

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN**  
**PERULANGAN WHILE DAN DO-WHILE**



**Disusun oleh:**

Ghinada Fathanawafa Algma  
2511533008

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

**Asisten Praktikum:**  
Rahmad Dwirizki Olders

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

**2025**

## KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya, laporan praktikum dengan judul “*Perulangan While dan Do-While pada Bahasa Pemrograman Java*” dapat diselesaikan dengan baik dan tepat waktu.

Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban kegiatan praktikum algoritma dan pemrograman, khususnya dalam memahami konsep dasar perulangan (*looping*) menggunakan struktur *while* dan *do-while*. Melalui praktikum ini, penulis berusaha mempelajari bagaimana kedua jenis perulangan tersebut bekerja, perbedaan mendasar di antara keduanya, serta penerapannya dalam menyelesaikan berbagai permasalahan komputasi.

Penyusunan laporan ini tidak terlepas dari bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada: Dosen pengampu mata kuliah yang telah memberikan arahan dan materi, Asisten praktikum yang telah membimbing selama pelaksanaan, Rekan-rekan praktikan yang turut memberikan masukan dan semangat.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat, baik bagi penulis sendiri maupun bagi pembaca yang ingin memahami lebih dalam mengenai perulangan *while* dan *do-while* pada Java.

Padang, November 2025

Penulis

## DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI .....	ii
BAB I PENDAHULUAN .....	1
1.1    Latar Belakang .....	1
1.2    Tujuan.....	2
1.3    Manfaat .....	2
BAB II PEMBAHASAN .....	3
2.1 Dasar Teori .....	3
2.1.1 Konsep Perulangan.....	3
2.1.2 Perulangan <i>While</i> .....	3
2.1.3 Perulangan <i>Do-While</i> .....	4
2.2 Langkah Kerja.....	4
2.2.1 Membuat <i>Package</i> dan <i>Class</i> .....	4
2.2.2 Program 1 : Perulangan <i>While</i> 1 .....	7
2.2.3 Program 2 : Lempar Dadu.....	8
2.2.4 Program 3 : Game Penjumlahan .....	9
2.2.5 Program 4 : Sentinel loop.....	10
2.2.6 Program 5 : Do-While 1 .....	11
BAB III PENUTUP .....	13
3.1 Kesimpulan .....	13
DAFTAR PUSTAKA .....	14

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Dalam pemrograman, perulangan (*looping*) merupakan salah satu konsep dasar yang sangat penting karena memungkinkan suatu instruksi dijalankan secara berulang tanpa harus dituliskan berkali-kali. Dengan adanya perulangan, efisiensi kode dapat tercapai, baik dari segi penulisan maupun eksekusi program.

Bahasa pemrograman Java menyediakan beberapa struktur perulangan, di antaranya *while* dan *do-while*. Kedua jenis perulangan ini memiliki fungsi yang serupa, yaitu mengeksekusi blok kode secara berulang selama kondisi tertentu terpenuhi. Namun, terdapat perbedaan mendasar dalam cara keduanya mengevaluasi kondisi:

- *While loop* akan mengecek kondisi terlebih dahulu sebelum menjalankan perintah.
- *Do-while loop* akan menjalankan perintah minimal satu kali terlebih dahulu, kemudian baru melakukan pengecekan kondisi.

Pemahaman mengenai perbedaan dan penerapan kedua jenis perulangan ini sangat penting bagi mahasiswa atau praktikan yang sedang mempelajari dasar-dasar pemrograman. Melalui praktikum ini, diharapkan peserta mampu:

- Memahami konsep dasar perulangan *while* dan *do-while*.
- Mengimplementasikan perulangan tersebut dalam program sederhana.
- Menganalisis kelebihan dan kekurangan masing-masing struktur perulangan dalam menyelesaikan masalah komputasi.

Dengan demikian, praktikum ini tidak hanya bertujuan untuk melatih keterampilan teknis dalam menulis kode, tetapi juga untuk menumbuhkan pola pikir logis,

sistematis, dan efisien dalam menyelesaikan permasalahan menggunakan bahasa pemrograman Java.

