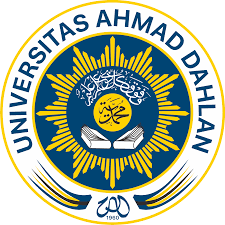
**PORTOFOLIO**

**DASAR SISTEM KOMPUTER**

**“PROGRAM ASSEMBLY PESAN MINUMAN DI CAFE”**

****

**DISUSUN OLEH:**

**NAMA : Dava Trista Argyatama**

**NIM : 24000104**

**KELAS : C**

**PROGRAM STUDI S1 INFORMATIKA**

**FAKULTAS TEKNOLOGI INDUSTRI**

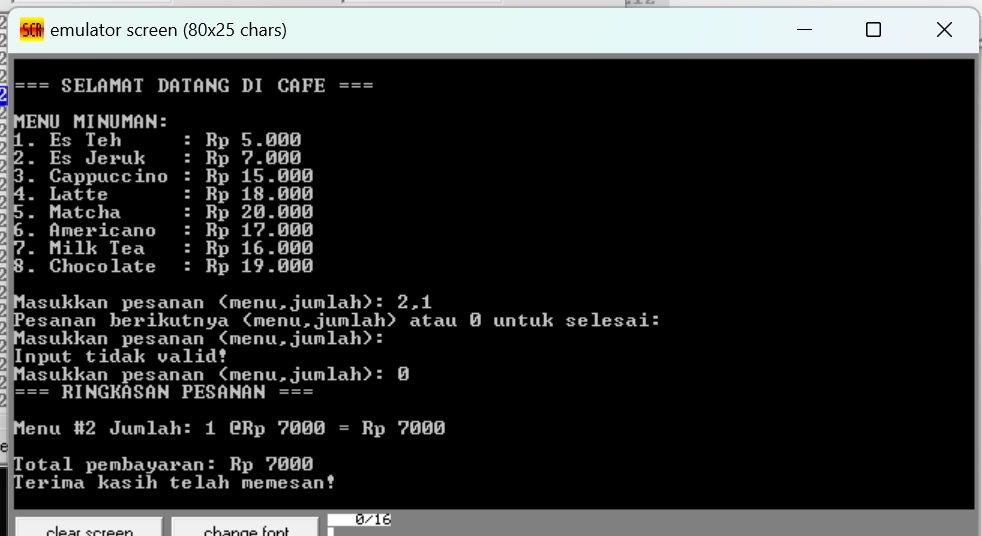
**UNIVERSITAS AHMAD DAHLAN**

**2024/2025**

1. **TAMPILAN AWAL APLIKASI**

Link Project Github :

https://github.com/DavaTrista/Program-Aplikasi-Pesan-Minuman-Di-CafeCapture Tampilan Aplikasi :



1. **RUAG LINGKUP**

Pembeli dan Warga lokal terdekat

1. **DESKRIPSI PROJEK**

Program ini adalah aplikasi sederhana yang menampilkan menu minuman dan memungkinkan pengguna untuk melakukan pemesanan dengan memasukkan nomor menu dan jumlahnya. Dimulai dengan menampilkan pesan selamat datang dan daftar menu, program kemudian meminta input dari pengguna dalam sebuah loop, menyimpan kuantitas pesanan dalam array jika input valid, atau menampilkan pesan kesalahan jika tidak.

Setelah pengguna selesai memasukkan pesanan, program menghitung dan menampilkan ringkasan pesanan, termasuk nomor menu, kuantitas, harga satuan, dan subtotal untuk setiap item. Total pembayaran akhir dihitung dari semua subtotal dan ditampilkan kepada pengguna bersama dengan pesan terima kasih. Dengan demikian, program ini menyediakan antarmuka interaktif yang sederhana untuk transaksi pemesanan minuman.

1. **DAFTAR SELURUH SPESIFIKASI APLIKASI SESUAI STUDI KASUS DAN SPESIFIKSI APLIKASI**

Fitur program adalah :

1. **Laman Selamat Datang**

Menampilkan pesan sambutan kepada pengguna.

1. **Daftar Menu Minuman**

Menampilkan pilihan minuman beserta harga masing-masing.

1. **Tempat Penginput Nomor Menu**

Memungkinkan pengguna untuk memasukkan nomor menu yang ingin dipesan.

1. **Tempat Penginput Jumlah Pesanan**

Memungkinkan pengguna untuk memasukkan jumlah minuman yang dipesan.

1. **Validasi Input**

Memeriksa validitas nomor menu dan jumlah pesanan yang dimasukkan oleh pengguna.

1. **Ringkasan Pesanan**

Menampilkan ringkasan pesanan yang mencakup nomor menu, kuantitas, harga satuan, subtotal, dan total pembayaran akhir.

1. **Pesan Kesalahan**

Menampilkan pesan kesalahan jika input tidak valid.

