

TUGAS AKHIR DSK
MEMBUAT APLIKASI MENGGUNAKAN
BAHASA ASSEMBLY



NAMA : GIGIH NURFADHILAH AKBAR
NIM : 2500018126
KELAS : C
MATKUL : DASAR SISTEM KOMPUTER
JUDUL APK : SIMULASI PENGOLAAN STOK GUDANG

PROGRAM STUDY INFORMATIKA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS AHMAD DAHLAN

2026

A. Ruang lingkup aplikasi

1. Fungsi utama

- Menerima input jumlah stok awal (0–9)
- Menerima pilihan operasi pengelolaan stok:
 - + untuk menambah stok
 - – untuk mengurangi stok
- Menerima input jumlah stok perubahan
- Menampilkan jumlah stok akhir
- Menangani kondisi stok bernilai negatif

2. Batasan Input

Masih banyak kekurangan dari apk ini, diantaranya :

- Input stok hanya berupa bilangan satu digit (0–9)
- Hanya mendukung satu jenis barang
- Operasi yang tersedia hanya penambahan dan pengurangan stok
- Belum mendukung penyimpanan data secara permanen

3. Bahasa program

Disini saya menggunakan Bahasa pemograman assembly, sesuai intruksi dari dosen pengampuh, yaitu Bpk Ali Tarmuji.

Bahasa assembly ini adalah Bahasa yang dekat dengan Bahasa mesin (Bahasa tingkat rendah)

Dan menggunakan tools emu8086

4. Target Pengguna

Aplikasi ditujukan bagi mahasiswa yang sedang mempelajari sistem bilangan di mata kuliah Dasar Sistem Komputer.

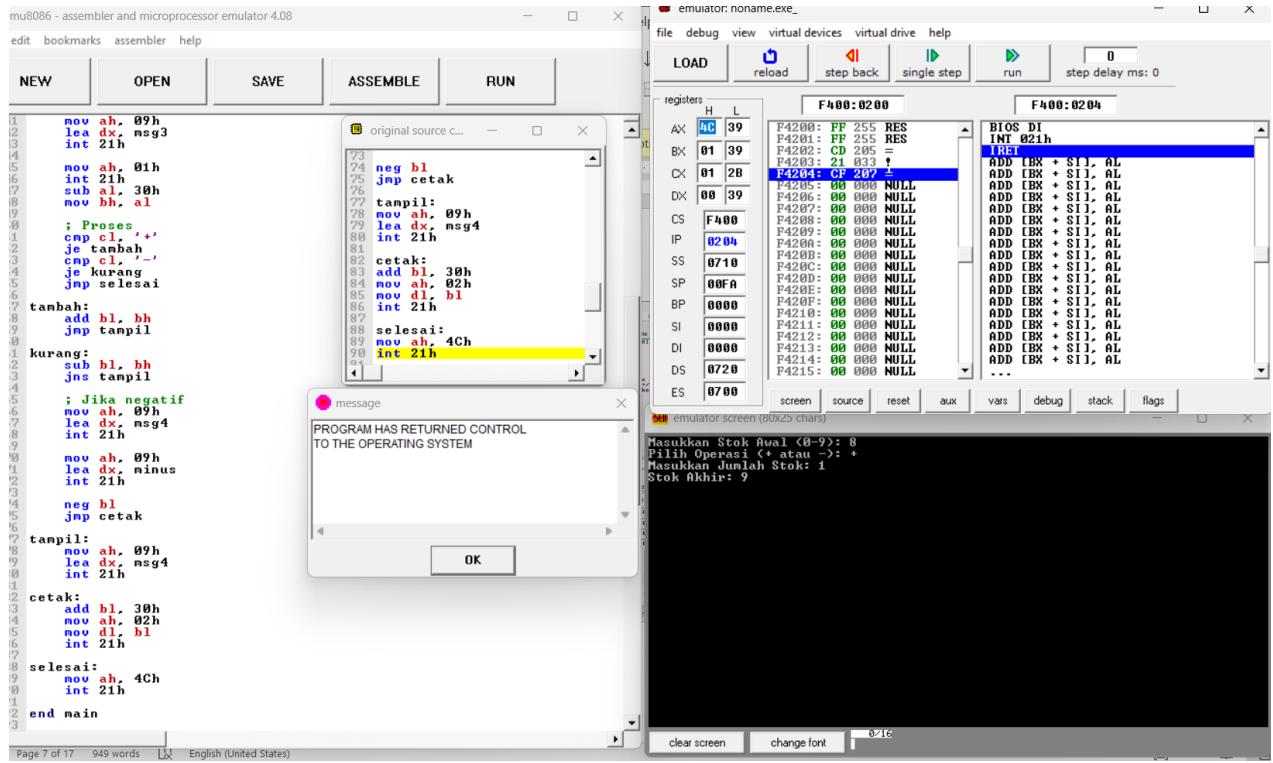
B. Visual Aplikasi

The screenshot shows a terminal window with the title "emulator screen (80x25 chars)". The window contains the following text:

```
Masukkan Stok Awal <0-9>: 8
Pilih Operasi (+ atau -): +
Masukkan Jumlah Stok: 1
Stok Akhir: 9
```

At the bottom of the window, there is a toolbar with three buttons: "clear screen", "change font", and a status indicator "0/16".

