**Millionaire**

LAPORAN STUDI KASUS TUGAS AOL  
MATA KULIAH COMP6360004 — ALGORITHM AND PROGRAMMING  
KELAS LB20

Company name

Description automatically generated with low confidence

Oleh :

2602160776 – Felix Juwono Purwoko

Semester Ganjil 2022

MALANG

**Daftar Isi**

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

Bab 2 Usulan Desain Program……………………………………………………………………….…6

Bab 3 Program dan Penjelasannya………………………………………………………………….…42

Bab 4 Tampilan Hasil……………………………………………………………...……………….....75

Bab 5 Kesimpulan……………………………………………………………...……..…………….....83

**Bab 1 Pendahuluan**

Pada hari Minggu 22 Januari 2023 akan digelar sebuah acara bergengsi yang dapat berhadiah hingga $100.000.000. Para peserta akan dikenakan biaya $1.000.000 hanya untuk satu kali kesempatan. Namun pihak penyelanggara sedang bingung untuk memikirkan bagaimana konsep dari acara ini agar dapat berlangsung dengan praktis dan dapat memeriahkan suasana. Oleh karena itu, pihak penyelanggara sepakat untuk menggunakan suatu sistem yang dapat memberikan 15 pertanyaan pengetahuan umum kepada para peserta .

Apabila terdapat peserta yang dapat menjawab semua pertanyaan tersebut dengan benar, maka pihak penyelenggara akan mengirimkan uang sejumlah $100.000.000 ,sedangkan untuk *player* yang salah dalam menjawab pertanyaan akan diberikan hadiah sejumlah uang berdasarkan *safe zone* para peserta, apabila terdapat peserta yang tidak memenuhi safe zone dan menjawab pertanyaan dengan salah, maka akan mendapat *score* 0 dan belum bisa untuk mendapatkan hadiah ,kecuali player menyerah sebelum menjawab dengan salah akan memperoleh hadiah sebesar ½ dari *prize*  yang di diperoleh oleh player .Player juga akan diberikan satu kali kesempatan untuk melakukan 50:50 yang artinya player dapat mengubah dari 4 pilihan jawaban menjadi hanya 2 pilihan jawaban. Semua *prize* tersebut akan dikirimkan melalui transfer rekening Bank BBB dan untuk bukti transfernya akan segera dikirimkan oleh pihak penyelenggara melalui chatting Whats App kepada para peraih hadiah(*awardee*).Pihak penyelanggara juga membutuhkan sistem yang terdapat menu Admin agar dapat mengedit baik kesalahan data yang dilakukan saat registrasi maupun untuk dapat menghapus akun user yang sudah tidak dipakai lagi.

Untuk membantu pihak penyelanggara dalam mengadakan acara yang meriah akan dibuatkan suatu sistem yang memiliki fitur sebagai berikut:

1.Registrasi untuk para *player*

2.Login *multiuser* untuk para *player*

3*.*Memberikan pertanyaan secara acak dan terdapat score di dalamnya

4.Melihat skor pemain beserta rangkingnya.

5. Melihat data (peraih hadiah) *awardee*.

6.Login *singleuser* yang hanya dapat diakses oleh satu admin

7.Mengedit kesalahan data para player yang hanya dapat diakses oleh admin.

8. Menghapus record akun para player yang sudah tidak digunakan dan hanya dapat diakses oleh admin

Kendali perulangan dibutuhkan untuk kebutuhan sebagai berikut:

-Membaca *file.*

-Menulis *file.*

-Validasi input.

-Menghasilkan angka random.

-Menghasilkan id untuk registrasi para *player.*

Kendali seleksi dibutuhkan untuk kebutuhan sebagai berikut:

-Pemilihan menu.

-Mengecekkan keberedaan data para peraih hadiah pada file dataregistrasi.txt berdasarkan id-user.

-Menampilkan validasi input.

-Menghasilkan id Ketika belum terdapat *record.*

-Mengecek jawaban para *player.*

-Memberikan fitur 50:50 dan menu *Surrender* pada *user.*

Media penyimpanan dibutuhkan untuk kebutuhan sebagai berikut:

-Terdapat file dataregistrasi.txt yang berfungsi untuk menyimpan data informasi penting yaitu *username* ,*password,*nomor *Whats app,*nomor rekening beserta id dari para player.

-Terdapat file scores.txt yang berfungsi untuk menyimpan nickname , tanggal dan waktu berakhirnya game(*Game over*). Beserta id dari para pemain.

Tipe-tipe data yang dibutuhkan sebagai berikut:

Terdapat 3 struct yaitu:

Struct data yang memiliki nama variable login yang dapat menampung 100 *record*.Struct data ini akan berfungsi untuk menampung informasi*-*informasi penting dari para user dan dapat dibaca dari file dataregistrasi.txt dengan variabel *character* *id[], character username[],character pass[],character noWA[],dan character noRekening[]..*

Struct q yang memiliki nama variable arr1 yang dapat menampung 15 *record .*Struct q ini akan berfungsi untuk menampung pertanyaan dari pihak penyelenggara yang dapat dibaca dari file questions.txt dengan variable *character* question[],*character* choice[] ,*integer* answer[].

Struct s yang memiliki nama variable arr2, awardee yang dapat menampung 999 *record.* Struct s ini akan berfungsi untuk menyimpan data score dari para player yang dapat dibaca dari file scores.txt dengan variable *int*eger id,char *character* name[],*long long* money, *character* Time[],*character* Date[]

*Integer untuk variable* account id, Q, S, pv,a,I,mid,move, win, lost, surrender, hint,ctrQ,j,found,selection,menu, ctrH1,ctrH2,movehint, choose, Available,id,validasi, actr, idedit, selection2

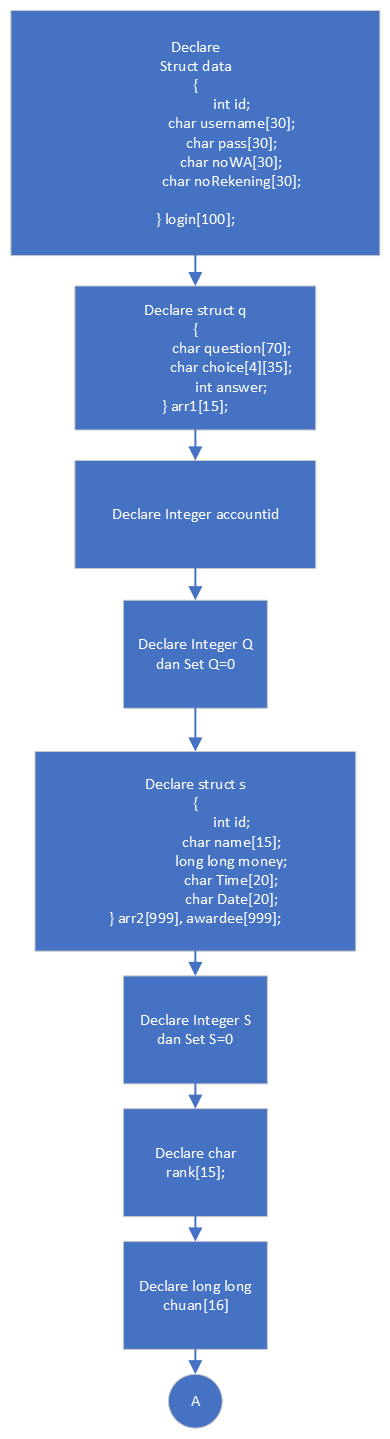
*Character* untuk variable rank, TimeGameOver, DateGameOver, nickname, choice1,choice2,username,pass, noWA,noRekening,usernamesearch,passwordsearch,ch, usernameAdmin, password

*Long long integer* untuk variable chuan,score,

Dalam program ini juga menggunakan memory allocation dengan tipe static pada variable Available

**Bab 2 Usulan Desain Program**

Global variable:

****Diagram

Description automatically generated

-Main function

Diagram

Description automatically generated

Module utama berikut ini pertama-tama akan membuka file scores.txt dan questions.txt, apabila file data questions.txt tidak ada maka program akan keluar dengan menampilkan Fail “to load \"questions.txt\" file.”,sedangkan apabila tidak terdapat scores.txt program masih bisa jalan namun tidak dapat menampilkan hasil *record* score player.Hasil record dari file scores.txt ini juga akan disorting berdasrakan score tertinggi menggunakan module quickSortArr2(0,S-1)

Diagram

Description automatically generated

Kemudian *user* akan diberikan pilihan berdasarkan login sebagai admin atau player dan untuk input 3 program akan menutup kedua file tersebut dan berakhir.

-Module swap(struct s\* a,struct s\*b)

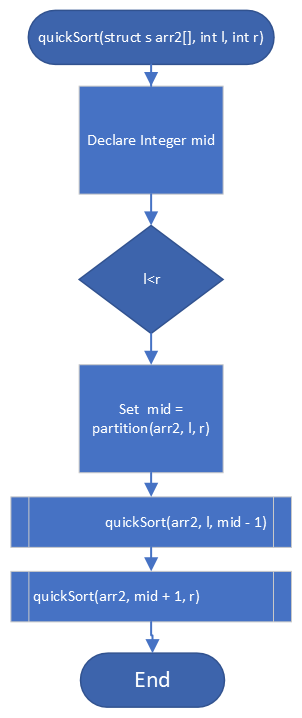
-Module partition(struct s arr2[], int l, int r)

-Module quicksort(struct s arr2[],int l,int r)

Diagram

Description automatically generated Diagram

Description automatically generated



Ketiga module berikut ini berfungsi untuk mengurutkan *record* data para player berdasarkan urutan skor tertinggi.

