

LAPORAN PRAKTIKUM PEMROGRAMAN JARINGAN

“Client – Server (Single Thread)”



Dibuat oleh :

Pandu Rafa Panatagama - 1203220063

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

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)

1.1. server.py

```
import socket

server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
HOST = 'localhost'
PORT = 12345
server_socket.bind((HOST, PORT))
```

Pertama, diimpor module socket yang menyediakan cara bekerja dengan socket. Socket bertipe TCP dibuat dan diikat dengan alamat localhost dan port 12345.

```
server_socket.listen(1)
print("Waiting...")
client_socket, client_address = server_socket.accept()
```

Socket yang telah dibuat menjalankan metode listen dengan backlog sejumlah 1, yang artinya saat ini socket menjadi pendengar pasif dengan 1 jumlah koneksi yang dapat diantri. Selanjutnya, socket menjalankan metode accept untuk membuka port yang telah terikat sehingga socket dapat mulai menerima koneksi dari client sekaligus informasinya disimpan didalam variable client_socket dan client_address.

```
data = client_socket.recv(1024)
angka = int(data.decode())
print("Request dari client: ", angka, " IP Client: ", client_address)
```

Jika client sudah terkoneksi dengan server, maka informasi client sudah tersimpan dalam variable client_socket. Dari informasi tersebut, dijalankan metode recv(1024) untuk mengambil data yang dikirimkan oleh client melalui socket dengan ukuran buffer 1024 byte. Pesan dari client kemudian disimpan dalam variable data, yang nantinya akan diterjemahkan dengan fungsi decode() dan disimpan dalam variable angka. Setelah itu, pesan dan informasi alamat client ditampilkan.

```
if angka % 2 == 0:
    response = "Angka " + str(angka) + " merupakan genap."
else:
    response = "Angka " + str(angka) + " merupakan ganjil."
```

Jika angka habis dibagi 2, maka respon disiapkan sebagai pesan yang menyampaikan bahwa angka tersebut merupakan bilangan genap, dan sebaliknya.

```
client_socket.sendall(response.encode())
client_socket.close()
server_socket.close()
```

Respon yang telah disiapkan, disandikan dengan metode encode() kemudian dikirim ke client. Setelah itu, socket client ditutup dilanjut penutupan socket server untuk mengakhiri koneksi.

1.2. client.py

```
import socket

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

Pertama, diimpor module socket yang menyediakan cara bekerja dengan socket. Socket bertipe TCP dibuat dan dikoneksikan dengan socket yang terikat dengan alamat localhost dan port 12345 (socket server).

```
pesan = input("Masukkan angka: ")
client_socket.sendall(pesan.encode())
```

Input yang diterima (angka) disimpan dalam variable pesan. Kemudian, pesan disandikan lalu dikirim ke socket yang telah terkoneksi (socket server).

```
data = client_socket.recv(1024)
print(data.decode())
```

Menggunakan metode recv(1024), client mengambil pesan yang dikirim (respon) dari server dengan ukuran buffer sebesar 1024 byte. Kemudian pesan diterjemahkan dan ditampilkan.

```
client_socket.close()
```

Akhirnya koneksi ditutup dengan menjalankan metode close() dari socket client.

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

2.1. server2.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("Menunggu koneksi client...")
client_socket, client_address = server_socket.accept()

data = client_socket.recv(1024)
pesan = data.decode()
print("Request dari client: \"", pesan, "\" IP Client: ",
      client_address)
```

```

response = "Jumlah karakter dalam pesan tersebut: " + str(len(pesan))
client_socket.sendall(response.encode())

client_socket.close()
server_socket.close()

```

Screenshot:

```

PS C:\Users\pandu\Documents\KULIAH\ITTS\Semester 4\Dokumen\Pemrograman Jaringan\Minggu 04 - 15 Maret 2024 - Praktikum> &
C:/Users/pandu/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/pandu/Documents/KULIAH/ITTS/Semester 4/Doku
men/Pemrograman Jaringan/Minggu 04 - 15 Maret 2024 - Praktikum/server02.py"
Menunggu koneksi client...
Request dari client: " Naik Ikiwir " IP Client: ('127.0.0.1', 51524)
PS C:\Users\pandu\Documents\KULIAH\ITTS\Semester 4\Dokumen\Pemrograman Jaringan\Minggu 04 - 15 Maret 2024 - Praktikum>

```

Program (socket server) dibuat kurang lebih sama dengan hasil percobaan praktikum sebelumnya, bedanya terletak pada bagaimana pesan yang diterima dari client diolah. Pesan yang diterima berupa baris karakter (string) yang dihitung menggunakan fungsi `len()` yang merupakan fungsi bawaan python yang berguna untuk menghitung jumlah karakter yang dimuat dalam sebuah string.

