

JOB SHEET 7
PRAKTIKUM DASAR
PEMROGAMAN

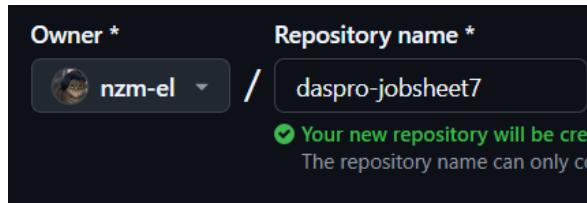


Nizam El Mullky Assalam
244107020041
D-IV TEKNIK INFORMATIKA

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG 2024

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

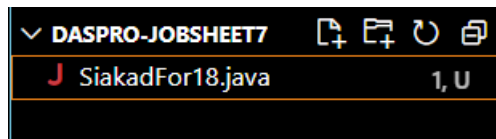
1. Buat repository baru pada akun Github Anda, beri nama daspro-jobsheet7



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

```
Lenovo@LAPTOP-1TKQ0FS5 MINGW64 ~/Downloads (master)
$ git clone https://github.com/nzm-el/daspro-jobsheet7.git
Cloning into 'daspro-jobsheet7'...
warning: You appear to have cloned an empty repository.
```

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().

```
import java.util.Scanner;
public class SiakadFor18 {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
```

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

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    double nilai, tertinggi = 0, terendah = 100;
```

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

```
Scanner sc = new Scanner(System.in);
double nilai, tertinggi = 0, terendah = 100;
for (int i = 1; i <= 10; i++) {
```

9. Buat struktur perulangan FOR dengan batas kondisi sesuai jumlah mahasiswa yaitu 100

```
double nilai, tertinggi = 0, terendah = 100;
for (int i = 1; i <= 10; i++) {
    System.out.print("Masukan nilai: ");
```

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

```
double nilai, tertinggi = 0, terendah = 100;
for (int i = 1; i <= 10; i++) {
    System.out.print("Masukan 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);
```

12. Compile dan run program

```
Lenovo@LAPTOP-1TKQ0F55 MINGW64 ~/Downloads/daspro-jobsheet7 (main)
$ git commit -m "Percobaan 1"
[main (root-commit) b18e25c] Percobaan 1
1 file changed, 19 insertions(+)
create mode 100644 SiakadFor18.java

Lenovo@LAPTOP-1TKQ0F55 MINGW64 ~/Downloads/daspro-jobsheet7 (main)
$ git push origin main
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 504 bytes | 168.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/nzm-el/daspro-jobsheet7.git
* [new branch]      main -> main
```

13. Commit dan push kode program ke Github



2.1.1 Verifikasi Hasil Percobaan

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

2.1.2 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!

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

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 kode program ke Github**

2.1.3 Jawaban Pertanyaan

1. Mm
2. Karena jika diinisialisasi tertinggi = 0 dan terendah = 100, maka outputnya Nilai tertinggi = 100 dan Nilai terendah = 0, karena pada perulangan FOR untuk mendapatkan nilai tertinggi menggunakan nilai \geq tertinggi karena tidak ada nilai yang lebih besar maka nilai yang diambil adalah nilai tertinggi = 100 begitupun sebaliknya dengan nilai terendah
3. Fungsinya adalah untuk mendeteksi nilai yang sudah dimasukan oleh user dan mengecek serta mengategorikannya pada nilai tertinggi dan nilai terendah secara berkala.

4. Maka hasilnya seperti ini :

