

LAPORAN PRATIKUM PEKAN 6
ALGORITMA DAN PEMROGRAMAN

disusun Oleh:

ALIF MUHAMMAD IHSAN

NIM 2511531004

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

Asisten Laboratorium: Muhammad Zaki Al Hafiz



DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS

PADANG

2025

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah Swt atas limpahan rahmat dan karunia-Nya sehingga praktikum pemrograman Java ini dapat diselesaikan dengan baik dan laporan ini dapat disusun sebagai bentuk pertanggungjawaban akademik.

Laporan praktikum ini disusun untuk mendokumentasikan proses pembelajaran yang telah dilakukan, khususnya dalam memahami struktur dasar pemrograman Java, penggunaan metode, serta penerapan logika prosedural dalam bentuk program sederhana. Praktikum ini diharapkan dapat memperkuat pemahaman peserta terhadap konsep-konsep dasar pemrograman serta meningkatkan kemampuan teknis dalam menulis kode yang terstruktur dan efisien.

Penulis menyadari bahwa penyusunan laporan ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada dosen pembimbing, asisten praktikum, serta rekan-rekan yang telah memberikan dukungan dan masukan selama pelaksanaan praktikum.

Akhir kata, penulis berharap laporan ini dapat memberikan manfaat dan menjadi referensi yang berguna bagi pembaca dalam memahami dasar-dasar pemrograman Java. Kritik dan saran yang membangun sangat diharapkan demi penyempurnaan laporan di masa mendatang.

Padang, September 2025

Alif Muhammad Ihsan

Daftar Isi

KATA PENGANTAR.....	i
BAB I.....	1
PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan.....	1
1.3 Manfaat Praktikum	2
BAB II.....	3
PEMBAHASAN.....	3
2.1 Kode Program.....	3
2.2 Langkah Kerja.....	11
2.3 Analisis Hasil	16
BAB III	20
PENUTUP.....	20
3.1 Kesimpulan.....	20
3.2 Saran.....	21
DAFTAR PUSTAKA	22

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era perkembangan teknologi informasi yang semakin pesat, pemahaman terhadap dasar-dasar pemrograman menjadi hal yang esensial bagi setiap individu yang ingin berkontribusi dalam bidang rekayasa perangkat lunak. Salah satu bahasa pemrograman yang banyak digunakan dalam dunia akademik maupun industri adalah Java. Bahasa ini dikenal karena sifatnya yang berorientasi objek, portabel, serta memiliki struktur sintaks yang relatif mudah dipahami oleh pemula.

Praktikum ini dilaksanakan sebagai bagian dari upaya untuk memperkuat pemahaman mahasiswa terhadap konsep dasar pemrograman Java, khususnya dalam hal struktur program, penggunaan metode, serta logika eksekusi program. Melalui kegiatan praktikum, peserta diharapkan mampu menerapkan teori yang telah dipelajari dalam bentuk implementasi nyata, sehingga tercipta keterkaitan antara aspek konseptual dan praktik langsung.

1.2 Tujuan

1. Memahami struktur dasar program Java, termasuk penggunaan metode, pemanggilan fungsi, dan alur eksekusi program secara sistematis.
2. Melatih kemampuan dalam menyusun algoritma prosedural, melalui implementasi langkah-langkah logis yang merepresentasikan proses nyata, seperti pembuatan kue.
3. Meningkatkan keterampilan teknis dalam menulis kode yang terstruktur dan efisien, serta membiasakan diri dengan praktik penulisan program yang baik dan mudah dipahami.

4. Mendorong kemampuan berpikir kritis dan analitis, khususnya dalam mengidentifikasi urutan proses, menyederhanakan logika, dan menerjemahkan aktivitas sehari-hari ke dalam bentuk algoritma.