## 1.2 Tujuan

1. Memahami konsep dasar perulangan pada Bahasa pemrograman Java, khususnya struktur *while* dan *do-while*.
2. Menganalisis perbedaan antara *while* dan *do-while* dalam hal alur eksekusi serta kondisi yang digunakan.
3. Mengimplementasikan perulangan *while* dan *do-while* dalam program sederhana untuk menyelesaikan permasalahan komputasi.
4. Melatih keterampilan logika dan algoritma, sehingga mahasiswa mampu menuliskan kode yang lebih efisien, sistematis, dan mudah dipahami.
5. Meningkatkan kemampuan *problem solving* melalui penerapan perulangan dalam berbagai studi kasus pemrograman.

## 1.3 Manfaat

1. Membantu mahasiswa dalam memahami konsep dasar perulangan pada bahasa pemrograman, khususnya perbedaan *while* dan *do-while*.
2. Melatih kemampuan menulis kode yang lebih efisien, terstruktur dan sesuai kebutuhan logika program.
3. Membiasakan mahasiswa berpikir logis dan sistematis dalam menyelesaikan permasalahan komputasi melalui penerapan perulangan.
4. Memberikan pengalaman praktis dalam mengimplementasikan perulangan untuk menyelesaikan berbagai permasalahan sederhana dalam kehidupan sehari-hari.
5. Menjadi dasar yang kuat untuk memahami konsep pemrograman yang lebih kompleks, seperti *nested loop*, *array processing*, maupun *data structure*.

## BAB II

### PEMBAHASAN

#### 2.1 Dasar Teori

##### 2.1.1 Konsep Perulangan

Perulangan adalah salah satu struktur kendali dalam pemrograman yang digunakan untuk mengeksekusi suatu blok perintah secara berulang selama kondisi tertentu terpenuhi. Dengan adanya perulangan, penulisan kode menjadi lebih efisien, mengurangi redundansi, serta mempermudah pemeliharaan program.

Secara umum, terdapat tiga jenis perulangan dalam bahasa pemrograman:

1. For loop → digunakan jika jumlah perulangan sudah diketahui. (pembahasan praktikum minggu lalu).
2. While loop → digunakan ketika jumlah perulangan belum pasti, tetapi bergantung pada kondisi logika.
3. Do-while loop → mirip dengan while, namun menjamin blok perintah dijalankan minimal satu kali.

##### 2.1.2 Perulangan *While*

Struktur *while* akan mengevaluasi kondisi terlebih dahulu sebelum menjalankan blok perintah. Jika kondisi bernilai benar (*true*), maka perintah di dalam blok akan dijalankan. Proses ini berulang hingga kondisi bernilai salah (*false*).

Sintaks dasarnya :

```
while (kondisi) {  
  
    // blok perintah yang akan diulang  
  
}
```

Ciri utama:

- Kondisi dicek di awal

- Jika kondisi awal salah, maka blok perintah tidak akan dijalankan sama sekali.

### 2.1.3 Perulangan *Do-While*

Berbeda dengan *while*, struktur *do-while* akan menjalankan blok perintah terlebih dahulu, kemudian baru mengevaluasi kondisi. Dengan demikian, perintah dalam blok akan selalu dijalankan minimal satu kali, meskipun kondisi bernilai salah sejak awal.

Sintaks dasar:

```
do {  
  
    // blok perintah yang akan diulang  
  
} while (kondisi);
```

