

LAPORAN

“Dokumentasi Pembuatan Aplikasi Game Ulo Asem”

Diajukan untuk memenuhi salah satu tugas Mata Kuliah Dasar Sistem Komputer yang di
ampu oleh:

Ali Tarmuji, S.T., M.Cs. Teknik Informatika



Disusun Oleh:

Mohammad Farid Hendianto Kelas I 2200018401

UNIVERSITAS AHMAD DAHLAN
FAKULTAS TEKNOLOGI INDUSTRI
PROGRAM STUDI INFORMATIKA
TAHUN 2023

KATA PENGANTAR

Puji dan syukur yang tak terhingga saya panjatkan ke hadirat Allah SWT, yang telah memberikan kemudahan dan kesempatan kepada saya untuk dapat menyelesaikan laporan dasar sistem komputer dengan judul "Dokumentasi Pembuatan Aplikasi Game Ulo Asem". Saya merasa sangat bersyukur karena telah diberikan kemampuan dan kekuatan untuk menyelesaikan laporan ini tepat pada waktunya.

Tidak lupa saya sampaikan terima kasih yang tak terhingga kepada Bapak Ali Tarmuji, S.T., M.Cs. Teknik Informatika, selaku dosen pembimbing yang telah memberikan bimbingan dan arahan yang sangat membantu saya dalam mengerjakan pembuatan aplikasi ini. Saya merasa sangat beruntung telah diberikan kesempatan untuk belajar dan mengikuti bimbingan dari Bapak.

Terimakasih juga kepada teman-teman saya yang mensupport saya dalam pembuatan aplikasi ini, serta tidak lupa juga kepada orang tua saya yang selalu memberikan dukungan dan semangat kepada saya sehingga saya dapat menyelesaikan tugas ini.

Meskipun saya berusaha semaksimal mungkin dalam menyusun laporan ini, saya sadar bahwa masih banyak kekurangan dan kelemahan yang terdapat pada laporan ini. Oleh karena itu, saya mengajak pembaca untuk memberikan kritik dan saran yang membangun kepada saya, agar laporan-laporan selanjutnya yang saya susun dapat menjadi lebih baik lagi. Kritik konstruktif sangat saya harapkan demi kesempurnaan laporan selanjutnya.

Akhir kata, saya berharap laporan ini dapat memberikan manfaat dan menjadi referensi yang bermanfaat bagi kita semua. Saya juga berharap agar pembuatan aplikasi di assembly dapat berjalan dengan lebih lancar dan sukses di masa yang akan datang.

Yogyakarta, Januari 2023

Penyusun



Mohammad Farid Hendianto

DAFTAR ISI

KATA PENGANTAR.....	2
DAFTAR ISI.....	3
BAB I PENDAHULUAN.....	4
1.1. Latar Belakang Masalah	4
1.2. Rumusan Masalah	5
1.3. Tujuan Penulisan	5
1.4. Manfaat Penulisan	6
BAB II PEMBAHASAN.....	7
2.1. Deskripsi Aplikasi	7
2.2. Spesifikasi Sistem	8
2.3. Langkah-Langkah Instalasi	8
2.4. Pembuatan Aplikasi.....	9
2.4.1. Flowchart Aplikasi.....	9
2.4.2. Desain Antarmuka Aplikasi.....	13
2.4.3. Kode program Assembly untuk menampilkan antarmuka	14
2.4.4. Kode program Assembly untuk proses	19
2.4.5. Percobaan Aplikasi.....	21
BAB IV PENUTUP	24
3.1. Kesimpulan	24
3.2. Kritik dan Saran	24
DAFTAR PUSTAKA.....	25

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Dalam era yang semakin maju ini, teknologi komputer merupakan salah satu aspek yang sangat penting dalam kehidupan manusia. Teknologi komputer telah mengalami kemajuan yang sangat pesat sejak pertama kali ditemukan, dan terus berkembang hingga saat ini.

Salah satu teknologi yang terus berkembang dalam dunia komputer adalah pemrograman. Pemrograman merupakan salah satu keahlian yang sangat penting bagi para insinyur komputer, karena pemrograman merupakan dasar dari segala sesuatu yang berkaitan dengan komputer.

Salah satu bahasa pemrograman yang cukup populer adalah bahasa Assembly. Bahasa Assembly merupakan bahasa pemrograman yang bersifat tingkat rendah, sehingga dapat memberikan kontrol yang lebih spesifik terhadap hardware komputer. Namun, bahasa Assembly juga merupakan bahasa yang cukup sulit dipelajari, karena memiliki sintaks yang cukup kompleks dan membutuhkan pemahaman yang mendalam mengenai struktur komputer.

Bahasa Assembly merupakan bahasa yang sangat berguna bagi para insinyur komputer, karena bahasa ini dapat digunakan untuk membuat program yang dapat berinteraksi secara langsung dengan hardware komputer. Selain itu, bahasa Assembly juga memungkinkan para insinyur untuk memanipulasi data yang ada di dalam memori komputer secara lebih efektif.

Untuk itu, dibutuhkan aplikasi yang dapat membantu para pemula dalam mempelajari bahasa Assembly, sehingga dapat memudahkan proses belajar para pemula tersebut. Aplikasi game merupakan salah satu cara yang dapat menarik perhatian para pemula untuk terus belajar dan memahami bahasa Assembly.

Melalui aplikasi game Ulo Asem ini, diharapkan dapat membantu para pemula dalam memahami konsep-konsep dasar bahasa Assembly seperti syntax, operand, operator, dan lain-lain. Selain itu, aplikasi game ini juga dapat menjadi sarana hiburan bagi para pemula yang sedang belajar bahasa Assembly.

Oleh karena itu, penulis tertarik untuk menyusun laporan dasar sistem komputer yang berjudul "Dokumentasi Pembuatan Aplikasi Game Ulo Asem" sebagai salah satu syarat untuk menyelesaikan tugas akhir mata kuliah Dasar Sistem Komputer.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah yang telah dijelaskan, maka dapat dirumuskan beberapa masalah yang dapat dipecahkan dalam pembuatan aplikasi game Ulo Asem ini, yaitu:

- 1) Bagaimana cara membuat aplikasi game sederhana yang berjudul "Ulo Asem" menggunakan bahasa pemrograman Assembly (MASM).
- 2) Bagaimana cara menyusun deskripsi cara kerja, proses bisnis, flow chart, atau alur program dari aplikasi game "Ulo Asem".
- 3) Bagaimana cara menyusun desain atau sketsa antarmuka yang akan digunakan sesuai dengan spesifikasi aplikasi.
- 4) Bagaimana cara menulis kode program Assembly untuk menampilkan antarmuka yang telah dibuat serta menulis kode program Assembly untuk proses yang ada di dalam aplikasi game "Ulo Asem".
- 5) Bagaimana hasil capture dari percobaan program hingga diperoleh hasil yang sesuai dengan yang telah direncanakan.

