

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN
PERULANGAN WHILE DAN DO=WHILE
Disusun Oleh:
Nama: Rifki Maulana
NIM: 2511533007
Dosen Pengampu: Dr. Wahyudi, S.T.,M.T.
Asisten Praktikum: Jovantri Immanuel Gulo



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
TAHUN 2025

KATA PENGANTAR

Puji syukur kehadirat Tuhan Yang Maha Esa karena dengan rahmat-Nya laporan praktikum ini dapat diselesaikan. Laporan ini disusun sebagai bagian dari pelaksanaan mata kuliah Praktikum Algoritma dan Pemrograman untuk memenuhi tugas dan memperdalam pemahaman praktis terhadap materi yang dibahas.

Terima kasih disampaikan kepada dosen pengampu, asisten praktikum, serta rekan-rekan yang telah memberikan bimbingan, bantuan, dan kerja sama selama kegiatan praktikum berlangsung. Penulis menyadari laporan ini masih jauh dari sempurna, oleh karena itu kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang.

Semoga laporan ini bermanfaat bagi penulis dan pembaca.

Padang, 06 November 2025

Rifki Maulana

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan Praktikum.....	1
1.3 Manfaat Praktikum.....	1
BAB 2 PEMBAHASAN.....	3
2.1 Uraian Kode Program.....	3
2.2 Langkah Kerja.....	5
2.3 Analisis hasil.....	7
BAB 3 KESIMPULAN.....	10
DAFTAR PUSTAKA.....	11

BAB I PENDAHULUAN

1.1 Latar Belakang

Pemrograman merupakan keterampilan dasar yang penting dalam bidang informatika. Salah satu konsep yang harus dipahami oleh mahasiswa adalah perulangan. Perulangan digunakan untuk mengeksekusi blok kode secara berulang dalam kondisi tertentu. Dengan perulangan program menjadi lebih efisien untuk tugas yang harus dilakukan berkali kali tanpa harus menulis kode yang sama berulang-ulang.

Dalam bahasa pemrograman java, terdapat beberapa jenis perulangan seperti for, while, dan dowhile yang memiliki karakteristik masing-masing. Selain itu, konsep *loop control* seperti *sentinel loop* juga sering digunakan untuk mengatur program. Melalui pemahaman konsep perulangan, programmer dapat membuat program yang lebih efisien dan fleksibel.

1.2 Tujuan Praktikum

Praktikum ini memiliki tujuan sebagai berikut:

1. Mahasiswa dapat memahami konsep dasar *while* dan *dowhile* pada java
2. Mahasiswa dapat memahami konsep *sentinel loop*
3. Mahasiswa mampu mengimplementasikan kelas *random* untuk mendapatkan angka acak
4. Mahasiswa mampu mengimplementasikan konsep *while*, *dowhile*, dan *sentinel loop*

1.3 Manfaat Praktikum

Mahasiswa memperoleh pemahaman mengenai konsep perulangan dalam bahasa pemrograman Java, baik menggunakan struktur *while*, *dowhile*, maupun penerapan *sentinel loop*. Pemahaman ini membantu mahasiswa berpikir secara

logika untuk menyusun program yang efisien dan terstruktur ketika menghadapi permasalahan yang memerlukan konsep perulangan. Praktikum ini melatih mahasiswa untuk menangani *input* dari pengguna, memanfaatkan kelas *Random* untuk menghasilkan nilai acak, serta mengontrol jalannya program berdasarkan kondisi logika.

BAB 2 PEMBAHASAN

2.1 Uraian Kode Program

```
1 package Pekan6_2511533007;
2 import java.util.Scanner;
3 |
4 public class doWhile1_2511533007 {
5     public static void main(String[] args) {
6
7         Scanner console = new Scanner(System.in);
8         String phrase;
9         do {
10             System.out.print("Input Password: ");
11             phrase = console.next();
12         } while (!phrase.equals("abcd"));
13         console.close();
14     }
15 }
16
```

Kode Program 2.1 doWhile1

```
1 package Pekan6_2511533007;
2 import java.util.Scanner;
3 import java.util.Random;
4
5 public class GamePenjumlahan_2511533007 {
6     public static void main(String[] args) {
7
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    //membuat soal penjumlahan dan ditampilkan ke user
24    public static int play (Scanner console, Random rand) {
25        // print the operands being added, and sum them
26        int operands = rand.nextInt(4) + 2;
27        int sum = rand.nextInt(10) +1;
28        System.out.print(sum);
29
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     (" = ");
37     // read users guess and report whether it was correct
38     int guess = console.nextInt();
39     if (guess==sum) {
40         return 1;
41     } else {
42         System.out.println("Wrongg !! The answer was " + sum);
43         return 0;
44     }
45 }
46 }
```

