

LAPORAN PRAKTIKUM  
ALGORITMA PEMROGRAMAN PEKAN ENAM  
*WHILE STATEMENT*

disusun Oleh:

Zahira Nur Asyifa

2411532015

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

Asisten Praktikum: Rahmad Dwirizki Olders



DEPARTEMEN INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
UNIVERSITAS ANDALAS  
2025

## KATAPENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT karena atas rahmat dan karunia-Nya, laporan praktikum Algoritma dan Pemrograman dengan topik Perulangan While ini dapat diselesaikan dengan baik. Laporan ini disusun untuk memenuhi tugas praktikum pekan keenam mata kuliah praktikum Algoritma dan Pemrograman, sekaligus sebagai sarana pembelajaran dalam memahami konsep dasar perulangan menggunakan bahasa Java.

Melalui praktikum ini penulis dapat mempelajari cara kerja dan penerapan struktur perulangan while, baik dalam bentuk sederhana maupun kompleks. Praktikum ini juga membantu penulis memahami perbedaan antara perulangan *for* dan *while*, serta bagaimana memilih struktur perulangan yang tepat sesuai kebutuhan program. Penulis menyadari bahwa laporan ini masih jauh dari sempurna, baik dari segi penjelasan maupun penyajian. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Akhir kata, penulis mengucapkan terima kasih kepada dosen pengampu, asisten praktikum, serta semua pihak yang telah memberikan bimbingan dan dukungan sehingga laporan ini dapat terselesaikan. Semoga laporan ini dapat memberikan manfaat bagi penulis sendiri maupun pembaca.

Padang, 2025

Penulis

## DAFTAR ISI

\

KATA PENGANTAR .....	i
DAFTAR ISI .....	ii
DAFTAR GAMBAR .....	iii
BAB I PENDAHULUAN .....	4
1.1    Latar Belakang .....	4
1.2    Tujuan .....	4
1.3    Manfaat .....	4
BAB II PEMBAHASAN .....	5
2.1    Class Perulangan While1 .....	5
2.2    Class LemparDadu .....	9
2.3    Class GamePenjumlahan .....	11
2.4    Class SentinelLoop .....	15
2.5    Class DoWhile1 .....	17
BAB III KESIMPULAN .....	19
3.1    Kesimpulan .....	19
3.2    Saran .....	19
DAFTAR PUSTAKA .....	20

## DAFTAR GAMBAR

Gambar 2.1 Membuat package baru .....	5
Gambar 2. 2 Membuat nama package.....	5
Gambar 2. 3 Membuat <i>class</i> peru;anganwhile1 .....	6
Gambar 2. 4 <i>Syntax Class PerulanganWhile1</i> .....	7
Gambar 2. 5 Penjelasan syntax.....	7
<i>Gambar 2. 6 Penjelasan syntax</i> .....	7
Gambar 2. 7 Penjelasan syntax.....	8
<i>Gambar 2. 8 Ouput PerulanganWhile1</i> .....	8
Gambar 2. 9 Class LemparDadu .....	9
Gambar 2. 10 Syntax LemparDadu.....	9
Gambar 2. 11 Penjelasan syntax .....	10
Gambar 2. 12 Penjelasan syntax.....	10
Gambar 2. 13 Output LemparDadu.....	10
Gambar 2. 14 Class Game Penjumlahan.....	11
Gambar 2. 15 Syntax Game Penjumlahan .....	12
Gambar 2. 16 Penjelasan syntax.....	12
Gambar 2. 17 Penjelasan syntax.....	13
Gambar 2. 18 Penjelasan syntax.....	13
Gambar 2. 19 Output GamePenjumlahan .....	14
Gambar 2. 20 Class SentinelLoop .....	15
Gambar 2. 21 Syntax SentinelLoop .....	15
Gambar 2. 22 Penjelasan syntax.....	16
Gambar 2. 23 Penjelasan syntax.....	16
Gambar 2. 24 Output SentinelLoop .....	16
Gambar 2. 25 Class DoWhile1 .....	17
Gambar 2. 26 Syntax DoWhile1 .....	17
Gambar 2. 27 Penjelasan syntax.....	18
Gambar 2. 28 Penjelasan syntax.....	18
Gambar 2. 29 Output DoWhile1 .....	18

## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

Perulangan pada java terbagi menjadi tiga jenis yaitu : For, While dan Do While. Pada praktikum sebelumnya telah menggunakan perulangan for yang dimana ketiga syarat ditulis dalam 1 baris perintah seperti `for (i = 1; i < 5; i++)`. Di dalam perulangan while, ketiga kondisi ini saling terpisah. Di bagian start biasanya ditulis perintah inisialisasi variabel counter, misalnya `i = 0`. Di bagian condition terdapat kondisi yang harus dipenuhi agar perulangan berjalan, misalnya `i < 5`. Kemudian perintah increment harus berada di dalam block perulangan agar bisa menaikkan nilai variabel counter, misalnya dengan perintah `i++`.