1.3. Tujuan Penulisan

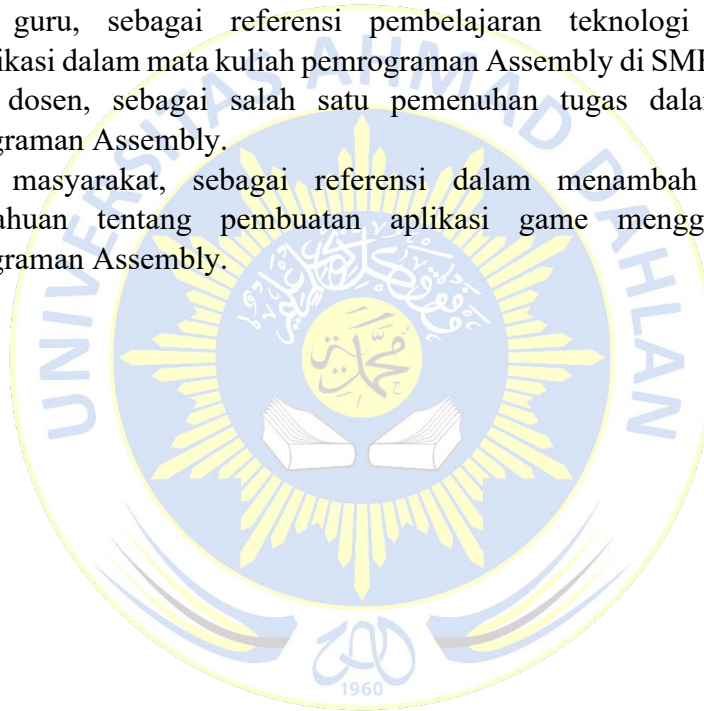
Adapun tujuan dari pembuatan aplikasi game Ulo Asem ini adalah:

- 1) Untuk memahami cara membuat aplikasi game sederhana menggunakan bahasa pemrograman Assembly.
- 2) Untuk memahami cara menyusun deskripsi cara kerja, proses bisnis, flow chart, atau alur program dari aplikasi game/.
- 3) Untuk mengetahui desain atau sketsa antarmuka yang akan digunakan sesuai dengan spesifikasi aplikasi.
- 4) Untuk mengetahui kode program Assembly untuk menampilkan antarmuka yang telah dibuat serta menulis kode program Assembly untuk proses yang ada di dalam aplikasi game.
- 5) Untuk memahami bagaimana proses percobaan program hingga diperoleh hasil yang sesuai dengan yang telah direncanakan.

1.4. Manfaat Penulisan

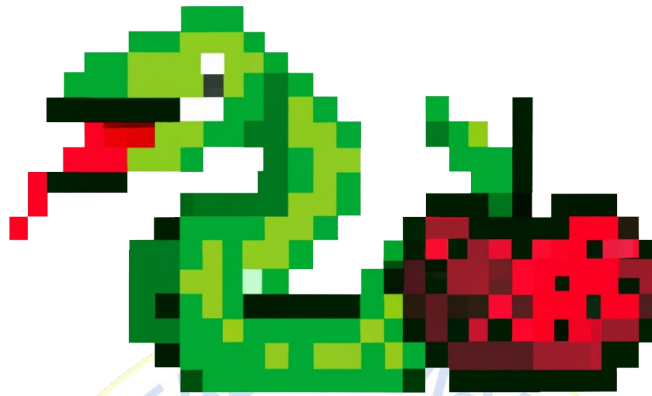
Berikut ini adalah manfaat penulisan laporan ini:

- 1) Untuk penulis, sebagai pengalaman dan ilmu tambahan dalam memahami dan mengetahui bagaimana cara membuat aplikasi game sederhana menggunakan bahasa pemrograman Assembly.
- 2) Untuk pelajar, sebagai referensi atau studi pustaka dalam memahami bagaimana cara membuat aplikasi game sederhana menggunakan bahasa pemrograman Assembly.
- 3) Untuk guru, sebagai referensi pembelajaran teknologi informasi dan komunikasi dalam mata kuliah pemrograman Assembly di SMK maupun SMA.
- 4) Untuk dosen, sebagai salah satu pemenuhan tugas dalam mata kuliah pemrograman Assembly.
- 5) Untuk masyarakat, sebagai referensi dalam menambah wawasan dan pengetahuan tentang pembuatan aplikasi game menggunakan bahasa pemrograman Assembly.



BAB II PEMBAHASAN

2.1. Deskripsi Aplikasi



Gambar 1 Logo Aplikasi Ulo Asem (Sumber: Penulis)

Ulo Asem adalah sebuah aplikasi game sederhana yang dibuat menggunakan bahasa pemrograman Assembly yaitu MASM. Aplikasi ini ditujukan untuk membantu para pemula dalam mempelajari bahasa Assembly dengan cara yang menyenangkan.

Aplikasi game Ulo Asem ini diberi nama dengan mengambil inspirasi dari bahasa Jawa yang merupakan salah satu bahasa daerah di Indonesia. Kata 'ulo' dalam bahasa Jawa merupakan salah satu sinonim untuk kata 'ular', sedangkan 'asem' merupakan kepanjangan dari bahasa Assembly yang merupakan bahasa pemrograman yang digunakan dalam pembuatan aplikasi ini. Sehingga, penamaan aplikasi game Ulo Asem ini merupakan gabungan dari kata 'ular' dan 'Assembly', yang diharapkan dapat menjadi salah satu sarana belajar bahasa Assembly yang menyenangkan bagi para pemula.

Dalam aplikasi game Ulo Asem ini, pengguna dapat memainkan game tersebut dengan menggunakan tombol panah pada keyboard untuk mengatur arah gerakan ular. Ular tersebut akan bergerak mengikuti arah yang ditunjukkan oleh pengguna, dan akan terus bergerak hingga menabrak tembok atau tubuh sendiri. Selain itu, setiap kali ular berhasil memakan sebuah apel, maka skor akan bertambah dan panjang ular akan bertambah.

Saat game berakhir, aplikasi game Ulo Asem akan menampilkan skor akhir yang telah diraih oleh pengguna. Skor tersebut akan ditampilkan sesuai dengan jumlah apel yang telah dimakan oleh ular. Selain itu, aplikasi game ini juga menyediakan fitur reset yang dapat digunakan oleh pengguna untuk mengulang permainan dari awal.

2.2. Spesifikasi Sistem

Untuk dapat menjalankan aplikasi game Ulo Asem, diperlukan spesifikasi sistem yang minimal sebagai berikut:

Sistem Operasi	: Windows XP 32 bit atau yangne lebih tinggi
Prosesor	: 512 Mhz atai lebih tinggi
Memory	: 32 MB atau lebih tinggi
Storage	: 1 MB ruang hardisk yang tersedia
Sound card	: Terdapat soundcard untuk menjalankan audio
Graphics	: Graphic card yang mendukung penampilan warna

Pastikan bahwa semua perangkat lunak yang diperlukan, seperti sistem operasi dan aplikasi pendukung, telah terinstal dengan benar pada komputer Anda agar game Ulo Asem dapat berjalan dengan lancar. Jika ada perangkat lunak yang kurang atau tidak terinstal dengan benar, maka kemungkinan game Ulo Asem akan mengalami error atau tidak dapat dijalankan sama sekali. Oleh karena itu, pastikan untuk melakukan instalasi semua perangkat lunak yang diperlukan dengan benar agar game dapat berjalan dengan lancar.

2.3. Langkah-Langkah Instalasi

Berikut adalah langkah-langkah untuk menjalankan aplikasi game Ulo Asem:

- 1) Download file aplikasi Ulo Asem yang telah tersedia di website github <https://github.com/IRedDragonICY/Ulo-Asem>
- 2) Pastikan bahwa komputer Anda sudah memenuhi spesifikasi yang dibutuhkan, seperti sistem operasi Windows XP 32 bit atau yang lebih tinggi, prosesor 512 Mhz atau lebih tinggi, memory 64 MB atau lebih tinggi, storage 3 MB ruang hardisk yang tersedia, sound card yang terdapat pada komputer, dan graphic card yang mendukung penampilan warna.
- 3) Buka file aplikasi Ulo Asem yang telah didownload.
- 4) Ikuti instruksi yang ada pada layar untuk menjalankan aplikasi game Ulo Asem.
- 5) Selamat bermain!

