

TUGAS KELOMPOK– 4C

PENAMBAHAN FITUR PADA CHAT CLIENT DAN SERVER



Kelas : Pemrograman Jaringan D

Amelia Puji 05111840000147

Evelyn Tjitrodjojo 05111840000099

Muhammmad Daffa' Aflah Syarif 05111840000030

Dosen :

Royyana Muslim Ijtihadie, S.Kom.,M.Kom., Ph.D.

Departemen Teknik Informatika

Fakultas Teknologi Elektro dan Informatika Cerdas (FTEIC)

Institut Teknologi Sepuluh Nopember (ITS)

Surabaya

2021

Tugas Setiap Member Kelompok

Amelia Puji 05111840000147

- Pengiriman file dari user ke user

Evelyn Tjitrodjojo 05111840000099

- Graphical User Interface untuk chat client

Muhammad Daffa Syarif 05111840000030

- Group message

Soal:

KERJAKAN SECARA BERKELOMPOK sesuai pembagian yang sudah dilakukan oleh asisten
Tambahkan kemampuan

- Graphical User Interface untuk chat client
- Pengiriman file dari user ke user
- Group message

Jawab:

PROTOKOL

FILE Client_GUI.py

TUJUAN:

Membentuk tampilan untuk chat client

ATURAN PROTOKOL:

- Server telah dijalankan
- String diinputkan pada entry yang tersedia
- Username dan password harus sudah didaftarkan dalam data

GUI

Tujuan : untuk menampilkan halaman login.

Parameter : tidak ada

Result: akan tampil berbentuk window GUI

```

class GUI:
    def __init__(self, ip_address, port):
        self.users = {}
        self.users['Amel']={'password':'1'}
        self.users['Evelyn']={'password':'2'}
        self.users['Syarif']={'password':'3'}
        self.sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.sock.connect((ip_address, port))

        self.Window = tk.Tk()
        self.Window.withdraw()

        self.login = tk.Toplevel()
        self.login.title("Login")
        self.login.resizable(width=False, height=False)
        self.login.configure(width=400, height=430, bg="#bbf1fa")

        self.keterangan = tk.Label(self.login, text="Made by Evelyn, Amelia, Syarif",
        self.keterangan.place(relheight=0.15, relx=0.30, rely=0.02)

        self.loginText = tk.Label(self.login, text="MOHON LOGIN TERLEBIH DAHULU", jus

```

Code lengkapnya silahkan melihat github.

masuk_fungsi(self, username, room_id=0, pw=0)

Tujuan :

- Untuk mengecek apakah username dan password cocok dengan data
- Untuk mengirimkan username dan password kepada server
- Untuk menampilkan halaman chat dan menutup halaman login
- Untuk menampilkan warning bila terdapat salah username/password

Parameter : username, password, dan room_id yang diinputkan user.

Result

Berhasil : status: OK, menuju halaman chat

Gagal : tetap pada halaman login

```

def masuk_fungsi(self, username, room_id=0, pw=0):
    self.name = username
    self.room = room_id
    if (username in self.users and self.users[username]['password']== pw):
        self.sock.send(str.encode(username))
        time.sleep(0.1)
        self.sock.send(str.encode(room_id))
        self.login.destroy()
        self.chat_GUI()
        terima = threading.Thread(target=self.terima)
        terima.start()
    else:
        self.warn=tk.Label(self.login,bg="#bbf1fa", text="Salah username/pa
        self.warn.place(relx=0.18, rely=0.90)

```

Code lengkapnya silahkan melihat github.

Chat_GUI

Tujuan : untuk menampilkan halaman chat.

