**LAPORAN**

**PROYEK PERANCANGAN PERMAINAN PAC-MAN DALAM C++**

Dosen Pengampu: Radinal Setyadinsa, S.Pd., M.T.I.



Disusun oleh

Ridho Anugrah Illahi 2310511088

Ruzbihan Salafi Bangko 2310511094

Dinda Marsa Alamsyah 2310511100

Rangga Budinovanto 2310511120

**PROGRAM STUDI S1 INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” JAKARTA**

**2023/2024**

**KATA PENGANTAR**

Puji syukur kehadirat kami panjatkan kehadirat Allah SWT. yang berkat Rahmat dan hidayat-Nya kami dapat menyelesaikan proyek akhir berupa laporan ini. Adapun topik dari proyek Pengenalan Pemrograman kami kali ini adalah membuat sebuah program permainan Pac-Man.

Laporan ini kami ajukan untuk memenuhi salah satu tugas mata kuliah Pengenalan Pemrograman dengan dosen pengampu Radinal Setyadinsa, S.Pd., M.T.I. Tidak pula lupa kami ucapkan terima kasih kepasa dosen pengampu mata kuliah pengenalan Pemrograman yang telah memberi bimbingan sehingga kami dapat menyelesaikan makalah ini.

Kami menyadari bahwa laporan ini masih jauh dari kata sempurna dikarenakan terbatasnya pengalaman dan pengetahuan yang kami miliki. Oleh karena keterbatasan waktu dan kemampuan kami, maka kritik dan saran yang membangun senantiasa kami harapkan demi kesempurnaan laporan ini. Semoga laporan ini dapat berguna bagi kami dan pihak yang berkepentingan pada umumnya.

Jakarta, 3 Desember 2023

Tim Penulis

**DAFTAR ISI**

KATA PENGANTAR…………………………………………………………………...……..i

DAFTAR ISI…………………………………………………………………….……..………ii

DAFTAR GAMBAR…………………………………………………….…………..………..iii

BAB I PENDAHULUAN……………………………………………….……….……….……1

1.1 Latar Belakang………………………..…..…………………...……………………..1

1.2 Tujuan………………..………………..……………………..….…....……………...1

1.3 Peran Anggota………………………….…………………..……...…….………..….1

BAB II PEMBAHASAN…………………………….…………………………………………2

2.1 Cara Kerja………………………………………………………………..…………..2

2.2 Proses Aplikasi……………………………………………………………………...10

2.3 Link……………………………………………………………………....................13

BAB III PENUTUP…………………………………………………………………………...14

3.1 Kesimpulan………………………………………………………………………....14

3.2 Saran………………………………………………………………………………..14

DAFTAR PUSTAKA………………………………………………………………………...15

**DAFTAR GAMBAR**

Gambar 2.1……………………………….…………………………………………………….2

Gambar 2.2……………………………….…………………………………………………….3

Gambar 2.3……………………………….…………………………………………………….4

Gambar 2.4……………………………….…………………………………………………….4

Gambar 2.5……………………………….…………………………………………………….5

Gambar 2.6……………………………….…………………………………………………….5

Gambar 2.7……………………………….…………………………………………………….6

Gambar 2.8……………………………….…………………………………………………….6

Gambar 2.9……………………………….…………………………………………………….7

Gambar 2.10……………………………….…………………………………………………...7

Gambar 2.11……………………………….…………………………………………………...8

Gambar 2.12……………………………….…………………………………………………...8

Gambar 2.13……………………………….…………………………………………………...9

Gambar 2.14……………………………….………………………………………………….10

Gambar 2.15……………………………….………………………………………………….10

Gambar 2.16……………………………….………………………………………………….11

Gambar 2.17……………………………….………………………………………………….12

Gambar 2.18……………………………….………………………………………………….12

**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang**

Pada masa ini, kita pasti sering sekali melihat *video game* dalam kehidupan kita. *Video game* atau permainan video, disusun dari dua kata, yaitu permainan dan video. Menurut Zimmerman (2004), permainan adalah aktivitas interaktif sukarela, di mana satu atau lebih pemain mengikuti aturan yang membatasi perilaku mereka, memberlakukan konflik buatan yang berakhir dengan hasil yang dapat diukur. Menurut Esposito (2005), *video game* adalah permainan yang kita mainkan berkat peralatan audiovisual dan yang dapat didasarkan pada sebuah cerita. Sejarah permainan video sendiri telah ada sejak tahun 1950-an. Salah satu contoh permainan video adalah Pac-Man.

