

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN**  
**“laporan Praktikum Pekan 6”**

Disusun Oleh:

Faiz Fikri Satria

2511533026

Dosen Pengampu : Dr. Wahyudi, S.T, M.T.  
Asisten Praktikum : Rahmad Dwirizki Olders



DAPARTEMEN INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS ANDALAS  
2025

## KATA PENGANTAR

Puji syukur saya panjatkan ke hadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga dapat menyelesaikan laporan praktikum ini dengan baik. Tidak lupa juga saya ucapkan terima kasih kepada bapak Wahyudi. Dr. S.T.M.T sebagai dosen pembimbing dan Rahmad Dwirizki Olders yang telah membantu dalam pelaksanaan praktikum ini. Laporan ini saya susun untuk memenuhi tugas praktikum Algoritma dan Pemrograman dari praktikum pertemuan ke enam mengenai pemahaman dasar pemrograman bahasa Java menggunakan IDE Eclipse. Praktikum ini bertema pembuatan kode Do While. Saya berharap laporan ini dapat memberikan manfaat bagi pembaca dalam memahami konsep dasar pemrograman Java.

Laporan ini dibuat dengan harapan dapat memberikan pemahaman dasar mengenai penggunaan Bahasa pemrograman Java, khususnya mengenai hal struktur dasar program dan fungsi.

Saya menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu, kritik dan saran sangat diharapkan demi penyempurnaan laporan di masa mendatang.

Padang 08 November, 2025

Penulis

## DAFTAR ISI

KATA PENGANTAR .....	ii
DAFTAR ISI .....	iii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Tujuan .....	1
1.3 Manfaat .....	1
BAB II PEMBAHASAN .....	2
2.1 <b>Do-While untuk Validasi Password</b> .....	2
2.1.1 Langkah-langkah .....	2
2.1.2 Contoh Pemrograman .....	3
2.1.3 Hasil Output .....	3
2.1.4 Analisis .....	3
2.1.5 Teori Mengenai Kode .....	3
2.2 <b>While untuk Game Penjumlahan</b> .....	4
2.2.1 Langkah-langkah .....	4
2.2.2 Contoh Pemrograman .....	5
2.2.3 Hasil Output .....	5
2.2.4 Analisis .....	4
2.2.5 Teori Mengenai Kode .....	5
2.3 <b>While untuk Simulasi Lempar Dadu</b> .....	6
2.2.1 Langkah-langkah .....	6

2.3.2 Contoh Pemrograman .....	7
2.3.3 Hasil Output .....	7
2.3.4 Analisis .....	7
2.3.5 Teori Mengenai Kode .....	7
<b>2.4 While untuk Perulangan dengan Konfirmasi .....</b>	<b>7</b>
2.4.1 Langkah-langkah .....	8
2.4.2 Contoh Pemrograman .....	8
2.4.3 Hasil Output .....	9
2.4.4 Analisis .....	9
2.4.5 Teori Mengenai Kode .....	9
<b>2.5 While untuk Sentinel Loop Penjumlahan.....</b>	<b>9</b>
2.5.1 Langkah-langkah .....	9
2.5.2 Contoh Pemrograman .....	10
2.5.3 Hasil Output .....	10
2.5.4 Analisis .....	10
2.5.5 Teori Mengenai Kode .....	10
<b>BAB III KESIMPULAN .....</b>	<b>11</b>
<b>DAFTAR PUSTAKA .....</b>	<b>12</b>

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Tujuan**

1. Tujuan dari praktikum ini adalah:
2. Memahami dan menerapkan konsep do-while untuk perulangan yang dieksekusi minimal sekali.
3. Menguasai penggunaan while loop untuk kondisi yang tidak diketahui jumlah iterasinya, termasuk dengan elemen random atau input.
4. Melatih kemampuan dalam menulis, menjalankan, dan menganalisis kode Java menggunakan Eclipse.
5. Mengidentifikasi aplikasi praktis perulangan seperti validasi password, game sederhana, dan akumulasi data.

