LAPORAN PRAKTIKUM PEMROGRAMAN JARINGAN

“Client – Server (Single Thread)”

Dibuat oleh :

Hammam Jauharul Karim - 1203222050

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)

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

```python
import socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
HOST = 'localhost'
PORT = 12345
client_socket.connect((HOST, PORT))
pesan = input("Masukkan angka :")
client_socket.sendall(pesan.encode())
data = client_socket.recv(1024)
print(data.decode())
client_socket.close()
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

PS D:\Kuliah\Semester 4\Pemrograman Jaringan\Kuliah\Praktikum\3> & C:/Users/hamma/AppD
/Semester 4/Pemrograman Jaringan/Kuliah/Praktikum/3/Server.py"
Waiting...
Request dari client : 5 IP client : ('127.0.0.1', 59833)
PS D:\Kuliah\Semester 4\Pemrograman Jaringan\Kuliah\Praktikum\3>
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

PS D:\Kuliah\Semester 4\Pemrograman Jaringan\Kuliah\Praktikum\3> & C:/Users/ha
/Semester 4/Pemrograman Jaringan/Kuliah/Praktikum/3/Client.py"
Masukkan angka :5
angka 5 merupakan ganjil
PS D:\Kuliah\Semester 4\Pemrograman Jaringan\Kuliah\Praktikum\3>
```

- **Analisa Hasil**

  Program tersebut merupakan contoh kode Python yang menggunakan socket untuk mengirim dan menerima data antara client dan server. Kode ini terdiri dari dua bagian, yaitu server dan client. Adapun program tesebut adalah sebuah program untuk saling berkomunikasi antara server dengan klien, dimana server menunggu koneksi dari klien dan setelah tersambung maka klien dapat menginputkan sebuah angka yang nantinya angka tersebut akan dikirim kepada server dan akan di cek apakah angka tersebut genap atau ganjil. Hasil output akan dikirim oleh server kepada klien.

- **Analisa Code Server**

  Pada code server ia menggunakan protocol Pv4 (socket.AF_INET) dan protokol stream (socket.SOCK_STREAM) sebagai semacam protocol aturan yang harus diikuti pada saat terjadinya komunikasi antara server dengan klien. Server membind socket ke alamat lokal (localhost) dan port (12345). Setelah itu, server mengulangi perintah listen(1) sehingga server dapat menerima koneksi client. Kemudian, server menunggu sampai ada koneksi client yang terhubung. Adapun data dari client akan diterima oleh server mengunakan metode accept() dan disimpan dalam sebuah variable data. Setelah itu, server mengirim balasan ke client menggunakan metode sendall(). Setelah itu, server menghentikan koneksi dengan client dan client_socket.

- **Analisa Code Client**

  Sama seperti pada server, klien menggunakan protocol Pv4 (socket.AF_INET) dan protokol stream (socket.SOCK_STREAM) untuk aturan komunikasinya. Client mengirinkan ke alamat lokal (localhost) dan port (12345). Setelah itu, client mengirim pesan ke server menggunakan metode sendall(). Adapun pesan balasan dari server atas kiriman pesan yang sebelumnya dikirimkan akan diterima menggunakan metode recv(). Ketika proses komunikasi sudah selesai clien akan memutus koneksi dari server oleh client_socket.close.

