

# LAPORAN PRATIUM PEMROGRAMAN ALGORITMA DAN PEMROGRAMAN

Perulangan While

Disusun oleh

Nama : Muhammad Yusuf Insy

NIM : 2511532003

Dosen Pengampu : DR. Wahyudi,S.T,M.T.

Asisten Praktikum : Aufan Taufikurrahman



DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS TAHUN 2025

## Daftar Isi

## Contents

BAB I PENDAHULUAN.....	3
1.1 Latar belakang.....	3
1.2 Tujuan praktikum .....	3
1.3     Manfaat praktikum.....	3
BAB II PEMBAHASAN.....	4
BAB III KESIMPULAN .....	11
DAFTAR PUSTAKA .....	12

## BAB I PENDAHULUAN

### 1.1 Latar belakang

Dalam pemrograman, seringkali diperlukan untuk menjalankan serangkaian instruksi secara berulang berdasarkan kondisi tertentu. Struktur kontrol ini dikenal sebagai perulangan (*looping*). Perulangan adalah konsep fundamental yang memungkinkan program untuk mengotomatisasi tugas-tugas berulang, mengurangi redundansi kode, dan meningkatkan efisiensi.

Perulangan while merupakan salah satu jenis struktur perulangan *pre-test* (uji kondisi di awal) yang paling dasar dalam bahasa pemrograman Java.

Perulangan while akan mengevaluasi kondisi logis di awal blok kode. Perulangan hanya akan dijalankan jika kondisi tersebut bernilai benar (*true*). Jika kondisi awal sudah bernilai salah (*false*), badan perulangan tidak akan dieksekusi sama sekali.

### 1.2 Tujuan praktikum

- A. Memahami Mekanisme Perulangan while
- B. Membedakan while dan do-while
- C. Mengenal Infinite Loop

### 1.3 Manfaat praktikum

- A. Penguasaan Logika Kontrol Program
- B. Peningkatan Efisiensi Kode (*Code Efficiency*)
- C. Kemampuan Mengolah Input Dinamis

## BAB II PEMBAHASAN

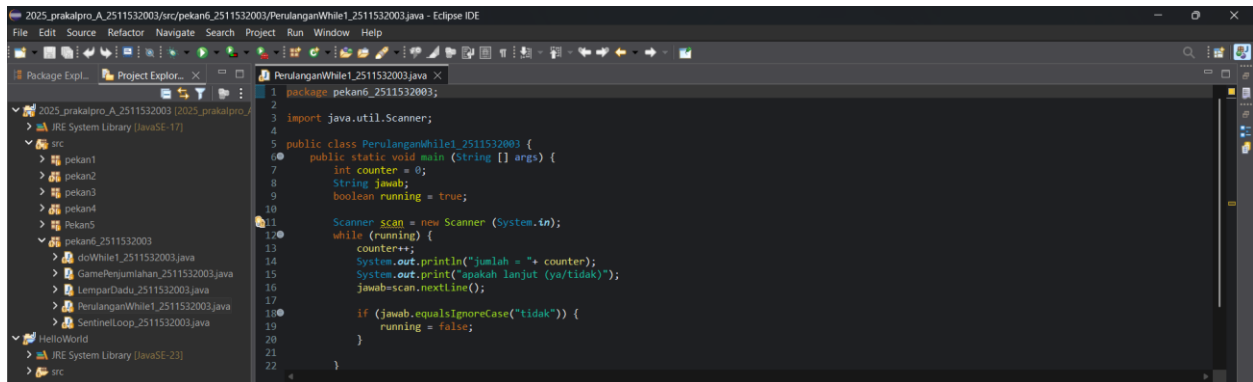
### 2.1 Langkah Kerja

1. Membuka IDE dan membuat project baru bernama pekan5.
2. Membuat kelas Java: PerulanganWhile1, LemparDadu, GamePenjumlahan ,SentineLoop, PerulanganDoWhile.
3. Menuliskan kode program sesuai instruksi praktikum.
4. Menjalankan program dengan beberapa input uji.
5. Mencatat hasil dan membuat analisis.

