LAPORAN UAS TEORI GAME DEV

DISUSUN OLEH

Dandy Dicky (V3920016)

PROGRAM STUDI D-III TEKNIK

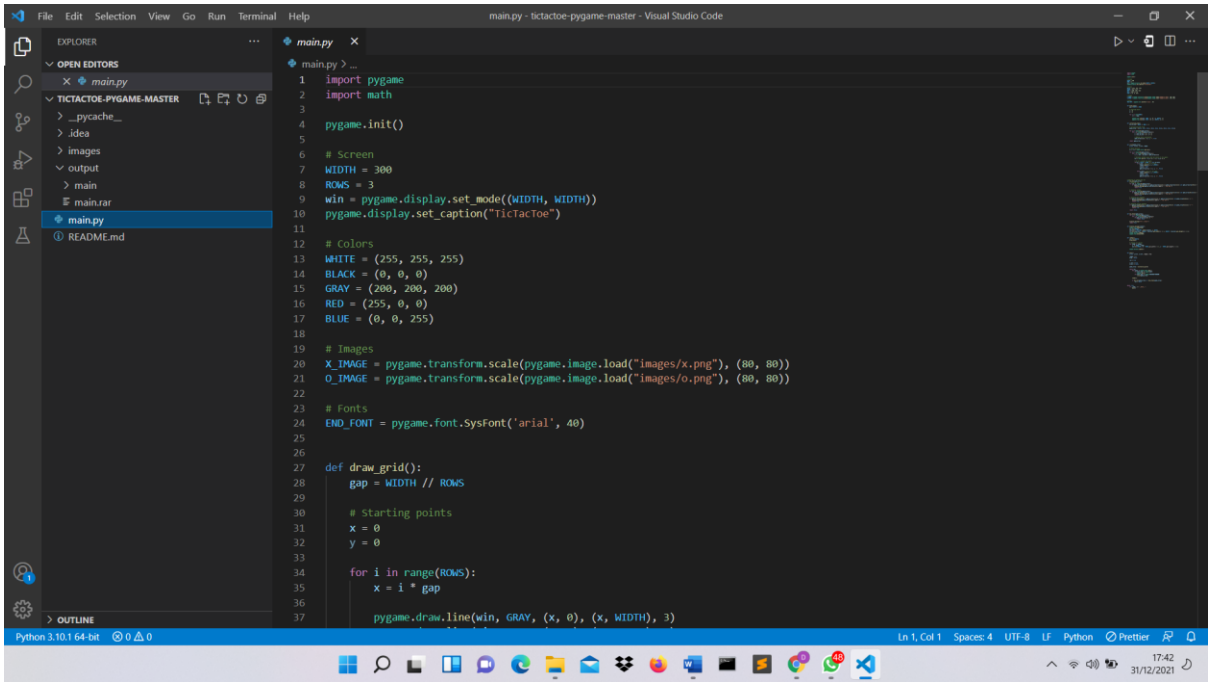
INFORMATIKA FAKULTAS SEKOLAH

VOKASI

