

PERMAINAN BATU GUNTING KERTAS

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



Oleh :
2602085920 – Rahel Andrea Arabella

Semester Ganjil 2022/2023

MALANG

DAFTAR ISI

DAFTAR ISI	i
BAB 1 : PENDAHULUAN	1
BAB 2 : USULAN DESAIN PROGRAM	5
BAB 3 : PROGRAM DAN PENJELASANNYA	12
BAB 4 : TAMPILAN HASIL	23
BAB 5 : KESIMPULAN	30

BAB 1 : PENDAHULUAN

Ide Project : Permainan Batu Gunting Kertas

Ide ini didapat dari keinginan penulis untuk menambah daya tarik program sehingga tidak hanya digunakan sebagai penyimpan atau pengolah data saja, melainkan juga sebagai sarana bermain batu gunting kertas melawan komputer. Permainan Batu Gunting Kertas yang dibuat ini adalah program yang bisa digunakan untuk bermain dan menyimpan data permainan. Berikut penjelasan detail mengenai kebutuhan struktur program ini :

1. Struct

Digunakan untuk menyimpan data dari file .txt sehingga mempermudah dalam penggunaan. Penggunaannya antara lain berada di fungsi sort (mengurutkan isi data), displayRecord (menampilkan isi data), deleteRecord (menghapus isi data), editRecord (mengubah/mengedit isi data : nama).

2. Gotoxy

Gotoxy merupakan fungsi untuk menempatkan kursor di tempat yang kita inginkan berdasarkan koordinat (x, y), dimana x adalah kolom dan y adalah baris.

3. Set Color

SetColor merupakan fungsi untuk membuat tulisan berwarna pada program, di program ini, penulis menggunakan setColor(10) yaitu berwarna hijau.

4. Loading Bar

Loading bar yang penulis gunakan ada 2 :

- a. Loading Bar yang bertuliskan "Loading...", yang digunakan di awal sebelum menuju fungsi-fungsi (play, displayRecord, deleteRecord, editRecord) tersebut [loadingBar()]
- b. Loading Bar yang bertuliskan "Bye-bye...", yang digunakan di awal sebelum program terhenti karena user memilih menu "Exit". [byeBar()]

Diberikan fungsi ini agar terlihat lebih bagus dan menampakkan progress. Disini kita menggunakan perulangan yang diberi fungsi Sleep, agar bar bisa berjalan bertahap sesuai dengan waktu Sleep yang diberikan. Di sini penulis menggunakan Sleep(75), di mana sudah diuji agar loading tidak terlalu lama dan juga tidak terlalu cepat.

5. Random

Dalam menentukan pilihan batu/gunting/kertas dari lawan yang merupakan sebuah komputer, penulis memilih untuk menggunakan fungsi random dalam menentukan pilihan komputer. Di fungsi ini, program akan mengeluarkan angka random yang kemudian dimodulus 100, agar hasilnya bisa tertebak yaitu dari angka 0-99.

Setelah mendapat angka random tersebut (dimasukkan variabel *n*), dibuat permisalan :

- a. Apabila nilai $n < 33$, maka komputer dianggap memilih batu / 's' / *stone*
- b. Apabila nilai $n \geq 33$ dan $n < 66$, maka komputer dianggap memilih gunting / 'c' / *scissors*
- c. Apabila nilai $n \geq 66$, maka komputer dianggap memilih kertas / 'p' / *paper*

Di akhir, fungsi ini akan mengembalikan char 's' atau 'c' atau 'p'

6. File Processing

Disini penulis menggunakan file processing :

- a. Create atau inputRecord, di sini digunakan parameter pass by reference (nama dan skor), jadi di fungsi ini kita menggunakan "a" karena akan selalu menambah data.
- b. Read atau displayRecord
 - 1) Mengurutkan isi data menggunakan fungsi sort()
 - 2) Membersihkan tampilan layar
 - 3) Membuka filenya dalam mode "r" atau read, dan menampilkan semua menggunakan for dan printf.
- c. Update atau editRecord
 - 1) Membersihkan tampilan layar
 - 2) Semua data akan ditampilkan
 - 3) User diminta menginput nama dari data yang ingin di-update
 - 4) Program akan mencocokkan dengan yang ada di array of structure lb
 - 5) Apabila ada yang cocok, maka program akan meminta inputan dari user mengenai nama yang baru, setelah menginput, akan keluar tanda apabila berhasil, dan men-display datanya kembali.
 - 6) Apabila tidak ada yang cocok dengan inputan user, maka program akan mengeluarkan tanda bahwa nama tidak ditemukan dan akan di-display datanya kembali.

- 7) User akan ditanya kembali apakah ingin meng-update lagi atau tidak, apabila menjawab 'y', maka fungsi ini akan memanggil dirinya sendiri, selain jawaban itu, program akan terhenti dan kembali ke menu awal.
- d. Delete atau deleteRecord
 - 1) Membersihkan tampilan layar
 - 2) Mengurutkan isi data menggunakan fungsi sort()
 - 3) Membuka file untuk memasukkan data-datanya ke dalam struct, setelah itu ditutup
 - 4) Membuka file kembali dalam mode "w" atau *write*
 - 5) User akan diminta menginput nama yang datanya ingin dihapus
 - 6) Program akan mencocokkan dengan yang ada di array of structure lb
 - 7) Apabila cocok, program akan lanjut dengan tidak memasukkan data struct tersebut di file
 - 8) Apabila tidak cocok, program akan memasukkan data struct tersebut di file
 - 9) User akan ditanya kembali apakah ingin menghapus lagi atau tidak, apabila menjawab 'y', maka fungsi ini akan memanggil dirinya sendiri, selain jawaban itu, program akan terhenti dan kembali ke menu awal.
- e. Sort atau mengurutkan

Di fungsi ini, program akan mengurutkan data sesuai dari skor yang tertinggi menuju skor yang terendah. Adapun sort di sini menggunakan bubble sort.

7. Tipe Data

Di program ini, penulis hanya menggunakan 3 jenis data, yaitu string untuk nama, char untuk pilihan-pilihan yang diberikan, dan int untuk skor.

8. Selection

- a. Penggunaan switch case dalam pengeksekusian menu di awal
- b. Penggunaan if dan nested if dalam penentuan beberapa permissalan

9. Media Penyimpanan

Menggunakan file .txt yang bernama LeaderBoard.txt, digunakan untuk menyimpan nama dan skor.

