**IQBAL FOOTBALL ACADEMY**

LAPORAN STUDI KASUS TUGAS AOL

MATA KULIAH COMP6360004 – ALGORITHM AND PROGRAMMING

KELAS LB20



Oleh :

2602134595 – Iqbal Rafi Maulana

Semester Ganjil 2022/2023

MALANG

Daftar Isi

Bab 1 Pendahuluan……………………………………………………………………………………3

Bab 2 Flowchart………………………………………………………………………………………4-6

Bab 3 Program dan Penjelasannya……………………………………………………………………7-18

Bab 4 Tampilan Hasil…………………………………………………………………………………19-22

Bab 5 Kesimpulan…………………………………………………………………………………….23

**BAB 1**

**Pendahuluan**

**Ide Project – Iqbal Football Academy**

Iqbal Football Academy adalah sebuah program yang di design untuk menyimpan data dari pendaftaran akademi sepakbola. Program ini dibuat agar user atau pelatih dapat melihat data pemain yang ada. Program ini bertujuan supaya user / pemain dapat juga mengetahui posisi pemain, tahun lahir, dan nomor pemain lain. Program ini dibuat menggunakan bahasa C. Berikut adalah beberapa unsur yang digunakan saat membuat program Iqbal Football Academy :

**Struct**

Struct digunakan untuk menyimpan data yang telah dibaca dari “pendaftaran.txt”. Di program ini terdapat 1 struct yaitu, struct pendaftaran. Struct ini berfungsi untuk menyimpan data dan file processing dari file “pendaftaran.txt” yang berisi nama pemain, tahun kelahiran, posisi pemain , kaki terkuat, nomor pemain, dan asal provinsi.

**File Processing**

File Processing digunakan untuk menambahkan data pendaftaran dari user lalu menyimpannya di file “pendaftaran.txt”, file “pendaftaran.txt” yang telah di-input oleh user. Dapat digunakan untuk mendisplay data pemain yang ada, mensorting sesuai dengan tahun lahir dan nomor pemain, player juga dapat menjalankan fungsi lainnya seperti searching data player berdasarkan kaki terkuat, user juga dapat mendelete data pemain yang ada di file “pendaftaran.txt”, user juga dapat memberbarui/ update data pemain yang ada di file “pendaftaran.txt”.

**Media Penyimpanan**

Media penyimpanan yang digunakan untuk menyimpan data input dari user adalah file txt bernama “pendaftaran.txt” yang akan digunakan untuk menyimpan data- data pendaftaran pemain.

**Tipe Data**

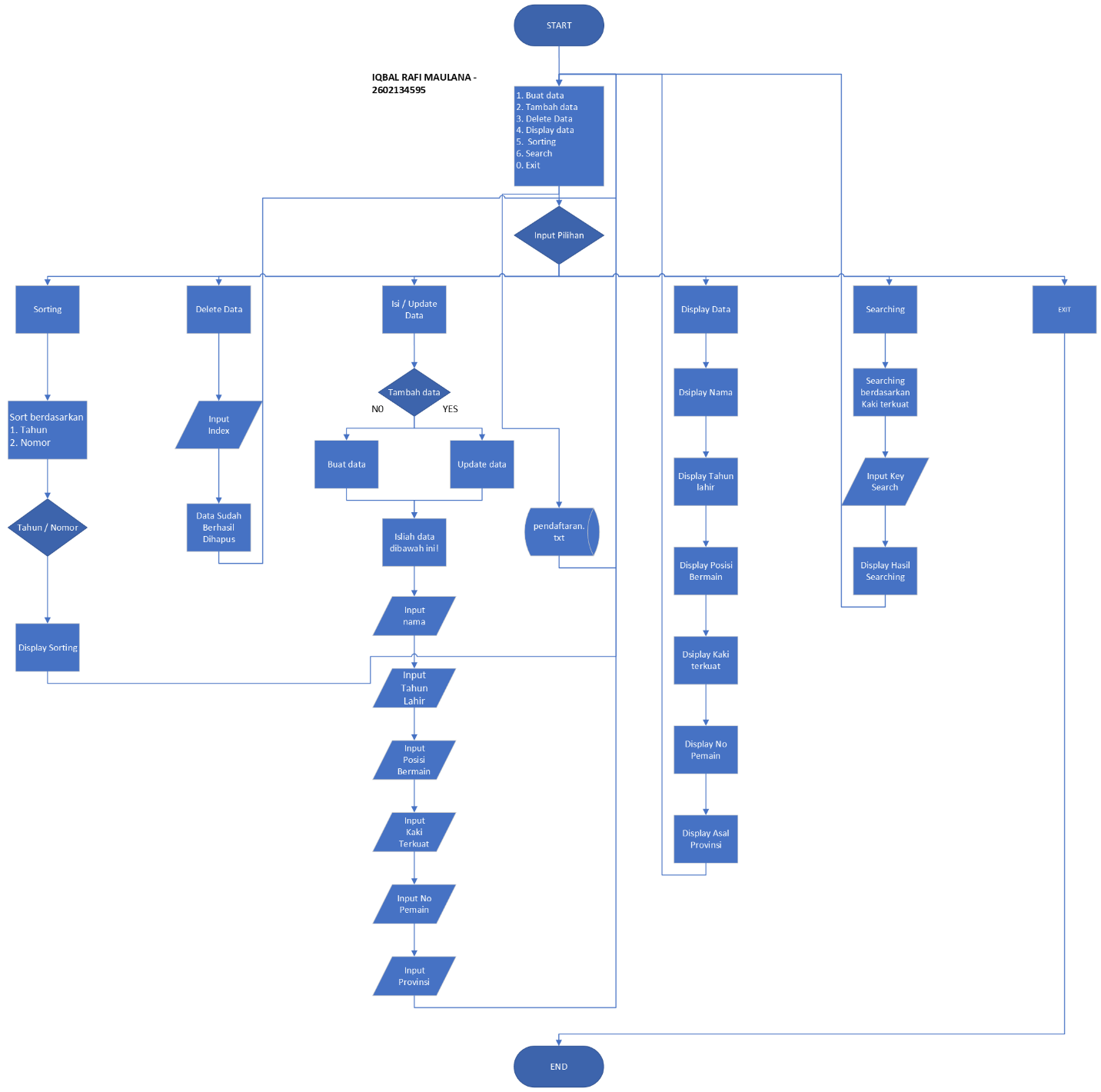
Tipe data yang digunakan di program ini adalah, “pendaftaran.txt” menyimpan semua data pendaftaran pendaftaran, yaitu : nama(string), tahun lahir(integer), posisi pemain(string), kaki terkuat(string), nomor pemain (interger), asal provinsi (string).

