

**LAPORAN PRATIKUM
PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN**

LAPORAN PRAKTIKUM PEKAN 5

**Disusun Oleh:
Muhammad Aufa Rafiki**

2511531012

Dosen Pengampu: Dr. Wahyudi S.T.M.T

Asisten Praktikum: Muhammad Zaki Al Hafiz



**DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS**

2025

KATA PENGANTAR

Pedoman ini disusun sebagai rujukan resmi bagi mahasiswa Departemen Informatika dalam penyusunan laporan praktikum pada mata kuliah *Pemrograman Dasar dengan Java*. Dokumen ini tidak hanya memberikan gambaran umum mengenai format penulisan, tetapi juga menguraikan secara rinci sistematika laporan, tata cara penyajian isi, serta contoh penulisan kode program yang dilengkapi dengan referensi ilmiah. Melalui panduan ini, mahasiswa diharapkan mampu menyusun laporan yang tidak sekadar memenuhi aspek administratif, tetapi juga mencerminkan ketelitian, keteraturan, dan penerapan kaidah penulisan akademik pada tingkat dasar. Dengan demikian, laporan praktikum yang dihasilkan dapat berfungsi sebagai media pembelajaran, dokumentasi kegiatan, sekaligus sarana untuk melatih keterampilan menulis ilmiah yang akan bermanfaat dalam jenjang studi selanjutnya.

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
BAB I PENDAHULUAN.....	1
1.1 Pengertian Pratikum.....	1
1.2 Tujuan Pratikum.....	1
1.3 Persyaratan Pratikum.....	1
1.4 Waktu dan Tempat Pratikum.....	2
BAB II PEMBAHASAN.....	3
2.1 For Loops	3
2.2 Perulangan For 1	3
2.3 Perulangan For 2	4
2.4 Perulangan For 3	4
2.5 Nested For 2	8
BAB III KESIMPULAN	10
DAFTAR PUSTAKA	11

BAB I

PENDAHULUAN

1.1 Pengertian Pratikum

Praktikum Java adalah kegiatan pembelajaran yang dilakukan di laboratorium komputer untuk mengasah keterampilan mahasiswa dalam memahami serta menerapkan konsep pemrograman Java. Kegiatan ini tidak hanya menekankan pada penguasaan teori, tetapi juga pada latihan penyusunan kode program, pengujian, hingga analisis hasil eksekusi. Praktikum dipandang sebagai wahana latihan yang menjembatani pemahaman konseptual dengan kemampuan teknis pemrograman.

1.2 Tujuan Pratikum

Tujuan dari pelaksanaan praktikum antara lain sebagai berikut:

- 1.** Membantu mahasiswa memahami konsep dasar pemrograman Java melalui penerapan langsung.
- 2.** Melatih kemampuan menulis, mengompilasi, dan mengeksekusi program dengan mengikuti aturan sintaksis Java.
- 3.** Meningkatkan keterampilan dalam memecahkan masalah (*problem solving*) dengan pendekatan algoritmik.
- 4.** Membiasakan mahasiswa bekerja sistematis dalam menyusun laporan yang memuat analisis hasil praktikum.
- 5.** Menanamkan sikap teliti, disiplin, serta tanggung jawab dalam melaksanakan kegiatan laboratorium.

1.3 Persyaratan Pratikum

Agar praktikum berjalan lancar, mahasiswa perlu memenuhi beberapa persyaratan berikut:

- 1.** Telah mengikuti perkuliahan teori Pemrograman Java sebagai dasar pemahaman.
- 2.** Membawa perlengkapan yang diperlukan, antara lain laptop atau komputer yang sudah terpasang Java Development Kit (JDK) dan Integrated Development Environment (IDE) yang direkomendasikan.

3. Mengikuti setiap sesi praktikum sesuai jadwal yang ditetapkan dan hadir minimal sesuai ketentuan program studi.
4. Mematuhi tata tertib laboratorium, termasuk menjaga keamanan data, perangkat, serta lingkungan kerja.
5. Menyusun laporan praktikum dengan format dan aturan yang telah ditetapkan dalam pedoman ini.

1.4 Waktu dan Tempat Pratikum

Pelaksanaan praktikum Java mengikuti kalender akademik yang berlaku pada program studi. Setiap sesi praktikum dilaksanakan sesuai jadwal yang ditentukan oleh dosen pengampu. Tempat kegiatan umumnya berlangsung di laboratorium komputer, namun pada kondisi tertentu dapat dilaksanakan secara mandiri dengan perangkat masing-masing, selama memenuhi syarat teknis yang ditetapkan.

