

**LAPORAN PRAKTIKUM 6**  
**ALGORITMA DAN PEMROGRAMAN**  
**PENGUNAAN WHILE DAN DO WHILE DALAM**  
**PEMROGRAMAN JAVA**

**Disusun Oleh:**

Dinda Amelia

(2511531020)

**Dosen Pengampu:**

Dr. Wahyudi, S.T, M.T

**Asisten Praktikum:**

Muhammad Zaki Al Hafiz



**DEPARTEMEN INFORMATIKA**  
**FAKULTAS TEKNOLOGI INFORMASI**  
**UNIVERSITAS ANDALAS**  
**PADANG 2025**

## **KATA PENGANTAR**

Segala puji bagi Allah SWT yang telah memberikan kesempatan serta kemudahan kepada penulis untuk dapat menyelesaikan Laporan Praktikum pada mata kuliah Algoritma Pemrograman, sehingga Laporan Praktikum ini dapat dikumpulkan dengan tepat waktu. Atas rahmat dan karunianya Laporan Praktikum dapat terselesaikan dengan baik. Sholawat serta salam juga penulis sampaikan kepada baginda tercinta yaitu Nabi Muhammad SAW. yang kita nantikan syafa'atnya di akhirat nanti. Laporan Praktikum ini bertujuan untuk menambah wawasan para pembaca untuk lebih memperdalam ilmu yang ada pada makalah ini.

Laporan pratikum ini disusun sebagai bentuk penanggung jawaban atas pelaksanaan kegiatan pratikum mata kuliah algoritma pemograman yang membahas tentang perulangan WHILE dan DO WHILE. Melalui pemograman ini penulis dapat mengetahui bagaimana struktur kontrol bertingkat untuk menghasilkan pola karakter untuk dilayar .Dalam penyusunan Laporan Praktikum ini, penulis mengalami banyak kesulitan dan penulis menyadari bahwa Laporan Praktikum ini masih jauh dari kesempurnaan. Untuk itu, penulis sangat mengharapkan kritik dan saran yang membangun demi kesempurnaan pada Laporan Praktikum ini.

Padang,November 2025

Dinda Amelia

## DAFTAR ISI

KATA PENGANTAR.....	2
DAFTAR ISI .....	3
BAB I .....	4
PENDAHULUAN.....	4
<b>1.1    Latar Belakang</b> .....	4
<b>1.2    Tujuan</b> .....	4
<b>1.3    Manfaat Praktikum</b> .....	4
BAB 2 .....	5
PEMBAHASAN .....	5
2.1 Class pertama ‘perulanganWhile1’ .....	5
2.1.1 Kode Program .....	5
2.1.2 Langkah Kerja.....	5
2.1.3 Kode Program .....	6
2.2 Class kedua ‘Lempardadu’ .....	6
2.2.1 Kode Program .....	6
2.2.2 Langkah Kerja.....	6
2.2.3 Analisis Hasil .....	7
2.3 Class ketiga Game Penjumlahan .....	8
2.3.1 Kode Program .....	8
2.2.3 Langkah Kerja.....	8
2.2.3 Analisis Hasil .....	9
2.4 Class keempat SentinelLoop.....	10
2.4.1 Kode Program .....	10
2.4.2 Langkah Kerja.....	10
2.4.3    Analisis Hasil.....	11
2.5 Class kelima Do-while .....	11
2.5.1 Kode Program .....	11
2.5.2 Langkah Kerja.....	11
2.5.3 Analisis Hasil .....	12
KESIMPULAN .....	13
DAFTAR PUSTAKA .....	14