1.3 Manfaat Praktikum

1. **Memperkuat pemahaman konsep dasar pemrograman Java**, sehingga peserta mampu mengaitkan teori dengan implementasi nyata.
2. **Mengasah keterampilan penyusunan algoritma prosedural**, melalui pembiasaan merinci setiap langkah kerja sebelum menerjemahkannya ke dalam kode.
3. **Meningkatkan kemampuan menulis kode yang modular dan efisien**, dengan memanfaatkan metode sebagai unit kerja yang terpisah dan terstruktur.
4. **Memberikan pengalaman langsung dalam penerjemahan proses sehari-hari ke dalam program**, sehingga peserta dapat melihat relevansi pemrograman dalam konteks kehidupan nyata.
5. **Melatih kedisiplinan, kerapian, dan kemampuan dokumentasi**, yang menjadi landasan penting bagi profesional di bidang rekayasa perangkat lunak.

BAB II

PEMBAHASAN

2.1 Kode Program

a. doWhile1

```
package Pekan6_2511531004;

import java.util.Scanner;

public class doWhile1_2511531004 {

    public static void main(String[] args) {

        // TODO Auto-generated method stub

        Scanner console = new Scanner(System.in);

        String phrase ;

        do {

            System.out.print("Input Password : ");

            phrase = console.next();

        } while ( !phrase.equals("abcd"));

        console.close();
```

```

    }
}

```

Hasil Output

Input Password : aaaaa

Input Password : ssss

Input Password : abcd

b. Game Penjumlahan

```

package Pekan6_2511531004;
import java.util.Scanner;
import java.util.Random;
public class GamePenjumlahan_2511531004 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Scanner console = new Scanner(System.in);
        Random rand = new Random();

        // play untill user gets 3 wrong
        int points = 0;
        int wrong = 0;
        while (wrong < 3) {
            int result = play(console,rand);

            if (result > 0) {
                points++;

            } else {
                wrong++;
            }
        }
    }
}

```

```

        }

    }

    System.out.println("You earned " + points + "total points.");

}

// membuat soal penjumlahan dan ditampilkan ke user
public static int play(Scanner console, Random rand) {

    // print the operand being added, and sum them
    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();
        sum += n;
    }

    System.out.print(" = ");

    // read users guess and report weather it it was correct
    int guess = console.nextInt();
    if (guess == sum) {
        return 1;
    } else {
        System.out.println("Wrong ! The answer was " +
sum);
        return 0;
    }
}

```



```
}
```

```
}
```

```
}
```

Hasil Output

9 = 7

Wrong ! The answer was 492721074

7 = 3

Wrong ! The answer was 1038068142

7 = 4

Wrong ! The answer was 507883269

You earned 0total points.

c. Lempar Dadu

```
package Pekan6_2511531004;
```

```
import java.util.Random;
```

```
public class LemparDadu_2511531004 {
```

```
    public static void main(String[] args) {
```

```
        // TODO Auto-generated method stub
```

```
        Random rand = new Random();
```

```
        int tries = 0;
```

```
        int sum = 0;
```

```
        while ( sum != 7) {
```

```
            // roll the dice one
```

```
            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");
}

}

```

Hasil Output

```

3+1=4
4+6=10
3+5=8
1+3=4
3+5=8
4+4=8
2+1=3
3+6=9
1+3=4
4+5=9
3+5=8
4+2=6
2+4=6
5+6=11
3+1=4
6+2=8
1+4=5
6+3=9
1+1=2
6+5=11
3+3=6

```

5+3=8

5+6=11

4+5=9

6+2=8

5+3=8

3+6=9

4+6=10

6+6=12

5+2=7

You won after 30 tries

d. Perulangan While 1

```
package Pekan6_2511531004;
import java.util.Scanner;
public class PerulanganWhile1_2511531004 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        int counter = 0;
        String Jawab;
        boolean running = true;

        // deklarasi scanner
        Scanner scan = new Scanner(System.in);

        while (running) {
            counter++;
            System.out.println("Jumlah = " + counter);
            System.out.println("Apakah lanjut (ya / tidak)");
            Jawab = scan.nextLine();
        }
    }
}
```

```

        // cek jawab = tidak, perulangan berhenti
        if (Jawab.equalsIgnoreCase("tidak")) {
            running = false;
        }

    }

    System.out.println("Anda sudah melakukan perulangan
    sebanyak " + counter + " kali" );
}

}