```
J SiakadFor18.java 1, M X
J SiakadFor18.java > Language Support for Java(TM) by Red Hat > SiakadFor18 > main(String[])
1  import java.util.Scanner;
2  public class SiakadFor18 {
3      Run | Debug | Run main | Debug main
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
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("Masukan nilai mahasiswa ke-" + i + ": ");
11             nilai = sc.nextDouble();
12             if (nilai > tertinggi) {
13                 tertinggi = nilai;
14             }
15             if (nilai < terendah) {
16                 terendah = nilai;
17             }
18             if (nilai >= 60) {
19                 lulus++;
20             } else {
21                 tidakLulus++;
22             }
23         }
24
25         System.out.println("Nilai tertinggi: " + tertinggi);
26         System.out.println("Nilai terendah: " + terendah);
27         System.out.println("Jumlah mahasiswa lulus: " + lulus);
28         System.out.println("Jumlah mahasiswa tidak lulus: " + tidakLulus);
29     }
30 }
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
at SiakadFor18.main(SiakadFor18.java:10)
PS C:\Users\Lenovo\Downloads\daspro-jobsheet7> c:\cd 'c:\Users\Lenovo\Downloads\daspro-jobsheet7'; & 'C:\Program F
ode\workspaceStorage\bb9c0706f9e37c3a0e76d5dc1d769b87\redhat.java\jdt_ws\daspro-jobsheet7_2d2288a0\bin' 'Siakad
Masukan nilai mahasiswa ke-1: 76,5
Masukan nilai mahasiswa ke-2: 82,3
Masukan nilai mahasiswa ke-3: 62,1
Masukan nilai mahasiswa ke-4: 88,4
Masukan nilai mahasiswa ke-5: 65,9
Masukan nilai mahasiswa ke-6: 67,9
Masukan nilai mahasiswa ke-7: 90,1
Masukan nilai mahasiswa ke-8: 55,3
Masukan nilai mahasiswa ke-9: 73,7
Masukan nilai mahasiswa ke-10: 78,6
Nilai tertinggi: 90.1
Nilai terendah: 55.3
Jumlah mahasiswa lulus: 9
Jumlah mahasiswa tidak lulus: 1
```

5.

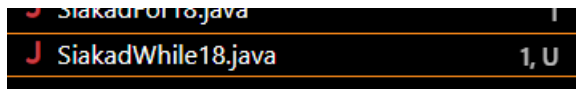
```
Lenovo@LAPTOP-1TKQ0F55 MINGW64 ~/Downloads/daspro-jobsheet7
$ git add .

Lenovo@LAPTOP-1TKQ0F55 MINGW64 ~/Downloads/daspro-jobsheet7
$ git commit -m "Percobaan 1 No 4"
[main 4e73407] Percobaan 1 No 4
1 file changed, 11 insertions(+)

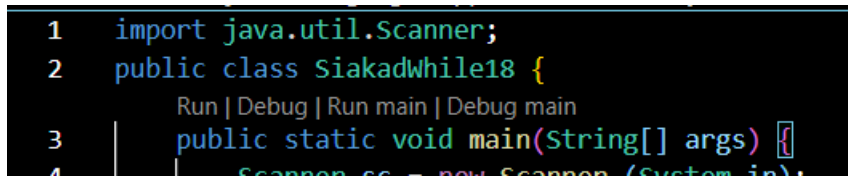
Lenovo@LAPTOP-1TKQ0F55 MINGW64 ~/Downloads/daspro-jobsheet7
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 407 bytes | 407.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from the cache)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/nzm-el/daspro-jobsheet7.git
   b18e25c..4e73407  main -> main
```

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

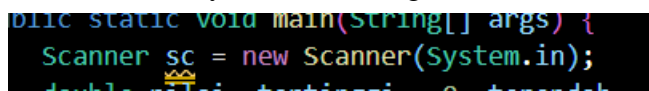
1. Buat file baru, beri nama SiakadWhileNoAbsen.java



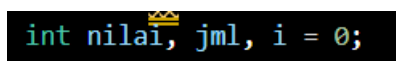
2. Buatlah struktur dasar program Java yang terdiri dari fungsi main().



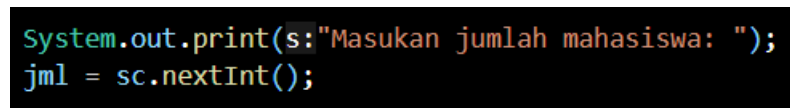
3. Tambahkan library Scanner di bagian atas (luar) class



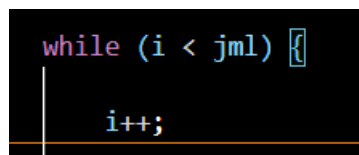
4. Buat deklarasi Scanner dengan nama variabel sc di dalam fungsi main()
5. Deklarasikan variabel nilai, jml, dan i (untuk perulangan) bertipe integer. Inisialisasi i dengan 0 sebagai nilai awal perulangan



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.



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



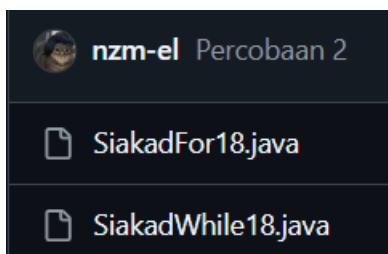
8. Di dalam perulangan WHILE tersebut, tambahkan perintah untuk memasukkan nilai mahasiswa. Setelah itu, buat kondisi pemilihan IF untuk mengecek valid atau tidaknya nilai yang dimasukkan, dengan syarat nilai harus berada pada rentang 0 hingga 100. Kemudian tambahkan kondisi pemilihan IF-ELSE IF-ELSE untuk menampilkan kategori nilai huruf berdasarkan ketentuan.

