

**SELASA 28 OKTOBER 2025**

**LAPORAN HASIL PRAKTIKUM**

**Jobsheet 7**



**Disusun oleh:**

**Rafif Rizdan Prastana**

**Kelas 1H/IT**

**254107020052**

**PROGRAM STUDI D-IV TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2025**

## 2. Praktikum

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

#### Waktu Percobaan: 90 menit

Di dalam Sistem Informasi Akademik (SIKAD), dosen mengisi nilai mata kuliah Praktikum Dasar Pemrograman yang ditempuh oleh mahasiswa. Dosen tersebut ingin mencari nilai tertinggi dan terendah Kuis dari 10 mahasiswa di dalam satu kelas. Dosen tersebut harus memasukkan nilai dari setiap siswa, kemudian menentukan dan menampilkan nilai tertinggi dan terendah. Perhatikan flowchart berikut ini:

#### 2.1.1 Langkah-langkah Percobaan

1. Buat folder baru pada repositori lokal Anda, beri nama jobsheet7
2. Buat file baru, beri nama SiakadForNoPresensi.java
3. Buatlah struktur dasar program Java yang terdiri dari fungsi main().
4. Tambahkan library Scanner di bagian atas (luar) class
5. Buat deklarasi Scanner dengan nama variabel sc di dalam fungsi main()
6. Deklarasikan variabel nilai, **tertinggi**, dan **terendah** bertipe double. Inisialisasi **tertinggi** dengan 0 dan **terendah** dengan 100



```
1
2 double nilai, tertinggi = 0, terendah = 100;
```

7. Buat struktur perulangan FOR dengan batas kondisi sesuai jumlah mahasiswa yaitu 10
8. 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**



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

9. Di luar perulangan FOR, tampilkan nilai **tertinggi** dan **terendah**

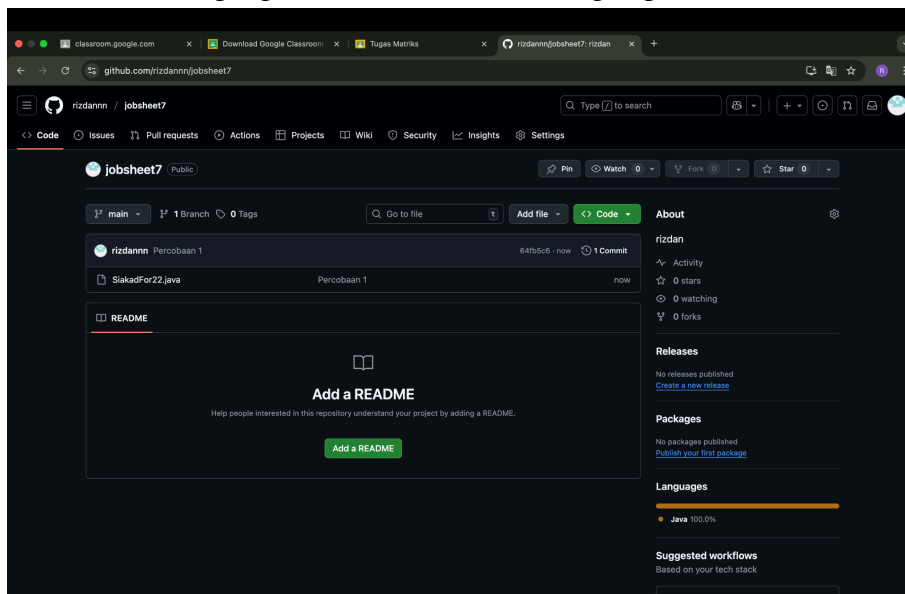


```
1 System.out.println("Nilai tertinggi: " + tertinggi);
2 System.out.println("Nilai terendah: " + terendah);
```

## 10. Compile dan run program

```
1  import java.util.Scanner;
2  public class SiakadFor22 {
3      public static void main(String[] args) {
4          Scanner sc = new Scanner (System.in);
5
6          double nilai, tertinggi = 0, terendah = 100;
7
8          for (int i = 1; i <= 10; i++) {
9              System.out.print("Masukkan nilai mahasiswa ke-" + i + ": ");
10             nilai = sc.nextDouble();
11             if (nilai > tertinggi) {
12                 tertinggi = nilai;
13             }
14             if (nilai < terendah) {
15                 terendah = nilai;
16             }
17         }
18
19         System.out.println("Nilai tertinggi: " + tertinggi);
20         System.out.println("Nilai terendah: " + terendah);
21     }
22 }
```

