

Nama :Muhammad Shabran
Nim :13020220056
Kelas :A2
Tugas4 :PBO

1. Apakah perbedaan antara struktur kontrol percabangan if-else dan switch-case?

Jawab:

If Else merupakan pilihan alternatif dari penyelesaian yang digunakan untuk mencari kondisi yang diinginkan. Kalau kita artikan ke bahasa Indonesia if sendiri berarti jika, sedangkan else adalah yang lain. Sesuai artinya, if else ini untuk menentukan kondisi, misal jika bukan A berarti hasilnya yang lain.

If sendiri diletakan setelah pernyataan if, kemudian else merupakan pilihan terakhir yang akan dijalankan jika semua pilihan tidak memiliki nilai benar pada kondisi yang ada atau yang sudah ditentukan.

Switch Case

Cara kerjanya hampir sama dengan if else di mana switch case merupakan suatu kondisi percabangan di mana jika suatu kondisi benar maka blok bernilai benar akan dijalankan.

Perbedaan mendasar antara if dan switch yaitu switch case ini digunakan untuk mengecek data yang tipenya karakter dan juga integer. Sementara pada if else pengecekan bisa menggunakan simbol atau operator seperti <, >, =, ==.

Switch sendiri digunakan untuk memilih kondisi yang biasanya memasukkan variabel. Lalu case digunakan sebagai kondisi-kondisi. Jika suatu kondisi bernilai benar maka blok yang benar (true) akan ditampilkan.

2. Kapan digunakan struktur kontrol if-else dan switch-case

Jawab:

If-else: Struktur if-else digunakan ketika Anda memiliki kondisi yang harus dievaluasi terhadap satu atau lebih ekspresi boolean. If-else memungkinkan Anda untuk membuat keputusan yang berbeda tergantung pada apakah kondisi yang dievaluasi adalah benar (true) atau salah (false). Jika kondisi yang diberikan benar, blok kode dalam bagian if akan dieksekusi. Jika tidak, maka blok kode dalam bagian else akan dieksekusi.

Switch-case: Struktur switch-case digunakan ketika Anda memiliki satu variabel dan ingin mengevaluasi nilai variabel tersebut terhadap beberapa kasus yang mungkin terjadi. Jika nilai variabel cocok dengan salah satu kasus, blok kode terkait dengan kasus tersebut akan dieksekusi. Switch-case sangat berguna ketika Anda memiliki banyak pilihan atau kasus yang mungkin dan ingin mengeksekusi kode berdasarkan nilai variabel. Namun, tidak semua bahasa pemrograman mendukung switch-case, dan beberapa bahasa memiliki pengecualian tertentu dalam penggunaannya.

3. Pada program 2, tambahkan perintah untuk memilih 2 opsi menggunakan kontrol switch..case.

opsi pilihah 1=inputNilai()

Pilihan 2=inputNilaiBaru()

Jawab:

=syntax

```
1 package pertemuan2.modul3.nilai;
2
3 import java.util.Scanner;
4
5 public class TestNilai {
6
7     Run | Debug
8     public static void main(String[] args) {
9         Scanner input = new Scanner(System.in);
10
11         System.out.print(s:"Masukkan Jumlah Data : ");
12         int banyakData = input.nextInt();
13         int nilai[] = new int[banyakData];
14
15         // Menampilkan menu pilihan
16         System.out.println(x:"Pilihan: ");
17         System.out.println(x:"1. Input Nilai");
18         System.out.println(x:"2. Input Nilai Baru");
19         System.out.print(s:"Masukkan Pilihan : ");
20         int pilihan = input.nextInt();
21
22         HitungRata hitung = new HitungRata(input);
23
24         // Memilih opsi dengan switch-case
25         switch (pilihan) {
26             case 1:
27                 hitung.inputNilai(nilai);
28                 System.out.println(x:"Daftar Nilai : ");
29                 hitung.cetakNilai(nilai);
```

```

25         case 1:
26             hitung.inputNilai(nilai);
27             System.out.println(x:"Daftar Nilai : ");
28             hitung.cetakNilai(nilai);
29             System.out.println("Rata Nilai : " + hitung.rataNilai(banyakData));
30             break;
31         case 2:
32             hitung.inputNilaiBaru(banyakData);
33             System.out.println(x:"Daftar Nilai Baru : ");
34             hitung.cetakNilaiBaru();
35             break;
36         default:
37             System.out.println(x:"Pilihan tidak valid!");
38     }
39 }
40 }
41
42 class HitungRata {
43
44     private Scanner input;
45     private int[] nilai;
46
47     public HitungRata(Scanner input) {
48         this.input = input;
49     }
50
51     public void inputNilai(int[] nilai) {