Salah satu hal yang harus selalu diingat ketika membuat perulangan while adalah, jangan lupa membuat perintah increment. Jika tidak, kondisi akhir tidak akan pernah terpenuhi dan perulangan akan berjalan terus menerus. Ini dikenal dengan istilah infinity loop.

### 1.2 Tujuan

1. Memahami konsep dasar struktur perulangan while dan do-while
2. Mengetahui perbedaan antara berbagai jenis perulangan

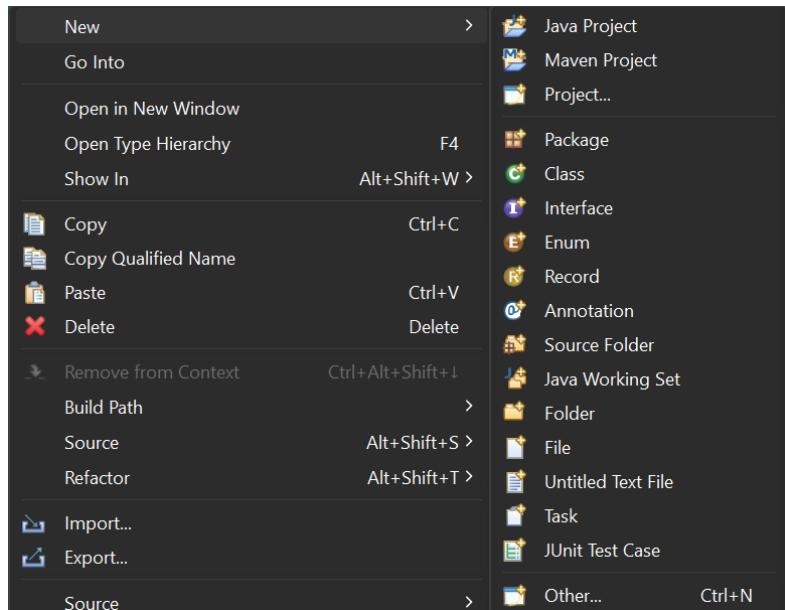
### 1.3 Manfaat

1. Mahasiswa dapat memahami cara kerja perulangan dalam mengotomatisasi proses berulang tanpa harus menulis kode secara berulang.
2. Dapat membedakan situasi yang tepat untuk menggunakan while, do-while, atau for.

