

**LAPORAN PRAKTIKUM MEMBANGUN WEB SERVER
HTTP SEDERHANA
MENGGUNAKAN SOCKET DI PYTHON**



Nama : L Hafidl Alkhair
Nim : 2023903430060
Kelas : TRKJ-2C
Jurusan : Teknologi Informasi dan Komputer
Program Studi : Teknologi Rekayasa Komputer dan Jaringan
Dosen Pengampu : Afla Nevrissa S.Kom, M.Kom



**PROGRAM STUDI TEKNOLOGI REKAYASA KOMPUTER JARINGAN
JURUSAN TEKNOLOGI INFORMASI DAN KOMPUTER
POLITEKNIK NEGERI LHOKSEUMAWE 2025**

LEMBAR PENGESAHAN

No. Praktikum : 04 /TIK/TRKJ-2C/ Pemrograman Jaringan
Judul : Laporan Praktikum
Nama : L Hafidl Alkhair
Nim : 202390343060
Kelas : TRKJ-2C
Jurusan : Teknologi Informasi Dann Komputer
Program Studi : Teknologi Rekayasa Komputer Jaringan
Tanggal Praktikum : 20 Mei 2025
Tanggal Penyerahan : 1 Juni 2025

Buketrata, 1 Juni 2025

Dosen Pembimbing,

Afla Nevrisa S.Kom, M.Kom

NIP. 199211172022032007

DAFTAR ISI

LEMBAR PENGESAHAN	i
DAFTAR ISI	ii
BAB I.....	1
A. Dasar teori.....	1
B. Tujuan Praktikum	1
C. Alat dan bahan	2
BAB II	3
1. Membuat Program Server.....	3
2. Membuat Tampilan.....	5
3. Membuat Style untuk tampilan nya.....	5
4. Membuat file notfound	5
5. Output program.....	7
6. Penjelasan dan Analisa	8
BAB III	9
A. Kesimpulan.....	9

BAB I

PENDAHULUAN

A. Dasar teori

a. Socket Programming

Socket adalah antarmuka komunikasi antara dua perangkat dalam jaringan.

Dalam praktik ini digunakan socket TCP yang digunakan untuk menerima permintaan (request) dari browser dan mengirimkan respon berupa halaman HTML atau file lain.

b. Protokol HTTP (Hypertext Transfer Protocol)

HTTP adalah protokol yang digunakan untuk komunikasi antara client (browser) dan server web. Dalam praktik ini, server secara manual membangun struktur response HTTP termasuk kode status, header, dan isi konten.

c. MIME Type

MIME type adalah format standar untuk menunjukkan jenis file. Server dalam praktik ini menggunakan modul mimetypes Python untuk menentukan jenis file yang diminta oleh client agar browser dapat menanganinya dengan tepat (seperti HTML, CSS, atau gambar).

d. File Handling dan Error Handling

Server memuat file berdasarkan URL yang diminta. Jika file tidak ditemukan, server mengembalikan halaman 404.html sebagai penanganan error yang sesuai dengan standar HTTP (kode status 404).

e. Struktur Web Sederhana (HTML & CSS)

Halaman yang disajikan oleh server menggunakan HTML sebagai struktur utama dan CSS untuk memperindah tampilan. Ini menunjukkan integrasi dasar antara backend dan frontend dalam sistem web sederhana.

B. Tujuan Praktikum

- Mempelajari cara kerja server HTTP secara manual menggunakan socket di Python.
- Membangun dan menyajikan halaman web statis (HTML, CSS, gambar)

dari server socket.

- Menerapkan konsep request-response dalam komunikasi berbasis HTTP.
- Mengimplementasikan error handling seperti halaman 404.
- Memahami dasar pengiriman konten web dari sisi server tanpa framework web.

C. Alat dan bahan

- Laptop/PC – Digunakan untuk menulis dan menjalankan kode.
- Python 3.x – Bahasa pemrograman utama yang digunakan.
- Text Editor (VS Code, Sublime, Notepad++) – Untuk membuat file Python dan HTML/CSS.
- Command Prompt / Terminal – Untuk menjalankan server.
- Browser (Chrome, Firefox, dll) – Sebagai client untuk mengakses server.
- Jaringan lokal / localhost – Untuk mengakses server melalui <http://localhost:8080>.

BAB II **PRAKTIKUM**

1. Membuat Program Server

Membuat file server_http.py

```
#!/usr/bin/env python3

import socket
import os
import mimetypes

# Fungsi untuk membaca file dan menentukan MIME type
def load_file(path):
    try:
        with open(path, 'rb') as f:
            content = f.read()
            content_type = mimetypes.guess_type(path)[0] or 'application/octet-stream'
            header = f"HTTP/1.1 200 OK\r\nContent-Type: {content_type}\r\n\r\n"
            return header.encode() + content
    except FileNotFoundError:
        with open("404.html", "rb") as f:
            content = f.read()
            header = "HTTP/1.1 404 Not Found\r\nContent-Type: text/html\r\n\r\n"
            return header.encode() + content

# Buat socket TCP
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind(('0.0.0.0', 8080))
server_socket.listen(5)

print("🌐 Server HTTP berjalan di http://localhost:8080 ...")
```

```

while True:
    client_socket, addr = server_socket.accept()
    print(f"[+] Terhubung dari {addr}")
    request = client_socket.recv(1024).decode('utf-8')
    print(f"[REQ]\n{request}")

    if not request:
        client_socket.close()
        continue

    # Parsing request
    try:
        filename = request.split()[1]
        if filename == '/':
            filename = '/index.html'
        filepath = filename.lstrip('/')
        response = load_file(filepath)

    except Exception as e:
        print(f"[ERR] {e}")
        response = b"HTTP/1.1 500 Internal Server Error\r\n\r\n<h1>500 Internal
Server Error</h1>"

    client_socket.send(response)
    client_socket.close()

