Soal Ujian

Praktikum Pemrograman Jaringan

Format Nama File : Ujian_Pjar_Nama.pdf

Nama : Ibrahim Bramullah

Npm : 50420562

Kelas : 4IA02

A. Web APP (20 Poin)

1) Buatlah Langkah-langkah WEB APP Menggunakan Framework FLASK output terminal dan penjelasannya
2) Ubahlah route pada URL menjadi http://xxx.x.x.x:5000/about
3) Buatlah output di URL http://xxx.x.x.x:5000/about menjadi hai, adalah halaman tentang saya (nama praktikkan)

B. Chat APP (20 Poin)

1) Buatlah Langkah-langkah Membuat Aplikasi Chat Sederhana MENGGUNAKAN PYTHON berserta dengan screenshoot output code program, terminal output dan penjelasannya
2) buatlah percakapan antara server dengan client tentang diri pribadi

C. P2P App (20 Poin)

1) Buatlah Langkah-langkah Membuat Aplikasi File Sharing P2P MENGGUNAKAN PYTHON berserta dengan screenshoot output code program, terminal output dan penjelasannya
2) Kirimkan File Dengan Format JPG dari client. Tentunya harus ada file JPG pada folder local

D. Video Player (20 Poin)

1) Buatlah Video Player yang bisa di putar, jeda, perbesar layar pada server local
2) Tambahkan video player menjadi 3 dengan sumber video yang berbeda dan bebas video apapun.
3) Ganti favicon menjadi logo Instagram

E. Game Sederhana (20 Poin)

1) Buatlah game sederhana apa saja menggunakan server local
2) Tampilkan output skor & waktu

A. WebApp

pip install flask

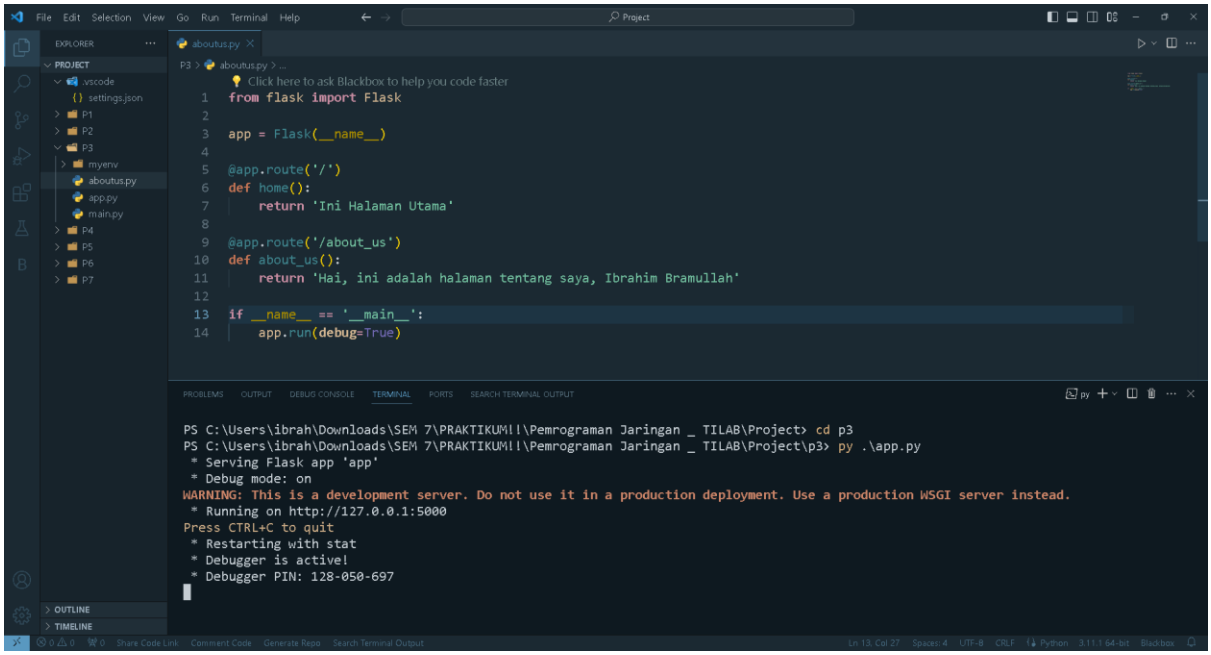
```
PS C:\Users\ibrah\Downloads\SEM 7\PRAKTIKUM!!\Pemrograman Jaringan _ TILAB\Project> cd p3
PS C:\Users\ibrah\Downloads\SEM 7\PRAKTIKUM!!\Pemrograman Jaringan _ TILAB\Project\p3> pip install flask
Requirement already satisfied: flask in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packages (2.3.2)
Requirement already satisfied: Werkzeug>=2.3.3 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packages (from flask
) (2.3.4)
Requirement already satisfied: Jinja2>=3.1.2 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packages (from flask)
(3.1.2)
Requirement already satisfied: itsdangerous>=2.1.2 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packages (from f
lask) (2.1.2)
Requirement already satisfied: click>=8.1.3 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packages (from flask) (
8.1.3)
Requirement already satisfied: blinker>=1.6.2 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packages (from flask)
 (1.6.2)
Requirement already satisfied: colorama in c:\users\ibrah\appdata\roaming\python\python311\site-packages (from click>=8.1.3->flask) (0
.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packages (from Jinja
2>=3.1.2->flask) (2.1.2)
PS C:\Users\ibrah\Downloads\SEM 7\PRAKTIKUM!!\Pemrograman Jaringan _ TILAB\Project\p3>
```

