

LAPORAN PRATIKUM PEKAN 6
ALGORITMA DAN PEMROGRAMAN

Disusun Oleh:

Rifqi Aditya

2511533002

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 ke hadirat Allah SWT karena atas rahmat dan karunia-Nya saya dapat menyelesaikan laporan praktikum mata kuliah Algoritma dan Pemrograman dengan judul "*Laporan Cuaca*".

Laporan ini disusun untuk memenuhi salah satu tugas praktikum sekaligus sebagai sarana pembelajaran dalam memahami dasar-dasar pemrograman menggunakan bahasa Java. Melalui praktikum ini, penulis dapat mempelajari penggunaan tipe data dasar seperti *integer*, *float*, *char*, dan *boolean*, serta bagaimana menampilkan informasi ke layar menggunakan perintah `System.out.println`.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis mengucapkan terima kasih kepada dosen pengampu, asisten praktikum, serta rekan-rekan yang telah membantu dalam proses penyusunan laporan ini.

Akhir kata, semoga laporan ini dapat memberikan manfaat, baik bagi penulis sendiri maupun bagi pembaca, khususnya dalam memperdalam pemahaman tentang pemrograman dasar dengan Java.

Padang, 26 September 2025

Penulis

Daftar Pustaka

BAB I PENDAHULUAN.....	4
1.1 Latar Belakang	4
1.2 Tujuan	4
1.3 Manfaat.....	4
BAB II PEMBAHASAN.....	5
2.1 Program Do-While.....	5
2.1.1 Uraian Kode Program.....	5
2.1.2 Uraian Program.....	5
2.1.3 Analisis Hasil.....	6
2.2 Program Perulangan While	6
2.2.1 Uraian Kode Program.....	6
2.2.2 Penjelasan Langkah Kerja	6
2.2.3 Analisis Hasil.....	7
2.3 Program Sentinel Loop	7
2.3.1 Uraian Kode Program.....	7
2.3.2 Penjelasan Langkah Kerja	7
2.3.3 Analisis Hasil.....	8
2.4 Program Lempar Dadu	8
2.4.1 Uraian Kode Program.....	8
2.4.2 Penjelasan Langkah Kerja	8
2.4.3 Analisis Hasil.....	9
2.5 Program Game Penjumlahan	9
2.5.1 Uraian Kode Program.....	10
2.5.2 Penjelasan Langkah Kerja	10
2.5.3 Analisis Hasil.....	11
BAB III	12
DAFTAR PUSTAKA	13

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pemrograman Java merupakan landasan penting dalam mempelajari algoritma. Untuk menguasainya, pemahaman mendalam tentang elemen dasar seperti tipe data sangatlah diperlukan. Praktikum ini bertujuan untuk memberikan pengalaman praktis dalam penggunaan perulangan do-while serta menggunakan sentinel loop dan akan di praktikkan menggunakan software IDE Eclipse

1.2 Tujuan

1. Memahami pengenalan perulangan while dan do-while
2. Mampu mengimplementasikan perulangan do-while untuk kasus dalam program
3. Menganalisis penggunaan while dan do-while untuk membangun program

1.3 Manfaat

1. Meningkatkan kompetensi dan keterampilan teknis dalam menulis kode Java yang efisien dan interaktif menggunakan berbagai jenis perulangan.
2. Mengembangkan kemampuan analisis untuk memilih struktur kendali perulangan yang paling tepat dan efisien sesuai dengan kebutuhan masalah yang dihadapi.
3. Mendapat pengalaman dalam mengintegrasikan perulangan dengan utilitas Java lain seperti Scanner untuk masukan dan Random untuk data probabilistik.

BAB II PEMBAHASAN

Bab ini menguraikan analisis mendalam terhadap lima program Java yang dipraktikkan. Setiap sub-bab akan membahas satu program secara spesifik, mencakup uraian kode program, penjelasan langkah kerja, serta analisis hasil yang didukung oleh teori-teori dasar perulangan dalam pemrograman.

2.1 Program Do-While

Program ini mendemonstrasikan penggunaan perulangan do-while untuk validasi masukan pengguna.

