

**LAPORAN PRATIKUM
PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN**

LAPORAN PRAKTIKUM PEKAN 6

**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 PUSTAKA	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 Perulangan While 1.....	3
2.2 Lempar Dadu	4
2.3 Game Penjumlahan.....	6
2.4 Sentinel Loop	8
2.5 Do While 1	9
BAB III KESIMPULAN	11
DAFTAR PUSTAKA	12

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 Perulangan While 1

1. Kode Program

```
1 package pekan6_2511531012;
2
3 import java.util.Scanner;
4
5 public class perulanganWhile1_2511531012 {
6
7     public static void main(String[] args) {
8
9         int counter=0;
10        String jawab;
11        boolean running = true;
12        //deklarasi scanner
13        Scanner scan = new Scanner (System.in);
14        while (running) {
15            counter++;
16            System.out.println("Jumlah = "+ counter);
17            System.out.print("Apakah lanjut (ya / tidak)");
18            jawab= scan.nextLine();
19            //cek jawab = tidak, perulangan berhenti
20            if (jawab.equalsIgnoreCase("tidak")) {
21                running= false;
22            }
23        }
24        System.out.println("Anda sudah melakukan perulangan sebanyak "+ counter+" kali");
25    }
26 }
27
28 }
29 }
```

Gambar 2.1

2. Output

```
Jumlah = 1
Apakah lanjut (ya / tidak)YA
Jumlah = 2
Apakah lanjut (ya / tidak)YA
Jumlah = 3
Apakah lanjut (ya / tidak)ya
Jumlah = 4
Apakah lanjut (ya / tidak)tidak
Anda sudah melakukan perulangan sebanyak 4 kali
```

Gambar 2.2

3. Kerja program

Program ini pada dasarnya adalah sebuah perulangan yang dikontrol oleh input pengguna. Saat program dimulai, ia menyiapkan sebuah penghitung (counter) bernilai 0 dan sebuah "saklar" (running) yang disetel ke true. Program kemudian masuk ke dalam perulangan while karena "saklar"

tersebut true. Di setiap putaran, program akan menambah penghitung, mencetak nilai penghitung saat ini, lalu bertanya kepada pengguna apakah ingin lanjut dengan mengetik "ya" atau "tidak". Program akan berhenti sejenak untuk membaca jawaban pengguna. Jika pengguna mengetik "tidak" (tidak peduli huruf besar atau kecil), "saklar" running akan diubah menjadi false. Jika pengguna mengetik jawaban lain (seperti "ya"), "saklar" running tetap true dan perulangan akan berlanjut ke putaran berikutnya. Begitu "saklar" running berubah menjadi false, perulangan akan berhenti total. Akhirnya, program akan mencetak satu kalimat penutup yang menampilkan jumlah total perulangan yang telah dilakukan berdasarkan nilai akhir counter.

2.2 Lempar dadu

1. Kode program

```
1 package pekan6_2511531012;
2 import java.util.Random;
3 public class Lempardadu_2511531012 {
4
5     public static void main(String[] args) {
6         Random rand = new Random();
7         int tries = 0;
8         int sum = 0;
9         while (sum !=7) {
10             // roll the dice once
11             int dadu1 = rand.nextInt(6) + 1;
12             int dadu2 = rand.nextInt(6) + 1;
13             sum = dadu1 + dadu2;
14             System.out.println(dadu1 + " + " + dadu2 + " = " + sum);
15             tries++;
16         }
17         System.out.println("You won after " + tries + "tries!");
18     }
19 }
20 }
21 }
```

Gambar 2.3

2. Output

```
6 + 4 = 10
1 + 6 = 7
You won after 2tries!
```

Gambar 2.4

```
2 + 4 = 6
3 + 2 = 5
5 + 4 = 9
6 + 4 = 10
2 + 1 = 3
3 + 6 = 9
1 + 6 = 7
You won after 7tries!
```

Gambar 2.5

3. Kerja program

Program ini adalah simulasi pelemparan dua buah dadu yang akan terus berulang sampai jumlah kedua dadu tersebut tepat bernilai 7. Pada awalnya, program menyiapkan sebuah generator angka acak serta dua variabel, `tries` (untuk menghitung jumlah percobaan) dan `sum` (untuk menyimpan jumlah dadu), yang keduanya diatur ke 0. Program kemudian memasuki sebuah perulangan `while` dengan kondisi "terus jalankan selama nilai `sum` tidak sama dengan 7". Di dalam setiap putaran, program akan mengacak dua angka (masing-masing antara 1 sampai 6) untuk mewakili `dadu1` dan `dadu2`, menjumlahkan kedua nilai dadu tersebut, dan memperbarui nilai `sum`. Hasil lemparan (contoh: "3 + 5 = 8") akan dicetak ke layar, dan penghitung `tries` akan ditambah satu. Proses ini akan terus diulangi mengocok dadu, menjumlahkan, mencetak hasil, dan menambah `tries` sampai suatu saat hasil `sum` dari kedua dadu itu tepat bernilai 7, yang akan menyebabkan kondisi perulangan `while` menjadi salah dan perulangan pun berhenti. Akhirnya, program akan mencetak pesan "You won after..." yang diikuti dengan jumlah total percobaan (`tries`) yang diperlukan untuk mendapatkan angka 7 tersebut.