## 2.2 Uraian Kode Program

### a. Kode Program PerulanganWhile1

Kode Program :



```
1 package pekan6_2511532003;
2
3 import java.util.Scanner;
4
5 public class PerulanganWhile1_2511532003 {
6     public static void main (String [] args) {
7         int counter = 0;
8         String jawab;
9         boolean running = true;
10
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
18             if (jawab.equalsIgnoreCase("tidak")) {
19                 running = false;
20             }
21         }
22     }
```

Uraian :

- running diinisialisasi sebagai true.
- Loop berjalan selama running bernilai true.
- Di dalam loop, program menampilkan hitungan saat ini (jumlah = counter) dan menanyakan: "apakah lanjut (ya/tidak)".
- Jika pengguna memasukkan "tidak" (tidak peduli huruf besar/kecil karena menggunakan equalsIgnoreCase), maka running diubah menjadi false, dan perulangan pun berhenti.

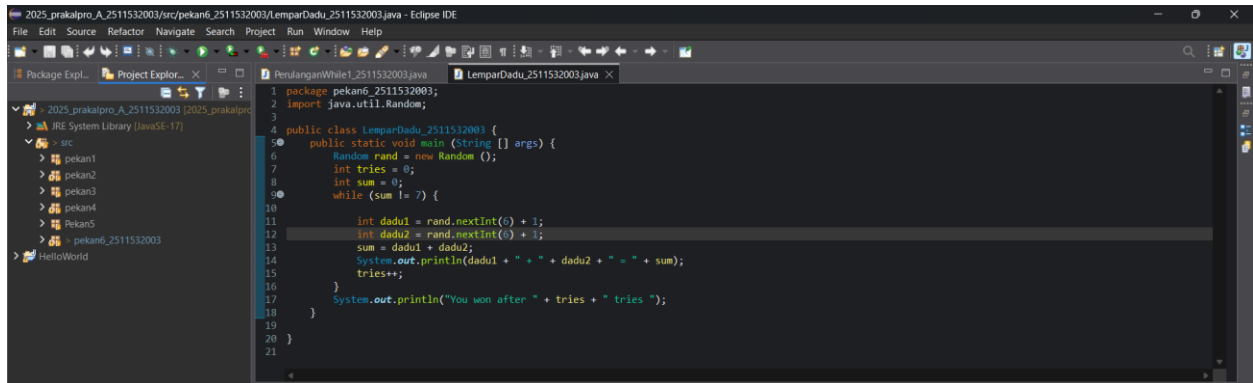
Output :



```
terminated- PerulanganWhile1_2511532003 (Java Application) C:\Users\USER\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_23.0.2.v20250131-0604\jre\bin\javaw.exe (Nov 5, 2025, 7:54:19 PM - 7:54:29 PM)
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
```

## b. Kode Program LemparDadu

Kode Program :

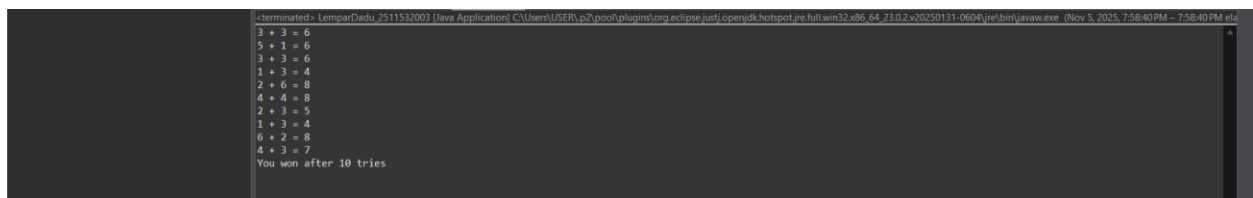


```
1 package pekan6_2511532003;
2 import java.util.Random;
3
4 public class LemparDadu_2511532003 {
5     public static void main (String [] args) {
6         Random rand = new Random ();
7         int tries = 0;
8         int sum = 0;
9         while (sum != 7) {
10
11             int dadu1 = rand.nextInt(6) + 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 }
```

