

Aplikasi Permainan “*Tic-Tac-Toe*” dengan Menggunakan *Finite Automata*

**LAPORAN TUGAS BESAR I IF2220
TEORI BAHASA FORMAL DAN OTOMATA**



Disusun oleh:
Johanes - 13517012

**PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
BANDUNG
2018**

DAFTAR ISI

KATA PENGANTAR	3
DESKRIPSI PERSOALAN	4
NOTASI DFA	5
PENJELASAN STATE DAN AKSI	8
SOURCE CODE	11
CONTOH INPUT/OUTPUT PROGRAM	23
REFERENSI	25

KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Esa yang atas segala rahmat dan berkat-Nya, laporan tugas besar ini dapat diselesaikan.

Laporan ini dibuat untuk memenuhi salah satu spesifikasi Tugas Besar I dari mata kuliah IF2220 Teori Bahasa Formal dan Otomata selain pembuatan program.

Dalam tugas besar ini penulis diminta untuk membuat membuat sebuah program yang dapat bermain *tic-tac-toe* dan tidak akan pernah kalah saat bermain dengan manusia. Dengan membuat tugas besar ini penulis mendapatkan pemahaman mengenai *Finite Automata* dan menambah pengetahuan mengenai proses pemrograman dengan *Finite Automata*.

Akhir kata, semoga laporan ini dapat bermanfaat bagi pembaca dan penulis dan tidak hanya menjadi sarana pemenuhan tugas saja.

DESKRIPSI PERSOALAN

Tic-tac-toe adalah sebuah permainan untuk dua orang yang secara bergiliran saling membuat huruf X dan O didalam sebuah kotak 3 x 3. Pemenang dari permainan ini adalah pemain pertama yang berhasil membuat tiga tanda 'X' atau tanda 'O' berurutan di dalam suatu permainan. Kondisi menang adalah saat seorang pemain membuat tiga tanda pada posisi horizontal, vertikal atau diagonal. Permainan juga berakhir ketika papan sudah terisi penuh dan hasil permainan adalah imbang.

Pada Tugas Besar I ini, penulis diminta membuat program yang mensimulasikan permainan *tic-tac-toe* dengan pemainnya adalah seorang *user*(pemain) dan sebuah komputer. Program dibuat sedemikian rupa dengan menggunakan *Finite Automata* yang terdiri atas banyak *state* dan dipastikan bahwa komputer tidak akan pernah kalah dari *user*. Komputer akan membaca *state* dari *file external* yang mendaftarkan semua jalan paling optimum dalam memainkan *tic-tac-toe*.

Adapula batasan dalam masalah ini adalah, siapapun yang jalan dahulu, baik komputer maupun *user* akan selalu mulai di kotak tengah. *User* akan selalu memainkan tanda 'X' sedangkan komputer akan selalu memainkan tanda 'O'. Simulasi permainan ini hanya akan dijalankan satu kali untuk setiap kali penjalanan program dan pada akhir program akan selalu dimunculkan *state-state* apa saja yang telah dilewati selama permainan. Program ini dibuat sehingga komputer tidak akan pernah kalah dan yang terkakhir, tidak diperkenankan menggunakan librari yang berkaitan dengan *finite automata* yang sudah ada.

NOTASI DFA

Berikut ini adalah tabel transisi setiap *state* dalam program ini:

Tabel 1. Tabel transisi lengkap setiap *state* dari program *tic-tac-toe*

State	1	2	3	4	5	6	7	8	9
q0	q25	q26	q27	q28	q1	q28	q30	q26	q32
q1	q43	q2	q3	q4	q43	q5	q6	q7	q6
q2	q43	q43	q8	q9	q43	q10	q11	q43	q12
q3	q43	q13	q43	q15	q43	q13	q43	q13	q13
q4	q43	q9	q15	q43	q43	q43	q17	q18	q19
q5	q43	q20	q20	q43	q43	q43	q17	q20	q20
q6	q43	q11	q43	q14	q43	q14	q14	q14	q14
q7	q43	q43	q33	q16	q43	q16	q16	q43	q16
q8	q43	q43	q43	q22	q43	q21	q43	q43	q21
q9	q43	q43	q23	q43	q43	q43	q23	q43	q23
q10	q43	q43	q20	q43	q43	q43	q23	q43	q20
q11	q43	q43	q43	q23	q43	q23	q23	q43	q23
q12	q43	q43	q21	q24	q43	q21	q43	q43	q43
q13	q43	q43	q43	q43	q43	q43	q43	q43	q43
q14	q43	q43	q43	q43	q43	q43	q43	q43	q43
q15	q43	q23	q43	q43	q43	q43	q43	q23	q24
q16	q43	q43	q43	q43	q43	q43	q43	q43	q43

