**Menampilkan Data History Match dalam Sebuah Game**

LAPORAN STUDI KASUS TUGAS AOL

MATA KULIAH COMP6360004 \_ ALGORITHM AND PROGRAMMING

KELAS LC20



Oleh :

2602211902 – ARI DANDA WIDHIA PUTRA

Semester Ganjil 2022/2023

MALANG

**DAFTAR ISI**

HALAMAN SAMPUL………………………………………………………………………………….i

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

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

BAB II USULAN DESAIN PROGRAM……………………………………………………………….2

BAB III PROGRAM DAN PENJELASANNYA………………………………………………………4

BAB IV TAMPILAN HASIL………………………………………………………………….............. 7

BAB 5 KESIMPULAN………………………………………………………………………………..11

**BAB I Pendahuluan**

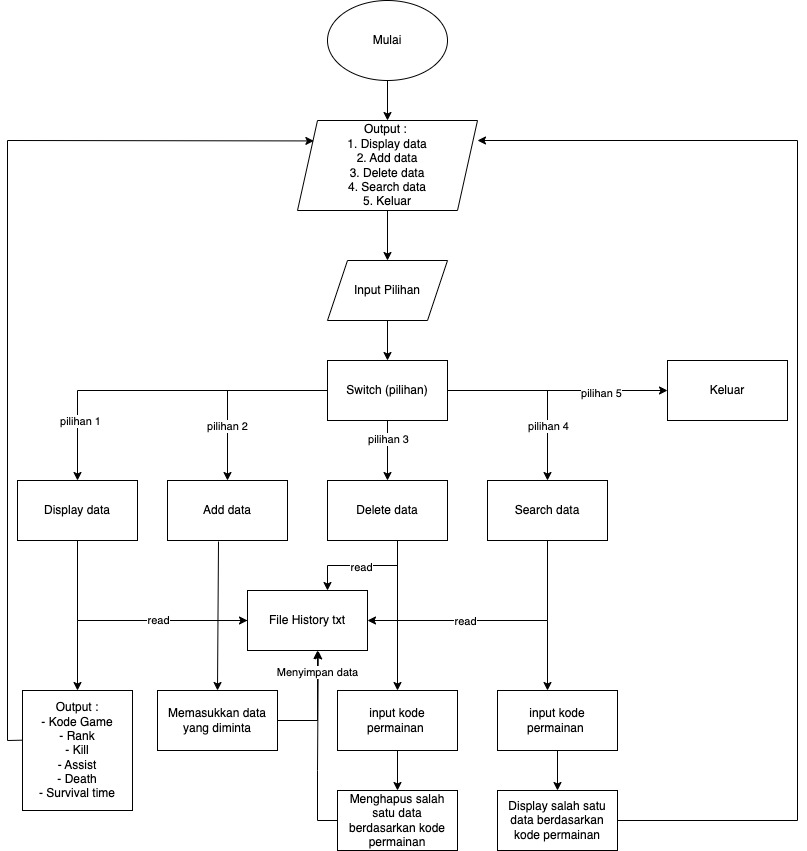
Program ini adalah suatu program yang menyimpan dan mengelola data tentang pertandingan game, menggunakan struct game untuk menyimpan data seperti kode pertandingan, peringkat, kill, assist, death, dan waktu bertahan hidup. Program ini memiliki tiga fungsi utama: display\_data, add\_data, dan delete\_data. Fungsi display\_data digunakan untuk menampilkan data yang tersimpan dalam file "history.txt" dalam format tabel. Fungsi add\_data digunakan untuk menambahkan data baru ke file "history.txt". Fungsi delete\_data digunakan untuk menghapus data dari pertandingan dari file "history.txt" dengan mencari kode pertandingan dan menghapusnya dari file.

Untuk menyelesaikan proyek ini, diperlukan beberapa struktur kendali seperti perulangan dan seleksi. Fungsi display\_data menggunakan perulangan untuk membaca data dari file "history.txt" baris perbaris dan menampilkannya dalam format tabel. Fungsi delete\_data juga menggunakan perulangan untuk mencari kode pertandingan dalam file "history.txt" dan menghapus data yang sesuai.

Tipe data yang diperlukan untuk proyek ini adalah string dan integer. String digunakan untuk menyimpan kode pertandingan, waktu bertahan hidup, dan informasi lain yang tidak dapat dinyatakan dalam bentuk angka. Integer digunakan untuk menyimpan peringkat, kill, assist, dan death.