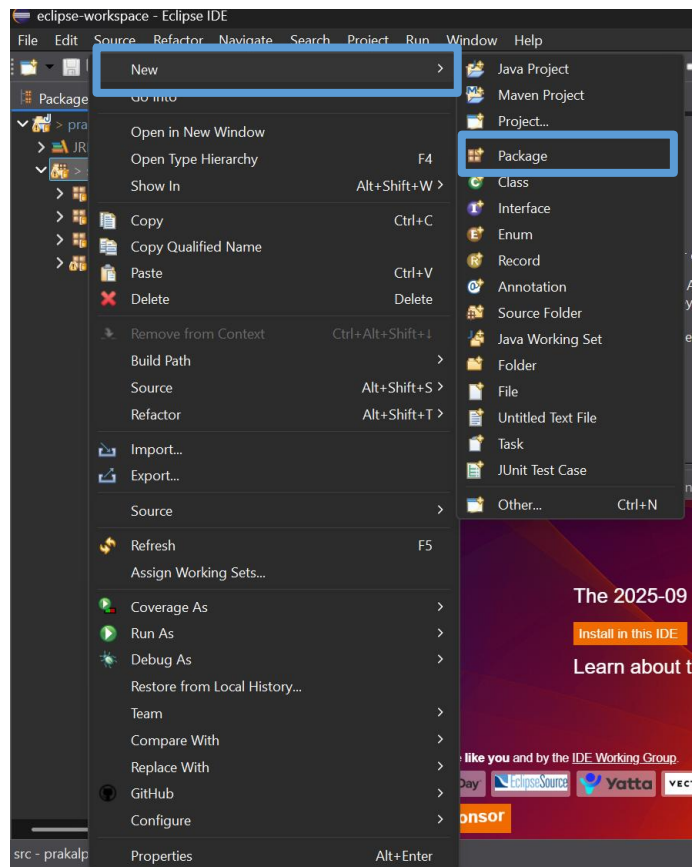
Ciri utama

- Kondisi dicek di akhir.
- Blok perintah dijalankan minimal satu kali.

