

LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGRAMAN

“String dan Class Java”

Dosen pengampu: Dr. Wahyudi S.T, M.T

Asisten Praktikum: Rahmad DRO



**Disusun oleh:
Muhammad Althaf Mulya
NIM: 2511533018**

**FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN INFORMATIKA
UNIVERSITAS ANDALAS
2025**

DAFTAR ISI

DAFTAR ISI.....	i
KATA PENGANTAR.....	ii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan Praktikum.....	1
1.3 Manfaat Praktikum.....	1
BAB II PEMBAHASAN.....	2
2.1 Bilangan Prima.....	2
2.2 Mahasiswa.....	5
2.3 Panggil Mahasiswa.....	8
2.4 Panggil Mahasiswa2.....	10
2.5 String 1.....	13
2.6 String 2.....	15
BAB III KESIMPULAN.....	17
3.1 Ringkasan Hasil Praktikum.....	17
DAFTAR PUSTAKA.....	iii

KATA PENGANTAR

Puji syukur kehadirat Allah SWT atas limpahan rahmat dan karunia-Nya sehingga laporan praktikum dengan judul “Class dan String dalam Bahasa Pemrograman Java” ini dapat saya susun dan selesaikan dengan baik. Laporan ini disusun sebagai salah satu bentuk pelaksanaan kegiatan praktikum pada mata kuliah Algoritma dan Pemrograman.

Ucapan terima kasih saya sampaikan kepada Bapak Dr. Wahyudi, S.T., M.T. selaku dosen pengampu, serta kepada seluruh asisten laboratorium yang telah memberikan bimbingan dan arahan selama kegiatan praktikum berlangsung.

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pemahaman mengenai class, object, dan String sangat penting dalam pembelajaran Java karena menjadi dasar dari pemrograman berorientasi objek.

Melalui praktikum ini, mahasiswa mempelajari cara membuat class, memanggil object, serta menggunakan berbagai method pada String. Contoh program yang digunakan meliputi pengecekan bilangan prima, class Mahasiswa, dan manipulasi String.

1.2 Tujuan Praktikum

1. Memahami konsep class, object, atribut, dan method.
2. Menerapkan class dalam program seperti pada contoh Mahasiswa.
3. Menggunakan method pada String untuk pengolahan teks.
4. Melatih kemampuan membuat program Java yang sederhana dan terstruktur.

1.3 Manfaat Praktikum

1. Meningkatkan pemahaman dasar OOP di Java.
2. Mampu membuat program menggunakan class dan object.
3. Mampu memanfaatkan method pada String.
4. Menambah pengalaman dalam menyusun program Java secara rapi dan benar.

BAB II

PEMBAHASAN

2.1 Program Bilangan Prima

Kode program:

```

akalpro_2025_D_2511533019. > src > pekan7_2511533018 > BilanganPrima_2511533018.java >
1  package pekan7_2511533018;
2
3  import java.util.Scanner;
4
5  public class BilanganPrima_2511533018 {
6
7
8      public static boolean isPrime(int n) {
9          int factors = 0;
10         for (int i = 1; i <= n; i++) {
11             if (n % i == 0) {
12                 factors++;
13             }
14             System.out.println("count " + i);
15         }
16         return (factors == 2);
17     }
18
19     public static void main(String[] args) {
20         Scanner input = new Scanner(System.in);
21         System.out.println("Input nilai n = ");
22
23         int a = input.nextInt();
24
25         if (isPrime(a)) {
26             System.out.println(a + " bilangan prima");
27         } else {
28             System.out.println(a + " bukan bilangan prima");
29         }
30         input.close();
31     }
32
33 }
34

```

Program ini dibuat untuk mengecek apakah suatu bilangan merupakan bilangan prima atau bukan menggunakan bahasa pemrograman Java. Program ini memanfaatkan sebuah method `isPrime()` untuk menghitung jumlah faktor dari bilangan yang diinputkan. Jika jumlah faktornya tepat dua (1 dan dirinya sendiri), maka bilangan tersebut adalah bilangan prima. Selain itu, program menggunakan input dari pengguna melalui class `Scanner` untuk membaca nilai yang akan diuji.

Langkah Penyelesaian:

1. Import class `Scanner` untuk menerima input dari keyboard.
2. Buat method `isPrime(int n)` yang bertugas menghitung faktor dari suatu bilangan:
 - a. Deklarasi variabel `factors` dengan nilai awal 0.

