

LAPORAN PRAKTIKUM  
PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN  
“STRING PADA BAHASA PEMROGRAMAN JAVA”

Disusun Oleh :

Dzhillan Dzhalila

2511531001

Dosen Pengampu:

Dr. Wahyudi, S.T, M.T

Asisten Praktikum:

Aufan Taufiqurrahman



FAKULTAS TEKNOLOGI INFORMASI  
DEPARTEMEN INFORMATIKA  
UNIVERSITAS ANDALAS  
2025

## KATA PENGANTAR

Puji syukur penulis panjatkan atas kehadiran Allah SWT Tuhan Yang Maha Esa yang telah memberikan rahmat dan berkatnya, sehingga penulis dapat menyelesaikan laporan praktikum Algoritma Pemrograman dan Pemrograman dengan baik. Laporan ini disusun sebagai dokumentasi dan refleksi atas praktikum yang telah dilaksanakan pada pertemuan minggu ke-7 untuk mata kuliah Praktikum Algoritma Pemrograman. Praktikum ini berfokus pada pembahasan konsep methods untuk tipe data String yang ada pada bahasa pemrograman Java.

Melalui praktikum ini, hasil yang diharapkan adalah penulis dapat memahami dan mengimplementasikan konsep dasar String dalam bahasa Java. Melalui projek sederhana yang dilakukan pada praktikum ini membantu penulis untuk memahami bagaimana pemilihan dan penggunaan beberapa jenis methods pada tipe data String sesuai dengan kebutuhan.

Penulis menyadari bahwa, laporan praktikum ini masih memiliki banyak kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan dan pengembangan di masa mendatang. Semoga laporan ini dapat memberikan manfaat, baik sebagai bahan pembelajaran penulis maupun referensi pembaca dalam memahami dasar-dasar pemrograman komputer.

Padang, 2025

Dzhillan Dzhalila

251531001

## DAFTAR ISI

KATA PENGANTAR .....	i
DAFTAR ISI.....	ii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Tujuan .....	2
1.3 Manfaat .....	2
BAB II PEMBAHASAN .....	3
2.1 Program Bilangan Prima .....	3
2.2 Program Mahasiswa .....	5
2.2.1. Langkah-Langkah .....	5
2.2.2 Penjelasan Program.....	6
2.3 Program Panggil Mahasiswa.....	7
2.3.1. Langkah-Langkah .....	7
2.3.2. Penjelasan Program.....	8
2.4 Program Panggil Mahasiswa 2.....	9
2.4.1. Langkah-Langkah Program.....	9
2.4.2 Penjelasan Program.....	10
2.5 Program String 1 .....	12
2.5.1. Langkah-Langkah .....	12
2.5.2 Penjelasan Program.....	13
2.6 Program String 2 .....	14
2.6.1. Langkah-Langkah .....	14
2.6.2. Penjelasan Program.....	15
BAB III KESIMPULAN.....	17
DAFTAR PUSTAKA .....	18

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Dalam pemrograman, string merupakan tipe data yang berisi deretan karakter baik itu huruf maupun angka yang diapit oleh tanda kutip. String pada program digunakan untuk menyimpan dan mempresentasikan teks. Teks bisa saja hanya terdiri dari 1 karakter yang biasanya dimasukkan ke dalam tipe data *char*. Namun, jika ingin menahan lebih dari satu karakter, kita harus menggunakan tipe data string. Tipe data string harus menggunakan tanda kutip ganda (“”).

Pada bahasa pemrograman Java, tipe data string tidak hanya berfungsi sebagai tempat penyimpanan teks. Tipe data ini juga bisa berfungsi sebagai alat untuk mengambil input dari pengguna dan penampil output program ke layer. String juga bisa digunakan jika kita ingin memanipulasi suatu teks, seperti kita bisa untuk mengubah huruf yang menjadi karakter dalam suatu string menjadi huruf besar semua atau huruf kecil semua. Pemanfaatan string tidak sampai di sana, dengan string, kita bisa untuk menggunakan beberapa method string yang tersedia hingga menjalankan aksi tertentu sesuai dengan program yang programmer set.

