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

A. 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.

B. DFA

I. Komputer pertama

1. 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.

2. **Simbol**

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

3. **State Awal**

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

4. **State Akhir**

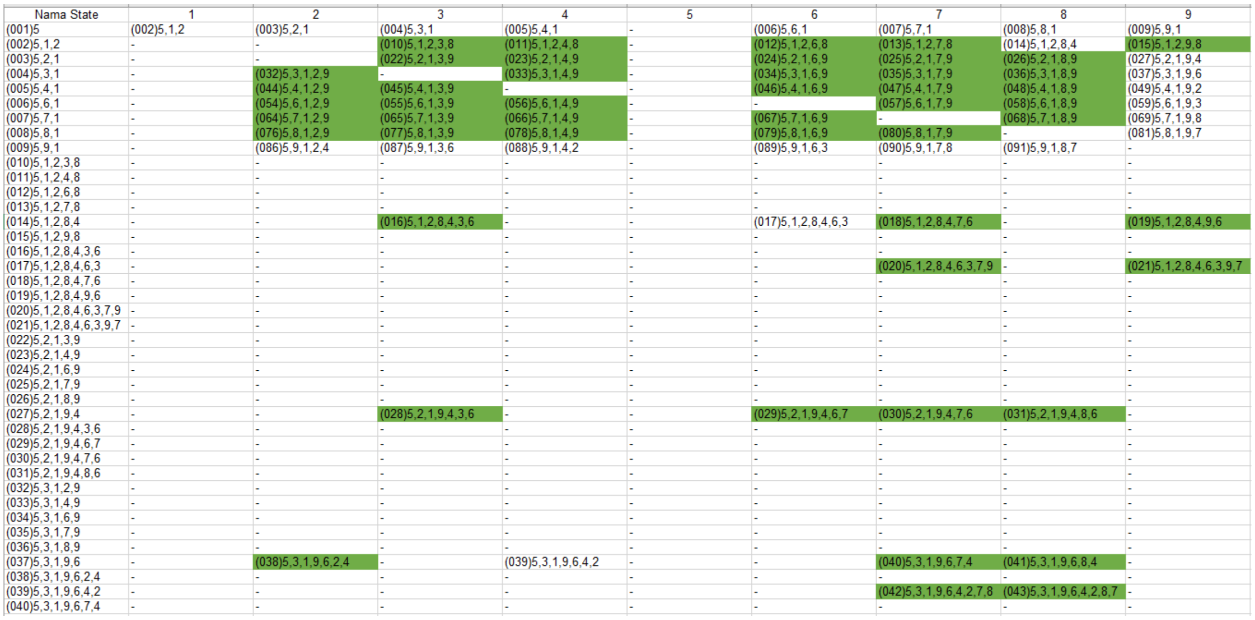
   Semua state yang merupakan final state sebanyak 92 State dengan rincian :

   (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

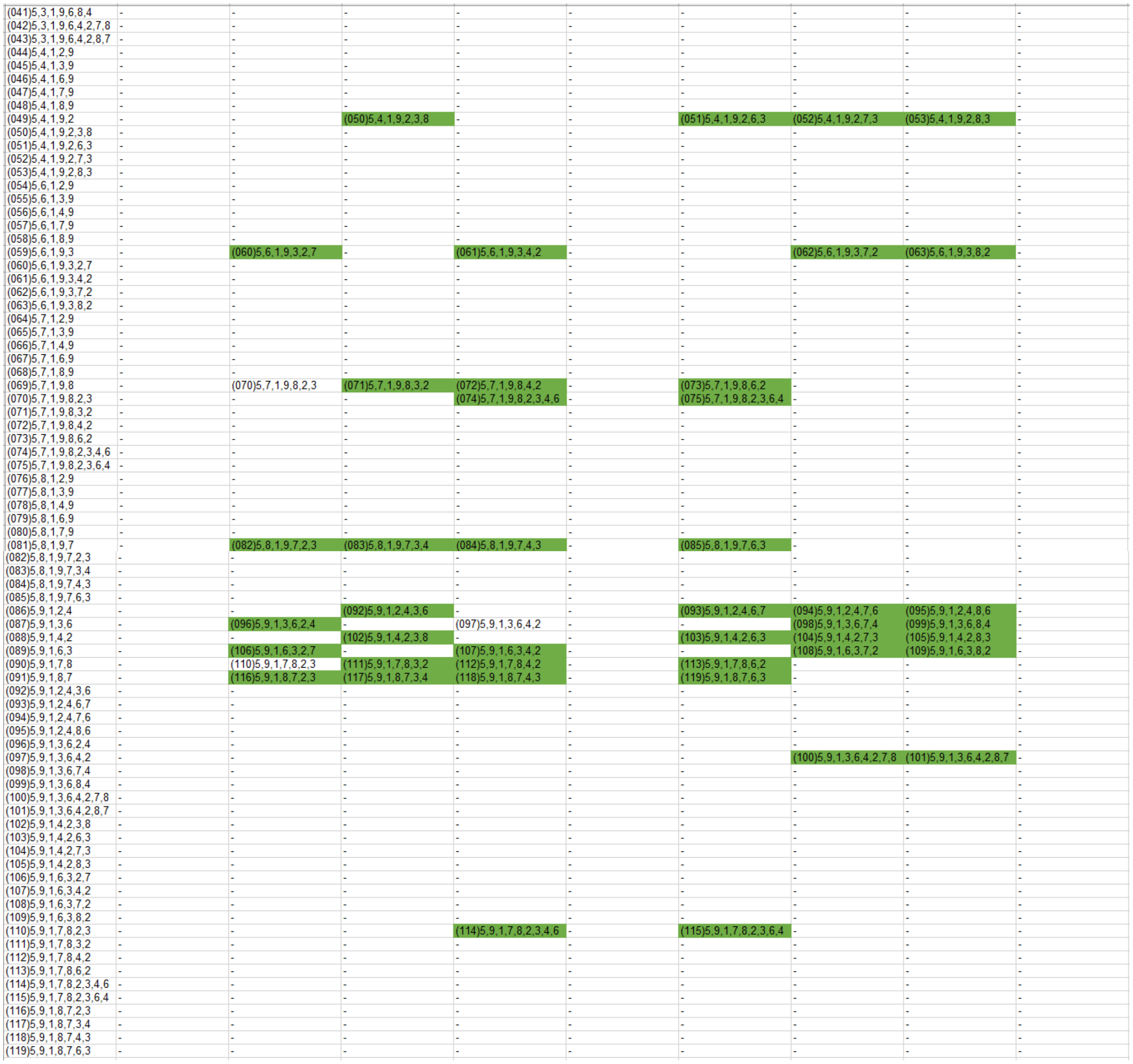
(094)5,9,1,2,4,7,6
(095)5,9,1,2,4,8,6
(096)5,9,1,3,6,2,4
(098)5,9,1,3,6,7,4
(099)5,9,1,3,6,8,4
(100)5,9,1,3,6,4,2,7,8
(101)5,9,1,3,6,4,2,8,7
(102)5,9,1,4,2,3,8
(103)5,9,1,4,2,6,3
(104)5,9,1,4,2,7,3
(105)5,9,1,4,2,8,3
(106)5,9,1,6,3,2,7
(107)5,9,1,6,3,4,2
(108)5,9,1,6,3,7,2
(109)5,9,1,6,3,8,2
(111)5,9,1,7,8,3,2
(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

