

LAPORAN TUGAS  
ALGORITMA DAN PEMROGRAMAN  
“PROGRAM TEBAK DADU PERULANGAN UNCOUNTEDED DALAM  
BAHASA JAVA”



disusun Oleh:  
Annisa Layli Ramadhani  
2511532024  
Informatika Kelas B

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 ke hadirat Allah SWT atas rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan praktikum pemrograman Java ini dengan baik. Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban kegiatan praktikum sekaligus untuk memperdalam pemahaman mengenai konsep dasar dan penerapan bahasa pemrograman Java. Dalam penyusunan laporan ini, penulis mengucapkan terima kasih kepada dosen pengampu, asisten laboratorium, serta rekan-rekan yang telah memberikan bimbingan, dukungan, dan kerja sama selama proses praktikum berlangsung. Penulis menyadari laporan ini masih jauh dari sempurna, sehingga kritik dan saran yang membangun sangat diharapkan demi perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat dan menambah wawasan bagi penulis maupun pembaca.

Padang, 6 November 2025

Penulis

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>ii</b>
<b>DAFTAR ISI.....</b>	<b>iii</b>
<b>BAB I.....</b>	<b>4</b>
<b>1.1 Latar Belakang .....</b>	<b>4</b>
<b>1.2 Tujuan .....</b>	<b>5</b>
<b>1.3 Manfaat .....</b>	<b>5</b>
<b>BAB II .....</b>	<b>6</b>
<b>2.1 Pseudocode .....</b>	<b>6</b>
<b>2.2 Flowchart .....</b>	<b>7</b>
<b>2.3 Kode Program.....</b>	<b>8</b>
<b>2.4 Output Program .....</b>	<b>9</b>
<b>BAB III.....</b>	<b>11</b>
<b>3.1 Kesimpulan .....</b>	<b>11</b>
<b>3.2 Saran.....</b>	<b>11</b>
<b>DAFTAR PUSTAKA .....</b>	<b>12</b>

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam dunia pemrograman, penggunaan struktur kontrol seperti perulangan (loop) memiliki peranan penting dalam mengatur proses eksekusi program agar dapat berjalan secara efisien dan berulang sesuai kebutuhan. Perulangan memungkinkan komputer untuk mengeksekusi blok kode yang sama berkali-kali tanpa harus menuliskan instruksi yang berulang secara manual. Salah satu jenis perulangan yang sering digunakan dalam bahasa pemrograman java adalah perulangan uncounted, yaitu perulangan yang tidak ditentukan jumlah pengulangannya sejak awal, melainkan bergantung pada suatu kondisi logika tertentu agar dapat berhenti.

Program tebak dadu menggunakan perulangan uncounted merupakan contoh penerapan sederhana dari konsep tersebut. Dalam program ini, komputer melakukan simulasi pelemparan dua buah dadu secara acak menggunakan class Random, kemudian menjumlahkan hasil keduanya. Pemain dinyatakan menang apabila jumlah dua dadu yang muncul bernilai 7, sedangkan jika hasilnya bukan 7, pemain diberi pilihan untuk melanjutkan atau menghentikan permainan. Proses ini berlangsung terus menerus hingga kondisi kemenangan tercapai atau pemain memilih berhenti.

Melalui program ini, konsep perulangan uncounted (do-while loop) dapat dipahami secara lebih nyata. Struktur do-while digunakan karena program harus menjalankan minimal satu kali proses pelemparan dadu sebelum mengevaluasi kondisi logika di dalamnya. Selain itu, program ini juga memperkenalkan penggunaan variabel logika (boolean) untuk menentukan status kemenangan, serta variabel numerik dan string untuk menampung data acak dan respon pengguna.

Penerapan program ini tidak hanya membantu mahasiswa memahami cara kerja perulangan tanpa batas jumlah tetap, tetapi juga

memperlihatkan hubungan antar struktur logika, percabangan, dan input pengguna dalam membangun sebuah aplikasi interaktif sederhana. Dengan demikian, program ini menjadi contoh yang efektif dalam memahami bagaimana teori perulangan, logika kondisi, dan penggunaan variabel dapat diimplementasikan secara langsung dalam bahasa pemrograman java.

