Tugas Kelompok 4C

Penambahan Fitur pada Chat Client dan Chat Server

Oleh Kelompok D05:

1)	Rida Adila	05111840000002
2)	Clement Prolifel Priyatama	05111840000013
3)	Irsyadhani Dwi Shubhi	05111840000022

Dosen Pengampu:
Royyana Muslim Ijtihadie, S.Kom.,M.Kom., Ph.D.

S1 TEKNIK INFORMATIKA

FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS

INSTITUT TEKNOLOGI SEPULUH NOPEMBER SURABAYA

2021

Tugas Anggota Kelompok:

Rida Adila 05111840000002

- Pengiriman file dari user ke user

Irsyadhani Dwi Shubhi 05111840000022

- Graphical User Interface untuk chat client

Clement Prolifel Priyatama 05111840000013

- 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

Jawaban:

File:

client_rida.py

client_clement.py

client_irsyad.py

Tujuan:

Membuat tampilan untuk chat client

Aturan:

- Server telah dijalankan

- String diinputkan pada entry yang tersedia

- Username dan password harus sudah didaftarkan dalam data

A. **Data untuk *Sign In***

Tujuan : untuk mendaftarkan data yang sudah disimpan.

Parameter : user, nama, negara, password, incoming

Result: pada saat login akan bisa masuk (*sign in*)

```
9   class ChatApplication():
10      def __init__(self):
11
12          Definitions of __init__                                  a': 'Lionel Messi', 'negara': 'Argentina', 'password': 'surabaya', 'incoming': {},
13          f client_irsyad.py:10  Python                            ing': {}}
14          f testserver.py:6  Python                                ': 'Jordan Henderson', 'negara': 'Inggris', 'password': 'surabaya',
15                                                                   ncoming': {}, 'outgoing': {}}
16          self.users['rida'] = {'nama': 'rida', 'negara': 'Inggris', 'password': 'surabaya', 'incoming': {},
17                                'outgoing': {}}
18
19          self.sign_in()
```

B. Tampilan Login

Tujuan : untuk menampilkan halaman login

Parameter : -

Result: Tampilan halaman Login

```
23      def sign_in(self):
24          self.login_win = Tk()
25          self.login_win.title('Sign In')
26          self.login_win.geometry('500x350')
27          self.login_win.configure(bg="light blue")
28
29          self.login_user = StringVar()
30          self.login_pass = StringVar()
31
32          self.msg1 = tk.Label(self.login_win, text='User Name', relief=GROOVE)
33          self.msg1.place(relx=0.2, rely=0.3, anchor=CENTER)
34
35          self.msg2 = tk.Label(self.login_win, text=' Password ', relief=GROOVE)
36          self.msg2.place(relx=0.2, rely=0.5, anchor=S)
37
38          self.user_name = Entry(self.login_win, textvariable=self.login_user, relief=GROOVE)
39          self.user_name.place(relx=0.6, rely=0.3, anchor=CENTER, width=300)
40
41          self.user_pass = Entry(self.login_win, show="*",
42                                 textvariable=self.login_pass, relief=GROOVE)
43          self.user_pass.place(relx=0.6, rely=0.5, anchor=S, width=300, )
44
45          self.button = tk.Button(self.login_win, text='Sign In', width=20, height=2, command= lambda: self.login(self.login_us
46                                  activebackground="dark grey", activeforeground="red", relief=GROOVE)
47          self.button.place(relx=0.5, rely=0.7, anchor=CENTER)
48
49          self.stop = tk.Button(self.login_win, text='EXIT', width=20, command=self.login_win.destroy,
50                                bg="red", activebackground="red", relief=GROOVE)
51          self.stop.place(relx=0.3, rely=1, anchor=SE)
52
53          self.login_win.mainloop()
```

C. Tampilan berhasil *Login*

Tujuan : untuk menampilkan halaman berhasil *login*

Parameter : -

Result: Tampilan halaman berhasil *login*

```
55 ▼    def successLogin(self):
56          self.success = Toplevel()
57          self.success.geometry("350x250")
58          self.success.configure(bg="light green")
59          self.success.title("Successfull Sign In")
60          self.stop = tk.Button(self.success, text='Success', width=25, height=2, command=self.success.destroy,
61                                bg="green", activebackground="light grey", relief=GROOVE)
62          self.stop.place(relx=0.5, rely=0.5, anchor=CENTER)
63
64          self.login_win.destroy
```