Media penyimpanan yang diperlukan untuk proyek ini adalah file teks. File "history.txt" digunakan untuk menyimpan data dari setiap pertandingan, dan digunakan oleh fungsi display\_data, add\_data, dan delete\_data. File "temp.txt" digunakan sebagai file sementara saat hendak menghapus data dari file "history.txt".

**BAB II Usulan Desain Program**

****

Program dimulai dengan menampilkan menu utama kepada pengguna yang terdiri dari pilihan untuk menampilkan data, menambah data dan menghapus data.

* Jika pengguna memilih opsi untuk menampilkan data, program melanjutkan ke fungsi Tampilkan Data, yang membaca data dari file "history.txt" dan menampilkannya dalam format tabel.
* Jika pengguna memilih opsi untuk menambahkan data, program melanjutkan ke fungsi Tambah Data, yang meminta pengguna memasukkan data yang diminta, lalu menulis data ke file "history.txt".
* Jika pengguna memilih opsi untuk menghapus data, program melanjutkan ke fungsi Hapus Data, yang meminta pengguna memasukkan kode permainan, lalu menghapus data di file "history.txt" dengan menemukan kode yang cocok dan menghapusnya .
* Jika pengguna memilih opsi Cari Data, yang meminta pengguna mamsukkan kode permainan, lalu mencocokan data pada “history.txt” dan menampilkan data history match berdasarkan kode permainan tersebut.

Program berputar kembali ke menu utama setelah setiap fungsi, memungkinkan pengguna untuk membuat banyak pilihan dan melakukan banyak tindakan.

**BAB III Program dan Penjelasannya**

// Program dimulai dengan menyertakan header stdio.h dan string.h

#include <stdio.h>

#include <string.h>

// Deklarasi struktur data game

struct game {

char code[10]; // Deklarasi variabel karakter untuk menyimpan kode game

int rank; // Deklarasi variabel integer untuk menyimpan peringkat

int kill; // Deklarasi variabel integer untuk menyimpan kill

int assist; // Deklarasi variabel integer untuk menyimpan assist

int death; // Deklarasi variabel integer untuk menyimpan death

char survival\_time[10]; // Deklarasi variabel karakter untuk menyimpan survival time

};

// Deklarasi fungsi display\_data yang digunakan untuk menampilkan data

void display\_data() {

char temp\_code[10]; // Deklarasi variabel karakter untuk menyimpan kode game sementara

struct game data; // Deklarasi variabel data untuk menyimpan data

FILE \*file = fopen("history.txt", "r"); // Membuka file history.txt

// Menampilkan header

printf("\n==============================================================\n");

printf("\t\t HISTORY MATCH\n");

printf("==============================================================\n");

printf("\n Game Code\tRank\tKill\tAssist\tDeath\tSurvival Time\n");

// Melakukan perulangan sampai membaca EOF (end of file)

while (fscanf(file, "%s %d %d %d %d %s\n", temp\_code, &data.rank, &data.kill, &data.assist, &data.death, data.survival\_time) != EOF) {

// Menampilkan data yang dibaca

printf(" %s\t\t%d\t%d\t%d\t%d\t%s\n\n", temp\_code, data.rank, data.kill, data.assist, data.death, data.survival\_time);

}

// Menutup file

fclose(file);

}

// Deklarasi fungsi add\_data yang digunakan untuk menambah data

void add\_data(struct game data) {

FILE \*file = fopen("history.txt", "a"); // Membuka file history.txt dengan mode a (append)

// Menambahkan data

fprintf(file, "%s %d %d %d %d %s\n", data.code, data.rank, data.kill, data.assist, data.death, data.survival\_time);

// Menutup file

fclose(file);

}

// Deklarasi fungsi delete\_data yang digunakan untuk menghapus data

void delete\_data(char code[10]) {

char temp\_code[10]; // Deklarasi variabel karakter untuk menyimpan kode game sementara

char temp[1000]; // Deklarasi variabel karakter untuk menyimpan data sementara

struct game data; // Deklarasi variabel data untuk menyimpan data

FILE \*file = fopen("history.txt", "r"); // Membuka file history.txt

FILE \*temp\_file = fopen("temp.txt", "w"); // Membuka file temp.txt dengan mode w (write)

// Melakukan perulangan sampai membaca EOF (end of file)

while (fscanf(file, "%s %d %d %d %d %s\n", temp\_code, &data.rank, &data.kill, &data.assist, &data.death, data.survival\_time) != EOF) {