Dengan mengetahui bahwa pemahaman akan fungsi string, kita tahu bahwa string adalah konsep dasar yang harus dipahami. Memahami konsep dasar string akan membantu kita sebagai programmer untuk menyimpan dan memanipulasi suatu teks.

## **1.2 Tujuan**

1. Memahami konsep dasar dan cara kerja string dalam bahasa pemrograman Java.
2. Menguasai penggunaan method string pada Bahasa pemrograman Java baik untuk memanipulasi data hingga menjalankan program tertentu.
3. Membuat berbagai macam methods menggunakan konsep string.

## **1.3 Manfaat**

1. Memberikan penguasaan akan keterampilan teknis dalam mengolah konsep string dan megimplementasikan berbagai method pada berbagai kasus pemrograman.
2. Meningkatkan kemampuan dalam memulis kode program yang lebih efektif dan efisien.

## BAB II

## PEMBAHASAN

### 2.1 Program Bilangan Prima



```
1 package pekan7_2511531001;
2 import java.util.Scanner;
3 public class BilanganPrima_2511531001 {
4     public static boolean isPrime(int n) {
5         int factors = 0;
6         for (int i=1; i<= n; i++) {
7             if (n % i == 0) {
8                 factors++;
9             }
10        }
11        return (factors == 2);
12    }
13    public static void main(String[] args) {
14        Scanner input =new Scanner (System.in);
15        System.out.print(" Input nilai n =");
16        int a = input.nextInt();
17        if (isPrime(a)) {
18            System.out.println(a+"bilangan prima");
19        }else {
20            System.out.println(a+" bukan bilangan prima");
21        }
22    }
23
24 }
```

Gambar 2.1 : Kode Program Menentukan Bilangan Pri

#### 2.1.1. Langkah-Langkah

Adapun Langkah Langkah untuk membuat program ini adalah sebagai berikut:

1. *import java.util.Scanner;* untuk membaca input dari pengguna melalui keyboard
2. Menggunakan method *booleaan isPrime(int )* yang bertugas untuk mengecek apakah bilangan “n” adalah bilangan prima atau tidak (method ini akan mengembalikan nilai true atau false).
3. Untuk deklarasi variabel pertama di dalam method *booleaan isPrime* adalah variabel *factors* yang berupa tipe data integer.
4. Di dalam method *booleaan isPrime*, kita juga menggunakan konsep perulangan *for* dengan inisialisasi *i=1*, kondisi apabila *i<=a* dan nilai *i* akan *diincrementkan*.
5. Dalam perulangan for akan ada konsep percabangan *if*, jika nilai sisa hasil bagi bilangan n dengan *i = 0* terpenuhi maka nilai *factors* akan bertambah 1.

6. *return factors==2* untuk menyatakan bahwa bilangan prima hanya punya 2 faktor yaitu 1 dan dirinya sendiri.
7. *Tmethod main = public static void main (String [] args)*
8. *Scanner input = new Scanner(System.in) =* Di dalam *method main* akan ada variabel “input” sebagai objek *scanner* untuk membaca input dari keyboard.
9. Penggunaan menginputkan nilai n.
10. *Int n = input.nextInt(); =* menyimpan input pengguna dalam variabel n.
11. Menggunakan percabangan *if*, dengan memanggil *method is Prime* untuk mengecek apakah n adalah bilangan prima.

### 2.1.2 Penjelasan Program

Tujuan dari program ini adalah untuk menentukan angka atau bilangan yang diinputkan oleh pengguna adalah bilangan prima atau tidak. Dengan urgensi tersebut, kita dapat menggunakan salah satu *method* yaitu adalah *isPrime*, yang mana menggunakan parameter bilangan n. *Method* ini akan menghitung jumlah faktor dari n dengan cara mengecek dari 1 sampai n yang bisa membagi n tanpa sisa. Jika jumlah faktor tepat 2 (yaitu 1 dan n sendiri) maka n adalah bilangan prima. Saat Boolean *method isPrime* bernilai *true*, maka program akan mencetak “n bilangan prima”. Dan jika bernilai *false*, maka program akan mencetak “n bukan bilangan prima”. Output dari program ini adalah sebagai berikut :

```
<terminated> BilanganPrima_2511531001 [Java Application] C:\U
Input nilai n = 13
13 adalah bilangan prima
```