D. Tampilan *password* salah

Tujuan : untuk menampilkan halaman *password* salah

Parameter : -

Result: Tampilan halaman *password* salah

```
66 ▼    def wrongPass(self):
67          self.wrong_pass = Toplevel()
68          self.wrong_pass.geometry("350x250")
69          self.wrong_pass.title("Wrong PassWord")
70          self.wrong_pass.configure(bg="yellow")
71          self.stop = tk.Button(self.wrong_pass, text='Wrong Password', width=25, height=2,
72                                command=self.wrong_pass.destroy, bg="red", activebackground="red", relief=GROOVE)
73          self.stop.place(relx=0.5, rely=0.5, anchor=CENTER)
74
```

E. Tampilan *user* salah

Tujuan : untuk menampilkan *user* salah

Parameter : -

Result: Tampilan halaman *user* salah

```
75 ▼    def userNotFound(self):
76          self.user_not_found = Toplevel()
77          self.user_not_found.geometry("350x250")
78          self.user_not_found.title("User not Found ")
79          self.user_not_found.configure(bg="yellow")
80 ▼       self.stop = tk.Button(self.user_not_found, text='User not Found Kindly register First', width=30,
81                                height=2, command=self.user_not_found.destroy, bg="red", activebackground="red",
82                                relief=GROOVE)
83          self.stop.place(relx=0.5, rely=0.5, anchor=CENTER)
84
```

F. Fungsi *login()* pada *client*

Tujuan : Memeriksa apakah autentikasi sudah benar

Parameter : self, username, password

Result: Apabila autentikasi sesuai dengan list *users,* maka aplikasi akan menjalankan tampilan *successLogin(),* menghubungkan ke server dengan fungsi *connect_to_server()*, menerima pesan sukses dari server dengan fungsi *contact_server(),* dan menampilkan tampilan chat dengan fungsi *chat_screen()*. Apabila password salah, maka akan menampilkan tampilan password salah pada fungsi *wrongPass(),* sedangkan apabila user tidak ditemukan, maka akan menampilkan halaman user tidak ditemukan dengan fungsi *userNotFound()*

```
85 ▼    def login(self, username, password):
86 ▼        if(username in self.users and password==self.users[username]['password']):
87              self.successLogin()
88              self.connect_to_server()
89              self.contact_server()
90              self.chat_screen()
91              # self.chat_screen()
92          elif(username in self.users and password!=self.users[username]['password']):
93              self.wrongPass()
94          else:
95              self.userNotFound()
```

G. Fungsi *connect_to_server()* pada *client*

Tujuan : Melakukan koneksi TCP dengan server pada localhost port 8000

Parameter : self

Result: Apabila koneksi gagal, maka aplikasi akan keluar dan *print* "Server Not Found!"

```
97          def connect_to_server(self):
98              self.s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
99              try:
100                 self.s.connect(("127.0.0.1", 8000))
101             except:
102                 print("Server Not found!")
103                 sys.exit()
```

H. Fungsi *contact_server()* pada *client*

Tujuan : untuk mengirimkan data user ke server.

Parameter : self

Result: Menerima pesan awal dari server yakni welcome_msg

```
105         def contact_server(self):
106             self.s.send(str.encode(self.login_user.get()))
107             self.welcome_msg = self.s.recv(2048).decode()
108             if "ERROR" in self.welcome_msg:
109                 print(self.welcome_msg)
110                 sys.exit()
111             self.is_connected = True
```

I. Fungsi *receive_message_from_server()* pada *client*

Tujuan : untuk menerima pesan saat melakukan chat

Parameter : self

Result: String message yang ditampilkan pada tampilan chatbox

```
113     def recieve_message_from_server(self):
114         while self.is_connected:
115             message = self.s.recv(2048).decode()
116             if self.login_user.get() in message.split(':')[0]:
117                 message = "Me :" + message.split(':')[1]
118             self.chatbox.insert(END, str(message))
```

J. Fungsi *send_messages_to_server()* pada *client*

Tujuan : untuk mengambil pesan yang dituliskan oleh user dan mengirimnya ke server

Parameter : self

Result: String yang dikirimkan ke server dan tulisan pada messagebox yang hilang

```
121     def send_messages_to_server(self):
122         message = self.messagebox.get()
123         self.s.send(str.encode(message))
124         self.messagebox.delete(0, END)
125
```

K. Fungsi *send_file_to_server()* pada *client*

Tujuan : untuk mengirimkan file text kepada semua *client* dalam group chat

Parameter : self

Result: File txt dikirimkan ke server

```
126     def send_file_to_server(self):
127         self.s.send("FILE".encode())
128         self.s.send(str("client_" + os.path.basename(self.filename)).encode())
129         file = open(self.filename, "rb")
130         print("Send : ", self.filename)
131         data = file.read(1024)
132         while data:
133             self.s.send(data)
134             data = file.read(1024)
```