Uraian :

- `Random rand = new Random();` digunakan untuk menghasilkan angka dadu.
- `dadu1 = rand.nextInt(6) + 1;` menghasilkan angka 1 sampai 6.
- Loop berjalan selama `sum != 7`.
- Di setiap putaran, dua dadu dilempar, jumlahnya (sum) dihitung, dan percobaan (tries) ditambah.
- Perulangan berhenti saat `dadu1 + dadu2 == 7`, dan program mencetak jumlah percobaan yang dibutuhkan.

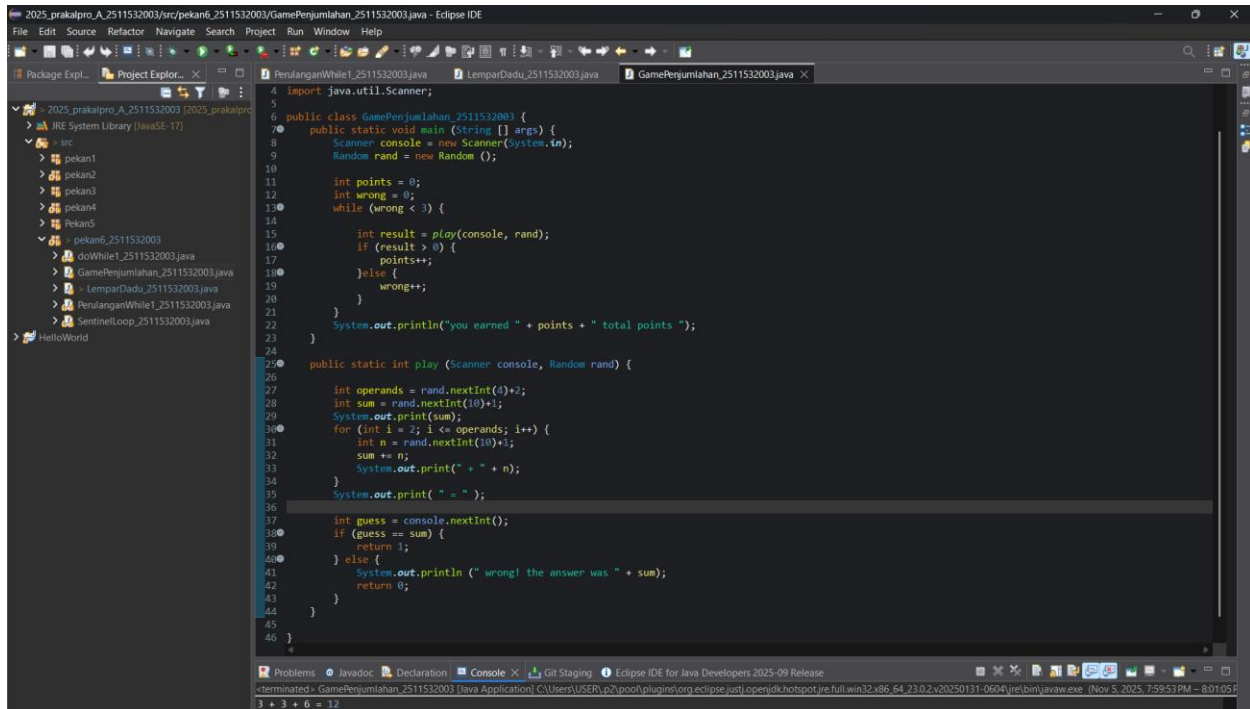
Output :



```
terminated> LemparDadu_2511532003 [Java Application] C:\Users\USER\p2\poo\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.23.0.2.v20250131-0604\jre\bin\javaw.exe (Nov 5, 2025, 7:58:40 PM - 7:58:40 PM etc)
3 + 3 = 6
5 + 1 = 6
3 + 3 = 6
1 + 3 = 4
2 + 6 = 8
4 + 4 = 8
2 + 3 = 5
1 + 3 = 4
6 + 2 = 8
4 + 3 = 7
You won after 10 tries
```

