

LAPORAN PRATIKUM
ALGORITMA DAN PEMROGRAMAN
PENERAPAN PERULANGAN DALAM JAVA



disusun Oleh:
Annisa Layli Ramadhani
2511532024

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

DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
TAHUN 2025

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT karena dengan rahmat-Nya laporan praktikum Algoritma dan Pemrograman tanggal 3 November 2025 dapat diselesaikan. Laporan ini membahas materi tentang Penerapan Perulangan dalam Java sebagai dasar penting dalam memahami pemrograman. Penulis mengucapkan terima kasih kepada dosen, asisten, dan rekan-rekan yang telah membantu selama praktikum. Penulis menyadari laporan ini masih jauh dari sempurna, sehingga kritik dan saran sangat diharapkan. Semoga laporan ini bermanfaat bagi pembaca.

Padang, 3 November 2025

Penulis

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
BAB I.....	4
1.1 Latar Belakang	4
1.2 Tujuan	5
1.3 Manfaat	5
BAB II	6
2.1 Dasar Teori	6
2.1.1 Perulangan	6
2.1.2 Perulangan While.....	6
2.1.3 Perulangan Do-While	7
2.2 Langkah Kerja.....	7
2.2.1 Pembuatan <i>Package</i> dan <i>Class</i> Pekan 6.....	7
2.2.2 Program Pertama: PerulanganWhile1_2511532024	9
2.2.3 Program Kedua: Lempardadu_2511532024	10
2.2.4 Program Ketiga: GamePenjumlahan_2511532024.....	10
2.2.5 Program Keempat: SentinelLoop_2511532024.....	11
2.2.6 Program Kelima: doWhile1_2511532024	12
2.2.7 <i>Commit</i> dan <i>Push</i> ke GitHub.....	12
BAB III.....	15
3.1 Kesimpulan	15
3.2 Saran.....	15
DAFTAR PUSTAKA	16

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam dunia pemrograman, konsep perulangan (looping) merupakan salah satu dasar penting yang digunakan untuk menjalankan instruksi secara berulang tanpa harus menulis kode yang sama berulang kali. Bahasa pemrograman Java menyediakan berbagai jenis struktur perulangan, di antaranya adalah while dan do-while. Kedua jenis perulangan ini memiliki fungsi utama yang sama, yaitu mengeksekusi blok perintah selama kondisi tertentu masih terpenuhi, namun memiliki perbedaan dalam cara evaluasi kondisinya.

Perulangan while akan mengevaluasi kondisi terlebih dahulu sebelum menjalankan blok perintah. Artinya, jika kondisi bernilai salah sejak awal, maka perulangan tidak akan dijalankan sama sekali. Struktur ini sangat berguna ketika jumlah iterasi belum diketahui secara pasti, dan program harus terus berjalan selama kondisi tertentu masih terpenuhi. Contohnya, program yang terus meminta input pengguna hingga mereka memasukkan nilai tertentu.

Sementara itu, perulangan do-while memiliki perbedaan penting, yaitu blok perintah dijalankan terlebih dahulu, baru kemudian kondisi diperiksa. Dengan demikian, perulangan ini akan selalu dijalankan minimal satu kali meskipun kondisi bernilai salah sejak awal. Struktur ini sering digunakan pada kasus yang membutuhkan eksekusi awal sebelum melakukan pengecekan, seperti menampilkan menu berulang hingga pengguna memilih untuk keluar.

Oleh karena itu, pembahasan mengenai perulangan while dan do-while dalam bahasa java menjadi penting dalam laporan ini, agar dapat memberikan pemahaman yang lebih mendalam tentang logika kontrol alur program, perbedaan cara kerja masing-masing struktur, serta penerapannya dalam pembuatan program yang interaktif dan efisien.

1.2 Tujuan

Tujuan dari pelaksanaan praktikum antara lain sebagai berikut:

- 1.2.1 Memahami konsep dasar perulangan while dan do-while dalam bahasa pemrograman java.
- 1.2.2 Mengetahui perbedaan antara perulangan while dan do-while dari segi alur eksekusi dan cara kerja.
- 1.2.3 Mampu mengimplementasikan perulangan while dan do-while dalam berbagai kasus pemrograman seperti berulang, validasi data, dan pengulangan menu program.

