

LAPORAN UAS TEORI GAME DEV



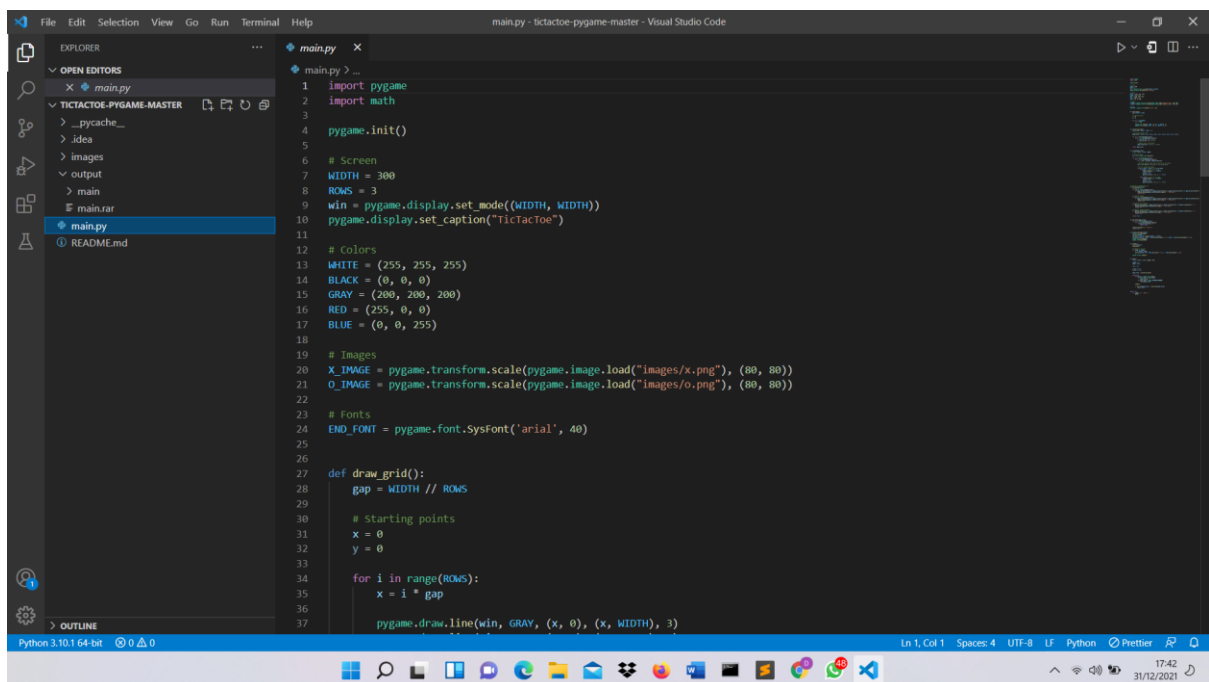
DISUSUN OLEH

Dandy Dicky

(V3920016)