Pac-Man adalah *franchise video game* Jepang yang dikembangkan, diterbitkan, dan dimiliki oleh Bandai Namco Entertainment, penerbit *video game* yang sebelumnya dikenal sebagai Namco. Pac-Man awalnya dirilis sebagai permainan *arcade* di Jepang pada 22 Mei 1980. Meskipun Pac-Man merupakan permainan lama, Pac-Man masih diketahui banyak orang sampai sekarang dan dianggap sebagai permainan yang ikonik.

Salah satu alasan kami memilih Pac-Man sebagai program yang dibuat untuk tugas proyek akhir ini adalah karena popularitasnya. Selain itu, Pac-Man secara *gameplay* juga cukup sederhana, karakter pemain harus memakan poin pada peta dan menghindari musuh agar tidak mendapatkan *game over*. Dalam laporan ini, kami gunakan konsep itu untuk lebih memahami bagaimana C++ dapat digunakan dalam program ini dan fungsi apa saja yang dibutuhkan.

* 1. **Tujuan**

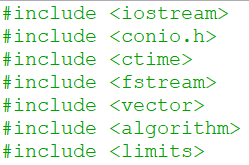
1. Memahami cara membuat sebuah program Pac-Man menggunakan bahasa C++.
2. Mengetahui bagaimana setiap bagian dari game Pac-Man bekerja.
   1. **Peran Anggota**

|  |  |
| --- | --- |
| **Nama** | **Peran** |
| Ridho Anugrah Illahi | Pembuatan Source Code dan Makalah |
| Ruzbihan Salafi Bangko | Pembuatan Source Code dan Powerpoint |
| Dinda Marsa Alamsyah | Perancang Utama Pembuatan Makalah |
| Rangga Budinovanto | Perancang Utama Pembuatan Makalah |

**BAB II**

**PEMBAHASAN**

* 1. **Cara Kerja**
* **Header**

****

**Gambar 2.1** Daftar header yang digunakan

1. iostream:

Digunakan untuk operasi input dan output di console.

1. conio.h:

Library khusus untuk operasi input/output console, seperti \_getch().

1. ctime:

Memberikan fungsi-fungsi untuk bekerja dengan waktu dan tanggal, digunakan untuk menghasilkan seed untuk fungsi random.

1. fstream:

Menyediakan fasilitas untuk operasi input/output ke file, digunakan untuk menyimpan dan membaca data skor pemain.

1. vector:

Memperkenalkan struktur data vector yang dinamis, digunakan untuk menyimpan dan mengelola skor pemain pada leaderboard.

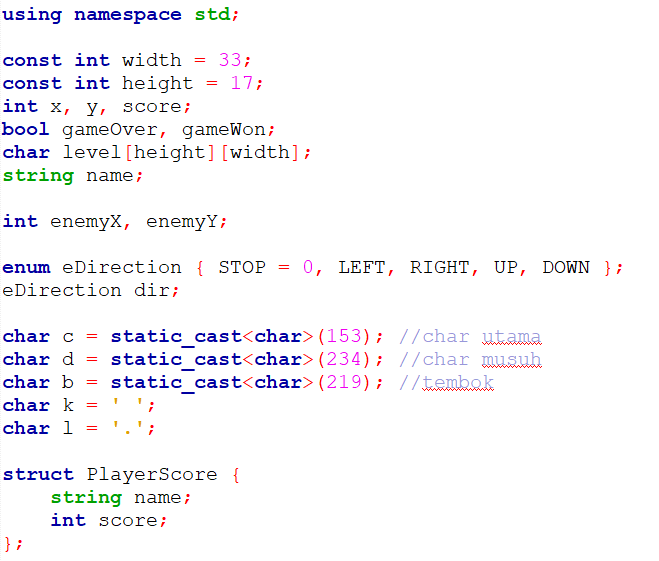
1. algorithm:

Menyediakan beberapa fungsi algoritma umum, digunakan di sini untuk mengurutkan skor pemain pada leaderboard.

1. limits:

Mengandung definisi nilai maksimum dan minimum untuk tipe data tertentu.

* **Pendeklarasian dan Struct**



**Gambar 2.2** Variabel dan Deklarasi

1. width dan height:

Mendeklarasikan konstanta yang menentukan ukuran peta permainan.

1. x, y, score:

Variabel untuk menyimpan posisi Pac-Man (`x` dan `y`) dan skor pemain (`score`).

1. gameOver, gameWon:

Variabel boolean yang menandakan apakah permainan berakhir atau dimenangkan.

1. level:

Array dua dimensi yang merepresentasikan peta permainan. Digunakan untuk menyimpan karakter seperti tembok (`b`), Pac-Man (`c`), musuh (`d`), dan elemen lainnya.

1. name:

Variabel untuk menyimpan nama pemain.

1. enemyX, enemy:

Variabel untuk menyimpan posisi musuh (Eater).

1. enum eDirection { STOP = 0, LEFT, RIGHT, UP, DOWN }; eDirection dir;

Enumerasi yang mendefinisikan arah pergerakan (berhenti, kiri, kanan, atas, bawah), dan variabel dir yang menyimpan arah pergerakan saat ini.

1. Karakter-karakter yang digunakan dalam permainan:

- `c`: Karakter untuk Pac-Man.

- `d`: Karakter untuk musuh (Eater).

- `b`: Karakter untuk tembok.

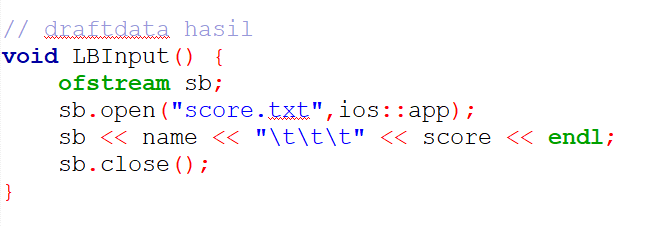
- `k`: Spasi.

- `l`: Titik atau dot.

1. PlayerScore:

untuk menyimpan nama pemain (name) dan skor (score).

