

**TUGAS BESAR 1**  
**IF2220 – TEORI BAHASA FORMAL DAN OTOMATA**

**APLIKASI PERMAINAN “TIC TAC TOE” DENGAN MENGGUNAKAN FINITE  
AUTOMATA**

Disusun oleh  
Saskia Imani  
13517142

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

## A. DESKRIPSI PERMASALAHAN

*Tic-tac-toe* adalah sebuah permainan melibatkan dua orang yang secara bergiliran saling membuat huruf X dan O didalam sebuah kotak berukuran 3 x 3. Pemenang dari permainan ini adalah pemain pertama yang berhasil membuat tiga tanda “X” atau tanda “O” berurutan di dalam suatu permainan.

Pada tugas ini, dibuat sebuah permainan *tic-tac-toe* sederhana yang akan dimainkan oleh komputer dan seorang player. Program harus bisa memastikan bahwa komputer tidak mungkin kalah didalam permainan. Aplikasi akan membuka file yang berisi informasi mengenai daftar *state*, daftar simbol, *state* awal, *state* akhir, dan *transition function*. Informasi dari file tersebut akan digunakan untuk mengecek masukan dari pengguna. Program diwajibkan untuk membaca konfigurasi dari file eksternal, dan logika state machine tidak boleh di-*hardcode* ke program secara langsung.

## B. DFA

Pada program ini, digunakan sebuah DFA untuk menentukan transisi yang akan dialami oleh *state* program, dengan rancangan sebagai berikut:

	p	c	1	2	3	4	5	6	7	8	9
→ q000	q126	q001	----	----	----	----	----	----	----	----	----
q001	----	----	q002	q003	q004	q005	----	q006	q007	q008	q009
q002	----	----	----	q010	q011	q012	----	q013	q014	q015	----
q003	----	----	q016	----	q017	q018	----	q019	----	q020	q021
q004	----	----	q022	q017	----	q023	----	q024	----	q025	q026
q005	----	----	q027	q018	----	----	----	q028	q029	q030	q031
q006	----	----	----	q032	q033	q034	----	----	q035	q036	q037
q007	----	----	q038	q039	----	q029	----	q040	----	q041	q042
q008	----	----	----	q043	q044	q045	----	q036	q046	----	q047
q009	----	----	----	q048	q049	q050	----	q037	q051	q047	----
q010	----	----	----	----	----	q052	----	q053	q054	q055	----
q011	----	----	----	----	----	q056	----	q057	q058	q059	----
q012	----	----	----	q052	q060	----	----	q061	----	q062	----
q013	----	----	----	q053	q063	q061	----	----	----	q064	----
q014	----	----	----	q065	q066	----	----	q067	----	q068	----
q015	----	----	----	q055	----	q062	----	q064	q069	----	----
* q016	----	----	----	----	----	----	----	----	----	----	----
q017	----	----	----	----	----	q070	----	q071	----	q072	q073
* q018	----	----	----	----	----	----	----	----	----	----	----
* q019	----	----	----	----	----	----	----	----	----	----	----
* q020	----	----	----	----	----	----	----	----	----	----	----
* q021	----	----	----	----	----	----	----	----	----	----	----
q022	----	----	----	----	----	q074	----	q075	----	q059	q076

[illegible]

[illegible]