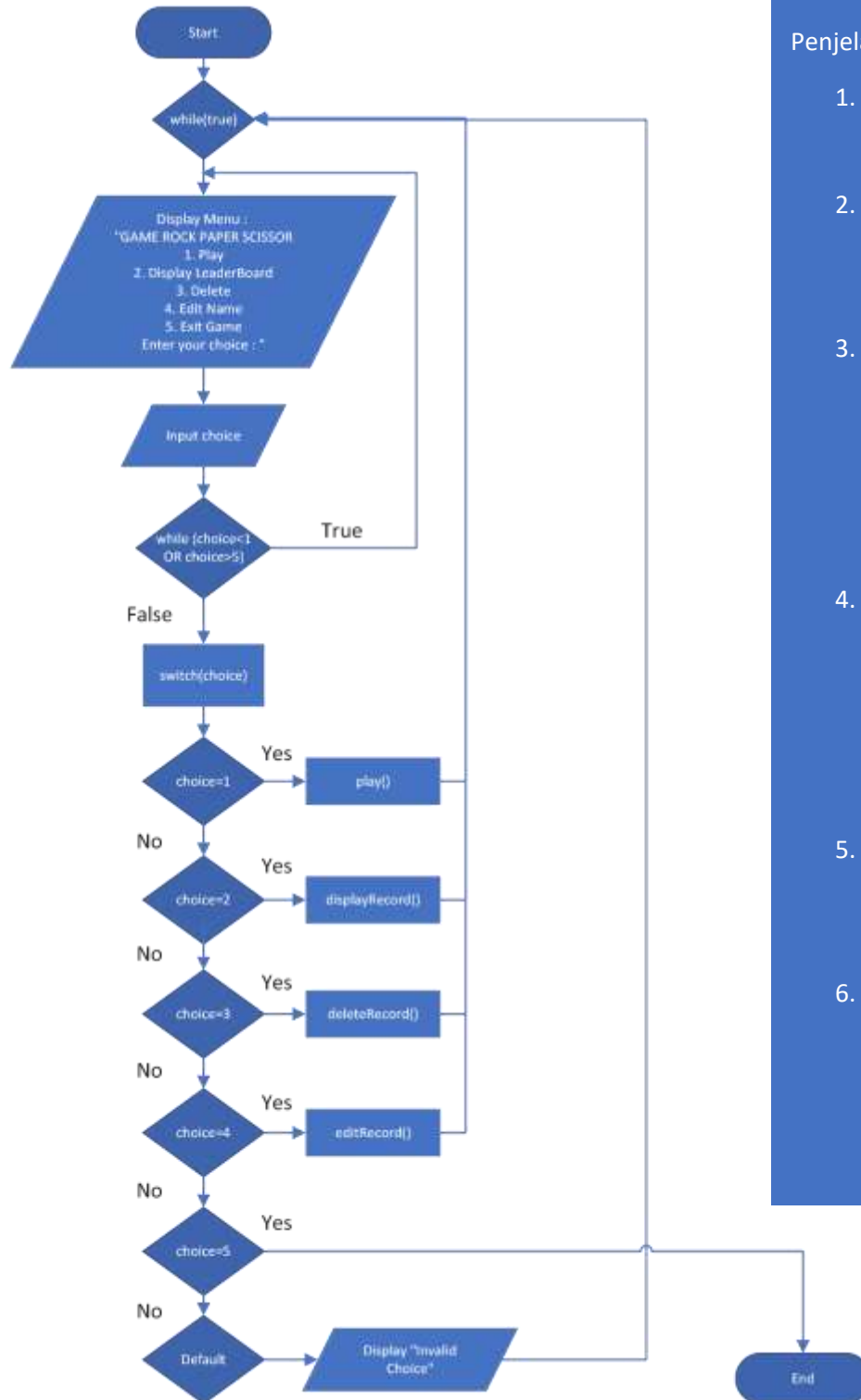
10. Perulangan

- a. Menggunakan do-while apabila terdapat suatu input yang harus divalidasi, apabila tidak tervalidasi, program akan terus berjalan hingga input sudah benar.

- b. Menggunakan while untuk mengulang menu terus menerus dan memasukkan data di file ke array of structure.
- c. Menggunakan for untuk mencocokkan data yang diinput dengan data yang ada di array of structure dan penggunaan sederhana lainnya.

BAB 2 : USULAN DESAIN PROGRAM

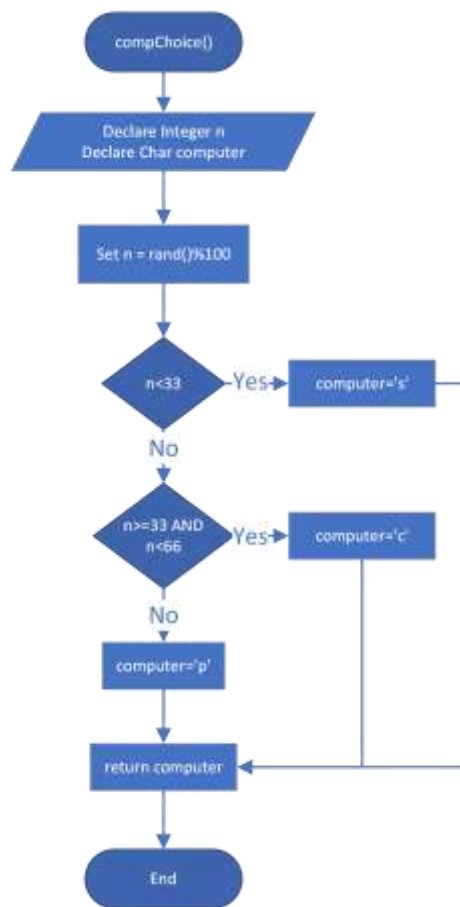
1. Desain Program Keseluruhan



Penjelasan :

1. Display Menu untuk dipilih user
2. User memilih dengan input choice (1/2/3/4/5)
3. Apabila user menginput angka < 1 atau angka > 5, maka program akan kembali ke nomor 1.
4. Apabila sudah menginput dengan benar, user akan menuju ke fungsi yang diinginkan.
5. Khusus untuk menu ke-5 / exit, program akan terhenti
6. Sedangkan menu lainnya akan membawa user pada nomor 1.

2. Fungsi compChoice → fungsi untuk mengembalikan nilai char computer



Penjelasan :

1. Program akan menggunakan fungsi random dalam int n untuk mendapat nilai n
2. Lalu akan dicocokkan dengan pengandaian yang ada
3. Char computer akan dikembalikan

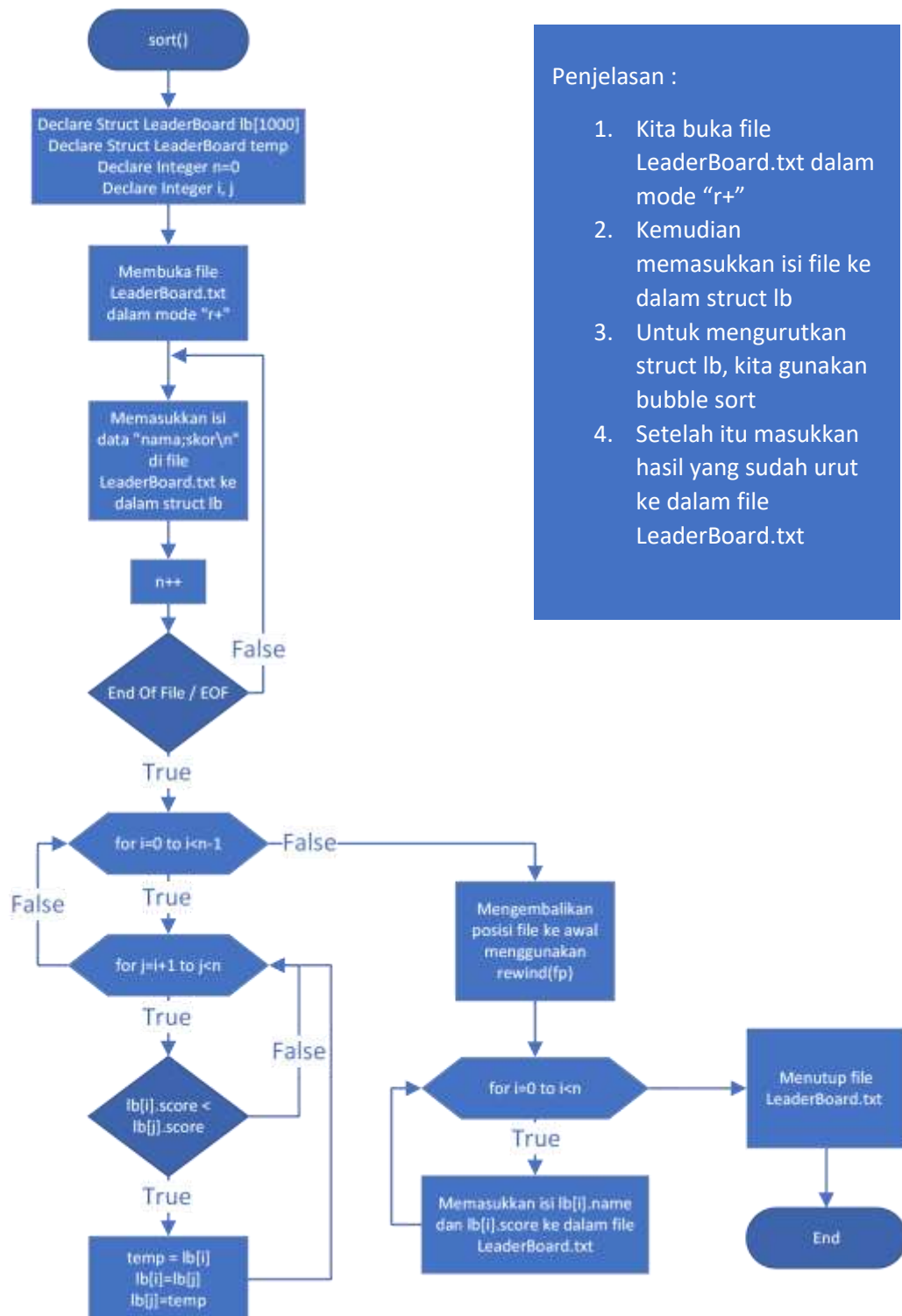
3. Fungsi inputRecord(char nama[6], int score)