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Looping atau perulangan adalah operator dalam bahasa pemrograman yang memungkinkan kita untuk mengeksekusi sebuah perintah hingga kondisinya bernilai FALSE. Selain menggunakan looping FOR, terdapat juga perulangan WHILE dan DO WHILE. Perulangan WHILE adalah salah satu operator looping yang digunakan dalam kondisi ketika tidak mengetahui pasti jumlah iterasi dan perulangan akan terus dilakukan selama memenuhi kondisi tertentu. WHILE akan diperiksa sebelum perulangan pertama dicetak. Apabila perulangan bernilai TRUE, maka program akan berlanjut hingga nilainya FALSE. Sedangkan perulangan DO WHILE berkebalikan dengan WHILE. Perulangan DO WHILE akan menjalankan perintah untuk pertama kali lalu diperiksa sesuai kondisi. Apabila perintah bernilai TRUE, maka perulangan akan dilanjutkan, namun jika bernilai FALSE, perulangan akan berhenti di perulangan pertama saja.

### 1.2 Tujuan

- 1.2.1 Untuk memahami bagaimana mengoperasikan perulangan WHILE dalam bahasa pemrograman Java
- 1.2.2 Untuk memahami bagaimana mengoperasikan perulangan DO WHILE dalam bahasa pemrograman Java.

### 1.3 Manfaat Praktikum

- 1.3.1 Membandingkan *While* dan *Do-While*: Laporan memaksa praktikan untuk menjelaskan dan mempraktikkan perbedaan mendasar antara *while* (kondisi dicek di awal, perulangan bisa tidak terjadi sama sekali) dan *do-while* (kondisi dicek di akhir, perulangan terjadi minimal satu kali).
- 1.3.2 Mengoptimalkan Penggunaan *Loop*: Praktikan belajar memilih perulangan yang paling efisien dan tepat untuk kasus tertentu.

## BAB 2

### PEMBAHASAN

#### 2.1 Class pertama ‘perulanganWhile1’

##### 2.1.1 Kode Program

```
1 package pekan6_2511531020;
2
3 import java.util.Scanner;
4
5 public class perulanganWhile1_2511531020 {
6     public static void main(String [] args) {
7
8         int counter=0;
9         String jawab ;
10        boolean running = true;
11        // deklarasi Scanner
12        Scanner scan = new Scanner (System.in);
13        while (running) {
14            counter++;
15            System.out.println("Jumlah = "+counter);
16            System.out.print("Apakah lanjut (ya/ tidak?)");
17            jawab= scan.nextLine();
18            // cek jawab = tidak, perulangan berhenti
19            if (jawab.equalsIgnoreCase("tidak")) {
20                running= false;
21            }
22        }
23        System.out.println("Anda sudah melakukan perulangan sebanyak "+counter+" kali");
24    }
25 }
26
27
28
```

##### 2.1.2 Langkah Kerja

Dalam class ini kita menggunakan Scanner untuk menginput jawaban.

- Start: int counter=0;  
String jawab;  
boolean running = true;
- Import scanner: Scanner scan = new Scanner(System.in);
- While (condition): while (running) {
- Increment/decrement: counter++ (untuk menaikkan value counter)
- Kode program:  
System.out.println("Jumlah = "+counter);  
System.out.print("Apakah lanjut (ya / tidak?)");
- Memasukkan value string jawab dari inputan keyboard:  
jawab= scan.nextLine();
- Menghentikan program jika jawaban bernilai false:  
if (jawab.equalsIgnoreCase("tidak")) {

```
        running= false;
    )
```

- `Println` terakhir untuk mencetak value akhir counter:

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

### 2.1.3 Kode Program

Jalankan program dengan memilih menu 'Run' sehingga akan menghasilkan output sebagai berikut.

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

## 2.2 Class kedua 'Lempardadu'

### 2.2.1 Kode Program

```
1 package pekan6_2511531020;
2
3 import java.util.Random;
4
5 public class Lempardadu_2511531020 {
6     public static void main(String [] args) {
7         Random rand = new Random();
8         int tries = 0;
9         int sum = 0;
10        while (sum !=7) {
11            //roll the dice once
12            int dadu1 = rand.nextInt(6) + 1;
13            int dadu2 = rand.nextInt(6) + 1;
14            sum = dadu1 + dadu2;
15            System.out.println(dadu1 + " + " + dadu2 + " = " + sum);
16            tries++;
17        }
18        System.out.println("You won after " + tries + " tries!");
19    }
20
21 }
22
```

### 2.2.2 Langkah Kerja

Dalam program ini, kita menggunakan class `Random` untuk menghasilkan bilangan acak dengan batas-batas.