- b. Gunakan perulangan for dari 1 sampai n.
 - c. Cek apakah $n \% i == 0$; jika iya, tambahkan factors.
 - d. Cetak proses perhitungan dengan `System.out.println("count " + i)`.
3. Kembalikan nilai true jika jumlah faktor sama dengan dua (bilangan prima).
4. Di dalam method `main()`:
 - a. Buat object Scanner untuk membaca input pengguna.
 - b. Tampilkan teks “Input nilai n = ” lalu simpan angka yang dimasukkan ke variabel a.
5. Panggil method `isPrime(a)` untuk menentukan apakah bilangan tersebut prima atau bukan.
6. Tampilkan hasil ke layar menggunakan `System.out.println()` sesuai kondisi:
 - a. Jika prima → tampilkan “bilangan prima”.
 - b. Jika bukan → tampilkan “bukan bilangan prima”.

Hasil:

Program akan menampilkan proses perhitungan faktor melalui teks “count ...” dari angka 1 sampai n. Setelah proses selesai, program akan mencetak apakah nilai yang diinputkan merupakan bilangan prima atau bukan. Jika jumlah faktor tepat dua, output menunjukkan bahwa bilangan tersebut adalah bilangan prima; jika lebih dari dua, maka bilangan tersebut bukan bilangan prima.

Output:

```
Input nilai n =
10
count 1
count 2
count 3
count 4
count 5
count 6
count 7
count 8
count 9
count 10
10 bukan bilangan prima
```

2.2 Program Mahasiswa

Kode program:

```
1 package pekan7_2511533018;
2
3 public class Mahasiswa_2511533018 {
4
5     private int nim;
6     private String nama, nim2;
7
8     public void setNim(int nim) {
9         this.nim = nim;
10    }
11
12    public void setNim2(String nim2) {
13        this.nim2 = nim2;
14    }
15
16    public void setName(String nama) {
17        this.nama = nama;
18    }
19
20    public int getNim() {
21        return nim;
22    }
23
24    public String getNim2() {
25        return nim2;
26    }
27
28    public String getName() {
29        return nama;
30    }
31
32    // method lain
33    public void Cetak() {
34        System.out.println("Nim " + nim);
35        System.out.println("Nama " + nama);
36    }
37
38    public void Cetak2() {
39        System.out.println("Nim " + nim2);
40        System.out.println("Nama " + nama);
41    }
42 }
```

Program ini dibuat untuk mendemonstrasikan penggunaan class, atribut (variable instance), serta method setter, getter, dan method cetak dalam konsep pemrograman berorientasi objek (OOP) pada Java. Class Mahasiswa_2511533018 berfungsi sebagai template untuk menyimpan data mahasiswa seperti NIM dan nama, serta menyediakan method untuk menampilkan data tersebut.

Langkah Penyelesaian:

1. Mendeklarasikan class Mahasiswa_2511533018 sebagai class utama yang berisi atribut dan method.
2. Membuat atribut private yaitu nim, nim2, dan nama sehingga hanya dapat diakses melalui method setter dan getter.
3. Membuat method setter (setNim, setNim2, setNama) untuk mengisi atau mengubah nilai atribut mahasiswa.
4. Membuat method getter (getNim, getNim2, getNama) untuk mengambil dan menampilkan nilai atribut.
5. Membuat method tambahan Cetak() untuk menampilkan data mahasiswa menggunakan NIM bertipe integer.
6. Membuat method Cetak2() untuk menampilkan data mahasiswa menggunakan NIM bertipe String.
7. Class ini akan dipanggil oleh class lain (seperti PanggilMahasiswa_2511533018) untuk membuat object mahasiswa dan menampilkan datanya.

Hasil:

Program akan menampilkan data mahasiswa sesuai nilai yang telah di-set melalui method setter.

- Jika menggunakan method Cetak(), output akan menampilkan NIM bertipe integer dan nama mahasiswa.
- Jika menggunakan Cetak2(), output akan menampilkan NIM bertipe String dan nama mahasiswa.

Class ini memungkinkan penyimpanan dan pengolahan data mahasiswa secara rapi, terstruktur, dan sesuai prinsip OOP pada Java.

2.3 Program Panggil Mahasiswa

Kode program:

```
1 package pekan7_2511533018;
2
3 public class PanggilMahasiswa_2511533018 {
4
5     Run | Debug
6     public static void main(String[] args) {
7
8         Mahasiswa_2511533018 ms = new Mahasiswa_2511533018();
9
10        ms.setNim(nim: 251153);
11        ms.setNama(nama: "Althaf");
12
13        System.out.println(ms.getNim());
14        System.out.println(ms.getNama());
15
16        ms.Cetak();
17    }
18 }
19
```

Program ini dibuat untuk memanggil class `Mahasiswa_2511533018`, membuat sebuah object mahasiswa, mengisi data mahasiswa melalui method setter, dan menampilkannya melalui method getter serta method `Cetak()`. Program ini menunjukkan cara penggunaan object dalam konsep OOP di Java.