q17	q43	q23	q43	q43	q43	q43	q43	q14	q14
q18	q43	q43	q23	q43	q43	q43	q16	q43	q16
q19	q43	q24	q43	q23	q43	q21	q14	q14	q21
q20	q43	q43	q43	q43	q43	q43	q43	q43	q43
q21	q43	q43	q43	q43	q43	q43	q43	q43	q43
q22	q43	q43	q43	q43	q43	q43	q43	q43	q43
q23	q43	q43	q43	q43	q43	q43	q43	q43	q43
q24	q43	q43	q43	q43	q43	q43	q43	q43	q43
q25	q43	q20	q43	q20	q43	q20	q34	q20	q20
q26	q20	q20	q43	q20	q43	q20	q35	q20	q20
q27	q43	q22	q43	q22	q43	q22	q22	q22	q36
q28	q43	q22	q22	q22	q43	q22	q22	q22	q37
q29	q43	q43	q43	q43	q43	q43	q43	q43	q43
q30	q43	q22	q22	q22	q43	q22	q43	q22	q38
q31	q43	q43	q43	q43	q43	q43	q43	q43	q43
q32	q20	q20	q43	q20	q43	q20	q39	q20	q43
q33	q43	q43	q43	q23	q43	q21	q43	q43	q21
q34	q43	q29	q43	q43	q43	q40	q43	q29	q29
q35	q29	q29	q43	q29	q43	q31	q43	q31	q43
q36	q43	q21	q43	q40	q43	q43	q21	q21	q43
q37	q43	q20	q43	q20	q43	q20	q14	q14	q43

q38	q43	q41	q14	q14	q43	q14	q43	q43	q43
q39	q14	q42	q43	q14	q43	q14	q43	q43	q43
q40	q43	q23	q43	q43	q43	q43	q14	q43	q14
q41	q43	q43	q13	q24	q43	q43	q43	q43	q43
q42	q29	q43	q43	q43	q43	q24	q43	q43	q43
q43	q43	q43	q43	q43	q43	q43	q43	q43	q43

Setiap *state* dituliskan dengan format ‘qx’ dengan x adalah angka yang membedakan setiap *state* dan ‘q’ yang merupakan konvensi umum untuk simbol *state*. Cara membaca tabel di atas adalah, kolom paling kiri melambangkan *state* yang ada lalu pada kolom-kolom setelahnya melambangkan *state* yang akan dicapai dari *state* asal jika di-*input* dengan bilangan. Misalkan, *state* awal adalah ‘q0’ lalu di-*input* bilangan 1 maka akan masuk ke dalam *state* ‘q25’ dan jika di-*input* 2 maka akan masuk ke *state* ‘q20’. Setiap kotak yang bukan *state* dan berwarna, berarti selanjutnya akan masuk ke *final state*, baris yang semua kolomnya berwarna sama melambangkan *final state*. *State* yang berwarna kuning melambangkan permainan berakhir dengan komputer sebagai pemenangnya, dan yang berwarna jingga melambangkan permainan berakhir dengan seri.

Posisi pada papan permainan dibagi menjadi sembilan kotak dengan konvensi sebagai berikut, kotak pada sudut kiri atas adalah kotak 1, 2, dan 3, kemudian baris selanjutnya dilanjutkan dengan kotak 4, 5, dan 6, dan baris terakhir adalah petak 7, 8, dan 9, yang dihitung dari kotak paling kiri.

1	2	3
4	5	6
7	8	9

Gambar 1. Representasi papan permainan

Sehingga, jika pemain memasukkan *input* 5, maka akan ada tanda silang di kotak 5 dan juga seterusnya.

PENJELASAN STATE DAN AKSI

State-state yang digunakan dalam pembuatan tugas kali ini ditulis dengan format 'qx' dengan x melambangkan urutan *state*-nya. *State* akan dimulai dari 'q0' dan akan berakhir di salah satu *final state*. Berikut ini adalah daftar *final state*-nya.

Tabel 2. *List Final State*

q13	q14	q16	q20	q21	q22	q23	q24	q29	q31
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Final state yang berwarna kuning melambangkan *state* saat permainan berakhir dan komputer menang, sedangkan warna jingga melambangkan saat permainan berakhir dan permainan berakhir imbang. Berikut ini adalah list definisi setiap *state* yang digunakan.

