

**NAMA : NEVITA TRIYA YULIANA**

**KELAS : TI 1D**

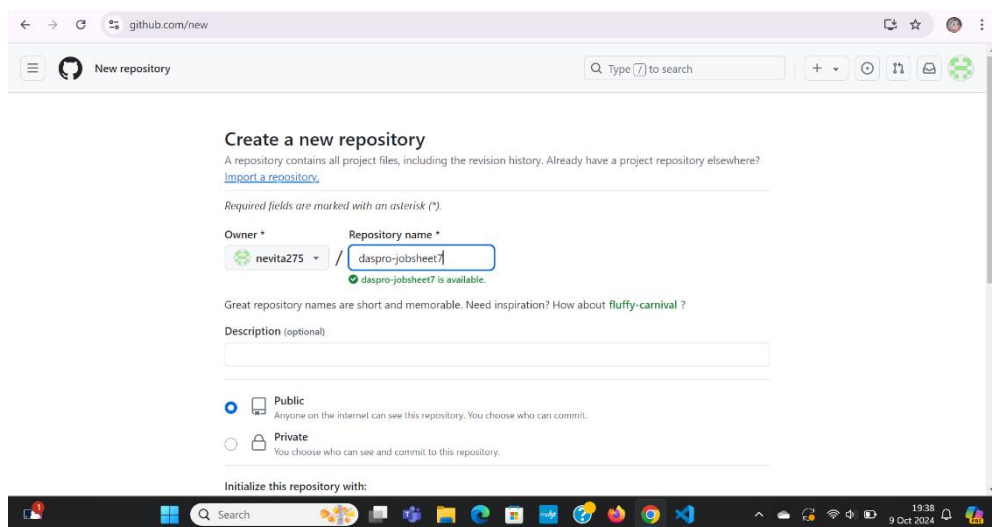
**ABSEN : 21**

**NIM : 244107020208**

## **JOBSHEET 7**

### **PERCOBAAN 1:**

1. Buat repository baru pada akun Github Anda, beri nama daspro-jobsheet7



2. Lakukan cloning repository tersebut menggunakan perintah git clone dari terminal

```
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP> git clone https://github.com/nevita275/daspro-jobsheet7.git
Cloning into 'daspro-jobsheet7'...
warning: You appear to have cloned an empty repository.

C:\Users\HP>
```

3. Buka folder repository tersebut menggunakan Visual Studio Code
4. Buat file baru, beri nama SiakadForNoAbsen.java
5. Buatlah struktur dasar program Java yang terdiri dari fungsi main().
6. Tambahkan library Scanner di bagian atas (luar) class
7. Buat deklarasi Scanner dengan nama variabel sc di dalam fungsi main()
8. Deklarasikan variabel nilai, tertinggi, dan terendah bertipe double. Inisialisasi tertinggi dengan 0 dan terendah dengan 100
9. Buat struktur perulangan FOR dengan batas kondisi sesuai jumlah mahasiswa yaitu 10

10. Di dalam perulangan FOR tersebut, tambahkan perintah untuk memasukkan nilai mahasiswa. Setelah itu, buat dua kondisi pemilihan secara terpisah untuk mengecek nilai tertinggi dan terendah dengan membandingkan nilai masukan dengan variabel tertinggi dan variabel terendah

```
8
9
10
11     for (int i = 1; i <= 10; i++) {
12         System.out.println("Masukkan nilai mahasiswa ke-" + i + ": ");
13         nilai = sc.nextDouble();
14         if (nilai > tertinggi) {
15             tertinggi = nilai;
16         }
17         if (nilai < terendah) {
18             terendah = nilai;
19         }
20     }
```