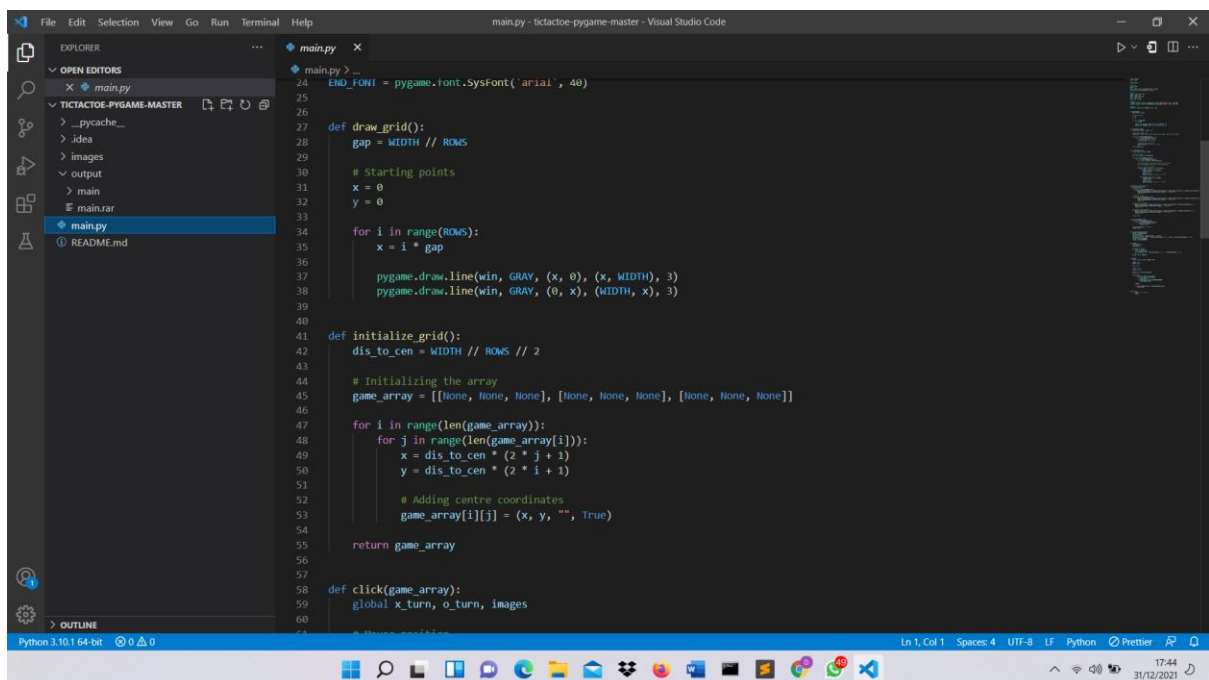
**PROGRAM STUDI D-III TEKNIK
INFORMATIKA FAKULTAS SEKOLAH
VOKASI
UNIVERSITAS SEBELAS MARET
SURAKARTA 2021/2022**

Langkah Langkah membuat game tic tac toe sederhana



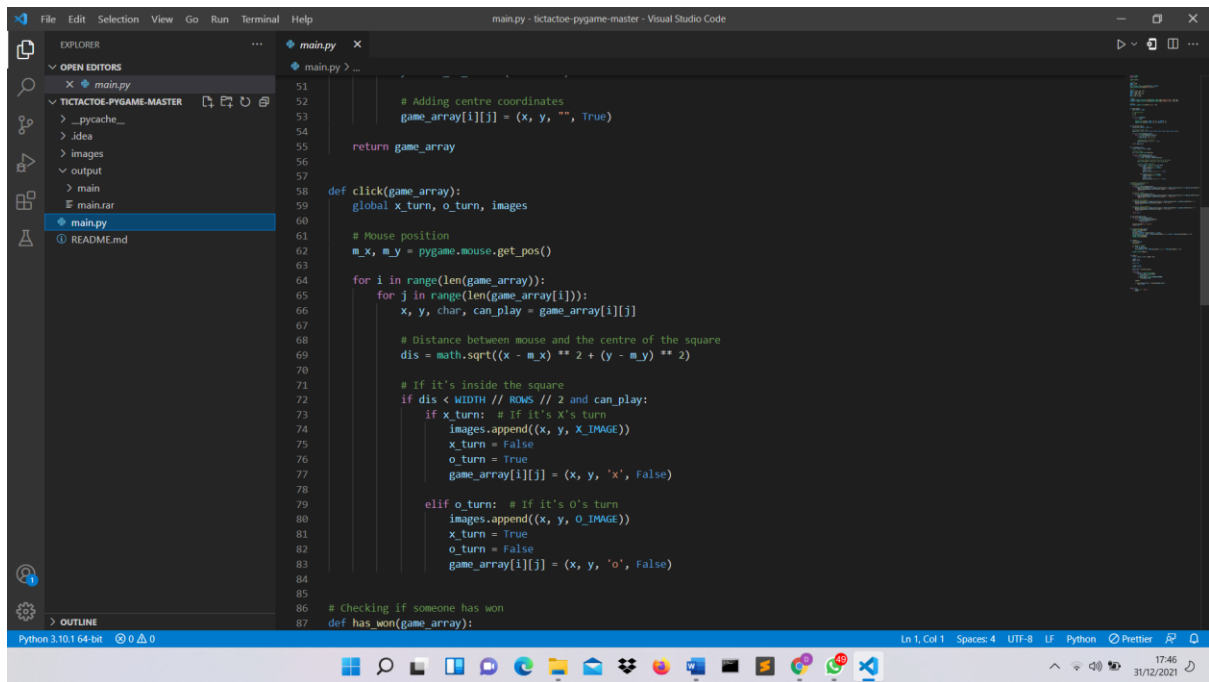
```
1 import pygame
2 import math
3
4 pygame.init()
5
6 # Screen
7 WIDTH = 300
8 ROWS = 3
9 win = pygame.display.set_mode((WIDTH, WIDTH))
10 pygame.display.set_caption("TicTacToe")
11
12 # Colors
13 WHITE = (255, 255, 255)
14 BLACK = (0, 0, 0)
15 GRAY = (200, 200, 200)
16 RED = (255, 0, 0)
17 BLUE = (0, 0, 255)
18
19 # Images
20 X_IMAGE = pygame.transform.scale(pygame.image.load("images/x.png"), (80, 80))
21 O_IMAGE = pygame.transform.scale(pygame.image.load("images/o.png"), (80, 80))
22
23 # Fonts
24 END_FONT = pygame.font.SysFont('arial', 40)
25
26
27 def draw_grid():
28     gap = WIDTH // ROWS
29
30     # Starting points
31     x = 0
32     y = 0
33
34     for i in range(ROWS):
35         x = i * gap
36
37         pygame.draw.line(win, GRAY, (x, 0), (x, WIDTH), 3)
```