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 :

```python
# Server
import socket

def main():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.settimeout(10)
    server_address = ('localhost', 12345)

    try:
        server_socket.bind(server_address)
        server_socket.listen(2)
        print(f'Server listening on {server_address}...')

        while True:
            try:
                client_socket, client_address = server_socket.accept()
                print(f'Menerima koneksi dari {client_address}')

                received_data = client_socket.recv(1024)
                message = f'Menerima {len(received_data)} bytes: {received_data.decode()}'
                print(message)

                response = f'Jumlah karakter: {len(received_data)}'
                client_socket.sendall(response.encode())
                print(f'Kirim Respon: {response}')

                client_socket.close()

            except socket.timeout:
                print('Batas waktu socket telah habis, mohon sambungkan kembali')
                continue
            except Exception as e:
                print(f'Error {e}')
                break
    finally:
        server_socket.close()

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

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

PS D:\Kuliah\Semester 4\Pemrograman Jaringan\Kuliah\Praktikum\
/Semester 4/Pemrograman Jaringan/Kuliah/Praktikum/3/Nomer2.py"
Server listening on ('localhost', 12345)...
Menerima koneksi dari ('127.0.0.1', 62138)
Menerima 9 bytes: Percobaan
Kirim Respon: Jumlah karakter: 9
```

- **Analisa Code**

  Program tersebut adalah implementasi dari server socket menggunakan modul socket. Ini adalah server yang sederhana yang akan menerima koneksi dari klien, mengirimkan pesan ke klien, dan kemudian menutup koneksi. Adapun penjelasan singkat mengenai baris code tersebut kurang lebih sebagai berikut :

  - **import socket** : Mengimpor modul socket yang digunakan untuk berkomunikasi melalui jaringan menggunakan socket.
  - **def main()** : Mendefinisikan fungsi utama main().
  - **server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM):** Membuat objek soket untuk server dengan

menggunakan alamat IPv4 (AF_INET) dan tipe soket stream (SOCK_STREAM).

- **server_socket.settimeout(10)** : Mengatur waktu tunggu (timeout) untuk soket menjadi 10 detik. Ini berarti jika tidak ada koneksi baru dalam waktu 10 detik, maka socket akan melemparkan pengecualian.
- **server_address = ('localhost', 12345**) : Menentukan alamat dan port server. Dalam hal ini, server berjalan di localhost (127.0.0.1) pada port 12345.
- **server_socket.bind(server_address)** : Mengikat (bind) socket server ke alamat dan port yang telah ditentukan sebelumnya.
- **server_socket.listen(2)** : Listen koneksi dari klien. Parameter 2 menentukan bahwa server dapat menerima hingga 2 koneksi klien secara bersamaan.
- **print(f'Server listening on {server_address}...')** : Mencetak pesan bahwa server sedang tersambung pada alamat dan port tertentu.
- **while True:** : Memulai loop tak terbatas untuk menerima koneksi dari klien.
- **client_socket, client_address = server_socket.accept()** : Menerima koneksi baru dari klien dan mendapatkan objek soket klien serta alamat klien.
- **received_data = client_socket.recv(1024)** : Menerima data yang dikirimkan oleh klien dengan maksimal 1024 bytes.
- **message = f'Menerima {len(received_data)} bytes: {received_data.decode()}'** : Membuat pesan yang berisi jumlah byte yang diterima dan isi pesan yang diterima dari klien.
- **client_socket.sendall(response.encode())** : Mengirimkan respons kembali kepada klien. Respons ini berupa jumlah karakter dari pesan yang diterima.
- **client_socket.close()** : Menutup koneksi dengan klien saat selesai berkomunikasi dengan klien tersebut.
- **except socket.timeout** : Menangkap pengecualian jika timeout terjadi.
- **Finally** : Bagian akhir yang akan dieksekusi terlepas dari apakah ada pengecualian atau tidak.
- **server_socket.close()** : Menutup soket server.

➢ **Analisa Hasil**

- Menampilkan pesan "Server listening on localhost:12345..." saat server siap menerima koneksi.

- Menampilkan pesan "Menerima koneksi dari client_address" saat menerima koneksi dari client.
- Menampilkan pesan "Menerima ... bytes: ..." dan "Kirim Respon: ..." setiap kali menerima pesan dari client dan mengirim respons ke client.
- Menampilkan pesan "Batas waktu socket telah habis, mohon sambungkan kembali" jika terjadi timeout.
- Menampilkan pesan kesalahan jika terjadi exception.

