

LAPORAN PRAKTIKUM
PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN
“PERULANGAN WHILE & DO WHILE”



disusun Oleh:

FILZI JELILA INDA ROBBANI

2511533019

Dosen Pengampu:

Dr. WAHYUDI, S.T, M.T.

Asisten Praktikum:

JOVANTRI IMMANUEL GULO

DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Dengan memanjatkan puji syukur kehadirat Allah SWT, atas limpahan rahmat dan karunia-Nya sehingga praktikum Algoritma dan Pemrograman kali ini dapat saya selesaikan dengan baik dan lancar. Shalawat serta salam semoga senantiasa tercurahkan kepada junjungan kita Nabi Muhammad SAW.

Laporan ini saya susun untuk memenuhi salah satu tugas pada mata kuliah Praktikum Algoritma dan Pemrograman di Universitas Andalas. Laporan ini diharapkan dapat menambah wawasan mengenai *perulangan While dan Do While* dalam bahasa pemrograman Java.

Saya menyampaikan terima kasih kepada dosen pengampu dan asisten praktikum yang telah membimbing serta memberikan arahan selama proses pembelajaran di kelas maupun di laboratorium komputer. Selain itu, saya juga berterima kasih kepada teman-teman praktikan atas bantuan dan dukungan yang diberikan, sehingga laporan ini dapat terselesaikan dengan baik.

Saya menyadari laporan ini masih jauh dari sempurna. Oleh karena itu, saya mohon maaf apabila terdapat kekurangan maupun kesalahan dalam penyusunan laporan ini. Harapan saya, semoga laporan ini dapat memberikan manfaat serta menambah pengetahuan bagi pembaca sekalian.

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan	1
1.3 Manfaat	2
BAB II PEMBAHASAN.....	3
2.1 Langkah Kerja Praktikum.....	3
2.1.1 Membuat program pertama (perulangan <i>While</i>)	3
2.1.2 Membuat program kedua (perulangan <i>While</i>).....	4
2.1.4 Membuat program keempat (perulangan <i>Do While</i>).....	8
2.1.5 Membuat program kelima (perulangan <i>Do While</i>).....	10
BAB III PENUTUP.....	12
3.1 Kesimpulan.....	12
DAFTAR PUSTAKA.....	13

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Pada praktikum sebelumnya, telah dipelajari mengenai struktur perulangan dalam bahasa pemrograman Java yaitu perulangan *for* dan *nested for*. Struktur perulangan atau *looping* merupakan salah satu konsep penting dalam pemrograman yang digunakan untuk mengeksekusi suatu blok kode secara berulang sesuai dengan jumlah iterasi yang ditentukan atau hingga syarat tertentu terpenuhi.. Setelah memahami konsep perulangan *for* dan *nested for*, langkah selanjutnya dalam mempelajari struktur perulangan pada pemrograman Java adalah memahami konsep *perulangan (looping) while* dan *do while*.

Dengan memahami struktur perulangan *while* dan *do-while*, kita dapat membuat program yang mampu menjalankan proses berulang berdasarkan kondisi tertentu. Kedua jenis perulangan ini sangat berguna dalam situasi ketika jumlah iterasi belum ditentukan sejak awal. Perulangan *while* akan terus berjalan selama kondisi bernilai *true*, sedangkan perulangan *do-while* menjamin bahwa suatu proses tetap dijalankan minimal satu kali sebelum pengecekan kondisi dilakukan.

Oleh sebab itu, pada praktikum kali ini kita akan mempelajari penggunaan perulangan *while* dan *do while* dalam penyelesaian berbagai permasalahan sederhana dalam pemrograman. Praktikum ini bertujuan agar mahasiswa mampu memahami cara kerja perulangan berbasis kondisi dan dapat mengimplementasikannya untuk membangun program yang lebih dinamis, fleksibel, dan logis.

1.2 Tujuan

Tujuan dilaksanakannya praktikum ini adalah sebagai berikut:

- a. Memahami konsep dasar perulangan (looping) dalam bahasa pemrograman Java

- b. Mempelajari cara kerja dan penggunaan struktur perulangan *While* dan *Do While*
- c. Mampu menerapkan perulangan *While* dan *Do While* untuk menyelesaikan masalah pemrograman yang melibatkan proses berulang
- d. Mengembangkan kemampuan logika pemrograman agar dapat menulis kode secara lebih efisien dan terstruktur

