

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN
LAPORAN PRAKTIKUM PEKAN 7

Disusun Oleh:

Syasya Halwa Gazwani
(2511531018)

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

Assalamu'alaikum warahmatullahi wabarakatuh,

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.

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. Penulis mengucapkan terima kasih kepada bapak Dr. Wahyudi, S.T, M.T. selaku dosen mata kuliah Algoritma Pemrograman yang telah memberikan tugas ini sehingga dapat menambah pengetahuan dan wawasan sesuai dengan mata kuliah yang penulis tekuni. Penulis juga mengucapkan terima kasih kepada semua pihak yang telah membagi sebagian pengetahuannya sehingga penulis dapat menyelesaikan Laporan Praktikum ini.

Padang, November 2025

Penulis

DAFTAR ISI

| | |
|----------------------------------|----|
| KATA PENGANTAR..... | i |
| DAFTAR ISI..... | ii |
| BAB I PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Tujuan..... | 1 |
| BAB II PEMBAHASAN..... | 3 |
| 2.1 Bilangan Prima | 3 |
| 2.1.1 Kode Program | 3 |
| 2.1.2 Langkah Kerja | 3 |
| 2.1.3 Analisis Hasil | 4 |
| 2.2 Mahasiswa | 5 |
| 2.2.1 Kode Program | 5 |
| 2.2.2 Langkah Kerja | 5 |
| 2.3 Panggil Mahasiswa | 7 |
| 2.3.1 Kode Program | 7 |
| 2.3.2 Langkah Kerja | 7 |
| 2.3.3 Analisis Hasil | 8 |
| 2.4 Panggil Mahasiswa 2 | 8 |
| 2.4.1 Kode Program | 8 |
| 2.4.2 Langkah Kerja | 9 |
| 2.4.3 Analisis Hasil | 11 |
| 2.5 String 1 | 11 |
| 2.5.1 Kode Program | 11 |
| 2.5.2 Langkah Kerja | 12 |
| 2.5.1 Analisis Hasil | 12 |
| 2.6 String 2 | 13 |
| 2.6.1 Kode Program | 13 |
| 2.6.2 Langkah Kerja | 14 |
| 2.6.3 Analisis Hasil | 15 |
| KESIMPULAN | 16 |
| DAFTAR PUSTAKA | 17 |

BAB I PENDAHULUAN

1.1 Latar Belakang

Pemrograman atau *programming* merupakan sebuah proses menulis, menguji dan memperbaiki (*debug*), serta memelihara kode yang dapat membangun suatu program komputer. Kode ini ditulis dalam berbagai bahasa pemrograman atau sering disebut juga bahasa komputer. Tujuan dari pemrograman adalah untuk memuat suatu program yang dapat melakukan suatu perhitungan atau “pekerjaan” sesuai dengan keinginan pemrograman. Untuk melakukan pemrograman, diperlukan keterampilan dalam algoritma, logika, bahasa pemrograman, dan pada banyak kasus, pengetahuan-pengetahuan lain seperti matematika. Bahasa Pemrograman, atau biasanya disebut bahasa komputer atau *computer language programming*, yaitu sebuah instruksi standar untuk mengendalikan sebuah komputer. Seperangkat aturan *syntax* dan *semantic* yang digunakan untuk memberikan sebuah definisi pada program komputer dikenal sebagai bahasa pemrograman. Dengan menggunakan bahasa ini, seorang *programmer* dapat dengan tepat menentukan data yang akan diproses oleh komputer, bagaimana penyimpanan dan transfernya, dan tindakan apa yang harus diambil dalam berbagai keadaan.

1.2 Tujuan

- 1.2.1 Memahami konsep dasar tipe data dalam bahasa pemrograman Java serta penerapannya dalam pembuatan program sederhana.
- 1.2.2 Mengembangkan keterampilan analisis terhadap hasil keluaran program sesuai dengan logika yang diterapkan.

