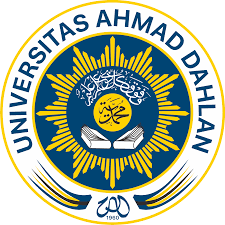
**PORTOFOLIO PEMBUATAN APLIKASI**

“**Dino Runner Assembly**”

Diajukan untuk memenuhi salah satu tugas mata kuliah Dasar Sistem Komputer

Yang diampu oleh :

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



**Disusun Oleh :**

**NAMA : M. MILKY GAZURA**

**NIM : 2200018396**

**KELAS : I**

**UNIVERSITAS AHMAD DAHLAN**

**FAKULTAS TEKNOLOGI INDUSTRI**

**PROGRAM STUDI INFORMATIKA**

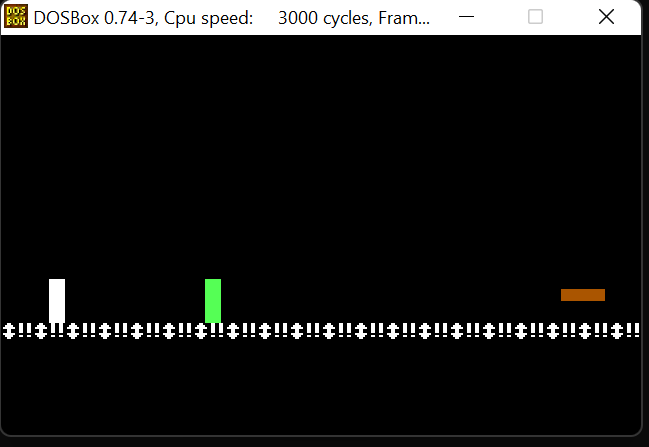
**2022/2023**

1.Cover: Identitas diri (NIM, NAMA, Kelas), judul projek, link projek di Gitlab/Github, capture tampilan awal aplikasi/ tampilan utama).

****

****

**Link Github : https://github.com/Ikydsk/Dinorasem-Dino-Runner-Assembly-Sederhana**

****

**IDENTITAS PROJEK**

|  |  |
| --- | --- |
| **NAMA PROGRAM** | Dino Runner Assembly Sederhana |
| **AUTHOR** | M. Milky Gazura |
| **DOSEN PENGAMPU** | Ali Tarmuji, S.T., M.Cs. |
| **MATA KULIAH** | Dasar Sistem Komputer |
| **PROGRAM STUDI** | Informatika |
| **FAKULTAS** | Teknologi Industri |



**Universitas Ahmad Dahlan Yogyakarta, Januari 2023**

**T.A. 2022/2023**

2. Ruang lingkup/ deskripsi dari projek

|  |
| --- |
| **Ruang Lingkup** |
| * Pelajar/Mahasiswa * Anak-anak * Programmer * Gamers * Youtuber * Masyarakat |
| **Deskripsi Program** |
| Dino Runner Assembly adalah sebuah permainan game klasik yang dapat dimainkan di Aplikasi Dosbox. Di dalam permainan ini, Anda berperan sebagai seorang dinosaurus yang harus menghindari obstaclenya yang terus bergerak ke kanan dengan menekan sebuah tombol untuk melompat. Tujuan utama dari permainan ini adalah untuk menghindari obstaclenya sebanyak mungkin. Pada dasarnya, Dino Runner Assembly adalah sebuah permainan yang sangat sederhana, tetapi sangat menyenangkan untuk dimainkan, terutama jika Anda menyukai tantangan saat bermain.  Terdapat tiga fitur yang terdapat pada Program ini yaitu:  1. Rintangan   * Rintangan yang terdiri dari objek fisik seperti dinding atau kaktus yang harus dihindari.   2. One Click   * Dalam permainan kamu hanya perlu mengklik tombol “w” untuk lompat dan “s” untuk menunduk   3. Easy to Play   * Pada saat GAMEOVER kamu hanya perlu mengklik tombol “w”, maka permainan akan Kembali ke awal tanpa perlu keluar dari aplikasi |

3. Daftar seluruh spesifikasi aplikasi sesuai studi kasus.

|  |  |
| --- | --- |
| **Spesifikasi Aplikasi** | |
| Sistem Operasi | Windows |
| Bahasa Pemrograman | Assembly |
| Jenis Aplikasi | Game Casual |
| Kemampuan Output | Dinosaurus dan rintangan pohon |
| Nama File Assembly | dino.asm |
| Ukuran File | 11 KB |
| Aplikasi Yang Digunakan | DOSBOX |

