

LAPORAN PRAKTIKUM ALGORITMA PEMROGRAMAN
STRUKTUR KONTROL DAN LOOPING



Muhamad farhansyah
201021400028
01SIFM001

STIMIK ERESHA
SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN
KOMPUTER ERESHA PAMULANG

Pendahuluan

Algoritma dan pemrograman merupakan ilmu yang sangat esensial dalam membuat program. Dengan pemrograman itu sendiri, sebuah program bisa disusun dibantu dengan algoritma. Algoritma membantu penyelesaian masalah yang ingin dipecahkan untuk membuat sebuah program, kemudian program disusun oleh programmer dengan mengcompile source code

menjadi sebuah program yang dapat dieksekusi. Algoritma dapat disusun dalam bentuk pseudo code maupun flowchart. Pseudo code adalah Kode atau tanda yang menyerupai (pseudo) program atau merupakan penjelasan cara menyelesaikan suatu masalah. Pseudo-code sering digunakan oleh manusia untuk menuliskan algoritma. Sementara flowchart adalah suatu teknik untuk menyusun rencana program yang telah dipergunakan oleh kalangan programmer komputer sebelum algoritma menjadi populer. Flowchart menggunakan simbol gambar yang menunjukkan aliran dari proses terhadap data. Listing pemrograman tersebut ditulis dalam bahasa pemrograman. Terdapat macam macam bahasa pemrograman, dari tingkat rendah seperti bahasa mesin sampai bahasa tingkat tinggi seperti Pascal, C, Delphi dll. Setiap bahasa pemrograman memiliki kelemahan dan kelebihan masing-masing. Tinggal peran programmer itu sendiri yang memilih bahasa pemrograman yang sesuai dengan kebutuhannya

Perulangan data atau yang biasa disebut dengan “looping” adalah proses yang dilakukan secara berulang-ulang sampai batas yang ditentukan. Biasanya bila dalam perulangan tersebut tidak disertakan batasnya maka syntax akan error karena proses itu akan berulang terus hingga tak terhingga sementara variabel dalam komputer masih terbatas.

Teori

1. Struktur Kontrol

Pada bagian ini, kita akan mempelajari tentang struktur kontrol dimana kita dapat mengubah cara eksekusi pada pernyataan yang dibuat diprogram kita. Struktur kontrol adalah pernyataan dari Java yang memungkinkan user untuk memilih dan mengeksekusi blok kode spesifik dan mengabaikan blok kode yang lain.

1.1 Statement IF

Pernyataan if akan menentukan sebuah pernyataan (atau blok kode) yang akan eksekusi jika dan hanya jika persyaratan bernilai benar(true).

1.2 Statement IF-ELSE

Pernyataan IF-ELSE digunakan apabila kita ingin mengeksekusi beberapa pernyataan dengan kondisi true dan pernyataan yang lain dengan kondisi false.

1.3 Statement IF-ELSE-IF

Pernyataan pada bagian kondisi else dari blok if-else dapat menjadi struktur if-else yang lain. Kondisi struktur seperti ini memungkinkan kita untuk membuat seleksi persyaratan yang lebih kompleks.

1.4 Statement SWITCH

Cara lain untuk membuat cabang adalah dengan menggunakan kata kunci switch. Switch mengkonstruksikan cabang untuk beberapa kondisi dari nilai

switch_expression adalah ekspresi integer atau karakter dan case_selector1, case_selector2 dan seterusnya adalah konstanta unik dari nilai integer atau karakter

2. Looping/Pengulangan

Pengulangan adalah berupa pernyataan dari Java yang memungkinkan kita untuk mengeksekusi blok code berulang-ulang sesuai dengan jumlah tertentu yang diinginkan. Ada tiga macam jenis dari struktur pengulangan yaitu while, do-while, dan for-loops.

2.1 While Loop

Pernyataan while loop adalah pernyataan atau blok pernyataan yang diulang-ulang sampai mencapai kondisi yang cocok. Bentuk pernyataan while,

Pernyataan didalam while loop akan dieksekusi berulang-ulang selama kondisi boolean_expression bernilai benar (true).