BAB II

PEMBAHASAN

2.1 FOR LOOPS

1. Kode Program

```
1 package pekan7_2511531012;
2 import java.util.Scanner;
3 public class BilanganPrima_2511531012 {
4
5     public static boolean isPrime(int n) {
6         int factors = 0;
7         for (int i=1; i <= n; i++) {
8             if (n%i == 0) {
9                 factors++;
10            }
11        }
12        return (factors == 2);
13    }
14
15    public static void main (String[] args) {
16        Scanner input = new Scanner (System.in);
17        System.out.print("Input nilai n = ");
18        int a = input.nextInt();
19        if (isPrime(a)) {
20            System.out.println(a+" bilangan prima");
21        } else {
22            System.out.println(a+" bukan bilangan prima");
23        }
24    }
25
26 }
```

Gambar 2.1

2. Output

```
Input nilai n = 17
17 bilangan prima
```

Gambar 2.2

```
Input nilai n = 16
16 bukan bilangan prima
```

Gambar 2.3

3. Kerja Program

Program Java ini dirancang untuk menentukan apakah sebuah bilangan bulat yang diinput oleh pengguna merupakan bilangan prima atau bukan. Alur kerja program dimulai dari metode main, yang merupakan titik masuk utama eksekusi. Pertama, program membuat sebuah objek Scanner untuk membaca input dari keyboard pengguna. Program kemudian menampilkan pesan "Input nilai n = " di konsol untuk meminta pengguna memasukkan

sebuah angka. Angka yang dimasukkan oleh pengguna tersebut selanjutnya dibaca dan disimpan ke dalam variabel integer bernama a. Setelah nilai a didapatkan, program akan memanggil metode `isPrime(a)` untuk mengevaluasi bilangan tersebut. Metode `isPrime` ini menerima satu parameter integer, yang dalam kasus ini adalah nilai a (disebut n di dalam metode). Di dalam metode `isPrime`, sebuah variabel bernama `factors` diinisialisasi dengan nilai 0. Program kemudian menjalankan perulangan (loop) `for` yang akan mengiterasi semua bilangan i mulai dari 1 hingga n (inklusif). Pada setiap iterasi, program akan memeriksa apakah n habis dibagi oleh i (menggunakan operasi modulo `%`). Jika `n % i == 0` (sisa baginya 0), itu berarti i adalah faktor dari n, dan nilai variabel `factors` akan ditambah satu (increment). Setelah perulangan selesai (setelah memeriksa semua bilangan dari 1 sampai n), metode `isPrime` akan mengembalikan hasil dari evaluasi `factors == 2`. Sebuah bilangan prima, menurut definisi, adalah bilangan yang hanya memiliki tepat dua faktor positif, yaitu 1 dan dirinya sendiri. Jika `factors` bernilai 2, metode akan mengembalikan `true`. Jika `factors` tidak bernilai 2 (misalnya 1, 3, 4, dst.), metode akan mengembalikan `false`. Nilai boolean (`true` atau `false`) ini kemudian dikirim kembali ke metode `main`. Di metode `main`, pernyataan `if` akan menangkap hasil ini. Jika hasilnya `true`, program akan mencetak [nilai a] bilangan prima. Jika hasilnya `false`, program akan mencetak [nilai a] bukan bilangan prima.

2.2 Mahasiswa

1. Kode Program

```

1 package pekan7_2511531012;
2
3 public class Mahasiswa_2511531012 {
4     //variabel global
5     private int nim;
6     private String nama,nim2;
7     // membuat mutator (setter)
8     public void setNim (int nim) {
9         this.nim=nim;
10    }
11    public void setNim2 (String nim2) {
12        this.nim2=nim2;
13    }
14    public void setNama (String nama) {
15        this.nama=nama;
16    }
17    //membuat accessor (getter)
18    public int getNim() {
19        return nim;
20    }
21    public String getNim2() {
22        return nim2;
23    }
24    public String getNama() {
25        return nama;
26    }
27    //metode lain
28    public void Cetak() {
29        System.out.println("Nim :"+nim);
30        System.out.println("Nama :"+nama);
31    }
32    public void Cetak2() {
33        System.out.println("Nim :"+nim2);
34        System.out.println("Nama :"+nama);
35    }