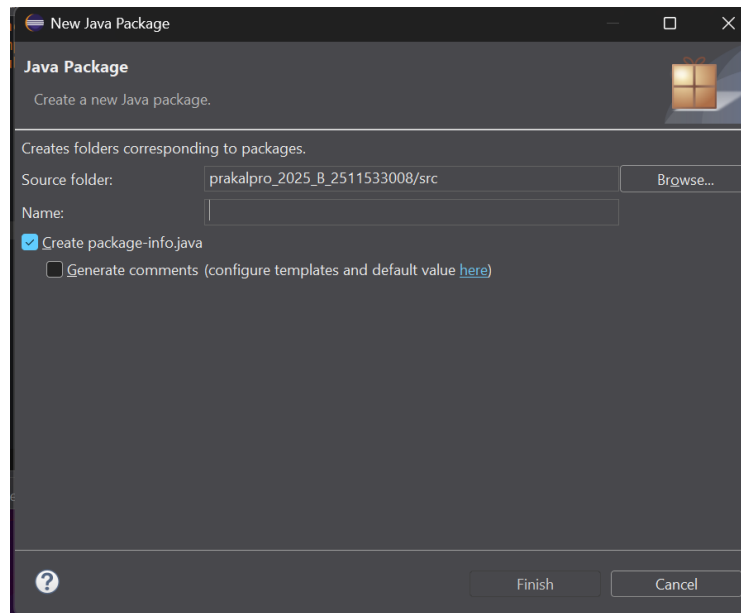
## 2.2 Langkah Kerja

### 2.2.1 Membuat *Package* dan *Class*

1. Sebelum membuat atau menuliskan kode program pertama kita harus membuat *package* dan *class* terlebih dahulu.

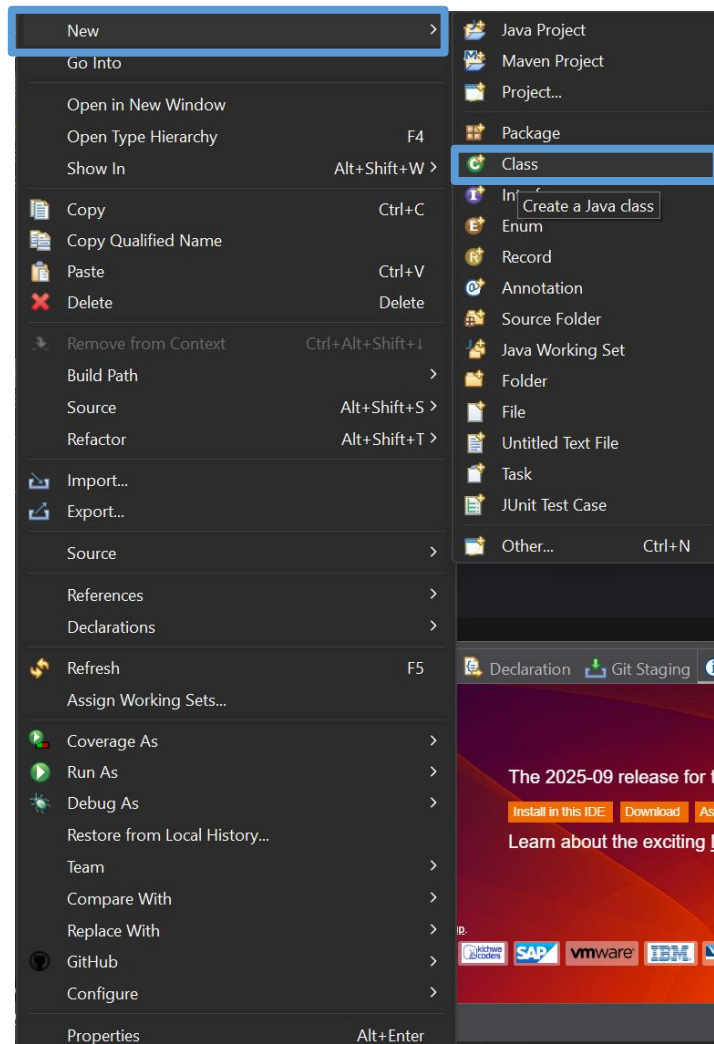


*Gambar 2.1 membuat package*

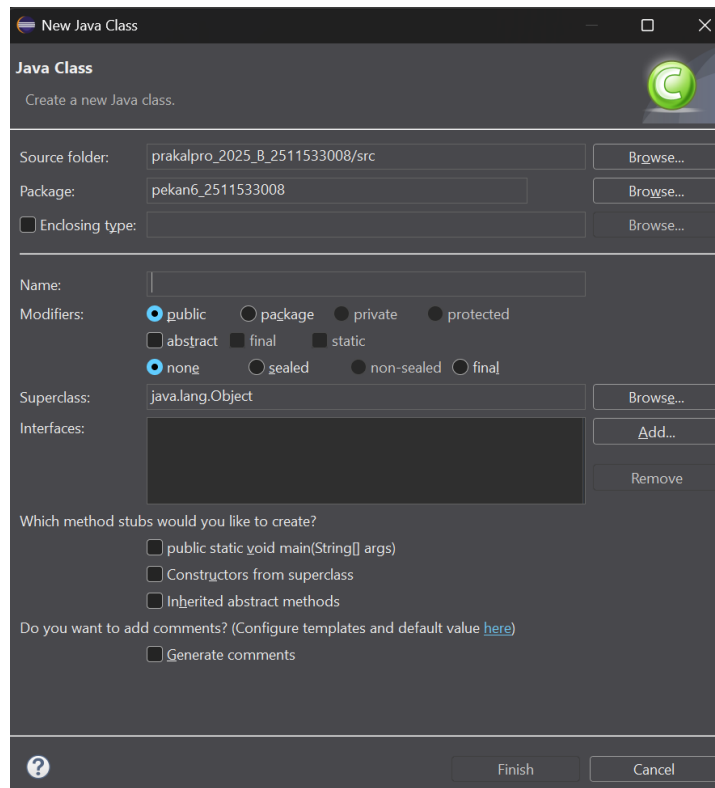


*Gambar 2.2 membuat nama package*





*Gambar 2.3 membuat class*



Gambar 2.4 membuat nama class

2. Setelah selesai membuat *package* dan *class* barulah kita mulai membuat kode program.

## 2.2.2 Program 1 : Perulangan *While* 1

1. Pada *class* yang telah dibuat tadi masukkan kode program berikut.

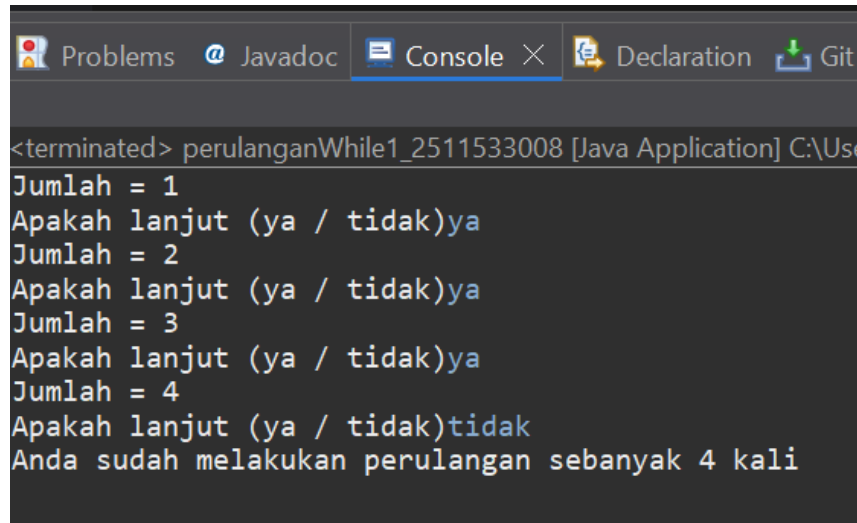
```

1 package pekan6_2511533008;
2
3 import java.util.Scanner;
4
5 public class perulanganWhile1_2511533008 {
6
7     public static void main(String[] args) {
8
9         int counter=0;
10        String jawab;
11        boolean running = true;
12        //deklarasi scanner
13        Scanner scan = new Scanner(System.in);
14        while (running) {
15            counter++;
16            System.out.println("Jumlah = "+counter);
17            System.out.print("Apakah lanjut (ya / tidak)");
18            jawab = scan.nextLine();
19            //cek jawab = tidak, perulangan berhenti
20            if (jawab.equalsIgnoreCase("tidak")) {
21                running = false;
22            }
23        }
24        System.out.println("Anda sudah melakukan perulangan sebanyak "+counter+" kali");
25
26    }
27
28 }
29

