

# TIC-TAC-TOE

**Tugas Besar Dasar - Dasar Pemrograman**

*Dikumpulkan untuk memenuhi sebagian persyaratan kelulusan mata kuliah Dasar-Dasar  
pemrograman*



**Kelompok A8 :**

**Difa Moch Fadillah – 201524004**

**Muhammad Azhar Alauddin – 201524013**

**Program Studi D4 Teknik Informatika  
Jurusan Teknik Komputer dan Informatika  
Politeknik Negeri Bandung  
2020/2021**

# BAB 1

## “Deskripsi Aplikasi”

### 1. Penjelasan Umum



Gambar 1 tic-tac-toe game

Game ini merupakan permainan klasik yang bernama “*Tic-tac-toe*” yang terdiri dari 2 komponen utama yaitu *board* (papan permainan) dan juga *player*, baik itu *bot* maupun *player* yang ditandai dengan tanda ‘X’ dan ‘O’. Dalam permainan ini, *board* yang bisa dimainkan berbentuk *grid* dengan ukuran 3x3, 5x5, dan 7x7. *Player 1* dapat menentukan akan melawan *bot* atau *player 2*. Jika *player 1* memilih *bot* sebagai lawan, maka *player 1* diwajibkan untuk memilih tingkat kesulitan dari *bot* itu sendiri. Setelah itu, *player 1* harus meng-inputkan *username*. Dan jika *player 1* memilih *player 2* sebagai lawan, maka *player 1* dan *player 2* harus meng-inputkan *username* masing-masing. Setiap *turn*/giliran, baik itu dari *player 1* maupun *player 2*, terdapat timer yang menunjukkan batas waktu untuk memindahkan tanda ‘X’ atau ‘O’ selama 10 detik. Jika *player* menandai dengan lebih dari 10 detik, maka giliran akan berpindah ke *player* lainnya atau ke *bot*. Selain itu juga, terdapat fitur *highscore* yang menyimpan jumlah kemenangan setiap permainan melawan *bot*, sehingga setelah menang melawan *bot*, *player* bisa melihat riwayat kemenangannya pada fitur *highscore* ini. Adapun format isi dari *highscore* ini adalah nomor sebagai jumlah dari riwayat kemenangan, time sebagai waktu mainnya, size sebagai ukuran board yang dimainkan, difficulty sebagai level kesulitan dari komputer dan winner sebagai nama dari *player* yang menang melawan *bot*.

“*Tic-tac-toe*” ini juga dilengkapi dengan menu *help* yang berfungsi untuk menampilkan aturan dan cara bermain dari *game* ini sehingga *player* yang baru pertama kali bermain *game* ini tidak kebingungan untuk cara bermainnya. Selain itu juga

terdapat menu *color theme* yang berisi pilihan warna dari game ini yang dapat dipilih oleh *player* sesuai dengan keinginan. Dengan adanya fitur ini, akan menambah desain tampilan dari *game* ini semakin menarik dan juga lebih atraktif. Adapun warna yang dipilih merupakan warna untuk tampilan text dan juga background dari permainan. *Player* juga bisa melihat identitas *developer* atau *creator* dari *game* ini dengan memilih menu *credits* dan juga bisa langsung keluar dari permainan dengan memilih menu *exit*. Dan setiap *player* yang sudah menuju ke salah satu pilihan di main menu, *player* juga bisa kembali dengan menekan tombol pada keyboard sesuai apa yang diperintahkan.

## 2. Skenario

### 2.1. Tampilan Main Menu

Pada saat *player* membuka permainan, maka tampilan yang pertama muncul adalah tampilan main menu. Pada tampilan pertama ini, terdapat beberapa pilihan yang harus dipilih oleh *player* sesuai dengan keinginan. Adapun rincian dari pilihan-pilihan yang terdapat di main menu sebagai berikut :

- a. *start-game* untuk memulai permainan
- b. *help* untuk menampilkan aturan dan cara bermain
- c. *color-theme* untuk memilih warna tampilan permainan ini
- d. *credits* untuk menampilkan identitas developer atau creator dari permainan ini
- e. *highscore* untuk menampilkan riwayat kemenangan *player* melawan bot
- f. *exit* untuk keluar dari permainan



Gambar 2 sketsa tampilan main menu

### 2.2. Tampilan Memilih Ukuran Grid

Setelah *player* memilih *start-game* pada main menu, maka *player* akan ditampilkan 3 pilihan ukuran grid yaitu 3x3, 5x5, dan 7x7. Adapun sketsa dari tampilan ini sebagai berikut



Gambar 3 sketsa tampilan memilih ukuran grid

### 2.3. Tampilan Memilih Lawan

Setelah *player* memilih ukuran grid, maka *player* akan ditampilkan pilihan lawan apakah akan melawan bot atau *player* 2, Adapun sketsa dari tampilan ini sebagai berikut



Gambar 4 sketsa tampilan memilih lawan

### 2.4. Tampilan Memilih *Level* Kesulitan Komputer

Setelah *player* memilih lawan, maka *player* akan ditampilkan tampilan pemilihan *level* kesulitan dari komputer. *Level* kesulitan dari bot nya itu sendiri terdapat 3 yaitu, *easy*, *normal*, dan *hard*. Adapun sketsa dari tampilan ini sebagai berikut



Gambar 5 sketsa tampilan memilih *level* komputer

## 2.5. Tampilan Meng-inputkan Username *Player*

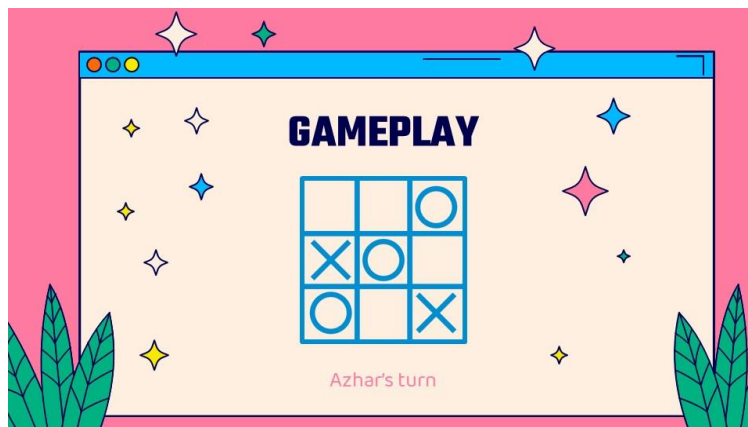
Setelah *player* memilih *level* kesulitan komputer, maka *player* akan ditampilkan tampilan pengisian nama *player*. Jika melawan bot, maka hanya *player* 1 yang harus menginputkan *username* dan jika melawan *player* 2, maka *player* 1 dan *player* 2 harus menginputkan *username*. Setelah masing-masing *player* menginputkan *username*, maka ditampilkan juga tanda dari masing-masing *player/bot*. Adapun sketsa dari tampilan ini sebagai berikut



Gambar 6 sketsa tampilan *input* nama *player*

## 2.7. Tampilan Utama Permainan (Cara bermain Tic-tac-toe)

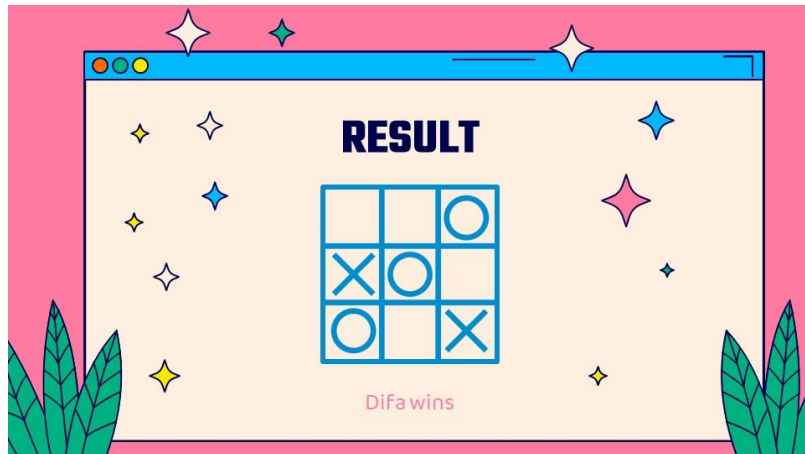
Selanjutnya, *player* akan ditampilkan dengan *gameplay* atau tampilan utama permainan. Pada bagian inilah, *bot* dan *player* bermain dengan memasukkan koordinat tandanya masing-masing. Dan disini juga terdapat keterangan *turn*/giliran *player* secara bergantian.



Gambar 7 sketsa tampilan permainan utama

## 2.8. Tampilan Hasil Permainan

Setelah permainan selesai, maka *player* akan ditampilkan dengan tampilan hasil permainan. Pada tampilan ini, terdapat nama *player1/player2* atau komputer sebagai pemenangnya. Adapun sketsanya sebagai berikut.



Gambar 8 sketsa tampilan hasil permainan

## 2.9. Tampilan Menu *Help*

Pada main menu, terdapat berbagai pilihan, salah satunya menu *help*. Pada menu ini ditampilkan cara bermain dan juga penjelasan dari setiap menu yang ada pada permainan ini dengan bahasa Indonesia maupun bahasa Inggris. Adapun sketsanya sebagai berikut.



Gambar 9 sketsa menu *help*

## 2.9. Tampilan Color Theme

Terdapat juga menu color theme pada main menu yang menampilkan pilihan warna beserta kodenya. Warna yang dipilih bisa untuk warna teks maupun warna background. Adapun sketsanya sebagai berikut.



Gambar 10 sketsa menu *help*

## 2.9. Tampilan Credit

Ada juga menu credit pada main menu yang menampilkan identitas developer dari permainan ini. Adapun sketsanya sebagai berikut.



Gambar 11 menu credit

## 2.9. Tampilan *Highscore*

Pada main menu, terdapat juga menu *highscore* yang menampilkan riwayat kemenangan *player* dalam melawan *bot*. Adapun sketsanya sebagai berikut.





Gambar 12 sketsa menu *highscore*

Seluruh sketsa skenario bisa dicek di :

[https://drive.google.com/drive/folders/1rJ1dUGtM1Q13ePEDBy6vf1BwWXQc1ND\\_?usp=sharing](https://drive.google.com/drive/folders/1rJ1dUGtM1Q13ePEDBy6vf1BwWXQc1ND_?usp=sharing)

### 3. Tingkat Kesulitan

*Difficulty level* atau tingkat kesulitan dari *bot* pada permainan ini terdapat 3 tingkat yaitu *easy*, *normal*, dan *hard*. Dari masing-masing *level* ini, *player* dapat memilih 1 *level* komputer yang akan menjadi lawannya.