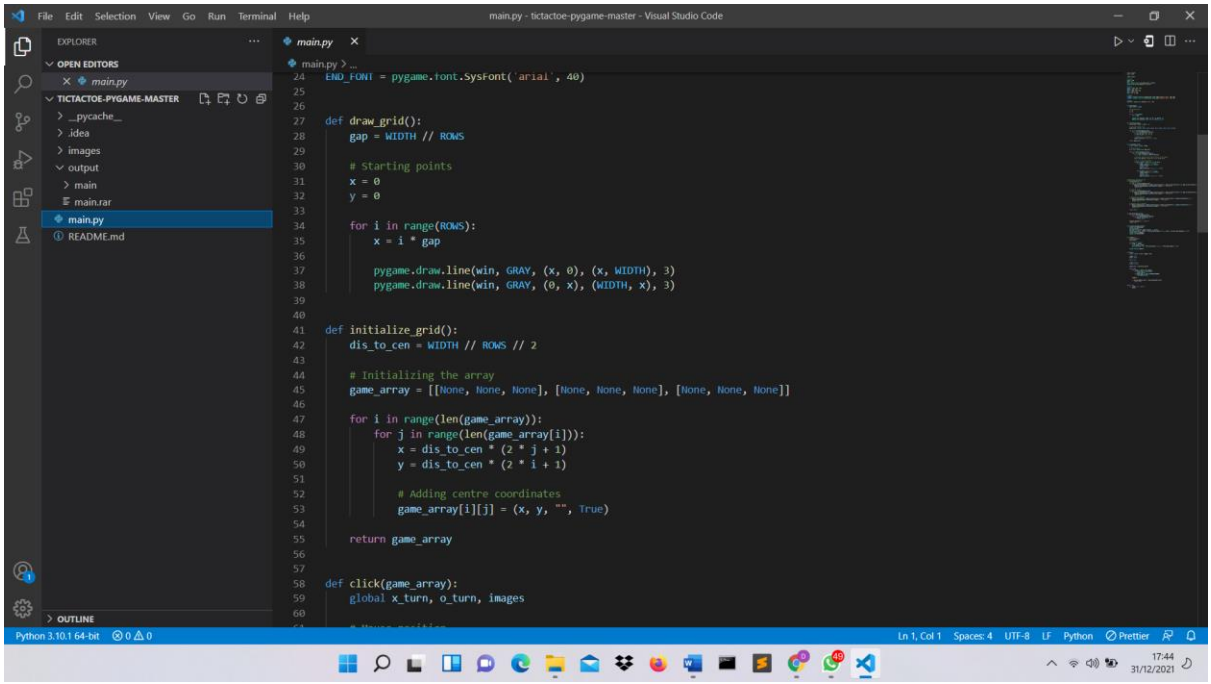
UNIVERSITAS SEBELAS MARET

SURAKARTA 2021/2022

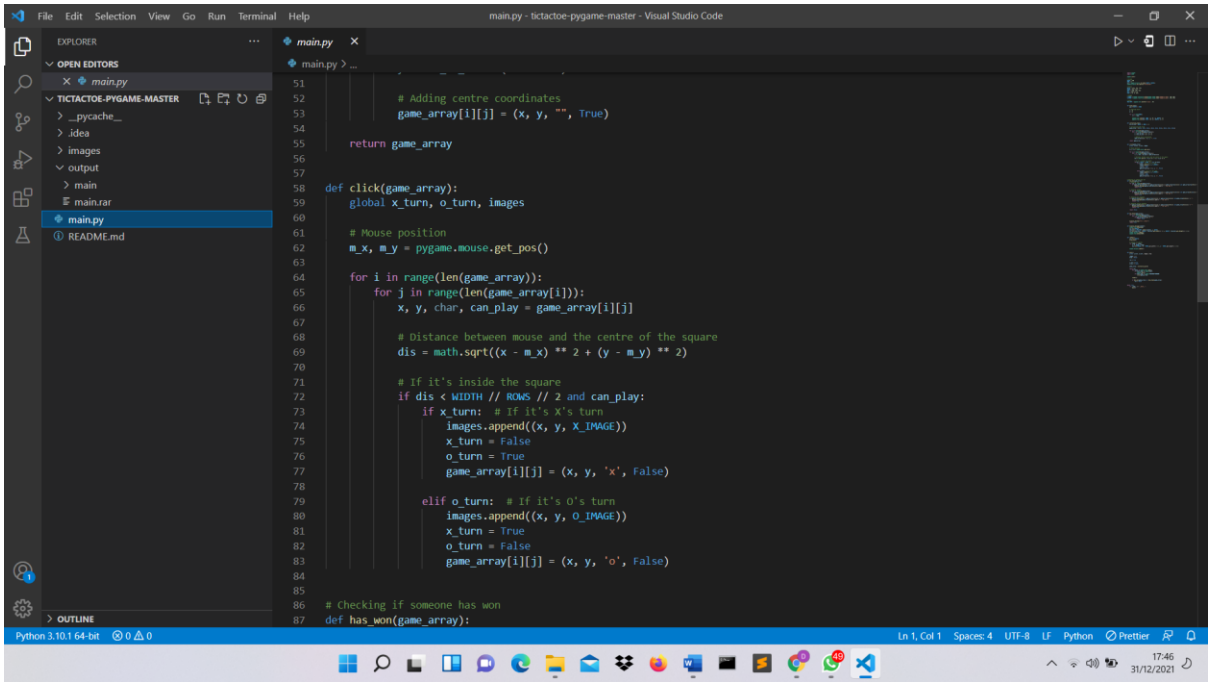
Langkah Langkah membuat game tic tac toe sederhana