-Module playermenu()

Diagram

Description automatically generated

Module playermenu() berfungsi untuk memberikan pilihan menu 1-3 pada user .Untuk input karakter selain angka ,maka program akan terus mengulang untuk meminta input berupa angka 1-3 terlebih dahulu ,baru dapat mengeksekusi *statement* selanjutnya.untuk input 1 akan mengekses menu registrasi,sedangkan untuk input 2 ditujukan pada *player* yang sudah memiliki akun .Untuk input 3 akan kembali ke menu sebelumnya,sedangkan untuk input di luar range 1-3 akan terus mengulangi menu yang sama.

-Module registrasi()

Chart, funnel chart

Description automatically generated

Module ini berfungsi untuk player dalam memasukkan data informasi penting untuk diinputkan ke dalam sistem .Namun terdapat input validasi yang terdapat pada module isusernamepassAvailable(),apabila *player* memasukkan input username yang sudah terdapat dalam file dataregistrasi.txt ,maka program ini akan terus mengulang untuk meminta input username yang belum pernah dimasukkan sebelumnya.Lalu program ini juga akan secara otomatis menghasilkan ID bagi para pendaftar dengan menggunakan module IDGenerator()

Module isusernamepassAvailable()

Chart, diagram

Description automatically generated

Module isusernamepassAvailable() berfungsi untuk mengecek apakah username yang telah diniputkan user telah terdapat pada file dataregistrasi.txt atau belum,apabila sudah ada maka module ini akan memberikan return 1,namun apabila belum ada akan memberikan return 0

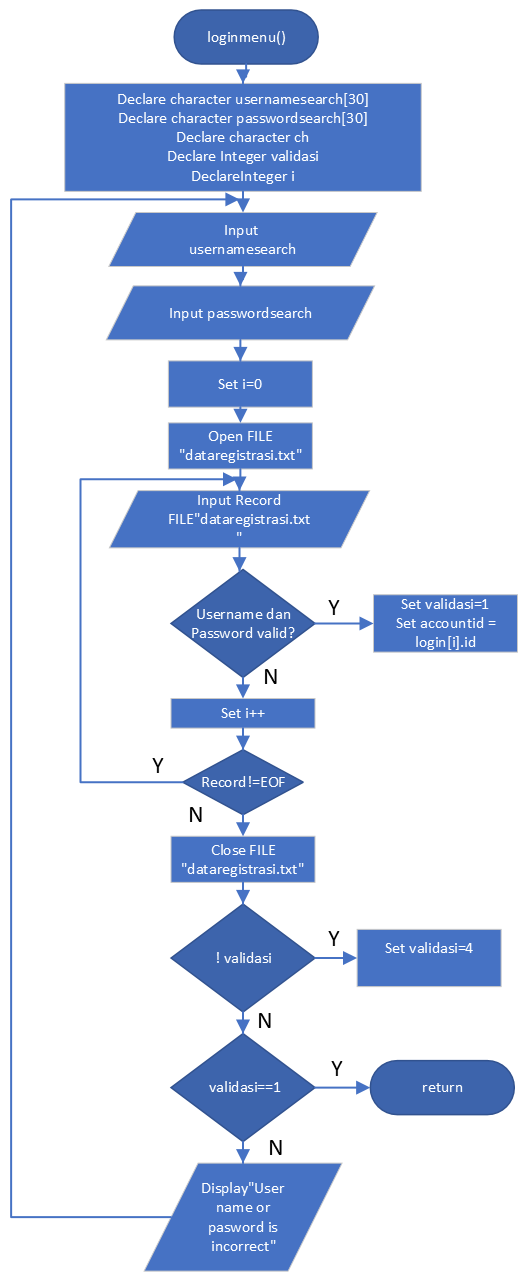
Module IDGenerator()

Diagram

Description automatically generated

Module IDGenerator() berfungsi untuk mengambil id record terakhir pada file dataregistrasi.txt dan id tersebut akan ditambahkan dengan 1 sebagai ID untuk pendaftar berikutnya,dan apabila masih belum terdapat record sebelumnya ,maka module ini akan secara otomatis menghasilkan id 1.

-Module loginmenu()



Module loginmenu() berfungsi untuk melakukan validasi username dan password para player apabila username yang diinput terdaftar dalam file dataregistrasi.txt maka player bisa masuk ke dalam game() dan program akan terus meminta input user sampai username dan password yang dimasukkan valid.

-Module game()

Diagram

Description automatically generated

Module Game() ini berfungsi untuk menampilkan menu pilihan bagi para user,apabila user memilih angka1 maka game akan dimulai,apabila user memilih angka 2 maka akan menampilkan hasil urutan score para *player,*untuk input 3 berarti mengakhiri dan menyimpan data terbaru dari program tersebut ke dalam file scores.txt.

-Module menuoption()

Diagram

Description automatically generated

Module menuoption() ini berfungsi sebagai core dalam fitur permainan Milliunaire berikut,pertama-tama sistem akan mereset Rank menggunakan module resetRank() dan juga pertanyaan menggunakan module resetquestion() lalu untuk semua variable juga akan direset terlebih dahulu karena kalua tidak program ini akan berisi data dari permainan sebelumnya kemudian akan diteruskan pada module randomQ() yang berfungsi untuk menghasilkan angka secara random yang dapat digunakan untuk membuat index acak pada *question* ,kemudian sistem akan mengeksekusi module tampilrangking() yang dapat menampilkan level permainan sedang berada di posisi mana dan menampilkan score yang akan didapat oleh *player* apabila dapat menjawab pertanyaan tersebut,kemudian akan mengeksekusi module play() yang berfungsi untuk memberikan tampilan berupa table dan pertanyaan yang ditujukan pada *player.*

Chart, funnel chart

Description automatically generated

User hanya dapat memberikan input ‘E ‘ sekali saja , apabila user memberikan input E sebanyak 2 kali ataupun memberikan input diluar A-D maka sistem akan meminta user untuk mengulang memberikan input kembali ,dan apabila *player* telah memberikan input berubah pilihan A-D ,maka program akan memastikan dengan meminta kembali verifikasi user berupa input‘Y’/’N’,apabila user belum memakai fitur 50:50 yang berupa inputan E maka sistem tersebut akan mengeksekusi module removewrongaswer() yang bertujuan untuk menghilangkan 2 pertanyaan yang salah secara acak.User harus memberikan input dalam huruf kapital agak dapat terbaca oleh program ini.

