

TUGAS MATA KULIAH
PRAKTIKUM DASAR PEMROGRAMAN

JOBSHEET 7
PERULANGAN 1

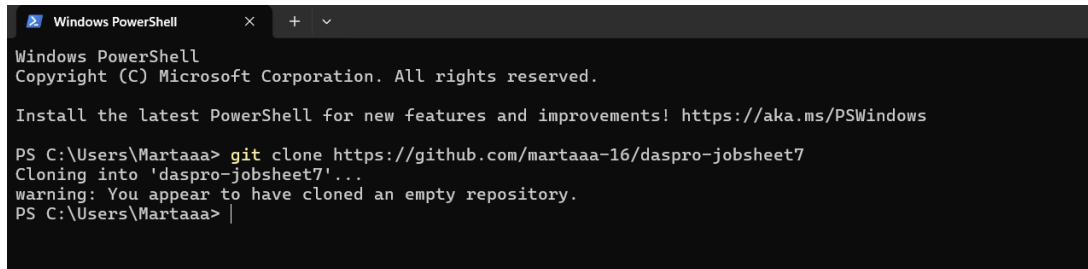


OLEH:
MARTA PRAMA DANISWARA
244107020205

PROGRAM STUDI D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
MALANG
2024

2.1 PERCOBAAN 1: Studi Kasus Nilai Mahasiswa di SIAKAD – Perulangan FOR

1. Buat repo baru di GitHub dengan nama **daspro-jobsheet7**, kemudian cloning di terminal

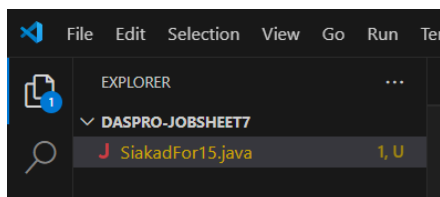


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

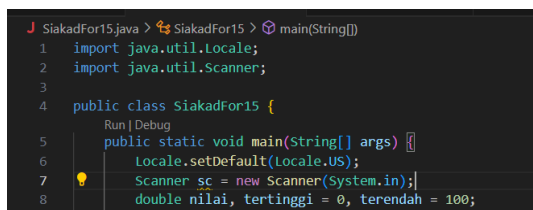
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Martaaa> git clone https://github.com/martaaa-16/daspro-jobsheet7
Cloning into 'daspro-jobsheet7'...
warning: You appear to have cloned an empty repository.
PS C:\Users\Martaaa>
```

2. Buka VS Code, buat file baru dengan nama **SiakadFor15.java**

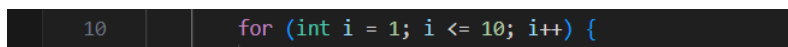


3. Buat struktur dasar Java dan deklarasi Scanner dengan variabel **sc**. Deklarasi variabel **nilai**, **tertinggi**, dan **terendah** bertipe *double*. Inisialisasi **tertinggi** dengan 0 dan **terendah** dengan 100. Tambahkan **import java.util.Locale;** diatas **import java.util.Scanner;**, kemudian tambahkan juga **Locale.setDefault(Locale.US);** dibawah psvm.



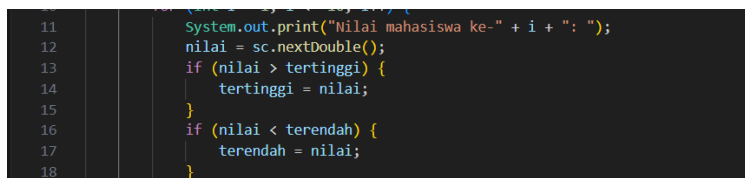
```
J SiakadFor15.java > SiakadFor15 > main(String[])
1  import java.util.Locale;
2  import java.util.Scanner;
3
4  public class SiakadFor15 {
5      Run | Debug
6      public static void main(String[] args) {
7          Locale.setDefault(Locale.US);
8          Scanner sc = new Scanner(System.in);
9          double nilai, tertinggi = 0, terendah = 100;
```

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



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

5. Buat dua kondisi pemilihan secara terpisah untuk mengecek **nilai** tertinggi dan terendah dengan membandingkan nilai masukan dengan variabel **tertinggi** dan variabel **terendah**