1.3 Manfaat

Manfaat dari pelaksanaan praktikum antara lain sebagai berikut:

- 1.3.1 Mempermudah pengulangan proses tertentu berdasarkan kondisi logis tanpa campur tangan manual.
- 1.3.2 Meningkatkan fleksibilitas program, terutama pada kasus yang melibatkan input pengguna atau proses yang berulang hingga kondisi tertentu tercapai.
- 1.3.3 Membantu memahami logika kontrol program, sehingga memudahkan pengembangan program yang lebih kompleks di masa depan.

BAB II

PEMBAHASAN

2.1 Dasar Teori

2.1.1 Perulangan

Perulangan adalah salah satu konsep dasar dalam pemrograman yang berfungsi untuk mengeksekusi satu atau beberapa perintah secara berulang-ulang selama kondisi tertentu masih terpenuhi. Dengan adanya perulangan, kita tidak perlu menulis baris kode yang sama berkali-kali, karena komputer akan secara otomatis menjalankan instruksi tersebut berulang sesuai dengan syarat atau batas yang telah ditentukan. Konsep ini sangat penting dalam pengembangan program yang efisien, terstruktur, dan mudah dikelola.

Dalam bahasa pemrograman java, perulangan termasuk dalam struktur kendali alur (control structure) yang digunakan untuk mengatur urutan eksekusi program. Tujuan utama penggunaan perulangan adalah untuk menghemat waktu penulisan kode, mempermudah pemrosesan data dalam jumlah banyak, serta meningkatkan efisiensi program. Misalnya, jika sebuah program harus menampilkan angka dari 1 sampai 100, kita tidak perlu menulis 100 baris kode, cukup menggunakan satu perulangan untuk mengatur pengulangannya.

2.1.2 Perulangan While

Perulangan while adalah salah satu jenis loop dalam bahasa pemrograman Java yang digunakan untuk mengeksekusi sebuah blok perintah selama kondisi tertentu bernilai benar (true). Struktur ini disebut juga sebagai loop berbasis kondisi awal (entry-controlled loop), karena kondisi diperiksa terlebih dahulu sebelum perintah di dalam blok dijalankan.

Jika kondisi bernilai true, maka program akan menjalankan isi perulangan, kemudian kembali memeriksa kondisi tersebut. Proses ini akan terus berulang sampai kondisi menjadi false. Jika sejak awal kondisi awal bernilai false, maka perulangan tidak akan dijalankan sama sekali.

Struktur umum while dalam java:

```
While (kondisi) {
    // blok perintah yang akan dijalankan
}
```

2.1.3 Perulangan Do-While

Perulangan do-while merupakan salah satu bentuk loop dalam bahasa pemrograman java yang digunakan untuk menjalankan suatu blok perintah setidaknya satu kali, kemudian mengulangi eksekusi tersebut selama kondisi masih bernilai benar (true).

Berbeda dengan while, perulangan do-while mengevaluasi kondisi di akhir perulangan terlebih dahulu, baru kemudian memeriksa apakah kondisi masih terpenuhi. Karena itu, walaupun kondisi bernilai salah sejak awal, blok perintah di dalam do-while tetap akan dijalankan minimal satu kali.

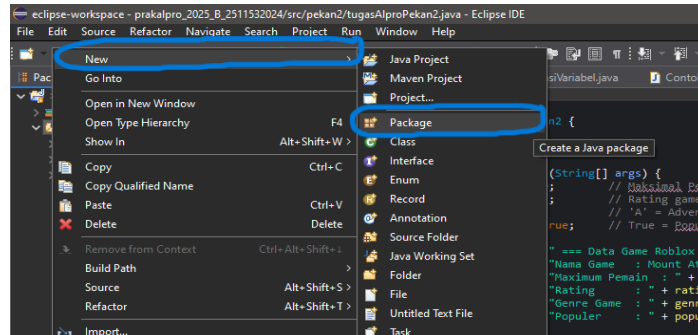
Struktur umum do-while dalam java:

```
Do {
    // blok perintah yang akan dijalankan
} while (kondisi);
```