```

Kode program 2.1 perulangan while 1

2. Untuk menjalankan kode program berikut klik menu run dan akan muncul output pada menu console seperti berikut.

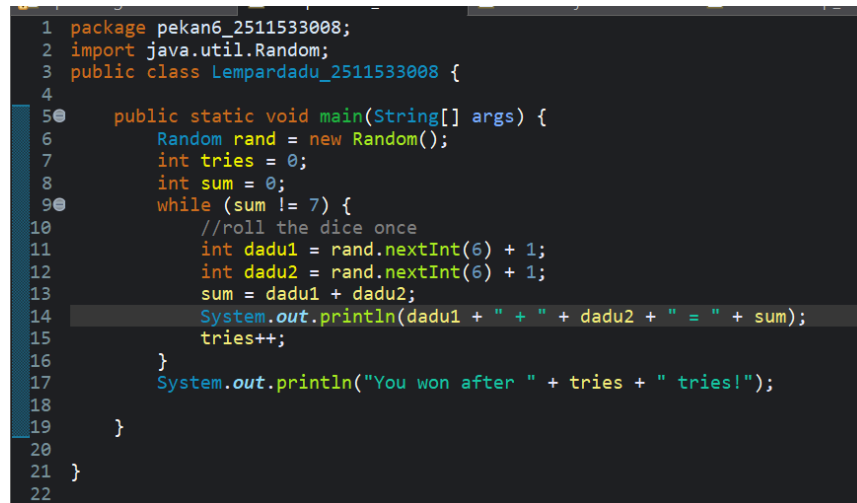


```
<terminated> perulanganWhile1_2511533008 [Java Application] C:\Use
Jumlah = 1
Apakah lanjut (ya / tidak)ya
Jumlah = 2
Apakah lanjut (ya / tidak)ya
Jumlah = 3
Apakah lanjut (ya / tidak)ya
Jumlah = 4
Apakah lanjut (ya / tidak)tidak
Anda sudah melakukan perulangan sebanyak 4 kali
```

*Gambar 2.5 output perulangan while 1*

### 2.2.3 Program 2 : Lempar Dadu

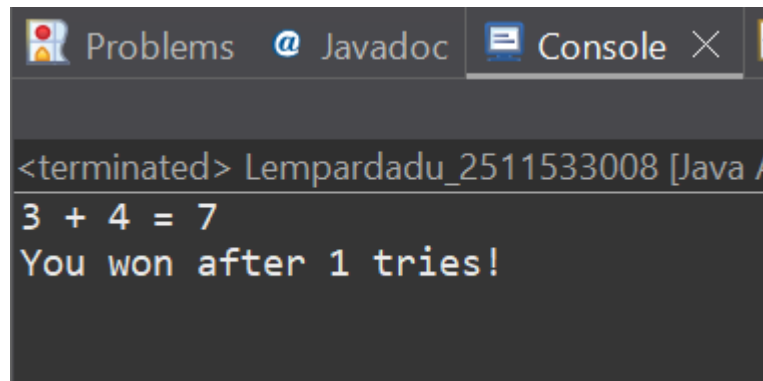
1. Untuk program kedua buat *class* baru dan masukkan kode program berikut.



```
1 package pekan6_2511533008;
2 import java.util.Random;
3 public class Lempardadu_2511533008 {
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 }
```

*Kode program 2.2 lempar dadu*

2. Sama seperti sebelumnya untuk menjalankan program klik menu run dan akan muncul output pada menu console seperti berikut.

