# LAPORAN PRAKTIKUM PEMROGRAMAN JARINGAN

## “Client – Server (Single Thread)”


Dibuat oleh :

Syarif Hidayatullah - 1203220094

**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)

Sisi Client

Pemrograman Jaringan > Modul 3 > client3.py > ...

Click here to ask Blackbox to help you code faster

```
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()
```

```
PS C:\Users\LENOVO\OneDrive\Documents\Tugas_Materi Kuliah\S4> python -u "
c:\Users\LENOVO\OneDrive\Documents\Tugas_Materi Kuliah\S4\Pemrograman Jar
ingan\Modul 3\client3.py"
Masukkan angka :24
angka 24 merupakan genap
PS C:\Users\LENOVO\OneDrive\Documents\Tugas_Materi Kuliah\S4>
```

Mengimpor modul socket menggunakan fungsi **'import socket'** untuk berkomunikasi melalui jaringan. Membuat objek socket pada client dengan **'client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)**, parameter 'socket.AF_INET' menunjukkan akan menggunakan alamat IPv4 dan parameter 'socket.SOCK_STREAM' menunjukkan socket akan menggunakan TCP sebagai protokol transport.

**HOST = 'localhost', PORT = 12345** menetapkan alamat host yang akan digunakan untuk terhubung ke server, 'localhost' mengindikasikan bahwa server berjalan pada mesin yang sama dengan client. Menghubungkan client socket ke alamat server yang ditentukan dengan **client_socket.connect((HOST, PORT))**.

**Pesan = input("Masukkan angka :")** mengambil input dari user melalui terminal (input ini berupa angka yang akan dikirim ke server).

**Client_socket.sendall(pesan.encode())** mengirim pesan ke server, 'encode' digunakan untuk mengubah string (pesan) menjadi bytes yang dikirim melalui jaringan.**Data = client_socket.recv(1024)** menerima respon dari server, 1024 adalah ukuran maksimal byte yang diterima dalam satu waktu.

**Print(data.dekode())** menampilkan pesan yang diterima dari server ke user, ‘decode’ digunakan untuk mengubah byte kembali menjadi string.
**Client_socket.close()** menutup koneksi socket.

Sisi Server

```
Click here to ask Blackbox to help you code faster
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()
```

```
PS C:\Users\LENOVO\OneDrive\Documents\Tugas_Materi Kuliah\S4> python -u "c:
\Users\LENOVO\OneDrive\Documents\Tugas_Materi Kuliah\S4\Pemrograman Jaringa
n\Modul 3\server3.py"
Waiting...
Request dari client : 24 IP client : ('127.0.0.1', 50743)
PS C:\Users\LENOVO\OneDrive\Documents\Tugas_Materi Kuliah\S4>
```

Mengimpor modul socket menggunakan fungsi ‘**import socket**’ untuk membuat objek socket dan melakukan operasi jaringan. Membuat objek socket server dengan ‘**server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)** dengan alamat IPv4 dan jenis socket stream.

**HOST = ‘localhost’, PORT = 12345** digunakan untuk menentukan host dan port server.

**Server_socket.bind((HOST, PORT))** mengikat socket server ke alamat dan port tertentu untuk menerima koneksi dari clinet. **Server_socket.listen(1)** mendengarkan koneksi yang masuk dengan parameter 1 yang menunjukkan jumlah koneksi yang diterima sekaligus.

**Print(“waiting...”)** mencetak pesan untuk menunggu koneksi dari client.
**Client_socket, client_address = server_socket.accept()** menerima koneksi dari

client, mengembalikan objek socket baru untuk berkomunikasi dengan client serta alamat IP client.

**Data = client_socket.recv(1024)** menerima data yang dikirim client dengan maksimum 1024 byte.

**Angka = int(data.decode())** mengkonversi data menjadi bilangan bulat.

**Print("request dari client :", angka, "IP client :", client_address)** menampilkan permintaan client berserta alamat IP client.

