



TVGAS

AKHIR

DASAR SISTEM KOMPVETER

SAFINA EKA ANINDITA





PENYUSUN



SAFINA EKA ANINDITA 2400018130





# FITUR KALKULATOR SEDERHANA



1

KALKULATOR SEDERHANA

2

PENJUMLAHAN

3

PENGURANGAN

4

PERKALIAN

5

PEMBAGIAN

6

HASIL AKHIR

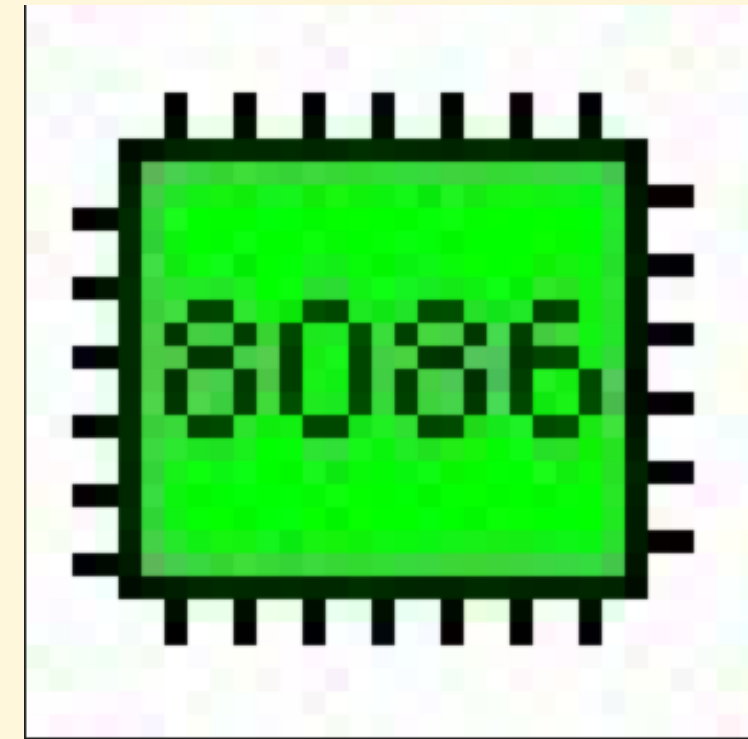


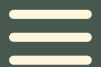
## DESKRIPSI PROGRAM SEDERHANA



Komputasi digital memainkan peran penting dalam kehidupan modern, mulai dari aplikasi sederhana hingga kompleks. Salah satu contoh dasar penerapan logika komputasi adalah kalkulator. Kalkulator sederhana berbasis kode assembly dapat digunakan sebagai latihan pemrograman dasar pada lingkungan emulator 8086 untuk memahami pengoperasian arsitektur prosesor dan sistem input/output.

Kode assembly memberikan kontrol langsung terhadap perangkat keras dengan menggunakan instruksi tingkat rendah. Dalam pembuatan kalkulator sederhana, programmer belajar tentang pengelolaan register, manipulasi data, dan pemrograman interaksi dengan sistem operasi melalui interupsi.





## TUJUAN PROGRAM



Memahami Konsep Dasar Assembly Language

Program ini dirancang untuk membantu pemula memahami struktur dasar bahasa assembly, seperti pengolahan data menggunakan register, memori, dan instruksi dasar (ADD, SUB, MUL, DIV).

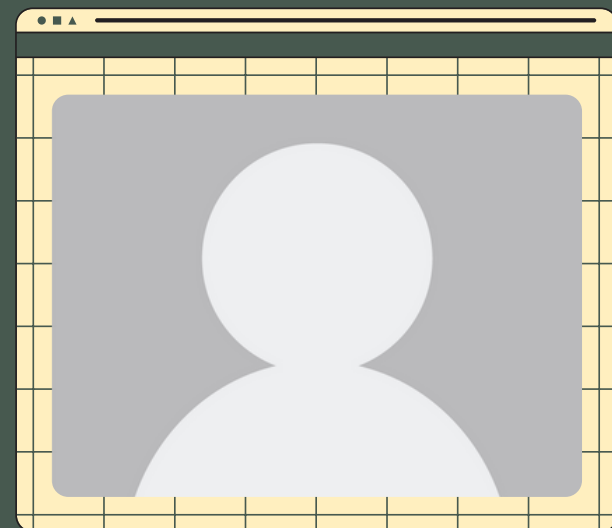
Melatih Penggunaan Interrupt BIOS

Program menggunakan interrupt seperti INT 21h untuk input dan output, memberikan wawasan tentang cara berinteraksi dengan perangkat keras pada level rendah.