1. **Pesan Terima Kasih**

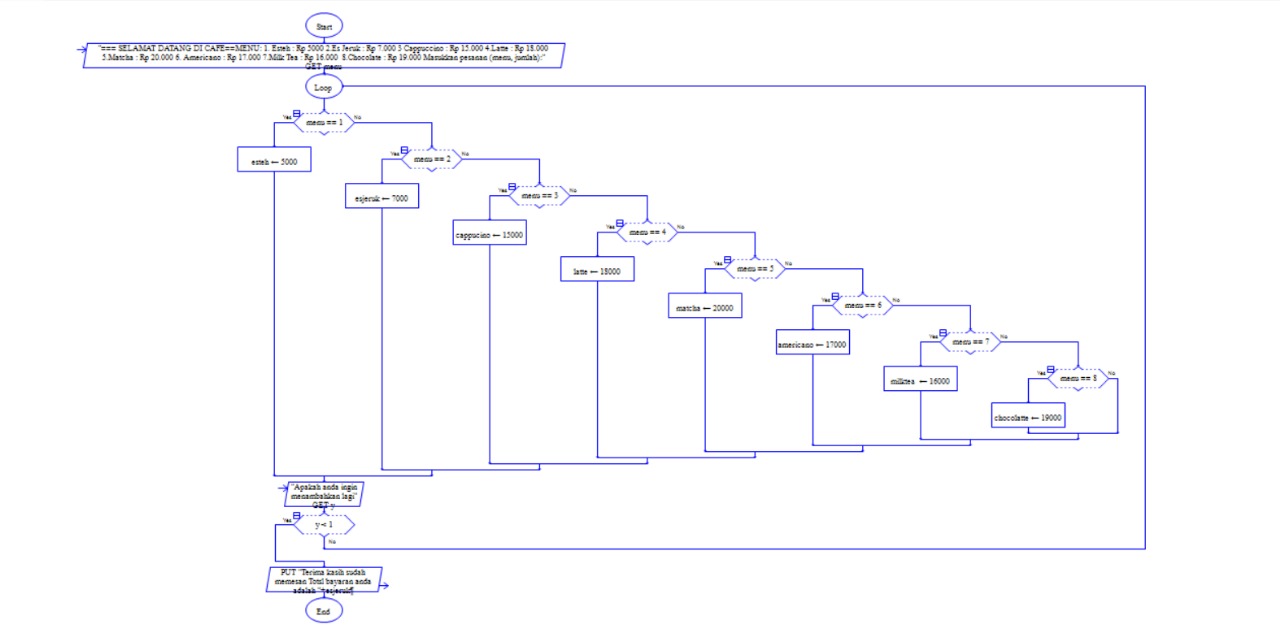
Menampilkan pesan terima kasih setelah proses pemesanan selesai

1. **SKETSA APLIKASI**

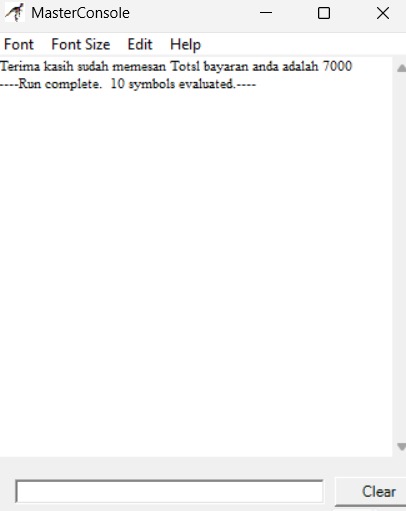


1. **DIAGRAM APLIKASI**

* **Tampilan flowchart**



* **Tampilan output flowchart**



1. **PENJELASAN/KODE PROGRAM**

.model small

.stack 100h

.data

; Teks Menu

welcome db 13,10,'=== SELAMAT DATANG DI CAFE ===',13,10,'$'

menu\_txt db 13,10,'MENU MINUMAN:',13,10

db '1. Es Teh : Rp 5.000',13,10

db '2. Es Jeruk : Rp 7.000',13,10

db '3. Cappuccino : Rp 15.000',13,10

db '4. Latte : Rp 18.000',13,10

db '5. Matcha : Rp 20.000',13,10

db '6. Americano : Rp 17.000',13,10

db '7. Milk Tea : Rp 16.000',13,10

db '8. Chocolate : Rp 19.000',13,10,'$'

input\_txt db 13,10,'Masukkan pesanan (menu,jumlah): $'

next\_txt db 13,10,'Pesanan berikutnya (menu,jumlah) atau 0 untuk selesai: $'

order\_sum db 13,10,'=== RINGKASAN PESANAN ===',13,10,'$'

item\_txt db 13,10,'Menu #$'

qty\_txt db ' Jumlah: $'

unit\_price db ' @Rp $'

price\_txt db ' = Rp $'

total\_txt db 13,10,13,10,'Total pembayaran: Rp $'

error\_txt db 13,10,'Input tidak valid!$'

thanks\_txt db 13,10,'Terima kasih telah memesan!$'

