ETS JARKOM

Nama: Riza Dwi Andhika

NRP: 05111940000149

1. Penjelasan Singkat

Deskripsi:

Aplikasi sosial media antara user dimana user dapat terhubung dengan user lainnya kedalam suatu group sehingga bisa berkomunikasi bersama-sama di dalamnya. Server menangani koneksi user menggunakan socket

Daftar fitur yang berhasil:

No	Fitur	Status	Keterangan
1	Chat biasa	Berhasil	Berfungsi dengan baik
2	LIST	Berhasil	Berfungsi dengan baik
3	LOG	Berhasil	Berfungsi dengan baik
4	DOWNZIP	Berhasil	Berfungsi dengan baik
5	SEND	Belum Dikerjakan	Masih belum 100% karena waktu tidak cukup

2. Source Code dan Dokumentasi

Aplikasi server:

Ide utama dari *source code server* adalah memilah tipe command yang dikirim dari client lalu melakukan *handling* kepada *handler function* sesuai command yang dikirim. Setiap client yang ingin terkoneksi akan dibuatkan `thread`

```
import os
import socket
import threading
import json
import shutil
from time import sleep

""" Set up socket """
IP_ADDRESS = '127.0.0.1'
PORT = 7000
CHUNK = 4096

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
```

```
server.bind((IP_ADDRESS, PORT))
server.listen(socket.SOMAXCONN)

list_of_clients = []
history_chat = []

""" Handler functions """
def list_handler(client, *args):
    """ COMMAND 'LIST' """

    folder = 'storage'

    onlyfiles = [f for f in os.listdir(folder) if os.path.isfile(os.path.join(folder, f))]
    payload = json.dumps({'type': 'LIST', 'content': onlyfiles})

    client['socket'].send(payload.encode())

def downzip_handler(client, *args):
    """ COMMAND 'DOWNZIP' """

    shutil.make_archive('updated', 'zip', 'storage')
    filesize = os.path.getsize('updated.zip')
    client['socket'].send(json.dumps({'type': 'DOWNZIP', 'size': filesize}).encode())

    sleep(2)

    f = open('updated.zip', 'rb')
    while True:
        l = f.read(CHUNK)
        if not l:
            break
        client['socket'].send(l)

    f.close()
    os.remove('updated.zip')

def send_handler(client, *args):
    # ! Belum selesai
    """ COMMAND 'SEND <file>' """

    print('Recieving file...')
    accumulator = int(0)
    f = open('client.zip', 'wb')

    size = data['size']

    while accumulator < size:
        print('acc', accumulator)
        data = sock.recv(CHUNK)
        accumulator += f.write(data)

    f.close()
    print('Recieved!')

def log_handler(client, *args):
    """ COMMAND LOG """

    client['socket'].send(json.dumps({'type': 'LOG', 'content': history_chat}).encode())

def chat_handler(client, content):
    """ COMMAND CHAT (default command) """

    history_chat.append('{}: {}\n'.format(client['name'], content))
    message = '<{}> {}\n'.format(client['name'], content)

    """ Kirim ke semua client (kecuali dirinya sendiri) """
```

```
    broadcast(client, json.dumps({'type': 'CHAT', 'content': message}))

HANDLE_COMMAND = {
    'LIST': list_handler,
    'DOWNZIP': downzip_handler,
    'SEND': send_handler,
    'LOG': log_handler,
    'CHAT': chat_handler
}

""" Thread untuk client (setiap koneksi client mendapatkan threadnya masing2) """
def clientthread(client):
    while True:
        try:
            """ Terima pesan dari clent """
            message = client['socket'].recv(CHUNK)
            if not message:
                print('\n\nClient disconnected\n\n')
                remove(client)
                continue

            """ Handle request client sesuai dengan command yang diberikan """
            message = message.decode()
            command, content = extract_command(message)
            print(command + '-apa')
            HANDLE_COMMAND[command](client, content)

        except:
            continue


def extract_command(message):
    content = ' '.join(message.split(' ')[1:])

    if message.startswith('LIST'):
        return 'LIST', content
    if message.startswith('DOWNZIP'):
        return 'DOWNZIP', content
    if message.startswith('SEND'):
        return 'SEND', content
    if message.startswith('LOG'):
        return 'LOG', content

    return 'CHAT', message

def broadcast(initiator, message):
    for client in list_of_clients:
        if client['socket'] != initiator['socket']:
            try:
                client['socket'].send(message.encode())
            except:
                client['socket'].close()
                remove(client)

def remove(client):
    if client in list_of_clients:
        client['socket'].close()
        list_of_clients.remove(client)

""" MAIN FUNCTION HERE """
count = 1
while True:
    print('Waiting for connection...')

    """ Simpan informasi koneksi client dalam dictionary """
    conn, addr = server.accept()
    client = {
```

```
        'socket': conn,
        'addr': addr,
        'name': 'Person {}'.format(count)
    }
    list_of_clients.append(client)

    print(addr[0] + ' connected!\n')

    """ Buat thread untuk client """
    threading.Thread(target=clientthread, args=(client,)).start()
    count += 1

conn.close()
```