2.4. Pembuatan Aplikasi

Untuk membuat aplikasi Ulo Asem, alat yang dibutuhkan adalah:

- Library Irvine32
- Visual Studio (disarankan menggunakan versi 2022 atau yang lebih baru)
- Setting MASM (disarankan menggunakan versi 6.15 atau yang lebih baru)

Langkah-langkah untuk membuat aplikasi Ulo Asem adalah sebagai berikut:

- 1) Buka Visual Studio dan buat proyek baru dengan memilih "File" -> "New" -> "Project".
- 2) Pilih "Assembly" -> "Assembly Project" dan beri nama proyek sesuai keinginan.
- 3) Tambahkan library Irvine32 dengan cara meng-include file irvine32.inc pada proyek.
- 4) Tuliskan kode program sesuai dengan spesifikasi yang diinginkan.
- 5) Compile kode program dengan memilih "Build" -> "Build Solution".
- 6) Jalankan aplikasi dengan memilih "Debug" -> "Start Debugging".

Untuk memodifikasi build Ulo Asem, Anda bisa melakukan fork dari repositori <https://github.com/IRedDragonICY/Ulo-Asem> dan meng-duplicate project tersebut ke dalam akun Anda. Anda kemudian dapat mengubah kode sesuai dengan kebutuhan Anda. Jika Anda menemukan kesalahan atau memiliki saran untuk pengembangan lebih lanjut, Anda juga dapat mengirimkan pull request ke repositori asli.

2.4.1. Flowchart Aplikasi

Berikut adalah alur program aplikasi Ulo Asem

- 1) Memuat dan menampilkan interface permainan
 - Menggunakan library Irvine32 untuk memanggil procedure clrscr yang akan membersihkan layar
 - Memanggil procedure TampilanDinding untuk menggambar tembok pada layar permainan
 - Memanggil procedure TampilanPapanSkor untuk menampilkan papan skor pada layar permainan
 - Memanggil procedure MemilihKecepatanUlo untuk memberikan pemain pilihan kecepatan ulo

- Memulai loop yang akan menampilkan ulo dengan panjang awal 5 segmen dengan procedure TampilanPemain
- Memanggil procedure Randomize dari library Irvine32 untuk mengacak posisi apel yang akan muncul pada layar
- Memanggil procedure BuatAcakApel untuk menentukan posisi apel yang akan muncul
- Memanggil procedure TampilanApel untuk menampilkan apel pada layar permainan.
- Loop akan terus berjalan dan menunggu input dari pemain sampai pemain memutuskan untuk keluar dari permainan dengan menekan tombol x atau sampai pemain mati.

2) Meminta input kecepatan Ulo dari pemain

Pada bagian ini, aplikasi akan meminta pemain untuk memasukkan kecepatan Ulo yang diinginkan pada awal permainan. Input yang dimasukkan akan digunakan sebagai parameter untuk menentukan seberapa cepat gerakan Ulo pada permainan. Setelah pemain memasukkan kecepatan yang diinginkan, aplikasi akan melanjutkan ke langkah berikutnya.

3) Menampilkan Ulo dengan panjang 5 segmen

Pada tahap ini, Ulo akan ditampilkan di layar dengan panjang 5 segmen. Setiap segmen ditampilkan dengan menggunakan koordinat x dan y yang disimpan dalam array xPosisi dan yPosisi. Posisi awal Ulo ditentukan secara acak di layar.

4) Menampilkan Apel secara acak di layar

Pada poin ini, aplikasi akan menampilkan apel secara acak di layar. Apel akan muncul di posisi yang tidak sama dengan posisi Ulo. Proses ini akan terus berulang setiap kali apel dimakan oleh Ulo.

5) Meminta input gerakan dari pemain

Pada poin ini, program akan meminta input dari pemain tentang arah gerakan Ulo. Input yang diperbolehkan adalah arah atas, bawah, kiri, dan kanan. Setelah input diterima, program akan memperbarui posisi Ulo sesuai dengan arah yang dipilih pemain.

6) Memeriksa apakah input yang dimasukkan valid atau tidak

Dalam tahap ini, program akan memeriksa apakah input gerakan yang dimasukkan pemain merupakan salah satu dari empat arah yang tersedia, yaitu ke atas, ke bawah, ke kiri, atau ke kanan. Jika input yang dimasukkan valid, maka program

akan melanjutkan ke tahap selanjutnya. Namun jika input yang dimasukkan tidak valid, program akan mengabaikan input tersebut dan terus menunggu input yang valid dari pemain.

7) Menggerakkan Ulo sesuai dengan input yang dimasukkan oleh pemain

Dalam tahap ini, program akan memeriksa apakah input yang diberikan pemain valid atau tidak. Jika input yang diberikan pemain valid, maka Ulo akan bergerak sesuai dengan arah yang diberikan pemain. Jika input yang diberikan pemain tidak valid, maka Ulo akan terus bergerak ke arah yang sama seperti sebelumnya. Misalnya, jika pemain memberikan input untuk bergerak ke kiri, tetapi Ulo sedang berada di ujung kanan layar, maka Ulo tidak akan bergerak ke kiri dan akan terus bergerak ke arah yang sama seperti sebelumnya.

8) Memeriksa apakah Ulo telah memakan Apel

Untuk tahap ini, program akan memeriksa apakah koordinat Ulo sama dengan koordinat Apel. Jika sama, maka Ulo akan memakan Apel dan panjang Ulo akan bertambah 1 segmen. Sebaliknya, jika tidak sama, maka Ulo tidak akan memakan Apel dan tidak ada perubahan pada panjang Ulo.

9) Jika Ulo telah memakan Apel, maka tambahkan 1 segmen pada Ulo dan menampilkan Apel baru secara acak

Tahap ini merupakan bagian dari permainan Ulo Asem di mana setelah Ulo berhasil memakan Apel, maka panjang Ulo akan ditambah sebanyak 1 segmen. Selain itu, Apel yang telah dimakan juga akan dihapus dan digantikan dengan Apel baru yang akan ditampilkan secara acak di layar permainan. Ini bertujuan untuk menjaga keseruan permainan, sehingga pemain harus terus-menerus mengontrol gerakan Ulo untuk memakan Apel yang muncul di layar.

10) Memeriksa apakah Ulo telah menabrak tembok atau menabrak dirinya sendiri

Tahap ini adalah tahap yang mengecek apakah Ulo telah menabrak tembok atau menabrak dirinya sendiri. Proses ini dilakukan dengan memeriksa posisi Ulo apakah sudah melewati batas layar atau menyentuh salah satu segmen tubuhnya sendiri. Jika kondisi tersebut terpenuhi, maka permainan akan dihentikan dan Ulo dinyatakan mati.