```

2. Membuat Tampilan

Buat file index.html

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
    <title>Halaman Utama</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <h1>Halo dari Server HTTP Sederhana!</h1>
    <p>Ini adalah halaman utama.</p>
    
</body>
</html>
```

3. Membuat Style untuk tampilan nya

```
body {
    background-color: #f0f0f0;
    font-family: Arial, sans-serif;
    color: #333;
    text-align: center;
    margin-top: 50px;
}
```

4. Membuat file notfound

```
<!DOCTYPE html>
<html lang="id">
<head>
    <meta charset="UTF-8">
```

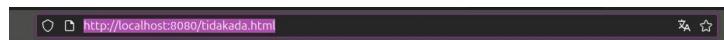
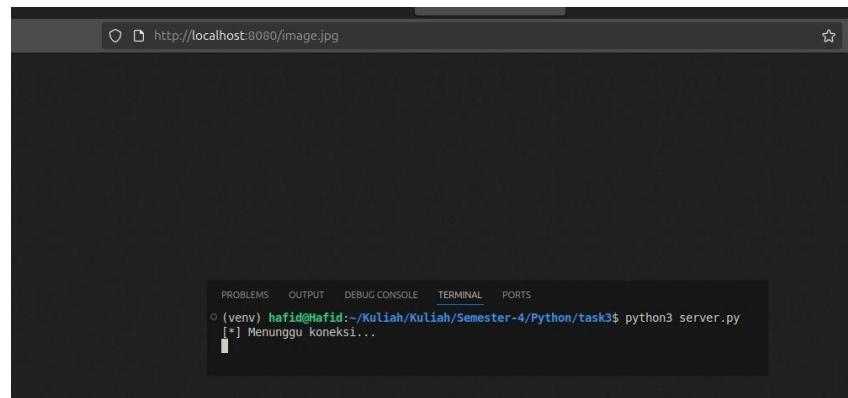
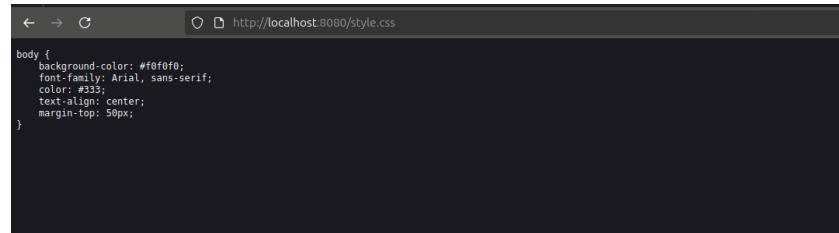
```
<title>404 Tidak Ditemukan</title>
</head>
<body style="text-align:center; padding-top:50px;">
    <h1>😔 404</h1>
    <p>Halaman yang Anda cari tidak ditemukan.</p>
</body>
</html>
```

5. Output program

a. Server dijalankan

```
hafid@Hafid:~/Kuliah/Kuliah/Semester-4/Python/task4$ python3 server_http.py
Server HTTP berjalan di http://localhost:8080 ...
```

b. Tampilan halaman



6. Penjelasan dan Analisa

Program Server (server_http.py)

Server dibuat menggunakan socket TCP dan berjalan pada localhost port 8080.

Ketika browser mengakses URL, server menerima permintaan (request), memproses path yang diminta, dan mencari file yang sesuai.

- Jika file ditemukan, server mengirim file tersebut disertai header HTTP 200.
- Jika file tidak ditemukan, maka server mengirim 404.html dengan header HTTP 404.
- Semua file dikirim dalam bentuk binary dan header dikirim sebagai string HTTP.

Program HTML dan CSS (index.html dan style.css)

index.html adalah halaman utama yang memuat teks dan gambar serta mengimpor style.css untuk tampilan. Halaman ini dapat ditampilkan ketika user mengakses root (/) dari server.

Halaman Not Found (404.html)

File ini akan muncul ketika pengguna mengakses file yang tidak ada di server, sehingga server merespons dengan halaman kesalahan standar.

Analisa Output

- Saat server dijalankan, akan muncul pesan "  Server HTTP berjalan di http://localhost:8080".
- Browser dapat menampilkan halaman HTML secara utuh, termasuk gaya dan gambar.
- Jika mencoba mengakses file yang tidak tersedia, akan muncul halaman 404 dengan pesan "Halaman tidak ditemukan".
- Ini membuktikan server mampu menangani permintaan file statis dan error handling sesuai protokol HTTP.

BAB III

PENUTUP

A. Kesimpulan

Melalui praktikum ini, dapat disimpulkan bahwa pembuatan web server sederhana menggunakan socket di Python memungkinkan pemahaman mendalam terhadap cara kerja dasar komunikasi HTTP antara client (browser) dan server. Server berhasil melayani permintaan file statis seperti HTML, CSS, dan gambar, serta mampu menangani kesalahan seperti file yang tidak ditemukan (404).

Dengan membangun server dari awal tanpa menggunakan framework, mahasiswa dapat memahami secara langsung proses parsing request HTTP, pembentukan response, dan pengiriman konten ke browser. Praktikum ini juga memperlihatkan pentingnya MIME type dan manajemen file dalam pengembangan aplikasi jaringan berbasis web.