```
while (i < jml) {  
    System.out.print("Masukan nilai mahasiswa ke-" + (i + 1) + ": ");  
    nilai = sc.nextInt();  
  
    if (nilai < 0 || nilai > 100) {  
        System.out.println(x:"Nilai tidak valid. Masukan 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 E");  
    }  
    i++;  
}
```

9. Compile dan run program

```
Lenovo@LAPTOP-1TKQ0F55 MINGW64 ~/Downloads/daspro-jobsheet7 (ma  
$ git add .  
  
Lenovo@LAPTOP-1TKQ0F55 MINGW64 ~/Downloads/daspro-jobsheet7 (ma  
$ git commit -m "Percobaan 2"  
[main 29c0a61] Percobaan 2  
1 file changed, 37 insertions(+)  
create mode 100644 SiakadWhile18.java  
  
Lenovo@LAPTOP-1TKQ0F55 MINGW64 ~/Downloads/daspro-jobsheet7 (ma  
$ git push origin main  
Enumerating objects: 4, done.  
Counting objects: 100% (4/4), done.  
Delta compression using up to 12 threads  
Compressing objects: 100% (3/3), done.  
Writing objects: 100% (3/3), 673 bytes | 336.00 KiB/s, done.  
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
To https://github.com/nzm-el/daspro-jobsheet7.git  
4e73407..29c0a61 main -> main
```

10. Commit dan push kode program ke Github



2.2.1 Verifikasi Hasil Percobaan

```

Masukan jumlah mahasiswa: 5
Masukan nilai mahasiswa ke-1: 85
Nilai mahasiswa ke-1 adalah A
Masukan nilai mahasiswa ke-2: 63
Nilai mahasiswa ke-2 adalah C+
Masukan nilai mahasiswa ke-3: 101
Nilai tidak valid. Masukan lagi nilai yang valid!
Masukan nilai mahasiswa ke-3: 23
Nilai mahasiswa ke-3 adalah E
Masukan nilai mahasiswa ke-4: -15
Nilai tidak valid. Masukan lagi nilai yang valid!
Masukan nilai mahasiswa ke-4: 70
Nilai mahasiswa ke-4 adalah B
Masukan nilai mahasiswa ke-5: 55
Nilai mahasiswa ke-5 adalah C

```

2.2.2 Pertanyaan

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;  
}
```

- nilai < 0 || nilai > 100**
 - continue**
- Mengapa sintaks **i++** dituliskan di akhir perulangan **WHILE**? Apa yang terjadi jika posisinya dituliskan di awal perulangan **WHILE**?
- Apabila jumlah mahasiswa yang dimasukkan adalah 19, berapa kali perulangan **WHILE** akan berjalan?
- Modifikasi kode program sehingga apabila terdapat mahasiswa yang mendapat nilai A, program menampilkan pesan tambahan "Bagus, pertahankan nilainya"!
- Commit dan push kode program ke Github**

2.2.3 Jawaban Pertanyaan

1.
 - a) Kode tersebut dibaca apabila nilai < 0 atau > 100 , kode tersebut berguna untuk membaca nilai yang dimasukan oleh user jika nilainya < 0 atau > 100 maka dia memberikan output "Nilai tidak valid. Masukan lagi nilai yang valid!"
 - b) Continue digunakan untuk mengulang kembali perintah diatasnya dengan nilai i yang sama sampai memenuhi persyaratan programnya
2. Sintaks `i++` dituliskan diakhir karena dia akan menambahkan data i setelah program diatasnya sudah selesai dijalankan jika program diatasnya belum selesai atau belum terpenuhi maka nilai i tidak bertambah, jika sintaks `i++` dituliskan diawal maka nilai i menjadi 2 karena program sebelumnya sudah membaca `i = 1` jadi jika sintaks `i++` dituliskan diawal maka outputnya tidak dimulai dari 1
3. Akan berjalan sebanyak 19 kali