pip install virtualenv

```
PS C:\Users\ibrah\Downloads\SEM 7\PRAKTIKUM!!\Pemrograman Jaringan _ TILAB\Project\p3> pip install virtualenv
Requirement already satisfied: virtualenv in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packages (20.23.0)
Requirement already satisfied: distlib<1,>=0.3.6 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packages (from vir
tualenv) (0.3.6)
Requirement already satisfied: filelock<4,>=3.11 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packages (from vir
tualenv) (3.12.0)
Requirement already satisfied: platformdirs<4,>=3.2 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packages (from
virtualenv) (3.5.1)
PS C:\Users\ibrah\Downloads\SEM 7\PRAKTIKUM!!\Pemrograman Jaringan _ TILAB\Project\p3>
```

Create my virtualenv

```
PS C:\Users\ibrah\Downloads\SEM 7\PRAKTIKUM!!\Pemrograman Jaringan _ TILAB\Project\p3> virtualenv myenv
created virtual environment CPython3.11.1.final.0-64 in 8408ms
  creator CPython3Windows(dest=C:\Users\ibrah\Downloads\SEM 7\PRAKTIKUM!!\Pemrograman Jaringan _ TILAB\Project\P3\myenv, clear=False,
no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=C:\Users\ibrah\AppData\Local\
pypa\virtualenv)
    added seed packages: distlib==0.3.7, filelock==3.13.0, pip==23.3.1, platformdirs==3.11.0, setuptools==68.2.2, virtualenv==20.24.6,
 wheel==0.41.3
  activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
PS C:\Users\ibrah\Downloads\SEM 7\PRAKTIKUM!!\Pemrograman Jaringan _ TILAB\Project\p3>
```



App.py

```
from flask import Flask
```

```python
app = Flask(__name__)

@app.route('/')

def home():
    return 'Hello, World!'

@app.route('/about_us')
def tentang():
    return 'Hai, ini adalah halaman tentang Saya (Ibrahim Bramullah)'

if __name__ == '__main__':
    app.run(debug=True)
```

Hai, ini adalah halaman tentang Saya (Ibrahim Bramullah)

