TUGAS AKHIR

Program Aplikasi Mini Game Tebak Angka

Dosen Pengampu:Ali Tarmuji,S.T,M.M.Cs.

Disusun oleh:

Nama:Fatih Akmal Hidayat

NIM : 2400018135

KELAS : C

Link Github:

<https://github.com/fatihakmal15/fatihakmalhidayat_mini-game-tebak-angka>

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS AHMAD DAHLAN

2024/2025

BAB 1: PENDAHULUAN

1.1 Latar Belakang

Aplikasi Mini Game Tebak Angka ini adalah aplikasi yang dimana pengguna akan diminta untuk menebak angka yang telah ditentukan oleh program.Program akan memberikan petunjuk apakah tebakan pengguna terlalu tinggi atau terlalu rendah hingga pengguna berhasil menebak angka rahasia(angka yang benar).

Program menggunakan bahasa Assembly yang memungkinkan kontrol langsung terhadap perangkat keras dan sumber daya sistem, memberikan efisiensi tinggi dan kemampuan untuk berinteraksi langsung dengan CPU.

1.2 Tujuan

1. Interaktif dan Edukatif: Program ini dibuat untuk memberikan pengalaman interaktif kepada pengguna sambil mempelajari bahasa Assembly.

2. Latihan Pemrograman: Memberikan latihan praktis dalam penulisan dan debugging program Assembly.

3. Penggunaan Interrupts: Menunjukkan bagaimana menggunakan interrupts untuk menangani input/output di sistem operasi DOS.

4. Manipulasi Data: Menyediakan contoh manipulasi data sederhana, seperti membaca input dari pengguna, membandingkan angka, dan menampilkan hasil.

5. Kontrol Alur: Memperkenalkan pengguna pada kontrol alur dalam Assembly, termasuk penggunaan jump dan loop untuk mengelola logika program.

6. Validasi Input: Menampilkan bagaimana memvalidasi input pengguna untuk memastikan bahwa angka yang dimasukkan berada dalam rentang yang diharapkan.

BAB II:DASAR TEORI

2.1 Bahasa Assembly

Bahasa Asemmbly adalah bahasa pemrograman tingkat rendah yang berkomunikasi langsung dengan perangkat keras melalui instruksi prosesor.Pada prosesor intel 8086,assembly bekerja dengan memanipulasikan register, memori, dan interrupt untuk menyelesaikan tugas-tugas tertentu.

2.2 MINI GAME TEBAK ANGKA

Mini game tebak angka adalah salah satu jenis permainan sederhana yang mengandalkan logika dan interaksi pengguna dengan komputer. Berikut adalah dasar teori yang mendukung permainan ini:

1. Konsep Permainan Interaktif

Definisi: Permainan interaktif adalah jenis permainan yang melibatkan interaksi langsung antara pengguna dan sistem komputer. Pengguna memberikan input (tebakan angka), dan sistem merespons dengan umpan balik yang sesuai (misalnya, apakah tebakan terlalu tinggi, terlalu rendah, atau tepat).

Tujuan: Meningkatkan keterlibatan pengguna dengan memberikan tantangan yang harus diselesaikan melalui interaksi berulang.

2. Logika Komparatif

Komparasi Angka: Logika utama dalam permainan ini adalah membandingkan angka yang ditebak oleh pengguna dengan angka rahasia yang telah ditentukan.

Percabangan: Menggunakan struktur kontrol seperti IF-ELSE atau JMP dalam Assembly untuk mengarahkan alur program berdasarkan hasil perbandingan.

3. Validasi Input

Rentang Angka: Input dari pengguna harus divalidasi agar berada dalam rentang tertentu. Dalam permainan ini, rentangnya adalah antara 1 dan 100.

Keamanan: Validasi input membantu mencegah kesalahan dan memastikan bahwa permainan berjalan sesuai dengan aturan yang ditentukan.

4. Interaksi Input/Output

Input: Mengambil angka dari pengguna, sering kali menggunakan keyboard.

Output: Memberikan umpan balik berupa pesan teks yang mengarahkan atau memberi tahu pengguna hasil dari input mereka.

5. Penggunaan Register dan Memori

Register: Digunakan untuk menyimpan data sementara seperti angka rahasia dan tebakan pengguna. Register dalam CPU memungkinkan manipulasi data yang cepat dan efisien.