#### **1.2 Manfaat**

6. Manfaat dari praktikum ini meliputi:
7. Meningkatkan kemampuan logika pemrograman mahasiswa melalui latihan perulangan interaktif dan random.
8. Memberikan pengalaman praktis dalam menggunakan Eclipse, yang berguna untuk proyek pemrograman di masa depan.
9. Membantu mahasiswa dalam memahami aplikasi nyata dari struktur kontrol seperti do-while dan while, misalnya untuk validasi input atau simulasi permainan.
10. Sebagai bahan referensi untuk pengembangan kemampuan troubleshooting kode, terutama pada kasus loop tak terbatas atau handling input.

## BAB II

### PEMBAHASAN

#### 2.1 Do-While untuk Validasi Password

Kode ini merupakan contoh penggunaan do-while loop dalam Java untuk memvalidasi input password dari pengguna hingga input sesuai dengan nilai yang ditentukan ("abcd"). Konsep ini berguna untuk memastikan eksekusi minimal satu kali sebelum memeriksa kondisi, yang ideal untuk skenario validasi input interaktif. Dengan import Scanner, kode ini melatih kemampuan handling input konsol dan perbandingan string menggunakan equals(). Secara keseluruhan, ini memperkenalkan aplikasi do-while dalam program keamanan sederhana, di mana loop terus berjalan hingga kondisi terpenuhi, membantu memahami perbedaan dengan while loop standar.

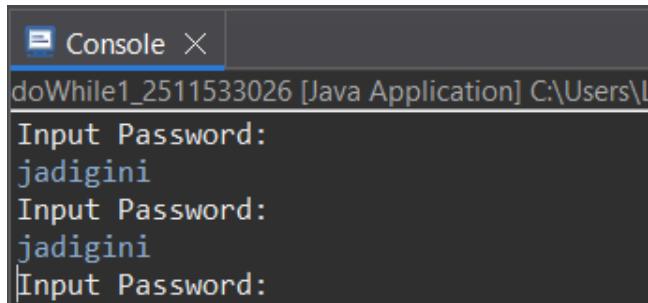
##### 2.1.1 Langkah-langkah

1. Buka Eclipse dan buat project baru dengan package  
Pekan\_enam\_2511533026.
2. Buat class doWhile1\_2511533026 dengan import Scanner.
3. Inisialisasi Scanner dan variabel phrase.
4. Gunakan do-while: cetak prompt, baca input, loop hingga phrase equals "abcd".
5. Jalankan program dan masukkan input untuk testing.

## 2.1.2 Contoh Pemrograman

```
3 import java.util.Scanner;
4
5 public class doWhile1_2511533026 {
6     public static void main(String[] args) {
7         Scanner console = new Scanner (System.in);
8         String phrase;
9         do {
10             System.out.println("Input Password: ");
11             phrase = console.next();
12         }while (!phrase.equals("abcd"));
13     }
14 }
```

## 2.1.3 Hasil Output



The screenshot shows a Java application window titled "doWhile1\_2511533026 [Java Application] C:\Users\...". The console tab is active, displaying the following text:

```
Input Password:  
jadigini  
Input Password:  
jadigini  
Input Password:
```

## 2.1.4 Analisis

Kode menampilkan prompt hingga password benar. Loop dieksekusi minimal sekali, cocok untuk validasi. Tidak ada error jika input string, tapi case-sensitive. Analisis: Efisien untuk input berulang, tapi bisa infinite jika tidak pernah benar.

## 2.1.5 Teori Mengenai Kode

Do-while loop mengeksekusi body terlebih dahulu sebelum cek kondisi, berbeda dengan while yang cek dulu. Teori: Berguna saat minimal satu eksekusi diperlukan. Di Java, equals() untuk banding string, bukan ==.

## **2.2 While untuk Game Penjumlahan**

Kode ini menunjukkan penggunaan while loop untuk game penjumlahan random, di mana pemain menjawab soal hingga 3 kali salah, dengan poin dihitung untuk jawaban benar. Dengan import Random dan Scanner, ini melatih integrasi elemen acak dan input pengguna dalam loop. While loop mengontrol permainan utama, sementara method play menggunakan for untuk generate soal. Secara keseluruhan, ini memperkuat konsep while untuk kondisi tak terbatas iterasi, berguna dalam simulasi game edukasi, dan menekankan modularitas kode dengan method terpisah.