2.2 Langkah Kerja

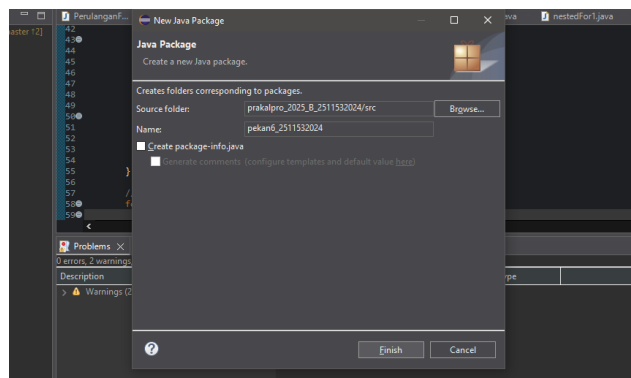
2.2.1 Pembuatan *Package* dan *Class* Pekan 6

1. Buka aplikasi eclipse, lalu buat package baru pada src klik kanan, lalu pilih “New” dan klik tulisan Package.



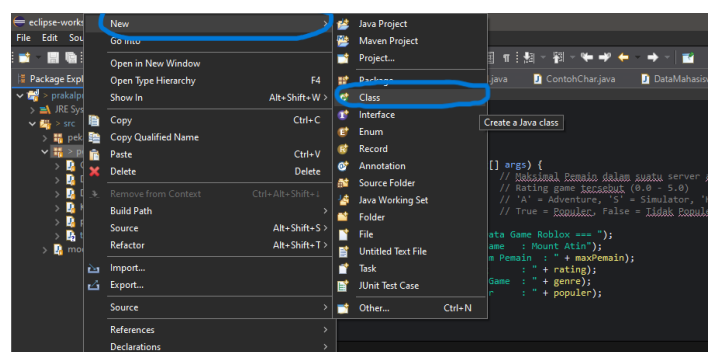
2.2.1 pembuatan package

2. Kemudian buat nama packagenya tanpa spasi, huruf kapital, karakter khusus. Beri nama “pekan6_2511532024”. Lalu klik finish.



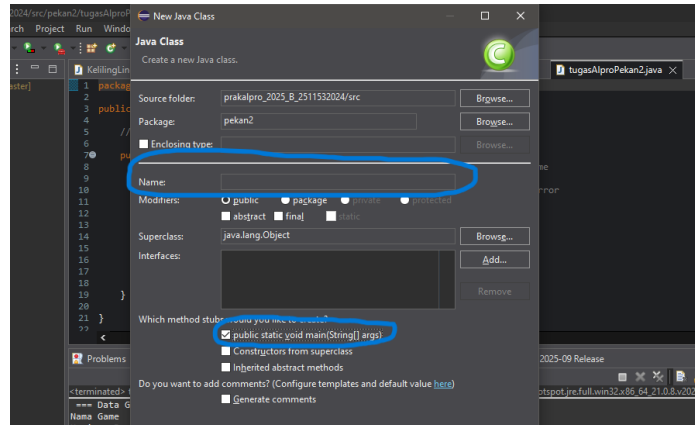
2.2.2 pembuatan package

3. Ketika package sudah jadi. Klik kanan pada bagian “pekan6_2511532024”. Lalu klik “new” dan pilih bagian “Class” untuk memulai membuat program.



2.2.3 pembuatan class

4. Buat nama class program yang akan dibuat pada bagian nama spasi dan menggunakan huruf kapital pada awal kata, lalu klik bagian “public static void main(string[]args)”. Kemudian klik “finish”.