```
11          System.out.print("Nilai mahasiswa ke-" + i + ": ");
12          nilai = sc.nextDouble();
13          if (nilai > tertinggi) {
14              tertinggi = nilai;
15          }
16          if (nilai < terendah) {
17              terendah = nilai;
18          }
```

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

```

19     }
20     System.out.println("Nilai tertinggi: " + tertinggi);
21     System.out.println("Nilai terendah: " + terendah);
22
23
24 }

```

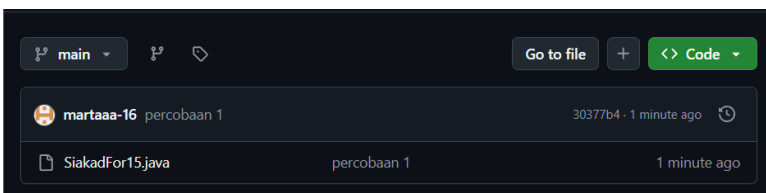
7. Run program

```

Nilai mahasiswa ke-1: 92.6
Nilai mahasiswa ke-2: 90
Nilai mahasiswa ke-3: 88.3
Nilai mahasiswa ke-4: 78.2
Nilai mahasiswa ke-5: 89.9
Nilai mahasiswa ke-6: 91.2
Nilai mahasiswa ke-7: 80.5
Nilai mahasiswa ke-8: 82.5
Nilai mahasiswa ke-9: 77
Nilai mahasiswa ke-10: 92.3
Nilai tertinggi: 92.6
Nilai terendah: 77.0
PS C:\Users\Martaaa>

```

8. Commit dan push ke GitHub



Jawaban

1. Komponen perulangan FOR pada kode program diatas, yaitu:

- **Inisialisasi:** `int i = 1` (variabel `i` diinisialisasi ke 1)
- **Kondisi:** `i <= 10` (perulangan berlanjut selama `i` kurang dari sama dengan 10)
- **Perubahan Nilai:** `i++` (setiap iterasi, `i` bertambah 1).
- **Blok Kode:** `System.out.println(i)` mencetak nilai `i`.

2. Jika nilai **tertinggi** diinisialisasi dengan angka kecil, maka setiap angka dalam data yang lebih besar dari 0 akan menggantikan nilai tersebut, sehingga kita mendapatkan nilai maksimum dari kumpulan data. Biasanya kita tidak tahu nilai minimum dari data, sehingga kita memilih 0 sebagai nilai default yang rendah. Jika nilai **terendah** diinisialisasi dengan angka besar, setiap angka yang lebih kecil dari 100 akan menggantikan nilai tersebut, sehingga kita dapat menemukan nilai terkecil dalam data.