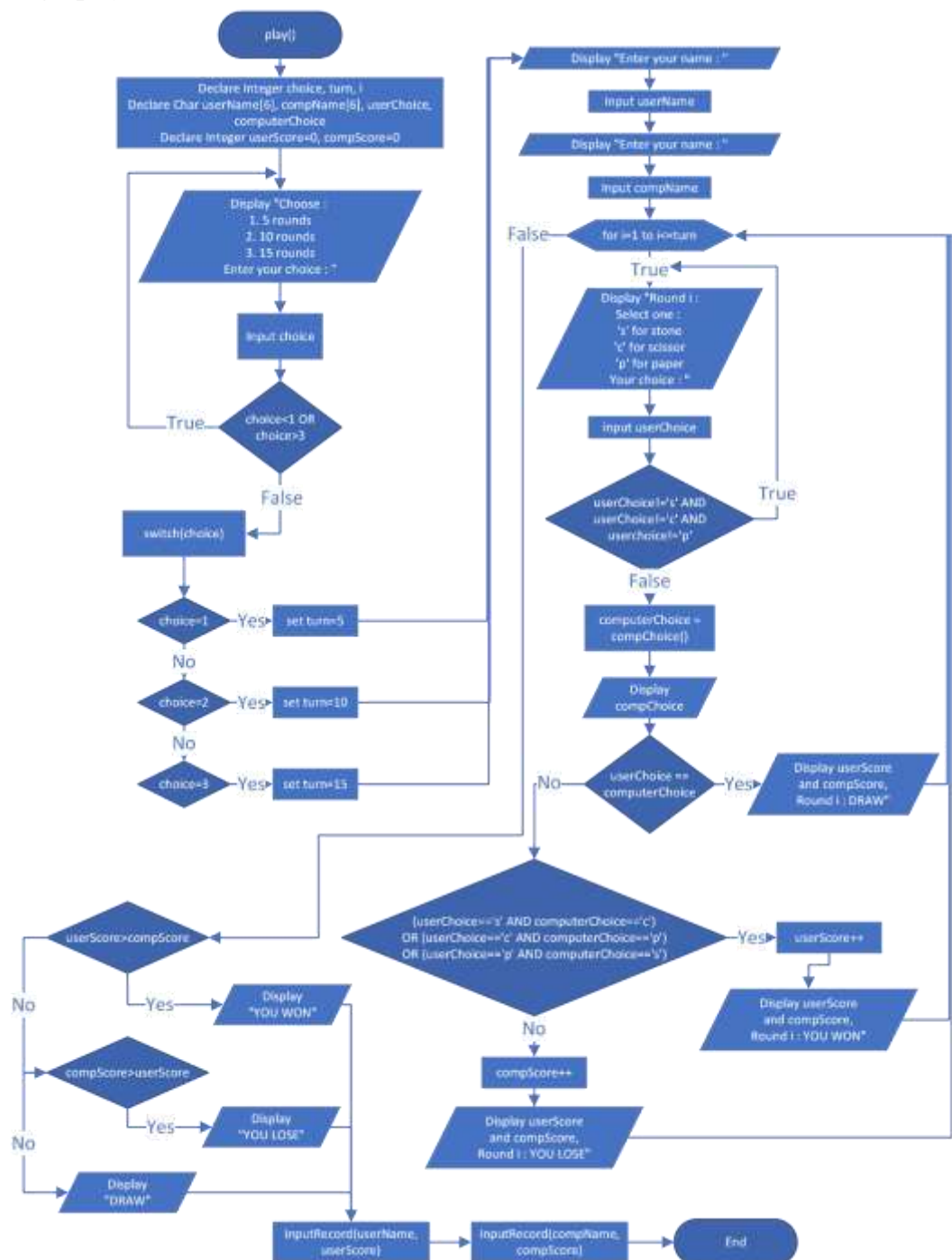
Penjelasan :

1. Merupakan fungsi untuk memasukkan data ke dalam file LeaderBoard.txt
2. Membuka file LeaderBoard.txt dalam mode "a"
3. Memasukkan data ke dalam file LeaderBoard.txt
4. Menutup file LeaderBoard.txt