**Selection**

Program ini memiliki beberapa menu, selection dapat membantu user untuk menyeleksi/ memilih menu.

**Bab 2**

**Flowchart**



Penjelasan :

User akan diberi 7 menu, yaitu :

1. Isi Data Pendaftaran

Jika user memilih menu ini, user akan diminta untuk meninput nama, tahun lahir, posisi pemain, kaki terkuat, nomor pemain, asal provinsi . Input dari user ini akan disimpan ke dalam file “pendaftaran.txt”. Setelah user selesai menjalankan fungsi Isi data pendaftaran. User bisa menjalankan fungsi lagi dengan mengetik (Y) dan jika tidak ketik (N).

1. Edit Data Pendaftaran

Jika user memilih menu ini, user akan diminta untuk me-input index dari data pemain yang ingin update. Lalu user akan diminta untuk meninput nama, tahun lahir, posisi pemain, kaki terkuat, nomor pemain, asal provinsi dan akan disimpan kedalam file “data.txt” sebagai isi dari index yang dipilih.

1. Delete Data

Jika user memilih menu ini, user dapat menghapus data pemain yang sebelumnya telah di input. Dengan cara memilih index dari data pemain. Lalu program akan menjalankan fungsi hapus() dan menghapus index yang telah dipilih oleh user.

1. Display Data

Jika user memililh menu ini, program akan ‘membaca’ file “pendaftaran.txt” dan menampilkan data yang ada di dalam file “pendaftaran.txt” kepada user dalam bentuk tabel.

1. Sorting Data

Jika user memilih menu ini, program akan ‘membaca’ file “pendaftaran.txt” lalu program akan melakukan fungsi sort(), mengurutkan data pemain dari tahun lahir dan nomor pemain dari terkecil ke terbesar. Lalu program akan menampilkan data yang telah di sorting dalam bentuk tabel.

1. Search

Jika user memilih menu ini, user akan diminta menginput (R\L) karena searching berdasarkan kaki terkuat. Lalu program akan ‘membaca’ file “pendaftaran.txt” lalu program akan melakukan fungsi search() sesuai dengan input dari user, setelah itu program akan menampilkan hasilnya kepada user dalam bentuk tabel.

1. Exit

Jika user sudah merasa tidak ingin menjalankan program , user dapat memilih menu exit program dan program akan selesai.

**BAB** 3

Program dan Penjelasannya

// Iqbal Rafi Maulana

// 2602134595

// LB20 - Project AOL

#include <stdio.h> // fungsi library input output

#include <stdlib.h> // fungsi library alokasi memory

#include <string.h> // fungsi library manipulasi string

#include <ctype.h> // fungsi library untuk character

struct pendaftaran{ // struct pendaftaran

char nama[100]; // declare type data string

int tahunlahir; // declare type data int

char posisi[100]; // declare type data string

char peakfoot[10]; // declare type data string

int number; // declare type data int

char province[100]; // declare type data int

}data[100]; // declare agar struct digunakan secara global

int idx; // declare variable global type data int untuk indikasi index value dari variable data untuk setiap inputan dalam struct pendaftaran

void entry(){ // membuat modul entry

char cekInput; // declare type data char untuk cek input

FILE \*fp; // declare pointer fp type data file

fp = fopen ("pendaftaran.txt", "a"); // open file bernama pendaftaran dan menggunakan fungsi 'a' untuk menambah data baru

printf("===================================================== Pengisian Data =================================================== \n");// print sbg tampilan awal

do{ // lakukan perulangan

printf("Isilah data dibawah ini!\n"); // print untuk mengingatkan user utk input data baru