Pertama lakukan inisiasi package yang diperlukan untuk membuild game yang akan dibuat. Mulai dari screen awal, warna, gambar yang akan digunakan, dan juga fonts.



```
24 END_FONT = pygame.font.SysFont('arial', 40)
25
26
27 def draw_grid():
28     gap = WIDTH // ROWS
29
30     # Starting points
31     x = 0
32     y = 0
33
34     for i in range(ROWS):
35         x = i * gap
36
37         pygame.draw.line(win, GRAY, (x, 0), (x, WIDTH), 3)
38         pygame.draw.line(win, GRAY, (0, x), (WIDTH, x), 3)
39
40
41 def initialize_grid():
42     dis_to_cen = WIDTH // ROWS // 2
43
44     # Initializing the array
45     game_array = [[None, None, None], [None, None, None], [None, None, None]]
46
47     for i in range(len(game_array)):
48         for j in range(len(game_array[i])):
49             x = dis_to_cen * (2 * j + 1)
50             y = dis_to_cen * (2 * i + 1)
51
52             # Adding centre coordinates
53             game_array[i][j] = (x, y, "", True)
54
55     return game_array
56
57
58 def click(game_array):
59     global x_turn, o_turn, images
```

Selanjutnya inisiasi kondisi awal game dan memastikan tampilan layar awal terdiri dari kotak 3x3 dengan kondisi awal kosong.



Selanjutnya untuk penggunaan mouse saat diklik. Dengan code seperti berikut :

Mouse position

```
m_x, m_y = pygame.mouse.get_pos()
```

```
for i in range(len(game_array)):
```

```
    for j in range(len(game_array[i])):
```

```
        x, y, char, can_play = game_array[i][j]
```

```
# Distance between mouse and the centre of the square
```

```
dis = math.sqrt((x - m_x) ** 2 + (y - m_y) ** 2)
```

```
# If it's inside the square
```

```
if dis < WIDTH // ROWS // 2 and can_play:
```

```
    if x_turn: # If it's X's turn
```

```
        images.append((x, y, X_IMAGE))
```

```
        x_turn = False
```

```
        o_turn = True
```

```
        game_array[i][j] = (x, y, 'x', False)
```