5. Fungsi Transisi

Fungsi Transisi untuk DFA ini adalah :



Nama State	1	2	3	4	5	6	7	8	9
(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
(002)5,1,2	-	-	(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
(003)5,2,1	-	-	(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
(004)5,3,1	-	(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
(005)5,4,1	-	(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
(006)5,6,1	-	(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
(007)5,7,1	-	(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
(008)5,8,1	-	(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
(009)5,9,1	-	(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	-
(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	-	-	(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
(015)5,1,2,9,8	-	-	-	-	-	-	-	-	-
(016)5,1,2,8,4,3,6	-	-	-	-	-	-	-	-	-
(017)5,1,2,8,4,6,3	-	-	-	-	-	-	(020)5,1,2,8,4,6,3,7,9	-	(021)5,1,2,8,4,6,3,9,7
(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	-
(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	-
(038)5,3,1,9,6,2,4	-	-	-	-	-	-	-	-	-
(039)5,3,1,9,6,4,2	-	-	-	-	-	-	(042)5,3,1,9,6,4,2,7,8	(043)5,3,1,9,6,4,2,8,7	-
(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	-
(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	-
(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	-	-	-
(070)5,7,1,9,8,2,3	-	-	-	(074)5,7,1,9,8,2,3,4,6	-	(075)5,7,1,9,8,2,3,6,4	-	-	-
(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	-	-	-
(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	-	-	(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	-
(087)5,9,1,3,6	-	(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	-
(088)5,9,1,4,2	-	-	(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	-
(089)5,9,1,6,3	-	(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	-
(090)5,9,1,7,8	-	(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	-	-	-
(091)5,9,1,8,7	-	(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	-	-	-
(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	-	-	-	-	-	-	(100)5,9,1,3,6,4,2,7,8	(101)5,9,1,3,6,4,2,8,7	-
(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	-	-	-	(114)5,9,1,7,8,2,3,4,6	-	(115)5,9,1,7,8,2,3,6,4	-	-	-
(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	-	-	-	-	-	-	-	-	-

II. Player pertama

1. 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

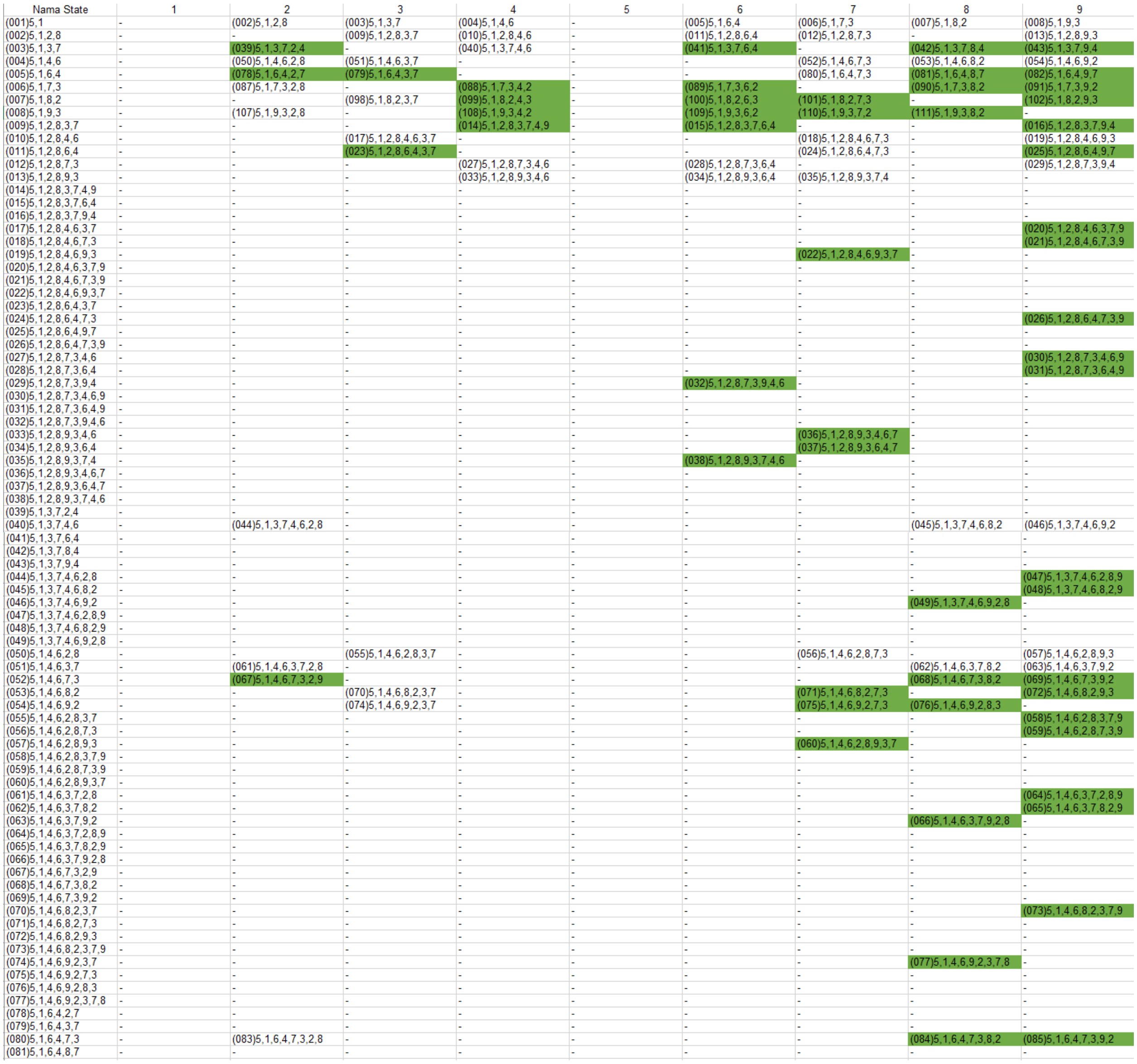
2. **Simbol**
   Simbol yang valid bagi DFA ini adalah angka dari 1 s/d 9 yang menyatakan petak petak pada papan permainan Tic-Tac-Toe.
3. **State Awal**
   State awal DFA ini adalah (001)5,1 yang menyatakan Player menaruh "X" di petak nomor 1 lalu CPU menaruh "O" di petak nomor 5.
4. **State Akhir**
   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