### 4. Aturan Bermain

- Ketika *player 1* memulai permainan, maka ada beberapa pilihan yang kemudian harus dipilih, seperti ukuran *grid*, mode permainan, *level* komputer (Jika melawan komputer), dan nama *player*.
- *Player1* mendapatkan selalu mendapatkan first move dan juga tanda 'X'. Sedangkan *bot/player 2* selalu mendapatkan tanda 'O' dan mulai menandai dari giliran yang kedua. Dan seterusnya permainan berlangsung secara bergantian hingga terdapat pemenang.
- Setiap giliran/*turn*, peng-*inputan* tanda 'X' ataupun 'O' untuk *player1* maupun *player2* akan diwaktu sebanyak 10 detik. Jika lebih dari itu, *player1* ataupun *player2* masih bisa menginput koordinat untuk tandanya. Hanya saja tanda yang mereka *input* tidak akan masuk ke dalam *grid*.
- Diantara *player1*, *player2* maupun *bot*, yang pertama kali membentuk 3 tandanya berturut-turut pada *grid* 3x3, 4 tandanya berturut-turut pada *grid* 5x5, dan 5 tandanya berturut-turut pada *grid* 7x7 adalah pemenang.
- Setiap kemenangan *player* melawan *bot*, riwayat kemenangan dari permainan tersebut akan tersimpan pada menu *highscore* dengan format *time*, *size*, *difficulty*, dan juga *nama player* yang bisa dilihat setelah permainan selesai.



## 5. Jumlah Player

Dalam game ini hanya terdapat 2 mode permainan yaitu *single player* dan juga *multi player*. Pada *single player*, pemain akan melawan *bot* secara bergantian, sedangkan pada *multi player*, pemain akan melawan *player* 2 secara bergantian pula

## 6. Karakteristik Computer

Pergerakan dari masing-masing *bot*, terdapat beberapa karakteristik berbeda bergantung pada tingkat kesulitan dan ukuran *grid* yang dipilih. Berikut tabel ilustrasi karakteristik yang dibuat dimana setiap nomor menunjukkan prioritas/urutan :

Level/Ukuran	Grid 3x3	Grid 5x5	Grid 7x7
<b>Easy</b>	<ol style="list-style-type: none"><li>1. Blocking untuk 2 tanda 'X' berturut-turut secara baris, kolom dan diagonal</li><li>2. Mengisi <i>random</i> di sekitar tanda 'X'</li><li>3. Mengisi <i>random</i> yang masih kosong</li></ol>	<ol style="list-style-type: none"><li>1. Mengisi <i>random</i> di sekitar tanda 'X'</li><li>2. Mengisi <i>random</i> yang masih kosong</li></ol>	<ol style="list-style-type: none"><li>1. Blocking untuk 4 tanda 'X' berturut-turut secara baris, kolom dan diagonal</li><li>2. Mengisi <i>random</i> di sekitar tanda 'X'</li><li>3. Mengisi <i>random</i> yang masih kosong</li></ol>
<b>Normal</b>	<ol style="list-style-type: none"><li>1. Blocking untuk 2 tanda 'X' berturut-turut secara baris, kolom dan diagonal</li><li>2. Pemantauan pergerakan pertama dari lawan</li><li>3. Mengisi <i>random</i> di sekitar tanda 'X'</li><li>4. Mengisi <i>random</i> yang masih kosong</li></ol>	<ol style="list-style-type: none"><li>1. Blocking untuk 3 tanda 'X' berturut-turut secara baris, kolom dan diagonal</li><li>2. Mengisi <i>random</i> di sekitar tanda 'X'</li><li>3. Mengisi <i>random</i> yang masih kosong</li></ol>	<ol style="list-style-type: none"><li>1. Blocking untuk 4 tanda 'X' berturut-turut secara baris, kolom dan diagonal</li><li>2. Blocking untuk 3 tanda 'X' berturut-turut secara baris, kolom dan diagonal</li><li>3. Mengisi <i>random</i> di sekitar tanda 'X'</li><li>4. Mengisi <i>random</i> yang masih kosong</li></ol>
<b>Hard</b>	<ol style="list-style-type: none"><li>1. Blocking untuk 2 tanda 'X' berturut-turut secara baris, kolom dan diagonal</li><li>2. Pemantauan pergerakan pertama dari lawan</li><li>3. Pemantauan pergerakan kedua dari lawan</li><li>4. Mengisi <i>random</i> di sekitar tanda 'X'</li><li>5. Mengisi <i>random</i> yang masih kosong</li></ol>	<ol style="list-style-type: none"><li>1. Blocking untuk 3 tanda 'X' berturut-turut secara baris, kolom dan diagonal</li><li>2. Blocking untuk 2 tanda 'X' berturut-turut secara baris, kolom dan diagonal</li><li>3. Mengisi <i>random</i> di sekitar tanda 'X'</li><li>4. Mengisi <i>random</i> yang masih kosong</li></ol>	<ol style="list-style-type: none"><li>1. Blocking untuk 4 tanda 'X' berturut-turut secara baris, kolom dan diagonal</li><li>2. Blocking untuk 3 tanda 'X' berturut-turut secara baris, kolom dan diagonal</li><li>3. Blocking untuk 2 tanda 'X' berturut-turut secara baris dan kolom saja</li><li>4. Mengisi <i>random</i> di sekitar tanda 'X'</li><li>5. Mengisi <i>random</i> yang masih kosong</li></ol>

**Keterangan :** Pengurutan nomor diatas diurutkan berdasarkan prioritas dengan nomor 1 paling priortias

## 6. Inisialisasi Program

- a. `Matrix[8][8]` => Seluruh isi array diisi dengan ' '.
- b. `Player` => 1 // sebagai penanda giliran pada setiap giliran
- c. `p1, p2` => Undefined // sebagai nama *player* pada permainan
- d. `done` => ' ' // sebagai penanda lanjut atau berhentinya permainan

## 7. Variabel Global

- a. `against` => (integer) Menampung pilihan mode permainan
- b. `player` => (integer) Menampung giliran dari *player*
- c. `chooseLanguage` => (integer) Menampung pilihan bahasa pada menu *Help*
- d. `p1[8]` => (char) Menampung nama *player* 1 dengan maksimal 8 karakter
- e. `p2[8]` => (char) Menampung nama *player* 2 dengan maksimal 8 karakter
- f. `done` => (char) Menampung status permainan apakah berakhir atau lanjut
- g. `kodeWarna [1]` => (char) Menampung pilihan warna pada menu *Color*

## BAB 2

### “Desain dan Perancangan”

#### 1. Kebutuhan Data (struktur data)

##### a. Data *Player*

Data ini bertujuan untuk menampung segala kebutuhan yang dibutuhkan pada permainan seperti ukuran *grid*, *namaPlayer*, *levelComputer*, dan waktu. Data ini memiliki tipe struct yang di dalamnya memiliki 4 field yang terdiri dari *grid* sebagai penampung pilihan ukuran *grid*, *namePlayer* sebagai penampung nama *player*, *levelComputer* sebagai penampung pilihan *level* kesulitan *bot*, dan *time* sebagai penampung waktu untuk *highscore*.

##### Struktur Data :

```
typedef struct data{  
    int grid; //menampung pilihan ukuran grid  
    char namePlayer[255]; //menampung nama player  
    int levelComputer; //menampung pilihan level kesulitan bot  
    char time[100]; //menampung waktu untuk highscore  
};
```

##### b. Data Papan Permainan

Data ini bertujuan untuk menampilkan papan permainan dengan tiga pilihan ukuran yaitu 3x3, 5x5, dan 7x7. Adapun papan permainan dibentuk dengan menggunakan tabel sehingga tampilan permainan akan terlihat lebih rapih. Data ini terdiri dari 1 data bertipe array 2 dimensi bertipe karakter dengan ukuran 8x8 dan 1 data bertipe integer sebagai pilihan user.

##### Struktur Data :

```
char matrix[8][8]; //menampilkan papan permainan  
int grid; //menampung pilihan ukuran papan permainan
```

##### c. Data Mode Permainan

Data ini bertujuan untuk menampung pilihan mode permainan. *Player* bisa memilih akan melawan *player* lagi atau komputer. Data ini terdiri dari 1 data bertipe integer sebagai penampung pilihan user untuk mode permainan.

##### Struktur Data :

```
int against; //menampung pilihan mode permainan
```

##### d. Data *Level Computer*

Data ini bertujuan untuk menampung pilihan *level* komputer oleh *player*. Data ini terdiri dari 1 data bertipe integer sebagai penampung pilihan user untuk *level* komputer.

Struktur Data :

int *levelComputer*; //menampung pilihan *level* komputer

#### e. Data Pergerakan Posisi *Player/Bot*

Data ini bertujuan untuk menampung posisi koordinat pergerakan permainan yang kemudian posisi ini menjadi index untuk array 2 dimensi dari papan permainan. Data ini terdiri dari 2 data bertipe integer sebagai posisi koordinat pergerakan permainan.

Struktur Data :

int x; //menampung pilihan posisi koordinat-x

int y; //menampung pilihan posisi koordinat-y

#### f. Data ColorTheme

Data ini bertujuan untuk pilihan kode warna permainan yang dipilih *player*. Data ini terdiri dari 1 data bertipe array of character sebagai kode warna permainan.

Struktur Data :

char kodeWarna[1]; //menampung pilihan kode warna permainan

#### g. Data Pilihan Main Menu

Data ini bertujuan untuk menampung pilihan yang ada di prosedur mainMenu. Data ini terdiri dari 1 data bertipe integer sebagai pilihan yang ada di prosedur mainMenu dan .

Struktur Data :

int choiceMainMenu; //menampung pilihan mainMenu

#### h. Data Lanjut atau Berakhirnya Permainan

Data ini bertujuan untuk menampung hasil keluaran dari function check. Data ini terdiri dari 1 data bertipe character dengan nilai keluaran 'X' maka *player* 1 menang, nilai keluaran 'O' maka *player* 2 atau komputer yang menang, dan nilai keluaran ' ' maka permainan tetapi lanjut

Struktur Data :

char done; //menampung nilai keluaran dari function check

#### i. Data Giliran *Player*

Data ini bertujuan untuk menampung giliran *player* apakah giliran *player* 1 atau *player* 2. Data ini terdiri dari 1 data bertipe integer dengan *player* = 1, maka giliran *player* 1, dan *player* = 2, maka giliran *player* 2

**Struktur Data :**

```
int player; //menampung giliran player
```

**j. Data Highscore**

Data ini bertujuan untuk menampung hal terkait yang ada pada menu *highscore*. Data ini memiliki tipe struct yang di dalamnya memiliki 4 field yang terdiri dari *grid* sebagai penampung pilihan ukuran *grid*, *namePlayer* sebagai penampung nama *player*, *levelComputer* sebagai penampung pilihan *level* kesulitan *bot*, dan *time* sebagai penampung waktu untuk *highscore*. Masing-masing field adalah penampung dari struct dataPemain sebelumnya untuk ditampilkan di menu *highscore*.