Parameter : tidak ada

Result: akan tampil berbentuk window GUI untuk chatroom

```
def chat_GUI(self):
    self.Window.deiconify()
    self.Window.title("ROOM "+self.room)
    self.Window.resizable(width=False, height=False)
    self.Window.configure(width=470, height=550, bg="#17202A")
    self.header = tk.Label(self.Window, bg = "#51c2d5", fg = "#ffffff", text="")
    self.header.place(relwidth = 1)

    self.text = tk.Text(self.Window, width=20, height=2, bg="#c0feff", fg="")
    self.text.place(relheight=0.745, relwidth=1, rely=0.08)

    self.labelBottom = tk.Label(self.Window, bg="#51c2d5", height=80)
    self.labelBottom.place(relwidth = 1, rely = 0.8)

    self.msgInput = tk.Entry(self.labelBottom, bg = "#c0feff", fg = "#000000")
    self.msgInput.place(relwidth = 0.74, relheight = 0.03, rely = 0.008, relytext=0.8)
    self.msgInput.focus()
```

Code lengkapnya silahkan melihat github.

browse_file

Tujuan : untuk mencari file yang akan dikirimkan ke user lain pada chatroom.

Parameter : tidak ada

Result: akan tampil mydocument untuk memilih file mana yang akan dikirim

```
def browse_file(self):
    self.filename = filedialog.askopenfilename(initialdir="/", title="Pilih file", filetypes = ('*.txt',))
    self.fileLocation.configure(text="File Dibuka: " + self.filename)
```

Kirim_file

Tujuan : untuk mengirim file ke server yang kemudian dikirimkan ke user lain pada chatroom.

Parameter : tidak ada

Result:

- Berhasil:
 - status: OK
 - filename: nama file yang dikirim
 - data : isi file yang dikirim
- Gagal:
 - status: ERROR

- data: pesan kesalahan

```
def kirim_file(self):
    self.sock.send("FILE".encode())
    time.sleep(0.1)
    self.sock.send(str("client_" + os.path.basename(self.filename)).encode())
    time.sleep(0.1)
    self.sock.send(str(os.path.getsize(self.filename)).encode())
    time.sleep(0.1)

    file = open(self.filename, "rb")
    data = file.read(1024)
    while data:
        self.sock.send(data)
        data = file.read(1024)
    self.text.config(state=tk.DISABLED)
    self.text.config(state=tk.NORMAL)
    self.text.insert(tk.END, "You: " + str(os.path.basename(self.filename)) + " Dikirim\n\n")
    self.text.config(state=tk.DISABLED)
    self.text.see(tk.END)
```

tombol_kirim

Tujuan : untuk menghapus text yang ada pada box input dan memanggil fungsi kirim message.

Parameter : message

Result:

Berhasil: Berhasil memanggil fungsi kirim message.

Error: Muncul pesan error

```
def tombol_kirim(self, msg):
    self.text.config(state=tk.DISABLED)
    self.msg=msg
    self.msgInput.delete(0, tk.END)
    snd= threading.Thread(target = self.kirim_message)
    snd.start()
```

kirim_message

Tujuan :

- Untuk mengirim pesan ke server yang nantinya akan diteruskan ke client pada ruang chat yang sama.
- Untuk menampilkan pesan yang dikirim pada chat sendiri dengan nama "you".

Parameter : tidak ada

Result:

Berhasil: Pesan terkirim pada server

Error: Muncul pesan error

```
def kirim_message(self):
    self.text.config(state=tk.DISABLED)
    while True:
        self.sock.send(self.msg.encode())
        self.text.config(state = tk.NORMAL)
        self.text.insert(tk.END, "You: " + self.msg + "\n\n")
        self.text.config(state = tk.DISABLED)
        self.text.see(tk.END)
        break
```

terima

Tujuan :

- Untuk menerima file dari server.
- Untuk menulis (write) file pada client.
- Untuk memberitahukan bahwa ada file yang diterima

Parameter : tidak ada

Result:

Berhasil: File diterima

Error: Muncul pesan error dan koneksi pada server terputus

```
def terima(self):
    while True:
        try:
            message = self.sock.recv(1024).decode()

            if str(message) == "FILE":
                nama_file = self.sock.recv(1024).decode()
                len_file = self.sock.recv(1024).decode()
                send_user = self.sock.recv(1024).decode()

                if os.path.exists(nama_file):
                    os.remove(nama_file)

                total = 0
                with open(nama_file, 'wb') as file:
                    while str(total) != len_file:
                        data = self.sock.recv(1024)
                        total = total + len(data)
                        file.write(data)

                self.text.config(state=tk.DISABLED)
                self.text.config(state = tk.NORMAL)
```

Code lengkapnya silahkan melihat pada github.