Jika variabel **tertinggi** diinisialisasi dengan nilai besar seperti 100, maka kita tidak akan menemukan nilai tertinggi yang benar jika semua nilai dalam kumpulan data lebih kecil dari 100. Akibatnya, nilai tertinggi tidak akan berubah dan tetap 100, padahal nilai tertinggi dalam data adalah 30. Jika variabel **terendah** diinisialisasi dengan 0, maka kita tidak akan menemukan nilai terendah yang benar jika ada nilai yang lebih besar dari 0 dalam kumpulan data. Akibatnya, nilai terendah tidak akan berubah dan tetap 0, padahal nilai terendah dalam data adalah 10.

3. **if (nilai > tertinggi)**, bertujuan untuk mengecek apakah nilai yang baru dimasukkan lebih besar dari nilai tertinggi yang sudah ada. Sedangkan **if (nilai > terendah)**, bertujuan untuk mengecek apakah nilai yang baru dimasukkan lebih kecil dari nilai tertinggi yang sudah ada

Kesimpulan: Potongan kode tersebut adalah bagian dari proses untuk menemukan nilai maksimum dan minimum dari serangkaian nilai yang dimasukkan, memastikan bahwa program dapat mengidentifikasi nilai tertinggi dan terendah dengan benar setelah semua input selesai.

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!

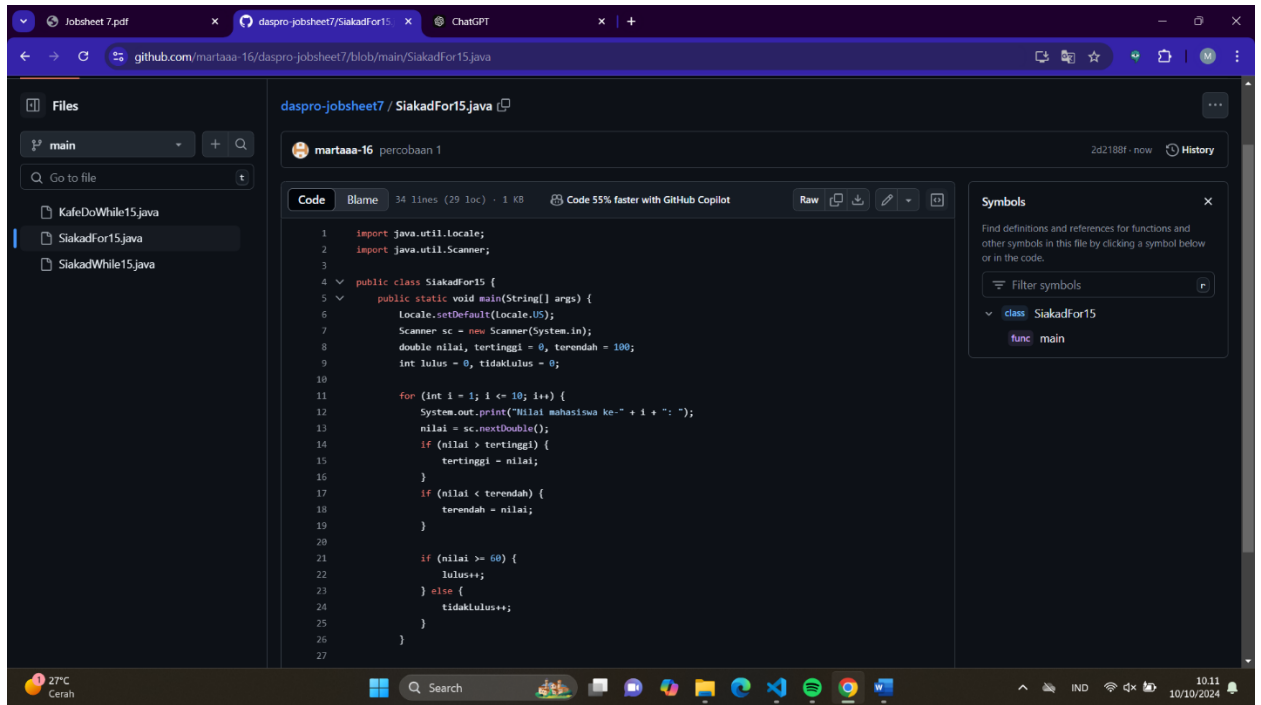
The screenshot shows an IDE with a project named 'DASPRO-JOBSHEET7'. The file explorer on the left lists several Java files: 'KafeDoWhile15.java', 'SiakadFor15.java', and 'SiakadWhile15.java'. The main editor displays the code for 'SiakadFor15.java'. The code imports 'java.util.Locale' and 'java.util.Scanner', defines a 'SiakadFor15' class, and implements a 'main' method. The 'main' method sets the locale to 'US', creates a 'Scanner' for input, and initializes variables for 'nilai', 'tertinggi', 'terendah', 'lulus', and 'tidaklulus'. It then enters a 'for' loop that iterates 10 times, prompting the user to enter a score. Inside the loop, it updates 'tertinggi' and 'terendah' if the current score is higher or lower than the current values. After the loop, it checks if the number of 'lulus' (passed) students is greater than or equal to 60. Finally, it prints the highest score, the lowest score, the number of students who passed, and the number of students who did not pass.

