

# **PROGRAM TIC-TAC-TOE**

## **LAPORAN Pengerjaan**

Diajukan untuk memenuhi tugas kuliah Teori Bahasa Formal dan Otomata

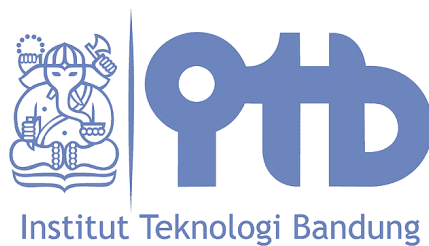
oleh

**Ahmad Darmawan**

**13513096**

**Fiqie Ulya S**

**13514602**



**TEKNIK INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**BANDUNG**

**2014**

## A. TIC TAC TOE

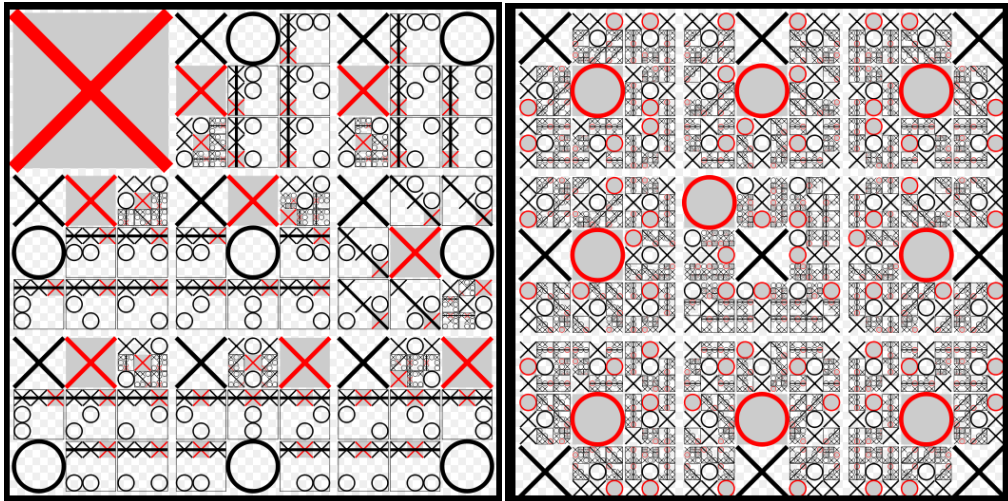
permainan kertas dan pensil untuk dua pemain, X dan O, yang bergiliran menandai ruang dalam grid  $3 \times 3$ . Pemain yang berhasil menempatkan tiga tanda masing-masing dalam sebuah baris horisontal, vertikal, atau diagonal memenangkan permainan.

Contoh permainan:



Game diatas menunjukan pemain pertama X, memenangkan permainan.

Beberapa strategi yang digunakan:



Pada Program TIC TAC TOE ini, penulis berusaha membuat Program sederhana TIC TAC TOE dengan Finite Automata yang optimal untuk langkah CPU. Sehingga program ini dibentuk dengan tujuan langkah optimal CPU dapat mengalahkan atau hanya dapat draw dengan user.

Penulis membuat Finite automata dengan mencoba segala kemungkinan yang bisa didapat. Finite Automata tersebut menangani CPU sebagai pemain pertama ( attack) dan CPU pemain ke dua (defend).

## B. FILE EXTERNAL

File external berupa file .txt yang dibuat di dalam Operating System Windows. Dalam penggunaannya File ini dapat di esekusi oleh Windows, untuk operating sistem Linux, file .txt akan mengalami perubahan dikarenakan karakter “enter” dalam window memiliki 2 karakter, yakni ‘\r’ dan ‘\n’.