B. Chat App

server.py

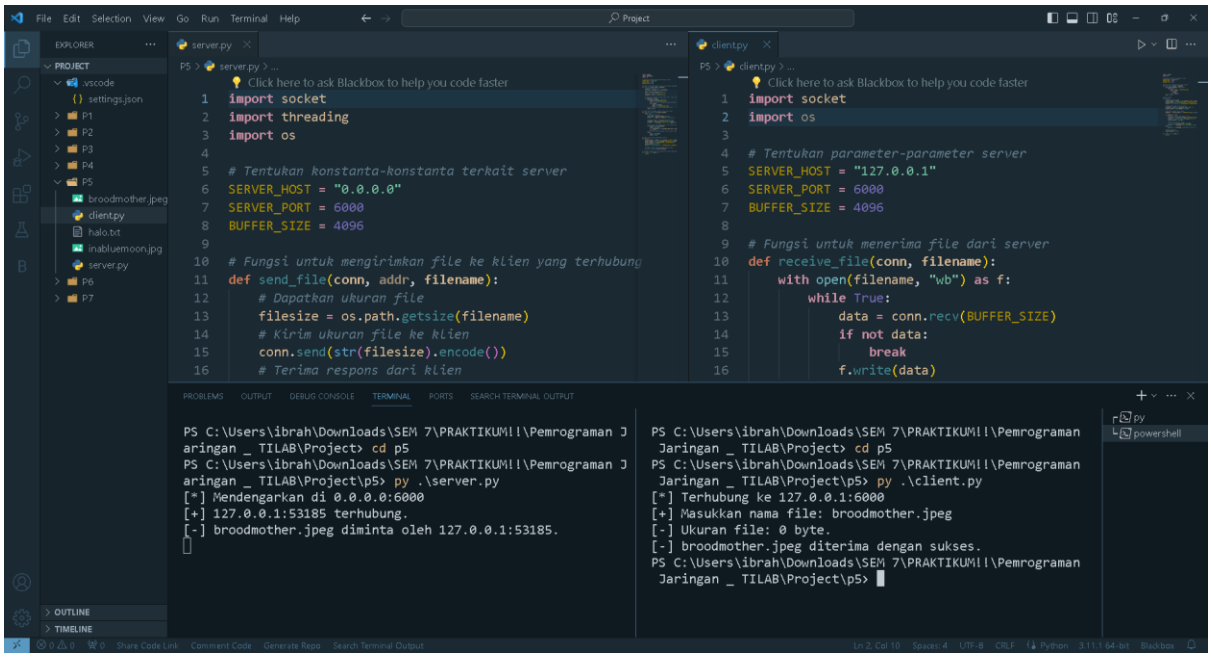
```python
import socket  # Import library socket untuk mengakses fungsi jaringan

HOST = '127.0.0.1'  # Menentukan alamat IP server (localhost)
PORT = 5555  # Menentukan nomor port yang akan digunakan untuk koneksi

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)  # Membuat objek socket dengan alamat IPv4 dan tipe koneksi TCP

sock.bind((HOST, PORT))  # Mengikat socket server ke alamat dan port yang telah ditentukan
sock.listen()  # Mendengarkan koneksi yang masuk
```

```
client, address = sock.accept()  # Menerima koneksi dari klien dan mendapatkan informasi klien

while True:  # Loop tak terbatas untuk menerima dan mengirim pesan
    message = client.recv(1024).decode('utf-8')  # Menerima pesan dari klien, mendekripsi dalam format UTF-8, dan menyimpan dalam variabel 'message'
    print(f"Received from {address}: {message}")  # Mencetak pesan yang diterima beserta alamat klien

    reply = input("Reply: ")  # Membaca pesan balasan dari server
    client.send(reply.encode('utf-8'))  # Mengirim pesan balasan ke klien setelah mengenkripsi dalam format UTF-8
```

client.py

```
import socket  # Import library socket untuk mengakses fungsi jaringan

HOST = '127.0.0.1'  # Menentukan alamat IP tujuan yang akan dihubungi (localhost)
PORT = 5555  # Menentukan nomor port yang akan digunakan untuk koneksi

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)  # Membuat objek socket dengan alamat IPv4 dan tipe koneksi TCP

sock.connect((HOST, PORT))  # Menghubungkan socket dengan alamat dan port yang telah ditentukan

while True:  # Loop tak terbatas untuk mengirim dan menerima pesan
    message = input("Message: ")  # Membaca pesan dari pengguna
    input()  # Mengabaikan input kosong (sepertinya ini adalah kesalahan)
    sock.send(message.encode('utf-8'))  # Mengirim pesan ke server setelah mengenkripsi dalam format UTF-8

    reply = sock.recv(1024).decode('utf-8')  # Menerima pesan dari server, mendekripsi dari UTF-8, dan menyimpan dalam variabel 'reply'
    print(f"Received: {reply}")  # Mencetak pesan yang diterima dari server
```