11) Jika Ulo telah menabrak tembok atau menabrak dirinya sendiri, maka permainan berakhir dan menampilkan skor akhir

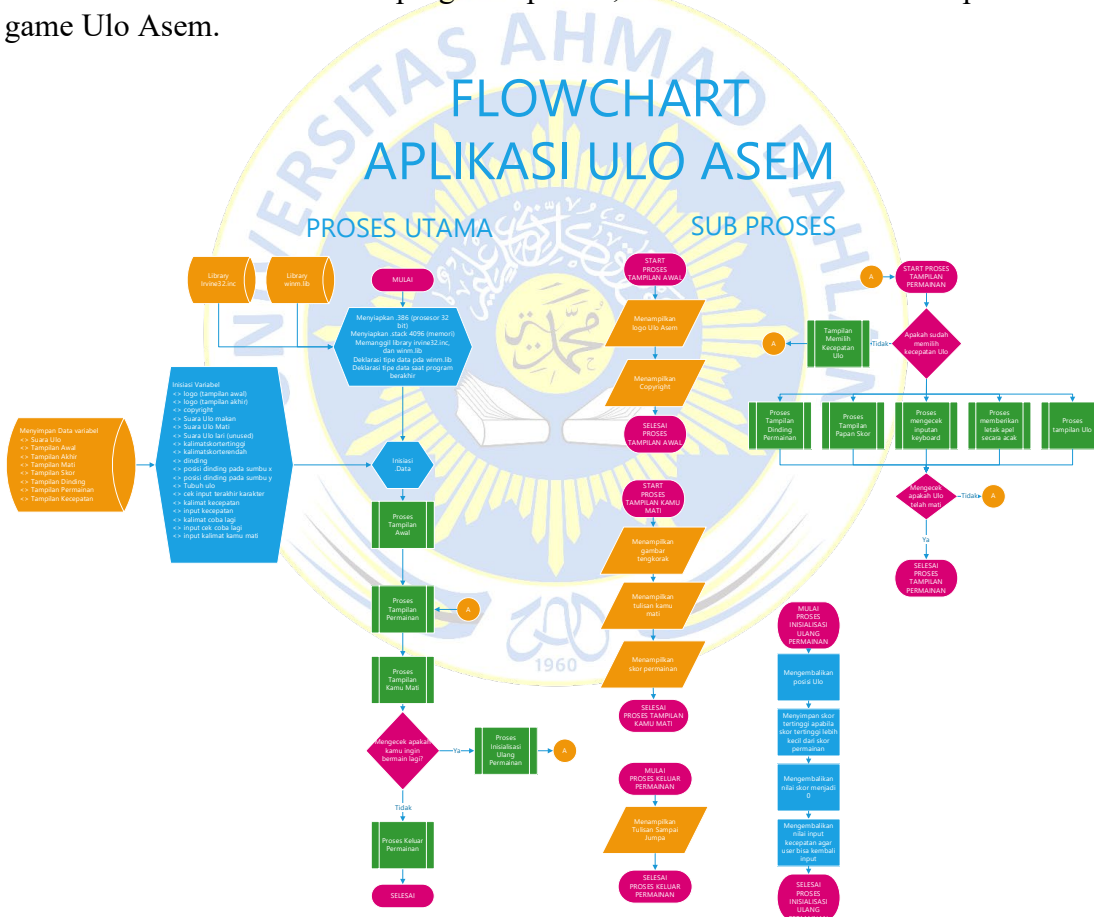
Apabila Ulo telah menabrak tembok atau menabrak dirinya sendiri, maka permainan berakhir dan menampilkan skor akhir. Tahap ini bertujuan untuk

menghentikan permainan jika kondisi tersebut terjadi. Skor akhir akan ditampilkan pada layar sebagai hasil dari permainan yang telah dimainkan oleh pemain.

12) Meminta input dari pemain untuk memainkan lagi atau keluar dari permainan

Jika pemain memilih untuk memainkan lagi, maka ulangi tahap 2 sampai 11. Jika pemain memilih untuk keluar dari permainan, maka program akan selesai dan menampilkan pesan "Sampai Jumpa".

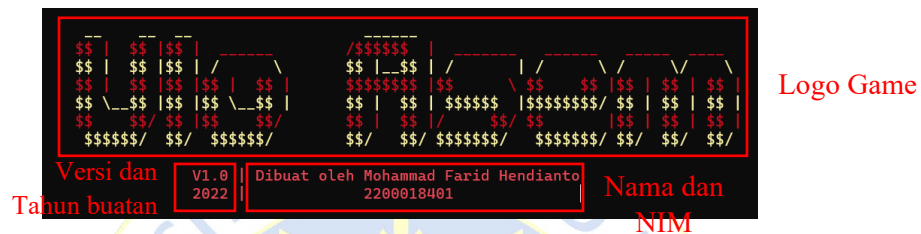
Untuk memahami alur program aplikasi, berikut adalah flowchart aplikasi game Ulo Asem.



Gambar 2 Flowchart Sederhana Program Aplikasi Game Ulo Asem (Sumber: Penulis)

2.4.2. Desain Antarmuka Aplikasi

Untuk desain antarmuka aplikasi Ulo Asem, diperlukan beberapa elemen utama yang perlu dipertimbangkan. Pada awal program akan ditampilkan sebuah logo dan disertai nama pembuat. Kemudian adalah area permainan di mana Ulo dan Apel akan ditampilkan. Selanjutnya ada panel skor yang menampilkan skor saat ini dari pemain. Lalu ada pengaturan kecepatan yang mengizinkan pemain untuk mengatur kecepatan Ulo saat bermain. Selanjutnya juga ada tampilan di mana untuk menampilkan Ulo jika mati (game over). Terakhir, tampilan akhir program sebelum aplikasi di tutup.



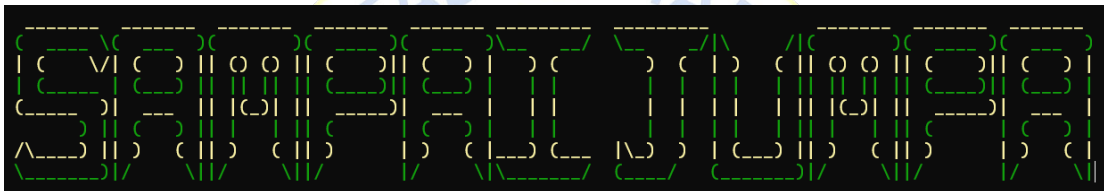
Gambar 3 Rancangan tampilan awal program saat dijalankan (Sumber: Penulis)



Gambar 4 Rancangan tampilan dalam permainan (Sumber: Penulis)



Gambar 5 Rancangan Tampilan Saat Ulo Mati [Game Over] (Sumber: Penulis)



Gambar 6 Rancangan Tampilan Saat Keluar Program (Sumber: Penulis)

2.4.3. Kode program Assembly untuk menampilkan antarmuka

Untuk menampilkan antarmuka, dalam assembly 32 bit, saya menggunakan library irvine32 untuk memudahkan pengerjaan assembly.

Untuk menaruh lokasi tulisan, bisa menggunakan call Gotoxy. Berikut adalah deklarasi lengkapnya

```
Mov dl,kolom
```

```
Mov dh,baris
```

```
call Gotoxy
```

Mov dl,kolom dan Mov dh,baris adalah dua perintah yang digunakan untuk memindahkan nilai dari variabel kolom dan baris ke dalam register DL dan DH masing-masing. Register DL dan DH merupakan register yang digunakan untuk menyimpan data-data yang akan diolah oleh sebuah prosesor.

Setelah nilai kolom dan baris tersebut berhasil dipindahkan ke dalam register DL dan DH, perintah call Gotoxy akan dijalankan. Call Gotoxy adalah perintah yang

digunakan untuk memanggil sebuah prosedur yang bernama Gotoxy. Prosedur Gotoxy merupakan sebuah prosedur yang telah didefinisikan sebelumnya dalam library irvine32 yang bertujuan untuk mengatur posisi cursor pada layar.