2.3 Game Penjumlahan

1. Kode program

```
1 package pekan6_2511531012;
2+ import java.util.Scanner;
4
5 public class GamePenjumlahan_2511531012 {
6
7-     public static void main(String[] args) {
8         Scanner console = new Scanner (System.in);
9         Random rand = new Random();
10        // play until user gets 3 wrong
11        int points = 0;
12        int wrong = 0;
13-        while (wrong < 3) {
14            int result = play(console, rand); //play one game
15-            if (result > 0) {
16                points++;
17-            } else {
18                wrong++;
19            }
20        }
21        System.out.println("You earned " + points + " total points.");
22
23    }
24    //membuat soal penjumlahan dan ditampilkan ke user
25-    public static int play(Scanner console, Random rand) {
26        //print the operands being added, and sum them
27        int operands = rand.nextInt(4) + 2;
28        int sum = rand.nextInt(10) + 1;
29        System.out.print(sum);
30-        for (int i = 2; i <= operands; i++) {
31            int n = rand.nextInt(10) + 1;
32            sum += n;
33            System.out.print(" + " + n);
34        }
35        System.out.print(" = ");
36        // read user's guess and report whether it was correct
```

Gambar 2.6

```
37        int guess = console.nextInt();
38-        if (guess == sum) {
39            return 1;
40-        } else {
41            System.out.println("Wrong! The answer was " + sum);
42            return 0;
43        }
44    }
45 }
46
```

Gambar 2.7

2. Output

```
4 + 1 + 6 = 11
6 + 9 + 6 + 5 + 8 = 34
8 + 6 = 14
10 + 10 + 6 + 3 = 24
Wrong! The answer was 29
6 + 9 + 8 + 9 = 34
Wrong! The answer was 32
6 + 10 + 2 + 10 = 22
Wrong! The answer was 28
You earned 3 total points.
```

Gambar 2.8

3. Kerja program

Program ini adalah sebuah permainan kuis penjumlahan sederhana yang akan terus memberikan soal sampai pemain melakukan 3 kali kesalahan. Saat program dijalankan, ia akan menyiapkan penghitung points (untuk skor) dan wrong (untuk kesalahan), keduanya dimulai dari 0. Program kemudian masuk ke dalam sebuah perulangan while yang hanya akan terus berjalan selama jumlah kesalahan (wrong) masih kurang dari 3. Di dalam setiap putaran, program akan memanggil metode play untuk menjalankan satu ronde permainan, yang kemudian akan mengembalikan nilai (1 jika benar, 0 jika salah). Di dalam metode play, program membuat sebuah soal penjumlahan: pertama, ia mengacak jumlah angka yang akan dijumlahkan (antara 2 sampai 5 angka), lalu ia mengacak setiap angka (antara 1 sampai 10) dan menampilkannya di layar (contoh: "8 + 3 + 5 = "). Program kemudian menunggu pengguna mengetikkan tebakan jawaban mereka. Setelah pengguna memasukkan jawaban, program akan membandingkannya dengan sum (hasil penjumlahan yang sebenarnya). Jika tebakan pengguna benar, metode play mengembalikan nilai 1, dan penghitung points di main akan bertambah. Jika tebakan pengguna salah, program akan mencetak "Wrong!" beserta jawaban yang benar, mengembalikan nilai 0, dan penghitung wrong di main akan bertambah. Siklus ini akan terus berulang—memberi soal baru, memeriksa jawaban, menambah points atau wrong—sampai penghitung wrong mencapai angka 3. Pada saat itu, kondisi while (wrong < 3) menjadi salah, perulangan berhenti, dan program akan mencetak total poin akhir yang berhasil didapat oleh pengguna.