Diagram

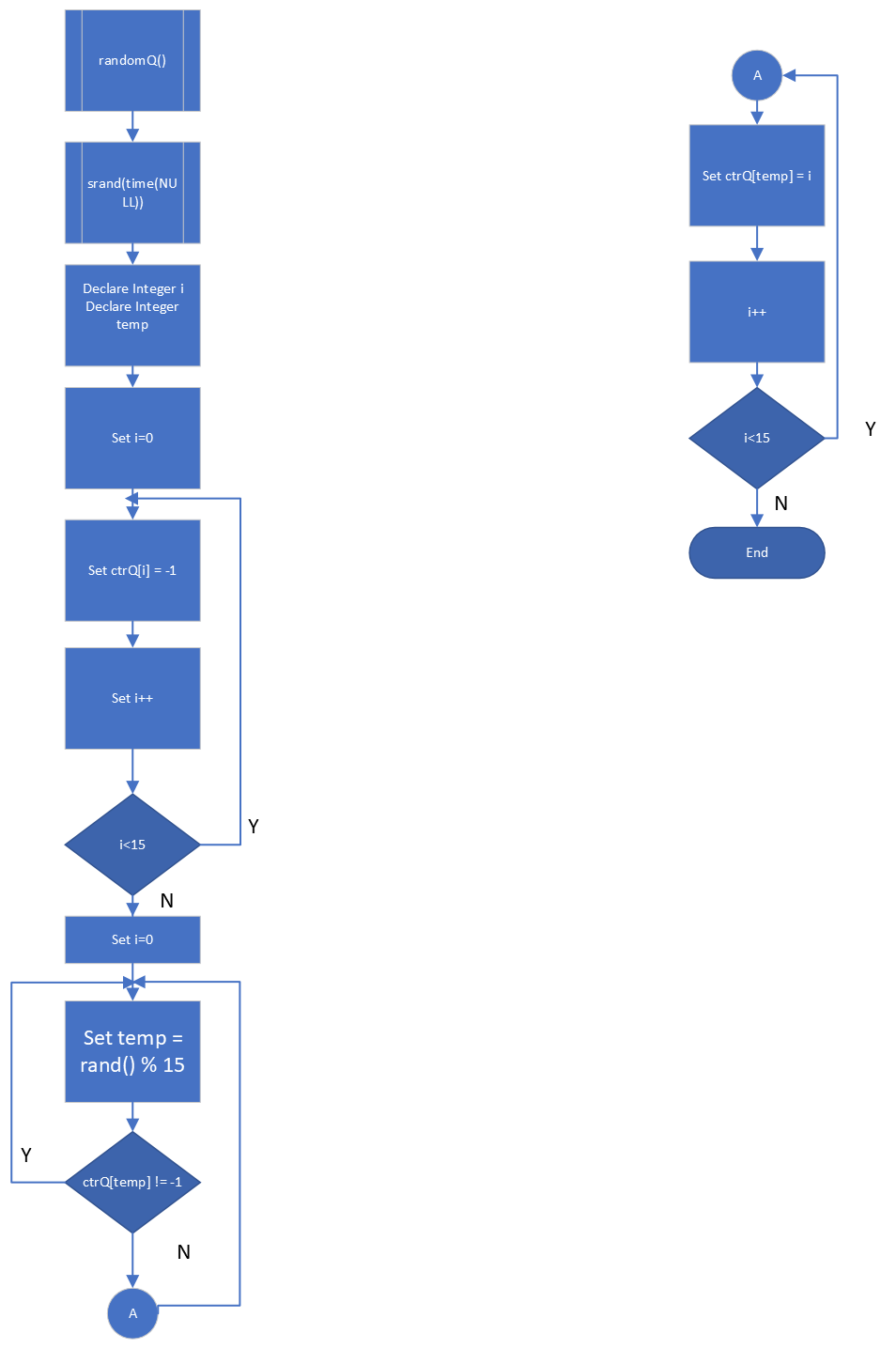
Description automatically generated

A screenshot of a computer

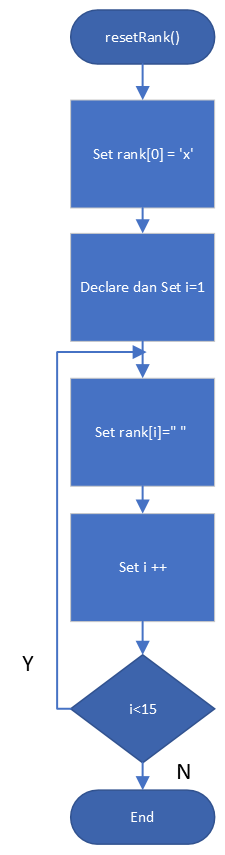
Description automatically generated with low confidence

Apabila *player* memilih F berarti user telah menyerah dan akan mendapat hadiah sejumlah ½ dari skor level *player*,dan menyimpan waktu dan tanggal berakhirnya permainan ini ,kemudian meminta *nickname* para *player* dan apabila jawaban yang diberikan user benar ,maka program ini akan mengulang untuk lanjut ke level berikutnya ,namun apabila jawaban salah maka program akan memberikan skor berdasarkan *safe zone* yang telah dilewati oleh para *player* dan menyimpan waktu dan tanggal berakhirnya permainan ini ,kemudian meminta *nickname* para *player.*

-Module randomQ()



-Modul resetRank()

**

-Modul resetquestions()

Diagram

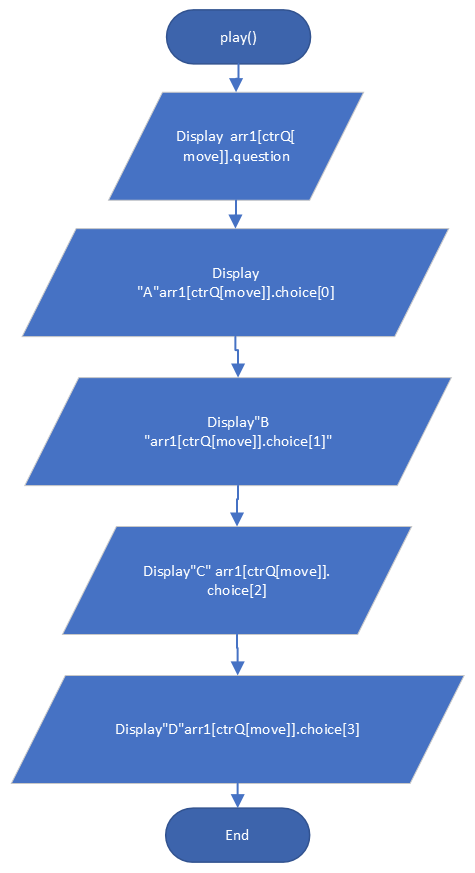
Description automatically generated

Module tampilrangking()

Diagram

Description automatically generated

-Module play()

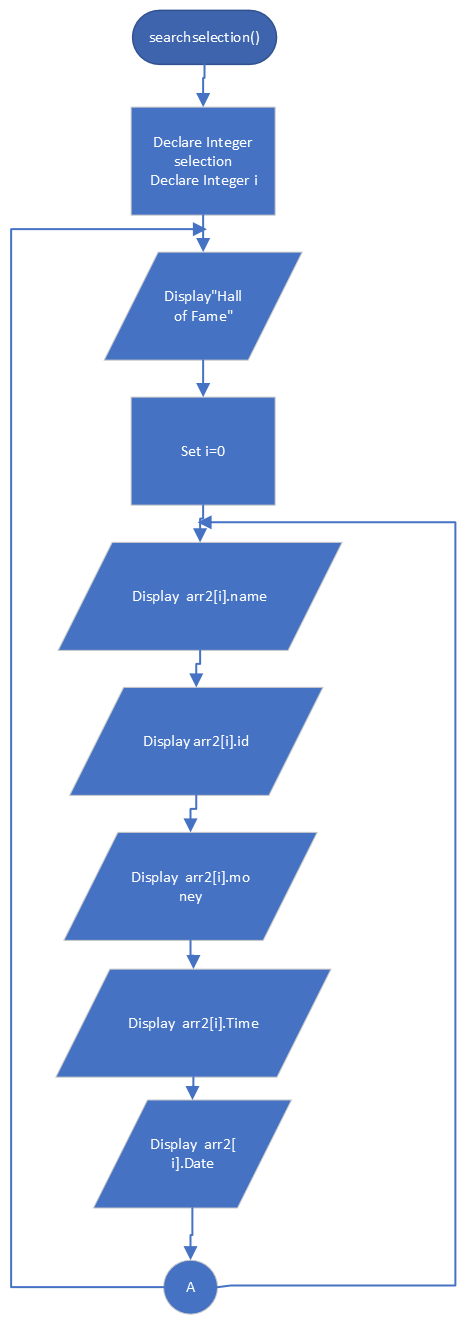


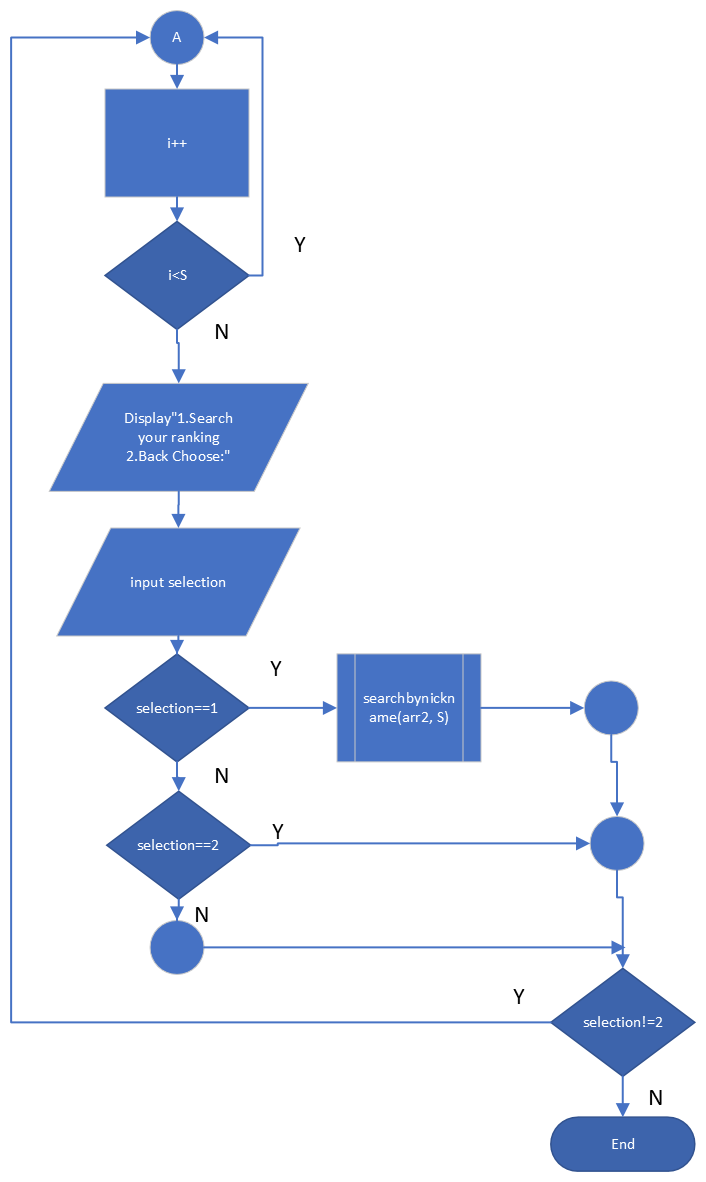
-Module removewronganswer()

Diagram

Description automatically generated

-Module searchselection()





Module searchselection() ini digunakan untuk menampilkan daftar player yang telah disortir sebelumnya menggunakan module quickSort .Di module ini juga memberikan pilihan pada user apabila user memberikan input 1 maka *player* dapat melakukan searching untuk record data nya berada di rank berapa,sedangkan untuk input 2 akan kembali ke menu sebelumnya yaitu module game()

-Module searchbynickname()

Diagram

Description automatically generated

Module searchbynickname() ini berguna agar player dapat langsung menemukan data recordnya tanpa harus melihat semua record satu persatu.