q0 start_state

q1 menambah 'o' di 1 dan 'x' di input user

q2 menambah 'o' di 8 dan 'x' di input user

q3 menambah 'o' di 7 dan 'x' di input user

q4 menambah 'o' di 6 dan 'x' di input user

q5 menambah 'o' di 4 dan 'x' di input user

q6 menambah 'o' di 3 dan 'x' di input user

q7 menambah 'o' di 2 dan 'x' di input user

q8 menambah 'o' di 7 dan 'x' di input user

q9 menambah 'o' di 6 atau 8 dan 'x' di input user

q10 menambah 'o' di 4 dan 'x' di input user

q11 menambah 'o' di 3 atau 8 dan 'x' di input user

q12 menambah 'o' di 7 dan 'x' di input user

q13 menambah 'o' di 4 dan 'x' di input user

q14 menambah 'o' di 2 dan 'x' di input user

q15 menambah 'o' di 7 atau 6 dan 'x' di input user

q16 menambah 'o' di 3 dan 'x' di input user

q17 menambah 'o' di 3 dan 'x' di input user

q18 menambah 'o' di 2 dan 'x' di input user

q19 menambah 'o' di 3 dan 'x' di input user

q20 menambah 'o' di 7 dan 'x' di input user

q21 menambah 'o' di 4 dan 'x' di input user

q22 menambah 'o' di 9 dan 'x' di input user

q23 menambah 'x' di input user dan

'o' di tempat kosong pertama di matriks dan

'x' di tempat kosong selanjutnya di matriks,

jika matriks tidak penuh

q24 menambah 'x' di input user dan

'o' di tempat kosong terakhir di matriks dan

'x' di tempat kosong sebelumnya di matriks,

jika matriks tidak penuh

q25 menambah 'o' di 5 dan 3 dan 'x' di input user
 q26 menambah 'o' di 5 dan 3 dan 'x' di input user
 q27 menambah 'o' di 5 dan 1 dan 'x' di input user
 q28 menambah 'o' di 5 dan 1 dan 'x' di input user
 q29 menambah 'o' di 6 dan 'x' di input user
 q30 menambah 'o' di 5 dan 1 dan 'x' di input user
 q31 menambah 'o' di 1 dan 'x' di input user
 q32 menambah 'o' di 5 dan 3 dan 'x' di input user
 q33 menambah 'o' di 7 dan 'x' di input user
 q34 menambah 'o' di 1 dan 'x' di input user
 q35 menambah 'o' di 9 dan 'x' di input user
 q36 menambah 'o' di 6 dan 'x' di input user
 q37 menambah 'o' di 3 dan 'x' di input user
 q38 menambah 'o' di 8 dan 'x' di input user
 q39 menambah 'o' di 8 dan 'x' di input user
 q40 menambah 'o' di 8 dan 'x' di input user
 q41 menambah 'o' di 6 dan 'x' di input user
 q42 menambah 'o' di 4 dan 'x' di input user
 q43 state buangan

Aksi-aksi yang dilakukan oleh program adalah:

1. Program membaca *start state* lalu menyimpannya ke suatu variabel global bernama *start_state*, *list final state* yang kemudian disimpan ke dalam array bernama *final_state*, dan membaca *transition table* yang disimpan ke dalam matriks bernama *transition* dari file external bernama "informasi.txt"
2. Selanjutnya program membuat sebuah matriks kosong bertipe karakter yang akan menyimpan keadaan papan permainan *tic-tac-toe*
3. Lalu program meminta masukkan dari pengguna untuk menyatakan pengguna jalan dahulu atau komputer dahulu, jika pengguna memasukan masukkan 5 maka pengguna jalan dahulu, jika masukkannya selain 5 maka komputer jalan dahulu dan komputer akan selalu jalan di petak 5 terlebih dahulu
4. Program akan mengecek apakah masukkan yang dimasukan sudah benar, yaitu dengan membandingkan masukkan dengan keadaan papan. Jika pada petak yang dimasukan, nilai elemennya masih kosong, maka program akan lanjut. Jika tidak maka program akan meminta pengguna untuk memasukan nilai yang lain
5. Program akan mencari *state* selanjutnya dengan membaca *current_state* dan kolom sesuai masukkan pengguna. Lalu mengeksekusi prosedur sesuai definisi setiap *state* di atas lalu menampilkan hasil papan ke layar
6. Aksi 3-5 akan terus diulang hingga *current_state* mencapai *final state*
7. Di akhir program, program akan menampilkan daftar *state* yang sudah pernah dilewati. Mulai dari *start state* hingga *final state*.

Berikut ini adalah beberapa asumsi yang digunakan dalam program ini:

1. Jika ada keadaan papan yang saat pemain memasukkan pilihannya dan menyebabkan permainan sudah pasti akan berakhir seri dengan kondisi papan yang sudah dapat diprediksi, maka permainan langsung selesai, Contohnya:

O		X
X	X	O
O		

Gambar 2. Keadaan papan yang pasti seri

Pada kondisi papan seperti gambar diatas, jika pengguna memasukkan di 2, maka komputer akan jalan di 8 dan pengguna pasti jalan di 9 sehingga oleh komputer permainan akan langsung diakhiri

O	X	X
X	X	O
O	O	X

Gambar 3. Keadaan papan saat seri

2. masukkan dari pengguna pasti selalu satu karakter. Masukkan boleh berupa angka atau karakter namun hanya dibatasi 1 karakter per masukkan

