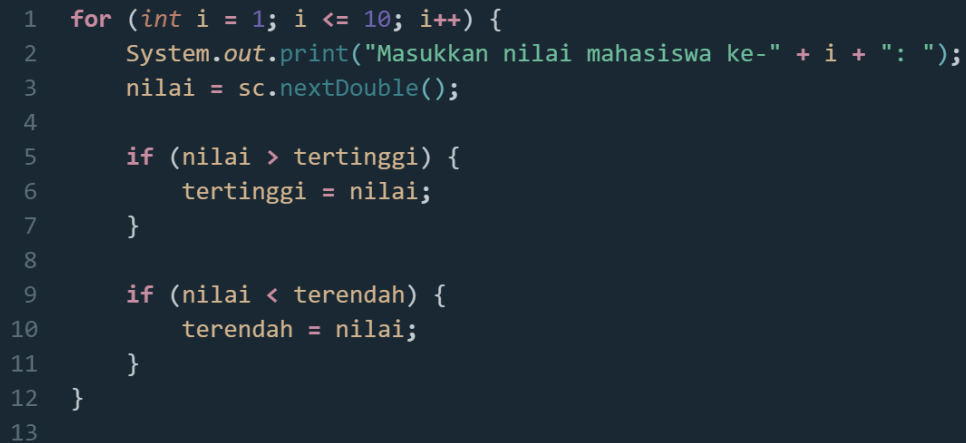


PERCOBAAN 1

1. Sebutkan dan tunjukkan masing-masing komponen perulangan FOR pada kode program Percobaan 1!



```
1  for (int i = 1; i <= 10; i++) {
2      System.out.print("Masukkan nilai mahasiswa ke-" + i + ": ");
3      nilai = sc.nextDouble();
4
5      if (nilai > tertinggi) {
6          tertinggi = nilai;
7      }
8
9      if (nilai < terendah) {
10         terendah = nilai;
11     }
12 }
13
```

Perulangan for memiliki komponen-komponen berupa

1. Inisialisasi

Berada di baris pertama kode pada gambar, di bagian
for (**inisialisasi** ; kondisi ; update)

2. Kondisi

Berada di baris pertama kode pada gambar, di bagian
for (inisialisasi ; **kondisi** ; update)

3. Update

Berada di baris pertama kode pada gambar, di bagian
for (inisialisasi ; kondisi ; **update**)

2. Mengapa variabel tertinggi diinisialisasi 0 dan terendah diinisialisasi 100? Apa yang terjadi jika variabel tertinggi diinisialisasi 100 dan terendah diinisialisasi 0?

Variabel tertinggi diinisialisasi 0 karena ketika program meminta input nilai, jika variabel nilai lebih besar dari variabel tertinggi, maka isi dari variabel nilai akan mengisi variabel tertinggi. Ketika program menerima input nilai yang lebih tinggi dari variabel tertinggi, maka isi dari variabel nilai akan mengisi variabel tertinggi, begitu seterusnya hingga program selesai, jadi variabel tertinggi akan selalu berisi

nilai tertinggi yang diterima oleh program.

Variabel terendah diinisialisasi 100 karena ketika program meminta input nilai, jika variabel nilai lebih kecil dari variabel terendah, maka isi dari variabel nilai akan mengisi variabel terendah. Ketika program menerima input nilai yang lebih kecil dari variabel terendah, maka isi dari variabel nilai akan mengisi variabel terendah, begitu seterusnya hingga program selesai, jadi variabel terendah akan selalu berisi nilai terendah yang diterima oleh program.

Jika variabel tertinggi diinisialisasi 100, maka variabel tertinggi selalu berisi 100, karena input yang diterima program terbatas pada 0-100, jadi variabel nilai tidak akan lebih tinggi dari 100, sehingga tidak akan memenuhi kondisi dimana variabel nilai lebih besar dari variabel tertinggi. Jadi variabel tertinggi selalu berisi 100, walaupun input tertinggi yang diterima program bukan 100. Kecuali jika program menerima input lebih dari 100, maka variabel tertinggi akan berisi variabel nilai yang lebih dari 100, karena program tidak memiliki validasi untuk menolak angka di luar rentang 0-100.

Jika variabel terendah diinisialisasi 0, maka variabel terendah selalu berisi 0, karena input yang diterima program terbatas pada 0-100, jadi variabel nilai tidak akan lebih rendah dari 0, sehingga tidak akan memenuhi kondisi dimana variabel nilai lebih kecil dari variabel terendah. Jadi variabel terendah selalu berisi 0, walaupun input terendah yang diterima program bukan 0. Kecuali jika program menerima input kurang dari 0, maka variabel terendah akan berisi variabel nilai yang kurang dari 0, karena program tidak memiliki validasi untuk menolak angka di luar rentang 0-100.