```

```

42     class HitungRata {
51         public void inputNilai(int[] nilai) {
52             this.nilai = nilai;
53             for (int i = 0; i < nilai.length; i++) {
54                 System.out.print(s:"Masukkan Nilai : ");
55                 nilai[i] = input.nextInt();
56             }
57         }
58
59         public void cetakNilai(int[] nilai) {
60             for (int i = 0; i < nilai.length; i++) {
61                 System.out.print(nilai[i] + " ");
62             }
63             System.out.println();
64         }
65
66         public double rataNilai(int banyakData) {
67             double total = 0;
68             for (int i = 0; i < nilai.length; i++) {
69                 total += nilai[i];
70             }
71             return total / banyakData;
72         }
73
74         public void inputNilaiBaru(int banyakData) {
75             for (int i = 0; i < nilai.length; i++) {
76                 System.out.print(s:"Masukkan Nilai Baru : ");
77                 nilai[i] = input.nextInt();
78             }

```

```

73
74     public void inputNilaiBaru(int banyakData) {
75         for (int i = 0; i < nilai.length; i++) {
76             System.out.print(s:"Masukkan Nilai Baru : ");
77             nilai[i] = input.nextInt();
78         }
79     }
80
81     public void cetakNilaiBaru() {
82         for (int i = 0; i < nilai.length; i++) {
83             System.out.print(nilai[i] + " ");
84         }
85         System.out.println();
86     }
87 }
88

```

```

Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. All rights reserved.

D:\SEMESTER 4\PEMOGRAMAN BERORIENTASI OBJEK>javac TestNilai.java

D:\SEMESTER 4\PEMOGRAMAN BERORIENTASI OBJEK>java TestNilai
Masukkan Jumlah Data : 5
Pilihan:
1. Input Nilai
2. Input Nilai Baru
Masukkan Pilihan : 1
Masukkan Nilai : 21
Masukkan Nilai : 22
Masukkan Nilai : 23
Masukkan Nilai : 24
Masukkan Nilai : 25
Daftar Nilai :
21 22 23 24 25
Rata Nilai : 23.0

```

4. Apakah perbedaan antara struktur kontrol perulangan while dan do-while?

Jawab:

- Struktur Perulangan While akan mengeksekusi program terlebih dahulu, jika false maka kode tidak akan dijalankan.
- Struktur Perulangan Do-While akan menjalankan kode terlebih dahulu minimal sekali setelah itu dia akan memeriksa kondisi.

5. Kapan digunakan struktur kontrol for?

Jawab:

Kesimpulannya, struktur kontrol for digunakan ketika Anda perlu melakukan iterasi atau perulangan sejumlah tertentu kali dalam program Anda. Ini adalah salah satu

struktur perulangan yang paling umum digunakan dalam pemrograman karena kemampuannya untuk menginisialisasi variabel loop, menentukan kondisi keluar dari loop, dan menentukan langkah iterasi dalam satu baris pernyataan. Dengan struktur ini, Anda dapat secara efisien mengulangi blok kode berulang kali dengan parameter kontrol yang jelas dan terorganisir. Jadi, jika Anda memiliki kebutuhan untuk melakukan iterasi dengan jumlah tertentu, pertimbangkan penggunaan struktur kontrol for dalam kode Anda.

6. Apakah perbedaan antara Array dan ArrayList?berilah contoh masing-masing!

Jawab:

Array dan ArrayList adalah dua struktur data yang umum digunakan dalam bahasa pemrograman Java untuk menyimpan kumpulan data. Berikut adalah beberapa perbedaan utama antara keduanya:

Array: Ukuran array harus ditentukan saat pembuatan dan tidak dapat diubah setelahnya.

Array: Array hanya dapat menyimpan elemen dengan tipe data yang sama.

Array: Array umumnya lebih cepat dalam operasi akses dan manipulasi data dibandingkan ArrayList.

Array: Array adalah struktur data built-in dalam Java.

ArrayList: Ukuran ArrayList bersifat dinamis dan dapat diubah secara otomatis saat data ditambahkan atau dihapus.

ArrayList: ArrayList dapat menyimpan elemen dengan berbagai tipe data (heterogeneous).

ArrayList: ArrayList lebih lambat dalam operasi akses dan manipulasi data dibandingkan array, tetapi lebih efisien dalam memori.

ArrayList: ArrayList adalah bagian dari library Java Collections Framework.