```

Gambar 2.4

2. Kerja Program

Program ini adalah sebuah Class (Kelas) yang berfungsi sebagai blueprint atau cetakan untuk mendefinisikan sebuah objek "Mahasiswa". File ini tidak bisa dijalankan dengan sendirinya karena ia tidak memiliki metode main. Tugasnya murni untuk mendeskripsikan struktur data dan perilaku dari seorang Mahasiswa. Di dalamnya, terdapat variabel private (disebut juga atribut) yaitu nim, nama, dan nim2, yang digunakan untuk menyimpan data. Karena bersifat private, variabel ini tidak bisa diakses langsung dari luar kelas, sehingga program ini menyediakan "jembatan" berupa metode public. Metode yang digunakan untuk mengisi/mengatur nilai ke dalam variabel private disebut Setter (atau mutator), dalam program ini contohnya adalah setNim(int nim), setNim2(String nim2), dan setNama(String nama). Sementara itu, metode yang digunakan untuk mengambil/membaca nilai dari variabel private disebut Getter (atau accessor), contohnya adalah getNim(), getNim2(), dan getNama(). Terakhir, program ini juga memiliki metode lain (Cetak() dan Cetak2()) yang mendefinisikan perilaku atau aksi yang bisa dilakukan oleh objek Mahasiswa, yaitu mencetak data nim dan nama miliknya sendiri ke konsol.

2.3 Panggil Mahasiswa

1. Kode Program

```
1 package pekan7_2511531012;
2
3 public class PanggilMahasiswa_2511531012 {
4
5     public static void main(String[] args) {
6         Mahasiswa_2511531012 a= new Mahasiswa_2511531012();
7         a.setNim(1531012);
8         a.setNama("Muhammad Aufa Rafiki");
9         System.out.println(a.getNim());
10        System.out.println(a.getNama());
11        a.Cetak();
12    }
13 }
14
15 }
16
```

Gambar 2.5

2. Output

```
1531012
Muhammad Aufa Rafiki
Nim :1531012
Nama :Muhammad Aufa Rafiki
```

Gambar 2.6

3. Kerja Program

Program ini adalah kelas eksekutor yang bertugas untuk menjalankan alur aplikasi. Inilah file yang Anda "Run" karena di dalamnya terdapat metode public static void main(String[] args), yang merupakan titik masuk atau awal mula eksekusi program Java. Proses di dalam main dimulai dengan membuat

objek (sebuah wujud nyata) dari blueprint yang telah kita definisikan sebelumnya; perintah `new Mahasiswa_2511531012()` adalah proses "mencetak" objek baru dari kelas Mahasiswa, yang kemudian disimpan dalam variabel bernama `a`. Setelah objek `a` tercipta, program kemudian menggunakan metode Setter (`a.setNim(...)` dan `a.setNama(...)`) untuk mengisi data ke dalam objek `a` tersebut. Selanjutnya, program ini menampilkan data yang ada di dalam `a` ke konsol dengan dua cara: pertama dengan memanggil metode Getter (`a.getNim()` dan `a.getNama()`) lalu mencetaknya, dan kedua dengan memanggil metode `a.Cetak()` yang menyuruh objek `a` untuk mencetak datanya sendiri.

2.4 Panggil Mahasiswa 2

1. Kode Program

```
1 package pekan7_2511531012;
2
3 import java.util.Scanner;
4
5 public class PanggilMahasiswa2 {
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_2511531012 a = new Mahasiswa_2511531012();
14        a.setNim2(x);
15        a.setNama(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 }
```

Gambar 2.7

2. Output

```
Nim: 2511531012
Nama: Muhammad Aufa Rafiki
Muhammad Aufa Rafiki anda angkatan 2025
Anda Mahasiswa Informatika
Nim :2511531012
Nama :Muhammad Aufa Rafiki
```