3. Jelaskan fungsi dan alur kerja dari potongan kode berikut!

```
if (nilai > tertinggi) {  
    tertinggi = nilai;  
}  
if (nilai < terendah) {  
    terendah = nilai;  
}
```

Potongan dari kode ini berfungsi untuk melakukan pengecekan terhadap input nilai tertinggi dan terendah.

apabila input nilai yang diterima program lebih dari variabel tertinggi (0), maka input nilai akan mengisi variabel tertinggi, kemudian berulang terus menerus hingga program berakhir, sehingga variabel tertinggi selalu berisi input nilai

tertinggi yang diterima program.

apabila input nilai yang diterima program kurang dari variabel terendah (100), maka input nilai akan mengisi variabel terendah, kemudian berulang terus menerus hingga program berakhir, sehingga variabel terendah selalu berisi input nilai tertinggi yang diterima program.

4. **Modifikasi kode program sehingga terdapat perhitungan untuk menentukan berapa mahasiswa yang lulus dan yang tidak lulus berdasarkan batas kelulusan (nilai minimal 60). Tampilkan jumlah mahasiswa lulus dan tidak lulus setelah menampilkan nilai tertinggi dan terendah!**

```
1 public static void main(String[] args) {
2     Scanner sc = new Scanner(System.in);
3     double nilai, tertinggi = 0, terendah = 100;
4     String status;
5     int jumlahLulus = 0, jumlahTidakLulus = 0;
6
7
8     for (int i = 1; i <= 10; i++) {
9         System.out.print("Masukkan nilai mahasiswa ke-" + i + ": ");
10        nilai = sc.nextDouble();
11
12        if (nilai > tertinggi) {
13            tertinggi = nilai;
14        }
15
16        if (nilai < terendah) {
17            terendah = nilai;
18        }
19
20        if (nilai >= 60) {
21            status = "Lulus";
22            jumlahLulus++;
23        } else {
24            status = "Tidak Lulus";
25            jumlahTidakLulus++;
26        }
27
28        System.out.println("Status mahasiswa ke-" + i + ": " + status + "\n");
29    }
30
31    System.out.println("Nilai tertinggi\t\t\t: " + tertinggi);
32    System.out.println("Nilai terendah\t\t\t: " + terendah);
33    System.out.println("Jumlah mahasiswa lulus\t\t: " + jumlahLulus);
34    System.out.println("Jumlah mahasiswa tidak lulus\t: " + jumlahTidakLulus);
35    sc.close();
36 }
```

```
Masukkan nilai mahasiswa ke-1: 90
Status mahasiswa ke-1: Lulus

Masukkan nilai mahasiswa ke-2: 80
Status mahasiswa ke-2: Lulus

Masukkan nilai mahasiswa ke-3: 60
Status mahasiswa ke-3: Lulus

Masukkan nilai mahasiswa ke-4: 59
Status mahasiswa ke-4: Tidak Lulus

Masukkan nilai mahasiswa ke-5: 12
Status mahasiswa ke-5: Tidak Lulus

Masukkan nilai mahasiswa ke-6: 10
Status mahasiswa ke-6: Tidak Lulus

Masukkan nilai mahasiswa ke-7: 78
Status mahasiswa ke-7: Lulus

Masukkan nilai mahasiswa ke-8: 3
Status mahasiswa ke-8: Tidak Lulus

Masukkan nilai mahasiswa ke-9: 12
Status mahasiswa ke-9: Tidak Lulus

Masukkan nilai mahasiswa ke-10: 45
Status mahasiswa ke-10: Tidak Lulus
```

```
Nilai tertinggi           : 90.0
Nilai terendah            : 3.0
Jumlah mahasiswa lulus    : 4
Jumlah mahasiswa tidak lulus : 6
PS C:\Users\sapin\Repository\PraktikumDaspro>
```

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

PERCOBAAN 2

1. Pada potongan kode berikut, tentukan maksud dan kegunaan dari sintaks berikut:

```

if (nilai < 0 || nilai > 100) {
    System.out.println(x:"Nilai tidak valid. Masukkan lagi nilai yang valid!");
    continue;
}

```

Maksud dan kegunaan dari program ini adalah untuk melakukan validasi input nilai sesuai dengan kondisi yang diinginkan.

a. nilai < 0 || nilai > 100

Ini adalah kondisi yang ada di dalam If, maksudnya adalah apabila nilai kurang dari 0 atau lebih dari 100, maka memenuhi kondisi if dan akan menjalankan statement yang ada di dalam if.

b. continue

Continue disini digunakan agar saat program menerima input yang memenuhi kondisi if, maka program akan langsung melakukan iterasi berikutnya dan mengabaikan program di bawah continue pada iterasi tersebut.

