

LAPORAN PRAKTIKUM ALGORITMA DAN PEMROGAMAN

PEKAN 6

Disusun Oleh:

IHSANUL ZAKY EL-MUHAMMADY

NIM:2511533001

DOSEN PENGAMPU: Dr.WAHYUDI S.T , M.T

ASISTEN LABORATORIUM: AUFAN TAUFIQUR RAHMAN



DEPARTEMEN INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

UNIVERSITAS ANDALAS

PADANG

2025

KATA PENGANTAR

Dengan penuh rasa syukur, saya mengucapkan terima kasih kepada Tuhan Yang Maha Esa, yang telah memberikan kemudahan dan kesempatan sehingga laporan praktikum ini dapat selesai dengan baik.

Saya juga ingin menyampaikan penghargaan yang setinggi-tingginya kepada Dr. Wahyudi, S.T., M.T. selaku dosen pengampu Mata Kuliah Algoritma dan Pemrograman, yang telah memberikan bimbingan dan motivasi selama perkuliahan. Ucapan terima kasih juga saya sampaikan kepada Uda Aufan Tafiqurrahman, asisten laboratorium, yang dengan sabar dan penuh perhatian telah membimbing kami dalam praktikum ini.

Laporan ini disusun sebagai salah satu tugas dalam Mata Kuliah Pemrograman Dasar.

Saya menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, segala masukan dan saran yang membangun sangat saya harapkan untuk perbaikan di masa mendatang.

Padang, 2025

IHSANUL ZAKY EL-MUHAMMADY

DAFTAR ISI

KATA PENGANTAR
DAFTAR ISI.....
BAB I PENDAHULUAN
1.1 Latar Belakang
1.2 Tujuan.....
1.3 Manfaat.....
BAB II PEMBAHASAN
2.1 Program Perulangan while 1
2.2 Program Lempar Dadu
2.3 Program Game Penjumlahan
2.4 Program Sentinel Loop.....
2.5 Program do while 1
BAB III KESIMPULAN
3.1 Kesimpulan
3.2 Saran.....
DAFTAR PUSTAKA.....

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Dalam pemrograman, perulangan atau looping digunakan untuk menjalankan perintah secara berulang sampai kondisi tertentu terpenuhi. Di Java, salah satu bentuk perulangan adalah while dan do-while, yang cocok digunakan ketika jumlah pengulangan belum diketahui sejak awal dan bergantung pada kondisi logis.

Perulangan while akan mengecek kondisi terlebih dahulu sebelum menjalankan perintah. Jika kondisi true, perintah dijalankan terus hingga kondisi menjadi false. Sementara itu, do-while akan menjalankan perintah minimal satu kali sebelum memeriksa kondisi. Dengan memahami perbedaan keduanya, programmer dapat menulis kode yang lebih efisien dan sesuai kebutuhan logika program.

1.2 Tujuan

1. Memahami konsep dasar serta mekanisme kerja perulangan while dan do-while.
2. Mengidentifikasi perbedaan antara struktur while dan do-while melalui contoh program.
3. Mengembangkan kemampuan logika pemrograman menggunakan struktur perulangan while dan do-while.

1.3 Manfaat

1. Memperluas pemahaman mengenai struktur perulangan **while** dan **do-while** dalam bahasa Java.
2. Meningkatkan keterampilan dalam menerapkan logika pemrograman untuk menyelesaikan berbagai permasalahan.
3. Menjadi landasan dalam mempelajari konsep pengendalian alur program (**control flow**) yang lebih kompleks.