SOURCE CODE

Berikut ini adalah isi dari tictactoe.h

```
//adt tictactoe
#ifndef TICTACTOE_H
#define TICTACTOE_H

#include "boolean.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

typedef struct {
    char mem[4][4];
} tictactoe; //tipe data tictactoe
//merupakan suatu tipe data array yang berisi karakter 'x', 'o', atau ' '
(spasi)

#define maxLength 52 //panjang string maximal dari setiap string
#define state 44 //jumlah state
#define symbol 9 //jumlah symbol

extern char transition[state][symbol+1][maxLength]; //transition table
extern char start_state[5]; //state awal;
extern char current_state[5]; //state sekarang
extern char final_state[13][5]; //list final states
extern char storage[9][5];
extern int counter;

#define elmt(m,i,j) (m).mem[(i)][(j)]

void baru(tictactoe *m);
//membuat tictactoe baru yang isinya ' ' (spasi) pada semua elemennya

void tulis(tictactoe m);
//prosedur menampilkan tictactoe ke layar

void isi(tictactoe *m, int i, int j, char input);
//prosedur mengisi tabel tictactoe

void ubah_input(int *i, int *j, char x);
//prosedur yang mengubah char x menjadi indeks pada array

boolean win_state();
//fungsi yang menyatakan final state sebagai menang atau draw

void cek_finish(tictactoe m,boolean *stop, boolean *win);
//mengecek apakah sudah state final dan apakah menang

void olah_state(tictactoe *m, char input);
//prosedur yang menjalankan giliran komputer berdasarkan input dan state
sekarang

void loading(tictactoe *m,char input, boolean *stop, boolean *win);
//prosedur yang merupakan proses berpikirnya komputer untuk mengintepretasi
//kondisi yang ada
```

```

void mulai();
//prosedur yang dilakukan saat awal program berjalan
//yaitu membaca informasi dari file external
void delta(char input);
//merupakan transition function untuk dfa
//menerima input current_state dan mengembalikan state selanjutnya

#endif

```

Berikut ini adalah isi dari file tictactoe.c

```

//implementasi tictactoe.h
#include "tictactoe.h"

char transition[state][symbol+1][maxLength]; //transition table
char start_state[5]; //state awal;
char current_state[5]; //state sekarang
char final_state[13][5]; //list final states
char storage[9][5];
int counter;

void baru(tictactoe *m)
//membuat tictactoe baru yang isinya ' ' (spasi) pada semua elemennya
{
    //kamus
    int i,j;
    //algoritma
    for (i=1;i<=3;i++) {
        for (j=1;j<=3;j++) {
            elmt(*m,i,j) = ' '; //mengisi setiap elemen dengan spasi
        }
    }
    counter =0;
}

void tulis(tictactoe m)
//prosedur menampilkan tictactoe ke layar
{
    //kamus
    int i,j;
    //algoritma
    for (i=1;i<=3;i++) {
        for (j=1;j<=3;j++) {
            printf("%c",elmt(m,i,j)); //menuliskan elemen tiap kolom ke samping
            if (j!=3) printf("|");
        }
        printf("\n"); //pindah baris baru untuk baris berikutnya
        if (i!=3) printf("~~~~~\n");
    }
}

void isi(tictactoe *m, int i, int j, char input)
//prosedur mengisi tabel tictactoe
{
    elmt(*m,i,j) = input;
}

void ubah_input(int *i, int *j, char x)
//prosedur yang mengubah char x menjadi indeks pada array

```

```

{
    //algoritma
    switch (x) {
        case '1' ... '3' :
            *i = 1;
            *j = (int) (x-'0');
            break;
        case '4' ... '6' :
            *i = 2;
            *j = (int) (x-'0') - 3;
            break;
        case '7'...'9' :
            *i = 3;
            *j = (int) (x-'0') -6;
            break;
        default : *i = *j = 0;
    }
}

boolean win_state()
//fungsi yang menyatakan final state sebagai menang atau draw
{
    //kamus
    boolean tmp;
    //algoritma
    tmp = (strcmp("q23",current_state)==0) ||
(strcmp("q24",current_state)==0);
    return !tmp;
}

void cek_finish(tictactoe m,boolean *stop, boolean *win)
//mengecek apakah sudah state final dan apakah menang
{
    //kamus
    int i;
    //algoritma
    i = 0;
    while (i<10 && strcmp(final_state[i],current_state)!=0) {
        i++;
    }/i==15 || final_state[i]==state
    *stop = (strcmp(final_state[i],current_state)==0);
    if (*stop) {
        *win = win_state(current_state);
    }
}

void olah_state(tictactoe *m, char input)
//prosedur yang menjalankan giliran komputer berdasarkan state sekarang
{
    //kamus
    int i,j;
    //algoritma
    delta(input);
    //pendefinisian state
    if (strcmp("q1",current_state)==0) {
        ubah_input(&i,&j,'1');
        isi(m,i,j,'o');
        ubah_input(&i,&j,input);
        isi(m,i,j,'x');
    } else if (strcmp("q2",current_state)==0) {
        ubah_input(&i,&j,'8');