## 11. Commit program Anda ke Github dengan pesan “Percobaan 1”



### 2.1.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.

```
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: 64,9
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
```

### 2.1.3 Pertanyaan

1. Sebutkan dan tunjukkan masing-masing komponen perulangan FOR pada kode program Percobaan 1!
2. Mengapa variabel tertinggi diinisialisasi 0 dan terendah diinisialisasi 100? Apa yang terjadi jika variabel tertinggi diinisialisasi 100 dan terendah diinisialisasi 0?
3. Jelaskan fungsi dan alur kerja dari potongan kode berikut!
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!
5. Commit dan push hasil modifikasi Anda ke Github dengan pesan "Modifikasi Percobaan 1"

### Jawaban

1. `int i = 1;` : Menentukan nilai awal variabel penghitung sebelum perulangan dimulai.  
`i <= 10` : Mengecek apakah perulangan masih harus dijalankan. Jika bernilai **true**, blok kode dijalankan; jika **false**, perulangan berhenti.  
`i++` : Mengubah nilai variabel penghitung di setiap akhir perulangan.
2. Variabel tertinggi diinisialisasi 0 agar nilai berapa pun yang lebih besar dari 0 bisa langsung menggantikannya.  
Variabel terendah diinisialisasi 100 agar nilai berapa pun yang lebih kecil dari 100 bisa langsung menggantikannya.  
Jika dibalik (tertinggi = 100 dan terendah = 0), maka hasilnya salah, karena tidak ada nilai yang lebih besar dari 100 atau lebih kecil dari 0, sehingga variabel tidak akan berubah.
3. Kode tersebut berfungsi untuk mencari nilai tertinggi dan terendah  
Jika nilai lebih besar dari tertinggi, maka tertinggi diubah menjadi nilai.  
Jika nilai lebih kecil dari terendah, maka terendah diubah menjadi nilai.

4.

```
1 import java.util.Scanner;
2 public class SiakadFor22 {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner (System.in);
5
6         double nilai, tertinggi = 0, terendah = 100;
7         int lulus = 0, tidakLulus = 0;
8
9         for (int i = 1; i <= 10; i++) {
10             System.out.print("Masukkan nilai mahasiswa ke-" + i + ": ");
11             nilai = sc.nextDouble();
12
13             if (nilai > tertinggi) {
14                 tertinggi = nilai;
15             }
16
17             if (nilai < terendah) {
18                 terendah = nilai;
19             }
20             if (nilai > 60) {
21                 lulus++;
22             } else {
23                 tidakLulus++;
24             }
25         }
26
27         System.out.println("Nilai tertinggi: " + tertinggi);
28         System.out.println("Nilai terendah: " + terendah);
29         System.out.println("Keterangan siswa : " + lulus);
30         System.out.println("Keterangan siswa : " + tidakLulus);
31     }
32 }
```

```

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
Siswa yang lulus: 9
Siswa yang tidak lulus: 1
rizdann@MacBook-Air-rafif jobsheet7-1 %

```

5.

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

### Waktu Percobaan: 90 menit

Seorang dosen ingin memasukkan nilai beberapa mahasiswa ke dalam SIAKAD untuk ditentukan kategori nilai hurufnya. Program harus meminta dosen untuk memasukkan nilai setiap mahasiswa. Jika dosen memasukkan nilai yang tidak valid (negatif atau lebih dari 100), program harus mengabaikan input tersebut dan meminta dosen untuk melakukan input ulang. Selanjutnya, nilai yang valid dikelompokkan ke dalam kategori huruf A ( $80 < \text{nilai} \leq 100$ ), B+ ( $73 < \text{nilai} \leq 80$ ), B ( $65 < \text{nilai} \leq 73$ ), C+ ( $60 < \text{nilai} \leq 65$ ), C ( $50 < \text{nilai} \leq 60$ ), D ( $39 < \text{nilai} \leq 50$ ), dan E ( $\text{nilai} \leq 39$ ).

