

**LAPORAN PRAKTIKUM ALGORITMA  
PEMROGRAMAN**

**“WHILE CONDITION”**

**DISUSUN OLEH:**

**NAYSHA APRILIA RIZKI**

**2511531014**

**DOSEN PENGAMPU: Dr. WAHYUDI, S.T,M.T**

**ASISTEN PRAKTIKUM: MUHAMMAD ZAKI AL  
HAFIZ**



**DEPARTEMEN INFORMATIKA FAKULTAS  
TEKNOLOGI INFORMASI PROGRAM STUDI  
INFORMATIKA UNIVERSITAS ANDALAS**

**2025**

## **KATA PENGANTAR**

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya, Laporan praktikum ini dapat diselesaikan dengan baik. Laporan praktikum ini disusun sebagai salah satu tugas dalam rangka memenuhi tugas yang diberikan oleh dosen dan juga asisten praktikum.

Padang, 05 November 2025

Naysha Aprilia Rizki

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Perkembangan teknologi informasi yang pesat dalam beberapa dekade terakhir telah memberikan dampak besar terhadap berbagai aspek kehidupan manusia, termasuk di bidang pendidikan, industri, dan komunikasi. Salah satu pilar utama dalam perkembangan teknologi tersebut adalah **pemrograman komputer**. Di antara banyak bahasa pemrograman yang ada, **Java** merupakan salah satu bahasa yang sangat populer dan banyak digunakan di berbagai belahan dunia.

Java adalah bahasa pemrograman tingkat tinggi yang bersifat **object- oriented**, dikembangkan oleh Sun Microsystems pada tahun 1995, dan kini dikelola oleh Oracle. Keunggulan utama Java terletak pada kemampuannya untuk berjalan di berbagai platform (platform- independent) berkat konsep **Write Once, Run Anywhere (WORA)** yang dimilikinya. Hal ini memungkinkan program yang dibuat dalam Java untuk dijalankan di berbagai sistem operasi tanpa perlu diubah ulang.

Selain itu, Java memiliki sintaks yang relatif mudah dipahami bagi pemula, dokumentasi yang lengkap, dan komunitas pengguna yang besar, sehingga sangat mendukung proses belajar dan pengembangan perangkat lunak. Java juga menjadi bahasa utama dalam pengembangan berbagai aplikasi, seperti aplikasi desktop, mobile (Android), web, hingga sistem enterprise berskala besar.

Dengan melihat pentingnya Java dalam dunia pemrograman dan aplikasinya yang luas, sangatlah relevan untuk mempelajari dasar-dasar bahasa pemrograman Java, baik dari segi teori maupun praktik. Oleh karena itu, laporan ini disusun untuk memberikan pemahaman yang lebih mendalam tentang bahasa Java, struktur dasarnya, serta penerapan konsep pemrograman berorientasi objek dalam bahasa tersebut

## **1.2 Tujuan**

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. Meningkatkan keterampilan dalam memecahkan masalah (problem solving) dengan pendekatan algoritmik

## **1.3 Manfaat Praktikum**

1. Mahasiswa dapat memahami konsep dasar pemrograman secara praktis, bukan hanya teori.
2. Melatih keterampilan logika dan analisis dalam menyusun serta mengeksekusi kode.
3. Memberikan pengalaman langsung dalam menggunakan bahasa Java sebagai salah satu bahasa pemrograman populer.
4. Menjadi bekal awal untuk memahami materi pemrograman yang lebih kompleks pada praktikum berikutnya.
5. Membiasakan mahasiswa dengan proses debugging dan pemecahan masalah dalam pemrograman.

