**LAPORAN TUGAS PROJEK**

**DASAR SISTEM KOMPUTER BAHASA ASSEMBLY**

**“PROGRAM INPUT NILAI SEMESTER”**



NAMA : WISNU SYARIFUDIN AJI

KELAS : A

NIM : 2300018039

**UNIVERSITAS AHMAD DAHLAN**

**PROGRAM STUDI INFORMATIKA S1**

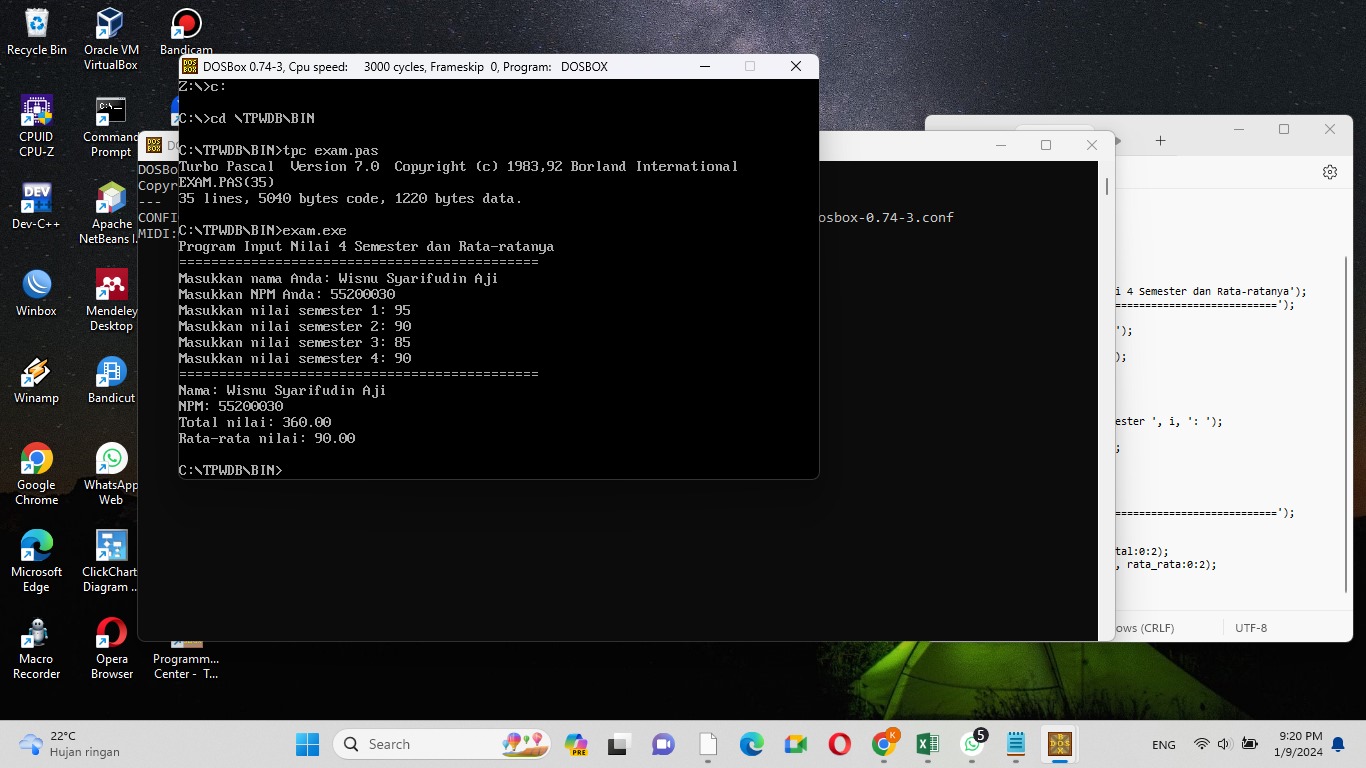
**FAKULTAS TEKNOLOGI INDUSTRI**

**TAHUN AKADEMIK 2023 / 2024**

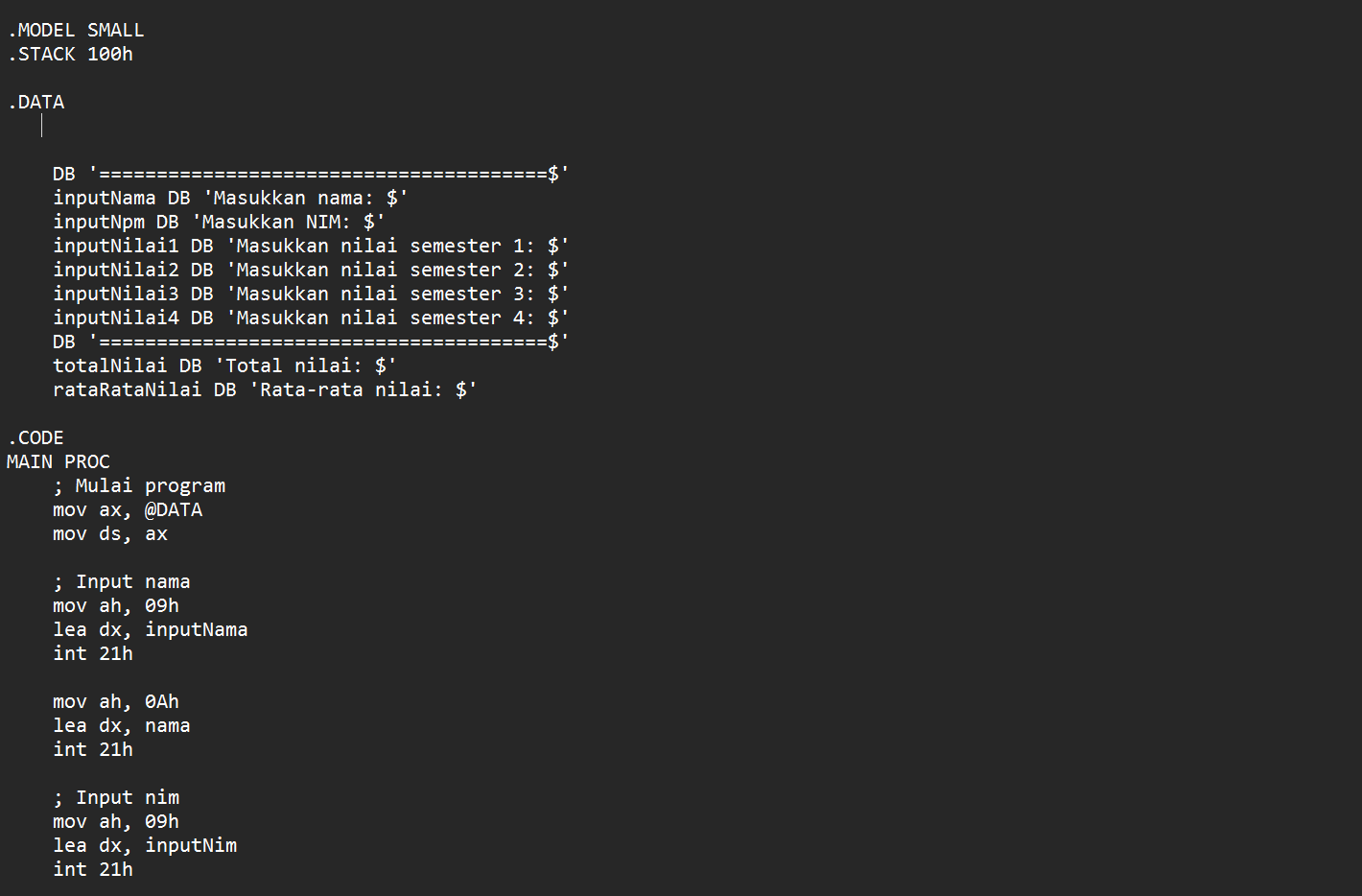
1. **Deskripsi Kerja**

Program yang dibuat yaitu *Program Input Nilai 4 semester dan rata-ratanya* yang memungkinkan pengguna menghitung menghitung total nilai dari 4 semester dan mengetahui hasil rata-ratanya. Pengguna diminta untuk memasukan nama dan juga Nomor Induk Mahasiswa(NIM). Kemudian Pengguna menginput niali secara urut dari semester satu sampai semester 4. Setelah input diterima, program akan menghitung total nilai dari keempat nilai tersebut dan akan keluar hasil rata-ratanya. Setelah hasil Muncul program pun selesai.