```
1 import java.util.Locale;
2 import java.util.Scanner;
3
4 public class SiakadFor15 {
5     public static void main(String[] args) {
6         Locale.setDefault(Locale.US);
7         Scanner sc = new Scanner(System.in);
8         double nilai, tertinggi = 0, terendah = 100;
9         int lulus = 0, tidaklulus = 0;
10
11         for (int i = 1; i <= 10; i++) {
12             System.out.print("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
21             if (nilai >= 60) {
22                 lulus++;
23             } else {
24                 tidaklulus++;
25             }
26         }
27         System.out.println("Nilai tertinggi: " + tertinggi);
28         System.out.println("Nilai terendah: " + terendah);
29         System.out.println("Jumlah mahasiswa yang lulus: " + lulus);
30         System.out.println("Jumlah mahasiswa yang tidak lulus: " + tidaklulus);
31     }
32 }
```

The terminal window shows the output of the Java program. It displays the scores for 10 students, the highest score (98.0), the lowest score (23.0), the number of students who passed (7), and the number of students who did not pass (3). The prompt 'PS C:\Users\Martaaa\daspro-jobsheet7>' is visible at the bottom.

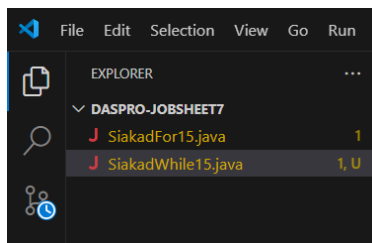
```
Nilai mahasiswa ke-1: 67
Nilai mahasiswa ke-2: 87
Nilai mahasiswa ke-3: 90
Nilai mahasiswa ke-4: 78
Nilai mahasiswa ke-5: 23
Nilai mahasiswa ke-6: 67
Nilai mahasiswa ke-7: 54
Nilai mahasiswa ke-8: 98
Nilai mahasiswa ke-9: 87
Nilai mahasiswa ke-10: 43
Nilai tertinggi: 98.0
Nilai terendah: 23.0
Jumlah mahasiswa yang lulus: 7
Jumlah mahasiswa yang tidak lulus: 3
PS C:\Users\Martaaa\daspro-jobsheet7>
```

5. Commit dan push code program ke Github



2.2 PERCOBAAN 2: Studi Kasus Nilai Mahasiswa di SIAKAD – Perulangan WHILE

1. Buat file baru, beri nama **SiakadWhile15.java**



2. Buat struktur dasar Java dan deklarasi Scanner dengan variabel **sc**. Deklarasi variabel **nilai**, **jml**, dan **i** (untuk perulangan) bertipe **integer**. Inisialisasi **i** dengan 0 sebagai nilai awal perulangan

```

1  import java.util.Scanner;
2
3  public class SiakadWhile15 {
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6          int nilai, jml, i = 0;
7      }

```

3. 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

```

7
8      System.out.print(s:"Masukkan jumlah nilai mahasiswa: ");
9      jml = sc.nextInt();
10

```

4. Buat struktur perulangan WHILE dengan batas kondisi sesuai jumlah mahasiswa yaitu 5. 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.

```

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

```

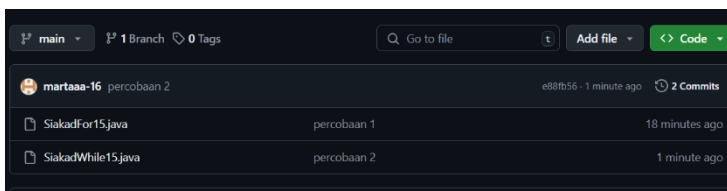
5. Run program