1.3 Manfaat Praktikum

- 1.3.1 Menambah wawasan dan keterampilan mahasiswa dalam mengelola data menggunakan bahasa pemrograman Java sebagai dasar untuk pemrograman yang lebih lanjut.

1.3.2 Memberikan pemahaman dasar kepada mahasiswa mengenai penggunaan tipe data dalam bahasa pemrograman Java, sehingga dapat menjadi bahan dalam mengembangkan program yang lebih kompleks di tahap berikutnya.

BAB II PEMBAHASAN

2.1 Bilangan Prima

2.1.1 Kode Program

```

1 package pekan7_2511531018;
2
3 import java.util.Scanner;
4
5 public class BilanganPrima_2511531018 {
6     public static boolean isPrime(int n) {
7         int factors = 0;
8         for (int i = 1; i<= n; i++) {
9             if (n % i == 0) {
10                 factors++;
11             }
12         }
13         return (factors == 2);
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
27 }
28 }
```

Gambar 2.1.1 Kode program Bilangan Prima.

2.1.2 Langkah Kerja

- 2.1.2.1 Semua kode berada dalam *class* BilanganPrima_2511531018 dan fungsi *isPrime(int n)* untuk menentukan apakah n adalah bilangan prima atau bukan.

```

5 public class BilanganPrima_2511531018 {
6     public static boolean isPrime(int n) {
```

Gambar 2.1.2.1 Deklarasi *class* dan fungsi *isPrime(int n)*.

- 2.1.2.2 *Factors* digunakan untuk menghitung berapa banyak pembagi dari n. Setiap kali $n \% i == 0$, berarti i adalah pembaginya (*factors++*), dan bilangan prima harus tepat memiliki 2 pembagi yaitu 1 dan dirinya sendiri.

```

7         int factors = 0;
8         for (int i = 1; i<= n; i++) {
9             if (n % i == 0) {
10                 factors++;
```

Gambar 2.1.2.2 Cara kerja fungsi *isPrime(int n)*.

2.1.2.3 Fungsi mengembalikan *true* jika *factors == 2*.

```
13         return (factors == 2);
```

Gambar 2.1.2.3 Fungsi *return*.

2.1.2.4 Membuat *Scanner*, program meminta pengguna memasukkan nilai n, dan nilai tersebut disimpan ke variabel a.

```
16  public static void main(String[] args) {  
17      Scanner input = new Scanner(System.in);  
18      System.out.print("Input nilai n = ");  
19      int a= input.nextInt();  
20  }
```

Gambar 2.1.2.4 *Scanner* dan menyimpan nilai dalam variabel.

2.1.2.5 Program memanggil fungsi *isPrime(a)*, jika hasilnya *true* maka cetak bahwa a adalah bilangan prima, dan jika *false* cetak bahwa a bukan bilangan prima.

```
20  if (isPrime(a)) {  
21      System.out.println(a+" bilangan prima");  
22  } else {  
23      System.out.println(a+" bukan bilangan prima");  
24  }
```

2.1.2.5 Memastikan nilai adalah bilangan prima atau tidak.

2.1.3 Analisis Hasil

| | |
|-------------------|------------------------|
| Input nilai n = 5 | Input nilai n = 9 |
| 5 bilangan prima | 9 bukan bilangan prima |

Gambar 2.1.3 Hasil *output* kode program Bilangan Prima.

Untuk *input* 5, program menghitung berapa banyak angka yang dapat dibagi 5 dengan hasil sisa nol. Setelah dilakukan pengecekan dari 1 sampai 5, hanya angka 1 dan 5 yang dapat membagi 5 secara tepat sehingga jumlah pembaginya ada dua. Karena bilangan prima didefinisikan sebagai bilangan yang hanya memiliki dua faktor, yaitu 1 dan dirinya sendiri, maka program menyimpulkan bahwa 5 adalah bilangan prima dan menampilkannya sebagai *output*.