Dengan menggunakan perintah `Mov dl,kolom` dan `Mov dh,baris`, kita dapat menentukan kolom dan baris yang diinginkan untuk posisi cursor. Nilai dari variabel kolom dan baris tersebut akan disimpan ke dalam register DL dan DH, dan kemudian akan diolah oleh prosedur Gotoxy untuk mengatur posisi cursor sesuai dengan yang diinginkan.

Untuk mencetak sebuah kalimat/ string, anda bisa menggunakan `call WriteString`. Berikut adalah langkah untuk deklarasi menggunakan `call WriteString` pada library Irvine32

```
mov edx, OFFSET nama_variabel  
call WriteString
```

`Mov edx, OFFSET nama_variabel` adalah perintah yang digunakan untuk memindahkan alamat dari sebuah variabel ke dalam register EDX. Register EDX merupakan register yang digunakan untuk menyimpan data-data yang akan diolah oleh sebuah prosesor.

OFFSET adalah sebuah operator yang digunakan untuk mengambil alamat dari sebuah variabel. Jadi, dengan menggunakan perintah `Mov edx, OFFSET nama_variabel`, kita dapat memindahkan alamat dari variabel `nama_variabel` ke dalam register EDX.

Setelah alamat dari variabel `nama_variabel` berhasil dipindahkan ke dalam register EDX, perintah `call WriteString` akan dijalankan. `Call WriteString` adalah perintah yang digunakan untuk memanggil sebuah prosedur yang bernama `WriteString`. Prosedur `WriteString` merupakan sebuah prosedur yang telah didefinisikan sebelumnya dalam library irvine32 yang bertujuan untuk menampilkan string pada layar.

Dengan menggunakan perintah `Mov edx, OFFSET nama_variabel`, kita dapat menentukan alamat dari sebuah variabel yang akan ditampilkan pada layar. Alamat dari variabel tersebut akan disimpan ke dalam register EDX, dan kemudian akan diolah oleh prosedur `WriteString` untuk menampilkan string pada layar sesuai dengan yang diinginkan.

Untuk menulis karakter, bisa menggunakan call WriteChar dari library irvine32

```
mov eax,"x"  
call WriteChar
```

Mov eax,"x" adalah perintah yang digunakan untuk memindahkan sebuah karakter ke dalam register EAX. Register EAX merupakan register yang digunakan untuk menyimpan data-data yang akan diolah oleh sebuah prosesor.

Dalam perintah ini, karakter "x" akan dipindahkan ke dalam register EAX. Karakter "x" merupakan sebuah karakter yang ditulis dengan menggunakan tanda petik dua (") di sebelah kiri dan kanan.

Setelah karakter "x" berhasil dipindahkan ke dalam register EAX, perintah call WriteChar akan dijalankan. Call WriteChar adalah perintah yang digunakan untuk memanggil sebuah prosedur yang bernama WriteChar. Prosedur WriteChar merupakan sebuah prosedur yang telah didefinisikan sebelumnya dalam library irvine32 yang bertujuan untuk menampilkan sebuah karakter pada layar.

Dengan menggunakan perintah Mov eax,"x", kita dapat menentukan karakter yang akan ditampilkan pada layar. Karakter tersebut akan disimpan ke dalam register EAX, dan kemudian akan diolah oleh prosedur WriteChar untuk menampilkan karakter pada layar sesuai dengan yang diinginkan.

Perintah Mov eax,"x" dan call WriteChar dapat digunakan untuk menampilkan karakter apapun yang diinginkan pada layar, tidak hanya karakter "x" saja. Hanya saja, kita perlu mengganti karakter "x" dengan karakter yang diinginkan pada perintah Mov eax,"x" untuk menampilkan karakter tersebut pada layar. Misalnya, jika kita ingin menampilkan karakter "a" pada layar, maka kita perlu mengganti karakter "x" dengan karakter "a" pada perintah Mov eax,"a".

Untuk mencetak angka, bisa menggunakan call WriteInt dari library irvine. Berikut ini adalah cara deklarasinya.

```
mov al,nama_variabel  
call WriteInt
```

Mov al,nama_variabel adalah perintah yang digunakan untuk memindahkan nilai dari sebuah variabel ke dalam register AL. Register AL merupakan register yang digunakan untuk menyimpan data-data yang akan diolah oleh sebuah prosesor.

Dalam perintah ini, nilai dari variabel nama_variabel akan dipindahkan ke dalam register AL. Nilai dari variabel nama_variabel merupakan sebuah bilangan bulat yang dapat berupa angka positif atau negatif.

Setelah nilai dari variabel `nama_variabel` berhasil dipindahkan ke dalam register AL, perintah `call WriteInt` akan dijalankan. `Call WriteInt` adalah perintah yang digunakan untuk memanggil sebuah prosedur yang bernama `WriteInt`. Prosedur `WriteInt` merupakan sebuah prosedur yang telah didefinisikan sebelumnya dalam library `irvine32` yang bertujuan untuk menampilkan sebuah bilangan bulat pada layar.

Dengan menggunakan perintah `Mov al,nama_variabel`, kita dapat menentukan nilai yang akan ditampilkan pada layar. Nilai tersebut akan disimpan ke dalam register AL, dan kemudian akan diolah oleh prosedur `WriteInt` untuk menampilkan bilangan bulat pada layar sesuai dengan yang diinginkan.

Perintah `Mov al,nama_variabel` dan `call WriteInt` dapat digunakan untuk menampilkan nilai apapun yang diinginkan pada layar, tidak hanya nilai dari variabel `nama_variabel` saja. Hanya saja, kita perlu mengganti `nama_variabel` dengan variabel yang diinginkan pada perintah `Mov al,nama_variabel` untuk menampilkan nilai tersebut pada layar. Misalnya, jika kita ingin menampilkan nilai dari variabel `bilangan`, maka kita perlu mengganti `nama_variabel` dengan `bilangan` pada perintah `Mov al,bilangan`.

Untuk memberikan warna tampilan pada console, kita bisa menggunakan `call SetTextColor` pada `Irvine32`. Berikut adalah cara deklarasinya.

```
mov  eax,warna+(warna*angka)
call SetTextColor
```

`Call settextcolor` adalah perintah yang digunakan untuk memanggil sebuah prosedur yang bernama `settextcolor`. Prosedur `settextcolor` merupakan sebuah prosedur yang telah didefinisikan sebelumnya dalam library `irvine32` yang bertujuan untuk mengubah warna teks pada layar.

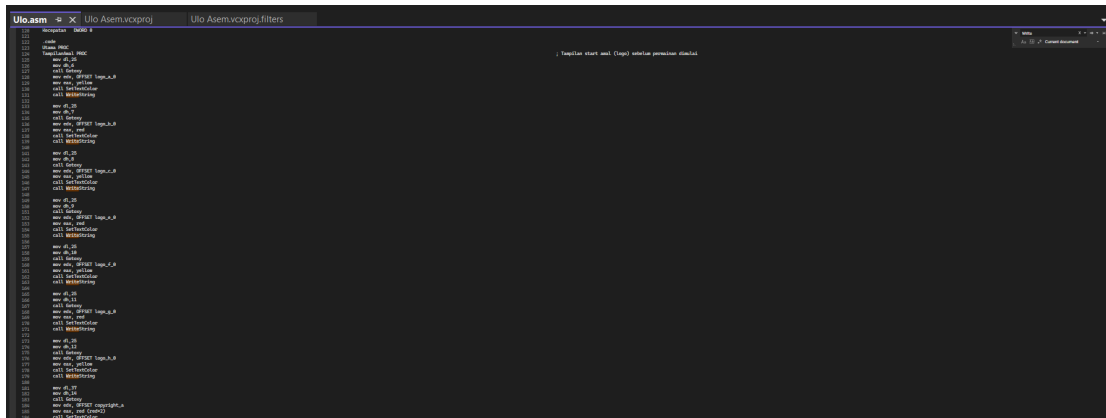
Perintah `call settextcolor` dapat digunakan dengan cara menuliskan `call settextcolor` diikuti oleh nomor warna yang diinginkan. Nomor warna tersebut merupakan sebuah bilangan bulat yang menunjukkan warna teks yang akan digunakan.