**Struktur Data :**

```
typedef struct highscore{  
    int grid; //menampung pilihan ukuran grid  
    char namePlayer[255]; //menampung nama player  
    int levelComputer; //menampung pilihan level kesulitan bot  
    char time[100]; //menampung waktu untuk highscore  
}Record;
```

**k. Data Pemilihan Bahasa Pada Menu Help**

Data ini bertujuan untuk pilihan bahasa dalam menu *help*. Data ini terdiri dari hanya 1 data yang bertipe integer. *Player* dapat memilih 2 pilihan, pilihan pertama untuk bahasa Indonesia dan pilihan kedua untuk bahasa inggris

**Struktur Data :**

```
int chooseLanguage; //menampung pilihan bahasa pada menu help
```

## **2. Modul program**

Bagian ini menjelaskan semua modul yang digunakan dalam program TicTacToe. Modul-modul tersebut dapat berupa function atau prosedur. Tabel ini adalah uraian mengenai modul-modul yang terkait :

No	Nama Modul	Jenis	Keterangan	PJ
1	main	Function	Program utama yang menjalankan keseluruhan program	Difa Mochammad Fadillah
2	mainMenu	Prosedur	Menampilkan pilihan Start, <i>Help</i> , Color Theme, Credit, <i>Highscore</i> , dan Exit.	M Azhar Alauddin
3	<i>help</i>	Prosedur	Menampilkan cara bermain, aturan bermain dan deskripsi singkat permainan dari sebuah file	M Azhar Alauddin
4	showColorTheme	Prosedur	Menampilkan warna background dan teks aplikasi dari sebuah file	M Azhar Alauddin
5	chooseColorTheme	Prosedur	Memilih warna background dan teks aplikasi.	M Azhar Alauddin
6	credit	Prosedur	Menampilkan identitas <i>developer</i> dan <i>creator</i> dari sebuah file	M Azhar Alauddin
7	introduction	Prosedur	Menampilkan introduction atau header	M Azhar Alauddin
8	disp_matrix_3X3	Prosedur	Menampilkan board game berukuran 3x3.	M Azhar Alauddin
9	disp_matrix_5X5	Prosedur	Menampilkan board game berukuran 5x5.	M Azhar Alauddin
10	disp_matrix_7X7	Prosedur	Menampilkan board game berukuran 7x7.	M Azhar Alauddin
11	init_matrix	Prosedur	Menginisialisasi matrix pada <i>grid</i> dengan ' '.	M Azhar Alauddin
12	<i>player1_move</i>	Prosedur	Penempatan tanda 'X' untuk <i>player</i> 1 pada setiap <i>turn</i>	M Azhar Alauddin

13	<i>player2_move</i>	Prosedur	Penempatan tanda ‘O’ untuk <i>player</i> 2 pada setiap <i>turn</i>	M Azhar Alauddin
14	<i>computer_1_move_3X3</i>	Prosedur	Penempatan tanda ‘O’ untuk komputer dengan <i>easy level</i> pada ukuran 3x3	M Azhar Alauddin
15	<i>computer_2_move_3X3</i>	Prosedur	Penempatan tanda ‘O’ untuk komputer dengan <i>normal level</i> pada ukuran 3x3	M Azhar Alauddin
16	<i>computer_3_move_3X3</i>	Prosedur	Penempatan tanda ‘O’ untuk komputer dengan <i>hard level</i> pada ukuran 3x3	M Azhar Alauddin
17	<i>computer_1_move_5X5</i>	Prosedur	Penempatan tanda ‘O’ untuk komputer dengan <i>easy level</i> pada ukuran 5x5	M Azhar Alauddin
18	<i>computer_2_move_5X5</i>	Prosedur	Penempatan tanda ‘O’ untuk komputer dengan <i>normal level</i> pada ukuran 5x5	M Azhar Alauddin
19	<i>computer_3_move_5X5</i>	Prosedur	Penempatan tanda ‘O’ untuk komputer dengan <i>hard level</i> pada ukuran 5x5	M Azhar Alauddin
20	<i>computer_1_move_7X7</i>	Prosedur	Penempatan tanda ‘O’ untuk komputer dengan <i>easy level</i> pada ukuran 7x7	M Azhar Alauddin
21	<i>computer_2_move_7X7</i>	Prosedur	Penempatan tanda ‘O’ untuk komputer dengan <i>normal level</i> pada ukuran 7x7	M Azhar Alauddin
22	<i>computer_3_move_7X7</i>	Prosedur	Penempatan tanda ‘O’ untuk komputer dengan <i>hard level</i> pada ukuran 7x7	M Azhar Alauddin
23	<i>check_3X3</i>	Function	Menentukan kondisi <i>Win/ Lose/ Draw</i> pada	M Azhar Alauddin



			ukuran 3x3 baik itu secara baris, kolom maupun diagonal	
24	check_5X5	Function	Menentukan kondisi Win/ Lose/ Draw pada ukuran 5x5 baik itu secara baris, kolom maupun diagonal	M Azhar Alauddin
25	check_7X7	Function	Menentukan kondisi Win/ Lose/ Draw pada ukuran 7x7 baik itu secara baris, kolom maupun diagonal	M Azhar Alauddin
26	menuStart	Prosedur	Menampung <i>gameplay</i> agar lebih ringkas	M Azhar Alauddin
27	menuHelp	Prosedur	Menampung <i>help</i> agar lebih ringkas	M Azhar Alauddin
28	menuColor	Prosedur	Menampung <i>colortheme</i> agar lebih ringkas	M Azhar Alauddin
29	menuCredit	Prosedur	Menampung <i>credit</i> agar lebih ringkas	M Azhar Alauddin
30	menuHighscore	Prosedur	Menampung <i>highscore</i> agar lebih ringkas	M Azhar Alauddin
31	menuExit	Prosedur	Menampung <i>exit</i> agar lebih ringkas	M Azhar Alauddin
32	readHighscore	Function	Menampilkan file riwayat kemenangan <i>player</i> dalam melawan <i>bot</i>	M Azhar Alauddin
33	saveHighscore	Function	Menyimpan riwayat kemenangan dalam file	M Azhar Alauddin

### 3. Spesifikasi Modul

Function main( )	
Deskripsi	Program utama yang menjalankan program dari awal hingga akhir
Initial State	Program belum berjalan
Final State	Keluar dari program
Modul Dipanggil	menuStart, menuHelp, menuColor, menuCredit, menuHighscore, menuExit
Author	Difa Mochammad Fadillah
Kamus Data	choiceMainMenu : integer
Algoritma	
lakukan tampilkan mainMenu jika choiceMainMenu = 1 maka menuStart() jika choiceMainMenu = 2 maka menuHelp() jika choiceMainMenu = 3 maka menuColor() jika choiceMainMenu = 4 maka menuCredit() jika choiceMainMenu = 5 maka menuHighscore() jika choiceMainMenu = 6 maka menuExit() return 0	

Prosedur mainMenu( )	
Deskripsi	Menampilkan pilihan Start, Help, Color Theme, Credit, Highscore, dan Exit
Initial State	Belum ada yang ditampilkan
Final State	Sudah ditampilkan pilihan Start, Help, Color Theme, Credit, Highscore, Exit
Modul Pemanggil	main
Author	Muhammad Azhar Alauddin
Algoritma	

lakukan tampilkan pilihan main menu masukkan pilihan main menu
--

Prosedur <i>help()</i>	
------------------------	--

Deskripsi	Menampilkan Deskripsi singkat terkait dengan permainan Tic Tac Toe, cara bermain, dan aturan bermain
Initial State	Belum ada yang ditampilkan
Final State	Sudah ditampilkan Deskripsi singkat terkait dengan permainan Tic Tac Toe, cara bermain, dan aturan bermain
Modul Pemanggil	menu <i>Help</i>
Author	Muhammad Azhar Alauddin
Parameter <i>Input</i>	Pilihan bahasa

Algoritma
-----------

lakukan tampilkan deskripsi singkat permainan tampilkan cara bermain tampilkan aturan bermain
--

Prosedur chooseColorTheme( )	
Deskripsi	Memilih warna background dan teks aplikasi.
Initial State	Belum ada yang ditampilkan
Final State	Sudah ditampilkan pilihan warna background dan warna teks
Modul Pemanggil	menuColor
Author	Muhammad Azhar Alauddin
Algoritma	
lakukan tampilkan pilihan warna background tampilkan pilihan warna text masukkan kode warna	

Prosedur showColorTheme( )	
Deskripsi	Menampilkan warna background dan warna teks
Initial State	Belum ada yang ditampilkan
Final State	Sudah ditampilkan warna background dan warna teks sesuai dengan pilihan yang <i>diinput</i>
Modul Pemanggil	menuColor
Author	Muhammad Azhar Alauddin
Parameter <i>input</i>	kode warna

Prosedur credit( )	
Deskripsi	Menampilkan identitas <i>developer</i> dan <i>creator</i> dari permainan ini
Initial State	Belum ada yang ditampilkan
Final State	Sudah ditampilkan identitas <i>developer</i> dan <i>creator</i> dari permainan ini
Modul Pemanggil	menuCredit
Author	Muhammad Azhar Alauddin
Prosedur menuStart ( )	
Deskripsi	Menampung <i>gameplay</i> agar lebih ringkas

Initial State	Belum ada yang ditampilkan
Final State	Sudah ditampilkan <i>gameplay</i> dari permainan
Modul Pemanggil	main
Modul Dipanggil	Introduction ( ), <i>computer_move_1_3x3</i> ( ), <i>computer_move_2_3x3</i> ( ), <i>computer_move_3_3x3</i> ( ), <i>computer_move_3_3x3</i> , <i>computer_move_1_5x5</i> ( ), <i>computer_move_2_5x5</i> , <i>computer_move_3_5x5</i> ( ), <i>computer_move_1_7x7</i> ( ), <i>computer_move_2_7x7</i> ( ), <i>computer_move_3_7x7</i> ( ), <i>check_3x3</i> ( ), <i>check_5x5</i> ( ), <i>check_7x7</i> ( ), <i>init_matrix</i> ( )
Author	Muhammad Azhar Alauddin

Prosedur menu <i>Help</i> ( )	
Deskripsi	Menampung <i>help</i> agar lebih ringkas
Initial State	Belum ada yang ditampilkan
Final State	Sudah ditampilkan <i>help</i> dari permainan
Modul Pemanggil	main
Modul Dipanggil	<i>help</i> ( )
Author	Muhammad Azhar Alauddin

Prosedur menuColor ( )	
Deskripsi	Menampung <i>colorTheme</i> agar lebih ringkas
Initial State	Belum ada yang ditampilkan
Final State	Sudah ditampilkan <i>ColorTheme</i> dari permainan
Modul Pemanggil	main
Modul Dipanggil	<i>chooseColorTheme</i> ( ), <i>showColorTheme</i> ( )
Author	Muhammad Azhar Alauddin