Untuk *input* 9, proses yang sama dilakukan. Program memeriksa pembagi dari 1 sampai 9, dan menemukan bahwa angka 9 dapat dibagi oleh 1, 3, dan 9. Karena jumlah pembaginya ada tiga, angka ini tidak memenuhi syarat sebagai bilangan prima. Oleh karena itu, program menampilkan bahwa 9 bukan bilangan prima. Analisis ini menunjukkan bahwa program bekerja dengan cara menghitung jumlah faktor suatu angka, lalu menentukan apakah angka tersebut prima berdasarkan jumlah faktornya.

2.2 Mahasiswa

2.2.1 Kode Program

```

1 package pekan7_2511531018;
2
3 public class Mahasiswa_2511531018 {
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    //metoda 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 2.2.1 Kode program Mahasiswa.

2.2.2 Langkah Kerja

2.2.2.1 Atribut dibuat *private* supaya tidak bisa diakses langsung dari luar kelas.

```

5      private int nim;
6      private String nama, nim2;

```

Gambar 2.2.2.1 Deklarasi variabel.

- 2.2.2.2 *Setter* digunakan untuk mengisi atau mengubah nilai atribut dan *this.nim* merujuk ke variabel dalam kelas, sedangkan parameter nim adalah *input* dari luar.

```

8@    public void setNim (int nim) {
9        this.nim=nim;

```

Gambar 2.2.2.2 Method *Setter*.

- 2.2.2.3 Digunakan untuk mengisi data mahasiswa.

```

11@    public void setNim2 (String nim2) {
12        this.nim2=nim2;
14@    public void setNama (String nama) {
15        this.nama=nama;

```

Gambar 2.2.2.3 Method *Setter*.

- 2.2.2.4 *Getter* digunakan untuk mengambil nilai atribut dan semua *getter* mengembalikan nilai atribut agar bisa digunakan oleh program lain.

```

18@    public int getNim() {
19        return nim;
21@    public String getNim2() {
22        return nim2;
24@    public String getNama() {
25        return nama;

```

Gambar 2.2.2.4 Method *getter*.

- 2.2.2.5 Method ini digunakan untuk menampilkan isi variabel. Method ini menampilkan nim dan nama.

```

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

```

Gambar 2.2.2.5 Method Cetak() dan Cetak2().

2.3 Panggil Mahasiswa

2.3.1 Kode Program

```

1 package pekan7_2511531018;
2
3 public class PanggilMahasiswa {
4     public static void main(String[] args) {
5         Mahasiswa_2511531018 a= new Mahasiswa_2511531018();
6         a.setNim(23532);
7         a.setNama("Rahmat");
8         System.out.println(a.getNim());
9         System.out.println(a.getNama());
10        a.Cetak();

```

Gambar 2.3.1 Kode program Panggil Mahasiswa.

2.3.2 Langkah Kerja

- 2.3.2.1 Membuat objek baru dari *class* Mahasiswa_2511531018 dan variabel lokal a menyimpan referensi ke objek tersebut.

```

5     Mahasiswa_2511531018 a= new Mahasiswa_2511531018();

```

Gambar 2.3.2.1 Membuat objek baru.

- 2.3.2.2 Memanggil *method* setNim pada objek a, memberi nilai NIM 23532 dan memanggil *method* setNama untuk menyimpan nama "Rahmat".

