

**LAPORAN PRATIKUM PEMROGRAMAN ALGORITMA DAN
PEMROGRAMAN**

PERULANGAN WHILE

disusun Oleh:

Habib Al Faruq

2511532010

Dosen Pengampu: Dr. Wahyudi S.T.M.T

Asisten Praktikum: Muhammad Zaki Al Hafiz



DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur atas kehadiran Tuhan Yang Maha Esa, karena berkat rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan praktikum algoritma pemrograman mengenai perulangan *while* di Java. Penulisan laporan ini bertujuan untuk membuat sebuah laporan tentang materi pemrograman mengenai penggunaan *while* agar program dapat dijalankan berulang kali sesuai dengan kondisi yang diberikan.

Penyusunan laporan praktikum ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, penulis ingin menyampaikan terima kasih kepada dosen pengampu mata kuliah praktikum pemrograman algoritma dan pemrograman, Bapak Dr. Wahyudi. S.T.M.T dan asisten praktikum, Uda Muhammad Zaki Al Hafiz yang telah memberikan arahan dan masukan dalam proses penyusunan laporan praktikum ini. Penulis berharap laporan praktikum ini dapat memberikan manfaat bagi pembaca dalam mempelajari konsep penggunaan *while*.

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang Praktikum	1
1.2 Tujuan Praktikum.....	1
1.3 Manfaat Praktikum	1
 BAB II PEMBAHASAN.....	2
2.1 Perulangan While.....	2
2.1.1 Penjelasan Kode Program.....	2
2.2 Lempar Dadu	3
2.2.1 Penjelasan Kode Program.....	3
2.2.2 Contoh output	4
2.3 Do While	4
2.3.1 Penjelasan Kode Progam	5
2.4 Sentinel Lopp.....	5
2.4.1 Penjelasan Kode Progam	6
2.5 Gmae Penjumlahan.....	6
2.5.1 Penjelasan Kode Progam	7
 BAB III KESIMPULAN.....	8
3.1 Ringkasan Hasil Praktikum.....	8
DAFTAR PUSTAKA.....	9

BAB I

PENDAHULUAN

1.1 Latar Belakang Praktikum

Dalam pemrograman Java, konsep perulangan merupakan salah satu struktur dasar yang sangat penting. Dua jenis perulangan yang sering digunakan dalam bahasa Java yaitu *while* dan *do while*. keduanya berfungsi untuk menjalankan kode program berdasarkan kondisi yang diberikan hingga berhenti pada suatu kondisi tertentu. perbedaan keduanya yaitu *while* memeriksa kondisi sebelum menjalankan isi perulangan, sedangkan *do while* menjalankan isi perulangan setidaknya satu kali.

1.2 Tujuan Praktikum

Tujuan utama dari praktikum ini adalah agar mahasiswa dapat memahami konsep perulangan pada pemrograman bahasa Java. Mahasiswa diharapkan dapat mengetahui bagaimana setiap jenis perulangan itu bekerja dan dapat mengimplementasikan apa yang telah dipelajari pada praktikum ini pada suatu kondisi yang memerlukan perulangan untuk menyelesaikannya. Selain hanya memahami teori perulangan saja, mahasiswa juga menerapkan konsep perulangan sederhana dalam praktikum ini.