### **2.2.1 Langkah-langkah**

1. Buat package Pekan\_enam\_2511533026 dan class GamePenjumlahan\_2511533026.
2. Import Random dan Scanner.
3. Inisialisasi variabel points, wrong, dan loop while hingga wrong < 3.
4. Buat method play untuk generate soal random dan cek jawaban.
5. Jalankan dan masukkan jawaban untuk simulasi game.

## 2.2.2 Contoh Pemrograman

```
3❶ import java.util.Random;❷
5
6 public class GamePenjumlahan_2511533026 {
7❶   public static void main (String[] args) {
8     Scanner console = new Scanner (System.in);
9     Random rand = new Random ();
10    //play until user gets 3 wrong
11    int points = 0;
12    int wrong = 0;
13❶    while (wrong < 3) {
14      int result = play(console, rand); //play one game
15❶      if (result > 0) {
16        points++;
17❶      }else {
18        wrong++;
19      }
20    }
21    System.out.println("You eaned " + points + " total points.");
22  }
23  //membuat soal penjumlahan dan ditampilkan ke user
24❶  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);

30❶    for (int i = 2; i <= operands; i++) {
31      int n = rand.nextInt(10) + 1;
32      sum += n;
33      System.out.print(" + " + n);
34    }
35    System.out.print(" = ");
36    // read user's guess and report whether it was correct
37    int guess = console.nextInt();
38❶    if (guess == sum) {
39      return 1;
40    } else {
41      System.out.println("Wrong! The answer was " + sum);
42      return 0;
43    }
44  }
45
46  }
47 }
```

## 2.2.3 Hasil Output



The screenshot shows a Java application window titled "Console". The title bar also displays the application name "GamePenjumlahan\_2511533026 [Java Application]" and the path "C:\Users\Lenovo\p2\pool\p". The console output is as follows:

```
4 + 7 + 5 = 2
Wrong! The answer was 16
7 + 3 + 4 + 6 + 1 = 9
Wrong! The answer was 21
3 + 8 + 6 + 3 = 20
1 + 10 + 3 + 10 = 24
3 + 5 + 8 =
```

## **2.2.4 Analisis**

Kode menghasilkan game interaktif dengan random soal. While loop berhenti setelah 3 salah. Analisis: Random membuat variasi, tapi tergantung seed. Method play membuat kode modular.

## **2.2.5 Teori Mengenai Kode**

While loop dieksekusi selama kondisi true, cocok untuk iterasi tak diketahui. Teori: Kombinasi dengan Random untuk simulasi, dan Scanner untuk input. Modularitas dengan method meningkatkan readability.

## **2.3 While untuk Simulasi Lempar Dadu**

Kode ini adalah implementasi while loop untuk simulasi lempar dadu hingga jumlah mata dadu mencapai 7, dengan hitungan percobaan. Dengan import Random, ini melatih penggunaan elemen acak dalam perulangan. While loop mengontrol roll dadu berulang hingga kondisi terpenuhi. Secara keseluruhan, ini membantu memahami aplikasi while dalam simulasi probabilitas, di mana jumlah iterasi tidak pasti, dan menekankan pentingnya variabel counter untuk tracking.

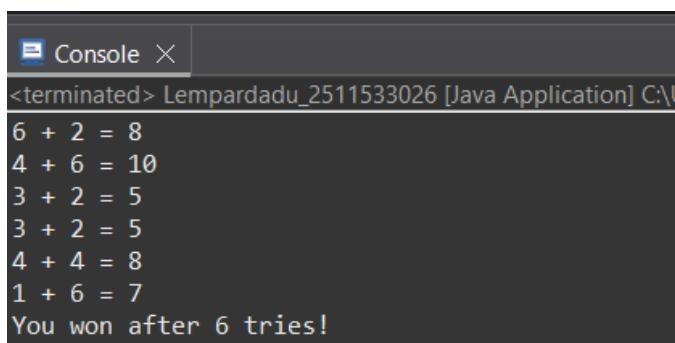
### **2.3.1 Langkah-langkah**

1. Buat package Pekan\_enam\_2511533026 dan class  
`Lempardadu_2511533026`
2. Import Random.
3. Inisialisasi tries, sum, dan while hingga `sum != 7`.
4. Generate dadu1 dan dadu2 random, hitung sum, increment tries.
5. Jalankan untuk lihat simulasi.