```

Hasil Output

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

e. SentinelLoop

```

package Pekan6_2511531004;
import java.util.Scanner;
public class SentinelLoop_2511531004 {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
    }
}

```

```

Scanner console = new Scanner(System.in);
int sum = 0;
int number= 12;

while (number != 0) {
    System.out.println("Masukkan angka (0 untuk
keluar) : ");

    number = console.nextInt();
    sum = sum + number;
}
System.out.println("Totalnya adalah " + sum);

console.close();
}
}

```

Hasil Output

Masukkan angka (0 untuk keluar) :

2

Masukkan angka (0 untuk keluar) :

2

Masukkan angka (0 untuk keluar) :

0

Totalnya adalah 4

2.2 Langkah Kerja

Langkah-langkah yang dilakukan dalam praktikum ini adalah sebagai berikut:

A. Langkah Kerja Program `doWhile1_2511531004`

1. Import Library Scanner

- Program mengimpor library `java.util.Scanner` untuk membaca input dari pengguna melalui keyboard.

2. Deklarasi Kelas dan Method `main`

- Program berada dalam kelas `doWhile1_2511531004` di package `Pekan6_2511531004`.
- Method `main` adalah titik awal eksekusi program.

3. Inisialisasi Scanner

- `Scanner console = new Scanner(System.in);`
- Membuat objek `console` untuk membaca input dari pengguna.

4. Deklarasi Variabel

- `String phrase;`
- Variabel `phrase` digunakan untuk menyimpan input password dari user.

5. Perulangan `do-while`

- Program masuk ke dalam blok `do-while` yang akan terus berjalan selama kondisi `!phrase.equals("abcd")` bernilai `true`.
- Di dalam blok `do`, program:
 - Menampilkan teks: `Input Password :`
 - Menunggu input dari user dan menyimpannya ke variabel `phrase` menggunakan `console.next();`
- Jika input tidak sama dengan `"abcd"`, maka perulangan akan diulang.
- Jika input sama dengan `"abcd"`, maka perulangan berhenti.

6. Menutup Scanner

- Setelah keluar dari perulangan, program menutup objek `Scanner` dengan `console.close();` untuk menghindari `memory leak`.

B. Langkah Kerja Program GamePenjumlahan_2511531004

1. Import Library

- Scanner → untuk membaca input dari user.
- Random → untuk menghasilkan angka acak sebagai soal penjumlahan.

2. Deklarasi Kelas dan Method main

- Kelas: GamePenjumlahan_2511531004 dalam package Pekan6_2511531004.
- Method main adalah titik awal eksekusi program.

3. Inisialisasi Objek

- Scanner console = new Scanner(System.in); → membaca input dari user.
- Random rand = new Random(); → menghasilkan angka acak.

4. Deklarasi Variabel Skor

- int points = 0; → menyimpan jumlah jawaban benar.
- int wrong = 0; → menyimpan jumlah jawaban salah.

5. Perulangan while

- Program akan terus berjalan selama wrong < 3.
- Di setiap iterasi, program memanggil method play(console, rand) untuk membuat soal dan mengecek jawaban.
- Jika jawaban benar (result > 0), poin bertambah.
- Jika jawaban salah, nilai wrong bertambah.

6. Menampilkan Skor Akhir

- Setelah user salah 3 kali, program menampilkan total poin yang didapat.

Langkah Kerja Method play()

1. Menentukan Jumlah Operand

- int operands = rand.nextInt(4) + 2; → jumlah angka yang akan dijumlahkan (antara 2 sampai 5).

2. Inisialisasi Soal

- int sum = rand.nextInt(10) + 1; → angka pertama antara 1 sampai 10.
- Ditampilkan ke layar: System.out.print(sum);

3. Menambahkan Operand Tambahan

- Perulangan dari $i = 2$ sampai operands.
- Di setiap iterasi:
- `int n = rand.nextInt();` → angka acak.
- `sum += n;` → menambahkan ke total.