Gambar 2.8

3. Kerja Program

Program PanggilMahasiswa2 ini berfungsi sebagai kelas eksekutor utama yang melibatkan interaksi dengan pengguna (user input) dan implementasi logika kondisional (if). Proses kerja program dimulai pada metode main dengan menginisialisasi objek Scanner untuk membaca input dari keyboard. Program kemudian akan meminta pengguna untuk memasukkan dua nilai berturut-turut: pertama, nilai NIM, yang dibaca sebagai String dan disimpan dalam variabel x, dan kedua, nilai Nama, yang disimpan dalam variabel y. Setelah menerima input, program menciptakan sebuah objek baru dari blueprint Mahasiswa_2511531012 dan memberinya nama referensi a. Nilai NIM (x) kemudian dimasukkan ke dalam objek a melalui metode setter a.setNim2(x), dan nilai Nama (y) dimasukkan melalui a.setNama(y). Selanjutnya, program memasuki tahap pengecekan kondisional yang melibatkan analisis terhadap string NIM (x). Pertama, program mengecek dengan if (x.startsWith("25")) apakah NIM tersebut diawali dengan karakter "25". Jika kondisi ini benar, program akan mencetak pesan yang mengindikasikan bahwa mahasiswa tersebut adalah angkatan 2025. Kedua, program melakukan pengecekan terpisah dengan if (x.contains("1153")) untuk melihat apakah NIM tersebut mengandung deret karakter "1153" di posisi manapun. Jika kondisi ini terpenuhi, program mencetak pesan yang mengidentifikasi pengguna sebagai Mahasiswa Informatika. Sebagai langkah akhir, program memanggil metode a.Cetak2() dari objek a untuk menampilkan kembali data NIM dan Nama yang tersimpan di objek tersebut ke konsol, sebelum menutup sumber daya Scanner (input.close()).

2.5 String1

1. Kode Program

```
1 package pekan7_2511531012;
2
3 public class String1_2511531012 {
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()); //output "Assalamualaikum"
9         System.out.println(salam.toLowerCase()); //output "assalamualaikum"
10        System.out.println(salam.indexOf("salam")); // outputs 2
11
12    }
13
14 }
15 }
```

Gambar 2.9

2. Output

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

Gambar 2.10

3. Kerja Program

Program Java ini dimulai dari metode main yang merupakan titik awal eksekusi. Pertama, program mendeklarasikan sebuah variabel string bernama `salam` dan memberinya nilai `"Assalamualaikum"`. Setelah itu, program menjalankan empat perintah cetak (`print`) secara berurutan. Perintah pertama, `salam.length()`, menghitung jumlah total karakter dalam string `salam` (termasuk huruf besar dan kecil) dan mencetak hasilnya, yaitu 15. Perintah kedua, `salam.toUpperCase()`, mengubah semua karakter dalam string `salam` menjadi huruf kapital dan mencetak hasilnya, yaitu `"ASSALAMUALAIKUM"`. Perintah ketiga, `salam.toLowerCase()`, mengubah semua karakter menjadi huruf kecil dan mencetak `"assalamualaikum"`. Perintah terakhir, `salam.indexOf("salam")`, mencari posisi (indeks) pertama kali substring `"salam"` ditemukan di dalam string `"Assalamualaikum"`. Karena pencarian ini case-sensitive (membedakan huruf besar dan kecil) dan substring `"salam"` (dengan 's' kecil) ditemukan mulai dari karakter ketiga (indeks dimulai dari 0), program mencetak angka 2.

2.6 String2

1. Kode Program

BAB III KESIMPULAN

Dari pembahasan yang telah dilakukan, dapat disimpulkan bahwa `for loops` merupakan salah satu struktur perulangan yang sangat penting dalam pemrograman karena memungkinkan eksekusi kode secara berulang dengan jumlah iterasi yang telah ditentukan. Dengan menggunakan `for loops`, proses yang bersifat repetitif dapat dilakukan dengan lebih efisien, terstruktur, dan mudah dipahami.

Ada bentuk pengembangan dari `for loop` yang memungkinkan penggunaan satu perulangan di dalam perulangan lainnya yang disebut `nested for loop` atau perulangan bersarang.

Secara keseluruhan, pemahaman yang baik tentang `for loop` dan `nested for loop` sangat penting bagi seorang programmer, karena kedua konsep ini menjadi dasar dalam pembuatan algoritma yang efektif dan efisien di berbagai bahasa pemrograman.

DAFTAR PUSTAKA

- [1] Oracle, *The for Statement (Java™ Tutorials)*, 2024. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/for.html>
- [2] J. Gosling, B. Joy, G. Steele, and G. Bracha, *The Java Language Specification*, 5th ed., Boston: Addison-Wesley, 2022.