4. Maka hasilnya seperti ini

```
SiakadWhile18.java > Language Support for Java(TM) by Red Hat > SiakadWhile18 > main(String[])
2 public class SiakadWhile18 {
3     public static void main(String[] args) {
4
5
6         System.out.print(s:"Masukan jumlah mahasiswa: ");
7         jml = sc.nextInt();
8
9         while (i < jml) {
10             System.out.print("Masukan nilai mahasiswa ke-" + (i + 1) + ": ");
11             nilai = sc.nextInt();
12
13             if (nilai < 0 || nilai > 100) {
14                 System.out.println(x:"Nilai tidak valid. Masukan lagi nilai yang valid!");
15                 continue;
16             }
17
18             if (nilai > 80 && nilai <= 100) {
19                 System.out.println("Nilai mahasiswa ke-" + (i+1) + " adalah A");
20                 System.out.println(x:"Bagus, pertahankan nilainya!");
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 }
```

PROBLEMS (2) OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

```
Masukan jumlah mahasiswa: 5
Masukan nilai mahasiswa ke-1: 85
Nilai mahasiswa ke-1 adalah A
Bagus, pertahankan nilainya!
Masukan nilai mahasiswa ke-2: 63
Nilai mahasiswa ke-2 adalah C+
Masukan nilai mahasiswa ke-3: 101
Nilai tidak valid. Masukan lagi nilai yang valid!
Masukan nilai mahasiswa ke-3: 23
Nilai mahasiswa ke-3 adalah E
Masukan nilai mahasiswa ke-4: -15
Nilai tidak valid. Masukan lagi nilai yang valid!
Masukan nilai mahasiswa ke-4: 70
Nilai mahasiswa ke-4 adalah B
Masukan nilai mahasiswa ke-5: 55
Nilai mahasiswa ke-5 adalah C
PS C:\Users\Lenovo\Downloads\daspro-jobsheet7>
```

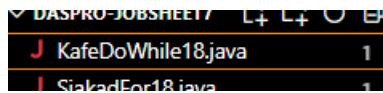
5.

```
Lenovo@LAPTOP-1TKQ0F55 MINGW64 ~/Downloads/daspro-jobsheet7 (n
$ git add .
Lenovo@LAPTOP-1TKQ0F55 MINGW64 ~/Downloads/daspro-jobsheet7 (n
$ git commit -m "Percobaan 2 No 4"
[main 9415eb4] Percobaan 2 No 4
1 file changed, 2 insertions(+), 1 deletion(-)

Lenovo@LAPTOP-1TKQ0F55 MINGW64 ~/Downloads/daspro-jobsheet7 (n
$ git push origin main
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 327 bytes | 327.00 KiB/s, done.
Total 3 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local c
To https://github.com/nzm-el/daspro-jobsheet7.git
29c0a61..9415eb4 main -> main
```

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

1. Buat file baru, beri nama KafeDoWhileNoAbsen.java



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

```
import java.util.Scanner;
public class KafeDoWhile18 {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner sc = new Scanner (System.in);
```

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

```
Scanner sc = new Scanner (System.in);
int kopi, teh, roti;
```

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.

```
Scanner sc = new Scanner (System.in);
int kopi, teh, roti;
String namaPelanggan;
double 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:"Masukan nama pelanggan (ketik 'batal' untuk
    keluar): ");
    namaPelanggan = sc.nextLine();
    if (namaPelanggan.equalsIgnoreCase(anotherString:"batal")) {
        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();

    double 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

```
Lenovo@LAPTOP-1TKQ0F55 MINGW64 ~/Downloads/daspro-jobsheet7 (mai
$ git add .

Lenovo@LAPTOP-1TKQ0F55 MINGW64 ~/Downloads/daspro-jobsheet7 (mai
$ git commit -m "Percobaan 3"
[main f1d93d8] Percobaan 3
1 file changed, 28 insertions(+)
create mode 100644 KafeDoWhile18.java

Lenovo@LAPTOP-1TKQ0F55 MINGW64 ~/Downloads/daspro-jobsheet7 (mai
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 742 bytes | 742.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/nzm-el/daspro-jobsheet7.git
9415eb4..f1d93d8 main -> main
```

9. Commit dan push kode program ke Github



2.3.1 Verifikasi Hasil Percobaan

