**LAPORAN TUGAS BESAR I**

**IF2220**

**TEORI BAHASA FORMAL DAN AUTOMATA**

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

****

**DIBUAT OLEH :**

**ADITYA PUTRA SANTOSA / 13517013**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**BANDUNG**

**2018**

1. **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.

Buat sebuah aplikasi permainan tic-tac-toe sederhana, dimana permainan ini akan dimainkan oleh komputer dan 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. Perlu diperhatikan bahwa program diwajibkan untuk membaca konfigurasi dari file eksternal, dan logika state machine tidak boleh di-hardcode ke program secara langsung. Batasan masalah : Pada langkah pertama permainan, player/CPU dipastikan meletakkan tanda “X” atau “O” di bagian tengah papan​. Setelah itu pastikan bahwa **CPU tidak pernah kalah** di dalam permainan ini. 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.

1. **DFA**
2. **Komputer pertama**
3. **State**

State untuk DFA Komputer Pertama sebanyak 119 state dengan rincian sebagai berikut :

(001)5

(002)5,1,2

(003)5,2,1

(004)5,3,1

(005)5,4,1

(006)5,6,1

(007)5,7,1

(008)5,8,1

(009)5,9,1

(010)5,1,2,3,8

(011)5,1,2,4,8

(012)5,1,2,6,8

(013)5,1,2,7,8

(014)5,1,2,8,4

(015)5,1,2,9,8

(016)5,1,2,8,4,3,6

(017)5,1,2,8,4,6,3

(018)5,1,2,8,4,7,6

(019)5,1,2,8,4,9,6

(020)5,1,2,8,4,6,3,7,9

(021)5,1,2,8,4,6,3,9,7

(022)5,2,1,3,9

(023)5,2,1,4,9

(024)5,2,1,6,9

(025)5,2,1,7,9

(026)5,2,1,8,9

(027)5,2,1,9,4

(028)5,2,1,9,4,3,6

(029)5,2,1,9,4,6,7

(030)5,2,1,9,4,7,6

(031)5,2,1,9,4,8,6

(032)5,3,1,2,9

(033)5,3,1,4,9

(034)5,3,1,6,9

(035)5,3,1,7,9

(036)5,3,1,8,9

(037)5,3,1,9,6

(038)5,3,1,9,6,2,4

(039)5,3,1,9,6,4,2

(040)5,3,1,9,6,7,4

(041)5,3,1,9,6,8,4

(042)5,3,1,9,6,4,2,7,8

(043)5,3,1,9,6,4,2,8,7

(044)5,4,1,2,9

(045)5,4,1,3,9

(046)5,4,1,6,9

(047)5,4,1,7,9

(048)5,4,1,8,9

(049)5,4,1,9,2

(050)5,4,1,9,2,3,8

(051)5,4,1,9,2,6,3

(052)5,4,1,9,2,7,3

(053)5,4,1,9,2,8,3

(054)5,6,1,2,9

(055)5,6,1,3,9

(056)5,6,1,4,9

(057)5,6,1,7,9

(058)5,6,1,8,9

(059)5,6,1,9,3

(060)5,6,1,9,3,2,7

(061)5,6,1,9,3,4,2

(062)5,6,1,9,3,7,2

(063)5,6,1,9,3,8,2

(064)5,7,1,2,9

(065)5,7,1,3,9

(066)5,7,1,4,9

(067)5,7,1,6,9

(068)5,7,1,8,9

(069)5,7,1,9,8

(070)5,7,1,9,8,2,3

(071)5,7,1,9,8,3,2

(072)5,7,1,9,8,4,2

(073)5,7,1,9,8,6,2

(074)5,7,1,9,8,2,3,4,6

(075)5,7,1,9,8,2,3,6,4

(076)5,8,1,2,9

(077)5,8,1,3,9

(078)5,8,1,4,9

(079)5,8,1,6,9

(080)5,8,1,7,9

(081)5,8,1,9,7

(082)5,8,1,9,7,2,3

(083)5,8,1,9,7,3,4

(084)5,8,1,9,7,4,3

(085)5,8,1,9,7,6,3

(086)5,9,1,2,4

(087)5,9,1,3,6

(088)5,9,1,4,2

(089)5,9,1,6,3

(090)5,9,1,7,8

(091)5,9,1,8,7

(092)5,9,1,2,4,3,6

(093)5,9,1,2,4,6,7

(094)5,9,1,2,4,7,6

(095)5,9,1,2,4,8,6

(096)5,9,1,3,6,2,4

(097)5,9,1,3,6,4,2

(098)5,9,1,3,6,7,4

(099)5,9,1,3,6,8,4

(100)5,9,1,3,6,4,2,7,8

(101)5,9,1,3,6,4,2,8,7

(102)5,9,1,4,2,3,8

(103)5,9,1,4,2,6,3

(104)5,9,1,4,2,7,3

(105)5,9,1,4,2,8,3

(106)5,9,1,6,3,2,7

(107)5,9,1,6,3,4,2

(108)5,9,1,6,3,7,2

(109)5,9,1,6,3,8,2

(110)5,9,1,7,8,2,3

(111)5,9,1,7,8,3,2

(112)5,9,1,7,8,4,2

(113)5,9,1,7,8,6,2

(114)5,9,1,7,8,2,3,4,6

(115)5,9,1,7,8,2,3,6,4

(116)5,9,1,8,7,2,3

(117)5,9,1,8,7,3,4

(118)5,9,1,8,7,4,3

(119)5,9,1,8,7,6,3

Dimana 3 digit di awal adalah nomor state dan sisanya adalah nama statenya.

1. **Simbol**

Simbol yang valid bagi DFA ini adalah angka dari 1 s/d 9 yang menyatakan petak petak pada papan permainan Tic-Tac-Toe.

1. **State Awal**

State awal DFA ini adalah (001)5 yang menyatakan CPU menaruh “O” di petak nomor 5.

1. **State Akhir**

Semua state yang merupakan final state sebanyak 92 State dengan rincian :