1.3 Manfaat

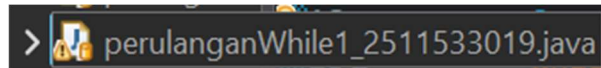
- a. Dapat memahami cara kerja perulangan berbasis kondisi.
- b. Menambah bekal dasar mempelajari konsep pemrograman yang lebih lanjut.
- c. Melatih keterampilan mahasiswa dalam menuliskan kode yang efektif dan efisien.
- d. Dapat mengimplementasikannya untuk membangun program yang lebih dinamis, fleksibel, dan logis

BAB II PEMBAHASAN

2.1 Langkah Kerja Praktikum

2.1.1 Membuat program pertama (perulangan *While*)

- 1) Pertama buka apl Eclipse IDE for Java Developers, lalu buatlah package baru dengan mengklik kanan di folder src. Setelah itu beri nama “pekan6_NIM”. Lalu buat new class dan beri nama class “PerulanganWhile1_NIM”.



- 2) Lalu, masukkan syntax seperti pada gambar.

```
1 package pekan6_2511533019;
2
3 import java.util.Scanner;
4
5 public final class perulanganWhile1_2511533019 {
6
7     public perulanganWhile1_2511533019() {
8         // TODO Auto-generated constructor stub
9     }
10
11     public static void main(String[] args) {
12         // TODO Auto-generated method stub
13
14         int counter=0;
15         String jawab;
16         boolean running=true;
17         //deklarasi scanner
18         Scanner scan=new Scanner (System.in);
19         while (running) {
20             counter++;
21             System.out.println("jumlah="+counter);
22             System.out.print("apakah lanjut (ya/tidak?)");
23             jawab= scan.nextLine();
24             //cek jawab=tidak, perulangan berhenti
25             if (jawab.equalsIgnoreCase("tidak")) {
26                 running=false;
27             }
28         }
29         System.out.println("anda sudah melakukan perulangan sebanyak"+counter+"kali");
30     }
31 }
32 }
```

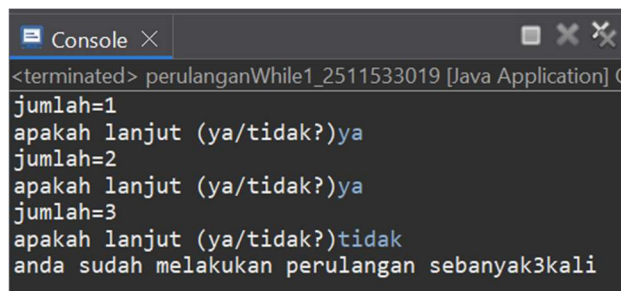
Pada program di atas, saya menggunakan struktur perulangan *While* untuk menjalankan proses berulang berdasarkan kondisi tertentu. Pertama, saya mendeklarasikan variabel *counter* dengan nilai awal 0 sebagai penghitung jumlah perulangan, serta variabel *running* bertipe *boolean* yang diset ke *true* agar perulangan dapat dimulai. Kemudian

dibuat objek *Scanner* untuk membaca input dari user. Perulangan *While* akan terus berjalan selama nilai *running* bernilai *true*. Di dalam blok perulangan, nilai *counter* akan ditambah satu setiap iterasi menggunakan “*counter++*”.

Selanjutnya, program menampilkan jumlah perulangan yang telah dilakukan menggunakan `System.out.println("jumlah="+counter);`. Setelah itu, program meminta input dari user dengan menampilkan pertanyaan “*apakah lanjut (ya/tidak?)*”. Input yang dimasukkan user disimpan pada variabel *jawab*. Jika user mengetik “tidak”, maka kondisi *running* diubah menjadi *false* sehingga perulangan berhenti. Setelah keluar dari perulangan, program akan menampilkan pesan jumlah total perulangan yang telah dilakukan.

Dengan demikian, program ini menunjukkan bagaimana perulangan *While* dapat digunakan untuk menjalankan program secara berulang berdasarkan input user, dan akan berhenti ketika kondisi tertentu terpenuhi.

