

**JARINGAN KOMPUTER
LAPORAN FINAL TUGAS BESAR**

Kelompok 11

KELAS IF 45 09

DOSEN: FAZ



**Telkom
University**

Dipersiapkan Oleh:

Syehan Fariz Gustomo - 1301210530

M.Rafi Athallah - 1301210210

Nandika Abiyoga Santosa - 1301213421

**PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
BANDUNG
2023**

Daftar Isi

Pendahuluan.....	3
Latar Belakang.....	3
Batasan Masalah.....	4
Sistem Yang Di Bangun.....	4
Cara Kerja Program.....	5
Hasil Program Dan Analisis.....	5
Kesimpulan.....	6
Referensi.....	6
Kontribusi Kelompok.....	6

Pendahuluan

Topik yang kami ambil adalah Web Server berbasis html, dimana saat klien mengirim request ke server, maka server akan memberikan balasan dan klien akan masuk ke halaman index dari html yang berupa bio data kelompok kami. Bila permintaan request dari klien tidak ditemukan oleh server maka server akan merespon dengan halaman html 404 Not Found

Latar Belakang

Web server merupakan salah satu teknologi yang berperan penting dalam infrastruktur internet di era digital saat ini. Web server dapat digunakan untuk melayani client melalui protokol HTTP atau HTTPS, dan memiliki fungsi penting dalam mengakses data serta informasi yang disimpan oleh server dengan cara memberikan respon berupa konten yang diminta oleh client.

Salah satu cara untuk membuat web server yang sederhana adalah dengan menggunakan TCP socket programming. Dalam metode TCP socket programming, server berfungsi untuk menerima koneksi dari client, kemudian server akan mengambil data yang diinginkan client serta memproses data tersebut yang kemudian akan mengirimkan respon balik ke client.

Batasan Masalah

Batasan masalah yang dibuat:

1. Web server dapat mengirim respon atas request dari client
2. Web server dapat menerima request yang dikirim client
3. Web server dapat mengirim file yang diminta oleh client (terbatas hanya pada file html,css dan txt)

Sistem Yang Di Bangun

```
import socket.py > handle_request
1  import socket # mengimpor modul socket, yang digunakan untuk membuat dan mengendalikan socket di Python.
2
3  HOST = '127.0.0.1' # menentukan alamat HOST (localhost) dan PORT (8080) yang akan digunakan oleh server socket.
4  PORT = 8080
5
6  # HTML for 404 error
7  not_found = """\
8  HTTP/1.1 404 Not Found
9  Content-Type: text/html; charset=utf-8
10
11  """
12  not_found += open("404notfound.html").read() # untuk membaca file html 404notfound
13
14  # HTML for index page
15  index_html = """\
16  HTTP/1.1 200 OK
17  Content-Type: text/html; charset=utf-8
18
19  """
20  index_html += open("jarkom tubes.html").read() # untuk membaca file html index
21
22  # Fungsi untuk menangani permintaan client
23  def handle_request(conn): # menerima objek 'conn' sebagai argumen, yang merupakan koneksi socket dengan client
24      # Get the client request
25      request = conn.recv(1024).decode('utf-8') # Fungsi akan menerima permintaan client dan mengkonversinya menjadi byte
26      if not request: # Fungsi mengembalikan nilai return bila tidak ada permintaan yang diterima dari client
27          return
28
29      # Mengambil nama file yang diminta dari request yang diterima
30      filename = request.split()[1] # Permintaan file tsb dibagi menjadi potongan2 berdasarkan spasi, dan elemen ke-2 '[1]' merupakan file yg diminta
31
32      # Serve index page by default if no filename provided
33      if filename == '/':
34          response = index_html.encode('utf-8')
35      else:
36          # Try to open the file
37          try:
```