## BAB II PEMBAHASAN

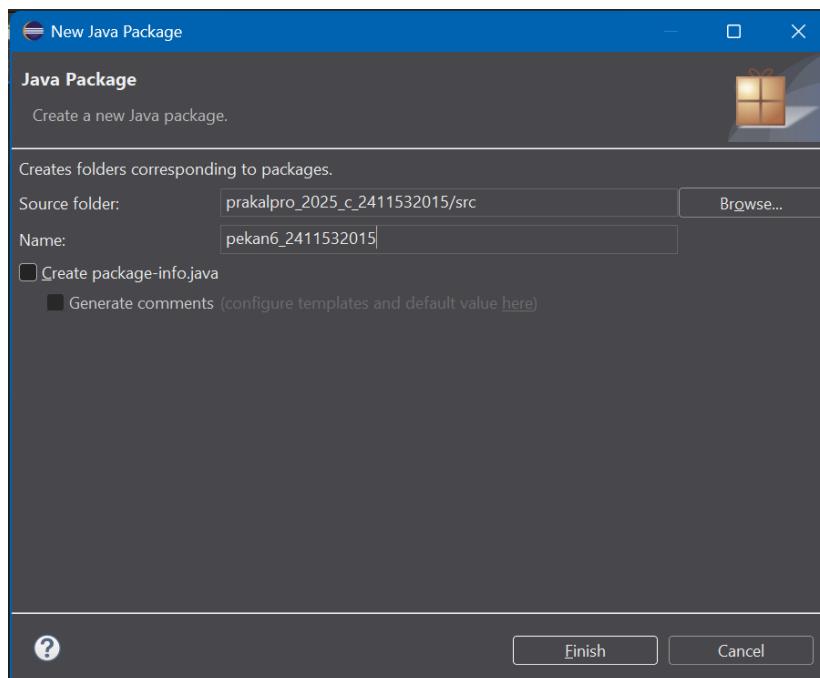
### 2.1 Class Perulangan While1

1. Buat Buat *package* baru dengan klik kanan pada *src*, pilih new lalu *package*



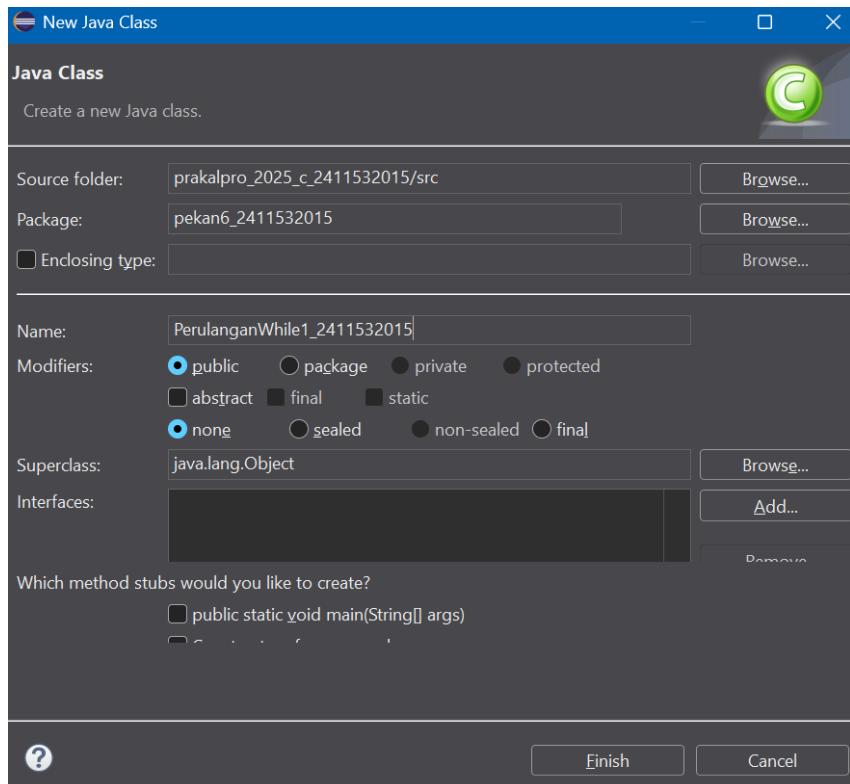
Gambar 2.1 Membuat package baru

2. Buat *package* dengan nama pekan 6



Gambar 2. 2 Membuat nama package

3. Buat *class* baru dengan nama “PerulanganWhile1”



Gambar 2. 3 Membuat *class* perulanganwhile1

4. Buat *syntax* seperti ini

```
1 package pekan6_2411532015;
2
3 import java.util.Scanner;
4
5 public class PerulanganWhile1_2411532015 {
6     public static void main (String[] args) {
7         int counter=0;
8         String jawab;
9         boolean running = true;
10        Scanner scan = new Scanner (System.in);
11        . . .
```

```
1  while(running) {  
2      counter++;  
3      System.out.println("Jumlah = "+counter);  
4      System.out.print("Apakah lanjut (ya/tidak?)");  
5      jawab = scan.nextLine();  
6  
7      if (jawab.equalsIgnoreCase("tidak")) {  
8          running = false;  
9      }  
10     }  
11     System.out.println("Anda sudah melakukan perulangan sebanyak "  
12         "+counter+ " kali");  
13     scan.close();  
14 }  
15 }
```

Gambar 2. 4 Syntax Class PerulanganWhile1

5. Mendeklarasikan variabel dan scanner

```
7     int counter=0;  
8     String jawab;  
9     boolean running = true;  
10    Scanner scan = new Scanner (System.in);
```

Gambar 2. 5 Penjelasan syntax

6. Memulai perulangan while yang akan terus dijalankan selama kondisi running bernilai true.

```
1  Scanner scan = new Scanner (System.in);  
2  while(running) {  
3      counter++;  
4      System.out.println("Jumlah = "+counter);  
5      System.out.print("Apakah lanjut (ya/tidak?)");  
6      jawab = scan.nextLine();  
7  
8      if (jawab.equalsIgnoreCase("tidak")) {  
9          running = false;  
10     }  
11 }
```

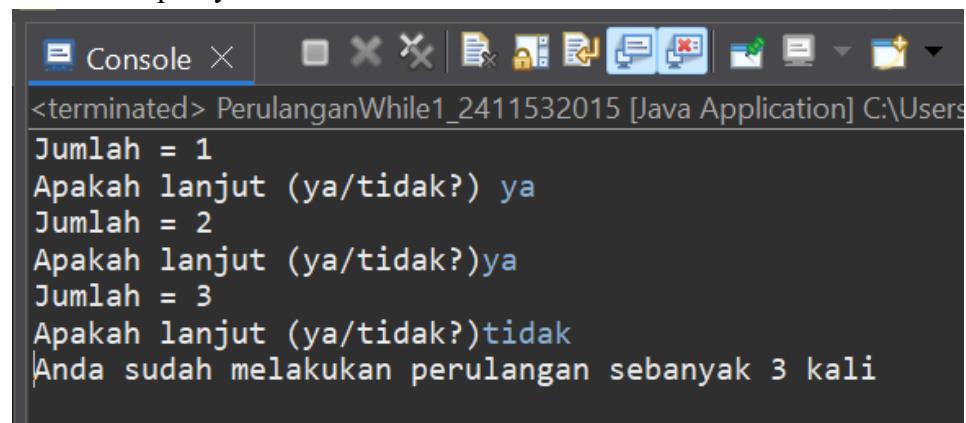
Gambar 2. 6 Penjelasan syntax

7. Setelah perulangan berhenti, baris ini menampilkan **pesan akhir** yang memberitahukan total jumlah perulangan yang telah dilakukan pengguna.

```
9     }
10    System.out.println("Anda sudah melakukan perulangan sebanyak "
11        +counter+ " kali");
12    scan.close();
13  }
14
15 }
```