Memori: String pesan disimpan di segmen data, yang diakses dan ditampilkan selama eksekusi program.

6. Pengulangan dan Iterasi

Looping: Permainan ini menggunakan loop untuk mengulang proses pengambilan input dan validasi hingga pengguna menebak angka dengan benar. Ini menunjukkan konsep iterasi di mana suatu blok kode dieksekusi berulang kali berdasarkan kondisi tertentu.

7. Feedback dan Motivasi

Feedback Positif/Negatif: Permainan memberikan umpan balik langsung kepada pengguna untuk setiap tebakan yang mereka buat, baik itu terlalu tinggi, terlalu rendah, atau tepat. Ini membantu pengguna belajar dan membuat tebakan yang lebih baik.

Motivasi: Umpan balik positif seperti "Selamat, tebakan Anda benar!" memberikan rasa pencapaian dan memotivasi pengguna untuk menyelesaikan permainan.

2.3 EMU8086

EMU8086 adalah emulator yang dirancang untuk mempelajari dan menjalankan program assembly berbasis prosesor Intel 8086. Fitur utama EMU8086 meliputi:

A.Debugger untuk melacak eksekusi instruksi.

B.Lingkungan simulasi DOS.

C.Dukungan interrupt berbasis real mode.

BAB III: IMPLEMENTASI

3.1 Alat Dan Bahan

1.Perangkat Lunak:

-EMU8086

-Text Editor Bawaan EMU8086

2.Perangkat Keras:

Laptop/PC

3.2 Langkah-Langkah Pembuatan

1. Siapkan Lingkungan: Instal assembler (TASM, MASM, atau NASM) dan text editor.

2. Inisialisasi Program: Tentukan offset awal (ORG 100H) dan titik masuk (START).

3. Definisikan Pesan dan Data:

Buat string pesan untuk interaksi pengguna.

Tetapkan angka rahasia (misalnya 40).

4. Tampilkan Pesan: Gunakan INT 21H dengan fungsi 09H untuk menampilkan pesan.

5. Ambil Input Pengguna: Gunakan INT 21H dengan fungsi 01H, lalu konversi ASCII ke angka.

6. Bandingkan Input: Gunakan CMP untuk membandingkan input dengan angka rahasia.

7. Berikan Umpan Balik: Tampilkan pesan sesuai hasil perbandingan (terlalu rendah, tinggi, atau benar).

8. Loop hingga Benar: Ulangi proses input dan perbandingan sampai tebakan benar.

9. Keluar Program: Gunakan INT 21H dengan fungsi 4CH untuk keluar.

10. Kompilasi dan Test: Kompilasi kode dan jalankan di EMU8086

3.3 Kode Program

Berikut adalah contoh kode program untuk game mini tebak angka:

ORG 100H ; Program dimulai pada offset 100H

START:

MOV AH, 09H ; Menampilkan teks

LEA DX, MSG1

INT 21H

; Set angka rahasia ke 51

MOV SI, 51 ; Menyimpan angka rahasia (51) ke SI

INPUT\_GUESS:

MOV AH, 09H ; Menampilkan pesan masukkan angka

LEA DX, MSG2

INT 21H

CALL GET\_NUMBER ; Memanggil fungsi untuk mengambil input angka

MOV DI, AX ; Menyimpan angka input ke DI

CMP DI, SI ; Membandingkan angka input dengan angka rahasia

JE CORRECT ; Jika sama, loncat ke CORRECT

JL TOO\_LOW ; Jika input < angka rahasia, loncat ke TOO\_LOW

JG TOO\_HIGH ; Jika input > angka rahasia, loncat ke TOO\_HIGH

TOO\_LOW:

MOV AH, 09H

LEA DX, MSG3 ; Pesan "Tebakan terlalu rendah"

INT 21H

JMP INPUT\_GUESS ; Ulangi tebak angka

TOO\_HIGH:

MOV AH, 09H

LEA DX, MSG4 ; Pesan "Tebakan terlalu tinggi"

INT 21H

JMP INPUT\_GUESS ; Ulangi tebak angka

CORRECT:

MOV AH, 09H

LEA DX, MSG5 ; Pesan "Tebakan benar"

INT 21H

JMP EXIT ; Keluar dari game

EXIT:

MOV AH, 4CH ; Perintah keluar

INT 21H

; Fungsi untuk mengambil input angka dari pengguna

GET\_NUMBER:

MOV AH, 01H ; Input karakter pertama

INT 21H

SUB AL, '0' ; Konversi ASCII ke angka

MOV BL, AL ; Simpan digit pertama

MOV AH, 01H ; Input karakter kedua (jika ada)

INT 21H

SUB AL, '0' ; Konversi ASCII ke angka

MOV BH, AL ; Simpan digit kedua (jika ada)

; Jika ada digit kedua, kombinasikan keduanya

MOV AL, BL ; AL = digit pertama

MOV CL, 10

MUL CL ; AL = digit pertama \* 10

ADD AL, BH ; AL = digit pertama \* 10 + digit kedua

; Jika hanya satu digit, AL sudah berisi digit pertama

RET

; Subroutine untuk menampilkan angka (desimal) di layar

DISPLAY\_NUM:

XOR CX, CX ; Membersihkan CX untuk menghitung digit

MOV BX, 10 ; Basis desimal

CONVERT\_LOOP:

XOR DX, DX ; Membersihkan DX untuk operasi DIV

DIV BX ; Membagi AX dengan 10

ADD DL, '0' ; Mengonversi sisa bagi ke karakter ASCII

PUSH DX ; Menyimpan digit di stack

INC CX ; Menambah jumlah digit

CMP AX, 0

JNE CONVERT\_LOOP ; Ulangi jika AX belum nol

PRINT\_DIGITS:

POP DX ; Mengambil digit dari stack

MOV AH, 02H ; Menampilkan karakter

INT 21H

LOOP PRINT\_DIGITS

RET

; Data segment

MSG1 DB 'Selamat datang di game Tebak Angka!$'

MSG2 DB 0DH, 0AH, 'Masukkan angka antara 1-100: $'

MSG3 DB 0DH, 0AH, 'Tebakan terlalu rendah.$'

MSG4 DB 0DH, 0AH, 'Tebakan terlalu tinggi.$'

MSG5 DB 0DH, 0AH, 'Selamat, tebakan Anda benar!$'

MSG\_SECRET DB 0DH, 0AH, 'Kode rahasia: $'

END START ; Menandai akhir program

BAB IV:HASIL DAN PEMBAHASAN

4.1 Hasil Implementasi

Mini Game Tebak Angka berhasil dijalankan menggunakan EMU8086 dengan hasil sebagai berikut:

1. Tampilan Awal

Saat program dijalankan, pengguna akan melihat pesan selamat datang dan petunjuk untuk memasukkan angka:

Selamat datang di game Tebak Angka!

Masukkan angka antara 50-100:

2. Input Angka

Pengguna memasukkan angka, misalnya 55.

3. Feedback Program

Jika Tebakan Terlalu Rendah:

Setelah pengguna memasukkan angka yang lebih kecil dari angka rahasia, program akan menampilkan:

Tebakan terlalu rendah.

Masukkan angka antara 1-100:

Jika Tebakan Terlalu Tinggi:

Jika angka yang dimasukkan lebih besar dari angka rahasia:

Tebakan terlalu tinggi.

Masukkan angka antara 1-100:

Jika Angka di Luar Rentang 1-100:

Jika angka di luar rentang 1-100 dimasukkan:

Input harus antara 1 dan 100!

Masukkan angka antara 1-100:

Jika Tebakan Benar:

Jika pengguna menebak angka yang benar (misalnya 40):

Selamat, tebakan Anda benar!

4. Program Keluar

Setelah tebakan yang benar, program akan keluar dengan menggunakan interrupt INT 21H fungsi 4CH.

4.2 ANALISIS

Keuntungan:

1. Sederhana dan Edukatif: Cocok untuk pemula dalam memahami logika dasar pemrograman Assembly.

2. Interaktif: Melibatkan pengguna dengan memberikan umpan balik langsung.

3. Ringan: Memiliki ukuran program yang kecil dan membutuhkan sumber daya minimal.

Kelemahan:

1. Terbatas: Hanya mendukung input angka sederhana dengan rentang tertentu.

2. Kurang User-Friendly: Tidak ada validasi input non-numerik, sehingga rentan terhadap kesalahan input.

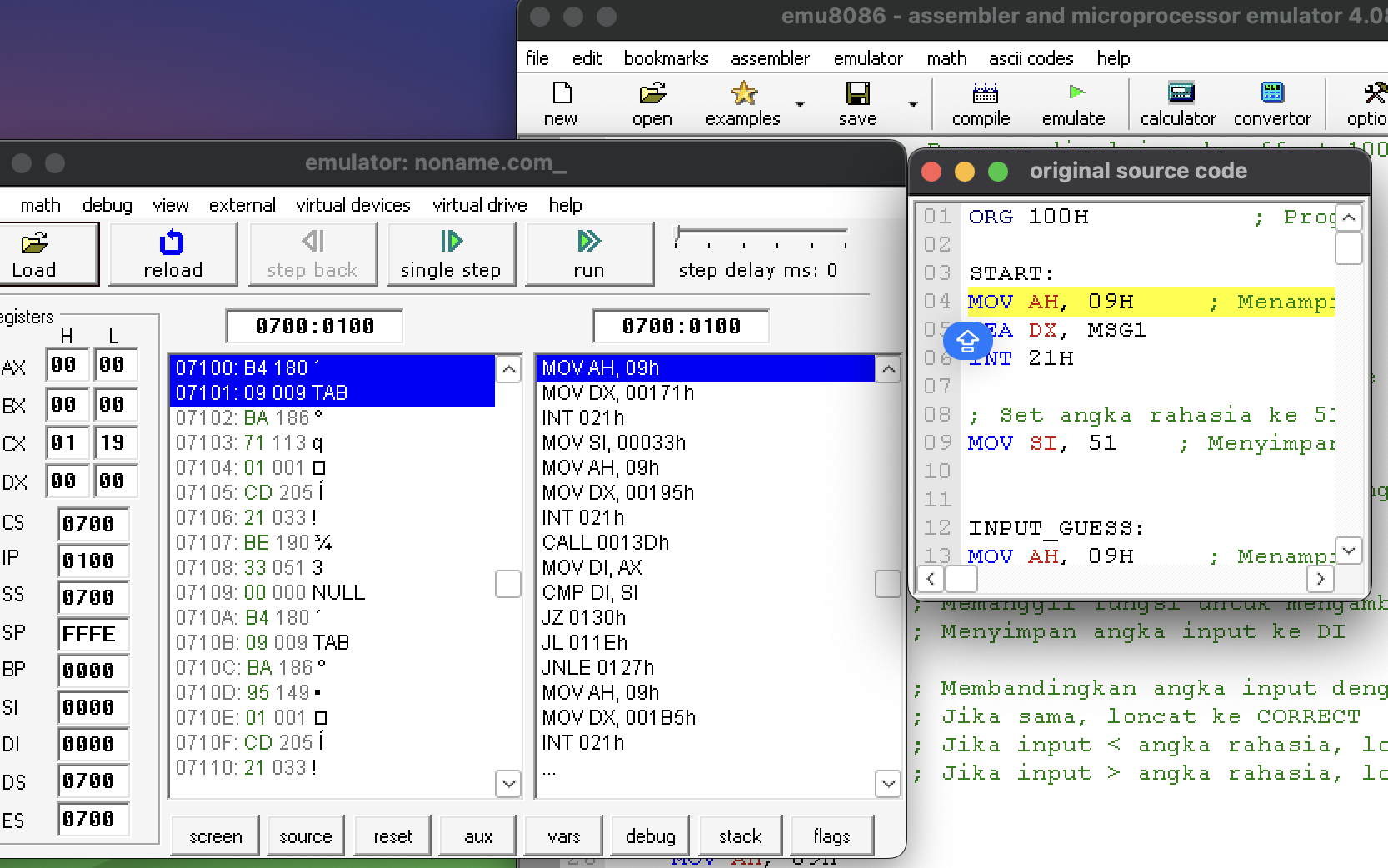
3. Dependensi pada DOS: Bergantung pada lingkungan DOS atau emulator seperti EMU8086, yang kurang relevan untuk sistem modern.

BAB V: PENUTUP

5.1 Kesimpulan

Mini game tebak angka dalam Assembly adalah alat pembelajaran yang sederhana dan efektif untuk memahami dasar-dasar pemrograman seperti input/output, logika perbandingan, dan kontrol alur. Meskipun bermanfaat untuk pemula, game ini memiliki keterbatasan dalam fleksibilitas dan user experience, serta bergantung pada emulator DOS seperti EMU8086.

LAMPIRAN DI EMU8086



LAMPIRAN DI GITHUB

