

Laporan Tugas Besar Jaringan Komputer
Mata Kuliah Jaringan Komputer
Program Web Server Sederhana Berbasis TCP Socket Programming



Disusun Oleh :

- 1. Abiyyu Daffa Haidar Setiawan(1301213519)**
- 2. Ariotegar Ramadhanto Putra(1301213441)**
- 3. Rivan Fauzan (1301210554)**

Kelas : IF – 45 – 01

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
BANDUNG
2023

DAFTAR ISI

DAFTAR ISI.....	2
BAB 1 PENDAHULUAN	3
1.1 Tujuan Penulisan Laporan	3
1.2 Latar Belakang.....	3
1.3 Jaringan Komputer	3
1.4 Python.....	3
BAB 2 PEMBAHASAN.....	5
2.1 Topik Tugas Besar	5
2.2 Penjelasan.....	5
2.3 Output File Terbaca.....	6
2.4 Code Tcp Server.....	7
2.5 Code Tcp Client.....	8
BAB 3 KESIMPULAN.....	9
DAFTAR PUSTAKA.....	10

BAB 1

PENDAHULUAN

1.1 Tujuan Penulisan Laporan

Tujuan dalam pembuatan program web server sederhana ini adalah untuk memahami konsep dasar jaringan komputer, khususnya protokol HTTP dan TCP socket programming. Tujuan dari tugas ini adalah untuk membuat web server yang mampu menerima HTTP request, memproses request tersebut, dan mengirimkan HTTP response yang sesuai.

1.2 Latar Belakang

Latar Belakang pembuatan laporan ini adalah untuk memenuhi laporan penugasan besar pada mata kuliah Jaringan Komputer kami dengan spesifikasi

1.3 Jaringan Komputer

Jaringan telekomunikasi yang memungkinkan antar komputer untuk saling berkomunikasi dengan bertukar data.^[1] Tujuan dari jaringan komputer adalah agar dapat mencapai tujuannya, setiap bagian dari jaringan komputer dapat meminta dan memberikan layanan (service). Pihak yang meminta/menerima layanan disebut klien (client) dan yang memberikan/mengirim layanan disebut peladen (server). Desain ini disebut dengan sistem client-server, dan digunakan pada hampir seluruh aplikasi jaringan komputer.

Dua buah komputer yang masing-masing memiliki sebuah kartu jaringan, kemudian dihubungkan melalui kabel maupun nirkabel sebagai medium transmisi data, dan terdapat perangkat lunak sistem operasi jaringan akan membentuk sebuah jaringan komputer yang sederhana. Apabila ingin membuat jaringan komputer yang lebih luas lagi jangkauannya, maka diperlukan peralatan tambahan seperti hub, bridge, switch, router, gateway sebagai peralatan interkoneksinya.

1.4 Python

Bahasa pemrograman tujuan umum yang ditafsirkan, tingkat tinggi. Dibuat oleh Guido van Rossum dan pertama kali dirilis pada tahun 1991, filosofi desain Python menekankan keterbacaan kode dengan penggunaan spasi putih yang signifikan. Konstruksi bahasanya dan pendekatan berorientasi objek bertujuan untuk membantu pemrogram menulis kode yang jelas dan logis untuk proyek skala kecil dan besar.^[28]

Python diketik secara dinamis dan pengumpulan sampah. Ini mendukung beberapa paradigma pemrograman, termasuk pemrograman terstruktur (terutama, prosedural), berorientasi objek, dan fungsional. Python sering dideskripsikan sebagai bahasa "termasuk baterai" karena perpustakaan standarnya yang komprehensif.^[29]

Python dibuat pada akhir 1980-an sebagai penerus bahasa ABC. Python 2.0, dirilis pada tahun 2000, memperkenalkan fitur-fitur seperti pemahaman daftar dan sistem pengumpulan sampah dengan penghitungan referensi.

Python 3.0, dirilis pada tahun 2008, adalah revisi utama dari bahasa yang tidak sepenuhnya kompatibel dengan versi sebelumnya, dan banyak kode Python 2 yang tidak berjalan tanpa modifikasi pada Python 3.

