**DASAR SISTEM KOMPUTER**

**“JUDUL APLIKASI: PROGRAM PENILAIAN NILAI MAHASISWA DENGAN ID DAN MENGURUTKANNYA BERDASARKAN NILAI**

Sebuah gambar berisi lambang, simbol, Merek dagang, logo

Deskripsi dibuat secara otomatis

**Disusun Oleh:**

**NAMA: MUHAMMAD NAUFAL AFRIZA**

**NIM : 2300018088**

**KELAS: B**

**UNIVERSITAS AHMAD DAHLAN**

**FAKULTAS TEKNOLOGI INDUSTRI**

**PROGRAM STUDI INFORMATIKA**

**TAHUN 2023**

1. DEKSRIPSI CARA KERJA APLIKASI:

1. Pengguna diminta untuk memasukkan jumlah mahasiswa dengan ketentuan batas tidak boleh lebih dari 1000.
2. Pengguna diminta untuk memasukkan ID mahasiswa berupa 3 digit akhir dari nim mahasiswa.
3. Pengguna diminta untuk memasukkan nilai mahasiswa.
4. Kemudian nilai tersebut akan diurutkan berdasarkan nilai (tertinggi – terendah).
5. Kemudian hasil output akan keluar berupa nilai yang telah di urutkan beserta dengan ID dari mahasiswa yang bersangkutan.

Sebuah gambar berisi teks, diagram, sketsa, gambar

Deskripsi dibuat secara otomatis

2. SKETSA ANTARMUKA APLIKASI

Sebuah gambar berisi teks, cuplikan layar, Font, aljabar

Deskripsi dibuat secara otomatis

**Penjelasan:** Antarmuka berupa text dan simbol, dan kompatibel dengan program assembly pada arsitektur x86, dengan inputan user pada setiap masukannya, dan menghasilkan output pada bagian bawah dengan susunan format ID dan Nilai Mahasiswa yang telah di urutkan.

3. KODE PROGRAM

include 'emu8086.inc'

JMP START

DATA SEGMENT

N DW ?

MARKS DB 1000 DUP (?) *; 1000 ADALAH JUMLAH MAKSIMUM MAHASISWA*

ID DB 1000 DUP (?)

MSG1 DB 'MASUKKAN JUMLAH MAHASISWA (tidak boleh lebih dari 1000): ', 0

MSG2 DB 0Dh, 0Ah, 0Dh, 0Ah, 'MASUKKAN ID MAHASISWA: ', 0

MSG3 DB 0Dh, 0Ah, 0Dh, 0Ah, 'MASUKKAN NILAI MAHASISWA: ', 0

HR DB 0Dh, 0Ah, 0Dh, 0Ah, '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*NILAI MAHASISWA YANG DIURUTKAN\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*', 0

MSG4 DB 0Dh, 0Ah, 0Dh, 0Ah, 'ID: ', 09H, 'NILAI:', 0

DATA ENDS

CODE SEGMENT

ASSUME DS:DATA CS:CODE

START:

MOV AX, DATA

MOV DS, AX

DEFINE\_SCAN\_NUM

DEFINE\_PRINT\_STRING

DEFINE\_PRINT\_NUM

DEFINE\_PRINT\_NUM\_UNS

*; BACA JUMLAH MAHASISWA*

LEA SI, MSG1

CALL PRINT\_STRING

CALL SCAN\_NUM

MOV N, CX

*; BACA ID MAHASISWA*

LEA SI, MSG2

CALL PRINT\_STRING

MOV SI, 0

LOOP1:

CALL SCAN\_NUM

MOV ID[SI], CL

INC SI

PRINT 0AH

PRINT 0DH

CMP SI, N

JNE LOOP1

*; BACA NILAI MAHASISWA*

LEA SI, MSG3

CALL PRINT\_STRING

MOV SI, 0

LOOP2:

CALL SCAN\_NUM

MOV MARKS[SI], CL

INC SI

PRINT 0AH

PRINT 0DH

CMP SI, N

JNE LOOP2

*; URUTKAN BERDASARKAN NILAI MENGGUNAKAN BUBBLE SORT*

DEC N

MOV CX, N

OUTER:

MOV SI, 0

INNER:

MOV AL, MARKS[SI]

MOV DL, ID[SI]

INC SI

CMP MARKS[SI], AL

JB SKIP

XCHG AL, MARKS[SI]

MOV MARKS[SI-1], AL

XCHG DL, ID[SI]