4. Fungsi sort()

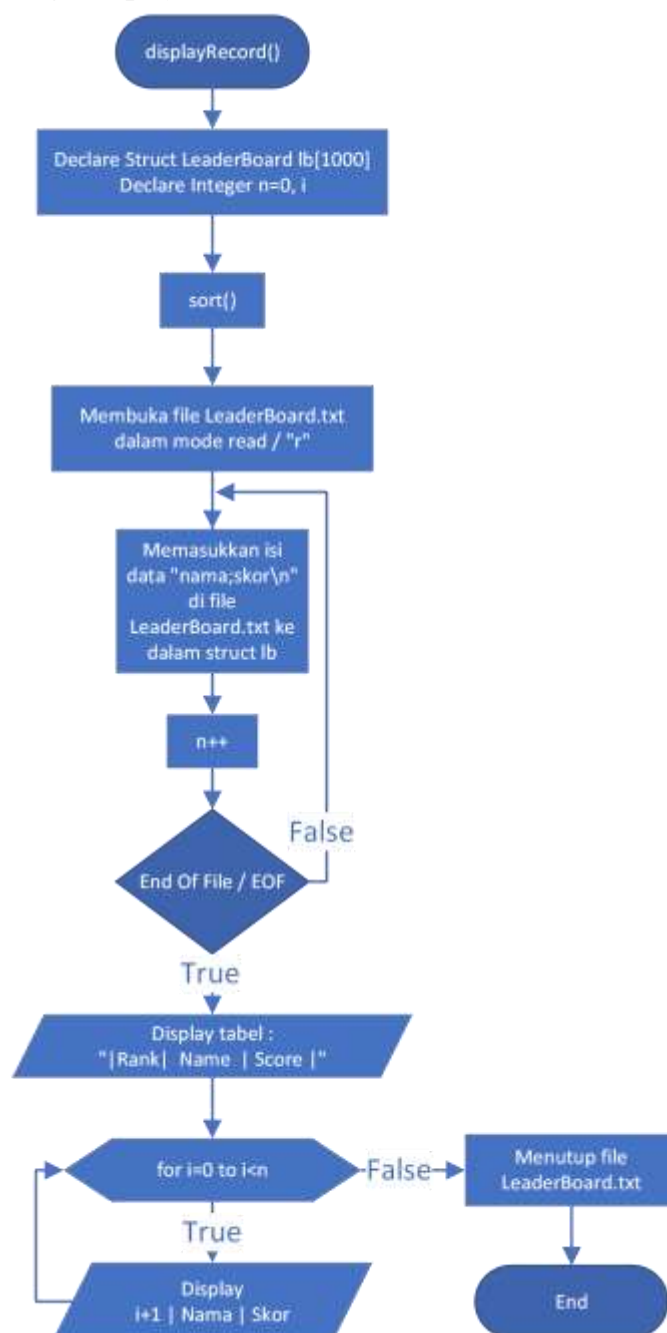


5. Fungsi play()



User diminta memilih ingin bermain 5/10/15 ronde, setelah itu user diminta menginput nama diri sendiri dan nama untuk lawan. Kemudian permainan dimulai dan user diminta menginput pilihan yang ada, setelah memilih, pilihan komputer [compChoice()] akan ditampilkan. Setelah itu di-cek siapakah yang menang dan diinput datanya menggunakan fungsi inputRecord().

6. Fungsi displayRecord()

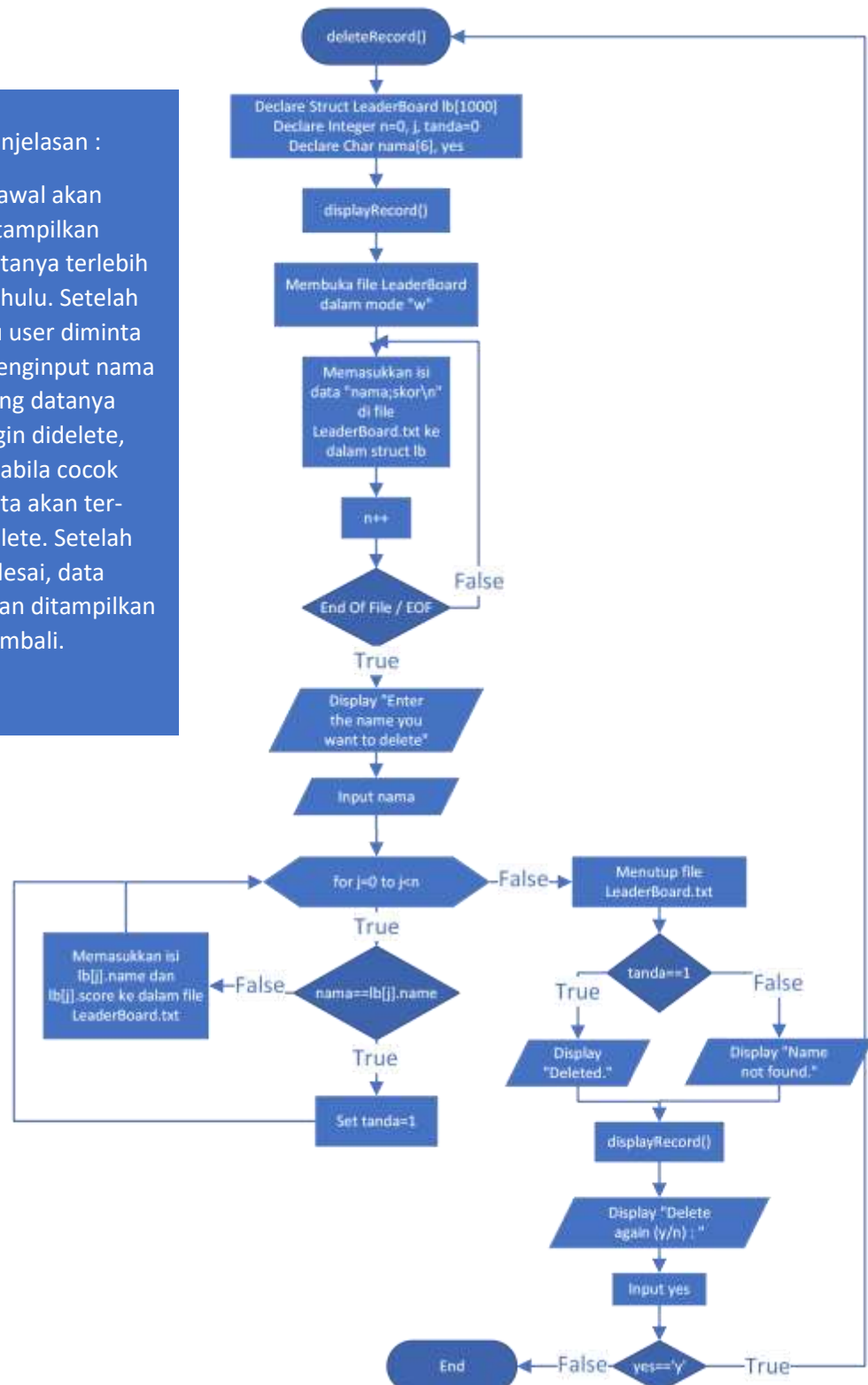


Merupakan fungsi untuk menampilkan data permainan. Sebelum ditampilkan, kita melakukan `sort()` terlebih dahulu, kemudian membuka file `LeaderBoard.txt` dalam mode "r". Kita masukkan dahulu datanya di struct `lb`. Kemudian print tabel untuk kemudian diikuti print dari data yang ada di struct `lb` dan rankingnya. Tutup file.

7. Fungsi deleteRecord()

Penjelasan :