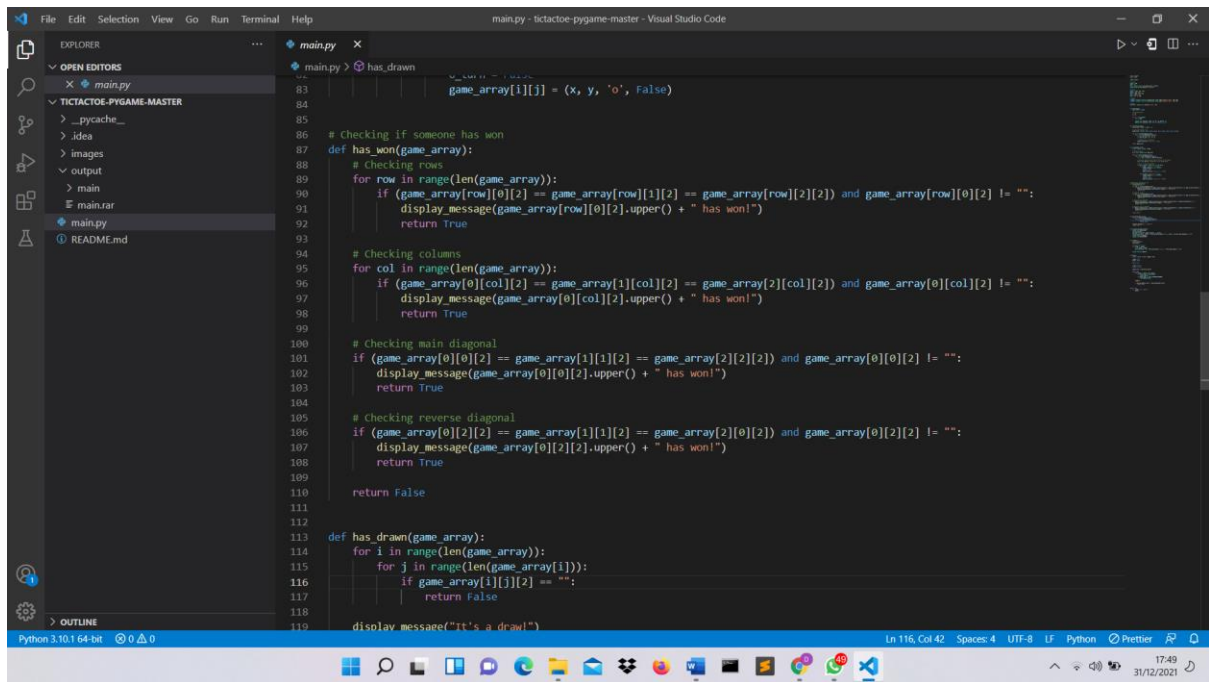
```
    elif o_turn: # If it's O's turn
```

```
        images.append((x, y, O_IMAGE))
```

```
        x_turn = True
```