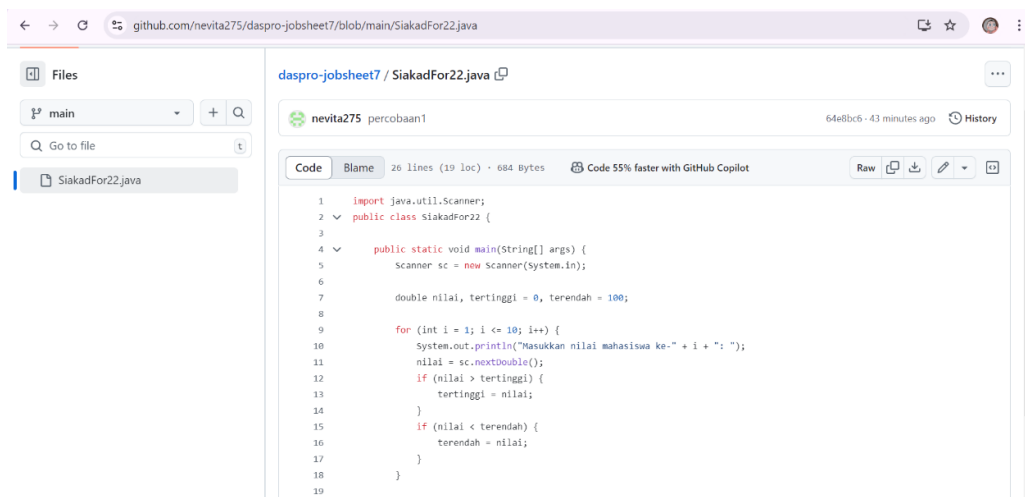
11. Di luar perulangan FOR, tampilkan nilai tertinggi dan terendah

```
20
21     System.out.println("Nilai tertinggi: " + tertinggi);
22     System.out.println("Nilai terendah: " + terendah);
23 }
```

12. Compile dan run program

```
Masukkan nilai mahasiswa ke-1:
76,5
Masukkan nilai mahasiswa ke-2:
82,3
Masukkan nilai mahasiswa ke-3:
62,1
Masukkan nilai mahasiswa ke-4:
88,4
Masukkan nilai mahasiswa ke-5:
65,9
Masukkan nilai mahasiswa ke-6:
67,9
Masukkan nilai mahasiswa ke-7:
90,1
Masukkan nilai mahasiswa ke-8:
55,3
Masukkan nilai mahasiswa ke-9:
73,7
Masukkan nilai mahasiswa ke-10:
78,6
Nilai tertinggi: 90.1
Nilai terendah: 55.3
```

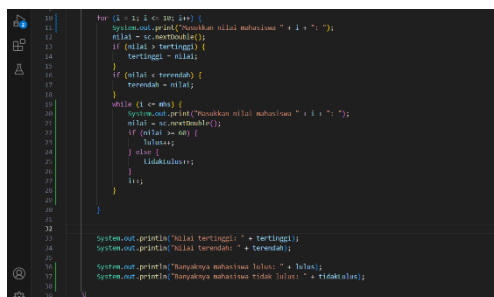
13. Commit dan push kode program ke Github



## PERTANYAAN :

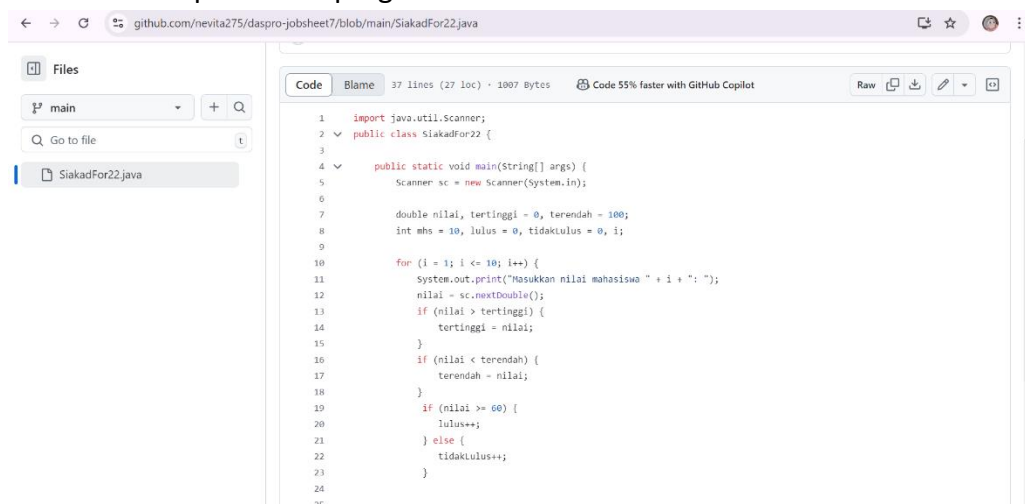
1. Sebutkan dan tunjukkan masing-masing komponen perulangan FOR pada kode program Percobaan 1!  
⇒ (int i = 10; i <= 10; i++)  
⇒ (int i=10;) inisialisasi, (i<=10;) kondisi, (i++) pembaruan/ update