Gambar 2. 7 Penjelasan syntax

8. Berikut outputnya



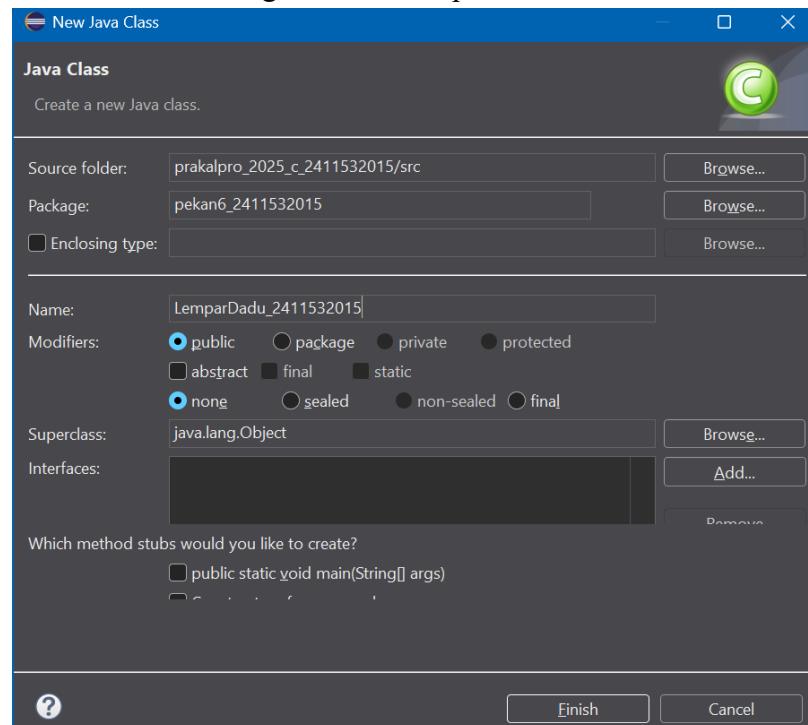
The screenshot shows a Java application window with the title 'PerulanganWhile1\_2411532015 [Java Application]'. The 'Console' tab is selected, displaying the following text:

```
<terminated> PerulanganWhile1_2411532015 [Java Application] C:\Users
Jumlah = 1
Apakah lanjut (ya/tidak?) ya
Jumlah = 2
Apakah lanjut (ya/tidak?)ya
Jumlah = 3
Apakah lanjut (ya/tidak?)tidak
Anda sudah melakukan perulangan sebanyak 3 kali
```

Gambar 2. 8 Ouput PerulanganWhile1

## 2.2 Class LemparDadu

1. Buat class baru dengan nama LemparDadu



Gambar 2. 9 Class LemparDadu

2. Buat syntax seperti ini

```
package pekan6_2411532015;

import java.util.Random;

public class LemparDadu_2411532015 {
    public static void main (String[] args) {
        Random rand = new Random();
        int tries = 0;
        int sum = 0;
        while (sum!=7) {
            int dadu1 = rand.nextInt(6) + 1;
            int dadu2 = rand.nextInt(6) + 1;
            sum = dadu1 + dadu2;
            System.out.println(dadu1 + " + " + dadu2 + " = " + sum);
            tries++;
        }
        System.out.println("You won after " + tries + " tries!");
    }
}
```

Gambar 2. 10 Syntax LemparDadu

3. Membuat objek Random Bernama rand dan mendeklarasikan variabel

```
public static void main (String[] args) {  
    Random rand = new Random();  
    int tries = 0;  
    int sum = 0;
```

Gambar 2. 11 Penjelasan syntax

4. Perulangan while terus berjalan selama jumlah dua dadu belum sama dengan

```
7  
while (sum!=7) {  
    int dadu1 = rand.nextInt(6) + 1;  
    int dadu2 = rand.nextInt(6) + 1;  
    sum = dadu1 + dadu2;  
    System.out.println(dadu1 + " + " + dadu2 + " = " + sum);  
    tries++;  
}  
System.out.println("You won after " + tries + " tries!");
```

Gambar 2. 12 Penjelasan syntax

- Rand.nextInt(6) memberi hasil angka antara 0-5, lalu ditambah 1 menjadi 1-6
- Sum untuk menjumlahkan hasil dadu yang akan dicek oleh while
- Lalu menampilkan hasil ke layar
- tries++ menambahkan nilai sebanyak 1 setiap perulangan terjadi