C. P2P App



server.py

```python
import socket
import threading
import os

# Tentukan konstanta-konstanta terkait server
SERVER_HOST = "0.0.0.0"
SERVER_PORT = 6000
BUFFER_SIZE = 4096

# Fungsi untuk mengirimkan file ke klien yang terhubung
def send_file(conn, addr, filename):
    # Dapatkan ukuran file
    filesize = os.path.getsize(filename)
    # Kirim ukuran file ke klien
    conn.send(str(filesize).encode())
    # Terima respons dari klien
    response = conn.recv(BUFFER_SIZE)

    # Jika klien mengakui (mengirimkan "OK"), mulai mengirimkan file
    if response == b"OK":
        with open(filename, "rb") as f:
            data = f.read(BUFFER_SIZE)
            # Kirim data file dalam potongan-potongan hingga seluruh file terkirim
            while data:
                conn.send(data)
                data = f.read(BUFFER_SIZE)
        # Tutup koneksi setelah mengirimkan file
        conn.close()

# Fungsi untuk menunggu koneksi masuk dari klien
def wait_for_connection(server_socket):
```

```
    while True:
        # Terima koneksi baru
        conn, addr = server_socket.accept()
        print(f"[+] {addr[0]}:{addr[1]} terhubung.")

        # Terima nama file yang diminta oleh klien
        filename = conn.recv(BUFFER_SIZE).decode()
        print(f"[-] {filename} diminta oleh {addr[0]}:{addr[1]}.")

        # Jika file yang diminta ada, kirim sinyal "OK" ke klien dan mulai kirim
file
        if os.path.exists(filename):
            conn.send(b"OK")
            t = threading.Thread(target=send_file, args=(conn, addr, filename))
            t.start()
        # Jika file tidak ditemukan, kirim sinyal "ERR" ke klien dan tutup koneksi
        else:
            conn.send(b"ERR")
            conn.close()

# Fungsi utama untuk menjalankan server
def main():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    server_socket.bind((SERVER_HOST, SERVER_PORT))
    server_socket.listen(5)
    print(f"[*] Mendengarkan di {SERVER_HOST}:{SERVER_PORT}")

    # Panggil fungsi untuk menunggu koneksi masuk
    wait_for_connection(server_socket)

# Jalankan program jika ini adalah file utama yang dieksekusi
if __name__ == "__main__":
    main()
```

client.py

```
import socket
import os

# Tentukan parameter-parameter server
SERVER_HOST = "127.0.0.1"
SERVER_PORT = 6000
BUFFER_SIZE = 4096

# Fungsi untuk menerima file dari server
def receive_file(conn, filename):
    with open(filename, "wb") as f:
        while True:
            data = conn.recv(BUFFER_SIZE)
            if not data:
```

```
                break
            f.write(data)
    conn.close()

# Fungsi utama
def main():
    # Membuat socket klien
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect((SERVER_HOST, SERVER_PORT))
    print(f"[*] Terhubung ke {SERVER_HOST}:{SERVER_PORT}")

    # Mengirimkan nama file ke server
    filename = input("[+] Masukkan nama file: ")
    client_socket.send(filename.encode())

    # Menerima respons dari server
    response = client_socket.recv(BUFFER_SIZE).decode()

    if response == "OK":
        # Menerima ukuran file dari server
        filesize = int(client_socket.recv(BUFFER_SIZE).decode())
        print(f"[-] Ukuran file: {filesize} byte.")
        client_socket.send(b"OK")

        # Menerima file dari server
        receive_file(client_socket, filename)
        print(f"[-] {filename} diterima dengan sukses.")
    else:
        print(f"[!] {filename} tidak ada di server.")

    # Menutup koneksi
    client_socket.close()

if __name__ == "__main__":
    main()
```