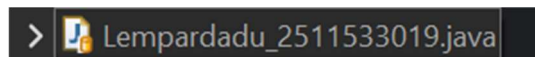
- 3) Run program dengan mengklik tombol run (bulatan hijau di bar menu) dan nanti akan muncul output dari program.



```
<terminated> perulanganWhile1_2511533019 [Java Application] C
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 sebanyak3kali
```

2.1.2 Membuat program kedua (perulangan *While*)

- 1) Buat new class dan beri nama class “Lempardadu_NIM”.



```
> Lempardadu_2511533019.java
```

- 2) Masukkan syntax seperti berikut.

```

1 package pekan6_2511533019;
2
3 import java.util.Random;
4
5 public class Lempardadu_2511533019 {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         Random rand = new Random();
10        int tries=0;
11        int sum=0;
12        while(sum !=7) {
13            //roll the dice once
14            int dadu1=rand.nextInt(6)+1;
15            int dadu2=rand.nextInt(6)+1;
16            sum = dadu1+dadu2;
17            System.out.println(dadu1+" "+dadu2+"="+sum);
18            tries++;
19        }
20        System.out.println("You won after "+tries+" tries!");
21    }
22 }
23 }

```

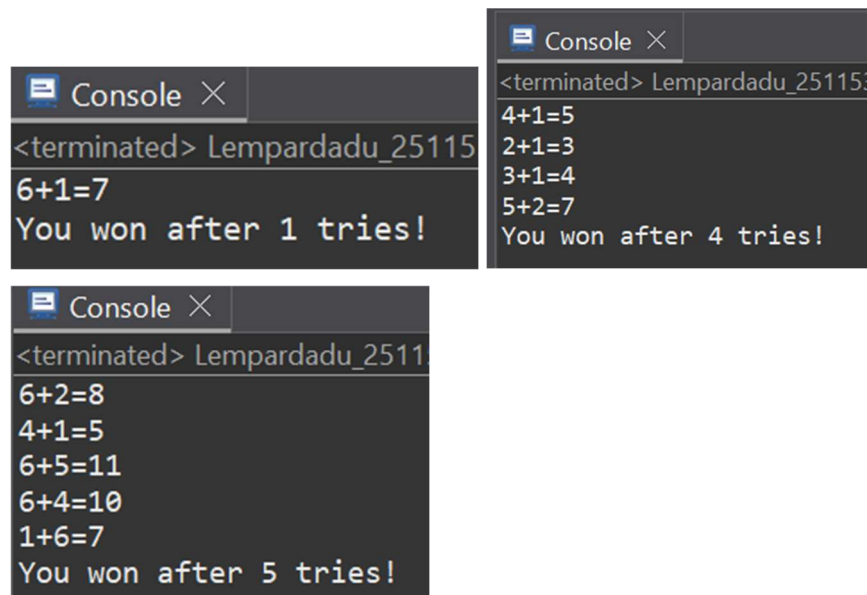
Pada program di atas, saya menggunakan struktur perulangan *While* untuk mensimulasikan permainan lempar dadu hingga mendapatkan jumlah nilai=7. Pertama, dibuat objek *Random* untuk menghasilkan angka acak yang merepresentasikan hasil lemparan dadu. Variabel *tries* digunakan untuk menghitung berapa kali percobaan dilakukan, sedangkan variabel *sum* digunakan untuk menyimpan total nilai dari dua dadu yang dilempar.

Perulangan *while* akan terus berjalan selama nilai *sum* belum=7 (*while(sum != 7)*). Di dalam blok perulangan, program menghasilkan dua angka acak dari 1-6 yang mewakili dua dadu (*dadu1* dan *dadu2*). Nilai kedua dadu tersebut dijumlahkan dan disimpan dalam variabel *sum*. Setelah itu, program menampilkan hasil lemparan dadu dan total nilainya pada layar menggunakan *System.out.println*. Setiap kali perulangan berlangsung, variabel *tries* akan ditambah satu untuk mencatat jumlah percobaan lemparan yang sudah dilakukan.

Perulangan berhenti ketika jumlah kedua dadu menghasilkan nilai=7. Setelah keluar dari perulangan, program menampilkan pesan bahwa pemain menang dan mencantumkan berapa kali percobaan yang

diperlukan untuk mendapatkan total nilai=7. Dengan demikian, program ini memperlihatkan bagaimana perulangan *While* dapat digunakan untuk menjalankan proses secara berulang sampai suatu kondisi tertentu tercapai.