## BAB II

### PEMBAHASAN

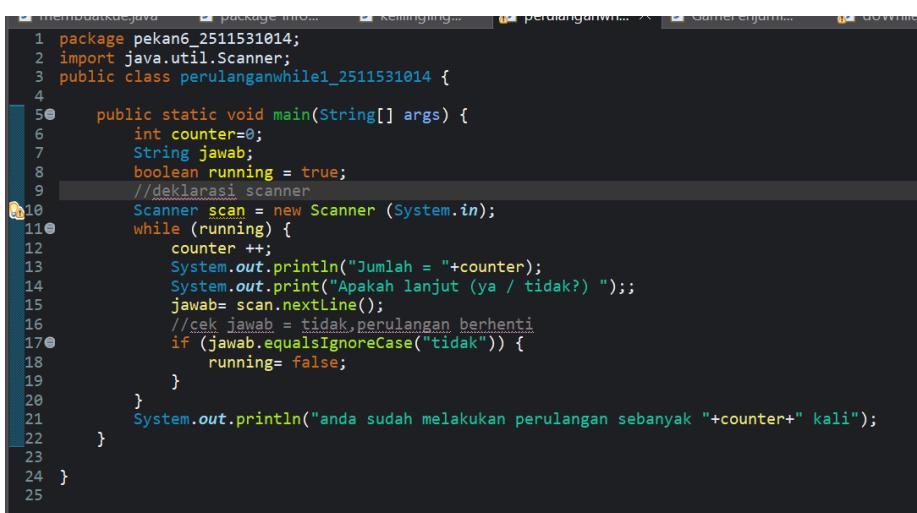
#### 2.1 Pengertian while conditional

Perulangan *while* adalah salah satu bentuk perulangan dalam Java yang digunakan untuk mengeksekusi blok kode selama kondisi yang diberikan bernilai *true*.

Jika kondisi bernilai *false*, maka perulangan akan berhenti dan program melanjutkan ke pernyataan berikutnya.

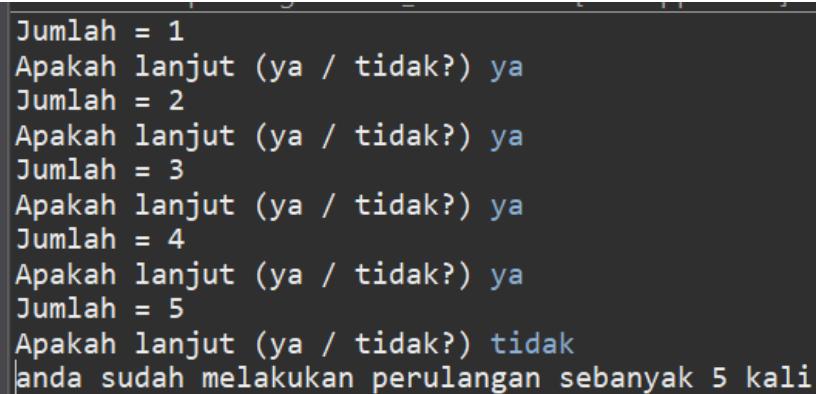
#### 2,2 Praktikum Pekan 6

##### 2.1.1 Perulangan *While* 1



```
1 package pekan6_2511531014;
2 import java.util.Scanner;
3 public class perulanganwhile1_2511531014 {
4
5     public static void main(String[] args) {
6         int counter=0;
7         String jawab;
8         boolean running = true;
9         //deklarasi scanner
10        Scanner scan = new Scanner (System.in);
11        while (running) {
12            counter++;
13            System.out.println("Jumlah = "+counter);
14            System.out.print("Apakah lanjut (ya / tidak?) ");
15            jawab= scan.nextLine();
16            //cek jawab = tidak_perulangan berhenti
17            if (jawab.equalsIgnoreCase("tidak")) {
18                running= false;
19            }
20        }
21        System.out.println("anda sudah melakukan perulangan sebanyak "+counter+" kali");
22    }
23
24 }
```

Pada pemrograman kali ini merupakan contoh dari perulangan *while* yang digabungkan dengan *Scanner* yang berguna untuk menentukan sampai berapa perulangan angka yang kita inginkan seperti output dari pemrograman ini.