### 2.3.2 Contoh Pemrograman

```
3 import java.util.Random;
4
5 public class Lempardadu_2511533026 {
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 }
```

### 2.3.3 Hasil Output



The screenshot shows a Java application window titled 'Console'. The title bar also includes '<terminated> Lempardadu\_2511533026 [Java Application] C:\U'. The console output displays a series of dice rolls and their sums until a total of 7 is reached, followed by a success message. The output is as follows:

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

### 2.3.4 Analisis

Kode mensimulasikan lempar dadu hingga 7. While loop efisien untuk kondisi acak. Analisis: Probabilitas 7 adalah 1/6, jadi tries variatif. Tidak ada error.

### 2.3.5 Teori Mengenai Kode

While loop untuk iterasi hingga kondisi false. Teori: Random.nextInt() untuk nilai acak, berguna simulasi. Counter tries untuk analisis performa loop.

## 2.4 While untuk Perulangan dengan Konfirmasi

Kode ini menunjukkan while loop dengan boolean flag untuk perulangan counter, di mana pengguna ditanya untuk lanjut atau tidak hingga jawaban "tidak". Dengan import Scanner, ini melatih handling input string dan ignore case. While loop bergantung pada flag running. Secara keseluruhan, ini memperkuat

pemahaman while untuk loop interaktif, berguna dalam menu program, dan menekankan penggunaan equalsIgnoreCase() untuk fleksibilitas input.

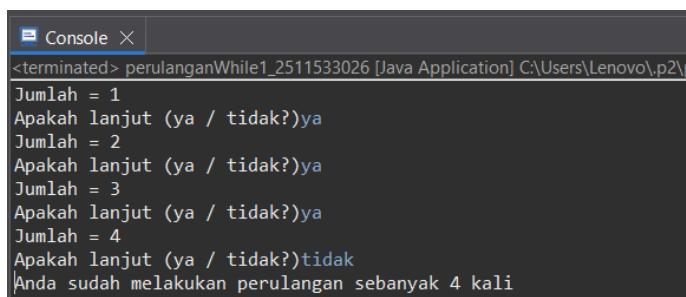
### 2.4.1 Langkah-langkah

1. Buat package Pekan\_enam\_2511533026 dan class perulanganWhile1\_2511533026.
2. Import Scanner.
3. Inisialisasi counter, jawab, running=true.
4. While running: increment counter, cetak, tanya jawab, if "tidak" set running false.
5. Jalankan dan masukkan input.

### 2.4.2 Contoh Pemrograman

```
1 package Pekan_enam_2511533026;
2
3 import java.util.Scanner;
4
5 public class perulanganWhile1_2511533026 {
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 }
```

### 2.4.3 Hasil Output



The screenshot shows a terminal window titled "Console". The output of the program is displayed, starting with the message "Jumlah = 1". It then asks the user "Apakah lanjut (ya / tidak?)". The user responds with "ya". This process repeats four times, with the counter increasing by one each time. After the fourth iteration, when the user types "tidak", the loop exits. The final message printed is "Anda sudah melakukan perulangan sebanyak 4 kali".

```
Console ×
<terminated> perulanganWhile1_2511533026 [Java Application] C:\Users\Lenovo\p2\
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?)tidak
Anda sudah melakukan perulangan sebanyak 4 kali
```

### **2.4.3 Hasil Output**

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

### **2.4.4 Analisis**

Kode menghitung perulangan hingga user berhenti. While dengan flag fleksibel. Analisis: equalsIgnoreCase() tangani variasi input. Potensi infinite jika input salah terus.

### **2.4.5 Teori Mengenai Kode**

While loop dengan flag boolean untuk kontrol manual. Teori: Scanner.nextLine() untuk input baris penuh. equalsIgnoreCase() untuk banding case-insensitive.

## **2.5 While untuk Sentinel Loop Penjumlahan**

Kode ini merupakan while loop dengan sentinel value (0) untuk menjumlahkan angka input hingga 0 dimasukkan, sebagai cara keluar loop. Dengan import Scanner, ini melatih konsep sentinel untuk terminasi loop. Inisialisasi number dengan dummy value. Secara keseluruhan, ini membantu memahami while untuk pengolahan data tak terbatas, berguna dalam program akumulasi seperti kalkulator, meski disebut for di deskripsi tapi sebenarnya while.