## **1.2 Tujuan**

Tujuan dari tugas praktikum antara lain sebagai berikut:

- 1.2.1 Memahami konsep dasar perulangan uncounted (khususnya struktur do-while) dalam bahasa pemrograman java.
- 1.2.2 Mempelajari penerapan logika kondisi dan variabel untuk mengontrol jalannya program interaktif.
- 1.2.3 Mengimplementasikan struktur kontrol (loop dan percabangan) untuk membuat program yang dinamis dan fleksibel.
- 1.2.4 Melatih kemampuan menggunakan class random dalam menghasilkan bilangan acak untuk simulasi lempar dadu.

## **1.3 Manfaat**

Manfaat dari tugas praktikum antara lain sebagai berikut:

- 1.3.1 Menambahkan pemahaman praktis mahasiswa terhadap penggunaan perulangan uncounted (do-while) yang umum digunakan dalam pemrograman berbasis kondisi.
- 1.3.2 Memberikan pengalaman langsung dalam menggabungkan konsep variabel, logika kondisi, dan perulangan menjadi satu kesatuan program yang berfungsi penuh.
- 1.3.3 Meningkatkan kemampuan berpikir logis dan sistematis, karena setiap bagian kode memiliki keterkaitan logika yang harus dipahami dengan benar.
- 1.3.4 Membantu memahami mekanisme kontrol program yang bergantung pada keputusan pengguna, sehingga program dapat bersifat interaktif.

## BAB II

### PEMBAHASAN

#### 2.1 Pseudocode

Pseudocode berikut digunakan untuk menggunakan alur logika dari program Tebak Dadu Menggunakan Perulangan Uncounted dalam Bahasa Java. Tujuannya adalah menjelaskan langkah-langkah kerja program secara sistematis sebelum diterjemahkan ke dalam kode program sebenarnya. Pseudocode ini memuat proses inisialisasi variabel, penggunaan perulangan do-while, kondisi pengecekan nilai jumlah dadu, serta keputusan pengguna untuk melanjutkan atau menghentikan permainan.

<b>Judul</b>
Program Tebak Dadu Menggunakan Perulangan Uncounted dalam Bahasa Java
<b>Deklarasi</b>
Jawab : String Menang : Boolean Dadu1, dadu2, sum, tries = integer
<b>Algoritma</b>
1. Mulai 2. Tries $\leftarrow$ 0 3. Sum $\leftarrow$ 0 4. Running $\leftarrow$ false 5. Dadu1 $\leftarrow$ random (1-6) 6. Dadu2 $\leftarrow$ random (1-6) 7. Sum $\leftarrow$ dadu1 + dadu2 8. Tries $\leftarrow$ tries + 1 9. Print dadu1 + " + " + dadu2 + " = " + sum 10. If sum = 7 then 11.   Print "Tebakan Anda Benar!" 12.   Anda Menang setelah " + tries + "percobaan!"

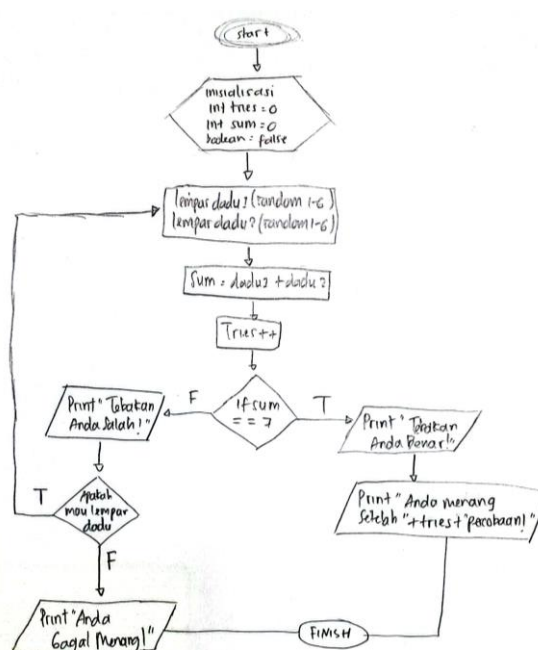
```

13.  Running ← true
14. Else
15.  Print “Tebakan Anda Salah!”
16.  Print “ Apakah mau lempar dadu? (ya/tidak)”
17.  Read answer
18.  If answer = “tidak” then
19.      Print “Anda gagal menang!”
20.  End if
21.  While (!Running)
22.  Close Scanner
23.  End