MOV ID[SI-1], DL

SKIP:

CMP SI, CX

JL INNER

LOOP OUTER

INC N

*; CETAK TABEL ID DAN NILAI SETELAH DIURUTKAN*

LEA SI, HR

CALL PRINT\_STRING

LEA SI, MSG4

CALL PRINT\_STRING

PRINT 0AH

PRINT 0DH

MOV SI, 0

LOOP3:

MOV AX, 0

MOV AL, ID[SI]

CALL PRINT\_NUM\_UNS

PRINT 09H

MOV AL, MARKS[SI]

CALL PRINT\_NUM\_UNS

PRINT 0AH

PRINT 0DH

INC SI

CMP SI, N

JNE LOOP3

CODE ENDS

END START

ret

**Penjelasan:**

Dengan menggunakan Kode program assembly, kode program tersebut akan menghasilkan antarmuka seperti yang telah dirancang pada slide sebelumnya, dengan kombinasi text dan simbol.

Sebuah gambar berisi teks, cuplikan layar, Font, aljabar

Deskripsi dibuat secara otomatis

4. KODE PROGRAM

include 'emu8086.inc'

JMP START

DATA SEGMENT

N DW ?

MARKS DB 1000 DUP (?) *; 1000 ADALAH JUMLAH MAKSIMUM MAHASISWA*

ID DB 1000 DUP (?)

MSG1 DB 'MASUKKAN JUMLAH MAHASISWA (tidak boleh lebih dari 1000): ', 0

MSG2 DB 0Dh, 0Ah, 0Dh, 0Ah, 'MASUKKAN ID MAHASISWA: ', 0

MSG3 DB 0Dh, 0Ah, 0Dh, 0Ah, 'MASUKKAN NILAI MAHASISWA: ', 0

HR DB 0Dh, 0Ah, 0Dh, 0Ah, '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*NILAI MAHASISWA YANG DIURUTKAN\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*', 0

MSG4 DB 0Dh, 0Ah, 0Dh, 0Ah, 'ID: ', 09H, 'NILAI:', 0

DATA ENDS

CODE SEGMENT

ASSUME DS:DATA CS:CODE

START:

MOV AX, DATA

MOV DS, AX

DEFINE\_SCAN\_NUM

DEFINE\_PRINT\_STRING

DEFINE\_PRINT\_NUM

DEFINE\_PRINT\_NUM\_UNS

*; BACA JUMLAH MAHASISWA*

LEA SI, MSG1

CALL PRINT\_STRING

CALL SCAN\_NUM

MOV N, CX

*; BACA ID MAHASISWA*

LEA SI, MSG2

CALL PRINT\_STRING

MOV SI, 0

LOOP1:

CALL SCAN\_NUM

MOV ID[SI], CL

INC SI

PRINT 0AH

PRINT 0DH

CMP SI, N

JNE LOOP1

*; BACA NILAI MAHASISWA*

LEA SI, MSG3

CALL PRINT\_STRING

MOV SI, 0

LOOP2:

CALL SCAN\_NUM

MOV MARKS[SI], CL

INC SI

PRINT 0AH

PRINT 0DH

CMP SI, N

JNE LOOP2

*; URUTKAN BERDASARKAN NILAI MENGGUNAKAN BUBBLE SORT*

DEC N

MOV CX, N

OUTER:

MOV SI, 0

INNER:

MOV AL, MARKS[SI]

MOV DL, ID[SI]

INC SI

CMP MARKS[SI], AL

JB SKIP

XCHG AL, MARKS[SI]

MOV MARKS[SI-1], AL

XCHG DL, ID[SI]

MOV ID[SI-1], DL

SKIP:

CMP SI, CX

JL INNER

LOOP OUTER

INC N

*; CETAK TABEL ID DAN NILAI SETELAH DIURUTKAN*

LEA SI, HR

CALL PRINT\_STRING

LEA SI, MSG4

CALL PRINT\_STRING

PRINT 0AH

PRINT 0DH

MOV SI, 0

LOOP3:

MOV AX, 0

MOV AL, ID[SI]

CALL PRINT\_NUM\_UNS

PRINT 09H

MOV AL, MARKS[SI]

CALL PRINT\_NUM\_UNS

PRINT 0AH

PRINT 0DH

INC SI

CMP SI, N

JNE LOOP3

CODE ENDS

END START

ret

**PENJELASAN MENGENAI KODE PROGRAM:**

**Awal Program**

include 'emu8086.inc'

JMP START

DATA SEGMENT

N DW ?