; Variabel

orders db 8 dup(0) ; Array untuk menyimpan jumlah pesanan setiap menu

prices dw 5000, 7000, 15000, 18000, 20000, 17000, 16000, 19000

menu\_names db 'Es Teh $'

db 'Es Jeruk $'

db 'Cappuccino$'

db 'Latte $'

db 'Matcha $'

db 'Americano $'

db 'Milk Tea $'

db 'Chocolate $'

choice db ?

quantity db ?

subtotal dw 0

grandtotal dd 0 ; Diubah menjadi double word untuk total yang lebih besar

.code

main proc

mov ax, @data

mov ds, ax

; Bersihkan layar

mov ax, 0003h

int 10h

; Tampilkan pesan selamat datang dan menu

lea dx, welcome

mov ah, 09h

int 21h

lea dx, menu\_txt

mov ah, 09h

int 21h

input\_loop:

; Tampilkan prompt input

lea dx, input\_txt

mov ah, 09h

int 21h

; Ambil pilihan menu

mov ah, 01h

int 21h

sub al, '0'

; Periksa jika selesai (0)

cmp al, 0

je calculate\_total

; Validasi pilihan menu

cmp al, 1

jl invalid

cmp al, 8

jg invalid

mov choice, al

; Ambil koma

mov ah, 01h

int 21h

; Ambil jumlah pesanan

mov ah, 01h

int 21h

sub al, '0'

; Validasi jumlah pesanan (1-9)

cmp al, 1

jl invalid

cmp al, 9

jg invalid

mov quantity, al

; Simpan pesanan dalam array

mov bl, choice

dec bl ; Sesuaikan untuk array yang berbasis 0

mov bh, 0

mov al, quantity

mov [orders + bx], al

; Tampilkan prompt pesanan berikutnya

lea dx, next\_txt

mov ah, 09h

int 21h

jmp input\_loop

calculate\_total:

; Tampilkan ringkasan pesanan

lea dx, order\_sum

mov ah, 09h

int 21h

; Reset grandtotal

mov word ptr grandtotal, 0

mov word ptr grandtotal+2, 0

; Loop melalui array pesanan

mov cx, 8 ; 8 item menu

mov si, 0 ; Indeks array

print\_orders:

mov al, [orders + si]

cmp al, 0

je next\_item ; Lewati jika jumlahnya 0

; Tampilkan nomor menu

lea dx, item\_txt

mov ah, 09h

int 21h

mov dl, si

inc dl ; Ubah menjadi nomor menu

add dl, '0'

mov ah, 02h

int 21h

; Tampilkan jumlah pesanan

lea dx, qty\_txt

mov ah, 09h

int 21h

mov dl, [orders + si]

add dl, '0'

mov ah, 02h

int 21h

; Tampilkan harga satuan

lea dx, unit\_price

mov ah, 09h

int 21h

mov bx, si

shl bx, 1 ; Kalikan dengan 2 untuk indeks array kata

mov ax, [prices + bx]

call display\_num

; Hitung subtotal

mov al, [orders + si] ; Ambil jumlah pesanan

xor ah, ah ; Bersihkan AH

mul word ptr [prices + bx] ; Kalikan dengan harga

mov subtotal, ax

; Tampilkan subtotal

lea dx, price\_txt

mov ah, 09h

int 21h

mov ax, subtotal

call display\_num

; Tambah ke grandtotal

mov ax, subtotal

add word ptr grandtotal, ax

adc word ptr grandtotal+2, 0 ; Tambah carry jika ada

next\_item:

inc si

loop print\_orders

; Tampilkan total grandtotal

lea dx, total\_txt

mov ah, 09h

int 21h

mov ax, word ptr grandtotal

call display\_num

***Penjelasan***:

**Baris 1-2:** Program dimulai dengan mendeklarasikan mode memori menggunakan .model small dan alokasi ukuran stack sebesar 256 byte menggunakan .stack 100h. Hal ini bertujuan untuk mendukung program dengan kebutuhan memori kecil dan mengatur stack sebagai penyimpanan sementara selama eksekusi.

**Baris 3-41:** Segmen data dideklarasikan dengan .data. Di dalamnya terdapat deklarasi string untuk berbagai pesan yang akan ditampilkan, seperti pesan selamat datang (welcome), daftar menu (menu\_txt), pesan input, serta pesan untuk ringkasan pesanan, total pembayaran, dan kesalahan input. Selain itu, terdapat array orders untuk menyimpan jumlah pesanan tiap menu, array prices untuk harga tiap menu, dan variabel lain seperti choice, quantity, subtotal, dan grandtotal untuk menyimpan data selama eksekusi program.

**Baris 43-44:** Program memasuki segmen kode dengan .code. Prosedur utama main dideklarasikan dengan main proc. Kemudian, segmen data diinisialisasi dengan memindahkan alamatnya ke register DS melalui mov ax, @data dan mov ds, ax.

**Baris 46-49:** Layar dibersihkan menggunakan interrupt BIOS int 10h dengan fungsi 03h (Clear Screen). Ini memastikan layar konsol kosong sebelum program mulai menampilkan output.