(010)5,1,2,3,8

(011)5,1,2,4,8

(012)5,1,2,6,8

(013)5,1,2,7,8

(015)5,1,2,9,8

(016)5,1,2,8,4,3,6

(018)5,1,2,8,4,7,6

(019)5,1,2,8,4,9,6

(020)5,1,2,8,4,6,3,7,9

(021)5,1,2,8,4,6,3,9,7

(022)5,2,1,3,9

(023)5,2,1,4,9

(024)5,2,1,6,9

(025)5,2,1,7,9

(026)5,2,1,8,9

(028)5,2,1,9,4,3,6

(029)5,2,1,9,4,6,7

(030)5,2,1,9,4,7,6

(031)5,2,1,9,4,8,6

(032)5,3,1,2,9

(033)5,3,1,4,9

(034)5,3,1,6,9

(035)5,3,1,7,9

(036)5,3,1,8,9

(038)5,3,1,9,6,2,4

(040)5,3,1,9,6,7,4

(041)5,3,1,9,6,8,4

(042)5,3,1,9,6,4,2,7,8

(043)5,3,1,9,6,4,2,8,7

(044)5,4,1,2,9

(045)5,4,1,3,9

(046)5,4,1,6,9

(047)5,4,1,7,9

(048)5,4,1,8,9

(050)5,4,1,9,2,3,8

(051)5,4,1,9,2,6,3

(052)5,4,1,9,2,7,3

(053)5,4,1,9,2,8,3

(054)5,6,1,2,9

(055)5,6,1,3,9

(056)5,6,1,4,9

(057)5,6,1,7,9

(058)5,6,1,8,9

(060)5,6,1,9,3,2,7

(061)5,6,1,9,3,4,2

(062)5,6,1,9,3,7,2

(063)5,6,1,9,3,8,2

(064)5,7,1,2,9

(065)5,7,1,3,9

(066)5,7,1,4,9

(067)5,7,1,6,9

(068)5,7,1,8,9

(071)5,7,1,9,8,3,2

(072)5,7,1,9,8,4,2

(073)5,7,1,9,8,6,2

(074)5,7,1,9,8,2,3,4,6

(075)5,7,1,9,8,2,3,6,4

(076)5,8,1,2,9

(077)5,8,1,3,9

(078)5,8,1,4,9

(079)5,8,1,6,9

(080)5,8,1,7,9

(082)5,8,1,9,7,2,3

(083)5,8,1,9,7,3,4

(084)5,8,1,9,7,4,3

(085)5,8,1,9,7,6,3

(092)5,9,1,2,4,3,6

(093)5,9,1,2,4,6,7

(094)5,9,1,2,4,7,6

(095)5,9,1,2,4,8,6

(096)5,9,1,3,6,2,4

(098)5,9,1,3,6,7,4

(099)5,9,1,3,6,8,4

(100)5,9,1,3,6,4,2,7,8

(101)5,9,1,3,6,4,2,8,7

(102)5,9,1,4,2,3,8

(103)5,9,1,4,2,6,3

(104)5,9,1,4,2,7,3

(105)5,9,1,4,2,8,3

(106)5,9,1,6,3,2,7

(107)5,9,1,6,3,4,2

(108)5,9,1,6,3,7,2

(109)5,9,1,6,3,8,2

(111)5,9,1,7,8,3,2

(112)5,9,1,7,8,4,2

(113)5,9,1,7,8,6,2

(114)5,9,1,7,8,2,3,4,6

(115)5,9,1,7,8,2,3,6,4

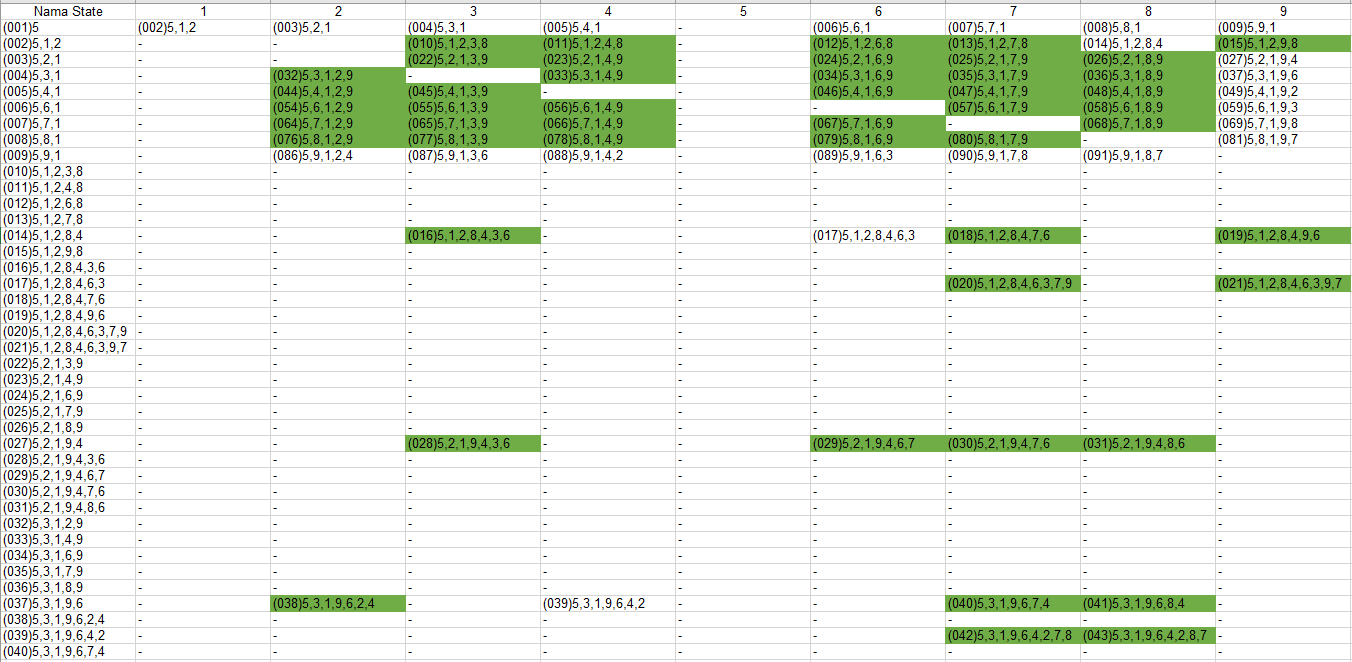
(116)5,9,1,8,7,2,3

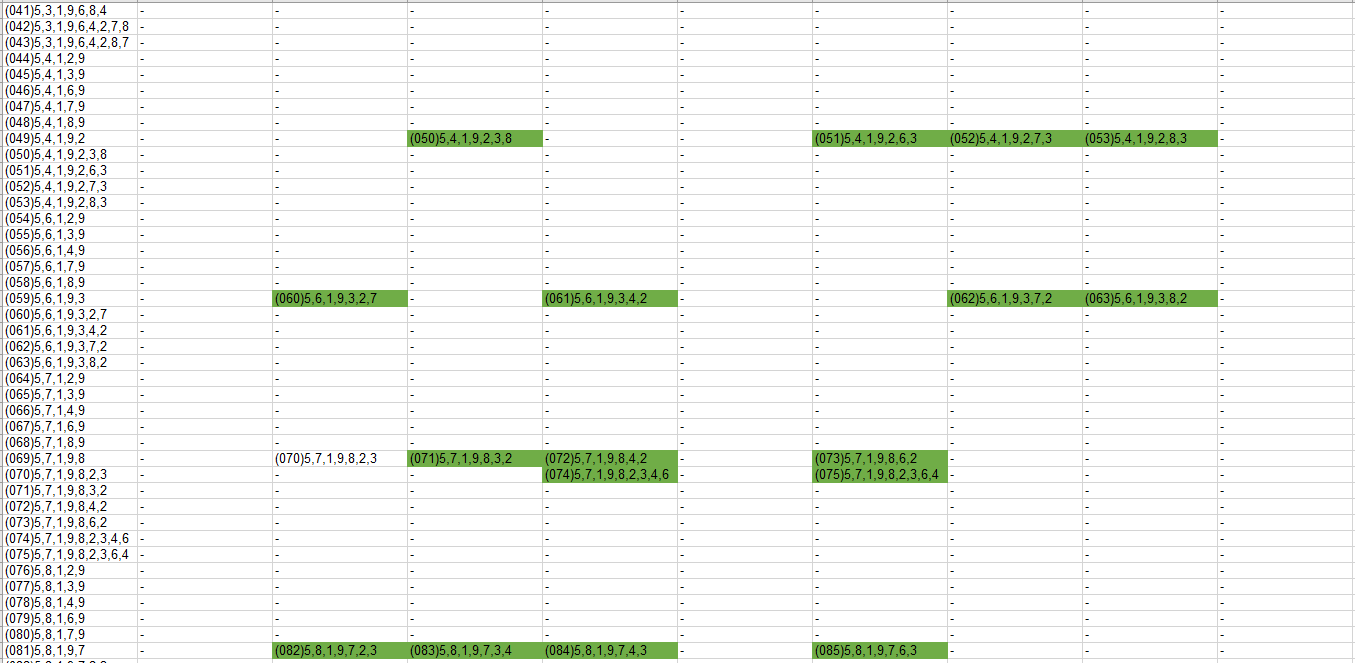
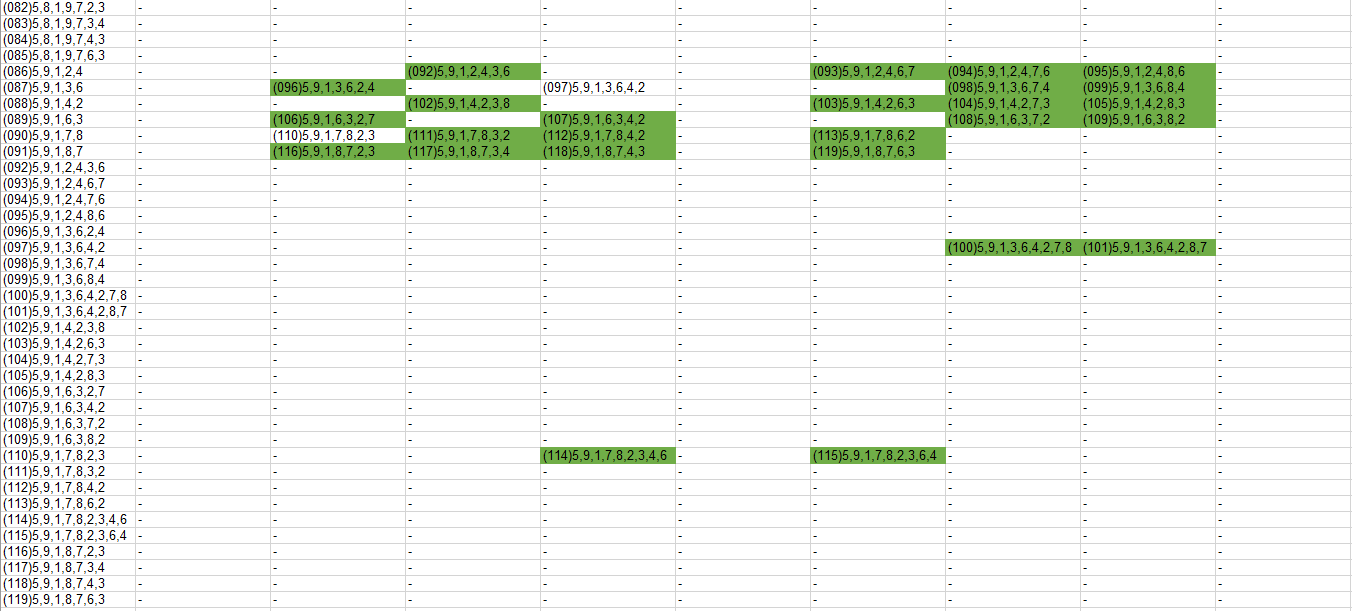
(117)5,9,1,8,7,3,4

(118)5,9,1,8,7,4,3

(119)5,9,1,8,7,6,3

1. **Fungsi Transisi**

Fungsi Transisi untuk DFA ini adalah :



1. **Player pertama**
2. **State**

State untuk DFA Player Pertama sebanyak 117 state dengan rincian sebagai berikut :