*	q103	----	----	----	----	----	----	-----	----	----	----	----
*	q104	----	----	----	----	----	----	-----	----	----	----	----
*	q105	----	----	----	----	----	----	-----	----	----	----	----
*	q106	----	----	----	----	----	----	-----	----	----	----	----
*	q107	----	----	----	----	----	----	-----	----	----	----	----
*	q108	----	----	----	----	----	----	-----	----	----	----	----
*	q109	----	----	----	----	----	----	-----	----	----	----	----
*	q110	----	----	----	----	----	----	-----	----	----	----	----
*	q111	----	----	----	----	----	----	-----	----	----	----	----
*	q112	----	----	----	----	----	----	-----	----	----	----	----
*	q113	----	----	----	----	----	----	-----	----	----	----	----
	q114	----	----	----	----	q124	-----	q125	----	----	----	----
*	q115	----	----	----	----	----	-----	----	----	----	----	----
*	q116	----	----	----	----	----	-----	----	----	----	----	----
*	q117	----	----	----	----	----	-----	----	----	----	----	----
*	q118	----	----	----	----	----	-----	----	----	----	----	----
*	q119	----	----	----	----	----	-----	----	----	----	----	----
*	q120	----	----	----	----	----	-----	----	----	----	----	----
*	q121	----	----	----	----	----	-----	----	----	----	----	----
*	q122	----	----	----	----	----	-----	----	----	----	----	----
*	q123	----	----	----	----	----	-----	----	----	----	----	----
*	q124	----	----	----	----	----	-----	----	----	----	----	----
*	q125	----	----	----	----	----	-----	----	----	----	----	----
	q126	----	----	q127	q128	q129	-----	q130	q131	q132	q133	
	q127	----	----	----	q134	q135	-----	q136	q137	----	q138	
	q128	----	----	q139	----	q140	-----	q141	----	q142	q143	
	q129	----	----	q135	q140	----	-----	----	q144	q145	q146	
	q130	----	----	q147	q141	----	-----	----	q148	q149	q150	
	q131	----	----	q137	----	q151	-----	q152	----	q153	q154	
	q132	----	----	----	q155	q156	-----	q157	q153	----	q158	
	q133	----	----	q159	----	q160	-----	q161	q154	q158	----	
	q134	----	----	----	----	q162	-----	q163	----	----	164	
	q135	----	----	----	q165	----	-----	----	q166	----	q166	
	q136	----	----	----	q163	----	-----	----	q167	----	q168	
	q137	----	----	----	----	q166	-----	q167	----	----	q167	
	q138	----	----	----	q164	q165	-----	q168	----	----	----	
*	q139	----	----	----	----	----	-----	----	----	----	----	
	q140	----	----	q165	----	----	-----	----	----	q169	q169	
*	q141	----	----	----	----	----	-----	----	----	----	----	
*	q142	----	----	----	----	----	-----	----	----	----	----	

*	q143	----	----	----	----	----	----	----	----	----	----
	q144	----	----	----	q170	----	----	----	----	q171	q172
	q145	----	----	----	----	q169	----	----	q171	----	q173
	q146	----	----	----	q165	q169	----	----	----	q169	----
*	q147	----	----	----	----	----	----	----	----	----	----
	q148	----	----	----	q167	----	----	----	----	q174	q175
*	q149	----	----	----	----	----	----	----	----	----	----
*	q150	----	----	----	----	----	----	----	----	----	----
*	q151	----	----	----	----	----	----	----	----	----	----
*	q152	----	----	----	----	----	----	----	----	----	----
*	q153	----	----	----	----	----	----	----	----	----	----
*	q154	----	----	----	----	----	----	----	----	----	----
	q155	----	----	----	----	q169	----	q176	----	----	q177
*	q156	----	----	----	----	----	----	----	----	----	----
*	q157	----	----	----	----	----	----	----	----	----	----
*	q158	----	----	----	----	----	----	----	----	----	----
	q159	----	----	----	----	q166	----	q167	q167	----	----
*	q160	----	----	----	----	----	----	----	----	----	----
*	q161	----	----	----	----	----	----	----	----	----	----
*	q162	----	----	----	----	----	----	----	----	----	----
*	q163	----	----	----	----	----	----	----	----	----	----
*	q164	----	----	----	----	----	----	----	----	----	----
*	q165	----	----	----	----	----	----	----	----	----	----
*	q166	----	----	----	----	----	----	----	----	----	----
*	q167	----	----	----	----	----	----	----	----	----	----
*	q168	----	----	----	----	----	----	----	----	----	----
*	q169	----	----	----	----	----	----	----	----	----	----
*	q170	----	----	----	----	----	----	----	----	----	----
*	q171	----	----	----	----	----	----	----	----	----	----
*	q172	----	----	----	----	----	----	----	----	----	----
*	q173	----	----	----	----	----	----	----	----	----	----
*	q174	----	----	----	----	----	----	----	----	----	----
*	q175	----	----	----	----	----	----	----	----	----	----
*	q176	----	----	----	----	----	----	----	----	----	----
*	q177	----	----	----	----	----	----	----	----	----	----
*	q178	----	----	----	----	----	----	----	----	----	----
*	q179	----	----	----	----	----	----	----	----	----	----
*	q180	----	----	----	----	----	----	----	----	----	----