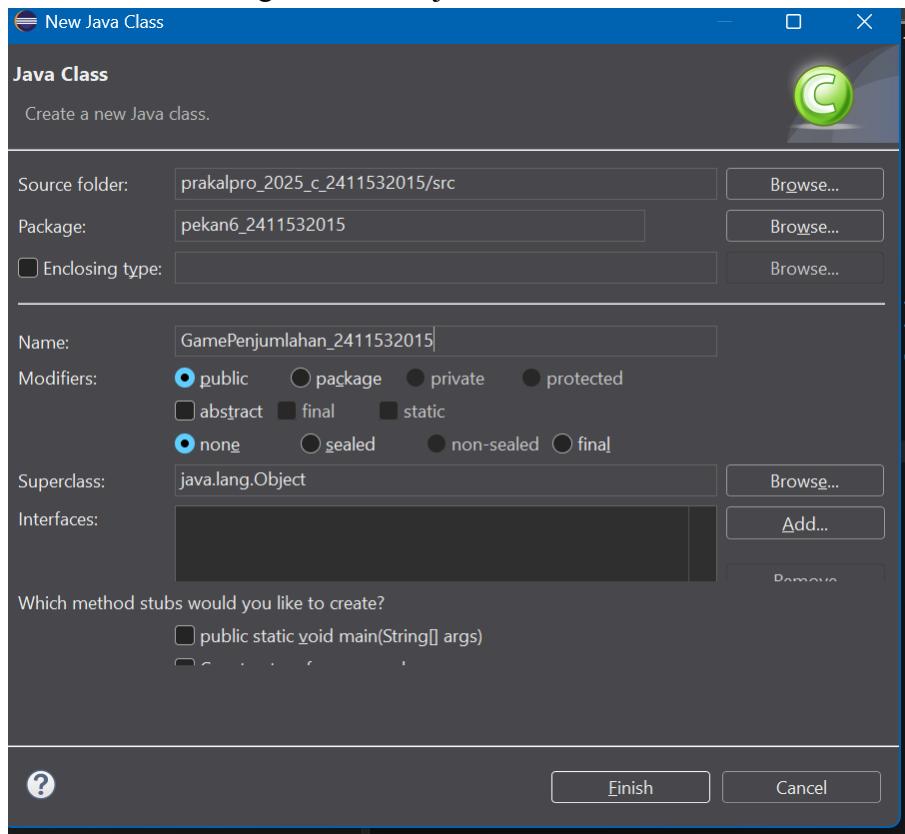
5. Output

```
Console  
<terminated> LemparDadu_24115320  
1 + 3 = 4  
2 + 3 = 5  
4 + 3 = 7  
You won after 3 tries!
```

Gambar 2. 13 Output LemparDadu

## 2.3 Class GamePenjumlahan

1. Buat class baru dengan GamePenjumlahan



Gambar 2. 14 Class Game Penjumlahan

2. Buat syntax seperti ini

```

1 package pekan6_2411532015;
2
3 import java.util.*;
4
5 public class GamePenjumlahan_2411532015 {
6     public static void main (String[] args) {
7         Scanner console = new Scanner (System.in);
8         Random rand = new Random();
9
10        int points = 0;
11        int wrong = 0;
12        while (wrong<3) {
13            int result = play(console, rand);
14            if(result>0) {
15                points++;
16            } else {
17                wrong++;
18            }
19        }
20        System.out.println("You eanrned " + points + " total points.");
21    }
22    public static int play(Scanner console, Random rand) {
23        int operands = rand.nextInt(4) +2;
24        int sum = rand.nextInt(10) +1;
25        System.out.print(sum);
26
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        int guess = console.nextInt();
34        if(guess==sum) {
35
36            return 1;
37        } else {
38            System.out.println("Wrong! The Answer was " + sum);
39            return 0;
40        }
41    }
42}

```

Gambar 2. 15 Syntax Game Penjumlahan

### 3. Deklarasi awal

```

6    public static void main (String[] args) {
7        Scanner console = new Scanner (System.in);
8        Random rand = new Random();
9
10       int points = 0;
11       int wrong = 0;

```

Gambar 2. 16 Penjelasan syntax

- Membuat objek scanner dan random
- Mendeklarasikan variabel points untuk jawaban benar dan wrong untuk jumlah kesalahan

4. Perulangan while akan terus berjalan selama jumlah kesalahan kurang dari 3

```
12@     while (wrong<3) {
13@         int result = play(console, rand);
14@         if(result>0) {
15@             points++;
16@         }else {
17@             wrong++;
18@         }
19@     }
20@     System.out.println("You eanred " + points + " total points.");
21@ }
```