// Jika kode game yang dicari tidak sama dengan kode yang dibaca

if (strcmp(temp\_code, code) != 0) {

// Masukkan data ke dalam temp

sprintf(temp, "%s %d %d %d %d %s\n", temp\_code, data.rank, data.kill, data.assist, data.death, data.survival\_time);

// Tulis data ke dalam file temp.txt

fputs(temp, temp\_file);

}

}

// Menutup file

fclose(file);

fclose(temp\_file);

remove("history.txt"); // Menghapus file history.txt

rename("temp.txt", "history.txt"); // Mengubah nama file temp.txt menjadi history.txt

}

// Deklarasi fungsi search\_data yang digunakan untuk mencari data

void search\_data(char code[10]) {

char temp\_code[10]; // Deklarasi variabel karakter untuk menyimpan kode game sementara

struct game data; // Deklarasi variabel data untuk menyimpan data

int found = 0; // Deklarasi variabel found bernilai 0

FILE \*file = fopen("history.txt", "r"); // Membuka file history.txt

// Melakukan perulangan sampai membaca EOF (end of file)

while (fscanf(file, "%s %d %d %d %d %s\n", temp\_code, &data.rank, &data.kill, &data.assist, &data.death, data.survival\_time) != EOF) {

// Jika kode game yang dicari sama dengan kode yang dibaca

if (strcmp(temp\_code, code) == 0) {

// Menampilkan header

printf("\n==============================================================\n");

printf("\t\t HISTORY MATCH\n");

printf("==============================================================\n");

printf("\n Game Code\tRank\tKill\tAssist\tDeath\tSurvival Time\n");

// Menampilkan data yang ditemukan

printf(" %s\t\t%d\t%d\t%d\t%d\t%s\n\n", temp\_code, data.rank, data.kill, data.assist, data.death, data.survival\_time);

// Ubah nilai found menjadi 1

found = 1;

break;

}

}

// Jika data tidak ditemukan

if (!found) {

printf("Data tidak ditemukan.\n");

}

// Menutup file

fclose(file);

}

// Deklarasi fungsi main

int main() {

int choice; // Deklarasi variabel choice untuk menyimpan pilihan menu

struct game data; // Deklarasi variabel data untuk menyimpan data

char code[10]; // Deklarasi variabel karakter untuk menyimpan kode game

// Melakukan perulangan sampai user memilih pilihan 5

while (1) {

// Menampilkan menu

printf("Menu:\n");

printf("1. Tampilkan data\n");

printf("2. Tambah data\n");

printf("3. Hapus data\n");

printf("4. Cari data\n");

printf("5. Keluar\n");

printf("Pilih: ");

scanf("%d", &choice);

// Menjalankan perintah sesuai dengan pilihan

switch (choice) {

case 1:

display\_data();

break;

case 2:

printf("\nMasukkan kode permainan: ");

scanf("%s", data.code);

printf("Masukkan peringkat: ");

scanf("%d", &data.rank);

printf("Masukkan kill: ");

scanf("%d", &data.kill);

printf("Masukkan assist: ");

scanf("%d", &data.assist);

printf("Masukkan death: ");

scanf("%d", &data.death);

printf("Masukkan survival time (hh:mm:ss): ");

scanf("%s", data.survival\_time);

add\_data(data);

printf("Data berhasil ditambahkan.\n");

break;

case 3:

printf("Masukkan kode permainan: ");

scanf("%s", code);

delete\_data(code);

printf("Data berhasil dihapus.\n");

break;

case 4:

printf("Masukkan kode permainan: ");

scanf("%s", code);

search\_data(code);

break;

case 5:

return 0;

default:

printf("Pilihan tidak valid.\n");

break;