3. Membuat sebuah program klien yang dapat terhubung ke server yang telah dibuat pada soal nomor 2. 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 :

```python
# Client
import socket

def main():
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_address = ('localhost', 12345)
    print(f'Connecting to {server_address}...')
    client_socket.connect(server_address)

    try:
        while True:
            message = input('Masukkan Pesan: ')
            client_socket.sendall(message.encode())

            try:
                data = client_socket.recv(1024)
                print(f'Menerima data dari server: {data.decode()}')
            except ConnectionAbortedError:
                print('Koneksi telah habis, mohon sambungkan kembali')
                break

    except KeyboardInterrupt:
        print('Koneksi Ditutup')
        client_socket.close()

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

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  COMMENTS

PS D:\Kuliah\Semester 4\Pemrograman Jaringan\Kuliah\Praktikum\3> & C:/U
/Semester 4/Pemrograman Jaringan/Kuliah/Praktikum/3/Nomer3.py"
Connecting to ('localhost', 12345)...
Masukkan Pesan: Percobaan
Menerima data dari server: Jumlah karakter: 9
```

➢ **Analisa Code**

Program implementasi dari sebuah client sederhana menggunakan modul socket. Client ini digunakan untuk terhubung ke server yang telah diimplementasikan

sebelumnya. Adapun penjelasan singkat mengenai baris code tersebut kurang lebih sebagai berikut

- **import socket** : Mengimpor modul socket yang digunakan untuk berkomunikasi melalui jaringan menggunakan socket.
- **def main()** : Mendefinisikan fungsi utama main().
- **client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)** : Membuat objek soket untuk klien dengan menggunakan alamat IPv4 (AF_INET) dan tipe soket stream (SOCK_STREAM).
- **server_address = ('localhost', 12345)** : Menentukan alamat dan port server yang akan disambung oleh klien. Dalam hal ini, klien akan terhubung ke localhost (127.0.0.1) pada port 12345, sesuai dengan server yang telah diimplementasikan sebelumnya.
- **print(f'Connecting to {server_address}...')** : Mencetak pesan bahwa klien sedang terhubung ke server yang ditentukan.
- **client_socket.connect(server_address)** : Membuat koneksi ke server menggunakan alamat dan port yang telah ditentukan sebelumnya.
- **while True** : Memulai loop tak terbatas untuk mengirim pesan ke server dan menerima respons dari server.
- **message = input('Masukkan Pesan: ')** : Meminta pengguna untuk memasukkan pesan yang akan dikirimkan ke server.
- **client_socket.sendall(message.encode())** : Mengirim pesan yang telah dimasukkan oleh pengguna ke server setelah mengkodeknya menjadi byte.
- **data = client_socket.recv(1024)** : Menerima data/respons dari server dengan maksimal 1024 bytes.
- **print(f'Menerima data dari server: {data.decode()}')** : Mencetak pesan/respons yang diterima dari server setelah didekodekan dari byte menjadi string.
- **except ConnectionAbortedError** : Menangkap pengecualian jika koneksi dengan server terputus.
- **except KeyboardInterrupt** : Menangkap pengecualian jika pengguna menekan tombol keyboard (CTRL+C) untuk menutup koneksi dengan server.
- **client_socket.close()** : Menutup koneksi dengan server setelah selesai berkomunikasi.

- **if __name__ == '__main__'** : Fungsi main() akan dijalankan saat file dieksekusi sebagai skrip utama.

➢ **Analisa Hasil**

- Menampilkan pesan "Connecting to localhost:12345..." saat mencoba terhubung ke server.
- Meminta pengguna untuk memasukkan pesan.
- Menampilkan pesan respons dari server setelah mengirim pesan.
- Menampilkan pesan "Koneksi telah habis, mohon sambungkan kembali" jika koneksi terputus secara tiba-tiba.
- Menampilkan pesan "Koneksi Ditutup" saat program client ditutup dengan KeyboardInterrupt.