Kode Program 2.2 GamePejumlahan

```

1 package Pekan6_2511533007;
2 import java.util.Random;
3
4 public class Lempardadu_2511533007 {
5
6     public static void main(String[] args) {
7
8         Random rand = new Random();
9         int tries = 0;
10        int sum = 0;
11        while (sum != 7) {
12            //roll the dice once
13            int dadu1 = rand.nextInt(6) + 1;
14            int dadu2 = rand.nextInt(6) + 1;
15            sum = dadu1 + dadu2 ;
16            System.out.println(dadu1 + " " + dadu2 + " = " + sum);
17            tries++;
18        }
19        System.out.println("You won after " + tries + " tries!");
20    }
21 }
```

Kode Program 2.3 Lempardadu

```

1 package Pekan6_2511533007;
2 import java.util.Scanner;
3 public class PerulanganWhile1_2511533007 {
4
5    public static void main(String[] args) {
6
7        int counter=0;
8        String jawab;
9        boolean running = true;
10       //deklarasi scanner
11       Scanner scan = new Scanner(System.in);
12       while (running) {
13           counter++;
14           System.out.println("Jumlah " + counter);
15           System.out.print("Apakah Lanjut (ya / tidak?) ");
16           jawab = scan.nextLine();
17           //cek jawab = tidak, perulangan berhenti
18           if (jawab.equalsIgnoreCase("tidak")) {
19               running=false;
20           }
21       }
22       System.out.println("Anda sudah melakukan perulangan sebanyak " +counter+ " kali");
23       scan.close();
24
25   }
26
27 }
```

Kode Program 2.4 PerulanganWhile1

```

1 package Pekan6_2511533007;
2 import java.util.Scanner;
3
4 public class SentinelLoop_2511533007 {
5
6    public static void main(String[] args) {
7
8        Scanner console = new Scanner(System.in);
9        int sum = 0;
10       int number = 7; // dummy value, anything but 0
11
12       while (number != 0) {
13           System.out.print("Masukkan angka (0 untuk keluar) : ");
14           number = console.nextInt();
15           sum = sum + number;
16       }
17       System.out.println("totalnya adalah " + sum);
18       console.close();
19
20   }
21
22 }
```

Kode Program 2.5 SentinelLoop

2.2 Langkah Kerja

Langkah-langkah menyusun program:

1. Program doWhile1

- a) Mulai
- b) Deklarasi scanner console
- c) Deklarasi var string phrase
- d) Tampilkan input password
- e) Buat phrase = console.next
- f) *While phrase* tidak sama dengan abcd (*loop* ulang jika belum sama)
- g) Selesai

2. Program GamePenjumlahan

- a) Mulai
- b) *Import scanner dan random*
- c) Deklarasi scanner console dan random rand
- d) Inisialisasi points = 0 dan wrong = 0
- e) While wrong<3 result = play(console, rand), jika result>0 maka points++,
else wrong++
- f) *Print* jumlah point
- g) Selanjutnya untuk langkah play, buat public static int play
- h) Kemudian tentukan jumlah operand, operands = rand.nextInt(4) + 2
- i) Buat sum = rand.nextInt(10) +1
- j) *Print* sum
- k) Untuk i = 2, i<= operands, i++, buat int n = rand.nextInt(10) +1, sum +=n,
print(" " + n)
- l) Baca tebakan pengguna dan laporkan apakah itu benar atau salah
- m) Buat guess =console.nextInt()
- n) Jika guess == sum, return 1, else *print* jawaban salah dan *return* ke 0
- o) Selesai

3. Program Lempardadu

- a) Mulai
- b) *Import random*
- c) Deklarasi random rand
- d) Inisialisasi tries = 0, sum = 0

- e) While sum != 7, buat dadu1=rand.nextInt(6) + 1, buat dadu2=rand.nextInt(6) + 1, hitung sum = dadu1 + dadu2, *print output*, tries++
- f) Setelah *loop* `println("you won after " + tries + " tries!")`
- g) Selesai

4. Program PerulanganWhile1

- a) Mulai
- b) Import scanner
- c) Deklarsi int counter = 0, string jawab, boolean running = true, scanner
- d) While running == true, counter++, *print* jumlah counter, tanya *user* apakah ingin lanjut, jawab = scan.nextLine(), jika jawab.equalsIgnoreCase("Tidak"), ,maka running = false
- e) *Print* sudah berapa kali melakukan perulangan
- f) Selesai

5. Program SentinelLoop

- a) Mulai
- b) Import scanner
- c) Deklarasi scanner
- d) Inisialisasi sum = 0, number = 7
- e) While number != 0, tampilkan masukkan angka (0 untuk keluar), number = console.nextInt(), sum = sum + number
- f) *Print* hasil sum
- g) selesai

2.3 Analisis Hasil

1. Analisis kode program 2.1 doWhile1

Program menjalankan *dowhile* yang mana kode dijalankan minimal satu kali sebelum kondisinya diperiksa. Program akan terus meminta *input* kata sandi hingga pengguna memasukkan *string* "abcd".

