

Web Server Sederhana Berbasis TCP Socket Programming

Disusun untuk memenuhi Tugas Besar Mata Kuliah Jaringan Komputer



Dosen pembimbing :

MUHAMAD FARIS FATHONI, S.T., M.T., Ph.D.

Disusun oleh :

Idlofi Zahir Rajaba (1301213407)

Lulu Maudhuna Alfani (1301213046)

Rofi'ah Budi Nadia (1301213244)

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

(2022/2023)

DAFTAR ISI

DAFTAR ISI.....	1
BAB I.....	2
A. Tujuan Penulisan Laporan.....	2
B. Landasan Teori.....	2
BAB II.....	5
LAMPIRAN.....	6

BAB I

PENDAHULUAN

A. Tujuan Penulisan Laporan

Pada penulisan laporan ini tujuan yang akan dicapai yaitu :

- a. Dapat membuat serta mengimplementasikan TCP socket dan mengaitkannya ke alamat dan port tertentu
- b. Membuat program web server dapat menerima dan memarsing HTTP request yang dikirimkan
- c. Membuat web server dapat mencari dan mengambil file (dari file system) yang diminta oleh client
- d. Membuat web server dapat membuat HTTP response message yang terdiri dari header dan konten file yang diminta
- e. Membuat web server dapat mengirimkan response message yang sudah dibuat ke browser (client) dan dapat ditampilkan dengan benar di sisi client
- f. Membuat web server yang dapat mengirimkan pesan “404 Not Found” dan dapat ditampilkan dengan benar di sisi client.

B. Landasan Teori

a. Jaringan Komputer

Jaringan komputer adalah jaringan telekomunikasi yang memungkinkan antar komputer saling berkomunikasi dengan cara bertukar data. Dua unit komputer dikatakan saling terkoneksi apabila keduanya dapat saling bertukar data atau informasi, berbagi resource yang dimiliki, seperti file, printer, media penyimpanan (hard disk, floppy disk, cd-rom, flash disk, dll). Data yang berupa teks, audio, maupun video bergerak melalui media kabel atau tanpa kabel (wireless) sehingga memungkinkan pengguna komputer dalam jaringan komputer dapat saling bertukar file atau data, mencetak pada printer yang sama dan menggunakan hardware atau software yang terhubung dalam jaringan secara bersama-sama. Sebuah jaringan komputer sekurang-kurangnya terdiri dari dua unit komputer atau lebih, dapat berjumlah puluhan komputer, ribuan, atau bahkan jutaan node yang terhubung satu sama lain. Node merupakan komputer, printer, atau periferal yang terhubung dalam jaringan.

Di dalam jaringan komputer dikenal sistem koneksi antar node, yakni:

1. Peer to peer

Peer to peer network merupakan jaringan komputer yang terdiri dari tidak lebih dari 10 komputer dengan 1-2 printer. Model peer to peer dapat dikembangkan untuk koneksi lebih dari 10 hingga 100 komputer untuk penggunaan khusus, seperti laboratorium komputer, riset, dan lain-lain.

2. Client - Server

Sistem ini dapat diterapkan dengan teknologi internet di mana ada suatu unit komputer yang berfungsi sebagai server yang memberikan layanan bagi komputer lain, dan client yang meminta layanan dari server. Client hanya bisa menggunakan resource yang disediakan server sesuai dengan otoritas yang diberikan administrator.

- b. Socket Programming

Jaringan komputer merupakan jalur komunikasi antara banyak entitas. Jalur ini menyediakan layanan untuk mengirim data dari pengirim ke penerima. Data yang dikirim akan berjalan melalui mekanisme lapisan jaringan yang diproses oleh program, yaitu pemrograman soket (socket programming). Socket programming merupakan salah satu metode untuk mengamankan jalur komunikasi pada suatu jaringan. Socket programming adalah protokol yang membuat koneksi antar perangkat dalam jaringan yang tujuannya agar suatu program dapat berinteraksi dengan program lainnya. Konsep komunikasi dengan socket hampir sama seperti operasi I/O. aplikasi dapat membaca atau menulis terhadap socket tersebut dengan mudah seperti ketika aplikasi menulis sesuatu dari atau ke dalam file. Socket juga dapat diibaratkan seperti kunci yang memberikan akses menuju port yang tepat. Protokol yang digunakan pada komunikasi socket pada sistem yang dibangun adalah TCP (Transmission Control Protocol).

- c. TCP

TCP (Transmission Control Protocol) yaitu protokol yang bertanggung jawab untuk pengiriman data dari sumber ke tujuan dengan benar. TCP dapat mendeteksi kesalahan atau hilangnya data dan kembali melakukan pengiriman sampai data diterima dengan lengkap. Untuk memastikan apakah data telah sampai di tempat tujuan, TCP akan selalu meminta konfirmasi setiap kali data dikirim. Kemudian, mengirim data berikutnya atau melakukan retransmisi (pengiriman ulang) apabila data sebelumnya tidak sampai atau rusak. Data yang dikirim dan diterima kemudian diatur berdasarkan nomor urut. Tugas TCP yaitu

menerima pesan elektronik dengan panjang sembarang dan membaginya dalam bagian-bagian berukuran 64 kb. Dengan adanya pembagian pesan tersebut, perangkat lunak yang mengontrol komunikasi jaringan dapat mengirim tiap bagian dan menyerahkan prosedur pemeriksaan bagian demi bagian.