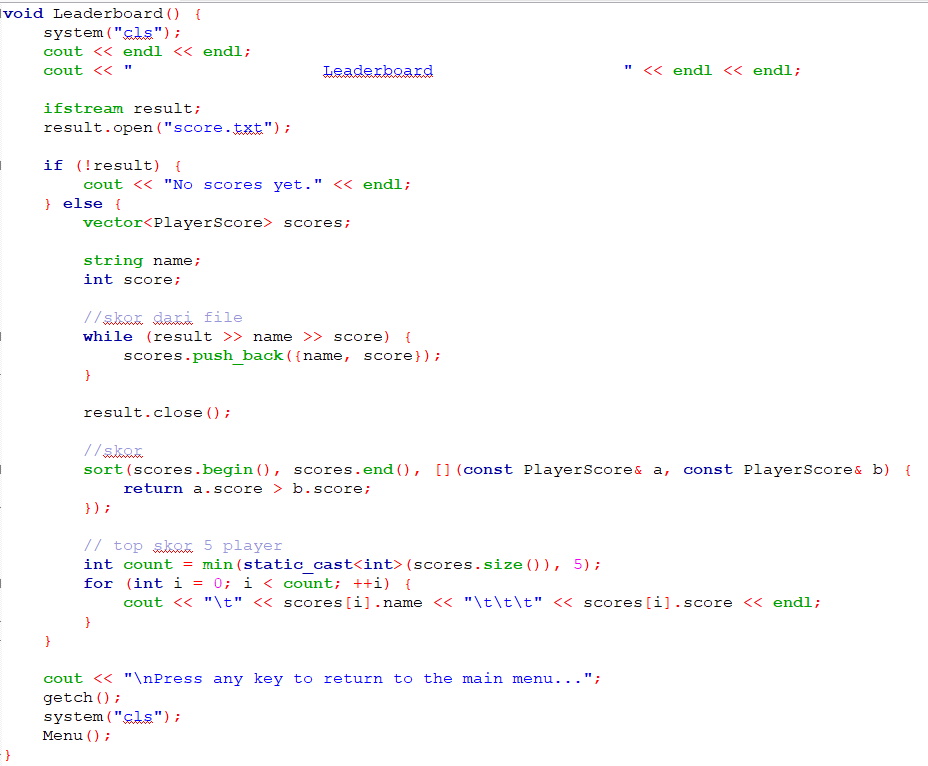
* **Void LBInput()**



**Gambar 2.3** Void LBInput()

Digunakan untuk memasukkan data skor ke file leaderboard, dalam file “score.txt”.

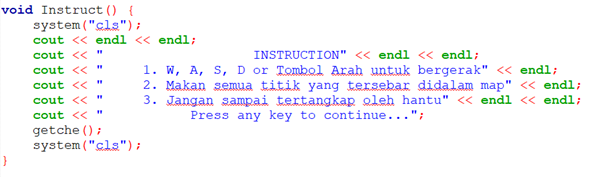
* **Void Leaderboard()**



**Gambar 2.4** Void Leaderboard()

Untuk menampilkan peringkat pemain berdasarkan skor tertinggi, dengan data disimpan dalam file “score.txt”.

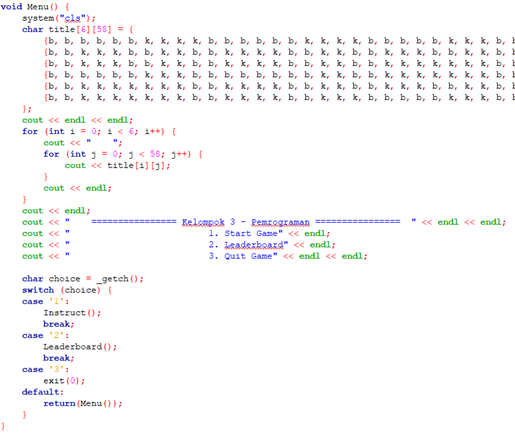
* **Void Instruct()**



**Gambar 2.5** Void Instruct()

Untuk memberikan petunjuk permainan kepada pemain.

* **Void Menu()**



**Gambar 2.6** Void Menu()

Untuk menampilkan menu utama dan mengarahkan ke pilihan yang dipilih.

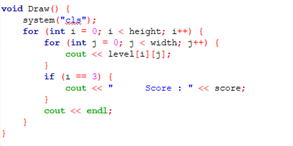
* **Void Setup()**



**Gambar 2.7** Void Setup()

Untuk menginisialisasi kondisi awal permainan, termasuk posisi pemain dan musuh, serta peta.

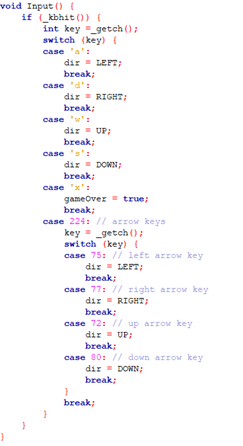
* **Void Draw()**



**Gambar 2.8** Void Draw()

Untuk Menggambar peta permainan, skor, dan elemen lainnya.