[Final State][State dalam Kombinasi][Alphabet Inputan][TRANSISI]

- Final State:** Karakter yang menunjukkan bahwa suatu state dikatakan Final State atau tidak, ‘w’ berarti CPU win, ‘d’ berarti CPU draw, dan ‘-’ berarti bukan state akhir.
- State dalam Kombinasi:** karakter yang menyatakan kombinasi bentuk array TICTACTOE yang sedang di jalankan.
- Alphabet Inputan:** karakter angkayang menyatakan inputan untuk state agar berpindah. Inputan ini berupa langkah yang di ambil dari user berupa address dalam array
- TRANSISI:** karakter angka yang menyatakan kemana langkah CPU selanjutnya, dilihat dari file external.

Contoh pembacaan file external:

State dimana player bermain sebagai pemain ke-2 dan CPU pemain ke-1,  
Direpresentasikan CPU sebagai ‘o’ dan user sebagai ‘x’.

-o-----|2|7  
-o-----|3|9  
-o-----|4|3  
-o-----|5|9  
-o-----|6|3  
-o-----|7|9  
-o-----|8|3  
-o-----|9|3

wOX----o--|3|4  
-oX----o--|4|5  
wOX----o--|5|4  
wOX----o--|6|4  
wOX----o--|8|4  
wOX----o--|9|4

Variabel yang digunakan dalam penyimpanan tic tac toe, digunakan **representasi array**, dari 0 hingga 11.

Index 0 digunakan untuk representasi bahwa state belum Final.

Indeks 1 sampai 9 untuk representasi state sekarang.

1	2	3
4	5	6
7	8	9

Indeks 10 hanya sebagai pemisah state dan transisi .

Indeks 11 sebagai inputan user.

CPU bermain pertama, oleh karena itu CPU akan meletakkan bidak di sudut. Misal di indeks 1, kemudian user meletakkan di indeks 2, maka state sekrang -o-----|2.

Array tersebut kemudian masuk kedalam fungsi transisi, dan akan diseleksi dengan file external.

Pada fungsi transisi, array state sekarang dibandingkan dengan file external yang kombinasinya cocok(dengan fungsi IsStateSama). Untuk karakter yang dibandingkan hanya karakter 1- 11, dan akan mengoutput karakter ke 13(Hasil Transisi). Setelah itu index ke 0 akan di inisialisasi sesuai dengan file external yang sesuai.

**State awal :** o-----

(menyimpan basis(flag) – bila dibutuhkan rotate, dan hanya dilakukan di state awal)

**Input user :** 2

**Array sekarang:** -o-----|2

>>Array sekarang diproses dalam fungsi Transisi

Inputan user, state awal dirotate sebanyak flag jika perlu

>>Program mulai membaca file external, dengan memanfaatkan ADT MesinKarakter, dan Kata, untuk setiap baris dalam file external akan dibaca 1 array karakter, dan enter di gunakan untuk pindah ke array karakter di bawahnya. (Anggap **Cstate**)

**Cstate** akan dibandingkan dengan Array sekarang hingga menemukan **Cstate** yang indeks 1-11 sama seperti **Arraysekarang**.