- 3) Run program dengan mengklik tombol run (bulatan hijau di bar menu) dan nanti akan muncul output yang berbeda-beda setiap run dari program, seperti keluaran dari program saya berikut(3x run).



```
Console X
<terminated> Lempardadu_25115
6+1=7
You won after 1 tries!
```

```
Console X
<terminated> Lempardadu_251153
4+1=5
2+1=3
3+1=4
5+2=7
You won after 4 tries!
```

```
Console X
<terminated> Lempardadu_2511
6+2=8
4+1=5
6+5=11
6+4=10
1+6=7
You won after 5 tries!
```

2.1.3 Membuat program ketiga (perulangan *While*)

- 1) Buat new class dan beri nama “GamePenjumlahan_NIM”

```
> GamePenjumlahan_2511533019.java
```

- 2) Masukkan syntax seperti berikut.

```

1 package pekan6_2511533019;
2
3 import java.util.Random;
4
5
6 public class GamePenjumlahan_2511533019 {
7
8     public static void main(String[] args) {
9         // TODO Auto-generated method stub
10        Scanner console= new Scanner(System.in);
11        Random rand= new Random();
12        //play until user gets 3 wrong
13        int points=0;
14        int wrong=0;
15        while (wrong<3) {
16            int result= play(console, rand); //play one game
17            if(result>0) {
18                points++;
19            }else {
20                wrong++;
21            }
22        }
23        System.out.println("You earned "+points+" total points.");
24    }
25
26    //membuat soal penjumlahan dan ditampilkan ke user
27    public static int play(Scanner console, Random rand) {
28        //print the operands being added, and sum them
29        int operands= rand.nextInt(4)+2;
30        int sum= rand.nextInt(10)+1;
31        System.out.print(sum);
32        for (int i=2;i<=operands;i++) {
33            int n = rand.nextInt(10)+1;
34            sum += n;
35            System.out.print("+"+n);
36        }
37        System.out.print("=");
38        //read user's guess and report whether it was correct
39        int guess= console.nextInt();
40        if (guess==sum){
41            return 1;
42        }else {
43            System.out.println("Wrong! the answer was " +sum);
44            return 0;
45        }
46    }
47
48 }

```

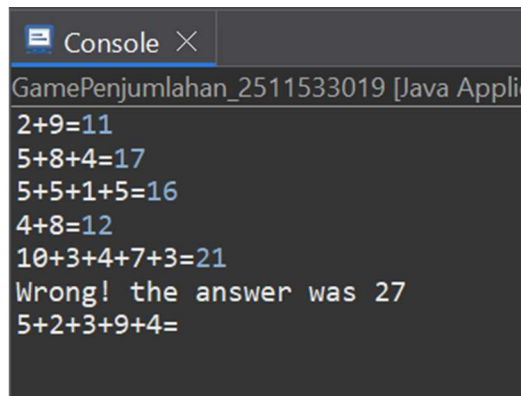
Pada program ini, saya bikin program game penjumlahan sederhana. Jadi nantinya user bakal dikasih soal penjumlahan acak, dan kalau jawabnya benar dapat poin, tapi kalau jawab salah, dihitung sebagai salah. Program bakal terus jalan sampai user salah 3 kali.

Pertama, saya gunakan *Scanner* untuk input dari user dan *Random* untuk ngacak angka. Lalu ada dua variabel, *points* untuk menghitung skor yang benar, dan *wrong* untuk menghitung berapa kali salah. Program memakai perulangan *while(wrong<3)*, artinya selama kesalahan user belum

mencapai 3 kali, game masih lanjut. Di dalam *While*, saya memakai fungsi *play(console, rand)* untuk bikin soal penjumlahan acak dan ngecek jawaban user. Fungsi *play* bakal balikin nilai 1 kalau jawabannya benar, dan 0 kalau salah. Nah kalau yang dikembalikan nilainya 1, poin ditambah satu, tapi kalau 0, variabel *wrong* akan ditambah satu.

Di bagian fungsi *play*, program bakal ngacak beberapa angka untuk dijumlahkan, terus ditampilkan ke layar. Dan nantinya user input jawabannya, dan program ngecek apakah jawabannya sesuai/tidak. Kalau benar program balikin 1, kalau salah, program akan kasih tau jawaban yang benar dan balikin 0. Di akhir, setelah user salah 3 kali, program akan menampilkan total poin yang sudah didapat.

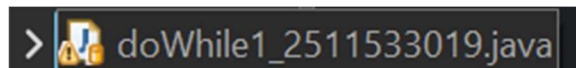
- 3) Run program dengan mengklik tombol run (bulatan hijau di bar menu) dan nanti akan muncul output dari program seperti berikut.



```
GamePenjumlahan_2511533019 [Java Appli
2+9=11
5+8+4=17
5+5+1+5=16
4+8=12
10+3+4+7+3=21
Wrong! the answer was 27
5+2+3+9+4=
```

2.1.4 Membuat program keempat (perulangan *Do While*)

- 1) Buat new class lalu beri nama class “doWhile1_NIM”



```
> doWhile1_2511533019.java
```

