

**LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN**  
**WHILE DAN DO WHILE LOOP**



disusun Oleh:

Tiara Amalia Insani

NIM 2511532019

Dosen Pengampu : DR.Wahyudi, S.T, M.T

Asisten Praktikum : Jovantri Immanuel Gulo

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

**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.

Padang, 6 November 2025

Penyusun

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>i</b>
<b>DAFTAR ISI.....</b>	<b>ii</b>
<b>DAFTAR GAMBAR.....</b>	<b>iii</b>
<b>BAB I.....</b>	<b>1</b>
<b>PENDAHULUAN .....</b>	<b>1</b>
<b>1.1 Latar Belakang.....</b>	<b>1</b>
<b>1.2 Tujuan Praktikum.....</b>	<b>1</b>
<b>1.3 Manfaat.....</b>	<b>1</b>
<b>BAB II .....</b>	<b>2</b>
<b>PEMBAHASAN .....</b>	<b>2</b>
<b>2.1 While dan do while loop .....</b>	<b>2</b>
2.1.1 Perulangan while.....	2
2.1.2 While loop.....	2
2.1.3 Sentinel Value .....	2
<b>2.2 Praktikum.....</b>	<b>2</b>
2.3.1 Alat dan Bahan.....	2
2.3.2 Program.....	3
<b>BAB III.....</b>	<b>11</b>
<b>KESIMPULAN DAN SARAN .....</b>	<b>11</b>
<b>3.1 Kesimpulan .....</b>	<b>11</b>
<b>3.2 Saran .....</b>	<b>11</b>
<b>DAFTAR PUSTAKA .....</b>	<b>12</b>

## DAFTAR GAMBAR

Gambar 2.1 .....	3
Gambar 2.2 .....	4
Gambar 2.3 .....	6
Gambar 2.4 .....	8
Gambar 2.5 .....	9

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Dalam pemrograman, perulangan memungkinkan program menjalankan perintah secara berulang tanpa harus menulis kode yang sama berulang kali. Salah satu bentuk perulangan yang harus dipelajari adalah *while*, dan *do while*, karena struktur ini sering kali digunakan untuk menyelesaikan permasalahan yang membutuhkan proses berulang dengan jumlah iterasi yang tidak diketahui sebelumnya.

### 1.2 Tujuan Praktikum

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. Membiasakan mahasiswa bekerja sistematis dalam menyusun laporan yang memuat analisis hasil praktikum.
4. Menanamkan sikap teliti, disiplin, serta tanggung jawab dalam melaksanakan kegiatan laboratorium.
5. Belajar bagaimana membuat program yang bisa mengambil keputusan otomatis sesuai kondisi
6. Mempelajari konsep perulangan menggunakan *while*, dan *do while*.
7. Mengetahui cara kerja dan membedakan penggunaan *while*, dan *do while*.

### 1.3 Manfaat

Mahasiswa diharapkan semakin terbiasa menggunakan perulangan dalam pemrograman, dan bisa memahami kapan sebaiknya menggunakan masing masing struktur perulangan sesuai kebutuhan program.

## **BAB II**

### **PEMBAHASAN**

#### **2.1 While dan do while loop**

##### **2.1.1 Perulangan while**

Berfungsi untuk menjalankan blok kode selama kondisi tertentu masih terpenuhi (true), namun pada while loop kondisi diperiksa terlebih dahulu sebelum blok perintah dijalankan, sehingga jika kondisi awal bernilai salah (false), maka blok tidak akan dijalankan sama sekali.

##### **2.1.2 While loop**

Berfungsi untuk menjalankan blok kode selama kondisi tertentu masih terpenuhi (true), tetapi berbeda dengan while, pada do while loop, blok perintah akan dijalankan terlebih dahulu setidaknya satu kali, baru kemudian dilakukan pengecekan kondisi.

##### **2.1.3 Sentinel Value**

Adalah sebuah nilai khusus yang digunakan untuk menandai akhir dari proses input atau perulangan. Konsep ini sangat berguna dalam situasi dimana jumlah data yang akan diolah tidak diketahui sejak awal, sehingga program dapat berhenti secara otomatis ketika pengguna memasukkan nilai penanda tertentu.