Implementasi Operasi Aritmatika Dasar  
Program ini bertujuan untuk mempraktikkan implementasi operasi aritmatika dasar (penjumlahan, pengurangan, perkalian, dan pembagian) dalam lingkungan assembly, dengan logika sederhana yang sesuai untuk arsitektur prosesor 8086.



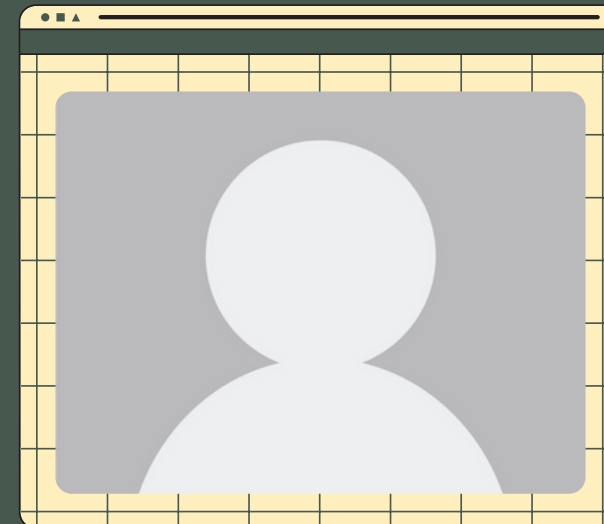
# KEKURANGAN & KELEBIHAN



## kekurangan

Kelebihan

- Struktur sederhana dan mudah dipahami.
- Menggunakan instruksi dasar yang cocok untuk pemula dalam assembly.



## kelebihan

Kekurangan

- Tidak ada validasi input selain operator.
- Tidak menangani angka negatif atau desimal.
- Tidak ada mekanisme penanganan kesalahan pembagian dengan nol.