MARKS DB 1000 DUP (?) ; 1000 ADALAH JUMLAH MAKSIMUM MAHASISWA

ID DB 1000 DUP (?)

MSG1 DB 'MASUKKAN JUMLAH MAHASISWA (tidak boleh lebih dari 1000): ', 0

MSG2 DB 0Dh, 0Ah, 0Dh, 0Ah, 'MASUKKAN ID MAHASISWA: ', 0

MSG3 DB 0Dh, 0Ah, 0Dh, 0Ah, 'MASUKKAN NILAI MAHASISWA: ', 0

HR DB 0Dh, 0Ah, 0Dh, 0Ah, '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*NILAI MAHASISWA YANG DIURUTKAN\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*', 0

MSG4 DB 0Dh, 0Ah, 0Dh, 0Ah, 'ID: ', 09H, 'NILAI:', 0

DATA ENDS

CODE SEGMENT

ASSUME DS:DATA CS:CODE

START:

MOV AX, DATA

MOV DS, AX

DEFINE\_SCAN\_NUM

DEFINE\_PRINT\_STRING

DEFINE\_PRINT\_NUM

DEFINE\_PRINT\_NUM\_UNS

* **include 'emu8086.inc'**: Menggunakan file header **emu8086.inc** yang kemungkinan berisi definisi-informasi spesifik untuk emulator atau assembler tertentu.
* **JMP START**: Melompat ke label **START** untuk memulai eksekusi program.
* Bagian **DATA SEGMENT** mendefinisikan segmen data, termasuk variabel N, array MARKS, array ID, dan beberapa string pesan.
* Bagian **CODE SEGMENT** menginisialisasi segmen kode dan data serta mendefinisikan label **START**.

**Sesi 1: Baca Jumlah Mahasiswa**

; BACA JUMLAH MAHASISWA

LEA SI, MSG1 ; Memuat alamat pesan string MSG1 ke dalam SI

CALL PRINT\_STRING ; Mencetak pesan untuk memasukkan jumlah mahasiswa

CALL SCAN\_NUM ; Memanggil rutin untuk membaca nilai numerik (jumlah mahasiswa)

MOV N, CX ; Menyimpan nilai yang dibaca (jumlah mahasiswa) ke dalam variabel N

* **LEA SI, MSG1**: Memuat alamat pesan string **MSG1** (pesan untuk memasukkan jumlah mahasiswa) ke dalam SI.
* **CALL PRINT\_STRING**: Memanggil rutin untuk mencetak string yang diarahkan oleh SI (pesan untuk memasukkan jumlah mahasiswa).
* **CALL SCAN\_NUM**: Memanggil rutin untuk membaca nilai numerik dari input pengguna dan menyimpannya di dalam register CX. Nilai ini kemudian disalin ke dalam variabel **N** menggunakan instruksi **MOV N, CX**.

**Sesi 2: Baca ID Mahasiswa**

; BACA ID SISWA

LEA SI, MSG2 ; Memuat alamat pesan string MSG2 ke dalam SI

CALL PRINT\_STRING ; Mencetak pesan untuk memasukkan ID mahasiswa

MOV SI, 0 ; Menginisialisasi SI dengan 0 (indeks array)

* **LEA SI, MSG2**: Memuat alamat pesan string **MSG2** (pesan untuk memasukkan ID mahasiswa) ke dalam SI.
* **CALL PRINT\_STRING**: Memanggil rutin untuk mencetak string yang diarahkan oleh SI (pesan untuk memasukkan ID mahasiswa).
* **MOV SI, 0**: Menginisialisasi SI dengan 0, yang kemungkinan akan digunakan sebagai indeks untuk array ID mahasiswa.

**Sesi 3: Loop Membaca ID Mahasiswa**

LOOP1:

CALL SCAN\_NUM ; Membaca ID mahasiswa dan menyimpannya dalam register CL

MOV ID[SI], CL ; Menyimpan ID mahasiswa di dalam array ID

INC SI ; Meningkatkan nilai SI untuk mengakses elemen berikutnya dalam array

PRINT 0AH ; Mencetak newline (baris baru) - mungkin untuk tata letak yang baik

PRINT 0DH ; Mencetak carriage return - mungkin untuk tata letak yang baik

CMP SI, N ; Membandingkan SI dengan N (jumlah mahasiswa)