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

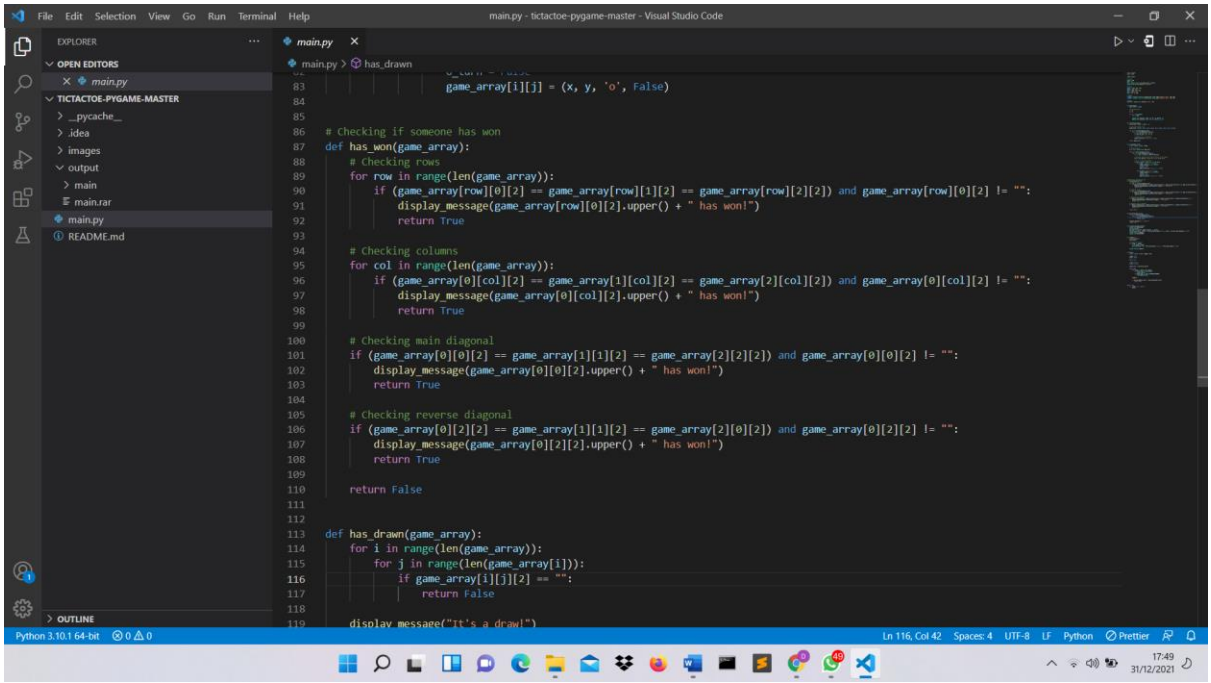


Selanjutnya inisiasi kondisi awal game dan memastikan tampilan layar awal terdiri dari kotak 3x3 dengan kondisi awal kosong.



