

# **LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGRAMAN**

**“Do While”**

**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 PUSTAKA.....	i
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 Perulangan While.....	2
2.2 Sentinel Loop.....	4
2.3 Do While.....	6
2.4 Game Penjumlahan.....	8
2.5 Lempar Dadu.....	1
BAB III KESIMPULAN.....	13
3.1 Ringkasan Hasil Praktikum.....	13

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Dalam pemrograman, perulangan (looping) merupakan salah satu konsep dasar yang digunakan untuk menjalankan suatu blok program secara berulang. Salah satu jenis perulangan adalah Do-While, yaitu perulangan yang memastikan bahwa proses di dalam loop dijalankan paling tidak satu kali sebelum kondisi diperiksa.

## 1.2 Tujuan Praktikum

Tujuan praktikum ini adalah untuk memahami konsep, cara kerja, dan penerapan perulangan Do-While dan versi advance nya dalam pemrograman. Mahasiswa diharapkan mampu membuat program sederhana menggunakan Do-While serta membedakan penggunaannya dengan jenis loop lainnya.

## 1.3 Manfaat Praktikum

Melalui praktikum ini, mahasiswa memperoleh kemampuan dalam menggunakan Do-While dan versi advance nya untuk menyelesaikan permasalahan yang membutuhkan minimal satu kali eksekusi sebelum pengecekan kondisi. Selain itu, praktikum ini meningkatkan keterampilan logika pemrograman dan memperkaya pemahaman tentang struktur kontrol alur dalam bahasa pemrograman.

# BAB II

## PEMBAHASAN

### 2.1 Perulangan While

Kode program:

```

package pekan6_2511533018;

import java.util.Scanner;

public class perulanganWhile1_2511533018 {

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

        int counter = 0;
        String jawab;
        boolean running = true;
        Scanner scan = new Scanner(System.in);

        while (running) {
            counter++;
            System.out.println("Jumlah = " + counter);
            System.out.print(s: "Apakah lanjut (ya/tidak?)");
            jawab = scan.nextLine();

            // Cek jawab = tidak, perulangan berhenti.
            if (jawab.equalsIgnoreCase(anotherString: "tidak")) {
                running = false;
            }
        }

        System.out.println("Anda sudah melakukan perulangan sebanyak " + counter + " kali");
        scan.close();
    }
}

```

Program ini dibuat untuk menampilkan jumlah perulangan yang dilakukan oleh pengguna menggunakan for while pada Java. Program akan terus berjalan selama pengguna menjawab “ya”, dan akan berhenti ketika pengguna menjawab “tidak”. Program memanfaatkan variabel penghitung untuk mencatat berapa kali proses perulangan terjadi, kemudian menampilkan hasilnya ke terminal.

#### Langkah Penyelesaian:

1. Import class Scanner untuk membaca input dari keyboard.
2. Deklarasikan variabel counter dengan base value 0 sebagai penghitung jumlah perulangan.
3. Buat variabel running bertipe boolean dengan nilai true sebagai syarat agar while tetap berjalan.
4. Jalankan perulangan while(running) untuk menjalankan proses selama kondisi true.
5. Setiap iterasi, naikan nilai counter dengan counter++ (incr) dan tampilkan jumlah perulangan.
6. Minta input dari pengguna dengan pertanyaan “Apakah lanjut (ya/tidak?)”.
7. Jika jawaban user adalah “tidak”, ubah running menjadi **false** sehingga perulangan berhenti.
8. Setelah loop selesai, tampilkan total jumlah perulangan yang telah dilakukan.