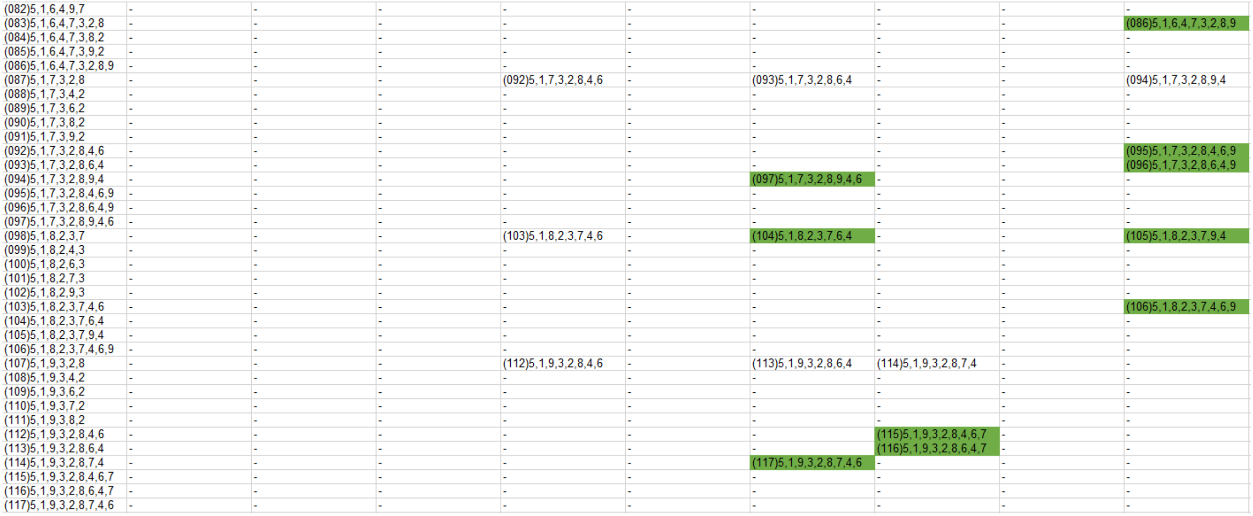
(102)5,1,8,2,9,3
(104)5,1,8,2,3,7,6,4
(105)5,1,8,2,3,7,9,4
(106)5,1,8,2,3,7,4,6,9
(108)5,1,9,3,4,2
(109)5,1,9,3,6,2
(110)5,1,9,3,7,2
(111)5,1,9,3,8,2
(115)5,1,9,3,2,8,4,6,7
(116)5,1,9,3,2,8,6,4,7
(117)5,1,9,3,2,8,7,4,6

5. Fungsi Transisi

Fungsi Transisi untuk DFA ini adalah :



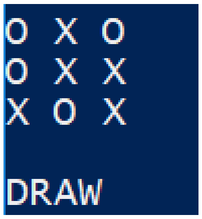
Nama State	1	2	3	4	5	6	7	8	9
(001)5,1	-	(002)5,1,2,8	(003)5,1,3,7	(004)5,1,4,6	-	(005)5,1,6,4	(006)5,1,7,3	(007)5,1,8,2	(008)5,1,9,3
(002)5,1,2,8	-	-	(009)5,1,2,8,3,7	(010)5,1,2,8,4,6	-	(011)5,1,2,8,6,4	(012)5,1,2,8,7,3	-	(013)5,1,2,8,9,3
(003)5,1,3,7	-	(039)5,1,3,7,2,4	-	(040)5,1,3,7,4,6	-	(041)5,1,3,7,6,4	-	(042)5,1,3,7,8,4	(043)5,1,3,7,9,4
(004)5,1,4,6	-	(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
(005)5,1,6,4	-	(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
(006)5,1,7,3	-	(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
(007)5,1,8,2	-	-	(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
(008)5,1,9,3	-	(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	-
(009)5,1,2,8,3,7	-	-	-	(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
(010)5,1,2,8,4,6	-	-	(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
(011)5,1,2,8,6,4	-	-	(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
(012)5,1,2,8,7,3	-	-	-	(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
(013)5,1,2,8,9,3	-	-	-	(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	-	-
(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	-	-	-	-	-	-	-	-	(020)5,1,2,8,4,6,3,7,9
(018)5,1,2,8,4,6,7,3	-	-	-	-	-	-	-	-	(021)5,1,2,8,4,6,7,3,9
(019)5,1,2,8,4,6,9,3	-	-	-	-	-	-	(022)5,1,2,8,4,6,9,3,7	-	-
(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	-	-	-	-	-	-	-	-	(026)5,1,2,8,6,4,7,3,9
(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	-	-	-	-	-	-	-	-	(030)5,1,2,8,7,3,4,6,9
(028)5,1,2,8,7,3,6,4	-	-	-	-	-	-	-	-	(031)5,1,2,8,7,3,6,4,9
(029)5,1,2,8,7,3,9,4	-	-	-	-	-	(032)5,1,2,8,7,3,9,4,6	-	-	-
(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	-	-	-	-	-	-	(036)5,1,2,8,9,3,4,6,7	-	-
(034)5,1,2,8,9,3,6,4	-	-	-	-	-	-	(037)5,1,2,8,9,3,6,4,7	-	-
(035)5,1,2,8,9,3,7,4	-	-	-	-	-	(038)5,1,2,8,9,3,7,4,6	-	-	-
(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	-	-	-	-	-	-	-	-	-
(040)5,1,3,7,4,6	-	(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
(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	-	-	-	-	-	-	-	-	(047)5,1,3,7,4,6,2,8,9
(045)5,1,3,7,4,6,8,2	-	-	-	-	-	-	-	-	(048)5,1,3,7,4,6,8,2,9
(046)5,1,3,7,4,6,9,2	-	-	-	-	-	-	-	(049)5,1,3,7,4,6,9,2,8	-
(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	-	-	(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
(051)5,1,4,6,3,7	-	(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
(052)5,1,4,6,7,3	-	(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
(053)5,1,4,6,8,2	-	-	(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
(054)5,1,4,6,9,2	-	-	(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	-
(055)5,1,4,6,2,8,3,7	-	-	-	-	-	-	-	-	(058)5,1,4,6,2,8,3,7,9
(056)5,1,4,6,2,8,7,3	-	-	-	-	-	-	-	-	(059)5,1,4,6,2,8,7,3,9
(057)5,1,4,6,2,8,9,3	-	-	-	-	-	-	(060)5,1,4,6,2,8,9,3,7	-	-
(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	-	-	-	-	-	-	-	-	(064)5,1,4,6,3,7,2,8,9
(062)5,1,4,6,3,7,8,2	-	-	-	-	-	-	-	-	(065)5,1,4,6,3,7,8,2,9
(063)5,1,4,6,3,7,9,2	-	-	-	-	-	-	-	(066)5,1,4,6,3,7,9,2,8	-
(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	-	-	-	-	-	-	-	-	(073)5,1,4,6,8,2,3,7,9
(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	-	-	-	-	-	-	-	(077)5,1,4,6,9,2,3,7,8	-
(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	-	(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
(081)5,1,6,4,8,7	-	-	-	-	-	-	-	-	-