### c. Kode Program GamePenjumlahan

Kode Program :



```
4 import java.util.Scanner;
5
6 public class GamePenjumlahan_2511532003 {
7     public static void main (String [] args) {
8         Scanner console = new Scanner (System.in);
9         Random rand = new Random ();
10
11         int points = 0;
12         int wrong = 0;
13         while (wrong < 3) {
14
15             int result = play(console, rand);
16             if (result > 0) {
17                 points++;
18             } else {
19                 wrong++;
20             }
21             System.out.println("you earned " + points + " total points ");
22         }
23     }
24
25     public static int play (Scanner console, Random rand) {
26
27         int operands = rand.nextInt(4)+2;
28         int sum = rand.nextInt(10)+1;
29         System.out.print(sum);
30         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
37         int guess = console.nextInt();
38         if (guess == sum) {
39             return 1;
40         } else {
41             System.out.println (" wrong! the answer was " + sum);
42             return 0;
43         }
44     }
45 }
46 }
```

Uraian :

- Permainan terus berlanjut selama jumlah kesalahan (wrong) **kurang dari 3**.
- Metode play menghasilkan soal:
  - Menentukan jumlah angka yang akan dijumlahkan (operands antara 2 hingga 6).
  - Menggunakan perulangan **for** di dalam play untuk menghasilkan angka-angka acak dan menghitung totalnya (sum).
  - Pengguna memasukkan tebakan.
- Jika tebakan benar, points bertambah. Jika salah, wrong bertambah, dan jawaban yang benar diungkapkan.
- Permainan berakhir ketika pengguna membuat 3 kesalahan, dan program mencetak total poin yang didapatkan.

Output :

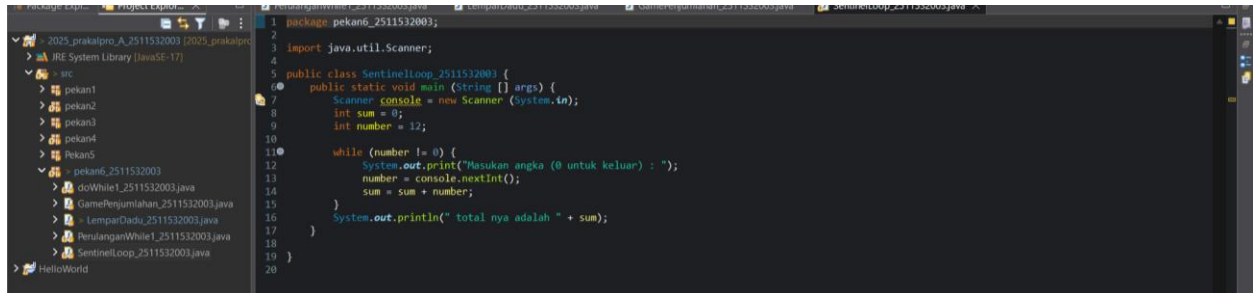
```
terminated> GamePengumtahan_2511532003 [Java Application] C:\Users\USER\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.23.0.2.x20250131-0604\jre\bin\javaw.exe - Nov 5, 2025, 7:59:51 PM - 8010511
> pekan1
> pekan2
> pekan3
> pekan4
> pekan5
  > pekan6_2511532003
    > doWhile1_2511532003.java
    > GamePengumtahan_2511532003.java
    > LempatDadu_2511532003.java
    > PerulanganWhile1_2511532003.java
    > SentinelLoop_2511532003.java
> HelloWorld

3 + 3 + 6 = 12
3 + 10 + 1 = 14
6 + 4 = 10
7 + 6 + 2 = 15
6 + 10 + 8 + 3 = 15
wrong! the answer was 27
5 + 10 + 5 + 1 + 1 = 22
2 + 9 + 8 + 5 = 24
8 + 10 = 18
9 + 8 + 9 = 26
8 + 9 + 1 + 8 + 3 = 29
9 + 5 + 3 + 8 = 25
3 + 8 = 1
wrong! the answer was 11
6 + 3 = 1
wrong! the answer was 9
you earned 10 total points
```



