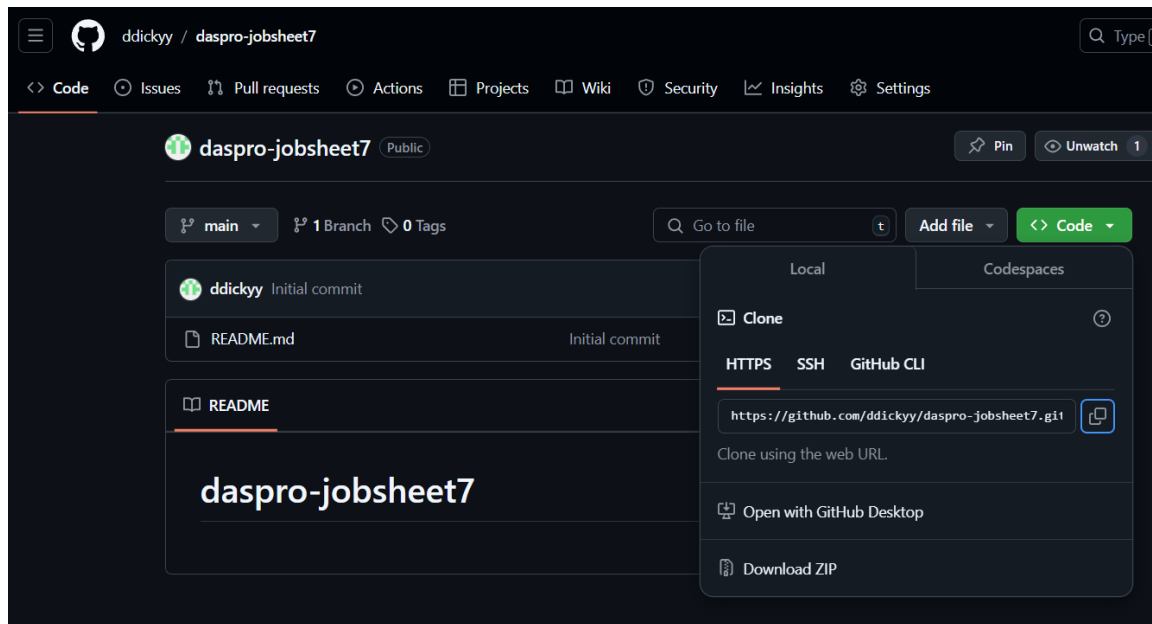


Nama : Dicky Darmawan
Kelas : TI 1B
No. Absen : 08
NIM : 244107020037

Percobaan 1: Studi Kasus Nilai Mahasiswa di SIAKAD – Perulangan FOR

Langkah-langkah:

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



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

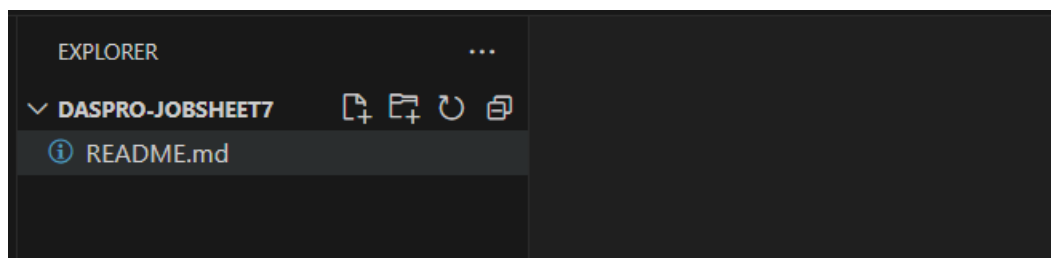
```
D:\JAVA>git clone https://github.com/ddickyy/daspro-jobsheet7.git
Cloning into 'daspro-jobsheet7'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.

D:\JAVA>cd daspro-jobsheet7

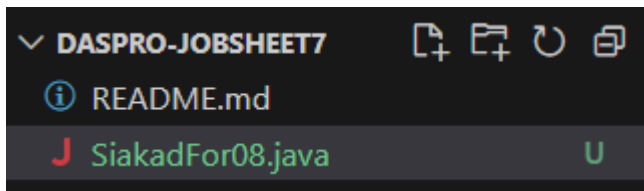
D:\JAVA\daspro-jobsheet7>code .

D:\JAVA\daspro-jobsheet7>
```