```

Masukkan jumlah nilai mahasiswa: 5
Nilai mahasiswa ke-1: 92
Hasil nilai mahasiswa ke-1 adalah A
Nilai mahasiswa ke-2: 78
Hasil nilai mahasiswa ke-2 adalah B+
Nilai mahasiswa ke-3: 65
Hasil nilai mahasiswa ke-3 adalah C+
Nilai mahasiswa ke-4: 109
Nilai tidak valid. Masukkan ulang nilai yang valid!
Nilai mahasiswa ke-4: 49
Hasil nilai mahasiswa ke-4 adalah D
Nilai mahasiswa ke-5: 36
Hasil nilai mahasiswa ke-5 adalah E
PS C:\Users\Martaaa>

```

6. Commit dan push ke GitHub



Jawaban

1. Tentukan maksud dan kegunaan dari sintaks berikut:

a. **nilai < 0 || nilai > 100**

- **nilai < 0**: Memeriksa apakah nilai yang dimasukkan lebih kecil dari 0 (nilai negatif tidak valid)
- **nilai > 100**: Memeriksa apakah nilai yang dimasukkan lebih besar dari 100. Jadi, nilai yang lebih tinggi dari 100 juga tidak valid.
- **|| (Operator OR)**: Menggabungkan kedua kondisi tersebut. Artinya, jika salah satu dari kondisi tersebut bernilai **true**, maka blok kode di dalam kurung kurawal akan dieksekusi.

b. **continue**

- Dalam perulangan **WHILE**, ketika **continue** dijalankan, kontrol program akan kembali ke awal dari perulangan. Jadi jika nilai tidak valid, program akan meminta pengguna untuk memasukkan nilai lagi tanpa menginkrementasi variabel penghitung (dalam hal ini, **i**), sehingga pengguna tetap pada mahasiswa yang sama.

2. Dengan menempatkan **i++** di akhir, setiap iterasi perulangan akan melakukan semua proses (seperti meminta input) terlebih dahulu, kemudian menginkrementasi nilai **i**. Ini memastikan bahwa semua operasi dalam perulangan dijalankan untuk nilai **i** saat ini sebelum beralih ke iterasi berikutnya.

Misalnya, jika **jml** adalah 5, perulangan akan berlangsung untuk nilai **i** dari 0 hingga 4 (5 iterasi), di mana setiap iterasi akan memproses mahasiswa ke-1 hingga mahasiswa ke-5.

3. **Total Iterasi**: Perulangan akan berjalan sebanyak **19 kali**, dengan **i** bertambah dari 0 hingga 18. Nilai **i** pada akhir perulangan adalah 19, di mana perulangan berhenti. Setiap iterasi mewakili satu mahasiswa yang input nilainya diminta.

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

```
19
20
21     if (nilai > 80 && nilai <= 100) {
22         System.out.println("Hasil nilai mahasiswa ke-" + (i + 1) + " adalah A");
23         if (nilai > 80 && nilai <= 100) {
24             System.out.println(x:"Bagus, pertahankan nilainya!");
```

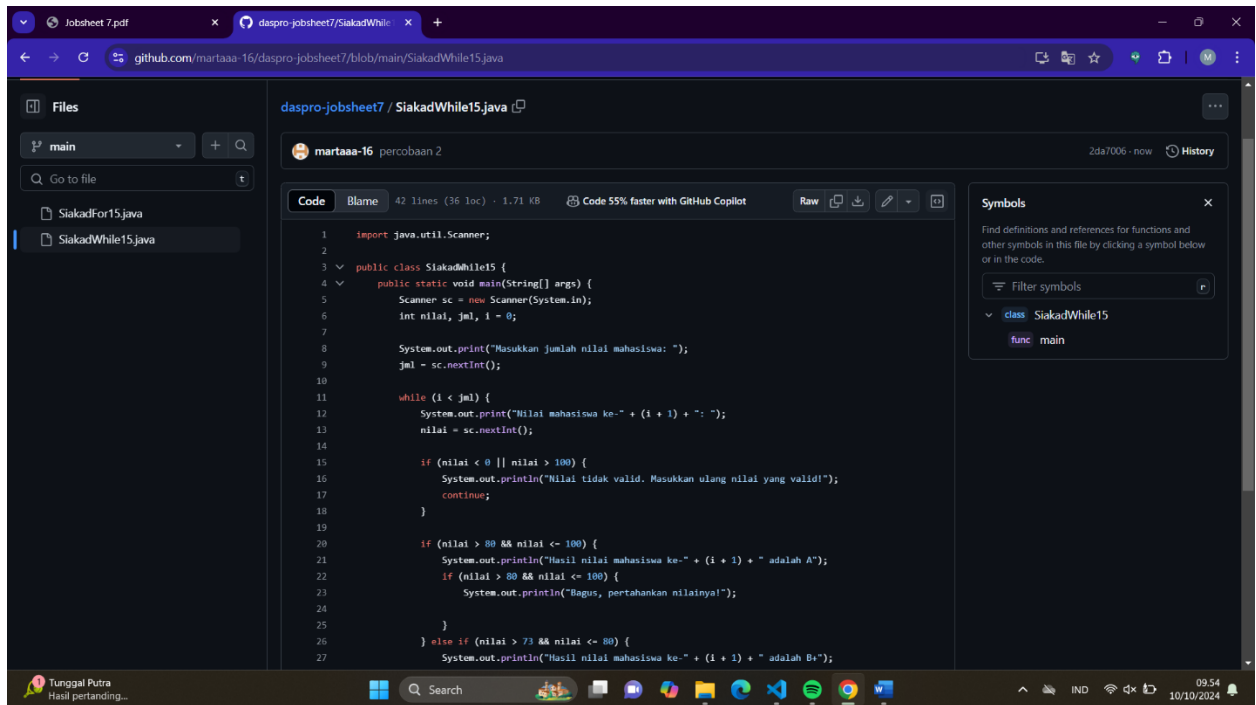