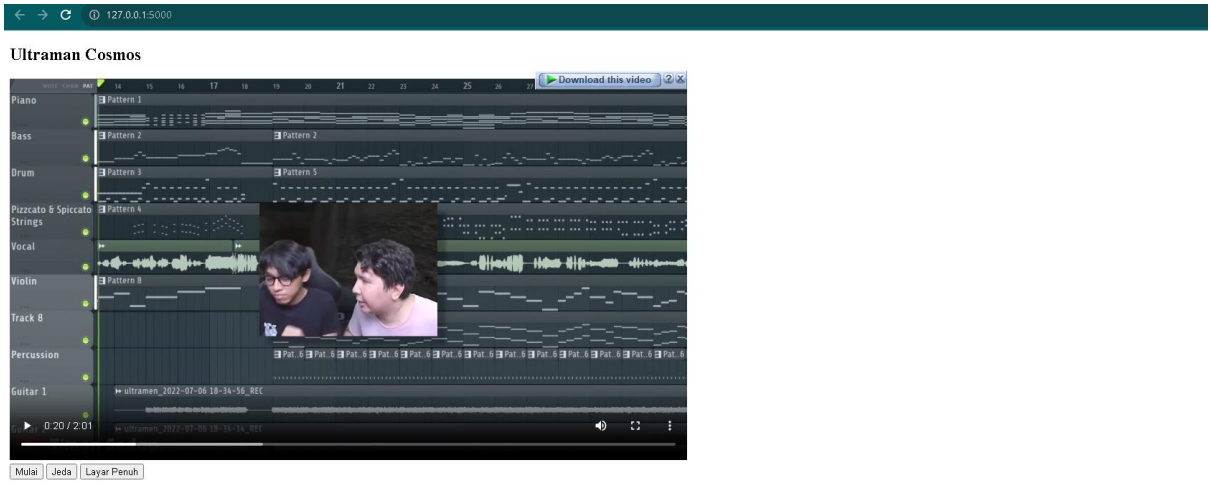
D. Video Player

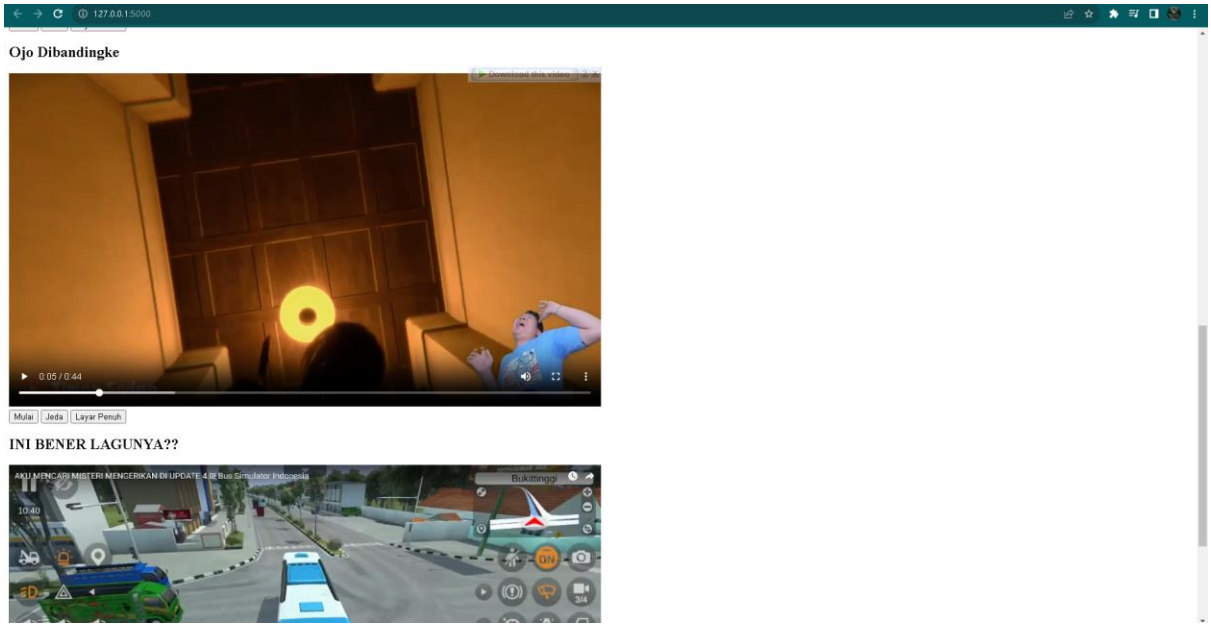
pip install flask

```
PS C:\Users\ibrah\Downloads\SEM 7\PRAKTIKUM!!\Pemrograman Jaringan _ TILAB\Project\p6> pip install flask
Requirement already satisfied: flask in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packages (2.3.2)
Requirement already satisfied: Werkzeug>=2.3.3 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packag
es (from flask) (2.3.4)
Requirement already satisfied: Jinja2>=3.1.2 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packages
 (from flask) (3.1.2)
Requirement already satisfied: itsdangerous>=2.1.2 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-pa
ckages (from flask) (2.1.2)
Requirement already satisfied: click>=8.1.3 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packages
(from flask) (8.1.3)
Requirement already satisfied: blinker>=1.6.2 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-package
s (from flask) (1.6.2)
Requirement already satisfied: colorama in c:\users\ibrah\appdata\roaming\python\python311\site-packages (from click>=8.
1.3->flask) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packag
es (from Jinja2>=3.1.2->flask) (2.1.2)
PS C:\Users\ibrah\Downloads\SEM 7\PRAKTIKUM!!\Pemrograman Jaringan _ TILAB\Project\p6>
```