```

## 2.2 Flowchart

Flowchart berikut menggambarkan alur logika dari program tebak dadu menggunakan perulangan uncounted dalam bahasa java. Diagram ini menunjukkan proses kerja program secara visual, mulai dari inisialisasi variabel, proses pelemparan dua dadu secara acak, pengecekan kondisi kemenangan (jumlah = 7), hingga keputusan pengguna untuk melanjutkan atau menghentikan permainan. Dengan adanya flowchart ini, alur program dapat dipahami dengan lebih mudah dan sistematis sebelum diimplementasikan ke dalam kode java.



### 2.3 Kode Program

```
1. package pekan6_2511532024;
2.
3. import java.util.Random;
4. import java.util.Scanner;
5.
6. public class TugasAlproPekan6_2511532024 {
7.
8.     public static void main(String[] args) {
9.         Scanner scan = new Scanner(System.in);
10.        Random rand = new Random();
11.        String jawab ;
12.        int tries = 0;
13.        int sum = 0;
14.        boolean running = false;
15.
16.        do {
17.
18.            int dadu1 = rand.nextInt(6) + 1;
19.            int dadu2 = rand.nextInt(6) + 1;
20.            sum = dadu1 + dadu2;
21.            tries++;
22.
23.            System.out.println(dadu1 + " + " +
dadu2 + " = " + sum);
24.
25.            if (sum == 7) {
26.                System.out.println("Tebakan Anda
Benar!");
27.                System.out.println("Anda menang
setelah "+tries+" percobaan!");
28.                running = true;
29.            } else {
30.                System.out.println("Tebakan Anda
Salah!");
31.                System.out.print("Apakah mau
lempar dadu? (Ya / Tidak) ");
32.                jawab = scan.nextLine();
33.                if(!jawab.equalsIgnoreCase("ya"))
{
34.                    System.out.println("Anda
gagal menang");
35.                    break;
36.                }
37.            }
38.
39.        } while (!running);
40.        scan.close();
41.    }
42.
43. }
```



## 2.4 Output Program

```

jumlahan_2511532024.java      1 + 2 = 3
dadu_2511532024.java          Tebakkan Anda Salah!
doWhile1_2511532024.java      Apakah mau lempar dadu? (Ya / Tidak) ya
oop_2511532024.java          1 + 2 = 3
proPekan6_2511532024.java     Tebakkan Anda Salah!
                                Apakah mau lempar dadu? (Ya / Tidak) ya
                                5 + 3 = 8
                                Tebakkan Anda Salah!
                                Apakah mau lempar dadu? (Ya / Tidak) ya
                                2 + 3 = 5
                                Tebakkan Anda Salah!
                                Apakah mau lempar dadu? (Ya / Tidak) ya
                                4 + 1 = 5
                                Tebakkan Anda Salah!
                                Apakah mau lempar dadu? (Ya / Tidak) ya
                                1 + 4 = 5
                                Tebakkan Anda Salah!
                                Apakah mau lempar dadu? (Ya / Tidak) ya
                                2 + 4 = 6
                                Tebakkan Anda Salah!
                                Apakah mau lempar dadu? (Ya / Tidak) ya
                                5 + 2 = 7
                                Tebakkan Anda Benar!
                                Anda menang setelah 8 percobaan!