Screenshot Tampilan GUI:

- Halaman Authentifikasi dan Validasi

Two screenshots of a login window titled "Login". The window has a light blue background and a dark blue header bar with the text "Login".

The left screenshot shows the login form with the following fields and buttons:

- Username:
- Password:
- Id Room:
- MASUK** button

The right screenshot shows the same form with the following values and an error message:

- Username: Amel
- Password: *
- Id Room:
- MASUK** button
- Salah username/password! (red text)

- Halaman Chat dan Kirim File

Two screenshots of a chat window titled "ROOM 1". The window has a light blue background and a dark blue header bar with the text "ROOM 1".

The left screenshot shows the chat interface for user "Amel". The chat area contains the following messages:

- Hai, Amel! Selamat Datang di Grup Baru - Code oleh Evelyn, Amelia, Syarif
- You: test
- Evelyn: hai
- You: test.txt Dikirim

The right screenshot shows the chat interface for user "Evelyn". The chat area contains the following messages:

- Hai, Evelyn! Selamat Datang di Grup 1 - Code oleh Evelyn, Amelia, Syarif
- Amel: test
- You: hai
- Amel: client_test.txt Diterima

Both screenshots show a "Kirim" button at the bottom right of the chat area and a "Browse" button at the bottom left of the chat area.

PROTOKOL

FILE Server.py

TUJUAN

Melayani client dalam request file server

ATURAN PROTOKOL:

- client harus mengirimkan request dalam bentuk string

Buat_koneksi(self, ip_address, ports)

Tujuan: menginisiasi (membangun) koneksi antara server dan client

Parameter: connection, ip_address, port (tergantung laptop masing-masing)

Result:

Berhasil: koneksi terbuat, akan muncul pesan "[IP_Address]: Connected", dan dibuatnya thread

Gagal: socket akan ditutup

```
def buat_koneksi(self, ip_address, port):
    self.ip_address = ip_address
    self.port = port
    self.sock.bind((self.ip_address, int(self.port)))
    self.sock.listen(100)

    while True:
        connection, address = self.sock.accept()
        print(str(address[0]) + ":" + str(address[1]) + " Connected")
        start_new_thread(self.thread_client, (connection,))

    self.sock.close()
```

Thread_client(self, connection)

Tujuan: untuk mengecek apakah

- Group (Room) sudah pernah dibuat sebelumnya
- Message yang diread apakah berupa FILE atau bukan

Parameter: connection

Result:

Berhasil:

- Apabila Room belum ada, maka akan muncul pesan "Hai, {user_id}! Selamat Datang di Grup Baru - Code oleh Evelyn,Amelia,Syarif"
- Apabila Room sudah ada, maka akan muncul pesan "Hai, {user_id}! Selamat Datang di Grup {room_id} - Code oleh Evelyn,Amelia,Syarif"
- Apabila message yang diread pada client1 berupa FILE, maka akan dikirimkan FILE ke client2 atau client lainnya menggunakan fungsi file_broadcast dibawah

- Apabila message yang diread pada client1 bukan FILE, maka akan dikirimkan text / pesan ke client2 atau client lainnya menggunakan fungsi broadcast dibawah