2. Mengapa sintaks i++ dituliskan di akhir perulangan WHILE? Apa yang terjadi jika posisinya dituliskan di awal perulangan WHILE?

Sintaks i++ dituliskan di akhir perulangan while, karena ketika nilai memenuhi kondisi (kurang dari 0 atau lebih dari 100), maka akan melakukan continue. Jadi nilai i pada iterasi berikutnya tetap akan sama dengan nilai i pada iterasi saat ini.

Jika sintaks i++ dituliskan di awal perulangan while, karena ketika nilai memenuhi kondisi (kurang dari 0 atau lebih dari 100), maka akan melakukan continue. Jadi nilai i pada iterasi berikutnya akan berbeda (+1) dengan nilai i pada iterasi saat ini.

3. Apabila jumlah mahasiswa yang dimasukkan adalah 19, berapa kali perulangan WHILE akan berjalan?

Perulangan while berjalan 19 kali, karena while akan terus melakukan loop ketika i kurang dari jumlah mahasiswa. i awal adalah 0, di akhir loop while i ditambah 1, sehingga ketika i = 19, kondisi sudah tidak memenuhi kondisi while, jadi loop akan berhenti.

4. Modifikasi kode program sehingga apabila terdapat mahasiswa yang mendapat nilai A, program menampilkan pesan tambahan "Bagus, pertahankan nilainya"!



```
1 public static void main(String[] args) {
2     Scanner sc = new Scanner(System.in);
3     int nilai, jml, i = 0;
4
5     System.out.print("Masukkan jumlah mahasiswa: ");
6     jml = sc.nextInt();
7
8     while (i < jml) {
9         System.out.print("Masukkan nilai mahasiswa ke-" + (i + 1) + ": ");
10        nilai = sc.nextInt();
11
12        if (nilai < 0 || nilai > 100) {
13            System.out.println("Nilai tidak valid. Masukkan lagi nilai yang valid!");
14            continue;
15        }
16
17        if (80 < nilai && nilai <= 100) {
18            System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah A");
19            System.out.println("Bagus, pertahankan nilainya!");
20
21        } else if (73 < nilai && nilai <= 80) {
22            System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah B+");
23
24        } else if (65 < nilai && nilai <= 73) {
25            System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah B");
26
27        } else if (60 < nilai && nilai <= 65) {
28            System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah C+");
29
30        } else if (50 < nilai && nilai <= 60) {
31            System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah C");
32
33        } else if (39 < nilai && nilai <= 50) {
34            System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah D");
35
36        } else {
37            System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah E");
38        }
39        i++;
40    }
41    sc.close();
42 }
```

```
Masukkan jumlah mahasiswa: 5
Masukkan nilai mahasiswa ke-1: 90
Nilai mahasiswa ke-1 adalah A
Bagus, pertahankan nilainya!
Masukkan nilai mahasiswa ke-2: 89
Nilai mahasiswa ke-2 adalah A
Bagus, pertahankan nilainya!
Masukkan nilai mahasiswa ke-3: 12
Nilai mahasiswa ke-3 adalah E
Masukkan nilai mahasiswa ke-4: 78
Nilai mahasiswa ke-4 adalah B+
Masukkan nilai mahasiswa ke-5: 45
Nilai mahasiswa ke-5 adalah D
```

```
PS C:\Users\sapin\Repository\PraktikumDaspro>
```

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

PERCOBAAN 3

- 1. Pada penggunaan DO-WHILE ini, apabila nama pelanggan yang dimasukkan pertama kali adalah “batal”, maka berapa kali perulangan dilakukan?**

Apabila nama pelanggan pertama diisi “batal”, maka program memenuhi kondisi keluar dari loop, sehingga program menyatakan “Transaksi dibatalkan.” kemudian langsung keluar dari perulangan do-while. jadi program hanya menjalankan program pada bagian do sekali, kemudian langsung keluar dari loop. jadi loop dilakukan sekali di bagian do.

- 2. Sebutkan kondisi berhenti yang digunakan pada perulangan DO-WHILE tersebut!**

Perulangan tersebut akan berhenti apabila program menerima input nama pelanggan “batal”.