```

PS C:\Users\Martaaa\daspro-jobsheet7>
Masukkan jumlah nilai mahasiswa: 2
Nilai mahasiswa ke-1: 78
Hasil nilai mahasiswa ke-1 adalah B+
Nilai mahasiswa ke-2: 99
Hasil nilai mahasiswa ke-2 adalah A
Bagus, pertahankan nilainya!
PS C:\Users\Martaaa\daspro-jobsheet7>

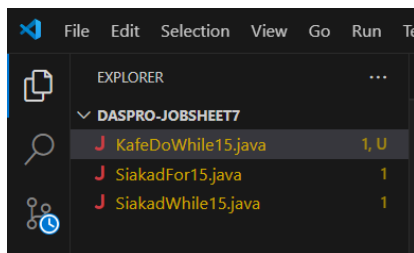
```

5. Commit dan push kode program ke GitHub



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

1. Buat file baru, beri nama **KafeDoWhile15.java**



- Buat struktur dasar Java dan deklarasi Scanner dengan variabel **sc**. Deklarasi 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. Deklarasi variable **totalHarga** untuk menghitung jumlah total harga yang ada.

```

J KafeDoWhile15.java > KafeDoWhile15
1  import java.util.Scanner;
2
3  public class KafeDoWhile15 {
4      Run | Debug
      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6          int kopi, teh, roti;
7          int hargaKopi = 12000, hargaTeh = 7000, hargaRoti = 20000;
8          int totalHarga;
9          String namaPelanggan;
10

```

3. Buat struktur perulangan DO-WHILE dengan kondisi **true**. 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

```

11      do {
12          System.out.print(s:"Nama pelanggan (ketik 'batal' untuk keluar): ");
13          namaPelanggan = sc.nextLine();
14          if (namaPelanggan.equalsIgnoreCase(anotherString:"batal")) {
15              System.out.println(x:"Transaksi dibatalkan.");
16              break;
17          }
18          System.out.print(s:"Jumlah kopi: ");
19          kopi = sc.nextInt();
20          System.out.print(s:"Jumlah teh: ");
21          teh = sc.nextInt();
22          System.out.print(s:"Jumlah roti: ");
23          roti = sc.nextInt();
24
25          totalHarga = (kopi*hargaKopi) + (teh*hargaTeh) + (roti*hargaRoti);
26          System.out.println("Total yang harus dibayar: Rp " + totalHarga);
27          sc.nextLine();
28      } while (true);
29      System.out.println(x:"Semua transaksi telah selesai.");
30
31  }

```

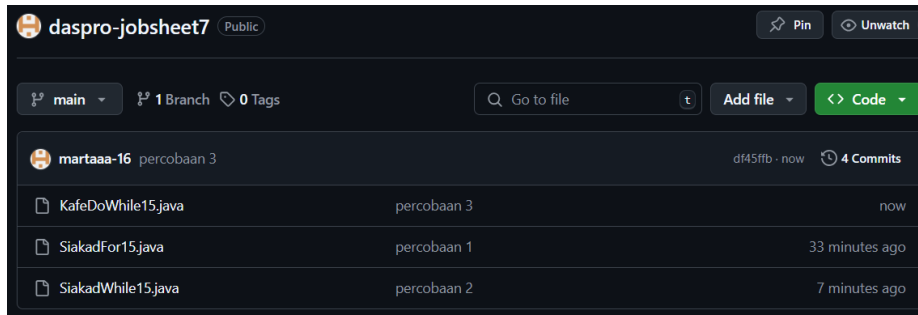
4. Run program