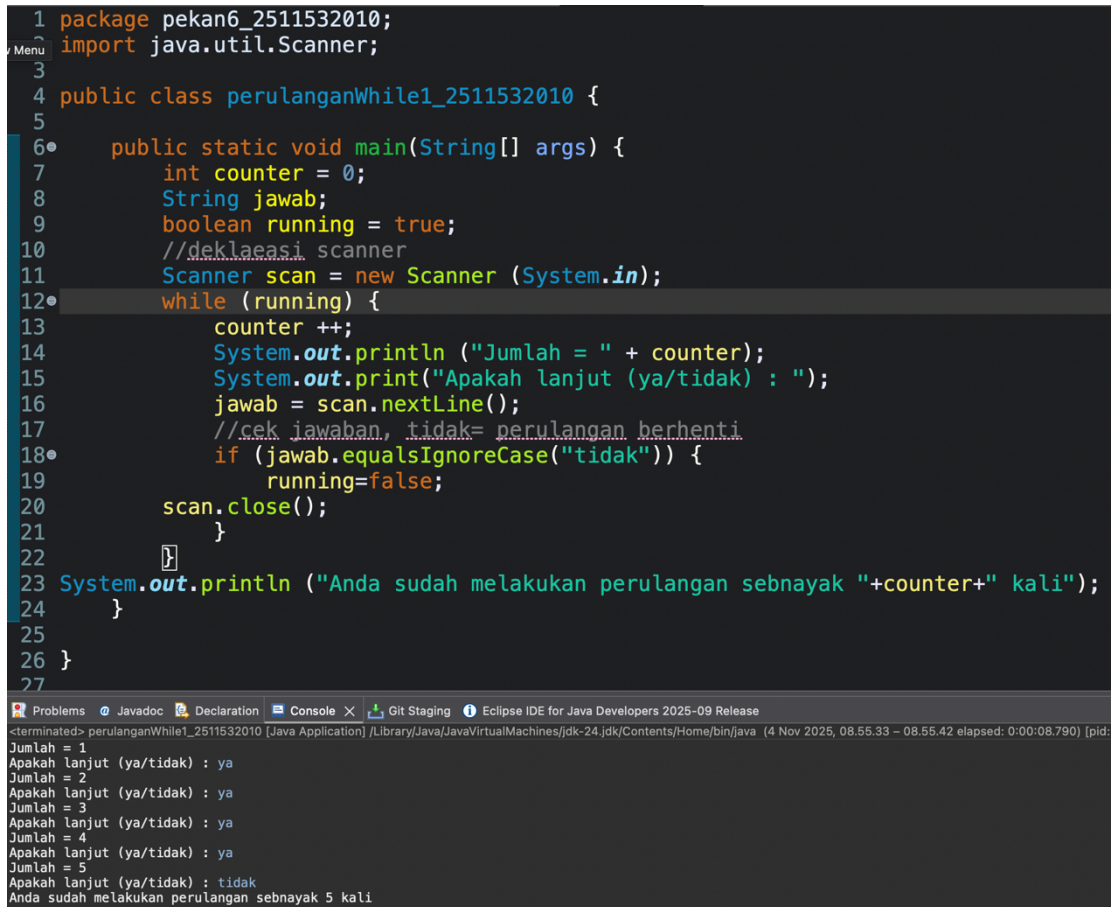
1.3 Manfaat Praktikum

Manfaat dari praktikum ini adalah untuk membiratkan pemahaman yang lebih mendalam kepada para mahasiswa mengenai cara kontrol alur pada program perulangan. Dengan konsep perulangan di java, mahasiswa dapat menulis program secara lebih efisien dan terstruktur. Selain itu, mahasiswa dilatih untuk memiliki kemampuan berpikir secara sistematis dan logis untuk menyelesaikan permasalahan pemrograman dengan melibatkan kondisi perulangan sehingga kemampuan analisis mahasiswa menjadi lebih terasah.

BAB II

PEMBAHASAN

2.1. Perulangan While



```
1 package pekan6_2511532010;
2 import java.util.Scanner;
3
4 public class perulanganWhile1_2511532010 {
5
6     public static void main(String[] args) {
7         int counter = 0;
8         String jawab;
9         boolean running = true;
10        //deklareasi 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 jawaban, tidak= perulangan berhenti
18            if (jawab.equalsIgnoreCase("tidak")) {
19                running=false;
20            }
21            scan.close();
22        }
23        System.out.println ("Anda sudah melakukan perulangan sebanyak "+counter+" kali");
24    }
25 }
26 }
27 }
```

Jumlah = 1
Apakah lanjut (ya/tidak) : ya
Jumlah = 2
Apakah lanjut (ya/tidak) : ya
Jumlah = 3
Apakah lanjut (ya/tidak) : ya
Jumlah = 4
Apakah lanjut (ya/tidak) : ya
Jumlah = 5
Apakah lanjut (ya/tidak) : tidak
Anda sudah melakukan perulangan sebanyak 5 kali