2.1.1 Uraian Kode Program

```
1 package Pekan6_2511533002;
2
3 import java.util.Scanner;
4
5 public class Dowhile1_2511533002 {
6
7     public static void main(String[] args) {
8         Scanner console = new Scanner(System.in);
9         String phrase;
10        do {
11            System.out.print("Inpur password :");
12            phrase = console.next();
13        } while (!phrase.equals("abcd"));
14    }
15
16 }
17 |
```

2.1.2 Uraian Program

1. Program dimulai dan objek Scanner diinisialisasi.
2. Blok do dieksekusi **setidaknya satu kali** tanpa pengecekan kondisi awal.
3. Program mencetak "Inpur password : " ke konsol.
4. Program berhenti sejenak, menunggu pengguna mengetik masukan dan menekan Enter. Masukan disimpan di variabel phrase.
5. Setelah blok do selesai, kondisi while dievaluasi: "Apakah phrase tidak sama dengan "abcd"?"

6. Jika kondisi bernilai true (kata sandi salah), eksekusi kembali ke langkah 3 (awal blok do).
7. Jika kondisi bernilai false (kata sandi benar, yaitu "abcd"), perulangan berhenti dan program selesai.

2.1.3 Analisis Hasil

Program akan terus-menerus meminta pengguna memasukkan kata sandi. Program tidak akan berhenti sampai pengguna mengetikkan string yang persis sama dengan "abcd".

2.2 Program Perulangan While

Program ini mendemonstrasikan penggunaan perulangan while yang dikontrol oleh sebuah variabel boolean.

2.2.1 Uraian Kode Program

```
1 package Pekan6_2511533002;
2
3 import java.util.Scanner;
4
5 public class PerulanganWhile1_2511533002 {
6
7     public static void main(String[] args) {
8         int counter=0;
9         String jawab;
10        boolean running = true;
11        //deklarasi scanner
12        Scanner scan = new Scanner(System.in);
13        while (running) {
14            counter++;
15            System.out.println("Jumlah = "+counter);
16            System.out.print("Apakah Lanjut (ya / tidak)");
17            jawab= scan.nextLine();
18            //cek jawab = tidak, perulangan berhenti
19            if (jawab.equalsIgnoreCase("tidak")) {
20                running= false;
21            }
22        }
23        System.out.println("Anda sudah melakukan perulangan sebanyak "+counter+" kali");
24    }
25 }
26
27 }
```

2.2.2 Penjelasan Langkah Kerja

1. Program dimulai. Variabel counter diatur ke 0
2. counter menjadi 1. Maka program mencetak "Jumlah = 1".
3. Program meminta masukan pengguna "Apakah Lanjut (ya / tidak)".
4. Pengguna memasukkan jawaban (misal, "ya").
5. Kondisi if (jawab.equalsIgnoreCase("tidak")). Jika pengguna memasukkan "ya", kondisi ini false.
6. Blok perulangan selesai, eksekusi kembali ke langkah 2.
7. Kondisi while dievaluasi lagi. dan nilai while masih true.
8. Langkah 3-7 diulangi.

9. Jika pada langkah 5 pengguna memasukkan "tidak", kondisi if pada langkah 6 akan bernilai true.
10. Variabel running diatur ke false.
11. Blok perulangan selesai, eksekusi kembali ke langkah 2.
12. Kondisi while dievaluasi. Karena while sekarang false, perulangan berhenti.
13. Program mencetak pesan akhir yang berisi nilai counter.

2.2.3 Analisis Hasil

Program akan terus berjalan dan menghitung jumlah perulangan, selama pengguna tidak menjawab "tidak" pada pertanyaan konfirmasi. Setelah pengguna menjawab "tidak", program berhenti dan melaporkan berapa kali perulangan telah terjadi.

2.3 Program Sentinel Loop

Program ini menunjukkan penggunaan perulangan while yang dikendalikan oleh nilai sentinel.