Ada beberapa warna teks yang dapat digunakan dalam perintah `call settextcolor`, yaitu:

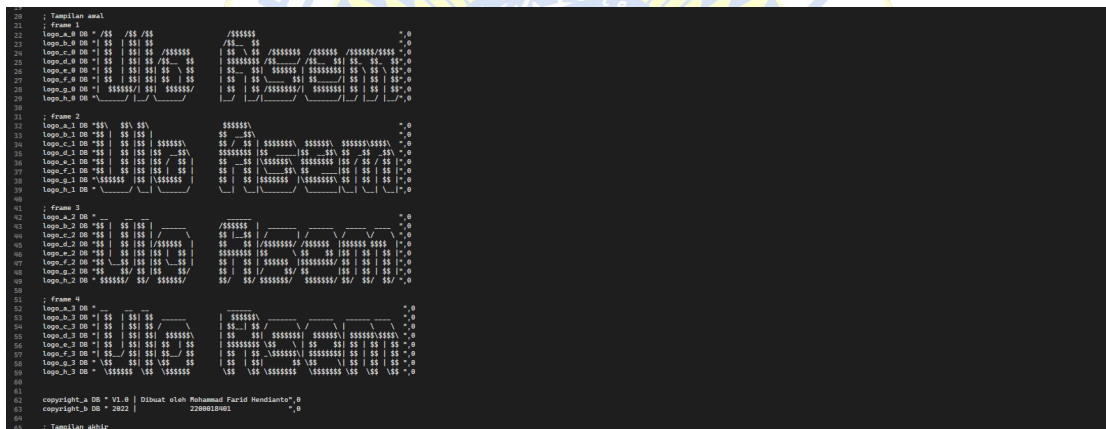
0 = Hitam	3 = Cyan	6 = Kuning
1 = Biru	4 = Merah	7 = Putih
2 = Hijau	5 = Magenta	

Selain itu ada prosedur call `ClrScr` untuk membersihkan console.

Contoh penggunaan untuk tampilan adalah seperti berikut



Gambar 7 Contoh kode program assembly untuk tampilan dalam Aplikasi Game Ulo Asem (Sumber: Penulis)



Gambar 8 Contoh menyimpan karakter yang ingin ditampilkan ke dalam sebuah variabel. (Sumber Penulis)

```

1663     mov inputChar, "+"
1664     dec yPosisiDinding[3]
1665     call ClrScr
1666     jmp Permainan
1667     InisialisasiUlangPermainan ENDP
1668     END Utama

```

Gambar 9 Membersihkan layar dengan call `ClrScr` (Sumber: Penulis)

Untuk lebih jelasnya, Anda bisa melihat source code pada akhir laporan ini.

2.4.4. Kode program Assembly untuk proses

Untuk membantu memproses program dalam aplikasi game Ulo Asem, salah satunya menggunakan call Randomize yang berasal dari library Irvine32.

Call randomize adalah perintah yang digunakan untuk memanggil sebuah prosedur yang bernama randomize. Prosedur randomize merupakan sebuah prosedur yang telah didefinisikan sebelumnya dalam library irvine32 yang bertujuan untuk mengacak sebuah bilangan bulat yang telah ditentukan sebelumnya.

Perintah call randomize dapat digunakan dengan cara menuliskan call randomize diikuti oleh nama variabel yang akan diacak. Variabel tersebut merupakan sebuah bilangan bulat yang akan diacak oleh prosedur randomize.

Contoh penggunaan perintah call randomize:

```
call randomize
```

Perintah di atas akan mengacak nilai dari variabel bilangan. Hasil dari pengacakan tersebut akan disimpan kembali ke dalam variabel bilangan.

Perintah call randomize dapat digunakan untuk mengacak bilangan bulat apapun yang diinginkan. Hanya saja, kita perlu mengganti nama variabel bilangan dengan variabel yang diinginkan pada perintah call randomize untuk mengacak nilai tersebut. Misalnya, jika kita ingin mengacak nilai dari variabel angka, maka kita perlu mengganti bilangan dengan angka pada perintah call randomize(angka).

```

991  Loop_TampilanUlo
992  call Randomize
993  call BicaraApel
994  call TampilanApel
995
; mengacak dengan library Irvine32.
; Menampilkan selesai

```

Gambar 10 Pemakaian Call Randomize untuk mengacak keberadaan letak apel pada papan permainan (Sumber: Penulis)

Kemudian ada call randomrange yang berasal dari library Irvine32. Call randomrange adalah perintah yang digunakan untuk memanggil sebuah prosedur yang bernama randomrange. Prosedur randomrange merupakan sebuah prosedur yang telah didefinisikan sebelumnya dalam library irvine32 yang bertujuan untuk mengacak sebuah bilangan bulat yang telah ditentukan sebelumnya dengan rentang nilai yang ditetapkan.

Perintah call randomrange dapat digunakan dengan cara menuliskan call randomrange diikuti oleh nama variabel yang akan diacak dan rentang nilai yang diinginkan. Variabel tersebut merupakan sebuah bilangan bulat yang akan diacak oleh prosedur randomrange dengan rentang nilai yang telah ditentukan.

Di dalam program tersebut, ada bagian untuk mengecek program aplikasi Cmp inputchar adalah perintah yang digunakan untuk membandingkan sebuah karakter dengan sebuah variabel yang telah ditentukan sebelumnya. Perintah cmp inputchar terdiri dari dua bagian, yaitu cmp dan inputchar.

Cmp merupakan perintah yang digunakan untuk membandingkan dua bilangan bulat atau dua karakter.

```

1000      mov     ecx, 0
1001      cmp     inputChar, 'x'
1002      je      HelaarPermainan
1003      cmp     inputChar, 'u'
1004      je      CekAtas
1005      cmp     inputChar, 's'
1006      je      CekBawah
1007      cmp     inputChar, 'a'
1008      je      CekKiri
1009      cmp     inputChar, 'd'
1010      je      CekKanan
1011      jmp     PermainanLamp
1012
1013      ; Cek keyboard (u,s,a,d) untuk bisa bergerak
1014      cmp     TerakhirInputKarakter, ""
1015      je      JagaPosisiDinding
1016      mov     cl, yPosisiDinding[1]
1017      dec     cl
1018      cmp     yPosisi[0], cl
1019      jl      PindahBawah
1020      je      Reti
1021      cmp     TerakhirInputKarakter, ""
1022      je      JagaPosisiKiri
1023      cmp     TerakhirInputKarakter, "a"
1024      je      JagaPosisiKanan
1025      mov     cl, yPosisiDinding[0]
1026      jmp     cl

