

Nama : Muh.Fiqri
Kelas : A2
Nim : 13020220071

1. Apakah perbedaan antara struktur kontrol percabangan if-else dan switch-case? Jawab:
Struktur kontrol percabangan if-else digunakan ingin mengevaluasi kondisi tertentu dan menjalankan blok kode yang berbeda-beda sesuai dengan hasil evaluasi tersebut. Sedangkan struktur kontrol percabangan switch-case digunakan ingin mengevaluasi suatu nilai dan menjalankan blok kode yang berbeda-beda sesuai dengan nilai tersebut.

2. Kapan digunakan struktur kontrol if-else dan switch-case Jawaban:
if-else lebih cocok digunakan ketika kita ingin mengevaluasi kondisi yang kompleks dan membutuhkan operasi yang lebih fleksibel. Selain itu, if-else dapat menangani banyak kasus yang berbeda dengan mudah dan dapat digabungkan dengan operator lainnya, seperti else if dan switch-case. Sedangkan switch-case lebih cocok digunakan ketika kita ingin melakukan pemilihan antara beberapa kasus yang sudah ditentukan, dan kondisi yang diperiksa adalah nilai tertentu. switch-case juga lebih mudah dibaca dan dapat meningkatkan kemudahan pemeliharaan kode.

3. Pada program 2, tambahkan perintah untuk memilih 2 opsi menggunakan kontrol switch..case. opsi pilihah 1=inputNilai()
Pilihan 2=inputNilaiBaru() Jawab:

```
package tugas4n2; import java.util.Scanner;

public class Tugas4N2 {
    public static void main(String[] args) { HitungRata hitung = new HitungRata(); Scanner input
    = new Scanner(System.in); System.out.print("Masukkan Jumlah Data : "); int banyakData =
    input.nextInt();
    int nilai[] = new int[banyakData]; System.out.print("Daftar Nilai : "); inputOpsi(input, hitung,
    banyakData, nilai);
    System.out.println("Rata Nilai : "+ hitung.rataNilai(banyakData)); inputOpsi(input, hitung,
    banyakData, nilai); System.out.print("Daftar Nilai Baru : ");
    hitung.cetakNilaiBaru();
    }

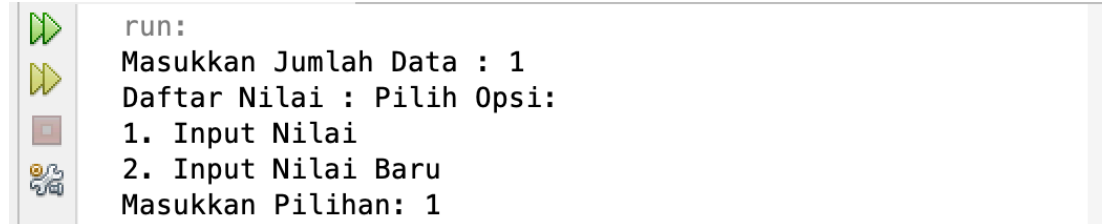
    public static void inputOpsi(Scanner input, HitungRata hitung, int banyakData, int nilai[])
    System.out.println("Pilih Opsi:"); System.out.println("1. Input Nilai"); System.out.println("2.
    Input Nilai Baru"); System.out.print("Masukkan Pilihan: "); int pilihan = input.nextInt();
    switch (pilihan) { case 1:
    hitung.inputNilai(nilai); hitung.cetakNilai(nilai); break;
    case 2:
```

```

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

```

Output:



```

run:
Masukkan Jumlah Data : 1
Daftar Nilai : Pilih Opsi:
1. Input Nilai
2. Input Nilai Baru
Masukkan Pilihan: 1

```

4. Apakah perbedaan antara struktur kontrol perulangan while dan do-while? Jawaban: Perbedaan antara struktur kontrol perulangan while dan do-while adalah ketika mengevaluasi kondisi untuk melakukan iterasi.

while loop mengevaluasi kondisi terlebih dahulu sebelum melakukan iterasi. Jika kondisi benar, blok kode loop akan dijalankan. Jika kondisi salah, blok kode loop akan dilompati.

Sedangkan do-while loop mengejalankan blok kode loop terlebih dahulu dan kemudian mengevaluasi kondisi. Jika kondisi benar, blok kode loop akan dijalankan kembali. Jika kondisi salah, loop akan dihentikan.