Prosedur menuCredit ( )	
Deskripsi	Menampung <i>credit</i> agar lebih ringkas

Initial State	Belum ada yang ditampilkan
Final State	Sudah ditampilkan credit dari permainan
Modul Pemanggil	main
Modul Dipanggil	credit ( )
Author	Muhammad Azhar Alauddin

<b>Prosedur menuExit ( )</b>	
Deskripsi	Menampung exit agar lebih ringkas
Initial State	Belum ada yang ditampilkan
Final State	Sudah ditampilkan exit dari permainan
Modul Pemanggil	main
Author	Muhammad Azhar Alauddin

<b>Prosedur introduction( )</b>	
Deskripsi	Menampilkan introduction atau header
Initial State	Belum ada yang ditampilkan
Final State	Sudah ditampilkan introduction atau header
Modul Pemanggil	menuStart
Author	Muhammad Azhar Alauddin

<b>Prosedur disp_matrix_3X3( )</b>	
Deskripsi	Menampilkan board game berukuran 3x3.
Initial State	Belum ada yang ditampilkan
Final State	Sudah ditampilkan board game berukuran 3x3.
Modul Pemanggil	menuStart
Author	Muhammad Azhar Alauddin

Prosedur <i>disp_matrix_5X5()</i>	
Deskripsi	Menampilkan board game berukuran 5x5.
Initial State	Belum ada yang ditampilkan
Final State	Sudah ditampilkan board game berukuran 5x5.
Modul Pemanggil	menuStart
Author	Muhammad Azhar Alauddin

Prosedur <i>disp_matrix_7X7()</i>	
Deskripsi	Menampilkan board game berukuran 7x7.
Initial State	Belum ada yang ditampilkan
Final State	Sudah ditampilkan board game berukuran 7x7.
Modul Pemanggil	menuStart
Author	Muhammad Azhar Alauddin

Prosedur <i>init_matrix()</i>	
Deskripsi	Menginisialisasi matrix pada <i>grid</i> dengan ‘ ‘
Initial State	Isi keseluruhan matrix belum diisi apapun
Final State	Isi keseluruhan matrix sudah diisi ‘ ‘
Modul Pemanggil	menuStart
Kamus Data	i : integer j : integer
Author	Muhammad Azhar Alauddin

Prosedur <i>player1_move()</i>	
Deskripsi	Penempatan tanda ‘X’ untuk <i>player</i> 1 pada setiap <i>turn</i> . Pada penempatan ini, <i>player</i> 1 diberikan waktu selama 10 detik. Jika lebih dari itu, maka <i>player</i> 1 tetap bisa mengisi tanda tetapi tidak akan tersimpan pada board game
Initial State	Belum ada penempatan tanda ‘X’ pada <i>turn</i> tertentu



Final State	Sudah ada penempatan tanda 'X' pada <i>turn</i> tertentu
Modul Pemanggil	menuStart
Kamus Data	waktuAwal : integer waktuAkhir : integer x : integer y : integer
Author	Muhammad Azhar Alauddin
Algoritma	
lakukan waktuAwal = clock( ) waktuAkhir = clock( ) masukkan titik koordinat x dan y untuk pergerakan  jika waktuAkhir-waktuAwal >= 10000 <b>maka</b> kembali ke modul pemanggil  jika matrix[--x][--y] tidak sama dengan spasi <b>maka</b> player1_move( )  selain itu <b>maka</b> matrix[x][y] = 'X'	

Prosedur player2_move( )	
Deskripsi	Penempatan tanda 'O' untuk <i>player</i> 2 pada setiap <i>turn</i> . Pada penempatan ini, <i>player</i> 2 diberikan waktu selama 10 detik. Jika lebih dari itu, maka <i>player</i> 2 tetap bisa mengisi tanda tetapi tidak akan tersimpan pada board game
Initial State	Belum ada penempatan tanda 'O' pada <i>turn</i> tertentu
Final State	Sudah ada penempatan tanda 'O' pada <i>turn</i> tertentu
Modul Pemanggil	menuStart
Kamus Data	waktuAwal : integer waktuAkhir : integer x : integer y : integer
Author	Muhammad Azhar Alauddin
Algoritma	
lakukan	

```
waktuAwal = clock( )
waktuAkhir = clock( )
masukkan titik koordinat x dan y untuk pergerakan
```

```
jika waktuAkhir-waktuAwal >= 10000 maka
    kembali ke modul pemanggil
```

```
jika matrix[--x][--y] tidak sama dengan spasi maka
    player1_move( )
```

```
selain itu maka
    matrix[x][y] = 'O'
```

#### Prosedur *computer\_move\_1\_3X3* ( )

Deskripsi	Penempatan tanda 'O' untuk <i>computer</i> pada setiap <i>turn</i> pada <i>grid</i> 3x3 dengan <i>easy level</i>
Initial State	Belum ada penempatan tanda 'O' pada <i>turn</i> tertentu
Final State	Sudah ada penempatan tanda 'O' pada <i>turn</i> tertentu
Modul Pemanggil	menuStart
Kamus Data	i : integer j : integer
Author	Muhammad Azhar Alauddin
Algoritma	
<b>lakukan</b> jika tanda 'X' sudah 2 kali berturut-turut <b>maka</b> blocking pergerakan tersebut baik itu secara baris, kolom ataupun diagonal selain itu <b>maka</b> Isi secara acak di sekitar tanda 'X' yang sudah ada selain itu <b>maka</b> Isi secara acak yang masih kosong	

#### Prosedur *computer\_move\_1\_5X5* ( )

Deskripsi	Penempatan tanda 'O' untuk <i>computer</i> pada setiap <i>turn</i> pada <i>grid</i> 5x5 dengan <i>easy level</i>
Initial State	Belum ada penempatan tanda 'O' pada <i>turn</i> tertentu
Final State	Sudah ada penempatan tanda 'O' pada <i>turn</i> tertentu
Modul Pemanggil	menuStart

Kamus Data	i : integer j : integer
Author	Muhammad Azhar Alauddin
Algoritma	
<b>lakukan</b> jika dalam board ada tanda 'X' <b>maka</b> Isi secara acak di sekitar tanda 'X' yang sudah ada selain itu <b>maka</b> Isi secara acak yang masih kosong	

Prosedur <i>computer_move_1_7X7</i> ( )	
Deskripsi	Penempatan tanda 'O' untuk <i>computer</i> pada setiap <i>turn</i> pada <i>grid</i> 7x7 dengan <i>easy level</i>
Initial State	Belum ada penempatan tanda 'O' pada <i>turn</i> tertentu
Final State	Sudah ada penempatan tanda 'O' pada <i>turn</i> tertentu
Modul Pemanggil	menuStart
Kamus Data	i : integer j : integer
Author	Muhammad Azhar Alauddin
Algoritma	
<b>lakukan</b> jika tanda 'X' sudah 4 kali berturut-turut <b>maka</b> blocking pergerakan tersebut baik itu secara baris, kolom ataupun diagonal selain itu <b>maka</b> Isi secara acak di sekitar tanda 'X' yang sudah ada selain itu <b>maka</b> Isi secara acak yang masih kosong	

Prosedur <i>computer_move_2_3x3</i> ( )	
Deskripsi	Penempatan tanda 'O' untuk <i>computer</i> pada setiap <i>turn</i> pada <i>grid</i> 3x3 dengan <i>normal level</i>
Initial State	Belum ada penempatan tanda 'O' pada <i>turn</i> tertentu
Final State	Sudah ada penempatan tanda 'O' pada <i>turn</i> tertentu
Modul Pemanggil	menuStart

Kamus Data	i : integer j : integer
Author	Muhammad Azhar Alauddin
Algoritma	
<b>lakukan</b> jika tanda 'X' sudah 2 kali berturut-turut <b>maka</b> blocking pergerakan tersebut baik itu secara baris, kolom ataupun diagonal selain itu <b>maka</b> pantau pergerakan pertama dari <i>player</i> 1 selain itu <b>maka</b> Isi secara acak di sekitar tanda 'X' yang sudah ada selain itu <b>maka</b> Isi secara acak yang masih kosong	

Prosedur <i>computer_move_2_5x5</i> ( )	
Deskripsi	Penempatan tanda 'O' untuk <i>computer</i> pada setiap <i>turn</i> pada <i>grid</i> 5x5 dengan <i>normal level</i>
Initial State	Belum ada penempatan tanda 'O' pada <i>turn</i> tertentu
Final State	Sudah ada penempatan tanda 'O' pada <i>turn</i> tertentu
Modul Pemanggil	menuStart
Kamus Data	i : integer j : integer
Author	Muhammad Azhar Alauddin
Algoritma	
<b>lakukan</b> jika tanda 'X' sudah 3 kali berturut-turut <b>maka</b> blocking pergerakan tersebut baik itu secara baris, kolom ataupun diagonal selain itu <b>maka</b> Isi secara acak di sekitar tanda 'X' yang sudah ada selain itu <b>maka</b> Isi secara acak yang masih kosong	

Prosedur <i>computer_move_2_7x7</i> ( )	
Deskripsi	Penempatan tanda 'O' untuk <i>computer</i> pada setiap <i>turn</i> pada <i>grid</i> 7x7 dengan <i>normal level</i>
Initial State	Belum ada penempatan tanda 'O' pada <i>turn</i> tertentu

Final State	Sudah ada penempatan tanda 'O' pada <i>turn</i> tertentu
Modul Pemanggil	menuStart
Kamus Data	i : integer j : integer
Author	Muhammad Azhar Alauddin
Algoritma	
<b>lakukan</b> jika tanda 'X' sudah 4 kali berturut-turut <b>maka</b> blocking pergerakan tersebut baik itu secara baris, kolom ataupun diagonal jika tanda 'X' sudah 3 kali berturut-turut <b>maka</b> blocking pergerakan tersebut baik itu secara baris, kolom ataupun diagonal selain itu <b>maka</b> Isi secara acak di sekitar tanda 'X' yang sudah ada selain itu <b>maka</b> Isi secara acak yang masih kosong	

Prosedur <i>computer_move_3_3x3</i> ( )	
Deskripsi	Penempatan tanda 'O' untuk <i>computer</i> pada setiap <i>turn</i> pada <i>grid</i> 3x3 dengan <i>hard level</i>
Initial State	Belum ada penempatan tanda 'O' pada <i>turn</i> tertentu
Final State	Sudah ada penempatan tanda 'O' pada <i>turn</i> tertentu
Modul Pemanggil	menuStart
Kamus Data	i : integer j : integer
Author	Muhammad Azhar Alauddin
Algoritma	
<b>lakukan</b> jika tanda 'X' sudah 2 kali berturut-turut <b>maka</b> blocking pergerakan tersebut baik itu secara baris, kolom ataupun diagonal selain itu <b>maka</b> pantau pergerakan pertama dari <i>player</i> 1 selain itu <b>maka</b> pantau pergerakan kedua dari <i>player</i> 1 selain itu <b>maka</b> Isi secara acak di sekitar tanda 'X' yang sudah ada selain itu <b>maka</b> Isi secara acak yang masih kosong	