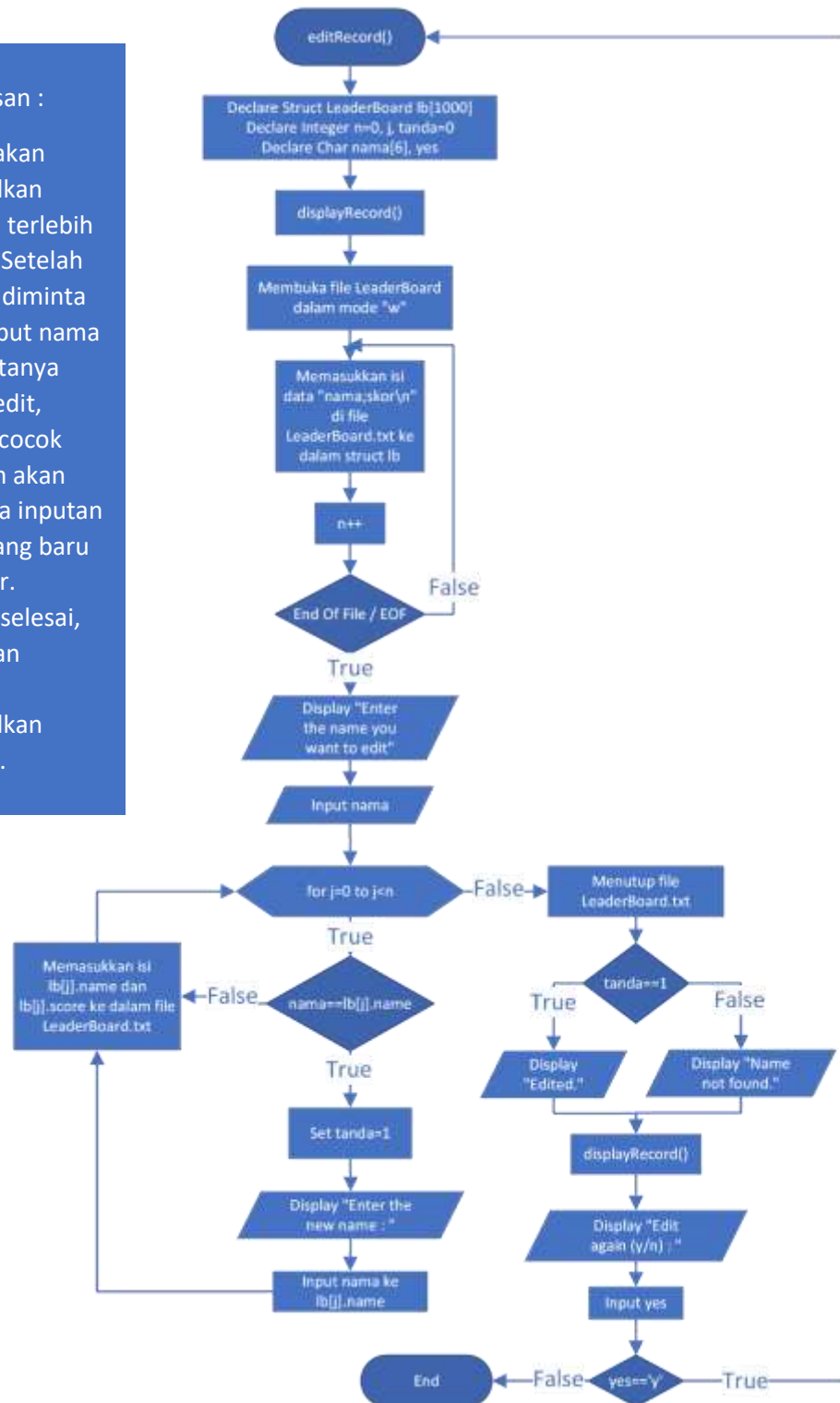
Diawal akan ditampilkan datanya terlebih dahulu. Setelah itu user diminta menginput nama yang datanya ingin didelete, apabila cocok data akan ter-delete. Setelah selesai, data akan ditampilkan kembali.



8. Fungsi editRecord()

Penjelasan :

Diawal akan ditampilkan datanya terlebih dahulu. Setelah itu user diminta menginput nama yang datanya ingin diedit, apabila cocok program akan meminta inputan nama yang baru dari user. Setelah selesai, data akan teredit ditampilkan kembali.



BAB 3 : PROGRAM DAN PENJELASANNYA

Berikut *source code* dari permainan batu gunting kertas :

```
#include <stdio.h> // untuk fungsi input output
#include <stdlib.h> // untuk operasi-operasi
#include <string.h> // untuk manipulasi string (strcmp, dll)
#include <windows.h> // untuk fungsi system()

// Menggunakan struct untuk menampung variabel" leaderboard
struct LeaderBoard{
    // string nama sepanjang maksimal 5 karakter, dibuat [6] untuk
    // menampung \0 dan meminimalisir error
    char name[6];
    // deklarasi skor menggunakan int (bilangan bulat)
    int score;
};

// fungsi untuk menempatkan kursor ditempat yang kita inginkan
// (koordinatnya)
void gotoxy(int x, int y){
    HANDLE hConsoleOutput;
    COORD dwCursorPosition;
    dwCursorPosition.X = x;
    dwCursorPosition.Y = y;
    hConsoleOutput = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleCursorPosition(hConsoleOutput, dwCursorPosition);
}

// fungsi untuk membuat tulisan berwarna pada program
void setColor(unsigned short color){
    HANDLE hCon = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(hCon, color);
}

// fungsi untuk membuat loading bar / tampilan loading
void loadingBar(){
    system("cls");
    // 0 black background
    // A green foreground
    system("color 0A");

    char a=177, b=219;

    gotoxy(50, 10); printf("Loading...");
    gotoxy(50, 12);
    for(int i=0; i<15; i++){
        printf("%c", a);
    }
    gotoxy(50, 12);
    for(int i=0; i<15; i++){
        printf("%c", b);
        Sleep(75);
    }
}
```

```

// fungsi untuk membuat tampilan loading bertulis "bye bye"
void byeBar(){
    system("cls");
    // 0 black background
    // A green foreground
    system("color 0A");

    char a=177, b=219;

    gotoxy(50, 10); printf("Bye-bye...");
    gotoxy(50, 12);
    for(int i=0; i<15; i++){
        printf("%c", a);
    }
    gotoxy(50, 12);
    for(int i=0; i<15; i++){
        printf("%c", b);
        Sleep(75);
    }
}

// fungsi untuk mendapatkan pilihan computer menggunakan random
(s/c/p)
int compChoice(){
    int n; char computer;
    // mengambil nilai random yang akan dimodulus dengan 100 agar
    mendapat nilai 0-99
    n = rand()%100;

    // batu / stone : apabila nilai n kurang dari 33
    if(n<33) computer='s';
    // gunting / scissor : apabila nilai n>=33 dan n<66
    else if(n>=33 && n<66) computer='c';
    // kertas / paper : apabila nilai n lebih dari 66
    else computer='p';

    // akan mengembalikan nilai char computer berupa 's'/'c'/'p'
    return computer;
}

// fungsi untuk memasukkan record ke dalam file LeaderBoard.txt
void inputRecord(char nama[6], int score){
    // kita akan buka dan tambahkan isinya menggunakan "a" di
    LeaderBoard.txt
    FILE *fp;
    fp = fopen("LeaderBoard.txt", "a");
    // kita print datanya menggunakan format "nama;score\n" di
    dalam filenya
    fprintf(fp, "%.5s;%d\n", nama, score);
    // tutup file
    fclose(fp);
}

// fungsi untuk mengurutkan skor dari yang tertinggi di dalam file
void sort(){
    // menggunakan array of structure untuk menampung nilai-nilai

```

```

    struct LeaderBoard lb[1000];
    // digunakan untuk menampung nilai yang digunakan sementara
    struct LeaderBoard temp;
    // membuka file dan membacanya
    FILE *fp;
    fp = fopen("LeaderBoard.txt", "r+");
    int n=0;
    // memasukkan nilai di file ke dalam array of structure lb
    dengan menghitung jumlahnya melalui variable n
    while(fscanf(fp, "%[^';'];%d\n", lb[n].name,
&lb[n].score) != EOF) {
        n++;
    }
    // menggunakan bubble sort, kita akan mengurutkan dari yang
    tertinggi
    for(int i=0; i<n-1; i++){
        for(int j=i+1; j<n; j++){
            // apabila data di atas lebih kecil dari yang dibawah,
            maka kedua data tersebut akan ditukar posisinya
            if(lb[i].score<lb[j].score){
                temp=lb[i];
                lb[i]=lb[j];
                lb[j]=temp;
            }
        }
    }
    // berfungsi untuk mengembalikan posisi FILE ke awal
    rewind(fp);
    // kita akan memasukkan nilai yang ada di array of structure lb
    tadi ke FILE LeaderBoard.txt menggunakan fprintf
    for(int i=0; i<n; i++){
        fprintf(fp, "%s;%d\n", lb[i].name, lb[i].score);
    }
    // menutup file
    fclose(fp);
}

// fungsi untuk menampilkan record yang sudah dalam keadaan urut
sesuai score
void displayRecord(){
    // akan membersihkan tampilan layar
    system("cls");
    // menggunakan array of structure lb untuk menampung nilai-
    nilai di file
    struct LeaderBoard lb[1000];
    // sebelum masuk ke tampilan, kita perlu sort terlebih dahulu
    isi filenya
    sort();
    // membuka file dalam mode "r" yaitu read atau membaca file
    saja
    FILE *fp;
    fp = fopen("LeaderBoard.txt", "r");
    int n=0;
    // memasukkan nilai di file ke dalam array of structure lb
    dengan menghitung jumlahnya melalui variable n
    while(fscanf(fp, "%[^';'];%d\n", lb[n].name,
&lb[n].score) != EOF) {

