Tugas 1 - Socket Programming On python

1. Membuat sebuah program server yang dapat menerima koneksi dari client menggunakan protokol TCP. Server ini akan menerima pesan dari client dan mengirimkan pesan balasan berisi jumlah karakter pada pesan tersebut. Gunakan port 12345 untuk server.

2. Membuat analisa dari hasil program tersebut

Membuat sebuah program client yang dapat terhubung ke server yang telah dibuat pada soal nomor 1. Client ini akan mengirimkan pesan ke server berupa inputan dari pengguna dan menampilkan pesan balasan jumlah karakter yang diterima dari server.

3. Membuat analisa dari hasil program tersebut

upload di github dan kumpulkan url repositorinya

Jawaban:

1. Server:

```python
import socket

def hitung_jumlah_karakter(pesan):
    return str(len(pesan))

server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.bind(('localhost', 12345))
server_socket.listen(5)

print("Server berjalan. Menunggu koneksi...")

while True:
    client_socket, client_address = server_socket.accept()
    print("Terhubung dengan", client_address)

    pesan_dari_klien = client_socket.recv(1024).decode()

    if pesan_dari_klien:
        print("Pesan dari klien:", pesan_dari_klien)

        jumlah_karakter = hitung_jumlah_karakter(pesan_dari_klien)

        client_socket.send(jumlah_karakter.encode())
        print("Jumlah karakter pesan:", jumlah_karakter)

    client_socket.close()
```

Client:

```python
import socket

client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect(('localhost', 12345))

pesan_ke_server = input("Masukkan pesan untuk dikirim ke server: ")

client_socket.send(pesan_ke_server.encode())

balasan_dari_server = client_socket.recv(1024).decode()

print("Balasan dari server:", balasan_dari_server)

client_socket.close()
```

Output server:

```
Server berjalan. Menunggu koneksi...
Terhubung dengan ('127.0.0.1', 51346)
Pesan dari klien: Hello, server!
Jumlah karakter pesan: 14
```

Output client:

```
Masukkan pesan untuk dikirim ke server: Hello, server!
Balasan dari server: 14
```

Analisa:

A. Server.py

- Import Library: Mengimpor modul socket untuk menggunakan fungsi-fungsi socket dalam Python.
- Fungsi hitung_jumlah_karakter: Fungsi ini menerima sebuah string dan mengembalikan jumlah karakternya dalam bentuk string.
- Inisialisasi Socket Server: Membuat socket server menggunakan socket.socket() dengan parameter socket.AF_INET untuk menentukan domain alamat (IPv4) dan socket.SOCK_STREAM untuk menentukan jenis socket (TCP).
- Bind Socket: Mengikat (bind) socket ke alamat localhost dan port 12345 menggunakan metode bind().
- Listen for Connections: Memulai mendengarkan (listen) koneksi masuk dengan metode listen(). Parameter 5 menentukan jumlah maksimum koneksi yang diterima oleh server.
- Loop untuk Menerima Koneksi: Server memasuki loop tak terbatas untuk terus menerima koneksi masuk dari client.
- Terima Koneksi: Ketika ada koneksi masuk, server menerima koneksi dan mendapatkan objek socket baru untuk berinteraksi dengan client. Fungsi accept() akan mengembalikan objek socket client dan alamat client.
- Terima Pesan dari client: Server menerima pesan dari client menggunakan metode recv() dengan buffer size 1024. Pesan tersebut kemudian di-decode dari byte menjadi string.

- Proses Pesan: Jika pesan diterima dari client, server akan menghitung jumlah karakter pesan menggunakan fungsi hitung_jumlah_karakter.
- Kirim Balasan: Server mengirim balasan berupa jumlah karakter pesan kembali ke client menggunakan metode send(). Sebelumnya, jumlah karakter dikonversi menjadi string dan diencode ke dalam byte.
- Tutup Koneksi: Setelah mengirim balasan, koneksi dengan client ditutup menggunakan metode close().

B. client.py

- Import Library: Mengimpor modul socket untuk menggunakan fungsi-fungsi socket dalam Python.
- Inisialisasi Socket client: Membuat socket client menggunakan socket.socket() dengan parameter socket.AF_INET untuk menentukan domain alamat (IPv4) dan socket.SOCK_STREAM untuk menentukan jenis socket (TCP).
- Koneksi ke Server: client melakukan koneksi ke server dengan menggunakan metode connect() dengan alamat localhost dan port 12345.
- Mengirim Pesan ke Server: client meminta pengguna untuk memasukkan pesan yang akan dikirim ke server, kemudian pesan tersebut dikirim menggunakan metode send(). Pesan tersebut diencode menjadi byte sebelum dikirim.
- Terima Balasan dari Server: client menerima balasan dari server menggunakan metode recv() dengan buffer size 1024. Balasan tersebut kemudian di-decode dari byte menjadi string.
- Tampilkan Balasan: Balasan dari server, yaitu jumlah karakter pesan, ditampilkan ke pengguna.
- Tutup Koneksi: Setelah menerima balasan, koneksi dengan server ditutup menggunakan metode close().

2. Client:

```python
import socket

def main():
    # Inisialisasi socket klien
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    # Koneksi ke server
    server_address = ('localhost', 12345)
    client_socket.connect(server_address)

    try:
        # Meminta pengguna untuk memasukkan pesan
        pesan = input("Masukkan pesan untuk dikirim ke server: ")

        # Kirim pesan ke server
        client_socket.sendall(pesan.encode())

        # Terima balasan dari server
        balasan = client_socket.recv(1024).decode()

        # Tampilkan balasan
        print("Balasan dari server:", balasan)

    finally:
        # Tutup koneksi
        client_socket.close()

if __name__ == "__main__":
    main()
```

Output client:

```
Masukkan pesan untuk dikirim ke server: Ini adalah sebuah pesan dari klien.
Balasan dari server: 30
```

Analisa:
Client diminta untuk memasukkan sebuah pesan. Dalam contoh ini, pengguna memasukkan "Ini adalah sebuah pesan dari client." Lalu client mengirimkan pesan tersebut ke server. Server menghitung jumlah karakter dalam pesan tersebut (30 karakter) dan mengirim balasan kembali ke client. Client menerima balasan dari server, yaitu "30", dan menampilkannya ke pengguna.