#### **2.2 Praktikum**

##### **2.3.1 Alat dan Bahan**

1. Laptop/computer dengan eclipse terinstal.
2. Materi praktikum mengenai The For Loop.

## 2.3.2 Program

### 1. PerulanganWhile1

```
package pekan6_2511532019;

import java.util.Scanner;

public class perulanganWhile1_2511532019 {

    public static void main (String[] args) {
        int counter=0;
        String jawab;
        boolean running=true;
        //deklarasi Scanner
        Scanner scan = new Scanner(System.in);
        while (running) {
            counter ++;
            System.out.println("jumlah="+ counter);
            System.out.print("apakah lanjut (ya/tidak?");
            jawab=scan.nextLine();
            //cek jawab=tidak, perulangan berhenti
            if (jawab.equalsIgnoreCase("tidak")) {
                running=false;
            }
        }
        System.out.println("Anda sudah melakukan perulangan sebanyak "+counter+" kali");
    }
}
```

Gambar 2.1

Langkah-langkah :

- a. *Public class* PerulanganWhile1\_2511532019 { ➔ mendeklarasikan class.
- b. *Public static void main (String[] args){* tempat program java dieksekusi pertama kali. Agar program bisa dijalankan.
- c. *Counter* ➔ variabel yang digunakan untuk menghitung berapa kali perulangan terjadi
- d. *Jawab* ➔ variabel untuk menyimpan jawaban pengguna (“ya atau tidak”)
- e. *Running* ➔ variabel logika Boolean yang mengontrol apakah perulangan masih berjalan (true) atau berhenti (false).
- f. *Scanner scan = new Scanner (System.in)* ➔ program menerima input dari keyboard dan membaca input tersebut melalui system in
- g. *While (running)* ➔ Selama nilai running masih true, perulangan akan terus dilakukan.
- h. *Counter++* ➔ setiap kali perulangan dilakukan, nilai counter bertambah 1
- i. *System.out.println(“jumlah=”+counter);* ➔ menampilkan jumlah perulangan ke layer

- j. `System.out.print(apakah lanjut("ya/tidak?"));` ➔ menanyakan kepada pengguna apakah ingin melanjutkan perulangan atau tidak.
- k. `Jawab=scan.nextLine();` ➔ menyimpan jawaban ke dalam variabel jawab.
- l. `equalsIgnoreCase()` ➔ membandingkan string tanpa membedakan huruf besar atau kecil (huruf besar dan huruf kecil dianggap sama)
- m. jika pengguna menjawab tidak maka running menjadi false, dan perulangan berhenti.
- n. `System.out.println ("Anda sudah melakukan perulangan sebanyak "counter+"kali");`

## 2. LemparDadu

```
package pekan6_2511532019;

import java.util.Random;

public class LemparDadu_2511532019 {

    public static void main(String[] args) {
        Random rand = new Random();
        int tries=0;
        int sum=0;
        while (sum!=7) {
            //roll the dice once
            int dadu1=rand.nextInt(6)+1;
            int dadu2=rand.nextInt(6)+1;
            sum=dadu1+dadu2;
            System.out.println(dadu1+" "+dadu2+"="+sum);
            tries++;
        }
        System.out.println ("you won after " + tries + " tries!");
    }
}
```

Gambar 2.2

Langkah-langkah :

- a. `Public class LemparDadu_2511532019{` ➔ mendeklarasikan class.
- b. `Public static void main (String[] args){` tempat program java dieksekusi pertama kali, agar program bisa dijalankan.
- c. `Random rand=new Random();` ➔ membuat objek rand untuk menghasilkan angka secara acak antara 1-6



- d. *Int tries=0;* ➔ variabel penghitung berapa kali dadu dilempar
- e. *Int sum=0* ➔ variabel untuk menyimpan jumlah dari dua dadu setiap kali dilempar
- f. *While (sum != 7){* ➔ selama jumlah dadu belum sama dengan 7, maka dadu akan terus dilempar, saat sum menjadi 7, perulangan berhenti (pemain “menang”)
- g. *Int dadu1=rand.nextInt(6)+1;* ➔ menghasilkan angka acak antara 1 sampai 6 untuk dadu 1
- h. *Int dadu2=rand.nextInt(6)+1;* ➔ menghasilkan angka acak antara 1 sampai 6 untuk dadu 2
- i. *Sum = dadu1+dadu2;* ➔ jumlahkan dadu 1 dan 2 untuk memperoleh totalnya
- j. *System.out.println(dadu1+""+dadu2+"="+sum);* ➔ menampilkan hasil lemparan dadu
- k. *Tries++* ➔ menambahkan hitungan berapa kali percobaan lempar dadu dilakukan
- l. *System.out.println("you win after"+ tries+"tries!");* ➔ setelah total penjumlahan 2 dadu = 7 maka program menampilkan pesan kemenangan, dan berapa kali percobaan dibutuhkan sampai mendapatkan nilai total 7