Penerjemah Python tersedia untuk banyak sistem operasi. Komunitas pemrogram global mengembangkan dan memelihara CPython, implementasi referensi^[30] yang bebas dan sumber terbuka. Sebuah organisasi nirlaba, Python Software Foundation, mengelola dan mengarahkan sumber daya untuk pengembangan Python dan CPython.

Python secara konsisten menempati peringkat sebagai salah satu bahasa pemrograman paling populer.



Gambar 1.4 Logo Python

BAB 2

PEMBAHASAN

2.1 TOPIK TUGAS BESAR'

Implementasi pembuatan TCP socket dan mengaitkannya ke alamat dan port tertentu (poin: 20)

Program web server dapat menerima dan memarsing HTTP request yang dikirimkan oleh browser (poin: 20)

Web server dapat mencari dan mengambil file (dari file system) yang diminta oleh client (poin: 15)

Web server dapat membuat HTTP response message yang terdiri dari header dan konten file yang diminta (poin: 20)

Web server dapat mengirimkan response message yang sudah dibuat ke browser (client) dan dapat ditampilkan dengan benar di sisi client (poin: 15)

Jika file yang diminta oleh client tidak tersedia, web server dapat mengirimkan pesan "404 Not Found" dan dapat ditampilkan dengan benar di sisi client. (poin: 10)

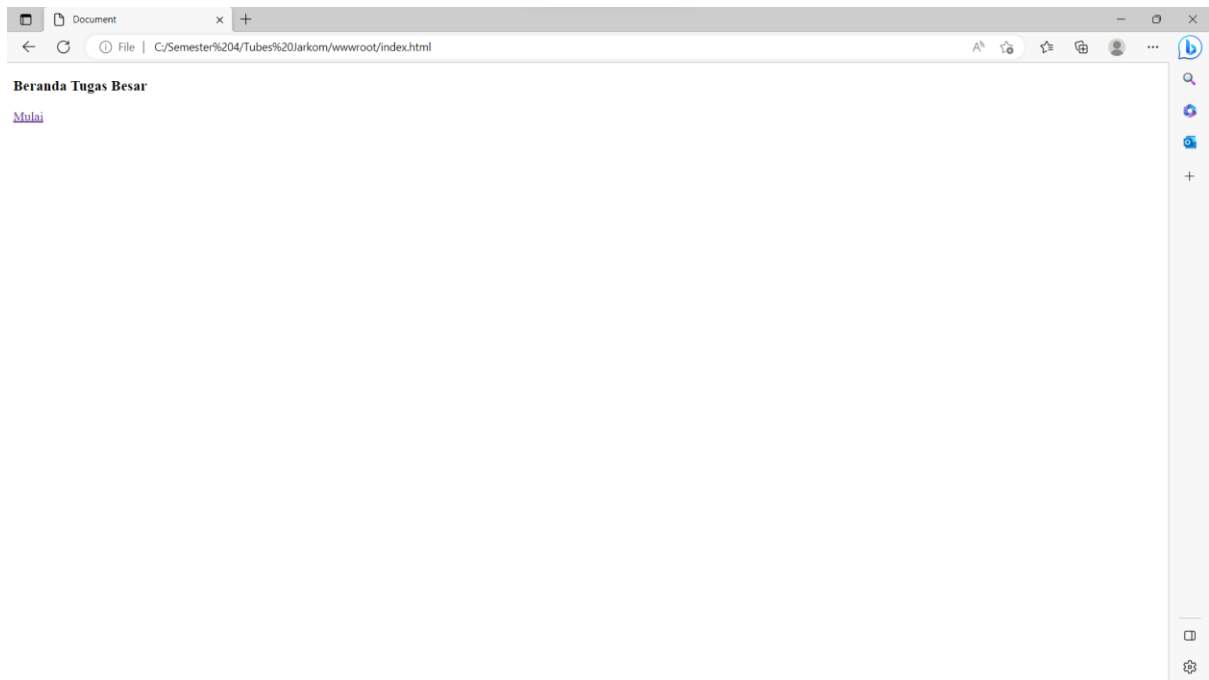
Gambar 2.1 Topik Tugas Besar

2.2 PENJELASAN

Dalam tugas besar ini kami membentuk program web server yang dapat menerima dan memarsing HTTP request yang dikirimkan oleh browser sehingga mendapatkan hasil seperti dibawah