4. Rancangan antarmuka.

|  |
| --- |
| **Rancangan Antarmuka** |
|  |
| Aplikasi Dino Runner Assembly memiliki antarmuka yang sederhana dan mudah digunakan. Pengguna hanya mengklik tombol “w” untuk lompat, dan tombol “s” untuk menunduk. Apabila menabrak suatu rintangan pengguna bisa mengklik tombol “w” untuk kembali bermain. |

5. Diagram proses aplikasi (bisa DFD/ UML/ flowchart/ Proses bisnis, atau diagram

lainnya yang digunakan untuk \*\* menjelaskan jalannya program)

|  |
| --- |
| **Diagram Proses** |
|  |
| Berikut ini adalah flowchart yang menggambarkan alur kerja pada kode Assembly di atas:   1. Mulai 2. Panggil prosedur CLEAR\_SCREEN 3. Pergi ke label TIME\_CHECK 4. Panggil prosedur TIME\_CHECK 5. Periksa apakah terdapat perubahan waktu 6. Jika tidak terdapat perubahan waktu, kembali ke label TIME\_CHECK 7. Jika terdapat perubahan waktu, panggil prosedur COLLISION\_CHECK 8. Periksa apakah terdapat tabrakan 9. Jika tidak terdapat tabrakan, lanjut ke langkah 10 10. Panggil prosedur CACTUS\_MOVE, BIRD\_MOVE, dan DINO\_MOVE 11. Panggil prosedur CLEAR\_SCREEN 12. Panggil prosedur GROUND\_DRAW 13. Periksa apakah terdapat perubahan waktu sebesar 20ms 14. Jika tidak terdapat perubahan waktu, kembali ke label TIME\_CHECK 15. Jika terdapat perubahan waktu, dan kembali ke label TIME\_CHECK 16. Jika terdapat tabrakan, panggil prosedur GAMEOVER\_GAM 17. Periksa apakah pemain memasukkan input "w" 18. Jika tidak, tutup permainan 19. Jika ya, panggil prosedur RESET\_GAME dan kembali ke langkah 2 20. Selesai |