- Import `Random`: `Random rand = new Random();`
- Deklarasi variabel:  
`int tries = 0;`

```
int sum = 0;
```

- While (condition): while (sum != 7)

- Kode program:

```
int dadu1 = rand.nextInt(6) + 1;
```

```
int dadu2 = rand.nextInt(6) + 1;
```

```
sum = dadu1 + dadu2;
```

```
System.out.println(dadu1 + " + " + dadu2 + " = " + sum);
```

- Increment/decrement: tries++; (untuk menaikkan value variabel tries)

- Hasil println terakhir:

```
System.out.println("You won after " + tries + " tries!");
```

Program ini adalah simulasi lempar dadu yang terus dilakukan hingga hasil jumlah dari dua dadu mencapai 7. Program dimulai dengan membuat objek Random untuk menghasilkan angka acak, serta mendeklarasikan variabel tries yang digunakan untuk menghitung jumlah percobaan dan variabel sum untuk menyimpan hasil jumlah kedua dadu. Dalam perulangan while, dua angka acak antara 1 hingga 6 dihasilkan untuk kedua dadu, dan jumlahnya disimpan dalam variabel sum. Program kemudian mencetak hasil jumlah kedua dadu pada setiap percobaan. Proses ini berulang hingga sum mencapai 7, dan setelah itu program menampilkan pesan yang menunjukkan jumlah percobaan yang dibutuhkan untuk mencapai hasil tersebut.

### 2.2.3 Analisis Hasil

Jalankan program dengan memilih menu 'Run' dan akan menghasilkan output sebagai berikut.

```
Simulasi Lempar Dadu
4 + 2 = 6
5 + 2 = 7
You won after 2 tries!
```

## 2.3 Class ketiga Game Penjumlahan

### 2.3.1 Kode Program

```
1 package pekan6_2511531020;
2
3 import java.util.Scanner;
4
5 public class GamePenjumlahan_2511531020 {
6     public static void main(String [] args) {
7         Scanner console = new Scanner (System.in);
8         Random rand = new Random ();
9         // play until user gets 3 wrong
10        int points = 0;
11        int wrong = 0;
12        while (wrong < 3) {
13            int result = play(console, rand); //play one game
14            if (result > 0) {
15                points++;
16            } else {
17                wrong++;
18            }
19        }
20        System.out.println("You earned " + points + " total points.");
21    }
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        for (int i = 2; i <= operands; i++) {
30            int n = rand.nextInt(10);
31            sum += n;
32            System.out.print(" + " + n);
33        }
34        System.out.print(" = ");
35    }
36
37    //read user's 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("Wrong! The answer was " + sum);
43        return 0;
44    }
45 }
46 }
47 }
```

### 2.2.3 Langkah Kerja

Program **GamePenjumlahan** merupakan permainan sederhana untuk melatih kemampuan penjumlahan menggunakan bahasa Java. Program ini menggunakan **perulangan while** untuk menjalankan permainan selama pengguna belum melakukan tiga kesalahan, serta **method play()** untuk membuat soal penjumlahan acak dan mengevaluasi jawaban pengguna. Dalam setiap permainan, program



menghasilkan beberapa angka acak menggunakan **class Random**, menampilkannya sebagai soal penjumlahan, lalu meminta input dari pengguna melalui **Scanner**. Jika jawaban benar, poin bertambah; jika salah, jumlah kesalahan meningkat. Setelah pengguna salah tiga kali, permainan berakhir dan menampilkan total poin yang diperoleh.