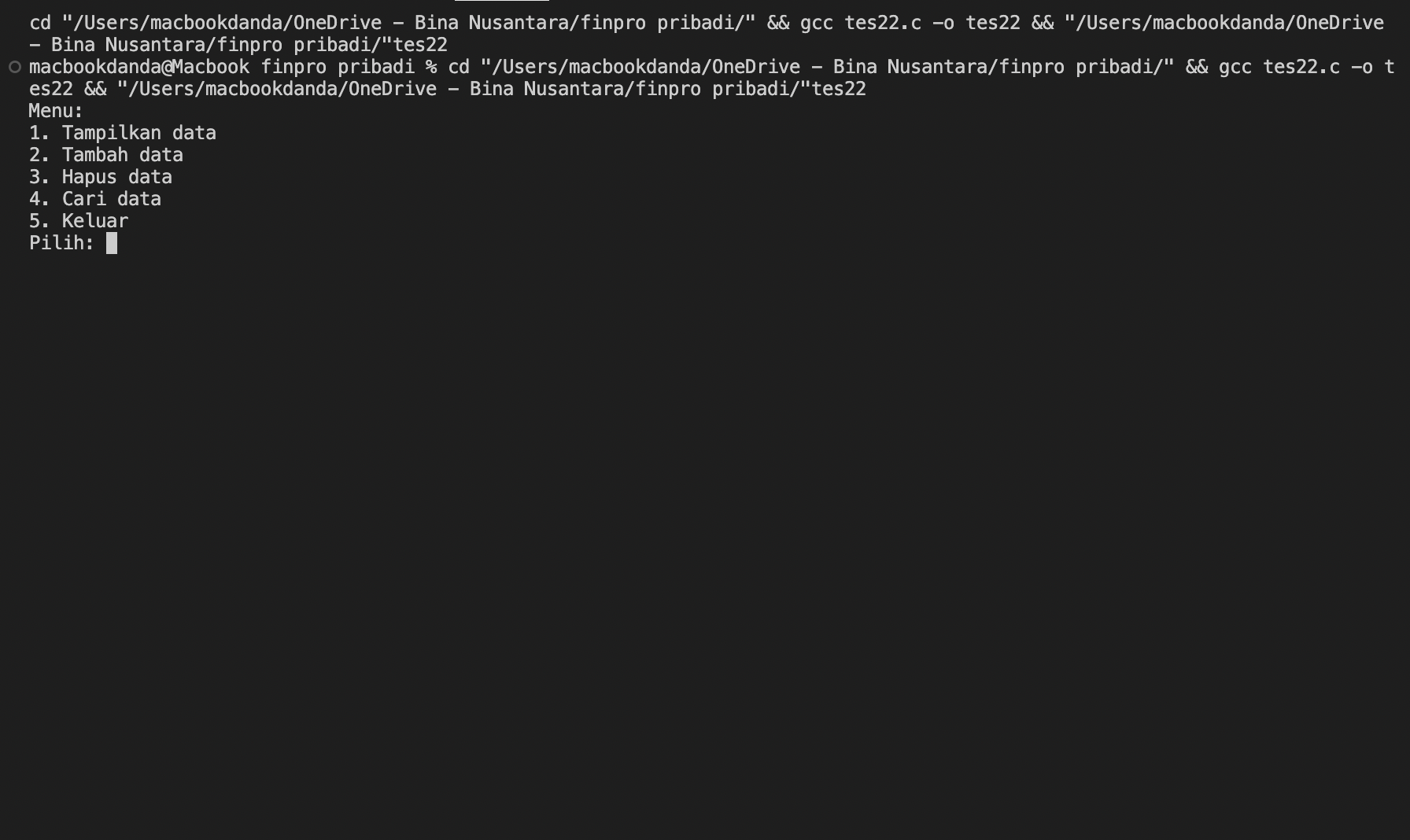
}

}

return 0;