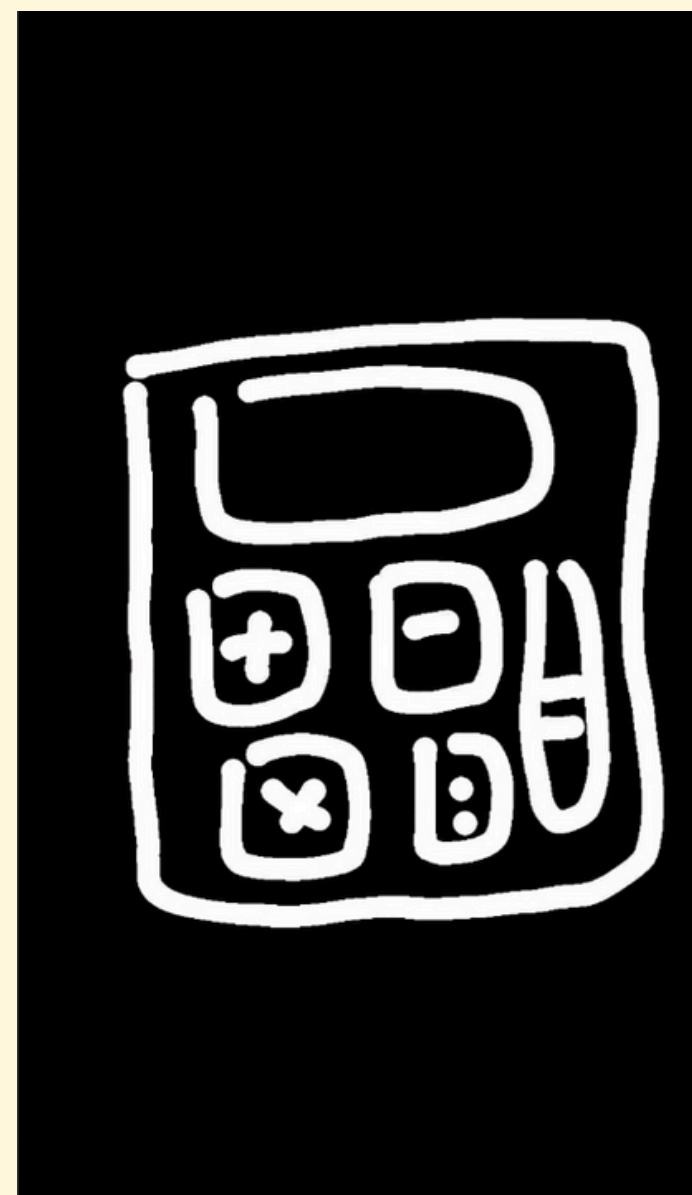




sketsa rancangan



ooo





## HASIL EKSEKUSI PROGRAM



1

BAGIAN AWAL

3

PROSES INPUT PROGRAM

2

MENAMPILKAN OPERATOR

4

MENAMPILKAN DETAIL KALKULATOR







# HASIL EKSEKUSI



1

TULISAN KALKULATOR SEDERHANA

4

3.PERKLIAN

2

1.PENJUMLAHAN

5

4.PEMBAKIAN


3

2.PENGURANGAN

6

HASIL INPUT DARI OPERASI TADI

# KODE PROGRAM



```
emu8086 - assembler and microprocessor emulator 4.08
file edit bookmarks assembler emulator math ascii codes help
new open examples save compile emulate calculator convertor options help about

0001 org 100h ; Mulai program di alamat 100h
0002
0003 ; Data
0004 menu db 'Kalkulator Sederhana', 0Dh, 0Ah
0005 db '1. Penjumlahan', 0Dh, 0Ah
0006 db '2. Pengurangan', 0Dh, 0Ah
0007 db 'Pilih operasi (1-2): $'
0008
0009 input_msg db 0Dh, 0Ah, 'Masukkan bilangan pertama: $'
0010 input_msg2 db 0Dh, 0Ah, 'Masukkan bilangan kedua: $'
0011 result_msg db 0Dh, 0Ah, 'Hasil: $'
0012 buffer db 6 dup('$') ; Buffer untuk angka yang akan ditampilkan
0013
0014 .code
0015 start:
0016 ; Menampilkan menu
0017 mov ah, 09h
0018 lea dx, menu
0019 int 21h
0020
0021 ; Meminta pengguna memilih operasi
0022 mov ah, 01h
0023 int 21h ; Membaca karakter pilihan
0024 sub al, '0' ; Mengonversi karakter ke angka
0025 mov bl, al ; Menyimpan pilihan operasi di BL
0026
0027 ; Meminta input angka pertama
0028 lea dx, input_msg
0029 mov ah, 09h
0030 int 21h ; Menampilkan prompt untuk angka pertama
0031 call input_number
0032 mov cx, ax ; Menyimpan bilangan pertama di CX
0033
0034 ; Meminta input angka kedua
0035 lea dx, input_msg2
0036 mov ah, 09h
0037 int 21h ; Menampilkan prompt untuk angka kedua
0038 call input_number
0039 mov dx, ax ; Menyimpan bilangan kedua di DX
0040
0041 ; Menangani operasi berdasarkan pilihan
0042 cmp bl, 1 ; Cek apakah pilihan 1 (Penjumlahan)
0043 je addition
0044 cmp bl, 2 ; Cek apakah pilihan 2 (Pengurangan)
0045 je subtraction
0046 jmp exit
0047
0048 addition:
0049 add cx, dx ; CX = CX + DX (Penjumlahan)
0050 mov ax, cx ; Menyimpan hasil penjumlahan ke AX
0051 call display_result
0052 jmp start ; Kembali ke menu utama
0053
0054 subtraction:
```