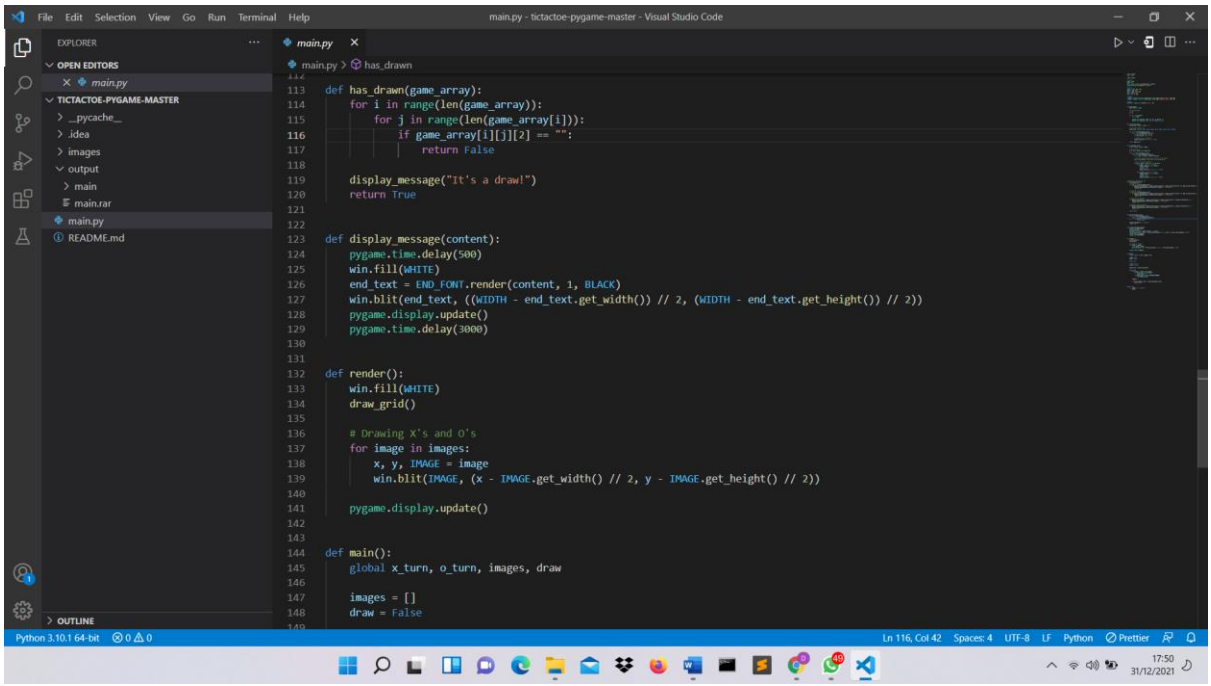
Selanjutnya untuk penggunaan mouse saat diklik. Dengan code seperti berikut :

```
# 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)
```



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


# Checking if someone has won
def has_won(game_array):
    # Checking rows
    for row in range(len(game_array)):
        if (game_array[row][0][2] == game_array[row][1][2] == game_array[row][2][2]) and game_array[row][0][2] != "":
            display_message(game_array[row][0][2].upper() + " has won!")
            return True

    # Checking columns
    for col in range(len(game_array)):
        if (game_array[0][col][2] == game_array[1][col][2] == game_array[2][col][2]) and game_array[0][col][2] != "":
            display_message(game_array[0][col][2].upper() + " has won!")
            return True

    # Checking main diagonal
    if (game_array[0][0][2] == game_array[1][1][2] == game_array[2][2][2]) and game_array[0][0][2] != "":
        display_message(game_array[0][0][2].upper() + " has won!")
        return True

    # Checking reverse diagonal
    if (game_array[0][2][2] == game_array[1][1][2] == game_array[2][0][2]) and game_array[0][2][2] != "":
        display_message(game_array[0][2][2].upper() + " has won!")
        return True

    return False


def has_drawn(game_array):
    for i in range(len(game_array)):
        for j in range(len(game_array[i])):
            if game_array[i][j][2] == "":
                return False

    display_message("It's a draw!")
```

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



```
def has_drawn(game_array):
    for i in range(len(game_array)):
        for j in range(len(game_array[i])):
            if game_array[i][j][2] == "":
                return False

    display_message("It's a draw!")
    return True


def display_message(content):
    pygame.time.delay(500)
    win.fill(WHITE)
    end_text = END_FONT.render(content, 1, BLACK)
    win.blit(end_text, ((WIDTH - end_text.get_width()) // 2, (WIDTH - end_text.get_height()) // 2))
    pygame.display.update()
    pygame.time.delay(3000)


def render():
    win.fill(WHITE)
    draw_grid()

    # Drawing X's and O's
    for image in images:
        x, y, IMAGE = image
        win.blit(IMAGE, (x - IMAGE.get_width() // 2, y - IMAGE.get_height() // 2))

    pygame.display.update()


def main():
    global x_turn, o_turn, images, draw

    images = []
    draw = False
```

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