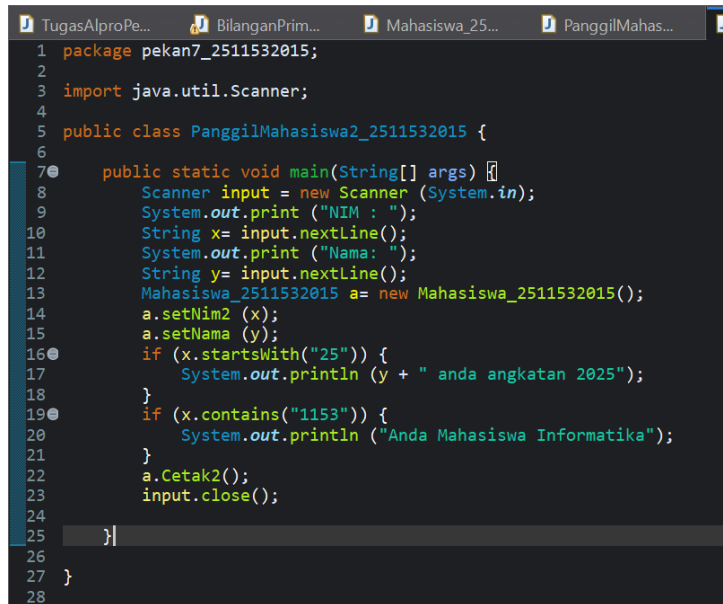
Gambar 2.3: Hasil output program PanggilMahasiswa

Dua baris pertama menampilkan nilai NIM dan nama secara terpisah sesuai dengan data yang diisi melalui metode set. Dua baris berikutnya menampilkan data yang sama namun dalam format yang lebih rapi dan terstruktur, seperti Nim: 23532 lalu dibawahnya Nama: Rahmat yang merupakan hasil dari pemanggilan metode Cetak().

2.4 Panggil Mahasiswa 2

Program ini merupakan implementasi sederhana dari konsep pemrograman berorientasi objek (OOP) dalam java, dimana kelas utama berperan sebagai pengendali alur program yang berinteraksi langsung dengan pengguna, sekaligus memanfaatkan kelas lain (Mahasiswa_2511532015) sebagai wadah penyimpanan dan pengelolaan data mahasiswa. Dengan memadukan input dari pengguna, logika percabang berdasarkan pola teks pada NIM, serta pemanggilan metode dari objek mahasiswa, program ini tidak hanya menampilkan data yang dimasukkan, tetapi juga memberi informasi tambahan seperti angkatan dan jurusan berdasarkan struktur NIM tersebut. Berikut langkah-langkah pembuatan programnya:

1. Sebelum mengetikkan kode program buat class baru di package7_2511532015 dengan nama 'PanggilMahasiswa2'.
2. Ketikkan kode programnya:



```

1 package pekan7_2511532015;
2
3 import java.util.Scanner;
4
5 public class PanggilMahasiswa2_2511532015 {
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_2511532015 a= new Mahasiswa_2511532015();
14        a.setNim2 (x);
15        a.setName (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: Kode program PanggilMahasiswa2

Program menggunakan class scanner untuk membaca input dari pengguna melalui keyboard. Lalu program meminta input NIM dan disimpan dalam bentuk string ke variabel x, dan input Nama disimpan dalam bentuk string ke variabel y. Selanjutnya, program membuat sebuah objek baru dari kelas Mahasiswa_2511532015 dengan perintah new Mahasiswa_2511532015() dan memberi nama objek tersebut dengan a. kemudian nilai x (NIM) disimpan ke dalam objek a melalui metode setNim2(x) dan nilai y disimpan ke dalam objek a melalui metode setName(y).

Pada blok selanjutnya, terdapat 2 method dari kelas String di Java, yaitu .startsWith() dan .contains().

Metode startsWith() di java digunakan untuk memeriksa apakah string dimulai dengan awalan yang diberikan. Awalan yang ingin diperiksa juga berupa string yang diteruskan sebagai parameter ke fungsi. Awalan juga bisa berupa karakter Tunggal. Jika string dimulai dengan awalan yang diberikan, ia akan mengembalikan true, atau false [5].

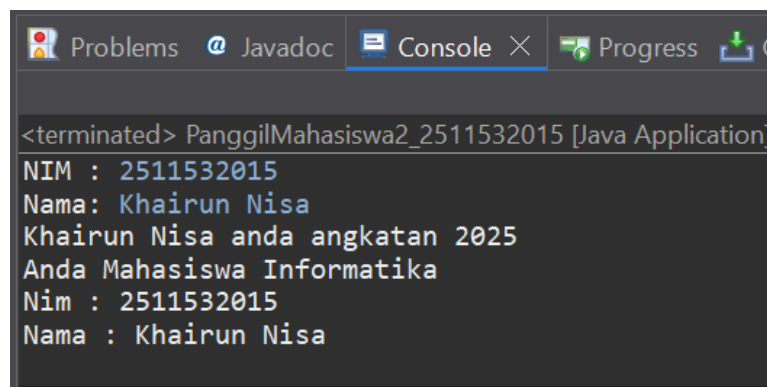
Metode contains() mengembalikan nilai boolean berdasarkan keberadaan sub-string dalam string ini atau tidak. Metode contains() mencari sub-string di seluruh string asli dan mengembalikan true jika dan

hanya jika string ini berisi urutan nilai karakter yang ditentukan, dan false sebaliknya [6].

Jika program (`x.startsWith("25")`) yang berarti `x` (`setNim2`) berawalan 25 maka program akan mencetak pesan "(inputan nama) anda angkatan 2025". Jika (`x.contains("1153")`) yang berarti `x`(`setNim2`) memiliki 1153 di keseluruhan stringnya dimanapun berada maka program akan menampilkan "Anda Mahasiswa Informatika", jika tidak ada 1153 dalam string tersebut maka tidak ada output "Anda Mahasiswa Informatika".

Terakhir, memanggil metode `a.Cetak2()` untuk menampilkan NIM dan nama dalam format rapi.

3. Setelah dijalankan program akan menghasilkan output:



```
<terminated> PanggilMahasiswa2_2511532015 [Java Application]
NIM : 2511532015
Nama: Khairun Nisa
Khairun Nisa anda angkatan 2025
Anda Mahasiswa Informatika
Nim : 2511532015
Nama : Khairun Nisa
```

Gambar 2.4: Hasil Output program PanggilMahasiswa2

Setelah dijalankan, program akan meminta user untuk memasukkan NIM dan Nama, dimana jika NIM berawalan 25 akan menampilkan Anda angkatan 2025, dan jika di string NIM terdapat 1153 akan menampilkan anda mahasiswa informatika, dan jika keduanya tidak ada atau false maka program akan langsung menampilkan output NIM dan Nama saja.

2.5 String 1

Program ini adalah contoh sederhana dalam bahasa Java yang bertujuan untuk memperkenalkan penggunaan berbagai metode bawaan (built-in methods) dari kelas String. Berikut langkah-langkah pembuatan programnya:

1. Sebelum mengetikkan kode, terlebih dahulu membuat class baru di package `7_2511532015` dengan nama 'String1_2511532015'.
2. Selanjutnya ketikkan kode programnya:

```

1 package pekan7_2511532015;
2
3 public class String1_2511532015 {
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")); //Outputs 2
11    }
12 }
13
14 }
15

```

Kode Program 2.5: Kode program String1

Program ini bekerja dengan menginisialisasikan sebuah variabel bertipe string bernama salam yang berisikan teks “Assalamualaikum”. Selanjutnya, program mencetak jumlah karakter dalam salam menggunakan metode `.length()`.

Dalam pemrograman Java, metode `length()` merupakan fitur fundamental dari kelas `String`, yang menyediakan cara mudah untuk menentukan jumlah karakter dalam sebuah string. Metode ini tanpa argument (tidak menerima input) dan mengembalikan nilai `int` yang mewakili Panjang string [7].

Selanjutnya, program memanggil `salam.toUpperCase()` dan `salam.toLowerCase()`. Metode `toUpperCase()` mengubah string menjadi huruf besar sebaliknya `toLowerCase()` mengubah string menjadi huruf kecil [8]. Lalu ada metode `indexOf()`, metode ini adalah alat canggih yang digunakan untuk menemukan posisi karakter atau substring tertentu dalam sebuah string. Fungsi ini mengembalikan indeks di dalam objek string pemanggil dari kemunculan pertama nilai yang ditentukan, memulai pencarian dari `formIndex`. Jika nilai tidak ditemukan, metode akan mengembalikan `-1`. Fungsional ini `indexOf()` sangat berguna untuk tugas manipulasi string, seperti penguraian dan validasi data [9].

3. Setelah dijalankan program akan menghasilkan output:

```

<terminated> String1_2511532015 [Java Application] C:\Use
Panjang salam adalah: 15
ASSALAMUALAIKUM
assalamualaikum
2

```

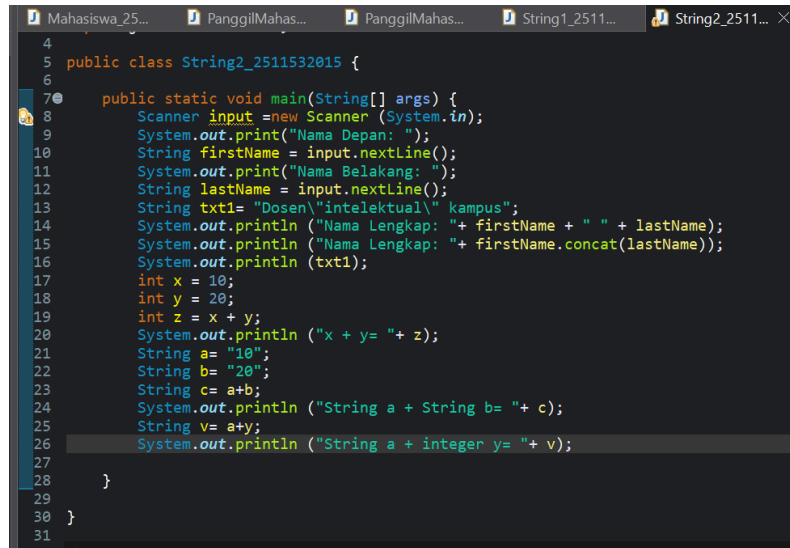
Gambar 2.5: Hasil Output program String1

Setelah dijalankan, program langsung mengeksekusi isi metode `main()` tanpa meminta input apapun dari pengguna. Variabel `salam` yang telah diinisialisasi langsung dieksekusi ke dalam metode `string`. Yang pertama metode `.length()` yang berfungsi menghitung jumlah string dimana jumlah string `salam` ada 15, lalu baris kedua `.toUpperCase()` yang berfungsi untuk menampilkan teks dalam huruf kapital akan menampilkan “ASSALAMUALAIKUM”, dan sebaliknya `.toLowerCase()` akan menampilkan “assalamualaikum”, baris keempat `salam` terdapat pada posisi (indeks) ke 2, maka output nya 2.

2.6 String 2

Program ini adalah contoh latihan sederhana yang menggabungkan beberapa konsep dasar Java: input pengguna, manipulasi string, operasi matematika, dan perilaku *concatenation* (penggabungan) antara string dan angka. Tujuannya bukan untuk membuat aplikasi nyata, tapi untuk membantu pemula memahami bagaimana data berbeda tipe (string, integer) bisa diolah dan ditampilkan di layar. Program ini juga menunjukkan perbedaan antara operasi matematika biasa (misalnya $x + y$) dengan penggabungan teks ("10" + "20"), serta cara menggunakan metode seperti `.concat()` dan `\n` untuk membuat baris baru. Dengan struktur yang jelas dan output langsung ke konsol, program ini sangat cocok sebagai bahan pembelajaran awal tentang interaksi pengguna dan manipulasi data. Berikut langkah-langkah dalam pembuatan programnya;

1. Sebelum mengetikkan kode, terlebih dahulu membuat class baru di `package7_2511532015` dengan nama ‘String2_2511532015’.
2. Lalu ketikkan kodenya:



```

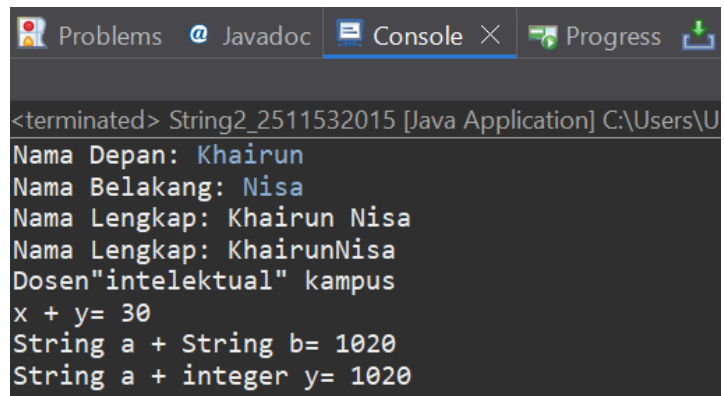
4
5 public class String2_2511532015 {
6
7     public static void main(String[] args) {
8         Scanner input = new Scanner(System.in);
9         System.out.print("Nama Depan: ");
10        String firstName = input.nextLine();
11        System.out.print("Nama Belakang: ");
12        String lastName = input.nextLine();
13        String txt1 = "Dosen\\intelektual\\ kampus";
14        System.out.println("Nama Lengkap: " + firstName + " " + lastName);
15        System.out.println("Nama Lengkap: " + firstName.concat(lastName));
16        System.out.println(txt1);
17        int x = 10;
18        int y = 20;
19        int z = x + y;
20        System.out.println("x + y = " + z);
21        String a = "10";
22        String b = "20";
23        String c = a+b;
24        System.out.println("String a + String b = " + c);
25        String v = a+y;
26        System.out.println("String a + integer y = " + v);
27    }
28 }
29
30
31

```

Kode Program 2.6: Kode program String2

Program dimulai dengan membuat objek Scanner untuk membaca input dari pengguna. Pertama, ia meminta nama depan dengan pesan "Nama Depan: " dan menyimpannya ke variabel `firstName`. Lalu, ia meminta nama belakang dengan pesan "Nama Belakang: " dan menyimpannya ke `lastName`. Setelah itu, program mencetak dua versi nama lengkap: satu dengan spasi (`firstName + " " + lastName`) dan satu lagi dengan metode `.concat()`, yang juga menggabungkan kedua nama. Selanjutnya, program menampilkan teks "Dosen\\intelektual\\ kampus" di mana `\` digunakan untuk "escape" tanda kutip agar tidak salah baca oleh compiler. Kemudian, program mendemonstrasikan operasi aritmatika sederhana: menjumlahkan dua angka ($x = 10$, $y = 20$) menjadi $z = 30$, lalu mencetak hasilnya. Terakhir, program menunjukkan perilaku unik Java saat menggabungkan string dan angka: variabel `a = "10"` dan `b = "20"` digabungkan menjadi "1020" (bukan 30), karena Java menganggapnya sebagai teks. Bahkan saat `a + y` (string + integer), Java secara otomatis mengubah angka `y` menjadi string dan menggabungkannya — sehingga hasilnya "1020", bukan 30. Semua hasil ini dicetak ke layar secara berurutan, tanpa memerlukan logika percabangan atau perulangan.

3. Setelah dijalankan, program akan menghasilkan output:

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