### 2.2.1 Langkah-langkah Percobaan

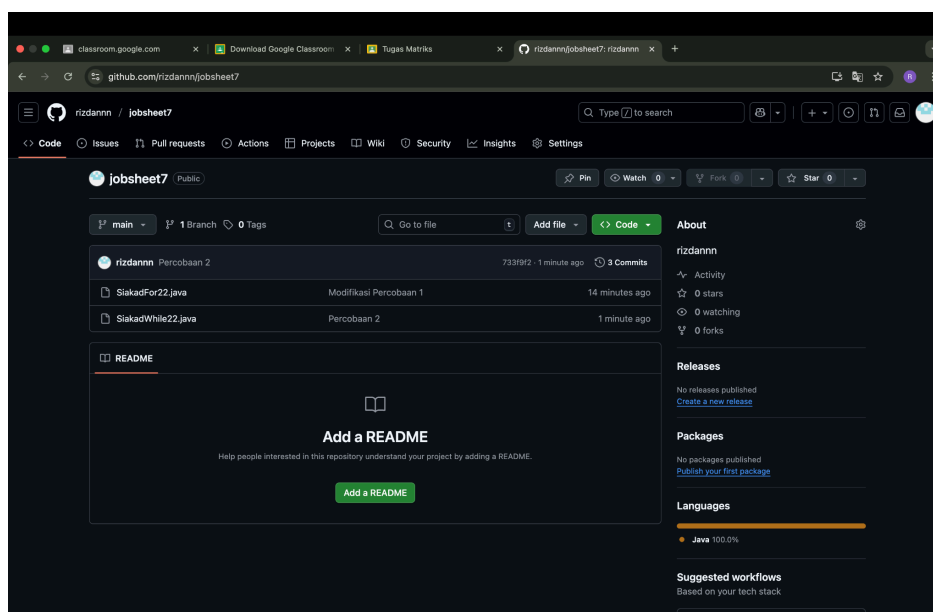
1. Buat file baru, beri nama SiakadWhileNoPresensi.java
2. Buatlah struktur dasar program Java yang terdiri dari fungsi main().
3. Tambahkan library Scanner di bagian atas (luar) class
4. Buat deklarasi Scanner dengan nama variabel sc di dalam fungsi main()
5. Deklarasikan variabel nilai, jml, dan i (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.



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.

```
1 import java.util.Scanner;
2 public class SiakadWhile22 {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         int nilai, jml, i = 0;
6
7         System.out.print("Masukkan jumlah mahasiswa: ");
8         jml = sc.nextInt();
9
10        while (i < jml) {
11            System.out.print("Masukkan nilai mahasiswa ke-" + (i + 1) + ": ");
12            nilai = sc.nextInt();
13
14            if (nilai < 0 || nilai > 100){
15                System.out.println("Nilai tidak valid. Masukkan lagi nilai yang valid!");
16                continue;
17            }
18
19            if (nilai > 80 && nilai <= 100){
20                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah A");
21            } else if (nilai > 73 && nilai <= 80){
22                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah B+");
23            } else if (nilai > 65 && nilai <= 73){
24                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah B");
25            } else if (nilai > 60 && nilai <= 65){
26                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah C+");
27            } else if (nilai > 50 && nilai <= 60){
28                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah C");
29            } else if (nilai > 39 && nilai <= 50){
30                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah D");
31            } else {
32                System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah E");
33            }
34            i++;
35        }
36    }
37 }
```

9. Compile dan run program
10. Commit program Anda ke Github dengan pesan “Percobaan 2”



### 2.2.2 Langkah-langkah Percobaan

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.

```
Masukkan jumlah mahasiswa: 5
Masukkan nilai mahasiswa ke-1: 85
Nilai mahasiswa ke-1 adalah A
Bagus, Pertahankan nilainya
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
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
rizdann@MacBook-Air-rafif jobsheet7-1 % □
```

### 2.2.3 Pertanyaan

1. Pada potongan kode berikut, tentukan maksud dan kegunaan dari sintaks berikut
2. Mengapa sintaks `i++` dituliskan di akhir perulangan `WHILE`? Apa yang terjadi jika posisinya dituliskan di awal perulangan `WHILE`?
3. Apabila jumlah mahasiswa yang dimasukkan adalah 19, berapa kali perulangan `WHILE` akan berjalan?
4. Modifikasi kode program sehingga apabila terdapat mahasiswa yang mendapat nilai A, program menampilkan pesan tambahan "Bagus, pertahankan nilainya"!
5. Commit dan push hasil modifikasi Anda ke Github dengan pesan "Modifikasi Percobaan 2"