Gambar 2.2 : Output Kode Program IsPrime

## 2.2 Program Mahasiswa

```

1 package pekan7_2511531001;
2
3 public class Mahasiswa_2511531001 {
4     // variabel global
5     private int nim;
6     private String nama,nim2;
7
8     //membuat mutator (setter)
9     public void setNim (int nim) {
10         this.nim=nim;
11     }
12     public void setNim2 (String nim2) {
13         this.nim2=nim2;
14     }
15     public void setNama (String nama) {
16         this.nama=nama;
17     }
18
19     //membuat accessor (getter)
20     public int getNim() {
21         return nim;
22     }
23     public String getNim2 () {
24         return nim2;
25     }
26     public String getNama () {
27         return nama;
28     }
29
30     //metode lain
31     public void Cetak() {
32         System.out.println("Nim : "+ nim);
33         System.out.println("Nama : "+ nama);
34     }
35     public void Cetak2() {
36         System.out.println("Nim : "+ nim2);
37         System.out.println("Nama : "+ nama);
38     }
39 }
40

```

Gambar 2.3 : Kode Program Mahasiswa

### 2.2.1. Langkah-Langkah

1. Deklarasi Variabel Global (Private) : diberi akses *private* agar tidak bisa diakses langsung dari luar *class* (prinsip enkapsulasi)
  - a. Private int nim
  - b. Private String nama,nim2
2. Membuat Method Setter (Mutator) : mengisi nilai ke variabel nim dan nama.
3. Membuat Method Getter (Accessor) : mengambil nilai dari variabel nim dan nama dan mengembalikan nilai sesuai tipe data masing masing.
4. Membuat Method Cetak() Tanpa Parameter = *method* untuk menampilkan data mahasiswa yang sudah disimpan di variabel

global. Menggunakan “this.nim” dan “this.nama” untuk mencetak nilai yang tersimpan.

5. Membuat Method Cetak(String nim, String nama) Dengan Parameter = *method* yang ditambahkan parameter sehingga dapat mev=cetak berdasarkan input parameter bukan dari variabel global.

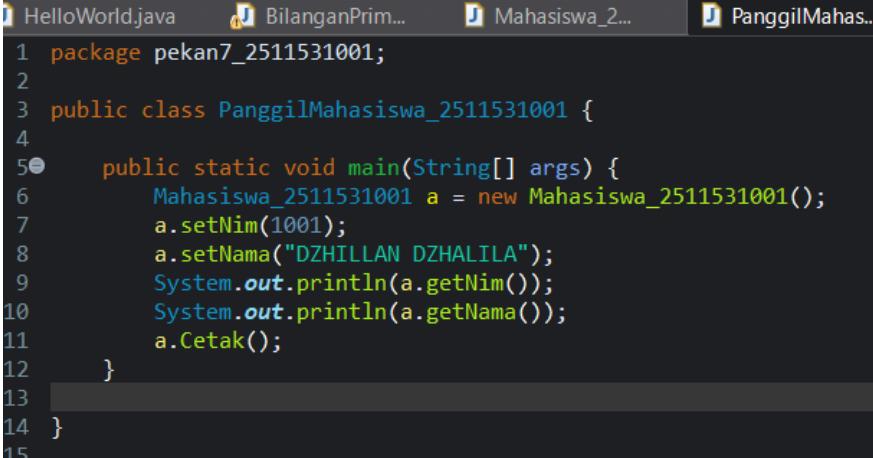
### 2.2.2 Penjelasan Program

Program ini merupakan rancangan *method* yang bisa digunakan untuk program lain. Program ini bermula dari penyettingan variabel global yang diberikan akses *private*. Karena variabel global tidak bisa untuk diakses langsung dari luar *class*, maka dibuatlah *method setter* untuk mengisi nilainya. Tidak lupa membuat *method getter* untuk mengambil nilainya kembali.