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().

```
public class SiakadFor08 {  
    Run main | Debug main | Run | Debug  
    public static void main(String[] args) {  
    }  
}
```

6. Tambahkan library Scanner di bagian atas (luar) class

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

7. Buat deklarasi Scanner dengan nama variabel sc di dalam fungsi main()

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

8. Deklarasikan variabel nilai, tertinggi, dan terendah bertipe double. Inisialisasi tertinggi dengan 0 dan terendah dengan 100

```
double nilai, tertinggi = 0, terendah = 100;
```

9. Buat struktur perulangan FOR dengan batas kondisi sesuai jumlah mahasiswa yaitu 10

```
for (int i = 1; i<= 10; i++) {
```

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

```
    for (int i = 1; i<= 10; i++) {  
        System.out.print("Masukkan nilai mahasiswa ke-" +i + ": ");  
        nilai = sc.nextDouble();  
        if (nilai > tertinggi) {  
            tertinggi = nilai;  
        }  
        if (nilai < terendah) {  
            terendah = nilai;  
        }  
    }
```

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

```
System.out.println("Nilai tertinggi : " +tertinggi);  
System.out.println("Nilai terendah : " +terendah);
```

Output:

```
Masukkan nilai mahasiswa ke-1: 88,8  
Masukkan nilai mahasiswa ke-2: 92,6  
Masukkan nilai mahasiswa ke-3: 90,5  
Masukkan nilai mahasiswa ke-4: 84,8  
Masukkan nilai mahasiswa ke-5: 82,3  
Masukkan nilai mahasiswa ke-6: 76,2  
Masukkan nilai mahasiswa ke-7: 72,9  
Masukkan nilai mahasiswa ke-8: 63,6  
Masukkan nilai mahasiswa ke-9: 94,8  
Masukkan nilai mahasiswa ke-10: 77,2  
Nilai tertinggi : 94.8  
Nilai terendah : 63.6
```

12. Compile dan run program

13. Commit dan push kode program ke Github

Pertanyaan

1. Sebutkan dan tunjukkan masing-masing komponen perulangan FOR pada kode program Percobaan 1!
 - inisialisasi: `int i = 1`
 - kondisi: `i <= 10`
 - update: `i++`
2. Mengapa variabel tertinggi diinisialisasi 0 dan terendah diinisialisasi 100? Apa yang terjadi jika variabel tertinggi diinisialisasi 100 dan terendah diinisialisasi 0?
nilai tertinggi diinisialisasi 0 semua nilai yang diinputkan akan lebih besar dari 0, sehingga nilai tertinggi bisa diupdate dengan nilai yang diinputkan. Nilai terendah diinisialisasi 100 semua nilai yang diinputkan akan lebih kecil atau sama dengan 100, sehingga nilai terendah bisa diupdate dengan nilai yang diinputkan. Jika dirubah maka kemungkinan output yang dihasilkan salah atau error.
3. Jelaskan fungsi dan alur kerja dari potongan kode berikut!

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

Fungsinya adalah untuk memilih input angka dari terendah hingga tertinggi dengan rentang 0-100,

Fungsi tersebut kemudian melakukan perulangan untuk menginputkan nilai-nilai yang akan dicari nilai tertinggi dan terendahnya. kemudian membandingkan nilai yang diinputkan dengan nilai tertinggi dan terendah yang sudah diinisialisasi.

Jika nilai yang diinputkan lebih besar dari nilai tertinggi, maka nilai tertinggi diupdate dengan nilai yang diinputkan. Jika nilai yang diinputkan lebih kecil dari nilai terendah, maka nilai terendah diupdate dengan nilai yang diinputkan.

Setelah perulangan selesai, fungsi tersebut mengembalikan nilai tertinggi dan terendah yang sudah diupdate.

4. Modifikasi kode program sehingga terdapat perhitungan untuk menentukan berapa mahasiswa yang lulus dan yang tidak lulus berdasarkan batas kelulusan (nilai minimal 60).

```
double nilai, tertinggi = 0, terendah = 100;
int lulus = 0, tidak_lulus = 0;

for (int i = 1; i <= 10; i++) {
    System.out.print("Masukkan nilai mahasiswa ke-" + i + ": ");
    nilai = sc.nextDouble();
    if (nilai > tertinggi) {
        tertinggi = nilai;
    }
    if (nilai < terendah) {
        terendah = nilai;
    }
    if (nilai >= 60) {
        lulus++;
    }
    if (nilai < 60) {
        tidak_lulus++;
    }
}

System.out.println("Nilai tertinggi : " + tertinggi);
System.out.println("Nilai terendah : " + terendah);
System.out.println("Jumlah mahasiswa yang lulus sebanyak " + lulus + " orang");
System.out.println("Jumlah mahasiswa yang tidak lulus sebanyak " + tidak_lulus + " orang");
```