```



```

        n++;
    }
    // menggunakan gotoxy agar tampilan berada di tengah
    gotoxy(45, 7); printf("=====\n");
    gotoxy(45, 8); printf("|Rank|   Name   | Score |\n");
    gotoxy(45, 9); printf("=====\n");
    int i;
    // akan menampilkan rank, name dan score dari file
LeaderBoard.txt
    for(i=0; i<n; i++){
        // untuk ranknya menggunakan (i+1, dimulai dari 1)
        gotoxy(45, 10+i); printf("|%4d|    %5s|%7d|\n", i+1,
lb[i].name, lb[i].score);
    }
    gotoxy(45, 10+i); printf("=====\n");
    gotoxy(45, 12+i); system("pause");
    // menutup file
    fclose(fp);
}

// fungsi untuk menghapus record : diawal akan ditampilkan datanya
(secara urut), kemudian user akan diminta menginput nama yang
datanya ingin dihapus
// program akan mencocokkan dan akan menghapus, lalu data akan
ditampilkan kembali
void deleteRecord(){
    // akan membersihkan tampilan layar
    system("cls");
    // menggunakan array of structure lb untuk menampung nilai-
nilai di file
    struct LeaderBoard lb[1000];
    // sebelum masuk ke tampilan, kita perlu sort terlebih dahulu
isi filenya
    sort();
    // membuka file dalam mode baca "r"/read
    FILE *fp;
    fp = fopen("LeaderBoard.txt", "r");
    int n=0;
    // memasukkan nilai di file ke dalam array of structure lb
dengan menghitung jumlahnya melalui variable n
    while(fscanf(fp, "%[^';'];%d\n", lb[n].name,
&lb[n].score)!=EOF){
        n++;
    }
    // menggunakan fitur gotoxy agar tampilan bisa di tengah sesuai
koordinat yang diberikan
    gotoxy(45, 7); printf("=====\n");
    gotoxy(45, 8); printf("|Rank|   Name   | Score |\n");
    gotoxy(45, 9); printf("=====\n");
    int i;
    for(i=0; i<n; i++){
        // akan menampilkan seluruh data di dalam file
LeaderBoard.txt
        gotoxy(45, 10+i); printf("|%4d|    %5s|%7d|\n", i+1,
lb[i].name, lb[i].score);
    }
    gotoxy(45, 10+i); printf("=====\n");

```

```

    char nama[6]; int tanda=0;
    // kita tutup dahulu filenya, lalu kita buka dalam mode
write/"w"
    fclose(fp);
    fp = fopen("LeaderBoard.txt", "w");
    // user akan diminta memasukkan nama yang datanya ingin dihapus
gotoxy(40, 12+i); printf("Enter the name you want to delete :
"); scanf(" %[^\\n]", nama); getchar();
    // program akan mencocokkan nama yang dimasukkan user dengan
nama yang ada di array of structure lb
    for(int j=0; j<n; j++){
        // apabila ada yang cocok, maka program akan berlanjut
dengan tidak meng-print data tersebut dan akan memberi nilai
tanda=1
        if(strcmp(nama, lb[j].name)==0){
            tanda=1;
            continue;
            // apabila tidak cocok, maka program akan memasukkan nilai-
nilai itu ke dalam file lagi
        }else{
            fprintf(fp, "%.5s;%d\\n", lb[j].name, lb[j].score);
        }
    }
    // menutup file karena sudah selesai digunakan
fclose(fp);
    // apabila tanda==1, maka itu berarti ada data yang cocok dan
berhasil dihapus
    if(tanda==1){
        gotoxy(45, 14+i); printf("Deleted.");
        gotoxy(45, 16+i); system("pause"); // system("pause")
digunakan menahan layar
        displayRecord(); // akan men-display data
    }
    // apabila tanda!=1, maka tidak ada kecocokan antara nama yang
diinput user dengan yang ada di file, sehingga mengeluarkan "Name
not found"
    else{
        gotoxy(45, 14+i); printf("Name not found.");
        gotoxy(45, 16+i); system("pause"); // system("pause")
digunakan menahan layar
        displayRecord(); // akan men-display data
    }
    system("cls"); // membersihkan tampilan layar
    char yes;
    // program meminta inputan user apakah ingin menghapus data
lagi atau tidak
    gotoxy(45, 10); printf("Delete again (y/n) : "); scanf(" %c",
&yes); getchar();
    // apabila user menjawab 'y', maka fungsi akan memanggil diri
sendiri (deleteRecord) (Recursive)
    // apabila menjawab selain 'y', maka fungsi ini akan terhenti
    if(yes=='y'){
        deleteRecord();
    }
}

// fungsi untuk mengedit nama di dalam data : diawal akan
ditampilkan datanya (secara urut), kemudian user akan diminta
menginput nama yang datanya ingin diedit (mengedit namanya saja)