**Cstate yang sama:** -o-----|2|7

Jika **Array sekarang = Cstate**, maka kita ambil elemen ke 13 dari Cstate

Hasil dan inputan user kemudian dimasukan ke Array sekarang. Sehingga

**State sekarang=** ox----o--

**Input user :** 3

**Array sekarang:** -ox----o--|3

**Cstate :** wox----o--|3|4

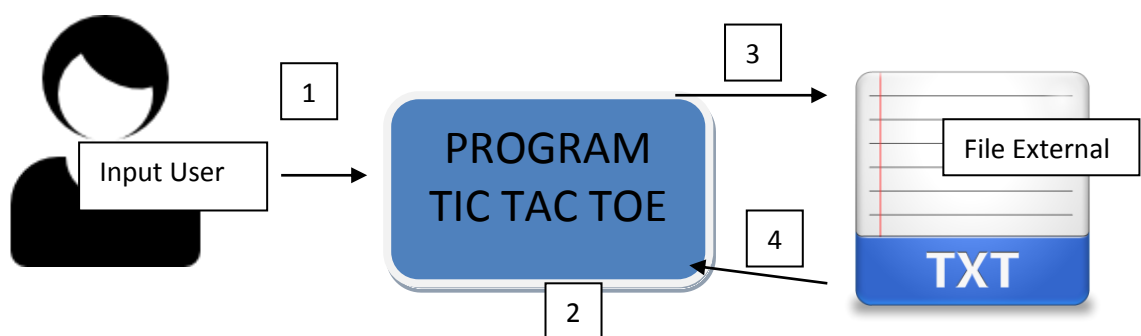
**Hasil=4;**

Ambil karakter ke 13, dan inialisasikan ke **dalam Array sekarang**

**Array sekarang=wox----o--|**

Karena sudah membaca karakter ke 0 berupa final state (d/w), maka program sudah selesai dieksekusi

### C. CARA KERJA



1. User memberikan inputan berupa angka 1..9 yang melambangkan letak bidak dari user.
2. Program Menangani inputan user, membentuk representasi array, melakukan proses rotasi untuk menangani inputan yang sejenis(sudut, tengah, pinggir).

Program disini hanya bertugas sebagai fasilitator pembaca file external, dan implementasi program.

3. Untuk mendapatkan langkah transisi, kombinasi state akan diseleksi oleh file external.
4. Program TIC TAC TOE mendapatkan transisi berdasarkan file external yang sudah dibuat.

#### D. TABEL TRANSISI FINITE OTOMATA

Langkah optimal yang diambil oleh penulis yakni langkah dimana CPU melakukan tindakan optimal terhadap USER, CPU berusaha untuk menang, atau seri. Namun disini penulis berasumsi sebuah game memungkinkan untuk pemain tidak memiliki peluang kalah lebih banyak, sehingga transisi yang digunakan memberikan peluang seri yang besar pula.

Finite Automata yang menangani CPU sebagai pemain pertama

State	Transisi								
	1	2	3	4	5	6	7	8	9
-> (o-----)Q1	-	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
(ox----o--)Q2	-	-	4-Q28	Q10	4-Q28	4-Q28	-	4-Q28	4-Q28
(o-x-----)Q3	-	5	-	5-Q28	Q11	5-Q28	5-Q28	5-Q28	-
(o-ox-----)Q4	-	Q12	-	-	2-Q28	2-Q28	2-Q28	2-Q28	2Q28
(o---x---o)Q5	-	Q13	Q14	Q15	-	Q16	Q17	Q18	-
(o-o--x---)Q6	-	Q19	-	2-Q28	2-Q28	-	2-Q28	2-Q28	2-Q28
(o-----x-o)Q7	-	5-Q28	5-Q28	5-Q28	Q20	5-Q28	-	5-Q28	-
(o-o----x-)Q8	-	Q21	-	2-Q28	2-Q28	2-Q28	2-Q28	-	2-Q28
(o-o-----x)Q9	-	Q22	-	2-Q28	2-Q28	2-Q28	2-Q28	2-Q28	-
(ox-xo-o--)Q10	-	-	9-Q28	-	-	3-Q28	-	3-Q28	3-Q28
(o-x-x-o-o)Q11	-	8-Q28	-	8-Q28	-	8-Q28	-	4-Q28	-
(oxoxo----)Q12	-	-	-	-	-	9-Q28	9-Q28	9-Q28	7-Q28
(ox--x--oo)Q13	-	-	7-Q28	7-Q28	-	7-Q28	Q23	-	-
(o-x-x-o-o)Q14	-	8-Q28	-	8-Q28	-	8-Q28	-	4-Q28	-
(o--xxo--o)Q15	-	3-Q28	Q24	-	-	-	3-Q28	3-Q28	-
(o--oxx--o)Q16	-	7-Q28	Q25	-	-	-	3-Q28	7-Q28	-
(o-o-x-x-o)Q17	-	6-Q28	-	2-Q28	-	2-Q28	-	2-Q28	-
(oo--x--xo)Q18	-	-	Q26	3-Q28	-	3-Q28	3-Q28	-	-
(oxo-ox---)Q19	-	-	-	9-Q28	-	-	9-Q28	9-Q28	7-Q28
(o-o-x-x-o)Q20	-	6-Q28	-	2-Q28	-	2-Q28	-	2-Q28	-
(oxo-o--x-)Q21	-	-	-	7-Q28	-	7-Q28	9-Q28	-	7-Q28
(oxo---o-x)Q22	-	-	-	5-Q28	4-Q28	5-Q28	-	5-Q28	-
(oxo-x-xoo)Q23	-	-	-	6-Q28	-	4-Q27	-	-	-
(o-xxxoo-o)Q24	-	8-Q28	-	-	-	-	-	2-Q27	-
(o-ooxxx-o)Q25	-	8-Q27	-	-	-	-	-	2-Q28	-
(oox-x-oxo)Q26	-	-	-	6-Q27	-	4-Q28	-	-	-