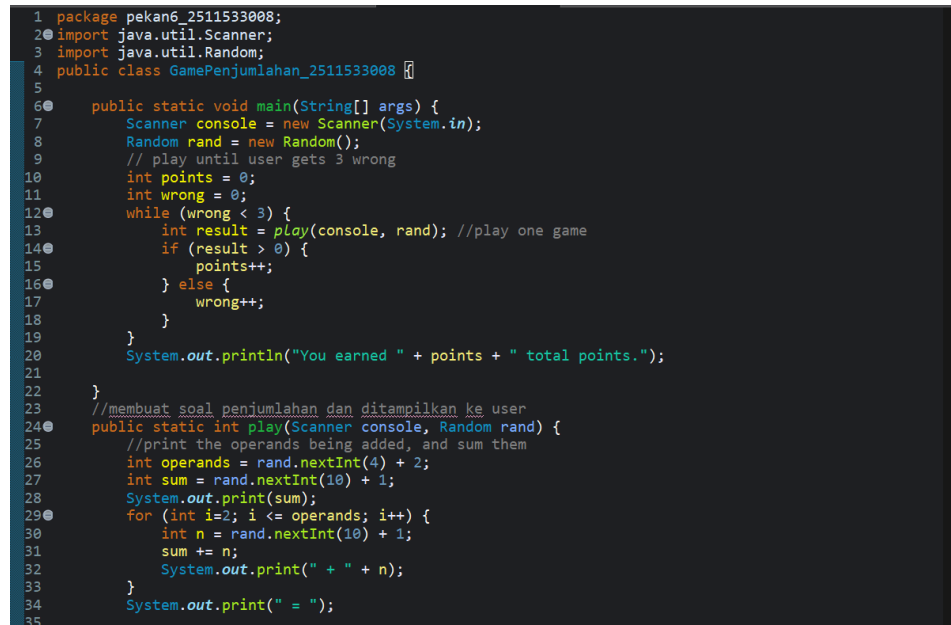


```
<terminated> Lempardadu_2511533008 [Java A
3 + 4 = 7
You won after 1 tries!
```

Gambar 2.6 output lempar dadu

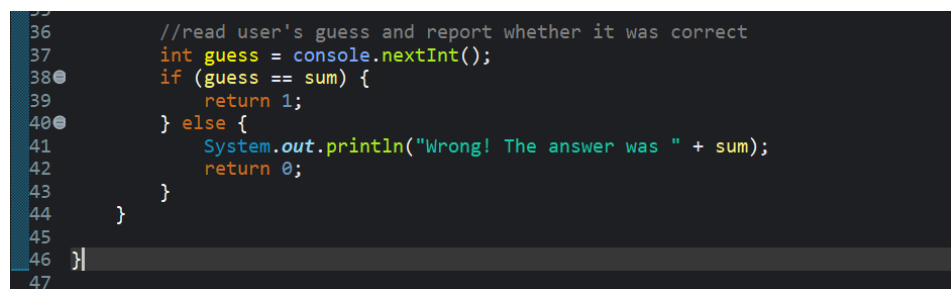
## 2.2.4 Program 3 : Game Penjumlahan

1. Sama seperti sebelumnya kita buat *class* baru dan masukkan kode program berikut.



```
1 package pekan6_2511533008;
2 import java.util.Scanner;
3 import java.util.Random;
4 public class GamePenjumlahan_2511533008 {
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 points = 0;
11        int wrong = 0;
12        while (wrong < 3) {
13            int result = play(console, rand); //play one game
14            if (result > 0) {
15                points++;
16            } else {
17                wrong++;
18            }
19        }
20        System.out.println("You earned " + points + " total points.");
21    }
22    //membuat soal penjumlahan dan ditampilkan ke user
23    public static int play(Scanner console, Random rand) {
24        //print the operands being added, and sum them
25        int operands = rand.nextInt(4) + 2;
26        int sum = rand.nextInt(10) + 1;
27        System.out.print(sum);
28        for (int i=2; i <= operands; i++) {
29            int n = rand.nextInt(10) + 1;
30            sum += n;
31            System.out.print(" + " + n);
32        }
33        System.out.print(" = ");
34    }
35}
```