Jika pengguna memasukkan kata lain selain "abcd", perulangan akan terus dilakukan. Program berhenti ketika kondisi `phrase.equals("abcd")` terpenuhi. Konsep perulangan *dowhile* memastikan satu kali eksekusi sebelum melihat kondisinya.

2. Analisis kode program 2.2 GamePenjumlahan

Program ini menerapkan *while* dengan batas salah (`wrong < 3`) untuk membuat game menebak hasil penjumlahan acak. Pada setiap iterasi, `play()` membentuk sejumlah bilangan acak menggunakan *Random* dan menghitung totalnya. Pengguna harus menebak hasil penjumlahan tersebut. Jika benar, skor meningkat (`points++`); jika salah, jumlah kesalahan bertambah (`wrong++`). Program berhenti saat kesalahan mencapai tiga kali.

Dari hasil percobaan, tampak bahwa setiap sesi permainan menghasilkan soal yang berbeda karena kondisi acak dari *Random*. Penggunaan kelas `java.util.Random` memungkinkan pembangkitan bilangan acak semu.

3. Analisis kode program 2.3 Lempardadu

Program ini melakukan lemparan dua dadu menggunakan `rand.nextInt(6) + 1` untuk mendapatkan nilai antara 1 sampai 6. Perulangan dilakukan hingga jumlah kedua dadu $= 7$. Variabel *tries* berfungsi menghitung berapa kali percobaan diperlukan sampai mendapatkan $\text{sum} = 7$.

Hasil percobaan menunjukkan jumlah lemparan sebelum mendapatkan total 7 bervariasi, karena *outputnya* bersifat acak. Hal ini dikarenakan kita menggunakan `java.util.random`.

4. Analisis kode program 2.4 PerulanganWhile1

Program ini melakukan perulangan dengan input pengguna. Setiap iterasi mencetak nilai counter dan langsung menanyakan apakah pengguna ingin melanjutkan (ya atau tidak).

Kondisi diberhentikan jika pengguna mengetik tidak, yang dicek menggunakan equalsIgnoreCase() agar menghiraukan format huruf besar/kecil.

Metode equalsIgnoreCase() digunakan untuk membandingkan string tanpa memperhatikan kapitalisasi, sedangkan *while* berfungsi untuk mengulangi perintah selama kondisinya bernilai benar.

5. Analisis kode program 2.5 SentinelLoop

Program ini menggunakan konsep *sentinel loop*, yaitu perulangan yang berhenti ketika ada sebuah nilai yang dimasukkan oleh pengguna. Dalam hal ini, nilai *sentinel* adalah 0. Program menjumlahkan setiap bilangan yang dimasukkan sampai pengguna mengetik 0. Nilai total akhir ditampilkan sebagai *output*.

Konsep *sentinel loop* ini umum digunakan untuk membaca data yang tidak diketahui jumlahnya sebelumnya, seperti *input* dari pengguna. *Sentinel loop* efisien untuk menghentikan input ketika kondisi tertentu terpenuhi.

BAB III KESIMPULAN

Laporan praktikum ini membahas implementasi dan analisis struktur perulangan *while*, *dowhile*, serta *sentinel loop* dalam bahasa pemrograman Java melalui beberapa contoh program. Berdasarkan hasil praktikum, dapat disimpulkan bahwa perulangan *while* cocok digunakan ketika jumlah iterasi belum diketahui secara pasti namun bergantung pada kondisi tertentu, sedangkan *dowhile* menjamin bahwa blok kode dieksekusi minimal satu kali sebelum evaluasi kondisi dilakukan. Adapun *sentinel loop* efektif untuk mengontrol berakhirnya proses perulangan berdasarkan input pengguna atau nilai penanda tertentu.

DAFTAR PUSTAKA

- [1] H. M. Deitel dan P. J. Deitel, *Java: How to Program*, 11th ed. Pearson, 2022
- [2] Oracle, “Class Random (Java Platform SE 17),” *Java SE Documentation*, Oracle, 2024. [Daring]. Tersedia pada:
<https://docs.oracle.com/javase/17/docs/api/java/util/Random.html>. [Diakses: 6-Nov-2025]
- [3] J. Stewart, *Calculus: Early Transcendentals*, 9th ed. Cengage Learning, 2021
- [4] Oracle, “Class String (Java Platform SE 17),” *Java SE Documentation*, Oracle, 2024. [Daring]. Tersedia pada:
<https://docs.oracle.com/javase/17/docs/api/java/lang/String.html>. [Diakses: 6-Nov-2025]