2.2.4 pembuatan class

2.2.2 Program Pertama: PerulanganWhile1_2511532024

1. Masukkan sintaks berikut.

```

1 package pekan_2511532024;
2
3 import java.util.Scanner;
4
5 public class perulanganWhile1_2511532024 {
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            //jika jawab = tidak maka 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 }

```

2.2.5 sintaks perulangan while 1

2. Lalu klik bagian “run” untuk melihat hasil sintaks yang di inputkan. Lalu akan menghasilkan output seperti berikut.

```

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

```

2.2.6 running kode dan output sintaks

2.2.3 Program Kedua: Lempardadu_2511532024

1. Ulangi pembuatan class seperti program sebelumnya dan beri nama class tersebut “Lempardadu_2511532024”. Lalu masukkan sintaks seperti berikut.



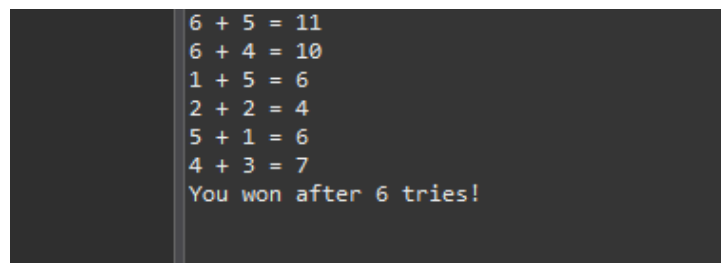
```

1 package pekan6_2511532024;
2
3 import java.util.Random;
4
5 public class Lempardadu_2511532024 {
6
7     public static void main(String[] args) {
8         Random rand = new Random();
9         int tries = 0;
10        int sum = 0;
11        while (sum != 7) {
12            // roll the dice once
13            int dadu1 = rand.nextInt(6) + 1;
14            int dadu2 = rand.nextInt(6) + 1;
15            sum = dadu1 + dadu2;
16            System.out.println(dadu1 + " + " + dadu2 + " = " + sum);
17            tries++;
18        }
19        System.out.println("You won after " + tries + " tries!");
20    }
21 }
22

```

2.2.7 sintaks lempar dadu

2. Klik “run” untuk melihat hasil sintaks yang sudah dibuat.



```

6 + 5 = 11
6 + 4 = 10
1 + 5 = 6
2 + 2 = 4
5 + 1 = 6
4 + 3 = 7
You won after 6 tries!

```

2.2.8 running kode dan output sintaks

2.2.4 Program Ketiga: GamePenjumlahan_2511532024

1. Ulangi pembuatan class seperti program sebelumnya dan beri nama class tersebut “GamePenjumlahan_2511532024”. Lalu masukkan sintaks seperti berikut.



```

1 package pekan6_2511532024;
2
3 import java.util.Random;
4 import java.util.Scanner;
5
6 public class GamePenjumlahan_2511532024 {
7
8     public static void main(String[] args) {
9         Scanner console = new Scanner(System.in);
10        Random rand = new Random();
11        //play until user gets 3 wrong
12        int points = 0;
13        int wrong = 0;
14        while (wrong < 3) {
15            int result = play(console, rand); // play one game
16            if (result > 0) {
17                points++;
18            } else {
19                wrong++;
20            }
21        }
22        System.out.println("You earned " + points + " total points.");
23    }
24

```

2.2.9 sintaks game penjumlahan

```

24 // membuat array penjumlahan dan memanggil ke user
25 public static int play(Scanner console, Random rand) {
26     // print the operands being added, and sum them
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     // read user's guess and report whether it was correct
38     int guess = console.nextInt();
39     if (guess == sum) {
40         return 1;
41     } else {
42         System.out.println("Wrong! The answer was " + sum);
43         return 0;
44     }
45 }
46 }

```

2.2.10 sintaks game penjumlahan

- Klik “run” untuk melihat hasil sintaks yang sudah dibuat.

```

6 + 3 + 3 + 7 = 19
5 + 3 + 2 + 9 + 10 = 29
10 + 7 = 17
6 + 7 + 9 = 22
4 + 8 + 2 + 1 + 9 = 24
3 + 3 + 3 = 9
8 + 5 + 9 = 21
Wrong! The answer was 22
9 + 7 + 5 + 6 = 27
4 + 6 = 10
3 + 4 + 6 + 9 = 22
8 + 8 + 8 + 8 + 6 = 1
Wrong! The answer was 38
4 + 4 = 4
Wrong! The answer was 8
You earned 9 total points.

