

LAPORAN PRAKTIKUM PEMROGRAMAN JARINGAN

“Client – Server (Single Thread)”



Dibuat oleh :

Afifah Fikriyah (1203220051)

**PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM UNIVERSITY
SURABAYA**

2024

TUGAS DAN LATIHAN PRAKTIKUM

1. Membuat laporan percobaan praktikum dan beri Analisa Hasil Percobaan tadi yang sudah dibuat Pembuatan Aplikasi Client-Server Sederhana (Single Thread)
2. Membuat sebuah program server yang dapat menerima koneksi dari klien menggunakan protokol TCP. Server ini akan menerima pesan dari klien dan mengirimkan pesan balasan berisi jumlah karakter pada pesan tersebut. Gunakan port 12345 untuk server. Membuat analisa dari hasil program tersebut

Screenshot :

3. Membuat sebuah program klien yang dapat terhubung ke server yang telah dibuat pada soal nomor 1. Klien ini akan mengirimkan pesan ke server berupa inputan dari pengguna dan menampilkan pesan balasan jumlah karakter yang diterima dari server. Membuat analisa dari hasil program tersebut

Screenshot :

1. Single Thread

```
client.py X
client.py > ...
1 import socket
2 client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
3 HOST = 'localhost'
4 PORT = 12345
5 client_socket.connect((HOST, PORT))
6 pesan = input("Masukkan angka :")
7 client_socket.sendall(pesan.encode())
8 data = client_socket.recv(1024)
9 print(data.decode())
10 client_socket.close()
11
```

```
server.py X
server.py > ...
1 import socket
2 server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
3 HOST = 'localhost'
4 PORT = 12345
5 server_socket.bind((HOST, PORT))
6 server_socket.listen(1)
7 print("Waiting...")
8 client_socket, client_address = server_socket.accept()
9 data = client_socket.recv(1024)
10 angka = int(data.decode())
11 print("Request dari client :", angka, "IP client :", client_address)
12 if angka % 2 == 0:
13     response = "angka " + str(angka) + " merupakan genap"
14 else:
15     response = "angka " + str(angka) + " merupakan ganjil"
16 client_socket.sendall(response.encode())
17 client_socket.close()
18 server_socket.close()
```

Pembuatan aplikasi client server sederhana (single thread) menggunakan python dapat memanfaatkan library socket yang telah disediakan oleh python

1.1 Client

- Merupakan perangkat lunak yang mengakses dan menggunakan layanan atau sumber daya yang disediakan oleh server, contoh jika server memberi akses untuk browser web maka yang akan direspon oleh client adalah akses pada web itu sendiri
- Cara membuat aplikasi client server sederhana adalah kita harus memperhatikan hal-hal berikut :

- a. Pastikan dalam client.py telah diimport library socket, dan objek socket telah dibuat

```
client.py > ...  
1 import socket  
2 client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

- b. Kemudian ikat alamat client pada alamat server (port,host,ip,dll)

```
3 HOST = 'localhost'  
4 PORT = 12345
```

- c. Jangan lupa untuk beri koneksi server ke client

```
5 client_socket.connect((HOST, PORT))
```

- d. Mengirim dan menerima data antara server dan client

```
pesan = input("Masukkan angka :")  
client_socket.sendall(pesan.encode())  
data = client_socket.recv(1024)  
print(data.decode())
```

- e. Menutup koneksi antara server dan client

```
10 client_socket.close()
```

1.2 Server

- Merupakan perangkat lunak yang menyediakan layanan atau sumber daya ke dalam jaringan, sebuah server biasanya menyimpan dan mengakses data yang akan digunakan dalam sebuah komunikasi
- Peran server dalam aplikasi client-server sederhana:
 - a. Adanya objek socket pada server

```
import socket  
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

- b. Menngikat objek socket pada server ke alamat tertentu

```
3 HOST = 'localhost'  
4 PORT = 12345  
5 server_socket.bind((HOST, PORT))  
6 server_socket.listen(1)  
7 print("waiting...")  
8 client_socket, client_address = server_socket.accept()
```

- c. Mengirim dan menerima data antara server dan client

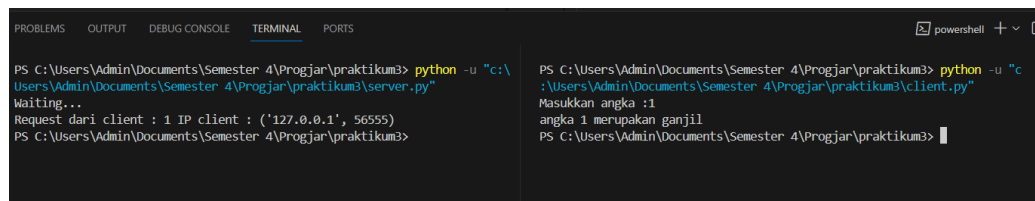
```
data = client_socket.recv(1024)  
angka = int(data.decode())  
print("Request dari client :", angka, "IP client :", client_address)  
if angka % 2 == 0:  
    response = "angka " + str(angka) + " merupakan genap"  
else:  
    response = "angka " + str(angka) + " merupakan ganjil"  
client_socket.sendall(response.encode())
```