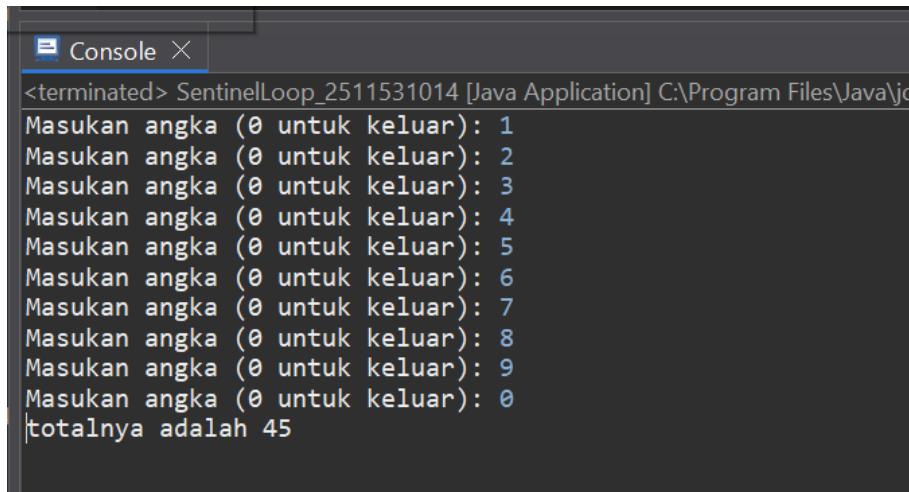


```
Jumlah = 1
Apakah lanjut (ya / tidak?) ya
Jumlah = 2
Apakah lanjut (ya / tidak?) ya
Jumlah = 3
Apakah lanjut (ya / tidak?) ya
Jumlah = 4
Apakah lanjut (ya / tidak?) ya
Jumlah = 5
Apakah lanjut (ya / tidak?) tidak
anda sudah melakukan perulangan sebanyak 5 kali
```

##### 2.1.2 Sentinel Loop

```
1 package pekan6_2511531014;
2 import java.util.Scanner;
3 public class SentinelLoop_2511531014 {
4
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("Masukan angka (0 untuk keluar): ");
12        number = console.nextInt();
13        sum = sum + number;
14    }
15    System.out.println("totalnya adalah " + sum);
16
17
18 }
19
20 }
21
```

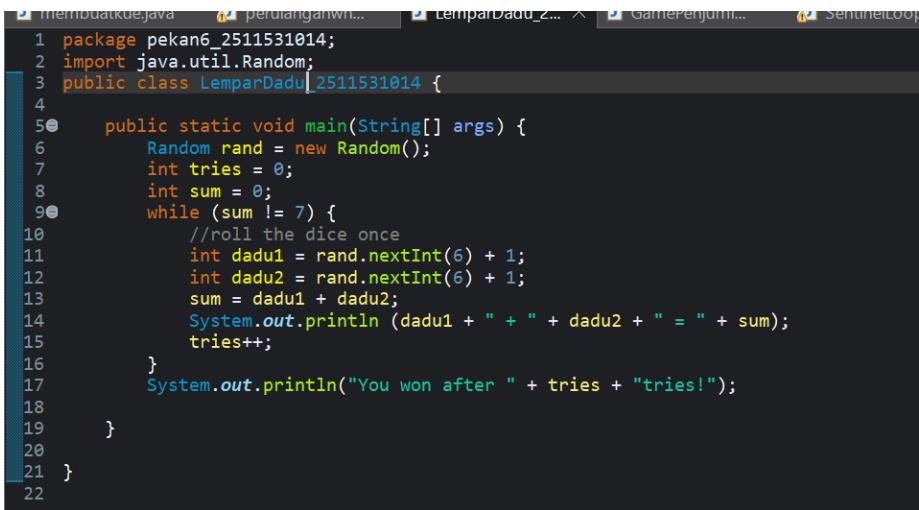
Pada pemrograman kali ini juga merupakan contoh dari perulangan while yang juga menggunakan *Scanner* untuk menemukan sampai berapa angka yang ingin kita tambahkan dari 1 – 9 sedangkan angka 0 sebagai penghenti dari pemrograman tersebut seperti pada output di bawah ini.