Dua *method* “Cetak()” digunakan untuk menampilkan data mahasiswa yang sudah diambil kembali oleh *method getter*. *Method* cetak pertama tidak memiliki parameter dan akan mencetak nilai “nim” dan “nama” yang sudah disimpan dalam objek. Sedangkan *method* kedua merupakan versi *overload* yang menerima parameter “nim” dan “nama” secara langsung, lalu mencetaknya tanpa menyimpan ke variabel global.

Untuk menggunakan *class* ini di program lain, langkah pertama adalah membuat objek dari *class* program ini di dalam *method main* pada *class* lain. Setelah objek dibuat, kita bisa memanggil *method setter* untuk mengisi data mahasiswa, lalu menggunakan *method Cetak ()* untuk menampilkan data tersebut.

## 2.3 Program Panggil Mahasiswa



```

1 package pekan7_2511531001;
2
3 public class PanggilMahasiswa_2511531001 {
4
5     public static void main(String[] args) {
6         Mahasiswa_2511531001 a = new Mahasiswa_2511531001();
7         a.setNim(1001);
8         a.setNama("DZHILLAN DZHALLIA");
9         System.out.println(a.getNim());
10        System.out.println(a.getNama());
11        a.Cetak();
12    }
13
14 }
15

```

Kode 2.4 : Kode Program Panggil Mahasiswa

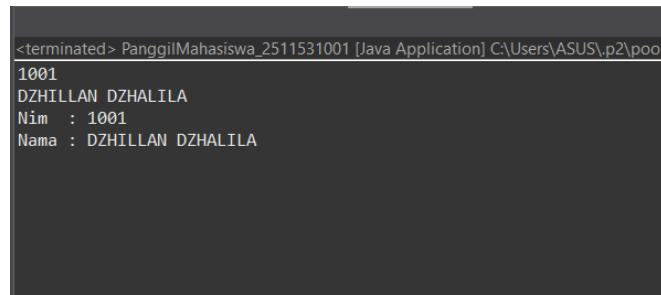
### 2.3.1. Langkah-Langkah

1. Mahasiswa\_2511531001 a = new Mahasiswa\_2511531001(); = membuat objek “a” yang akan digunakan untuk mengakses *method* dan menyimpan data mahasiswa.
2. a.setNim(1001); = memanggil *method* setNim()
3. a.setNama("DZHILLAN DZHALLIA"); = memanggil *method* setNama(String nama)
4. System.out.println(a.getNim());,  
System.out.println(a.getNama()) = menampilkan data dengan *method getter*, data yang ditampilkan adalah data yang telah disimpan dalam objek menggunakan *method getNim()* dan *getNama()*.
5. a.Cetak() + menampilkan data mahasiswa secara lengkap dengan format yang sudah ditentukan di dalam *method Cetak()* dari *class* Mahasiswa\_2511531001.

### 2.3.2. Penjelasan Program

Alur program ini dimulai dari dalam *method* main, objek dibuat dari *class* mahasiswa, lalu digunakan untuk mengakses *method* setter seperti setNim() dan setNama() guna mengisi data mahasiswa. Namun, pemanggilan setNim() tidak disertai parameter, padahal *method* tersebut membutuhkan input bertipe int, sehingga perlu diperbaiki agar program dapat berjalan dengan benar.

Setelah data dimasukkan, program melanjutkan dengan mencetak nilai nim dan nama menggunakan *method* getter getNim() dan getNama(), yang mengambil nilai dari variabel global dalam objek. Hasilnya ditampilkan ke layar menggunakan System.out.println. Sebagai langkah akhir, *method* Cetak() dipanggil untuk menampilkan data mahasiswa secara lengkap dengan format yang telah ditentukan dalam *class* Mahasiswa\_2511531001. Program ini menunjukkan alur dasar pemrograman berorientasi objek, mulai dari pembuatan objek, pengisian data, pengambilan data, hingga pencetakan informasi. Output dari program ini adalah:



```
<terminated> PanggilMahasiswa_2511531001 [Java Application] C:\Users\ASUS\p2\pool
1001
DZHILLAN DZHALILA
Nim : 1001
Nama : DZHILLAN DZHALILA
```