# KODE PROGRAM



16 - assembler and microprocessor emulator 4.08

bookmarks assembler emulator math ascii codes help

open examples save compile emulate calculator convertor options help about

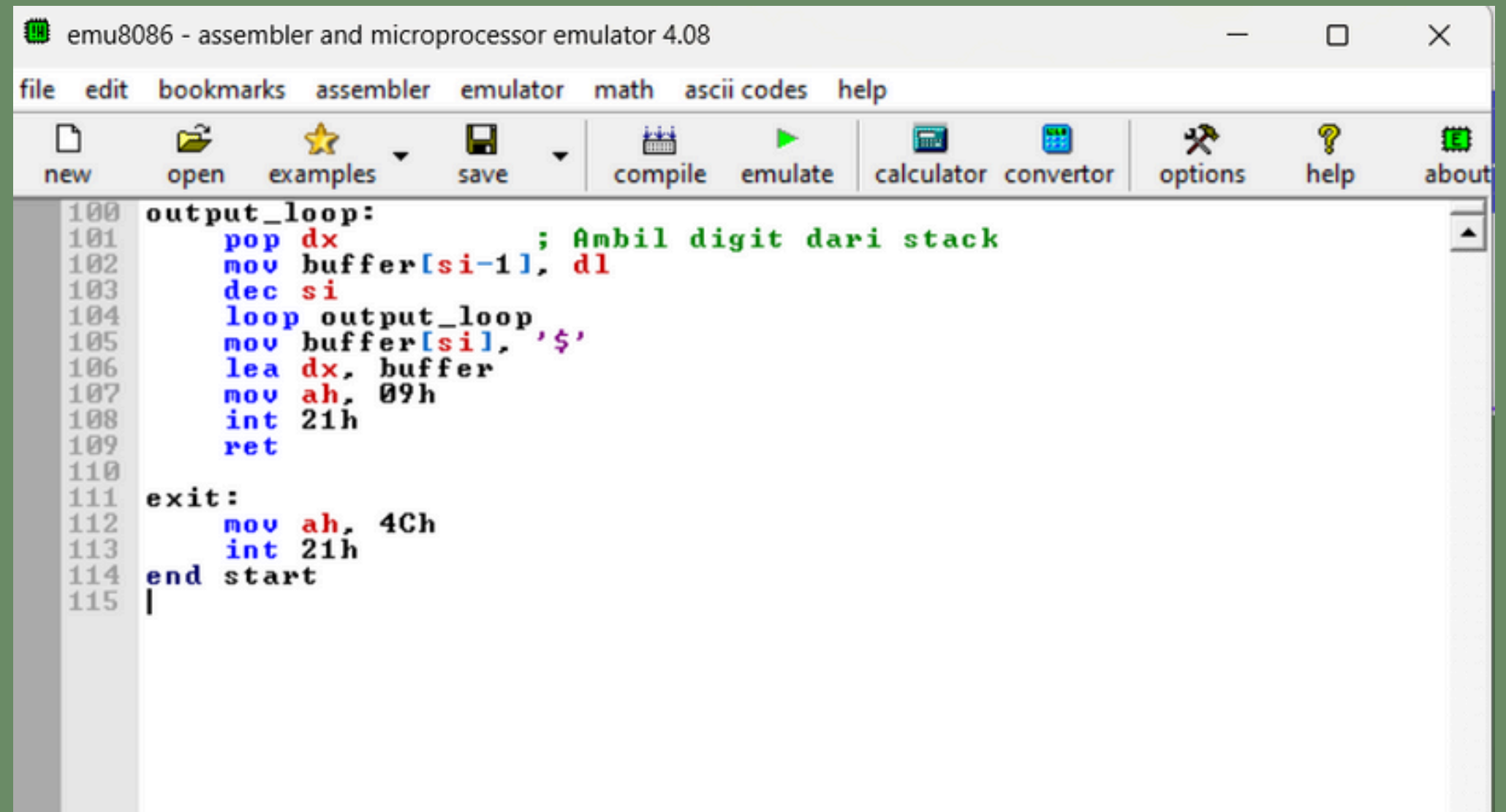
```
subtraction:
    sub cx, dx        ; CX = CX - DX <Pengurangan>
    mov ax, cx        ; Menyimpan hasil pengurangan ke AX
    call display_result
    jmp start         ; Kembali ke menu utama

; Fungsi untuk input bilangan
input_number:
    xor ax, ax        ; Bersihkan AX
    mov cx, 0         ; Menyimpan hasil input
    mov bl, 10        ; Basis 10 untuk konversi
input_loop:
    mov ah, 01h       ; Membaca karakter
    int 21h
    sub al, '0'       ; Mengonversi karakter ke angka
    cmp al, 0         ; Cek apakah angka valid
    jl done_input
    mul bl            ; AX = AX * 10
    add ax, cx        ; Menambahkan angka ke hasil konversi
    mov cx, ax        ; Simpan hasil konversi
    jmp input_loop
done_input:
    ret

; Fungsi untuk menampilkan hasil
display_result:
    mov ah, 09h
    lea dx, result_msg
    int 21h
    mov ax, cx
    call print_number
    ret

; Fungsi untuk mencetak angka
print_number:
    xor bx, bx        ; Bersihkan BX
    mov cx, 10        ; Basis 10
    mov si, 0         ; Indeks buffer
convert_loop:
    xor dx, dx
    div cx            ; Membagi AX dengan 10
    add dl, '0'       ; Konversi ke karakter ASCII
    push dx           ; Simpan digit ke stack
    inc si            ; Increment indeks
    test ax, ax       ; Cek apakah sudah habis
    jnz convert_loop
output_loop:
    pop dx            ; Ambil digit dari stack
    mov buffer[si-1], dl
    dec si
    loop output_loop
    mov buffer[si], '$'
    lea dx, buffer
```