```

Gambar 11 Contoh untuk mengecek input dari keyboard (Sumber: Penulis)

```

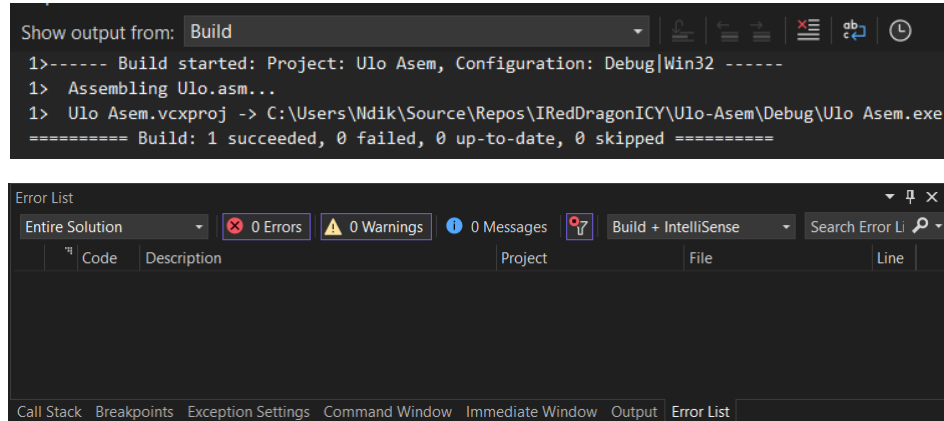
1000      InisialisasiUlangPermainan PROC
1001      mov     xPosisi[0], 45
1002      mov     yPosisi[1], 44
1003      mov     xPosisi[2], 43
1004      mov     yPosisi[3], 42
1005      mov     xPosisi[4], 41
1006      mov     yPosisi[5], 15
1007      mov     yPosisi[1], 15
1008      mov     yPosisi[2], 15
1009      mov     yPosisi[3], 15
1010      mov     yPosisi[4], 15
1011      mov     Skor, 0
1012      mov     TerakhirInputKarakter, ""
1013      mov     inputChar, ""
1014      dec     yPosisiDinding[3]
1015      Call     GetChar
1016      jmp     Permainan
1017      InisialisasiUlangPermainan ENDP

```

Gambar 12 Kode untuk proses menginisialisasi ulang permainan kembali (Sumber: Penulis)

Untuk lebih jelasnya membedakan kode program assembly untuk proses dan untuk menampilkan antarmuka adalah, Kode program assembly proses dideklarasikan dengan label, PROC ENDP atau PROC END.

2.4.5. Percobaan Aplikasi



Gambar 13 Hasil Compile Aplikasi Game Ulo Asem tanpa error (Sumber: Penulis)

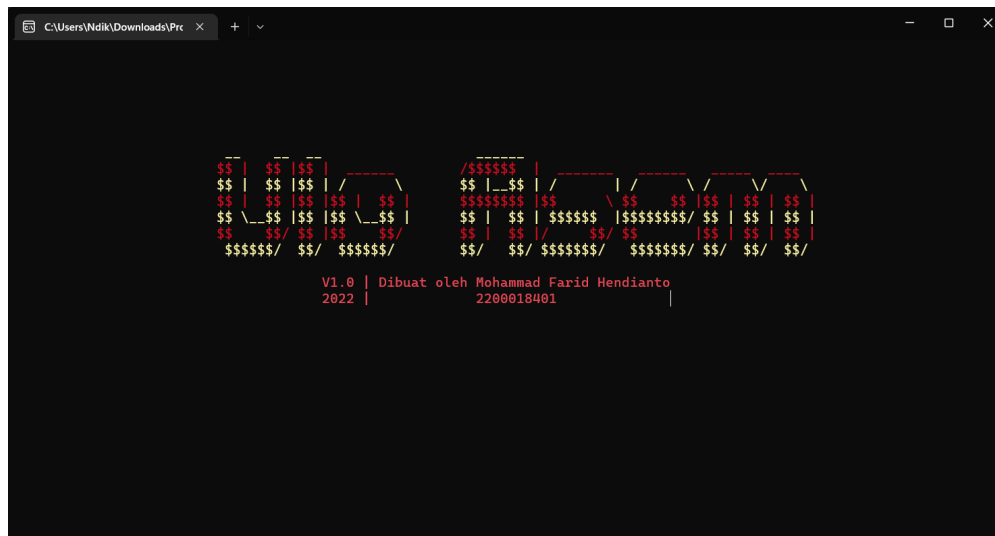
Setelah menulis kode program yang sesuai dengan yang diinginkan, saya melakukan compile terhadap program tersebut menggunakan nasm yang terinstall di dalam Visual Studio 2022. Proses compile berlangsung lancar tanpa terjadi error apapun. Hal ini menunjukkan bahwa kode program yang saya tulis sudah sesuai dengan sintaks yang benar dan dapat dieksekusi dengan baik.

Setelah proses compile selesai, saya menjalankan aplikasi game Ulo Asem yang telah dibuat. Aplikasi game tersebut berjalan dengan lancar sesuai dengan yang diharapkan. Pengguna dapat memainkan game dengan menggunakan tombol panah pada keyboard untuk mengatur arah gerakan ular, dan setiap kali ular berhasil memakan sebuah apel, skor akan bertambah sesuai dengan jumlah apel yang dimakan. Saat game berakhir, aplikasi game Ulo Asem akan menampilkan skor akhir yang telah diraih oleh pengguna.

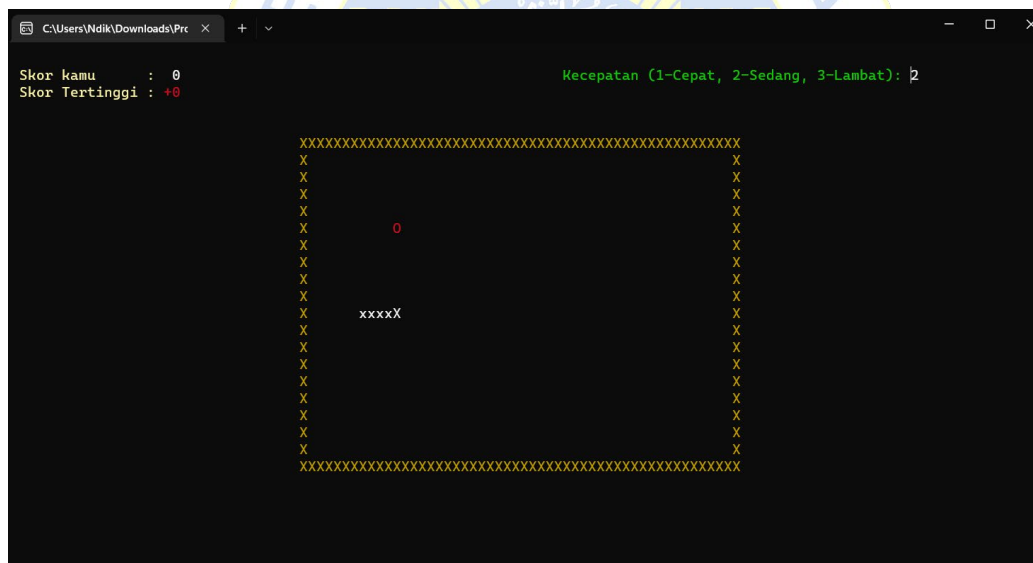
Saya juga melakukan beberapa percobaan dengan menambahkan fitur-fitur baru yang diinginkan seperti penambahan warna pada ular dan apel, serta penambahan mekanisme penyimpanan skor tertinggi. Semua percobaan tersebut juga berjalan dengan lancar dan sesuai dengan yang diharapkan.