```

6         a.setNim(23532);
7         a.setNama("Rahmat");

```

Gambar 2.3.2.2 Memanggil *method* setNim dan setNama.

- 2.3.2.3 Cetak getNim() pada objek a untuk mengambil nilai NIM yang tersimpan, cetak getNama() untuk mengambil nama (“Rahmat”) dan Memanggil *method* Cetak() pada objek a.

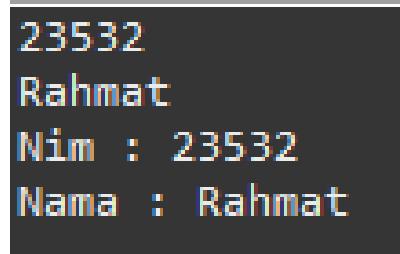
```

8      System.out.println(a.getNim());
9      System.out.println(a.getNama());
10     a.Cetak();

```

Gambar 2.3.2.3 Cetak hasil akhir.

2.3.3 Analisis Hasil



```

23532
Rahmat
Nim : 23532
Nama : Rahmat

```

Gambar 2.3.3 Hasil *output* kode program Panggil Mahasiswa.

Hasil yang muncul menunjukkan bahwa data seorang mahasiswa berhasil disimpan dan ditampilkan dengan benar. Angka dan nama yang pertama kali muncul menandakan bahwa data yang dimasukkan sudah tersimpan sesuai yang diinginkan. Setelah itu, program menampilkan kembali informasi yang sama dalam bentuk yang lebih rapi, lengkap dengan keterangan “Nim” dan “Nama”. Hal ini menunjukkan bahwa proses penyimpanan, pengambilan, dan penampilan data berjalan lancar. Program berhasil menampilkan identitas mahasiswa secara urut, mulai dari data mentah hingga format yang lebih informatif, tanpa adanya kesalahan dalam prosesnya.

2.4 Panggil Mahasiswa 2

2.4.1 Kode Program

```

1 package pekan7_2511531018;
2
3 import java.util.Scanner;
4
5 public class PanggilMahasiswa2_2511531018 {
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_2511531018 a = new Mahasiswa_2511531018();
13        a.setNim2(x);
14        a.setNama(y);
15        if(x.startsWith("25")) {
16            System.out.println(y + " anda angkatan 2025");
17        }
18        if(x.contains("1153")) {
19            System.out.println("Anda Mahasiswa Informatika");
20        }
21        a.Cetak2();
22        input.close();
23    }
24 }
```

Gambar 2.4.1 Kode program Panggil Mahasiswa 2.

2.4.2 Langkah Kerja

2.4.2.1 Program membuat objek *Scanner* untuk menerima data dari pengguna dan program menampilkan tulisan di layar agar pengguna memasukkan NIM, lalu menyimpan nilai yang diketik ke dalam variabel x.

```

7     Scanner input = new Scanner(System.in);
8     System.out.print("NIM: ");
9     String x= input.nextLine();
```

Gambar 2.4.2.1 Membuat objek *Scanner* dan menampilkan tulisan pada layar.

2.4.2.2 Program menampilkan tulisan agar pengguna memasukkan Nama dan hasil ketikan disimpan ke dalam variabel y.

```

10    System.out.print("Nama: ");
11    String y= input.nextLine();
```

Gambar 2.4.2.2 Menampilkan tulisan pada layar.

2.4.2.3 Setelah data didapat, program membuat objek mahasiswa baru dari *class* Mahasiswa_2511531018 dan program mengisi data mahasiswa tersebut dengan memanggil fungsi untuk

menyimpan NIM dan Nama yang sudah dimasukkan sebelumnya.

```
14      sambung(y);
15      sambung(x);
16      s = n66 MewMSIz66M eMSIz66M - S2T2T018():
```

Gambar 2.4.2.3 Membuat objek baru dan mengisi data Mahasiswa.

- 2.4.2.4 Program kemudian memeriksa apakah NIM diawali dengan angka “25”. Jika iya, maka program menampilkan pesan bahwa mahasiswa tersebut adalah angkatan 2025, disertai dengan nama yang dimasukkan.

```
15 if(x.startsWith("25")) {
16     System.out.println(y + " anda angkatan 2025");
```

Gambar 2.4.2.4 Memeriksa NIM.

- 2.4.2.5 Program lanjut memeriksa apakah NIM mengandung angka “1153”. Jika iya, program menampilkan pesan “Anda Mahasiswa Informatika” karena kode itu menandakan jurusan.