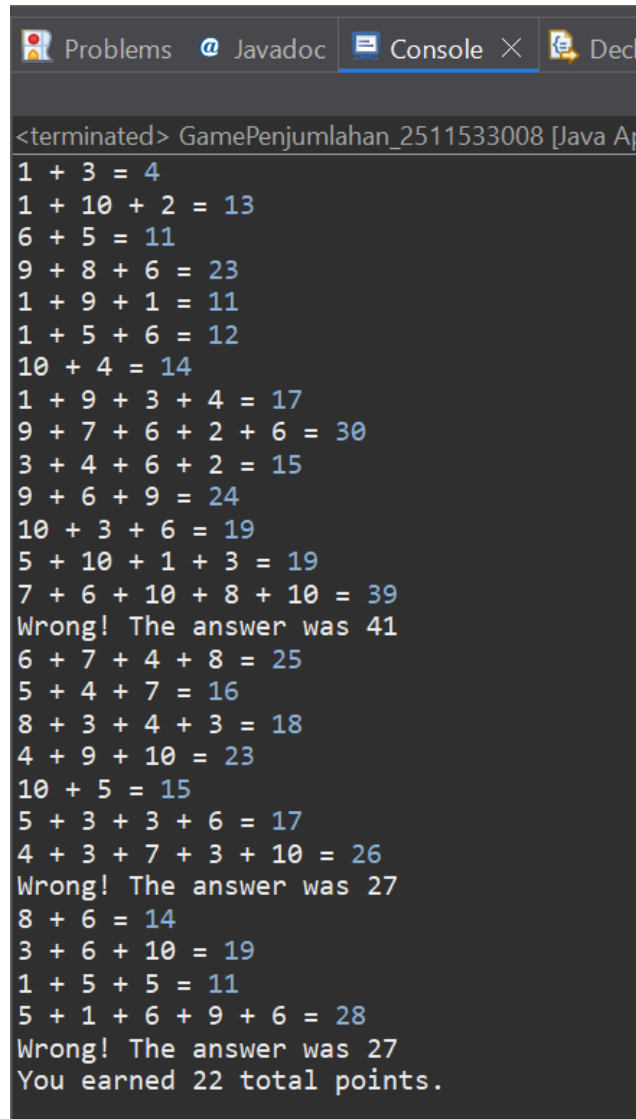
Kode program 2.3 game penjumlahan 1



```
36        //read user's guess and report whether it was correct
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 }
47
```

Kode program 2.4 game penjumlahan 2

2. Sama seperti sebelumnya untuk menjalankan program klik run dan akan tampil output pada menu console seperti berikut.



```
<terminated> GamePenjumlahan_2511533008 [Java Ap
1 + 3 = 4
1 + 10 + 2 = 13
6 + 5 = 11
9 + 8 + 6 = 23
1 + 9 + 1 = 11
1 + 5 + 6 = 12
10 + 4 = 14
1 + 9 + 3 + 4 = 17
9 + 7 + 6 + 2 + 6 = 30
3 + 4 + 6 + 2 = 15
9 + 6 + 9 = 24
10 + 3 + 6 = 19
5 + 10 + 1 + 3 = 19
7 + 6 + 10 + 8 + 10 = 39
Wrong! The answer was 41
6 + 7 + 4 + 8 = 25
5 + 4 + 7 = 16
8 + 3 + 4 + 3 = 18
4 + 9 + 10 = 23
10 + 5 = 15
5 + 3 + 3 + 6 = 17
4 + 3 + 7 + 3 + 10 = 26
Wrong! The answer was 27
8 + 6 = 14
3 + 6 + 10 = 19
1 + 5 + 5 = 11
5 + 1 + 6 + 9 + 6 = 28
Wrong! The answer was 27
You earned 22 total points.
```

*Gambar 2.7 output game penjumlahan*

Pada output kita akan menjawab pertanyaan tentang penjumlahan secara random yang diberikan sistem lalu jika sudah salah 3 kali maka akan muncul poin yang di dapat.

### 2.2.5 Program 4 : Sentinel loop

1. Sama seperti sebelumnya kita membuat *class* baru dan memasukkan kode program seperti berikut.

```

1 package pekan6_2511533008;
2 import java.util.Scanner;
3 public class SentinelLoop_2511533008 {
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("Masukkan 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 }

```

*Kode program 2.5 sentinel loop*

2. Sama seperti sebelumnya untuk menjalankan program klik run dan akan muncul output pada menu console seperti berikut.

```

<terminated> SentinelLoop_2511533008 [Java Application]
Masukkan angka (0 untuk keluar) : 2
Masukkan angka (0 untuk keluar) : 177
Masukkan angka (0 untuk keluar) : 78
Masukkan angka (0 untuk keluar) : 65
Masukkan angka (0 untuk keluar) : 7
Masukkan angka (0 untuk keluar) : 53
Masukkan angka (0 untuk keluar) : 25
Masukkan angka (0 untuk keluar) : 0
totalnya adalah 407