Langkah Penyelesaian:

1. Membuat object baru dari class `Mahasiswa_2511533018` dengan sintaks:

```
Mahasiswa_2511533018 ms = new Mahasiswa_2511533018();
```

2. Mengisi data mahasiswa menggunakan method setter, yaitu:
 - a. `ms.setNim(251153);` untuk mengatur NIM
 - b. `ms.setNama("Althaf");` untuk mengatur nama
3. Mengambil data yang telah di-set menggunakan method getter:
 - a. `ms.getNim()` untuk menampilkan NIM
 - b. `ms.getNama()` untuk menampilkan nama

4. Memanggil method tambahan Cetak() yang berada di dalam class Mahasiswa untuk menampilkan data mahasiswa secara lengkap.
5. Program kemudian mencetak semua hasil ke layar menggunakan System.out.println().

Hasil:

Program akan mencetak NIM dan nama mahasiswa dua kali:

1. Pertama, melalui pemanggilan getter (getNim() dan getNama()).
2. Kedua, melalui pemanggilan method Cetak() yang menampilkan data dalam format:

Output:

```
Althaf  
Nim 251153  
Nama Althaf
```

2.4 Program Panggil Mahasiswa 2

Kode program:

```

package pekan7_2511533018;

import java.util.Scanner;

public class PanggilMahasiswa2_2522533018 {

    Run | Debug
    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.println(x: "Nama: ");
        String x = input.nextLine();

        System.out.println(x: "NIM: ");
        String y = input.nextLine();

        Mahasiwa_2511533018 ms = new Mahasiwa_2511533018();
        ms.setNim2(x);
        ms.setNama(y);

        if (x.startsWith(prefix: "25")) {
            System.out.println(ms.getNama() + " anda angkatan 2025.");
        }
        if (ms.getNim2().contains(s: "1153")) {
            System.out.println(y + " Anda mahasiswa Informatika.");
        }

        ms.Cetak2();
        input.close();
    }
}

```

Program ini dibuat untuk membaca data mahasiswa melalui input keyboard, menyimpan data tersebut ke dalam class Mahasiwa_2511533018, kemudian melakukan pengecekan terhadap pola tertentu pada NIM menggunakan method bawaan String. Selain itu, program menampilkan hasil berupa angkatan dan jurusan berdasarkan kondisi yang terpenuhi.

Langkah Penyelesaian:

1. Import class Scanner untuk membaca input dari pengguna.
2. Membuat object Scanner dan meminta pengguna memasukkan nama dan NIM menggunakan nextLine().
3. Membuat object mahasiswa dengan sintaks:

Mahasiwa_2511533018 ms = new Mahasiwa_2511533018();

4. Menyimpan data ke dalam class Mahasiswa menggunakan:
 - a. ms.setNim2(x); untuk menyimpan NIM (tipe String)
 - b. ms.setNama(y); untuk menyimpan nama
5. Melakukan pengecekan String pada NIM:
 - a. startsWith("25") → Menentukan apakah mahasiswa angkatan 2025.
 - b. contains("1153") → Menentukan apakah mahasiswa jurusan Informatika.
6. Menampilkan hasil pengecekan menggunakan System.out.println().
7. Memanggil method Cetak2() untuk menampilkan kembali NIM dan nama mahasiswa dari class.
8. Menutup Scanner dengan input.close().

Hasil:

Program akan menghasilkan output berupa:

1. Keterangan apakah mahasiswa merupakan angkatan 2025, jika NIM diawali "25".
2. Keterangan apakah mahasiswa adalah mahasiswa Informatika, jika NIM mengandung "1153".
3. Data mahasiswa akan ditampilkan ulang oleh method Cetak2() dalam format:

Output:

```
Nama:
althaf
NIM:
2511
Nim althaf
Nama 2511
```

2.5 Program String 1

Kode program:

```
package pekan7_2511533018;

public class String1 {

    Run | Debug
    public static void main(String[] args) {

        String salam = "Assalamualaikum";

        System.out.println("Panjang salam adalah: " + salam.length());
        System.out.println(salam.toUpperCase());
        System.out.println(salam.toLowerCase());
        System.out.println(salam.indexOf(str: "salam"));
        System.out.println(salam.lastIndexOf(str: "l"));

    }

}
```

Program ini dibuat untuk mendemonstrasikan penggunaan beberapa method penting pada class String di Java. Program ini memanfaatkan sebuah variabel String bernama salam lalu menampilkan panjang teks, mengubah huruf menjadi

kapital dan kecil, serta mencari posisi kata atau karakter tertentu di dalam String.