Gambar 2.2 Hasil Program Jika File tidak Ditemukan



Gambar 2.3 Output Ketika File Terbaca

Untuk mendapatkan hasil seperti diatas kami harus membuat program agar memiliki output seperti **Gambar 2.2** dan **Gambar 2.3**

```
TCPServer.py x index.html nama.html TCPClient.py
C:\Semester 4> Tubes Jarkom > TCPServer.py >...
1 import socket #menyediakan low-level interface untuk sebuah jaringan komunikasi
2 import os #menyediakan cara untuk berinteraksi dengan operating system yang bisa melakukan beberapa jenis tugas
3 import mimetypes #mengidentifikasi format file spesifik di internet
4
5 #untuk membuat socket baru, menunjuk IPv4, serta menunjuk bahwa format ini menggunakan TCP
6 server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
7
8 #mengatur alamat IP dan port
9 IP = "127.0.0.1"
10 port = 8080
11
12 # Mengaitkan socket dengan alamat IP dan port
13 server_socket.bind((IP, port))
14
15 #membuat listening socket untuk menerima koneksi
16 server_socket.listen(5)
17
18 print(f"Web server is running on {IP}:{port}...") #print
19
20 while True:
21     client_socket, client_addr = server_socket.accept() #untuk menerima
22     request = client_socket.recv(1024).decode() #untuk menerima data client
23     print(f"Request from {client_addr}: {request}") #print
24
25     request_line = request.split("\r\n")[0] #untuk mengekstrak line pertama dari http request
26     method, path, _ = request_line.split(" ") #untuk memisahkan line http request dengan komponen individu
27
28     #ketika mendapatkan GET maka index.html akan terdapat link
29     if method == "GET":
```

```

30     if path == "/":
31         path = "/index.html"
32
33     #untuk mengikuti 1 atau lebih komponen path
34     file_path = os.path.join("wwwroot", path[1:])
35
36     #memeriksa apakah file ada dalam sistem
37     if os.path.exists(file_path) and os.path.isfile(file_path):
38
39         #menerima ukuran dari file yang diminta
40         file_size = os.path.getsize(file_path)
41
42         #untuk menerima MIME yang diminta dari file extension
43         mime_type, _ = mimetypes.guess_type(file_path)
44
45         #line ini membuat respons dari header yang akan dikirim ke client dengan status OK
46         response = "HTTP/1.1 200 OK\r\n"
47         response += "Content-Type: " + mime_type + "\r\n"
48         response += "Content-Length: " + str(file_size) + "\r\n"
49         response += "\r\n"
50         #mengirim HTTP respons ke client berbentuk bytes menggunakan metode 'client_socket.send()'
51         client_socket.send(response.encode())
52
53         #meminta file dengan ukuran 1024 bytes dan mengirimnya menggunakan metode 'client_socket.send()'
54         with open(file_path, 'rb') as f:
55             while True:
56                 file_data = f.read(1024)
57                 if not file_data:
58                     break
59
60                 client_socket.send(file_data)
61
62         #memanggil link html ketika file html tidak ditemukan
63         else:
64             response = "HTTP/1.1 404 Not Found\r\n"
65             response += "Content-Type: text/html\r\n"
66             response += "\r\n"
67             response += "<html><body><h1>404 Not Found</h1></body></html>"
68             client_socket.send(response.encode())
69
70     #menutup socket
71     client_socket.close()