pip install flask-bootstrap

```
PS C:\Users\ibrah\Downloads\SEM 7\PRAKTIKUM!!\Pemrograman Jaringan _ TILAB\Project\p6> pip install flask-bootstrap
Requirement already satisfied: flask-bootstrap in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packag
es (3.3.7.1)
Requirement already satisfied: Flask>=0.8 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packages (f
rom flask-bootstrap) (2.3.2)
Requirement already satisfied: dominate in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packages (fro
m flask-bootstrap) (2.9.0)
Requirement already satisfied: visitor in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packages (from
 flask-bootstrap) (0.1.3)
Requirement already satisfied: Werkzeug>=2.3.3 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packag
es (from Flask>=0.8->flask-bootstrap) (2.3.4)
Requirement already satisfied: Jinja2>=3.1.2 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packages
 (from Flask>=0.8->flask-bootstrap) (3.1.2)
Requirement already satisfied: itsdangerous>=2.1.2 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-pa
ckages (from Flask>=0.8->flask-bootstrap) (2.1.2)
Requirement already satisfied: click>=8.1.3 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packages
(from Flask>=0.8->flask-bootstrap) (8.1.3)
Requirement already satisfied: blinker>=1.6.2 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-package
s (from Flask>=0.8->flask-bootstrap) (1.6.2)
Requirement already satisfied: colorama in c:\users\ibrah\appdata\roaming\python\python311\site-packages (from click>=8.
1.3->Flask>=0.8->flask-bootstrap) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\ibrah\appdata\local\programs\python\python311\lib\site-packag
es (from Jinja2>=3.1.2->Flask>=0.8->flask-bootstrap) (2.1.2)
PS C:\Users\ibrah\Downloads\SEM 7\PRAKTIKUM!!\Pemrograman Jaringan _ TILAB\Project\p6>
```





App.py

```python
from flask import Flask, render_template, request
from flask_bootstrap import Bootstrap

app = Flask(__name__)
Bootstrap(app)

@app.route("/")
def index():
    return render_template("index.html")

if __name__ == '__main__':
```

```
    app.run(debug=True)
```

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Video Streaming</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='css/bootstrap.min.css')
}}">
    <!-- Memanggil Bootstrap dari CDN -->
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
    integrity="sha384-OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QLQk1iFV12LyOWqyj8p8WZO+3q7a"
crossorigin="anonymous">
    <link rel="icon" type="image/ico" href="{{ url_for('static',
filename='video/logo_instagram_ico.ico') }}">
    <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
    <style>
        body, html {
            height: 100%;
        }
        #videoPlayer {
            width: 50%;
            height: 50%;
        }
        #videoPlayer2 {
            width: 50%;
            height: 50%;
        }
        #videoPlayer3 {
            width: 50%;
            height: 50%;
        }
        #videoPlayer4 {
            width: 50%;
            height: 50%;
        }
    </style>
</head>
<body>
    <div class="container mt-5">
        <div class="row">
            <div class="col-md-8 offset-md-2">
                <h2 class="text-center mb-4">Ultraman Cosmos</h2>
                <video id="videoPlayer" class="" controls autoplay>
                    <source src="{{ url_for('static', filename='video/Ultraman Cosmos.mp4')
}}" type="video/mp4">
                </video>
                <div class="mt-3">
                    <button id="playBtn" class="btn btn-primary mr-3">Mulai</button>
                    <button id="pauseBtn" class="btn btn-primary mr-3">Jeda</button>
                    <button id="fullScreenBtn" class="btn btn-primary">Layar Penuh</button>
                </div>
```

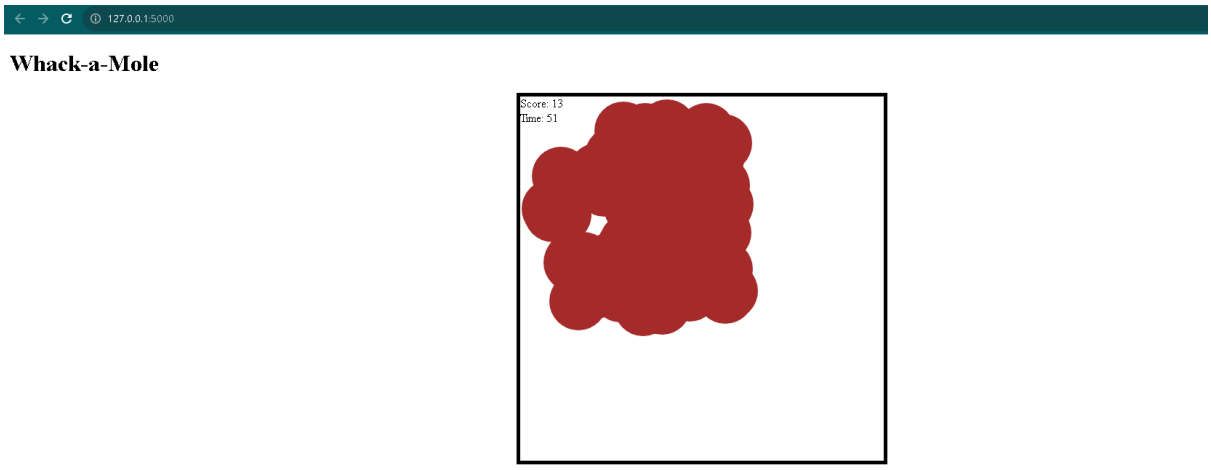
```
                <h2 class="text-center mb-4">Challange - Bang Aku Telat Nga?</h2>
                <video id="videoPlayer2" class="" controls autoplay>
                    <source src="{{ url_for('static', filename='video/BANG AKU TELAT
NGA.mp4') }}" type="video/mp4">
                </video>
                <div class="mt-3">
                    <button id="playBtn2" class="btn btn-primary mr-3">Mulai</button>
                    <button id="pauseBtn2" class="btn btn-primary mr-3">Jeda</button>
                    <button id="fullScreenBtn2" class="btn btn-primary">Layar
Penuh</button>
                </div>

