

LAPORAN PRATIKUM
ALGORITMA PEMROGRAMAN
“PERULANGAN WHILE DAN DO WHILE”

DISUSUN OLEH:

NABIL FIKRI

2511533011

DOSEN PENGAMPU:

Dr. WAHYUDI, S.T, M.T

ASISTEN PRAKTIKUM:

JOVANTRI IMMANUEL GULO



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

2025

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT yang telah memberikan kelimpahan rahmat dan hidayahnya yaitu kesehatan dan kekuatan , atas rahmat tersebut penulis dapat mengerjakan laporan pratikum pekan ke-6 ini dengan tepat waktu. Laporan ini disusun sebagai tugas mingguan mata kuliah Pratikum Algoritma Pemrograman yaitu dengan topik “Perulangan while dan do-while” pada Java.

Melalui laporan ini, penulis bertujuan untuk mempelajari serta memahami konsep pada perulangan while dan do-while dalam bahasa Java, khususnya penggunaan struktur while dan do-while. Kedua jenis perulangan tersebut memiliki peranan penting dalam pengembangan program yang efisien, terstruktur, dan mudah dikelola. Pratikum ini juga diharapkan dapat membantu mahasiswa dalam mengasah kemampuan logika serta pemahaman terhadap alur eksekusi program.

Penulis menyadari laporan ini masih jauh dari sempurna, oleh karena itu kritik dan saran sangat diharapkan demi perbaikan ke depannya. Semoga laporan ini dapat bermanfaat bagi penulis maupun pembaca.

DAFTAR ISI

KATA PENGANTAR.....	i
DAFTAR ISI	ii
BAB I PENDAHULUAN	1
1.2 Tujuan Pratikum.....	1
1.3 Manfaat Pratikum.....	1
BAB II PEMBAHASAAN.....	3
2.1 Pengertian While dan Do-While	3
2.2 Struktur Perulangan While	3
2.3 Kode Program	4
2.3.1 doWhile1_2511533011	4
2.3.2 GamePenjumlahan_2511533011	5
2.3.3 LemparDadu_2511533011	6
2.3.4 PerulanganWhile1_2511533011	7
2.3.5 SentinelLoop_2511533011	8
BAB III PENUTUP	10
3.1 Kesimpulan.....	10
3.2 Saran.....	10
DAFTAR PUSTAKA.....	11

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perulangan while adalah salah satu konsep dasar dalam pemrograman yang memungkinkan program untuk mengulangi serangkaian pernyataan selama kondisi tertentu terpenuhi. Dalam bahasa pemrograman Java, perulangan while digunakan untuk mengulangi tugas-tugas selama kondisi yang ditentukan tetap true. Pada laporan ini mahasiswa akan membahas pengertian dan struktur perulangan while dalam bahasa pemrograman Java, serta memberikan contoh penggunaan dan outputnya.

Dengan menggunakan perulangan while, pengguna dapat mengulangi serangkaian pernyataan selama kondisi yang ditentukan tetap true, sehingga memungkinkan Pengguna untuk melakukan tugas-tugas yang memerlukan pengulangan tanpa batas waktu tertentu. Pastikan untuk memastikan bahwa kondisi pada perulangan while berubah seiring waktu untuk menghindari perulangan tak terbatas.

1.2 Tujuan Pratikum

1. Memberikan mahasiswa pengetahuan baru tentang penggunaan perulangan while dan do-while.
2. Memahami penggunaan perulangan while di berbagai situasi pada program Java, serta penulisan sintaks, kelas, dan metode utama.
3. Melatih keterampilan mahasiswa dalam menulis, mengompilasi, dan mengeksekusi program sederhana menggunakan perulangan while.
4. Memberikan pemahaman yang matang dengan memberikan mahasiswa Latihan menggunakan program perulangan while.

1.3 Manfaat Pratikum

1. Mahasiswa dapat memahami konsep perulangan while dan do-while secara praktis, bukan hanya teori.
2. Melatih keterampilan logika dan analisis dalam menyusun serta mengeksekusi kode.