(001)5,1

(002)5,1,2,8

(003)5,1,3,7

(004)5,1,4,6

(005)5,1,6,4

(006)5,1,7,3

(007)5,1,8,2

(008)5,1,9,3

(009)5,1,2,8,3,7

(010)5,1,2,8,4,6

(011)5,1,2,8,6,4

(012)5,1,2,8,7,3

(013)5,1,2,8,9,3

(014)5,1,2,8,3,7,4,9

(015)5,1,2,8,3,7,6,4

(016)5,1,2,8,3,7,9,4

(017)5,1,2,8,4,6,3,7

(018)5,1,2,8,4,6,7,3

(019)5,1,2,8,4,6,9,3

(020)5,1,2,8,4,6,3,7,9

(021)5,1,2,8,4,6,7,3,9

(022)5,1,2,8,4,6,9,3,7

(023)5,1,2,8,6,4,3,7

(024)5,1,2,8,6,4,7,3

(025)5,1,2,8,6,4,9,7

(026)5,1,2,8,6,4,7,3,9

(027)5,1,2,8,7,3,4,6

(028)5,1,2,8,7,3,6,4

(029)5,1,2,8,7,3,9,4

(030)5,1,2,8,7,3,4,6,9

(031)5,1,2,8,7,3,6,4,9

(032)5,1,2,8,7,3,9,4,6

(033)5,1,2,8,9,3,4,6

(034)5,1,2,8,9,3,6,4

(035)5,1,2,8,9,3,7,4

(036)5,1,2,8,9,3,4,6,7

(037)5,1,2,8,9,3,6,4,7

(038)5,1,2,8,9,3,7,4,6

(039)5,1,3,7,2,4

(040)5,1,3,7,4,6

(041)5,1,3,7,6,4

(042)5,1,3,7,8,4

(043)5,1,3,7,9,4

(044)5,1,3,7,4,6,2,8

(045)5,1,3,7,4,6,8,2

(046)5,1,3,7,4,6,9,2

(047)5,1,3,7,4,6,2,8,9

(048)5,1,3,7,4,6,8,2,9

(049)5,1,3,7,4,6,9,2,8

(050)5,1,4,6,2,8

(051)5,1,4,6,3,7

(052)5,1,4,6,7,3

(053)5,1,4,6,8,2

(054)5,1,4,6,9,2

(055)5,1,4,6,2,8,3,7

(056)5,1,4,6,2,8,7,3

(057)5,1,4,6,2,8,9,3

(058)5,1,4,6,2,8,3,7,9

(059)5,1,4,6,2,8,7,3,9

(060)5,1,4,6,2,8,9,3,7

(061)5,1,4,6,3,7,2,8

(062)5,1,4,6,3,7,8,2

(063)5,1,4,6,3,7,9,2

(064)5,1,4,6,3,7,2,8,9

(065)5,1,4,6,3,7,8,2,9

(066)5,1,4,6,3,7,9,2,8

(067)5,1,4,6,7,3,2,9

(068)5,1,4,6,7,3,8,2

(069)5,1,4,6,7,3,9,2

(070)5,1,4,6,8,2,3,7

(071)5,1,4,6,8,2,7,3

(072)5,1,4,6,8,2,9,3

(073)5,1,4,6,8,2,3,7,9

(074)5,1,4,6,9,2,3,7

(075)5,1,4,6,9,2,7,3

(076)5,1,4,6,9,2,8,3

(077)5,1,4,6,9,2,3,7,8

(078)5,1,6,4,2,7

(079)5,1,6,4,3,7

(080)5,1,6,4,7,3

(081)5,1,6,4,8,7

(082)5,1,6,4,9,7

(083)5,1,6,4,7,3,2,8

(084)5,1,6,4,7,3,8,2

(085)5,1,6,4,7,3,9,2

(086)5,1,6,4,7,3,2,8,9

(087)5,1,7,3,2,8

(088)5,1,7,3,4,2

(089)5,1,7,3,6,2

(090)5,1,7,3,8,2

(091)5,1,7,3,9,2

(092)5,1,7,3,2,8,4,6

(093)5,1,7,3,2,8,6,4

(094)5,1,7,3,2,8,9,4

(095)5,1,7,3,2,8,4,6,9

(096)5,1,7,3,2,8,6,4,9

(097)5,1,7,3,2,8,9,4,6

(098)5,1,8,2,3,7

(099)5,1,8,2,4,3

(100)5,1,8,2,6,3

(101)5,1,8,2,7,3

(102)5,1,8,2,9,3