1. **Desain Atau Sketsa Dari Antarmuka Yang Digunakan**



1. **Kode Program Assembly Untuk Menampilkan Antarmuka Yang Telah Dibuat di file .txt(Notepad)**

****

Penjelasan syntax :

1. .MODEL SMALL untuk Menunjukkan bahwa program ini menggunakan model memori SMALL.
2. STACK 100h digunakan untuk menentukan ukuran stack untuk program.
3. DATA merupakan bagian untuk mendefinisikan segmen data program.
4. DB '=======================================$': Ini adalah instruksi untuk mendefinisikan string atau data byte (DB). Baris ini mendefinisikan string yang terdiri dari tanda "=" yang panjangnya 40 karakter, diikuti oleh tanda "$", digunakan untuk membuat garis pemisah atau header dalam output
5. inputNama DB 'Masukkan nama: $': Ini mendefinisikan string 'Masukkan nama: $' dan menyimpannya dalam variabel inputNama, digunakan untuk meminta pengguna memasukkan nama.
6. inputNpm DB 'Masukkan NIM: $' mendefinisikan string 'Masukkan NIM: $' dan menyimpannya dalam variabel inputNim, digunakan untuk meminta pengguna memasukkan NIM.
7. inputNilai1 DB 'Masukkan nilai semester 1: $': Mendefinisikan string 'Masukkan nilai semester 1: $' dan menyimpannya dalam variabel inputNilai1, digunakan untuk meminta pengguna memasukkan nilai semester 1.
8. Untuk “inputNilai2 DB 'Masukkan nilai semester 2: $'”, “inputNilai3 DB 'Masukkan nilai semester 3: $'”, “inputNilai4 DB 'Masukkan nilai semester 4: $'” ketiga syntax tersebyt memiliki fungsi yang sama yaitu untuk memberikan informasi tentang rata-rata nilai setelah pengguna memasukkan nilai-nilai.
9. MAIN PROC label yang menandakan awal dari sebuah prosedur utama (main procedure)
10. mov ax, @DATA: Ini adalah instruksi yang digunakan untuk memindahkan nilai ke dalam register ax. @DATA adalah segmen data default yang digunakan untuk menyimpan data dalam program. Jadi, baris ini memuat alamat segmen data ke dalam register ax
11. mov ds, ax: Instruksi ini memindahkan nilai dari register ax ke dalam register ds. Register ds (Data Segment) adalah register yang menyimpan alamat segmen data. Jadi, baris ini mengatur register ds agar menunjuk ke segmen data yang telah diinisialisasi sebelumnya.
12. Input nama

mov ah, 09h

lea dx, inputNama

int 21h

mov ah, 0Ah

lea dx, nama

int 21h

; Input nim

mov ah, 09h

lea dx, inputNim

int 21h

mov ah, 0Ah

lea dx, npm

int 21h

Penjelasan kode diatas diurutkan dari baris teratas :

* mov ah, 09h: Instruksi ini mengatur nilai register ah dengan 09h, yang merupakan kode layanan untuk mencetak string ke layar menggunakan fungsi DOS. Ini digunakan untuk mencetak prompt atau pesan kepada pengguna.
* lea dx, inputNama: Instruksi ini menggunakan lea (Load Effective Address) untuk memuat alamat dari string inputNama ke dalam register dx. Ini menyiapkan alamat string yang akan dicetak.
* int 21h: Instruksi ini memanggil fungsi DOS melalui interrupt 21h dengan kode layanan yang telah diatur sebelumnya. Dalam hal ini, 09h digunakan untuk mencetak string ke layar. Sebagai hasilnya, pesan "Masukkan nama: $" akan ditampilkan.
* mov ah, 0Ah: Instruksi ini mengatur nilai register ah dengan 0Ah, yang merupakan kode layanan untuk membaca string dari masukan pengguna. Ini akan digunakan untuk membaca input nama yang dimasukkan oleh pengguna.
* lea dx, nama: Instruksi ini memuat alamat dari variabel nama ke dalam register dx. Ini menyiapkan tempat di mana string yang dimasukkan oleh pengguna akan disimpan.