3. Memberikan pengalaman langsung dalam menggunakan bahasa Java sebagai salah satu bahasa pemrograman populer.
4. Menjadi bekal awal untuk memahami materi pemrograman yang lebih kompleks pada praktikum berikutnya.
5. Membiasakan mahasiswa dengan proses debugging dan pemecahan masalah dalam pemrograman.

BAB II

PEMBAHASAAN

2.1 Pengertian While dan Do-While

Perulangan while digunakan ketika Anda ingin menjalankan serangkaian pernyataan selama kondisi tertentu terpenuhi. Jika kondisi awal bernilai true, blok pernyataan dalam while akan dieksekusi. Setelah setiap iterasi, kondisi akan dievaluasi lagi, dan jika masih true, perulangan akan terus berlanjut. Perulangan ini akan berhenti ketika kondisi menjadi false.

Sementara itu, Perulangan Do While merupakan modifikasi dari perulangan While, yakni dengan memindahkan posisi pemeriksaan kondisi ke akhir perulangan. Artinya, lakukan dahulu sebuah perulangan, baru periksa apakah kondisi *variabel counter* sudah terpenuhi atau belum di akhir perulangan.

2.2 Struktur Perulangan While

Berikut adalah bentuk dasar dari perulangan while dalam bahasa pemrograman Java:

while (kondisi) {
// Blok pernyataan yang akan diulang
}

- kondisi adalah ekspresi atau kondisi yang dievaluasi. Selama kondisi ini bernilai true, perulangan akan terus berlanjut; begitu kondisi ini menjadi false, perulangan akan berhenti.
- Blok pernyataan dalam while adalah tugas-tugas yang akan diulang selama kondisi tetap true.

2.3 Kode Program

2.3.1 doWhile1_2511533011

```

1 package pekan6_2511533011;
2 import java.util.Scanner;
3 public class doWhile_2511533011 {
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        } while (!phrase.equals("abcd"));
12    }
13 }
14

```

Kode Program 2.3.1

Program ini merupakan program dasar dari pengulangan do-while dengan meminta output sederhana yaitu pengguna harus memasukkan *password* dan terus meminta input ulang sampai *password* yang dimasukkan benar. Jika sesuai program maka *password* yang benar yaitu “abcd”. Penggunaan doWhile dengan do yang artinya blok kode didalam do akan dijalankan setidaknya sekali, lalu kondisi while berfungsi untuk menentukan apakah perulangan harus dilanjutkan atau tidak. Selain itu, *String phrase* berfungsi untuk menyimpan input password dan *!phrase.equals* berfungsi untuk menyimpan password yang benar selama password yang diinput belum sesuai dengan yang diminta.

Langkah Pembuatan :

1. Import library java.util.Scanner untuk membaca input dari pengguna.
2. Buat objek Scanner console = new Scanner(System.in);.
3. Deklarasikan variabel String phrase; untuk menampung input.
4. Gunakan struktur do-while agar perintah dijalankan minimal satu kali.
5. Di dalam blok do, tampilkan pesan “Input Password:” dan baca input dari user.
6. Kondisikan while (!phrase.equals("abcd")); agar loop berhenti hanya jika password yang dimasukkan benar.

2.3.2 GamePenjumlahan_2511533011

```

1 package pekan6_2511533011;
2 import java.util.Random;
3
4 public class GamePenjumlahan_2511533011 {
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
29        for (int i = 2; i <= operands; i++) {
30            int n = rand.nextInt(10) + 1;
31            sum += n;
32            System.out.print(" + " + n);
33        }
34        System.out.print(" = ");
35
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 }

```

Kode Program 2.3.2

Program ini adalah kode program sebuah game sederhana penjumlahan. Pengguna akan diberi soal penjumlahan dengan jumlah angka (operand) yang acak (antara 2 hingga 5), dan setiap angkanya juga acak (1-10). Jika pengguna menjawab benar, ia mendapat poin. Jika salah, ia mendapat kesalahan. Permainan berhenti setelah pengguna melakukan 3 kesalahan, lalu menampilkan total poin yang didapat.

