TUGAS PEMBUATAN APLIKASI ASSEMBLY

MATA KULIAH DASAR SISTEM KOMPUTER

Nama : Nazwa Kiranty Syamsury

Nim : 2500018121

Kelas : C

Judul Aplikasi : Mobile Banking Sederhana

Ruang Lingkup M-Bank

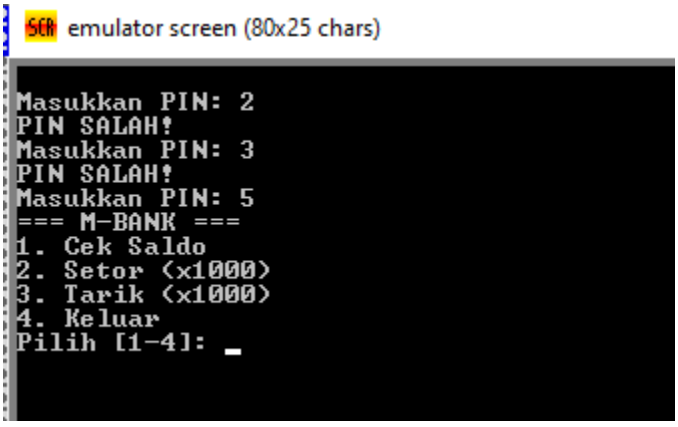
Ruang lingkup aplikasi M-Bank sederhana ini mencakup pengembangan sebuah sistem simulasi layanan perbankan dasar berbasis teks (console) yang dibuat menggunakan bahasa Assembly dan dijalankan pada lingkungan EMU8086. Aplikasi ini dirancang untuk melayani satu pengguna dengan mekanisme keamanan dasar berupa verifikasi PIN sebelum mengakses layanan utama. Setelah PIN berhasil diverifikasi, pengguna dapat mengakses menu utama yang menyediakan fitur cek saldo, setor uang, tarik uang, dan keluar dari aplikasi. Saldo disimpan sementara di memori selama program berjalan dan akan berubah sesuai dengan transaksi yang dilakukan pengguna. Aplikasi ini tidak mencakup fitur lanjutan seperti transfer antar rekening, multiuser, penyimpanan database, enkripsi data, riwayat transaksi, maupun koneksi jaringan, sehingga ruang lingkupnya terbatas pada pembelajaran logika program, pengelolaan data sederhana, serta penggunaan interupsi EMU8086 untuk input dan output.

1. Sketsa Alur Aplikasi M-Bank

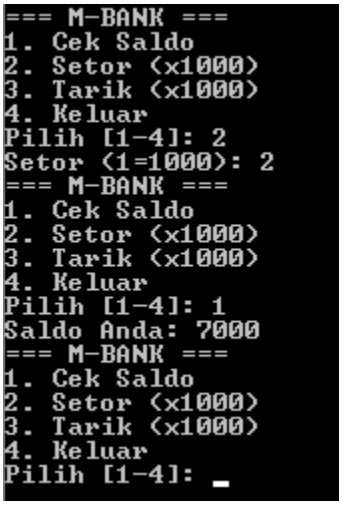
```
START
  |
  |-- Input PIN
  |      ├─ Jika salah → tampilkan "PIN SALAH" → ulangi
  |      └─ Jika benar → Menu Utama
  |
```

```
|-- Menu Utama
|      ├─ 1. Cek Saldo
|      ├─ 2. Setor Uang
|      ├─ 3. Tarik Uang
|      └─ 4. Keluar
|
|-- Proses sesuai pilihan
|
END
```

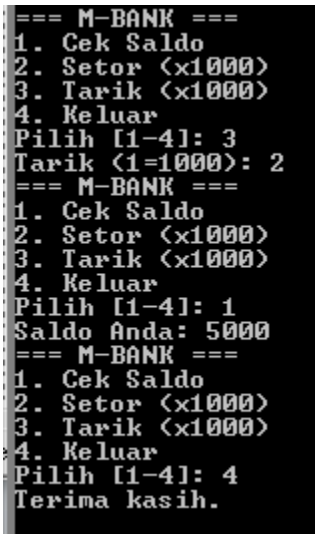
2. Gambar Tampilan Layar Aplikasi



Input pin terlebih dahulu, jika salah maka akan mengeluarkan output “PIN SALAH!”, setelah itu ouput yang keluar adalah menu aplikasinya, diminta input pilihan untuk lanjut ke proses.