C. Hasil Akhir



Disini saya menggunakan tools emu 8086.

Pada gambar hasil akhir diatas kita bisa lihat, code program, proses tracking setiap baris code dan hasil dari program yang dibuat menggunakan Bahasa assembly.

Kenapa saya menggunakan emu 8086?

Karena tampilan UI yang lebih mudah digunakan dan banyak fitur yang dapat digunakan, salah satunya tracking code perbaris, sehingga kita bisa lebih mudah mengetahui di bagian mana yang error.

1. Deskripsi Program

Program **Simulasi Pengolaan Stok Gudang** adalah aplikasi berbasis Assembly (DOS mode) yang mensimulasikan proses pengelolaan stok dengan operasi penambahan dan pengurangan.

Program menerima input berupa stok awal, jenis operasi, dan nilai perubahan stok. Hasil akhir berupa stok gudang ditampilkan ke layar. Program mampu menangani stok negatif dan menampilkan hasil hingga dua digit.

2. Struktur Utama Program

Inisialisasi Program

Pada bagian awal program digunakan deklarasi:

- Model Small
- Stack
- Data

Bagian ini berfungsi untuk menyiapkan segmentasi memori, stack, serta variabel pesan (string) yang akan ditampilkan ke layar.

Manajemen Data

Instruksi:

`mov ax, @data` dan `mov dxx, ax`

3. Mekanisme Input dan Konversi ASCII

Input dari keyboard dibaca menggunakan interrupt:

INT 21h, fungsi 01h

Karena input masih berupa karakter ASCII, maka dilakukan konversi ke nilai numerik menggunakan:

sub al, 30h

Instruksi ini mengubah karakter ASCII menjadi nilai angka yang dapat diproses secara aritmatika sebagai stok.

4. Logika Pengolahan Stok dan Kontrol Alur

Program menggunakan instruksi:

- CMP untuk membandingkan operator
- JE dan JMP untuk menentukan alur program

Jika operasi +, maka stok akan ditambah. Jika operasi -, maka stok akan dikurangi.

Untuk menangani stok negatif, program menggunakan pengecekan **Sign Bit (MSB)** dengan instruksi:

TEST bl, 80h

Jika hasil negatif, maka program akan menampilkan tanda minus (-) sebelum menampilkan nilai stok.

5. Mekanisme Output

Jika stok akhir memiliki dua digit (≥ 10), maka program:

- Membagi nilai dengan 10 menggunakan DIV
- Memisahkan digit puluhan dan satuan
- Mengonversi kembali ke ASCII dengan menambahkan 30h

Output kemudian ditampilkan menggunakan interrupt:

6. Tabel Ringkasan

MOV	Transfer data	Peran dalam Simulasi Stok Gudang
ADD	Operasi Aritmatika	Menambahkan jumlah stok ke stok awal (proses stok masuk).

SUB	Operasi Aritmatika	Mengurangi stok awal dengan jumlah stok(proses stok keluar).
CMP	Perbandigan Data	Membandingkan operator input (+ atau – untuk menentukan proses stok
JE	Percabangan Bersyarat	Melompat ke proses menambahkan stok jika oprator +.
JNS	Percabangan Tanda	Mengecek apakah hasil stok bernilai poositif atau negatif
NEG	Negasi bilangan	Mengubah stok negative menjadii positif sebelum di tampilkan
INT 21h	Interrupt DOS	Digunakan untuk input keyboard, output teks, dan mengakhiri program.
SUB AL	Konversi ASCII	Mengubah input karakter angka menjadi nilai numerik stok
ADD BL, 30h	Konversi ke ASCII	Mengubah nilai menjadi karakter agar bisa di sampaikan

D. Code Program

```
.model small
.stack 100h

.data
    msg1 db 'Masukkan Stok Awal
(0-9): $'
    msg2 db 13,10,'Pilih Operasi
(+/-): $'
    msg3 db 13,10,'Masukkan
Jumlah Stok: $'
    msg4 db 13,10,'Stok Akhir: $'
    minus db '-$'

.code
main proc
    mov ax, @data
    mov ds, ax

    ; Input stok awal
    mov ah, 09h
    lea dx, msg1
    int 21h

    mov ah, 01h
    int 21h
    sub al, 30h
    mov bl, al
```

; Input operasi

mov ah, 09h

lea dx, msg2

int 21h

mov ah, 01h

int 21h

mov cl, al

; Input jumlah stok

mov ah, 09h

lea dx, msg3

int 21h

mov ah, 01h

int 21h

sub al, 30h

mov bh, al

; Proses

cmp cl, '+'

je tambah

cmp cl, '-'

je kurang

jmp selesai

tambah:

add bl, bh

jmp tampil

kurang:

sub bl, bh

jns tampil

; Jika stok negatif

mov ah, 09h

lea dx, msg4

int 21h

mov ah, 09h

lea dx, minus

int 21h

neg bl

jmp cetak

tampil:

mov ah, 09h

lea dx, msg4

int 21h

cetak:

add bl, 30h

mov ah, 02h

mov dl, bl

int 21h

selesai:

mov ah, 4Ch

int 21h

```
main endp  
end main
```