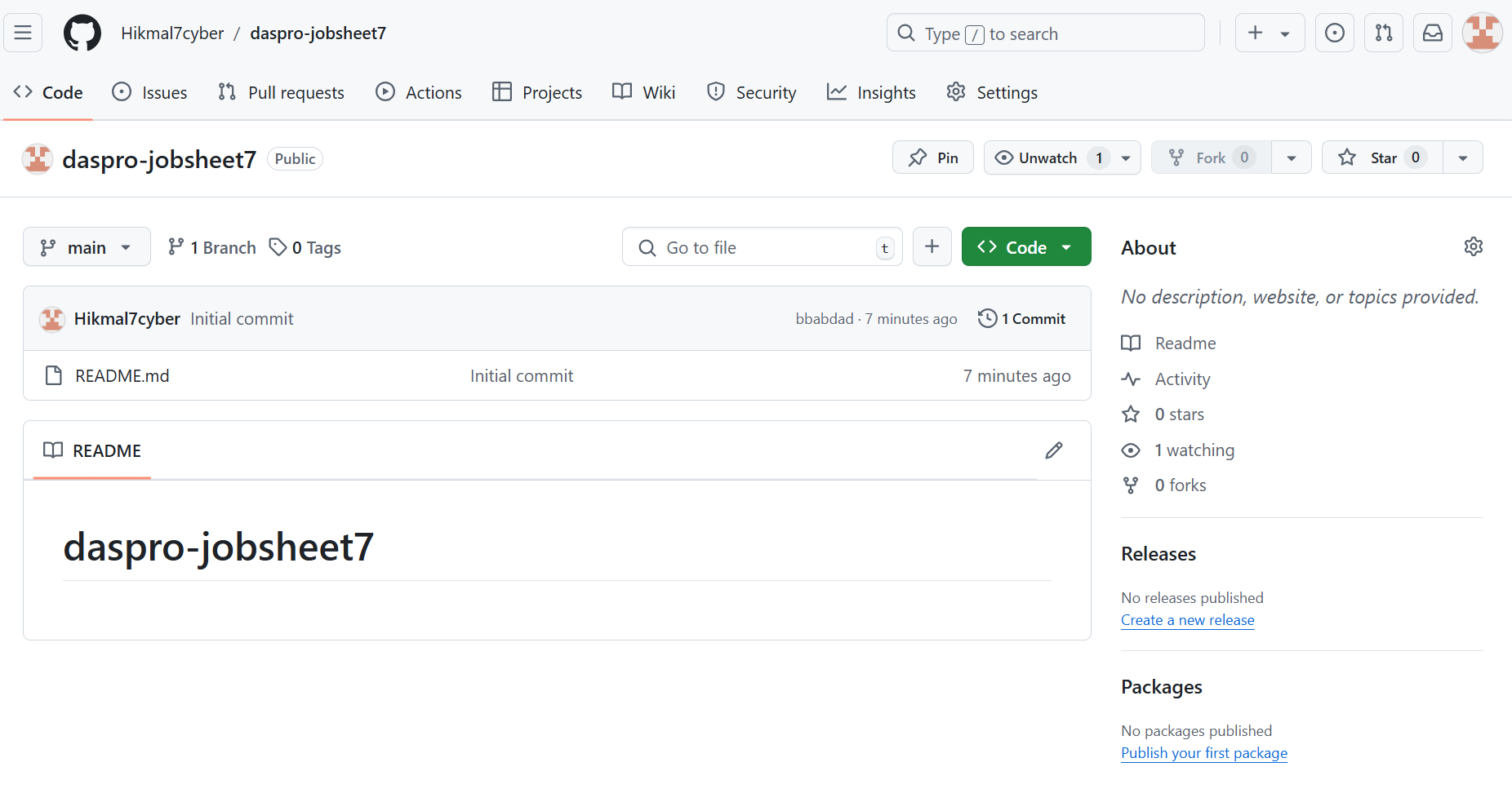
Nama : Muhammad Hikmal Al-Ghifary

Prodi : D – IV Teknik Informatika (1B)