Langkah Pembuatan :

1. Deklarasikan Scanner dan Random untuk input dan angka acak.
2. Buat variabel points dan wrong untuk menghitung skor dan kesalahan.

3. Gunakan perulangan while ($wrong < 3$) agar permainan berlanjut selama salah kurang dari 3 kali.
4. Di dalam loop, panggil metode play() untuk menampilkan soal dan memeriksa jawaban.
5. Tambah poin jika benar, tambah kesalahan jika salah.
6. Setelah keluar dari loop, tampilkan total poin yang diperoleh
7. Dalam metode play(), buat angka acak, tampilkan soal, baca jawaban pengguna, lalu kembalikan 1 (benar) atau 0 (salah).

2.3.3 LemparDadu_2511533011

```

1 package pekan6_2511533011;
2 import java.util.Random;
3
4 public class LemparDadu_2511533011 {
5
6     public static void main(String[] args) {
7         Random rand = new Random();
8         int tries = 0;
9         int sum = 0;
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 }
23

```

Kode Program 2.3.3

Program ini mensimulasikan permainan melempar dua dadu. Tujuannya adalah terus melempar dadu sampai jumlah kedua dadu sama dengan 7. Setiap kali melempar, program akan mencetak hasil lemparan dan jumlahnya. Ketika akhirnya mendapatkan jumlah 7, program akan memberi tahu berapa kali percobaan (tries) yang dibutuhkan untuk menang.

Langkah Pembuatan :

1. Import library java.util.Random untuk menghasilkan angka acak.
2. Buat objek Random rand = new Random(); agar bisa digunakan untuk lempar dadu.

3. Inisialisasi variabel tries untuk menghitung jumlah percobaan dan sum untuk menyimpan hasil penjumlahan dadu.
4. Gunakan perulangan while ($\text{sum} \neq 7$) agar program terus berjalan sampai hasil dadu berjumlah 7.
5. Di dalam loop, buat dua angka acak dadu1 dan dadu2 dengan $\text{rand.nextInt}(6) + 1$.
6. Jumlahkan kedua dadu, tampilkan hasilnya, lalu tambahkan nilai tries.
7. Setelah keluar dari loop, cetak pesan kemenangan dengan jumlah percobaan yang dilakukan.

2.3.4 PerulanganWhile1_2511533011

```

1 package pekang_2511533011;
2 import java.util.Scanner;
3 public class PerulanganWhile1_2511533011 {
4
5     public static void main(String[] args) {
6         int counter = 0;
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.println("Apakah lanjut (ya / tidak?)");
15            jawab = scan.nextLine();
16            //cek jawab = tidak, perulangan berhenti
17            if (jawab.equalsIgnoreCase("tidak")) {
18                running = false;
19            }
20        }
21        System.out.println("Anda sudah melakukan perulangan sebanyak "+counter+" kali" );
22    }
23 }
24 }
25

```

Kode Program 2.3.4

Program ini membuat perulangan sederhana yang bisa dihentikan oleh penggunaanya. Setiap kali perulangan berjalan, program akan:

- Menampilkan jumlah iterasi (counter),
- Bertanya apakah ingin melanjutkan (ya atau tidak),
- Jika pengguna menjawab "tidak" (tidak peduli huruf besar/kecil), perulangan berhenti.
- Setelah berhenti, program menampilkan total jumlah perulangan yang telah dilakukan.

Langkah Pembuatan :

1. Deklarasikan variabel counter, jawab, dan running.
2. Buat objek Scanner untuk membaca input dari pengguna.
3. Gunakan **perulangan while** dengan kondisi `running == true`.
4. Di dalam perulangan, tampilkan nilai counter dan minta input “ya” atau “tidak”.
5. Jika pengguna menjawab “tidak”, ubah nilai running menjadi false agar perulangan berhenti.
6. Setelah keluar dari perulangan, tampilkan jumlah total perulangan yang telah dilakukan.

