

LAPORAN TUGAS I IF2220 TEORI BAHASA FORMAL DAN AUTOMATA
APLIKASI PERMAINAN “TIC TAC TOE” DENGAN MENGGUNAKAN
FINITE AUTOMATA



LOUIS CAHYADI
13517126

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

I

DESKRIPSI PERSOALAN

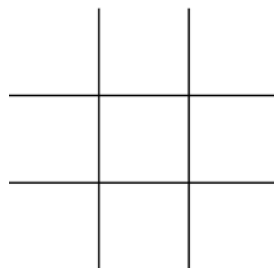
Tic-tac-toe adalah sebuah permainan untuk dua orang yang secara bergiliran saling membuat huruf X dan O di dalam sebuah kotak 3×3 . Pemenang dari permainan ini adalah pemain pertama yang berhasil membuat tiga tanda “X” atau tanda “O” berurutan di dalam suatu permainan, baik horizontal, vertikal maupun diagonal.



Dalam tugas ini, dibuat sebuah aplikasi permainan tic-tac-toe sederhana, yang mana permainan ini akan dimainkan komputer dan player. Program harus bisa dipastikan bahwa komputer tidak mungkin kalah dalam permainan. Aplikasi juga dibuat dengan menggunakan finite automata.

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.

tic tac toe



Pada aplikasi, konvensi yang digunakan ialah petak pada sudut kiri atas adalah petak 1,2, dan 3, kemudian baris selanjutnya dilanjutkan dengan petak 4,5,6, dan baris terakhir adalah petak 7,8, dan 9.

Batasan masalah :

Pada langkah pertama dalam setiap permainan, player atau CPU dipastikan meletakkan tanda “X” atau “O” di bagian tengah papan.

Implementasi :

1. Program dibuat dalam bahasa pemrograman prosedural C.
2. Program akan membaca file text yang berisi deskripsi state machine.
3. Program akan menerima masukan aksi terhadap state machine, sebagai contoh :
 - a. 5 berarti memasukan pada petak ke – 5
 - b. 7 berarti memasukan pada petak ke – 7
4. Pemain memainkan tanda X, sementara komputer memainkan tanda O
5. Keluaran berupa papan permainan tic-tac-toe setelah state dimasukan, serta kondisi terakhir permainan jika telah diketahui hasilnya (menang/kalah/seri).
Pada kondisi terakhir, program juga harus menampilkan state mana saja yang sudah dilalui selama permainan.
6. Tidak menggunakan library terkait finite automata yang sudah ada.