### 3. GamePenjumlahan

```
package pekan6_2511532019;

import java.util.Scanner;

public class GamePenjumlahan_2511532019 {
    public static void main (String[] args) {
        Scanner console=new Scanner (System.in);
        Random rand= new Random();
        //play until user gets 3 wrong
        int points=0;
        int wrong=0;
        while (wrong<3) {
            int result=play(console, rand); //play one game
            if (result>0) {
                points++;
            }else {
                wrong++;
            }
        }
        System.out.println("you earned "+points+" total points");
    }

    //membuat soal penjumlahan dan ditampilkan ke user
    public static int play (Scanner console, Random rand) {
        //print the operands being added, and sum them
        int operands=rand.nextInt(4)+2;
        int sum=rand.nextInt(10)+1;
        System.out.print(sum);
        for (int i=2;i<= operands;i++) {
            int n=rand.nextInt(10)+1;
            sum += n;
            System.out.print(" + "+ n);
        }
        System.out.print(" = ");

        //read user's guess and report whether it was correct
        int guess=console.nextInt();
        if (guess==sum) {
            return 1;
        }else {
            System.out.println("Wrong! the answer was "+sum);
            return 0;
        }
    }
}
```

Gambar 2.3

Langkah-langkah:

- a. *Public class* GamePenjumlahan\_2511532019{ → mendeklarasikan class.

- b. *Public static void main (String[] args){* tempat program java dieksekusi pertama kali agar program bisa dijalankan.
- c. *Int jumlah=0;* ➔ mendeklarasikan variabel jumlah sebagai tempat menyimpan hasil penjumlahan.
- d. *Scanner console* ➔ untuk membaca jawaban pengguna dari keyboard
- e. *Random rand* ➔ untuk menghasilkan bilangan acak yang diperlukan untuk soal pada game penjumlahan
- f. *Points* ➔ menyimpan jumlah point jika menjawab benar.
- g. *Wrong* ➔ menyimpan jumlah kesalahan dalam menjawab soal
- h. *While(wrong<3)* ➔ perulangan akan terus dilakukan selama pemain belum melakukan kesalahan sebanyak 3 kali
- i. *Play(console,rand)* ➔ memanggil fungsi play yang menjalankan satu soal penjumlahan
- j. *If(result>0){points++}* ➔ jika result>0 berarti jawaban benar dan point bertambah
- k. *Else{wrong++}* ➔ jika tidak maka jumlah kesalahan yang bertambah.
- l. *Public static int play (Scanner console, Random rand)* ➔ menjalankan satu sesi permainan penjumlahan.
- m. *Rand.nextInt(4)+2;* ➔ menghasilkan angka acak antara 0-3, ditambah 2, jadi angka 2-5 angka yang akan dijumlahkan
- n. *Int sum=rand.nextInt(10)+1;* ➔ angka pertama dari soal acak antara 1-10
- o. *System.out.print(sum);* ➔ menampilkan angka pertama tanpa Ganti baris
- p. *For (int i=2;i<=operands;i++)* ➔ akan berulang sebanyak (operands-1) kali. Yaitu untuk menambah suku ke 2 hingga suku ke operands
- q. *Int n=rand.nextInt(10)+1* ➔ menghasilkan suku acak berikutnya yang nilainya antara 1-10
- r. *Sum += n;* ➔ menambahkan nilai n ke variabel sum
- s. *System.out.print(“+ ”+n);* ➔ menampilkan + diikuti n
- t. *System.out.print(“= ”);* ➔ menampilkan = untuk mengakhiri tampilan soal.

