

LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN
PERULANGAN WHILE DAN DO WHILE



Oleh: Afif Naufal Zahran

NIM: 2511533009

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

FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN INFORMATIKA
UNIVERSITAS ANDALAS

Kata Pengantar

Pedoman ini disusun sebagai rujukan resmi bagi mahasiswa Departemen Informatika dalam penyusunan laporan praktikum pada mata kuliah *Pemrograman Dasar dengan Java*. Dokumen ini tidak hanya memberikan gambaran umum mengenai format penulisan, tetapi juga menguraikan secara rinci sistematika laporan, tata cara penyajian isi, serta contoh penulisan kode program yang dilengkapi dengan referensi ilmiah. Melalui panduan ini, mahasiswa diharapkan mampu menyusun laporan yang tidak sekadar memenuhi aspek administratif, tetapi juga mencerminkan ketelitian, keteraturan, dan penerapan kaidah penulisan akademik pada tingkat dasar. Dengan demikian, laporan praktikum yang dihasilkan dapat berfungsi sebagai media pembelajaran, dokumentasi kegiatan, sekaligus sarana untuk melatih keterampilan menulis ilmiah yang akan bermanfaat dalam jenjang studi selanjutnya.

Padang, 2025

Tim Penyusun

DAFTAR ISI

BAB I	
1.1 Latar Belakang.....	
1.2 Tujuan	
1.3 Manfaat	
BAB II	
2,1 Teori.....	
2,2 Program.....	
BAB III.....	
3,1 Kesimpulan.....	
3,2 Saran	

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada bahasa pemograman ketika kita ingin menjalankan suatu program berulang kali dengan batas yang ditentukan atau hingga suatu kondisi terpenuhi kita dapat menggunakan perulangan dengan *keyword* yang telah disediakan dari bahasa pemograman yang kita gunakan. Untuk *Java* sendiri kita dapat menggunakan perulangan *For* dan *While*. Pada pertemuan minggu ke-6 kita akan membahas perulangan *while* atau *do while*.

1.2 Tujuan

Tujuan dari laporan praktikum ini agar pembaca dan penulis dapat mendalami dan memahami tentang perulangan *while* atau *do while* dan implementasinya di bahasa pemograman *Java*.

1.3 Manfaat

Manfaat dari praktikum tentang perulangan *while* agar pembaca mendapatkan pengetahuan tentang perulangan *While* atau *Do While*. Manfaat untuk penulis agar dapat memahami *While* pada bahasa pemograman *Java*.

BAB II

PEMBAHASAN

2,1 Teori

Perulangan *While* adalah perulangan yang biasa digunakan untuk mengulangi perintah hingga suatu statement *true*. Struktur *While* lebih efisien dari lainnya karena simple.

Berikut contoh sintaks untuk kode *While* di *Java*:

```
while(statement){  
    }
```

atau

```
do{  
    }while(statement);
```

Atau kode java nya:

```
while(true){  
    //Kode yang akan diulang sebanyak 10 kali  
}
```

2,2 Program

Pada pertemuan pekan ke-6 terdapat 5 program yang di buat yakni GamePenjumlahan.java, Lempardadu.java, SentinelLoop.java, doWhile1.java, perulanganWhile1.java.

a. GamePenjumlahan.java

```
4  public class GamePenjumlahan_2511533009 {
5
6  public static void main(String[] args) {
7      Scanner console = new Scanner(System.in);
8      Random rand = new Random();
9
10     int points = 0;
11     int wrong = 0;
12     while (wrong < 3) {
13         int result = play(console, rand);
14         if (result > 0){
15             points++;
16         } else {
17             wrong++;
18         }
19     }
20     System.out.println("You earned " + points + " total points.");
21 }
22
23 public static int play(Scanner console, Random rand) {
24     int operand = rand.nextInt(4) + 2;
25     int sum = rand.nextInt(10) + 1;
26     System.out.print(sum);
27     for(int i = 2; i <= operand; i++) {
28         int n = rand.nextInt(10) + 1;
29         sum += n;
30         System.out.print(" + " + n);
31     }
32     System.out.print(" = ");
33
34     // read user's guess and report whether it was correct
35     int guess = console.nextInt();
36     if (guess == sum) {
37         return 1;
38     } else {
39         System.out.println("Wrong! The answer was " + sum);
40         return 0;
41     }
42 }
43
44 }
```

gambar 2.1

Pada gambar 2.1 merupakan contoh penggunaan perulangan *while* dimana jika variabel *wrong* di atas 3 perulangan akan berhenti.

b. Lempardadu.java

```
4  public class Lempardadu_2511533009 {
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 }
```

gambar 2.2

Pada gambar 2.2 merupakan contoh lain penggunaan while loop dengan kondisi sum tidak sama dengan 7.

c. SentinelLoop.java

```
3  public class SentinelLoop_2511533009 {
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 }
```

gambar 2.3

Pada gambar berikut contoh penggunaan sentinel loop dimana program menerima input dan jika input tersebut bernilai 0 maka perulangan akan berhenti.

d. doWhile1.java

```
1  package pekan6_2511533009;
2  import java.util.Scanner;
3  ▼ public class doWhile1_2511533009 {
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 }
```

gambar 2.4

Pada program berikut merupakan contoh penggunaan *do while* dimana perulangan berikut sama seperti perulangan *while* sebelumnya namun diawali dengan “do{}”.

e. perulanganWhile1.java

```
3  public class perulanganWhile1_2511533009 {
4
5  public static void main(String[] args) {
6
7      int counter=0;
8      String jawab;
9      boolean running=true;
10     //deklarasi scanner
11     Scanner scan = new Scanner(System.in);
12     while (running){
13         counter++;
14         System.out.println("Jumlah = "+counter);
15         System.out.print("Apakah lanjut (ya / tidak?)");
16         jawab=scan.nextLine();
17         //cek jawaban - tidak, perulangan berhenti
18         if(jawab.equalsIgnoreCase("tidak")){
19             running=false;
20         }
21     }
22     System.out.println("Anda sudah melakukan perulangan sebanyak "+counter+" kali");
23 }
24
25 }
```

gambar 2.5

Pada gambar 2.5 merupakan penggunaan perulangan *while* dengan kondisi jika input dari pengguna “tidak” nilai variabel *running* akan berubah menjadi *false* dan menghentikan perulangan.

BAB III

PENUTUPAN

3,1 Kesimpulan

Perulangan *while* pada bahasa pemrograman *Java* sangat penting karena hal yang sangat fundamental pada bahasa pemrograman terutama bahasa pemrograman *Java*.

3,2 Saran

Laporan praktikum ini masih memiliki kekurangan. Karena itu, penulis sangat terbuka terhadap saran dan kritikan agar dapat meningkatkan kualits laporan ini dan laporan-laporan selanjutnya.