#### Penjelasan kode penting:

1. `while (wrong < 3)` — mengatur agar permainan terus berjalan selama pengguna belum salah tiga kali.
2. `int result = play(console, rand);` — memanggil method yang membuat soal dan mengecek jawaban.
3. `if (result > 0) points++; else wrong++;` — menentukan apakah pengguna mendapat poin atau menambah kesalahan.
4. `System.out.println("You earned " + points + " total points.");` — menampilkan hasil akhir permainan.

#### 2.2.3 Analisis Hasil

Jalankan program dengan memilih menu 'Run' dan akan menghasilkan output sebagai berikut.

```
1 + 5 + 8 + 7 + 6 = 27
4 + 5 + 0 = 9
10 + 9 = 20
Wrong! The answer was 19
6 + 1 + 8 + 6 + 5 = 30
Wrong! The answer was 26
3 + 8 = 8
Wrong! The answer was 11
You earned 2 total points.
```

## 2.4 Class keempat SentinelLoop

### 2.4.1 Kode Program

```
1 package pekan6_2511531020;
2
3 import java.util.Scanner;
4 public class SentinelLoop_2511531020 {
5     public static void main(String [] args) {
6         Scanner console = new Scanner(System.in);
7         int sum = 0;
8         int number=12; // "dummy value", anything but 0
9
10        while ( number != 0) {
11            System.out.print("Masukkan angka (0 untuk keluar): ");
12            number = console.nextInt();
13            sum = sum + number;
14        }
15        System.out.println("totalnya adalah " + sum);
16    }
17
18 }
```

### 2.4.2 Langkah Kerja

Import scanner: `Scanner console = new Scanner(System.in);`

- Deklarasi variabel:
- `int sum = 0;`  
`int number = 12` (nilai sementara)
- While (condition): `while (number != 0)`
- Kode program:  
`System.out.print("Masukkan angka (0 untuk keluar): ");`  
`number = console.nextInt();`  
`sum = sum + number;`
- Hasil `println` terakhir: `System.out.println("totalnya adalah " + sum);`

Program ini adalah implementasi dari sentinel loop yang meminta pengguna untuk memasukkan angka secara berulang hingga angka 0 dimasukkan, yang berfungsi sebagai sentinel untuk menghentikan perulangan. Program dimulai dengan mendeklarasikan objek Scanner untuk membaca input dari pengguna dan variabel sum untuk menyimpan total penjumlahan angka yang dimasukkan. Di dalam perulangan while, program meminta pengguna untuk memasukkan angka, dan angka tersebut akan ditambahkan ke dalam variabel sum. Perulangan akan terus berjalan hingga pengguna memasukkan angka 0, yang menandakan akhir dari input. Setelah perulangan berhenti, program akan menampilkan total jumlah angka yang telah dimasukkan oleh pengguna sebelum angka 0

### 2.4.3 Analisis Hasil

Program akan menghasilkan output sebagai berikut:

```
Masukkan angka (0 untuk keluar): 2
Masukkan angka (0 untuk keluar): 3
Masukkan angka (0 untuk keluar): 0
totalnya adalah 6
```

## 2.5 Class kelima Do-while

### 2.5.1 Kode Program

```
1 package pekan6_2511531020;
2
3 import java.util.Scanner;
4
5 public class doWhile1_2511531020 {
6     public static void main(String [] args) {
7         Scanner console = new Scanner(System.in);
8         String phrase;
9         do {
10             System.out.print("Input passswaord: ");
11             phrase = console.next();
12         } while (!phrase.equals("abcd"));
13     }
14
15 }
```

### 2.5.2 Langkah Kerja

Import scanner: `Scanner console = new Scanner(System.in);`

- Deklarasi string: `String phrase;`
- Do dan kode program:  

```
do {  
    System.out.print("Input Password: ");  
    phrase = console.next();  
}
```
- While (condition): `while (!phrase.equals("abcd"));`