(082)5,1,6,4,9,7	-	-	-	-	-	-	-	-	-
(083)5,1,6,4,7,3,2,8	-	-	-	-	-	-	-	-	(086)5,1,6,4,7,3,2,8,9
(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	-	-	-	(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
(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	-	-	-	-	-	-	-	-	(095)5,1,7,3,2,8,4,6,9
(093)5,1,7,3,2,8,6,4	-	-	-	-	-	-	-	-	(096)5,1,7,3,2,8,6,4,9
(094)5,1,7,3,2,8,9,4	-	-	-	-	-	(097)5,1,7,3,2,8,9,4,6	-	-	-
(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	-	-	-	(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
(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	-	-	-	-	-	-	-	-	(106)5,1,8,2,3,7,4,6,9
(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	-	-	-	(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	-	-
(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	-	-	-	-	-	-	(115)5,1,9,3,2,8,4,6,7	-	-
(113)5,1,9,3,2,8,6,4	-	-	-	-	-	-	(116)5,1,9,3,2,8,6,4,7	-	-
(114)5,1,9,3,2,8,7,4	-	-	-	-	-	(117)5,1,9,3,2,8,7,4,6	-	-	-
(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	-	-	-	-	-	-	-	-	-

C. 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 :



```
O X O
O X X
X O X

DRAW
```

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).

D. Source Code

dfa.h

```c
#ifndef DFA_H
#define DFA_H

#include <stdbool.h>
#include "papan.h"

#define banyakState 120 //Maksimal banyak state
#define banyakTipeInput 9

typedef struct
{
    int tabelTransisi[banyakState][banyakTipeInput];
    char daftarState[banyakState][23];
    int start;
    bool final[banyakState];
    int curState;
    char stateDilewati[7][23];
    int daftarSimbol[banyakTipeInput];
    int turn;
} DFA;

//Selektor
#define State(D, S, I) (D).tabelTransisi[(S)][(I)]
#define Daftar(D, I) (D).daftarState[(I)]
#define Final(D,I) (D).final[(I)]
#define Current(D) (D).curState
#define Start(D) (D).start
#define isFinal(D) (D).final[Current(D)]
#define StateDilewati(D, I) (D).stateDilewati[(I)]
#define Simbol(D,I) (D).daftarSimbol[(I)]
#define Turn(D) (D).turn

//fungsi
int nomorState(char *namaState);
//mendapat nomor state dari nama state yang berformat (nomor)namaState

void initDFA(DFA *D,char *namaDaftarSimbol, char *namaDaftarState,char *namaStateAwal, char *namaTabel, char *namaFinishState,int turn);
//Load DFA dengan daftar state dan tabel transisinya, serta mengisi pula finish statenya

void next(DFA *D, int X);
//DFA pindah ke state setelah diberi input X

int getMove(DFA D);
//Mendapat move dari current state

bool inputValid(DFA D,int input,PAPAN P);
//Mengecek apakah input valid
#endif
```

dfa.c

```c
#include <stdbool.h>
#include <stdio.h>
#include <string.h>
#include "dfa.h"
//fungsi

int nomorState(char* namaState)
//mendapat nomor state dari nama state yang berformat (nomor)namaState
{
    //"001" = 0*100 + 0 * 10 + 1
    //"012" = 0*100 + 1 * 10 + 2
    int ratusan = (int)namaState[1] - 48;
    int puluhan = (int)namaState[2] - 48;
    int satuan = (int)namaState[3] - 48;
    return 100*ratusan+10*puluhan+satuan;
}
void initDFA(DFA *D,char *namaDaftarSimbol, char *namaDaftarState,char *namaStateAwal, char *namaTabel, char *namaFinishState,int turn)
//Load DFA dengan daftar state dan tabel transisinya, serta mengisi pula finish statenya
{
    bool baca;
    //baca daftar state
    FILE *fileDaftar = fopen(namaDaftarState, "r");
    baca = true;
    int nomor = 0;
    while (baca)
    {
        char dummy[23] = "";
        if (fscanf(fileDaftar, "%s", dummy) != EOF)
        {
            //Belum eof
            strcpy(Daftar(*D, nomor), dummy);
            nomor++;
        }
        else
        {
            //EOF
            baca = false;
        }
    }
    fclose(fileDaftar);
    //baca tabel transisi
    FILE *fileTabel = fopen(namaTabel, "r");
    baca = true;
    int bar = 0;
    while (baca)
    {
        char temp[9][23];
        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)
        {
            //0 jika tidak pindah state
            for (int i = 0; i < 9; i++)
            {
                //cek apakah yang dibaca merupakan - atau nama state
                if (temp[i][0] == '-')
                {
                    //jika - maka tabel transisinya mengacu balik ke stateawal nya
                    State(*D, bar, i) = bar;
                }
                else
                {
                    State(*D, bar, i) = nomorState(temp[i]);
                }
            }
            bar += 1;
        }
        else
        {
            //EOF
            baca = false;
        }
    }
    fclose(fileTabel);
    //baca finish state
    FILE *fileFinish = fopen(namaFinishState, "r");
    baca = true;
    //assign semua false
    for (int i = 0; i < banyakState; i++)
    {
        Final(*D, i) = false;
    }
    while (baca)
    {
        char dummy[23];
        if (fscanf(fileDaftar, "%s", dummy) != EOF)
        {
            //Belum eof
            //Dapatkan nomor state
            int nomor = nomorState(dummy);
            Final(*D, nomor) = true;
        }
        else
        {
            //EOF
            baca = false;
        }
    }
    fclose(fileFinish);
    //Baca start State dari file external
    FILE *fileStart = fopen(namaStateAwal, "r");
           fscanf(fileStart, "%d",&Start(*D));
           fclose(fileStart);
           Current(*D) = Start(*D);
           //Baca daftar simbol
           FILE *fileSimbol = fopen(namaDaftarSimbol, "r");
           for(int i = 0;i< banyakTipeInput;i++)
           {
               fscanf(fileSimbol,"%d",&Simbol(*D,i));
           }
           fclose(fileSimbol);

           Turn(*D) = turn;
           for(int i = 0;i<7;i++){
               strcpy(StateDilewati(*D,i),"");
           }
       }

       void next(DFA *D, int X)
       //DFA pindah ke state setelah diberi input X
       {
           //Prekondisi : Input valid
           //Final : DFA pindah ke state yang sesuai setelah diberi input X
           Current(*D) = State(*D, Current(*D)-1, X-1); //-1 karena array dari 0
       }

       int getMove(DFA D)
       //Mendapat move dari current state
       {
           char state[23];
           strcpy(state, Daftar(D, Current(D)-1));  //-1 karena array dari 0
           return (int)state[strlen(state) - 1] - 48; //minus 48 karena '1' = 49
       }

       bool inputValid(DFA D,int input,PAPAN P)
       //Mengecek apakah input valid
       {
           //cek apakah berada di range 1-9
           bool valid = false;
           for(int i = 0;i<banyakTipeInput;i++)
           {
               if(Simbol(D,i) == input){
                   valid = true;
                   break;
               }
           }
           if (valid)
           {
               //cek apakah diletakan di tempat yang kosong
               int bar = (input - 1) / 3, kol = (input - 1) % 3;
               if (isi(P, bar, kol) == '_')
               {
                   return true;
               }
               else
               {
                   return false;
               }
           }
           else
           {
               return false;
           }
       }
```