Hasil:

Program akan menampilkan jumlah perulangan dimulai dari 1 dan terus meningkat setiap kali pengguna menjawab “ya”. Ketika pengguna mengetik “tidak”, program berhenti dan menampilkan total berapa kali perulangan sudah dilakukan.

```
ro_2025_D_2511533018/pekan6_2511533018.perulanganwhite1_2511533018
Jumlah = 1
Apakah lanjut (ya/tidak?)ya
Jumlah = 2
Apakah lanjut (ya/tidak?)tidak
Anda sudah melakukan perulangan sebanyak 2 kali
```

## 2.2 Sentinel Loop

Kode program:

```

import java.util.Scanner;

public class SentinelLoop_2511533018 {

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

        Scanner console = new Scanner(System.in);
        int sum = 0;
        int number = 12;

        while (number != 0) {
            System.out.print(s: "Masukan angka (0 untuk exit)");
            number = console.nextInt();
            sum = sum + number;
        }
        System.out.println("Totalnya adalah " + sum);
        console.close();
    }
}

```

Program ini dibuat untuk menjumlahkan angka yang dimasukkan oleh pengguna menggunakan konsep Sentinel Loop pada Java. Program akan terus meminta input angka dari pengguna hingga angka **0** dimasukkan. Angka 0 berfungsi sebagai *sentinel* atau tanda berhenti. Setiap angka yang dimasukkan (selain 0) akan dijumlahkan dan hasil akhirnya ditampilkan ke layar.

#### Langkah Penyelesaian:

1. Import class Scanner untuk membaca input angka dari user.
2. Deklarasikan variabel sum dengan base value 0 untuk menyimpan total penjumlahan.
3. Buat variabel number bernilai awal 12 (nilai awal bebas, hanya agar while dapat berjalan pertama kali).
4. Jalankan perulangan while(number != 0) sehingga loop terus berlangsung selama input bukan 0.
5. Minta user memasukkan angka dengan teks “Masukan angka (0 untuk exit)”.
6. Simpan input ke variabel number menggunakan nextInt().
7. Tambahkan nilai number ke dalam sum menggunakan sum = sum + number.
8. Jika user memasukkan angka **0**, perulangan berhenti.

9. Setelah loop selesai, tampilkan hasil total penjumlahan ke layar dengan `System.out.println`.

Hasil:

Program akan menjumlahkan seluruh angka yang dimasukkan oleh pengguna. Ketika pengguna memasukkan angka 0, program berhenti dan menampilkan total akumulasi angka yang dimasukkan sebelumnya.

```
ro_2025_D_2511533018/pekan6_2511533018.SentinelLoop_2511533018
Masukan angka (0 untuk exit)1
Masukan angka (0 untuk exit)1
Masukan angka (0 untuk exit)2
Masukan angka (0 untuk exit)3
Masukan angka (0 untuk exit)4
Masukan angka (0 untuk exit)4
Masukan angka (0 untuk exit)0
Totalnya adalah 15
```

## 2.3 Do While

Kode program:

```

package pekan6_2511533018;

import java.util.Scanner;

public class DoWhile_2511533018 {

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

        Scanner console = new Scanner(System.in);
        String phrase;

        do {
            System.out.print(s: "Input password ");
            phrase = console.next();
        } while (!phrase.equals(anObject: "abcd"));
        console.close();

    }
}

```

Program ini dibuat untuk meminta pengguna memasukkan password menggunakan perulangan do-while pada Java. Perulangan ini memastikan bahwa input dari pengguna dijalankan setidaknya satu kali sebelum kondisi diperiksa. Program akan terus meminta input hingga pengguna memasukkan password yang benar yaitu "abcd".

#### Langkah Penyelesaian:

1. Import class **Scanner** untuk membaca input dari pengguna.
2. Deklarasikan variabel phrase bertipe String untuk menyimpan input password.
3. Gunakan perulangan do-while agar program meminta input minimal satu kali.
4. Dalam blok do, tampilkan teks "Input password" dan simpan input pengguna ke variabel phrase.
5. Periksa kondisi di while: selama input tidak sama dengan "abcd", perulangan akan terus berjalan.
6. Setelah pengguna memasukkan password "abcd", perulangan berhenti dan program selesai.
7. Tutup Scanner menggunakan console.close().