**Baris 51-56:** Program menampilkan pesan selamat datang (welcome) dan daftar menu minuman (menu\_txt) menggunakan interrupt DOS int 21h dengan fungsi 09h, yang digunakan untuk menampilkan string berakhiran $.

**Baris 58-109:** Label input\_loop menangani proses input dari pengguna. Program meminta pengguna memasukkan pesanan dengan menampilkan pesan input\_txt menggunakan interrupt DOS. Kemudian, karakter yang dimasukkan dibaca menggunakan int 21h dengan fungsi 01h dan diubah menjadi angka dengan sub al, '0'.

**Baris 62-65:** Jika pengguna memasukkan 0, program melompat ke label calculate\_total untuk menghitung total pesanan.

**Baris 67-73:** Validasi input dilakukan untuk memastikan bahwa menu yang dipilih berada di antara 1 dan 8. Jika input tidak valid, program melompat ke label invalid. Jika valid, menu pilihan disimpan di variabel choice.

**Baris 75-82:** Program membaca jumlah pesanan setelah tanda koma, memvalidasi input agar berada di antara 1 dan 9, dan menyimpan nilai jumlah ke variabel quantity. Kemudian, jumlah tersebut disimpan di array orders berdasarkan indeks menu yang dipilih.

**Baris 84-87:** Setelah input pesanan berhasil, program menampilkan pesan next\_txt untuk meminta input tambahan. Proses kembali ke input\_loop untuk menerima pesanan berikutnya.

**Baris 109-152:** Label calculate\_total menangani perhitungan total pesanan. Program menampilkan ringkasan pesanan dengan pesan order\_sum, mereset nilai grandtotal, dan menggunakan loop untuk memeriksa semua menu dalam array orders.

**Baris 115-146:** Dalam loop print\_orders, jika terdapat pesanan pada suatu menu (nilai di array orders tidak nol), program menampilkan nomor menu, jumlah pesanan, harga satuan, dan subtotal. Subtotal dihitung dengan mengalikan jumlah pesanan dengan harga per menu, dan hasilnya ditambahkan ke grandtotal.

**Baris 148-152:** Setelah semua pesanan diproses, program menampilkan total pembayaran (total\_txt) dan pesan terima kasih (thanks\_txt).

**Baris 154-157:** Label invalid menangani kesalahan input dengan menampilkan pesan error\_txt dan mengarahkan kembali ke input\_loop.

**Baris 159-161:** Label exit mengakhiri program dengan menggunakan interrupt DOS int 21h fungsi 4Ch, yang menghentikan eksekusi dan mengembalikan kontrol ke sistem operasi.

**Baris 163-182:** Prosedur display\_num digunakan untuk menampilkan angka desimal. Angka dipecah menjadi digit-digits dengan operasi pembagian (div), dan setiap digit ditampilkan secara terbalik menggunakan interrupt DOS int 21h fungsi 02h.

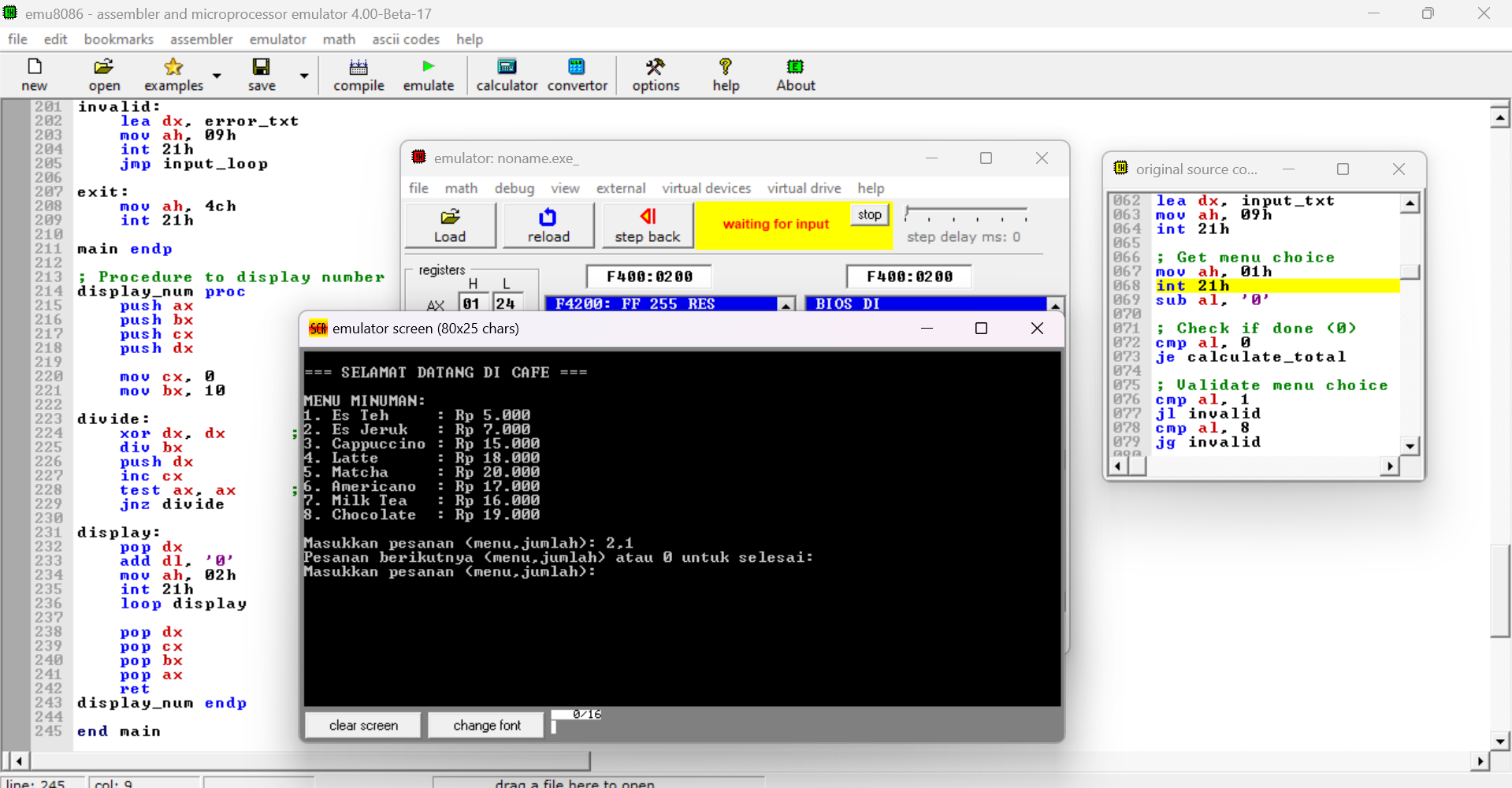
**Baris 184:** Program berakhir dengan deklarasi end main, menunjukkan titik masuk utama program.