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 mencari nilai maksimum dari data yang nilainya selalu lebih besar dari 0 sedangkan variabel terendah diinisialisasi 100 karena mencari nilai minimum dari data yang nilainya selalu kurang dari 100.
  - ⇒ Jika variabel tertinggi diinisialisasi 100 maka hasil akhir untuk nilai tertinggi tidak akan terpengaruh oleh inisialisasi ini. Sebaliknya, jika variabel terendah diinisialisasi 0 maka inisialisasi ini akan menyebabkan hasil akhir untuk nilai terendah menjadi tidak valid, karena tidak akan ada nilai yang lebih kecil dari 0.
3. Jelaskan fungsi dan alur kerja dari potongan kode berikut!
  - ⇒ `if (nilai > tertinggi) { tertinggi = nilai; }` berfungsi untuk mencari nilai tertinggi, Jika nilai saat ini (nilai) lebih besar daripada nilai tertinggi yang sudah ada (tertinggi), maka nilai tertinggi akan diperbarui menjadi nilai saat ini. Ini memastikan bahwa setelah semua nilai diperiksa, variabel tertinggi akan menyimpan nilai maksimum dari kumpulan data.
  - ⇒ `if (nilai < terendah) { terendah = nilai; }` berfungsi untuk mencari nilai terendah, Jika nilai saat ini (nilai) lebih kecil daripada nilai terendah yang sudah ada (terendah), maka nilai terendah akan diperbarui menjadi nilai saat ini. Ini memastikan bahwa setelah semua nilai diperiksa, variabel terendah akan menyimpan nilai minimum dari kumpulan data.
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!



```
10 for (i = 1; i <= mhs; i++) {
11     System.out.print("Masukkan nilai mahasiswa " + i + ": ");
12     nilai = sc.nextDouble();
13     if (nilai > tertinggi) {
14         tertinggi = nilai;
15     }
16     if (nilai < terendah) {
17         terendah = nilai;
18     }
19     while (i <= mhs) {
20         System.out.print("Masukkan nilai mahasiswa " + i + ": ");
21         nilai = sc.nextDouble();
22         if (nilai >= 60) {
23             lulus++;
24         } else {
25             tidaklulus++;
26         }
27         i++;
28     }
29 }
30
31 System.out.println("Nilai tertinggi: " + tertinggi);
32 System.out.println("Nilai terendah: " + terendah);
33
34 System.out.println("Banyaknya mahasiswa lulus: " + lulus);
35 System.out.println("Banyaknya mahasiswa tidak lulus: " + tidaklulus);
36 }
```

5. Commit dan push kode program ke Github



## PERCOBAAN 2:

1. Buat file baru, beri nama SiakadWhileNoAbsen.java
2. Buatlah struktur dasar program Java yang terdiri dari fungsi main().
3. Tambahkan library Scanner di bagian atas (luar) class
4. Buat deklarasi Scanner dengan nama variabel sc di dalam fungsi main()
5. Deklarasikan variabel nilai, jml, dan i (untuk perulangan) bertipe integer. Inisialisasi i dengan 0 sebagai nilai awal perulangan
6. Tuliskan kode program untuk menerima input banyaknya mahasiswa yang disimpan ke variabel jml. Dengan demikian, batas perulangan akan dinamis sesuai masukan dari pengguna melalui keyboard.

```
8
9      System.out.print(s:"Masukkan jumlah mahasiswa: ");
10     jml = sc.nextInt();
```