Prosedur <i>computer_move_3_5x5</i> ( )	
Deskripsi	Penempatan tanda 'O' untuk <i>computer</i> pada setiap <i>turn</i> pada <i>grid</i> 5x5 dengan <i>hard level</i>
Initial State	Belum ada penempatan tanda 'O' pada <i>turn</i> tertentu
Final State	Sudah ada penempatan tanda 'O' pada <i>turn</i> tertentu
Modul Pemanggil	menuStart
Kamus Data	i : integer j : integer
Author	Muhammad Azhar Alauddin
Algoritma	
<b>lakukan</b> jika tanda 'X' sudah 3 kali berturut-turut <b>maka</b> blocking pergerakan tersebut baik itu secara baris, kolom ataupun diagonal jika tanda 'X' sudah 2 kali berturut-turut <b>maka</b> blocking pergerakan tersebut baik itu secara baris, kolom ataupun diagonal selain itu <b>maka</b> Isi secara acak di sekitar tanda 'X' yang sudah ada selain itu <b>maka</b> Isi secara acak yang masih kosong	

Prosedur <i>computer_move_3_7x7</i> ( )	
Deskripsi	Penempatan tanda 'O' untuk <i>computer</i> pada setiap <i>turn</i> pada <i>grid</i> 7x7 dengan <i>hard level</i>
Initial State	Belum ada penempatan tanda 'O' pada <i>turn</i> tertentu
Final State	Sudah ada penempatan tanda 'O' pada <i>turn</i> tertentu
Modul Pemanggil	menuStart
Kamus Data	i : integer j : integer
Author	Muhammad Azhar Alauddin
Algoritma	
<b>lakukan</b> jika tanda 'X' sudah 4 kali berturut-turut <b>maka</b> blocking pergerakan tersebut baik itu secara baris, kolom ataupun diagonal	

jika tanda 'X' sudah 3 kali berturut-turut **maka**  
 blocking pergerakan tersebut baik itu secara baris, kolom ataupun diagonal  
 jika tanda 'X' sudah 2 kali berturut-turut **maka**  
 blocking pergerakan tersebut baik itu secara baris ataupun kolom  
 selain itu **maka**  
 Isi secara acak di sekitar tanda 'X' yang sudah ada  
 selain itu **maka**  
 Isi secara acak yang masih kosong

#### Function check\_3X3 ( )

Deskripsi	Menentukan kondisi Win/ Lose/ Draw pada ukuran 3x3 baik itu secara baris, kolom maupun diagonal
Modul Pemanggil	menuStart
Kamus Data	i : integer j : integer
Author	Muhammad Azhar Alauddin
Algoritma	
<b>lakukan</b> jika tanda 'X' sudah 3 kali berturut-turut baik itu secara baris, kolom ataupun diagonal <b>maka</b> kembalikan 'X' ke modul pemanggil jika tanda 'O' sudah 3 kali berturut-turut baik itu secara baris, kolom ataupun diagonal <b>maka</b> kembalikan 'O' ke modul pemanggil	

#### Function check\_5X5 ( )

Deskripsi	Menentukan kondisi Win/ Lose/ Draw pada ukuran 5x5 baik itu secara baris, kolom maupun diagonal
Modul Pemanggil	menuStart
Kamus Data	i : integer j : integer
Author	Muhammad Azhar Alauddin
Algoritma	
<b>lakukan</b> jika tanda 'X' sudah 4 kali berturut-turut baik itu secara baris, kolom ataupun diagonal <b>maka</b> kembalikan 'X' ke modul pemanggil	



jika tanda 'O' sudah 4 kali berturut-turut baik itu secara baris, kolom ataupun diagonal **maka**  
kembalikan 'O' ke modul pemanggil

#### Function check\_7X7 ( )

Deskripsi	Menentukan kondisi Win/ Lose/ Draw pada ukuran 7x7 baik itu secara baris, kolom maupun diagonal
Modul Pemanggil	menuStart
Kamus Data	i : integer j : integer
Author	Muhammad Azhar Alauddin
Algoritma	
<p><b>lakukan</b>            jika tanda 'X' sudah 5 kali berturut-turut baik itu secara baris, kolom ataupun diagonal <b>maka</b>                kembalikan 'X' ke modul pemanggil            jika tanda 'O' sudah 5 kali berturut-turut baik itu secara baris, kolom ataupun diagonal <b>maka</b>                kembalikan 'O' ke modul pemanggil</p>	

#### Prosedur saveHighscore( )

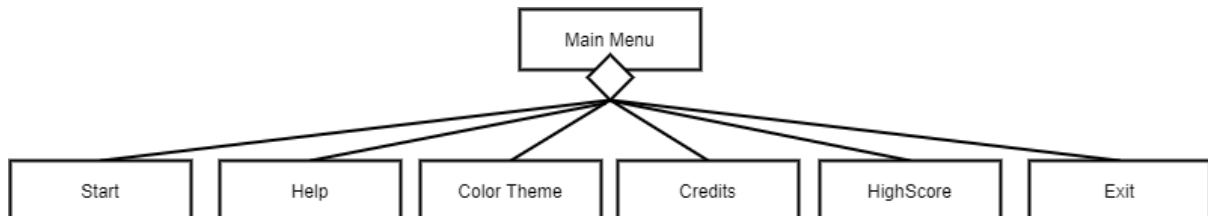
Deskripsi	Menyimpan riwayat kemenangan melawan <i>bot</i> dalam file
Initial State	Riwayat kemenangan belum tersimpan dalam file
Final State	Riwayat kemenangan sudah tersimpan dalam file
Modul Pemanggil	menuHighscore
Author	Muhammad Azhar Alauddin (credit : Rangga Yudha Yudistira)

#### Prosedur readHighscore( )

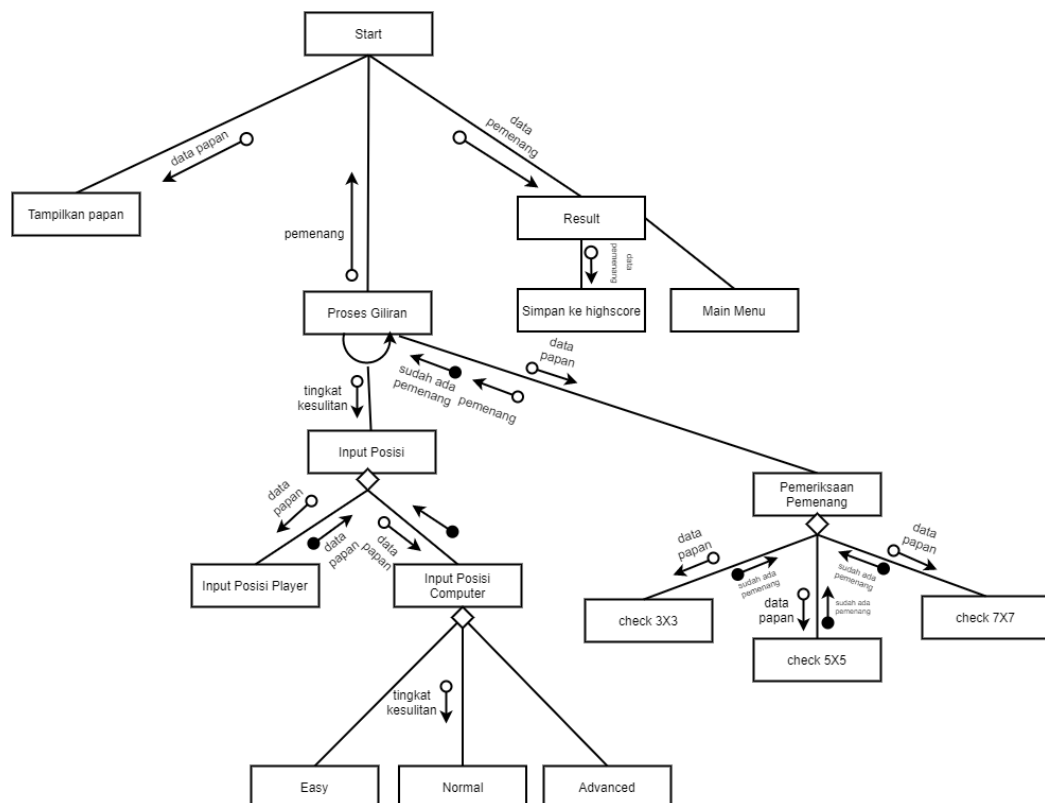
Deskripsi	Menampilkan file riwayat kemenangan <i>player</i> dalam melawan <i>bot</i>
Initial State	Riwayat kemenangan belum ditampilkan di implementasi program
Final State	Riwayat kemenangan sudah ditampilkan di implementasi program
Modul Pemanggil	menuHighscore

### 3. Structure chart

Berikut kami ilustrasikan interaksi modul yang sudah kami buat dalam bentuk struktur chart



Gambar 13 structure chart prosedur main menu



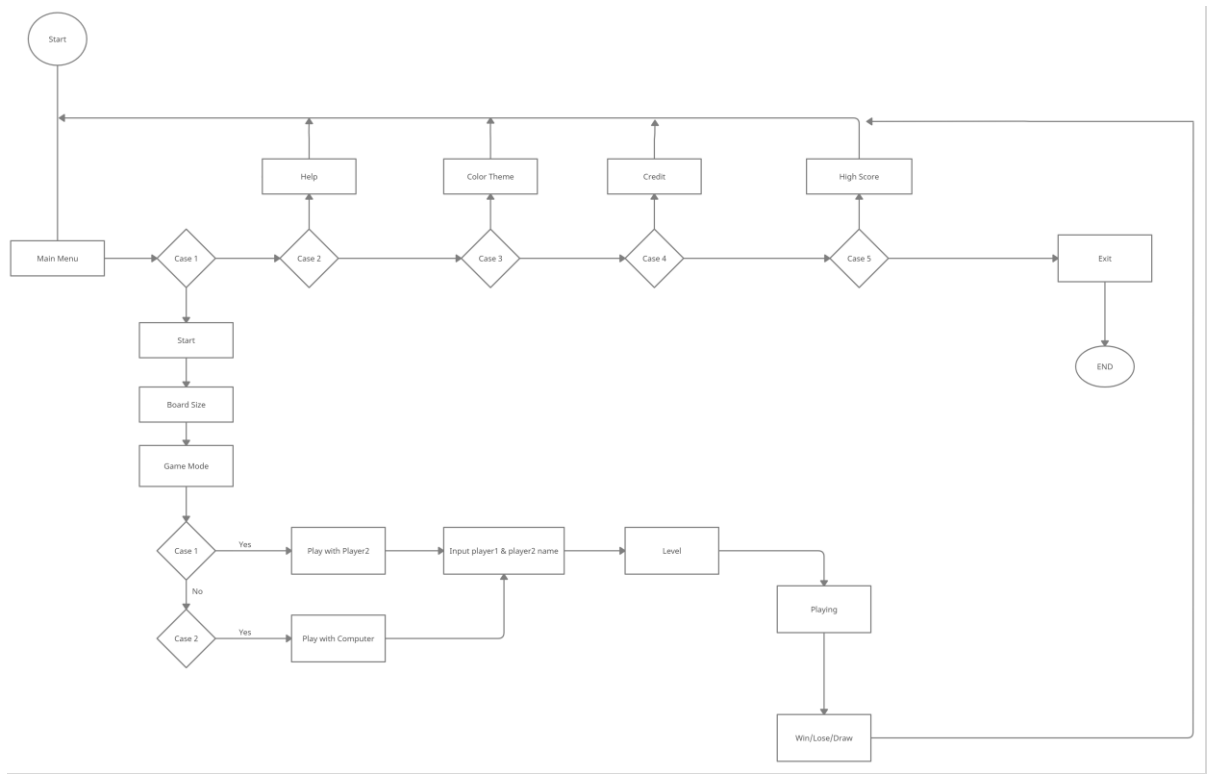
Gambar 14 structure chart gameplay (permainan utama)

Selengkapnya :

[https://drive.google.com/drive/folders/1m\\_GFMMsKfY-eKOILIVprkVNHhuUm-mQa?usp=sharing](https://drive.google.com/drive/folders/1m_GFMMsKfY-eKOILIVprkVNHhuUm-mQa?usp=sharing)

#### 4. Flowchart Program Utama

Berikut kami ilustrasikan alur program utama yang sudah kami buat dalam bentuk *flow chart* (alur diagram).