Dalam pembuatannya, tabel transisi ini menggunakan MS Excel, dan sebelum dibentuk menjadi tabel berisi nama *state* saja, dilakukan gambaran setiap kotak dalam permainan

terlebih dahulu untuk membayangkan perubahan yang terjadi secara jelas. Gambaran tersebut kemudian dipecah menjadi tabel transisi di atas dan sebuah kamus transisi yang menggambarkan isi setiap kotak dalam permainan ketika *state* adalah *state* tertentu.

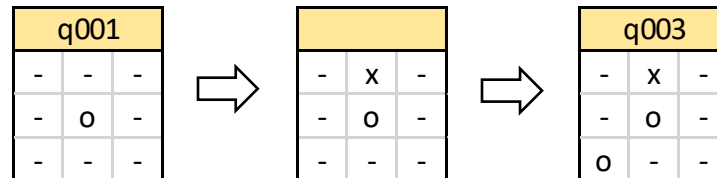
Berikut adalah kamus *state*:

q000	-----	q046	o---o-xxo	q092	oxo-oxo-x	q138	ox--x-oox
q001	----o----	q047	o---o-oxx	q093	o-oxoxo-x	q139	oxxox-o--
q002	x---o---o	q048	ox--o-o-x	q094	ooo-oxx-x	q140	o-xxxoo--
q003	-x--o-o--	q049	o-x-oo--x	q095	o-o-oxoxx	q141	o-xoxxo--
q004	--x-o-o--	q050	o-oxo---x	q096	xxoooox--	q142	o-xox-ox-
q005	--oxo----	q051	o---o-xox	q097	x-ooooxx-	q143	o-xox-o-x
q006	o---ox---	q052	xxoxo-o-o	q098	x-oooox-x	q144	o-oxxox--
q007	--o-o-x--	q053	xxo-oxo-o	q099	-xoxoox-o	q145	oo-xxo-x-
q008	o---o--x-	q054	xxo-oox-o	q100	oxo-oxx-o	q146	o--xxoo-x
q009	o---o---x	q055	xxo-o-oxo	q101	oxo-o-xxo	q147	ox-oxxo--
q010	xxo-o---o	q056	xoxxo--oo	q102	oooxoxx--	q148	o-ooxxx--
q011	xox-o---o	q057	xox-ox-oo	q103	ooo-oxxx-	q149	o--oxxox-
q012	x--xo-o-o	q058	xox-o-xoo	q104	-xo-ooxxo	q150	o--oxxo-x
q013	x---oxo-o	q059	xox-o-oxo	q105	--oxooxxo	q151	oooxx-x--
q014	x--oo-x-o	q060	x-xxo-ooo	q106	o-o-oxxxo	q152	ooo-xxx--
q015	x-o-o--xo	q061	x-oxoxo-o	q107	oxo-o-oxx	q153	ooo-x-xx-
q016	xxo-o-o--	q062	x-oxo-oxo	q108	o-oxo-oxx	q154	ooo-x-x-x
q017	oxx-o-o--	q063	x-x-oxooo	q109	oxoxo-o-x	q155	oox-x-ox-
q018	-xoxo-o--	q064	x-o-oxoxo	q110	ox-oo-oxx	q156	oooxx--x-
q019	-xo-oxo--	q065	xx-ooox-o	q111	oxxooo--x	q157	ooo-xx-x-
q020	-xo-o-ox-	q066	x-xooox-o	q112	o-xooox-x	q158	ooo-x--xx
q021	-xo-o-o-x	q067	x-oooxx-o	q113	o-xooo-xx	q159	oxo-x--ox
q022	xox-o-o--	q068	x--oooxxo	q114	oxo-o-xox	q160	oooxx---x
q023	--xxo-o-o	q069	x-o-ooxxo	q115	oox-o-xox	q161	ooo-xx--x
q024	--x-oxo-o	q070	oxxxo-o-o	q116	oo-xo-xox	q162	oxxxx-ooo
q025	o-x-o-ox-	q071	oxx-oxo-o	q117	oo--oxxox	q163	oxxoxxoo-
q026	--x-ooo-x	q072	oxxoo-ox-	q118	xxxxoooox	q164	oxxox-oox
q027	x-oxo-o--	q073	oxxoo-o-x	q119	xxoxoxoxo	q165	oxxxoooox
q028	--oxoxo--	q074	xoxxo-oo-	q120	xxooooxxoo	q166	oxoxxoxox
q029	o-oxo-x--	q075	xox-oxoo-	q121	xooooxxxo	q167	oxooooxxox
q030	--oxo-ox-	q076	xox-o-oox	q122	oxxxoooox	q168	ox-oxxoox
q031	--oxo-o-x	q077	-xxxo-ooo	q123	ooxxooooxx	q169	ooxxxooxx
q032	ox--ox--o	q078	--xxoxooo	q124	oxoxooxox	q170	oxoxxox-o
q033	o-x-ox--o	q079	o-xxo-oxo	q125	oxooooxxox	q171	oooxxoxx-
q034	o--xox--o	q080	-xx-oxooo	q126	o---x----	q172	oooxxox-x
q035	o---oxx-o	q081	o-x-oxoxo	q127	ox--x--o-	q173	oooxxo-xx
q036	o---ox-xo	q082	oxx-o-oxo	q128	o-x-x-o--	q174	ooooxxxx-
q037	o-o-ox--x	q083	o-xoo-oxx	q129	o--xxo---	q175	ooooxxx-x
q038	x-ooo-x--	q084	x-xoooo-x	q130	o--oxx---	q176	ooxoxxox-
q039	-xo-o-x-o	q085	-xxoooo-x	q131	o-o-x-x--	q177	ooxox-oxx
q040	o-o-oxx--	q086	o-xxooo-x	q132	oo--x--x-	q178	xoo-o-xox
q041	--o-o-xxo	q087	--xooooxx	q133	o-o-x---x	q179	-ooxo-xox
q042	--o-o-xox	q088	oxoxo-x-o	q134	oxx-x-oo-	q180	-oo-oxxox
q043	ox--o--xo	q089	o-oxoxx-o	q135	ox-xxo-o-		
q044	o-x-o--xo	q090	o-oxo-xxo	q136	ox-oxx-o-		
q045	o--xo--xo	q091	oooxo-x-x	q137	oxo-x-xo-		