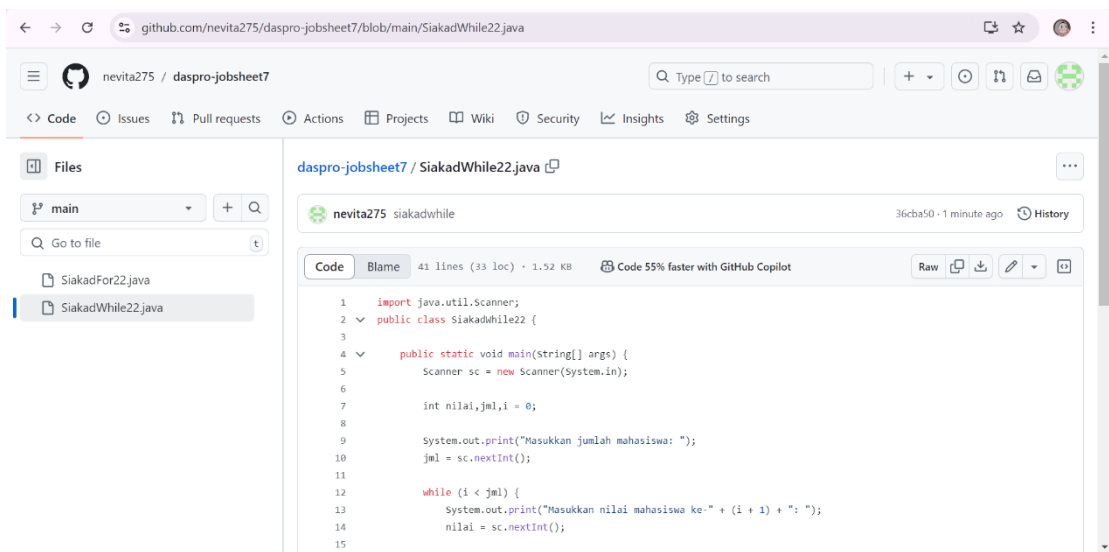
7. Buat struktur perulangan WHILE dengan batas kondisi sesuai jumlah mahasiswa yaitu 5. Perhatikan simbol yang digunakan adalah < karena perulangan variabel i dimulai dari 0, bukan 1
8. Di dalam perulangan WHILE tersebut, tambahkan perintah untuk memasukkan nilai mahasiswa. Setelah itu, buat kondisi pemilihan IF untuk mengecek valid atau tidaknya nilai yang dimasukkan, dengan syarat nilai harus berada pada rentang 0 hingga 100. Kemudian tambahkan kondisi pemilihan IF-ELSE IF-ELSE untuk menampilkan kategori nilai huruf berdasarkan ketentuan.

```
12     while (i < jml) {
13         System.out.print("Masukkan nilai mahasiswa ke-" + (i + 1) + ": ");
14         nilai = sc.nextInt();
15
16         if (nilai < 0 || nilai > 100) {
17             System.out.println(x:"Nilai tidak valid. Masukkan lagi nilai yang valid!");
18             continue;
19         }
20
21         if (nilai > 80 && nilai <= 100) {
22             System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah A");
23             System.out.println(x:"Bagus, pertahankan nilainya!");
24         } else if (nilai > 73 && nilai <= 80) {
25             System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah B+");
26         } else if (nilai > 65 && nilai <= 73) {
27             System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah B");
28         } else if (nilai > 60 && nilai <= 65) {
29             System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah C+");
30         } else if (nilai > 50 && nilai <= 60) {
31             System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah C");
32         } else if (nilai > 39 && nilai <= 50) {
33             System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah D");
34         } else {
35             System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah E");
36         }
37         i++;
38     }
```