II

DFA

1. Daftar state :

1. AAAAAAAAAA	48. OXXOXAOOX	95. OXAAOAA XO
2. AAAAXAOAA	49. OXAOXXOOX	96. XAXOOOAAA
3. XAAAXAOAO	50. XXOXXOOAO	97. AXXOOOAAA
4. AXAAXAOOA	51. XAXXXOOOO	98. AAXOOOXAA
5. OAXAXAOAA	52. XAOXXOOXO	99. AAXOOOAXA
6. AAAXXOOAA	53. XXAXXOOOO	100. AAXOOOAA X
7. AAAOXXOAA	54. AXXXXOOOO	101. XAOXOAOAA
8. AOAXXAOXA	55. XXAOXXOOO	102. AXOXOAOAA
9. OAAAXAOAX	56. XAXOXXOOO	103. AAOXOXOAA
10. XOAXXAOXO	57. OOXOXXOXA	104. AAOXOAXOA
11. OXAAXAOOX	58. OOA OXXOXX	105. AAOXOAOAX
12. OAXXXOOAA	59. OOXOXAOXX	106. XAOAOXOAA
13. XAAXXOOAO	60. OOOXXAOXX	107. AXOAOXOAA
14. AXAXXOOOA	61. XOXXXOOXO	108. AAOAOXOXA
15. AOXXXOOAA	62. XOXOXXOXO	109. AAOAOXOAX
16. AOAXXOOXA	63. OXXXXOOOX	110. XAAOOOXAA
17. OAAXXOOAX	64. OOXXXOOXX	111. AXA OOOXAA
18. XAAOXXOAO	65. AAAAOAAAA	112. AAAOOOXXA
19. AOXAXOOXA	66. XOOAOAAAA	113. AAAOOOXAX
20. AOAOXXOXA	67. OXAAOAAAA	114. OAXAOAA XO
21. OOAAXAOXX	68. AAXAOOAAA	115. OAXOAA XO
22. XOXAXOOXO	69. AAAXOAOAA	116. OAAA OXAXO
23. XOAXXOOXO	70. AAOAOXAAA	117. OAAA OAXXO
24. XOA OXXOXO	71. AAAOOAXAA	118. AOXA OAAOX
25. OXAXXOOOX	72. AAAAOAA XO	119. AOXA OAAOX
26. OXXXXOOOA	73. AAAAOAA OX	120. AOAA OXA OX
27. OOXXXOOXA	74. XOOAOOAXA	121. AOAA OAXOX
28. OOXXXOOAX	75. OXAOOAAAX	122. XOXOOOAXA
29. XOXXXOOAO	76. AAXXOOAOA	123. XOA OOOXXA
30. OOAXXOOXX	77. AAXXOAOOA	124. XOA OOOAXX
31. OOXAXOOXX	78. AOOAOXXAA	125. OXXOOOAA X
32. XXAAXAOOO	79. AOAO OXXAA	126. OXA OOXOAX
33. XAXAXAOOO	80. XAAA OOA XO	127. OXA OOOXAX
34. XAAAXXOOO	81. AXAOOAA OX	128. OXA OOOAXX
35. XAAXXA OOO	82. XOAX OOOXA	129. XOX XOOAOA
36. AXXAXAOOO	83. OXXX OOA OA	130. AOXX OOXOA
37. AXAXXA OOO	84. AOAO OXXXO	131. AOXX OOA OX
38. AXAAXXOOO	85. AXOO OXA OX	132. XOX XOA OOA
39. OXXOXAOAA	86. XOXAOAA OA	133. AXXX OAOOO
40. OAXOXXOAA	87. XOAXOAA OA	134. AOXX OXOOA
41. OAXOXAOXA	88. XOOAOAXOA	135. AOXX OAOOX
42. OAXOXAOAX	89. XOOAOXAOA	136. XOOAOXXOA
43. OXA OXXOAA	90. XOOAOAA OX	137. OOOX OXXAA
44. OAA OXXOXA	91. OXXAOAA AO	138. OOOAOXXXA
45. OAA OXXOAX	92. OXAXOAA AO	139. OOOAOXXAX
46. OXA OXAOAX	93. OXAA OXAAO	140. XOA OOX XOA
47. OAA OXAOXX	94. OXAA OAXAO	141. AOX OOX XOA

- | | | |
|----------------|----------------|----------------|
| 142. AOA00XXOX | 148. AXX000A0X | 154. X0000XXXO |
| 143. XXOA00AXO | 149. AXA000XOX | 155. 00X00XXXO |
| 144. XAX000AXO | 150. X0XX000XO | 156. XX000X00X |
| 145. XAOX00AXO | 151. X00X000XX | 157. 0X000XXOX |
| 146. XAA000XXO | 152. 0XXX00X0O | |
| 147. XXA000A0X | 153. 0XXX0000X | |