*(draw) Q27	-	-	-	-	-	-	-	-	-
*(win) Q28	-	-	-	-	-	-	-	-	-

Keterangan:

- ➔ : State Awal
- \* : State Akhir (FINAL STATE)
- O : CPU
- X : USER

Untuk CPU sebagai pemain kedua, transisi tidak penulis lampirkan dalam laporan ini dikarenakan jumlah state yang banyak. Secara garis besar, untuk kondisi ini dibagi menjadi 3 (tiga) state awal yaitu: di sudut (posisi ke-1,3,7,9), di pinggir (2,4,6,8), dan di tengah (5).

#### **E. ALGORITMA FINITE AUTOMATA TIC TAC TOE**

Penulis membagi kedalam 2 Logika, dimana logika pertama State awal dari user (User- Pemain 1 ) sedangkan CPU – Pemain 2 (Defend), dan State awal dari CPU (Attack) sedangkan user - pemain 2.

##### **➤ USER – PLAYER 1**

Pada bagian ini CPU lebih banyak menangani state, dikarenakan state awal dapat berupa 9 inputan yang berbeda. Pada Logika ini CPU juga bertindak lebih besar sebagai Defender, dikarenakan kemungkinan draw lebih besar. Penulis mengatasinya dengan mengelompokkannya dalam 2 solusi, yakni solusi state awal CPU sebagai ‘corner’ , CPU sebagai ‘midle’.

##### **➤ USER – PLAYER 2**

Langkah optimal yang diambil CPU saat mendapat giliran pertama yakni meletakkan bidaknya di sudut TICTACTOE dengan langkah random dari program. CPU melakukan Attack.

➤ TEKNIK ROTATE

		O
O		

Teknik rotate ini penulis gunakan untuk menangani jumlah kemungkinan state yang banyak. Dengan teknik rotate, penulis hanya perlu membuat transisi berdasarkan basis (sudut, pinggir, dan tengah) sehingga 1 transisi bisa dicerminkan untuk 4 pola.

O		

		O

Pada representasi diatas, posisi 1, 7, 3, 9 sebenarnya adalah bentuk rotasi 'sudut'. >

	O	

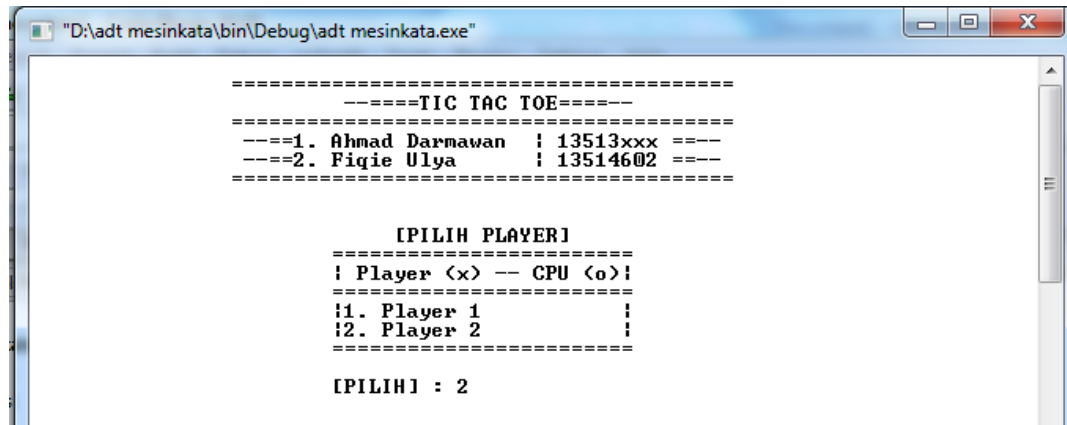
Representasi diatas menunjukan untuk rotate 'tengah'.