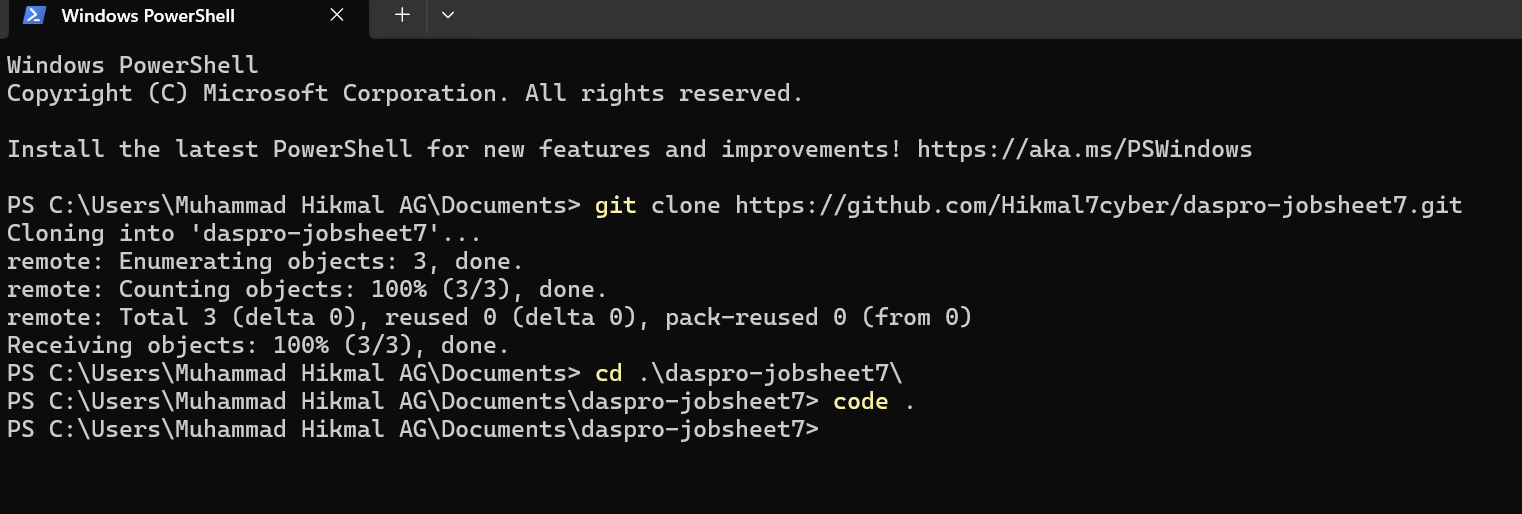
Matkul : Praktikum Dasar Pemrograman

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

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



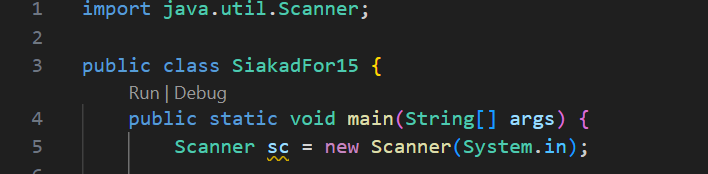
1. Lakukan cloning repository tersebut menggunakan perintah git clone dari terminal. Buka folder repository tersebut menggunakan Visual Studio Code



1. Buat file baru, beri nama SiakadForNoAbsen.java



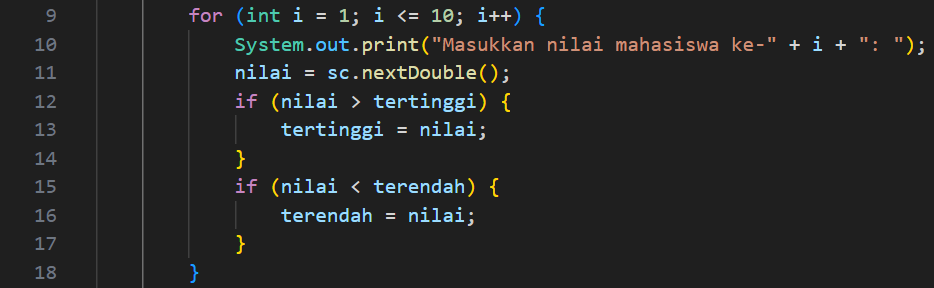
1. Buatlah struktur dasar program Java yang terdiri dari fungsi main(). Tambahkan library Scanner di bagian atas (luar) class. Buat deklarasi Scanner dengan nama variabel sc di dalam fungsi main()



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



1. Buat struktur perulangan FOR dengan batas kondisi sesuai jumlah mahasiswa yaitu 10.
2. 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