* **Void Input()**



**Gambar 2.9** Void Input()

Untuk Mengambil input dari pemain untuk menggerakkan atau keluar dari permainan.

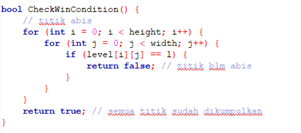
* **Void MoveEnemy()**



**Gambar 2.10** Void MoveEnemy()

Untuk menggerakkan musuh (hantu) dalam Pac-Man. Logika pemilihan langkah didasarkan pada posisi Pac-Man dan ketersediaan langkah di sekitar musuh. Jika semua arah terblokir, musuh akan memilih langkah acak. Jika tidak, musuh akan memilih langkah yang paling efisien berdasarkan jarak ke Pac-Man, baik secara horizontal maupun vertikal.

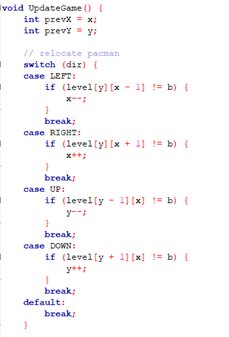
* **Void CheckWinCondition()**



**Gambar 2.11** Void CheckWinCondition()

Untuk memeriksa apakah pemain telah memenangkan permainan dengan mengumpulkan semua titik.