### Jawaban

1. - Sintaks `nilai < 0 || nilai > 100` digunakan untuk memeriksa apakah nilai berada di luar rentang 0–100. Jika benar, berarti nilai tidak valid  
- Sintaks `continue` digunakan untuk melewati sisa perintah dalam satu iterasi dan langsung lanjut ke perulangan berikutnya.
2. - Sintaks `i++` dituliskan di akhir perulangan `WHILE` agar perulangan dijalankan dulu baru nilai `i` bertambah.  
- Jika `i++` ditulis di awal perulangan, maka nilai `i` langsung bertambah sebelum proses utama dijalankan, akibatnya jumlah perulangan berkurang satu.
3. Jika jumlah mahasiswa yang dimasukkan 19, maka perulangan `WHILE` akan berjalan sebanyak 19 kali, karena loop akan mengulang satu kali untuk setiap mahasiswa

```

1  import java.util.Scanner;
2  public class SiakadWhile22 {
3      public static void main(String[] args) {
4          Scanner sc = new Scanner(System.in);
5          int nilai, jml, i = 0;
6
7          System.out.print("Masukkan jumlah mahasiswa: ");
8          jml = sc.nextInt();
9
10         while (i < jml) {
11             System.out.print("Masukkan nilai mahasiswa ke-" + (i + 1) + ": ");
12             nilai = sc.nextInt();
13
14             if (nilai < 0 || nilai > 100){
15                 System.out.println("Nilai tidak valid. Masukkan lagi nilai yang valid!");
16                 continue;
17             }
18
19             if (nilai > 80 && nilai <= 100){
20                 System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah A");
21                 System.out.println("Bagus, pertahankan nilainya");
22             } else if (nilai > 73 && nilai <= 80){
23                 System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah B+");
24             } else if (nilai > 65 && nilai <= 73){
25                 System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah B");
26             } else if (nilai > 60 && nilai <= 65){
27                 System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah C+");
28             } else if (nilai > 50 && nilai <= 60){
29                 System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah C");
30             } else if (nilai > 39 && nilai <= 50){
31                 System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah D");
32             } else {
33                 System.out.println("Nilai mahasiswa ke-" + (i + 1) + " adalah E");
34             }
35             i++;

```

4.

5.

The screenshot shows a web browser with multiple tabs open, including 'classroom.google.com', 'Download Google Classroom', 'Tugas Matriks', and 'Commits - rizdannn/jobsheet7'. The active tab displays the GitHub repository page for 'rizdannn / jobsheet7'. The page header includes navigation links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The main section is titled 'Commits' and shows a list of recent commits on the 'main' branch, filtered for 'All users' and 'All time'. The commits are listed in descending order of time, with the most recent at the top.

| Commit Message         | Author   | Time Ago                 | Commit Hash |
|------------------------|----------|--------------------------|-------------|
| Modifikasi Percobaan 2 | rizdannn | committed now            | 6ee4ce0     |
| Percobaan 2            | rizdannn | committed 5 minutes ago  | 733f9f2     |
| Modifikasi Percobaan 1 | rizdannn | committed 19 minutes ago | 5791656     |
| Percobaan 1            | rizdannn | committed 23 minutes ago | 31d7758     |