L. Fungsi *browseFile()* pada *client*

Tujuan : untuk menampilkan *window file explorer* yang dapat digunakan untuk memilih file yang akan dirikim

Parameter : self

Result: Tampilan *file explorer*

```
136     def browseFile(self):
137         self.filetypes = (
138             ('text files', '*.txt'),
139             ('All files', '*.*')
140         )
141 
142         self.filename = fd.askopenfilename(
143             title='Open a file',
144             initialdir='/',
145             filetypes=self.filetypes)
146 
147         self.messageFile.configure(text=self.filename)
148 
```

M. Fungsi *Chat Screen*

Tujuan : untuk menampilkan *Chat Screen*

Parameter : -

Result: Tampilan halaman *Chat Screen*

```
149     def chat_screen(self):
150         self.chatroom_size = '750x450+250+100'
151         self.chatroom_title = 'Chatroom ' +str(self.login_user.get())
152         self.backgroungcolor = 'pink'
153         self.chatbox_size = (90, 25)
154         self.messagebox_width = 70
155         self.send_button_background_color = 'cyan'
156         self.send_button_foreground_color = 'black'
157 
158         self.root = Tk()
159         self.root.geometry(self.chatroom_size)
160         self.root.title(self.chatroom_title)
161         self.root.config(bg=self.backgroungcolor)
162 
163         self.chatbox = Listbox(self.root, height=self.chatbox_size[1], width=self.chatbox_size[0])
164         self.chatbox.grid(row=0, column=0, padx=(35, 20), pady=30, columnspan=3)
165 
166         self.messagebox = Entry(self.root, width=self.messagebox_width)
167         self.messagebox.grid(row=1, column=0, columnspan=2, padx=(35, 0))
168         self.messageFile = Label(self.root, width=50, text="Pilih File")
169         self.messageFile.grid(row=3, column=0, columnspan=2, padx=(35, 0))
170 
171         self.send_button = Button(self.root, text='Send',
172                                   bg=self.send_button_background_color,
173                                   fg=self.send_button_foreground_color,
174                                   command=self.send_messages_to_server)
175         self.send_button.grid(row=1, column=2)
176         self.file_button = Button(self.root, text='Browse File',
177                                   bg=self.send_button_background_color,
178                                   fg=self.send_button_foreground_color,
179                                   command=self.browseFile)
180         self.file_button.grid(row=3, column=2)
```

```
181
182         self.fileSubmit_button = Button(self.root, text='Send File',
183                                          bg=self.send_button_background_color,
184                                          fg=self.send_button_foreground_color,
185                                          command= self.send_file_to_server)
186         self.fileSubmit_button.grid(row=4, column=2)
187
188         self.chatbox.insert(END, str(self.welcome_msg))
189
190         listen_for_messages_thread = threading.Thread(target=self.recieve_message_from_server)
191         listen_for_messages_thread.start()
192
193         self.root.mainloop()
194
195
196 if __name__ == "__main__":
197     chat_app = ChatApplication()
```

N. Potongan Kode Fungsi listen_for_messages pada *testserver.py*

Tujuan : sebagai penghubung antar *client*

Parameter : self, client_data

Result: Menunggu client mengirimkan pesan, kemudian mengirimkan pesan tersebut kepada semua orang yang ada di *group chat*

```
14      def listen_for_messages(self, client_data):
15          is_connected = True
16          while is_connected:
17              if len(self.users) > 0:
18                  try:
19                      msg = client_data[1].recv(self.msg_size).decode()
20
21                  except:
22                      is_connected = False
23                      self.users.remove(client_data)
24                  if(str(msg)=="FILE"):
25                      f = open('hasil.txt', 'wb')
26                      while True:
27                          msg = client_data[1].recv(self.msg_size)
28                          while (msg):
29                              print
30                              "Receiving..."
31                              f.write(msg)
32                              msg = client_data[1].recv(self.msg_size)
33                  else:
34                      msg = f"{client_data[0]} : {msg}"
35
36                  for user in self.users:
37                      try:
38                          user[1].send(str.encode(msg))
39                      except:      ridaadila, 2 hours ago • add file
40                          continue
```