5. Kapan digunakan struktur kontrol for? Jawaban

Struktur kontrol for di pemrograman biasanya digunakan ketika ingin melakukan perulangan dengan jumlah iterasi yang ter batas dan diketahui sebelumnya. Contohnya, jika ingin mencetak angka dari 1 hingga 10, Anda dapat menggunakan for loop. Dengan menggunakan for, Anda dapat menentukan jumlah iterasi dan inisialisasi nilainya.

6. Apakah perbedaan antara Array dan ArrayList?berilah

contoh masing-masing! Jawaban:

Perbedaan antara Array dan ArrayList adalah:

- Array memiliki ukuran yang fixed, sedangkan ArrayList memiliki ukuran yang tidak fixed.
- Array dapat mengandung tipe data primer maupun objek, sedangkan ArrayList hanya dapat menampung objek, bukan tipe data primer.
- ArrayList internalkan menggunakan array untuk penyimpanan data, tetapi secara otomatis mengelola resizing saat diperlukan.

Contoh Implemtasi dari Array package array;

```

public class Array {

```

```

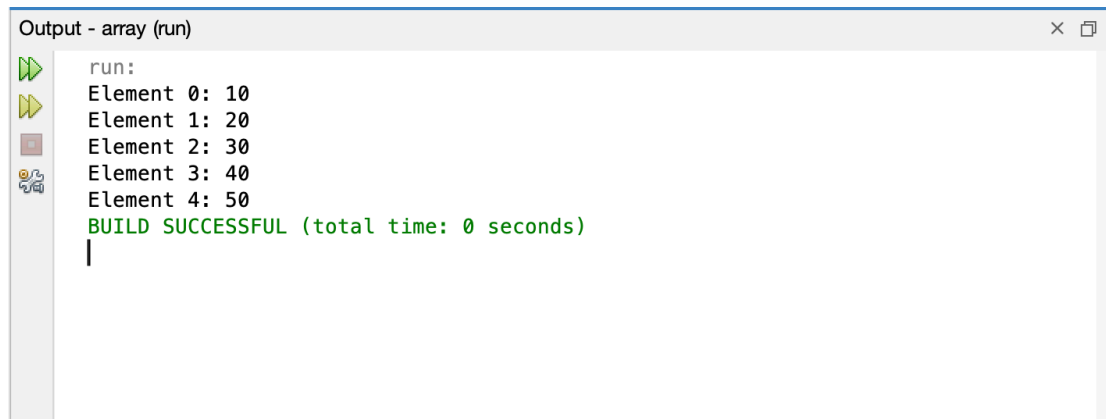
public static void main(String[] args) {
// Deklarasi dan inisialisasi array
int[] numbers = new int[5]; // Array dengan panjang 5

// Mengisi nilai array
numbers[0] = 10;
numbers[1] = 20;
numbers[2] = 30;
numbers[3] = 40;
numbers[4] = 50;

// Mencetak nilai array System.out.println("Isi array:");
for (int i = 0; i < numbers.length; i++) { System.out.println("Element " + i + ": " +
numbers[i]);
}
}
}
}

```

Output:



```

Output - array (run)
run:
Element 0: 10
Element 1: 20
Element 2: 30
Element 3: 40
Element 4: 50
BUILD SUCCESSFUL (total time: 0 seconds)

```

Contoh Implementasi dari ArrayListy package no6;
import java.util.ArrayList;

```

package no6;
import java.util.ArrayList;

public class No6 {
    public static void main(String[] args) {
        // Deklarasi ArrayList
        java.util.ArrayList<Integer> numbers = new java.util.ArrayList<>();

        // Menambahkan elemen ke ArrayList
        numbers.add(10);
    }
}