game.h

```c
#ifndef GAME_H
#define GAME_H

#include "dfa.h"
#include "papan.h"

void startGame(DFA* D, char start);
//Memulai game dengan dfa D dan karakter awal start (X atau O)

#endif
```

game.c

```c
#include <stdio.h>
#include <string.h>
#include "dfa.h"
#include "game.h"
#include "papan.h"

void startGame(DFA *D, char start)
//Memulai game dengan dfa D dan karakter awal start (X atau O)
{
    PAPAN game;
    int pilihan, noStateDilewati = 0;
    if (start == 'X')
    {
        printf("Game dimulai, player harus menaruh di tengah\n");
    }
    else
    {
        printf("Game dimulai, cpu harus menaruh di tengah\n");
    }
    //Catat state awal
    strcpy(StateDilewati(*D,noStateDilewati),Daftar(*D,Start(*D)-1));  //Isi -1 karena nomor State dimulai dari 1 dan indeksnya 0
    noStateDilewati += 1;
    initPapan(&game, start);
    printBoard(game);
    Turn(*D) += 1;               //Ganti turn
    while (cekMenang(game) == 0) //Berhenti saat game telah selesai
    {
        if (Turn(*D) % 2)
        {
            //jika 1, maka if akan true, menandakan giliran player skrng
            printf("Sekarang giliranmu, input angka 1-9 yang menandakan tempat di papan : ");
            scanf("%d", &pilihan);
            //Validasi input
            while (!inputValid(*D, pilihan, game))
            {
                printf("Input tidak valid, masukkan 1-9 dan di tempat yang kosong : ");
                scanf("%d", &pilihan);
            }
            //Sudah valid
            //pasang gerakan disana
            int bar = (pilihan - 1) / 3, kol = (pilihan - 1) % 3;
            isi(game, bar, kol) = 'X';
            //next state dfa
            next(D, pilihan);
            //Catat perubahan state
            strcpy(StateDilewati(*D,noStateDilewati),Daftar(*D,Current(*D)-1));
            //Current(*D)-1 karena nomor state dimulai dari 1, dan di indeks dimulai dari 0
            noStateDilewati += 1;
            //print papan ke layar
            printBoard(game);
        }
        else
        {
            //jika 0, maka if akan false, menandakan giliran komp skrng
            printf("Sekarang giliran cpu\n");
            //Dapatkan move paling optimal
            int optimal = getMove(*D);
            int bar = (optimal - 1) / 3, kol = (optimal - 1) % 3;
            isi(game, bar, kol) = 'O';
            //print papan ke layar
            printBoard(game);
        }
        Turn(*D) += 1; //Ganti turn
    }
    //Game selesai, output hasil permainan
    int kondisiAkhir = cekMenang(game);
    if (kondisiAkhir == 1)
    {
        printf("CPU MENANG\n");
    }
    else if (kondisiAkhir == 2)
    {
        printf("PLAYER MENANG\n");
    }
    else if (kondisiAkhir == 3)
    {
        printf("DRAW\n");
    }
    //Print semua state yang dilewati
    printf("State yang telah dilewati : \n");
    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
    {
        printf("%s\n",StateDilewati(*D,i));
    }
}
```

papan.h

```c
#ifndef BOARD_H
#define BOARD_H

#include<stdbool.h>

typedef struct{
    char data[3][3];
}PAPAN;

#define isi(P,R,C) (P).data[(R)][(C)]
#define giliran(P) (P).giliran

void initPapan(PAPAN* P,char start);
//Init papan dengan 0 semua, giliran = true
void printBoard(PAPAN P);
//Print papan ke layar
int cekMenang(PAPAN P);
//return 0 jika belum ada yang menang,1 jika O menang, 2 jika X menang, 3 jika draw
#endif
```