(103)5,1,8,2,3,7,4,6

(104)5,1,8,2,3,7,6,4

(105)5,1,8,2,3,7,9,4

(106)5,1,8,2,3,7,4,6,9

(107)5,1,9,3,2,8

(108)5,1,9,3,4,2

(109)5,1,9,3,6,2

(110)5,1,9,3,7,2

(111)5,1,9,3,8,2

(112)5,1,9,3,2,8,4,6

(113)5,1,9,3,2,8,6,4

(114)5,1,9,3,2,8,7,4

(115)5,1,9,3,2,8,4,6,7

(116)5,1,9,3,2,8,6,4,7

(117)5,1,9,3,2,8,7,4,6

1. **Simbol**

Simbol yang valid bagi DFA ini adalah angka dari 1 s/d 9 yang menyatakan petak petak pada papan permainan Tic-Tac-Toe.

1. **State Awal**

State awal DFA ini adalah (001)5,1 yang menyatakan Player menaruh “X” di petak nomor 1 lalu CPU menaruh “O” di petak nomor 5.

1. **State Akhir**

Semua state yang merupakan final state sebanyak 65 State dengan rincian :

(014)5,1,2,8,3,7,4,9

(015)5,1,2,8,3,7,6,4

(016)5,1,2,8,3,7,9,4

(020)5,1,2,8,4,6,3,7,9

(021)5,1,2,8,4,6,7,3,9

(022)5,1,2,8,4,6,9,3,7

(023)5,1,2,8,6,4,3,7

(025)5,1,2,8,6,4,9,7

(026)5,1,2,8,6,4,7,3,9

(030)5,1,2,8,7,3,4,6,9

(031)5,1,2,8,7,3,6,4,9

(032)5,1,2,8,7,3,9,4,6

(036)5,1,2,8,9,3,4,6,7

(037)5,1,2,8,9,3,6,4,7

(038)5,1,2,8,9,3,7,4,6

(039)5,1,3,7,2,4

(041)5,1,3,7,6,4

(042)5,1,3,7,8,4

(043)5,1,3,7,9,4

(047)5,1,3,7,4,6,2,8,9

(048)5,1,3,7,4,6,8,2,9

(049)5,1,3,7,4,6,9,2,8

(058)5,1,4,6,2,8,3,7,9

(059)5,1,4,6,2,8,7,3,9

(060)5,1,4,6,2,8,9,3,7

(064)5,1,4,6,3,7,2,8,9

(065)5,1,4,6,3,7,8,2,9

(066)5,1,4,6,3,7,9,2,8

(067)5,1,4,6,7,3,2,9

(068)5,1,4,6,7,3,8,2

(069)5,1,4,6,7,3,9,2

(071)5,1,4,6,8,2,7,3

(072)5,1,4,6,8,2,9,3

(073)5,1,4,6,8,2,3,7,9

(075)5,1,4,6,9,2,7,3

(076)5,1,4,6,9,2,8,3

(077)5,1,4,6,9,2,3,7,8

(078)5,1,6,4,2,7

(079)5,1,6,4,3,7

(081)5,1,6,4,8,7

(082)5,1,6,4,9,7

(084)5,1,6,4,7,3,8,2

(085)5,1,6,4,7,3,9,2

(086)5,1,6,4,7,3,2,8,9

(088)5,1,7,3,4,2

(089)5,1,7,3,6,2

(090)5,1,7,3,8,2

(091)5,1,7,3,9,2

(095)5,1,7,3,2,8,4,6,9

(096)5,1,7,3,2,8,6,4,9

(097)5,1,7,3,2,8,9,4,6

(099)5,1,8,2,4,3

(100)5,1,8,2,6,3

(101)5,1,8,2,7,3

(102)5,1,8,2,9,3

(104)5,1,8,2,3,7,6,4

(105)5,1,8,2,3,7,9,4

(106)5,1,8,2,3,7,4,6,9

(108)5,1,9,3,4,2

(109)5,1,9,3,6,2

(110)5,1,9,3,7,2

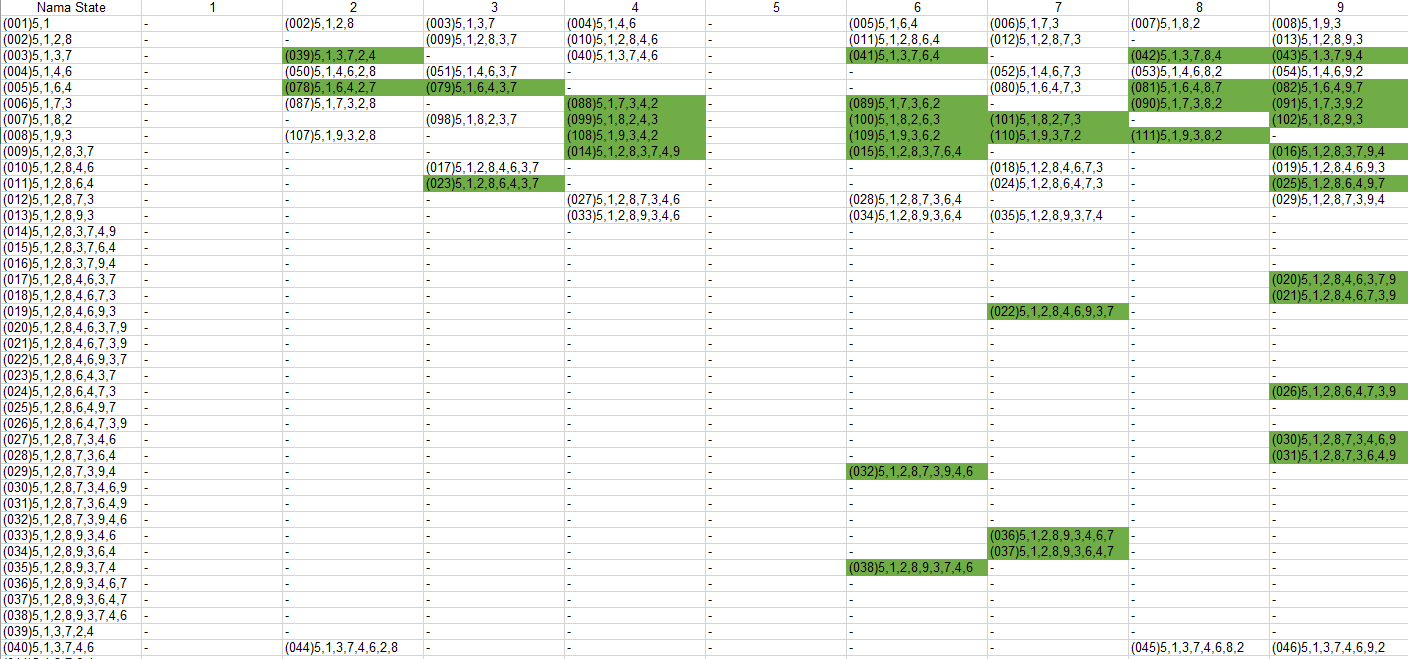
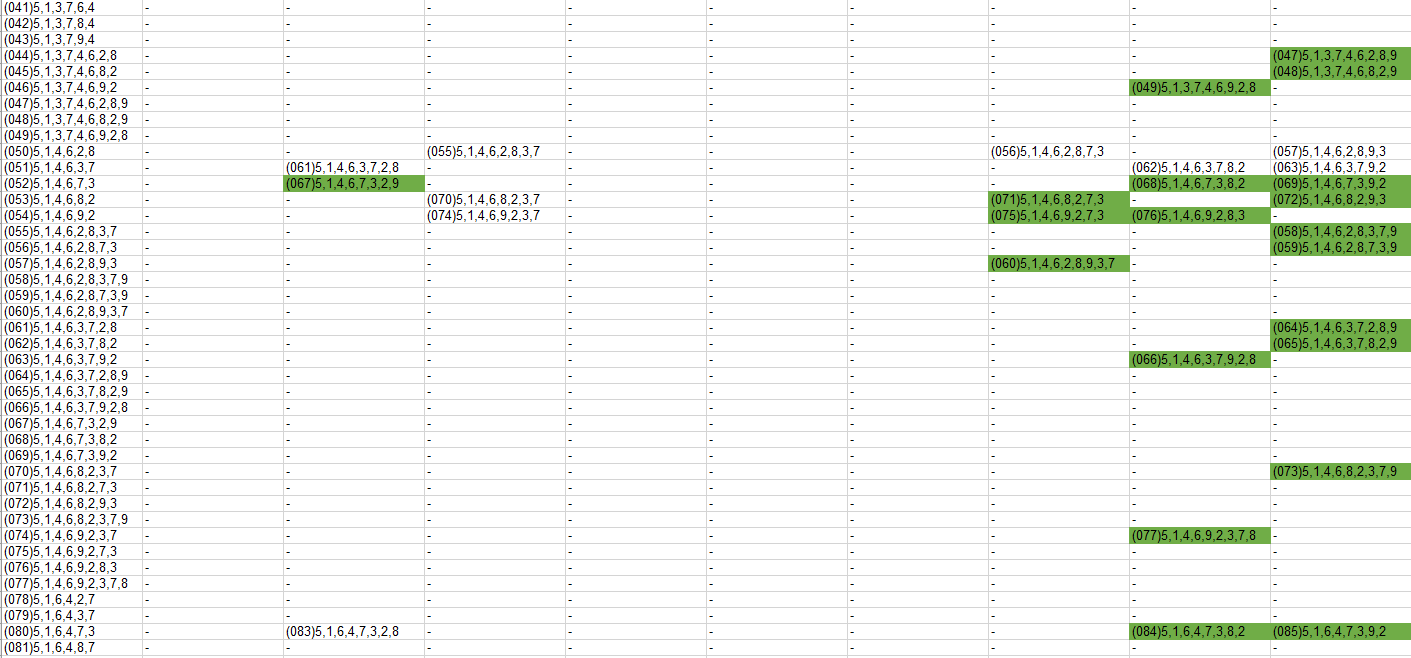
(111)5,1,9,3,8,2