6. Penjelasan koding/ skrip program.

|  |
| --- |
| **Penjelasan Skrip Program** |
| Kode di atas merupakan bagian dari kode program yang ditulis dalam 2048 assembly. Bagian ini memperkenalkan sebuah segment 2048 “STACK” dan menyatakan bahwa segment tersebut merupakan tipe “PARA STACK”.  Segment “STACK” merupakan bagian dari memori 2048 yang digunakan sebagai tempat menyimpan data sementara saat program dijalankan. Data yang disimpan di stack ini biasanya berupa nilai-nilai yang dibutuhkan oleh program, seperti nilai, nilai fungsi, dll.  Segment “STACK” juga merupakan tempat dimana program menyimpan informasi tentang lokasi saat mengeksekusi fungsi. Saat program memanggil suatu fungsi, program akan menyimpan informasi tentang lokasi di stack, sehingga setelah fungsi tersebut selesai dijalankan, program dapat digunakan ke lokasi yang tepat. “.MODEL SMALL” merupakan perintah yang digunakan untuk memberitahu assembler tentang model memori yang akan digunakan dalam program. Ada beberapa model memori yang tersedia, di antaranya adalah “SMALL”, “COMPACT”, “MEDIUM”, “LARGE”, dan “HUGE”.  Model memori “SMALL” menyatakan bahwa program yang ditulis hanya akan menggunakan kecil dari memori. Dengan kata lain, program yang ditulis hanya akan menggunakan sejumlah kecil memori yang tersedia. Model memori ini biasanya digunakan untuk program-program yang tidak terlalu besar atau yang tidak membutuhkan banyak memori.  Bagian “DB 2048 DUP(“ “)” menyatakan bahwa segment STACK terdiri dari 2048 byte data yang diinisialisasi dengan karakter spasi. Jumlah byte yang disimpan di stack ini bisa berbeda-beda tergantung kebutuhan program. Misalnya, jika program membutuhkan lebih banyak tempat untuk menyimpan data, maka jumlah byte yang disediakan di stack bisa ditambah. Sebaliknya, jika program tidak membutuhkan banyak tempat penyimpanan, maka jumlah byte yang disediakan di stack bisa dikurangi.  Secara umum, tujuan dari bagian kode di atas adalah untuk menyediakan sebuah segment memori yang dapat digunakan oleh program sebagai tempat penyimpanan sementara saat program dijalankan. Segment tersebut disebut “STACK” dan terdiri dari 2048 byte data yang diinisialisasi dengan karakter spasi.    fungsi dari setiap program yang tercantum dalam kode tersebut:   1. LEBAR GAME: Berisi informasi mengenai lebar layar game. 2. DATA DINOSAURUS: Berisi informasi mengenai posisi, ukuran, dan lainnya mengenai dinosaurus dalam game. 3. DATA CACTUS: Berisi informasi mengenai posisi, ukuran, dan kecepatan cactus dalam game. 4. DATA BURUNG: Berisi informasi mengenai posisi, ukuran, dan lainnya mengenai burung dalam game. 5. VARIABEL WAKTU: Berisi informasi mengenai nilai dari waktu dalam game. 6. FLAGS: Berisi informasi mengenai flags yang akan digunakan dalam game. 7. VARIABEL SCORE: Berisi informasi mengenai skor yang diperoleh dalam game. 8. STRING UNTUK DIPRINT: Berisi informasi mengenai string yang akan diprint pada layar. 9. MACRO GAMEOVER\_GAM: Merupakan macro yang bertugas mengeprint "GAMEOVER" pada layar saat gameover.   Semua data yang didefinisikan di atas akan digunakan dalam program untuk mengontrol game.   * ;DATA BURUNG BIRD\_X DW 200H BIRD\_Y DW 85H BIRD\_X\_SIZE DW 16H BIRD\_Y\_SIZE DW 6H * ;VARIABEL WAKTU TIME\_VAR DB 05H * Variabel ini menyimpan kode waktu dalam game. TIME\_VAR disimpan dalam bentuk (05H), yang sama dengan 5 dalam satuan bilangan. * ;FLAGS FLAG1 DW 0H FLAG2 DW 0H EXIT\_FLAG DW 0H * Variabel-variabel ini merupakan flags atau bendera yang digunakan dalam game. FLAG1 dan FLAG2 mungkin digunakan untuk menandai suatu kondisi atau yang terjadi dalam game. EXIT\_FLAG mungkin digunakan untuk menandai kondisi dimana game harus berhenti atau keluar dari game. Semua ini disimpan dalam bentuk (0H), yang sama dengan 0 dalam satuan bilangan. * ;STRING UNTUK DIPRINT DISPLAY\_TEXT1 DB ‘0’,’$’ DISPLAY\_TEXT2 DB ‘0’,’$’ DISPLAY\_TEXT3 DB ‘0’,’$’ DISPLAY\_TEXT4 DB ‘0’,’$’ DISPLAY\_TEXTH1 DB ‘0’,’$’ DISPLAY\_TEXTH2 DB ‘0’,’$’ DISPLAY\_TEXTH3 DB ‘0’,’$’ DISPLAY\_TEXTH4 DB ‘0’,’$’ GAMEOVER\_TEXT DB ‘GAMEOVER’,’$’ * Variabel-variabel ini merupakan string yang mungkin digunakan untuk ditampilkan atau diprint pada layar game. DISPLAY\_TEXT1, DISPLAY\_TEXT2, DISPLAY\_TEXT3, dan DISPLAY\_TEXT4 mungkin digunakan untuk menampilkan skor yang telah dicapai dalam game. DISPLAY\_TEXTH1, DISPLAY\_TEXTH2, DISPLAY\_TEXTH3, dan DISPLAY\_TEXTH4 mungkin digunakan untuk menampilkan skor tertinggi yang pernah dicapai dalam game. GAMEOVER\_TEXT mungkin digunakan untuk menampilkan pesan “GAMEOVER” saat game berakhir atau keluar dari game.     Kode di atas merupakan definisi macro assembly GAMEOVER\_GAM yang digunakan untuk mengeprint pesan “GAMEOVER” pada layar game saat kondisi gameover terjadi. Macro ini terdiri dari beberapa perintah-perintah yang akan dijalankan secara berurutan dipanggil dalam kode program.  MOV AH,02H MOV BH,00H MOV DH,01H MOV DL,01H INT 10H  Perintah-perintah di atas digunakan untuk menggerakkan cursor pada game ke lokasi tertentu yang telah ditentukan. Nilai DH dan DL digunakan untuk menentukan koordinat x dan y dari lokasi yang akan dituju, sedangkan nilai BH digunakan untuk menentukan page number (halaman) yang akan digunakan.  