Hasil:



Program akan meminta pengguna memasukkan password berulang kali. Perulangan dijalankan minimal satu kali, dan akan terus meminta input hingga pengguna memasukkan password yang benar yaitu "abcd".

```
Input password ahsja
Input password abcd
alhaf@Alhaf 2025_prakalpro_D_2511533018 %
```

## 2.4 Game Penjumlahan

Kode program:

```
public class GamePenjumlahan_2511533018 {  
    Run | Debug  
    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;  
        int limit = 3;  
  
        // Block play the program  
        while (wrong < 3) {  
            int result = play(console, rand);  
            if (result > 0) {  
                points++;  
            } else {  
                wrong++;  
            }  
            if (limit == points || limit == wrong) {  
                System.out.println("You have reach the limit, " + limit + " times");  
                return;  
            }  
        }  
        System.out.println("You earned " + points + " total points.");  
    }  
}
```

```
public static int play(Scanner console, Random rand) {  
    int operands = rand.nextInt(bound: 4) + 2;  
    int sum = rand.nextInt(bound: 10) + 1;  
    System.out.print(sum);  
  
    for (int i = 2; i <= operands; i++) {  
        int n = rand.nextInt(bound: 10) + 1;  
        sum += n;  
        System.out.print(" + " + n);  
    }  
    System.out.print(s: " = ");  
  
    int guess = console.nextInt();  
    if (guess == sum) {  
        return 1;  
    } else {  
        System.out.println("Wrong! The answer was " + sum);  
        return 0;  
    }  
}
```

Program ini dibuat untuk membuat permainan sederhana operasi penjumlahan acak menggunakan Java. Program akan menampilkan beberapa angka secara acak yang harus dijumlahkan oleh pengguna. Pemain akan mendapatkan poin jika jawaban benar, dan permainan berhenti jika salah menjawab 3 kali atau berhasil mencapai 3 poin.

Langkah Penyelesaian:

1. Import class Scanner untuk membaca input dari pengguna dan class Random untuk menghasilkan angka acak.

2. Deklarasikan variabel points untuk menghitung skor, wrong untuk menghitung kesalahan, dan limit untuk batas poin/kesalahan maksimal (3 kali).
3. Jalankan perulangan while(wrong < 3) agar permainan terus berlangsung selama pemain belum salah 3 kali.
4. Panggil method play(console, rand) untuk memainkan satu ronde penjumlahan.
5. Dalam method play:
  - a. Tentukan jumlah operand acak antara 2 sampai 5.
  - b. Buat angka pertama secara acak dan tampilkan ke layar.
  - c. Tambahkan angka acak berikutnya sesuai jumlah operand dan tampilkan sebagai penjumlahan.
  - d. Minta pemain memasukkan hasil penjumlahan.
  - e. Jika jawaban benar, kembalikan nilai 1; jika salah, kembalikan 0 dan tampilkan jawaban yang benar.
6. Di loop utama:
  - a. Tambahkan points jika jawaban benar, tambahkan wrong jika jawaban salah.
  - b. Jika points atau wrong mencapai limit, tampilkan pesan bahwa batas telah tercapai dan hentikan permainan.
7. Setelah permainan selesai, tampilkan total poin yang diperoleh pemain.

Hasil:

Program akan menampilkan soal penjumlahan acak ke layar. Pemain menjawab hasilnya, dan program akan menghitung skor. Jika pemain menjawab salah 3 kali atau mencapai 3 poin, permainan berhenti dan total poin ditampilkan.

```
nt/2025_prakalpro_D_2511533018/prakalpro_2025_D_2511533019.
18.GamePenjumlahan_2511533018
5 + 10 + 7 + 3 + 2 = 10
Wrong! The answer was 27
3 + 8 + 8 = 22
Wrong! The answer was 19
7 + 4 = 11
7 + 7 + 7 = 22
Wrong! The answer was 21
You have reach the limit, 3 times
```

## 2.5 Lempar Dadu

Kode program:

```

public class LemparDadu_2511533018 {

    Run | Debug
    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(bound: 6) + 1;
            int dadu2 = rand.nextInt(bound: 6) + 1;
            sum = dadu1 + dadu2;
            System.out.println(dadu1 + " + " + dadu2 + " = " + sum );
            tries++;
        }
        System.out.println("You won after " + tries + " tries!");
    }
}

```

Program ini dibuat untuk mensimulasikan permainan lempar dadu menggunakan Java. Program akan terus melempar dua dadu hingga jumlah keduanya sama dengan 7. Program juga menghitung berapa kali lemparan dilakukan sebelum mendapatkan jumlah 7.