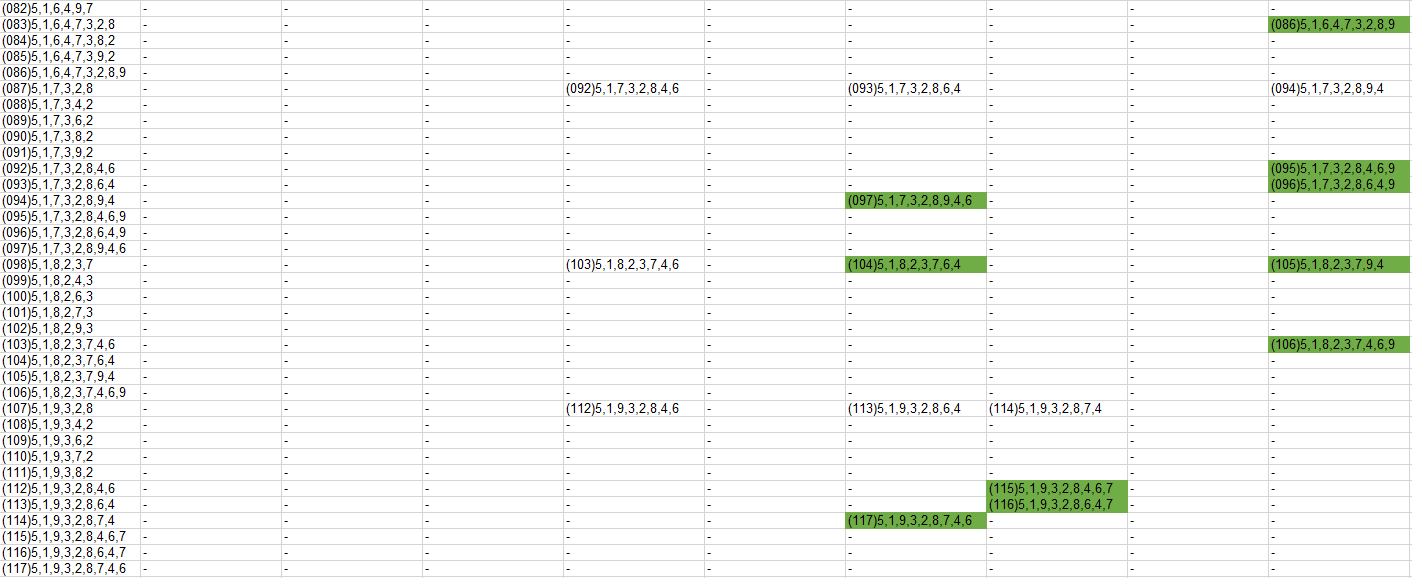
(115)5,1,9,3,2,8,4,6,7

(116)5,1,9,3,2,8,6,4,7

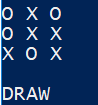
(117)5,1,9,3,2,8,7,4,6

1. **Fungsi Transisi**

Fungsi Transisi untuk DFA ini adalah :



1. **Penjelasan State dan Aksi**

State yang berada di file external berupa (###)A,B,C,D,… dengan ### menyatakan nomor statenya dan A,B,C,D menyatakan langkah langkah yang terlah terjadi. Jadi secara tidak langsung nama state tersebut juga telah mengencode apa saja yang terjadi selama permainan berlangsung. Contohnya adalah State di DFA Player (097)5,1,7,3,2,8,9,4,6 player pada awalnya menaruh X di petak 5, lalu cpu menaruh O di petak 1, lalu player di petak 7, lalu cpu di petak 3, dan seterusnya hingga akhirnya player menaruh X di petak 6 dan permainan berakhir seri karena tidak ada lagi petak yang bisa ditempatkan. Berikut adalah bentuk board yang dihasilkan dari contoh state tersebut :

Cara komputer menentukan dimanakah dia harus menaruh O adalah dengan mengambil angka terakhir pada nama state. Contoh : Pada awalnya DFA mulai di state (001)5,1 komputer mengambil angka paling belakang yaitu 1 dan menaruh O di 1. Lalu player menginput 7 sehingga DFA menuju state sesuai tabel transisi yaitu (006)5,1,7,3 komputer mengambil angka paling belakang yaitu 3 dan menaruh O di petak nomor 3, begitu seterusnya selama permainan belum berakhir (Game Over / salah satu menang).