                <h2 class="text-center mb-4">Ojo Dibandingke</h2>
                <video id="videoPlayer3" class="" controls autoplay>
                    <source src="{{ url_for('static', filename='video/Ojo Dibandingke.mp4')
}}" type="video/mp4">
                </video>
                <div class="mt-3">
                    <button id="playBtn3" class="btn btn-primary mr-3">Mulai</button>
                    <button id="pauseBtn3" class="btn btn-primary mr-3">Jeda</button>
                    <button id="fullScreenBtn3" class="btn btn-primary">Layar
Penuh</button>
                </div>

                <h2 class="text-center mb-4">INI BENER LAGUNYA??</h2>
                <video id="videoPlayer4" class="" controls autoplay>
                    <source src="{{ url_for('static', filename='video/INI GUYS
LAGUNYA.mp4') }}" type="video/mp4">
                </video>
                <div class="mt-3">
                    <button id="playBtn4" class="btn btn-primary mr-3">Mulai</button>
                    <button id="pauseBtn4" class="btn btn-primary mr-3">Jeda</button>
                    <button id="fullScreenBtn4" class="btn btn-primary">Layar
Penuh</button>
                </div>
        </div>
    </div>
    <script src="{{ url_for('static', filename='js/jquery.min.js') }}"></script>
    <script src="{{ url_for('static', filename='js/video.js') }}"></script>
    <script src="{{ url_for('static', filename='js/video2.js') }}"></script>
    <script src="{{ url_for('static', filename='js/video3.js') }}"></script>
</body>
</html>
```

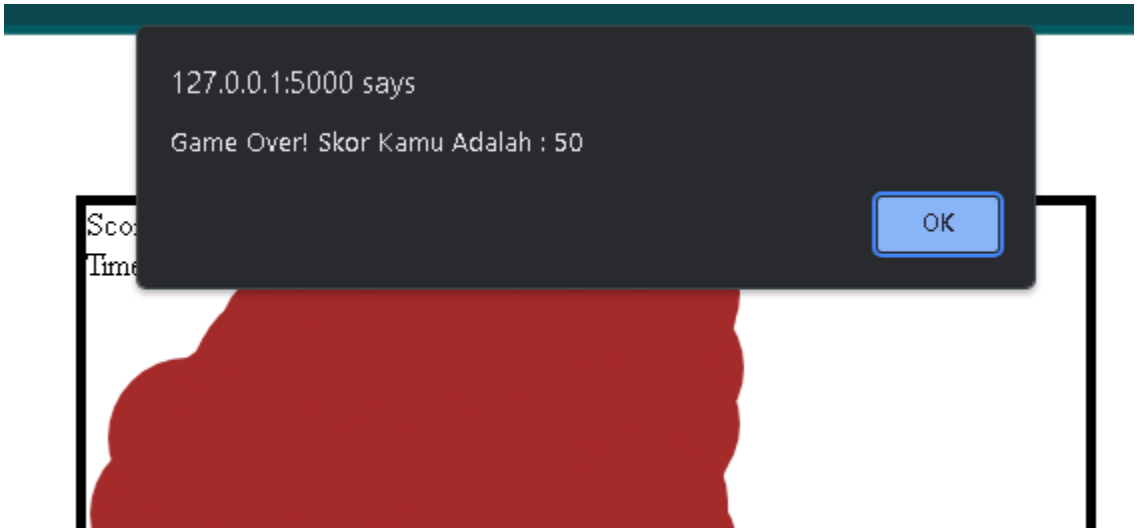
Video.js (1 code dari 4 video js)