Gagal: koneksi akan diremove

```
def thread_client(self, connection):
    user_id = connection.recv(1024).decode().replace("User ", "")
    room_id = connection.recv(1024).decode().replace("Join ", "")

    if room_id not in self.rooms:
        connection.send(f"Hai, {user_id}! Selamat Datang di Grup Baru - Code oleh Evelyn,Amelia,Syarif".encode())
    else:
        connection.send(f"Hai, {user_id}! Selamat Datang di Grup {room_id} - Code oleh Evelyn,Amelia,Syarif".encode())

    self.rooms[room_id].append(connection)

    while True:
        try:
            message = connection.recv(1024)
            print(str(message.decode()))
            if message:
                if str(message.decode()) == "FILE":
                    self.file_broadcast(connection, room_id, user_id)
                else:
                    message_to_send = str(user_id) + ": " + message.decode()
                    self.broadcast(message_to_send, connection, room_id)
            else:
                self.remove(connection, room_id)
        except Exception as e:
            print(repr(e))
            print("Client disconnected")
            break
```

File_broadcast(self, connection, room_id, user_id)

Tujuan: untuk mengirimkan FILE yang diread oleh client satu ke client lainnya

Parameter: connection, room_id, dan user_id yang diinputkan user

Result:

Berhasil: File yang diread akan terkirim ke client lainnya dengan tambahan pesan “Terkirim”

Gagal: client akan otomatis terclose koneksinya (koneksi akan diremove)

```
def file_broadcast(self, connection, room_id, user_id):
    nama_file = connection.recv(1024).decode()
    len_file = connection.recv(1024).decode()
    for client in self.rooms[room_id]:
        if client != connection:
            try:
                client.send("FILE".encode())
                time.sleep(0.1)
                client.send(nama_file.encode())
                time.sleep(0.1)
                client.send(len_file.encode())
                time.sleep(0.1)
                client.send(user_id.encode())
            except:
                client.close()
                self.remove(client, room_id)

    total = 0
    print(nama_file, len_file)
    while str(total) != len_file:
        data = connection.recv(1024)
        total = total + len(data)
        for client in self.rooms[room_id]:
            if client != connection:
                try:
                    client.send(data)
                    # time.sleep(0.1)
                except:
                    client.close()
                    self.remove(client, room_id)
    print("Terkirim")
```

broadcast(self, message_to_send, connection, room_id)

Tujuan: untuk message / pesan yang diread oleh client satu ke client lainnya

Parameter: message_to_send (format pesan yang akan dikirim), connection, dan room_id yang diinputkan user

Result:

Berhasil: Message yang diread akan terkirim ke client lainnya

Gagal: client akan otomatis terclose koneksinya (koneksi akan diremove)

```
def broadcast(self, message_to_send, connection, room_id):
    for client in self.rooms[room_id]:
        if client != connection:
            try:
                client.send(message_to_send.encode())
            except:
                client.close()
                self.remove(client, room_id)
```

remove(self, connection, room_id)

Tujuan: untuk memutuskan koneksi client di server

Parameter: connection dan room_id yang diinputkan user

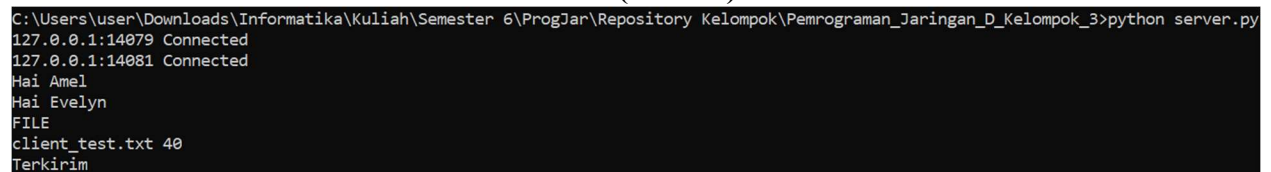
Result:

Berhasil: Koneksi akan terputus dari server

Gagal: -

```
def remove(self, connection, room_id):
    if connection in self.rooms[room_id]:
        self.rooms[room_id].remove(connection)
```

Hasil Screenshot Server di Command Prom (2 Klien)



```
C:\Users\user\Downloads\Informatika\Kuliah\Semester 6\ProgJar\Repository Kelompok\Pemrograman_Jaringan_D_Kelompok_3>python server.py
127.0.0.1:14079 Connected
127.0.0.1:14081 Connected
Hai Amel
Hai Evelyn
FILE
client_test.txt 40
Terkirim
```