

TUGAS BESAR JARINGAN KOMPUTER
MEMBUAT WEB SERVER BERBASIS TCP
DENGAN MENERAPKAN SOCKET PROGRAMMING



IT – 46 - 01

Kelompok 8

Andhika Akbar	1303220128
Yobel Alvino Kastanja	1303220142
I Gusti Ngurah Agung Dwigata Kusuma	1303220075

PROGRAM STUDI S-1 TEKNOLOGI INFORMASI
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
BANDUNG
2024

Kode Program Server Multi

```
import socket
import threading

IP = socket.gethostbyname(socket.gethostname())
PORT = 12000
ADDR = (IP, PORT)
SIZE = 1024
FORMAT = "utf-8"

def handle_client(conn, addr):
    print(f"[NEW CONNECTION] {addr} connected.")

    try:
        request = conn.recv(SIZE).decode(FORMAT)
        headers = request.split('\n')
        filename = headers[0].split()[1]

        if filename == '/':
            filename = '/index.html'

        try:
            filepath = '.' + filename
            if filename.endswith('.html'):
                content_type = 'text/html'
                with open(filepath, 'r') as f:
                    content = f.read()
                response = f'HTTP/1.1 200 OK\nContent-Type: {content_type}\n\n' + content
                conn.sendall(response.encode(FORMAT))
            elif filename.endswith(('.jpg', '.jpeg')):
                content_type = 'image/jpeg'
                with open(filepath, 'rb') as f:
                    content = f.read()
                response = f'HTTP/1.1 200 OK\nContent-Type: {content_type}\n\n'.encode(FORMAT) + content
                conn.sendall(response)
            elif filename.endswith('.png'):
                content_type = 'image/png'
                with open(filepath, 'rb') as f:
                    content = f.read()
                response = f'HTTP/1.1 200 OK\nContent-Type: {content_type}\n\n'.encode(FORMAT) + content
                conn.sendall(response)
            else:
                response = 'HTTP/1.1 415 Unsupported Media Type\n\nUnsupported Media Type'
                conn.sendall(response.encode(FORMAT))
        except FileNotFoundError:
            response = 'HTTP/1.1 404 NOT FOUND\n\nFile Not Found'
            conn.sendall(response.encode(FORMAT))
    except Exception as e:
        print(f"Error handling client {addr}: {e}")
    finally:
        conn.close()
    print(f"[DISCONNECTED] {addr} disconnected.")

def main():
    print("[STARTING] Server is starting...")
    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server.bind(ADDR)
    server.listen()
    print(f"[LISTENING] Server is listening on {IP}:{PORT}")

    while True:
        conn, addr = server.accept()
        thread = threading.Thread(target=handle_client, args=(conn, addr))
        thread.start()
        print(f"[ACTIVE CONNECTIONS] {threading.active_count() - 1}")

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

Server menerima permintaan dari klien, memprosesnya, dan kemudian mengirimkan respons. Jika file yang diminta tidak ditemukan, akan ditampilkan pesan 404 Not Found. Server ini mampu menangani beberapa koneksi klien secara bersamaan dengan menggunakan konsep threading.

Kode Program Server Single

```
import socket
import threading

def handle_client(client_socket):
    request = client_socket.recv(1024).decode()
    headers = request.split('\n')
    filename = headers[0].split()[1]

    if filename == '/':
        filename = '/index.html'

    try:
        # Tentukan path file
        filepath = '.' + filename
        # Tentukan apakah file adalah biner atau teks
        if filename.endswith(('.png', '.jpg', '.jpeg', '.gif')):
            with open(filepath, 'rb') as fin:
                content = fin.read()
                response = b'HTTP/1.1 200 OK\nContent-Type: image/png\n\n' + content
        else:
            with open(filepath, 'r', encoding='utf-8') as fin:
                content = fin.read()
                response = 'HTTP/1.1 200 OK\n\n' + content
                response = response.encode()
    except FileNotFoundError:
        response = 'HTTP/1.1 404 Not Found\n\nFile Not Found'.encode()

    client_socket.sendall(response)
    client_socket.close()

def main():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.bind(('0.0.0.0', 12000))
    server_socket.listen(5)
    print('Server listening on port 12000')

    while True:
        client_socket, addr = server_socket.accept()
        print(f'Accepted connection from {addr}')
        client_handler = threading.Thread(target=handle_client, args=(client_socket,))
        client_handler.start()

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

Server ini menggunakan protokol HTTP untuk menerima permintaan dari klien, mengambil file yang diminta (jika tersedia), dan mengirimkannya kembali kepada klien sebagai respons. Kode ini memanfaatkan modul socket dan threading untuk mengelola koneksi dari banyak klien secara bersamaan.