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

### Waktu Percobaan: 60 menit

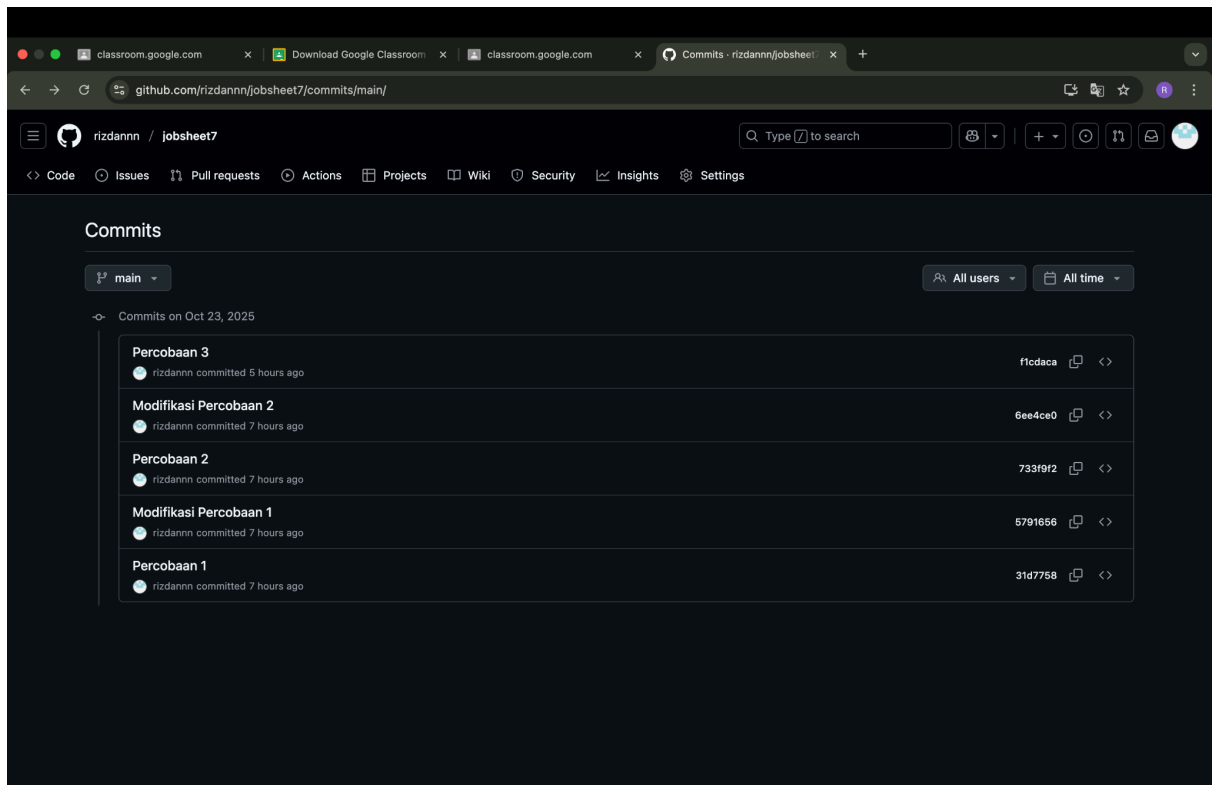
Di sebuah kafe, kasir ingin memproses transaksi beberapa pelanggan. Pelanggan dapat membeli lebih dari satu item (kopi dengan harga Rp 12.000, teh dengan harga Rp 7.000, dan roti dengan harga Rp 20.000), dan kasir akan terus memasukkan jumlah pembelian untuk setiap pelanggan. Jika ada pelanggan yang memutuskan untuk membatalkan transaksi (dengan memasukkan "batal"), maka kasir akan menghentikan input transaksi dan program berhenti.

#### 2.3.1 Langkah-langkah Percobaan

1. Buat file baru, beri nama KafeDoWhileNoPresensi.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 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.

```
1  do {
2      System.out.print("Masukkan nama pelanggan (ketik 'batal' untuk keluar): ");
3      namaPelanggan = sc.nextLine();
4      if (namaPelanggan.equalsIgnoreCase("batal")){
5          System.out.println("Transaksi dibatalkan");
6          break;
7      }
8      System.out.println("Jumlah kopi: ");
9      kopi = sc.nextInt();
10     System.out.println("Jumlah teh: ");
11     teh = sc.nextInt();
12     System.out.println("Jumlah roti: ");
13     roti = sc.nextInt();
14     int totalHarga = (kopi * hargaKopi) + (teh * hargaTeh) + (roti * hargaRoti);
15     System.out.println("Total yang harus dibayar: Rp " +totalHarga);
16     sc.nextLine();
17 } while (true);
```

8. Compile dan run program
9. Commit program Anda ke Github dengan pesan “Percobaan 3”



### 2.3.2 Langkah-langkah Percobaan

```
Masukkan nama pelanggan (ketik 'batal' untuk keluar): Rena
Jumlah kopi:
3
Jumlah teh:
0
Jumlah roti:
1
Total yang harus dibayar: Rp 56000
Masukkan nama pelanggan (ketik 'batal' untuk keluar): Yuni
Jumlah kopi:
1
Jumlah teh:
4
Jumlah roti:
2
Total yang harus dibayar: Rp 80000
Masukkan nama pelanggan (ketik 'batal' untuk keluar): BATAL
Transaksi dibatalkan
Semua transaksi selesai
rzdannn@MacBook-Air-rafif jobsheet7 %
```

### 2.3.3 Pertanyaan

1. Pada penggunaan DO-WHILE ini, apabila nama pelanggan yang dimasukkan pertama kali adalah “batal”, maka berapa kali perulangan dilakukan?
2. Sebutkan kondisi berhenti yang digunakan pada perulangan DO-WHILE tersebut!
3. Apa fungsi dari penggunaan nilai true pada kondisi DO-WHILE?
4. Mengapa perulangan DO-WHILE tersebut tetap berjalan meskipun tidak ada komponen inisialisasi dan update?