Pada tabel di atas, *state* menang diberi warna hijau sedangkan *state* seri diberi warna biru.

### C. STATE DAN AKSI

*State* pada DFA merupakan *state* di mana pemain baru saja meletakkan X berikutnya dan komputer kemudian membalas gerakan pemain. Sebagai contoh, *state* q001 merupakan *state* di mana komputer mulai terlebih dahulu. Pemain kemudian dapat meletakkan X di posisi 1-4 atau 6-9. Misalkan pemain meletakkan X di 2, maka komputer akan membalas gerakan tersebut dengan meletakkan O di posisi 7. *State* berubah menjadi q003.



Gerakan komputer ditentukan berdasarkan gerakan yang terbaik untuk memenangkan atau membuat seri permainan tic-tac-toe secara cepat. Referensi untuk gerakan komputer terbaik diambil dari Wikipedia dan Quora.

Dalam program digunakan sebuah fungsi yang diberi nama Trans untuk menentukan transisi, dengan parameter CState dan x, di mana CState adalah *state* permainan saat ini dan x adalah indeks transisi yang dicari. Indeks transisi untuk p (kondisi ketika pemain mulai terlebih dahulu) adalah -1 dan c (kondisi ketika komputer mulai terlebih dahulu) adalah 0, sedangkan untuk x = 1...9 indeks adalah x + 2. Untuk fungsi Trans secara lengkap, serta fungsi-fungsi lainnya, terdapat pada bagian *source code* laporan ini.