1. **Source Code**

dfa.h

1. #ifndef DFA\_H
2. #define DFA\_H
4. #include <stdbool.h>
5. #include "papan.h"
7. #define banyakState 120 //Maksimal banyak state
8. #define banyakTipeInput 9
10. **typedef** **struct**
11. {
12. **int** tabelTransisi[banyakState][banyakTipeInput];
13. **char** daftarState[banyakState][23];
14. **int** start;
15. **bool** final[banyakState];
16. **int** curState;
17. **char** stateDilewati[7][23];
18. **int** daftarSimbol[banyakTipeInput];
19. **int** turn;
20. } DFA;
22. //Selektor
23. #define State(D, S, I) (D).tabelTransisi[(S)][(I)]
24. #define Daftar(D, I) (D).daftarState[(I)]
25. #define Final(D,I) (D).final[(I)]
26. #define Current(D) (D).curState
27. #define Start(D) (D).start
28. #define isFinal(D) (D).final[Current(D)]
29. #define StateDilewati(D, I) (D).stateDilewati[(I)]
30. #define Simbol(D,I) (D).daftarSimbol[(I)]
31. #define Turn(D) (D).turn
33. //fungsi
34. **int** nomorState(**char** \*namaState);
35. //mendapat nomor state dari nama state yang berformat (nomor)namaState
37. **void** initDFA(DFA \*D,**char** \*namaDaftarSimbol, **char** \*namaDaftarState,**char** \*namaStateAwal, **char** \*namaTabel, **char** \*namaFinishState,**int** turn);
38. //Load DFA dengan daftar state dan tabel transisinya, serta mengisi pula finish statenya
40. **void** next(DFA \*D, **int** X);
41. //DFA pindah ke state setelah diberi input X
43. **int** getMove(DFA D);
44. //Mendapat move dari current state
46. **bool** inputValid(DFA D,**int** input,PAPAN P);
47. //Mengecek apakah input valid
48. #endif

dfa.c

1. #include <stdbool.h>
2. #include <stdio.h>
3. #include <string.h>
4. #include "dfa.h"
5. //fungsi
7. **int** nomorState(**char**\* namaState)
8. //mendapat nomor state dari nama state yang berformat (nomor)namaState
9. {
10. //"001" = 0\*100 + 0 \* 10 + 1
11. //"012" = 0\*100 + 1 \* 10 + 2
12. **int** ratusan = (**int**)namaState[1] - 48;
13. **int** puluhan = (**int**)namaState[2] - 48;
14. **int** satuan = (**int**)namaState[3] - 48;
15. **return** 100\*ratusan+10\*puluhan+satuan;
16. }
17. **void** initDFA(DFA \*D,**char** \*namaDaftarSimbol, **char** \*namaDaftarState,**char** \*namaStateAwal, **char** \*namaTabel, **char** \*namaFinishState,**int** turn)
18. //Load DFA dengan daftar state dan tabel transisinya, serta mengisi pula finish statenya
19. {
20. **bool** baca;
21. //baca daftar state
22. **FILE** \*fileDaftar = fopen(namaDaftarState, "r");
23. baca = **true**;
24. **int** nomor = 0;
25. **while** (baca)
26. {
27. **char** dummy[23] = "";
28. **if** (fscanf(fileDaftar, "%s", dummy) != EOF)
29. {
30. //Belum eof
31. strcpy(Daftar(\*D, nomor), dummy);
32. nomor++;
33. }
34. **else**
35. {
36. //EOF
37. baca = **false**;
38. }
39. }
40. fclose(fileDaftar);
41. //baca tabel transisi
42. **FILE** \*fileTabel = fopen(namaTabel, "r");
43. baca = **true**;
44. **int** bar = 0;
45. **while** (baca)
46. {
47. **char** temp[9][23];
48. **if** (fscanf(fileTabel, "%s %s %s %s %s %s %s %s %s", temp[0], temp[1], temp[2], temp[3], temp[4], temp[5], temp[6], temp[7], temp[8]) != EOF)
49. {
50. //0 jika tidak pindah state
51. **for** (**int** i = 0; i < 9; i++)
52. {
53. //cek apakah yang dibaca merupakan - atau nama state
54. **if** (temp[i][0] == '-')
55. {
56. //jika - maka tabel transisinya mengacu balik ke stateawal nya
57. State(\*D, bar, i) = bar;
58. }
59. **else**
60. {
61. State(\*D, bar, i) = nomorState(temp[i]);
62. }
63. }
64. bar += 1;
65. }
66. **else**
67. {
68. //EOF
69. baca = **false**;
70. }
71. }
72. fclose(fileTabel);
73. //baca finish state
74. **FILE** \*fileFinish = fopen(namaFinishState, "r");
75. baca = **true**;
76. //assign semua false
77. **for** (**int** i = 0; i < banyakState; i++)
78. {
79. Final(\*D, i) = **false**;
80. }
81. **while** (baca)
82. {
83. **char** dummy[23];
84. **if** (fscanf(fileDaftar, "%s", dummy) != EOF)
85. {
86. //Belum eof
87. //Dapatkan nomor state
88. **int** nomor = nomorState(dummy);
89. Final(\*D, nomor) = **true**;
90. }
91. **else**
92. {
93. //EOF
94. baca = **false**;
95. }
96. }
97. fclose(fileFinish);
98. //Baca start State dari file external
99. **FILE** \*fileStart = fopen(namaStateAwal, "r");
100. fscanf(fileStart, "%d",&Start(\*D));
101. fclose(fileStart);
102. Current(\*D) = Start(\*D);
103. //Baca daftar simbol
104. **FILE** \*fileSimbol = fopen(namaDaftarSimbol, "r");
105. **for**(**int** i = 0;i< banyakTipeInput;i++)
106. {
107. fscanf(fileSimbol,"%d",&Simbol(\*D,i));
108. }
109. fclose(fileSimbol);
111. Turn(\*D) = turn;
112. **for**(**int** i = 0;i<7;i++){
113. strcpy(StateDilewati(\*D,i),"");
114. }
115. }
117. **void** next(DFA \*D, **int** X)
118. //DFA pindah ke state setelah diberi input X
119. {
120. //Prekondisi : Input valid
121. //Final : DFA pindah ke state yang sesuai setelah diberi input X
122. Current(\*D) = State(\*D, Current(\*D)-1, X-1); //-1 karena array dari 0
123. }
125. **int** getMove(DFA D)
126. //Mendapat move dari current state
127. {
128. **char** state[23];
129. strcpy(state, Daftar(D, Current(D)-1));  //-1 karena array dari 0
130. **return** (**int**)state[strlen(state) - 1] - 48; //minus 48 karena '1' = 49
131. }
133. **bool** inputValid(DFA D,**int** input,PAPAN P)
134. //Mengecek apakah input valid
135. {
136. //cek apakah berada di range 1-9
137. **bool** valid = **false**;
138. **for**(**int** i = 0;i<banyakTipeInput;i++)
139. {
140. **if**(Simbol(D,i) == input){
141. valid = **true**;
142. **break**;
143. }
144. }
145. **if** (valid)
146. {
147. //cek apakah diletakan di tempat yang kosong
148. **int** bar = (input - 1) / 3, kol = (input - 1) % 3;
149. **if** (isi(P, bar, kol) == '\_')
150. {
151. **return** **true**;
152. }
153. **else**
154. {
155. **return** **false**;
156. }
157. }
158. **else**
159. {
160. **return** **false**;
161. }
162. }

game.h

1. #ifndef GAME\_H
2. #define GAME\_H
4. #include "dfa.h"
5. #include "papan.h"
7. **void** startGame(DFA\* D, **char** start);
8. //Memulai game dengan dfa D dan karakter awal start (X atau O)
10. #endif

