LAPORAN TUGAS AKHIR

SISTEM OPERASI KELAS B

Project : Tic Tac Toe

Disusun Oleh:

M. Awaluddin ilham zuhri 21083010112

PROGRAM STUDI SAINS DATA

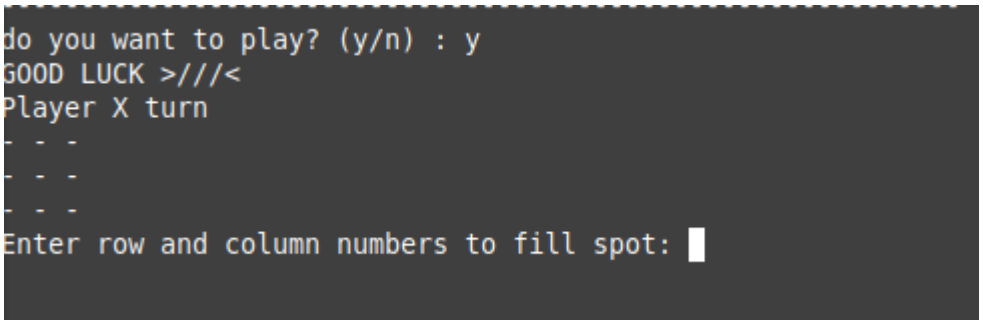
FAKULTAS ILMU KOMPUTER

UNIVERSITAS PEMBANGUNAN NEGERI “VETERAN” JAWA TIMUR

1. Tampilan Awal

Tampilan awal berisi nama dari game yaitu “TIC TAC TOE”, dan juga berisi pertanyaan “do you want to play? (y/n)”, pada pertanyaan ini user akan diminta menjawab dengan “y” yang berarti yess dan “n” yang berarti no.

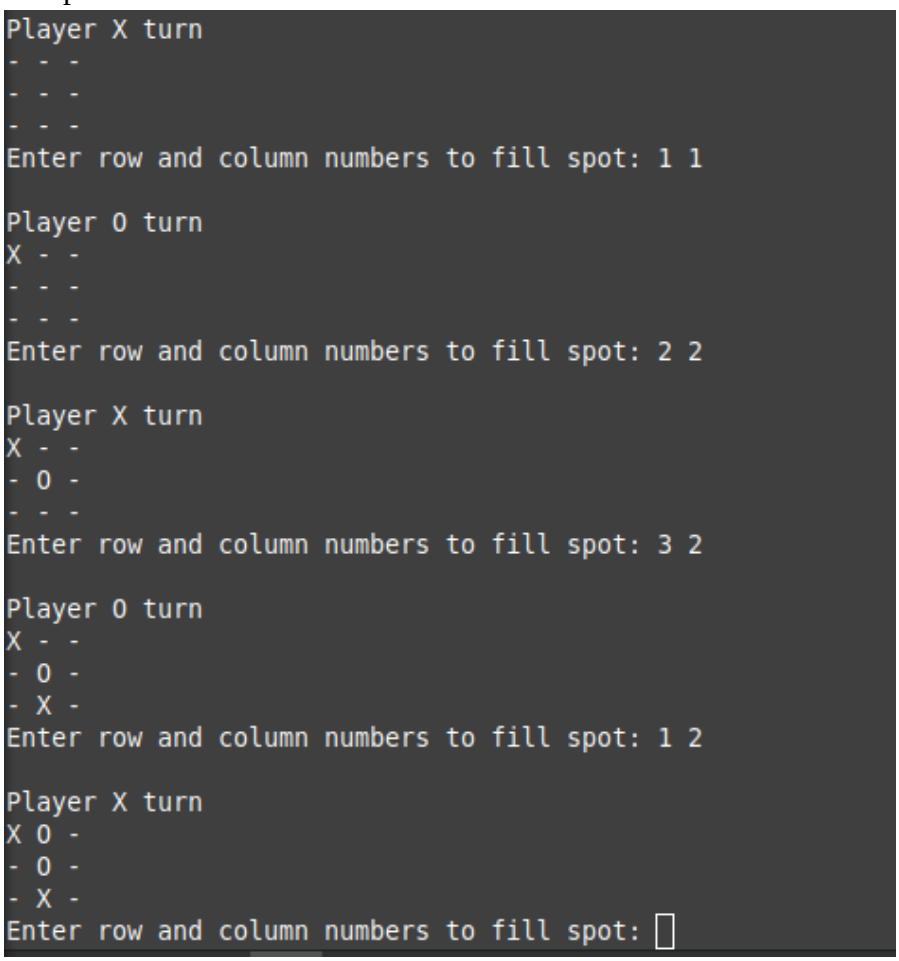
2. Tampilan Awal Game



Setelah user menjawab pertanyaan pertama tadi dengan “y”, maka tampilan seperti pada gambar akan keluar, cara bermain dari game ini adalah :