2.4 Sentinel Loop

1. Kode program

```
1 package pekan6_2511531012;
2
3 import java.util.Scanner;
4
5 public class SentinelLoop_2511531012 {
6
7     public static void main(String[] args) {
8         Scanner console = new Scanner (System.in);
9         int sum = 0;
10        int number=12;
11
12        while (number !=0) {
13            System.out.print("Maasukkan angka (0 untuk keluar):");
14            number = console.nextInt();
15            sum = sum + number;
16        }
17        System.out.println("totalnya adalah " + sum);
18    }
19 }
20
21 }
22 }
```

Gambar 2.9

2. Output

```
Maasukkan angka (0 untuk keluar):2
Maasukkan angka (0 untuk keluar):4
Maasukkan angka (0 untuk keluar):6
Maasukkan angka (0 untuk keluar):3
Maasukkan angka (0 untuk keluar):0
totalnya adalah 15
```

Gambar 2.10

3. Kerja program

Program ini berjalan sebagai kalkulator penjumlahan yang akan terus meminta input dari pengguna sampai sebuah nilai "sentinel" (yaitu angka 0) dimasukkan. Awalnya, program menginisialisasi total sum ke 0 dan sebuah variabel number ke 12 agar dapat masuk ke dalam perulangan while (karena 12 tidak sama dengan 0). Di dalam setiap putaran, program akan meminta pengguna untuk memasukkan sebuah angka dan memberi tahu bahwa angka 0 akan menghentikan program. Angka yang dimasukkan oleh pengguna kemudian akan dibaca, disimpan ke dalam variabel number, dan langsung ditambahkan ke total sum saat ini. Proses ini akan terus berulang, mengakumulasi nilai sum, selama pengguna memasukkan angka selain 0. Begitu pengguna mengetikkan 0, angka 0 tersebut akan ditambahkan ke sum, dan pada pengecekan kondisi while berikutnya (karena number kini bernilai 0), perulangan akan berhenti. Akhirnya, program akan mencetak total akhir

dari semua angka yang telah dimasukkan dengan pesan "totalnya adalah" diikuti oleh nilai sum.

2.5 Do While 1

1. Kode program

```
1 package pekan6_2511531012;
2
3 import java.util.Scanner;
4
5 public class doWhile1_2511531012 {
6
7     public static void main(String[] args) {
8         Scanner console = new Scanner (System.in);
9         String phrase;
10        do {
11            System.out.print ("Input Password: ");
12            phrase = console.next();
13        } while (!phrase.equals("abcd"));
14
15
16    }
17
18 }
```

Gambar 2.11

2. Output

```
Input Password: 1212
Input Password: dafa
Input Password: aufa
Input Password: a
Input Password: s
Input Password: ad
Input Password: da12
Input Password: asdfghjklzxcv
Input Password: abcd
|
```

gambar 2.12

3. Kerja program

Secara keseluruhan, program ini adalah sebuah sistem validasi password sederhana yang akan terus meminta input dari pengguna sampai password

yang benar dimasukkan. Pertama, program mengimpor dan menginisialisasi Scanner untuk membaca input dari keyboard, serta mendeklarasikan variabel String bernama phrase. Program kemudian langsung masuk ke dalam perulangan do-while, yang menjamin bahwa badan loop (do) dieksekusi setidaknya satu kali. Di dalam loop, program mencetak "Input Password: " ke layar dan menunggu pengguna mengetikkan sesuatu, yang kemudian disimpan dalam variabel phrase. Setelah itu, program memeriksa kondisi di bagian while, yaitu !phrase.equals("abcd"). Kondisi ini berarti "ulangi loop selama isi dari phrase tidak sama dengan 'abcd'". Akibatnya, jika pengguna memasukkan password yang salah, kondisi akan bernilai benar (true) dan program akan kembali meminta password. Perulangan ini akan terus berlanjut hingga pengguna mengetik "abcd", yang akan membuat kondisi bernilai salah (false), sehingga loop berhenti dan program selesai.

BAB III

KESIMPULAN

perulangan (loop) adalah sebuah instruksi fundamental dalam pemrograman yang memungkinkan eksekusi satu blok kode secara berulang kali, sehingga program menjadi lebih efisien. Kunci dari setiap perulangan adalah kondisi berhenti (termination condition), yang menentukan kapan repetisi harus berakhir. Seperti yang terlihat dari berbagai contoh, kondisi ini sangat fleksibel: bisa berupa input spesifik dari pengguna (seperti "tidak" atau "0" sebagai sentinel), pencapaian target tertentu (seperti jumlah dadu 7), pencapaian batas hitungan (seperti 3 kali kesalahan), atau validasi input yang benar (seperti password "abcd"). Agar perulangan dapat berhenti, nilai yang dievaluasi dalam kondisi tersebut (seperti variabel input atau counter) harus dimodifikasi di dalam badan loop. Perbedaan struktur, seperti antara while (yang mengecek kondisi di awal) dan do-while (yang mengecek di akhir), memberikan kontrol lebih lanjut, misalnya untuk menjamin eksekusi minimal satu kali.

DAFTAR PUSTAKA

[1] R. C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship*. Boston: Prentice Hall, 2009.

[2] B. Eckel, *Thinking in Java, 4th ed.* Upper Saddle River, NJ: Prentice Hall, 2006.