game.c

1. #include <stdio.h>
2. #include <string.h>
3. #include "dfa.h"
4. #include "game.h"
5. #include "papan.h"
7. **void** startGame(DFA \*D, **char** start)
8. //Memulai game dengan dfa D dan karakter awal start (X atau O)
9. {
10. PAPAN game;
11. **int** pilihan, noStateDilewati = 0;
12. **if** (start == 'X')
13. {
14. printf("Game dimulai, player harus menaruh di tengah\n");
15. }
16. **else**
17. {
18. printf("Game dimulai, cpu harus menaruh di tengah\n");
19. }
20. //Catat state awal
21. strcpy(StateDilewati(\*D,noStateDilewati),Daftar(\*D,Start(\*D)-1));  //Isi -1 karena nomor State dimulai dari 1 dan indeksnya 0
22. noStateDilewati += 1;
23. initPapan(&game, start);
24. printBoard(game);
25. Turn(\*D) += 1;               //Ganti turn
26. **while** (cekMenang(game) == 0) //Berhenti saat game telah selesai
27. {
28. **if** (Turn(\*D) % 2)
29. {
30. //jika 1, maka if akan true, menandakan giliran player skrng
31. printf("Sekarang giliranmu, input angka 1-9 yang menandakan tempat di papan : ");
32. scanf("%d", &pilihan);
33. //Validasi input
34. **while** (!inputValid(\*D, pilihan, game))
35. {
36. printf("Input tidak valid, masukkan 1-9 dan di tempat yang kosong : ");
37. scanf("%d", &pilihan);
38. }
39. //Sudah valid
40. //pasang gerakan disana
41. **int** bar = (pilihan - 1) / 3, kol = (pilihan - 1) % 3;
42. isi(game, bar, kol) = 'X';
43. //next state dfa
44. next(D, pilihan);
45. //Catat perubahan state
46. strcpy(StateDilewati(\*D,noStateDilewati),Daftar(\*D,Current(\*D)-1));
47. //Current(\*D)-1 karena nomor state dimulai dari 1, dan di indeks dimulai dari 0
48. noStateDilewati += 1;
49. //print papan ke layar
50. printBoard(game);
51. }
52. **else**
53. {
54. //jika 0, maka if akan false, menandakan giliran komp skrng
55. printf("Sekarang giliran cpu\n");
56. //Dapatkan move paling optimal
57. **int** optimal = getMove(\*D);
58. **int** bar = (optimal - 1) / 3, kol = (optimal - 1) % 3;
59. isi(game, bar, kol) = 'O';
60. //print papan ke layar
61. printBoard(game);
62. }
63. Turn(\*D) += 1; //Ganti turn
64. }
65. //Game selesai, output hasil permainan
66. **int** kondisiAkhir = cekMenang(game);
67. **if** (kondisiAkhir == 1)
68. {
69. printf("CPU MENANG\n");
70. }
71. **else** **if** (kondisiAkhir == 2)
72. {
73. printf("PLAYER MENANG\n");
74. }
75. **else** **if** (kondisiAkhir == 3)
76. {
77. printf("DRAW\n");
78. }
79. //Print semua state yang dilewati
80. printf("State yang telah dilewati : \n");
81. **for** (**int** i = 0; i < noStateDilewati; i++)    //Loop dari 0 sampai Turn(\*D)-1 karena nilai Turn saat selesai adalah banyak state yang dilewati+1
82. {
83. printf("%s\n",StateDilewati(\*D,i));
84. }
85. }

papan.h

1. #ifndef BOARD\_H
2. #define BOARD\_H
4. #include<stdbool.h>
6. **typedef** **struct**{
7. **char** data[3][3];
8. }PAPAN;
10. #define isi(P,R,C) (P).data[(R)][(C)]
11. #define giliran(P) (P).giliran
13. **void** initPapan(PAPAN\* P,**char** start);
14. //Init papan dengan 0 semua, giliran = true
15. **void** printBoard(PAPAN P);
16. //Print papan ke layar
17. **int** cekMenang(PAPAN P);
18. //return 0 jika belum ada yang menang,1 jika O menang, 2 jika X menang, 3 jika draw
19. #endif

papan.c

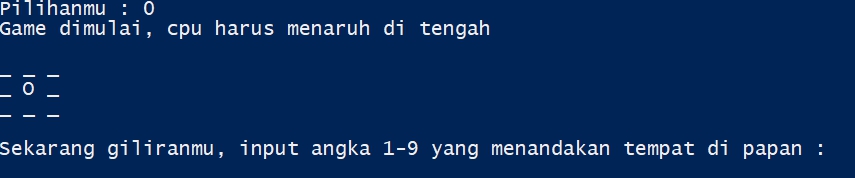
1. #include <stdbool.h>
2. #include <stdio.h>
3. #include "papan.h"
5. **void** initPapan(PAPAN \*P, **char** start)
6. //Init papan dengan 0 semua, giliran = true
7. {
8. **for** (**int** i = 0; i < 3; i++)
9. {
10. **for** (**int** j = 0; j < 3; j++)
11. {
12. isi(\*P, i, j) = '\_';
13. }
14. }
15. isi(\*P, 1, 1) = start;
16. }
17. **void** printBoard(PAPAN P)
18. //Print papan ke layar
19. {
20. printf("\n");
21. **for** (**int** i = 0; i < 3; i++)
22. {
23. **for** (**int** j = 0; j < 3; j++)
24. {
25. printf("%c ", isi(P, i, j));
26. }
27. printf("\n");
28. }
29. printf("\n");
30. }
31. **int** cekMenang(PAPAN P)
32. //return 0 jika belum ada yang menang,1 jika O menang, 2 jika X menang, 3 jika draw
33. {
34. //Cek baris
35. **for** (**int** i = 0; i < 3; i++)
36. {
37. **if** ((isi(P, i, 0) == isi(P, i, 1)) && (isi(P, i, 0) == isi(P, i, 2)))
38. {
39. **char** cek = isi(P, i, 0);
40. **if** (cek == 'O')
41. {
42. **return** 1;
43. }
44. **else** **if** (cek == 'X')
45. {
46. **return** 2;
47. }
48. }
49. }
50. //Cek kolom
51. **for** (**int** i = 0; i < 3; i++)
52. {
53. **if** ((isi(P, 0, i) == isi(P, 1, i)) && (isi(P, 0, i) == isi(P, 2, i)))
54. {
55. **char** cek = isi(P, 0, i);
56. **if** (cek == 'O')
57. {
58. **return** 1;
59. }
60. **else** **if** (cek == 'X')
61. {
62. **return** 2;
63. }
64. }
65. }
66. //Cek diagonal
67. **if** ((isi(P, 0, 0) == isi(P, 1, 1)) && (isi(P, 0, 0) == isi(P, 2, 2)))
68. {
69. **char** cek = isi(P, 0, 0);
70. **if** (cek == 'O')
71. {
72. **return** 1;
73. }
74. **else** **if** (cek == 'X')
75. {
76. **return** 2;
77. }
78. }
79. //Cari apakah ada yang kosong
80. **for** (**int** i = 0; i < 3; i++)
81. {
82. **for** (**int** j = 0; j < 3; j++)
83. {
84. **if** (isi(P, i, j) == '\_')
85. {
86. **return** 0;
87. }
88. }
89. }
90. //Jika sampai sini, berarti draw
91. **return** 3;
92. }