printf("Nama Pendaftar : "); // print untuk input nama

scanf("%s", data[idx].nama); // scan data bertipe string

getchar(); // mengambil blank space

printf("Tahun Kelahiran : "); // print untuk input tahun

scanf("%d", &data[idx].tahunlahir); // scan data bertipe integer

getchar(); // mengambil blank space

printf("Posisi Bermain : "); // print untuk posisi bermain

scanf("%[^\n]", data[idx].posisi); // scan data bertipe string

getchar(); // mengambil blank space

printf("Kaki Terkuat (R/L) : "); // print untuk kaki terkuat

scanf("%[^\n]", data[idx].peakfoot); // scan data bertipe string

getchar(); // mengambil blank space

printf("Nomor Pemain : "); // print untuk input nomor pemain

scanf("%d", &data[idx].number); // scan data bertipe integer

getchar();// mengambil blank space

printf("Asal Provinsi: "); // print untuk input asal provinsi

scanf("%[^\n]", data[idx].province); // scan data bertipe string

getchar();// mengambil blank space

fprintf(fp, "%s\t%d\t%s\t%s\t%d\t%s\n", data[idx].nama,data[idx].tahunlahir,data[idx].posisi,data[idx].peakfoot,data[idx].number,data[idx].province);// print inputan user ke dalam file pendaftaran.txt

idx ++; // post incremnt idx

printf("\nApakah anda ingin menginput data lagi?(Y/N) : "); // print untuk menanyakan apakah user ingin input data lagi atau tidak

scanf("\n%c", &cekInput); // scan apakah user ingin input data lagi atau tidak

}while(toupper(cekInput) != 'N');// jika input user 'N'/'n' maka berhenti mengisi entry jika selain 'n' langsung menampilkan seperti di awal

fclose(fp);// menutup file pendaftaran.txt

printf("Press ENTER to go back to main menu...\n");// print agar bisa kembali ke main menu harus press enter

getchar();// agar bisa memproses enter agar bisa kembali ke main menu

}

void update(){ // membuat modul update

int index; // declare index data type int

char cekInput; // declare cekInput data type char

printf("======================================================== Update Data =================================================== \n"); // print sbg tampilan awal

do{ // lakukan perulangan

printf("Isilah data dibawah ini!\n"); // print untuk mengingatkan user utk input data baru

printf("Choose index : "); // print untuk input index keberapa yang mau di update

scanf("%d", &index); // scan index keberapa yang mau di update

printf("Nama Pendaftar : "); // print untuk input nama

scanf("%s", data[index-1].nama); // scan data bertipe string

getchar(); // mengambil blank space

printf("Tahun Kelahiran : ");// print untuk input tahun

scanf("%d", &data[index-1].tahunlahir); // scan data bertipe int

getchar(); // mengambil blank space

printf("Posisi Bermain : "); // print untuk posisi bermain

scanf("%[^\n]", data[index-1].posisi); // scan data bertipe string

getchar(); // mengambil blank space

printf("Kaki Terkuat (R/L) : "); // print untuk kaki terkuat

scanf("%[^\n]", data[index-1].peakfoot); // scan data bertipe string

getchar(); // mengambil blank space

printf("Nomor Pemain : "); // print untuk input nomor pemain

scanf("%d", &data[index-1].number); // scan data bertipe int

getchar(); // mengambil blank

printf("Asal Provinsi: "); // print untuk input asal provinsi

scanf("%[^\n]", data[index-1].province); // scan data bertipe string

getchar(); // mengambil blank space

FILE \*fp; // declare pointer fp type data file

fp = fopen ("pendaftaran.txt", "w"); // open file bernama pendaftaran dan menggunakan fungsi 'w' untuk menambah data baru

for(int i=0; i < idx; i++){ // lakukan perluangan jika idx lebih besar dari i

fprintf(fp, "%s\t%d\t%s\t%s\t%d\t%s\n", data[i].nama,data[i].tahunlahir,data[i].posisi,data[i].peakfoot,data[i].number,data[i].province); // print inputan user ke dalam file pendaftaran.txt

} fclose(fp); // menutup file pendaftaran.txt

printf("\nApakah anda ingin menginput data lagi?(Y/N) : ");// print untuk menanyakan apakah user ingin input data lagi atau tidak

scanf("\n%c", &cekInput); // scan apakah user ingin input data lagi atau tidak

}while(toupper(cekInput) != 'N'); // jika input user 'N'/'n' maka berhenti mengisi entry jika selain 'n' langsung menampilkan seperti di awal

printf("Press ENTER to go back to main menu...\n"); // print agar bisa kembali ke main menu harus press enter

getchar(); // agar bisa memproses enter agar bisa kembali ke main menu

}

void hapus(){ // membuat modul hapus data

int index; // declare index type data int

printf("====================================================== Delete Data =====================================================\n"); // print sbg tampilan awal

printf("Select index: "); // print untuk input index keberapa yang mau di hapus

scanf("%d", &index); // scan index keberapa yang mau di hapus

for(int i=index-1; i<idx; i++){ // lakukan perulangan jika idx > i

data[i]=data[i+1]; // fungsi swap datd

}

idx--; // post increment idx

FILE \*fp; // declare pointer fp type data file

fp=fopen("pendaftaran.txt", "w"); // open file bernama pendaftaran dan menggunakan fungsi 'w' untuk melakukan perubahan

for(int i=0; i < idx; i++){ // lakukan perulangan ketika idx > i

fprintf(fp, "%s\t%d\t%s\t%s\t%d\t%s\n", data[i].nama,data[i].tahunlahir,data[i].posisi,data[i].peakfoot,data[i].number,data[i].province); // print inputan user ke dalam file pendaftaran.txt