```

// program akan mencocokkan dan ketika cocok, program akan meminta inputan user mengenai nama yang baru, dan setelah itu nama yang baru akan dimasukkan ke dalam file dan ditampilkan kembali di akhir fungsi

```
void editRecord(){
    system("cls"); // membersihkan tampilan layar
    struct LeaderBoard lb[1000]; // array of structure lb
    sort(); // mengurutkan data di dalam file terlebih dahulu
    // membuka file dalam mode read / "r"
    FILE *fp;
    fp = fopen("LeaderBoard.txt", "r");
    int n=0;
    // memasukkan nilai di file ke dalam array of structure lb
    dengan menghitung jumlahnya melalui variable n
    while(fscanf(fp, "%[^';];%d\n", lb[n].name,
    &lb[n].score)!=EOF){
        n++;
    }
    // menggunakan fitur gotoxy agar tampilan bisa di tengah sesuai
    koordinat yang diberikan
    gotoxy(45, 7); printf("=====\n");
    gotoxy(45, 8); printf("|Rank|   Name   | Score |\n");
    gotoxy(45, 9); printf("=====\n");
    int i;
    for(i=0; i<n; i++){
        // akan menampilkan seluruh data di dalam file
        LeaderBoard.txt
        gotoxy(45, 10+i); printf("|%4d|    %5s|%7d|\n", i+1,
        lb[i].name, lb[i].score);
    }
    gotoxy(45, 10+i); printf("=====\n");
    char nama[6]; int tanda=0;
    // kita tutup dahulu filenya, lalu kita buka dalam mode
    write/"w"
    fclose(fp);
    fp = fopen("LeaderBoard.txt", "w");
    gotoxy(40, 12+i); printf("Enter the name you want to edit : ");
    scanf(" %[^\\n]", nama); getchar();
    // program akan mencocokkan nama yang dimasukkan user dengan
    nama yang ada di array of structure lb
    for(int j=0; j<n; j++){
        // apabila ada yang cocok, maka program akan meminta
        inputan nama yang baru dan akan dimasukkan ke dalam file
        LeaderBoard.txt
        if(strcmp(nama, lb[j].name)==0){
            tanda=1;
            gotoxy(40, 14+i); printf("Enter the new name : ");
            scanf(" %[^\\n]", lb[j].name); getchar();
            fprintf(fp, "%.5s;%d\n", lb[j].name, lb[j].score);
            // apabila tidak cocok, maka program akan memasukkan nilai-
            nilai itu ke dalam file lagi
        }else{
            fprintf(fp, "%.5s;%d\n", lb[j].name, lb[j].score);
        }
    }
    }// menutup file karena sudah selesai digunakan
    fclose(fp);
```

```

        // apabila tanda==1, maka itu berarti ada data yang cocok dan
        berhasil diedit
        if(tanda==1){
            gotoxy(45, 16+i); printf("Edited.");
            gotoxy(45, 18+i); system("pause");// system("pause")
            digunakan menahan layar
            displayRecord(); // akan men-display data
        }// apabila tanda!=1, maka tidak ada kecocokan antara nama yang
        diinput user dengan yang ada di file, sehingga mengeluarkan "Name
        not found"
        else{
            gotoxy(45, 14+i); printf("Name not found.");
            gotoxy(45, 16+i); system("pause");// system("pause")
            digunakan menahan layar
            displayRecord(); // akan men-display data
        }
        system("cls"); // membersihkan tampilan layar
        char yes;
        // program meminta inputan user apakah ingin mengedit data lagi
        atau tidak
        gotoxy(45, 10); printf("Edit again (y/n) : "); scanf(" %c",
        &yes); getchar();
        // apabila user menjawab 'y', maka fungsi akan memanggil diri
        sendiri (editRecord) (Recursive)
        // apabila menjawab selain 'y', maka fungsi ini akan berhenti
        if(yes=='y'){
            editRecord();
        }
    }

    /* fungsi untuk memainkan gamenya : di awal user akan diberi
    pilihan ingin main berapa ronde (5/10/15)
    Setelah memilih ingin berapa ronde, user akan memasukkan nama diri
    dan nama lawan
    Game dimulai, user diminta menginput
    's':stone/'c':scissor/'p':paper
    Sesuai dengan aturan batu gunting kertas, user akan bertambah 1
    poin apabila berhasil menang, dan sebaliknya
    diakhir ronde akan muncul tulisan besar bertuliskan "YOU WON"
    apabila skor user lebih tinggi daripada lawan
    "YOU LOSE" apabila skor user lebih rendah dari skor lawan dan
    "DRAW" apabila skor user dan lawan sama
    data nama dan skor akan dimasukkan dalam file LeaderBoard.txt */
    void play(){
        system("cls"); // membersihkan tampilan layar
        int choice;
        // ketika inputan user tidak sesuai, maka program akan
        mengulang meminta user menginput kembali
        do{
            // menggunakan fitur gotoxy agar tampilan bisa di tengah
            sesuai koordinat yang diberikan
            gotoxy(50, 10); printf("Choose :\n");
            gotoxy(50, 11); printf("1. 5 rounds\n");
            gotoxy(50, 12); printf("2. 10 rounds\n");
            gotoxy(50, 13); printf("3. 15 rounds\n");
            gotoxy(50, 14); printf("Enter your choice : "); scanf("
            %d", &choice);

```

```

}while(choice<1 || choice>3);
int turn;
if(choice==1) turn=5; // akan main sebanyak 5 ronde
else if(choice==2) turn=10; // akan main sebanyak 10 ronde
else if(choice==3) turn=15; // akan main sebanyak 15 ronde
char userName[6], compName[6];
// meminta inputan nama user
system("cls");
gotoxy(25, 10); printf("Enter your name (max 5 character) : ");
scanf(" %[^\\n]", userName); getchar();
// meminta inputan nama lawan
system("cls");
gotoxy(25, 10); printf("Enter your opponent's name (max 5
character) : "); scanf(" %[^\\n]", compName); getchar();

int userScore = 0, compScore = 0;

for(int i=1; i<=turn; i++){
    char userChoice;
    // akan mengeluarkan pilihan untuk dipilih user, apabila
    user salah menginput, maka akan diminta inputan kembali
    do{
        system("cls");
        gotoxy(50, 6); printf("Round %d :\\n", i);
        gotoxy(50, 8); printf("=====");
        gotoxy(50, 9); printf("|Select one :   |\\n");
        gotoxy(50, 10); printf("|'s' for stone   |\\n");
        gotoxy(50, 11); printf("|'c' for scissor|\\n");
        gotoxy(50, 12); printf("|'p' for paper   |\\n");
        gotoxy(50, 13); printf("=====");
        gotoxy(50, 15); printf("Your choice : "); scanf(" %c",
&userChoice); getchar();
    }while(userChoice != 's' && userChoice != 'c' && userChoice
!= 'p');

    // mengambil nilai char computerChoice / pilihan komputer
    berdasarkan fungsi random di compChoice()
    char computerChoice = compChoice();
    Sleep(1000); // timer agar tidak langsung keluar yang
    selanjutnya
    gotoxy(50, 16); printf("%.5s choice : %c\\n", compName,
computerChoice);
    Sleep(1000);
    // apabila pilihannya sama antara user dengan komputer,
    maka skor tetap dan akan mengeluarkan "DRAW"
    if(userChoice == computerChoice){
        gotoxy(50, 18); printf("Your score : %d\\n", userScore);
        gotoxy(50, 19); printf("%.5s score : %d\\n", compName,
compScore);
        gotoxy(50, 21); printf("=====\\n");
        gotoxy(50, 22); printf("= Round %d : DRAW =\\n", i);
        gotoxy(50, 23); printf("=====\\n");
        gotoxy(50, 25); system("pause");
        // apabila pilihan user menang dari pilihan komputer, maka
        skor user akan bertambah dan akan mengeluarkan "YOU WON"

```

[illegible]


```
        // berdasarkan yang user pilih, akan menuju ke fungsi yang  
berkaitan
```

```
        switch(choice){  
            case 1:  
                loadingBar();  
                play();  
                break;  
  
            case 2:  
                loadingBar();  
                displayRecord();  
                break;  
  
            case 3:  
                loadingBar();  
                deleteRecord();  
                break;  
  
            case 4:  
                loadingBar();  
                editRecord();  
                break;  
  
            case 5:  
                // keluar dari program, terhenti dari while  
                byeBar();  
                exit(0);  
                break;  
  
            default:  
                printf("\nInvalid Choice\n");  
                system("pause");  
        }  
    }  
}
```


BAB 4 : TAMPILAN HASIL

1. Menu awal



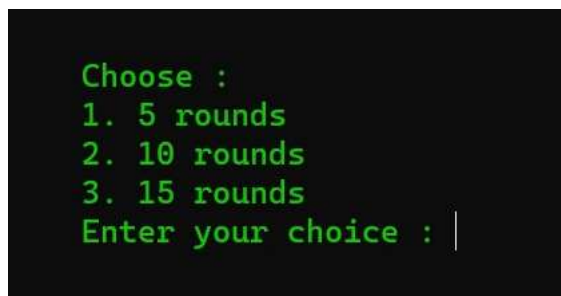
Di awal setiap program akan selalu ditampilkan menu dari program permainan batu gunting kertas ini, setelah melakukan menu 1/2/3/4, user juga akan ditampilkan tampilan menu seperti ini.