Dengan demikian, dapat disimpulkan bahwa proses pembuatan aplikasi game Ulo Asem berjalan dengan lancar dan sesuai dengan yang diharapkan. Kode program yang saya tulis dapat di-compile dengan baik dan aplikasi game yang telah dibuat dapat berjalan dengan lancar sesuai dengan yang diinginkan. Selain itu, percobaan-percobaan yang saya lakukan dengan menambahkan fitur-fitur baru juga berjalan dengan lancar, sehingga dapat disimpulkan bahwa aplikasi game Ulo Asem ini sudah siap untuk dijadikan sebagai salah satu sarana belajar bahasa Assembly yang menyenangkan bagi para pemula.

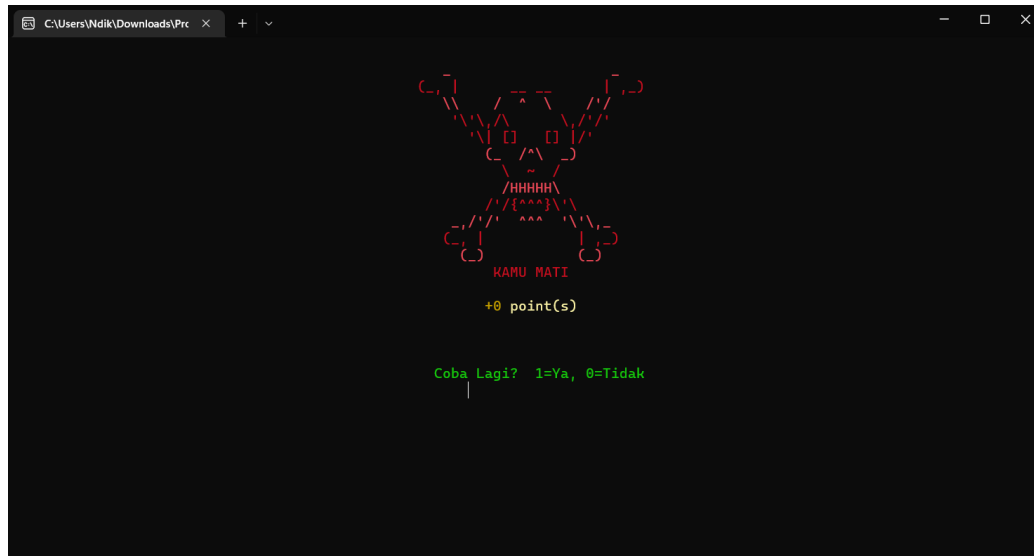
Berikut adalah tampilan program yang sesuai dengan dirancang



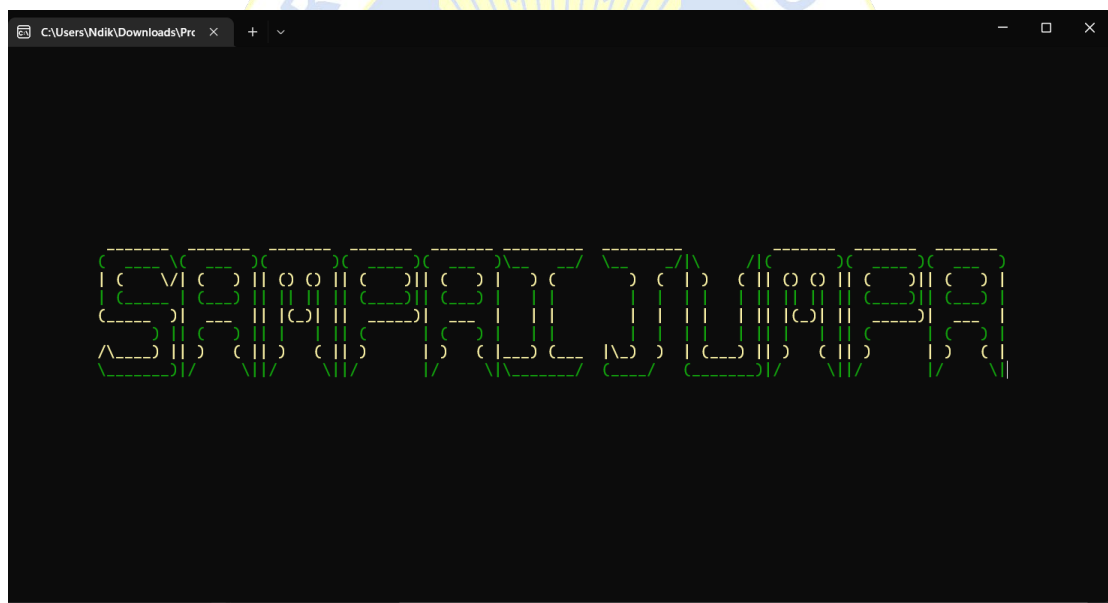
Gambar 14 Tampilan depan (Sumber: Penulis)



Gambar 15 Tampilan dalam Permainan (Sumber: Penulis)



Gambar 16 Tampilan saat Ulo mati (Sumber: Penulis)



Gambar 17 Tampilan Saat Keluar Permainan (Sumber: Penulis)

BAB IV

PENUTUP

3.1. Kesimpulan

Dokumentasi pembuatan aplikasi game Ulo Asem ini merupakan salah satu laporan yang membahas tentang proses pembuatan aplikasi game sederhana yang dibuat menggunakan bahasa pemrograman Assembly. Aplikasi game ini bertujuan untuk membantu para pemula dalam mempelajari bahasa Assembly dengan cara yang menyenangkan.

Dalam proses pembuatan aplikasi game Ulo Asem ini, kita dapat menggunakan tombol panah pada keyboard untuk mengatur arah gerakan ular. Ular tersebut akan bergerak mengikuti arah yang ditunjukkan oleh pengguna, dan akan terus bergerak hingga menabrak tembok atau tubuh sendiri. Selain itu, setiap kali ular berhasil memakan sebuah apel, maka skor akan bertambah dan panjang ular akan bertambah.

Saat game berakhir, aplikasi game Ulo Asem akan menampilkan skor akhir yang telah diraih oleh pengguna. Skor tersebut akan ditampilkan sesuai dengan jumlah apel yang telah dimakan oleh ular. Selain itu, aplikasi game ini juga menyediakan fitur reset yang dapat digunakan oleh pengguna untuk mengulang permainan dari awal.

Dengan demikian, dapat disimpulkan bahwa aplikasi game Ulo Asem ini dapat dijadikan salah satu sarana belajar bahasa Assembly yang menyenangkan bagi para pemula. Selain itu, aplikasi game ini juga dapat dijadikan sebagai hiburan sederhana yang dapat dinikmati oleh semua kalangan.

3.2. Kritik dan Saran

Adapun kritik dan saran yang dapat diberikan terkait dengan dokumentasi pembuatan aplikasi game Ulo Asem ini adalah sebagai berikut:

- Aplikasi game ini masih terbilang sederhana dan mungkin hanya cocok bagi pemula yang baru mempelajari bahasa Assembly. Penambahan fitur-fitur baru yang lebih kompleks mungkin dapat meningkatkan tingkat kesulitan dan menjadi lebih menarik bagi pengguna yang sudah terbiasa dengan bahasa Assembly.
- Selain menambah fitur-fitur baru, penggunaan bahasa pemrograman yang lebih modern seperti C atau C++ mungkin juga dapat menjadi pilihan yang lebih baik untuk pengembangan aplikasi game ini.

DAFTAR PUSTAKA

Link projek Aplikasi Game Ulo Asem:

<https://github.com/IRedDragonICY/Ulo-Asem>

Link Library Irvine:

<https://github.com/Eazybright/Irvine32>