BAB 2

PEMBAHASAN

2.1 Program Perulangan while 1

1. int counter = 0;
Menginisialisasi variabel penghitung dengan nilai awal 0.
2. boolean running = true;
Membuat variabel flag bernama running yang bernilai true, agar perulangan dapat berjalan.
3. Scanner scan = new Scanner(System.in);
Menyiapkan objek Scanner untuk membaca input dari keyboard.
4. while (running) {
Memulai perulangan yang akan terus berjalan selama running bernilai true.
 - counter++;
Menambah nilai counter sebesar 1 setiap kali perulangan dimulai. Pada iterasi pertama, nilai berubah dari 0 menjadi 1.
 - System.out.println("Jumlah = " + counter);
Menampilkan nilai counter saat ini.
 - System.out.print("Apakah lanjut (ya/tidak) ");
Mencetak pertanyaan tanpa pindah baris untuk menerima jawaban pengguna.
 - jawab = scan.nextLine();
Membaca input pengguna (misal: "ya" atau "tidak").
 - if (jawab.equalsIgnoreCase("tidak")) { running = false; }
Jika pengguna mengetik "tidak" (tanpa memperhatikan huruf besar/kecil), maka running diubah menjadi false, sehingga perulangan berhenti.
5. Setelah perulangan selesai,
System.out.println("Anda sudah melakukan perulangan sebanyak " + counter + " kali");
Program menampilkan jumlah total perulangan yang telah dilakukan.

The screenshot shows the Eclipse IDE interface. In the top editor pane, there is a Java file named 'perulanganWhile1_2511533001.java' with the following code:

```

7     String jawab;
8     boolean running = true;
9     //deklarasi scanner
10    Scanner scan = new Scanner(System.in);
11    while (running) {
12        counter++;
13        System.out.println ("Jumlah = "+counter);
14        System.out.print("Apakah lanjut (ya / tidak?)");
15        jawab= scan.nextLine ();
16        //cek jawab = tidak, perulangan berhenti
17        if (jawab.equalsIgnoreCase("tidak")) {
18            running= false;
19        }
20    }

```

In the bottom terminal window, the program's output is shown:

```

<terminated> perulanganWhile1_2511533001 [Java Application] C:\Users\FA506NCG\p2\pool\plugins\
Jumlah = 1
Apakah lanjut (ya / tidak?)ya
Jumlah = 2
Apakah lanjut (ya / tidak?)tidak
Anda sudah melakukan perulangan sebanyak 2 kali|

```

2.2 Program Lempar Dadu

1. Random rand = new Random();

Membuat objek rand untuk menghasilkan angka acak antara 1 sampai 6.

2. int tries = 0; int sum = 0;

Menginisialisasi variabel tries sebagai penghitung jumlah percobaan, dan sum untuk menyimpan total hasil dua dadu.

3. while (sum != 7)

Memulai perulangan yang akan terus berjalan selama jumlah dua dadu belum bernilai 7.

4. Di dalam perulangan:

- int dadu1 = rand.nextInt(6) + 1;

Menghasilkan angka acak 1–6 untuk dadu pertama.

- int dadu2 = rand.nextInt(6) + 1;

Menghasilkan angka acak 1–6 untuk dadu kedua.

- sum = dadu1 + dadu2;

Menjumlahkan nilai kedua dadu.

- System.out.println(dadu1 + "+" + dadu2 + "=" + sum);

Menampilkan hasil setiap lemparan.

- tries++;

Menambah jumlah percobaan setiap kali loop dijalankan.

5. Ketika sum bernilai 7, kondisi while menjadi false sehingga perulangan berhenti.

6. System.out.println("You won after " + tries + " tries!");

Menampilkan informasi jumlah percobaan yang dibutuhkan sampai hasil lemparan mencapai angka 7.

```

perulanganWhile1_2511533001.java Lempardadu_2511533001.java
er] 9
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
21 }
22

```

Problems @ Javadoc Declaration Console Terminal 1 Git Staging Install Java 25 Superuser

<terminated> Lempardadu_2511533001 [Java Application] C:\Users\FA506NCG\p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jvm算了

4 + 1 = 5
2 + 1 = 3
4 + 3 = 7
You won after3 tries!