Kode Program Client

```
import socket
import sys

def http_request(host, port, path):
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect((host, port))
    request = f"GET {path} HTTP/1.1\r\nHost: {host}\r\n\r\n"
    client_socket.send(request.encode())
    response = client_socket.recv(4096)
    print(response.decode())
    client_socket.close()

if __name__ == "__main__":
    if len(sys.argv) != 4:
        print("Usage: python TubesClient.py <host> <port> <path>")
    else:
        host = sys.argv[1]
        port = int(sys.argv[2])
        path = sys.argv[3]
        http_request(host, port, path)
```

Client HTTP yang mengirimkan permintaan GET ke server dan menampilkan respons yang diterima. Selain itu, client juga menentukan alamat server, port, dan nama file yang diminta.

Kode Program HTML

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <title>SOSOK SCP MISTERIUS</title>
7 </head>
8 <body>
9     <h1>SOSOK SCP 69</h1>
10    <p>Ini adalah sosok SCP 69:</p>
11    
12 </body>
13 </html>
```

Berikut adalah kode program dari HTML untuk menampilkan resource yang diminta

Running program melalui terminal

```
C:\Users\ASUS\OneDrive - Telkom University\Documents\TUBES JARKOM>TubesClient.py 192.168.56.1 12000 /TubesSingle.py
HTTP/1.1 200 OK

import socket
import threading
import os

def handle_client(client_socket):
    request = client_socket.recv(1024).decode()
    headers = request.split('\n')
    filename = headers[0].split()[1]

    if filename == '/':
        filename = '/index.html'

    try:
        # Tentukan path file
        filepath = '.' + filename
        # Tentukan apakah file adalah biner atau teks
        if filename.endswith(('png', '.jpg', '.jpeg', '.gif')):
            fin = open(filepath, 'rb')
```

```
C:\Windows\System32\cmd.e x + v - □ x
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.

C:\Users\andhika dhio\Desktop\Jarkom>TubesClient.py 1
72.26.48.1 12000 /TubesHTML.html
HTTP/1.1 200 OK
Content-Type: text/html

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width
, initial-scale=1.0">
  <title>SOSOK SCP MISTERIUS</title>
</head>
<body>
  <h1>SOSOK SCP 69</h1>
  <p>Ini adalah sosok SCP 69:</p>
  
</body>
</html>

C:\Users\andhika dhio\Desktop\Jarkom>
```

```
C:\Windows\System32\cmd.e x + v - □ x
Microsoft Windows [Version 10.0.22631.3593]
(c) Microsoft Corporation. All rights reserved.

C:\Users\andhika dhio\Desktop\Jarkom>TubesServer.py
[STARTING] Server is starting...
[LISTENING] Server is listening on 172.26.48.1:12000
[NEW CONNECTION] ('172.26.48.1', 53584) connected.
[ACTIVE CONNECTIONS] 1
[DISCONNECTED] ('172.26.48.1', 53584) disconnected.
[NEW CONNECTION] ('172.26.48.1', 53585) connected.
[ACTIVE CONNECTIONS] 1
[DISCONNECTED] ('172.26.48.1', 53585) disconnected.
[NEW CONNECTION] ('172.26.48.1', 53586) connected.
[ACTIVE CONNECTIONS] 1
[DISCONNECTED] ('172.26.48.1', 53586) disconnected.
[NEW CONNECTION] ('172.26.48.1', 53603) connected.
[ACTIVE CONNECTIONS] 1
[DISCONNECTED] ('172.26.48.1', 53603) disconnected.
```

Client mengirim permintaan ke server di 172.26.48.1(Ip host) pada port 12000 untuk file index.html. Respons dari server adalah HTTP/1.1 200 OK, menunjukkan bahwa permintaan berhasil.

```
C:\Users\ASUS\OneDrive - Telkom University\Documents\TUBES JARKOM>python TubesClient.py
Usage: python TubesClient.py <host> <port> <path>
```

Hasil akan seperti ini jika client hanya menginputkan “python TubesClient.py”, server akan memberikan balasan untuk cara request html.

Running program melalui browser

SOSOK SCP 69

Ini adalah sosok SCP 69:



Diakses secara lokal melalui [http:// 172.26.48.1:12000/TubesHTML.html](http://172.26.48.1:12000/TubesHTML.html)