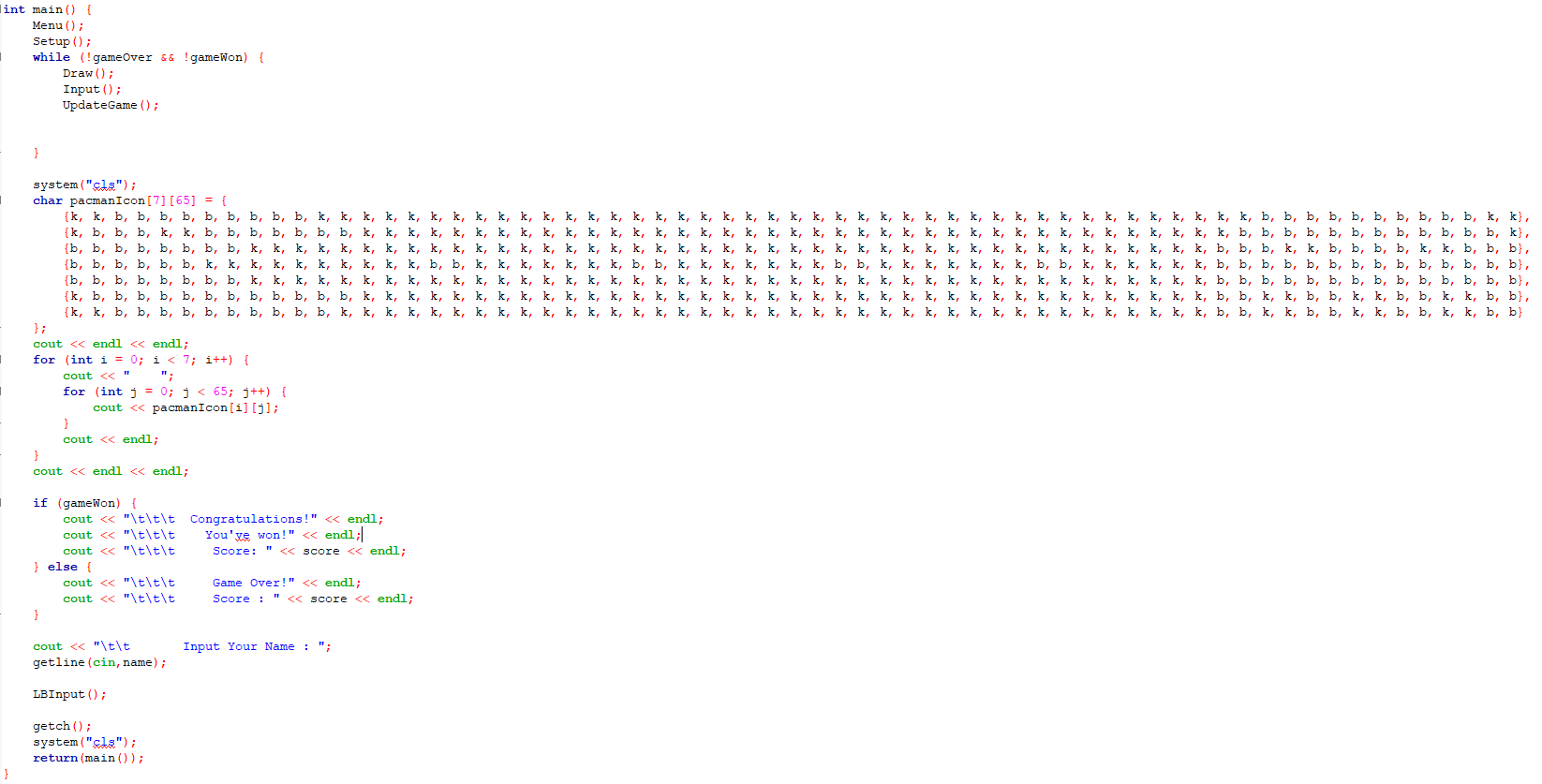
* **Void UpdateGame()**



**Gambar 2.12** Void UpdateGame()

Digunakan untuk mengatur pembaruan kondisi permainan, termasuk pergerakan pemain, pembaruan skor, dan pemeriksaan kondisi kemenangan atau kekalahan.

* **Int main()**



**Gambar 2.13** Int main()

Fungsi ini menjadi titik awal program, memanggil fungsi-fungsi lain dan mengatur alur utama permainan.

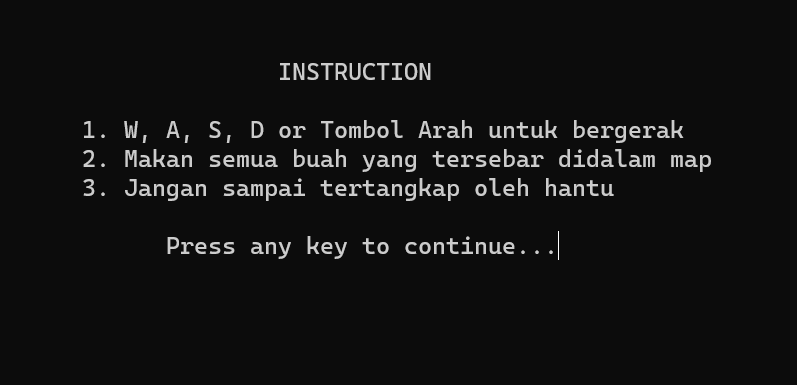
* 1. **Proses Aplikasi**
* **Menu Utama**



**Gambar 2.14** Menu Utama

Pada bagian ini akan ditampilkan output untuk bagian dari source code ‘void menu()’. Seperti yang kita ketahui pada fungsi main, line pertama diawali dengan pendeklarasian fungsi void menu. Disini ditampilkan judul serta perancang dari projek tersebut. Di bawahnya ditampilkan beberapa menu yang diantaranya yaitu Start game (untuk memulai permainan), Leaderboard (Menampilkan 5 pemain dengan skor tertinggi yang pernah tercatat), dan Quit game (Menutup Permainan).

* **Instruksi**



**Gambar 2.15** Instruksi

Apabila menekan tombol 1 pada menu (yang merupakan tombol untuk memasuki permainan), output akan diarahkan pada bagian instruksi terlebih dahulu sebelum masuk ke permainan, seperti yang tertera pada switch case 1 void menu yang membawa kita pada ‘void instruct()’, pada bagian ini akan dijelaskan cara kerja permainan tersebut. Dengan menekan tombol apapun akan membawa pengguna pada permainannya.