Pertama, kami mengimport socket untuk membuat socket di python, kemudian menentukan alamat IP dan port. Setelah itu kita memasukkan file html 404 Not Found dan index sebagai output dari respon web ke klien. Lalu terdapat fungsi handle_request dimana bila web server menerima request maka akan dikonversi menjadi byte dengan .decode('utf-8'). Bila tidak ada permintaan dari klien maka fungsi akan mengembalikan nilai return.

```

32 # Serve index page by default if no filename provided
33 if filename == '/':
34     response = index_html.encode('utf-8')
35 else:
36     # Try to open the file
37     try:
38         with open('.' + filename, 'rb') as f: # mencoba membuka file yang diminta ('.' + filename) dalam mode baca biner ('rb')
39             content = f.read() # jika file tersebut ditemukan, kontennya akan dibaca dan disimpan dalam variabel content.
40         # Build the response # Baris ini membangun respons yang akan dikirimkan kembali ke klien.
41         response = b'HTTP/1.1 200 OK\n' # inisiasi
42         if filename.endswith('.html'): # tipe konten (content-type) yang sesuai ditambahkan ke respons berdasarkan ekstensi yang diminta.
43             response += b'Content-Type: text/html; charset=utf-8\n'
44         elif filename.endswith('.css'):
45             response += b'Content-Type: text/css; charset=utf-8\n'
46         else: # respons untuk ekstensi selain tipe konten html dan css.
47             response += b'Content-Type: text/plain\n'
48         response += b'\n' + content # konten file ditambahkan ke respons menggunakan b'\n' + content.
49     except FileNotFoundError:
50         # File not found, return 404 error
51         response = not_found.encode('utf-8') # Jika file tidak ditemukan, maka response di enkripsi menjadi byte menggunakan utf-8
52
53 # Send the response
54 conn.sendall(response) # mengirim response ke klien
55 conn.close() # Setelah terkirim, socket ditutup

```

Baris kode diatas untuk menampilkan file yang diminta oleh klien, apabila klien meminta file html, maka web server akan menampilkan html dan bitupun untuk jenis file css dan html. Apabila klien meminta selain dari ke-tiga jenis tsb atau file tidak ditemukan, maka web server akan mengembalikan file html 404notfound.

```

57 # Create the socket and start listening
58 with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s: # membuat socket menggunakan socket.socket(socket.AF_INET, socket.SOCK_STREAM)
59     s.bind((HOST, PORT)) # lalu mengaitkannya dengan alamat HOST dan PORT menggunakan s.bind((HOST, PORT)).
60     s.listen() # kemudian, socket mulai mendengarkan permintaan masuk menggunakan s.listen().
61     print(f'Server listening on http://{HOST}:{PORT}') # Pesan yang mencetak URL server saat sedang 'listening'
62     # Accept incoming connections
63     while True:
64         conn, addr = s.accept() # socket menerima koneksi masuk
65         print('Connected by {addr[0]}:{addr[1]}') # ketika koneksi diterima, informasi tentang koneksi tersebut dicetak
66         handle_request(conn) # fungsi ini dipanggil untuk menangani permintaan dari klien tersebut.

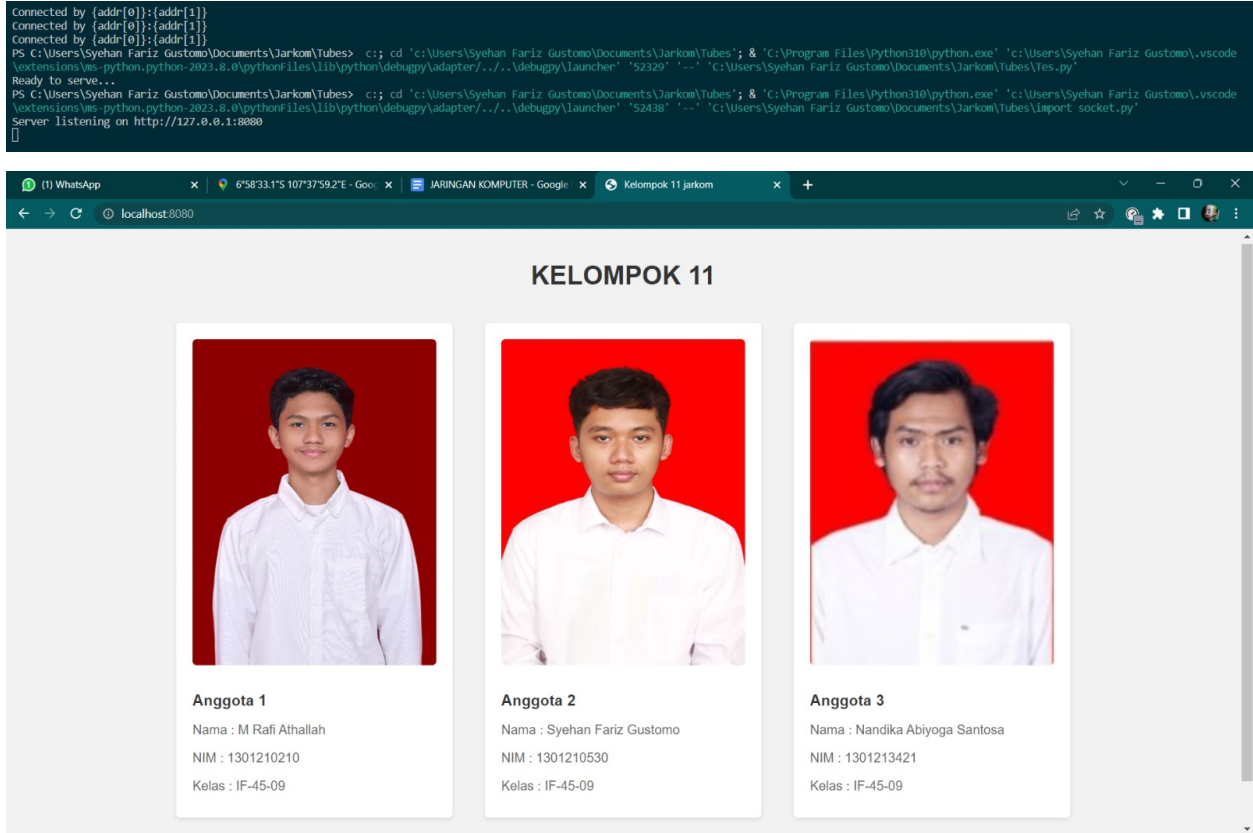
```