-Module loginadmin()

Diagram

Description automatically generated

Module beikut ini berfungsi untuk meminta input user lalu mengecek kevalidan username dan passwordnya

-Module MenuAdmin()

Timeline

Description automatically generated

A picture containing text, clock, sign

Description automatically generated

Modul ini berfungsi untuk memberikan 3 pilihan menu pada user yaitu pilihan 1-4

Apabila user memberi input 1 ,maka akan mengeksekusi biodataawardee().

Apabila user memberi input 2 ,maka akan mengeksekusi displayuser(), kemudian editdata(),kemudian displayuser()

Apabila user memberi input 3 ,maka akan mengeksekusi displayuser(),kemudian deletedata()

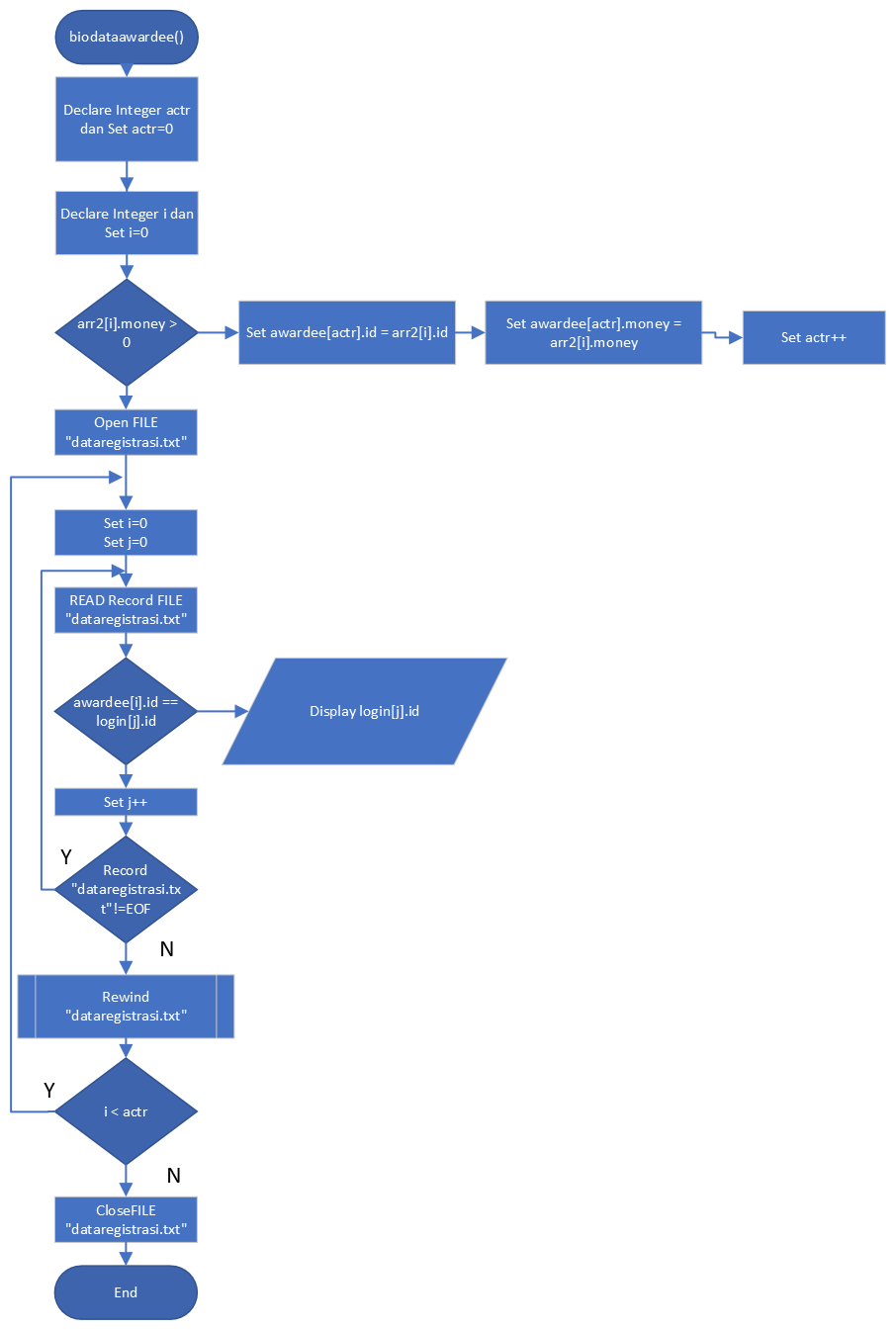
Apabila user memberi input 4,maka akan mengakhiri program

Apabila user memberi input di luar range 1-4 maka program akan mengulang

Modul ini juga dilengkapi input validasi bilangan integer.

-Module biodataawardee()

Module berikut ini digunakan untuk menampilkan data pribadi para peraih hadiah



-Module displayuser()

Diagram

Description automatically generated

Module berikut ini digunakan untuk menampilkan record pada dataregristrasi.txt

-Module editdata()

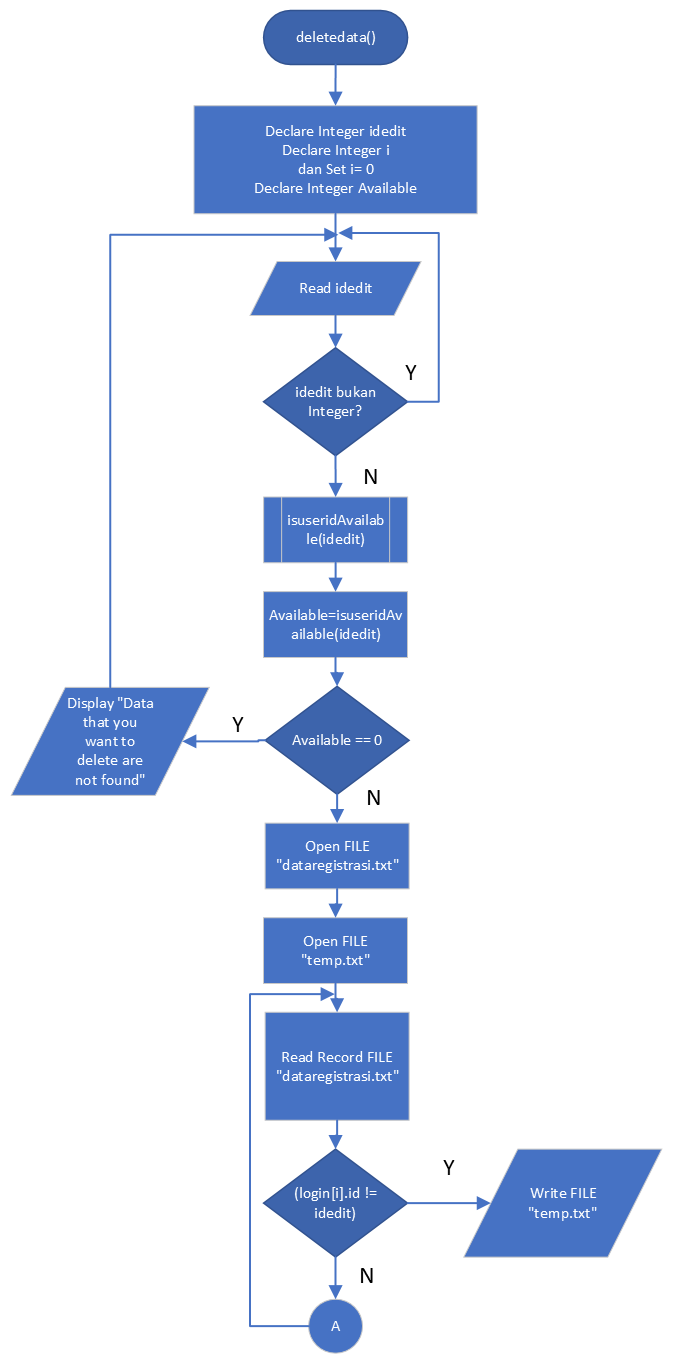
Chart, diagram

Description automatically generated Chart, funnel chart

Description automatically generated

Module editdata() berfungsi untuk melakukan edit pada data informasi yang salah diinputkan saat registrasi. Pada module ini akan dilakukan pengecekan apabila id *player* tidak ada maka program akan menampilkan”*Data that you want ot edit are not found*” kemudian mengulangi untuk meminta input.

Module deletedata()

****

**Diagram

Description automatically generated**

Module berikut ini digunakan untuk menghapus record para *player* yang sudah tidak digunakan . Pada module ini akan dilakukan pengecekan apabila id *player* tidak ada maka program akan menampilkan”*Data that you want ot update are not found*” kemudian mengulangi untuk meminta input.