2. Alfabet yang digunakan = {1,2,3,4,5,6,7,8,9}

3. Start state = AAAAAAAAAA

4. Final state :

- | | | |
|---------------|---------------|----------------|
| 1. XXAAXA000 | 36. XOAAOAXOA | 71. XOA000XXA |
| 2. XAXAXA000 | 37. XOAAOXAOA | 72. XOA000AXX |
| 3. XAAAXX000 | 38. XOAA0AA0X | 73. OXX000AAX |
| 4. XAAXXA000 | 39. OXXA0AAAA | 74. OXA00XOAX |
| 5. AXXAXA000 | 40. OXAXOAAAA | 75. OXA000XAX |
| 6. AXAXXA000 | 41. OXAAOXAAO | 76. OXA000AXX |
| 7. AXAAXX000 | 42. OXAAOAXAO | 77. X0XX00A0A |
| 8. OXXOXAOAA | 43. OXAA0AAXO | 78. AOXX00XOA |
| 9. OAXOXXOAA | 44. XAX000AAA | 79. AOXX00A0X |
| 10. OAXOXAOXA | 45. AXX000AAA | 80. X0XXOAO0A |
| 11. OAXOXAOAX | 46. AAX000XAA | 81. AXXXOAO0O |
| 12. OXA0XXOAA | 47. AAX000AXA | 82. AOXXOX00A |
| 13. OAA0XX0XA | 48. AAX000AAX | 83. AOXXOAO0X |
| 14. OAA0XXOAX | 49. XAOXOAOAA | 84. X00A0XX0A |
| 15. OXA0XAOAX | 50. AXOXOAOAA | 85. 000X0XXAA |
| 16. OAA0XA0XX | 51. AAOXOXOAA | 86. 000A0XXXX |
| 17. OXXOXAO0X | 52. AAOXOAOXA | 87. 000A0XXAX |
| 18. OXA0XX00X | 53. AAOXOAOAX | 88. XOAO0XXOA |
| 19. XXOXX00AO | 54. XAOAOXOAA | 89. AOX00XXOA |
| 20. XAXXX0000 | 55. AXOAOXOAA | 90. AOAO0XXOX |
| 21. XAOXX00XO | 56. AAOAOX0XA | 91. XXOAO0AXO |
| 22. XXAXX0000 | 57. AAOAOXOAX | 92. XAX000AXO |
| 23. AXXXX0000 | 58. XAA000XAA | 93. XAOX00AXO |
| 24. XXAOXX000 | 59. AXA000XAA | 94. XAA000XXO |
| 25. XAX0XX000 | 60. AAA000XXA | 95. XXA000A0X |
| 26. 00X0XX0XA | 61. AAA000XAX | 96. AXX000A0X |
| 27. 00A0XX0XX | 62. OAXAOAAXO | 97. AXA000XOX |
| 28. 00X0XA0XX | 63. OAXOAAAXO | 98. X0XX000XO |
| 29. 000XXA0XX | 64. OAAA0XAXO | 99. X00X000XX |
| 30. X0XXX00XO | 65. OAAA0AXXO | 100. OXXX00X0O |
| 31. XOX0XXOXO | 66. AOXA0AA0X | 101. OXXX0000X |
| 32. OXXX0000X | 67. AOAXOAA0X | 102. X0000XXXO |
| 33. 00XXX00XX | 68. AOAAOXAOX | 103. 00X00XXXO |
| 34. XOXAOAAOA | 69. AOAAOAXOX | 104. XX000X00X |
| 35. XOAXOAAOA | 70. XOX000AXA | 105. 0X000XXOX |

5. Transition table

[illegible]

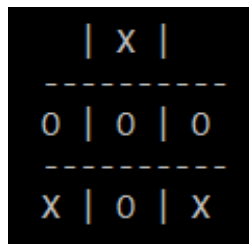
[illegible]

III

PENJELASAN STATE DAN AKSI

A. STATE

Setiap state yang ada merupakan string yang terdiri dari 9 karakter. Karakter pada posisi ke – i merepresentasikan kondisi pada petak nomor i, jika “A” berarti kosong, “X” berarti berisi tanda “X” dan “O” menandakan petak tersebut berisi tanda “O”. Sebagai contoh state AXXOOOXOX pada nomor 149, merepresentasikan keadaan papan seperti berikut



	X	
O	O	O
X	O	X

Pada file state.txt yang dilampirkan bersamaan dengan laporan ini, setiap final state ditambahkan karakter ke 10 dengan karakter 1 untuk menandakan state tersebut merupakan final state dengan kondisi komputer menang, atau karakter 2 untuk menandakan state tersebut merupakan final state dengan kondisi permainan berakhirimbang.

B. AKSI

Pada source code terdapat tiga buah prosedur dan dua buah fungsi. Prosedur pertama ialah menuAwal, yang menampilkan tampilan awal permainan serta menu awal permainan. Selanjutnya terdapat fungsi isMenang yang menerima input sebuah state q dan mengembalikan nilai true jika state q tersebut merupakan final state dalam kondisi komputer telah memenangkan permainan.

Lebih lanjut terdapat juga fungsi `isSeri` yang menerima input sebuah state `q` dan mengembalikan nilai `true` jika state `q` tersebut merupakan final state dalam kondisi permainan berakhir denganimbang.

Berikutnya terdapat prosedur `tulisState` yang menerima masukan state `q` dan menampilkan papan permainan tic-tac-toe sesuai dengan keadaan state seperti yang dijelaskan pada bagian state. Dan yang terakhir terdapat prosedur `pindahState` yang menerima masukan state `q` dan sebuah bilangan bulat `i`. Prosedur tersebut merubah state `q` dengan state yang dituju jika menerima alfabet `i` dengan cara mengambil string mulai dari posisi $(11*i)$ sebanyak 9 karakter pada transition table. Jika pengguna memberi input di tempat yang telah ada tandanya maka state akan kembali ke diri sendiri.