1. ; input nilai semester 1

mov ax, 4

mov bx, 1

mov cx, nilai1\_prompt

mov dx, nilai1\_prompt\_len

int 21h

mov ax, 3

mov bx, 0

lea cx, [nilai1]

mov dx, 4

int 21h

Penjelasan kode diatas diurutkan dari baris teratas :

* mov ax, 4: Instruksi ini mengatur register ax dengan nilai 4, yang merupakan kode layanan DOS untuk mencetak string ke layar.
* mov bx, 1: Instruksi ini mengatur register bx dengan nilai 1. Nilai ini biasanya digunakan untuk menentukan handle layar standar.
* mov cx, nilai1\_prompt: Instruksi ini mengatur register cx dengan alamat string nilai1\_prompt. Ini adalah string prompt yang akan dicetak untuk meminta input nilai semester 1.
* mov dx, nilai1\_prompt\_len: Instruksi ini mengatur register dx dengan panjang string prompt nilai1\_prompt. Ini diperlukan oleh layanan DOS untuk mengetahui berapa karakter yang akan dicetak.
* int 21h: Instruksi ini memanggil layanan DOS melalui interrupt 21h dengan kode layanan 4. Ini akan mencetak string prompt ke layar.
* mov ax, 3: Instruksi ini mengatur register ax dengan nilai 3, yang merupakan kode layanan DOS untuk membaca string dari masukan pengguna.
* mov bx, 0: Instruksi ini mengatur register bx dengan nilai 0, yang menandakan standar input.
* lea cx, [nilai1]: Instruksi ini menggunakan lea (Load Effective Address) untuk memuat alamat dari variabel nilai1 ke dalam register cx. Ini menyiapkan tempat di mana string yang dimasukkan oleh pengguna akan disimpan.
* mov dx, 4: Instruksi ini mengatur register dx dengan panjang maksimal string yang dapat dibaca. Dalam hal ini, panjang maksimal adalah 4 karakter.

1. Untuk input nilai semester dua sampai semester 4 sama seperti semester satu.
2. ; Menghitung rata-rata nilai

;Hitung total

mov ax, dword [nilai1]

add ax, dword [nilai2]

add ax, dword [nilai3]

add ax, dword [nilai4]

mov [total], ax

mov ax, dword [total]

mov bx, 4

int21h

Penjelasan kode diatas diurutkan dari baris teratas :

* mov ax, [nilai1]: Memuat nilai dari variabel nilai1 ke dalam register ax.
* add ax, [nilai2]: Menambahkan nilai dari variabel nilai2 ke dalam register ax
* add ax, [nilai3]: Menambahkan nilai dari variabel nilai3 ke dalam register ax
* add ax, [nilai4]: Menambahkan nilai dari variabel nilai4 ke dalam register ax.
* mov [total], ax: Menyimpan hasil penjumlahan ke dalam variabel total.
* mov ax, [total]: Memuat nilai total kembali ke dalam register ax.
* mov bx, 4: Mengatur register bx dengan nilai 4, yang akan digunakan sebagai pembagi.
* idiv bx: Instruksi ini melakukan pembagian nilai di register ax dengan nilai di register bx. Hasil pembagian (rata-rata) disimpan di register ax, dan sisa pembagian (tidak digunakan dalam konteks ini) disimpan di register dx.

1. ; Output rata-rata

mov ax, 4

mov bx, 1

mov cx, rata\_rata

mov dx, 4

int 21h

Penjelasan kode diatas diurutkan dari baris teratas :

* mov ax, 4: Memuat nilai 4 ke dalam register ax.
* mov bx, 1: Memuat nilai 1 ke dalam register bx. Nilai 1 menunjukkan file descriptor standar output (stdout), yang digunakan untuk menampilkan teks ke layar.
* mov cx, rata\_rata: Memuat nilai yang akan dicetak (rata\_rata) ke dalam register cx. Ini menentukan nilai rata-rata yang akan ditampilkan.
* mov dx, 4: Memuat nilai 4 ke dalam register dx. Ini menentukan panjang data yang akan dicetak, dalam hal ini, diasumsikan bahwa rata\_rata adalah integer 4-byte.
* int 21h: Instruksi ini adalah interrupt DOS yang digunakan untuk memanggil layanan sistem. Dengan parameter yang telah diatur sebelumnya, di sini digunakan untuk mencetak nilai rata-rata ke layar.