```

Program “Tebak dadu menggunakan perulangan uncounted” merupakan sebuah simulasi permainan sederhana yang menggunakan dua buah dadu untuk menghasilkan angka acak. Ketika program dijalankan, sistem secara otomatis melempar dua dadu dengan memanfaatkan kelas Random untuk menghasilkan nilai antara 1 hingga 6 pada masing-masing dadu. Hasil dari kedua dadu tersebut kemudian dijumlahkan dan ditampilkan di layar dalam format “dadu1 + dadu2 = sum”. Program ini memiliki tujuan agar pengguna dapat menebak apakah hasil penjumlahan kedua dadu tersebut sama dengan tujuh. Jika hasilnya tepat tujuh, program akan menampilkan pesan bahwa tebakan pengguna benar dan menampilkan jumlah percobaan yang telah dilakukan hingga berhasil. Namun, apabila hasilnya bukan tujuh, maka program akan menampilkan pesan bahwa tebakan salah dan memberikan pilihan kepada pengguna untuk melanjutkan atau menghentika permainan.

Struktur utama dari program ini menggunakan do-while, termasuk dalam kategori perulangan uncounted karena jumlah perulangannya tidak ditentukan sejak awal dan akan terus berjalan selama kondisi belum terpenuhi. Setiap kali pengguna memilih untuk melanjutkan, program akan kembali melempar dadu dan menampilkan hasil acak yang baru. Jika pengguna memilih berhenti, maka permainan akan dihentikan.

dengan pesan bahwa pengguna gagal menang. Melalui alur ini, program menunjukkan penerapan konsep dasar logika pemrograman yang meliputi penggunaan perulangan, percabangan kondisi (if-else), serta pengambilan input pengguna menggunakan class Scanner.

Secara keseluruhan, program ini menghasilkan output yang bersifat interaktif dan dinamis karena nilai dadu selalu berubah setiap kali dijalankan. Hal ini menggambarkan penerapan logika acak dan kendali alur menggunakan kondisi logis dalam bahasa pemrograman java. Dengan kombinasi antara proses acak, perulangan tanpa batas pasti, dan interaksi langsung dengan pengguna, program ini menjadi contoh sederhana namun efektif dalam memahami dasar-dasar pemrograman terstruktur menggunakan java.

## **BAB III**

### **PENUTUP**

#### **3.1 Kesimpulan**

Program tebak dadu menggunakan perulangan uncounted dalam bahasa java merupakan implementasi sederhana dari konsep dasar logika pemrograman, khususnya penggunaan perulangan, percabangan, dan input pengguna. Melalui program ini, pengguna dapat memahami bagaimana perulangan do-while bekerja dengan kondisi tertentu terpenuhi, yaitu ketika jumlah kedua dadu bernilai tujuh. Program ini juga menunjukkan penerapan Class Random untuk menghasilkan nilai acak serta penggunaan Class Scanner untuk menerima input dari pengguna, sehingga tercipta interaksi dua arah antara sistem dan pemain. Dari hasil pengujian, program dapat berjalan dengan baik dan menampilkan hasil yang berbeda setiap kali dijalankan karena nilai acaknya selalu berubah.

#### **3.2 Saran**

Untuk mengembangkan lebih lanjut, program ini dapat ditingkatkan dengan menambahkan fitur tambahan seperti sistem skor, batas maksimum percobaan, atau tampilan hasil dalam bentuk tabel agar lebih informatif. Selain itu, dapat pula ditambahkan validasi input agar program tetap berjalan dengan aman meskipun pengguna memasukkan data yang tidak sesuai. Dari sisi tampilan, penggunaan antarmuka grafis (GUI) juga dapat menjadi alternatif menarik agar permainan terlihat interaktif dan mudah digunakan.

## DAFTAR PUSTAKA

- [1] Oracle, *The Java™ Tutorials: Learning the Java Language*, Oracle Corporation, 2023. [Online]. Available: <https://docs.oracle.com/javase/tutorial/>
- [2] Oracle, *Class Random (Java Platform SE 8)*, Oracle Corporation, 2023. [Online]. Available: <https://docs.oracle.com/javase/8/docs/api/java/util/Random.html>
- [3] Oracle, *Class Scanner (Java Platform SE 8)*, Oracle Corporation, 2023. [Online]. Available: <https://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html>
- [4] Oracle, *The while and do-while Statements*, Oracle Corporation, 2023. [Online]. Available: <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/while.html>