# KODE PROGRAM

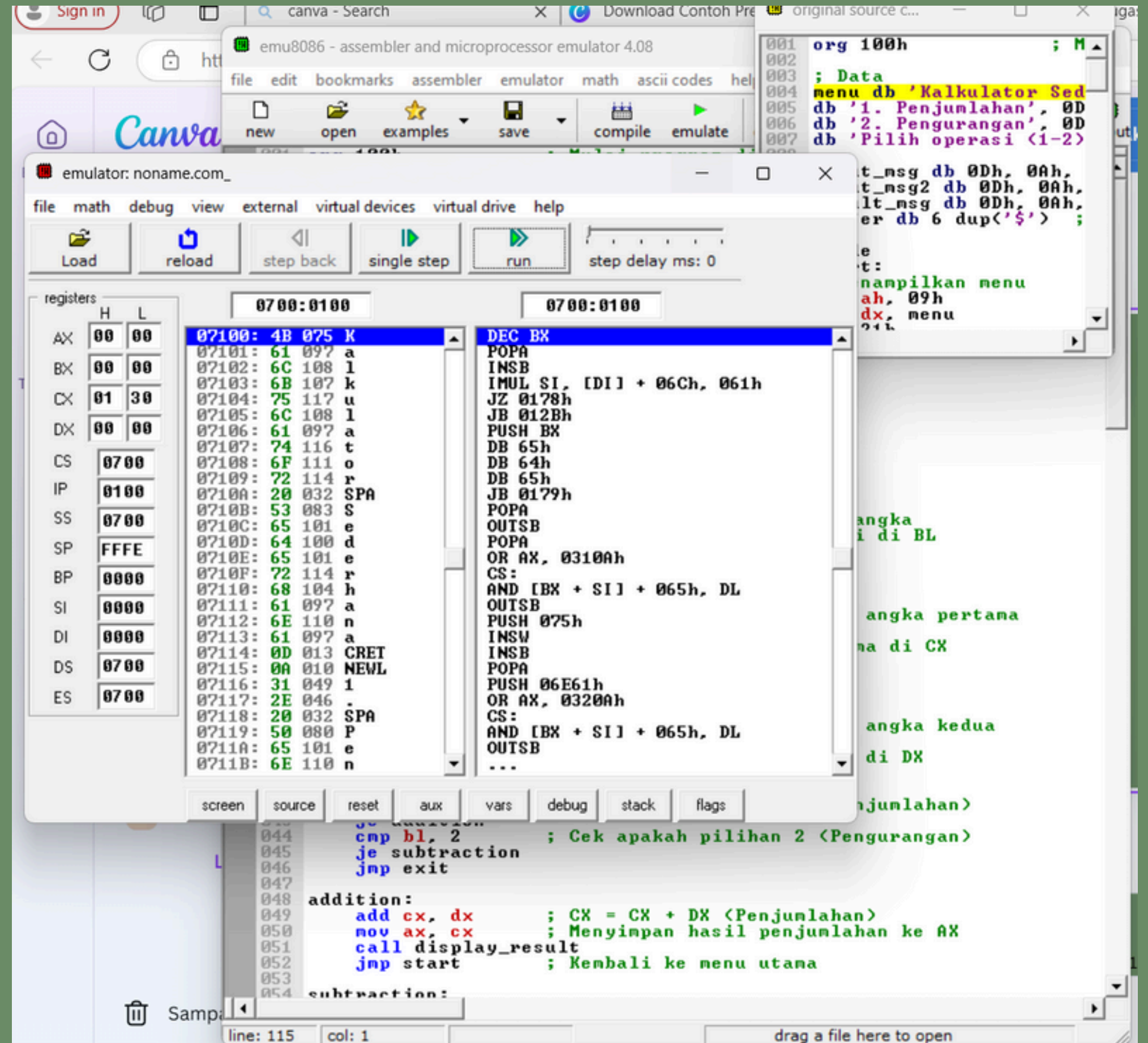


The screenshot shows the emu8086 - assembler and microprocessor emulator 4.08 window. The menu bar includes file, edit, bookmarks, assembler, emulator, math, ascii codes, and help. The toolbar contains icons for new, open, examples, save, compile, emulate, calculator, convertor, options, help, and about. The main window displays the following assembly code:

```
100 output_loop:
101     pop dx                ; Ambil digit dari stack
102     mov buffer[si-1], dl
103     dec si
104     loop output_loop
105     mov buffer[si], '$'
106     lea dx, buffer
107     mov ah, 09h
108     int 21h
109     ret
110
111 exit:
112     mov ah, 4Ch
113     int 21h
114 end start
115 |
```



# TAMPILAN PROGRAM



# OUTPUT PERTAMBAHAN



emulator screen (80x25 chars)

Kalkulator Sederhana:

1. Penjumlahan

2. Pengurangan

3. Perkalian

4. Pembagian

Pilih operasi (1-4): 1

Masukkan bilangan: 2

Masukkan bilangan: 3

Hasil: 5



# OUTPUT PENGURANGAN

Scr emulator screen (80x25 chars)

```
Kalkulator Sederhana:  
1. Penjumlahan  
2. Pengurangan  
3. Perkalian  
4. Pembagian  
Pilih operasi <1-4>: 2  
Masukkan bilangan: 9  
Masukkan bilangan: 4  
Hasil: 5
```

# OUTPUT PERKALIAN

 emulator screen (80x25 chars)