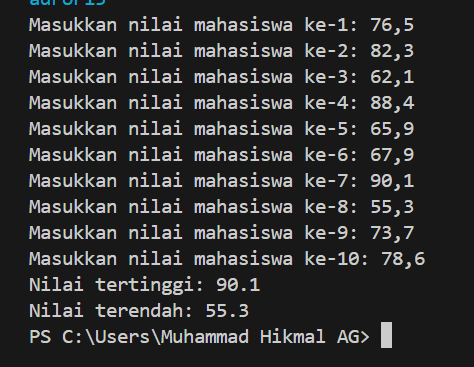


1. Di luar perulangan FOR, tampilkan nilai tertinggi dan terendah



1. Compile dan run program kemudian Commit dan push kode program ke Github

**HASIL** **COMPILE PROGRAM BERDASARKAN CONTOH**



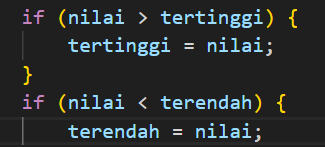
PERTANYAAN

1. Komponen perulangan FOR pada kode program Percobaan 1

Struktur FOR secara umum yakni => *for (inisialisasi; kondisi; update)*



1. Inisialisasi : *int i* = 1. Ini berarti dalam perulangan tersebut variabel i dimulai dari angka 1
2. Kondisi : *i<= 10*. Ini berarti perulangan for tersebut menerima perulangan sampai dengan 10 kali perulangan dan akan berhenti jika telah mencapai perulangan ke 10 (terakhir).
3. Update : *i ++*. Ini berarti setelah iterasi selesai, maka nilai i akan ditambah 1 secara otomatis untuk lanjut ke perulangan tahap berikutnya.
4. Variabel tertinggi diinisialisasi 0 dan terendah diinisialisasi 100 karena variabel tersebut digunakan sebagai nilai pembanding.



Jika variabel nilai dimasukkan, maka akan dilakukan perbandingan dengan variabel tertinggi maupun terendah. Setiap perbandingan dilakukan, akan ada update untuk setiap perulangannya. Misal :

*Nilai = 90*

*Tertinggi = 0*

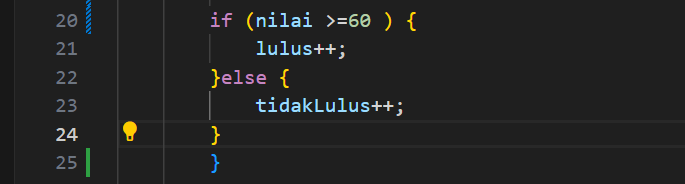
*Terendah = 100*

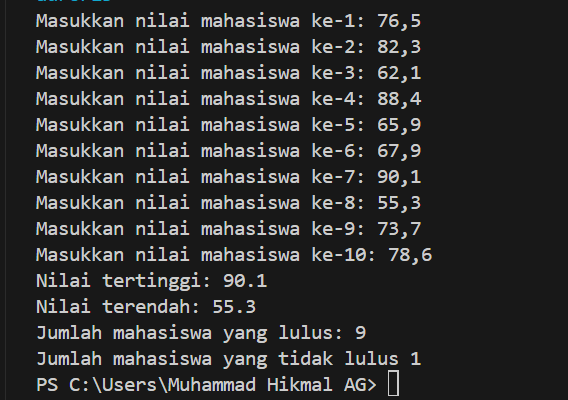
*Maka : 90 > 0 => akan dilakukan update untuk nilai tertinggi menjadi 90, dan seterusnya.*

*90< 0 => akan akan dilakukan update untuk nilai terendah menjadi 90, dan seterusnya.*

Yang akan terjadi jika dibalik jika nilai tertinggi diset adalah nilai tertinggi akan selalu 100, sampai ada nilai yang lebih besar dari 100 yang dimasukkan barulah nilai tertinggi akan berubah. Sama halnya dengan set nilai terendah = 0, maka nilai terendah akan terus 0 dan akan berubah jika diinput nilai yang lebih kecil dari 0.