Gambar 2.1

2.1.1 Penjelasan Kode Program

Pada gambar 2.1, kita membuat perulangan while yang mendeklarasikan counter sebagai integer dengan nilai 0, jawab sebagai string, dan running sebagai boolean dengan nilai true. Selanjutnya kita buat fungsi scan agar kita bisa menginputkan sesuatu. lalu, kita buat *while* atau ketika running maka counter++, print "Jumlah = " + counter, print "Apakah lanjut (ya / tidak) lalu masukkan input jawab ke

dalamnya. Jika input jawab yang kita masukkan sama dengan tidak maka running false dan program input otomatis akan berhenti dan program akan langsung mencetak "Anda sudah melakukan perulangan sebanyak " + counter + " kali".

2.2 Lempar Dadu

```
1 package pekan6_2511532010;
2 import java.util.Random;
3 public class Lempardadu_2511532010 {
4
5     public static void main(String[] args) {
6         Random rand = new Random();
7         int tries=0;
8         int sum=0;
9         while (sum !=7) {
10             // roll dice once
11             int dadu1 = rand.nextInt(6) +1;
12             int dadu2 = rand.nextInt(6) + 1;
13             sum= dadu1+dadu2;
14             System.out.println (dadu1 + "+" + dadu2 + "=" +sum);
15             tries++;}
16 System.out.println ("You won after " + tries + " tries!");}}
17
```

Gambar 2.2

2.2.1 Penjelasan Kode Program

Pada gambar 2.2, kita buat dulu import java.util.Random; agar bisa melakukan fungsi random di dalam program . Lalu kita buat Random rand = new Random();. Kita deklarasikan tries sebagai integer dengan nilai 0, sum sebagai integer dengan nilai 0. *While* atau ketika sum!=7 maka nilai dadu1 itu random dari satu sampai enam, nilai dari dadu2 random dari satu sampai enam, sum = dadu1 + dadu2. Setelah itu di print dadu1 + "+" + dadu2 + "=" +sum, tries++. Setelah keluar dari program fungsi while print "You won after " + tries + " tries!".

Makna dari program ini adalah dadu 1 dan dadu 2 akan menghasilkan angka random dari satu hingga enam. Lalu, kedua angka random itu kita tambahkan dan menyimpan nilainya di sum. Selama sum tidak sama dengan 7 maka program akan terus berjalan dan menghasilkan penjumlahan angka random lainnya. Namun, jika sum

sama dengan 7 maka program akan otomatis keluar dari fungsi while dan akan mencetak banyak percobaan yang dilakukan hingga sum sam dengan 7.

2.2.2 Contoh Output

```
4+6=10
4+4=8
4+1=5
4+6=10
2+5=7
You won after 5 tries!
```

Gambar 2.3

```
6+3=9
4+2=6
3+6=9
4+4=8
6+5=11
4+5=9
5+6=11
4+2=6
2+4=6
1+1=2
4+5=9
5+1=6
2+4=6
3+2=5
5+1=6
3+2=5
6+3=9
5+4=9
2+4=6
3+3=6
6+6=12
6+6=12
4+3=7
You won after 23 tries!
```

Gambar 2.4

2.3 Do While

```

1 package pekan6_2511532010;
2 import java.util.Scanner;
3 public class doWhile1_2511532010 {
4
5     public static void main(String[] args) {
6         Scanner console = new Scanner (System.in);
7         String phrase ;
8         do {
9             System.out.print ("Input Password : ");
10            phrase = console. next();
11
12        }while (!phrase.equals("abcd"));}

```

Problems Javadoc Declaration Console X Git Staging Eclipse IDE for Java Developers 2025-09 Release

<terminated> doWhile1_2511532010 [Java Application] /Library/Java/JavaVirtualMachines/jdk-24.jdk/Contents/Home/bin/java (4 Nov 2025, 12.34.10 – 12.34.20 elapsed: 0:00:10.259) [pid: 42537]

Input Password : 1234
Input Password : 4321
Input Password : dbca
Input Password : abcd

