

PERCOBAAN 1

1. **Jika isi masing-masing elemen array bil diubah dengan angka 5.0, 12867, 7.5, 2000000. Apa yang terjadi? Mengapa bisa demikian?**

Jika isi masing-masing array diubah menjadi angka-angka tersebut, maka akan terjadi error yaitu “Type mismatch: cannot convert from double to int”. Hal ini terjadi karena array bil adalah array tipe int, sedangkan angka 5.0 dan 7.5 yang akan diisikan adalah tipe double.

Array di java memiliki sifat homogen, jadi 1 array hanya dapat berisi 1 tipe data yang sama dengan tipe data array. Jika berisi tipe yang berbeda dengan tipe array maka akan terjadi error.

2. **Modifikasi kode program di atas dengan melakukan inisialisasi elemen array sekaligus pada saat deklarasi array.**
3. **Ubah statement pada langkah No 4 menjadi seperti berikut**

```
for (int i = 0; i < 4; i++){  
    System.out.println(bil[i]);  
}
```

Apa keluaran dari program? Jelaskan maksud dari statement tersebut.

Program mengeluarkan output yang sama dengan sebelumnya. Hal ini terjadi karena syntax tersebut akan menjalankan (println(bil[i])) sebanyak 4 kali sama dengan jumlah elemen array. Ketika i = 0 di iterasi pertama maka program akan menjalankan print(bil[0]). Ketika i = 1 di iterasi kedua maka program akan menjalankan print(bil[1]), begitu seterusnya hingga i = 3. Kemudian ketika i = 4, i sudah tidak memenuhi kondisi loop, jadi loop akan berhenti.

4. **Jika kondisi pada statement for-loop di atas diubah menjadi: i <= 4, apa keluaran dari program? Mengapa demikian?**

Program tersebut tetap mencetak ke 4 elemen array namun kemudian akan terjadi runtime error setelah elemen ke 4. Program akan mengeluarkan pesan error berupa “ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 4”. Hal ini terjadi karena ketika iterasi ke 5 dimana i = 4, i tetap memenuhi kondisi loop (i <= 4), jadi program menjalankan print (bil[4]). Sedangkan tidak ada elemen index ke 4 di dalam array bil.

5. **Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan 1”**

PERCOBAAN 2

1. Ubah statement pada langkah nomor 5 menjadi seperti berikut ini:

```
for (int i = 0; i < nilaiAkhir.length; i++){  
    System.out.print("Masukkan nilai akhir ke-"+i+" : ");  
    nilaiAkhir[i] = sc.nextInt();  
}
```

Jalankan program. Apakah terjadi perubahan? Mengapa demikian?

Ketika program dijalankan, output yang keluar tidak ada perubahan. Hal ini karena fungsi dari “.length” adalah menghitung jumlah elemen di dalam array (panjang array). sehingga nilaiAkhir.length = 10, sama dengan syntax awal yaitu 10.

2. Apa yang dimaksud dengan kondisi: `i < nilaiAkhir.length` ?

Syntax ini adalah kondisi perulangan yaitu perulangan akan terus berjalan selama memenuhi kondisi `i < nilaiAkhir.length`. Syntax nilaiAkhir.length berfungsi untuk menghitung panjang array (jumlah elemennya). jika nilaiAkhir berisi 4 elemen, maka nilaiAkhir.length akan menghasilkan nilai yang sama yaitu 4.

3. Ubah statement pada langkah nomor 6 menjadi seperti berikut ini, sehingga program hanya menampilkan nilai Mahasiswa yang lulus saja (yaitu mahasiswa yang memiliki nilai > 70):

```
for (int i = 0; i < nilaiAkhir.length; i++){  
    if (nilaiAkhir[i] > 70){  
        System.out.println("Mahasiswa ke-"+i+" lulus!");  
    }  
}
```

Jalankan program dan jelaskan alur program!

Alur program berubah dari yang sebelumnya menampilkan semua nilai mahasiswa yang di input, menjadi hanya menampilkan nilai mahasiswa yang lebih dari 70 dengan keterangan lulus. Jadi program akan mengambil input nilai 10 mahasiswa. Kemudian sesuai dengan nilaiAkhir.length, loop akan berjalan sebanyak jumlah elemen di dalam array (panjang array). Kemudian program akan masuk ke kondisi if, jika nilaiAkhir yang telah di input sesuai dengan nomor index tersebut lebih dari 70, maka print(mahasiswa ke {i} lulus) akan di jalankan.