Memeriksa angka inputan genap atau ganjil menggunakan fungsi **if angka%2 == 0 dan else** dengan **response = "angka"+str(angka)+"merupakan genap /(ganjil)"** menyiapkan pesan respons yang akan dikirim kembali kepada client berdasarkan hasil periksaan sebelumnya.

**Client_socket.sendall(response.encode())** mengirim respons kepada client setelah dikodekan dalam format byte.

Menutup koneksi dengan client menggunakan **client_socket.close()**, dan menutup socket server dengan **server_socker.close()** setelah berkomunikasi.

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 :

```
21  import socket
22
23  server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
24
25  HOST = 'localhost'
26  PORT = 12345
27
28  server_socket.bind((HOST, PORT))
29  server_socket.listen(1)
30
31  print("Waiting...")
32
33  client_socket, client_address = server_socket.accept()
34
35  pesan = client_socket.recv(1024).decode()
36  jumlah_karakter = len(pesan)
37  pesan_balasan = f"Jumlah karakter pesan: {jumlah_karakter}"
38
39  client_socket.sendall(pesan_balasan.encode())
40
41  client_socket.close()
42  server_socket.close()
```

```
PS C:\Users\LENOVO\OneDrive\Documents\Tugas_Materi Kuliah\S4> python -u "c:
\Users\LENOVO\OneDrive\Documents\Tugas_Materi Kuliah\S4\Pemrograman Jaringa
n\Modul 3\server3.py"
Waiting...
PS C:\Users\LENOVO\OneDrive\Documents\Tugas_Materi Kuliah\S4>
```

Membuat server dengan menggunakan modul socket python, server akan mengikat socket ke alamat IP ‘localhost’ dengan port ‘12345’ dan kemudian mendengarkan koneksi. Ketika ada koneksi masuk dari klien, server akan menerima koneksi tersebut dan menerima pesan yang dikirimkan oleh klien. Pesan yang diterima oleh server berupa data dalam format string yang akan di dekodekan untuk mengubahnya menjadi sebuah integer.

Setelah menerima pesan dari klien, server akan menghitung jumlah karakter pesan tersebut dan server akan membuat pesan balasan yang berisi jumlah karakter pesan dan mengirimkannya kembali pada klien. Setelah mengirim pesan balasan server akan menutup koneksi dengan klien dan menutup socket.

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 :

```
12
13  import socket
14
15  client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
16
17  HOST = 'localhost'
18  PORT = 12345
19
20  client_socket.connect((HOST, PORT))
21  pesan = input("Masukkan pesan: ")
22
23  client_socket.sendall(pesan.encode())
24  pesan_balasan = client_socket.recv(1024).decode()
25
26  print("Pesan balasan dari server:", pesan_balasan)
27
28  client_socket.close()
```

```
PS C:\Users\LENOVO\OneDrive\Documents\Tugas_Materi Kuliah\S4> python -u "
c:\Users\LENOVO\OneDrive\Documents\Tugas_Materi Kuliah\S4\Pemrograman Jar
ingan\Modul 3\client3.py"
Masukkan pesan: Nama Syarif H
Pesan balasan dari server: Jumlah karakter pesan: 13
PS C:\Users\LENOVO\OneDrive\Documents\Tugas_Materi Kuliah\S4>
```

Membuat klien objek socket untuk berkomunikasi dengan server menggunakan modul ‘socket’ yang akan terhubung ke server dengan menggunakan alamat ‘localhost’ dan port ‘12345’. Klien meminta input dari user berupa pesan yang akan dikirimkan ke server menggunakan metode ‘sendall()’.

Klien menerima respons dari server dengan metode ‘recv()’, respons yang diterima dari server adalah pesan balasan yang berisi jumlah karakter dari pesan yang dikirimmkan oleh klien. Klien akan menampilkan pesan balasan dari server. Klien menutup koneksi dengan server setelah selesai berkomunikasi.