2.2 Do-While Loop

Do-while loop mirip dengan while-loop. Pernyataan di dalam do-while loop akan dieksekusi beberapa kali selama kondisi bernilai benar(true). Perbedaan antara while dan do-while loop adalah dimana pernyataan di dalam do-while loop akan dieksekusi sedikitnya satu kali. Perbedaan antara while dan do-while loop adalah dimana pernyataan di dalam do-while loop akan dieksekusi sedikitnya satu kali

2.3 For Loop

Pernyataan for loop memiliki kondisi hampir mirip seperti struktur pengulangan sebelumnya yaitu melakukan pengulangan untuk mengeksekusi kode yang sama sebanyak jumlah yang telah ditentukan

2.4 Percabangan dalam Pengulangan

Pernyataan percabangan memungkinkan kita untuk mengatur aliran eksekusi program. Java memberikan tiga bentuk pernyataan percabangan: break, continue dan return.

Penjelasan

1. Statement IF

```
package nilai_terbesar;

/**
 *
 * @author Boang
 */
import java.util.Scanner;
public class Nilai_Terbesar {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        Scanner keyboard = new Scanner(System.in);
        int Nilai_1, Nilai_2, Nilai_3;

        System.out.print("Masukan Nilai 1 = ");
        Nilai_1 = keyboard.nextInt();
        System.out.print("Masukan Nilai 2 = ");
        Nilai_2 = keyboard.nextInt();
        System.out.print("Masukan Nilai 2 = ");
        Nilai_3 = keyboard.nextInt();
        if(Nilai_1 < Nilai_2)
        {
            if(Nilai_1 < Nilai_3)
            System.out.print("Nilai Terkecil adalah Nilai 1 = " + Nilai_1 + "\n");
            else
            System.out.print("Nilai Terkecil adalah Nilai 3 = " + Nilai_3 + "\n");
        }
        else
        {
            if(Nilai_2 < Nilai_3)
            System.out.print("Nilai Terkecil adalah Nilai 2 = " + Nilai_2 + "\n");
            else
            System.out.print("Nilai Terkecil adalah Nilai 3 = " + Nilai_3 + "\n");
        }
    }
}
```

run:

Masukan Nilai 1 = 20

Masukan Nilai 2 = 12


Masukan Nilai 2 = 3

Nilai Terkecil adalah Nilai 3 = 3

BUILD SUCCESSFUL (total time: 3 minutes 2 seconds)

2. Statement IF ELSE

SourceHistory



```
5  */
6
7  package angka.terkecil;
8
9  /**
10   *
11   * @author Boang
12   */
13  import java.util.Scanner;
14  public class AngkaTerkecil {
15
16      /**
17       * @param args the command line arguments
18       */
19      public static void main(String[] args) {
20          // TODO code application logic here
21          Scanner keyboard = new Scanner(System.in);
22          int Nilai_1, Nilai_2, Nilai_3;
23
24          System.out.print("Masukan Nilai 1 = ");
25          Nilai_1 = keyboard.nextInt();
26          System.out.print("Masukan Nilai 2 = ");
27          Nilai_2 = keyboard.nextInt();
28          System.out.print("Masukan Nilai 3 = ");
29          Nilai_3 = keyboard.nextInt();
30          if(Nilai_1 < Nilai_2)
31          {
32              if(Nilai_1 < Nilai_3)
33                  System.out.print("Nilai Terkecil adalah Nilai 1 = " + Nilai_1 + "\n");
34              else
35                  System.out.print("Nilai Terkecil adalah Nilai 3 = " + Nilai_3 + "\n");
36          }
37          else
38          {
39              if(Nilai_2 < Nilai_3)
40                  System.out.print("Nilai Terkecil adalah Nilai 2 = " + Nilai_2 + "\n");
41              else
42                  System.out.print("Nilai Terkecil adalah Nilai 3 = " + Nilai_3 + "\n");
43          }
44      }
45
46  }
47
```