4. Modifikasi program agar menampilkan status kelulusan semua mahasiswa berdasarkan nilai, yaitu dengan menampilkan status mana mahasiswa yang lulus dan tidak lulus, seperti ilustrasi output berikut:

```
Masukkan nilai akhir ke-0 : 87
Masukkan nilai akhir ke-1 : 65
Masukkan nilai akhir ke-2 : 78
Masukkan nilai akhir ke-3 : 95
Masukkan nilai akhir ke-4 : 92
Masukkan nilai akhir ke-5 : 58
Masukkan nilai akhir ke-6 : 89
Masukkan nilai akhir ke-7 : 67
Masukkan nilai akhir ke-8 : 85
Masukkan nilai akhir ke-9 : 78
Mahasiswa ke-0 lulus!
Mahasiswa ke-1 tidak lulus!
Mahasiswa ke-2 lulus!
Mahasiswa ke-3 lulus!
Mahasiswa ke-4 lulus!
Mahasiswa ke-5 tidak lulus!
Mahasiswa ke-6 lulus!
Mahasiswa ke-7 tidak lulus!
Mahasiswa ke-8 lulus!
Mahasiswa ke-9 lulus!
```

5. Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan 2”

PERCOBAAN 3

1. Modifikasi kode program pada praktikum percobaan 3 di atas (ArrayRataNilaiXX.java) agar program dapat menampilkan banyaknya mahasiswa yang lulus, yaitu mahasiswa yang memiliki lebih besar dari 70 (>70).
2. Modifikasi program pada praktikum percobaan 3 di atas (ArrayRataNilaiXX.java) sehingga program menerima jumlah elemen berdasarkan input dari pengguna dan mengeluarkan output seperti berikut ini:

```
Masukkan jumlah mahasiswa : 5
Masukkan nilai mahasiswa ke-1 : 80
Masukkan nilai mahasiswa ke-2 : 60
Masukkan nilai mahasiswa ke-3 : 90
Masukkan nilai mahasiswa ke-4 : 85
Masukkan nilai mahasiswa ke-5 : 65
Rata-rata nilai lulus = 85.0
Rata-rata nilai tidak lulus = 62.5
```

3. Commit dan push hasil modifikasi Anda ke Github dengan pesan “Modifikasi Percobaan 3”

PERCOBAAN 4

1. Jelaskan maksud dari statement `break`; pada baris ke-10 kode program percobaan 4 di atas.

Statement `break` di kode program tersebut berfungsi untuk menghentikan loop jika memenuhi kondisi (`key == arrNilai[i]`). Ini digunakan untuk menghentikan pencarian nilai menggunakan loop saat variabel `key` telah ditemukan pertama kali.

2. Modifikasi kode program pada percobaan 4 di atas sehingga program dapat menerima input berupa banyaknya elemen array nilai, isi array, dan sebuah nilai (`key`) yang ingin dicari. Lalu cetak ke layar indeks posisi elemen dari nilai (`key`) yang dicari. Contoh hasil program:

```
Masukkan banyaknya nilai yang akan diinput: 6
Masukkan nilai mahasiswa ke-1: 80
Masukkan nilai mahasiswa ke-2: 90
Masukkan nilai mahasiswa ke-3: 75
Masukkan nilai mahasiswa ke-4: 83
Masukkan nilai mahasiswa ke-5: 78
Masukkan nilai mahasiswa ke-6: 92
Masukkan nilai yang ingin dicari: 78

Nilai 78 ketemu, merupakan nilai mahasiswa ke-5
```

3. Modifikasi program pada percobaan 4 di atas, sehingga program akan memberikan pesan "Nilai yang dicari tidak ditemukan" jika nilai yang dicari (`key`) tidak ada di dalam array. Contoh tampilan program sebagai berikut:

```
Masukkan banyaknya nilai yang akan diinput: 6
Masukkan nilai mahasiswa ke-1: 80
Masukkan nilai mahasiswa ke-2: 90
Masukkan nilai mahasiswa ke-3: 75
Masukkan nilai mahasiswa ke-4: 82
Masukkan nilai mahasiswa ke-5: 95
Masukkan nilai mahasiswa ke-6: 70
Masukkan nilai yang ingin dicari: 85

Nilai yang dicari tidak ditemukan
```

4. Commit dan push hasil modifikasi Anda ke Github dengan pesan "Modifikasi Percobaan 4"