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



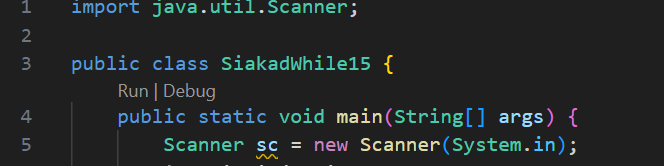


**Percobaan 2: Studi Kasus Nilai Mahasiswa di SIAKAD – Perulangan WHILE**

1. Buat file baru, beri nama SiakadWhileNoAbsen.java



1. Buatlah struktur dasar program Java yang terdiri dari fungsi main(). Tambahkan library Scanner di bagian atas (luar) class. Buat deklarasi Scanner dengan nama variabel sc di dalam fungsi main()



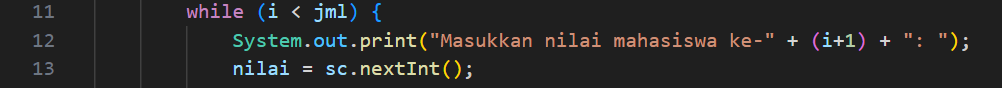
1. Deklarasikan variabel nilai, jml, dan i (untuk perulangan) bertipe integer. Inisialisasi i dengan 0 sebagai nilai awal perulangan



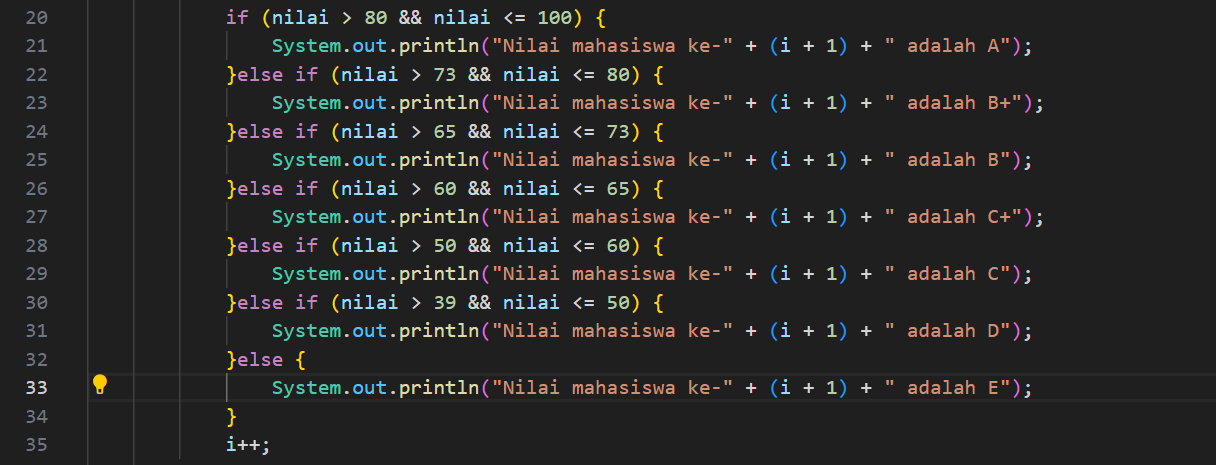
1. Tuliskan kode program untuk menerima input banyaknya mahasiswa yang disimpan ke variabel jml. Dengan demikian, batas perulangan akan dinamis sesuai masukan dari pengguna melalui keyboard.



1. 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



1. 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.