- d. Menutup koneksi antara server dan client

```
client_socket.close()  
server_socket.close()
```

1.3 Kesimpulan:

Sebuah server sangat berperan penting dalam komunikasi yang terjadi antara client dan server dikarenakan server memiliki alamat yang akan dihubungkan pada client, dan ketika alamat itu dihubungkan server akan mencoba mendengarkan sinyal yang ada pada server jika berhasil client socket dan client address akan diterima oleh server dan di antara server dan client akan dilakukan kegiatan mengirim data di mana client meminta input dari server dan server akan memberi balasan input yang diminta oleh client, sebagai contoh kasus pada penentuan angka ganjil atau genap oleh server yang diminta oleh client dan client menerima hasil penentuan angka ganjil atau genap oleh server yang dapat dilihat di bawah ini:



```
PS C:\Users\Admin\Documents\Semester 4\Progar\praktikum3> python -u "c:\Users\Admin\Documents\Semester 4\Progar\praktikum3\server.py"
Waiting...
Request dari client : 1 IP client : ('127.0.0.1', 56555)
PS C:\Users\Admin\Documents\Semester 4\Progar\praktikum3>

PS C:\Users\Admin\Documents\Semester 4\Progar\praktikum3> python -u "c:\Users\Admin\Documents\Semester 4\Progar\praktikum3\client.py"
Masukkan angka :1
angka 1 merupakan ganjil
PS C:\Users\Admin\Documents\Semester 4\Progar\praktikum3>
```

2. Server.py

```
import socket

server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

HOST = 'localhost'
PORT = 12345

server_socket.bind((HOST, PORT))
server_socket.listen(1)

print("Waiting...")

client_socket, client_address = server_socket.accept()

data = client_socket.recv(1024)
input_string = data.decode()

print(f"Request dari client :", input_string, "IP client :", client_address)
response = "Jumlah Karakter : " + str(len(input_string))
client_socket.sendall(response.encode())

client_socket.close()
server_socket.close()
```

Seperti biasa di dalam server kita harus membuat objek socket terlebih dahulu untuk melakukan komunikasi nanti kepada client, protokol yang digunakan adalah TCP terlihat dari objek socket yang diisi parameter SOCK_STREAM, kemudian inisialisasi host dan port agar dapat server memiliki alamat yang dapat dihubungkan kepada client, lalu host dan port akan diikat melalui fungsi bind dan melakukan pendengaran sinyal dengan fungsi listen

Jika berhasil maka objek socket dari client beserta alamatnya akan diterima dan akan melakukan pengiriman data, dalam hal ini client akan meminta server untuk menghitung jumlah karakter yang diinput oleh client

3. Client.py

```
1
2 import socket
3
4 client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
5 HOST = 'localhost'
6 PORT = 12345
7 client_socket.connect((HOST, PORT))
8
9 pesan = input("Masukkan kalimat: ")
10 client_socket.sendall(pesan.encode())
11 data = client_socket.recv(1024)
12 print(data.decode())
13
14 client_socket.close()
```

Sama halnya dengan server, client juga harus memiliki objek socket yang berprotokol sama dengan server, client juga harus memiliki host dan port yang sama dengan server, kemudian client akan mencoba untuk mengirim sinyal pada server dengan fungsi connect jika berhasil maka pengiriman data dapat dilakukan, dalam hal ini client akan memasukkan kalimat yang akan dihitung total hurufnya pada server dan hasilnya akan dibaca oleh client

4. Output

```
PS C:\Users\Admin\Documents\Semester 4\Progar\praktikum3> python -u "c:\Users\Admin\Documents\Semester 4\Progar\praktikum3\tugass\server.py"
Waiting...
Request dari client : bismillah bisa! IP client : ('127.0.0.1', 56592)
PS C:\Users\Admin\Documents\Semester 4\Progar\praktikum3>

PS C:\Users\Admin\Documents\Semester 4\Progar\praktikum3> python -u "c:\Users\Admin\Documents\Semester 4\Progar\praktikum3\tugass\client.py"
Masukkan kalimat: bismillah bisa!
Jumlah Karakter : 15
PS C:\Users\Admin\Documents\Semester 4\Progar\praktikum3>
```