}

fclose(fp); // menutup file pendaftaran.txt

}

void display() // membuat modul display

{

printf("======================================================== Display =======================================================\n"); // print sbg tampilan awal

printf("-------------------------------------------------------------------------------------------------"); // print untuk memberi batas

printf("\nNama Pendaftar\t "); // print untuk dijadikan tabel

printf("Tahun Kelahiran "); // print untuk dijadikan tabel

printf("Posisi Bermain "); // print untuk dijadikan tabel

printf("Kaki Terkuat (R/L) "); // print untuk dijadikan tabel

printf("Nomor Pemain "); // print untuk dijadikan tabel

printf("Asal Provinsi\t "); // print untuk dijadikan tabel

printf("\n"); // membuat line baru

printf("-------------------------------------------------------------------------------------------------\n");// print sbg tampilan awal

for(int j=0; j<idx; j++) // lakukan perulangan print jika idx > j

{

if (strlen(data[j].province) >= 9) // jika panjang string data province lebih dari sama dengan 9 maka lakukan fungsi dibawahnya

{

printf(" %s\t\t\t%d\t\t %s\t\t%s\t\t %d\t\t%s\n", data[j].nama,data[j].tahunlahir,data[j].posisi,data[j].peakfoot,data[j].number,data[j].province); // langsung print data dari nama, tahun lahir, posisi, kaki terkuat, nomor, dan provinsi

}

else // jika panjang string data province kurang dari 9 maka lakukan fungsi dibawahnya

{

printf(" %s\t\t\t%d\t\t %s\t\t%s\t\t %d\t\t%s\n", data[j].nama,data[j].tahunlahir,data[j].posisi,data[j].peakfoot,data[j].number,data[j].province); // langsung print data dari nama, tahun lahir, posisi, kaki terkuat, nomor, dan provinsi

}

}

printf("-------------------------------------------------------------------------------------------------\n");// print untuk memberi batas dan memberi tahu jika program sudah selesai dijalankan

exit: // untuk keluar dari program

getchar(); // menerima blank space

printf("\nPress ENTER to go back to main menu...\n"); // print agar bisa kembali ke main menu harus press enter

getchar(); // agar bisa memproses enter agar bisa kembali ke main menu

}

void sort(){ // membuat modul sort

struct pendaftaran sorted[100], temp; // declare struct pendaftaran ber variable sorted dan temp

FILE \*fp; // declare pointer fp type data file

fp = fopen ("pendaftaran.txt", "r"); // open file bernama pendaftaran dan menggunakan fungsi 'r' untuk read data dalam file

int i = 0; // declare int i bervalue 0

while (fscanf(fp, "%s\t%d\t%s\t%s\t%d\t%s\n", &sorted[i].nama, &sorted[i].tahunlahir, &sorted[i].posisi, &sorted[i].peakfoot, &sorted[i].number, &sorted[i].province) != EOF) { // ketika scan file untuk mendapatkan nama, tahun lahir, posisi, kaki terkuat, nomor, dan provinsi

i++; //menambahkan indeks

}

int memilih; // declare type data int agar bisa digunakan dalam switch case

printf("======================================================== Sorting =======================================================\n"); // print sbg tampilan awal

printf("\nSorting Berdasarkan : \n"); // print untuk menjelaskan bahwa ada 2 jenis yang di sorting

printf("1. Tahun Kelahiran\n"); // pilihan pertama sorting berdasarkan tahun

printf("2. Nomor Pemain\n"); // pilihan kedua sorting berdasarkan nomor pemain

printf("\nPilih Menu : ");// print pilih menu

scanf("%d", &memilih);// scan int untuk switch case

switch (memilih){ // membuka program sesuai inputan int yang dipilih user

case 1: { // masuk ke case 1 jika user memilih sorting berdasarkan tahun

for(int j = 0; j < i; j++){ // lakukan perulangan jika i > j

for(int k = 0; k < i-1; k++){ // lakukan perulangan jika i -1 > k

if(sorted[k].tahunlahir> sorted[k+1].tahunlahir){ // jika sorted tahun lahir lebih besar dari sorted tahun lahir+1

temp = sorted[k]; // maka lakukan swap

sorted[k] = sorted[k + 1]; // antara sorted dan sorted +1

sorted[k+1] = temp; // hasil setelah swap

}

}

} printf("\nSort Data Berdasarkan Tahun Kelahiran\n"); // memberi pemberitahuan jika hasil sort berdasarkan tahun lahir

break; // fungsi agar tidak lanjut ke program selanjutnya

}case 2:{ // masuk ke case 2 jika user memilih sorting berdasarkan nomor

for(int j = 0; j < i; j++){ // lakukan perulangan jika i > j

for(int k = 0; k < i-1; k++){ // lakukan perulangan jika i -1 > k

if(sorted[k].number> sorted[k+1].number){ // jika sorted nomor lebih besar dari sorted nomor+1

temp = sorted[k]; // maka lakukan swap

sorted[k] = sorted[k + 1]; // antara sorted dan sorted +1

sorted[k+1] = temp; // hasil setelah swap

}

}

} printf("\nSort Data Berdasarkan Nomor Pemain\n"); // memberi pemberitahuan jika hasil sort berdasarkan nomor