The screenshot shows a Java application window titled "Console". The title bar also includes the text "<terminated> SentinelLoop\_2511531014 [Java Application] C:\Program Files\Java\jre1.8.0\_201\bin\javaw.exe". The main area of the window displays the following text:

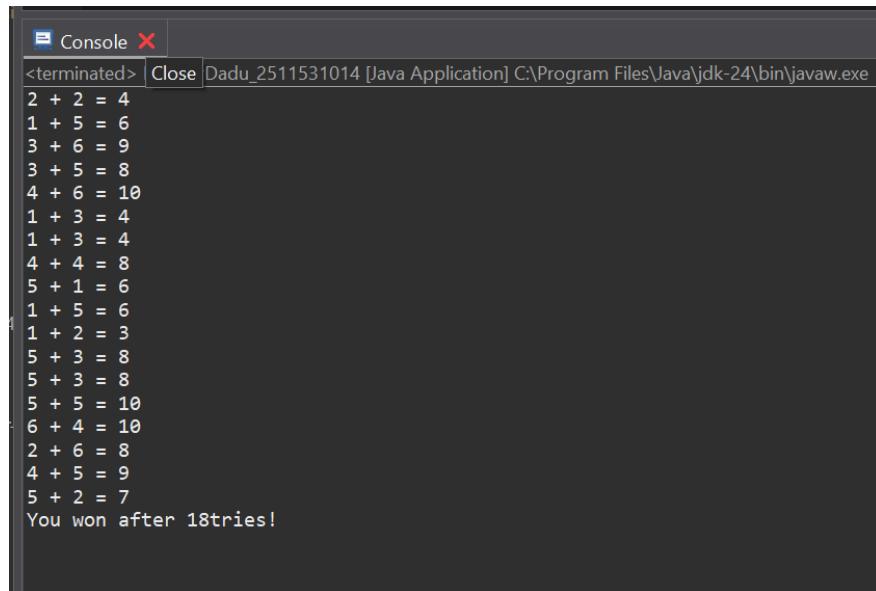
```
Masukan angka (0 untuk keluar): 1
Masukan angka (0 untuk keluar): 2
Masukan angka (0 untuk keluar): 3
Masukan angka (0 untuk keluar): 4
Masukan angka (0 untuk keluar): 5
Masukan angka (0 untuk keluar): 6
Masukan angka (0 untuk keluar): 7
Masukan angka (0 untuk keluar): 8
Masukan angka (0 untuk keluar): 9
Masukan angka (0 untuk keluar): 0
totalnya adalah 45
```

### 2.1.3 Lempar dadu



```
1 package pekan6_2511531014;
2 import java.util.Random;
3 public class LemparDadu_2511531014 {
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 the 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         }
17         System.out.println("You won after " + tries + " tries!");
18     }
19 }
20
21 }
22 }
```

Pada pemrograman kali ini juga menggunakan *while* yang digabungkan dengan *Random* yang berguna untuk menghasilkan angka yang acak dari 1 – 6 dan pemrograman ini pemrograman ini akan berhenti apa bila angka yang ditambahkan menghasilkan angka 7 seperti output di bawah ini.



```
2 + 2 = 4
1 + 5 = 6
3 + 6 = 9
3 + 5 = 8
4 + 6 = 10
1 + 3 = 4
1 + 3 = 4
4 + 4 = 8
5 + 1 = 6
1 + 5 = 6
1 + 2 = 3
5 + 3 = 8
5 + 3 = 8
5 + 5 = 10
6 + 4 = 10
2 + 6 = 8
4 + 5 = 9
5 + 2 = 7
You won after 18tries!
```