```
<terminated> String2_2511532015 [Java Application] C:\Users\U
Nama Depan: Khairun
Nama Belakang: Nisa
Nama Lengkap: Khairun Nisa
Nama Lengkap: KhairunNisa
Dosen"intelektual" kampus
x + y= 30
String a + String b= 1020
String a + integer y= 1020
```

Gambar 2.6: Hasil Output String2

BAB III

KESIMPULAN

3.1 Kesimpulan

Secara keseluruhan, rangkaian program-program ini merupakan latihan bertahap dalam memahami dasar-dasar pemrograman Java, khususnya dalam dua pilar utama: (1) pemrograman berorientasi objek (OOP) dan (2) manipulasi data teks (String). Program Mahasiswa mengenalkan konsep enkapsulasi melalui penggunaan variabel *private* dan metode *getter/setter*, sementara program PanggilMahasiswa dan PanggilMahasiswa2 menunjukkan bagaimana kelas tersebut digunakan sebagai model data oleh program lain membaca input pengguna, menyimpannya ke objek, menganalisis pola (misalnya dari NIM), lalu menampilkan informasi secara terstruktur. Di sisi lain, program String dan String2 memperkuat pemahaman tentang tipe data String: mulai dari metode bawaan seperti `.length()`, `.toUpperCase()`, `.indexOf()`, hingga perilaku penggabungan teks (+ dan `.concat()`), penggunaan karakter escape (`\`), serta perbedaan mendasar antara operasi matematika dan operasi string. Bersama-sama, semua program ini membentuk fondasi penting bagi pemula: tidak hanya mengajarkan sintaks Java, tetapi juga cara berpikir terstruktur, memisahkan logika, dan memahami bagaimana data dikelola, diproses, dan ditampilkan dalam dunia pemrograman.

DAFTAR PUSTAKA

- [1] S. Dirman, "String & Method dalam Java," [Online]. Available: <https://id.scribd.com/document/541543347/Bab-3-Input-dan-Output>. [Accessed 12 November 2025].
- [2] A. Syamsudin, "String Java," 2014. [Online]. Available: <https://id.scribd.com/doc/220857513/5-string-java>. [Accessed 12 November 2025].
- [3] Muis, "Enkapsulasi," 2017. [Online]. Available: <https://id.scribd.com/document/382443566/Enkapsulasi>. [Accessed 14 November 2025].
- [4] B. Akyong, "Setter dan Getter pada Java," Medium, 11 Maret 2019. [Online]. Available: <https://medium.com/java-and-other-java-framework/setter-dan-getter-pada-java-46820df13b3>. [Accessed 14 December 2025].
- [5] S. Bhagwat, "startsWith() di Java," Scaler, 30 Maret 2022. [Online]. Available: <https://www.scaler.com/topics/startswith-in-java/>. [Accessed 14 November 2025].
- [6] "Periksa apakah suatu string berisi substring di Java," Atta, 8 January 2020. [Online]. Available: [https://attacomsian.com/blog/java-string-contains-substring#:~:text=digunakan%20%E2%80%94%20contains\(\)%20.-,1.,dalam%20string%20ini%20atau%20tidak.&text=Metode%20contains\(\)%20akan%20memunculkan,yang%20diberikan%20ada%20atau%20tidak..](https://attacomsian.com/blog/java-string-contains-substring#:~:text=digunakan%20%E2%80%94%20contains()%20.-,1.,dalam%20string%20ini%20atau%20tidak.&text=Metode%20contains()%20akan%20memunculkan,yang%20diberikan%20ada%20atau%20tidak..) [Accessed 14 November 2025].
- [7] V. Agarwal, "Metode Panjang String Java()," Kode Naukri 360, 19 September 2025. [Online]. Available: <https://www.naukri.com/code360/library/string-length>. [Accessed 14 November 2025].
- [8] "Metode Java String toUpperCase()," w3schools, [Online]. Available: https://www.w3schools.com/java/ref_string_toupper.asp. [Accessed 14 November 2025].
- [9] "Java String indexOf() - Temukkan Indeks Karakter," vultr, 17 December 2024. [Online]. Available: <https://docs.vultr.com/java/standard-library/java/lang/String/indexOf>. [Accessed 14 November 2025].