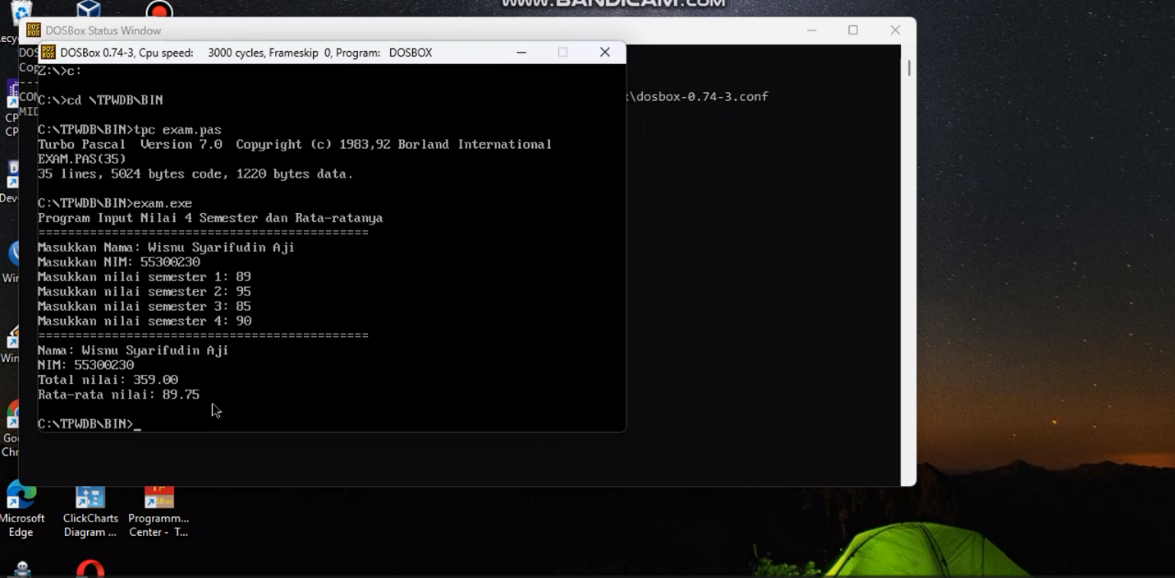
1. ; Selesai program

mov ah, 4Ch, mov ah, 4Ch: Instruksi ini memuat nilai hexadecimal 4Ch ke dalam register ah. Nilai 4Ch adalah kode layanan DOS yang menunjukkan bahwa program ingin keluar dan mengakhiri eksekusi.

int 21h, untuk mengeksekusi program.

1. MAIN ENDP, MAIN ENDP: Ini menandakan akhir dari suatu prosedur. ENDP singkatan dari "End Procedure". Dalam hal ini, ini menunjukkan bahwa prosedur utama (MAIN) telah selesai.

END MAIN, END MAIN: Ini menandakan akhir dari program Assembly. Dalam hal ini, program diakhiri dan ini adalah titik akhir dari kode Assembly. MAIN adalah label atau nama yang diberikan kepada prosedur utama, dan END MAIN mengakhiri program setelah prosedur utama selesai dieksekusi.

1. ****

Penjelasan :

Ketika di RUN atau Emulate Program akan muncul kemudian pengguna atau user memasukkan nama dan juga nomor induk mahasiswa(NIM). Kemudian aka nada perintah memasukan nilai semester dimulai dari semester satu sampai dengan semester empat. Nilai akan ditotal dan juga akan menghitung rata-rata nilainya. Terakhir program akan memunculkan output nama,nim,nilai semester satu sampai empat,total nilainya dan juga hasil akhir rata-ratanya.