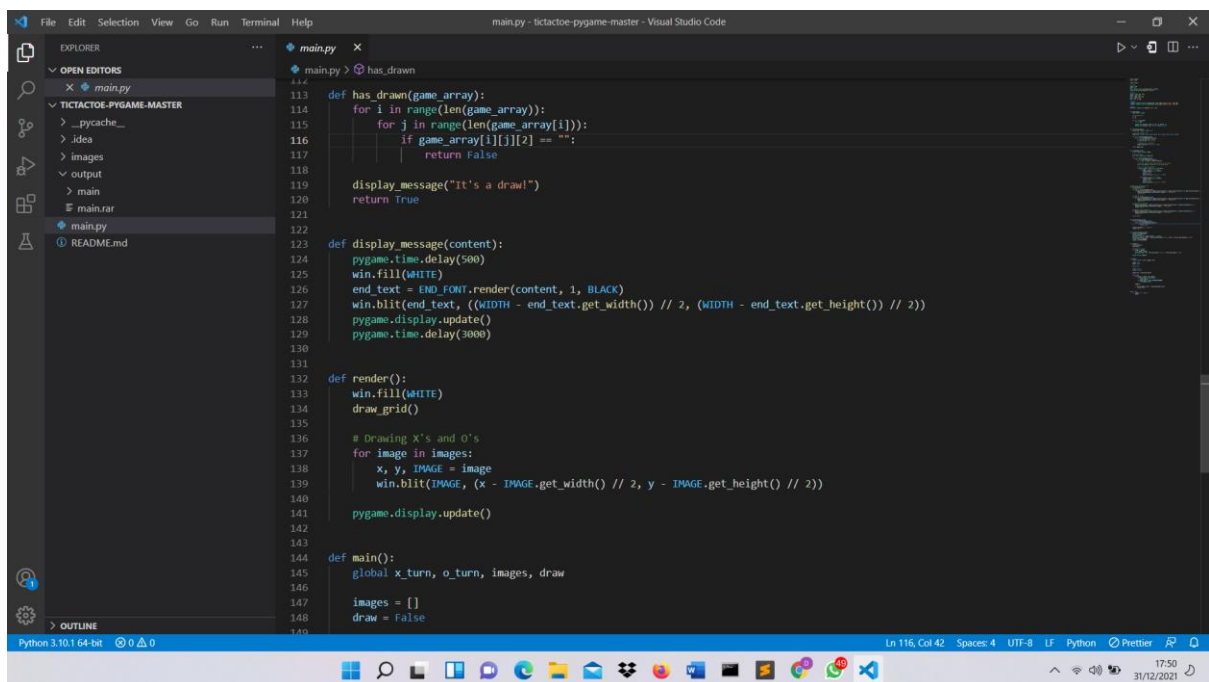
```
        o_turn = False
```

```
        game_array[i][j] = (x, y, 'o', False)
```



```
83 game_array[1][j] = (x, y, 'o', False)
84
85
86 # Checking if someone has won
87 def has_won(game_array):
88     # Checking rows
89     for row in range(len(game_array)):
90         if (game_array[row][0][2] == game_array[row][1][2] == game_array[row][2][2] and game_array[row][0][2] != ""):
91             display_message(game_array[row][0][2].upper() + " has won!")
92             return True
93
94     # Checking columns
95     for col in range(len(game_array)):
96         if (game_array[0][col][2] == game_array[1][col][2] == game_array[2][col][2] and game_array[0][col][2] != ""):
97             display_message(game_array[0][col][2].upper() + " has won!")
98             return True
99
100     # Checking main diagonal
101     if (game_array[0][0][2] == game_array[1][1][2] == game_array[2][2][2] and game_array[0][0][2] != ""):
102         display_message(game_array[0][0][2].upper() + " has won!")
103         return True
104
105     # Checking reverse diagonal
106     if (game_array[0][2][2] == game_array[1][1][2] == game_array[2][0][2] and game_array[0][2][2] != ""):
107         display_message(game_array[0][2][2].upper() + " has won!")
108         return True
109
110     return False
111
112
113 def has_drawn(game_array):
114     for i in range(len(game_array)):
115         for j in range(len(game_array[i])):
116             if game_array[i][j][2] == "":
117                 return False
118
119     display_message("It's a draw!")
```

Selanjutnya disini ada source code untuk menentukan atau check siapa pemenang dengan mengidentifikasi semua column dan rows.



```
113 def has_drawn(game_array):
114     for i in range(len(game_array)):
115         for j in range(len(game_array[i])):
116             if game_array[i][j][2] == "":
117                 return False
118
119     display_message("It's a draw!")
120     return True
121
122
123 def display_message(content):
124     pygame.time.delay(500)
125     win.fill(WHITE)
126     end_text = END_FONT.render(content, 1, BLACK)
127     win.blit(end_text, ((WIDTH - end_text.get_width()) // 2, (WIDTH - end_text.get_height()) // 2))
128     pygame.display.update()
129     pygame.time.delay(3000)
130
131
132 def render():
133     win.fill(WHITE)
134     draw_grid()
135
136     # Drawing X's and O's
137     for image in images:
138         x, y, IMAGE = image
139         win.blit(IMAGE, (x - IMAGE.get_width() // 2, y - IMAGE.get_height() // 2))
140
141     pygame.display.update()
142
143
144 def main():
145     global x_turn, o_turn, images, draw
146
147     images = []
148     draw = False
149
150
```

Namun apabila tidak ada yang memenuhi syarat atau kondisi untuk memenangkan game maka game dinyatakan draw.