Lalu kita akan mengaitkan socket dengan alamat host dan port, kemudian socket mulai mendengarkan permintaan masuk menggunakan s.listen(). Terdapat juga perulangan yang berisi socket yang menerima permintaan masuk, dan ketika koneksi diterima, informasi tersebut akan dicetak kemudian fungsi handle_request akan dipanggil kembali untuk menangani request dari klien.

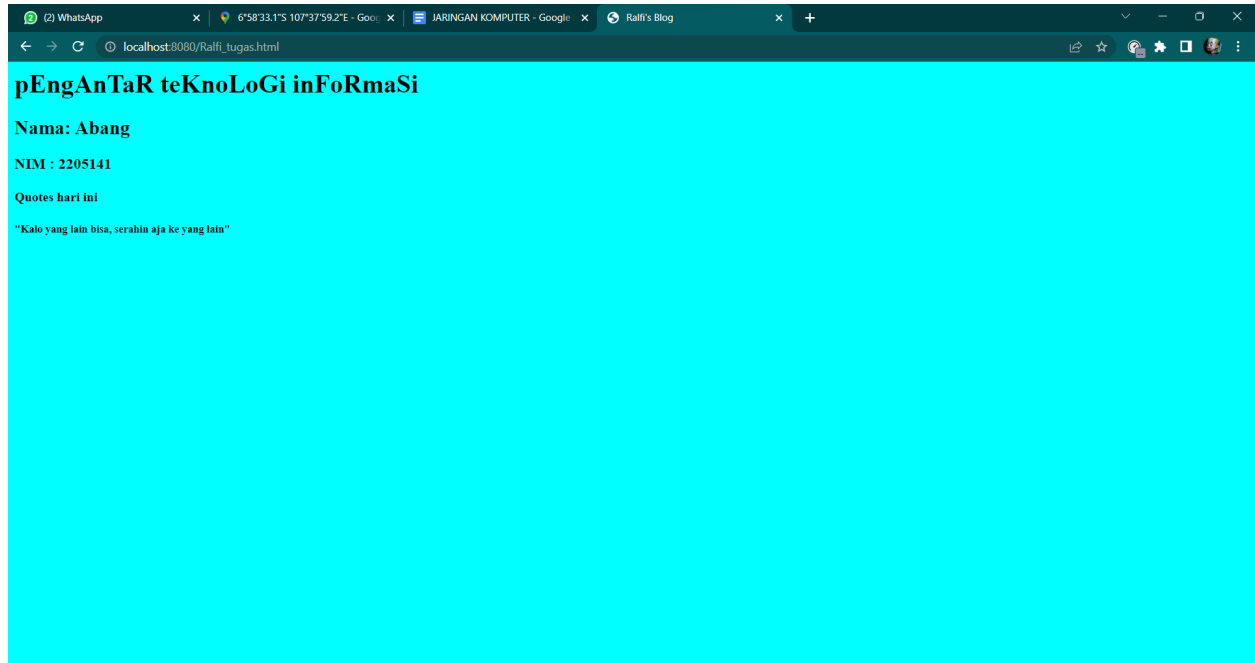
Cara Kerja Program

Kami membuat sistem Web Server untuk menampilkan file yang sesuai dengan klien, ketika klien mengakses web server akan menampilkan file index yang berisi biodata kelompok kami. Lalu ketika client meminta file yag lebih spesifik maka server akan merespon file yang diminta oleh klien namun terbatas hanya pada tipe jenis file html, css, dan txt. ketika client meminta file selain tipe jenis file html, css, dan txt atau file nya tidak ditemukan maka server akan mengeluarkan halaman 404 NOT FOUND.

