

Nama: Muhammad Rafi Andeo Praja

NIM: 1301200278

Kelas: IF-44-01

## 1. Life Cycle TCP Socket

- TCP Server
  - (a) Pembuatan socket dengan melakukan import socket, setting ip, port, dan buffer size socket().
  - (b) Melakukan setting socket dengan ip dan port yang sudah dibuat sebelumnya serta melakukan penyetingan ipv4 dan tipe socket TCP bind().
  - (c) Penyetingan listen agar berjalan listen().
  - (d) Server akan menerima pesan dari client accept().
  - (e) Server akan memblokir sampai terdapat sebuah koneksi dengan client.
- TCP Client
  - (f) Pembuatan socket dengan melakukan import socket, setting ip, port, buffer size, dan pesan yang sama dengan server socket().
  - (g) Melakukan setting dengan ip dan port yang sudah dibuat sebelumnya serta melakukan penyetingan ipv4 dan tipe socket TCP dan membuat koneksi dengan server connect().
  - (h) Mengirim pesan yang sudah dibuat ke server write().
- TCP Server
  - (i) Server membaca pesan dari client read().
  - (j) Server mengirim pesan ke client jika pesan dapat diterima write().
- TCP Client
  - (k) Client membaca pesan dari server read().
  - (l) Client dapat menutup socket close().
- TCP Server
  - (m) Server membaca pesan dari client read().
  - (n) Server dapat menutup socket close().

## 2. Life Cycle UDP Socket

- UDP Server
  - (a) Pembuatan socket dengan melakukan import socket, setting ip dan port socket().
  - (b) Melakukan setting socket dengan ip dan port yang sudah dibuat sebelumnya serta melakukan penyetingan ipv4 dan tipe socket UDP bind().
- UDP Client
  - (c) Pembuatan socket dengan melakukan import socket, setting ip, port, dan pesan yang sama dengan server socket().
  - (d) Mengirim pesan ke server sendto().
- UDP Server
  - (e) Server menerima pesan dari client recvfrom().
  - (f) Server dapat mengirim pesan ke client sendto().
- UDP Client
  - (g) Client menerima pesan dari server recvfrom().
  - (h) Client dapat menutup socket close().

## Server:

```
import socket
import select

HEADER_LENGTH = 100

TCP_IP = "127.120.120.1"
TCP_PORT = 1234

server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
server_socket.bind((TCP_IP, TCP_PORT))
server_socket.listen()
sockets_list = [server_socket]
clients = {}

print(f'Koneksi terbentuk dari IP: {TCP_IP} Port: {TCP_PORT}')
def receive_message(client_socket):
    try:
        message_header = client_socket.recv(HEADER_LENGTH)
        if not len(message_header):
            return False
        message_length = int(message_header.decode('utf-8').strip())
        return {'header': message_header, 'data': client_socket.recv(message_length)}
    except:
        return False

while True:
    read_sockets, _, exception_sockets = select.select(sockets_list, [], sockets_list)
    for notified_socket in read_sockets:
        if notified_socket == server_socket:
            client_socket, client_address = server_socket.accept()
            user = receive_message(client_socket)
            if user is False:
                continue
            sockets_list.append(client_socket)
            clients[client_socket] = user
            print('Koneksi diterima dari {}: {}, nama: {}'.format(*client_address, user['data'].decode('utf-8')))
        else:
            message = receive_message(notified_socket)
            if message is False:
                print('Koneksi ditutup dari {}'.format(clients[notified_socket]['data'].decode('utf-8')))
                sockets_list.remove(notified_socket)
                del clients[notified_socket]
                continue
            user = clients[notified_socket]
            print(f'Pesan diterima dari {user["data"].decode("utf-8")}: {message["data"].decode("utf-8")}')
            for client_socket in clients:
                if client_socket != notified_socket:
                    client_socket.send(user['header'] + user['data'] + message['header'] + message['data'])
    for notified_socket in exception_sockets:
        sockets_list.remove(notified_socket)
        del clients[notified_socket]
```

## Output CMD:

```
C:\Windows\System32\cmd.exe - python server.py
Microsoft Windows [Version 10.0.19044.2130]
(c) Microsoft Corporation. All rights reserved.

D:\Python Project>python server.py
Koneksi terbentuk dari IP: 127.120.120.1 Port: 1234
Koneksi diterima dari 127.0.0.1:51333, nama: Dian
Koneksi diterima dari 127.0.0.1:51336, nama: Rafi
Pesan diterima dari Dian: Hello World
Pesan diterima dari Rafi: Hello World
```

Client:

```
import socket

HEADER_LENGTH = 100

TCP_IP = "127.120.120.1"
TCP_PORT = 1234

my_username = input("Masukkan nama anda: ")
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect((TCP_IP, TCP_PORT))
client_socket.setblocking(False)
username = my_username.encode('utf-8')
username_header = f"{len(username):<{HEADER_LENGTH}}".encode('utf-8')
client_socket.send(username_header + username)

while True:
    message = input(f'{my_username} > ')
    if message:
        message = message.encode('utf-8')
        message_header = f"{len(message):<{HEADER_LENGTH}}".encode('utf-8')
        client_socket.send(message_header + message)
```

Output CMD:

```
C:\Windows\System32\cmd.exe - python client.py
Microsoft Windows [Version 10.0.19044.2130]
(c) Microsoft Corporation. All rights reserved.

D:\Python Project>python client.py
Masukkan nama anda: Rafi
Rafi > Hello World
Rafi >
```

```
C:\Windows\System32\cmd.exe - python client.py
Microsoft Windows [Version 10.0.19044.2130]
(c) Microsoft Corporation. All rights reserved.

D:\Python Project>python client.py
Masukkan nama anda: Dian
Dian > Hello World
Dian >
```