2.2. client2.py

```

import socket

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

pesan = input("Pesan untuk server: ")
client_socket.sendall(pesan.encode())

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

client_socket.close()

```

Screenshot:

```

PS C:\Users\pandu\Documents\KULIAH\ITTS\Semester 4\Dokumen\Pemrograman Jaringan\Minggu 04 - 15 Maret 2024 - Praktikum> &
C:/Users/pandu/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/pandu/Documents/KULIAH/ITTS/Semester 4/Doku
men/Pemrograman Jaringan/Minggu 04 - 15 Maret 2024 - Praktikum/client2.py"
Pesan untuk server: Naik Ikiwir
Jumlah karakter dalam pesan tersebut: 13
PS C:\Users\pandu\Documents\KULIAH\ITTS\Semester 4\Dokumen\Pemrograman Jaringan\Minggu 04 - 15 Maret 2024 - Praktikum>

```

Program (socket client) dibuat kurang lebih sama dengan hasil percobaan praktikum sebelumnya, bedanya hanya keterangan input yang diminta.

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.

3.1. server.py (modified)

```
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...")

while True:
    client_socket, client_address = server_socket.accept()
    data = client_socket.recv(1024)
    try:
        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."
    except ValueError:
        pesan = data.decode()
        print("Request dari client: \"", pesan, "\" IP Client: ",
client_address)
        response = "Jumlah karakter dalam pesan tersebut: " +
str(len(pesan))

    client_socket.sendall(response.encode())
    client_socket.close()
```

Screenshot:

```
PS C:\Users\pandu\Documents\KULIAH\ITTS\Semester 4\Dokumen\Pemrograman Jaringan\Minggu 04 - 15 Maret 2024 - Praktikum> &
C:/Users/pandu/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/pandu/Documents/KULIAH/ITTS/Semester 4/Doku
men/Pemrograman Jaringan/Minggu 04 - 15 Maret 2024 - Praktikum/server03.py"
Waiting...
Request dari client: " Beautiful Morning " IP Client: ('127.0.0.1', 59509)
```

Melanjutkan kode program socket server hasil percobaan praktikum (server.py) dengan menambahkan blok kode yang mengelola pesan bertipe non-integer dari client, yaitu blok kode yang menghitung jumlah karakter dari pesan yang dikirim client. Pengondisian memanfaatkan try and except. Pertama, server akan mencoba untuk casting pesan yang telah diterjemahkan menjadi nilai integer, dan apabila gagal, maka jalankan blok kode yang disiapkan dalam pengecualian yang akan menghitung jumlah karakter.

3.2. client3.py (Sama saja dengan client2.py)

```
import socket

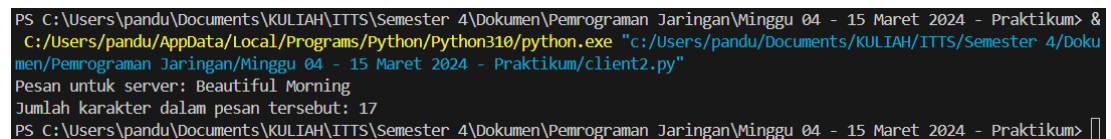
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
HOST = 'localhost'
PORT = 12345
client_socket.connect((HOST, PORT))

pesan = input("Pesan untuk server: ")
client_socket.sendall(pesan.encode())

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

client_socket.close()
```

Screenshot:



```
PS C:\Users\pandu\Documents\KULIAH\ITTS\Semester 4\Dokumen\Pemrograman Jaringan\Minggu 04 - 15 Maret 2024 - Praktikum> &
C:/Users/pandu/AppData/Local/Programs/Python/Python310/python.exe "c:/Users/pandu/Documents/KULIAH/ITTS/Semester 4/Doku
men/Pemrograman Jaringan/Minggu 04 - 15 Maret 2024 - Praktikum/client2.py"
Pesan untuk server: Beautiful Morning
Jumlah karakter dalam pesan tersebut: 17
PS C:\Users\pandu\Documents\KULIAH\ITTS\Semester 4\Dokumen\Pemrograman Jaringan\Minggu 04 - 15 Maret 2024 - Praktikum>
```

Program socket client sama saja dengan program socket client pada soal nomor 2 (client2.py), yaitu program socket client yang mengoneksikan diri ke alamat dan port server, lalu mengirimkan inputan string dari user ke server, hingga kemudian menerima dan menampilkan respon dari server.