#### d. Kode Program SentinelLoop

Kode Program :

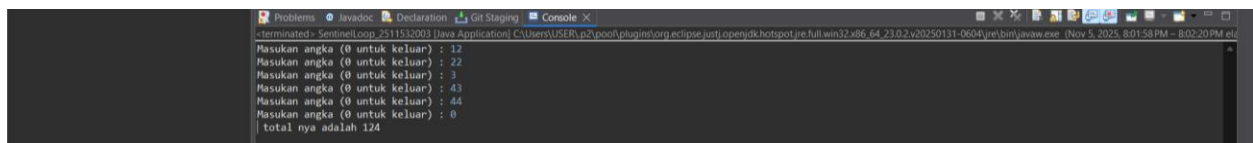


```
1 package pekan6_2511532003;
2
3 import java.util.Scanner;
4
5 public class SentinelLoop_2511532003 {
6     public static void main (String [] args) {
7         Scanner console = new Scanner (System.in);
8         int sum = 0;
9         int number = 12;
10
11         while (number != 0) {
12             System.out.print("Masukan angka (0 untuk keluar) : ");
13             number = console.nextInt();
14             sum = sum + number;
15         }
16         System.out.println(" total nya adalah " + sum);
17     }
18 }
19
20 }
```

Uraian :

- int number = 1; diinisialisasi agar perulangan while bisa dimulai.
- Loop berjalan selama number != 0.
- Program meminta pengguna memasukkan angka, dengan instruksi: "**(0 untuk keluar)**".
- Angka **0** adalah nilai **sentinel** (penghenti).
- Setiap angka yang dimasukkan (selain 0) akan ditambahkan ke variabel sum.
- Setelah 0 dimasukkan, perulangan berhenti, dan program mencetak **total penjumlahan** seluruh angka.

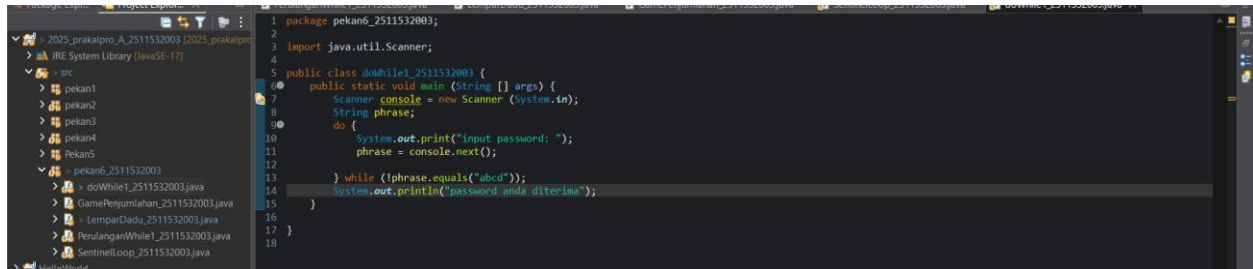
Output :



```
terminated: SentinelLoop_2511532003 [Java Application] C:\Users\USER\p2\pool\plugin\org.eclipse.jdt.ui\openjdk.hotspot.jre.full.win32.x86_64.23.0.2.v20250131-0604\jre\bin\java.exe (Nov 5, 2025, 8:01:58 PM - 8:02:20 PM EDT)
Masukan angka (0 untuk keluar) : 12
Masukan angka (0 untuk keluar) : 22
Masukan angka (0 untuk keluar) : 3
Masukan angka (0 untuk keluar) : 43
Masukan angka (0 untuk keluar) : 44
Masukan angka (0 untuk keluar) : 0
total nya adalah 124
```