Gambar 2.5

2.3.1 Penjelasan Kode Program

Pada gambar 2.5, kita buat dulu import java.util.Scanner; agar bisa melakukan fungsi input. Selanjutnya kita deklarasikan variabel phrase sebagai String. Kita buat do yang di dalamnya kita print "Input Password : " lalu kita menginputkan variabel phrase ke dalamnya. *While* atau ketika phrase nya sama dengan abcd maka program akan otomatis berhenti. Contohnya pada output yang jika kita inputkan angka apapun maka ia akan terus meminta kita untuk menginputkan password. Namun, jika kita menginputkan persis "abcd" maka program akan berhenti.

2.4 Sentinel Loop

```

1 package pekan6_2511532010;
2 import java.util.Scanner;
3
4 public class SentinelLoop_2511532010 {
5
6     public static void main(String[] args) {
7         Scanner console = new Scanner (System.in);
8         int sum =0;
9         int number = 12;
10
11         while (number != 0) {
12             System.out.print("Masukkan angka (0 untuk keluar) : ");
13             number = console.nextInt();
14             sum = sum + number;
15         }System.out.println ("totalnya adalah " + sum);}
16

```

Problems Javadoc Declaration Console X Git Staging Eclipse IDE for Java Developers 2025-09 Release

<terminated> SentinelLoop_2511532010 [Java Application] /Library/Java/JavaVirtualMachines/jdk-24.jdk/Contents/Home/bin/java (4 Nov 2025, 12.34.10 – 12.34.20 elapsed: 0:00:10.259) [pid: 42537]

Masukkan angka (0 untuk keluar) : 1
Masukkan angka (0 untuk keluar) : 2
Masukkan angka (0 untuk keluar) : 34
Masukkan angka (0 untuk keluar) : 5
Masukkan angka (0 untuk keluar) : 0
totalnya adalah 42

Gambar 2.6

2.4.1 Penjelasan Kode Program

Pada gambar 2.6, kita buat dulu import `java.util.Scanner`; agar bisa melakukan fungsi input. Selanjutnya kita deklarasikan variabel `sum` sebagai integer dengan nilai 0 dan variabel `number` sebagai integer dengan 12. *While* atau ketika `number != 0` maka kita print "Masukkan angka (0 untuk keluar) : " lalu menginputkan variabel `number` ke dalamnya, `sum = sum + number`. Jika sudah keluar dari *while* maka kita akan mencetak "totalnya adalah " + `sum`. Maksud dari program ini adalah selama kita tidak menginputkan angka 0 maka program akan terus berjalan dan meminta kita untuk menginputkan suatu angka. Namun, jika angka yang kita inputkan 0 maka program akan otomatis berhenti dan menambahkan semua angka yang telah kita inputkan.

2.5. Game Penjumlahan

```
1 package pekan6_2511532010;
2 import java.util.Random;
3 import java.util.Scanner;
4 public class GamePenjumlahan_2511532010 {
5
6     public static void main(String[] args) {
7         Scanner console = new Scanner (System.in);
8         Random rand = new Random ();
9         int points = 0;
10        int wrong = 0;
11        while (wrong < 3 ) {
12            int result = play(console, rand);
13            if (result > 0) {
14                points++;
15            } else wrong++; } System.out.println("You earned " + points + " total points.");
16
17    public static int play(Scanner console, Random rand) {
18        int operands = rand.nextInt (4) + 2;
19        int sum = rand.nextInt (10) + 1;
20        System.out.print (sum);
21        for (int i = 2; i <= operands; i++) {
22            int n = rand.nextInt(10) + 1;
23            sum += n;
24            System.out.print(" + " + n);
25        }
26        System.out.print(" = ");
27        int guess = console.nextInt();
28        if (guess==sum) {return 1;} else {System.out.println ("Wrong! The Answer Was " + sum);
29        return 0;}}
```

Problems Javadoc Declaration Console X Git Staging Eclipse IDE for Java Developers 2025-09 Release