Module int isuseridAvailable(int idedit)

Module berikut ini digunakan untuk mengecek apakah id yang diinput benar benar tersedia dalam file”dataregistrasi.txt”

Diagram

Description automatically generated

**Bab 3 Program dan Penjelasannya**

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

#include <time.h>

#include <conio.h>

#define USERNAME "adminmillionaire"

#define PASSWORD "admin1234"

FILE \*fp1;

FILE \*fp2;

FILE \*fp;

FILE \*fw;

int accountid;

//Module untuk tampilan awal pada game

void home()

{

printf(" \_\_ \_\_ \_ \_ \_ \_ \_\n");

printf("| \\/ |(\_)| || |(\_) (\_)\n");

printf("| \\ / | \_ | || | \_ \_\_\_ \_ \_\_ \_\_ \_ \_ \_ \_\_ \_\_\_\n");

printf("| |\\/| || || || || | / \_ \\ | '\_ \\ / \_` || || '\_\_|/ \_ \\\n");

printf("| | | || || || || || (\_) || | | || (\_| || || | | \_\_/\n");

printf("|\_| |\_||\_||\_||\_||\_| \\\_\_\_/ |\_| |\_| \\\_\_,\_||\_||\_| \\\_\_\_|\n\n\n");

printf("MENU\n");

printf("1. Let's Play!!\n");

printf("2. Hall of Fame\n");

printf("3. Exit and Save Hall of Fame\n");

}

struct data

{

int id;

char username[30];

char pass[30];

char noWA[30];

char noRekening[30];

} login[100];

struct q

{

char question[70];

char choice[4][35];

int answer;

} arr1[15];

int Q = 0;

void readQuestions()

{

while (!feof(fp1))

{

fscanf(fp1, "%[^#]#%[^#]#%[^#]#%[^#]#%[^@]@%d\n", &arr1[Q].question, &arr1[Q].choice[0], &arr1[Q].choice[1], &arr1[Q].choice[2], &arr1[Q].choice[3], &arr1[Q].answer);

Q++;

}

}

struct s

{

int id;

char name[15];

long long money;

char Time[20];

char Date[20];

} arr2[999], awardee[999];

int S = 0;

//Module untuk membaca file scores.txt

void readScores()

{

while (!feof(fp2))

{

fscanf(fp2, "%d#%[^#]#%lld#%[^#]#%[^#]#\n", &arr2[S].id, arr2[S].name, &arr2[S].money, arr2[S].Time, arr2[S].Date);

S++;

}

}

//Module untuk melakukan swap struct s yang berisi data record dari file scores.txt

void swap(struct s \*a, struct s \*b)

{

struct s temp = \*a;

\*a = \*b;

\*b = temp;

}

//Module yang digunakan untuk menemukan posisi pivot yang sesuai

int partition(struct s arr2[], int l, int r)

{

int i = l - 1;

int pv = arr2[r].money;

for (int a = l; a <= r - 1; a++)

{

if (arr2[a].money >= pv)

{

i++;

swap(&arr2[i], &arr2[a]);

}

}

i++;

swap(&arr2[i], &arr2[r]);

return i;

}

//Module untuk melakukan quick sort

void quickSort(struct s arr2[], int l, int r)

{

if (l < r)

{

int mid = partition(arr2, l, r);

quickSort(arr2, l, mid - 1);

quickSort(arr2, mid + 1, r);

}

}

char rank[15];

long long chuan[16] = {500, 1000, 2000, 3000, 5000, 7500, 10000, 15000, 25000, 50000, 75000, 150000, 250000, 500000, 1000000};

long long score;

int move;

int win, lost, surrender, hint;

char TimeGameOver[20];

char DateGameOver[20];

int ctrQ[15];

//Module untuk generate pertanyaan secara acak

void randomQ()

{

srand(time(NULL));

//assign array pada ctrQ dengan nilai-1

for (int i = 0; i < 15; i++){

ctrQ[i] = -1;

}

for (int i = 0; i < 15; i++)

{

//generate index counter question untuk record pada questions.txt secara acak

int temp = rand() % 15;

//apabila index telah digunakan maka akan secara berulang generate index baru

while (ctrQ[temp] != -1)

{

temp = rand() % 15;

}

//ctr[0-14]ini akan memiliki nilai yang acak dari range 0-14

ctrQ[temp] = i;

}

}

//Module yang digunakan untuk mereset tampilan ranking awal setelah game selesai dan kalah

void resetRank()

{

rank[0] = 'x';

for (int i = 1; i < 15; i++)

rank[i] = ' ';

}

//Module untuk menampilkan pertanyaan

void play()

{

//untuk membuat table pertanyaan

system("cls");

for (int i = 0; i < 73; i++)

{

if (i == 0)

printf("%c", 201);

else if (i == 72)

printf("%c", 187);

else

printf("%c", 205);

}

printf("\n%c %-70s%c\n", 186, arr1[ctrQ[move]].question, 186);

//untuk membuat table dan pilihan jawaban

for (int i = 0; i < 73; i++)

{

if (i == 0)

printf("%c", 204);

else if (i == 36)

printf("%c", 203);

else if (i == 72)

printf("%c", 185);

else

printf("%c", 205);

}

printf("\n%c A: %-31s%c B: %-31s%c\n", 186, arr1[ctrQ[move]].choice[0], 186, arr1[ctrQ[move]].choice[1], 186);

//untuk membuat table dan pilihan jawaban

for (int i = 0; i < 73; i++)

{

if (i == 0)

printf("%c", 204);

else if (i == 36)

printf("%c", 206);

else if (i == 72)

printf("%c", 185);

else

printf("%c", 205);

}

printf("\n%c C: %-31s%c D: %-31s%c\n", 186, arr1[ctrQ[move]].choice[2], 186, arr1[ctrQ[move]].choice[3], 186);

//untuk membuat table dan pilihan jawaban

for (int i = 0; i < 73; i++)

{

if (i == 0)

printf("%c", 200);

else if (i == 36)

printf("%c", 202);

else if (i == 72)

printf("%c", 188);

else

printf("%c", 205);

}

puts("");

}

//Module untuk melakukan searching daftar pemain berdasarkan nicknamenya

void searchbynickname(struct s arr2[], int S)

{

char nickname[30];

int i, j;

int found = 0;

printf("What is your nickname?");

scanf("%[^\n]", nickname);

getchar();

system("cls");

for (i = 0; i < S; i++)

{

if (strcmp(arr2[i].name, nickname) == 0)

{

found = 1;

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

printf("||Rank||Nama\t\t||ID||\tMoney\t||\tTime\t||\tDate\t\t||\n");

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

printf("||%03d. %-18s %-5d %7lld %14s %18s\t||\n", i + 1, arr2[i].name, arr2[i].id, arr2[i].money, arr2[i].Time, arr2[i].Date);

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

}

}

if (found == 0)

{

printf("\033[1;31m");

printf("\*Record not found\n");

printf("\033[0m");

}

system("pause");

system("cls");

}

//Module untuk menampilkan data Hall of Fame dan menanyakan input apakah ingin melakukan searching atau tidak

void searchselection(struct s arr2[], int S)

{

int selection;

do

{

printf("Hall of Fame\n");

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

printf("||Rank||Nama\t\t||ID||\tMoney\t||\tTime\t||\tDate\t\t||\n");

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

for (int i = 0; i < S; i++)

printf("||%03d. %-18s %-5d %7lld %14s %18s\t||\n", i + 1, arr2[i].name, arr2[i].id, arr2[i].money, arr2[i].Time, arr2[i].Date);

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

puts("");

printf("1.Search your ranking\n2.Back\nChoose:");

scanf("%d", &selection);

getchar();

switch (selection)

{

case 1:

searchbynickname(arr2, S);

break;

case 2:

break;

}

} while (selection != 2);

}

//Module untuk permainan millionaire

int game()

{

int menu = 0;

//apabila input menu selain 3 maka program akan kembali ke sini

while (menu != 3)

{

system("cls");

home();

//Input validasi apabila input user di luar range 1-3

do

{

printf(">> ");

scanf("%d", &menu);

getchar();

} while (menu < 1 || menu > 3);

//Berikut ini saya asumsikan sebagai menuoption() untuk di flowchart:

//apabila user memberi input 1 maka akan mulai permainan

//apabila user memberi input 2 maka akan melihat data Hall of Fame

//apabila user memberi input 3 maka akan keluar dari program

switch (menu)

{

case 1:

{

char choice1[30];

char choice2[30];

int ctrH1;

int ctrH2;

int movehint;

resetRank();

//Berikut ini saya asumsikan sebagai resetquestions()untuk di flowchart:

for(int i=0;i<15;i++){

if(i==movehint&&hint ==1&&surrender==1||i==movehint&&hint ==1&&lost==1){

strcpy(arr1[i].choice[ctrH1],choice1);

strcpy(arr1[i].choice[ctrH2],choice2);

}

}

score = 0;

move = 0;//Variabel move disini digunakan sebagai stage/level dalam permainan ini

win = 0, lost = 0, surrender = 0, hint = 0;//variable hint ini digunakan sebagai penanda apakah 50:50 sudah digunakan atau belum

char choose, lock;

//Generate counter question dari index 0-14 dengan memiliki nilai yang acak.

randomQ();

while (win == 0 && lost == 0 && surrender == 0 && move < 15)

{

system("cls");

//Menampilkan Stage/level permainan

//Berikut ini saya asumsikan sebagai tampilrangking() untuk di flowchart:

for (int i = 14; i >= 0; i--)

{

printf("[%c] $%7lld", rank[i], chuan[i]);

//Menampilkan safe zone pada setiap element kelipatan 5

if ((i + 1) % 5 == 0)

printf("\t<- Safe Zone %d", (i + 1) / 5);

puts("");