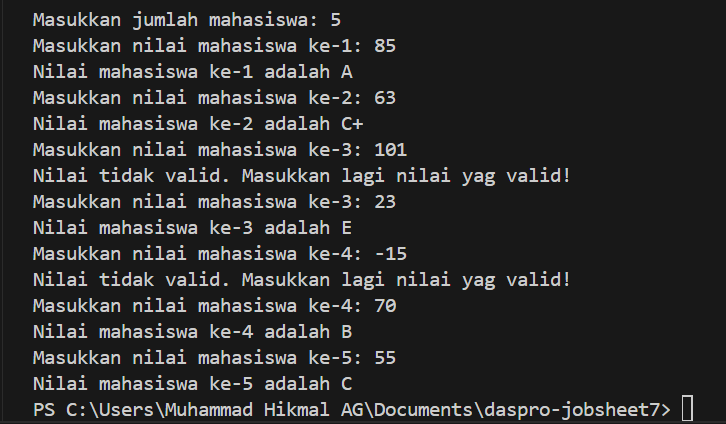


1. Compile dan run program, kemudian Commit dan push kode program ke Github

**HASIL COMPILE PROGRAM BERDASARKAN CONTOH**

Jumlah mahasiswa diinput = 5

Tidak valid = 2

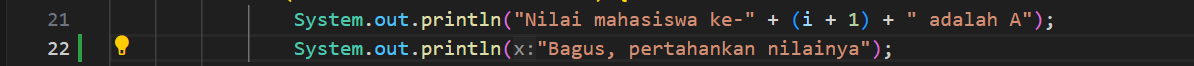


**PERTANYAAN**

1. Kegunaan dari sintaks
2. **nilai < 0 || nilai > 100** adalah untuk membatasi rentang nilai yang dapat diinput kedalam program tersebut. Artinya, pengguna tidak bisa memasukkan angka yang lebih kecil dari 0 (negatif; nilai desimal [0,1]) dan angka yang lebih besar dari 100. Dapat disimpulkan bahwa angka yang dapat diinput berada pada rentang 0 – 100.
3. **Continue** berguna untuk melanjutkan atau meloncati satu langkah perulangan. Sintaks ini dimasukkan untuk melanjutkan ke tahap berikutnya apabila kondisi di perulangan sebelumnya telah dijalankan dan terpenuhi
4. Sintaks **i++** dituliskan di akhir perulangan WHILE untuk memperbaharui nilai **i** (perulangan) setelah setiap iterasi. Hal ini dlakukan agar tidak terjadi kesalahan dan kondisi perulangan dapat dievaluasi dengan benar di iterasi berikutnya.

Apabila **i++** diletakkan di awal, maka nilai **i** akan diperbarui sebelum kondisi while dievaluasi.

1. Apabila jumlah mahasiswa yang dimasukkan adalah 19, maka perulangan WHILE akan berjalan sebanyak 19 kali pula. Hal ini terjadi karena dalam program tersebut variabel i dideklarasikan sebagai i = 0 yang berarti perhitungan dimulai dari 0. Berbeda halnya jika deklarasi berupa i = 2, maka perulangan akan terjadi sebanyak 17 kali karena mulai menghitung dari 2.
2. Modifikasi kode program sehingga apabila terdapat mahasiswa yang mendapat nilai A, program menampilkan pesan tambahan "Bagus, pertahankan nilainya"

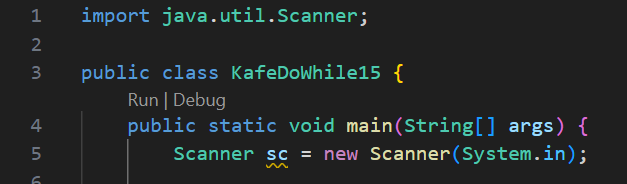


**Percobaan 3: Studi Kasus Transaksi di Kafe – Perulangan DO-WHILE**

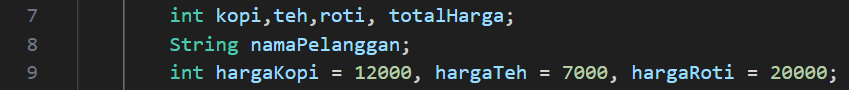
1. Buat file baru, beri nama KafeDoWhileNoAbsen.java



1. Buatlah struktur dasar program Java yang terdiri dari fungsi main(). Tambahkan library Scanner di bagian atas (luar) class. Buat deklarasi Scanner dengan nama variabel sc di dalam fungsi main()