JNE LOOP1 ; Melompat ke LOOP1 jika SI belum mencapai N

* **CALL SCAN\_NUM**: Memanggil rutin untuk membaca nilai numerik dari input pengguna dan menyimpannya dalam register CL. Ini kemungkinan adalah ID mahasiswa.
* **MOV ID[SI], CL**: Menyimpan nilai ID mahasiswa (yang telah dibaca) dalam array ID pada indeks yang ditunjukkan oleh SI.
* **INC SI**: Meningkatkan nilai SI, sehingga SI sekarang menunjukkan ke elemen berikutnya dalam array ID.
* **PRINT 0AH** dan **PRINT 0DH**: Mencetak newline (baris baru) dan carriage return untuk tata letak yang baik.
* **CMP SI, N**: Membandingkan nilai SI dengan N (jumlah mahasiswa).
* **JNE LOOP1**: Jika SI belum mencapai N, program akan melompat kembali ke label LOOP1 untuk membaca ID mahasiswa berikutnya.

**Sesi 4: Loop Membaca Nilai Mahasiswa**

; BACA NILAI MAHASISWA

LEA SI, MSG3 ; Memuat alamat pesan string MSG3 ke dalam SI

CALL PRINT\_STRING ; Mencetak pesan untuk memasukkan nilai mahasiswa

MOV SI, 0 ; Menginisialisasi SI dengan 0 (indeks array)

* **LEA SI, MSG3**: Memuat alamat pesan string **MSG3** (pesan untuk memasukkan nilai mahasiswa) ke dalam SI.
* **CALL PRINT\_STRING**: Memanggil rutin untuk mencetak string yang diarahkan oleh SI (pesan untuk memasukkan nilai mahasiswa).
* **MOV SI, 0**: Menginisialisasi SI dengan 0, yang kemungkinan akan digunakan sebagai indeks untuk array nilai mahasiswa (**MARKS**).

**Sesi 5: Loop Membaca Nilai Mahasiswa (Lanjutan)**

LOOP2:

CALL SCAN\_NUM ; Membaca nilai mahasiswa dan menyimpannya dalam register CL

MOV MARKS[SI], CL ; Menyimpan nilai mahasiswa di dalam array MARKS

INC SI ; Meningkatkan nilai SI untuk mengakses elemen berikutnya dalam array

PRINT 0AH ; Mencetak newline (baris baru) - mungkin untuk tata letak yang baik

PRINT 0DH ; Mencetak carriage return - mungkin untuk tata letak yang baik

CMP SI, N ; Membandingkan SI dengan N (jumlah mahasiswa)

JNE LOOP2 ; Melompat ke LOOP2 jika SI belum mencapai N

* **CALL SCAN\_NUM**: Memanggil rutin untuk membaca nilai numerik dari input pengguna dan menyimpannya dalam register CL. Ini kemungkinan adalah nilai mahasiswa.
* **MOV MARKS[SI], CL**: Menyimpan nilai mahasiswa (yang telah dibaca) dalam array MARKS pada indeks yang ditunjukkan oleh SI.
* **INC SI**: Meningkatkan nilai SI, sehingga SI sekarang menunjukkan ke elemen berikutnya dalam array MARKS.
* **PRINT 0AH** dan **PRINT 0DH**: Mencetak newline (baris baru) dan carriage return untuk tata letak yang baik.
* **CMP SI, N**: Membandingkan nilai SI dengan N (jumlah mahasiswa).
* **JNE LOOP2**: Jika SI belum mencapai N, program akan melompat kembali ke label LOOP2 untuk membaca nilai mahasiswa berikutnya.

**Sesi 6: Bubble Sort untuk Mengurutkan Nilai Mahasiswa**

; URUTKAN BERDASARKAN NILAI MENGGUNAKAN BUBBLE SORT

DEC N ; Mengurangkan N untuk mengakses array dari 0 hingga N-1

MOV CX, N ; Menyalin nilai N ke dalam register CX

OUTER:

MOV SI, 0

INNER:

MOV AL, MARKS[SI]

MOV DL, ID[SI]

INC SI

CMP MARKS[SI], AL

JB SKIP

XCHG AL, MARKS[SI]

MOV MARKS[SI-1], AL

XCHG DL, ID[SI]

MOV ID[SI-1], DL

SKIP:

CMP SI, CX

JL INNER

LOOP OUTER

INC N ; Menambahkan N setelah proses Bubble Sort selesai

* **DEC N**: Mengurangkan nilai **N** untuk mengakses array dari 0 hingga N-1.
* **MOV CX, N**: Menyalin nilai **N** ke dalam register CX untuk digunakan dalam loop **OUTER**.
* Loop **OUTER**: Loop utama untuk melakukan Bubble Sort. Menggunakan loop **INNER** untuk membandingkan dan menukar nilai-nilai.
* **XCHG AL, MARKS[SI]**: Instruksi ini bertanggung jawab untuk menukar nilai yang dibandingkan dan disimpan dalam register AL dengan nilai pada indeks yang ditunjukkan oleh SI dalam array MARKS.
* **XCHG DL, ID[SI]**: Instruksi serupa untuk menukar nilai ID mahasiswa.
* **CMP SI, CX** dan **JL INNER**: Melakukan perbandingan dan melompat ke label **INNER** jika SI masih kurang dari CX.
* **LOOP OUTER**: Loop ini akan terus berjalan selama register CX tidak mencapai 0. Loop ini akan mengulang proses Bubble Sort sebanyak N kali.

**Sesi 7: Cetak Tabel ID dan Nilai yang Sudah Diurutkan**

; CETAK TABEL ID DAN NILAI SETELAH DIURUTKAN

LEA SI, HR ; Memuat alamat pesan string HR ke dalam SI

CALL PRINT\_STRING ; Mencetak pesan header

LEA SI, MSG4 ; Memuat alamat pesan string MSG4 ke dalam SI

CALL PRINT\_STRING ; Mencetak pesan tabel

PRINT 0AH ; Mencetak newline (baris baru)

PRINT 0DH ; Mencetak carriage return

MOV SI, 0

LOOP3:

MOV AX, 0

MOV AL, ID[SI] ; Mengambil ID mahasiswa dari array ID

CALL PRINT\_NUM\_UNS ; Mencetak ID mahasiswa

PRINT 09H ; Mencetak tab sebagai pemisah antara ID dan nilai

MOV AL, MARKS[SI] ; Mengambil nilai mahasiswa dari array MARKS

CALL PRINT\_NUM\_UNS ; Mencetak nilai mahasiswa

PRINT 0AH ; Mencetak newline (baris baru)

PRINT 0DH ; Mencetak carriage return

INC SI ; Meningkatkan nilai SI untuk mengakses elemen berikutnya

CMP SI, N ; Membandingkan SI dengan N (jumlah mahasiswa)

JNE LOOP3 ; Melompat ke LOOP3 jika SI belum mencapai N

* **LEA SI, HR**: Memuat alamat pesan string **HR** (pesan header) ke dalam SI.
* **CALL PRINT\_STRING**: Memanggil rutin untuk mencetak string yang diarahkan oleh SI (pesan header).
* **LEA SI, MSG4**: Memuat alamat pesan string **MSG4** (pesan tabel) ke dalam SI.
* **CALL PRINT\_STRING**: Memanggil rutin untuk mencetak string yang diarahkan oleh SI (pesan tabel).
* **PRINT 0AH** dan **PRINT 0DH**: Mencetak newline (baris baru) dan carriage return untuk tata letak yang baik.
* Loop **LOOP3**: Loop untuk mencetak ID dan nilai mahasiswa yang sudah diurut

**Akhir Program**

CODE ENDS

END START

ret

* **CODE ENDS**: Menutup segmen kode.
* **END START**: Menandakan akhir program.
* **ret**: Instruksi **ret** digunakan untuk mengembalikan kendali dari suatu fungsi.

5. HASIL PROGRAM

Sebuah gambar berisi software, cuplikan layar, Software multimedia, teks

Deskripsi dibuat secara otomatis

Sebuah gambar berisi teks, software, Software multimedia, cuplikan layar

Deskripsi dibuat secara otomatis

**Penjelalasan :** Dengan jumlah mahaiswa sebanyak 3 orang dan dengan ID mahasiswa serta nilai mahasiswa, menghasilkan nilai yang telah di urutkan dari tertinggi – terendah dengan disertai nim mahasiswa yang bersangkutan.

Sebuah gambar berisi teks, software, Software multimedia, cuplikan layar

Deskripsi dibuat secara otomatis

**Penjelalasan :** Dengan jumlah mahaiswa sebanyak 5 orang dan dengan ID mahasiswa serta nilai mahasiswa, menghasilkan nilai yang telah di urutkan dari tertinggi – terendah dengan disertai nim mahasiswa yang bersangkutan.