}

printf("\nGet ready for $%lld question! Press ENTER to continue...", chuan[move]);

getchar();

do

{

//Menampilkan soal dan jawaban

play();

do

{

do

{ //apabila fitur 50:50 belum digunakan ,maka akan menampilkan "Input 'E' to use 50:50"

if (hint == 0)

printf("\nInput 'E' to use 50:50");

printf("\nInput 'F' to surrender\n");

printf("Your answer: ");

scanf("%c", &choose);

getchar();

//apabila fitur 50:50 telah digunakan,maka akan menampilkan "You have used your 50:50!\n"

if (choose == 'E' && hint != 0)

printf("You have used your 50:50!\n");

//apabila inputan user diluar A-F atau inputan adalah E dan telah menggunakan fitur 50:50 maka akan terus mengulangi program do

} while ((choose < 'A' || choose > 'F') || (choose == 'E' && hint != 0));

//apabila inputan user adalah A-D, maka akan mengeksekusi statement yang ada di dalam nya

if (choose >= 'A' && choose <= 'D')

{

do

{

printf("Lock your answer to %c? [Y/N]: ", choose);

scanf("%c", &lock);

getchar();

//apabila input user tidak samd dengan Y dan tidak sama dengan N maka program akan terus meminta input user

} while (lock != 'Y' && lock != 'N');

}//apabila user memberikan input N yang berarti No,maka program akan terus mengulang untuk mengambil input variabel choose

} while (lock == 'N' && choose >= 'A' && choose <= 'D');

//apabila user menggunakan fitur 50:50 akan mengeksusi statement berikut ini

if (choose == 'E' && hint == 0)

{

//Berikut ini saya asumsikan sebagai removewronganswer() untuk di flowchart:

srand(time(NULL));

hint = 1;

//generate random number antara 0-3

ctrH1 = rand() % 4;

//apabila jawabannya benar maka akan terus generate random number dengan range angka 0-3

//kalau sudah ketemu jawaban yang salah maka akan diganti dengan value '-'

while (ctrH1 == arr1[ctrQ[move]].answer - 1)

{

ctrH1 = rand() % 4;

}

movehint=ctrQ[move];

strcpy(choice1,arr1[ctrQ[move]].choice[ctrH1]);

strcpy(arr1[ctrQ[move]].choice[ctrH1], "-");

//generate random number antara 0-3

ctrH2 = rand() % 4;

//apabila jawabannya benar atau sama dengan yang sudah dihilangkan sebelumnya maka akan terus generate

//random number dengan range 0-3

//kalau sudah ketemu jawaban yang salah maka akan diganti dengan value'-'

while (ctrH1 == ctrH2 || ctrH2 == arr1[ctrQ[move]].answer - 1)

{

ctrH2 = rand() % 4;

}

strcpy(choice2,arr1[ctrQ[move]].choice[ctrH2]);

strcpy(arr1[ctrQ[move]].choice[ctrH2], "-");

}

//Setiap user memberi input E maka program akan mengulang untuk menampilkan pertanyaan

} while (choose == 'E');

//Apabila user memberikan input F maka berarti user menyerah dan mengeksekusi statement sebagai berikut

//dan player akan mendapatkan 1/2 dari prize sebenarnya

if (choose == 'F')

{

surrender = 1;

strcpy(TimeGameOver, \_\_TIME\_\_);

strcpy(DateGameOver, \_\_DATE\_\_);

score /= 2;

system("cls");

printf("Thank you for playing.\n");

printf("You got $%lld\n", score);

}

//apabila jawaban yang diinputkan oleh player benar maka akan mengeksekusi statement berikut ini:

else if (arr1[ctrQ[move]].answer == choose - 'A' + 1)

{

score = chuan[move];

rank[move % 15] = ' ';

rank[(move + 1) % 15] = 'x';

move++;

printf("\nYou are right! You won $%lld\n", score);

printf("Press ENTER to continue...");

getchar();

//apabila telah menjawab 15 pertanyaan dengan tepat maka program akan mengeksekusi statement berikut ini:

if (move == 15)

{

system("cls");

printf("Congratulations! You are the next Millionare\n");

}

}

//Apabila jawabannya salah maka mengeksekusi statement berikut ini:

else if (arr1[ctrQ[move]].answer != choose - 'A' + 1)

{

strcpy(TimeGameOver, \_\_TIME\_\_);

strcpy(DateGameOver, \_\_DATE\_\_);

lost = 1;

move--;

//Apabila level/stage pertanyaan melewati safe zone maka stage/level akan dikurangi sampai dengan area safe zone

//jadi apabila player salah dalam menjawab pertanyaan maka akan mendapatkan score sebesar prize di area

//safe zone.

while ((move + 1) % 5 != 0 && move > 0)

move--;

//apabila move atau level atau stage tidak sama dengan 0,maka player masih akan mendapatkan prize sesuai dengan safe zone terdekat

if (move != 0)

score = chuan[move];

else

score = 0;

system("cls");

printf("Time Game Over:%s\n", TimeGameOver);

printf("Date Game Over:%s\n", DateGameOver);

printf("Sorry, you got the wrong answer\n");

printf("Thank you for playing.\n");

printf("You got $%lld\n", score);

}

}

char player[15];

do

{

printf("\nPlease input your nickname [5-10 characters](doesn't allow any space): ");

scanf("%s", player);

getchar();

//apabila input nickname yang dimasukkan oleh user diluar range 5-10,maka akan terus mengulang untuk meminta input user

} while (strlen(player) < 5 || strlen(player) > 10);

arr2[S].money = score;

strcpy(arr2[S].name, player);

strcpy(arr2[S].Time, TimeGameOver);

strcpy(arr2[S].Date, DateGameOver);

arr2[S].id = accountid;

S++;

printf("\nPress ENTER to continue...");

getchar();

break;

}

case 2:

{

quickSort(arr2, 0, S - 1);

system("cls");

searchselection(arr2, S);

printf("Press ENTER to continue...");

getchar();

break;

}

case 3:

{

quickSort(arr2, 0, S - 1);

fp2 = fopen("scores.txt", "w");

//mencetak kembali ke scores.txt data yang telah di update dan disortir secara descending berdasarkan money yang diperoleh oleh player

for (int i = 0; i < S; i++)

fprintf(fp2, "%d#%s#%lld#%s#%s#\n", arr2[i].id, arr2[i].name, arr2[i].money, arr2[i].Time, arr2[i].Date);

puts("");

printf("\"scores.txt\" file has been updated\n");

printf("Thank you for playing!\n\nfor prize please wait any information from us on Whats App\n\n");

printf("Press ENTER to exit...\n");

getchar();

exit(0);

break;

}

}

}

}

//Modul untuk mengecek apakah username telah ada di dalam file dataregistrasi.txt atau belum

int isusernamepassAvailable(char nama[])

{

int i = 0;

//fp sebagai file streamer untuk baca file "dataregistrasi.txt"

fp = fopen("dataregistrasi.txt", "r");

//Membaca semua record yang ada di dalam file "dataregistrasi.txt" dan ditampung ke dalam variable struct login

while (fscanf(fp, "%d,%[^,],%[^,],%[^,],%[^,],\n", &login[i].id, login[i].username, login[i].pass, login[i].noWA, login[i].noRekening) != EOF)

{

//Mengecek apabila username telah ada di dalam file "dataregistrasi.txt",maka akan meberikan return 1

if (strcmp(nama, login[i].username) == 0)

{

fclose(fp);

return 1;

}

i++;

}

fclose(fp);

return 0;

}

//Module untuk player melakukan registrasi

void registrasi()

{

int i;

int a;

int id;

char username[30];

char pass[30];

char noWA[30];

char noRekening[30];

static int Available;

do

{

printf("Username:");

scanf("%[^\n]", username);

getchar();

Available = isusernamepassAvailable(username);

//untuk mengecek apabila username yang dimasukkan telah ada,maka user harus mengulangi untuk memasukkan input

if (Available == 1)

{

printf("Username that you want to register has been exist please insert other username\n");

}

} while (Available == 1);

printf("Password:");

scanf("%[^\n]", pass);

getchar();

printf("Whatsapp Number:");

scanf("%s", noWA);

getchar();

printf("Rekening Number:");

scanf("%s", noRekening);

getchar();

i = 0;

//Berikut ini saya asumsikan sebagai IDGenerator() untuk di flowchart:

//ID GENERATOR:

//Mengambil index untuk recordan data terakhir:

//fp sebagai file streamer untuk baca file "dataregistrasi.txt"

fp = fopen("dataregistrasi.txt", "r");

//Membaca semua record yang ada di dalam file "dataregistrasi.txt" dan ditampung ke dalam variable struct login

while (fscanf(fp, "%d,%[^,],%[^,],%[^,],%[^,],\n", &login[i].id, login[i].username, login[i].pass, login[i].noWA, login[i].noRekening) != EOF)

{

i++;

}

fclose(fp);

a = i - 1;

i = 0;

rewind(fp);

//fp sebagai file streamer untuk baca record file "dataregistrasi.txt"

fp = fopen("dataregistrasi.txt", "r");

//Membaca semua record yang ada di dalam file "dataregistrasi.txt" dan ditampung ke dalam variable struct login

while (fscanf(fp, "%d,%[^,],%[^,],%[^,],%[^,],\n", &login[i].id, login[i].username, login[i].pass, login[i].noWA, login[i].noRekening) != EOF)