Jika memilih 2 maka akan dimintaa input nominal angka yang akan dikalikan dengan 1000, setelah itu nominal tersebut akan tersimpan pada proses cek saldo dan jika memilih pilihan 1 maka saldo akan ditampilkan.



Jika memilih pilihan 3 sama dengan pilihan 2 meminta nominal yang akan dikalikan dengan 1000 dan dikurangkan dengan jumlah yang ada pada proses cek saldo, dan jika memilih pilihan terakhir yaitu 4 maka akan aplikasi bakal berhenti dan output terakhir adalah ucapan terima kasih.

3. Sketsa Struktur Program Assembly

```
DATA SEGMENT
    ├─ PIN
    ├─ SALDO
    ├─ PESAN / MENU
DATA ENDS

CODE SEGMENT
  ■ START
    ├─ INIT
    ├─ CEK_PIN
    ├─ MENU
    ├─ PROSES
```

- END

CODE ENDS

4. Code Assembly

```
.model small
.stack 100h
.data
    pin db '5'
    msgPin   db 13,10,"Masukkan PIN: $"
    msgSalah db 13,10,"PIN SALAH!$"
    menu db 13,10,"=== M-BANK ===",13,10
         db "1. Cek Saldo",13,10
         db "2. Setor (x1000)",13,10
         db "3. Tarik (x1000)",13,10
         db "4. Keluar",13,10
         db "Pilih [1-4]: $"

    msgSaldo db 13,10,"Saldo Anda: $"
    msgSetor db 13,10,"Setor (1=1000): $"
    msgTarik db 13,10,"Tarik (1=1000): $"
    msgGagal db 13,10,"Saldo tidak cukup!$"
    msgExit  db 13,10,"Terima kasih.$"
    saldo dw 5000       ; saldo awal = 5000
.code
main proc
    mov ax, @data
    mov ds, ax

; ===== LOGIN =====
login:
    mov ah, 09h
    lea dx, msgPin
    int 21h

    mov ah, 01h
    int 21h
```

```
    cmp al, pin
    jne salah
    jmp menu_utama


salah:
    mov ah, 09h
    lea dx, msgSalah
    int 21h
    jmp login

; ===== MENU =====
menu_utama:
    mov ah, 09h
    lea dx, menu
    int 21h

    mov ah, 01h
    int 21h

    cmp al, '1'
    je cek_saldo
    cmp al, '2'
    je setor
    cmp al, '3'
    je tarik
    cmp al, '4'
    je keluar
    jmp menu_utama

; ===== CEK SALDO =====
cek_saldo:
    mov ah, 09h
    lea dx, msgSaldo
    int 21h

    mov ax, saldo
    call tampil_angka
```

```
    jmp menu_utama



; ===== SETOR =====
setor:
    mov ah, 09h
    lea dx, msgSetor
    int 21h

    mov ah, 01h
    int 21h
    sub al, '0'     ; 1 digit

    mov ah, 0
    mov bx, 1000
    mul bx          ; AX = input × 1000

    add saldo, ax
    jmp menu_utama

; ===== TARIK =====
tarik:
    mov ah, 09h
    lea dx, msgTarik
    int 21h

    mov ah, 01h
    int 21h
    sub al, '0'

    mov ah, 0
    mov bx, 1000
    mul bx          ; AX = input × 1000

    cmp saldo, ax
    jb gagal
```

```
    sub saldo, ax
    jmp menu_utama

gagal:
    mov ah, 09h
    lea dx, msgGagal
    int 21h
    jmp menu_utama

; ===== KELUAR =====
keluar:
    mov ah, 09h
    lea dx, msgExit
    int 21h

    mov ah, 4Ch
    int 21h
main endp

; ===== TAMPIL ANGKA WORD =====
tampil_angka proc
    push ax
    push bx
    push cx
    push dx

    mov bx, 10
    xor cx, cx

ulang:
    xor dx, dx
    div bx
    push dx
    inc cx
    cmp ax, 0
    jne ulang

cetak:
```

```
    pop dx
    add dl, '0'
    mov ah, 02h
    int 21h
    loop cetak

    pop dx
    pop cx
    pop bx
    pop ax
    ret
tampil_angka endp

end main
```