```

Nama pelanggan (ketik 'batal' untuk keluar): K
Jumlah kopi: 5
Jumlah teh: 9
Jumlah roti: 6
Total yang harus dibayar: Rp 243000
Nama pelanggan (ketik 'batal' untuk keluar): batal
Transaksi dibatalkan.
Semua transaksi telah selesai.
PS C:\Users\Martaaa\daspro-jobsheet>

```

5. Commit dan push ke GitHub



Jawaban

1. Jika **batal** dimasukkan sebagai input pertama, blok perulangan tetap akan dieksekusi **satu kali**.
2. Kondisi berhenti untuk perulangan **do-while** ini adalah ketika nama pelanggan yang dimasukkan sama dengan **batal**. Ketika itu terjadi, perintah **break** menghentikan perulangan dan keluar dari program.
3. Penggunaan **true** pada kondisi **while (true)** dalam perulangan **do-while** membuat perulangan menjadi **tak terbatas** (infinite loop) sampai ada mekanisme lain yang menghentikannya. Perulangan dengan kondisi **true** berarti bahwa kondisi ini akan selalu **benar**, sehingga perulangan tidak akan berhenti dengan sendirinya.
4. Karena kode tersebut dirancang untuk menunggu input dari pengguna sebagai kondisi untuk melanjutkan atau menghentikan perulangan. Tanpa input "batal", perulangan akan terus berlangsung.

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.

JAWAB:

- Kode program