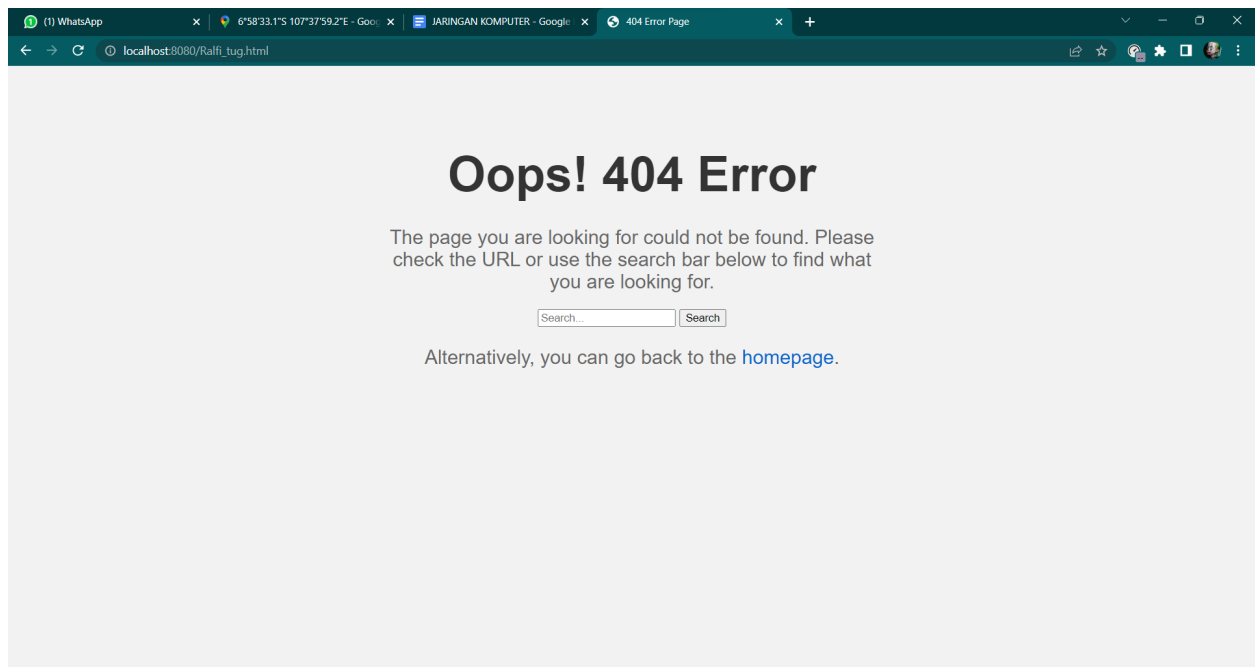
Hasil Program Dan Analisis



Gambar diatas adalah hasil dari program yang kami buat. Ketika client meminta menampilkan bio data dari kelompok kami.



Web server akan merespon file yang diminta oleh klien, karena klien meminta file html, maka server menampilkan file html.



Web server akan menampilkan 404notfound karena request klien tidak ditemukan.

Kesimpulan

Code dapat berjalan dengan baik karena dapat merespon klien sesuai apa yang dibutuhkan dari klien dan apabila jika klient meminta yang tidak sesuai maka akan menampilkan halan 404 NOT FOUND

Referensi

<https://stackoverflow.com/>
<https://www.w3schools.com/>
<https://www.youtube.com/>

Kontribusi Kelompok

Rafi: :membuat HTML dan CSS dan menyumbangkan ide pikiran web server berbasis socket programming dan membuat laporan.

Syehan : ikut menyumbangkan ide untuk membuat web server berbasis socket programing dan ikut membuat laporan.

Nandika : Laporan

Untuk codingan, kami membuat bersama di satu laptop dan di satu tempat.

Link video youtube = <https://youtu.be/6uvF6jxR0Js>