papan.c

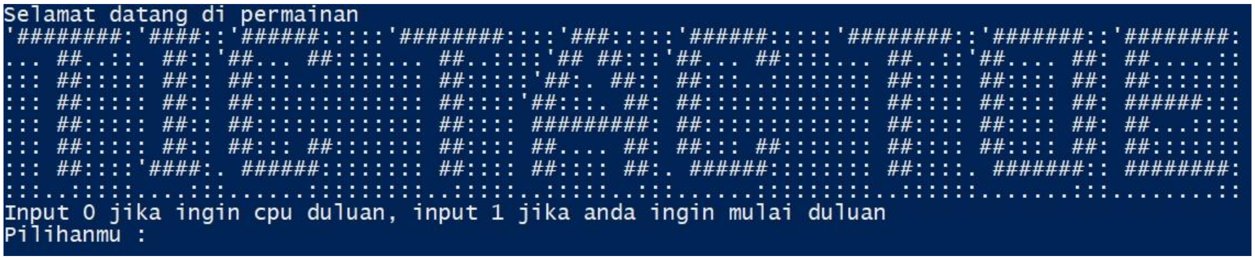
```c
#include <stdbool.h>
#include <stdio.h>
#include "papan.h"

void initPapan(PAPAN *P, char start)
//Init papan dengan 0 semua, giliran = true
{
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            isi(*P, i, j) = '_';
        }
    }
    isi(*P, 1, 1) = start;
}
void printBoard(PAPAN P)
//Print papan ke layar
{
    printf("\n");
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            printf("%c ", isi(P, i, j));
        }
        printf("\n");
    }
    printf("\n");
}
int cekMenang(PAPAN P)
//return 0 jika belum ada yang menang,1 jika O menang, 2 jika X menang, 3 jika draw
{
    //Cek baris
    for (int i = 0; i < 3; i++)
    {
        if ((isi(P, i, 0) == isi(P, i, 1)) && (isi(P, i, 0) == isi(P, i, 2)))
        {
            char cek = isi(P, i, 0);
            if (cek == 'O')
            {
                return 1;
            }
            else if (cek == 'X')
            {
                return 2;
            }
        }
    }
    //Cek kolom
    for (int i = 0; i < 3; i++)
    {
        if ((isi(P, 0, i) == isi(P, 1, i)) && (isi(P, 0, i) == isi(P, 2, i)))
        {
            char cek = isi(P, 0, i);
            if (cek == 'O')
            {
                return 1;
            }
            else if (cek == 'X')
            {
                return 2;
            }
        }
    }
    //Cek diagonal
    if ((isi(P, 0, 0) == isi(P, 1, 1)) && (isi(P, 0, 0) == isi(P, 2, 2)))
    {
        char cek = isi(P, 0, 0);
        if (cek == 'O')
        {
            return 1;
        }
        else if (cek == 'X')
        {
            return 2;
        }
    }
    //Cari apakah ada yang kosong
    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            if (isi(P, i, j) == '_')
            {
                return 0;
            }
        }
    }
    //Jika sampai sini, berarti draw
    return 3;
}
```

main.c

```c
#include<stdio.h>
#include "dfa.h"
#include "game.h"

int main()
{
    int pilihan;
    printf("Selamat datang di permainan \n");
    printf("'########:'####::'######::::'########::::'###:::::'######::::'########::'#######::'########:\n");
    printf("... ##..::. ##::'##... ##:::... ##..::::'## ##:::'##... ##:::... ##..::'##.... ##: ##.....::\n");
    printf("::: ##::::: ##:: ##:::..:::::::: ##:::::'##:. ##:: ##:::..:::::::: ##:::: ##:::: ##: ##:::::::\n");
    printf("::: ##::::: ##:: ##::::::::::::: ##::::'##:::. ##: ##::::::::::::: ##:::: ##:::: ##: ######:::\n");
    printf("::: ##::::: ##:: ##::::::::::::: ##:::: #########: ##::::::::::::: ##:::: ##:::: ##: ##...::::\n");
    printf("::: ##::::: ##:: ##::: ##::::::: ##:::: ##.... ##: ##::: ##::::::: ##:::: ##:::: ##: ##:::::::\n");
    printf("::: ##::::'####:. ######:::::::: ##:::: ##:::: ##:. ######:::::::: ##::::. #######:: ########:\n");
    printf(":::..:::::....:::......:::::::::..:::::..:::::..:::......:::::::::..::::::.......:::........::\n");
    printf("Input 0 jika ingin cpu duluan, input 1 jika anda ingin mulai duluan\n");
    printf("Pilihanmu : ");
    scanf("%d",&pilihan);
    if(pilihan==0){
        DFA cpuFirst;
        initDFA(&cpuFirst,"State/daftarSimbol.txt","State/daftarStates.txt","State/stateAwal.txt","State/states.txt","State/finishState.txt",0);
        startGame(&cpuFirst,'O');
    }else{
        DFA playerFirst;
        initDFA(&playerFirst,"State/daftarSimbol.txt","State/daftarStatesPlayer.txt","State/stateAwalPlayer.txt","State/statesPlayer.txt","State/finishStatePlayer.txt",1);
        startGame(&playerFirst,'X');
    }
    return 0;
}
```

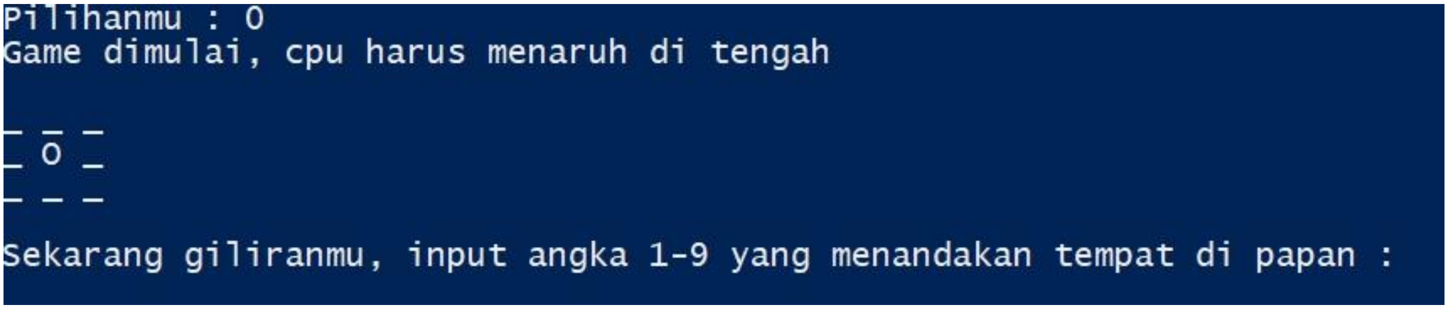
E. Contoh Input dan Output Program

Tampilan awal :