break; // fungsi agar tidak lanjut ke program selanjutnya

}

}

printf("-------------------------------------------------------------------------------------------------"); // print sbg tampilan awal

printf("\nNama Pendaftar\t "); // print untuk dijadikan tabel

printf("Tahun Kelahiran "); // print untuk dijadikan tabel

printf("Posisi Bermain "); // print untuk dijadikan tabel

printf("Kaki Terkuat (R/L) "); // print untuk dijadikan tabel

printf("Nomor Pemain "); // print untuk dijadikan tabel

printf("Asal Provinsi\t "); // print untuk dijadikan tabel

printf("\n");// print untuk new line

printf("-------------------------------------------------------------------------------------------------\n"); // print sbg tampilan awal

for (int i = 0; i < idx; i++) // lakukan perulangan jika idx > i

{

printf(" %s\t\t\t%d\t\t %s\t\t%s\t\t %d\t\t%s\n", sorted[i].nama,sorted[i].tahunlahir,sorted[i].posisi,sorted[i].peakfoot,sorted[i].number,sorted[i].province); // langsung print data sorting dari nama, tahun lahir, posisi, kaki terkuat, nomor, dan provinsi

}

printf("-------------------------------------------------------------------------------------------------\n"); //print untuk memberi batas dan memberi tahu jika program sudah selesai dijalankan

fclose(fp); // menutup file pendaftaran.txt

getchar(); // menerima blank space

printf("\nPress ENTER to go back to main menu...\n"); // print agar bisa kembali ke main menu harus press enter

getchar(); // agar bisa memproses enter agar bisa kembali ke main menu

}

void search(){ // modul seacrh

char foot[100]; // declare type data char sebagai data yang digunakan sbg key search

printf("======================================================== Search =======================================================\n"); // print sbg tampilan awal

printf("Search Berdasarkan Kaki Terkuat\n"); // print untuk menjelaskan bahwa seacrh berdasarkan kaki terkuat

printf("\nInput foot (R/L) : "); // print untuk memberi tau format input yang bisa dimasukkan

scanf("%s", &foot); // scan inputan untuk keysearch

printf("-------------------------------------------------------------------------------------------------"); // print sbg tampilan awal

printf("\nNama Pendaftar\t "); // print untuk dijadikan tabel

printf("Tahun Kelahiran "); // print untuk dijadikan tabel

printf("Posisi Bermain "); // print untuk dijadikan tabel

printf("Kaki Terkuat (R/L) "); // print untuk dijadikan tabel

printf("Nomor Pemain "); // print untuk dijadikan tabel

printf("Asal Provinsi\t "); // print untuk dijadikan tabel

printf("\n"); // print untuk new line

printf("-------------------------------------------------------------------------------------------------\n"); // print sbg tampilan awal

for(int i = 0; i-1 < idx; i++) // lakukan perulangan jika idx > i-1

{

if (strcmp(foot, data[i].peakfoot ) == 0) // metode seacrh string menggunakan string compare

{

printf(" %s\t\t\t%d\t\t %s\t\t%s\t\t %d\t\t%s\n", data[i].nama,data[i].tahunlahir,data[i].posisi,data[i].peakfoot,data[i].number,data[i].province);// jika search ditemukan maka print nama, tahun lahir, posisi, kaki terkuat, nomor, dan provinsi

}

}

printf("-------------------------------------------------------------------------------------------------\n"); // sprint untuk memberi batas dan memberi tahu jika program sudah selesai dijalankan

getchar(); // menerima blank space

printf("Press ENTER to go back to main menu...\n"); // print agar bisa kembali ke main menu harus press enter

getchar(); // agar bisa memproses enter agar bisa kembali ke main menu

}

int main () // fungsi main

{

FILE \*fp; // declare pointer fp type data file

fp = fopen("pendaftaran.txt", "r"); // open file bernama pendaftaran dan menggunakan fungsi 'r' untuk read data dalam file

idx = 0; // declare bahwa idx bervalue 0

while (fscanf(fp, "%s\t%d\t%s\t%s\t%d\t%s\n", &data[idx].nama, &data[idx].tahunlahir, &data[idx].posisi, &data[idx].peakfoot, &data[idx].number, &data[idx].province) != EOF) { // ketika scan file untuk mendapatkan nama, tahun lahir, posisi, kaki terkuat, nomor, dan provinsi

idx ++; //menambahkan indeks

}

fclose(fp); // menutup file pendaftaran.txt

int pilihan; // declare type data int agar bisa digunakan dalam switch case

do{ // lakukan perulangan

system("cls"); // fungsi membersihkan layar

printf("====================================================== Iqbal Academy ===================================================\n"); // sbg tampilan awal