### D. SOURCE CODE

Source code program terdiri dari sebuah program utama yang menjalankan fungsi Play, serta fungsi-fungsi sebagai berikut:

#### a. Prosedur Print

Prosedur Print adalah fungsi yang digunakan untuk membaca kamus state pada file eksternal dan mencetak gambaran setiap kotak permainan dalam bentuk *grid* pada saat *state* tertentu. Prosedur ini memiliki parameter CState, yaitu *state* permainan pada saat ini.

```
void Print (char * state) {
    FILE *fp;
    char MARK[6], label[5], content[10];

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

    fscanf(fp, "%s", MARK);
    while (strcmp(MARK, "KAMUS") != 0) {
```



```

        fscanf(fp, "%s", MARK);
    }

    fscanf(fp, "%s", label);
    while (strcmp(label, state) != 0) {
        fscanf(fp, "%s", label);
        fscanf(fp, "%s", label);
    }
    fscanf(fp, "%s", content);
    printf("%c %c %c\n", content[0], content[1], content[2]);
    printf("%c %c %c\n", content[3], content[4], content[5]);
    printf("%c %c %c\n", content[6], content[7], content[8]);
};

```

## b. Fungsi Trans

Fungsi Trans adalah fungsi yang digunakan untuk membaca tabel transisi pada file eksternal dan menentukan transisi *state* permainan setelah pemain memasukkan gerakannya selanjutnya. Fungsi ini memiliki parameter CState dan x, di mana CState adalah *state* permainan pada saat ini dan x adalah indeks transisi. Indeks transisi untuk p adalah -1 dan c adalah 0, sedangkan untuk x = 1...9 indeks adalah x + 2. Keluaran dari fungsi ini berupa “string” yang berisi *state* permainan yang sudah diubah.

```

char* Trans (char * state, int x) {
    FILE *fp;
    char MARK[9], skip[2], s[5], to[5], *final;

    fp = fopen("file.txt", "r");
    fscanf(fp, "%s", MARK);
    while (strcmp(MARK, "TRANSISI") != 0) {
        fscanf(fp, "%s", MARK);
    }

    for (int i = 1; i <= 11; i++) {
        fscanf(fp, "%s", skip);
    }

    fscanf(fp, "%s", s);
    while (strcmp(s, state) != 0) {
        fscanf(fp, "%s", s); //transisi p
        fscanf(fp, "%s", s); //transisi c
        fscanf(fp, "%s", s); //transisi 1
        fscanf(fp, "%s", s); //transisi 2
        fscanf(fp, "%s", s); //transisi 3
        fscanf(fp, "%s", s); //transisi 4
    }
}

```

```

        fscanf(fp, "%s", s); //transisi 5
        fscanf(fp, "%s", s); //transisi 6
        fscanf(fp, "%s", s); //transisi 7
        fscanf(fp, "%s", s); //transisi 8
        fscanf(fp, "%s", s); //transisi 9
        fscanf(fp, "%s", s); //state berikutnya
    }

    for (int i = 1; i <= x+2; i++) {
        fscanf(fp, "%s", to);
    }
    if (strcmp(to, "----") != 0) {
        final = to;
    } else {
        printf("That spot is occupied.\n");
        final = state;
    }
    return final;
}

```

### c. Prosedur CheckWin

Prosedur CheckWin adalah fungsi yang digunakan untuk membaca daftar *state* menang pada file eksternal dan menentukan apakah *state* saat ini merupakan *state* di mana komputer sudah menang. Prosedur ini memiliki parameter CState dan win, di mana CState adalah *state* permainan pada saat ini dan win adalah “boolean” yang menyatakan apakah komputer menang atau tidak.

```

void CheckWin (char * state, int * win) {
    FILE *fp;
    char MARK[9], s[5];

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

    fscanf(fp, "%s", MARK);
    while (strcmp(MARK, "MENANG") != 0) {
        fscanf(fp, "%s", MARK);
    }

    fscanf(fp, "%s", s);
    while (strcmp(s, "END") != 0) {
        if (strcmp(s, state) == 0) {
            *win = 1;
            break;
        } else {
            fscanf(fp, "%s", s);
        }
    }
}

