Laporan Praktikum

Algoritma Pemrograman



Disusun Oleh :

**Ahda Rindang Al-Amin (2311531003)**

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

Asisten Labor : Rahmad Dwirizki Olders

Departemen Informatika

Fakultas Teknologi Informasi

Universitas Andalas

Tahun 2023

**Perulangan *do-while* dan *while***

1. **Tujuan Praktikum**
2. Memahami perulangan *do-while* pada bahasa pemrograman Java.
3. Memahami perulangan *while* pada bahasa pemrograman Java.
4. **Pendahuluan**

 [Diterjemahkan dari bahasa Inggris ke bahasa Indonesia - www.onlinedoctranslator.com](https://www.onlinedoctranslator.com/id/?utm_source=onlinedoctranslator&utm_medium=docx&utm_campaign=attribution)

 [Diterjemahkan dari bahasa Inggris ke bahasa Indonesia - www.onlinedoctranslator.com](https://www.onlinedoctranslator.com/id/?utm_source=onlinedoctranslator&utm_medium=docx&utm_campaign=attribution)

1. Perulangan *while*

Perulangan Java lainnya adalah **while**. Bentuk umum dari perulangan **while** adalah

while(*kondisi*) *pernyataan*;

dimana *pernyataan* dapat berupa pernyataan tunggal atau blok pernyataan, dan *kondisi* mendefinisikan kondisi yang mengendalikan perulangan. Kondisinya bisa berupa ekspresi Boolean apa pun yang valid. Perulangan berulang selama kondisinya benar. Ketika kondisi menjadi salah, kendali program diteruskan ke baris yang segera mengikuti perulangan.

Seperti halnya perulangan **for**, **while** memeriksa ekspresi kondisional di bagian atas perulangan, yang berarti bahwa kode perulangan mungkin tidak dijalankan sama sekali. Hal ini menghilangkan kebutuhan untuk melakukan tes terpisah sebelum loop.

1. Perulangan *do-while*

Perulangan Java yang terakhir adalah **do-while**. Berbeda dengan perulangan **for** dan **while** yang kondisinya diuji di bagian atas perulangan, perulangan **do-while** memeriksa kondisinya di bagian bawah perulangan. Artinya, perulangan **do-while** akan selalu dijalankan minimal satu kali. Bentuk umum dari perulangan **do-while** adalah

do {

*pernyataan*;

} while(*kondisi*);

Meskipun kurung kurawal tidak diperlukan ketika hanya ada satu pernyataan, kurung kurawal sering digunakan untuk meningkatkan keterbacaan konstruksi **do-while**, sehingga mencegah kebingungan dengan while. Perulangan **do-while** dijalankan selama ekspresi kondisinya benar.

1. **Metode Praktikum**
2. Perulangan *do-while*

Pada program ini, perulangan terletak pada counter yang terus bertambah berdasarkan kondisi selama string jawab yang diinput adalah ‘ya’. Jika input diisi dengan ‘tidak’, maka perulangan akan terhenti.

public class dowhile1 {

public static void main(String[] args) {

int counter=0;

String jawab;

boolean running=true;

Scanner scan = new Scanner(System.***in***);

do {

counter++;

System.***out***.println("Jumlah= " + counter);

System.***out***.print("Apakah lanjut? (ya/tidak)");

jawab = scan.next();

if (jawab.equalsIgnoreCase("tidak")) {

running = false;

}

} while (running) ;

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

scan.close();

}

}

Contoh output dari program diatas adalah sebagai berikut. Dapat dilihat jumlah counter bertambah selama input adalah ‘ya’. Saat input diisi dengan ‘tidak’, perulangan terhenti dan dilanjutkan pada statement setelahnya.

Jumlah= 1

Apakah lanjut? (ya/tidak)ya

Jumlah= 2

Apakah lanjut? (ya/tidak)ya

Jumlah= 3

Apakah lanjut? (ya/tidak)ya

Jumlah= 4

Apakah lanjut? (ya/tidak)tidak

Anda sudah melakukan perulangan sebanyak 4 kali

Program diatas dapat dimodifikasi dengan menghilangkan boolean running dan pernyataan kondisional **if**. Caranya dengan memindahkan kondisi dalam **if** menjadi kondisi  **while**. Berikut adalah program yang telah dimodifikasi :

import java.util.Scanner;

public class DoWhile2 {

public static void main(String[] args) {

int counter=0;

String jawab;

Scanner scan = new Scanner(System.***in***);

do {

counter++;

System.***out***.println("Jumlah= " + counter);

System.***out***.print("Apakah lanjut? (ya/tidak)");

jawab = scan.next();

} while (!jawab.equalsIgnoreCase("tidak") ) ;

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

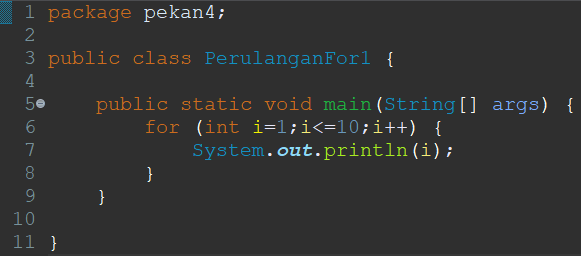
scan.close();

}

}

Output dari program diatas sama dengan output program sebelum dimodifikasi.