2. Loading Bar



Apabila user memilih menu 1/2/3/4, sebelum menuju fungsi yang lebih spesifik, akan ditampilkan loading bar sebagai tanda progress.

3. Play



User akan diminta untuk memilih ingin memainkan permainan ini berapa kali, di mana terdapat 3 pilihan yang bisa dipilih user.

```
Enter your name (max 5 character) : Rahel|
```

User akan diminta untuk menginput nama yang diinginkan.

```
Enter your opponent's name (max 5 character) : Lawan|
```

User akan diminta untuk menginput nama lawan yang diinginkan.

```
Round 1 :
```

```
=====
|Select one : |
|'s' for stone |
|'c' for scissor|
|'p' for paper  |
|              |
=====
```

```
Your choice : |
```

Ini merupakan tampilan sebelum user memilih. Disediakan 3 pilihan untuk user pilih, apabila user salah menginput, program akan menampilkan ulang dan user akan diminta inputan hingga valid.

```
Round 1 :  
  
=====|  
|Select one : |  
|'s' for stone |  
|'c' for scissor|  
|'p' for paper |  
=====|  
  
Your choice : s  
Lawan choice : c  
  
Your score : 1  
Lawan score : 0  
  
=====|  
= Round 1 : YOU WON =  
=====|  
  
Press any key to continue . . . |
```

Setelah user menginput, tidak lama akan muncul pilihan dari lawan(komputer). Di sini program akan mengecek siapakah yang menang dan siapa yang kalah. Yang menang akan mendapat poin yang akan ditunjukkan di bawah. Setelah itu akan ada tulisan yang merepresentasikan status game anda apakah di ronde ini menang atau kalah.

```
YOU WON  
  
Press any key to continue . . . |
```

Berikut adalah tampilan akhir setelah menyelesaikan semua ronde dan skor user unggul dari lawan (komputer).



Berikut adalah tampilan akhir setelah menyelesaikan semua ronde dan skor user sama dengan skor lawan (komputer).



Berikut adalah tampilan akhir setelah menyelesaikan semua ronde dan skor lawan (komputer) unggul dari skor user.

4. Display

=====			
Rank	Name	Score	
=====			
1	Steve	9	
2	Rahel	3	
3	Hello	2	
4	Sapan	2	
5	Rahel	2	
6	Lawan	1	
7	Nancy	1	
=====			
Press any key to continue . . .			

Berikut merupakan tampilan dari data. Di mana data yang ditampilkan tersebut sudah urut berdasarkan skornya.

5. Delete

```
=====
|Rank| Name | Score |
=====
| 1| Steve| 9|
| 2| Rahel| 2|
| 3| Arch| 2|
| 4| Nancy| 1|
| 5| Toni| 1|
=====

Enter the name you want to delete : Arch

Deleted.

Press any key to continue . . .
```

Di fungsi ini, program akan menampilkan datanya, dan user akan diminta untuk menginput nama yang datanya ingin dihapus, apabila inputan user dengan data yang ada di layar sama, maka data akan di delete dan akan mengeluarkan tulisan “Deleted.”

```
=====
|Rank| Name | Score |
=====
| 1| Steve| 9|
| 2| Rahel| 2|
| 3| Nancy| 1|
| 4| Toni| 1|
=====

Enter the name you want to delete : cobab

Name not found.

Press any key to continue . . . |
```

Sedangkan apabila tidak cocok, maka akan keluar tulisan “Name not found.”

```
=====
|Rank| Name | Score |
=====
| 1| Steve| 9|
| 2| Hello| 2|
| 3| Sapan| 2|
| 4| Lawan| 1|
| 5| Nancy| 1|
=====

Press any key to continue . . . |
```

Setelah proses delete selesai, program akan menampilkan kembali data yang baru.

```
Delete again (y/n) : |
```

Setelah semua tahapan selesai, user akan diminta inputan berdasarkan apakah ingin menghapus data lagi atau tidak.

6. Update

```
=====
|Rank| Name | Score |
=====
| 1| Steve| 9|
| 2| Rahel| 2|
| 3| Nancy| 1|
| 4| Toni| 1|
=====

Enter the name you want to edit : Toni

Enter the new name : Tony

Edited.

Press any key to continue . . . |
```

Berikut merupakan tampilan update apabila inputan user sesuai dengan data yang ada. Data akan berhasil diedit dan menunjukkan tulisan “Edited.”

```
=====
|Rank| Name | Score |
=====
| 1| Steve| 9|
| 2| Hello| 2|
| 3| Rahel| 2|
| 4| Lawan| 1|
| 5| Nancy| 1|
=====

Press any key to continue . . . |
```

Ini merupakan tampilan setelah berhasil ataupun gagal edit, bisa dilihat awalnya ada nama “Sapan” tetapi sudah diedit dan diubah menjadi “Rahel” di data.

```
=====
|Rank|  Name  | Score |
=====
|  1|  Steve|    9|
|  2|  Rahe1|    2|
|  3|  Nancy|    1|
|  4|   Tony|    1|
=====

Enter the name you want to edit : Tonnn

Name not found.

Press any key to continue . . . |
```

Berikut merupakan contoh yang gagal, dimana inputan user tidak sesuai dengan yang ada di data, maka akan mengeluarkan tulisan “Name not found.”

```
Edit again (y/n) : |
```

Setelah menyelesaikan semua tahapan, user akan diminta inputan berdasarkan apakah ingin mengedit lagi atau tidak.

7. Exit

```
Bye-bye...
████████████████████████████████████████████████████████████████████████████████
```

Apabila user memilih menu ke-5 / exit, maka user akan diberikan tampilan loading bar tetapi dengan tulisan “Bye-bye”, lalu program akan berhenti dan selesai.

BAB 5 : KESIMPULAN

Berdasarkan penjabaran di atas, dapat disimpulkan program permainan batu gunting kertas ini membutuhkan :

- a. 1 struct LeaderBoard
- b. 1 file : LeaderBoard.txt
- c. 9 fungsi void :
 - 1) Void gotoxy : untuk menempatkan kursor di koordinat yang diinginkan
 - 2) Void setColor : untuk mengubah warna teks
 - 3) Void loadingBar dan byeBar : untuk menampilkan progress di layar
 - 4) Void inputRecord(char nama[6], int score) : untuk memasukkan data ke file LeaderBoard.txt
 - 5) Void sort() : untuk mengurutkan data dan memasukkannya ke file LeaderBoard.txt
 - 6) Void displayRecord : untuk menampilkan seluruh data di LeaderBoard.txt
 - 7) Void deleteRecord : untuk menghapus data di LeaderBoard.txt
 - 8) Void editRecord : untuk mengedit nama di sebuah data di LeaderBoard.txt
 - 9) Void play : untuk mulai permainan batu gunting kertas yang kemudian akan memanggil inputRecord untuk memasukkan data permainan
- d. 1 fungsi int :
 - 1) Int compChoice : berguna untuk mengembalikan nilai char computer (pilihan komputer dalam permainan batu gunting kertas)
- e. 1 driver code : int main() : untuk memberikan alur dalam program ini.

Program ini akan mampu memberikan pengalaman bermain batu gunting kertas yang berbeda pada user, adanya LeaderBoard juga memberikan kesan tertentu kepada user seperti layaknya game pada umumnya.