### e. Kode Program DoWhile 1

Kode Program :

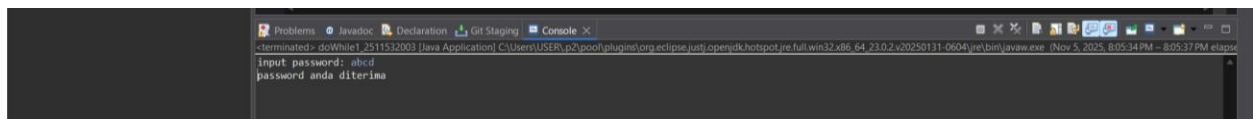


```
1 package pekan6_2511532003;
2
3 import java.util.Scanner;
4
5 public class doWhile1_2511532003 {
6     public static void main (String [] args) {
7         Scanner console = new Scanner (System.in);
8         String phrase;
9         do {
10             System.out.print("Input password: ");
11             phrase = console.next();
12         } while (!phrase.equals("abcd"));
13         System.out.println("password anda diterima");
14     }
15 }
16
17
18
```

Uraian :

- do-while menjamin bahwa blok kode akan dieksekusi minimal satu kali sebelum kondisi dicek.
- do: Program meminta dan menerima input password dari pengguna.
- while (phrase.equals("abcd")): Perulangan akan terus berjalan selama input yang dimasukkan sama persis dengan "abcd" (karena menggunakan equals() yang case-sensitive).
- Catatan Logika: Logika ini kebalikan dari validasi password pada umumnya. Dalam kasus ini, jika input benar ("abcd"), program mengulang (do lagi). Jika input salah (selain "abcd"), program berhenti dan mencetak "password anda diterima".

Output :



```
terminated> doWhile1_2511532003 [Java Application] C:\Users\US4R\62\poo\plugins\org.eclipse.jdt.openide.hotspot.jre.full.win32.x86_64.23.0.2.v20230131-0604\jre\bin\java.exe (Nov 5, 2025, 8:05:34 PM) elapsed
Input password: abcd
password anda diterima
```

### BAB III KESIMPULAN

Perulangan *while* merupakan salah satu struktur kontrol paling esensial dalam pemrograman Java, berfungsi untuk mengotomatisasi serangkaian instruksi secara efisien. Praktikum ini menyimpulkan bahwa perulangan *while* beroperasi sebagai mekanisme *pre-test*, di mana kondisi diuji sebelum blok kode dieksekusi, memastikan bahwa pengulangan hanya terjadi selama kondisi tersebut bernilai benar. Keberhasilan implementasi *while* mutlak bergantung pada pembaruan variabel kontrol atau *flag* di dalam *loop body* untuk menjamin terminasi yang terencana dan menghindari *infinite loop*. Lebih lanjut, materi ini mencakup variasi penting seperti perulangan *Sentinel* yang diakhiri oleh nilai khusus (0), perulangan yang dikontrol *flag* untuk interaksi pengguna, dan perulangan *do-while* (*post-test*) yang sangat berguna untuk memastikan blok kode dijalankan minimal sekali, khususnya dalam validasi input. Secara keseluruhan, penguasaan perulangan *while* memungkinkan mahasiswa untuk menciptakan program yang lebih ringkas, efisien, dan dinamis, baik untuk simulasi sederhana (seperti lempar dadu) maupun untuk aplikasi interaktif yang membutuhkan pengolahan input data berulang.

## DAFTAR PUSTAKA

Garrison, T. M. (Tahun Publikasi Terbaru). Java Programming: Logic and Design (Edisi ke-X).

Deitel, P. J., & Deitel, H. M. (Tahun Publikasi Terbaru). Java How to Program: Early Objects (Edisi ke-X). Pearson Education.

Oracle. (Tahun Publikasi Terbaru). The while and do-while Statements.

Oracle. (Tahun Publikasi Terbaru). Class Scanner.

Oracle. (Tahun Publikasi Terbaru). Class Random.