Program perulangan yang menggunakan sintaks **for** dapat diubah menjadi program perulangan **do-while** atau **while**. Program selanjutnya adalah modifikasi program perulangan **for** berikut :



Jika program tersebut diubah menjadi program perulangan **do-while**, maka akan seperti program berikut :

public class dowhile3 {

public static void main(String[] args) {

int i=1;

do {

System.***out***.println(i);

i++;

} while (i<=10) ;

}

}

Pada perulangan **for**, inisialisasi, kondisi, dan iterasi diletakkan bersejajar disebelah **for**. Pada perulangan **do-while**, inisialisasi ditulis bebas, seperti inisialisasi data biasa, lalu kondisi diletakkan dalam kurung setelah **while**, dan iterasi ditulis dalam kurung kurawal **do-while**. Walaupun menggunakan sintaks perulangan yang berbeda, output yang dihasilkan tetap sama :

1

2

3

4

5

6

7

8

9

10

1. Perulangan *while*

Jika program sebelumnya diubah menjadi perulangan **while**, maka letak inisialisasi, kondisi dan iterasi sama seperti perulangan **do-while.** Dimana inisialisasi ditulis bebas, kondisi diletakkan dalam kurung setelah **while**, dan iterasi ditulis dalam kurung kurawal **while**. Output yang dihasilkan juga sama dengan program yang menggunakan sintaks perulangan lain. Berikut program yang telah dimodifikasi menjadi menggunakan perulangan **while** :

public class while1 {

public static void main(String[] args) {

int i=1;

while (i<=10) {

System.***out***.println(i);

i++;

}

}

}

Selanjutnya, program berikut adalah modifikasi dari perulangan **do-while** yang pertama menjadi perulangan **while**:

import java.util.Scanner;

public class While2 {

public static void main(String[] args) {

int counter=0;

String jawab="ya";

Scanner scan = new Scanner(System.***in***);

while (!jawab.equalsIgnoreCase("tidak") ) {

counter++;

System.***out***.println("Jumlah= " + counter);

System.***out***.print("Apakah lanjut? (ya/tidak)");

jawab = scan.next();

}

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

scan.close();

}

}

Perbedaannya, nilai String jawab, yang menjadi kondisi **while**, harus diinisialisasi dahulu sebelum dilakukan perulangan. Output dari program tersebut sama dengan output program yang menggunakan perulangan **do-while**. Berikut output nya:

Jumlah= 1

Apakah lanjut? (ya/tidak)ya

Jumlah= 2

Apakah lanjut? (ya/tidak)ya

Jumlah= 3

Apakah lanjut? (ya/tidak)ya

Jumlah= 4

Apakah lanjut? (ya/tidak)tidak

Anda sudah melakukan perulangan sebanyak 4 kali

Program selanjutnya adalah modifikasi dari program praktikum pekan ketiga. Pada pekan ketiga, program hanya menerapkan struktur kondisional untuk menentukan dewasa atau tidak, dan boleh atau tidaknya membawa motor. Dengan menerapkan perulangan, penentuan dapat dilakukan berulang-ulang. Pertanyaan umur dan kepemilikan sim akan berulang jika kondisi yang dibutuhkan untuk melakukan perulangan terpenuhi. Berikut programnya:

package pekan5;

import java.util.Scanner;

public class While3 {

public static void main(String[] args) {

int umur;

char sim;

String jawab="ya";

Scanner input = new Scanner(System.***in***);

while(!jawab.equalsIgnoreCase("tidak")) {

System.***out***.println("Berapa Umurmu?");

umur=input.nextInt();

System.***out***.println("Apakah anda sudah memiliki SIM C?");

sim=input.next().charAt(0);

if((umur>=17)&&(sim=='y')) {

System.***out***.println("Anda sudah dewasa dan boleh bawa motor");

} if ((umur>=17)&&(sim!='y')) {

System.***out***.println("Anda sudah dewasa, tetapi tidak boleh bawa motor");

} if ((umur<17)&&(sim!='y')) {

System.***out***.println("Anda masih bocah dan belum boleh bawa motor");

}

System.***out***.println("ulang? (ya/tidak)");

jawab=input.next();

}

System.***out***.println("Terimakasih");

input.close();

}

}

Selama String jawab yang diinput bukan ‘tidak’, maka pertanyaan umur dan kepemilikan sim akan diulang kembali. Jika String jawab yang diinput adalah ‘tidak’ maka perulangan akan terhenti dan menjalankan pernyataan setelahnya. Berikut contoh output dari program di atas:

Berapa Umurmu?

18

Apakah anda sudah memiliki SIM C?

y

Anda sudah dewasa dan boleh bawa motor

ulang? (ya/tidak)

ya

Berapa Umurmu?

23

Apakah anda sudah memiliki SIM C?

t

Anda sudah dewasa, tetapi tidak boleh bawa motor

ulang? (ya/tidak)

tidak

Terimakasih

1. **Kesimpulan Praktikum**

Pada bahasa pemrograman Java, anda dapat mengeksekusi serangkaian kode berulang kali dengan membuat loop. Perulangan digunakan kapan pun Anda perlu melakukan tugas yang berulang karena jauh lebih sederhana dan mudah daripada mencoba menulis urutan pernyataan yang sama berulang kali. Java menyediakan beragam konstruksi loop yang kuat, yaitu perulangan **for**, **do-while**, dan **while**.

Perulangan **for** dipakai jika kondisi kapan perulangan dihentikan dan banyaknya perulangan diketahui. Perulangan **while** digunakan jika kondisi kapan perulangan dihentikan diketahui, namun banyaknya proses perulangan tidak diketahui. Pada perulangan **do-while**, hampir sama dengan perulangan **while** yaitu sebuah perulangan yang kondisi berhenti sudah di ketahui tetapi banyaknya proses perulangan yang dilakuakan belum di ketahui. perbedaannya dengan **while** yaitu perulangan **do-while** pasti akan dilakukan minimal satu kali. hal ini terjadi karena pengecekan kondisi dilakukan diakhir statemen.