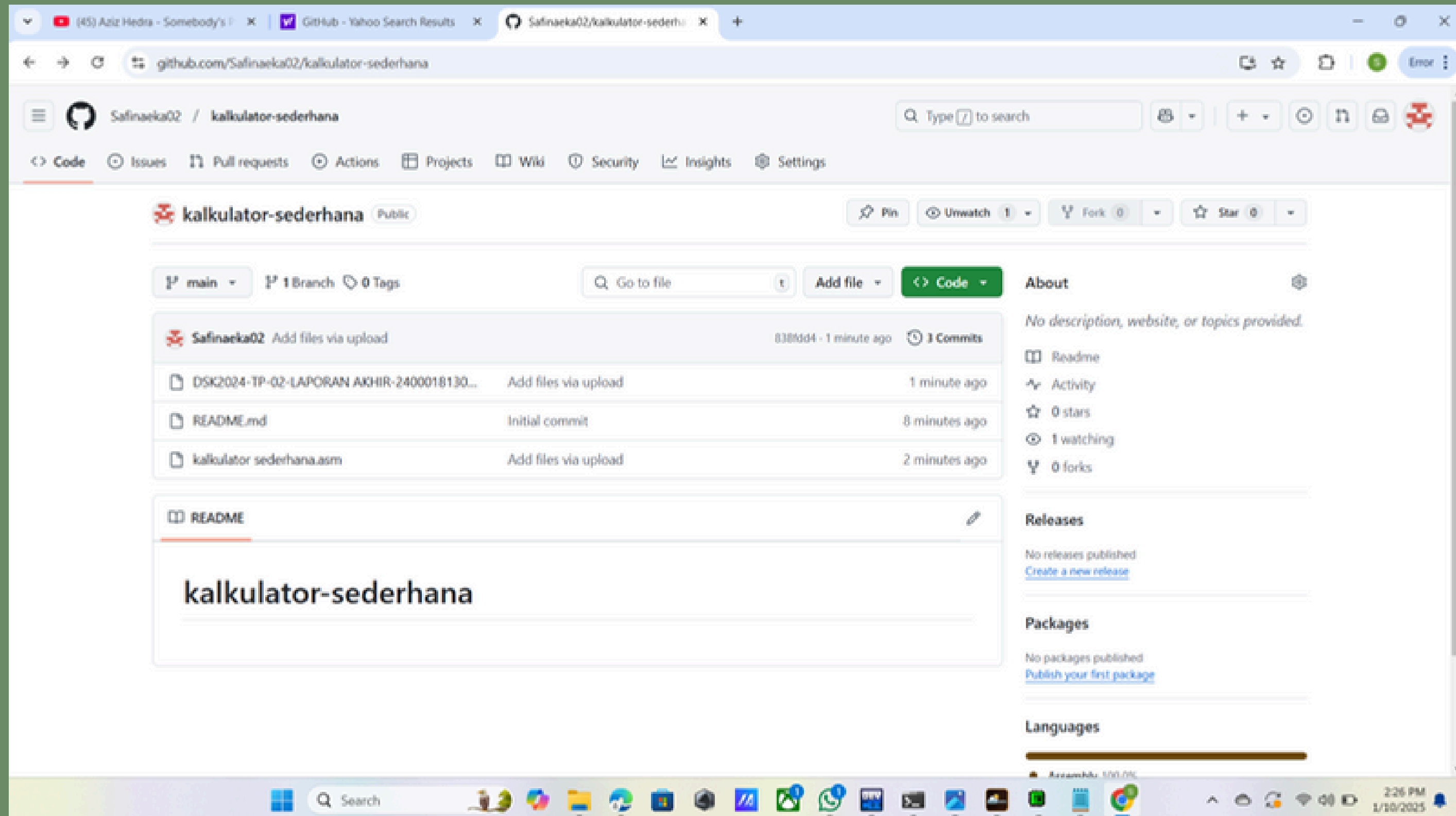
```
Kalkulator Sederhana:  
1. Penjumlahan  
2. Pengurangan  
3. Perkalian  
4. Pembagian  
Pilih operasi (1-4): 3  
Masukkan bilangan: 3  
Masukkan bilangan: 2  
Hasil: 3
```

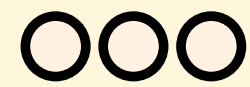
# OUTPUT PEMBAGIAN

 emulator screen (80x25 chars)

```
Kalkulator Sederhana:  
1. Penjumlahan  
2. Pengurangan  
3. Perkalian  
4. Pembagian  
Pilih operasi <1-4>: 4  
Masukkan bilangan: 5  
Masukkan bilangan: 5
```

# TAMPILAN GITHUB





## APA AJA SIH SARANNYA

### 1. Validasi Input

- o Tambahkan validasi untuk memastikan input berupa angka dan operator yang valid.
- o Tangani kasus seperti pembagian dengan nol agar program tidak mengalami error.

#### 1. Mendukung Angka Negatif dan Desimal

- o Perluas fungsionalitas kalkulator untuk mendukung angka negatif dan desimal.
- o Gunakan algoritma konversi untuk bilangan desimal ke binary floating-point jika diperlukan.

#### 1. Looping dan Navigasi Menu

- o Tambahkan opsi untuk menghitung ulang atau keluar dari program tanpa perlu menjalankan ulang emulator.
- o Gunakan logika branching untuk memberikan pengalaman pengguna yang lebih baik.





JANGAN LUPA BACA KESIMPULANNYA YA!



ooo

Kalkulator sederhana dalam bahasa assembly untuk emulator 8086 adalah program yang dirancang untuk memahami operasi dasar prosesor 8086, seperti pengolahan input, operasi aritmatika, dan menampilkan hasil menggunakan interrupt BIOS.

Kode ini memperlihatkan bagaimana memanfaatkan instruksi assembly dasar seperti ADD, SUB, MUL, dan DIV, serta bagaimana menggunakan interrupt INT 21h untuk interaksi input/output. Kalkulator ini cocok untuk pemula karena menyederhanakan logika pemrograman dalam konteks instruksi hardware langsung. Namun, kalkulator ini memiliki beberapa keterbatasan, seperti kurangnya validasi input, tidak mendukung angka negatif atau desimal, dan minimnya fitur penanganan kesalahan.





TERIMA KASIH