```

```

    }
}

```

#### d. Prosedur CheckTie

Prosedur CheckTie adalah fungsi yang digunakan untuk membaca daftar *state* seri pada file eksternal dan menentukan apakah *state* saat ini merupakan *state* di mana terjadi seri. Prosedur ini memiliki parameter CState dan tie, di mana CState adalah *state* permainan pada saat ini dan tie adalah “boolean” yang menyatakan apakah permainan seri atau tidak.

```

void CheckTie (char * state, int * tie) {
    FILE *fp;
    char MARK[9], s[5];

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

    fscanf(fp, "%s", MARK);
    while (strcmp(MARK, "SERI") != 0) {
        fscanf(fp, "%s", MARK);
    }

    fscanf(fp, "%s", s);
    while (strcmp(s, "END") != 0) {
        if (strcmp(s, state) == 0) {
            *tie = 1;
            break;
        } else {
            fscanf(fp, "%s", s);
        }
    }
}

```

#### e. Prosedur Play

Prosedur Play adalah prosedur yang menggabungkan fungsi Print, Trans, CheckWin, dan CheckTie. Prosedur ini tidak memiliki parameter.

```

void play () {
    char CState[5] = "q000";
    char history[100] = "q000";
    char first, dump[100];
    int move;
    int win, tie = 0;

    printf("\nFirst player (x = you, o = CPU): ");
    scanf(" %c", &first);
}

```

```

while ((first != 'x') && (first != 'o')) {
    printf("Your input is not 'x' or 'o'.\n");
    printf("Re-input: "); scanf(" %c", &first);
}

printf("\nFirst move is filling the middle slot.\n");
if (first == 'x') {
    strcpy(CState, Trans(CState, -1));
}
if (first == 'o') {
    strcpy(CState, Trans(CState, 0));
}
Print(CState);
strcat(history, " -> "); strcat(history, CState);

while ((win == 0) && (tie == 0)) {
    printf("Your next move (1-9): ");
    if (scanf("%d", &move)) {
        if ((move >= 1) && (move <= 9)) {
            strcpy(CState, Trans(CState, move));
            printf("\n"); Print(CState);
            strcat(history, " -> "); strcat(history, CState);
            CheckWin(CState, &win);
            CheckTie(CState, &tie);
        } else {
            printf("You can only input numbers 1-9.\n\n");
        }
    } else {
        printf("You can only input a number!\n\n");
        scanf("%s", dump);
    }
}

if (win) {
    printf("\nCPU wins.\n");
} else if (tie) {
    printf("\nIt's a tie.\n");
} else {
    printf("\nDefintely an error...\n");
}

printf("\nStates history: %s\n", history);
}

```

## E. CONTOH INPUT-OUTPUT

Untuk memastikan program berjalan dengan baik, dilakukan beberapa pengujian. Berikut merupakan contoh input dan output program. Output ditebalkan dan diberi garis bawah.

### *Tes 1:*

First player (x = you, o = CPU): y

Your input is not 'x' or 'o'.

Re-input: a

Your input is not 'x' or 'o'. Also, I can only read one word at a time.  
Please don't tease me.

Re-input: x

First move is filling the middle slot.

```
o - -  
- x -  
- - -
```

Your next move (1-9): 7

```
o - o  
- x -  
x - -
```

Your next move (1-9): 2

```
o x o  
- x -  
x o -
```

Your next move (1-9): 4

```
o x o  
x x o  
x o x
```

It's a tie.

States history: q000 -> q126 -> q131 -> q137 -> q166

**Test 2:**

First player (x = you, o = CPU): o

First move is filling the middle slot.

```
- - -  
- o -  
- - -
```

Your next move (1-9): 0

You inserted 0. You can only input numbers 1-9.

Your next move (1-9): a

You can only input numbers!

Your next move (1-9): 3

```
- - x  
- o -  
o - -
```

Your next move (1-9): 2

```
o x x  
- o -  
o - -
```

Your next move (1-9): 4

```
o x x  
x o -  
o - o
```

CPU wins.

States history: q000 -> q001 -> q004 -> q017 -> q070

**F. REFERENSI**

Subramaniam, Arjun. (2015). *Is there a way to never lose at Tic-Tac-Toe?*. Diakses September 2018, dari

<https://www.quora.com/Is-there-a-way-to-never-lose-at-Tic-Tac-Toe>.

Wikipedia. (2018). *Tic-tac-toe*. Diakses September 2018, dari

<https://en.wikipedia.org/wiki/Tic-tac-toe>.