```

Gambar 2.4 TCP Server Input Program

```

C:\Semester 4 > Tubes Jarkom > TCPClient.py > ...
1  import socket      #menyediakan low-level interface untuk sebuah jaringan komunikasi
2
3  #untuk membuat socket baru, menunjuk IPv4, serta menunjuk bahwa format ini menggunakan TCP dan untuk direktori client
4  client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5
6  #mengaitkan alamat IP dan port
7  client_socket.connect(("127.0.0.1", 8080))
8
9  #menandakan bahwa sinyal dari server sudah di dapat
10 http_request = "GET /index.html HTTP/1.1\r\n"
11 http_request += "Host: 127.0.0.1:8080\r\n"
12 http_request += "\r\n"
13
14 #menerima sinyal http request dari server
15 client_socket.send(http_request.encode())
16
17 #merespon sinyal http request
18 response = client_socket.recv(4096).decode()
19
20 #menerima response dari server
21 print("Received response from server:")
22 print(response)
23
24 #menutup socket
25 client_socket.close()

```

2.5 TCP Client Input Program

Setiap program di atas harus berkesinambungan agar dapat membentuk output yang diinginkan. Dari pemanggilan socket, pencarian file di dalam folder semuanya harus tepat.

Analisa TCPServer

1. Panggil socket, os, dan mimetypes untuk menyediakan interface dan cara untuk berinteraksi dengan operating system
2. Buat object socket baru, socket.AF_INET mengindikasikan bahwa kita akan menggunakan protokol IPv4. Dan socket.SOCK_STREAM mengindikasikan bahwa kita akan menggunakan protokol TCP.
3. Atur alamat IP dan port
4. Kaitkan alamat IP dan port dengan server socket
5. Membuat listening socket untuk menerima koneksi
6. Lalu kita membuat perulangan untuk menerima koneksi dari client, meminta data client, dan print permintaan dari client address
7. Lalu kita meminta line pertama dari http request dengan catatan memisahkan line http request dengan komponen individu
8. Ketika kita mendapat sinyal GET maka index html dapat menampilkan file yang berada di dalam folder wwwroot
9. Wwwroot adalah folder yang kami gunakan untuk menyimpan file .html kami sehingga dapat tertata dengan rapih
10. Lalu kita mencari filepath di dalam sistem, menerima ukuran file yang diminta, dan membuat response ke client bahwa server sudah siap.
11. Jika file tidak ditemukan maka response dari server adalah mengeluarkan 1 buah screen 404 not found di search engine yang kita cari.
12. Menutup socket sebagai penutup TCPServer kami

Analisa TCPClient

1. Memanggil socket untuk menyediakan low – level interface sebuah jaringan
2. Buat object socket baru, socket.AF_INET mengindikasikan bahwa kita akan menggunakan protokol IPv4. Dan socket.SOCK_STREAM mengindikasikan bahwa kita akan menggunakan protokol TCP.
3. Mengaitkan alamat IP dan port
4. Di TCPClient kami berfungsi untuk menerima sinyal dan menandakan server sudah siap atau belum.
5. Ketika sudah menerima sinyal maka akan terprint (“Received response from server : ”)
6. Menutup socket sebagai penutup TCPClient kami

BAB III

KESIMPULAN

Berdasarkan hasil pengerjaan tugas besar Jaringan Komputer, maka dapat disimpulkan :

1. Membuat jaringan aplikasi : Memahami penggunaan socket di python dan basic skill untuk pembuatan jaringan computer
2. Memahami Protokol : Memahami socket dan jaringan menggunakan python dan juga mengerti protocol HTTP dan TCP
3. Debugging masalah jaringan : Memahami socket dan jaringan menggunakan python dapat membantu untuk menganalisis dan debug masalah tentang jaringan dari software yang digunakan

DAFTAR PUSTAKA

[Building a basic HTTP Server from scratch in Python | Codementor](#)

[Python \(programming language\) - Wikipedia](#)

https://id.wikipedia.org/wiki/Jaringan_komputer

Modul Praktikum Jaringan Komputer S1 Informatika

LAMPIRAN

Penugasan :

1. Abiyyu Daffa Haidar Setiawan (1301213519) = Membuat TCPServer bagian pemanggilan socket, pemanggilan server socket, dan pembuatan laporan
2. Ariotegar Ramadhanto Putra(1301213441) = Membuat TCPserver bagian pemanggilan file
3. Rivan Fauzan (1301210554) = Membuat TCPClient