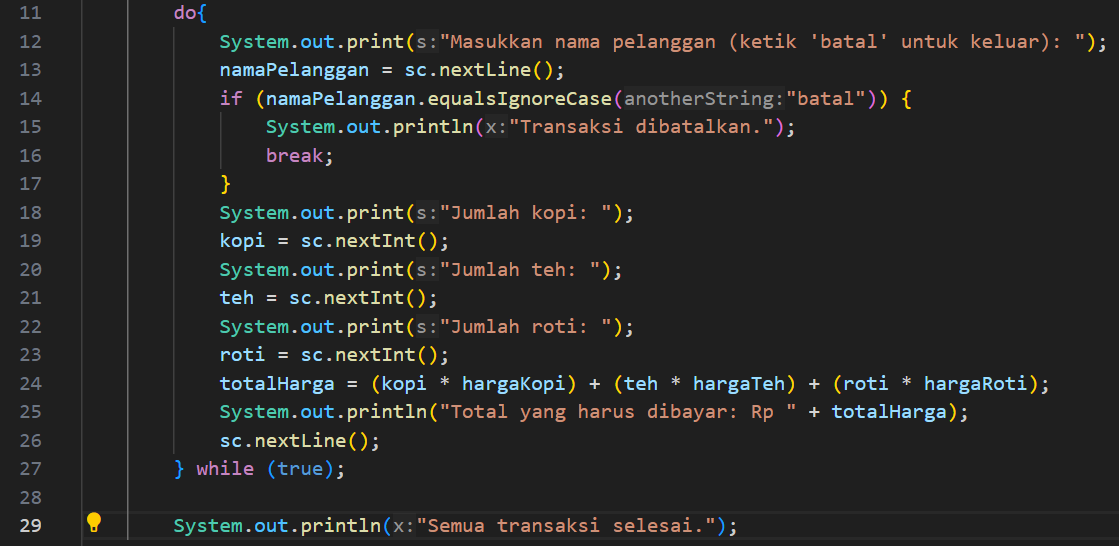


1. Deklarasikan variabel kopi, teh, dan roti bertipe integer untuk menampung banyaknya item yang dibeli pelanggan, serta namaPelanggan bertipe String. Deklarasi dan inisialisasi hargaKopi dengan 12000, hargaTeh dengan 7000, hargaRoti dengan 20000.



1. 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.



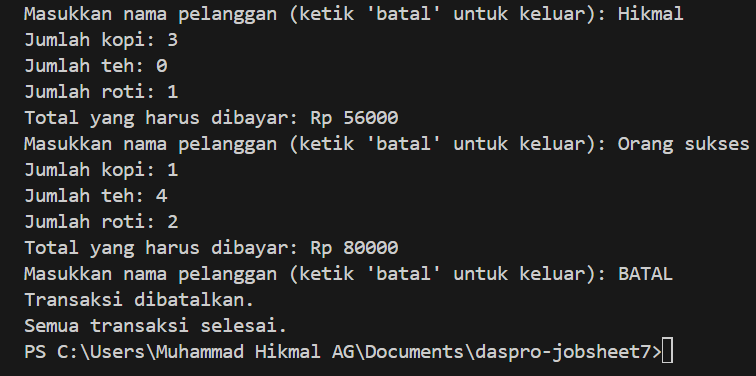
*Keterangan: sc.nextLine(); setelah print totalHarga merupakan sintaks untuk membersihkan newline dari buffer*