Aplikasi client:

Ide utama dari *source code server* adalah membuat 2 thread untuk menerima pesan dari server dan mengirim pesan ke server.

```
import socket
import os
import json
import random
from threading import Thread
from time import sleep

""" Set up socket"""
IP_ADDRESS = '127.0.0.1'
PORT = 7000
CHUNK = 4096

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.connect((IP_ADDRESS, PORT))

""" Thread untuk mengirim pesan ke server """
def send_msg(sock):
    while True:
        """ Menerima Input command user """
        data = input('Input >> ')

        """ Jika commandnya itu selain SEND, maka langsung kirim commandnya ke server """
        if not data.startswith('SEND'):
            sock.send(data.encode())
            continue

        """ Jika command adalah SEND, lakukan persiapan untuk streaming file ke server """
        filepath = data.split(' ')[1]
        # check file exist
        if not os.path.isfile(filepath):
            print('File not found!')
            continue

        # send file size
        sock.send(json.dumps({
            'type': 'SEND',
            'size': os.path.getsize(filepath),
            'name': os.path.basename(filepath)
        }).encode())

        sleep(2)

        f = open(filepath, 'rb')
```

```
        while True:
            l = f.read(CHUNK)
            if not l:
                break
            sock.send(l)

        f.close()

""" Thread untuk menerima pesan dari server """
def recv_msg(sock):
    while True:
        """ Terima informasi meta dari server (untuk menentukan tipe pesan apa yang dikirim server) """
        data = sock.recv(CHUNK)
        if not data:
            print('Server closed!')
            sock.close()
            break

        print(data.decode())
        data = json.loads(data.decode())


        """ Tangani response sesuai dengan commandnya """
        if data['type'] == 'CHAT' or data['type'] == 'LIST':
            print(data['content'])
        elif data['type'] == 'LOG':
            print(data['content'])
            f = open('log-{}.txt'.format(random.randint(1, 1000)), 'w')
            f.write(json.dumps(data['content']))
            f.close()
        elif data['type'] == 'DOWNZIP':
            print('Downloading file...')
            accumulator = int(0)
            f = open('client.zip', 'wb')

            size = data['size']

            while accumulator < size:
                print('acc', accumulator)
                data = sock.recv(CHUNK)
                accumulator += f.write(data)

            f.close()
            print('Downloaded!')


""" Menjalankan thread untuk mengirim dan menerima pesan menuju/dari server """
Thread(target=send_msg, args=(server,)).start()
Thread(target=recv_msg, args=(server,)).start()

while True:
    pass
```

3. Teknis Pengoperasian

1. Fitur chat

Lanjutan

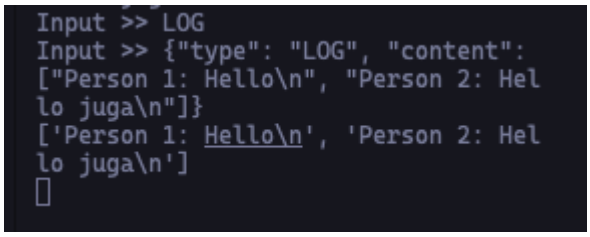
3. Teknis Pengoperasian

1. Chat

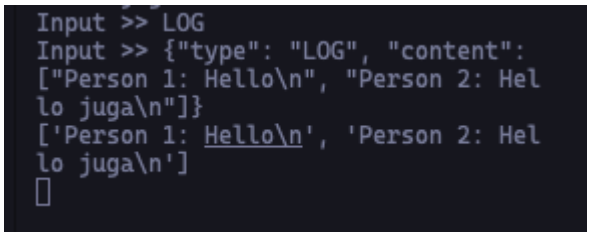
   Cukup mengirim pesan ke server

2. LOG

   Kirim command `LOG`



   Akan menghasilakn file log-{id}.txt berisi

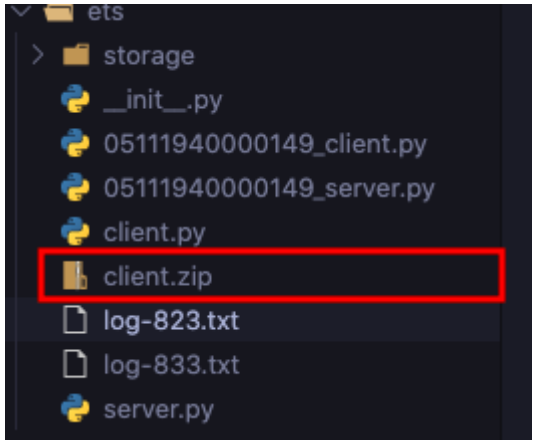


3. DOWNZIP

   Ketik command `DOWNZIP`

```
DOWNZIP
Input >> {"type": "DOWNZIP", "size":
670}
Downloading file...
acc 0
Downloaded!
```

   Akan muncul `client.zip`



4. LIST

   Ketika `LIST` untuk melihat isi file storage server

```
LIST
Input >> {"type": "LIST", "content":
["__init__ copy.py", "c copy.py"]}
['__init__ copy.py', 'c copy.py']
```