## F. DOKUMENTASI

Tampilan awal program TICTACTOE, user memilih peran sebagai player 1 atau player 2.



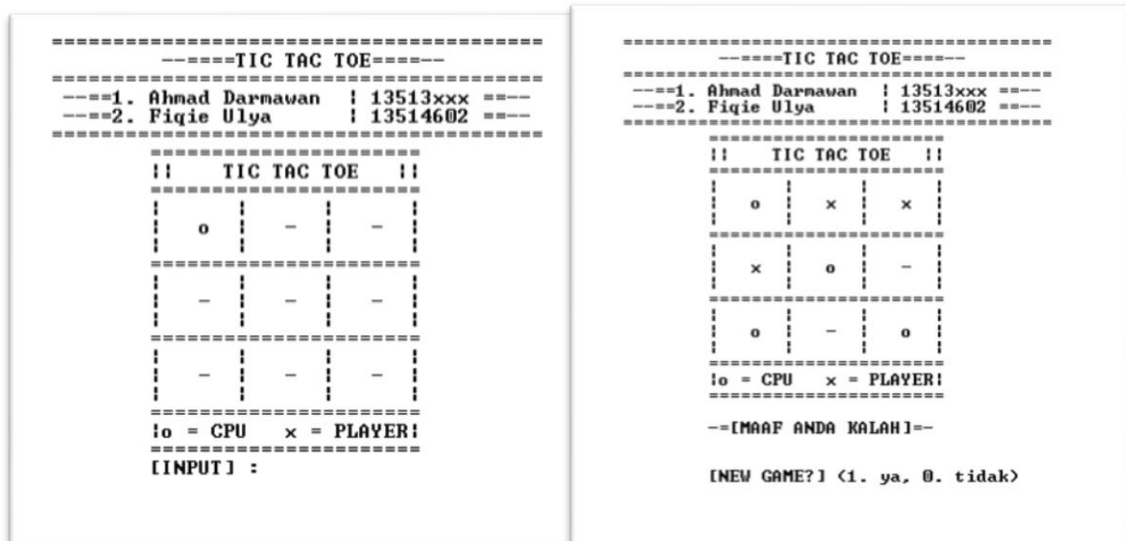
```
"D:\adt mesinkata\bin\Debug\adt mesinkata.exe"

=====
-----TIC TAC TOE-----
=====
---=1. Ahmad Darmawan | 13513xxx ---
---=2. Fiqie Ulya    | 13514602 ---
=====

[PILIH PLAYER]
=====
! Player <x> -- CPU <o>!
=====
!1. Player 1          !
!2. Player 2          !
=====

[PILIH] : 2
```

Tampilan Board TIC TAC TOE dan Hasil permainan (User kalah)



```
=====
-----TIC TAC TOE-----
=====
---=1. Ahmad Darmawan | 13513xxx ---
---=2. Fiqie Ulya    | 13514602 ---
=====

!!   TIC TAC TOE   !!
=====
| o | - | - |
|---|
| - | - | - |
|---|
| - | - | - |
=====
!o = CPU  x = PLAYER!
=====
[INPUT] :

=====
-----TIC TAC TOE-----
=====
---=1. Ahmad Darmawan | 13513xxx ---
---=2. Fiqie Ulya    | 13514602 ---
=====

!!   TIC TAC TOE   !!
=====
| o | x | x |
|---|
| x | o | - |
|---|
| o | - | o |
=====
!o = CPU  x = PLAYER!
=====
--[MAAF ANDA KALAH]--
=====
[NEW GAME?] <1. ya, 0. tidak>
```