## 2.1.4 Game Penjumlahan

```

1 package pekan6_2511531014;
2 import java.util.Scanner;
3 import java.util.Random;
4 public class GamePenjumlahan_2511531014 {
5
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 point = 0;
11        int wrong = 0;
12        while (wrong < 3) {
13            int result = play(console, rand); //play one game
14            if (result > 0) {
15                point++;
16            } else {
17                wrong++;
18            }
19            System.out.println("you earned " + point + " total point.");
20        }
21        //membuat soal penjumlahan dan ditampilkan ke user
22        public static int play (Scanner console, Random rand) {
23            //print the operands being added, and sum them
24            int operands = rand.nextInt (4) + 2;
25            int sum = rand.nextInt(10) + 1;
26            System.out.print(sum);
27            for (int i = 2; i <= operands; i++) {
28                int n = rand.nextInt(10) + 1;
29                sum += n;
30                System.out.print(" + " + n);
31            }
32        }
33        //membuat soal penjumlahan dan ditampilkan ke user
34        public static int play (Scanner console, Random rand) {
35            //print the operands being added, and sum them
36            int operands = rand.nextInt (4) + 2;
37            int sum = rand.nextInt(10) + 1;
38            System.out.print(sum);
39            for (int i = 2; i <= operands; i++) {
40                int n = rand.nextInt(10) + 1;
41                sum += n;
42                System.out.print(" + " + n);
43            }
44            System.out.print(" = ");
45            //read user guess and report whether it was correct
46            int guess = console.nextInt();
47            if (guess == sum) {
48                return 1;
49            } else {
50                System.out.println("wrong! The answer was " + sum);
51                return 0;
52            }
53        }
54    }
55}

```

Pada pemrograman kali ini juga merupakan *while* yang digabungkan dengan *Scanner* dan *Random* yang berguna untuk permainan penjumlahan pada pemrograman di atas, *Random* yang berguna untuk mengacak angka dari 1-10 Dan *Scanner* berguna untuk memasukkan angka hasil dari penjumlahan yang diberikan dengan angka yang acak dan juga pada penjumlahan ini kita diberikan nyawa 3, jika salah memasukkan angka sampai 3x dari soal penjumlahan tersebut maka kita akan kalah dan terdapat poin dari hasil benar menjawab soal penjumlahan tersebut seperti output di bawah ini.

```
Console ×
<terminated> GamePenjumlahan_2511531014 [Java Application] C:\Program F
1 + 3 = 4
7 + 2 = 3
wrong! The answer was 9
9 + 4 + 5 + 3 + 8 = 4
wrong! The answer was 29
8 + 9 + 3 = 3
wrong! The answer was 20
you earned 1 total point.
```

### 2.1.5 ***DoWhile1***

```
membuatkue.java perulanganwh... LemparDadu_2... GamePenjuml... S
1 package pekan6_2511531014;
2 import java.util.Scanner;
3 public class doWhile1_2511531014 {
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        } while (!phrase.equals("abcd"));
12
13    }
14
15 }
16
```

Pada pemrograman kali ini berbeda dengan *while* di atas, kali ini ditambahkan *Do* yang berguna untuk selalu menjalankan minimal satu kali, walaupun kondisi awal salah. Seperti pada output di bawah ini.

```
Console ×
terminated> doWhile1_2511531014 [Java Application] C:\Program File
Input Password: 3uh
Input Password: ebwi
Input Password: abcd
```

## **BAB III**

### **PENUTUP**

#### **3.1. Kesimpulan**

Perulangan *while* dalam bahasa Java digunakan untuk menjalankan blok kode berulang kali selama kondisi bernilai *true*.

Struktur *while* sangat berguna ketika jumlah perulangan tidak diketahui sebelumnya. Namun, programmer harus berhati-hati dalam menulis kondisi agar tidak terjadi perulangan tanpa akhir.

Selain itu, pemahaman perbedaan antara *while* dan *do-while* penting untuk menentukan kapan masing-masing digunakan.

#### **3.2. Saran**

Pemahaman konsep dasar perulangan sangat penting bagi pemrogram Java pemula. Disarankan agar mahasiswa mempraktikkan berbagai contoh while dan do-while untuk memahami perilakunya secara mendalam. Dan diharapkan agar mahasiswa lebih teliti dalam memahami syntax agar meminimalisir kesalahan penulisan kode. Terutama saat menggunakan perulangan while, pastikan kondisi pengontrol dan nilai variabel benar, agar program tidak mengalami infinite loop.

## **DAFTAR PUSTAKA**

Sumber Daring (website):

- [1] Deitel, P. J., & Deitel, H. M. (2017). *Java: How to Program* (10th Edition). Pearson Education.
- [2] Oracle. (2024). *The while and do-while Statements (Java Documentation)*. Diakses dari: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/while.html>
- [3] Wahana Komputer. (2020). *Pemrograman Java untuk Pemula*. Yogyakarta: Andi Publisher.