Langkah Penyelesaian:

1. Mendeklarasikan variabel String dengan nilai "Assalamualaikum".
2. Menampilkan panjang teks menggunakan `salam.length()`.
3. Mengubah seluruh huruf menjadi kapital menggunakan `salam.toUpperCase()`.
4. Mengubah seluruh huruf menjadi huruf kecil menggunakan `salam.toLowerCase()`.
5. Mencari posisi awal dari substring "salam" menggunakan `salam.indexOf("salam")`.
6. Mencari posisi terakhir huruf 'l' menggunakan `salam.lastIndexOf("l")`.
7. Semua hasil ditampilkan ke layar menggunakan `System.out.println()`.

Output yang dihasilkan menunjukkan bagaimana method String bekerja untuk mengolah dan menganalisis teks.

Output:

```
praktikum_D_2511333018/praktikum_2023
Panjang salam adalah: 15
ASSALAMUALAIKUM
assalamualaikum
2
9
```

2.6 Program String 2

Kode program:

```
package pekan7_2511533018;

import java.util.Scanner;

public class String2 {

    Run | Debug
    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.print(s: "Nama depan: ");
        String firstName = input.nextLine();

        System.out.print(s: "Nama belakang: ");
        String lastName = input.nextLine();

        String txt1 = "Dosen\\"Intelktual\\"Kampus";
        System.out.println("Nama lengkap: " + firstName + " " + lastName);
        System.out.println("Nama lengkap: " + firstName.concat(lastName));
        System.out.println(txt1);

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

        String a = "10";
        String b = "20";
        String c = a + b;
        System.out.println("String a + String b = " + c);

        String v = a + y;
        System.out.println("String a + Integer y = " + v);

        input.close();
    }
}
```

Program ini dibuat untuk menunjukkan penggunaan berbagai operasi pada tipe data String, termasuk penggabungan teks, penggunaan concat(), menampilkan karakter escape, serta perbedaan antara penjumlahan angka dan penggabungan String. Program juga menggunakan input dari pengguna untuk membaca nama depan dan nama belakang.

Langkah Penyelesaian:

1. Mengimpor dan membuat object Scanner untuk membaca input nama dari pengguna.
2. Meminta input nama depan dan nama belakang menggunakan `nextLine()`.
3. Menggabungkan nama dengan dua cara:
 - a. Menggunakan operator `+` \rightarrow `firstName + " " + lastName`
 - b. Menggunakan method `concat()` \rightarrow `firstName.concat(lastName)`
4. Menampilkan teks dengan tanda kutip menggunakan escape string seperti `"Dosen\"Intelktual\"Kampus"`.
5. Mendemonstrasikan penjumlahan angka dengan variabel integer `x` dan `y`, lalu menghitung `z = x + y`.
6. Membandingkan dengan penggabungan String, misalnya:
 - a. `a + b` di mana kedua variabel bertipe String \rightarrow menghasilkan gabungan teks.
 - b. `a + y` (String + integer) \rightarrow menghasilkan penggabungan karena String bersifat dominan.
7. Menampilkan seluruh hasil dengan `System.out.println()`.
8. Menutup Scanner untuk mengakhiri input.

Program ini memperlihatkan perbedaan jelas antara penjumlahan numerik dan penggabungan String, serta cara-cara manipulasi String di Java.

Output:

```
Nama depan: Althaf
Nama belakang: mulya
Nama lengkap: Althaf mulya
Nama lengkap: Althafmulya
Dosen"Intelktual"Kampus
x + y = 30
String a + String b = 1020
String a + Integer y = 1020
~1th501th5f 2025-09-10 10:00:00 D: 2511522010
```

BAB III

KESIMPULAN

3.1 Ringkasan Hasil Praktikum

Praktikum ini menghasilkan pemahaman dasar mengenai penggunaan *class* dan *string* dalam Java. Melalui beberapa percobaan, saya berhasil membuat class sederhana, mengakses atribut dan metode, serta memanipulasi string menggunakan fungsi bawaan Java seperti `length()`, `substring()`, `replace()`, dan `equalsIgnoreCase()`. Praktikum juga menunjukkan bagaimana objek dibuat dan digunakan dalam program Java. Secara keseluruhan, tujuan praktikum telah tercapai.

DAFTAR PUSTAKA

1. Deitel, P. J., & Deitel, H. M. (2018). *Java: How to Program* (11th ed.). Pearson.
2. Oracle. (2025). *Java Platform, Standard Edition Documentation*.
3. Schildt, H. (2019). *Java: The Complete Reference* (11th ed.). McGraw-Hill.