```
18 if(x.contains("1153")) {
19     System.out.println("Anda Mahasiswa Informatika");
```

Gambar 2.4.2.5 Memeriksa NIM mengandung angka 1153.

- 2.4.2.6 Setelah pemeriksaan selesai, program memanggil fungsi Cetak2() untuk menampilkan data lengkap mahasiswa dan objek Scanner ditutup.

```
21     a.Cetak2();
22     input.close();
```

Gambar 2.4.2.6 Fungsi Cetak2() dan menutup objek Scanner.

2.4.3 Analisis Hasil

```
NIM: 2511531018
Nama: halwa
halwa anda angkatan 2025
Anda Mahasiswa Informatika
Nim : 2511531018
Nama : halwa

NIM: 2411531018
Nama: naya
Anda Mahasiswa Informatika
Nim : 2411531018
Nama : naya
```

Gambar 2.4.3 Hasil output kode program Panggil Mahasiswa 2.

Hasil yang muncul menunjukkan bahwa program berhasil membaca dan memproses data yang dimasukkan oleh pengguna. Setelah pengguna mengetikkan NIM dan Nama, program langsung menampilkan kembali informasi tersebut sebagai bukti bahwa data telah tersimpan dengan benar. Selanjutnya, karena NIM yang dimasukkan dimulai dengan angka 25, program mengenali bahwa mahasiswa tersebut termasuk dalam angkatan 2025, sehingga muncul pesan yang menyebutkan nama diikuti dengan keterangan angkatan.

Kemudian, karena di dalam NIM terdapat angka 1153, program juga menampilkan pesan tambahan bahwa mahasiswa tersebut berasal dari jurusan Informatika. Setelah menampilkan kedua keterangan tersebut, program mencetak ulang data lengkap mahasiswa berupa NIM dan nama dalam format yang lebih rapi. Dari hasil tersebut dapat disimpulkan bahwa program telah berjalan dengan baik, berhasil melakukan pengecekan terhadap pola pada NIM, serta menampilkan data sesuai logika yang telah ditentukan.

2.5 String 1

2.5.1 Kode Program

```
1 package pekan7_2511531018;
2
3 public class String1_2511531018 {
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")); //Outputs 2
10    }
11 }
```

Gambar 2.5.1 Kode program String 1.

2.5.2 Langkah Kerja

- 2.5.2.1 Di dalamnya dibuat sebuah variabel teks bernama salam yang berisi tulisan "Assalamualaikum". Variabel ini digunakan untuk menampung kalimat yang akan diolah.

```
5     String salam = "Assalamualaikum";
```

Gambar 2.5.2.1 Membuat variabel.

- 2.5.2.2 Program kemudian menghitung jumlah karakter dalam teks tersebut menggunakan fungsi untuk mendapatkan panjang *string*, lalu menampilkannya bersama kalimat penjelasan di layar, program menampilkan kembali teks yang sama, tetapi seluruh hurufnya diubah menjadi huruf besar, program menampilkan teks yang sama dalam bentuk huruf kecil seluruhnya, dan program mencari posisi awal dari kata "salam" di dalam teks. Hasilnya berupa angka yang menunjukkan letak pertama kali kata tersebut muncul.