9. Compile dan run program

```
Masukkan jumlah mahasiswa: 5
Masukkan nilai mahasiswa ke-1: 85
Nilai mahasiswa ke-1 adalah A
Masukkan nilai mahasiswa ke-2: 63
Nilai mahasiswa ke-2 adalah C+
Masukkan nilai mahasiswa ke-3: 101
Nilai tidak valid. Masukkan lagi nilai yang valid!
Masukkan nilai mahasiswa ke-3: 23
Masukkan nilai mahasiswa ke-3: 23
Masukkan nilai mahasiswa ke-3: 23
Nilai mahasiswa ke-3 adalah E
Masukkan nilai mahasiswa ke-4: -15
Nilai tidak valid. Masukkan lagi nilai yang valid!
Masukkan nilai mahasiswa ke-4: 70
Nilai mahasiswa ke-4 adalah B
Masukkan nilai mahasiswa ke-5: 55
Nilai mahasiswa ke-5 adalah C
PS C:\Users\HP\daspro-jobsheet>
```

## 10. Commit dan push kode program ke Github

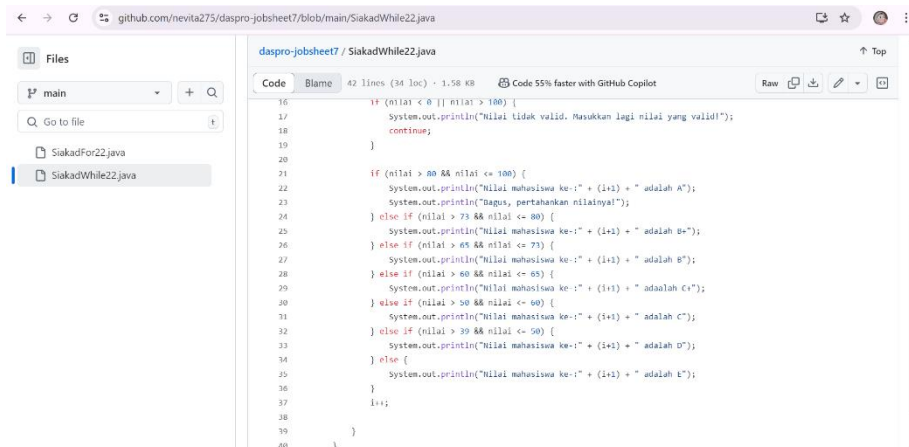


### PERTANYAAN :

- Pada potongan kode berikut, tentukan maksud dan kegunaan dari sintaks berikut:
  - `nilai < 0 || nilai > 100`  
untuk memeriksa validitas dari nilai yang dimasukkan  
(nilai < 0) adalah apakah nilai yang dimasukkan kurang dari 0. Jika ya, maka nilai tersebut tidak valid, karena nilai mahasiswa tidak bisa negatif.  
(nilai > 100) adalah apakah nilai yang dimasukkan lebih dari 100. Jika ya, maka nilai tersebut juga tidak valid, karena skala nilai biasanya berkisar antara 0 hingga 100.
  - `continue` untuk melewati beberapa kondisi tertentu dalam loop tanpa keluar dari loop secara keseluruhan.
- Mengapa sintaks `i++` dituliskan di akhir perulangan WHILE? Apa yang terjadi jika posisinya dituliskan di awal perulangan WHILE?
  - ⇒ Jika posisinya di akhir nilai `i` (0) akan dicetak, dan loop akan berjalan dengan benar hingga nilai `i` menjadi 5.
  - ⇒ Jika posisinya di awal nilai awal `i` (0) tidak akan dicetak, karena `i` akan ditambah satu terlebih dahulu, sehingga loop akan mulai dari 1 dan hanya mencetak nilai dari 1 hingga 5.
- Apabila jumlah mahasiswa yang dimasukkan adalah 19, berapa kali perulangan WHILE akan berjalan?
  - ⇒ Akan berjalan tepat 19 kali sebelum berhenti
- Modifikasi kode program sehingga apabila terdapat mahasiswa yang mendapat nilai A, program menampilkan pesan tambahan "Bagus, pertahankan nilainya!"

```
21 if (nilai > 80 && nilai <= 100) {
22     System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah A");
23     System.out.println(x:"Bagus, pertahankan nilainya!");
24 } else if (nilai > 73 && nilai <= 80) {
25     System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah B+");
26 } else if (nilai > 65 && nilai <= 73) {
27     System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah B");
28 } else if (nilai > 60 && nilai <= 65) {
29     System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah C+");
30 } else if (nilai > 50 && nilai <= 60) {
31     System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah C");
32 } else if (nilai > 39 && nilai <= 50) {
33     System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah D");
34 } else {
35     System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah E");
36 }
37 i++;
38
```