4. Menampilkan Soal

- Setelah semua operand ditambahkan, tampilkan tanda = untuk meminta jawaban.

5. Membaca Jawaban User

- `int guess = console.nextInt();` → input tebakan dari user.

6. Cek Jawaban

- Jika `guess == sum`, return 1 (jawaban benar).
- Jika salah, tampilkan pesan dan return 0.

C. Langkah Kerja Program Lempardadu_2511531004

1. Import Library

- `import java.util.Random;` → digunakan untuk menghasilkan angka acak sebagai simulasi lemparan dadu.

2. Deklarasi Kelas dan Method main

- Kelas: `Lempardadu_2511531004` dalam package `Pekan6_2511531004`.
- Method `main()` adalah titik awal eksekusi program.

3. Inisialisasi Objek dan Variabel

- `Random rand = new Random();` → objek untuk menghasilkan angka acak.
- `int tries = 0;` → menghitung berapa kali dadu dilempar.
- `int sum = 0;` → menyimpan jumlah dua dadu.

4. Perulangan while

- Program akan terus mengulang selama `sum != 7`.
- Di setiap iterasi:
- `dadu1 = rand.nextInt(6) + 1;` → menghasilkan angka 1–6 untuk dadu pertama.

- `dadu2 = rand.nextInt(6) + 1;` → menghasilkan angka 1–6 untuk dadu kedua.
- `sum = dadu1 + dadu2;` → menjumlahkan kedua dadu.
- Menampilkan hasil lemparan: `System.out.println(dadu1 + "+" + dadu2 + "=" + sum);`
- Menambah jumlah percobaan: `tries++;`

5. Menampilkan Hasil Akhir

- Setelah keluar dari perulangan (karena `sum == 7`), program menampilkan:
You won after X tries
- di mana X adalah jumlah lemparan yang dibutuhkan untuk mendapatkan total 7.

D. Langkah Kerja Program Perulangan While1_2511531004

Deklarasi Variabel:

- counter bertipe int, digunakan untuk menghitung jumlah perulangan.
- Jawab bertipe String, menyimpan input pengguna.
- running bertipe boolean, sebagai penentu kondisi perulangan.
- scan bertipe Scanner, digunakan untuk membaca input dari pengguna.

Langkah Kerja:

1. Inisialisasi variabel counter dengan nilai awal 0 dan running dengan nilai true.
2. Buat objek Scanner untuk membaca input dari pengguna melalui konsol.
3. Jalankan perulangan while selama nilai running bernilai true.
4. Di dalam perulangan:
 - Tambahkan nilai counter sebanyak satu.
 - Tampilkan jumlah perulangan saat ini ke layar.
 - Tampilkan pertanyaan kepada pengguna: “Apakah lanjut (ya / tidak)”.
 - Simpan jawaban pengguna ke dalam variabel Jawab.
 - Jika jawaban pengguna adalah “tidak” (tanpa memperhatikan huruf besar/kecil), ubah nilai running menjadi false agar perulangan berhenti.

5. Setelah keluar dari perulangan, tampilkan jumlah total perulangan yang telah dilakukan.

E. Langkah Kerja Program SentinelLoop_2511531004

Deklarasi Variabel:

- console bertipe Scanner, digunakan untuk membaca input dari pengguna.
- sum bertipe int, menyimpan total penjumlahan dari angka yang dimasukkan.
- number bertipe int, menyimpan angka input dari pengguna dan berfungsi sebagai nilai penjaga (sentinel).

Langkah Kerja:

1. Inisialisasi variabel sum dengan nilai 0 dan number dengan nilai awal 12 (nilai bebas selain 0).
2. Buat objek Scanner untuk membaca input dari pengguna melalui konsol.
3. Jalankan perulangan while selama nilai number tidak sama dengan 0.
4. Di dalam perulangan:
 - Tampilkan pesan “Masukkan angka (0 untuk keluar)” ke layar.
 - Baca input angka dari pengguna dan simpan ke dalam variabel number.
 - Tambahkan nilai number ke dalam variabel sum.
5. Jika pengguna memasukkan angka 0, maka perulangan berhenti.
6. Setelah keluar dari perulangan, tampilkan total penjumlahan dari semua angka yang telah dimasukkan.
7. Tutup objek Scanner untuk mengakhiri proses input.