1. **DESKRIPSI TAMPILAN**

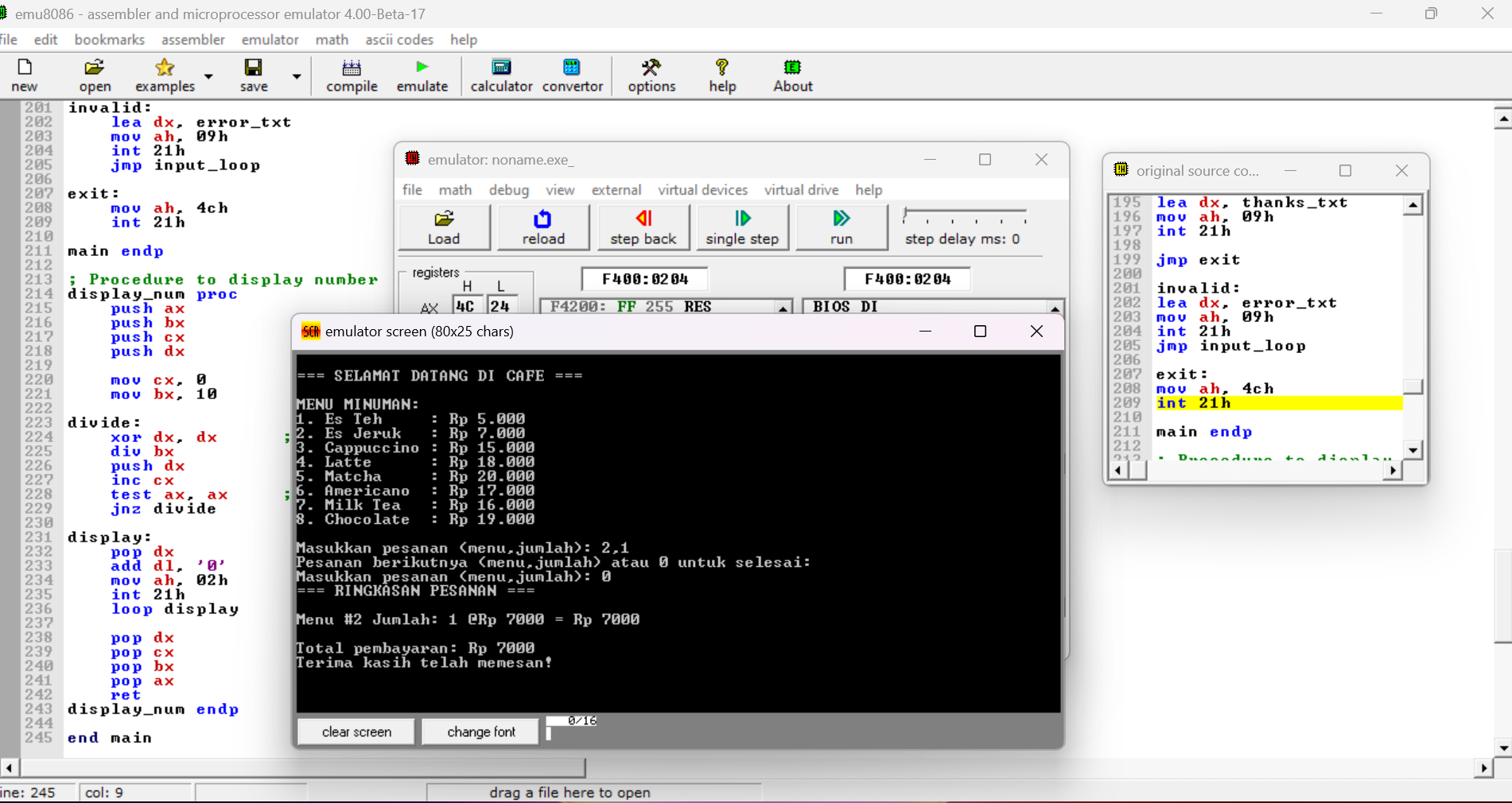
* ***Program dijalankan***

****

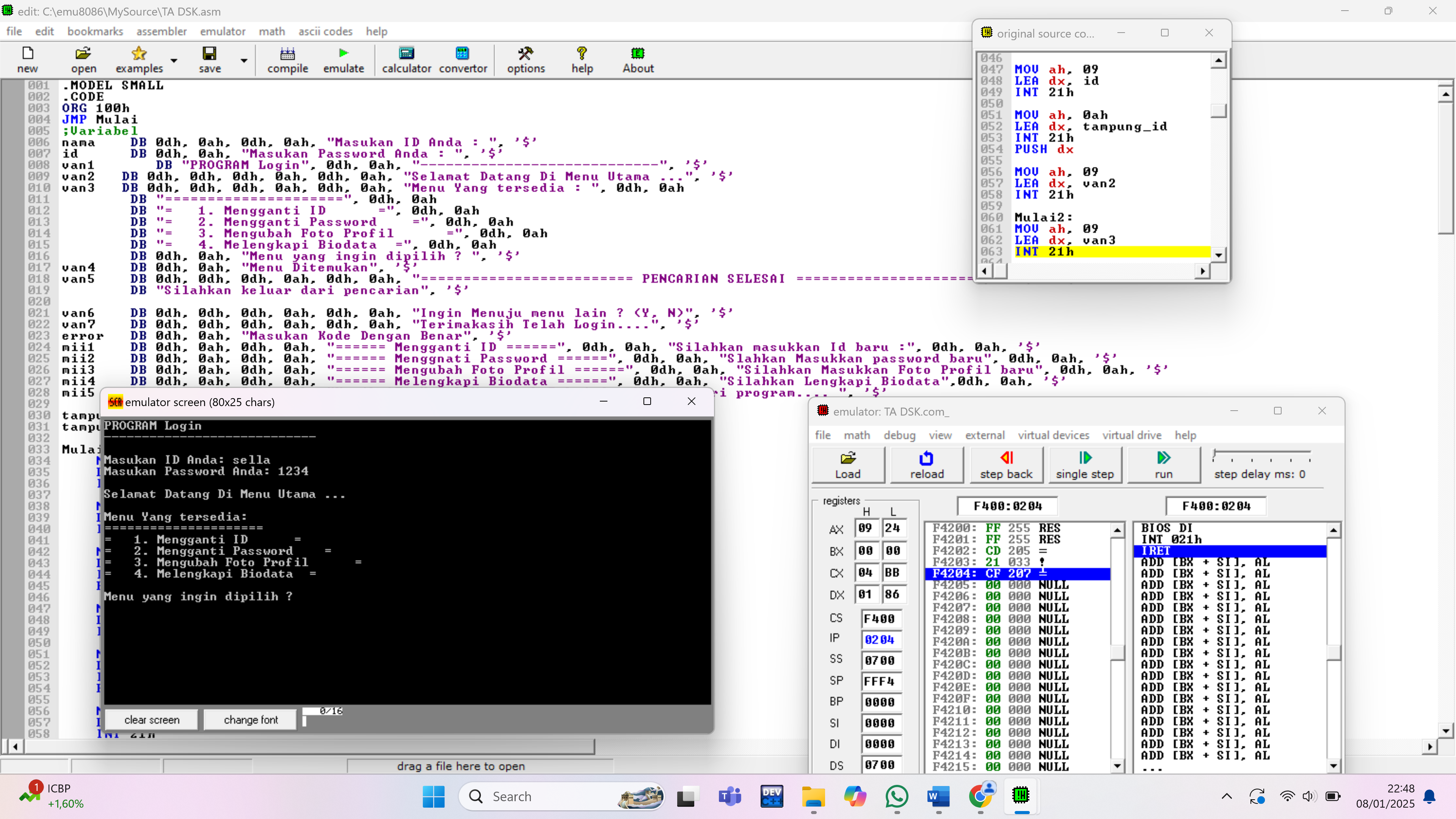
* **Pengguna meminta input pesanan**

****

* **Pengguna meminta memasukkan angka o untuk mengakhiri**

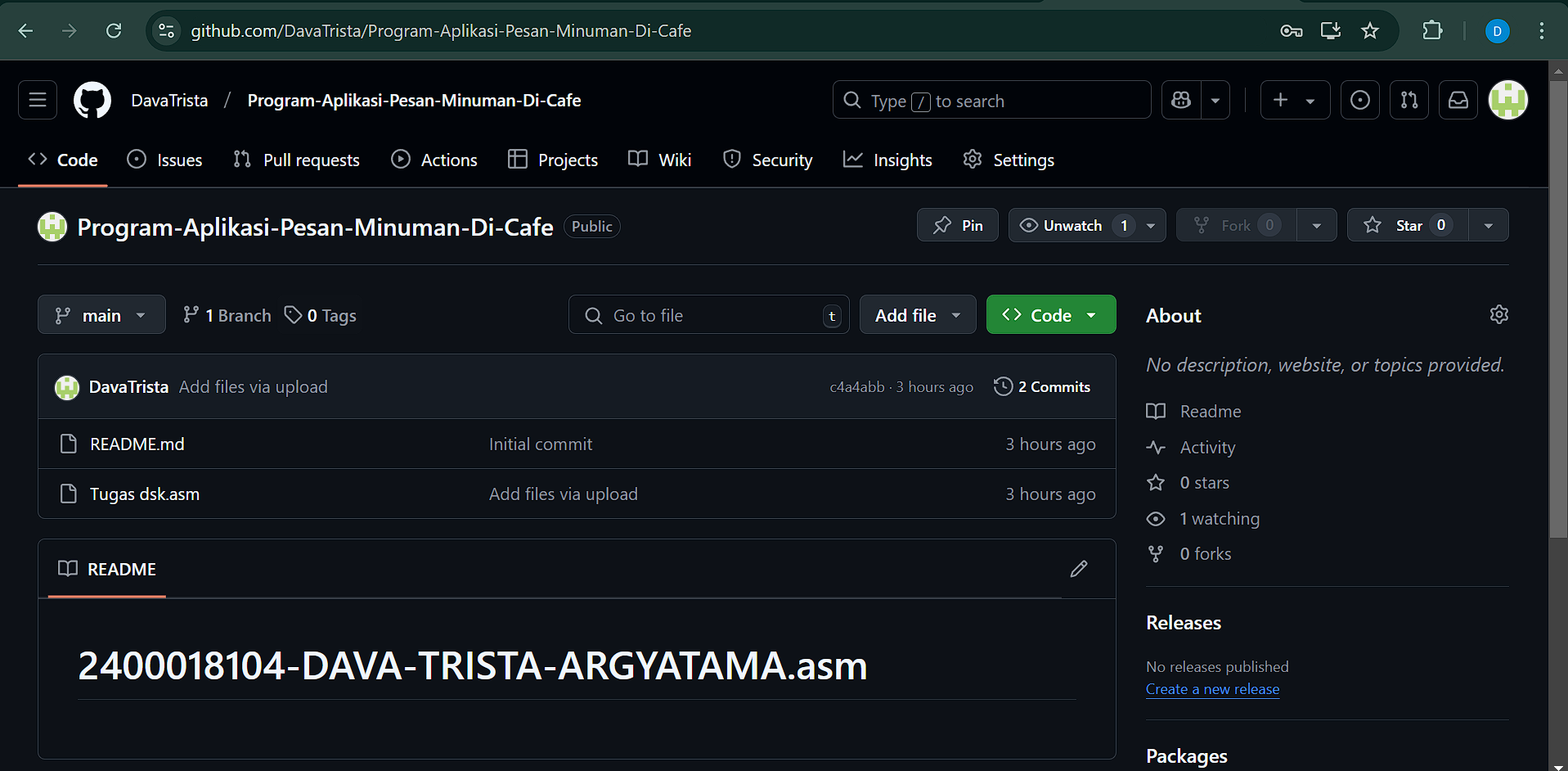
****

* **Menampilkan menu utama**

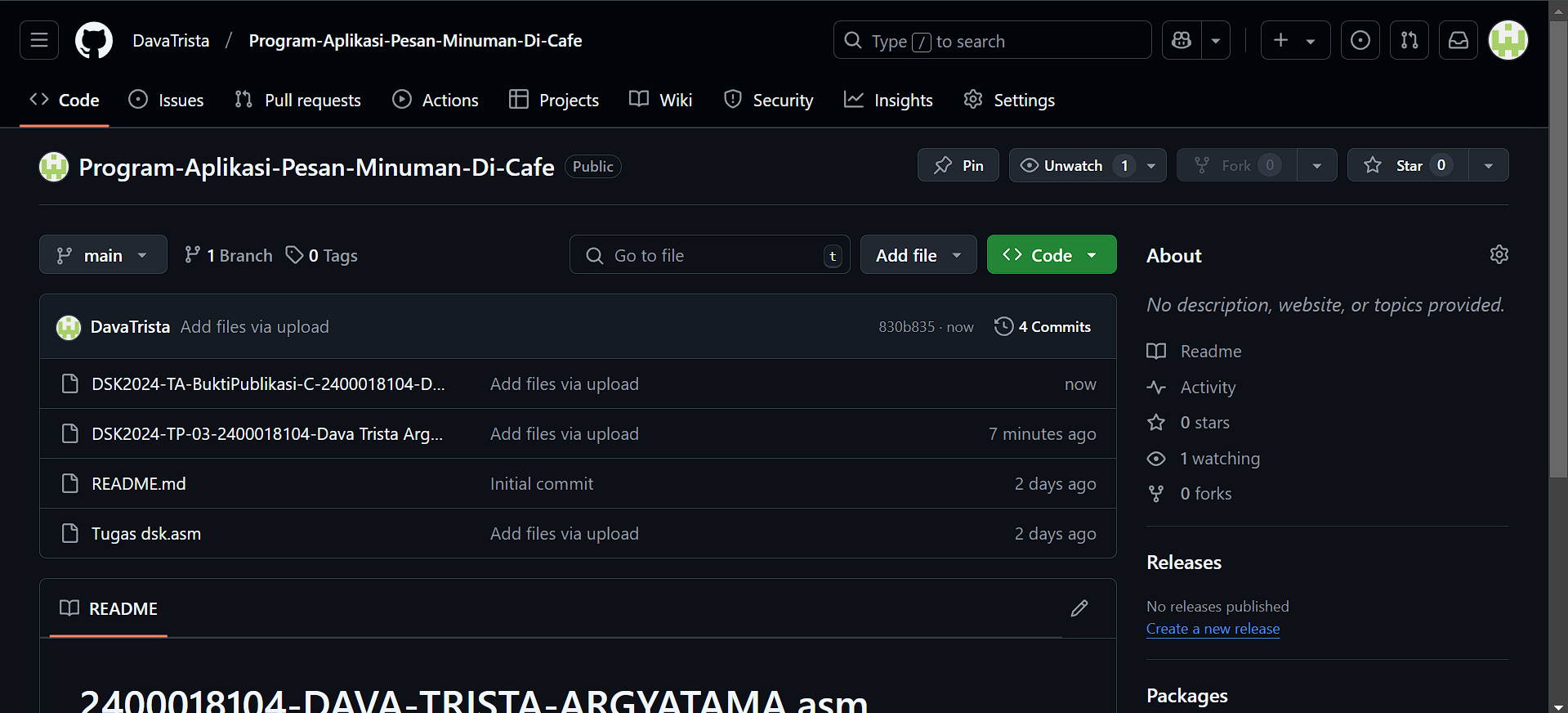
****

1. **DESKRIPSI SCREENSHOOT GITHUB**

* Tampilan Unggahan awal pada GitHub yang berisi file Program TA DSK.asm dan file readme.



* Tampilan unggahan GitHub setelah diperbarui.