Program ini adalah aplikasi Java yang menggunakan do-while loop untuk meminta input dari pengguna hingga kata sandi yang dimasukkan benar, yaitu "abcd". Program dimulai dengan mengimpor kelas Scanner untuk membaca input pengguna. Di dalam metode main, objek Scanner bernama console dibuat untuk menangkap input, dan variabel phrase digunakan untuk menyimpan input tersebut. Dalam blok do, program menampilkan "input Password:" dan meminta pengguna memasukkan kata sandi. Setelah itu, program memeriksa apakah input yang diberikan cocok dengan string "abcd" menggunakan metode equals. Jika input salah, loop akan terus meminta kata sandi hingga yang benar dimasukkan, dan program berhenti ketika kata sandi yang tepat, "abcd", dimasukkan. Berikut outputnya.

### 2.5.3 Analisis Hasil

Jalankan program dengan memilih menu 'Run' dan akan menghasilkan output sebagai berikut.

```
Input passswaord: 1234
Input passswaord: 4567
Input passswaord: abcd
```

## KESIMPULAN

Pada praktikum pekan 6 ini, mahasiswa belajar memahami dan menerapkan berbagai jenis **perulangan (loop)** dalam bahasa pemrograman Java. Setiap jenis perulangan memiliki fungsi dan karakteristik yang berbeda dalam mengontrol alur program. Melalui latihan ini, peserta praktikum dapat melihat bagaimana **struktur kontrol** seperti while, do-while, dan sentinel loop bekerja dalam situasi yang berbeda sesuai kebutuhan logika program.

Program-program pada praktikum pekan 6 ini menunjukkan penggunaan berbagai jenis perulangan seperti perulangan while 1, lempar dadu, game penjumlahan, sentinel loop, do while. Program pertama perulangan while 1 digunakan untuk menjalankan proses secara berulang selama kondisi bernilai benar. Program kedua lempar dadu dengan menggunakan while loop hingga jumlah dua dadu mencapai angka 7. Program ketiga game penjumlahan mengajarkan cara menggunakan perulangan, method, dan input pengguna untuk membuat permainan interaktif dan melalui program ini, kita belajar menerapkan logika kontrol sederhana untuk menghitung skor dan batas kesalahan. Program keempat menggunakan sentinel loop untuk terus meminta pengguna memasukkan angka hingga angka 0 dimasukkan, yang kemudian menampilkan jumlah total angka yang dimasukkan. Program terakhir menggunakan do-while untuk meminta input kata sandi dari pengguna hingga input yang benar dimasukkan. Semua program ini memanfaatkan teknik perulangan dan pengkondisian untuk berinteraksi dengan pengguna dan memanipulasi data dalam berbagai cara.

## DAFTAR PUSTAKA

- [1] Abid, N. J., Sharif, B., Dragan, N., Alrasheed, H., & Maletic, J. I. (2019, May). Developer reading behavior while summarizing java methods: Size and context matters. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)* (pp. 384-395). IEEE.
- Wangsawidjaja, D., Christian, N. M., Immanuel, A. R., & Gumelar, A. B. (2025). Implementasi Algoritma Perhitungan Cicilan Pinjaman Dengan Metode Perulangan Do-While Dalam Bahasa C. *JISEM (Jurnal Informatika, Sistem Informasi, dan Elektro Modern)*, 1(01), 14-19.
- [2] Damopolii, Z. D. P., Koniyo, M. H., & Takdir, R. (2021). Perancangan media pembelajaran berbasis audio visual pada materi looping mata kuliah algoritma dan pemrograman dasar universitas negeri gorontalo. *Inverted: Journal of Information Technology Education*, 1(2), 64-74.
- [3] Suryana, T. (2022). Perulangan Dalam Java Script.
- [4] Hartono, B. (2022). Pemrograman Java untuk Pemula. *Penerbit Yayasan Prima Agus Teknik*, 1-89. J