```
$(document).ready(function () {
    // Mendapatkan element video dan tombol play, pause, dan fullscreen
    var video = document.getElementById("videoPlayer");
    var playBtn = document.getElementById("playBtn");
    var pauseBtn = document.getElementById("pauseBtn");
    var fullScreenBtn = document.getElementById("fullScreenBtn");

   // Menambahkan event listener pada tombol play
   playBtn.addEventListener("click", function () {
        video.play();
    });
```

```
    // Menambahkan event listener pada tombol pause
    pauseBtn.addEventListener("click", function () {
        video.pause();
    });

    // Menambahkan event listener pada tombol fullscreen
    fullScreenBtn.addEventListener("click", function () {
        if (video.requestFullscreen) {
            video.requestFullscreen();
        } else if (video.mozRequestFullScreen) {
            video.mozRequestFullScreen();
        } else if (video.webkitRequestFullscreen) {
            video.webkitRequestFullscreen();
        }
    });
});
```

E. Game Sederhana





app.py

```python
from flask import Flask, render_template

app = Flask(__name__)
app.debug = True

@app.route('/')
def index():
    return render_template('index.html')

if __name__ == '__main__':
    app.run()
```

index.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Whack-a-Mole Game</title>
    <link rel="stylesheet" type="text/css" href="{{ url_for('static', filename='style.css')
}}">
    <link rel="icon" type="image/ico" href="{{ url_for('static', filename='Logo-
Gunadarma.ico') }}">
  </head>
  <body>
    <h1>Whack-a-Mole</h1>
    <div id="game-board">
      <!-- Area permainan -->
    </div>
     <script src="{{ url_for('static', filename='js/jquerry.js') }}"></script>
    <script src="{{ url_for('static', filename='script.js') }}"></script>
  </body>
</html>
```

script.js

```javascript
const gameBoard = document.getElementById("game-board");
const scoreDisplay = document.createElement("div");
const timerDisplay = document.createElement("div");
const restartButton = document.createElement("button");
let score = 0;
let moleTimer;
let timeRemaining = 10;

function startGame() {
  score = 0;
  scoreDisplay.textContent = "Score: 0";
  gameBoard.appendChild(scoreDisplay);
  timeRemaining = 60;
  gameBoard.appendChild(timerDisplay);
  moleTimer = setInterval(createMole, 100);
  startTimer();
}
```

```
function createMole() {
  const mole = document.createElement("div");
  mole.classList.add("mole");
  mole.style.left = Math.random() * 250 + "px";
  mole.style.top = Math.random() * 250 + "px";
  mole.addEventListener("click", whackMole);
  gameBoard.appendChild(mole);
}

function whackMole() {
  score++;
  this.parentNode.removeChild(this);
  scoreDisplay.textContent = "Score: " + score;
}

function startTimer() {
  const timerInterval = setInterval(() => {
    timeRemaining--;
    timerDisplay.textContent = "Time: " + timeRemaining;

    if (timeRemaining <= 0) {
      clearInterval(timerInterval);
      endGame();
    }
  }, 1000);
}

function endGame() {
  clearInterval(moleTimer);
  alert("Game Over! Skor Kamu Adalah : " + score);
  gameBoard.removeChild(scoreDisplay);
  gameBoard.removeChild(timerDisplay);
  gameBoard.innerHTML = "";

  createRestartButton();
}

function createRestartButton() {
  restartButton.textContent = "Restart";
  restartButton.addEventListener("click", restartGame);
  gameBoard.appendChild(restartButton);
}

function restartGame() {
  gameBoard.removeChild(restartButton);
  startGame();
}

startGame();
```

style.css

```
#game-board {
    width: 500px;
    height: 500px;
```

```
  border: 5px solid black;
  margin: 0 auto;
  position: relative;
}

.mole {
  width: 80px;
  height: 80px;
  background-color: brown;
  border-radius: 50%;
  position: absolute;
  cursor: pointer;
}
```