Contoh Array

```
int[] numbers = new int[5];  
numbers[0] = 1;  
numbers[1] = 2;  
numbers[2] = 3;  
numbers[3] = 4;  
numbers[4] = 5;
```

```
for (int i = 0; i < numbers.length; i++) {
    System.out.println(numbers[i]);
}
```

Contoh ArrayList

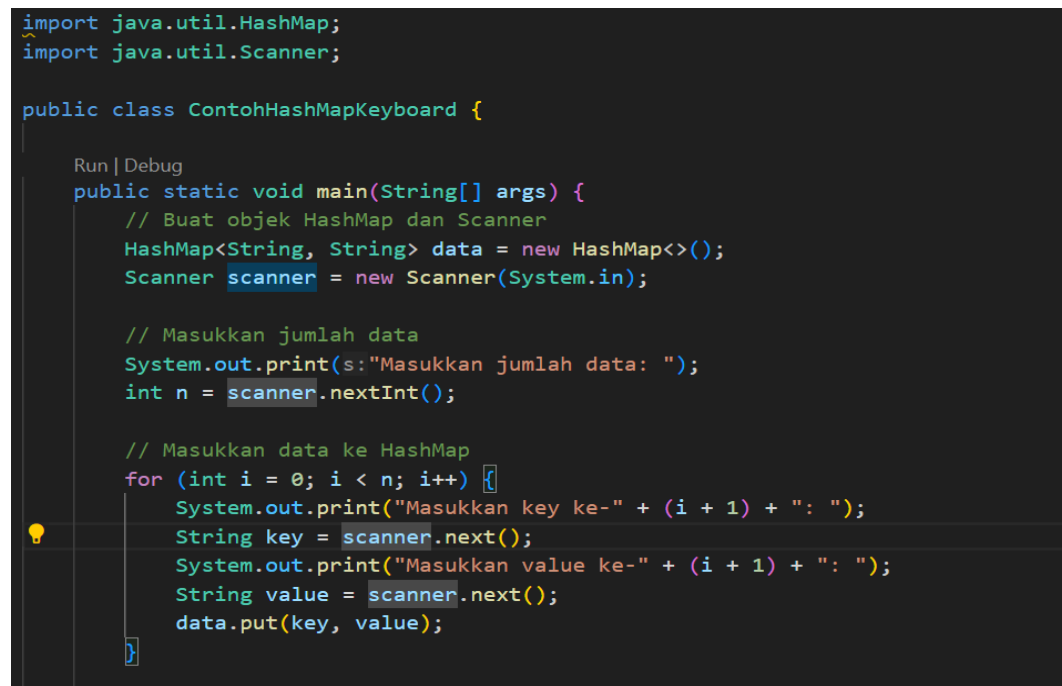
```
ArrayList<String> names = new ArrayList<>();
names.add("John");
names.add("Mary");
names.add("Bob");
```

```
for (String name : names) {
    System.out.println(name);
}
```

7. Buatlah contoh program yang mengimplementasikan HashMap dengan memasukkan nilai dan key melalui keyboard!

Jawab:

=syntax



```
import java.util.HashMap;
import java.util.Scanner;

public class ContohHashMapKeyboard {
    Run | Debug
    public static void main(String[] args) {
        // Buat objek HashMap dan Scanner
        HashMap<String, String> data = new HashMap<>();
        Scanner scanner = new Scanner(System.in);

        // Masukkan jumlah data
        System.out.print(s:"Masukkan jumlah data: ");
        int n = scanner.nextInt();

        // Masukkan data ke HashMap
        for (int i = 0; i < n; i++) {
            System.out.print("Masukkan key ke-" + (i + 1) + ": ");
            String key = scanner.next();
            System.out.print("Masukkan value ke-" + (i + 1) + ": ");
            String value = scanner.next();
            data.put(key, value);
        }
    }
}
```

```

        // Tampilkan data HashMap
        System.out.println(x:"Data HashMap:");
        for (String key : data.keySet()) {
            System.out.println("Key: " + key + ", Value: " + data.get(key));
        }
    }
}

```

=Output

```

D:\SEMESTER 4\PEMOGRAMAN BERORIENTASI OBJEK>javac ContohHashMapKeyboard.java

D:\SEMESTER 4\PEMOGRAMAN BERORIENTASI OBJEK>java ContohHashMapKeyboard
Masukkan jumlah data: 2
Masukkan key ke-1: shabran
Masukkan value ke-1: 21
Masukkan key ke-2: dzul
Masukkan value ke-2: 23
Data HashMap:
Key: shabran, Value: 21
Key: dzul, Value: 23

```