- 3. Apa fungsi dari penggunaan nilai true pada kondisi DO-WHILE?**

Fungsi penggunaan nilai true pada kondisi do-while adalah agar perulangan tersebut menjadi indefinite, sehingga jumlah perulangan tidak ditentukan dan hanya akan keluar ketika memenuhi kondisi keluar (nama pelanggan “batal”).

- 4. Mengapa perulangan DO-WHILE tersebut tetap berjalan meskipun tidak ada komponen inisialisasi dan update?**

Karena sesungguhnya komponen inisialisasi dan update adalah komponen yang bersifat opsional (boleh ada atau tidak). Fungsi dari komponen inisialisasi dan update sendiri adalah untuk mengatur variable counter (variabel pengontrol perulangan). Sedangkan perulangan do-while tersebut adalah perulangan indefinite yang akan keluar apabila memenuhi suatu kondisi (nama pelanggan “batal”). Sehingga komponen inisialisasi dan update sudah tidak diperlukan karena fungsinya digantikan oleh adanya sentinel (pembatas loop) berupa kondisi (nama pelanggan “batal”).

TUGAS

```
1 package jobsheet7;
2 import java.util.*;
3
4 public class Tugas1 {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         int pelanggan = 1;
9         int harga = 50000;
10        double totalHarga = 0;
11        int totalTiket = 0;
12        double jumlahHarga = 0;
13        int jumlahTiket = 0;
14        String lanjut;
15
16        // header
17        System.out.println("===== TOTAL PENJUALAN TIKET DALAM SATU HARI =====");
18
19        // perulangan pelanggan
20        while (true) {
21            // meminta input jumlah tiket
22            System.out.printf("\nMasukkan jumlah tiket pelanggan ke-%d: ", pelanggan);
23            jumlahTiket = sc.nextInt();
24
25            // membersihkan buffer
26            sc.nextLine();
27
28            // validasi nilai negatif
29            if (jumlahTiket < 0) {
30                System.out.println("Jumlah tiket tidak valid.");
31                continue;
32            }
33
34            // menentukan diskon atau tidak
35            if (jumlahTiket > 4 && jumlahTiket <= 10) {
36                jumlahHarga = (jumlahTiket * harga) * 0.9; // Diskon 10%
37            } else if (jumlahTiket > 10) {
38                jumlahHarga = (jumlahTiket * harga) * 0.85; // Diskon 15%
39            } else {
40                jumlahHarga = jumlahTiket * harga; // Tanpa diskon
41            }
42
43            // menambahkan ke total
44            totalHarga += jumlahHarga;
45            totalTiket += jumlahTiket;
46
47            // menampilkan jumlah dan harga tiket yang di beli
48            System.out.println("Total tiket yang dibeli saat ini\t\t: " + jumlahTiket);
49            System.out.println("Total harga \t\t\t\t\t: Rp " + jumlahHarga);
50
51            // menanyakan lanjut kagak?
52            System.out.print("\nApakah ada pembelian lagi? (y/n): ");
53            lanjut = sc.nextLine();
54
55            // jika n maka keluar loop
56            if (lanjut.equalsIgnoreCase("n")) {
57                break;
58            }
59
60            // Lanjut ke pelanggan berikutnya
61            pelanggan +=1;
62        }
63
64        // menampilkan total seluruh pembelian dalam 1 hari
65        System.out.println("\nTotal harga tiket yang dibeli dalam 1 hari\t\t: Rp " + totalHarga);
66        System.out.println("Total tiket yang dibeli dalam 1 hari\t\t\t: " + totalTiket);
67
68        // footer
69        System.out.println("=====");
70
71        // menutup scanner
72        sc.close();
73    }
74 }
75
76
77
```

1.

```

===== TOTAL PENJUALAN TIKET DALAM SATU HARI =====

Masukkan jumlah tiket pelanggan ke-1: 4
Total tiket yang dibeli saat ini      : 4
Total harga                          : Rp 200000.0

Apakah ada pembelian lagi? (y/n): y

Masukkan jumlah tiket pelanggan ke-2: 5
Total tiket yang dibeli saat ini      : 5
Total harga                          : Rp 225000.0

Apakah ada pembelian lagi? (y/n): y

Masukkan jumlah tiket pelanggan ke-3: 10
Total tiket yang dibeli saat ini      : 10
Total harga                          : Rp 450000.0

Apakah ada pembelian lagi? (y/n): y

Masukkan jumlah tiket pelanggan ke-4: 11
Total tiket yang dibeli saat ini      : 11
Total harga                          : Rp 467500.0

Apakah ada pembelian lagi? (y/n): y

Masukkan jumlah tiket pelanggan ke-5: -9
Jumlah tiket tidak valid.

Masukkan jumlah tiket pelanggan ke-5: 1
Total tiket yang dibeli saat ini      : 1
Total harga                          : Rp 50000.0

Apakah ada pembelian lagi? (y/n): n

Total harga tiket yang dibeli dalam 1 hari : Rp 1392500.0
Total tiket yang dibeli dalam 1 hari      : 31
=====
PS C:\Users\sapin\Repository\PraktikumDaspro>