```

2.2.11 running kode dan output sintaks

2.2.5 Program Keempat: SentinelLoop_2511532024

- Ulangi pembuatan Class seperti program sebelumnya dan beri nama Class tersebut “SentinelLoop_2511532024”.
Lalu masukkan sintaks seperti berikut.

```

1 package pekan6_2511532024;
2
3 import java.util.Scanner;
4
5 public class SentinelLoop_2511532024 {
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
19 }

```

2.2.12 sintaks sentinel loop

- Klik “run” untuk melihat hasil sintaks yang sudah dibuat.

```

masukkan angka (0 untuk keluar): 12
masukkan angka (0 untuk keluar): 5
masukkan angka (0 untuk keluar): 67
masukkan angka (0 untuk keluar): 10
masukkan angka (0 untuk keluar): 0
totalnya adalah 94

```

2.2.13 running kode dan output sintaks

2.2.6 Program Kelima: doWhile1_2511532024

1. Ulangi pembuatan Class seperti program sebelumnya dan beri nama “doWhile_2511532024”. Lalu masukkan sintaks seperti berikut.

```

1 package pekan6_2511532024;
2
3 import java.util.Scanner;
4
5 public class doWhile1_2511532024 {
6
7     public static void main(String[] args) {
8         Scanner console = new Scanner(System.in);
9         String phrase;
10        do {
11            System.out.print("Input Password: ");
12            phrase = console.next();
13        } while (!phrase.equals("abcd"));
14    }
15 }
16
17 }
18

```

2.2.14 sintaks do while 1

2. klik “run” untuk melihat hasil sintaks yang sudah dibuat.

```

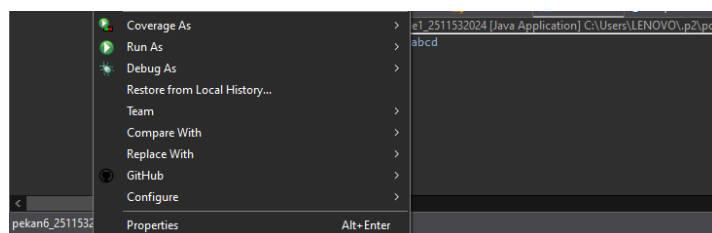
<terminated> doWhile1_2511532024 [Java Application] C:\Use
Input Password: abcd