Gambar 2.5 : Ouput Kode Program Panggil Mahasiswa

## 2.4 Program Panggil Mahasiswa 2

```

1 package pekan7_2511531001;
2 import java.util.Scanner;
3 public class PanggilMahasiswa2_2511531001 {
4
5     public static void main(String[] args) {
6         Scanner input = new Scanner(System.in);
7         System.out.print("NIM :");
8         String x = input.nextLine();
9         System.out.print("Nama :");
10        String y = input.nextLine();
11        Mahasiswa_2511531001 a = new Mahasiswa_2511531001();
12        a.setNim(x);
13        a.setNama(y);
14
15        if (x.startsWith("25")){
16            System.out.println(y + " anda angkatan 2025");
17        }
18        if (x.contains("1153")){
19            System.out.println(y + " Anda Mahasiswa Informatika");
20        }
21        a.Cetak2();
22        input.close();
23    }
24
25 }
26

```

Gambar 2.6 : Kode Panggil Mahasiswa 2

### 2.4.1. Langkah-Langkah Program

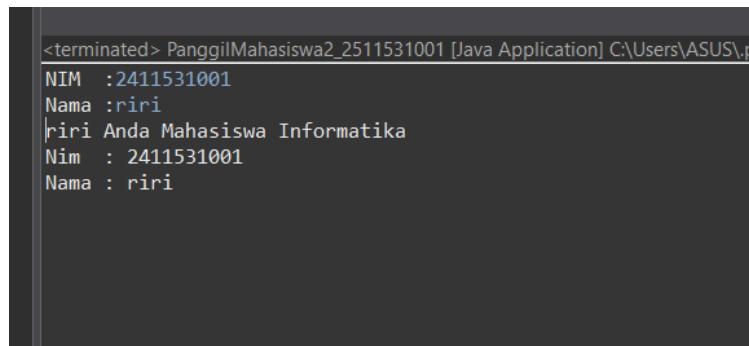
1. `import java.util.Scanner` = mengimpor *class Scanner* untuk membaca input dari keyboard.\
2. Input data dari pengguna
  - a. `System.out.print("NIM ")`; = menampilkan perintah untuk memasukkan NIM.
  - b. `String x = input.nextLine();` = menyimpan input NIM ke variabel x.
  - c. `System.out.print("Nama ")`; = menampilkan perintah untuk memasukkan nama.
  - d. `String y = input.nextLine();` = menyimpan input nama ke variabel y.
3. `Mahasiswa_2511512015 a = new Mahasiswa_2511512015()`; = membuat objek a dari *class Mahasiswa\_2511531001*
4. Mengisi data ke objek;

- a. a.setNim(x) = mengisi NIM ke objek menggunakan *method setter*.
  - b. a.setNama(y) = mengisi nama ke objek menggunakan *method setter*.
5. Pengecekan angkatan :
- a. f (x.startsWith("25")) = mengecek apakah NIM dimulai dengan "25".
  - b. System.out.println(y + " anda angkatan 2025") = menampilkan pesan jika kondisi terpenuhi.
6. Pengecekan jurusan:
- a. if (x.contains("1153")) Mengecek apakah NIM mengandung "1153".
  - b. System.out.println(y + " Anda Mahasiswa Informatika"); Menampilkan pesan jika kondisi terpenuhi.
7. a.Cetak2(); = memanggil *method Cetak2()* untuk menampilkan data mahasiswa.
8. input.close(); Menutup objek Scanner untuk menghindari kebocoran resource.

#### 2.4.2 Penjelasan Program

Program PanggilMahasiswa2\_2511512015 dimulai dengan membaca input dari pengguna menggunakan objek Scanner. Pengguna diminta untuk memasukkan NIM dan nama, yang kemudian disimpan dalam variabel x dan y. Setelah itu, program membuat objek a dari class Mahasiswa\_2511512015, lalu mengisi data ke dalam objek tersebut menggunakan method setNim(x) dan setNama(y). Data yang dimasukkan akan disimpan dalam variabel internal objek, sesuai dengan prinsip enkapsulasi dalam pemrograman berorientasi objek.