<terminated> GamePenjumlahan_2511532010 [Java Application] /Library/Java/JavaVirtualMachines/jdk-24.jdk/Contents/Home/bin/java (4 Nov 2025, 12:54:39 – 12:56:07 elapsed: 0:01:27.265)

```
2 + 4 + 6 + 6 + 2 = 20
8 + 3 + 10 + 1 + 4 = 26
1 + 9 = 10
4 + 6 + 2 + 7 + 10 = 29
5 + 4 + 8 + 6 = 0
Wrong! The Answer Was 23
7 + 1 = 12
Wrong! The Answer Was 8
2 + 2 + 1 = 16
Wrong! The Answer Was 5
You earned 4 total points.
```

Gambar 2.7

2.5.1 Penjelasan Kode Program

Pada gambar 2.7, kita buat dulu `import java.util.Scanner;` agar bisa melakukan fungsi input dan `import java.util.Scanner;` agar bisa melakukan fungsi input. Selanjutnya kita deklarasikan variabel `points` sebagai integer dengan nilai 0, variabel `wrong` sebagai integer dengan nilai 0. *While* atau ketika `wrong < 3`, kita memanggil metode `play` yang dikirim objek `console` untuk menginput sesuatu dan `rand` untuk menghasilkan angka random. Jika `result > 0`, `points ++`, jika tidak `wrong ++`. Print "You earned " + `points` + " total points."

Kita mendefinisikan sebuah metode bernama `play`. Operands menghasilkan angk acak dari 2 sampai 5 dan `sum` menghasilkan angka acak dari 1 sampai 10, lalu kita print `sum`. *For* atau untuk `i = 2`, `i <= operands`, `i++`, buat variabel baru sebagai integer dengan nilai angka random dari 1 sampai 10, `sum += n`, dan print " + " + `n`. Kelua dari *for* kita print (" = "). Kita deklarasikan variabel `guess` sebagai integer dengan `console` agar kita bisa menginputkan suatu angka. Jika `guess == sum`, return 1, jika tidak, print "Wrong! The Answer Was " + `sum`, return 0.

Maksud dari program ini yaitu kita diberikan suatu fungsi penjumlahan dengan banyaknya angka yang ingin dijumlahkan berkisar antara 2 sampai lima angka. Sedangkan besaran angka yang akan dijumlahkan itu berkisar antara 1 hingga 0. Kita harus menebak hasil penjumlahan angka random yang diberikan. Jika benar maka ia akan memberikan angka random lainnya untuk dijumlahkan dan poin kita akan bertambah satu. Jika kita salah menjawab maka ia akan memberikan jawaban yang benar dan jika kita salah menjawab sebanyak tiga kali maka program akan otomatis berhenti dan akan mencetak seberapa banyak kita menang atau benar ketika menjawab sebelumnya. Hal tersebut sama persis dengan output pada gambar 2.7.

BAB III

KESIMPULAN

3.1 Ringkasan Hasil Praktikum

Berdasarkan program yang telah kita lakukan, dapat kita simpulkan bahwa perulangan *while* dan *do while* merupakan struktur pengulangan yang sangat penting untuk menentukan dan mengontrol alur suatu program. Perulangan *while* dieksekusi ketika kondisi masih terpenuhi, maka dari itu sangat cocok untuk digunakan jika kita ingin menjalankan program agar program tersebut berjalan pada suatu kondisi tertentu yang diberikan. Sementara *do while* sedikit berbeda dengan proses mengecek dan memastikan bahwa blok kode dieksekusi minimal sekali sebelum pengecekan kondisi dilakukan.

Dengan memahami kedua jenis pengulangan ini pada praktikum kali ini, kita sebagai mahasiswa akan dapat lebih memperkuat pemahaman kita mengenai konsep dasar perulangan pada pemrograman bahasa Java. Pemahaman yang baik terhadap program dasar perulangan pada java membantu mahasiswa dalam mengembangkan kemampuan untuk berpikir logis dan terstruktur.

DAFTAR PUSTAKA

- [1] Oracle, *The Java® Tutorials — Primitive Data Types*. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>
- [2] GeeksforGeeks, *Java Data Types – GeeksforGeeks*. [Online]. Available: <https://www.geeksforgeeks.org/data-types-in-java>