Masukan Nilai 1 = 20
Masukan Nilai 2 = 12
Masukan Nilai 3 = 3
Nilai Terkecil adalah Nilai 3 = 3
BUILD SUCCESSFUL (total time: 9 seconds)

3. Statement IF-ELSE-IF

```
1  import java.util.*;
2  public class Nilaimahasiswa{
3
4      public static void main(String[] args) {
5          Scanner Keyboard = new Scanner(System.in);
6
7          int nilai;
8          String nama;
9
10         System.out.print("Masukan Nama Mahasiswa =");
11         nama = Keyboard.nextLine();
12         System.out.print("Masukan Nilai Mahasiswa = ");
13         nilai = Keyboard.nextInt();
14
15         if (nilai >= 36 && nilai <= 45)
16             System.out.print(nama+" Mendapatkan D ");
17         else
18             if (nilai >= 46 && nilai <= 55 )
19                 System.out.println(nama+" Mendapatkan C ");
20             else
21                 if (nilai >= 56 && nilai <= 65 )
22                     System.out.println(nama+" Mendapatkan C+ ");
23                 else
24                     if (nilai >= 66 && nilai <= 75 )
25                         System.out.println(nama+" Mendapatkan B ");
26                     else
27                         if (nilai >= 76 && nilai <= 85 )
28                             System.out.println(nama+" Mendapatkan B+ ");
29                         else
30                             if (nilai >= 86 && nilai <= 100 )
31                                 System.out.println(nama+" Mendapatkan A ");
32                             else
33                                 System.out.println("NILAI TIDAK VALID ");
34
35
36     }
37 }
```

```
C:\Users\Boang\Documents\Latihan Java>java Nilaimahasiswa
Masukan Nama Mahasiswa =ADI
Masukan Nilai Mahasiswa = 70
ADI Mendapatkan B
```

```
C:\Users\Boang\Documents\Latihan Java>java Nilaimahasiswa
Masukan Nama Mahasiswa =Budi
Masukan Nilai Mahasiswa = 65
Budi Mendapatkan C+
```

```
C:\Users\Boang\Documents\Latihan Java>java Nilaimahasiswa
Masukan Nama Mahasiswa =Caca
Masukan Nilai Mahasiswa = 90
Caca Mendapatkan A
```

```
C:\Users\Boang\Documents\Latihan Java>java Nilaimahasiswa
Masukan Nama Mahasiswa =Deny
Masukan Nilai Mahasiswa = 75
Deny Mendapatkan B
```

4. While Loop

```
package do_while_01;

/**
 *
 * @author Boang
 */
public class Do_while_01 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        int x = 0;
        do
        {
            System.out.println( x );
            x++;
        }
        while (x < 10 );
    }

}
```

```
compile:
run:
0
1
2
3
4
5
6
7
8
9
BUILD SUCCESSFUL (total time: 1 second)
|
```


5. For looping

```
package for_01;

/**
 *
 * @author Boang
 */
public class For_01 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        for( int i = 0; i < 10; i++ )
        {
            System.out.print( i );
        }
    }

}
```

```
Compiling 1 source file to C:\Users\Boang\Documents\
compile:
run:
0123456789BUILD SUCCESSFUL (total time: 1 second)
```

6. Percabangan dalam Perulangan

```
package for_if_break_search_label_01;

/**
 *
 * @author Boang
 */
public class For_if_break_search_label_01 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        searchLabel:
        for(int i=0;i<6;i++)
        {
            if (i==5)
            {
                System.out.println(i);
                break searchLabel;
            }
        }
    }
}
```

```
run:
5
BUILD SUCCESSFUL (total time: 1 second)
|
```

Kesimpulan

Struktur Kontrol Percabangan dan Looping sangat penting dan mempermudah seorang programmer melakukan atau menjalankan suatu program dan sangat berguna juga di pakai untuk melakukan pendataan pada suatu bidang tertentu.