Gambar 2. 17 Penjelasan syntax

- Int result memanggil method play() yang mengembalikan nilai 1 jika benar dan 0 jika salah
- Blok if mengecek hasil play() jika result bernilai 1 tambahkan 1 points, jika 0 tambahkan 1 ke wrong
- Setelah perulangan berhenti program menampilkan pesan beserta total poin

5. Method play()

```
21@     }
22@     public static int play(Scanner console, Random rand) {
23@         int operands = rand.nextInt(4) +2;
24@         int sum = rand.nextInt(10) +1;
25@         System.out.print(sum);
26@ 
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@         int guess = console.nextInt();
34@         if(guess==sum) {
35@ 
36@             return 1;
37@         } else {
38@             System.out.println("Wrong! The Answer was " + sum);
39@             return 0;
40@         }
41@     }
42@ }
```

Gambar 2. 18 Penjelasan syntax

- int operands menentukan jumlah bilangan yang akan dijumlahkan dari 2-5
- int sum menghasilkan bilangan acak dari 1-10
- blok for untuk menambahkan bilangan acak berikutnya
- int guess membaca jawaban dari input pengguna
- blok if untuk mengecek apakah jawaban pengguna benar

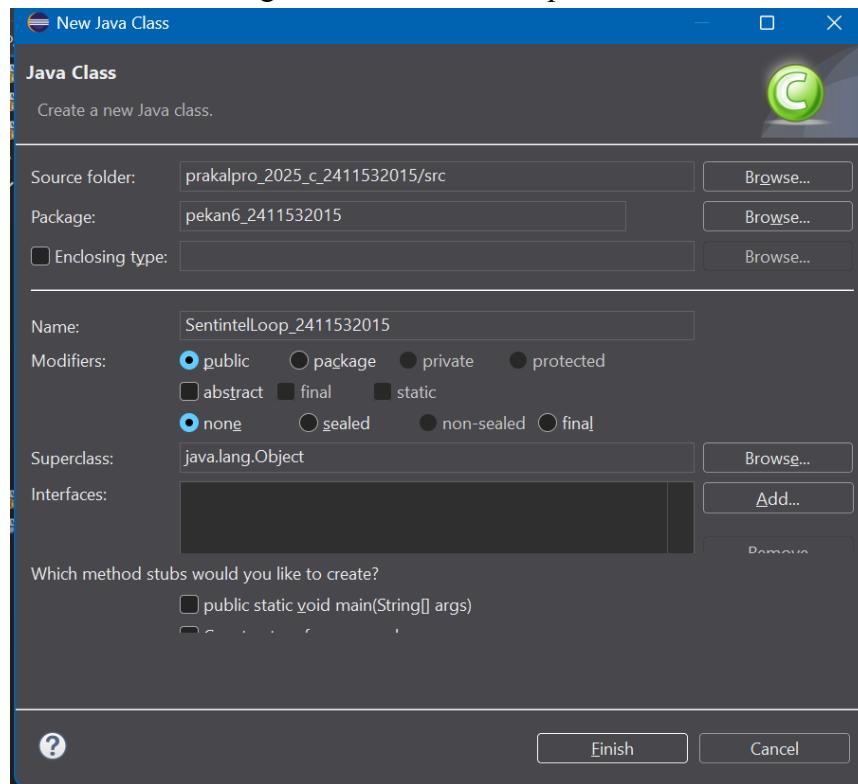
6. Output

```
<terminated> GamePenjumlahan_241153201
5 + 10 + 3 + 4 + 4 = 26
5 + 6 + 3 + 2 = 16
8 + 10 + 5 + 2 + 2 = 27
6 + 6 + 8 + 6 + 7 = 33
1 + 3 + 6 + 2 = 12
1 + 5 + 4 + 1 = 11
2 + 4 + 4 + 9 = 19
2 + 6 + 10 = 28
Wrong! The Answer was 18
2 + 9 + 9 = 20
9 + 1 = 10
7 + 6 + 2 + 10 = 25
7 + 10 + 2 + 6 + 5 = 30
5 + 10 + 2 = 17
5 + 5 + 4 + 5 + 8 = 27
10 + 2 + 3 + 4 + 5 = 24
4 + 10 = 14
6 + 1 = 7
10 + 4 + 6 + 3 = 23
7 + 2 + 5 = 14
2 + 8 + 9 + 9 + 5 = 2
Wrong! The Answer was 33
10 + 8 = 18
1 + 5 + 2 + 10 + 4 = 22
8 + 3 + 6 = 17
10 + 9 + 2 + 2 = 23
7 + 7 + 3 + 5 + 5 = 23
Wrong! The Answer was 27
You eanrned 22 total points.
```