- u. *Int guess=console.nextInt();* → membaca input dari pengguna melalui objek console dan menyimpannya dalam variabel guess.
- v. *If (guess==sum){return 1}* → memeriksa apakah jawaban pengguna sama dengan sum, jika benar maka metode mengembalikan nilai 1
- w. *Else{System.out.println("Wrong! The answer was "+sum); return 0;}* → jika salah maka menampilkan pesan dan memberitahu nilai yang benar lalu metode mengembalikan nilai 0

#### 4. SentinelLoop

```
package pekan6_2511532019;

import java.util.Scanner;

public class SentinelLoop_2511532019 {

    public static void main(String[] args) {
        Scanner console=new Scanner(System.in);
        int sum=0;
        int number=12; //"dummy value", anything but 0
        while (number !=0) {
            System.out.print("masukkan angka (0 untuk keluar) : " );
            number = console.nextInt();
            sum=sum +number;
        }
        System.out.println("totalnya adalah " + sum);
    }
}
```

Gambar 2.4

Langkah-langkah :

- a. *Import java.util.Scanner;* → mengimpor class scanner untuk membaca input dari keyboard
- b. *Public class SentinelLoop\_2511532019{* → mendeklarasikan class.
- c. *Public static void main (String[] args){* tempat program java dieksekusi pertama kali. Agar program bisa dijalankan
- d. *Scanner console* → untuk membaca input angka dari pengguna
- e. *Sum* → menyimpan total penjumlahan semua angka yang dimasukkan
- f. *Number* → diberi nilai awal selain 0 agar loop while bisa mulai dijalankan.

- g. *While (number !=0)* → loop akan terus berjalan selama number bukan 0
- h. *System.out.print("masukkan angka (0 untuk keluar):");* → menampilkan ke layer
- i. *Number=console.nextInt();* → membaca angka yang dimasukkan pengguna dan menyimpannya di variabel number.
- j. *Sum=sum+number;* → menambahkan nilai number yang baru dimasukkan ke dalam sum
- k. *System.out.println("totalnya adalah"+sum);* → setelah perulangan berhenti, program mencetak hasil dari sum.

#### 5. doWhile1

```
package pekan6_2511532019;

import java.util.Scanner;

public class doWhile1_2511532019 {
    public static void main (String[] args) {
        Scanner console = new Scanner (System.in);
        String phrase;
        do {
            System.out.print("Input Password : ");
            phrase = console.next ();
        } while (!phrase.equals("abcd"));
        console.close();
    }
}
```

Gambar 2.5

Langkah-langkah:

- a. *Import java.util.Scanner;* → mengimpor class scanner untuk membaca input dari keyboard
- b. *Public class doWhile\_2511532019{* → mendeklarasikan class.
- c. *Public static void main (String[] args){* tempat program java dieksekusi pertama kali. Agar program bisa dijalankan.
- d. *Scanner console* → untuk membaca input angka dari pengguna

- e. `String phrase` ➔ mendeklarasikan variabel `phrase` bertipe `String`
- f. `do{` ➔ kode dalam blok `do` dieksekusi minimal satu kali sebelum kondisi perulangan diperiksa.
- g. `System.out.println("input password :");` ➔ meminta input pengguna
- h. `Phrase=console.next();` ➔ membaca satu kata dari input pengguna dan menyimpannya ke variabel `phrase`.
- i. `}while(!phrase.equals("abcd"));` ➔ perulangan akan terus berjalan selama `phrase` tidak sama dengan `"abcd"`

## **BAB III**

### **KESIMPULAN DAN SARAN**

#### **3.1 Kesimpulan**

Dari praktikum ini dapat disimpulkan bahwa while berfungsi mengeksekusi blok kode secara berulang selama suatu kondisi bernilai benar dan tidak akan dieksekusi sama sekali jika kondisi awal yang diperiksa bernilai salah, sedangkan perulangan do while mengeksekusi minimal 1 kali sebelum diperiksa. Pada sentinel loop program terus meminta input dan terus berputar hingga input tersebut memenuhi kriteria penghentian yang telah ditetapkan seperti memasukkan angka. Jadi while digunakan Ketika jumlah iterasi tidak diketahui sebelumnya.

#### **3.2 Saran**

1. Praktikum berikutnya sebaiknya dilengkapi dengan lebih banyak contoh program nyata (kasus sehari-hari) supaya lebih mudah memahami penerapan tipe data dalam menyelesaikan masalah.
2. Materi pembelajaran sebaiknya diberikan sebelum praktikum dimulai, agar dapat dipahami terlebih dahulu, sehingga pada saat praktikum lebih mudah mengerti.

## DAFTAR PUSTAKA

- Dokumen Institusi

[1] Departemen Informatika, *Pedoman Penulisan Laporan Praktikum Java*. Padang: Universitas Andalas, 2025.

- Sumber daring

[2] Oracle, "The while and do-while Statements," [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/while.html>.

[Accessed: Nov. 6, 2025].