Setelah data mahasiswa tersimpan, program melakukan dua pengecekan terhadap NIM. Jika NIM dimulai dengan "25", maka program menampilkan pesan bahwa mahasiswa tersebut adalah angkatan 2025. Jika NIM mengandung "1153", maka program menampilkan bahwa mahasiswa tersebut berasal dari jurusan Informatika. Sebagai langkah akhir, program memanggil method Cetak2() dari objek a untuk menampilkan data mahasiswa secara lengkap, lalu menutup objek Scanner untuk mengakhiri proses input. Alur ini menunjukkan integrasi antara input pengguna, manipulasi objek, dan logika kondisi berbasis string. Output dari program ini adalah :



```
<terminated> PanggilMahasiswa2_2511531001 [Java Application] C:\Users\ASUS\p
NIM :2411531001
Nama :riri
riri Anda Mahasiswa Informatika
Nim : 2411531001
Nama : riri
```

Gambar 2.7 : Ouput Kode Program Mahasiswa 2

## 2.5 Program String 1

```

1 HelloWorld.java   2 MembuatKue.java   3 SentinelLoop...   4 PanggilMahas...   5 String1.java
1 package pekan7_2511531001;
2
3 public class String1_2511531001 {
4
5     public static void main(String[] args) {
6         String salam = "Assalamu'alaikum";
7         System.out.println("panjang salam adalah " + salam.length());
8         System.out.println(salam.toUpperCase());
9         System.out.println(salam.toLowerCase());
10        System.out.println(salam.indexOf("salam"));
11    }
12
13 }
14

```

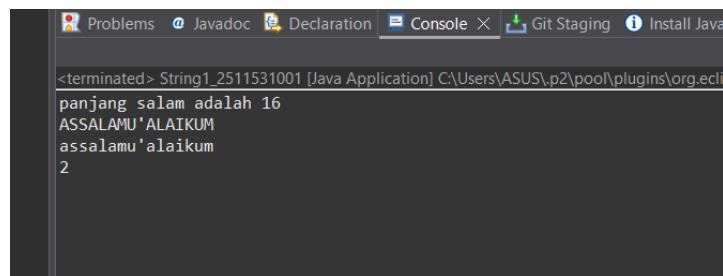
Gambar 2.8 : Kode Program String 1

### 2.5.1. Langkah-Langkah

1. String salam = "Assalamu'alaikum"; = Mendeklarasikan variabel salam bertipe String dan mengisinya dengan teks "Assalamu'alaikum".
2. System.out.println("panjang salam adalah " + salam.length()); = Menampilkan panjang karakter dari string salam menggunakan method .length().
3. System.out.println(salam.toUpperCase()); = Menampilkan string salam dalam huruf kapital seluruhnya menggunakan method .toUpperCase().
4. System.out.println(salam.toLowerCase()); = Menampilkan string salam dalam huruf kecil seluruhnya menggunakan method .toLowerCase().
5. System.out.println(salam.indexOf("salam")); = Menampilkan posisi indeks awal dari substring "salam" dalam string salam menggunakan method .indexOf().

### 2.5.2 Penjelasan Program

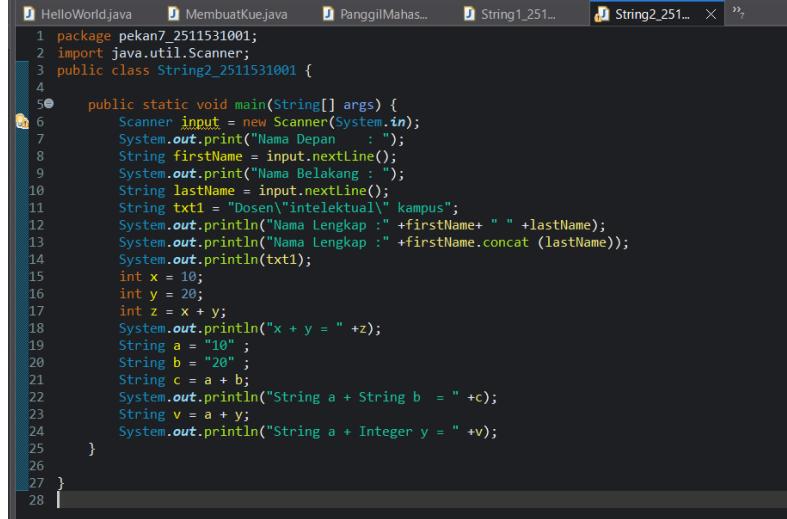
Program String1\_2511531001 dimulai dengan mendeklarasikan sebuah variabel bertipe String bernama salam yang berisi teks "Assalamu'alaikum". Setelah itu, program menjalankan beberapa operasi string menggunakan method bawaan dari class String. Pertama, program mencetak panjang dari string salam menggunakan method .length(), lalu mencetak versi huruf kapital dari string tersebut dengan .toUpperCase(), dan versi huruf kecilnya dengan .toLowerCase(). Terakhir, program mencari posisi awal dari substring "salam" di dalam string salam menggunakan .indexOf("salam"). Output dari program ini adalah:



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the following text:  
<terminated> String1\_2511531001 [Java Application] C:\Users\ASUS\.p2\pool\plugins\org.eclipse.jdt.core\src\panjang\_salam\_adalah\_16  
ASSALAMU'ALAIKUM  
assalamu'alaikum  
2

Gambar 2.9 : Output Kode Program String 1

## 2.6 Program String 2



```

1 package pekan7_2511531001;
2 import java.util.Scanner;
3 public class String2_2511531001 {
4
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

```

Gambar 2.10 : Kode Program String 2

### 2.6.1. Langkah-Langkah

1. Scanner input = new Scanner(System.in); = Membuat objek Scanner untuk membaca input dari keyboard.
2. System.out.print("Nama Depan "); = Menampilkan prompt agar pengguna memasukkan nama depan.
3. String firstName = input.nextLine(); = Menyimpan input nama depan ke variabel firstName.
4. System.out.print("Nama Belakang "); = Menampilkan prompt agar pengguna memasukkan nama belakang.
5. String lastName = input.nextLine(); = Menyimpan input nama belakang ke variabel lastName.
6. String txt1 = "Dosen" + "Institut" + "Kampus"; = Menggabungkan tiga string menjadi satu dan menyimpannya ke variabel txt1.
7. System.out.println("Nama Lengkap = " + firstName + " " + lastName); = Menampilkan nama lengkap dengan spasi di antara nama depan dan belakang.

8. System.out.println("Nama Lengkap = " +  
firstName.concat(lastName)); = Menampilkan nama lengkap  
tanpa spasi menggunakan method .concat().
9. System.out.println(txt1); = Menampilkan hasil penggabungan  
string "DosenInstitutKampus".
10. int x = 10; int y = 20; = Mendeklarasikan dua variabel integer x  
dan y.
11. System.out.println("x + y = " + (x + y)); = Menampilkan hasil  
penjumlahan angka: x + y = 30.
12. String a = "10"; String b = "20"; = Mendeklarasikan dua variabel  
string berisi angka.
13. String c = a + b; = Menggabungkan string "10" dan "20" menjadi  
"1020".
14. System.out.println("String a + String b = " + c); = Menampilkan  
hasil penggabungan string: "String a + String b = 1020".
15. String w = a + y; = Menggabungkan string "10" dengan integer  
20, hasilnya "1020" karena integer otomatis dikonversi ke string.
16. System.out.println("String a + Integer y = " + w); = Menampilkan  
hasil penggabungan string dan integer: "String a + Integer y =  
1020"

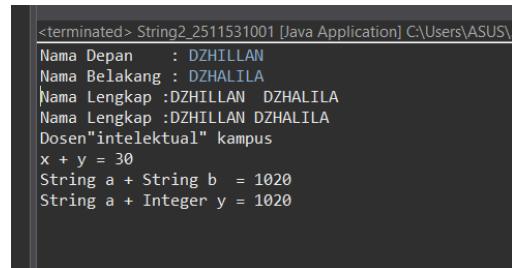
#### 2.6.2. Penjelasan Program

Program String2\_2511531001 dimulai dengan membaca input dari pengguna menggunakan objek Scanner. Pengguna diminta untuk memasukkan nama depan dan nama belakang, yang kemudian disimpan dalam variabel firstName dan lastName. Setelah itu, program membuat sebuah string baru txt1 yang merupakan gabungan dari tiga kata: "Dosen", "Institut", dan "Kampus". Program lalu mencetak nama lengkap pengguna dalam dua format: pertama dengan spasi di antara nama depan dan belakang menggunakan

operator +, dan kedua tanpa spasi menggunakan method .concat(). Kemudian, string txt1 ditampilkan ke layar.