printf("\n1. Isi Data Pendaftaran\n"); // tampilan menu

printf("2. Edit Data Pendaftaran\n"); // tampilan menu

printf("3. Delete Data\n"); // tampilan menu

printf("4. Display Data\n"); // tampilan menu

printf("5. Sorting Data\n"); // tampilan menu

printf("6. Search\n"); // tampilan menu

printf("0. Exit\n"); // tampilan menu

printf("\n"); // membuat new line

printf("Pilih Menu : "); // print pilih menu

scanf("%d", &pilihan); // scan int untuk swicth case

switch(pilihan){ // membuka program sesuai inputan int yang dipilih user

case 1:{ // masuk ke case 1 jika user memilih untuk menambah / mengisi data

system("cls"); // fungsi membersihkan layar

entry(); // memanggil modul entry

break; // fungsi agar tidak lanjut ke program selanjutnya

}

case 2:{ // masuk ke case 2 jika user memilih untuk meng update data

system("cls"); // fungsi membersihkan layar

update(); // memanggil modul update

break; // fungsi agar tidak lanjut ke program selanjutnya

}

case 3:{ // masuk ke case 3 jika user memilih untuk mehapus data

system("cls"); // fungsi membersihkan layar

hapus(); // memanggil modul hapus

break; // fungsi agar tidak lanjut ke program selanjutnya

}

case 4 :{ // masuk ke case 4 jika user memilih untuk mendisplay data

system("cls"); // fungsi membersihkan layar

display(); // memanggil modul display

break; // fungsi agar tidak lanjut ke program selanjutnya

}

case 5: { // masuk ke case 5 jika user memilih untuk sorting data

system("cls"); // fungsi membersihkan layar

sort(); // memanggil modul sorting

break; // fungsi agar tidak lanjut ke program selanjutnya

}

case 6: { // masuk ke case 6 jika user memilih untuk seacrh data

system("cls"); // fungsi membersihkan layar

search(); // memanggil modul search

break; // fungsi agar tidak lanjut ke program selanjutnya

}

}

}

while(pilihan != 0); // lakukan perulangan selama inputan bukan 0, jika inputan user 0 maka langsung hentikan program

printf("You choose to EXIT the program, See You!"); // farewell messages

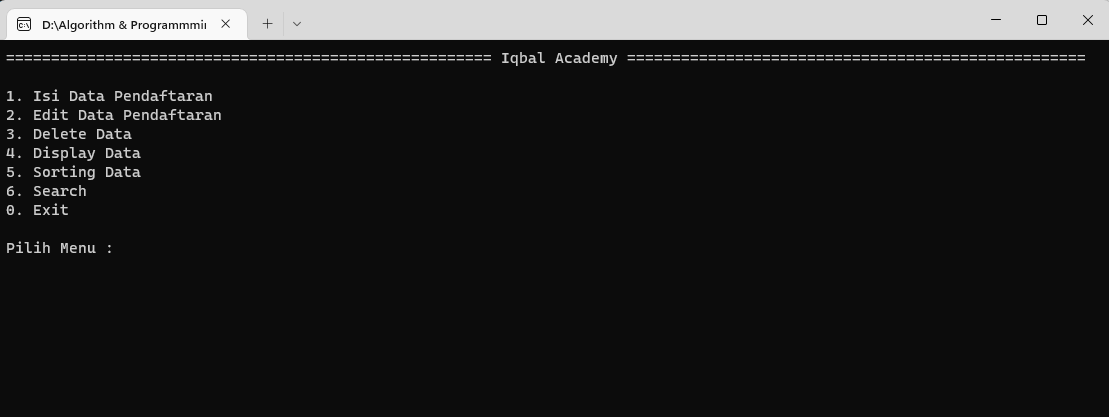
return 0; // fungsi untuk menghentikan program