```

*Gambar 2.8 output sentinel loop*

### 2.2.6 Program 5 : Do-While 1

1. Sama seperti sebelumnya kita buat *class* baru terlebih dahulu lalu masukkan kode program seperti berikut.

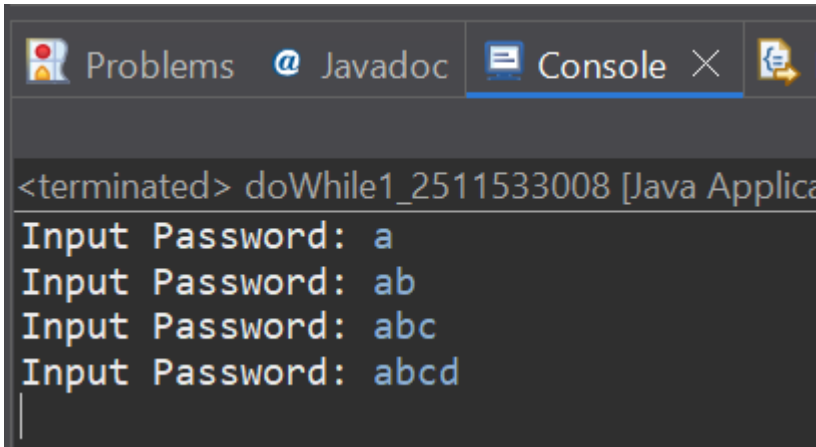
```

1 package pekan6_2511533008;
2 import java.util.Scanner;
3 public class doWhile1_2511533008 {
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"));
13
14    }
15
16 }
17

```

*Kode program 2.6 do-while 1*

2. Sama seperti sebelumnya untuk menjalankan program klik run dan akan muncul output pada menu console seperti berikut.



```

<terminated> doWhile1_2511533008 [Java Applica
Input Password: a
Input Password: ab
Input Password: abc
Input Password: abcd

```

*Gambar 2.9 output do-while 1*

Pada program ini jika input password yang di masukkan salah atau tidak sesuai maka program akan terus meminta untuk input password sampai password yang di masukkan benar atau sesuai.

## **BAB III**

### **PENUTUP**

#### **3.1 Kesimpulan**

Berdasarkan hasil praktikum yang telah dilakukan, dapat disimpulkan bahwa:

1. Perulangan merupakan konsep penting dalam pemrograman karena memungkinkan eksekusi perintah secara berulang tanpa penulisan kode yang berulang-ulang, sehingga program menjadi lebih efisien dan mudah dipahami.
2. Perulangan *while* melakukan pengecekan kondisi di awal, sehingga blok perintah hanya dijalankan jika kondisi bernilai benar. Hal ini membuat *while* cocok digunakan ketika jumlah perulangan belum pasti dan bisa saja tidak dijalankan sama sekali.
3. Perulangan *do-while* melakukan pengecekan kondisi di akhir, sehingga blok perintah akan selalu dijalankan minimal satu kali, meskipun kondisi bernilai salah sejak awal.
4. Perbedaan utama antara *while* dan *do-while* terletak pada posisi pengecekan kondisi, yang memengaruhi jumlah minimal eksekusi blok perintah.
5. Melalui praktikum ini, mahasiswa memperoleh pemahaman konseptual dan keterampilan praktis dalam mengimplementasikan kedua jenis perulangan, sekaligus melatih kemampuan logika, algoritma, dan pemecahan masalah dalam bahasa pemrograman Java.



## DAFTAR PUSTAKA

- [1] P. J. Deitel and H. M. Deitel, *Java: How to Program (Early Objects)*, 11th ed. Pearson Education, 2017.
- [2] W. J. Savitch, *Absolute Java*, 5th ed. Pearson, 2014.
- [3] Oracle, "The while and do-while Statements (Java Tutorials)," [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/while.html>. . [Accessed: Nov. 4, 2025].
- [4] Wahana Komputer, *Pemrograman Java untuk Pemula*. Yogyakarta: Andi Offset, 2019.
- [5] A. Nugroho, *Algoritma dan Pemrograman dengan Java*. Bandung: Informatika, 2018.