Selanjutnya, program mendeklarasikan dua variabel integer x dan y, lalu mencetak hasil penjumlahannya. Program juga mendeklarasikan dua string "10" dan "20" sebagai a dan b, lalu menggabungkannya menjadi "1020" dan mencetak hasilnya. Terakhir, string a digabungkan dengan integer y, menghasilkan "1020" karena integer otomatis dikonversi menjadi string saat digabungkan. Output program akan menampilkan nama lengkap pengguna, hasil penggabungan string, hasil penjumlahan angka, dan perbedaan antara penjumlahan numerik dan penggabungan string.

Output dari program ini adalah :



```
<terminated> String2_2511531001 [Java Application] C:\Users\ASUS\.
Nama Depan : DZHILLAN
Nama Belakang : DZHALILA
Nama Lengkap :DZHILLAN DZHALILA
Nama Lengkap :DZHILLAN DZHALILA
Dosen"intelektual" kampus
x + y = 30
String a + String b = 1020
String a + Integer y = 1020
```

Gambar 2.11 : Ouput Program String 2

## **BAB III**

### **KESIMPULAN**

Dari praktikum yang telah dilaksanakan, dapat disimpulkan bahwa praktikum ini memperkenalkan dan memperdalam pemahaman mahasiswa terhadap konsep dasar pemrograman Java, khususnya dalam konteks pemrograman berorientasi objek (OOP) dan manipulasi string. Melalui berbagai program yang telah dijalankan, mahasiswa belajar bagaimana membuat dan menggunakan class, mendeklarasikan atribut dengan akses kontrol private, serta mengimplementasikan method setter dan getter untuk menjaga prinsip enkapsulasi. Program seperti Mahasiswa\_2511531001 dan PanggilMahasiswa\_2511531001 menunjukkan bagaimana objek dapat dibuat, diisi datanya, dan digunakan untuk mencetak informasi secara modular. Selain itu, program PanggilMahasiswa2\_2511512015 memperluas konsep dengan menambahkan logika kondisi berbasis string, seperti pengecekan angkatan dan jurusan berdasarkan pola NIM.

Di sisi lain, praktikum juga membahas manipulasi string secara langsung melalui method bawaan Java seperti .length(), .toUpperCase(), .toLowerCase(), .indexOf(), dan .concat(). Program String1\_2511531001 dan String2\_2511531001 memberikan gambaran jelas tentang bagaimana teks dapat diolah, digabungkan, dan dianalisis, serta membedakan antara operasi numerik dan penggabungan string. Dengan menggabungkan input pengguna, logika kondisi, dan pemrosesan data, seluruh rangkaian praktikum ini membekali mahasiswa dengan keterampilan dasar yang penting untuk membangun aplikasi Java yang terstruktur, efisien, dan interaktif.

## **DAFTAR PUSTAKA**

- 1] DuniaIlkom, “Tutorial OOP Java: Cara Membuat Method bahasa Java,” Oct. 11, 2021. [Daring]. Tersedia: <https://www.duniaIlkom.com/tutorial-oop-java-cara-membuat-method-bahasa-java/> [Diakses : 13 Nov 2025].
- [2] STMIK STEKOM, *Modul Praktikum Pemrograman Berorientasi Objek*, Fakultas Ekonomi dan Bisnis, 2022, pp. 23–31, 45–52. [Daring]. Tersedia: [https://digilib.stekom.ac.id/assets/dokumen/ebook/feb\\_BMuBPtvpXwUkhZqdyUPA7LyV7948c7ZdhjGj8z2EkAjSpNgD\\_njQSpM\\_1656322622.pdf](https://digilib.stekom.ac.id/assets/dokumen/ebook/feb_BMuBPtvpXwUkhZqdyUPA7LyV7948c7ZdhjGj8z2EkAjSpNgD_njQSpM_1656322622.pdf) . [Diakses: 13 Nov 2025].