2.3 Program Game Penjumlahan

1. Scanner console = new Scanner(System.in); dan Random rand = new Random();
Membuat objek untuk membaca input dari pengguna serta menghasilkan angka acak.
2. int points = 0; int wrong = 0;
Menginisialisasi variabel points untuk menghitung jumlah jawaban benar dan wrong untuk menghitung jumlah jawaban salah.
3. while (wrong < 3)
Memulai perulangan utama, di mana permainan terus berjalan selama jumlah kesalahan masih kurang dari tiga.
4. Di dalam perulangan:
 - o Program memanggil metode play(console, rand) untuk menampilkan soal dan menerima jawaban dari pengguna.
 - o Jika nilai yang dikembalikan lebih dari 0 (jawaban benar), maka nilai points bertambah satu.
 - o Jika nilai yang dikembalikan adalah 0 (jawaban salah), maka variabel wrong bertambah satu.
5. Setelah pengguna melakukan tiga kesalahan, perulangan berhenti.
6. System.out.println("You earned " + points + " total points.");
Menampilkan total poin yang berhasil diperoleh selama permainan berlangsung.
7. Metode play() bertugas membuat soal penjumlahan acak:
 - Menentukan jumlah bilangan yang akan dijumlahkan, antara 2 hingga 5.
 - Menghasilkan angka acak antara 1–10, menampilkannya, dan menghitung totalnya.
 - Meminta pengguna memasukkan hasil penjumlahan.

- Jika jawaban benar, metode mengembalikan 1; jika salah, menampilkan jawaban yang benar dan mengembalikan 0.

The screenshot shows the Eclipse IDE interface with several tabs open at the top: 'perulanganWh...', 'Lempardadu...', 'GamePenjumlah...', and 'SentinelLoo...'. The code in the editor is:

```

13    while (wrong < 3) {
14        int result = play(console, rand); // play one game
15        if (result > 0) {
16            points++;
17        } else {
18            wrong++;
19        }
20    }
21    System.out.println("You earned " + points + " total points.");
22 }
23 // membuat soal penjumlahan dan ditampilkan ke user
24 public static int play(Scanner console, Random rand) {
25     //print the operands being added, and sum them
26     int operands = rand.nextInt(4) + 2;

```

In the 'Console' tab below, the output is:

```

GamePenjumlahan_2511533001 [Java Application] C:\Users\FA506NCG\p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.j
9 +8 +2 +10 = 20
Wrong! The answer was29
2 +3 +10 +8 +5 =

```

2.4 Program Sentinel Loop

1.Scanner console = new Scanner(System.in);

Membuat objek Scanner untuk membaca input dari keyboard.

2.int sum = 0; int number = 12;

Menginisialisasi variabel sum sebagai penampung total penjumlahan, dan number diberi nilai awal agar perulangan dapat mulai berjalan.

3.while (number != 0)

Memulai perulangan yang akan terus berlangsung selama nilai number tidak sama dengan 0.

4.Di dalam perulangan:

- Program menampilkan pesan "Masukan angka (0 untuk keluar): " untuk meminta input angka dari pengguna.
- Nilai yang dimasukkan pengguna dibaca dan disimpan ke variabel number.
- Nilai tersebut kemudian ditambahkan ke variabel sum.

5.Jika pengguna memasukkan nilai 0, kondisi number != 0 menjadi false sehingga perulangan berhenti.

6.Setelah keluar dari perulangan, program menampilkan total penjumlahan semua angka yang dimasukkan pengguna melalui perintah:

System.out.println("totalnya adalah " + sum);

The screenshot shows the Eclipse IDE interface. In the top bar, there are several tabs: Lempardadu_..., GamePenjumlahan_..., SentinelLoop_..., and doWhile1_25... . Below the tabs, the code editor displays the following Java code:

```
1 package Pekan6_2511533001;
2 import java.util.Scanner;
3 public class SentinelLoop_2511533001 {
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.println("Masukkan angka (0 untuk keluar): ");
12        number = console.nextInt();
13        sum = sum + number;
14    }
15}
```