```

```

1  package jobsheet7;
2  import java.util.*;
3
4  public class Tugas2 {
5      public static void main(String[] args) {
6          Scanner sc = new Scanner(System.in);
7
8          int jenis, durasi, total = 0;
9          int tarif = 0;
10         int kendaraan = 1;
11
12         // header
13         System.out.println("===== MENGHITUNG TOTAL PEMBAYARAN PARKIR =====");
14
15         // perulangan pelanggan
16         while (true) {
17             // meminta input jenis kendaraan
18             System.out.print(
19                 "\nMasukkan jenis kendaraan ke-" + kendaraan +
20                 "\n1. Mobil" +
21                 "\n2. Motor" +
22                 "\n(pilih 0 untuk keluar)" +
23                 "\nSilahkan pilih jenis kendaraan (1-2)\t: ");
24             jenis = sc.nextInt();
25
26             // membersihkan buffer
27             sc.nextLine();
28
29             // validasi nilai negatif
30             if (jenis < 0 || jenis > 2) {
31                 System.out.println("pilihan tidak valid.");
32                 continue;
33             }
34
35             // menentukan diskon atau tidak
36             if (jenis == 1 || jenis == 2) {
37                 while (true) {
38                     System.out.print("\nMasukkan durasi parkir (dalam jam)\t: ");
39                     durasi = sc.nextInt();
40
41                     // validasi durasi
42                     if (durasi <= 0) {
43                         System.out.println("Durasi tidak valid.");
44                         continue;
45                     }
46
47                     // menghitung tarif
48                     if (durasi > 5) {
49                         tarif = 12500;
50                     } else if (jenis == 1) {
51                         tarif = durasi * 3000;
52                     } else if (jenis == 2) {
53                         tarif = durasi * 2000;
54                     }
55                     total += tarif;
56                     break;
57                 }
58             } else if (jenis == 0) {
59                 break;
60             }
61
62             // menampilkan total yang harus dibayar
63             System.out.println("\nTarif kendaraan ke-" + kendaraan + "\t\t: Rp " + tarif);
64
65             // footer
66             System.out.println("-----");
67
68             // Lanjut ke kendaraan berikutnya
69             kendaraan += 1;
70         }
71
72         // menampilkan total seluruh pembelian dalam 1 hari
73         System.out.println("\nTotal pembayaran parkir dalam 1 hari\t: Rp " + total);
74         System.out.println("Total kendaraan yang diparkir\t: " + (kendaraan - 1));
75
76         // menutup scanner
77         sc.close();
78     }
79 }

```

===== MENGHITUNG TOTAL PEMBAYARAN PARKIR =====

Masukkan jenis kendaraan ke-1

1. Mobil

2. Motor

(pilih 0 untuk keluar)

Silahkan pilih jenis kendaraan (1-2) : 1

Masukkan durasi parkir (dalam jam) : 5

Tarif kendaraan ke-1 : Rp 15000

Masukkan jenis kendaraan ke-2

1. Mobil

2. Motor

(pilih 0 untuk keluar)

Silahkan pilih jenis kendaraan (1-2) : 1

Masukkan durasi parkir (dalam jam) : 6

Tarif kendaraan ke-2 : Rp 12500

Masukkan jenis kendaraan ke-3

1. Mobil

2. Motor

(pilih 0 untuk keluar)

Silahkan pilih jenis kendaraan (1-2) : 2

Masukkan durasi parkir (dalam jam) : 2

Tarif kendaraan ke-3 : Rp 4000

Masukkan jenis kendaraan ke-4

1. Mobil

2. Motor

(pilih 0 untuk keluar)

Silahkan pilih jenis kendaraan (1-2) : 0

Total pembayaran parkir dalam 1 hari : Rp 31500

Total kendaraan yang diparkir : 3

PS C:\Users\sapin\Repositori\PraktikumDaspro> █