### **2.5.1 Langkah-langkah**

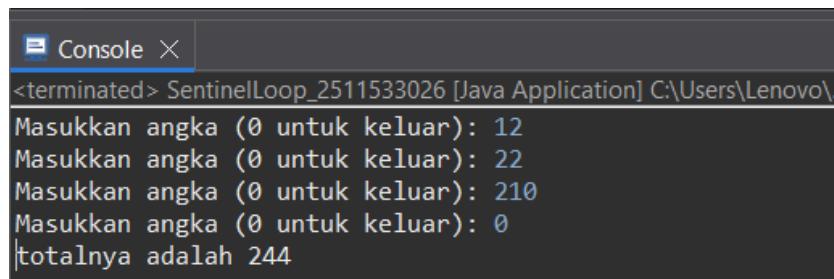
1. Buat package Pekan\_enam\_2511533026 dan class SentinelLoop\_2511533026.
2. Import Scanner.
3. Inisialisasi sum=0, number=12 (dummy).
4. While number !=0: prompt input, tambah ke sum.

5. Cetak total setelah loop.

### 2.5.2 Contoh Pemrograman

```
1 package Pekan_enam_2511533026;
2
3 import java.util.Scanner;
4
5 public class SentinelLoop_2511533026 {
6     public static void main (String[] args) {
7         Scanner console = new Scanner (System.in);
8         int sum = 0;
9         int number = 12; // "dummy value", anything but 0
10
11        while (number !=0) {
12            System.out.print("Masukkan angka (0 untuk keluar): ");
13            number = console.nextInt();
14            sum = sum + number;
15        }
16        System.out.println("totalnya adalah " + sum);
17    }
18 }
```

### 2.5.3 Hasil Output



The screenshot shows a terminal window titled 'Console'. The application has been terminated and is shown as 'SentinelLoop\_2511533026 [Java Application] C:\Users\Lenovo\'. The console output displays the following interaction:

```
Masukkan angka (0 untuk keluar): 12
Masukkan angka (0 untuk keluar): 22
Masukkan angka (0 untuk keluar): 210
Masukkan angka (0 untuk keluar): 0
|totalnya adalah 244
```

### 2.5.4 Analisis

Kode menjumlahkan hingga sentinel 0. While cocok untuk input tak terbatas. Analisis: Dummy value memastikan loop masuk. Tidak handle non-integer input.

### 2.5.5 Teori Mengenai Kode

While dengan sentinel untuk terminasi berdasarkan nilai khusus. Teori: Berguna saat jumlah iterasi unknown. Di Java, nextInt() untuk input integer.

## **BAB III**

### **KESIMPULAN**

Praktikum pertemuan kedua ini berhasil memberikan pemahaman dasar Praktikum pertemuan kedua ini berhasil memberikan pemahaman dasar mengenai do-while dan while loop dalam bahasa Java menggunakan Eclipse. Melalui kode-kode sederhana, mahasiswa dapat melihat bagaimana do-while menjamin eksekusi minimal satu kali, sementara while cocok untuk kondisi yang bergantung pada input atau random. Aplikasi seperti validasi password, game, dan simulasi memperkuat logika perulangan. Secara keseluruhan, praktikum ini membangun fondasi untuk program interaktif.

Dari analisis, terlihat bahwa kesalahan seperti input tak sesuai dapat menyebabkan loop tak berujung, sehingga penting menggunakan kondisi yang tepat dan handling error. Manfaat praktikum mencakup keterampilan modularitas kode dan penggunaan library seperti Random dan Scanner. Penggunaan package meningkatkan organisasi proyek.

Akhirnya, praktikum ini mendorong eksperimen lebih lanjut, seperti menambahkan fitur error handling atau variasi loop. Pemahaman ini dapat diterapkan pada aplikasi nyata seperti software edukasi. Saran: Tambahkan elemen GUI untuk interaksi lebih baik di praktikum mendatang.

## **DAFTAR PUSTAKA**

1. Schildt, Herbert. (2019). *Java: The Complete Reference*. McGraw-Hill Education.
2. Oracle. (2023). *Java Documentation*.
3. Deitel, Paul & Deitel, Harvey. (2020). *Java How to Program*. Pearson.
4. Tutorialspoint. (2024). *Java Loops Tutorial*.