MOV AH,09H LEA DX,GAMEOVER\_TEXT INT 21H  Perintah-perintah di atas digunakan untuk mengeprint string yang telah disimpan pada layer GAMEOVER\_TEXT. Nilai AH diset ke 09H yang merupakan kode untuk memanggil fungsi print string pada interrupt 21H. Kemudian, perintah LEA (Load Effective Address) digunakan untuk menyimpan alamat dari string pada layer GAMEOVER\_TEXT ke register DX. Setelah itu, perintah INT 21H digunakan untuk memanggil interrupt yang bertugas mengatur fungsi-fungsi input/output (I/O).  MOV AH,00H INT 16H  Perintah-perintah di atas digunakan untuk meminta input dari pengguna. Nilai AH diset ke 00H yang merupakan kode untuk memanggil fungsi baca tombol keyboard pada interrupt 16H. Setelah itu, perintah INT 16H digunakan untuk memanggil interrupt yang bertugas mengatur fungsi-fungsi input dari keyboard.  CMP AL,77h JNE EXITS  Perintah-perintah di atas digunakan untuk membandingkan nilai dari register AL dengan 77H. Jika nilai yang terdapat pada register AL tidak sama dengan 77H, maka program akan melanjutkan ke label EXITS. Jika nilai yang terdapat pada register AL sama dengan 77H, maka program akan melanjutkan ke perintah selanjutnya.       1. CALL CLEAR\_SCREEN: memanggil prosedur yang bertugas menghapus layar sehingga setiap frame akan terlihat baru. 2. TIME\_CHECK: prosedur yang bertugas mengecek waktu di layer setiap frame. 3. MOV AH,2CH: mengisi register AH dengan nilai 2CH sebagai kode untuk memanggil fungsi interrupt yang akan memberikan informasi waktu. 4. INT 21H: memanggil interrupt dengan kode 21H yang akan memberikan informasi waktu pada layar 5. CMP DL,TIME\_VAR: membandingkan nilai DL dengan TIME\_VAR untuk mengecek apakah terjadi perubahan waktu. 6. JE TIME\_CHECK: jika tidak terjadi perubahan waktu, maka kembali ke TIME\_CHECK. 7. INC SCORE\_TIME: menambah nilai SCORE\_TIME sebesar 1. 8. CMP SCORE\_TIME,5H: membandingkan nilai SCORE\_TIME dengan 5H (hexadecimal) untuk mengecek apakah terjadi perubahan waktu sebesar 20ms. 9. JNE NO\_SCORE: jika tidak terjadi perubahan waktu sebesar 20ms, maka ke NO\_SCORE. 10. MOV SCORE\_TIME,00H: mengisi SCORE\_TIME dengan 0. 11. NO\_SCORE: prosedur yang bertugas mengecek apakah terjadi tabrakan antara dino dan objek lainnya. 12. CALL COLLISION\_CHECK: memanggil prosedur yang bertugas mengecek apakah terjadi tabrakan antara dino dan objek lainnya. 13. CMP EXIT\_FLAG,00H: membandingkan nilai EXIT\_FLAG dengan 0 untuk mengecek apakah terjadi tabrakan. 14. JNE GAMEOVER: jika terjadi tabrakan, maka ke GAMEOVER. 15. JMP GAMEON: jika tidak terjadi tabrakan, maka ke GAMEON. 16. GAMEOVER: prosedur yang bertugas mengeluarkan pesan “gameover” pada layar. 17. CALL CACTUS\_MOVE: memanggil prosedur yang bertugas menggerakkan cactus dari kiri. 18. CALL BIRD\_MOVE: memanggil prosedur yang bertugas menggerakkan burung dari kiri. 19. MOV TIME\_VAR,DL: mengisi TIME\_VAR dengan nilai DL (informasi waktu yang didapat dari interrupt). 20. CALL CLEAR\_SCREEN: memanggil prosedur yang bertugas menghapus layar sehingga setiap frame akan terlihat baru. 21. CALL GROUND\_DRAW: memanggil prosedur yang bertugas mengeprint tanah ke layar. 22. CALL DINO\_MOVE: memanggil prosedur yang bertugas menggerakkan dino. 23. CMP FLAG2,1H: membandingkan nilai FLAG2 dengan 1H untuk mengecek apakah terjadi input ‘s’ (menunduk). 24. JE NUNDUK: jika terjadi input ‘s’, maka ke NUNDUK. 25. CALL DINO\_DRAW\_STAND: memanggil prosedur yang bertugas mengeprint dino yang berdiri ke layar. 26. JMP EXIT3: keluar dari prosedur. 27. NUNDUK: prosedur yang bertugas mengeprint dino yang menunduk ke bawah. 28. MOV FLAG2,00H: mengisi FLAG2 dengan 0. 29. EXIT3: keluar dari prosedur. 30. CALL CACTUS\_DRAW: memanggil prosedur yang bertugas mengeprint cactus ke layar. 31. CALL BIRD\_DRAW: memanggil prosedur yang bertugas mengeprint burung ke layar. 32. JMP TIME\_CHECK: menggerakkan ke TIME\_CHECK untuk memulai proses pada frame berikutnya. 33. EXITS: keluar dari program.     Kode Program di atas menggambar tanah dengan menggunakan bahasa assembly.      Kode program diatas digunakan untuk menggambarkan dino saat berdiri dan menunduk dengan Bahasa assembly.    Kode program di atas digunakan untuk menggerakkan dino, digerakkan ke atas maupun ke bawah (gravitasi).    Kode program di atas digunakan untuk menggambar rintangan kaktus, kemudian digerakkan dengan kecepatan-kecepatan tertentu.    Kode diatas digunakan untuk menggambar burung.    Kode tersebut digunakan untuk menggerakkan burung, apakah burung sudah dipojok kiri, dan kecepatan geraknya sama dengan kaktus.    Kode di atas bertujuan untuk mengecek terjadinya tabrakan antara dinosaurus dengan cactus atau burung di dalam game    Kode di atas bertujuan untuk mengatur ulang semua parameter ke kondisi awal di dalam game. Jadi, fungsi utama dari kode di atas adalah untuk mengatur ulang semua parameter ke kondisi awal sehingga game dapat dimulai dari awal.    Kode di atas bertujuan untuk mengatur agar dapat dibersihkan dengan latar belakang warna hitam. Jadi, fungsi utama dari kode di atas adalah untuk mengatur mode video dan warna latar belakang agar dapat dibersihkan dengan warna hitam. Ini berguna untuk membentuk efek penghapusan objek-objek yang ada di layar sehingga dapat dibuat animasi yang lebih dinamis. |