```

```

isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q3",current_state)==0) {
ubah_input(&i,&j,'7');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q4",current_state)==0) {
ubah_input(&i,&j,'6');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q5",current_state)==0) {
ubah_input(&i,&j,'4');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q6",current_state)==0) {
ubah_input(&i,&j,'3');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q7",current_state)==0) {
ubah_input(&i,&j,'2');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q8",current_state)==0) {
ubah_input(&i,&j,'7');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q9",current_state)==0) {
ubah_input(&i,&j,'6');
if (elmt(*m,i,j)=='o') ubah_input(&i,&j,'8');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q10",current_state)==0) {
ubah_input(&i,&j,'4');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q11",current_state)==0) {
ubah_input(&i,&j,'3');
if (elmt(*m,i,j)=='o') ubah_input(&i,&j,'8');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q12",current_state)==0) {
ubah_input(&i,&j,'7');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q13",current_state)==0) {
ubah_input(&i,&j,'4');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q14",current_state)==0) {

```

```

    ubah_input(&i,&j,'2');
    isi(m,i,j,'o');
    ubah_input(&i,&j,input);
    isi(m,i,j,'x');
} else if (strcmp("q15",current_state)==0) {
    ubah_input(&i,&j,'7');
    if (elmt(*m,i,j)=='o') ubah_input(&i,&j,'6');
    isi(m,i,j,'o');
    ubah_input(&i,&j,input);
    isi(m,i,j,'x');
} else if (strcmp("q16",current_state)==0) {
    ubah_input(&i,&j,'3');
    isi(m,i,j,'o');
    ubah_input(&i,&j,input);
    isi(m,i,j,'x');
} else if (strcmp("q17",current_state)==0) {
    ubah_input(&i,&j,'3');
    isi(m,i,j,'o');
    ubah_input(&i,&j,input);
    isi(m,i,j,'x');
} else if (strcmp("q18",current_state)==0) {
    ubah_input(&i,&j,'2');
    isi(m,i,j,'o');
    ubah_input(&i,&j,input);
    isi(m,i,j,'x');
} else if (strcmp("q19",current_state)==0) {
    ubah_input(&i,&j,'3');
    isi(m,i,j,'o');
    ubah_input(&i,&j,input);
    isi(m,i,j,'x');
} else if (strcmp("q20",current_state)==0) {
    ubah_input(&i,&j,'7');
    isi(m,i,j,'o');
    ubah_input(&i,&j,input);
    isi(m,i,j,'x');
} else if (strcmp("q21",current_state)==0) {
    ubah_input(&i,&j,'4');
    isi(m,i,j,'o');
    ubah_input(&i,&j,input);
    isi(m,i,j,'x');
} else if (strcmp("q22",current_state)==0) {
    ubah_input(&i,&j,'9');
    isi(m,i,j,'o');
    ubah_input(&i,&j,input);
    isi(m,i,j,'x');
} else if (strcmp("q23",current_state)==0) {
    ubah_input(&i,&j,input);
    isi(m,i,j,'x');
    char x='1';
    ubah_input(&i,&j,x);
    while (x!='9' && elmt(*m,i,j)!=' ') {
        x++;
        ubah_input(&i,&j,x);
    } //x=='9' || elmt(*m,i,j)==' '
    isi(m,i,j,'o');
    while (x!=':' && elmt(*m,i,j)!=' ') {
        x++;
        ubah_input(&i,&j,x);
    } //x=='9' || elmt(*m,i,j)==' '
    if (x!=':') isi(m,i,j,'x');
} else if (strcmp("q24",current_state)==0) {

```

```

ubah_input(&i,&j,input);
isi(m,i,j,'x');
char x='9';
ubah_input(&i,&j,x);
while (x!='1' && elmt(*m,i,j)!=' ') {
    x--;
    ubah_input(&i,&j,x);
} //x=='9' || elmt(*m,i,j)==' '
isi(m,i,j,'o');
while (x!='0' && elmt(*m,i,j)!=' ') {
    x--;
    ubah_input(&i,&j,x);
} //x=='9' || elmt(*m,i,j)==' '
if (x!='0') isi(m,i,j,'x');
} else if (strcmp("q25",current_state)==0) {
    ubah_input(&i,&j,'5');
    isi(m,i,j,'o');
    ubah_input(&i,&j,'3');
    isi(m,i,j,'o');
    ubah_input(&i,&j,input);
    isi(m,i,j,'x');
} else if (strcmp("q26",current_state)==0) {
    ubah_input(&i,&j,'5');
    isi(m,i,j,'o');
    ubah_input(&i,&j,'3');
    isi(m,i,j,'o');
    ubah_input(&i,&j,input);
    isi(m,i,j,'x');
} else if (strcmp("q27",current_state)==0) {
    ubah_input(&i,&j,'5');
    isi(m,i,j,'o');
    ubah_input(&i,&j,'1');
    isi(m,i,j,'o');
    ubah_input(&i,&j,input);
    isi(m,i,j,'x');
} else if (strcmp("q28",current_state)==0) {
    ubah_input(&i,&j,'5');
    isi(m,i,j,'o');
    ubah_input(&i,&j,'1');
    isi(m,i,j,'o');
    ubah_input(&i,&j,input);
    isi(m,i,j,'x');
} else if (strcmp("q29",current_state)==0) {
    ubah_input(&i,&j,'6');
    isi(m,i,j,'o');
    ubah_input(&i,&j,input);
    isi(m,i,j,'x');
} else if (strcmp("q30",current_state)==0) {
    ubah_input(&i,&j,'5');
    isi(m,i,j,'o');
    ubah_input(&i,&j,'1');
    isi(m,i,j,'o');
    ubah_input(&i,&j,input);
    isi(m,i,j,'x');
} else if (strcmp("q31",current_state)==0) {
    ubah_input(&i,&j,'1');
    isi(m,i,j,'o');
    ubah_input(&i,&j,input);
    isi(m,i,j,'x');
} else if (strcmp("q32",current_state)==0) {
    ubah_input(&i,&j,'5');