#### Langkah Penyelesaian:

1. Import class Random (java.util) untuk menghasilkan angka acak pada lemparan dadu.
2. Deklarasikan variabel tries untuk menghitung jumlah lemparan dan sum untuk menyimpan jumlah kedua dadu.
3. Jalankan perulangan while(sum != 7) agar dadu terus dilempar hingga jumlah kedua dadu sama dengan 7.
4. Dalam setiap iterasi:
  - a. Hasil lemparan dadu pertama dadu1 dan dadu kedua dadu2 dibuat secara acak antara 1 sampai 6.
  - b. Hitung sum = dadu1 + dadu2.
  - c. Tampilkan hasil lemparan di layar dengan format “dadu1 + dadu2 = sum”.
  - d. Tambahkan 1 ke variabel tries untuk menghitung jumlah lemparan.
5. Setelah jumlah kedua dadu sama dengan 7, perulangan berhenti.
6. Tampilkan jumlah lemparan yang dibutuhkan untuk memenangkan permainan.

Hasil:

Program akan menampilkan setiap lemparan dua dadu dan jumlahnya. Perulangan berhenti ketika jumlah kedua dadu sama dengan 7, kemudian program menampilkan berapa kali lemparan diperlukan untuk mencapai kemenangan.

```
...ents/Home/bin/java -XX:+ShowCodeDetailsInExceptionMe
nt/2025_prakalpro_D_2511533018/prakalpro_2025_D_25115
18.Lempardadu_2511533018
5 + 4 = 9
5 + 4 = 9
4 + 2 = 6
3 + 4 = 7
You won after 4 tries!
❖ althaf@Althaf 2025_prakalpro_D_2511533018 %
```

## **BAB III**

### **KESIMPULAN**

#### **3.1 Ringkasan Hasil Praktikum**

1. Perulangan While

- a. Program menghitung jumlah perulangan berdasarkan input pengguna.
  - b. Perulangan berjalan selama pengguna menjawab “ya”.
  - c. Hasil akhir menampilkan total berapa kali perulangan dilakukan.
2. Sentinel Loop (SentinelLoop\_2511533018)
- a. Program menjumlahkan angka yang dimasukkan pengguna hingga memasukkan angka 0.
  - b. Angka 0 berfungsi sebagai *sentinel* atau tanda berhenti perulangan.
  - c. Hasil akhir menampilkan total penjumlahan angka yang dimasukkan.
3. Do-While
- a. Program meminta input password berulang kali hingga password yang benar dimasukkan (contoh: "abcd").
  - b. Perulangan dijalankan minimal satu kali karena menggunakan `do...while`.
  - c. Hasil akhir: program berhenti ketika password benar.
4. Game Penjumlahan
- a. Program memberikan soal penjumlahan acak kepada pengguna.
  - b. Pemain mendapat poin jika jawaban benar dan kesalahan dihitung.
  - c. Permainan berhenti ketika pemain mencapai 3 poin atau salah 3 kali.
  - d. Hasil akhir menampilkan total poin yang diperoleh pemain.
5. Lempar Dadu
- a. Program melempar dua dadu secara acak hingga jumlahnya sama dengan 7.
  - b. Menghitung berapa kali lemparan diperlukan untuk mendapatkan jumlah 7.
  - c. Hasil akhir menampilkan jumlah lemparan yang dilakukan sebelum menang.

## DAFTAR PUSTAKA

1. Deitel, P. J., & Deitel, H. M. (2017). *Java How to Program* (11th Edition). Pearson.
2. Schildt, H. (2018). *Java: The Complete Reference* (11th Edition). McGraw-Hill Education.
3. Oracle. (2025). *The Java™ Tutorials – Lesson: Loops*. Retrieved from



<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/while.html>

4. Oracle. (2025). *The Java™ Tutorials – Lesson: Random Numbers*. Retrieved from

<https://docs.oracle.com/javase/8/docs/api/java/util/Random.html>