Gambar 15 *flow chart* program utama

Selengkapnya :

<https://drive.google.com/file/d/127sdFnGZ4WdYJh0mmxy5PBnLBjBLhNKN/view?usp=sharing>

## BAB IV : HASIL AKHIR PROGRAM

### 1. Link github

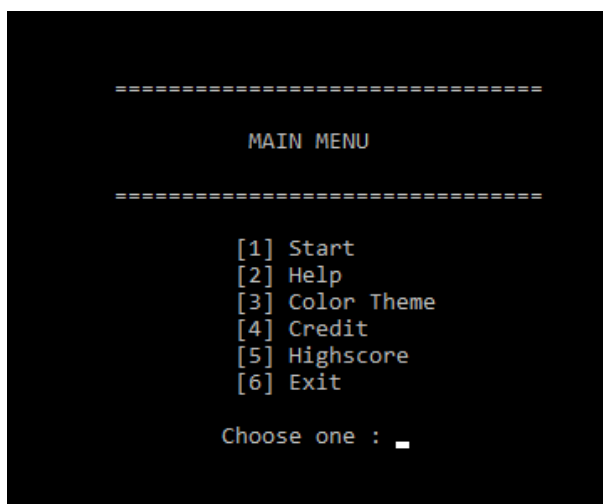
<https://github.com/zharmedia/Tugas-Besar-Tic-Tac-Toe.git>

### 2. Interface dan Data Testing

- Pre-Game

- a) Tampilan Menu Utama

Pada tampilan main menu, *player* akan ditampilkan dengan 5 pilihan yaitu start, *help*, *colortheme*, *credit*, *highscore*, dan exit. Jika *player* memilih 1, maka *player* akan ditampilkan ke *gameplay* utama seperti yang ada pada gambar 27, jika *player* memilih 2 maka *player* akan ditampilkan ke *help* seperti yang ada pada gambar 21, jika *player* memilih 3 maka *player* akan ditampilkan ke *colorTheme* seperti yang ada pada gambar 24, jika *player* memilih 4 maka *player* akan ditampilkan ke *credit* seperti yang ada pada gambar 25, jika *player* memilih 5 maka *player* akan ditampilkan ke *highscore* seperti yang ada pada gambar 26, dan jika *player* memilih 6 maka *player* akan keluar dari permainan. Sedangkan, jika *player* memilih angka di luar dari pilihan tersebut maka *player* akan diminta dan terus diminta untuk memilih pilihan yang tersedia (OKE)



Gambar 16 Hasil implementasi tampilan utama

### b) Tampilan Memilih Ukuran Grid

Setelah *player* memilih 1 pada menu utama, maka *player* akan ditampilkan pada tampilan untuk memilih ukuran *grid* dimana disini *player* bisa memilih 3 untuk ukuran *grid* 3x3, memilih 5 untuk *grid* ukuran 5x5, dan memilih 7 untuk *grid* ukuran 7x7. Selain daripada itu, maka *player* akan diminta dan terus diminta untuk memilih pilihan yang tersedia (OKE)

```
*** TIC TAC TOE GAME ***

## Halo! Selamat datang di game ini
## Berikut ada beberapa hal yang harus anda penuhi

Board Game
Press 3 and Enter for [3X3]
Press 5 and Enter for [5X5]
Press 7 and Enter for [7X7]

Choose one :
```

Gambar 17 Hasil implementasi tampilan memilih ukuran *grid*

### c) Tampilan Memilih Lawan

Setelah memilih ukuran *grid*, maka *player* akan ditampilkan pada tampilan mode permainan atau dengan siapa *player* akan main. Disini terdapat 2 pilihan, yaitu 1 untuk bermain bersama *bot* dan 2 untuk bermain bersama *player* 2. Selain daripada itu, maka *player* akan diminta dan terus diminta untuk memilih pilihan yang tersedia (OKE)

```
*** TIC TAC TOE GAME ***

## Halo! Selamat datang di game ini
## Berikut ada beberapa hal yang harus anda penuhi

Whom do you wanna play?
Press 1 and Enter for playing with computer
Press 2 and Enter for playing with player2

Choose one : _
```

Gambar 18 Hasil implementasi tampilan memilih lawan

### d) Tampilan Menginput Nama Player

Setelah menginput mode permainan, *player* akan ditampilkan pada tampilan yang mana *player* harus menginput namanya dengan maksimal 8 huruf. Dan jika *player* sebelumnya memilih *player* 2 sebagai lawan, maka *player* 2 pun harus menginput namanya. Nama disini diperuntukkan ketika *player* mulai bermain dan juga setelah bermain. (OKE)

```
*** TIC TAC TOE GAME ***

## Halo! Selamat datang di game ini
## Berikut ada beberapa hal yang harus anda penuhi

Enter player's name : Azhar
```

Gambar 19 Hasil implementasi tampilan *input* nama *player*

**e) Tampilan Memilih *Level* Kesulitan**

Setelah *player* menginput nama, maka *player* harus memilih *level* kesulitan dari komputer. Disini terdapat 3 *level* kesulitan, yaitu *easy*, *normal* dan *hard*. Jika *player* memilih 1 maka *player* akan melawan komputer dengan *level easy*. Jika *player* memilih 2 maka *player* akan melawan komputer dengan *level normal*. Dan jika *player* memilih 3 maka *player* akan memilih komputer dengan *level hard*. Selain daripada itu, maka *player* akan diminta dan terus diminta untuk memilih pilihan yang tersedia (OKE)

```
*** TIC TAC TOE GAME ***

## Halo! Selamat datang di game ini
## Berikut ada beberapa hal yang harus anda penuhi

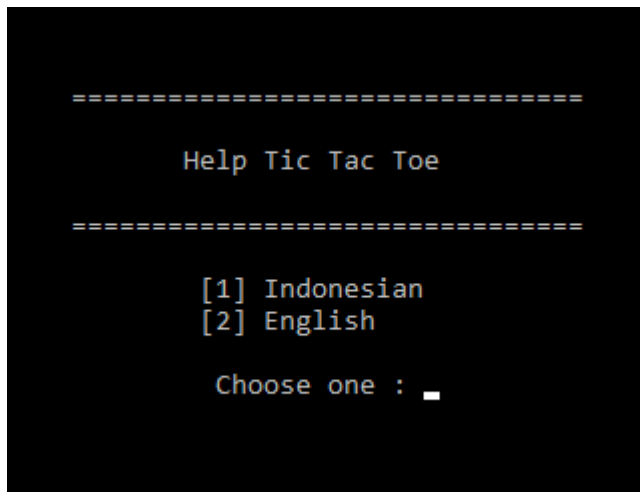
Enter level komputer :
[1] Easy
[2] Normal
[3] Hard

Choose one : 3
```

Gambar 20 Hasil implementasi tampilan *level* kesulitan

**f) Tampilan *Help* 1.0**

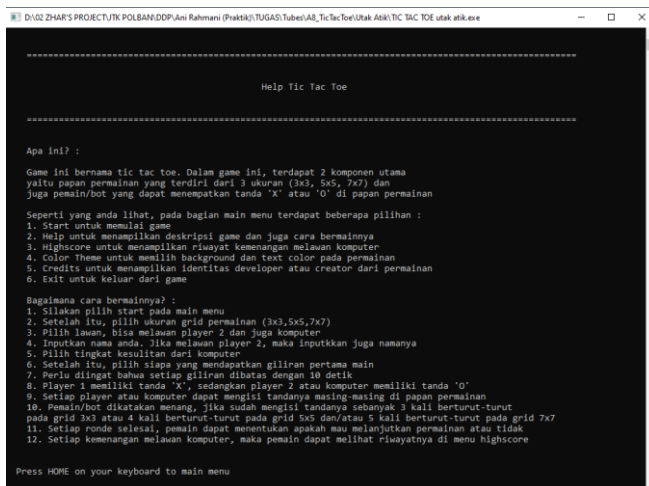
Jika *player* memilih 2 pada main menu, maka *player* akan ditampilkan pada tampilan menu *help* dimana pada menu *help* ini terdapat penjelasan terkait dengan cara bermain, aturan bermain dan juga deskripsi singkat. Dan pada tampilan ini, terdapat 2 bahasa, jika *player* memilih 1 maka *player* akan ditampilkan penjelasan terkait cara bermain, aturan bermain dan juga deskripsi singkat dalam bahasa Indonesia, dan jika *player* memilih 2 maka *player* akan ditampilkan penjelasan terkait cara bermain, aturan bermain dan juga deskripsi singkat dalam bahasa Inggris. Selain daripada itu, maka *player* akan diminta dan terus diminta untuk memilih pilihan yang tersedia (OKE)



Gambar 21 Hasil implementasi tampilan *Help* 1.0 (Memilih bahasa)

### g) Tampilan *Help* 1.1

Berikut tampilan penjelasan terkait cara bermain, aturan bermain dan juga deskripsi singkat dalam bahasa Indonesia. Tampilan ini akan muncul jika *player* memilih 1 pada menu *help* sebelumnya. Untuk kembali ke main menu, maka *player* harus menekan HOME pada keyboard. Jika menekan selain HOME, maka tidak akan kembali ke main menu (OKE)



Gambar 22 Hasil implementasi tampilan *Help* 1.1

### h) Tampilan *Help* 1.2

Berikut tampilan penjelasan terkait cara bermain, aturan bermain dan juga deskripsi singkat dalam bahasa Inggris. Tampilan ini akan muncul jika *player* memilih 1 pada menu *help* sebelumnya. Untuk kembali ke main menu, maka *player* harus menekan