```



```

isi(m,i,j,'o');
ubah_input(&i,&j,'3');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q33",current_state)==0) {
ubah_input(&i,&j,'7');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q34",current_state)==0) {
ubah_input(&i,&j,'4');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q35",current_state)==0) {
ubah_input(&i,&j,'9');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q36",current_state)==0) {
ubah_input(&i,&j,'6');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q37",current_state)==0) {
ubah_input(&i,&j,'3');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q38",current_state)==0) {
ubah_input(&i,&j,'8');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q39",current_state)==0) {
ubah_input(&i,&j,'8');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q40",current_state)==0) {
ubah_input(&i,&j,'8');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q41",current_state)==0) {
ubah_input(&i,&j,'6');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
} else if (strcmp("q42",current_state)==0) {
ubah_input(&i,&j,'4');
isi(m,i,j,'o');
ubah_input(&i,&j,input);
isi(m,i,j,'x');
}
}

void loading(tictactoe *m,char input, boolean *stop, boolean *win)
//prosedur yang merupakan proses berfikirnya komputer untuk mengintepretasi
//kondisi yang ada

```

```

{
    //algoritma
    olah_state(m,input);
    cek_finish(*m,stop,win);
}

void mulai() {
    //kamus
    FILE *info;
    char *retval;
    int i,j;
    char tmp[maxLength],*parse;
    //algoritma
    info = fopen("informasi.txt","r"); //dibuka untuk dibaca
    for (i=1;i<=64;i++) retval = fgets(tmp,maxLength,info); //skip 86 baris
    pertama
    retval = fgets(tmp,maxLength,info); //baca start_state
    parse = strtok(tmp,"\n");
    strcpy(start_state,parse);
    for (i=1;i<=2;i++) retval = fgets(tmp,maxLength,info); //skip 2 baris
    //membaca final_state
    for (i=0;i<10;i++) {
        retval = fgets(tmp,maxLength,info);
        parse = strtok(tmp,"\n");
        strcpy(final_state[i],parse);
    }
    for (i=1;i<=2;i++) retval = fgets(tmp,maxLength,info); //skip 2 baris
    //membaca transition table
    for (i=0;i<state;i++){
        retval = fgets(tmp,maxLength,info);
        parse = strtok(tmp," \n");
        j = 0;
        strcpy(transition[i][j],parse); //ambil elemen pertama
        j++;
        for (j;j<symbol+1;j++) {
            parse = strtok(NULL," \n");
            strcpy(transition[i][j],parse); //elemen baris pertama
        }
    }

    fclose(info); //ditutup
}

void delta(char input) {
    //prosedur mengisi current_state dengan state selanjutnya berdasarkan
    input
    //kamus
    int i;
    char *first,tmp[4];
    //algoritma
    first = (&current_state[1]);
    strcpy(tmp,first);
    i = atoi(tmp);
    strcpy(current_state,transition[i][(int)(input-'0')]);
    strcpy(storage[counter],current_state);
    counter++;
}