```
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
```

Gambar 2.5.2.2 Mencetak hasil.

2.5.1 Analisis Hasil

```
panjang salam adalah: 15
ASSALAMUALAIKUM
assalamualaikum
2
```

Gambar 2.5.1 Hasil *output* kode program String 1.

Hasil yang ditampilkan menunjukkan bahwa program berhasil melakukan beberapa operasi terhadap teks dengan benar. Angka **15**

menandakan jumlah seluruh karakter dalam kalimat yang disimpan, termasuk semua huruf tanpa spasi karena teksnya tidak memiliki jarak antar kata. Baris berikutnya menampilkan versi teks yang seluruh hurufnya diubah menjadi huruf besar, sedangkan baris sesudahnya menampilkan versi huruf kecil semua. Ini menunjukkan bahwa fungsi pengubah bentuk huruf bekerja sesuai yang diharapkan. Angka 2 di baris terakhir merupakan posisi awal dari kata “salam” di dalam teks, yang berarti bagian kata tersebut mulai ditemukan pada karakter ke 3 jika dihitung dari posisi pertama bernilai nol. Secara keseluruhan, hasil ini memperlihatkan bahwa program dapat menghitung panjang teks, mengubah bentuk huruf, serta menemukan posisi kata tertentu di dalam kalimat dengan benar.

2.6 String 2

2.6.1 Kode Program

```

1 package pekan7_2511531018;
2
3 public class String2_2511531018 {
4     public static void main(String[] args) {
5         String firstName = "Syifa";
6         String lastName = "Muhammad";
7         String txt1 = "Dosen\"intelektual\" kampus";
8         System.out.println("Nama Lengkap: "+firstName + " " + lastName);
9         System.out.println("Nama Lengkap: "+firstName.concat(lastName));
10        System.out.println(txt1);
11        int x = 10;
12        int y = 20;
13        int z = x + y;
14        System.out.println("x+ y= "+z);
15        String a = "10";
16        String b = "20";
17        String c = a + b;
18        System.out.println("String a + string b =" +c);
19        String v = a + y;
20        System.out.println("String a + integer y = " +v);
21
22    }
23
24 }
```

Gambar 2.6.1 Kode program String 2.

2.6.2 Langkah Kerja

- 2.6.2.1 Di dalamnya dibuat beberapa variabel bertipe *string* untuk menyimpan nama depan, nama belakang, dan sebuah kalimat yang berisi kata “Dosen”, “intelektual”, dan “kampus”.

```

5      String firstName = "Syifa";
6      String lastName = "Muhassanah";
7      String txt1 = "Dosen\"intelektual\" kampus";

```

Gambar 2.6.2.1 Membuat variabel.

- 2.6.2.2 Program kemudian menampilkan gabungan nama depan dan nama belakang dengan cara menyisipkan spasi di antara keduanya, sehingga membentuk nama lengkap yang lebih rapi, program juga menampilkan hasil penggabungan nama depan dan nama belakang menggunakan fungsi khusus untuk menggabungkan teks, dan kalimat yang disimpan di variabel teks lainnya juga ditampilkan di layar.

```

8      System.out.println("Nama Lengkap: "+firstName + " " + lastName);
9      System.out.println("Nama Lengkap: "+firstName.concat(lastName));
10     System.out.println(txt1);

```

Gambar 2.6.2.2 Menampilkan hasil.

- 2.6.2.3 Program selanjutnya membuat dua variabel bilangan bulat, lalu menjumlahkan keduanya dan menampilkan hasilnya di layar sebagai bentuk penjumlahan angka.

```

11      int x = 10;
12      int y = 20;
13      int z = x + y;
14      System.out.println("x+ y= "+z);

```

Gambar 2.6.2.3 Membuat dua variabel.

- 2.6.2.4 Program membuat dua variabel teks yang isinya berupa angka dalam bentuk tulisan, bukan angka sebenarnya.

```

15      String a = "10";
16      String b = "20";
17      String c = a + b;

```

Gambar 2.6.2.4 Membuat dua variabel berupa angka dalam bentuk tulisan.

2.6.2.5 Kedua teks tersebut digabungkan, bukan dijumlahkan secara matematis, melainkan disambung seperti kata, kemudian hasilnya ditampilkan dan program juga menggabungkan teks dengan angka sebenarnya. Dalam hal ini angka diubah secara otomatis menjadi teks sebelum digabung, sehingga hasilnya juga menjadi gabungan tulisan, bukan penjumlahan.