2.3.1 Uraian Kode Program

```

1 package Pekan6_2511533002;
2
3 import java.util.Scanner;
4
5 public class SentinelLoop_2511533002 {
6
7     public static void main(String[] args) {
8         Scanner console = new Scanner(System.in);
9         int sum = 0;
10        int number = 12; // "dummy value", anything but 0
11
12        while (number != 0) {
13            System.out.print("Masukkan angka (0 untuk keluar): ");
14            number = console.nextInt();
15            sum = sum + number;
16        }
17        System.out.println("totalnya adalah "+sum);
18    }
19 }
20 }
21

```

2.3.2 Penjelasan Langkah Kerja

1. Program dimulai. sum diatur ke 0, number diatur ke 12.
2. Kondisi while (number != 0) dievaluasi. (12 != 0) adalah true. Badan perulangan dieksekusi.
3. Program meminta masukan angka.
4. Pengguna memasukkan angka (misal, 5). Nilai ini disimpan di number.
5. sum diperbarui: sum = sum + number (misal, sum = 0 + 5, sum menjadi 5).

6. Eksekusi kembali ke langkah 2. Kondisi while dievaluasi. ($5 \neq 0$) adalah true.
7. Langkah 3-6 diulangi. Setiap angka yang dimasukkan pengguna ditambahkan ke sum.
8. Pada satu titik, pengguna memasukkan angka 0.
9. Eksekusi kembali ke langkah 2. Kondisi while ($\text{number} \neq 0$) dievaluasi. ($0 \neq 0$) adalah false.
10. Perulangan berhenti.
11. Program mencetak nilai akhir sum.

2.3.3 Analisis Hasil

Program akan menjumlahkan semua angka yang dimasukkan oleh pengguna. Program akan berhenti meminta masukan ketika pengguna memasukkan angka 0, dan kemudian menampilkan total penjumlahan dari semua angka yang telah dimasukkan (tidak termasuk angka 0 itu sendiri).

2.4 Program Lempar Dadu

Program ini mendemonstrasikan perulangan while yang di acak

2.4.1 Uraian Kode Program

```
1 package Pekan6_2511533002;
2
3 import java.util.Random;
4
5 public class Lempardadu_2511533002 {
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 }
23 }
```

2.4.2 Penjelasan Langkah Kerja

1. Program dimulai. Objek Random dibuat. tries dan sum adalah 0.
2. Kondisi while ($\text{sum} \neq 7$) dievaluasi. ($0 \neq 7$) adalah true. Badan perulangan dieksekusi.
3. dadu1 mendapat nilai acak (misal, 3).
4. dadu2 mendapat nilai acak (misal, 5).
5. sum dihitung: $\text{sum} = 3 + 5 = 8$.

6. Program mencetak " $3 + 5 = 8$ ".
7. tries dinaikkan menjadi 1.
8. Eksekusi kembali ke langkah 2. Kondisi while dievaluasi. ($8 \neq 7$) adalah true.
9. Proses ini berlanjut sampai suatu saat lemparan menghasilkan sum 7 (misal, $4 + 3 = 7$).
10. Program mencetak " $4 + 3 = 7$ ".
11. tries dinaikkan (misal, menjadi 9).
12. Eksekusi kembali ke langkah 2. Kondisi while ($\text{sum} \neq 7$) dievaluasi. ($7 \neq 7$) adalah false.
13. Perulangan berhenti.
14. Program mencetak pesan "You won after 9 tries!".

2.4.3 Analisis Hasil

Program akan menyimulasikan pelemparan dua dadu berulang kali. Setiap lemparan akan ditampilkan. Program akan berhenti secara otomatis ketika jumlah kedua dadu adalah 7. Hasil akhirnya adalah laporan tentang berapa banyak percobaan (lemparan) yang diperlukan untuk mendapatkan angka 7.

2.5 Program Game Penjumlahan

Program ini menggabungkan beberapa konsep, termasuk perulangan, metode, dan logika permainan.

2.5.1 Uraian Kode Program

```
import java.util.Scanner;

public class GamePenjumlahan_2511533002 {

    public static void main(String[] args) {
        Scanner console = new Scanner(System.in);
        Random rand = new Random();
        //play until user get 3 wrong
        int point = 0;
        int wrong = 0;
        while (wrong < 3) {
            int result = play(console, rand); //play one game
            if(result > 0) {
                point++;
            } else {
                wrong++;
            }
        }
        System.out.println("You earned "+point+" total points.");
    }
    //membuat soal penjumlahan dan ditampilkan ke user
    public static int play(Scanner console, Random rand) {
        //print the operands being added, and sum then
        int operands = rand.nextInt(4) + 2;
        int sum = rand.nextInt(10) + 1;
        System.out.print(sum);

        for (int i = 2; i <= operands; i++) {
            int n = rand.nextInt(10) + 1;
            sum += n;
            System.out.print(" + " + n);
        }
        System.out.print(" = ");

        //read user guess and report whether it was correct
        int guess = console.nextInt();
        if (guess == sum) {
            return 1;
        } else {
            System.out.println("wrong! the answer was "+ sum);
            return 0;
        }
    }
}
```