## 5. Commit dan push kode program ke Github



### PERCOBAAN 3 :

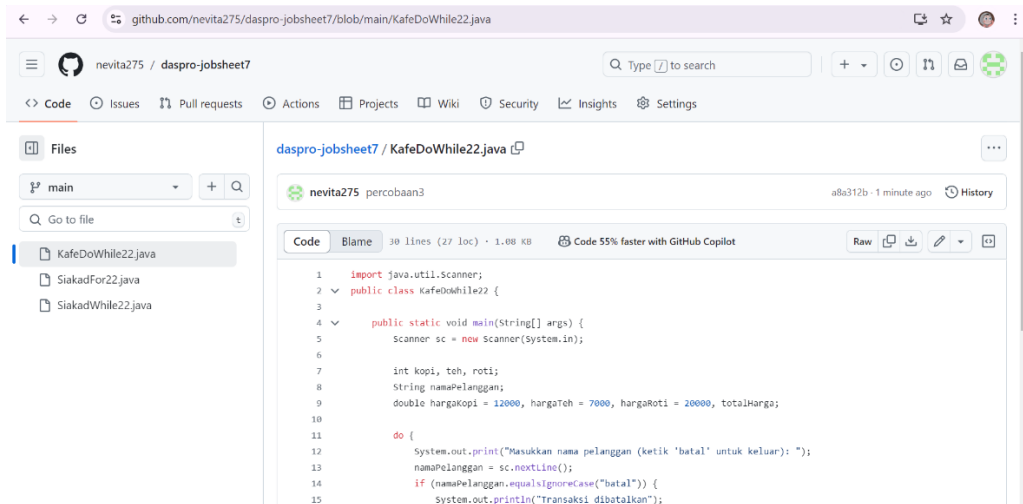
1. Buat file baru, beri nama KafeDoWhileNoAbsen.java
2. Buatlah struktur dasar program Java yang terdiri dari fungsi main().
3. Tambahkan library Scanner di bagian atas (luar) class
4. Buat deklarasi Scanner dengan nama variabel sc di dalam fungsi main()
5. Deklarasikan variabel kopi, teh, dan roti bertipe integer untuk menampung banyaknya item yang dibeli pelanggan, serta namaPelanggan bertipe String. Deklarasi dan inisialisasi hargaKopi dengan 12000, hargaTeh dengan 7000, hargaRoti dengan 20000.
6. Buat struktur perulangan DO-WHILE dengan kondisi true
7. Di dalam perulangan DO-WHILE tersebut, tambahkan perintah untuk memasukkan namaPelanggan. Kemudian tambahkan kondisi IF untuk mengecek isi variable namaPelanggan. Selanjutnya, tambahkan perintah untuk memasukkan banyaknya item yang dibeli pelanggan untuk setiap menu, apabila masukan nama pelanggan bukan "batal". Hitung total harga pembelian dan tampilkan hasilnya

```
do {
    System.out.print(s:"Masukkan nama pelanggan (ketik 'batal' untuk keluar): ");
    namaPelanggan = sc.nextLine();
    if (namaPelanggan.equalsIgnoreCase(anotherString:"batal")) {
        System.out.println(x:"Transaksi dibatalkan");
        break;
    }
    System.out.print(s:"Jumlah kopi: ");
    kopi = sc.nextInt();
    System.out.print(s:"Jumlah teh: ");
    teh = sc.nextInt();
    System.out.print(s:"Jumlah roti: ");
    roti = sc.nextInt();
    totalHarga = (kopi * hargaKopi) + (teh * hargaTeh) + (roti * hargaRoti);
    System.out.println("Total yang harus dibayar: Rp " + totalHarga);
    sc.nextLine();
} while (true);
System.out.println(x:"Semua transaksi selesai.");
}
```