2.3 Analisis Hasil

A. Program doWhile1

Program ini menerima masukan berupa kata sandi dari pengguna dan memvalidasinya menggunakan struktur perulangan do-while. Program akan terus meminta input hingga pengguna mengetikkan kata sandi yang sesuai, yaitu "abcd". Karena struktur do-while menjamin blok perulangan dijalankan minimal satu kali, maka proses input akan tetap terjadi meskipun kondisi awal belum terpenuhi.

Logika do-while bekerja dengan benar, mengeksekusi blok perulangan terlebih dahulu sebelum memeriksa kondisi. Hal ini memastikan bahwa pengguna diberikan kesempatan untuk memasukkan kata sandi setidaknya satu kali. Pendekatan ini sangat sesuai untuk proses validasi input yang bersifat interaktif dan berulang.

Referensi:

- Deitel, P. J., & Deitel, H. M. (2017). *Java: How to Program*. Pearson Education.
- Nugroho, A. (2020). *Logika dan Algoritma dalam Pemrograman*. Penerbit Informatika.

B. Program Game Penjumlahan

Program ini merupakan permainan edukatif yang menguji kemampuan pengguna dalam menjawab soal penjumlahan. Program akan terus memberikan soal secara acak hingga pengguna menjawab salah sebanyak tiga kali. Setiap jawaban yang benar akan menambah skor, sedangkan jawaban salah akan menambah jumlah kesalahan.

Struktur kontrol while digunakan untuk mengatur batas permainan, yaitu maksimal tiga kesalahan. Fungsi play() digunakan untuk menghasilkan soal penjumlahan secara acak dan mengevaluasi jawaban pengguna. Jika jawaban benar, fungsi mengembalikan nilai 1; jika salah, mengembalikan 0 dan menampilkan jawaban yang benar.

Struktur ini mencerminkan prinsip pemrograman modular, di mana logika utama permainan dipisahkan dari logika pembuatan soal. Pendekatan

ini memudahkan pengembangan dan pemeliharaan program. Selain itu, penggunaan kelas Random memungkinkan variasi soal yang dinamis, sehingga permainan tidak monoton.

Referensi:

- Deitel, P. J., & Deitel, H. M. (2017). *Java: How to Program*. Pearson Education.
- Nugroho, A. (2020). *Logika dan Algoritma dalam Pemrograman*. Penerbit Informatika.

C. Program Lempar Dadu

Program ini mensimulasikan permainan lempar dadu menggunakan dua buah dadu. Tujuan dari permainan adalah mendapatkan jumlah total lemparan yang bernilai 7. Program akan terus melakukan simulasi hingga kondisi tersebut tercapai, lalu menampilkan jumlah percobaan yang dibutuhkan.

Struktur kontrol while digunakan untuk menjalankan perulangan selama jumlah dua dadu belum sama dengan 7. Di setiap iterasi, dua angka acak antara 1 hingga 6 dihasilkan menggunakan objek Random, yang merepresentasikan hasil lemparan masing-masing dadu. Nilai dari kedua dadu dijumlahkan, ditampilkan ke layar, dan dihitung sebagai satu percobaan.

Referensi:

- Deitel, P. J., & Deitel, H. M. (2017). *Java: How to Program*. Pearson Education.
- Nugroho, A. (2020). *Logika dan Algoritma dalam Pemrograman*. Penerbit Informatika.

D. Program Perulangan

Program ini menggunakan struktur perulangan `while` untuk menjalankan proses secara berulang berdasarkan input pengguna. Tujuan dari program adalah menghitung berapa kali perulangan dilakukan, selama pengguna memilih untuk melanjutkan.