1. Compile dan run program, kemudian Commit dan push kode program ke Github

**HASIL COMPILE PROGRAM BERDASARKAN CONTOH**

Input berhasil = 2

Input end = batal

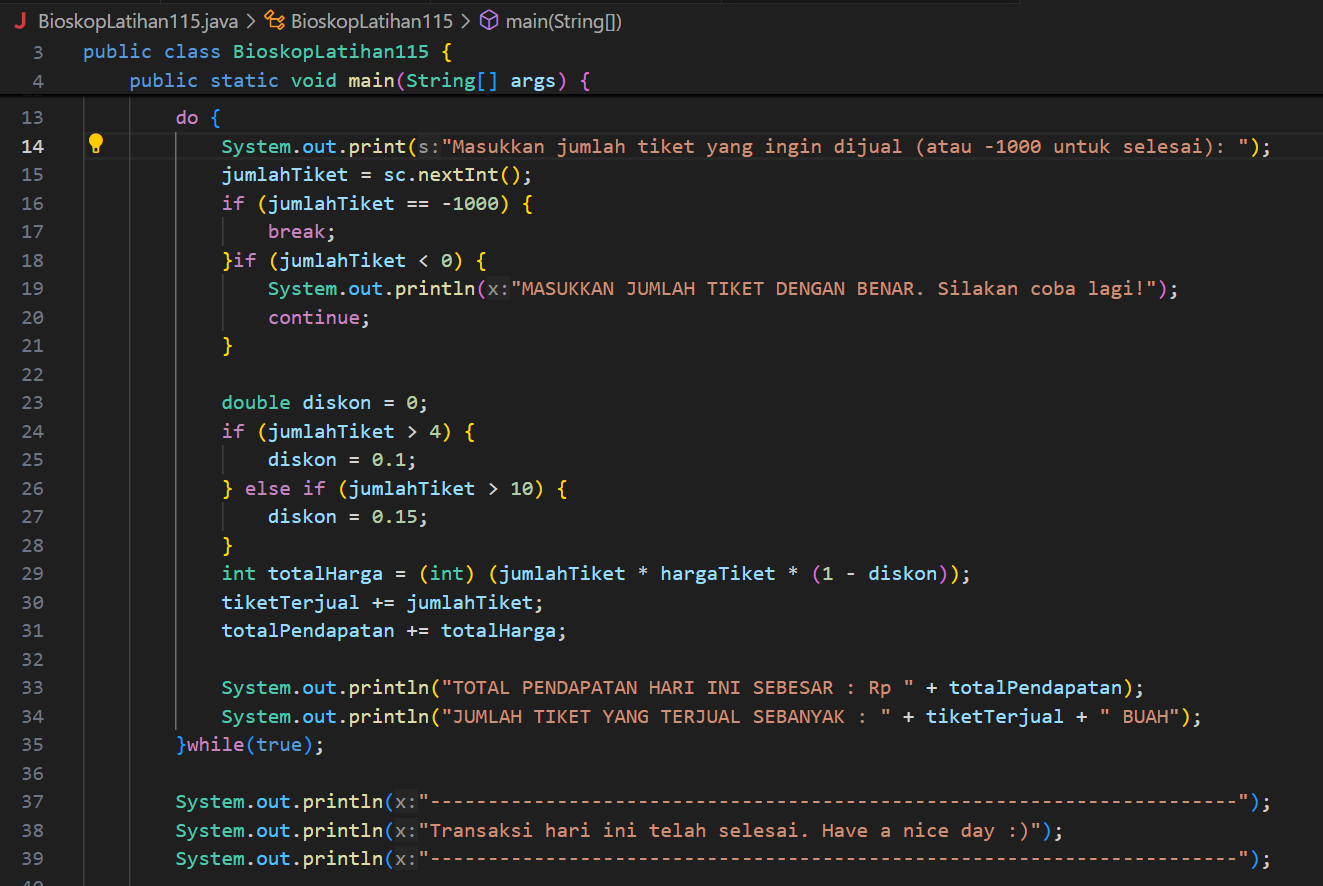


**PERTANYAAN**

1. Pada penggunaan DO-WHILE ini, apabila nama pelanggan yang dimasukkan pertama kali adalah “batal”, maka perulangan akan dilakukan sebanyak 1 kali saja. Berdasarkan teori, penggunaan DO-WHILE akan tetap berjalan minimal 1 kali walaupun dari awal kondisinya salah. Dalam kasus ini, kondisi input awal ‘batal’ bernilai false yang artinya kode tersebut akan terus berlanjut hingga bertemu while (true). Karena bernilai salah, maka tidak ada perulangan tahap 2.
2. Kondisi berhenti yang digunakan pada perulangan DO-WHILE tersebut ditandai dengan menginput ‘batal’ pada input pengguna dan adanya sintaks break pada program. Sintaks ini berfungsi untuk membatasi proses yang apabila sudah ditemukan kondisi yang benar, proses akan berhenti sampai di posisi itu saja dan mengabaikan proses dibawahnya. Input ‘batal’ bernilai false, kemudian ditahan oleh sintaks “*break*” melewati proses dibawahnya dan langsung masuk ke **while(true)**. Karena bernilai false, maka tidak ada perulangan ke 2
3. Fungsi dari penggunaan nilai true pada kondisi DO-WHILE adalah untuk mengecek apakah perlu melakukan pengulangan atau tidak. Jika proses diatasnya bernilai :
4. True : maka aka nada perulangan ke tahap berikutnya
5. False : maka perulangan tersebut merupakan perulangan terakhir dan tidak aka nada perulangan tahap berikutnya.
6. Perulangan DO-WHILE tersebut tetap berjalan meskipun tidak ada komponen inisialisasi dan update karena dalam perulangan ini, setiap kondisi baik bernilai TRUE ataupun FALSE akan tetap dilakukan proses pengecekan. Proses pengecekan akan terus berlangsung hingga n kali apabila terus ditemukan nilai TRUE. Pengecekan akan langsung berhenti jika ditemukan nilai FALSE

**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:
2. Jika pelanggan membeli lebih dari 4 tiket, pelanggan mendapatkan diskon 10%.
3. Jika pelanggan membeli lebih dari 10 tiket, pelanggan mendapatkan diskon 15%.
4. Jika input jumlah tiket tidak valid (negatif), program akan mengabaikan input tersebut dan meminta input ulang.



*Keterangan : pemilihan angka “-1000” untuk mengakhiri perhitungan adalah agar terhindar dari kesalahan. Karena diasumsikan tidak mungkin orang salah mengetik hingga -1000.*

Program tersebut menggunakan pemilihan **DO-WHILE**

1. 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!