## 8. Compile dan run program

```
Masukkan nama pelanggan (ketik 'batal' untuk keluar): Rena
Jumlah kopi: 3
Jumlah teh: 0
Jumlah roti: 1
Total yang harus dibayar: Rp 56000.0
Jumlah kopi: 1
Jumlah teh: 0
Jumlah roti: 1
Total yang harus dibayar: Rp 32000.0
Masukkan nama pelanggan (ketik 'batal' untuk keluar): BATAL
Transaksi dibatalkan
Semua transaksi selesai.
```

## 9. Commit dan push kode program ke Github



### PERTANYAAN :

1. Pada penggunaan DO-WHILE ini, apabila nama pelanggan yang dimasukkan pertama kali adalah “batal”, maka berapa kali perulangan dilakukan?  
⇒ Jika nama pelanggan yang dimasukkan pertama kali adalah "batal", perulangan do-while hanya dilakukan 1 kali.
2. Sebutkan kondisi berhenti yang digunakan pada perulangan DO-WHILE tersebut!  
⇒ Ketika nama pelanggan yang dimasukkan adalah "batal" maka perulangan akan berhenti.
3. Apa fungsi dari penggunaan nilai true pada kondisi DO-WHILE?  
⇒ Perulangan DO-WHILE akan terus berlanjut tanpa batas sampai ada pernyataan “break”
4. Mengapa perulangan DO-WHILE tersebut tetap berjalan meskipun tidak ada komponen inisialisasi dan update?  
⇒ Karena kondisi evaluasi di bagian WHILE dapat bergantung pada nilai yang diperbarui di dalam blok DO. Selama kondisi tersebut tetap true, perulangan akan terus berlangsung.

## TUGAS :

1. Seorang pengelola bioskop ingin membuat program untuk menghitung total penjuantiket dalam satu hari. Tiket dijual dengan harga Rp 50.000 per tiket. Program harus menghitung total tiket yang terjual dan total harga penjualan tiket selama satu hari dengan ketentuan sebagai berikut:
  - Jika pelanggan membeli lebih dari 4 tiket, pelanggan mendapatkan diskon 10%.
  - Jika pelanggan membeli lebih dari 10 tiket, pelanggan mendapatkan diskon 15%.
  - Jika input jumlah tiket tidak valid (negatif), program akan mengabaikan input tersebut dan meminta input ulang

```
1  import java.util.Scanner;
2  public class Tugas2 {
3
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6
7          int jenis, durasi, total = 0;
8          double tarifMobil = 3000, tarifMotor = 2000, totalPembayaran = 0;
9
10         while (true) {
11             System.out.print("Masukkan jenis kendaraan (1 untuk Mobil, 2 untuk Motor, 0 untuk keluar): ");
12             jenis = sc.nextInt();
13
14             if (jenis == 0) {
15                 break;
16             }
17
18             System.out.print("Masukkan durasi parkir (jam): ");
19             durasi = sc.nextInt();
20
21             if (durasi > 5) {
22                 total += 12500;
23             }
24             else {
25                 if (jenis == 1) {
26                     total += durasi * 3000;
27                 } else if (jenis == 2) {
28                     total += durasi * 2000;
29                 } else {
30                     System.out.println("Jenis kendaraan tidak valid.");
31                     continue;
32                 }
33             }
34
35             totalPembayaran += total;
36             System.out.println("Pembayaran untuk kendaraan: Rp " + total);
37
38             System.out.println("Total pembayaran parkir: Rp " + totalPembayaran);
39
40         }
41     }
42 }
```