Protokol TCP memiliki karakteristik sebagai berikut:

1. *Connection oriented*, aplikasi meminta koneksi dan menggunakannya dalam transfer data.
2. *Point to point communication*, setiap koneksi TCP pasti memiliki dua titik.
3. *Reliability*, TCP menjamin bagi data yang dikirimkan dalam koneksi dapat terkirim dengan pasti tanpa ada yang hilang atau double.
4. *Full duplex connection*, koneksi TCP memperbolehkan data untuk berkomunikasi dari salah satu titik koneksi setiap saat.
5. *Stream interface*, TCP memperbolehkan aplikasi untuk mengirimkan koneksi yang berkesinambungan.
6. *Reliable startup*, membutuhkan persetujuan dari kedua aplikasi untuk melakukan koneksi baru.
7. *Graceful shutdown*, aplikasi dapat membuka aplikasi, mengirim data dan menutup koneksi serta menjamin bahwa data sampai sebelum koneksi terputus.

TCP bersifat *connection oriented*, artinya sebelum proses transmisi data terjadi, dua aplikasi TCP harus melakukan pertukaran control informasi (*handshaking*). TCP juga bersifat reliable karena menerapkan fitur deteksi kesalahan dan retransmisi apabila ada data yang rusak. Program pengirim hanya perlu mengulang transmisi bagian yang rusak dan tidak perlu mengulang dari awal apabila terjadi kerusakan selama transmisi.

BAB II

PELAKSANAAN

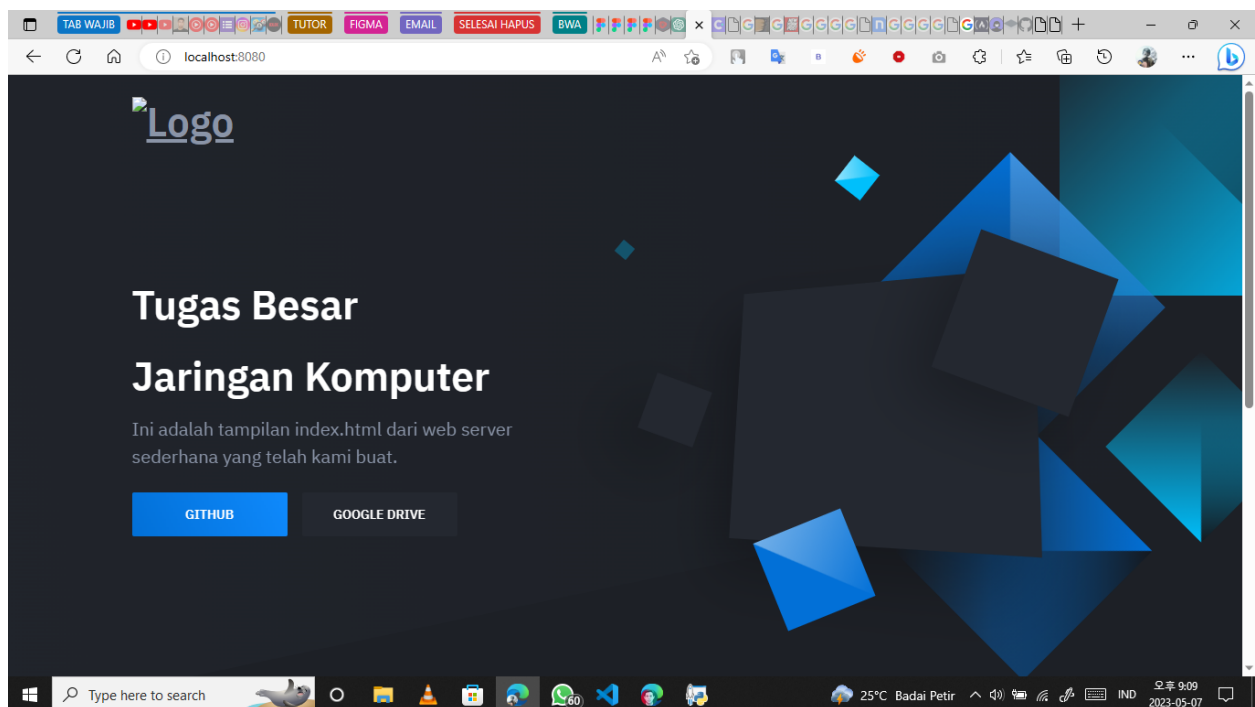
1. Isi file server.py

```
1 import socket
2 import os
3
4 # fungsi untuk menangani request yang masuk
5 def handle_request(request):
6     headers = request.split('\n')
7     filename = headers[0].split()[1]
8     if filename == '/':
9         filename = '/index.html'
10
11     try:
12         with open('.' + filename, 'rb') as fin:
13             content = fin.read()
14             response = 'HTTP/1.0 200 OK\n\n'.encode() + content
15     except FileNotFoundError:
16         response = 'HTTP/1.0 404 NOT FOUND\n\nFile Not Found'.encode()
17
18     return response
19
20 # definisikan host dan port
21 HOST = 'localhost'
22 PORT = 8080
23
24 # buat socket dan pasang ke host dan port tertentu
25 server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
26 server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
27 server_socket.bind((HOST, PORT))
28 server_socket.listen(1)
29 print(f'Listening on port {PORT} ...')
30
31 while True:
32     # tunggu koneksi dari client
33     client_connection, client_address = server_socket.accept()
34
35     # terima request dari client
36     request = client_connection.recv(1024).decode()
37     print(request)
38
39     # kirim response ke client
40     response = handle_request(request)
41     client_connection.sendall(response)
42
43     # tutup koneksi
44     client_connection.close()
45
46 # tutup socket
47 server_socket.close()
48
```

2. Jalankan server.py



3. Hasil running localhost:8080



LAMPIRAN

Link Github:

https://github.com/idlofizahir86/tubes_jarkom