{

// mengecek id pada record index terakhir dan akan ditambahkan dengan angka 1 untuk generate id secara urut berdasarkan id pada record sebelumnya

if (i == a)

{

id = login[i].id + 1;

}

i++;

}

//ketika masih belum ada record di dalam file "dataregistrasi.txt" yang masuk maka akan generate id =1

if (a == -1)

{

id = 1;

}

fclose(fp);

//fp sebagai file streamer untuk menambahkan record file "dataregistrasi.txt"

fp = fopen("dataregistrasi.txt", "a");

fprintf(fp, "\n");

fprintf(fp, "%d,%s,%s,%s,%s,", id, username, pass, noWA, noRekening);

fclose(fp);

printf("You are succesfully registered\n");

}

//Module untuk melakukan validasi username dan password multiuser

void loginmenu()

{

char usernamesearch[30];

char passwordsearch[30];

char ch;

int validasi;

int i;

do

{

printf("Username:");

scanf("%[^\n]", usernamesearch);

getchar();

printf("Password:");

scanf("%[^\n]", passwordsearch);

getchar();

i = 0;

//fp sebagai file streamer untuk baca record file "dataregistrasi.txt"

fp = fopen("dataregistrasi.txt", "r");

//Membaca semua record yang ada di dalam file "dataregistrasi.txt" dan ditampung ke dalam variable struct login

while (fscanf(fp, "%d,%[^,],%[^,],%[^,],%[^,],\n", &login[i].id, login[i].username, login[i].pass, login[i].noWA, login[i].noRekening) != EOF)

{

//Mengecek apabila username dan password yang diinputkan ada dalam file dataregistrasi.txt, maka akan menyimpan idnya

//dan assign value 1 di variable validasi

if (strcmp(login[i].username, usernamesearch) == 0 && strcmp(login[i].pass, passwordsearch) == 0)

{

validasi = 1;

accountid = login[i].id;

}

i++;

}

fclose(fp);

//mengecek apabila variable validasi tidak memiliki value di dalamnya ,maka validasi akan di assign dengan value 4

if (!validasi)

{

validasi = 4;

}

//Mengecek apabila validasi=1,maka akan kembali ke main function dan lanjut untuk mengeksekusi statement berikutnya

if (validasi == 1)

{

return;

}

else

{

printf("Username or pasword is incorrect\n");

}

//Kalau validasi belum bernilai 1 maka program akan terus mengulang untuk meminta input username dan password

} while (validasi != 1);

}

//Module untuk menampilkan record player yang memperoleh prize

void biodataawardee()

{

int actr = 0;

//mengulangi untuk generate index dari 0 sampai kurang dari jumlah nickname player(S)

for (int i = 0; i < S; i++)

{

//Mengecek player yang memiliki nilai prize lebih dari 0 dan akan ditampung id dan money yang diperoleh pada variabel awardee

if (arr2[i].money > 0)

{

awardee[actr].id = arr2[i].id;

awardee[actr].money = arr2[i].money;

actr++;

}

}

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

printf("||ID ||Username\t||Password||Nomor Whatsapp ||\tNo Rekening\t||Prize ||\n");

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

//fp sebagai file streamer untuk baca file "dataregistrasi.txt"

fp = fopen("dataregistrasi.txt", "r");

for (int i = 0; i < actr; i++)

{

int j = 0;

//Membaca semua record yang ada di dalam file "dataregistrasi.txt" dan ditampung ke dalam variable struct login

while (fscanf(fp, "%d,%[^,],%[^,],%[^,],%[^,],\n", &login[j].id, login[j].username, login[j].pass, login[j].noWA, login[j].noRekening) != EOF)

{

//mengecek apabila sesuai dengan id penerima hadiah,maka record pada dataregistrasi.txt akan diprint

if (awardee[i].id == login[j].id)

{

printf("||%-3d||%-9s||%-8s||%-15s||%-19s||%-7lld||\n", login[j].id, login[j].username, login[j].pass, login[j].noWA, login[j].noRekening, awardee[i].money);

}

j++;

}

rewind(fp);

}

fclose(fp);

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

}

//Module untuk menampilkan record dataregistrasi.txt

void displayuser()

{

int i = 0;

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

printf("||ID ||Username\t||Password||Nomor Whatsapp ||No Rekening\t||\n");

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

//fp sebagai file streamer untuk baca file "dataregistrasi.txt"

fp = fopen("dataregistrasi.txt", "r");

//Membaca semua record yang ada di dalam file "dataregistrasi.txt" dan ditampung ke dalam variable struct login

while (fscanf(fp, "%d,%[^,],%[^,],%[^,],%[^,],\n", &login[i].id, login[i].username, login[i].pass, login[i].noWA, login[i].noRekening) != EOF)

{

printf("||%-3d||%-9s||%-8s||%-15s||%-19s||\n", login[i].id, login[i].username, login[i].pass, login[i].noWA, login[i].noRekening);

i++;

}

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

fclose(fp);

}

//Module untuk mengecek apakah user ID nya sudah tersedia dalam dataregistrasi.txt atau belum

int isuseridAvailable(int idedit)

{

int i = 0;

//fp sebagai file streamer untuk baca file "dataregistrasi.txt"

fp = fopen("dataregistrasi.txt", "r");

//Membaca semua record yang ada di dalam file "dataregistrasi.txt" dan ditampung ke dalam variable struct login

while (fscanf(fp, "%d,%[^,],%[^,],%[^,],%[^,],\n", &login[i].id, login[i].username, login[i].pass, login[i].noWA, login[i].noRekening) != EOF)

{

//Apabila id yang dicari ada dalam file "dataregistrasi.txt" ,maka akan memberikan return 1

if (idedit == login[i].id)

{

fclose(fp);

return 1;

}

i++;

}

fclose(fp);

return 0;

}

//Module untuk mengedit record dataregistrasi.txt

void editdata()

{

int idedit;

int i = 0;

int Available;

do

{

printf("What ID number of data do you want to edit?");

//Input validasi apabila user menginput selain integer maka program akan terus mengulang untuk meminta input

while (scanf("%d", &idedit) != 1)

{

printf("Your input can only be numbers\n");

printf("ID number: ");

fflush(stdin);

}

getchar();

Available = isuseridAvailable(idedit);

//Apabila user ID yang akan dihapus tidak ada,maka program akan menampilkan "Data that you want to delete are not found"

//dan akan terus mengulang untuk meminta input sampai ID yang diinputkan ada di dalam dataregistrasi.txt

if (Available == 0)

{

printf("Data that you want to edit are not found\n");

}

} while (Available == 0);

//fp sebagai file streamer untuk baca record file "dataregistrasi.txt" dan fw sebagai file streamer untuk menulis record ke dalam

//file "temp.txt" yang digunakan sebagai file penyimpanan sementara

fp = fopen("dataregistrasi.txt", "r");

fw = fopen("temp.txt", "w");

//Membaca semua record yang ada di dalam file "dataregistrasi.txt" dan ditampung ke dalam variable struct login

while (fscanf(fp, "%d,%[^,],%[^,],%[^,],%[^,],\n", &login[i].id, login[i].username, login[i].pass, login[i].noWA, login[i].noRekening) != EOF)

{

//Memasukkan record yang tidak diedit ke dalam temp file

if (login[i].id != idedit)

{

fprintf(fw, "%d,%s,%s,%s,%s,\n", login[i].id, login[i].username, login[i].pass, login[i].noWA, login[i].noRekening);

}

//Mengubah record yang akan diedit kemudian dimasukkan ke dalam temp file

else

{

printf("New username:");

scanf("%[^\n]", login[i].username);

getchar();

printf("New password:");

scanf("%[^\n]", login[i].pass);

getchar();

printf("New Whatsapp number:");

scanf("%s", login[i].noWA);

getchar();

printf("New rekening number:");

scanf("%s", login[i].noRekening);

getchar();

fprintf(fw, "%d,%s,%s,%s,%s,\n", login[i].id, login[i].username, login[i].pass, login[i].noWA, login[i].noRekening);

}

i++;

}

fclose(fp);

fclose(fw);

//menghapus "dataregistrasi.txt"

remove("dataregistrasi.txt");

//mengubah nama "temp.txt" menjadi "dataregistrasi.txt"

rename("temp.txt", "dataregistrasi.txt");

printf("Record has been succesfully updated\n");

}

//Module untuk menghapus record dataregistrasi.txt

void deletedata()

{

int idedit;

int i = 0;

int Available;

do

{

printf("What ID number of data do you want to delete?");

//Input validasi apabila user menginput selain integer maka program akan terus mengulang untuk meminta input

while (scanf("%d", &idedit) != 1)

{

printf("Your input can only be numbers\n");

printf("ID number: ");

fflush(stdin);

}

getchar();

Available = isuseridAvailable(idedit);

//Apabila user ID yang akan dihapus tidak ada,maka program akan menampilkan "Data that you want to delete are not found"

//dan akan terus mengulang untuk meminta input sampai ID yang diinputkan ada di dalam dataregistrasi.txt

if (Available == 0)

{

printf("Data that you want to delete are not found\n");

}

} while (Available == 0);

//fp sebagai file streamer untuk baca record file "dataregistrasi.txt" dan fw sebagai file streamer untuk menulis record ke dalam

//file "temp.txt" yang digunakan sebagai file penyimpanan sementara

fp = fopen("dataregistrasi.txt", "r");

fw = fopen("temp.txt", "w");