The screenshot shows a GitHub repository page for a user named nevita275. The repository is named 'daspro-jobsheet7'. The file 'Tugas1.java' is selected, showing its code. The code is a Java program that calculates the total payment for parking tickets. It uses a Scanner to take input for the number of tickets and the type of vehicle (1 for Mobil, 2 for Motor, 0 for exit). It then calculates the total payment based on the number of tickets and the type of vehicle, applying a 10% discount for more than 4 tickets and a 15% discount for more than 10 tickets. The code is 36 lines long, has 29 lines of code, and is 1.24 KB in size. It is 55% faster with GitHub Copilot. The file is named 'Tugas1.java' and was last modified by fadd97 45 minutes ago. The repository also contains other files like 'KafeDoWhile22.java', 'SiakadFor22.java', 'SiakadWhile22.java', and 'Tugas2.java'.

```
1  import java.util.Scanner;
2  public class Tugas1 {
3      public static void main(String[] args) {
4          Scanner sc = new Scanner(System.in);
5
6          double harga = 50000, totalHarga = 0, diskon = 0;
7          int totalTiket = 0, jumlahPelanggan, jumlahTiket;
8
9          System.out.print("Masukkan jumlah pelanggan = ");
10         jumlahPelanggan = sc.nextInt();
11
12         for (int i = 0; i < jumlahPelanggan; i++) {
13             while (true) {
14                 System.out.print("Masukkan jumlah tiket untuk pelanggan " + (i + 1) + ": ");
15                 jumlahTiket = sc.nextInt();
```



2. Sebuah tempat parkir ingin membuat program untuk menghitung total pembayaran parkir dari beberapa kendaraan. Tarif parkir adalah Rp 3.000 per jam untuk mobil dan Rp 2.000 per jam untuk motor. Namun, jika durasi parkir lebih dari 5 jam, diberikan tarif tetap sebesar Rp 12.500 untuk semua kendaraan. Program akan terus meminta masukan selama input bukan 0. Implementasikan flowchart tersebut ke dalam bentuk kode program Java!

```
1 import java.util.Scanner;
2 public class Tugas1 {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5
6         double harga = 50000, totalHarga = 0, diskon = 0;
7         int totalTiket = 0, jumlahPelanggan, jumlahTiket;
8
9         System.out.print("Masukkan jumlah pelanggan = ");
10        jumlahPelanggan = sc.nextInt();
11
12        for (int i = 0; i < jumlahPelanggan; i++) {
13            while (true) {
14                System.out.print("Masukkan jumlah tiket untuk pelanggan " + (i + 1) + ": ");
15                jumlahTiket = sc.nextInt();
16
17                if (jumlahTiket < 0) {
18                    System.out.println("Jumlah tiket tidak valid. Masukkan lagi.");
19                } else {
20                    break;
21                }
22            }
23            if (jumlahTiket > 10) {
24                diskon = (jumlahTiket * harga) * 0.15;
25            } else if (jumlahTiket > 4) {
26                diskon = (jumlahTiket * harga) * 0.10;
27            }
28            totalTiket += jumlahTiket;
29            totalHarga = (harga * jumlahTiket) - diskon;
30        }
31        System.out.println("Total tiket yang terjual: " + totalTiket);
32        System.out.println("Total harga penjualan tiket: Rp " + totalHarga);
33    }
34 }
35
```

github.com/nevita275/daspro-jobsheet7/blob/main/Tugas2.java

nevita275 / daspro-jobsheet7

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Files

- main
- KafeDoWhile22.java
- SiakadFor22.java
- SiakadWhile22.java
- Tugas1.java
- Tugas2.java

daspro-jobsheet7 / Tugas2.java

nevita275 Tugas2 c2ba08d · now History

Code Blame 42 lines (32 loc) · 1.22 KB Code 55% faster with GitHub Copilot

```
1 import java.util.Scanner;
2 public class Tugas2 {
3
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         int jenis, durasi, total = 0;
8         double tarifMobil = 3000, tarifMotor = 2000, totalPembayaran = 0;
9
10        while (true) {
11            System.out.print("Masukkan jenis kendaraan (1 untuk Mobil, 2 untuk Motor, 0 untuk keluar): ");
12            jenis = sc.nextInt();
13
14            if (jenis == 0) {
15                break;
```