Gambar 2. 19 Output GamePenjumlahan

## 2.4 Class SentinelLoop

1. Buat class baru dengan nama SentinelLoop



Gambar 2. 20 Class SentinelLoop

2. Buat syntax seperti ini

```
package pekan6_2411532015;

import java.util.Scanner;

public class SentintelLoop_2411532015 {

    public static void main(String[] args) {
        Scanner console = new Scanner(System.in);
        int sum = 0;
        int number=1;

        while (number != 0) {
            System.out.print("Masukkan angka (0 untuk keluar): ");
            number = console.nextInt();
            sum = sum + number;
        }
        System.out.println("totalnya adalah " + sum);
        console.close();
    }
}
```

Gambar 2. 21 Syntax SentinelLoop

3. Membuat objek scanner dan mendeklarasikan dua varibael

```
public static void main(String[] args) {  
    Scanner console = new Scanner(System.in);  
    int sum = 0;  
    int number=1;
```

Gambar 2. 22 Penjelasan syntax

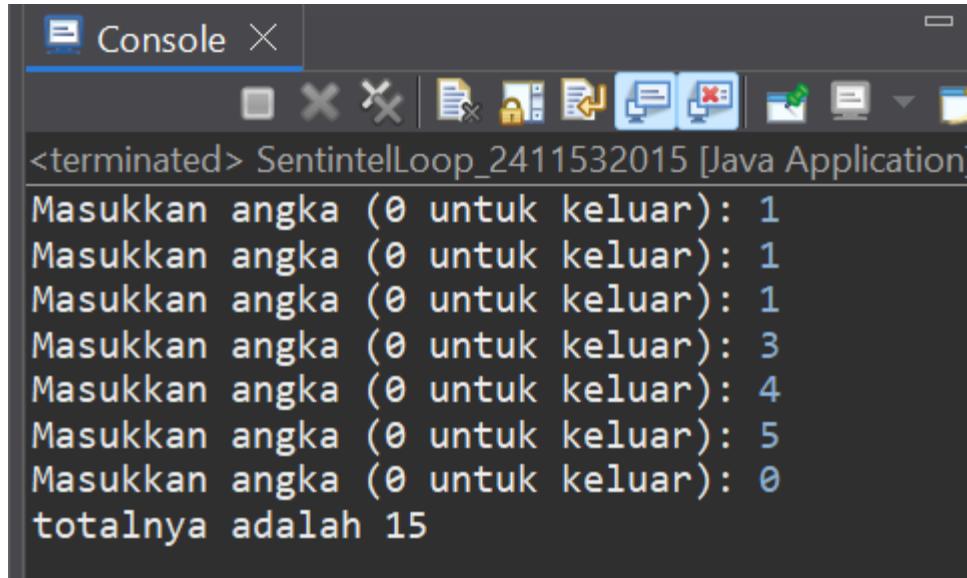
4. Perulangan berjalan selama nilai number tidak sama dengan 0

```
while (number != 0) {  
    System.out.print("Masukkan angka (0 untuk keluar): ");  
    number = console.nextInt();  
    sum = sum + number;  
}  
System.out.println("totalnya adalah " + sum);  
console.close();  
}
```

Gambar 2. 23 Penjelasan syntax

- Number = conseole.nextInt(); membaca input pengguna
- sum menambahkan angka yang dimasukkan ke total