7. Penjelasan screenshot tampilan yang dihasilkan aplikasi.

|  |
| --- |
| **Screenshot Tampilan Aplikasi** |
|  |
| Tampilan awal program tersebut menampilkan sebuah dinosaurus, dan rintangan-rintangannya pohon(berwarna hijau), burung(berwarna coklat), serta sebuah tanah yang digambarkan dengan tanda seru dan tanda panah. |
|  |
| Tampilan ketika user telah menekan tombol “w” , maka dinosaurus akan melompat (JUMP) sesuai pada gambar diatas. |

|  |
| --- |
| **Screenshot Tampilan Aplikasi** |
|  |
| Tampilan ketika user menekan tombol “s” , maka dinosaurus akan menunduk seperti contoh pada gambar diatas. |
|  |
| Tampilan ketika dinosaurus menabrak sesuatu rintangan, maka akan ada tampilan GAMEOVER pada kiri pojok atas. |
|  |
| Kemudian apabila kita ingin main kembali, kita dapat mengklik tombol w saat ada tulisan GAMEOVER, maka program akan Kembali berjalan seperti semula. |

8. Penjelasan screenshot status unggah skrip di Gitlab/Github hingga projek final.

|  |
| --- |
| **Screenshot Status Unggahan Github** |
|  |
| Status unggahan pada Github tahap awal yang berisi file dino.asm beserta file README. |
|  |
| Status unggahan pada Github tahap akhir setelah diperbarui dengan program yang telah disempurnakan, file laporan, file presentasi, dan link video presentasi YouTube. |

9. Analisis pengerjaan projek (tinjauan dari sisi waktu, ketercapaian spesifikasi, biaya

yang dibutuhkan, kendala, tantangan masa depan, dan atau lain-lain jika ada).

|  |  |
| --- | --- |
| **Analisis Pengerjaan Proyek** | |
| Waktu Pengerjaan | 7 Hari |
| Ketercapaian Spesifikasi | * Kompatibel dengan sistem operasi Windows/DOS. * Dapat digunakan dengan berbagai jenis laptop yang terdapat aplikasi DOSBOX |
| Biaya yang Dibutuhkan | Tidak ada |
| Kendala | * Keterbatasan pengetahuan mengenai bahasa pemrograman yang digunakan. * Keterbatasan waktu pengerjaan disela-sela tugas dan aktivitas kuliah. * Keterbatasan koneksi internet kadang-kadang lelet. |
| Tantangan Masa Depan | * Penambahan fitur game Highscore pada saat melewati rintangan. * Penambahan fitur untuk jenis karakter-karakter dinosaurus yang dimainkan. * Penambahan fitur game online yang dapat dimainkan lebih dari satu orang * Penambahan fitur layar background dengan berbagai wallpaper background yang menarik. |