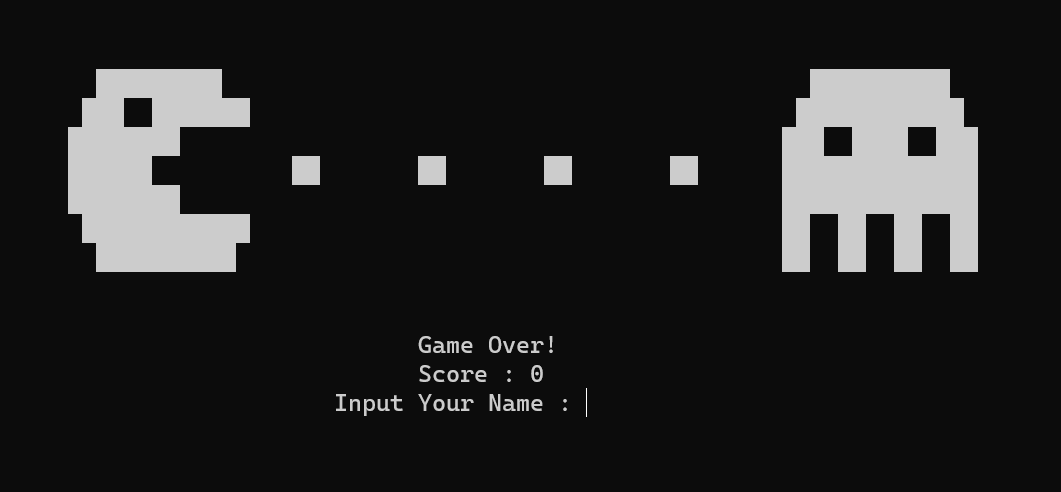
* **Game Screen**



**Gambar 2.16** Game Screen

Setelah membaca instruksi permainan, game pun dimulai. Pemetaan arena permainan yang dilakukan pada ‘void Setup()’ akan digambar pada output dengan fungsi ‘void Draw()’, hal yang sama terjadi pada titik mulai pemain, musuh hingga akumulasi score. Pergerakan pemain dan musuh akan dicatat frame by frame dengan ‘void Input()’ yang menjadi fungsi dalam pengontrolan karakter pemain, sedangkan pergerakan musuh dilakukan secara random oleh AI dengan mempertimbangkan jalur efektif dalam mengejar pemain sesuai yang tertera pada ‘void MoveEnemy()’. Tiap titik yang diambil oleh pemain akan mendapatkan akumulasi score sebesar 10 setiap titiknya. Untuk setiap gerakannya akan di proses pada ‘void UpdateGame()’ dengan refresh rate yang tertera pada fungsi main (tepatnya deklarasi \_sleep()), yaitu 400 milidetik. Perputaran dari fungsi-fungsi tersebut akan terus berulang hingga deklarasi gamewon atau gameover menjadi true, sebagai mana yang kita ketahui pada ‘void CheckWinCondition()’ apabila seluruh titik pada map telah habis, maka pemain dinyatakan menang (GameWon = True). Sedangkan apabila pemain bersentuhan dengan enemy, maka pemain dinyatakan kalah (GameOver = True).