1. **ANALISIS PENGERJAAN TUGAS PROJEK**

Proyek ini diselesaikan dalam waktu 3 hingga 5 hari, dengan fokus pada logika pemrograman dan bahasa Assembly. Saya menggunakan berbagai referensi dari internet, tutorial YouTube, dan diskusi dengan kakak tingkat yang berpengalaman dalam pemrograman sistem komputer dasar. Proyek ini tidak memerlukan biaya tambahan karena memanfaatkan perangkat lunak emulator yang sudah ada.Selama pengerjaan, saya menghadapi tantangan terutama terkait pemahaman bahasa Assembly yang memiliki sintaksis spesifik dan mendekati bahasa mesin. Ini memerlukan pembelajaran mendalam tentang penggunaan register, pengelolaan array, dan manipulasi data dengan operasi aritmatika dasar.

Waktu pengerjaan juga menjadi kendala karena harus disesuaikan dengan tugas kuliah dan kegiatan sehari-hari.Program yang dibuat bertujuan untuk menciptakan aplikasi simulasi sederhana sistem pemesanan di kafe. Aplikasi ini mencakup tampilan menu, penerimaan input pesanan, perhitungan total biaya, dan ringkasan pesanan dengan validasi input untuk menghindari kesalahan eksekusi.Tantangan ke depan adalah mengembangkan program ini untuk meningkatkan efisiensi dan menambah fitur seperti dukungan menu lebih banyak, pengelolaan data pelanggan, serta menciptakan antarmuka yang lebih ramah pengguna.

Persaingan dengan aplikasi modern yang lebih canggih menjadi motivasi untuk terus meningkatkan keterampilan dalam bahasa pemrograman baik tingkat rendah maupun tinggi.Melalui proyek ini, saya memperoleh pengalaman berharga dalam pengelolaan data di tingkat sistem operasi, khususnya penggunaan interrupt di DOS serta pengelolaan array, loop, dan kondisi dalam pemrograman berbasis Assembly.