- 2) Masukkan syntax seperti berikut.

```

1 package pekan6_2511533019;
2
3 import java.util.Scanner;
4
5 public class doWhile1_2511533019 {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         Scanner console = new Scanner(System.in);
10        String phrase;
11        do {
12            System.out.print("input password: ");
13            phrase = console.next();
14        }while (!phrase.equals("abcd"));
15    }
16
17 }

```

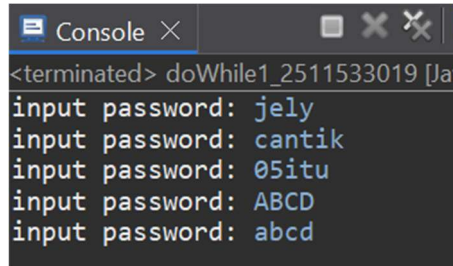
Untuk program ini, saya menggunakan perulangan *Do While*. Pertama, saya memakai objek *Scanner* untuk membaca input dari keyboard dengan perintah *Scanner console = new Scanner(System.in);*. Kemudian saya mendeklarasikan variabel *phrase* yang akan digunakan untuk menyimpan input dari user.

Program akan meminta user untuk memasukkan password dengan perintah *System.out.print("input password: ");*. Nilai yang dimasukkan oleh user akan disimpan ke dalam variabel *phrase* menggunakan *console.next();*.

Setelah itu, perulangan *Do While* digunakan untuk memeriksa apakah input user sudah sesuai dengan password yang benar, yaitu "abcd" atau tidak. Selama nilai *phrase* belum = "abcd", program akan terus mengulang dan meminta user untuk menginputkan password kembali. Jika user sudah memasukkan password yang benar, maka perulangan berhenti dan program selesai dijalankan.

Dengan demikian, program ini berfungsi untuk meminta user menginputkan password hingga input yang diberikan sesuai dengan kata sandi/password yang benar.

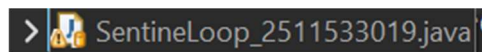
- 3) Run program, lalu akan tampil output akan seperti berikut



```
<terminated> doWhile1_2511533019 [Java]
input password: jely
input password: cantik
input password: 05itu
input password: ABCD
input password: abcd
```

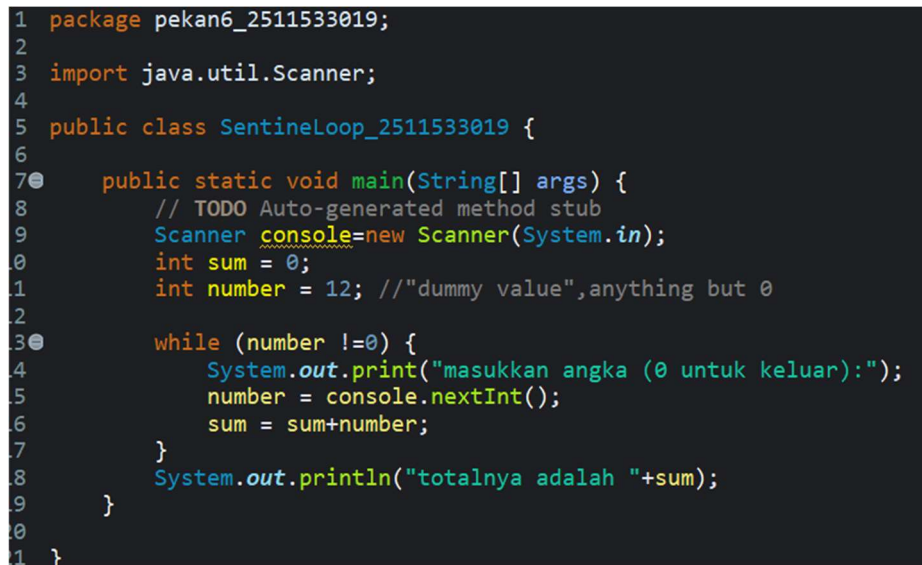
2.1.5 Membuat program kelima (perulangan *Do While*)

- 1) Buat new class dan beri nama class “SentinelLoop_NIM”



```
> SentinelLoop_2511533019.java
```

- 2) Masukkan syntax seperti berikut.



```
1 package pekan6_2511533019;
2
3 import java.util.Scanner;
4
5 public class SentinelLoop_2511533019 {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         Scanner console = new Scanner(System.in);
10        int sum = 0;
11        int number = 12; // "dummy value", anything but 0
12
13        while (number != 0) {
14            System.out.print("masukkan angka (0 untuk keluar):");
15            number = console.nextInt();
16            sum = sum + number;
17        }
18        System.out.println("totalnya adalah " + sum);
19    }
20 }
21 }
```

Pada program ini, saya menggunakan perulangan *While* dengan konsep *sentinel loop*. Pertama, saya membuat objek *Scanner* untuk membaca input dari user melalui keyboard menggunakan perintah *Scanner console = new Scanner(System.in);*.

