

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

Struktur kontrol percabangan if-else dan switch-case berbeda:

if-else: menjalankan blok kode jika kondisi logis benar. Jika tidak, blok kode lain akan dieksekusi.

Switch-case: Mengevaluasi ekspresi satu dan kemudian membandingkannya dengan berbagai nilai yang mungkin. Nilai yang cocok akan dieksekusi jika cocok.

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

if-else: Ketika terdapat beberapa kondisi yang harus diperiksa secara berurutan atau saling terkait, istilah "if-else" digunakan.

Switch-case: cocok digunakan ketika ada ekspresi tunggal yang harus dibandingkan dengan banyak nilai yang mungkin.

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

opsi pilihah 1=inputNilai()

Pilihan 2=inputNilaiBaru()

```
package pertemuan2.modul3.nilai;  
import java.util.Scanner;
```

```
public class TestNilai {  
    public static void main(String[] args) {  
        HitungRata hitung = new HitungRata();  
        Scanner input = new Scanner(System.in);  
  
        System.out.println("Pilih opsi:");  
        System.out.println("1. Input Nilai");  
        System.out.println("2. Input Nilai Baru");  
  
        int pilihan = input.nextInt();  
  
        switch (pilihan) {  
            case 1:  
                System.out.print("Masukkan Jumlah Data : ");  
                int banyakData = input.nextInt();  
                int nilai[] = new int[banyakData];  
                System.out.print("Masukkan Nilai : ");  
                hitung.inputNilai(nilai);  
                System.out.print("Daftar Nilai : ");  
                hitung.cetakNilai(nilai);  
                System.out.println("Rata Nilai : "+ hitung.rataNilai(banyakData));  
                break;  
            case 2:  
                System.out.print("Masukkan Jumlah Data Baru : ");  
                int banyakDataBaru = input.nextInt();  
                System.out.print("Masukkan Nilai Baru: ");  
                hitung.inputNilaiBaru(banyakDataBaru);  
                System.out.print("Daftar Nilai Baru : ");
```

```

        hitung.cetakNilaiBaru();
        break;
    default:
        System.out.println("Pilihan tidak valid.");
    }
}
}

```

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

While Loop

- Pada loop while, iterasi pertama dilakukan sebelum kondisi evaluasi. Jika kondisi pertama tidak terpenuhi, iterasi tidak akan dilakukan sama sekali.
- Dengan kata lain, blok kode di dalam while loop tidak akan pernah dieksekusi jika kondisi awalnya salah.
- Blok kode di dalam while loop akan dieksekusi jika kondisi benar. Kemudian kondisi akan diperiksa lagi, dan jika hasilnya masih benar, iterasi akan dilanjutkan.

Contoh :

```

int i = 0;
while (i < 5) {
    System.out.println(i);
    i++;
}

```

Do-While Loop

- Kondisi evaluasi dilakukan setelah iterasi pertama selesai pada loop do-while.
- Ini menunjukkan bahwa akan ada setidaknya satu iterasi bahkan jika kondisi awalnya tidak terpenuhi.
- Setelah iterasi pertama dieksekusi, kondisi dinilai. Jika kondisi benar, iterasi akan dilanjutkan, jika tidak, loop akan berhenti.

Contoh:

```

int i = 0;
do {
    System.out.println(i);
    i++;
} while (i < 5);

```

5. Kapan digunakan struktur kontrol for?

While Loop

- Sebelum iterasi pertama, pengecekan kondisi dilakukan.
- Iterasi tidak akan dilakukan jika kondisi tidak terpenuhi.
- Hanya jika kondisi terpenuhi, iterasi akan dilakukan.

Do-While Loop

- Setelah iterasi pertama, pengecekan kondisi dilakukan.
- Bahkan jika kondisi awal tidak terpenuhi, iterasi akan dilakukan setidaknya sekali.
- Iteration akan berhenti jika kondisi tidak terpenuhi setelah iterasi pertama.

Struktur kontrol for digunakan ketika:

- Jumlah iterasi yang akan dilakukan sudah jelas.
- Iterasi pada kumpulan data seperti array atau koleksi lainnya diperlukan.
- Iterasi harus dilakukan dengan teratur.

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

Array:

- Struktur data yang menyimpan elemen dari tipe data yang sama disebut array.
- Ukuran array tidak dapat diubah setelah deklarasi karena sudah ditentukan sebelumnya.
- Untuk menyimpan elemennya, array menggunakan tipe data primitif atau objek.
- Karena ukurannya tetap, array biasanya menggunakan lebih banyak memori.

ArrayList:

- ArrayList adalah versi Java dari interface List.
- Dengan kemampuan untuk diperluas, ArrayList memiliki kemampuan untuk menyimpan elemen dari tipe data yang sama tanpa batasan ukuran.
- Ukuran ArrayList dapat diubah secara dinamis jika diperlukan.
- Tidak seperti ArrayList, yang tidak dapat menyimpan tipe data primitif secara langsung, autoboxing dan unboxing dapat digunakan untuk mengatasi masalah ini

Contoh :

Array

```
int[] arrayAngka = new int[5];  
arrayAngka[0] = 10;  
arrayAngka[1] = 20;
```

ArrayList

```
import java.util.ArrayList;  
  
ArrayList<Integer> daftarAngka = new ArrayList<Integer>();  
daftarAngka.add(10);  
daftarAngka.add(20);
```

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

melalui keyboard!

```
import java.util.HashMap;
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        HashMap<String, Integer> nilaiHashMap = new HashMap<String, Integer>();
        System.out.print("Masukkan jumlah data: ");
        int jumlahData = scanner.nextInt();
        scanner.nextLine();

        for (int i = 0; i < jumlahData; i++) {
            System.out.print("Masukkan key: ");
            String key = scanner.nextLine();
            System.out.print("Masukkan nilai: ");
            int nilai = scanner.nextInt();
            scanner.nextLine();
            nilaiHashMap.put(key, nilai);
        }
        System.out.println("Isi HashMap:");
        for (String k : nilaiHashMap.keySet()) {
            System.out.println("Key: " + k + ", Nilai: " + nilaiHashMap.get(k));
        }
    }
}
```