**BAB IV Tampilan Hasil**

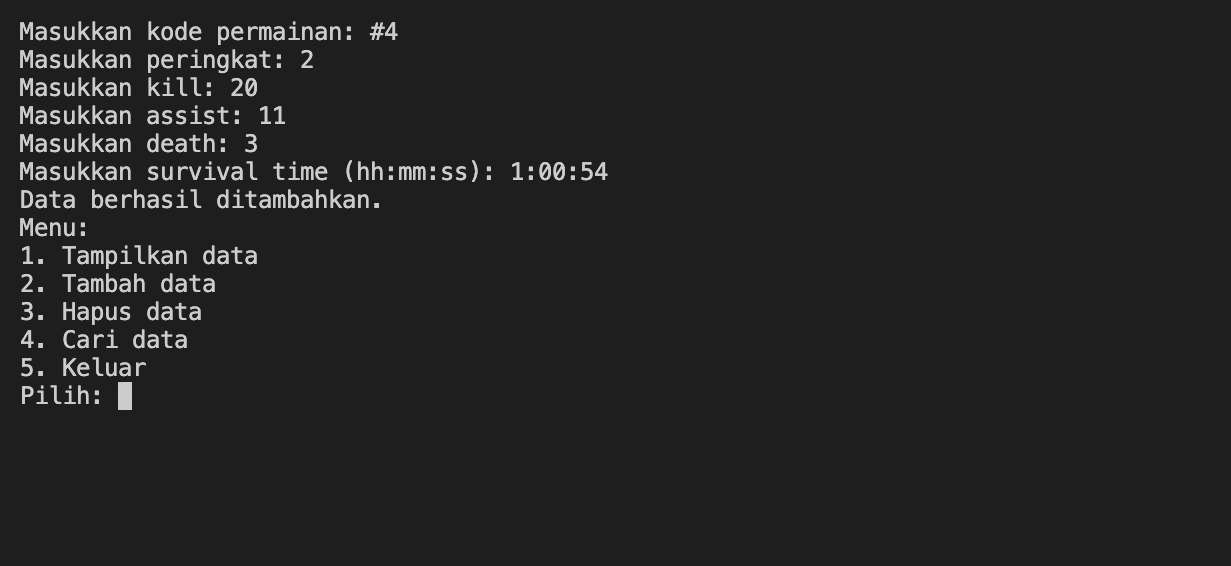
* Menu awal sesaat menjalankan program.



* Fitur pertama adalah menampilkan data yang terdapat pada txt.



* Fitur Kedua adalah menmbahkan data yang belum terdapat pada txt.



Setelah ditambahkan maka data yang tersimpan adalah seperti ini :



* Fitur Ketiga adalah menghapus data dengan memilih kode permainan yang hendak dihapus.

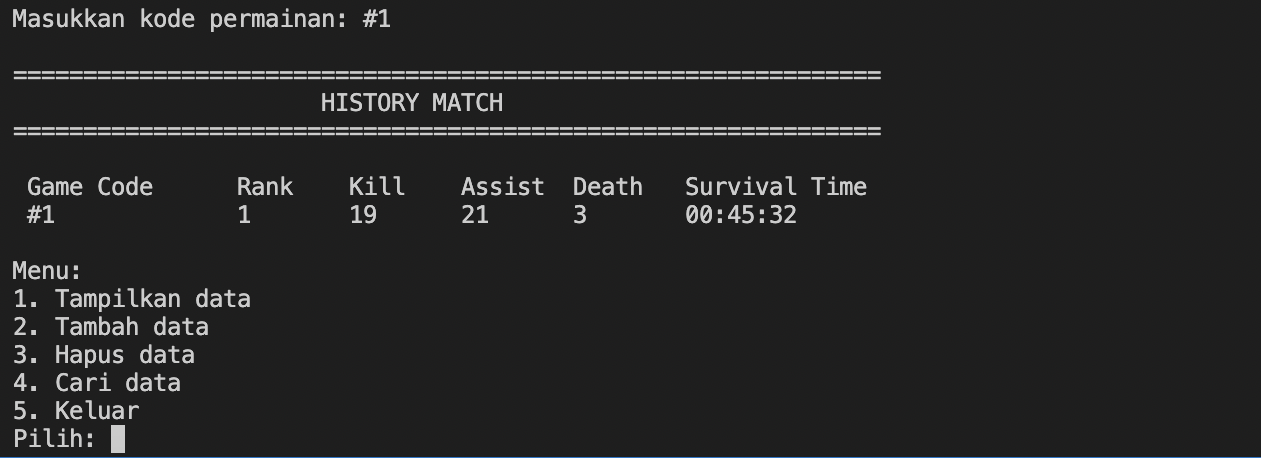
Sebuah gambar berisi teks

Deskripsi dibuat secara otomatis

Maka data setelah kode game #2 dihapus yaitu :



* Fitur keempat yaitu mencari data yang terimpan dalam file txt berdasarkan kode permainan.



Jika anda memilih 5, maka program akan berhenti.

**BAB V Kesimpulan**

Kesimpulan yang dapat ditarik dari program diatas yakni program tersebut terbentuk agar digunakan untuk menyimpan data history match sebuah game. Selain itu, user juga bisa menggunakan fitur-fitur yang tersedia pada program guna mengelola data yang tersedia. Fitur tersebut antara lain : add data (menambahkan data yang belum terdapat pada file txt), delete data (menghapus data yang sudah terdapat pada txt berdasarkan kode permainan yang user input), dan search data (menampilkan data yang sesuai inputan kode permainan dari user).