```

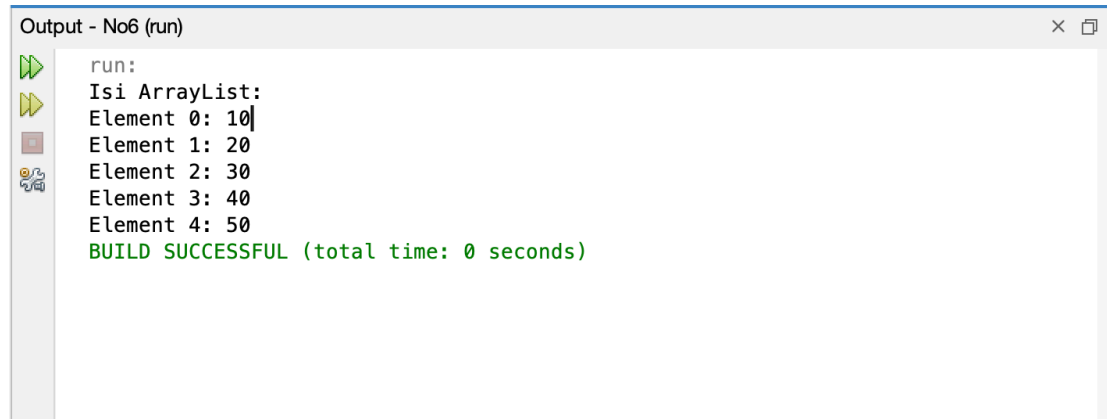
```

        numbers.add(20);
        numbers.add(30);
        numbers.add(40);
        numbers.add(50);

        // Mencetak ArrayList
        System.out.println("Isi ArrayList:");
        for (int i = 0; i < numbers.size(); i++) {
            Sys
        }
    }
}

```

Output:



```

Output - No6 (run)
run:
Isi ArrayList:
Element 0: 10
Element 1: 20
Element 2: 30
Element 3: 40
Element 4: 50
BUILD SUCCESSFUL (total time: 0 seconds)

```

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

Jawab:

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

```

public class tes {
    public static void main(String[] args) {
        tem.out.println("Element " + i + ": " + numbers.get(i));
        HashMap<Integer, String> hashMap = new HashMap<Integer, String>(); Scanner input =

```

```
new Scanner(System.in);
```

```

        System.out.println("Masukkan pasangan nilai kunci yang dipisahkan dengan spasi.
        Masukkan '0' untuk berhenti.");

```

```
while (true) {
```

```
int key = input.nextInt(); if (key == 0) {  
break;  
}
```

```
String nilai = input.next();
```

```
hashMap.put(key, nilai);  
}
```

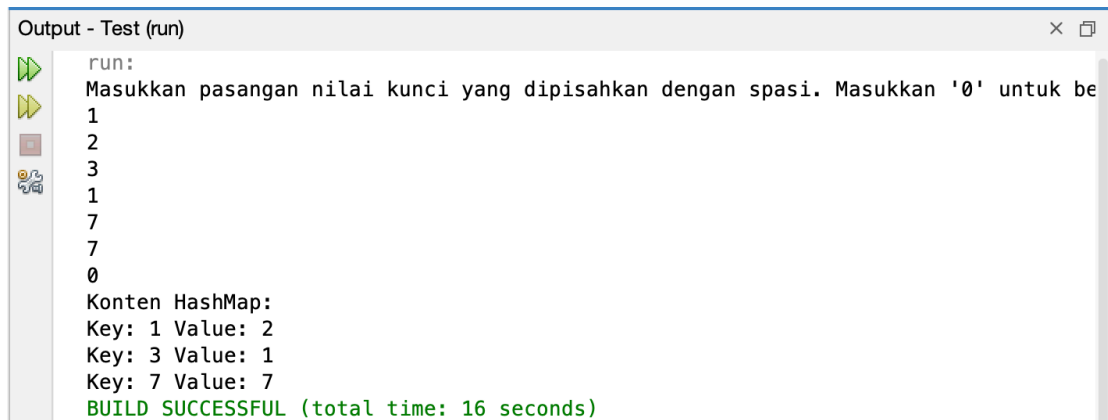
```
System.out.println("Konten HashMap: "); for (Integer theKey : hashMap.keySet()) {  
    System.out.println("Key: " + theKey + " Value: " + hashMap.get(theKey));
```

```
    }
```

```
}
```

```
}
```

Output:



```
run:  
Masukkan pasangan nilai kunci yang dipisahkan dengan spasi. Masukkan '0' untuk be  
1  
2  
3  
1  
7  
7  
0  
Konten HashMap:  
Key: 1 Value: 2  
Key: 3 Value: 1  
Key: 7 Value: 7  
BUILD SUCCESSFUL (total time: 16 seconds)
```