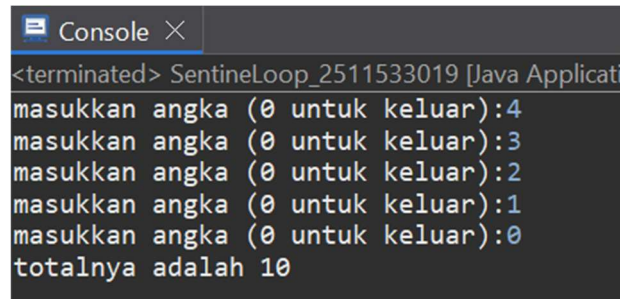
Selanjutnya, saya mendeklarasikan dua variabel, yaitu *sum* dengan nilai awal 0 untuk menyimpan hasil penjumlahan, dan *number* dengan nilai awal 12 sebagai nilai sementara agar perulangan dapat dimulai. Program akan meminta user untuk memasukkan angka dengan menggunakan perintah *System.out.print("masukkan angka (0 untuk keluar): ");*.

Angka yang dimasukkan kemudian disimpan ke dalam variabel *number* menggunakan *console.nextInt()*; Setiap kali user memasukkan angka selain 0, nilai tersebut akan ditambahkan ke variabel *sum* menggunakan *sum = sum + number*;

Perulangan akan terus berjalan selama nilai *number* tidak=0. Jika user memasukkan angka 0, maka perulangan berhenti dan program menampilkan total hasil penjumlahan dari semua angka yang telah dimasukkan dengan perintah *System.out.println("totalnya adalah " + sum)*;

Dengan demikian, program ini berfungsi untuk menjumlahkan semua angka yang diinputkan oleh user hingga angka 0 dimasukkan sebagai tanda berhenti.

- 3) Run program , lalu akan terlihat output seperti berikut.(dengan percobaan saya menginputkan beberapa angka yang ingin saya jumlahkan)



```
<terminated> SentinelLoop_2511533019 [Java Applicat
masukkan angka (0 untuk keluar):4
masukkan angka (0 untuk keluar):3
masukkan angka (0 untuk keluar):2
masukkan angka (0 untuk keluar):1
masukkan angka (0 untuk keluar):0
totalnya adalah 10
```

BAB III

PENUTUP

3.1 Kesimpulan

Dari praktikum ini dapat disimpulkan bahwa penggunaan struktur perulangan *While* dan *Do While* sangat penting dalam pemrograman Java untuk menangani proses yang harus dijalankan secara berulang hingga kondisi tertentu terpenuhi.

Perulangan *While* digunakan ketika pengecekan kondisi dilakukan sebelum proses dijalankan, sehingga perulangan hanya akan berjalan jika kondisi bernilai benar. Sebaliknya, perulangan *Do While* menjalankan proses terlebih dahulu sebelum mengecek kondisi, sehingga minimal satu kali proses pasti akan dijalankan meskipun kondisi awal bernilai salah.

Melalui praktikum ini, kita dapat memahami kapan harus menggunakan *While* atau *Do While*, khususnya pada program yang membutuhkan pengecekan kondisi dinamis, seperti input angka dari user atau proses yang tidak memiliki jumlah pengulangan pasti sejak awal. Dengan menguasai kedua jenis perulangan ini, dapat memberikan kita dasar yang kuat dalam membangun logika program yang fleksibel, terkontrol, dan efisien.

DAFTAR PUSTAKA

- [1] Wahyudi, “*Perulangan While Do While di Java*” PowerPoint slides, Mata Kuliah Algoritma dan pemrograman, Universitas Andalas, Padang, 2025.
- [2] Oracle, “*The while and do-while Statements*,” Oracle Java Documentation. [Online]. Tersedia: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/while.html>
- [3] H. M. Deitel and P. J. Deitel, *Java: How to Program*, 10th ed. Pearson, 2016. Tersedia: <https://www.rmhjaved.com/oop/Java%20-%20How%20to%20Program%2010th%20Ed%20-%20Early%20Objects%20Version%20-%20Deitel.pdf>