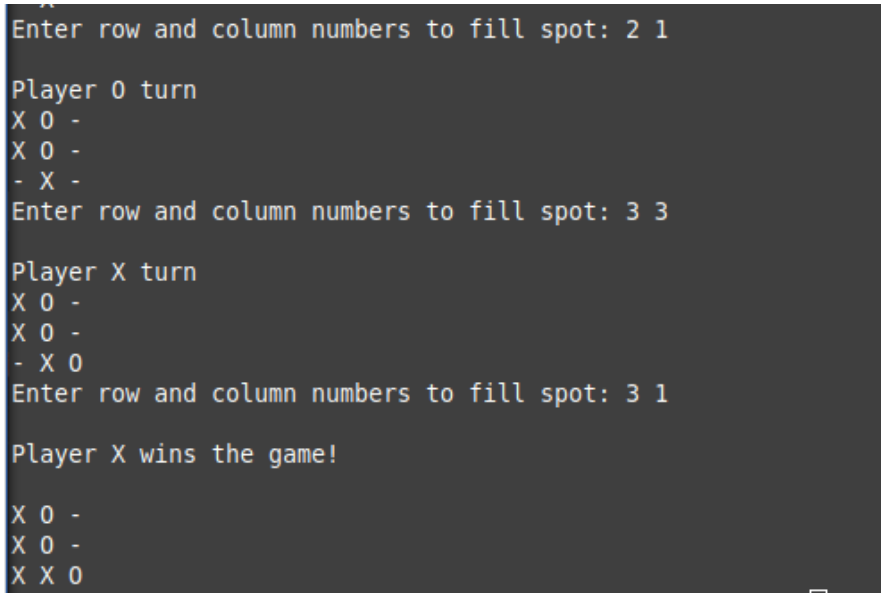
- Game ini dimainkan dengan 2 player, player X dan O
- Cara bermainnya player akan mengisikan koordinat dari kolom yang disediakan dengan pola “baris” (spasi) “kolom”. contoh: 1 1, berarti bahwa player mengisi baris 1 dan kolom 1
- Player akan otomatis berganti setelah kita melakukan input baris dan kolom

3. Tampilan Game



Tampilan game kurang lebih seperti pada gambar, disini player X dan O bergantian dalam melakukan giliran hingga Kotak Terisi oleh logo X atau O

4. Tampilan Akhir



Tampilan Akhir dari game akan muncul jika salah satu player menang atau seri, pada gambar game diselesaikan dengan player X yang menang

Script Program

Link github :
https://github.com/ilhamzuhri/21083010112/blob/master/Tugas%20Akhir/finpro_Tic%20Tac%20Toe_B.py

```
import random
import pyfiglet

class TicTacToe:

    def __init__(self):
        self.board = []

    def create_board(self):
        for i in range(3):
            row = []
            for j in range(3):
                row.append('-')
            self.board.append(row)

    def get_random_first_player(self):
        return random.randint(0, 1)

    def fix_spot(self, row, col, player):
        self.board[row][col] = player

    def is_player_win(self, player):
        win = None

        n = len(self.board)

        # checking rows
        for i in range(n):
            win = True
            for j in range(n):
                if self.board[i][j] != player:
                    win = False
                    break
            if win:
                return win

        # checking columns
        for i in range(n):
```

```
            win = True
            for j in range(n):
                if self.board[j][i] != player:
                    win = False
                    break
            if win:
                return win

        # checking diagonals
        win = True
        for i in range(n):
            if self.board[i][i] != player:
                win = False
                break
        if win:
            return win

        win = True
        for i in range(n):
            if self.board[i][n - 1 - i] != player:
                win = False
                break
        if win:
            return win
        return False

        for row in self.board:
            for item in row:
                if item == '-':
                    return False
        return True

    def is_board_filled(self):
        for row in self.board:
            for item in row:
                if item == '-':
                    return False
        return True

    def swap_player_turn(self, player):
        return 'X' if player == 'O' else 'O'

    def show_board(self):
        for row in self.board:
```

```
            for item in row:
                print(item, end=" ")
            print()

    def start(self):
        self.create_board()

        player = 'X' if self.get_random_first_player() == 1 else 'O'
        while True:
            print(f"Player {player} turn")

            self.show_board()

            # taking user input
            row, col = list(
                map(int, input("Enter row and column numbers to fill spot: ").split()))
            print()

            # fixing the spot
            self.fix_spot(row - 1, col - 1, player)

            # checking whether current player is won or not
            if self.is_player_win(player):
                print(f"Player {player} wins the game!")
                break

            # checking whether the game is draw or not
            if self.is_board_filled():
                print("Match Draw!")
                break


            # swapping the turn
            player = self.swap_player_turn(player)

        # showing the final view of board
        print()
        self.show_board()


# starting the game
tic_tac_toe = TicTacToe()
def welcome():
    ascii_banner = pyfiglet.figlet_format("TIC TAC TOE")
```

```
    print('-'*60)
    print(ascii_banner)
    print('-'*60)
    welcome = str(input("do you want to play? (y/n) : "))
    if welcome == "y":
        print("GOOD LUCK >///<")
        tic_tac_toe.start()
    elif welcome == "n":
        exit
    else:
        exit

welcome()
```