```
ta\Roaming\Code\User\workspacestorage\bb9c0706f9e37c3a0e76d5dc
Masukan nama pelanggan (ketik 'batal' untuk keluar): Rena
Jumlah kopi: 3
Jumlah teh: 0
Jumlah roti: 1
Total yang harus dibayar: Rp 56000.0
Masukan nama pelanggan (ketik 'batal' untuk keluar): Yuni
Jumlah kopi: 1
Jumlah teh: 4
Jumlah roti: 2
Total yang harus dibayar: Rp 80000.0
Masukan nama pelanggan (ketik 'batal' untuk keluar): BATAL
Semua transaksi selesai.
```

2.3.2 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?

2.3.3 Jawaban Pertanyaan

1. Perulangan dilakukan satu kali
2. Perulangan dihentikan menggunakan pernyataan 'break'
3. Berfungsi untuk membuat perulangan menjadi perulangan tak terbatas (infinite loop)
4. Karena DO-WHILE tidak memerlukan komponen inisiasi dan update, DO-WHILE akan tetap berjalan meskipun tidak ada komponen inisiasi dan update.

2.4 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.

```
Bioskop18.java > Language Support for Java(TM) by Red Hat > Bioskop18
1  import java.util.Scanner;
2  public class Bioskop18 {
3      Run | Debug | Run main | Debug main
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6          int hargaTiket = 50000;
7          int totalTiketTerjual = 0;
8          double totalPenjualan = 0.0;
9
10         while (true) {
11             System.out.print(s:"Masukan jumlah tiket yang dibeli (atau ketik '0' untuk keluar): ");
12             int totalTiket = sc.nextInt();
13             if (totalTiket == 0) {
14                 break;
15             }
16             if (totalTiket < 0) {
17                 System.out.println(x:"Jumlah tiket tidak valid. Silahkan masukkan jumlah yang benar");
18                 continue;
19             }
20             double hargaTotal = totalTiket * hargaTiket;
21             double diskon = 0.0;
22
23             if (totalTiket > 10) {
24                 diskon = 0.15;
25             } else if (totalTiket > 4) {
26                 diskon = 0.10;
27             }
28             double hargaTotalDisc = hargaTotal * (1 - diskon);
29             totalTiketTerjual += totalTiket;
30             totalPenjualan += hargaTotalDisc;
31             System.out.println("Harga total untuk " + totalTiket + " tiket adalah: Rp " + hargaTotalDisc);
32         }
33         System.out.println(x:"=====");
34         System.out.println("Total tiket terjual : " + totalTiketTerjual);
35         System.out.println("Total penjualan tiket: Rp " + totalPenjualan);
36     }
37 }
```

```
Masukan jumlah tiket yang dibeli (atau ketik '0' untuk keluar): 6
Harga total untuk 6 tiket adalah: Rp 270000.0
Masukan jumlah tiket yang dibeli (atau ketik '0' untuk keluar): 0
=====
Total tiket terjual : 6
Total penjualan tiket: Rp 270000.0
PS C:\Users\Lenovo\Downloads\daspro-jobsheet7>
```

- 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!

```

import java.util.Scanner;

public class Parkir18 {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int jenis, durasi;
        int total = 0;

        while (true) {
            System.out.print(s:"Masukan jenis kendaraan (1 Mobil, 2 Motor, 0 keluar): ");
            jenis = sc.nextInt();
            if (jenis == 0) {
                break;
            } else if (jenis == 1 || jenis == 2) {
                System.out.print(s:"Masukan durasi parkir dalam jam: ");
                durasi = sc.nextInt();
                if (durasi > 5) {
                    total = 12500;
                } else {
                    if (jenis == 1) {
                        total = durasi * 3000;
                    } else if (jenis == 2) {
                        total = durasi * 2000;
                    }
                }
                System.out.println("Total yang harus dibayar: Rp " + total);
            } else {
                System.out.println(x:"Jenis kendaraan tidak valid. Silahkan masukkan 1 untuk Mobil atau 2 untuk Motor.");
            }
        }
    }
}

```

```
Masukan jenis kendaraan (1 Mobil, 2 Motor, 0 keluar): 1
Masukan durasi parkir dalam jam: 4
Total yang harus dibayar: Rp 12000
Masukan jenis kendaraan (1 Mobil, 2 Motor, 0 keluar): 2
Masukan durasi parkir dalam jam: 5
Total yang harus dibayar: Rp 10000
Masukan jenis kendaraan (1 Mobil, 2 Motor, 0 keluar): 2
Masukan durasi parkir dalam jam: 6
Total yang harus dibayar: Rp 12500
Masukan jenis kendaraan (1 Mobil, 2 Motor, 0 keluar): 0
```