5. Output

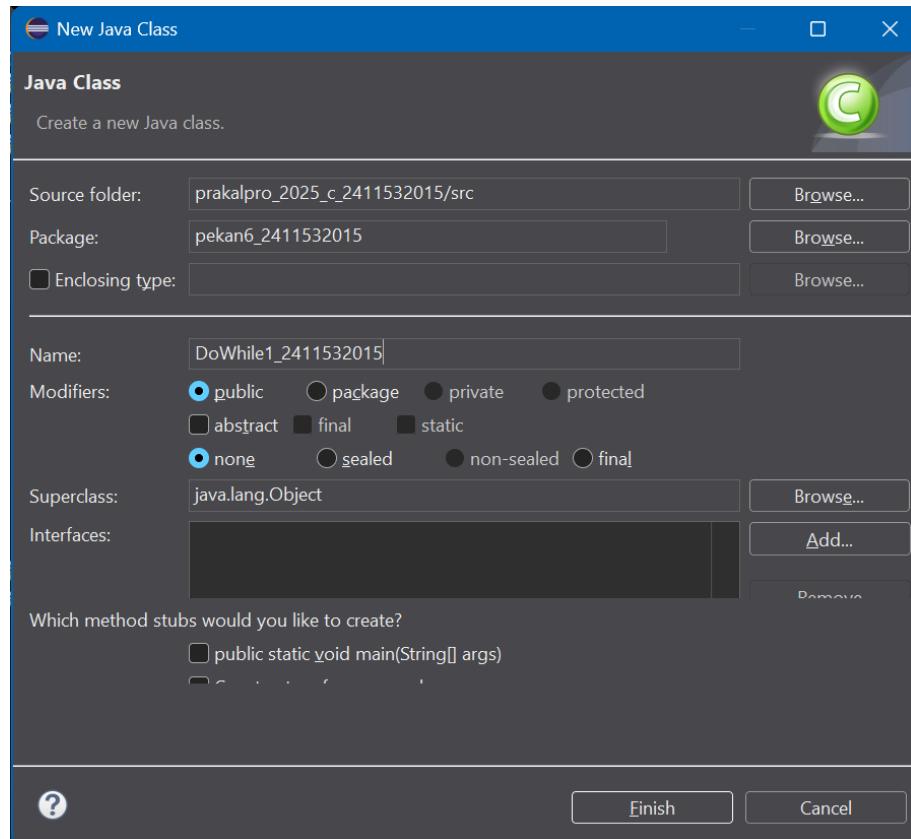


The screenshot shows a Java application window titled 'Console'. The title bar has a close button. Below the title bar is a toolbar with several icons. The main area of the window is a text terminal. It displays the following text:  
<terminated> SentinelLoop\_2411532015 [Java Application]  
Masukkan angka (0 untuk keluar): 1  
Masukkan angka (0 untuk keluar): 1  
Masukkan angka (0 untuk keluar): 1  
Masukkan angka (0 untuk keluar): 3  
Masukkan angka (0 untuk keluar): 4  
Masukkan angka (0 untuk keluar): 5  
Masukkan angka (0 untuk keluar): 0  
totalnya adalah 15

Gambar 2. 24 Output SentinelLoop

## 2.5 Class DoWhile1

- Buat class baru dengan nama DoWhile1



Gambar 2. 25 Class DoWhile1

- Buat syntax seperti ini

```
package pekan6_2411532015;

import java.util.Scanner;

public class DoWhile1_2411532015 {
    public static void main (String[] args) {
        Scanner console = new Scanner(System.in);
        String phrase;
        do {
            System.out.print("Input Password: ");
            phrase = console.next();
        } while (!phrase.equals("abcd"));
        console.close();
    }
}
```

Gambar 2. 26 Syntax DoWhile1

3. Membuat objek scanner dan mendeklarasikan variabel

```
public static void main (String[] args) {  
    Scanner console = new Scanner(System.in);  
    String phrase;
```

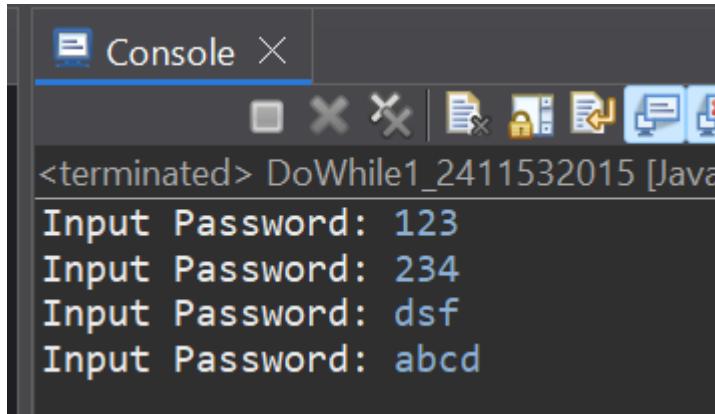
Gambar 2. 27 Penjelasan syntax

4. Perulangan do while dijalankan minimal sekali. Program akan terus meminta input password sampai pengguna memasukkan abcd

```
do {  
    System.out.print("Input Password: ");  
    phrase = console.next();  
} while (!phrase.equals("abcd"));  
console.close();  
}
```

Gambar 2. 28 Penjelasan syntax

5. Output



```
Console  
<terminated> DoWhile1_2411532015 [Java]  
Input Password: 123  
Input Password: 234  
Input Password: dsf  
Input Password: abcd
```

Gambar 2. 29 Output DoWhile1

## **BAB III KESIMPULAN**

### **3.1 Kesimpulan**

Dari hasil praktikum dapat disimpulkan bahwa struktur perulangan merupakan salah satu konsep fundamental dalam pemrograman karena memungkinkan proses yang berulang dijalankan secara otomatis tanpa menulis kode berulang. Setiap jenis perulangan memiliki karakteristik dan kegunaan tersendiri tergantung pada kebutuhan logika program.

### **3.2 Saran**

Saran untuk pengembangan dan pembelajaran selanjutnya adalah agar mahasiswa selalu memperhatikan kondisi logika dan validasi input agar program berjalan dengan benar dan tidak terjadi kesalahan tak terbatas. Selain itu, penambahan komentar pada setiap bagian kode sangat dianjurkan agar program lebih mudah dipahami dan dipelajari kembali di masa mendatang.

## **DAFTAR PUSTAKA**

[1] duniaikom, “Tutorial Belajar Java Part 34: Perulangan WHILE Bahasa Java”. [Daring]. Tersedia pada: <https://www.duniaikom.com/tutorial-belajar-java-perulangan-while-bahasa-java/>