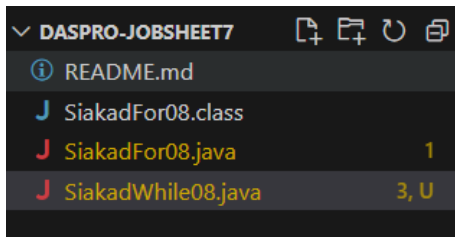
5. Tampilkan jumlah mahasiswa lulus dan tidak lulus setelah menampilkan nilai tertinggi dan terendah!

```
Masukkan nilai mahasiswa ke-1: 60
Masukkan nilai mahasiswa ke-2: 50
Masukkan nilai mahasiswa ke-3: 8
Masukkan nilai mahasiswa ke-4: 95
Masukkan nilai mahasiswa ke-5: 82
Masukkan nilai mahasiswa ke-6: 64
Masukkan nilai mahasiswa ke-7: 59
Masukkan nilai mahasiswa ke-8: 60
Masukkan nilai mahasiswa ke-9: 92
Masukkan nilai mahasiswa ke-10: 86
Nilai tertinggi : 95.0
Nilai terendah : 8.0
Jumlah mahasiswa yang lulus sebanyak 7 orang
Jumlah mahasiswa yang tidak lulus sebanyak 3 orang
```

Percobaan 2: Studi Kasus Nilai Mahasiswa di SIAKAD – Perulangan WHILE

Langkah-langkah:

1. Buat file baru, beri nama SiakadWhileNoAbsen.java



2. Buatlah struktur dasar program Java yang terdiri dari fungsi main().

```
public class SiakadWhile08 {  
  
    Run main | Debug main | Run | Debug  
    public static void main(String[] args) {
```

3. Tambahkan library Scanner di bagian atas (luar) class

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

4. Buat deklarasi Scanner dengan nama variabel sc di dalam fungsi main()

```
public class SiakadWhile08 {  
  
    Run main | Debug main | Run | Debug  
    public static void main(String[] args) {  
        Scanner sc = new Scanner (System.in);  
    }  
}
```

5. Deklarasikan variabel nilai, jml, dan i (untuk perulangan) bertipe integer. Inisialisasi i dengan 0 sebagai nilai awal perulangan

```
int i = 0, jml, nilai;
```

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.

```
System.out.print(s:"Masukkan jumlah mahasiswa: ");  
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

```
while (i < jml) {
```

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

```
System.out.print("Masukkan nilai mahasiswa ke " + (i + 1) + ": ");
nilai = sc.nextInt();

if (nilai < 0 || nilai > 100) {
    System.out.println("Nilai tidak valid. Masukkan lagi nilai yang valid");
    continue;
}
if (nilai > 80 && nilai <= 100) {
    System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah A");
} else if (nilai > 73 && nilai <= 80) {
    System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah B+");
} else if (nilai > 65 && nilai <= 73) {
    System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah B");
} else if (nilai > 60 && nilai <= 65) {
    System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah C+");
} else if (nilai > 50 && nilai <= 60) {
    System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah C");
} else if (nilai > 39 && nilai <= 50) {
    System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah D");
} else {
    System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah D");
}
i++;
```

Output:

```
Masukkan jumlah mahasiswa: 5
Masukkan nilai mahasiswa ke 1: 88
Nilai mahasiswa ke-1 adalah A
Masukkan nilai mahasiswa ke 2: 56
Nilai mahasiswa ke-2 adalah C
Masukkan nilai mahasiswa ke 3: 91
Nilai mahasiswa ke-3 adalah A
Masukkan nilai mahasiswa ke 4: 62
Nilai mahasiswa ke-4 adalah C+
Masukkan nilai mahasiswa ke 5: 28
Nilai mahasiswa ke-5 adalah D
```

9. Compile dan run program
10. Commit dan push kode program ke Github

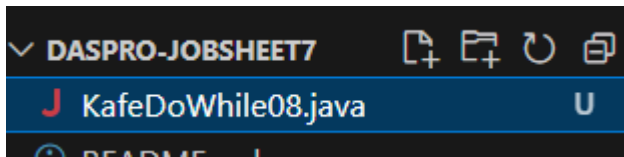
Pertanyaan

1. Pada potongan kode berikut, tentukan maksud dan kegunaan dari sintaks berikut:
 - a. `nilai < 0 || nilai > 100`
operator relasi (OR), berguna untuk membaca input apakah sesuai dengan operator atau tidak, operator OR berguna agar memastikan logika/angka memenuhi salah satu syarat
 - b. `continue`
digunakan untuk melewati (skip) iterasi saat ini (1 iterasi saja) dan melanjutkan ke iterasi berikutnya. Pada kasus ini berguna untuk mengulang kembali input jika terjadi kesalahan dan skip iterasi selanjutnya
2. Mengapa sintaks `i++` dituliskan di akhir perulangan WHILE? Apa yang terjadi jika posisinya dituliskan di awal perulangan WHILE?
dituliskan di akhir perulangan WHILE karena berfungsi sebagai penghitung untuk mengontrol jumlah perulangan. Nilai `i` akan bertambah 1 setiap kali perulangan dilakukan, jika dituliskan di awal perulangan WHILE, maka perulangan tidak akan pernah berhenti.
3. Apabila jumlah mahasiswa yang dimasukkan adalah 19, berapa kali perulangan WHILE akan berjalan?
Berjalan sebanyak 20 kali
4. Modifikasi kode program sehingga apabila terdapat mahasiswa yang mendapat nilai A, program menampilkan pesan tambahan "Bagus, pertahankan nilainya"!
5. Commit dan push kode program ke Github

Percobaan 3: Studi Kasus Transaksi di Kafe – Perulangan DO-WHILE

Langkah-langkah:

1. Buat file baru, beri nama KafeDoWhileNoAbsen.java



2. Buatlah struktur dasar program Java yang terdiri dari fungsi main().

```
public class KafeDoWhile08 {  
    Run main | Debug main | Run | Debug  
    public static void main(String[] args) {}  
}
```

3. Tambahkan library Scanner di bagian atas (luar) class

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

4. Buat deklarasi Scanner dengan nama variabel sc di dalam fungsi main()

```
import java.util.Scanner;  
  
public class KafeDowhile08 {  
    Run main | Debug main | Run | Debug  
    public static void main(String[] args) {  
        Scanner sc = new Scanner (System.in);  
    }  
}
```

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.

```
int kopi, teh, roti;  
String namaPelanggan;  
int hargaKopi = 12000, hargaTeh = 7000, hargaRoti = 20000;
```


6. Buat struktur perulangan DO-WHILE dengan kondisi true

```
do {  
  
} while (true);
```

7. Di dalam perulangan DO-WHILE tersebut, tambahkan perintah untuk memasukkan namaPelanggan. Kemudian tambahkan kondisi IF untuk mengecek isi variabel 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 dibayar: Rp" +totalHarga);  
    sc.nextLine();  
} while (true);  
  
System.out.println(x:"Semua transaksi selesai.");
```

Output:

```
Masukkan nama pelanggan (ketik 'batal' untuk keluar): Dicky  
Jumlah kopi: 1  
Jumlah teh: 1  
Jumlah roti: 1  
Total yang dibayar: Rp39000  
Masukkan nama pelanggan (ketik 'batal' untuk keluar): batal  
Transaksi dibatalkan  
Semua transaksi selesai.
```

8. Compile dan run program
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?
Sebanyak 1 kali
2. Sebutkan kondisi berhenti yang digunakan pada perulangan DO-WHILE tersebut!
Nama pelanggan yang dimasukkan adalah “batal”
3. Apa fungsi dari penggunaan nilai true pada kondisi DO-WHILE?
Berfungsi ketika selama nama pelanggan dimasukkan selain “batal” maka perulangan akan terus berlanjut.
4. Mengapa perulangan DO-WHILE tersebut tetap berjalan meskipun tidak ada komponen inisialisasi dan update?
karena kondisi while (true) selalu bernilai true berarti bahwa loop tidak memiliki kondisi berhenti yang bergantung pada variabel atau kondisi lain. Selama kata yang dimasukkan benar maka akan terus berlanjut.

Tugas

1. Seorang pengelola bioskop ingin membuat program untuk menghitung total penjualan tiket 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
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