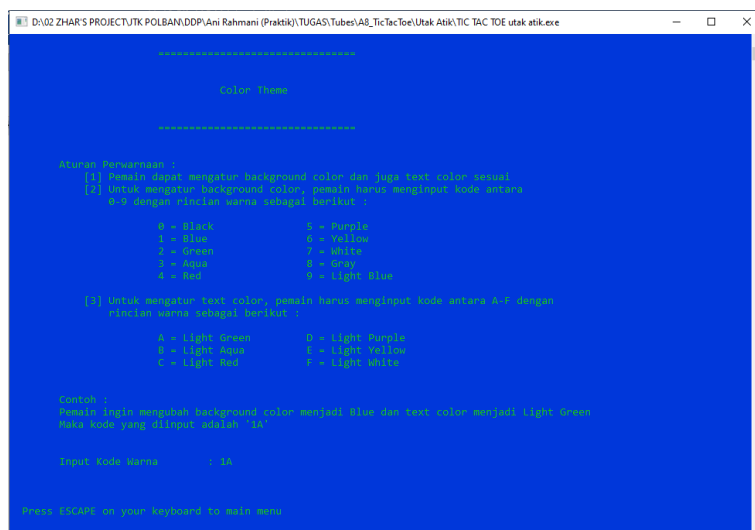
HOME pada keyboard. Jika menekan selain HOME, maka tidak akan kembali ke main menu (OKE)



Gambar 23 Hasil implementasi tampilan *Help* 1.2

## i) Tampilan Color Theme

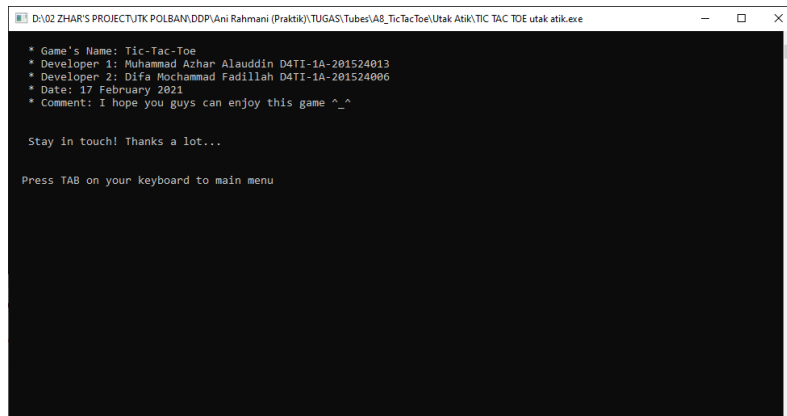
Jika *player* memilih 3 pada main menu, maka *player* akan ditampilkan pada pilihan warna berikut ini, dimana *player* bisa menentukan warna dari teks maupun background sesuai dengan keinginan. Sebagai contoh, *player* ingin memilih warna biru sebagai background dan warna hijau cerah sebagai teks, maka *player* bisa menginputkan dengan kode warna 1A. Untuk kembali ke main menu, maka *player* harus menekan ESCAPE pada keyboard. Jika menekan selain ESCAPE, maka tidak akan kembali ke main menu (OKE)



Gambar 24 Hasil implementasi tampilan Color Theme

### j) Tampilan Credits

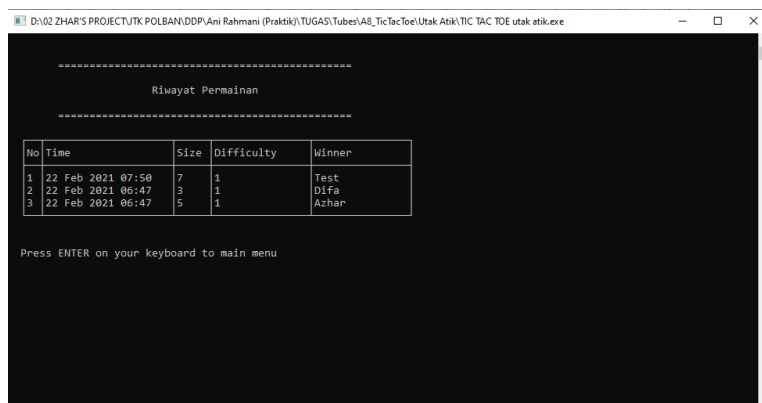
Jika pada main menu *player* memilih 4, maka *player* akan ditampilkan pada tampilan credit yang berupa identitas *developer* maupun *creator* dari permainan ini seperti pada gambar berikut. Untuk kembali ke main menu, maka *player* harus menekan TAB pada keyboard. Jika menekan selain TAB, maka tidak akan kembali ke main menu (OKE)



Gambar 25 Hasil implementasi tampilan Credits

### k) Tampilan High Score

Jika *player* memilih 5 pada main menu, maka *player* akan ditampilkan dengan tampilan *highscore* seperti gambar di bawa dimana disini akan ditampilkan waktu permainan berlangsung, ukuran *grid*, *level* komputer, dan nama *player* yang memenangkan permainan melawan komputer. Untuk kembali ke main menu, maka *player* harus menekan ENTER pada keyboard. Jika menekan selain ENTER, maka tidak akan kembali ke main menu (OKE)



Gambar 26 Hasil implementasi tampilan High Score

- **Gameplay**

- a) **Tampilan Game Play**

Berikut tampilan *gameplay* atau permainan utama, dimana pada tampilan ini *player* maupun *computer* menginputkan tandanya masing-masing. Pada tampilan ini, nama *player* dan ukuran *player* itu ditentukan oleh *inputan* sebelumnya. Seperti contoh disini diinputkan nama “Azhar” sebagai *player* dan 7x7 sebagai ukuran *grid*nya. Jika koordinat yang diinputkan tidak termasuk pada koordinat pada *grid* atau sudah terisi, maka akan ditampilkan “*invalid move, try again*” dan harus menginputkan kembali tandanya. (OKE)

```

Player1(X)
Player2/Computer(O)

Azhar as player 1
Player1's turn

(x,y) 1 2 3 4 5 6 7
1 | X |  |  |  |  |  |
2 | O |  |  |  |  |  |
3 |  |  |  |  |  |  |
4 |  |  |  | O | X |  |
5 |  |  |  | X |  |  |
6 |  |  |  | O |  |  |
7 |  |  |  |  |  | O | X |

Enter x and y co-ordinates of your move: 

```

```

Player1(X)
Player2/Computer(O)

Test4 as player 1
Player1's turn

(x,y) 1 2 3 4 5 6 7
1 |  |  |  |  |  |  |
2 | O | X |  |  |  |  |
3 |  | O | X |  |  |  |
4 |  |  | O | X |  |  |
5 |  |  |  |  |  |  |
6 |  |  |  |  |  |  |
7 |  |  |  |  |  |  |

Enter x and y co-ordinates of your move: 4 4
Invalid Move, try again

Enter x and y co-ordinates of your move: 

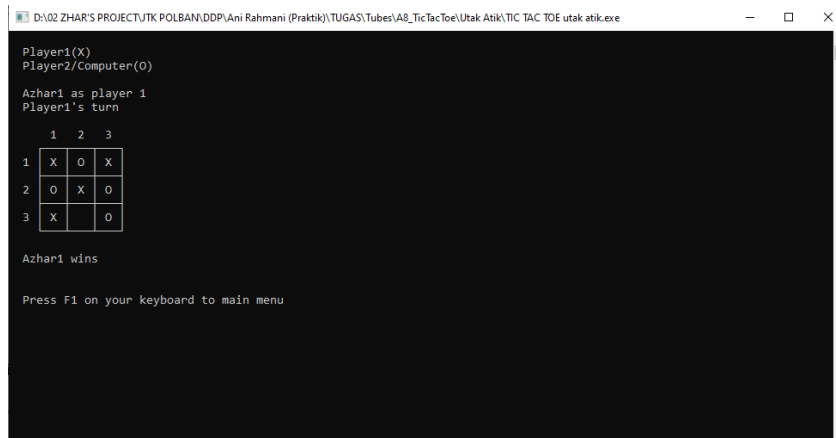
```

Gambar 27 Hasil implementasi tampilan *Gameplay*

- **End-Game**

- a) **Tampilan Result**

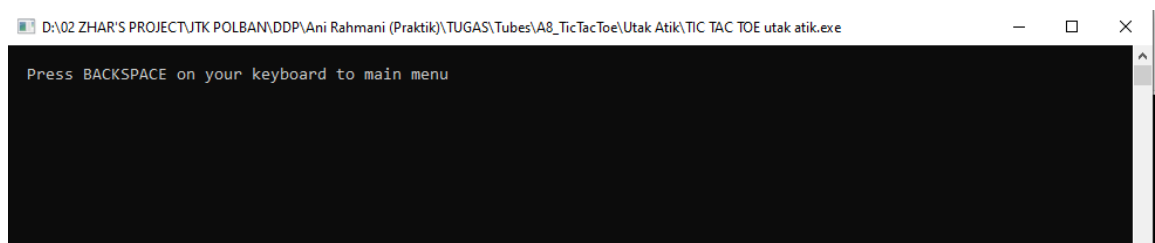
Berikut tampilan hasil permainan setelah kemudian *player* maupun komputer menempatkan tandanya sesuai kondisi berakhir. Pada gambar di bawah ditampilkan *player* 1 memenangkan permainan dengan berhasil menempatkan 3 tanda berturut-turut dan juga ditampilkan nama sesuai yang di-*inputkan*. Jika permainan sudah selesai, maka *player* bisa kembali ke main menu dengan menekan F1. Jika menekan selain F1, maka tidak akan kembali ke main menu (OKE)



Gambar 28 Hasil implementasi tampilan Result

- b) **Tampilan Exit**

Jika *player* memilih 6 pada main menu, maka *player* akan keluar dari game seperti yang ditampilkan pada tampilan berikut. Dan jika *player* menekan BACKSPACE, maka akan dikembalikan ke main menu. Jika menekan selain BACKSPACE, maka tidak akan kembali ke main menu (OKE)



Gambar 29 Hasil implementasi tampilan Exit

### 3. Petunjuk penggunaan

- bahasa Indonesia

Apa ini?

Game ini bernama tic tac toe. Dalam game ini, terdapat 2 komponen utama yaitu papan permainan yang terdiri dari 3 ukuran (3x3, 5x5, 7x7) dan juga *player/bot* yang dapat

menempatkan tanda 'X' atau 'O' di papan permainan

Seperti yang anda lihat, pada bagian main menu terdapat beberapa pilihan :

1. Start untuk memulai game
2. *Help* untuk menampilkan deskripsi game dan juga cara bermainnya
3. *Highscore* untuk menampilkan riwayat kemenangan melawan komputer
4. Color Theme untuk memilih background dan text color pada permainan
5. Credits untuk menampilkan identitas *developer* atau *creator* dari permainan
6. Exit untuk keluar dari game

Bagaimana cara bermainnya?