```

18     System.out.println("String a + string b =" + c);
19     String v = a + y;
20     System.out.println("String a + integer y =" + v);

```

Gambar 2.6.2.5 Mencetak hasil.

2.6.3 Analisis Hasil

```

Nama Lengkap: Syifa Muhsanah
Nama Lengkap: SyifaMuhsanah
Dosen "intelektual" kampus
x+ y= 30
String a + string b =1020
String a + integer y =1020

```

Gambar 2.6.3 Hasil *output* kode program String 2.

Hasilnya menunjukkan bahwa program berhasil menggabungkan teks dan angka dengan benar. Nama depan dan belakang digabung membentuk “Syifa Muhsanah”, lalu kalimat tambahan juga tampil sesuai isi variabel. Penjumlahan angka menghasilkan nilai murni 30, sedangkan penggabungan teks “10” dan “20” menghasilkan “1020”. Saat teks digabung dengan angka, hasilnya juga menjadi tulisan, bukan hasil hitung. Ini membuktikan bahwa Java membedakan antara operasi penjumlahan dan penggabungan teks.

KESIMPULAN

Kumpulan kode program yang telah dijelaskan sebelumnya menunjukkan penerapan konsep dasar dalam pemrograman Java, khususnya dalam pengolahan data bertipe *string* dan *numerik*. Melalui berbagai contoh, program memperlihatkan bagaimana bahasa Java dapat digunakan untuk melakukan manipulasi teks, penggabungan data, serta pengolahan logika sederhana untuk menampilkan informasi yang sesuai dengan kondisi yang ditentukan.

Pada bagian yang berhubungan dengan teks, program menunjukkan kemampuan Java dalam melakukan manipulasi *string*, seperti menghitung panjang teks, mengubah huruf menjadi kapital atau huruf kecil, mencari posisi suatu kata dalam kalimat, dan menggabungkan dua atau lebih potongan teks menjadi satu kesatuan yang utuh. Hal ini mencerminkan bahwa pengolahan data bertipe *string* dalam Java bersifat fleksibel dan efisien, serta sangat berguna dalam pembuatan aplikasi yang berinteraksi dengan pengguna melalui teks.

Dengan demikian, dapat disimpulkan bahwa keseluruhan program merupakan penerapan nyata dari konsep dasar pemrograman Java yang meliputi deklarasi variabel, manipulasi string, operasi numerik, logika kondisi, serta interaksi input dan output. Melalui contoh-contoh tersebut, pemahaman mengenai cara kerja Java dalam memproses data teks dan angka dapat diperoleh dengan lebih jelas. Program-program ini juga menjadi dasar penting bagi pengembangan aplikasi yang lebih kompleks di masa mendatang, baik dalam bidang pendidikan, administrasi, maupun sistem informasi berbasis komputer.

DAFTAR PUSTAKA

- [1] H. Schildt, *Java: The Complete Reference*, 12th ed. New York: McGraw-Hill Education, 2021.
- [2] J. Bloch, *Effective Java*, 3rd ed. Boston: Addison-Wesley, 2018.
- [3] P. Deitel and H. Deitel, *Java: How to Program, Early Objects*, 11th ed. Pearson Education, 2019.
- [4] C. Horstmann, *Core Java Volume I: Fundamentals*, 12th ed. Hoboken, NJ: Prentice Hall, 2022.
- [5] W. Savitch, *Absolute Java*, 6th ed. Boston: Pearson, 2016.
- [6] M. Naftalin and P. Wadler, *Java Generics and Collections*, Sebastopol, CA: O'Reilly Media, 2019.
- [7] D. Liang, *Introduction to Java Programming and Data Structures: Comprehensive Version*, 12th ed. Boston: Pearson, 2020.
- [8] R. Johnson, *Java SE 17 Development: A Comprehensive Guide*, London: Packt Publishing, 2022.