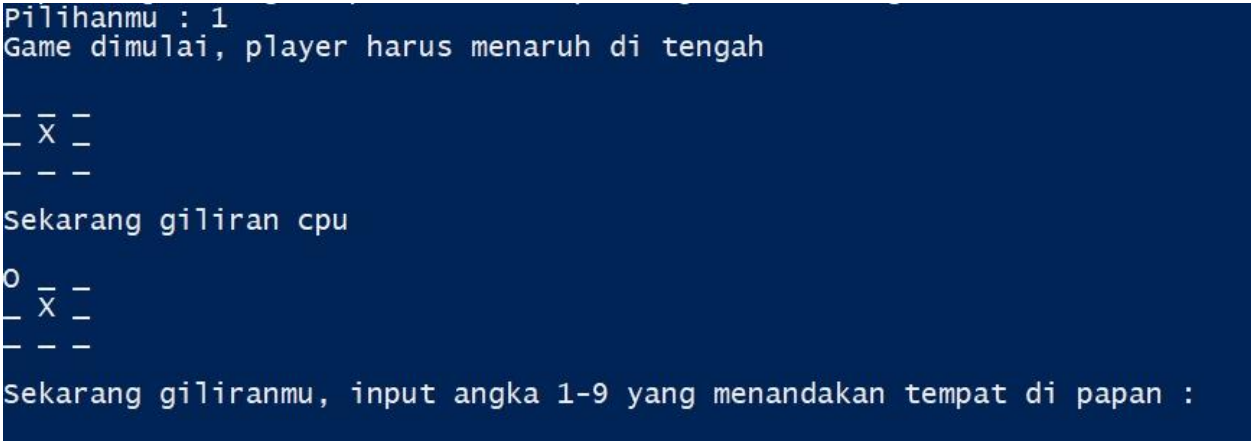
```
Selamat datang di permainan
'########:'####::'######::::'########::::'###:::::'######::::'########::'#######::'########:
... ##..::. ##::'##... ##:::... ##..::::'## ##:::'##... ##:::... ##..::'##.... ##: ##.....::
::: ##::::: ##:: ##:::..:::::::: ##:::::'##:. ##:: ##:::..:::::::: ##:::: ##:::: ##: ##:::::::
::: ##::::: ##:: ##::::::::::::: ##::::'##:::. ##: ##::::::::::::: ##:::: ##:::: ##: ######:::
::: ##::::: ##:: ##::::::::::::: ##:::: #########: ##::::::::::::: ##:::: ##:::: ##: ##...::::
::: ##::::: ##:: ##::: ##::::::: ##:::: ##.... ##: ##::: ##::::::: ##:::: ##:::: ##: ##:::::::
::: ##::::'####:. ######:::::::: ##:::: ##:::: ##:. ######:::::::: ##::::. #######:: ########:
:::..:::::....:::......:::::::::..:::::..:::::..:::......:::::::::..::::::.......:::........::
Input 0 jika ingin cpu duluan, input 1 jika anda ingin mulai duluan
Pilihanmu :
```

Cpu mulai duluan :



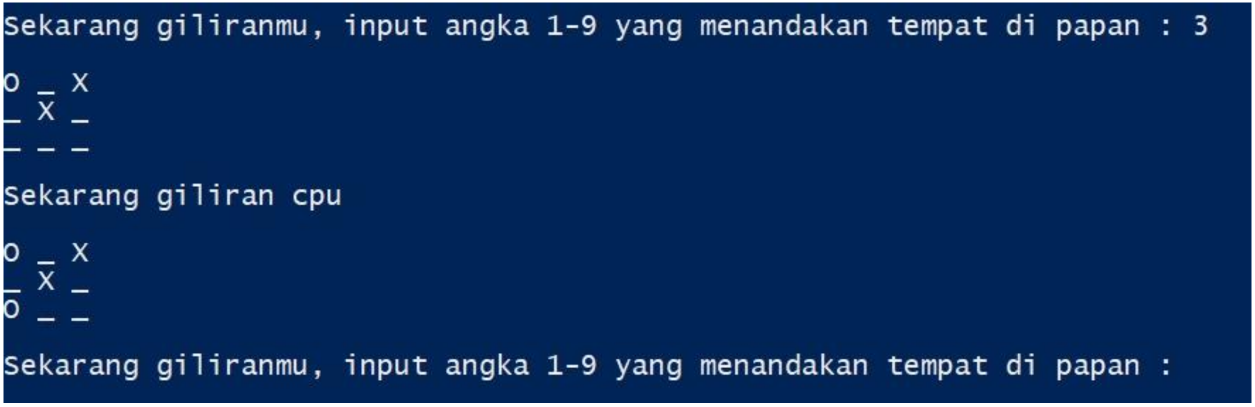
```
Pilihanmu : 0
Game dimulai, cpu harus menaruh di tengah

_ _ _
_ O _
_ _ _

Sekarang giliranmu, input angka 1-9 yang menandakan tempat di papan :
```

Player mulai duluan :



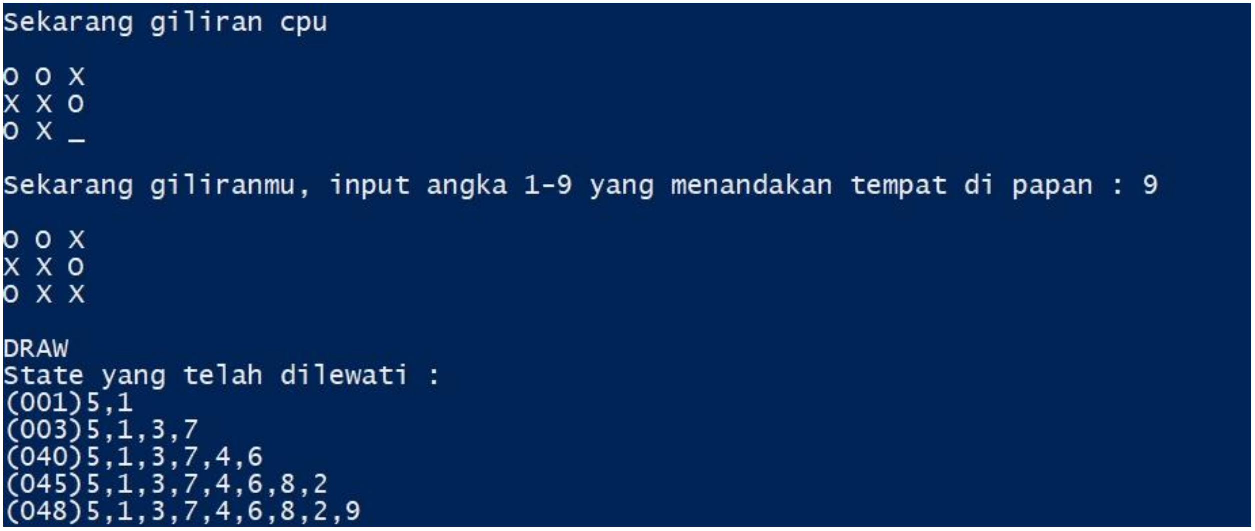
```
Pilihanmu : 1
Game dimulai, player harus menaruh di tengah

_ _ _
_ X _
_ _ _

Sekarang giliran cpu

O _ _
_ X _
_ _ _

Sekarang giliranmu, input angka 1-9 yang menandakan tempat di papan :
```

Contoh input pengguna dan reaksi CPU :