### Jawaban

1. Pada struktur do-while, perintah di dalam blok selalu dijalankan minimal satu kali, karena pengecekan kondisi dilakukan di akhir perulangan. Jadi, jika nama pelanggan pertama yang dimasukkan adalah "batal", perulangan tetap dijalankan satu kali sebelum berhenti.
2. Perulangan akan berhenti jika input = "batal"
3. Fungsi penggunaan true pada kondisi do-while adalah untuk membuat perulangan berjalan terus (tak terbatas)sampai ada perintah break di dalamnya.
4. Inisialisasi dan update bukan bagian wajib dari struktur DO-WHILE

## 3. Tugas

### 1. Waktu Percobaan : 120 Menit

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

## Commit dan push program Anda ke Github dengan pesan “Tugas 1”

```
1 import java.util.Scanner;
2
3 public class penjualanTiketBioskop22 {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         int HARGA_TIKET = 50000;
8         int totalTiketTerjual = 0;
9         double totalPenjualan = 0;
10        int i = 0;
11
12        System.out.println("=== Program Penjualan Tiket Bioskop ===");
13        System.out.println("Masukkan jumlah tiket yang dibeli oleh setiap pelanggan.");
14        System.out.println("Masukkan '0' untuk mengakhiri");
15
16        while (true) {
17            System.out.print("Jumlah tiket dibeli hari ke " + (i + 1) + "(ketik '0' untuk mengakhiri): ");
18            int jumlah = sc.nextInt();
19
20            if (jumlah == 0) {
21                break;
22            }
23
24            if (jumlah < 0) {
25                System.out.println("Jumlah tiket tidak boleh negatif! Coba lagi.\n");
26                continue;
27            }
28
29            double totalHarga = jumlah * HARGA_TIKET;
30            double diskon = 0;
31
32            if (jumlah > 10) {
33                diskon = 0.15 * totalHarga;
34            } else if (jumlah > 4) {
35                diskon = 0.10 * totalHarga;
36            }
37
38            double totalBayar = totalHarga - diskon;
39
40            System.out.println("Harga sebelum diskon : Rp." + (int) totalHarga);
41            System.out.println("Diskon : Rp." + (int) diskon);
42            System.out.println("Total bayar : Rp." + (int) totalBayar);
43
44            totalTiketTerjual += jumlah;
45            totalPenjualan += totalBayar;
46
47            i++;
48        }
49
50        System.out.println("\n=== Rangkuman Penjualan ===");
51        System.out.println("Total tiket terjual : " + totalTiketTerjual);
52        System.out.println("Total penjualan : Rp." + (int) totalPenjualan);
53    }
54 }
```

## 2. Perhatikan flowchart berikut!

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!

## Commit dan push program Anda ke Github dengan pesan “Tugas 2”

```
1 import java.util.Scanner;
2
3 public class parkirKendaraan22 {
4     public static void main(String[] args) {
5         Scanner input = new Scanner(System.in);
6         int jenis, durasi;
7         int total = 0;
8
9         do {
10             System.out.println("Masukkan jenis kendaraan (1 Mobil, 2 Motor, 0 Keluar): ");
11             jenis = input.nextInt();
12
13             if (jenis == 1 || jenis == 2) {
14                 System.out.print("Masukkan durasi parkir (jam): ");
15                 durasi = input.nextInt();
16
17                 if (durasi > 5) {
18                     total += 12500;
19                 } else if (jenis == 1) {
20                     total += durasi * 3000;
21                 } else if (jenis == 2) {
22                     total += durasi * 2000;
23                 }
24             }
25
26             } while (jenis != 0);
27
28             System.out.println("Total biaya parkir: Rp " + total);
29             input.close();
30         }
31     }
```

The screenshot shows a web browser displaying the GitHub repository page for 'rizzdennn/jobsheet7'. The repository is public and has 7 commits. The commit history table lists the following files and their commit messages:

| File                         | Commit Message         | Time Ago      |
|------------------------------|------------------------|---------------|
| KafeDoWhile22.java           | Percobaan 3            | 5 days ago    |
| SiakadFor22.java             | Modifikasi Percobaan 1 | 5 days ago    |
| SiakadWhile22.java           | Modifikasi Percobaan 2 | 5 days ago    |
| parkirKendaraan22.java       | Tugas 2                | 3 minutes ago |
| penjualanTiketBioskop22.java | Tugas 1                | 3 days ago    |

The repository also features a 'README' section with a prompt to 'Add a README' and a 'Languages' section showing 'Java' at 100.0%.