2.3.5 SentinelLoop_2511533011

```

1 package pekan6_2511533011;
2 import java.util.Scanner;
3 public class SentinelLoop_2511533011 {
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

```

Kode Program 2.3.5

Program ini adalah contoh klasik Sentinel Loop yaitu perulangan yang terus berjalan sampai pengguna memasukkan nilai "sentinel" (penanda). Dalam kasus ini, nilai sentinel-nya adalah 0. Tujuannya yaitu membaca sejumlah angka dari pengguna, lalu menjumlahkannya, dan berhenti saat pengguna memasukkan 0. Setelah itu, menampilkan total jumlah semua angka yang dimasukkan (kecuali 0).

Langkah Pembuatan :

1. Deklarasikan `int sum = 0;` untuk menyimpan total.

2. Siapkan variabel `int number` (bisa diinisialisasi non-nol) untuk kontrol loop.
3. Buat `Scanner console = new Scanner(System.in);` untuk baca input.
4. Gunakan `while (number != 0)` sebagai loop sentinel.
5. Di dalam loop: tampilkan prompt, baca `number = console.nextInt();`, lalu `sum += number;`.
6. Setelah loop selesai (user mengetik 0), cetak sum dengan `System.out.println("Totalnya adalah " + sum);`.

BAB III

PENUTUP

3.1 Kesimpulan

Berdasarkan praktikum yang telah dilakukan, dapat disimpulkan bahwa perulangan while dan do-while inti dari program atau pemegang kendali dalam pemrograman Java yang memungkinkan eksekusi berulang suatu blok kode berdasarkan kondisi tertentu. Perbedaan utama keduanya terletak pada waktu pengecekan kondisi. While memeriksa kondisi terlebih dahulu sebelum menjalankan blok perintah, sehingga memungkinkan blok tidak pernah dijalankan jika kondisi awal salah, sedangkan do-while menjamin blok dieksekusi minimal satu kali karena pengecekan dilakukan setelah eksekusi. Melalui berbagai contoh program seperti validasi password, simulasi lempar dadu, game penjumlahan, dan sentinel loop, mahasiswa mampu memahami penerapan praktis kedua jenis perulangan tersebut dalam berbagai konteks, sekaligus melatih logika pemrograman, penggunaan kelas bawaan seperti Scanner dan Random, serta pengelolaan variabel kontrol seperti counter dan flag.

3.2 Saran

Untuk meningkatkan efektivitas kegiatan praktikum ke depannya, disarankan agar penyampaian materi dilakukan secara lebih interaktif dengan disertai contoh-contoh program yang relevan dan bervariasi. Dosen maupun asisten laboratorium sebaiknya memberikan penjelasan yang lebih mendalam mengenai kesalahan umum yang sering terjadi saat menulis kode agar mahasiswa dapat belajar dari kesalahan tersebut. Selain itu, waktu pelaksanaan praktikum diharapkan dapat disesuaikan agar peserta memiliki kesempatan yang cukup untuk mencoba, memahami, dan mendiskusikan hasil programnya. Praktikum berikutnya juga sebaiknya disertai latihan tambahan atau mini proyek agar mahasiswa dapat memperkuat pemahaman konsep dan penerapannya dalam kasus nyata.

DAFTAR PUSTAKA

Bibliography

- [1] Andre, "Tutorial Belajar Java Part 35: Perulangan DO WHILE Bahasa Java," Duniaikom, 26 September 2020. [Online]. Available: <https://www.duniaikom.com/tutorial-belajar-java-perulangan-do-while-bahasa-java/>. [Accessed 06 November 2025].
- [3] I. Adityawan, "Belajar Bahasa Pemrograman Java #34 | Perulangan WHILE Bahasa Java," Minarsih, 04 Mei 2023. [Online]. Available: <https://www.minarsih.com/artikel/belajar-bahasa-pemrograman-java-34-perulangan-while-bahasa-java>.