2.5.2 Penjelasan Langkah Kerja

1. Program dimulai. Scanner, Random, point (0), dan wrong (0) diinisialisasi.
2. Kondisi while (wrong < 3) dievaluasi. (0 < 3) adalah true.
3. Metode play dipanggil.
4. Di dalam play: Soal dibuat. Misal: operands = 3. sum awal = 5. Mencetak "5".
Perulangan for berjalan 1 kali (i=2). n = 8. sum menjadi 13. Mencetak " + 8".
Perulangan for berjalan 1 kali (i=3). n = 2. sum menjadi 15. Mencetak " + 2".
Program mencetak " = ".
5. Di dalam play: Pengguna memasukkan guess (misal, 15).
6. Di dalam play: if (15 == 15) bernilai true. Metode play mengembalikan 1.
7. Di dalam main: result adalah 1. if (result > 0) adalah true. point menjadi 1.
8. Eksekusi kembali ke langkah 2. Kondisi while (wrong < 3) dievaluasi (0 < 3).
9. Metode play dipanggil lagi. Pengguna memberikan jawaban salah (misal, guess 10 padahal sum 12).
10. Di dalam play: if (10 == 12) false. Blok else dieksekusi, mencetak jawaban benar.
Metode play mengembalikan 0.

11. Di dalam main: result adalah 0. if (result > 0) false. Blok else dieksekusi. wrong menjadi 1.
12. Proses ini berulang hingga wrong menjadi 3.
13. Kondisi while (wrong < 3) dievaluasi ($3 < 3$) adalah false. Perulangan berhenti.
14. Program mencetak total point yang didapat.

2.5.3 Analisis Hasil

Pengguna akan disajikan serangkaian soal penjumlahan acak. Pengguna mendapatkan poin untuk jawaban benar, dan "kesempatan" berkurang untuk jawaban salah. Permainan akan berakhir setelah pengguna membuat 3 kesalahan. Skor akhir kemudian ditampilkan.

BAB III

KESIMPULAN

Berdasarkan hasil praktikum dan analisis terhadap kelima program yang telah dibuat, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Praktikum ini menegaskan perbedaan antara perulangan while dan do-while. while yaitu menguji kondisi sebelum badan perulangan dieksekusi, sehingga ada kemungkinan badan perulangan tidak dieksekusi sama sekali. Sebaliknya, do-while adalah post-test loop yang menguji kondisi setelah badan perulangan dieksekusi, sehingga menjamin eksekusi badan perulangan minimal satu kali.
2. Struktur perulangan while menunjukkan fleksibilitas tinggi dan dapat dikendalikan oleh berbagai jenis kondisi. Praktikum ini telah berhasil mengimplementasikan tiga skenario kontrol yang berbeda:
 - Kontrol Bendera (Flag-Controlled): (Program PerulanganWhile1)
Menggunakan variabel boolean untuk menghentikan perulangan berdasarkan konfirmasi eksplisit dari pengguna.
 - Kontrol Sentinel (Sentinel-Controlled): (Program SentinelLoop)
Menggunakan nilai masukan khusus (0) untuk menandai akhir dari pemrosesan data, ideal ketika jumlah data tidak diketahui sebelumnya.
 - Kontrol Peristiwa (Event-Controlled): (Program Lempardadu) Menggunakan hasil dari suatu proses acak (jumlah dadu = 7) sebagai kondisi berhenti, ideal untuk simulasi.

DAFTAR PUSTAKA

- [1] Deitel, P. J. (2018). *Java: How to Program, Early Objects*.
- [2] Horstmann, C. S. (2019). *Core Java Volume I--Fundamentals*.
- [3] Rosihan Ari Yuana, S. M. (2022). *Pemrograman Java*.