Struktur kontrol `while` digunakan untuk mengatur alur perulangan yang bergantung pada kondisi logika boolean. Di setiap iterasi, nilai counter akan bertambah satu, kemudian program menampilkan jumlah saat ini dan meminta input dari pengguna. Jika pengguna menjawab “tidak”, maka nilai `running` diubah menjadi `false`, sehingga perulangan berhenti.

Referensi:

- Deitel, P. J., & Deitel, H. M. (2017). *Java: How to Program*. Pearson Education.
- Nugroho, A. (2020). *Logika dan Algoritma dalam Pemrograman*. Penerbit Informatika.

E. Program Sentinel Loop

Program ini menerima masukan berupa angka dari pengguna dan menjumlahkannya secara berulang. Proses perulangan akan terus berlangsung hingga pengguna memasukkan angka 0 sebagai penanda berhenti (nilai penjaga atau *sentinel*). Setelah perulangan selesai, program menampilkan total dari seluruh angka yang telah dimasukkan.

Struktur kontrol `while` digunakan untuk mengatur perulangan berdasarkan kondisi nilai `number`. Selama nilai tersebut tidak sama dengan 0, perulangan akan terus berjalan. Di setiap iterasi, pengguna diminta memasukkan angka, lalu angka tersebut ditambahkan ke variabel `sum`. Ketika pengguna memasukkan angka 0, perulangan berhenti dan hasil penjumlahan ditampilkan.

Referensi:

- Deitel, P. J., & Deitel, H. M. (2017). *Java: How to Program*. Pearson Education.
- Nugroho, A. (2020). *Logika dan Algoritma dalam Pemrograman*. Penerbit Informatika.

BAB III

PENUTUP

3.1 Kesimpulan

Berdasarkan seluruh program yang telah dibuat dan dianalisis, dapat disimpulkan bahwa:

- Pemahaman terhadap **struktur dasar Java** seperti for, while, if-else, dan scanner sangat penting dalam membangun logika program yang efisien.
- Penggunaan **perulangan dan nested loop** memungkinkan pembuatan pola, tabel, dan simulasi logika yang kompleks secara sistematis.
- Proses **debugging dan analisis output** membantu memperkuat pemahaman terhadap alur program dan memperbaiki kesalahan logika.
- Penerapan **pseudocode dan flowchart** terbukti efektif dalam merancang solusi sebelum implementasi kode.
- Setiap program yang dibuat menunjukkan peningkatan dalam kemampuan berpikir algoritmik dan penyusunan solusi yang lebih rapi dan user-friendly.

3.2 Saran

Agar proses belajar dan pengembangan program ke depan lebih optimal, berikut beberapa saran:

1. **Tingkatkan eksplorasi ke konsep OOP (Object-Oriented Programming)** seperti class, method, dan inheritance untuk memperluas cakupan pemrograman Java.
2. **Gunakan komentar kode dan dokumentasi internal** agar program lebih mudah dipahami dan dipresentasikan.
3. **Lakukan eksperimen dengan input dinamis dan validasi data** untuk meningkatkan interaktivitas dan keamanan program.
4. **Coba integrasi dengan GUI sederhana (seperti Java Swing)** jika ingin membuat program lebih menarik dan visual.
5. **Terus gunakan pseudocode dan flowchart** sebelum coding agar logika tetap terstruktur dan minim error.
6. **Lakukan review mandiri atau diskusi dua arah** untuk memperkuat

DAFTAR PUSTAKA

Deitel, H. M., & Deitel, P. J. (2017). *Java: How to Program* (10th Edition). Pearson Education.

Oracle. (n.d.). *The Java™ Tutorials*. Diakses dari <https://docs.oracle.com/javase/tutorial/>

W3Schools. (n.d.). *Java Tutorial*. Diakses dari <https://www.w3schools.com/java/>

GeeksforGeeks. (n.d.). *Java Programming Examples*. Diakses dari <https://www.geeksforgeeks.org/java/>

Dokumentasi dan catatan pribadi selama proses belajar dan debugging program.