main.c

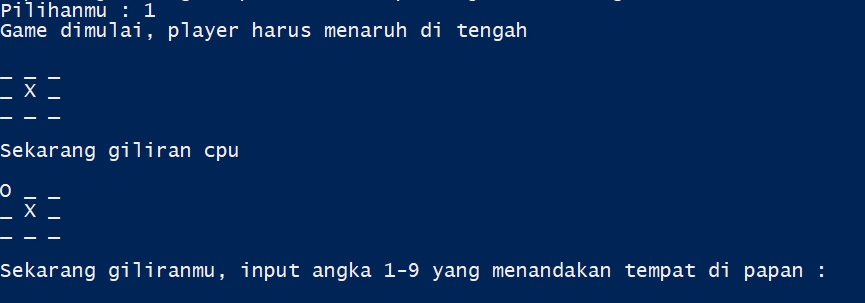
1. #include<stdio.h>
2. #include "dfa.h"
3. #include "game.h"
5. **int** main()
6. {
7. **int** pilihan;
8. printf("Selamat datang di permainan \n");
9. printf("'########:'####::'######:::::'########::::'###:::::'######:::::'########::'#######::'########:\n");
10. printf("... ##..::. ##::'##... ##::::... ##..::::'## ##:::'##... ##::::... ##..::'##.... ##: ##.....::\n");
11. printf("::: ##::::: ##:: ##:::..:::::::: ##:::::'##:. ##:: ##:::..:::::::: ##:::: ##:::: ##: ##:::::::\n");
12. printf("::: ##::::: ##:: ##::::::::::::: ##::::'##:::. ##: ##::::::::::::: ##:::: ##:::: ##: ######:::\n");
13. printf("::: ##::::: ##:: ##::::::::::::: ##:::: #########: ##::::::::::::: ##:::: ##:::: ##: ##...::::\n");
14. printf("::: ##::::: ##:: ##::: ##::::::: ##:::: ##.... ##: ##::: ##::::::: ##:::: ##:::: ##: ##:::::::\n");
15. printf("::: ##::::'####:. ######:::::::: ##:::: ##:::: ##:. ######:::::::: ##::::. #######:: ########:\n");
16. printf(":::..:::::....:::......:::::::::..:::::..:::::..:::......:::::::::..::::::.......:::........::\n");
17. printf("Input 0 jika ingin cpu duluan, input 1 jika anda ingin mulai duluan\n");
18. printf("Pilihanmu : ");
19. scanf("%d",&pilihan);
20. **if**(pilihan==0){
21. DFA cpuFirst;
22. initDFA(&cpuFirst,"State/daftarSimbol.txt","State/daftarStates.txt","State/stateAwal.txt","State/states.txt","State/finishState.txt",0);
23. startGame(&cpuFirst,'O');
24. }**else**{
25. DFA playerFirst;
26. initDFA(&playerFirst,"State/daftarSimbol.txt","State/daftarStatesPlayer.txt","State/stateAwalPlayer.txt","State/statesPlayer.txt","State/finishStatePlayer.txt",1);
27. startGame(&playerFirst,'X');
28. }
29. **return** 0;
30. }
31. **Contoh Input dan Output Program**

Tampilan awal :

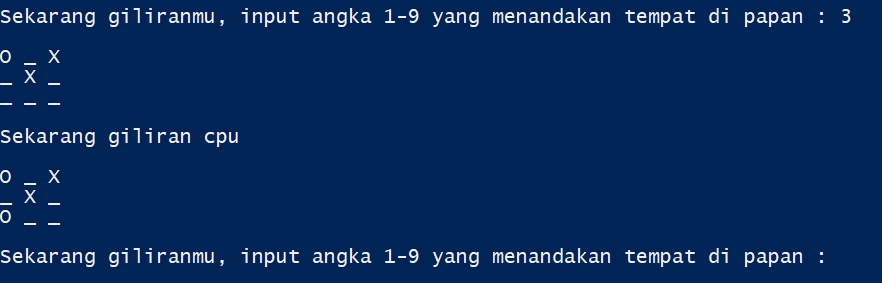


Cpu mulai duluan :

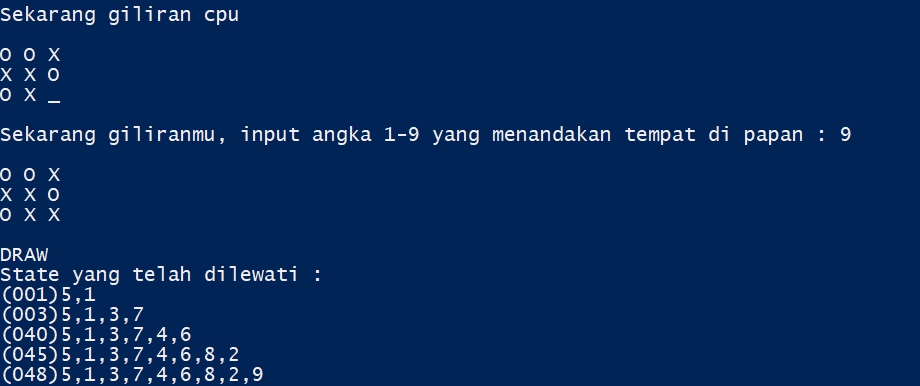
Player mulai duluan :

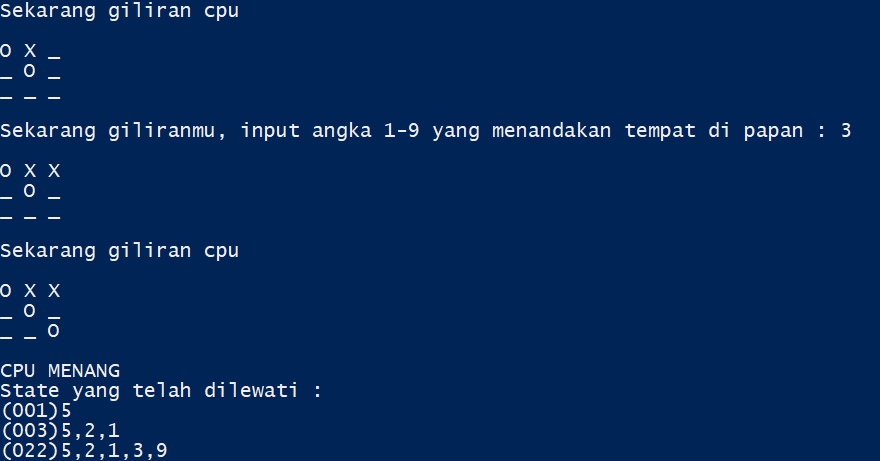


Contoh input pengguna dan reaksi CPU :

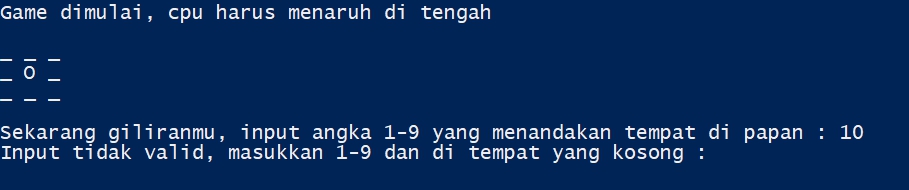


Game Over (Draw) dan state yang telah dilewati :



Game Over(CPU menang) dan state yang telah dilewati:

Input tidak valid (bukan 1-9):



Input tidak valid (bukan petak kosong):