}

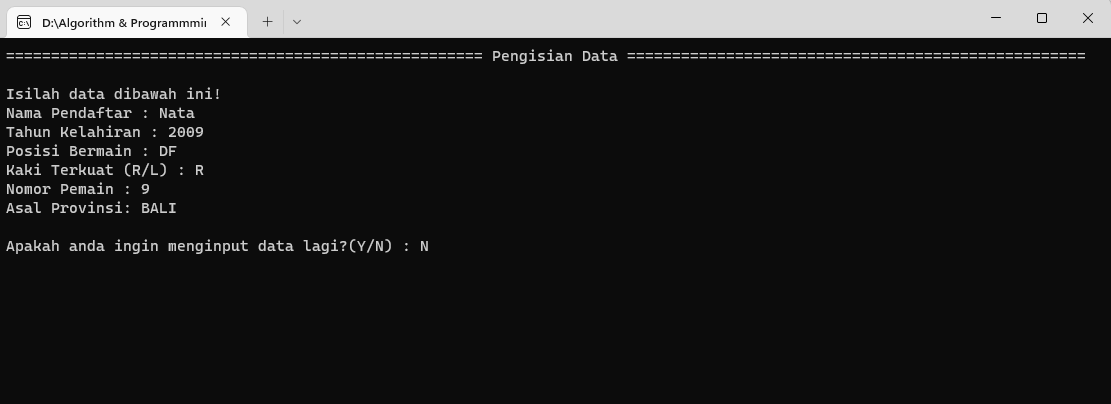
BAB 4

Tampilan Hasil

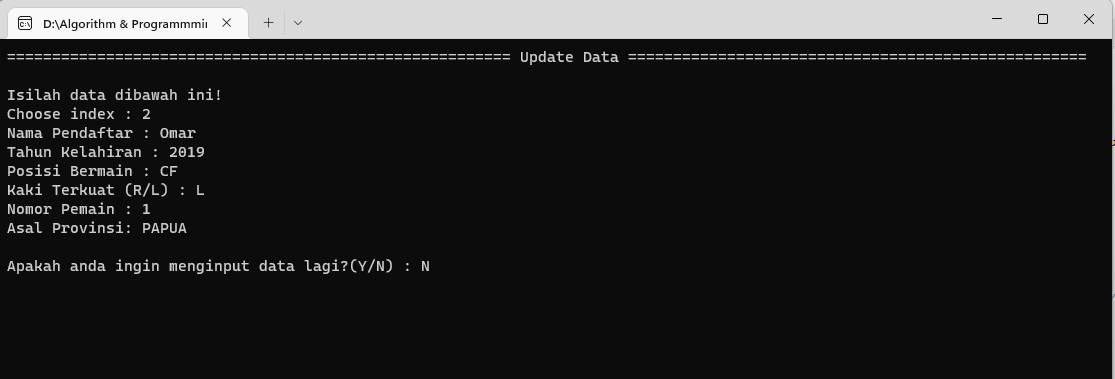
**Menu Utama / Main menu**



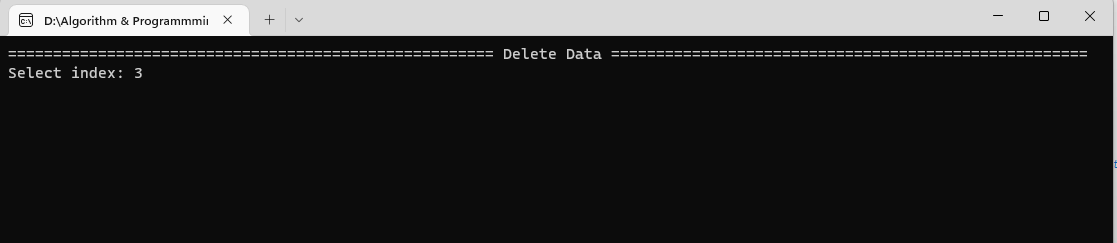
**Menu 1 🡪 Isi Data Pemdaftaran**



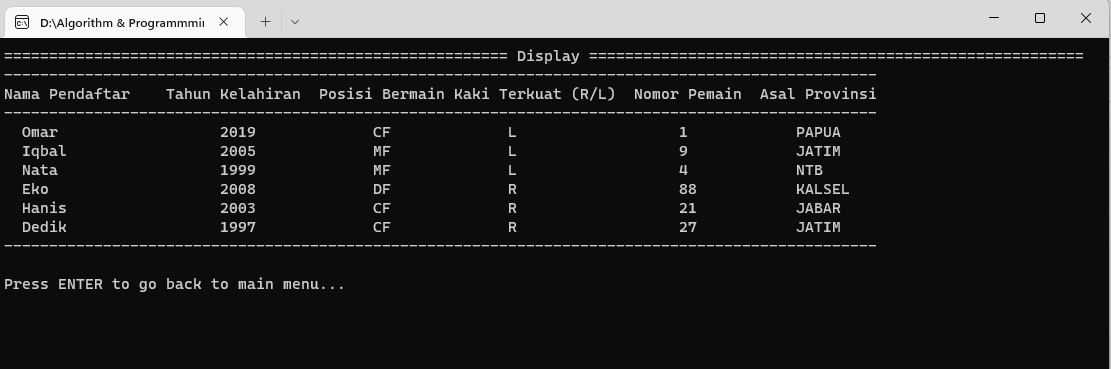
**Menu 2 🡪 Edit Data Pendaftaran**



**Menu 3 🡪 Delete Data**

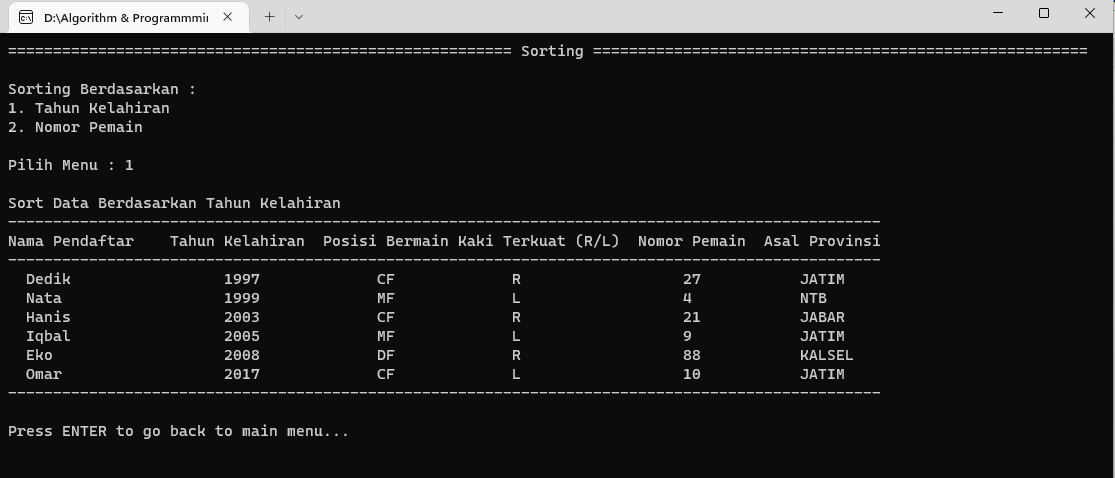


**Menu 4 🡪 Display Data**

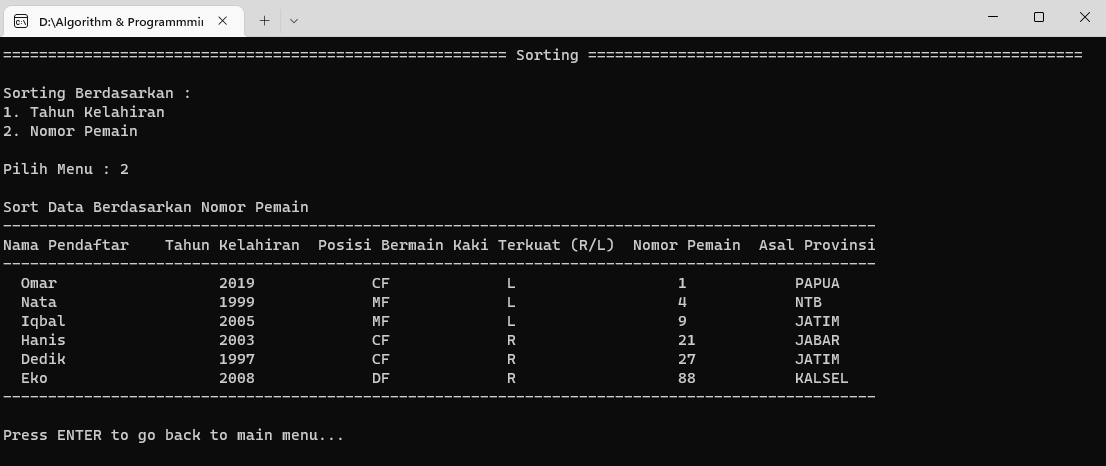


**Menu 5 🡪 Sorting Data**

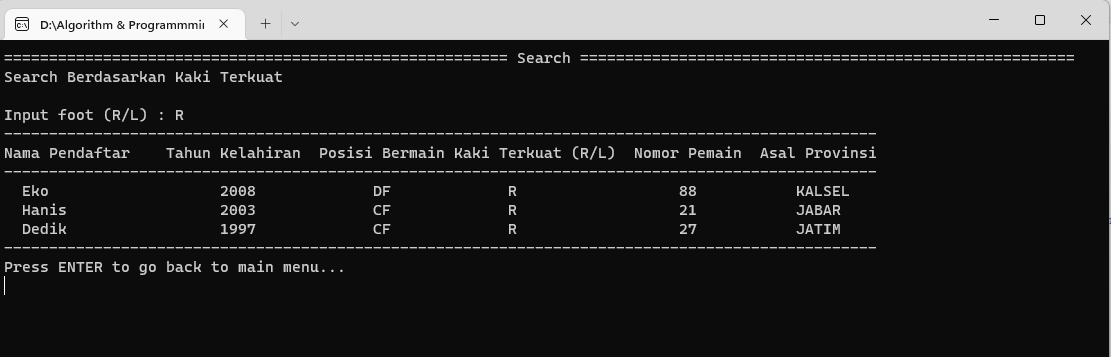
Sorting Berdasarkan Tahun Kelahiran



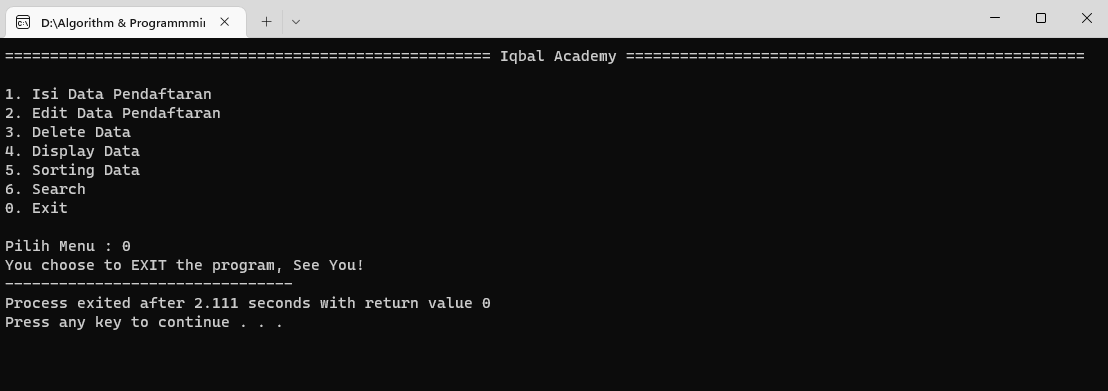
Sorting Berdasarkan Nomor Pemain



**Menu 6 🡪 Search Data**



**Menu 0 🡪** Exit Program



**BAB 5**

**Kesimpulan**

Iqbal Football Academy adalah program yang didesain untuk menampung, menampilkan, menupdate, mensorting lalu menampilkan, mencari, menghapus data yang telah diinput oleh user. Iqbal Football Academy dapat digunakan oleh pelatih sepakbola yang ingin melihat data para pemain yang terdaftar. Fitur programming di bahasa C seperti file processing, struct, serta selection digunakan untuk membantu pembuatan fitur- fitur di program ini.