```

2.2.15 runing kode dan output sintaks

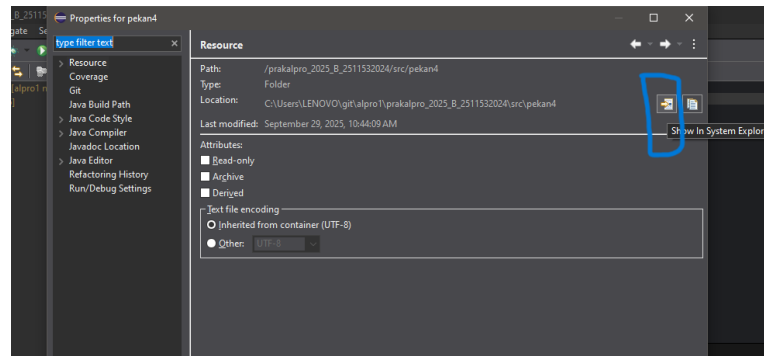
2.2.7 Commit dan Push ke GitHub

1. Klik kanan pada bagian “pekan6_2511532024”. Kemudian pilih bagian “properties”.



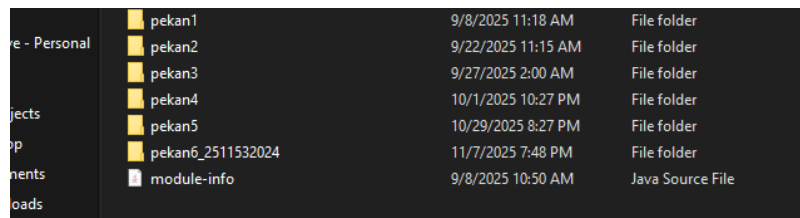
2.2.16 Proses commit dan push ke GitHub

2. Setelah itu keluar opsi properties. Kemudian klik “Show In System Explorer”.



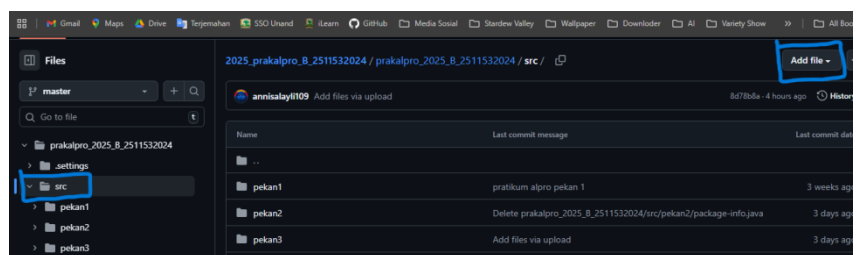
2.2.17 Proses commit dan push ke GitHub

3. Setelah itu akan terbuka folder file tersebut di file explorer.



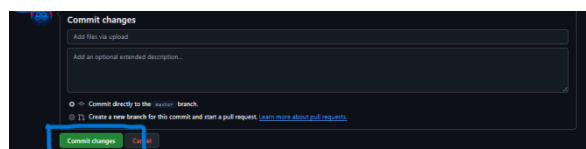
2.2.18 Proses commit dan push ke GitHub

4. Kemudian buka akun GitHub dan buka pada bagian src. Dan klik pada bagian “add file”. Dan pilih “upload file”.



2.2.19 Proses commit dan push ke GitHub

5. Setelah itu tarik file “pekan6_2511532024” dari file explorer ke bagian upload file dan kemudian tunggu loading file diunggah ke GitHub. Dan setelah itu klik “Commit changes”.



2.2.20 Proses commit dan push ke GitHub

6. Kemudian cek akun GitHub untuk melihat apakah file yang diupload sudah masuk atau belum. Jika sudah maka upload sudah selesai.



2.2.21 Proses commit dan push ke GitHub

BAB III

PENUTUP

3.1 Kesimpulan

Perulangan `while` dan `do-while` sama-sama digunakan untuk menjalankan perintah secara berulang selama kondisi bernilai benar. Perbedaan utamanya, `while` memeriksa kondisi sebelum menjalankan perintah, sedangkan `do-while` memeriksa setelah menjalankan perintah, sehingga selalu dijalankan minimal satu kali. Keduanya membantu membuat program lebih efisien, terutama pada proses yang memerlukan pengulangan dengan kondisi tertentu.

3.2 Saran

Dalam penggunaan perulangan `while` dan `do-while`, sebaiknya kita memahami dengan baik alur logika dan kondisi yang digunakan agar tidak terjadi kesalahan seperti `infinite loop`. Gunakan `while` ketika perlu melakukan pengecekan kondisi sebelum menjalankan perintah, sedangkan `do-while` lebih tepat digunakan saat perintah harus dijalankan minimal satu kali sebelum kondisi diperiksa. Selain itu, pastikan selalu ada perubahan pada variabel kondisi agar perulangan dapat berhenti dengan benar. Dengan terus berlatih dan mencoba berbagai kasus, pemahaman terhadap konsep perulangan ini akan semakin kuat dan dapat diterapkan secara efektif dalam pembuatan program yang interaktif dan efisien.

DAFTAR PUSTAKA

- [1] Oracle, “The while and do-while Statements,” *Java Documentation*, 2024.
[Online]. Available:
<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/while.html>
- [2] Oracle, “The Java™ Tutorials – Control Flow Statements,” *Oracle Java Documentation*, 2024. [Online]. Available:
<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/>