```

Berikut ini adalah isi dari mainProgramTTT.c

```
//Tubes TBFO TicTacToe
//13517012/Johanes

#include "tictactoe.h"

int main() {
    //KAMUS
    int i,j,k;
    tictactoe m;
    char input[1],xo;
    boolean stop,win;
    //algoritma
    printf("\nTugas Besar TBFO\nMembuat robot Tic Tac Toe yang tidak dapat kalah\n");
    printf("dibuat oleh: Johanes/13517012\n");

    mulai(); //membaca flie external
    baru(&m); //membuat tictactoe baru
    stop = false;
    win = false;
    strcpy(current_state,start_state);
    strcpy(storage[counter],current_state);
    counter++;
    printf("\nKetik 5 untuk mulai dahulu atau\nketik angka yang lain (1..9) untuk komputer mulai dahulu\n");
    printf("komputer akan selalu mulai dari 5\n");
    printf("anda bertanda 'x', komputer bertanda 'o'\n");
    while(!stop) { //mengulang hingga kondisi berhenti
        printf("\nmasukan input anda: ");
        scanf("%s",input);
        ubah_input(&i,&j,input[0]);
        while (elmt(m,i,j)!=' ') { //validasi input
            printf("input salah, ulangi\n");
            printf("\nmasukan input anda: ");
            scanf("%s",input);
            ubah_input(&i,&j,input[0]);
        }
        loading(&m,input[0],&stop,&win); //menjalankan giliran komputer
        tulis(m); //menuliskan papan ke layar
        if (stop) { //jika sudah kondisi berhenti
            printf("\nHasil permainan: ");
            if(win) printf("ANDA KALAH!\n"); //jika menang
            else printf("DRAW\n"); //jika draw
        }
    }
    printf("\n");
    printf("state yang sudah dilewati:\n");
    for(int k=0;k<counter;k++) //menuliskan semua state yang pernah dilewati
        printf("%s\n",storage[k]);
    printf("\npress enter to continue.....\n"); //memastikan tidak langsung tertutup cmd nya saat diexecusi langsung
    char tahan;
    scanf("%c",&tahan);
    scanf("%c",&tahan);
    return 0;
}
```

Berikut ini adalah isi dari boolean.h

```
/* Definisi type boolean */

#ifndef _BOOLEAN_h
#define _BOOLEAN_h

#define boolean unsigned char
#define true 1
#define false 0

#endif
```

Berikut ini adalah isi dari informasi.txt

```
daftar states:
q0 start_state
q1 menambah 'o' di 1 dan 'x' di input user
q2 menambah 'o' di 8 dan 'x' di input user
q3 menambah 'o' di 7 dan 'x' di input user
q4 menambah 'o' di 6 dan 'x' di input user
q5 menambah 'o' di 4 dan 'x' di input user
q6 menambah 'o' di 3 dan 'x' di input user
q7 menambah 'o' di 2 dan 'x' di input user
q8 menambah 'o' di 7 dan 'x' di input user
q9 menambah 'o' di 6 atau 8 dan 'x' di input user
q10 menambah 'o' di 4 dan 'x' di input user
q11 menambah 'o' di 3 atau 8 dan 'x' di input user
q12 menambah 'o' di 7 dan 'x' di input user
q13 menambah 'o' di 4 dan 'x' di input user
q14 menambah 'o' di 2 dan 'x' di input user
q15 menambah 'o' di 7 atau 6 dan 'x' di input user
q16 menambah 'o' di 3 dan 'x' di input user
q17 menambah 'o' di 3 dan 'x' di input user
q18 menambah 'o' di 2 dan 'x' di input user
q19 menambah 'o' di 3 dan 'x' di input user
q20 menambah 'o' di 7 dan 'x' di input user
q21 menambah 'o' di 4 dan 'x' di input user
q22 menambah 'o' di 9 dan 'x' di input user
q23 menambah 'x' di input user dan
    'o' di tempat kosong pertama di matriks dan
    'x' di tempat kosong selanjutnya di matriks,
    jika matriks tidak penuh
q24 menambah 'x' di input user dan
    'o' di tempat kosong terakhir di matriks dan
    'x' di tempat kosong sebelumnya di matriks,
    jika matriks tidak penuh
q25 menambah 'o' di 5 dan 3 dan 'x' di input user
q26 menambah 'o' di 5 dan 3 dan 'x' di input user
q27 menambah 'o' di 5 dan 1 dan 'x' di input user
q28 menambah 'o' di 5 dan 1 dan 'x' di input user
q29 menambah 'o' di 6 dan 'x' di input user
q30 menambah 'o' di 5 dan 1 dan 'x' di input user
q31 menambah 'o' di 1 dan 'x' di input user
q32 menambah 'o' di 5 dan 3 dan 'x' di input user
q33 menambah 'o' di 7 dan 'x' di input user
q34 menambah 'o' di 1 dan 'x' di input user
q35 menambah 'o' di 9 dan 'x' di input user
q36 menambah 'o' di 6 dan 'x' di input user
```

```

q37 menambah 'o' di 3 dan 'x' di input user
q38 menambah 'o' di 8 dan 'x' di input user
q39 menambah 'o' di 8 dan 'x' di input user
q40 menambah 'o' di 8 dan 'x' di input user
q41 menambah 'o' di 6 dan 'x' di input user
q42 menambah 'o' di 4 dan 'x' di input user
q43 state buangan
~~~
daftar simbol:
1
2
3
4
5
6
7
8
9
~~~
state awal:
q0
~~~
state akhir:
q13
q14
q16
q20
q21
q22
q23
q24
q29
q31
~~~
transisition function:
q0 q25 q26 q27 q28 q1 q28 q30 q26 q32
q1 q43 q2 q3 q4 q43 q5 q6 q7 q6
q2 q43 q43 q8 q9 q43 q10 q11 q43 q12
q3 q43 q13 q43 q15 q43 q13 q43 q13 q13
q4 q43 q9 q15 q43 q43 q43 q17 q18 q19
q5 q43 q20 q20 q43 q43 q43 q17 q20 q20
q6 q43 q11 q43 q14 q43 q14 q14 q14 q14
q7 q43 q43 q33 q16 q43 q16 q16 q43 q16
q8 q43 q43 q43 q22 q43 q21 q43 q43 q21
q9 q43 q43 q23 q43 q43 q43 q23 q43 q23
q10 q43 q43 q20 q43 q43 q43 q23 q43 q20
q11 q43 q43 q43 q23 q43 q23 q23 q43 q23
q12 q43 q43 q21 q24 q43 q21 q43 q43 q43
q13 q43 q43 q43 q43 q43 q43 q43 q43 q43
q14 q43 q43 q43 q43 q43 q43 q43 q43 q43
q15 q43 q23 q43 q43 q43 q43 q43 q23 q24
q16 q43 q43 q43 q43 q43 q43 q43 q43 q43
q17 q43 q23 q43 q43 q43 q43 q43 q14 q14
q18 q43 q43 q23 q43 q43 q43 q16 q43 q16
q19 q43 q24 q43 q23 q43 q21 q14 q14 q21
q20 q43 q43 q43 q43 q43 q43 q43 q43 q43
q21 q43 q43 q43 q43 q43 q43 q43 q43 q43
q22 q43 q43 q43 q43 q43 q43 q43 q43 q43
q23 q43 q43 q43 q43 q43 q43 q43 q43 q43
q24 q43 q43 q43 q43 q43 q43 q43 q43 q43
q25 q43 q20 q43 q20 q43 q20 q34 q20 q20