Below the code editor, the terminal window shows the output of the program:

```
SentinelLoop_2511533001 [Java Application] C:\Users\FA506NCG\p2\pool\plugins\org.eclipse.jdt.core\hot
Masukkan angka (0 untuk keluar):
```

2.5 Program do while 1

1. Scanner console = new Scanner(System.in);
Membuat objek Scanner untuk membaca input dari keyboard.
2. String phrase;
Mendeklarasikan variabel phrase untuk menyimpan input yang dimasukkan pengguna.
3. do { ... } while (!phrase.equals("abcd"));
Memulai struktur perulangan do-while, yaitu perulangan yang selalu dieksekusi minimal satu kali sebelum pengecekan kondisi dilakukan.
4. Di dalam blok do:
 - Program menampilkan pesan "Input Password: " untuk meminta pengguna memasukkan password.
 - Input yang diberikan pengguna dibaca dan disimpan ke variabel phrase melalui perintah phrase = console.nextInt();
5. Setelah input dimasukkan, program mengecek kondisi !phrase.equals("abcd"):
 - Jika password tidak sama dengan "abcd", kondisi bernilai true dan perulangan dijalankan kembali.
 - Jika password sesuai "abcd", kondisi bernilai false sehingga perulangan berhenti.
6. Karena tidak ada perintah lanjutan setelah perulangan, program selesai secara otomatis ketika pengguna berhasil memasukkan password yang benar.

The screenshot shows the Eclipse IDE interface with several tabs at the top: Lempardadu_..., GamePenjuml..., SentinelLoo..., doWhile1_25..., and a blank tab. Below the tabs is a code editor containing the following Java code:

```
1 package Pekan6_2511533001;
2
3 import java.util.Scanner;
4
5 public class doWhile1_2511533001 {
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       console.close();
15   }
16 }
```

Below the code editor is a toolbar with icons for Problems, Javadoc, Declaration, Console, Terminal, and Git Status. The 'Console' tab is selected. The console output window displays the following text:

```
doWhile1_2511533001 [Java Application] C:\Users\FA506NCG\p2\pool\plugins\org.eclipse.justj.openjdk.
Input Password: Zaky keren
Input Password: Input Password: Zaky keren
Input Password: Input Password: Zaky keren
Input Password: Input Password:
```

BAB 3

KESIMPULAN

3.1 Kesimpulan

Berdasarkan percobaan yang dilakukan, dapat disimpulkan bahwa perulangan while dan do-while sama-sama digunakan untuk mengeksekusi kode secara berulang selama kondisi bernilai benar. Perbedaan utamanya terdapat pada proses pengecekan kondisi. Pada perulangan while, kondisi dicek sebelum kode dijalankan sehingga jika kondisi awal salah, perulangan tidak berjalan sama sekali. Sebaliknya, pada do-while, pengecekan kondisi dilakukan setelah eksekusi kode, sehingga perintah di dalamnya selalu dijalankan minimal satu kali, meskipun kondisi awal tidak terpenuhi. Dengan demikian, struktur do-while lebih sesuai digunakan ketika program harus menjalankan suatu proses terlebih dahulu sebelum memvalidasi kondisi, seperti dalam penerimaan input atau pengecekan password.

3.2 Saran

Sebaiknya dosen menyelenggarakan sesi pra-praktikum di kelas sebagai pengantar materi, agar mahasiswa memperoleh pemahaman awal yang lebih baik. Dengan demikian, potensi kepanikan atau kesalahan selama praktikum dapat diminimalkan. Disarankan agar materi praktikum dibagikan terlebih dahulu melalui platform iLearn, sehingga mahasiswa memiliki waktu yang cukup untuk mempelajari dan mempersiapkan diri sebelum pelaksanaan praktikum.

DAFTAR PUSTAKA

Oracle. (2025). The while and do-while Statements (Java Tutorials). Oracle Documentation.
Diakses dari <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/while.html>