Pada program utama, pertama kali memunculkan menu awal dan assign `stateSekarang` sebagai `AAAAAAAAAA`, lalu dilanjutkan dengan menerima input siapa yang akan memulai giliran pertama. Selanjutnya lakukan pengulangan selama `isMenang` dan `isSeri` belum terpenuhi, untuk menerima input dari user dan melakukan prosedur pindah state serta ditampilkan. Jika sudah selesai print keadaan akhir serta menampilkan state-state yang telah dilewati.

IV

SOURCE CODE

```
#include <stdio.h>
#include <string.h>
#include <stdbool.h>

//Membuat tipe bentukan state yang merupakan array of char
typedef struct {
    char T[10];
} state;
//Deklarasi array untuk menampung state-state yang telah dilewati
state arrOfState[100];

void menuAwal();
bool isMenang(state q);
bool isSeri(state q);
void tulisState(state q);
void pindahState(state *q, int i);

int main(){
    int pilihan;
    int i,j;
    state qSekarang; //Menampung state yang sedang terjadi sekarang
    state q;

    menuAwal();
    printf("\n>> ");
    scanf("%d",&pilihan);

    //inisiasi start state
    qSekarang.T[0] = 'A';
    qSekarang.T[1] = 'A';
    qSekarang.T[2] = 'A';
    qSekarang.T[3] = 'A';
    qSekarang.T[4] = 'A';
    qSekarang.T[5] = 'A';
    qSekarang.T[6] = 'A';
    qSekarang.T[7] = 'A';
    qSekarang.T[8] = 'A';
    qSekarang.T[9] = '\0';

    //masukkan pertama
    pindahState(&qSekarang,pilihan);

    tulisState(qSekarang);
    arrOfState[1] = qSekarang;
    j = 2;

    while (!isMenang(qSekarang) && !isSeri(qSekarang)){
```

```

printf("\nMasukan nomor kotak yang ingin Anda beri tanda : ");
scanf("%d",&i);

pindahState(&qSekarang,i);
tulisState(qSekarang);
arrOfState[j] = qSekarang;
j += 1;
} //isMenang = true atau isSeri = true
//Menampilkan status akhir permainan
if (isMenang(qSekarang)){
    printf("\nPermainan telah selesai dan dimenangkan oleh komputer\n\n");
} else if (isSeri(qSekarang)){
    printf("\nPermainan telah selesai dengan imbang\n\n");
}
//Menampilkan daftar state yang telah dilewati
printf("Berikut ini ialah daftar state yang telah dilewati :\n");
for (i = 1; i < j; i++){
    printf("%d. %s\n",i,arrOfState[i]);
}

printf("\nKetik sembarang angka untuk keluar : ");
scanf("%d",&j);
return 0;
}

void menuAwal(){
//Prosedur untuk menampilkan menu di awal permainan
printf("-----\n");
printf("----- TIC TAC TOE -----\n");
printf("-----\n");
printf("");
printf("Siapa yang akan main pada giliran pertama?\n1. Komputer\n2. Anda\n");
}

bool isMenang(state q){
//Mengembalikan true bila state q merupakan final state dan dalam kondisi komputer menang
FILE *pFile;
char stateSebaris[110];
char stateMenang[2] = "1";
state qcek;
state q1;

pFile = fopen ("state.txt", "r");
while (!feof(pFile)){
    if (fgets(stateSebaris, 111, pFile) == NULL){
        break;
    } else {
        memcpy(qcek.T, &stateSebaris[0],9);
        qcek.T[9] = '\0';

        if (strcmp(qcek.T,q.T) == 0){
            memcpy(q1.T, &stateSebaris[9],1);
            q1.T[1] = '\0';
            if (strcmp(q1.T,stateMenang) == 0){
                return true;
            }
        }
    }
}
}

```

```

        } else {
            return false;
        }
    }
}
}
fclose(pFile);
}

```

bool isSeri(state q){
//Mengembalikan true apabila state q merupakan final state dan dalam kondisi permainan berakhirimbang.

```

    FILE *pFile;
    char stateSebaris[110];
    char stateSeri[2] = "2";
    state qcek;
    state q1;

    pFile = fopen ("state.txt", "r");
    while (!feof(pFile)){
        if (fgets(stateSebaris, 111, pFile) == NULL){
            break;
        } else {
            memcpy(qcek.T, &stateSebaris[0],9);
            qcek.T[9] = '\0';

            if (strcmp(qcek.T,q.T) == 0){
                memcpy(q1.T, &stateSebaris[9],1);
                q1.T[1] = '\0';
                if (strcmp(q1.T,stateSeri) == 0){
                    return true;
                } else {
                    return false;
                }
            }
        }
    }
    fclose(pFile);
}