//Membaca semua record yang ada di dalam file "dataregistrasi.txt" dan ditampung ke dalam variable struct login

while (fscanf(fp, "%d,%[^,],%[^,],%[^,],%[^,],\n", &login[i].id, login[i].username, login[i].pass, login[i].noWA, login[i].noRekening) != EOF)

{

//Memasukkan record yang tidak dihapus ke dalam temp file

if (login[i].id != idedit)

{

fprintf(fw, "%d,%s,%s,%s,%s,\n", login[i].id, login[i].username, login[i].pass, login[i].noWA, login[i].noRekening);

}

i++;

}

fclose(fp);

fclose(fw);

//menghapus "dataregistrasi.txt"

remove("dataregistrasi.txt");

//mengubah nama "temp.txt" menjadi "dataregistrasi.txt"

rename("temp.txt", "dataregistrasi.txt");

printf("Record has been succesfully deleted\n");

}

//Module untuk memberikan pilihan menu admin

void MenuAdmin()

{

int selection;

do

{

printf("Menu\n1.Biodata Awardee\n2.Edit User Account\n3.Delete User Account\n4.Back\nChoose:");

//Input validasi apabila user menginput selain integer maka program akan terus mengulang untuk meminta input

while (scanf("%d", &selection) != 1)

{

printf("Your input can only be numbers\n");

printf("Choose: ");

fflush(stdin);

}

getchar();

//Apabila user memberi input 1 ,maka akan mengeksekusi biodataawardee()

//Apabila user memberi input 2 ,maka akan mengeksekusi displayuser(), kemudian editdata(),kemudian displayuser()

//Apabila user memberi input 3 ,maka akan mengeksekusi displayuser(),kemudian deletedata()

//Apabila user memberi input 4,maka akan mengakhiri program

//Apabila user memberi input diluar dari range 1-4,maka program akan terus mengulang untuk menanyakan menu apa yang dipilih

switch (selection)

{

case 1:

system("cls");

biodataawardee();

printf("Please click enter to continue");

getchar();

system("cls");

break;

case 2:

system("cls");

displayuser();

editdata();

system("pause");

system("cls");

printf("User data after editing please check\n");

displayuser();

printf("Please click enter to continue");

getchar();

system("cls");

break;

case 3:

system("cls");

displayuser();

deletedata();

system("pause");

system("cls");

printf("User data after editing please check\n");

displayuser();

printf("Please click enter to continue");

getchar();

system("cls");

break;

case 4:

break;

default:

printf("Your input is out of range ");

break;

}

} while (selection != 4);

}

//Module untuk mengecek validasi username dan password

void loginadmin()

{

char usernameAdmin[30], password[30];

do

{

printf("USERNAME:");

scanf("%[^\n]", usernameAdmin);

getchar();

printf("PASSWORD:");

scanf("%[^\n]", password);

getchar();

//Cek validasi username dan password

//Apabila username dan password yang diinput sudah terdaftar di dalam dataregistrasi.txt,maka akan bisa lanjut mengeksekusi

//statement berikutnya

if ((strcmp(USERNAME, usernameAdmin)) == 0 && (strcmp(PASSWORD, password)) == 0)

{

return;

}

//Apabila username dan password yang diinput belum terdaftar di dalam dataregistrasi.txt,maka akan

//menampilkan "Your Username or Password is incorrect" dan terus mengulang sampai username dan password yang dimasukkan ada

//di dalam dataregistrasi.txt

if ((strcmp(USERNAME, usernameAdmin)) != 0 && (strcmp(PASSWORD, password)) != 0)

{

printf("Your Username or Password is incorrect\n");

}

} while ((strcmp(USERNAME, usernameAdmin)) != 0 && (strcmp(PASSWORD, password)) != 0);

}

//Module untuk memberikan pertanyaan pilihan menu pada player apakah sudah punya akun atau belum,apabila belum mempunyai akun dapat

//melakukan registrasi terlebih dahulu

void playermenu()

{

int selection2;

do

{

system("cls");

printf("If you want to edit or delete your account you can contact the admin\n1.Registrasi\n2.Login\n3.Back\nChoose:");

//Input validasi apabila user menginput selain integer maka program akan terus mengulang untuk meminta input

while (scanf("%d", &selection2) != 1)

{

printf("Your input can only be numbers\n");

printf("Choose: ");

fflush(stdin);

}

getchar();

//Apabila user memberi input 1 ,maka akan mengeksekusi registrasi()

//Apabila user memberi input 2 ,maka akan mengeksekusi loginmenu(), kemudian game()

//Apabila user memberi input 3 ,maka akan kembali ke menu sebelumnya.

//Apabila user memberi input diluar dari range 1-3,maka program akan terus mengulang untuk menanyakan menu apa yang dipilih

switch (selection2)

{

case 1:

registrasi();

system("pause");

system("cls");

break;

case 2:

loginmenu();

game();

break;

case 3:

break;

default:

printf("Your input is out of range\n");

break;

}

} while (selection2 != 3);

}

int main()

{ //fp1 sebagai file streamer untuk baca record file "questions.txt" dan fp2 sebagai file streamer untuk baca record file "scores.txt"

int selection;

fp1 = fopen("questions.txt", "r");

fp2 = fopen("scores.txt", "r");

//jika file"questions.txt" tidak ada maka akan menampilkan "Fail to load \"questions.txt\" file.

//Aborting the application..."

if (fp1 == NULL)

{

printf("Fail to load \"questions.txt\" file.\n");

printf("Aborting the application...");

getchar();

return 0;

}

else{

//jika file"questions.txt" ada maka akan membaca file "questions.txt"

readQuestions();

}

//jika file "scores.txt" tidak ada, maka akan menampilkan "Fail to load \"questions.txt\" file."

//Press ENTER to continue...

if (fp2 == NULL)

{

printf("Fail to load \"scores.txt\" file.\n");

printf("Press ENTER to continue...");

getchar();

}

else

{

//jika file "scores.txt" ada maka akan membaca file "scores.txt"

readScores();

quickSort(arr2, 0, S - 1);

}

do

{

system("cls");

printf("1.Player\n2.Admin\n3.Exit\nChoose:");

//Input validasi apabila user menginput selain integer maka program akan terus mengulang

while (scanf("%d", &selection) != 1)

{

printf("Your input can only be numbers\n");

printf("Choose: ");

fflush(stdin);

}

getchar();

//Apabila user memberi input 1 ,maka akan mengeksekusi playermenu()

//Apabila user memberi input 2 ,maka akan mengeksekusi loginadmin(), kemudian MenuAdmin()

//Apabila user memberi input 3 ,maka akan mengakhiri program.

//Apabila user memberi input diluar dari range 1-3,maka program akan terus mengulang untuk menanyakan menu apa yang dipilih

switch (selection)

{

case 1:

//Modul untuk player

playermenu();

break;

case 2:

//Modul untuk Login admin

system("cls");

loginadmin();

system("cls");

//Modul untuk menu admin

MenuAdmin();

break;

case 3:

//exit

exit(0);

break;

default:

printf("Your input is out of range\n");

break;

}

} while (selection != 3);

fclose(fp1);

fclose(fp2);

return 0;

}

**Bab 4 Tampilan Hasil**

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Data yang baru diinput telah berhasil kesimpan di dalam file dataregistrasi.txt

Text

Description automatically generated

Text

Description automatically generated with medium confidence

Text

Description automatically generated

Text

Description automatically generated

Jawaban benar

Text

Description automatically generated

Jawaban salah

Text

Description automatically generated

Cek Score dan Rangking

Text

Description automatically generated

Text

Description automatically generated with medium confidence

Text

Description automatically generated

Text

Description automatically generated

Graphical user interface, text, application, chat or text message

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

A picture containing graphical user interface

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Data sebelum diupdate:

Graphical user interface, text

Description automatically generated

Setelah di update:

Text

Description automatically generated with medium confidence

Hapus akun

Graphical user interface, text, application

Description automatically generated

Data sebelum dihapus:

Graphical user interface, text

Description automatically generated

Data setelah dihapus:

Text

Description automatically generated

**Bab 5 Kesimpulan**

Program *millionaire* adalah program yang sengaja dirancang untuk kebutuhan *event* *entertainment* .Program ini akan menghasil sejumlah uang yang nominalnya dihitung berdasarkan ketentuan permainan yang berlaku.Dalam program ini para *player* akan diminta untuk mengisi data-data penting seperti *username,password,*nomor *Whatss App* dan nomor rekening.Hadiah dari program ini akan dikirim kepada para pemenang melalui rekening Bank BBB dan akan segera diinformasikan untuk bukti transfernya kepada para penerima hadiah.

Program yang dibuat terinspirasi dari adanya acara British talkshow berjudul *”Who Wants to Be a millionaire”*.Namun untuk menjadikan acara ini dapat berlangsung lebih praktis maka dibuatlah program ini yang dibuat dalam bahasa C.Dikarenakan bahasa C memiliki kelebihan pada kecepatan eksekusi programnya.

Dengan adanya program millionaire diharapkan dapat menghibur para audience dan dapat menarik kalangan public untuk semakin tertarik mengikuti acara ini.Tanpa menutup kemungkinan, jika terjadi *bug* pada program ini di masa depan.Segala kritik dan saran akan dengan senang hati kami terima demi perkembangan program ini menjadi lebih baik.