```
J Tugas1.java > Tugas1 > main(String[])
1  import java.util.Scanner;
2
3  public class Tugas1 {
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6          int jumlahTiket, hargaTiket = 50000, totalTiket = 0, tiketTerjual = 0;
7          double totalHarga = 0;
8
9          do {
10             System.out.print(s:"Jumlah pembelian tiket (ketik '0' untuk pembatalan): ");
11             jumlahTiket = sc.nextInt();
12
13             if (jumlahTiket == 0) {
14                 System.out.println(x:"Dibatalan.");
15                 break;
16             }
17             if (jumlahTiket < 0) {
18                 System.out.println(x:"Data tidak valid.");
19                 continue;
20             }
21
22             totalHarga = jumlahTiket * hargaTiket;
23
24             if (jumlahTiket > 10) {
25                 totalHarga = totalHarga * 0.85;
26             } else if (jumlahTiket > 4) {
27                 totalHarga = totalHarga * 0.9;
28             }
29
30             System.out.println("Total harga untuk " + jumlahTiket + " tiket yang harus dibayarkan " + totalHarga);
31
32             totalTiket += jumlahTiket;
33             tiketTerjual += totalHarga;
34
35         } while (true);
36         System.out.print(s:"Hasil yang terjual: ");
37         System.out.println("Jumlah tiket yang terjual: " + totalTiket);
38         System.out.println("Total pendapatan: Rp " + tiketTerjual);
39     }
40 }
41
```

- Output

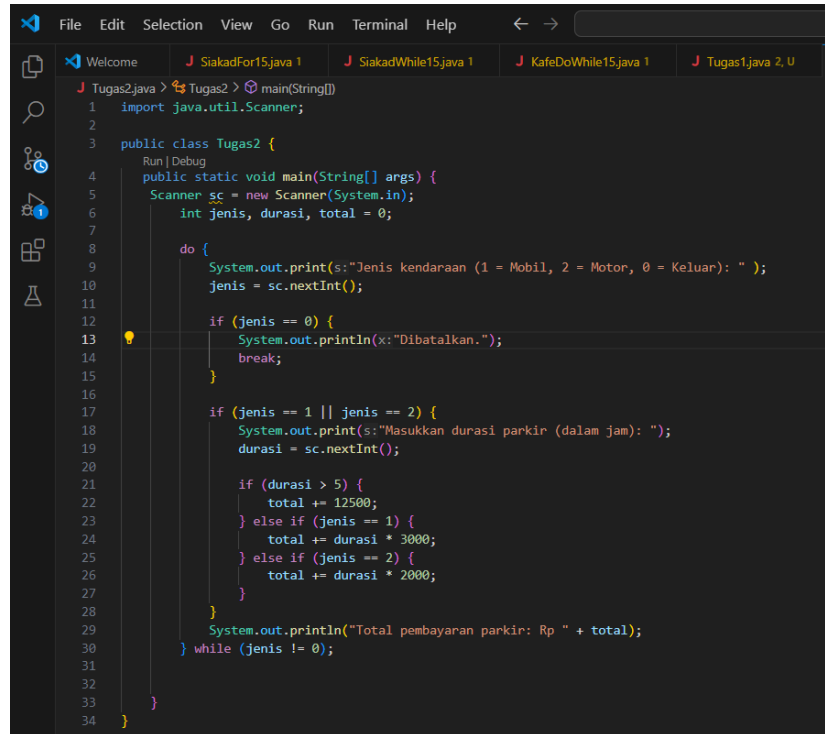
```
Jumlah pembelian tiket (ketik '0' untuk pembatalan): 8
Total harga untuk 8 tiket yang harus dibayarkan 360000.0
Jumlah pembelian tiket (ketik '0' untuk pembatalan): 6
Total harga untuk 6 tiket yang harus dibayarkan 270000.0
Jumlah pembelian tiket (ketik '0' untuk pembatalan): 0
Dibatalan.
Hasil yang terjual: Jumlah tiket yang terjual: 14
Total pendapatan: Rp 630000
PS C:\Users\Martaaa\daspro-jobsheet7>
```

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!

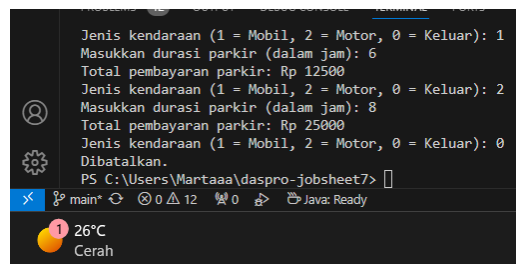
JAWAB:

- Kode program



```
File Edit Selection View Go Run Terminal Help
Tugas2.java > SiakadFor15.java 1 SiakadWhile15.java 1 KafeDoWhile15.java 1 Tugas1.java 2, U
Tugas2.java > main(String[])
1 import java.util.Scanner;
2
3 public class Tugas2 {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         int jenis, durasi, total = 0;
7
8         do {
9             System.out.print(s:"Jenis kendaraan (1 = Mobil, 2 = Motor, 0 = Keluar): " );
10            jenis = sc.nextInt();
11
12            if (jenis == 0) {
13                System.out.println(x:"Dibatalkan.");
14                break;
15            }
16
17            if (jenis == 1 || jenis == 2) {
18                System.out.print(s:"Masukkan durasi parkir (dalam jam): ");
19                durasi = sc.nextInt();
20
21                if (durasi > 5) {
22                    total += 12500;
23                } else if (jenis == 1) {
24                    total += durasi * 3000;
25                } else if (jenis == 2) {
26                    total += durasi * 2000;
27                }
28            }
29            System.out.println("Total pembayaran parkir: Rp " + total);
30        } while (jenis != 0);
31    }
32
33 }
34 }
```

- Output



```
Jenis kendaraan (1 = Mobil, 2 = Motor, 0 = Keluar): 1
Masukkan durasi parkir (dalam jam): 6
Total pembayaran parkir: Rp 12500
Jenis kendaraan (1 = Mobil, 2 = Motor, 0 = Keluar): 2
Masukkan durasi parkir (dalam jam): 8
Total pembayaran parkir: Rp 25000
Jenis kendaraan (1 = Mobil, 2 = Motor, 0 = Keluar): 0
Dibatalkan.
PS C:\Users\Martaaa\daspro-jobsheet7>
```