```

void tulisState(state q){
//Menampilkan state q ke layar berupa papan permainan tictactoe

```

    int i;
    for (i=0;i <= 8;i++){
        if (q.T[i] == 'A'){
            q.T[i] = ' ';
        }
    }

    printf("\n%c | %c | %c\n",q.T[0],q.T[1],q.T[2]);
    printf("-----\n");
    printf("%c | %c | %c\n",q.T[3],q.T[4],q.T[5]);
    printf("-----\n");
    printf("%c | %c | %c\n",q.T[6],q.T[7],q.T[8]);
}

```

```

void pindahState(state *q, int i){
//Merubah state q ke state yang dituju jika diberi input i
FILE *pFile;
int j;
char stateSebaris[110];
state qcek;
state qi;

pFile = fopen ("state.txt", "r");
while (!feof(pFile)){
    if (fgets(stateSebaris, 111, pFile) == NULL){
        break;
    } else {
        memcpy(qcek.T, &stateSebaris[0],9);
        qcek.T[9] = '\0';

        if (strcmp(qcek.T,(*q).T) == 0){
            memcpy(qi.T, &stateSebaris[11*i],9);
            qi.T[9] = '\0';

            *q = qi;
            break;
        }
    }
}
fclose(pFile);
}

```

CONTOH MASUKAN DAN KELUARAN

```

-----
----- TIC TAC TOE -----
-----

Siapa yang akan main pada giliran pertama?
1. Komputer
2. Anda

>> 1

  |  |
-----
  | 0 |
-----
  |  |

Masukan nomor kotak yang ingin Anda beri tanda : 2

0 | X |
-----
  | 0 |
-----
  |  |

Masukan nomor kotak yang ingin Anda beri tanda : 9

0 | X |
-----
0 | 0 |
-----
  |  | X

Masukan nomor kotak yang ingin Anda beri tanda : 6

0 | X |
-----
0 | 0 | X
-----
0 |  | X

Permainan telah selesai dan dimenangkan oleh komputer

Berikut ini ialah daftar state yang telah dilewati :
1. AAAA0AAAA
2. OXAA0AAAA
3. OXA00AAAX
4. OXA00X0AX

Ketik sembarang angka untuk keluar :

```

Contoh permainan yang mana komputer memulai permainan dan berakhir dengan komputer sebagai pemenang

```

-----
----- TIC TAC TOE -----
-----

Siapa yang akan main pada giliran pertama?
1. Komputer
2. Anda

>> 2

  |  |
-----
  | X |
-----
0 |  |

Masukan nomor kotak yang ingin Anda beri tanda : 1

X |  |
-----
  | X |
-----
0 |  | 0

Masukan nomor kotak yang ingin Anda beri tanda : 1

X |  |
-----
  | X |
-----
0 |  | 0

Masukan nomor kotak yang ingin Anda beri tanda : 3

X |  | X
-----
  | X |
-----
0 | 0 | 0

Permainan telah selesai dan dimenangkan oleh komputer

Berikut ini ialah daftar state yang telah dilewati :
1. AAAAXAOAA
2. XAAAXAOAO
3. XAAAXAOAO
4. XAXAXA000

Ketik sembarang angka untuk keluar :

```

Contoh permainan yang mana pengguna memulai permainan dan sempat terjadi input ke kotak yang telah ada isinya serta berakhir dengan komputer sebagai pemenang

```

1. Komputer
2. Anda

>> 2

  |  |
  ---
  | x |
  ---
0 |  |

Masukan nomor kotak yang ingin Anda beri tanda : 1

x |  |
  ---
  | x |
  ---
0 |  | 0

Masukan nomor kotak yang ingin Anda beri tanda : 8

x | 0 |
  ---
  | x |
  ---
0 | x | 0

Masukan nomor kotak yang ingin Anda beri tanda : 6

x | 0 |
  ---
0 | x | x
  ---
0 | x | 0

Masukan nomor kotak yang ingin Anda beri tanda : 3

x | 0 | x
  ---
0 | x | x
  ---
0 | x | 0

Permainan telah selesai denganimbang

Berikut ini ialah daftar state yang telah dilewati :
1. AAAAXA0AA
2. XAAAXA0AO
3. XOAXA0XO
4. XOA0XXOXO
5. XOX0XXOXO

```

Contoh permainan yang mana pengguna memulai permainan dan berakhir dengan imbang.