1. Silakan pilih start pada main menu
2. Setelah itu, pilih ukuran *grid* permainan (3x3, 5x5, 7x7)
3. Pilih lawan, bisa melawan *player 2* dan juga komputer
4. *Inputkan* nama anda. Jika melawan *player 2*, maka *inputkan* juga namanya
5. Pilih tingkat kesulitan dari komputer
6. Setelah itu, pilih siapa yang mendapatkan giliran pertama main
7. Perlu diingat bahwa setiap giliran dibatas dengan 10 detik
8. *Player 1* memiliki tanda 'X', sedangkan *player 2* atau komputer memiliki tanda 'O'
9. Setiap *player* atau komputer dapat mengisi tandanya masing-masing di papan permainan
10. *Player/bot* dikatakan menang, jika sudah mengisi tandanya sebanyak 3 kali berturut-turut pada *grid* 3x3 atau 4 kali berturut-turut pada *grid* 5x5 dan/atau 5 kali berturut-turut pada *grid* 7x7
11. Setiap ronde selesai, *player* dapat menentukan apakah mau melanjutkan permainan atau tidak
12. Setiap kemenangan melawan komputer, maka *player* dapat melihat riwayatnya di menu *highscore*

- bahasa Inggris

What is this?

So, this game is called tic-tac-toe. In this game, there's a board game in the form of *grid* (3x3, 5x5, 7x7) and two *players*, X and O, who take *turns* marking spaces ' ' in the board game. The *player* who takes the *turn* can mark in the row, column or diagonal side in the board game.

Also, as you can see in main menu, there's a couple of choices :

1. Start to start the game.
2. *Help* to display game descriptions, rules and how to play.
3. *Highscore* to display the history of the *player's* victory against the *computer*.
4. Color Theme to select the display color of this game.
5. Credits for displaying the identity of the *developer* or *creator* of this game.
6. Exit to exit the game.

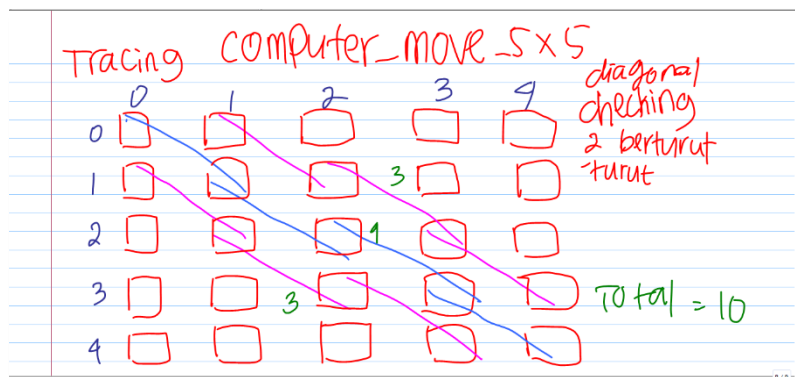
How to play this game? :

1. Select Start.
2. Select the *size* of the game board (3x3, 5x5, 7x7).
3. Choose who you want to play with (*Player/Computer*).
4. Enter the *player name*.
5. Choose a *difficulty level*.
6. Choose who will take the first *turn*.
7. Each *turn* is limited to 10 seconds.
8. You are 'X', your friend (or the *computer* in this case) is 'O'. *Players* take *turns* putting their marks in empty squares.
9. The first *player* to get all of her marks in a row (up, down, across, or diagonally) is the winner.
10. When all squares are full, the game is over. If no *player* has won, the game ends in a tie.

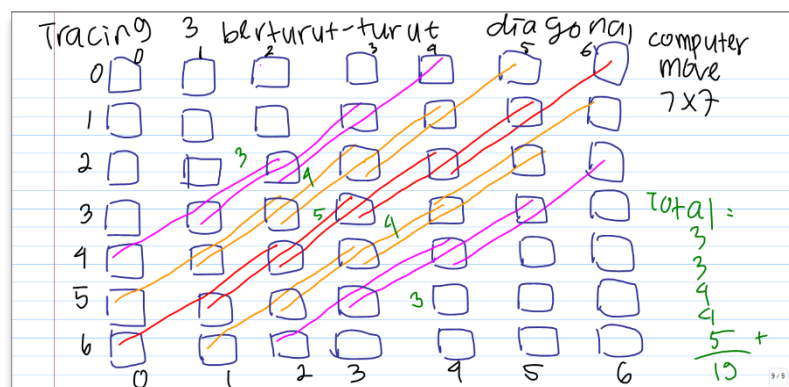
## BAB IV : PENUTUP

### 1. Dokumentasi

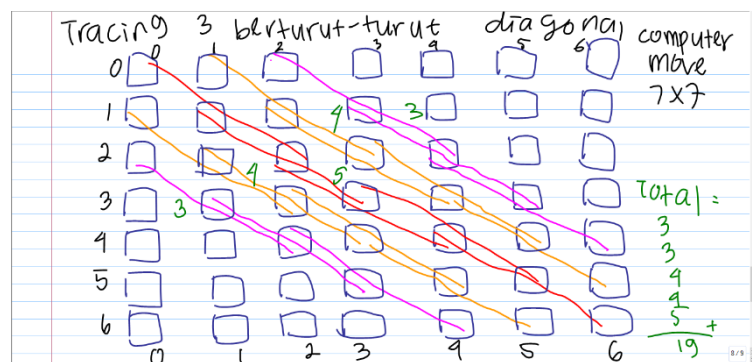
Berikut kami tampilkan beberapa hasil tracing dari pergerakan komputer dalam masing-masing ukuran *grid* nya sebelum kemudian mengimplementasikannya.



Gambar 30 tracing computer\_move\_5x5



Gambar 31 tracing computer\_move\_7x7



Gambar 32 tracing computer\_move\_7x7

Selengkapnya:

[https://drive.google.com/drive/folders/1DfFFqLDxrD4pDa\\_JJPK8fS9Gj9AuI3W7?usp=sharing](https://drive.google.com/drive/folders/1DfFFqLDxrD4pDa_JJPK8fS9Gj9AuI3W7?usp=sharing)

## 2. Timeline Pengerjaan

Pada awalnya, kami merencanakan dan juga merancang segala hal terkait dengan permainan apa yang akan dibuat secara umum. Setelah dirancang, maka kami mulai menentukan kebutuhan apa saja yang kira-kira dibutuhkan di implementasinya. Setelah ditentukan apa saja yang dibutuhkan, maka kami lanjut untuk menentukan modul-modul apa saja yang harus ada baik itu berupa prosedur ataupun function.

Setelah kemudian modulnya ditentukan, maka kami melanjutkan untuk mulai terjun ke implementasi disertai dengan datacase dan juga tracing. Seiring berjalannya waktu, kami juga saling me-review bersama, konsultasi dan juga mengoreksi apabila ada kesalahan. Setelah dirasa semuanya hampir atau bahkan selesai, maka kami mulai merancang untuk laporan dan juga bahan presentasi kami.

Berikut timeline singkat kami yang menggambarkan alur pengerjaan dari tugas besar ini :

- a. Akhir Desember 2020 : Penerimaan tugas besar
- b. Awal Januari 2021 : Merancang dan merencanakan permainan yang akan dibuat
- c. Pertengahan Januari 2021 : Mulai membuat program permainan
- d. Akhir Januari 2021 : Konsultasi dan perbaikan konsep dan juga program permainan
- e. 1 - 15 Februari 2021 : Finalisasi program permainan dan juga konsultasi
- f. 15 - 16 Februari 2021 : Testing program dan menyelesaikan laporan dan juga bahan untuk presentasi

## 3. Ringkasan Kontribusi Setiap Anggota

M Azhar Alauddin	Difa Mochammad Fadillah
Membuat keseluruhan modul, kecuali modul main	Mencari kebutuhan referensi permainan tic-tac-toe dari yang sudah ada di internet dan lainnya
Pada laporan mengerjakan bab 2, <i>structure</i>	Pada laporan mengerjakan bab 1, bab 3, <i>flow</i>



<i>chart</i> , dan sketsa skenario	<i>chart</i> , dan hasil akhir program
Me-review segala pengerjaan yang sudah dibuat	Membuat PPT untuk keperluan presentasi
Mengerjakan tracing dan data testing dari setiap implementasi program yang sudah dibuat	Membuat modul main pada implementasi program

#### 4. Kelebihan dan Kekurangan

Kelebihan	Kekurangan
Menu highscore untuk menyimpan dan menampilkan riwayat kemenangan player dalam melawan computer dalam segala level dan ukuran grid	Untuk pergerakan bot pada grid 7x7, setiap 2 kali tanda 'X' berturut-turut, hanya terdapat pengecekan untuk baris dan kolom saja
Timer selama 10 detik dari setiap turn/giliran dari setiap player, kecuali komputer	Source code terdiri dari 5500++ baris (kurang efektif)
Beberapa pilihan warna yang dapat dipilih untuk teks dan background permainan	Pergerakan computer menggunakan kondisi manual (tidak menggunakan mini max)
Level bot terdiri dari 3 level, yaitu easy, normal, dan hard	Fitur highscore mengambil dari referensi (tidak murni buatan sendiri)
Board game menggunakan table (lebih rapih)	

Gambar 33 kelebihan dan kekurangan program

#### 5. Lesson Learned

M Azhar Alauddin	Difa Mochammad Fadillah
<p>Bagi saya pribadi dalam seluruh rangkaian pengerjaan tugas besar ini baik itu pengerjaan skenario, implementasi maupun laporan, pelajaran besar yang bisa diambil adalah sesuatu hal besar pasti dimulai dari proses dan langkah yang kecil yang disertai dengan keseriusan dan konsisten. Awalnya, program ini terasa berat dan panjang, terutama ketika mengerjakan implementasi program untuk pergerakan komputer pada <i>grid</i> 7x7.</p> <p>Tetapi karena saya yakin bahwa hal itu akan selesai bila saya mau memulai dan juga melanjutkan sedikit demi sedikit mulai dari</p>	<ol style="list-style-type: none"> <li>1. Memahami pentingnya kerangka sebelum mengimplentasikan,</li> <li>2. Memahami <i>structure chart</i></li> <li>3. Memahami <i>source code</i> yang terstruktur.</li> <li>4. <i>Requirement</i> sangat penting, dimana di sana terletak informasi dan batasan - batasan.</li> <li>5. Lebih memahami penggunaan <i>function</i> dengan definisi, deklarasi, parameter.</li> </ol>

<p>tracing dan juga membuat implementasi programnya sehingga saya terus lanjutkan setiap ada waktu dan akhirnya semua programnya selesai dengan lebih dari 5400 baris. Dari sini, saya mengerti betul bahwa hal yang paling utama adalah kita harus memulai dan konsisten.</p>	
--	--