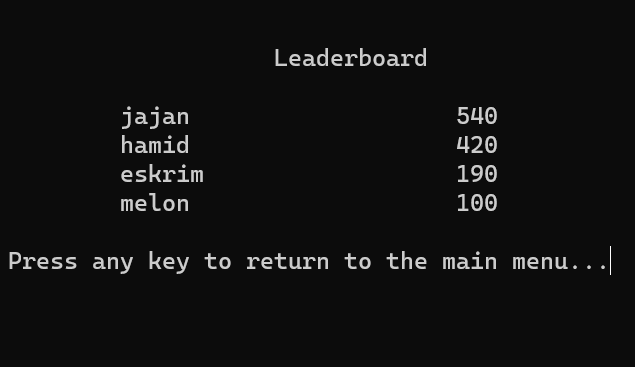
* **Ending Screen**



**Gambar 2.17** Ending Screen

Permainan diakhiri dengan 2 kemungkinan, yaitu apabila pemain kalah (Game Over), atau pemain menang (Game Won). Hasil output yang keluar pada ending screen tergantung pada kondisi permainan tersebut berakhir. Pada ending screen ini juga terdapat total kumulatif score dari permainan yang dilakukan, disertai dengan penginputan nama pemain untuk diproses pada Leaderboard. Nama pemain dan scorenya akan di input pada file score.txt yang telah disiapkan menggunakan metode ofstream seperti yang tertera pada ‘void LBInput()’. Setelah pengisian nama, pemain akan diarahkan kembali menuju halaman menu utama.

* **Leaderboard**



**Gambar 2.18** Leaderboard

Setelah beberapa skor akhir tercatat pada score.txt, data pemain dan skornya akan diproses pada halaman leaderboard. Halaman ini dapat diakses melalui menu utama dengan menekan tombol 2 pada keyboard, dimana kita akan diarahkan pada ‘void Leaderboard()’. Pada halaman ini, akan ditampilkan 5 pemain dengan skor tertinggi yang tercatat sepanjang permainan dengan melakukan pernyortiran secara descended (menurun). File dari score.txt yang merupakan tempat penyimpanan data pemain akan dibaca menggunakan deklarasi ifstream menggunakan sistem vector, sehingga data tersebut akan tercantum pada output bagian leaderboard tersebut.

* 1. **Link**

Bentuk .cpp dari program kami juga telah kami unggah pada laman (GitHub/Google Drive) dengan link berupa (link):

<https://github.com/JotunheimxAvalon/Laporan-Akhir-Pemrograman.git>

**BAB III**

**PENUTUP**

* 1. **Kesimpulan**

Dalam proyek akhir ini, kami berhasil mengimplementasikan versi sederhana dari permainan Pac-Man menggunakan bahasa pemrograman C++. Kami memulai dengan merancang struktur dasar permainan, termasuk inisialisasi peta, pemain, dan musuh. Melalui penggunaan fungsi-fungsi seperti Setup, Draw, Input, dan UpdateGame, kami berhasil membuat alur permainan yang interaktif dan menyenangkan.

Penggunaan C++ sebagai bahasa pemrograman memberikan kami fleksibilitas dalam mengelola logika permainan, penggambaran peta, dan pengaturan skor pemain. Konsep-konsep dasar dalam pemrograman, seperti penggunaan array, struct, dan file I/O, juga terintegrasi dengan baik dalam pembuatan program ini.

Pada bagian Pembahasan, kami memperkenalkan setiap bagian program serta memberikan penjelasan, mulai dari header, pendeklarasian variabel, fungsi-fungsi seperti LBInput, Leaderboard, hingga fungsi-fungsi yang mengatur logika permainan seperti MoveEnemy dan CheckWinCondition.

* 1. **Saran**

Sebagai penutup proyek ini, kami merangkum beberapa saran yang dapat kami gunakan untuk pengembangan lebih lanjut kedepannya:

1. Sebagai proyek sederhana, pengembangan grafis dapat ditawarkan untuk memberikan tampilan visual yang lebih menarik.
2. Meskipun program berjalan dengan baik, selalu ada ruang untuk optimisasi kode. Pemahaman lebih lanjut tentang algoritma pencarian musuh atau struktur data yang efisien dapat memperbaiki performa.
3. Untuk membuat permainan lebih kompleks, pertimbangkan untuk menambahkan elemen-elemen baru seperti level-level yang berbeda, daya tambahan, atau musuh dengan karakteristik khusus.

**DAFTAR PUSTAKA**

1. Esposito, N. (2005). A Short and Simple Definition of What a Videogame Is.
2. Zimmerman, E. (2004). Narrative, interactivity, play, and games: Four naughty concepts in need of discipline. *First person: New media as story, performance, and game*, *154*.