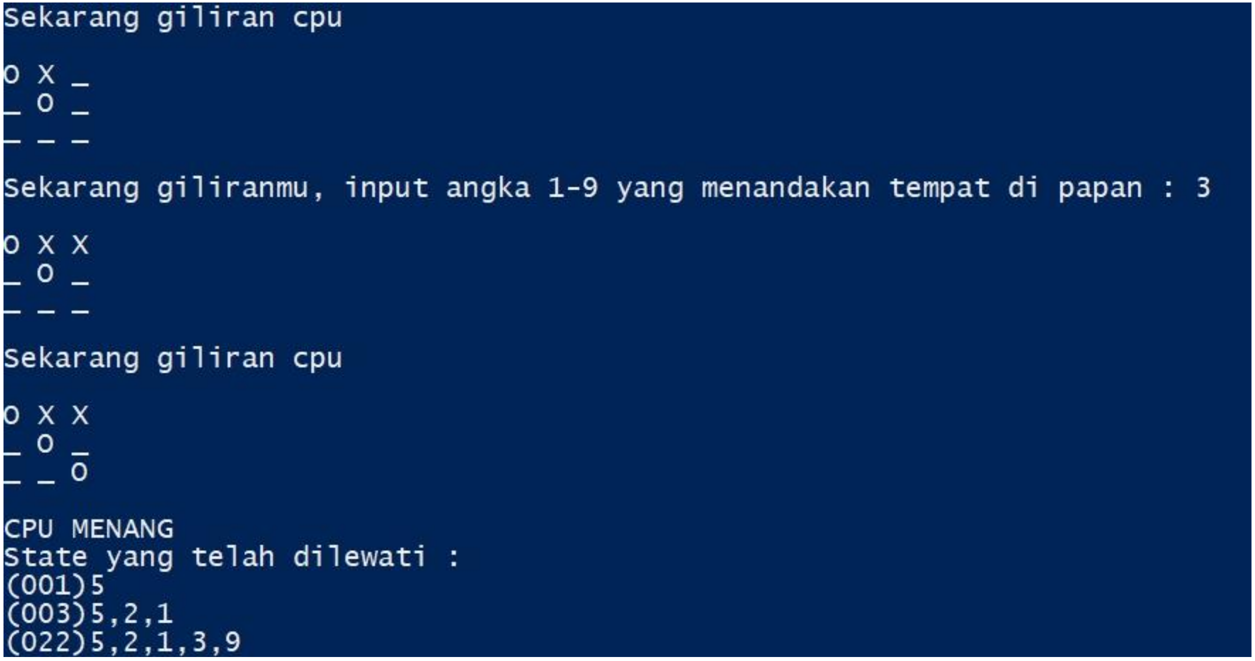
```
Sekarang giliranmu, input angka 1-9 yang menandakan tempat di papan : 3

O _ X
_ X _
_ _ _

Sekarang giliran cpu

O _ X
_ X _
O _ _

Sekarang giliranmu, input angka 1-9 yang menandakan tempat di papan :
```

Game Over (Draw) dan state yang telah dilewati :



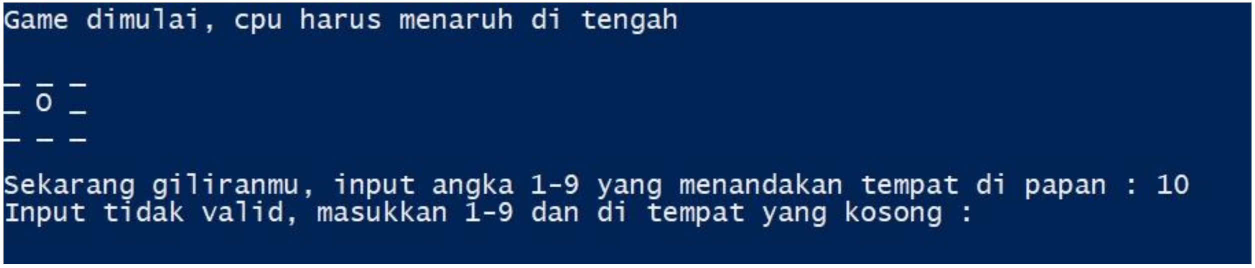
```
Sekarang giliran cpu

O O X
X X O
O X _

Sekarang giliranmu, input angka 1-9 yang menandakan tempat di papan : 9

O O X
X X O
O X X

DRAW
State yang telah dilewati :
(001)5,1
(003)5,1,3,7
(040)5,1,3,7,4,6
(045)5,1,3,7,4,6,8,2
(048)5,1,3,7,4,6,8,2,9
```

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



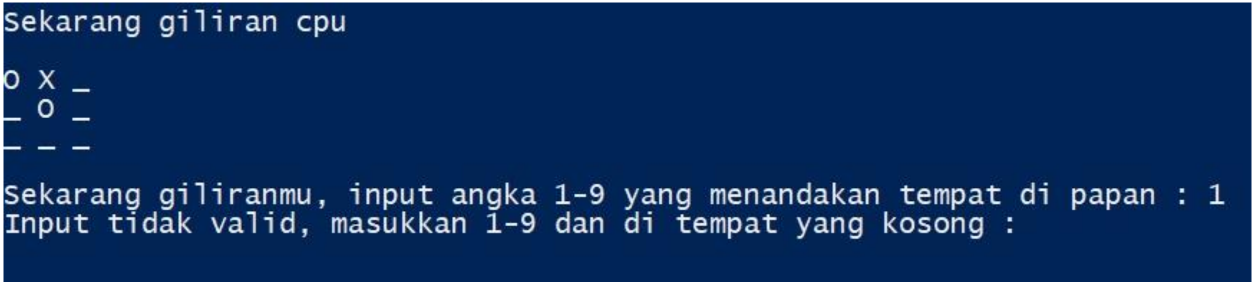
```
Sekarang giliran cpu

O X _
_ O _
_ _ _

Sekarang giliranmu, input angka 1-9 yang menandakan tempat di papan : 3

O X X
_ O _
_ _ _

Sekarang giliran cpu

O X X
_ O _
_ _ O

CPU MENANG
State yang telah dilewati :
(001)5
(003)5,2,1
(022)5,2,1,3,9
```

Input tidak valid (bukan 1-9):



```
Game dimulai, cpu harus menaruh di tengah

_ _ _
_ O _
_ _ _

Sekarang giliranmu, input angka 1-9 yang menandakan tempat di papan : 10
Input tidak valid, masukkan 1-9 dan di tempat yang kosong :
```

Input tidak valid (bukan petak kosong):



```
Sekarang giliran cpu

O X _
_ O _
_ _ _

Sekarang giliranmu, input angka 1-9 yang menandakan tempat di papan : 1
Input tidak valid, masukkan 1-9 dan di tempat yang kosong :
```