```

q26	q20	q20	q43	q20	q43	q20	q35	q20	q20
q27	q43	q22	q43	q22	q43	q22	q22	q22	q36
q28	q43	q22	q22	q22	q43	q22	q22	q22	q37
q29	q43	q43	q43	q43	q43	q43	q43	q43	q43
q30	q43	q22	q22	q22	q43	q22	q43	q22	q38
q31	q43	q43	q43	q43	q43	q43	q43	q43	q43
q32	q20	q20	q43	q20	q43	q20	q39	q20	q43
q33	q43	q43	q43	q23	q43	q21	q43	q43	q21
q34	q43	q29	q43	q43	q43	q40	q43	q29	q29
q35	q29	q29	q43	q29	q43	q31	q43	q31	q43
q36	q43	q21	q43	q40	q43	q43	q21	q21	q43
q37	q43	q20	q43	q20	q43	q20	q14	q14	q43
q38	q43	q41	q14	q14	q43	q14	q43	q43	q43
q39	q14	q42	q43	q14	q43	q14	q43	q43	q43
q40	q43	q23	q43	q43	q43	q43	q14	q43	q14
q41	q43	q43	q13	q24	q43	q43	q43	q43	q43
q42	q29	q43	q43	q43	q43	q24	q43	q43	q43
q43	q43	q43	q43	q43	q43	q43	q43	q43	q43
~~~									

## CONTOH *INPUT/OUTPUT* PROGRAM

```
Tugas Besar TBFO
Membuat robot Tic Tac Toe yang tidak dapat kalah
dibuat oleh: Johanes/13517012

Ketik 5 untuk mulai dahulu atau
ketik angka yang lain (1..9) untuk komputer mulai dahulu
komputer akan selalu mulai dari 5
anda bertanda 'x', komputer bertanda 'o'

masukan input anda: 5
o| |
~~~~~
|x|
~~~~~
| |

masukan input anda: 9
o| |o
~~~~~
|x|
~~~~~
| |x

masukan input anda: 2
o|x|o
~~~~~
|x|
~~~~~
|o|x

masukan input anda: 6
o|x|o
~~~~~
o|x|x
~~~~~
x|o|x

Hasil permainan: DRAW

state yang sudah dilewati:
q0
q1
q6
q11
q23
```

Contoh *input output*, pemain jalan dahulu dan hasilnya seri

Tugas Besar TBFO

Membuat robot Tic Tac Toe yang tidak dapat kalah  
dibuat oleh: Johanes/13517012

Ketik 5 untuk mulai dahulu atau  
ketik angka yang lain (1..9) untuk komputer mulai dahulu  
komputer akan selalu mulai dari 5  
anda bertanda 'x', komputer bertanda 'o'

masukan input anda: 8

```
| |o
~~~~~
|o|
~~~~~
|x|
```

masukan input anda: 7

```
| |o
~~~~~
|o|
~~~~~
x|x|o
```

masukan input anda: 6

```
o| |o
~~~~~
|o|x
~~~~~
x|x|o
```

Hasil permainan: ANDA KALAH!

state yang sudah dilewati:

```
q0
q26
q35
q31
```

Contoh *input output*, komputer jalan dahulu dan hasilnya komputer menang



## **REFERENSI**

Wikipedia. (2018). Tic-tac-toe . Retrieved September, 2018, from <https://en.wikipedia.org/wiki/Tic-tac-toe>