O. Potongan Kode Fungsi connection pada *testserver.py*

Tujuan : untuk melakukan *binding* dan *accept* koneksi dari *client*

Parameter : self

Result: Menunggu client mengirimkan pesan, kemudian mengirimkan pesan tersebut kepada semua orang yang ada di *group chat*

```
43    def connection(self):
44        self.s.bind((self.connection_addr[0], self.connection_addr[1]))
45        self.s.listen(5)
46        print("Listening to connections....")
47        while self.connection_status:
48            conn, addr = self.s.accept()
49            new_user = conn.recv(self.msg_size).decode()
50            user_taken = False
51            for user in self.users:
52                if user[0] == new_user:
53                    conn.send(str.encode("ERROR : USER NAME TAKEN!"))
54                    user_taken = True
55
56            if user_taken:
57                continue
58
59            for user in self.users:
60                user[1].send(str.encode(f"{new_user} has entered the chat"))
61
62            self.users.append((new_user, conn))
63          ridaadila, 2 hours ago • add file
64            new_user_thread = threading.Thread(target=self.listen_for_messages, args=((new_user, conn),))
65            new_user_thread.start()
66
67            print(f"Connection established with {new_user} at {addr}")
68            conn.send(str.encode(self.connection_msg))
```

P. Potongan Kode main pada *testserver.py*

Tujuan : untuk menjalankan server

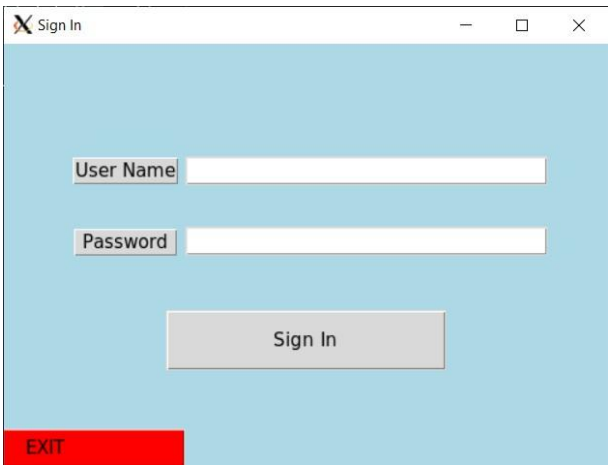
Parameter : -

Result: Membuat koneksi pada localhost dengan port 8000

```
71  if __name__ == "__main__":
72      server_connection = ServerForChatroom(('127.0.0.1', 8000))
73      server_connection.connection()        ridaadila, 2 hours ago •
```

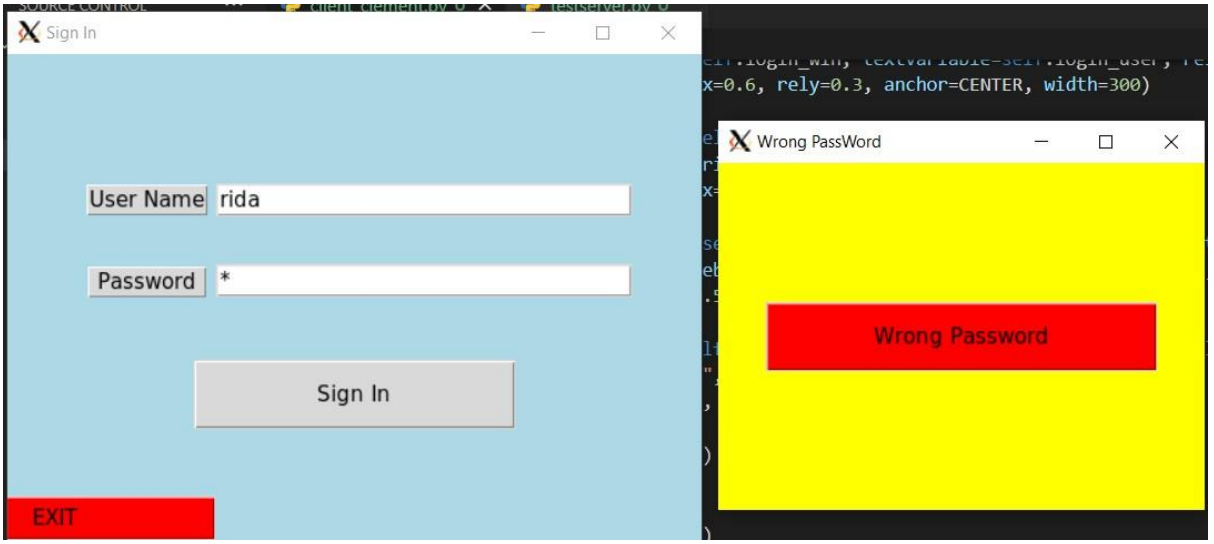
Hasil Tampilan Screenshot GUI *Chat Application*:

- Halaman *Sign In*

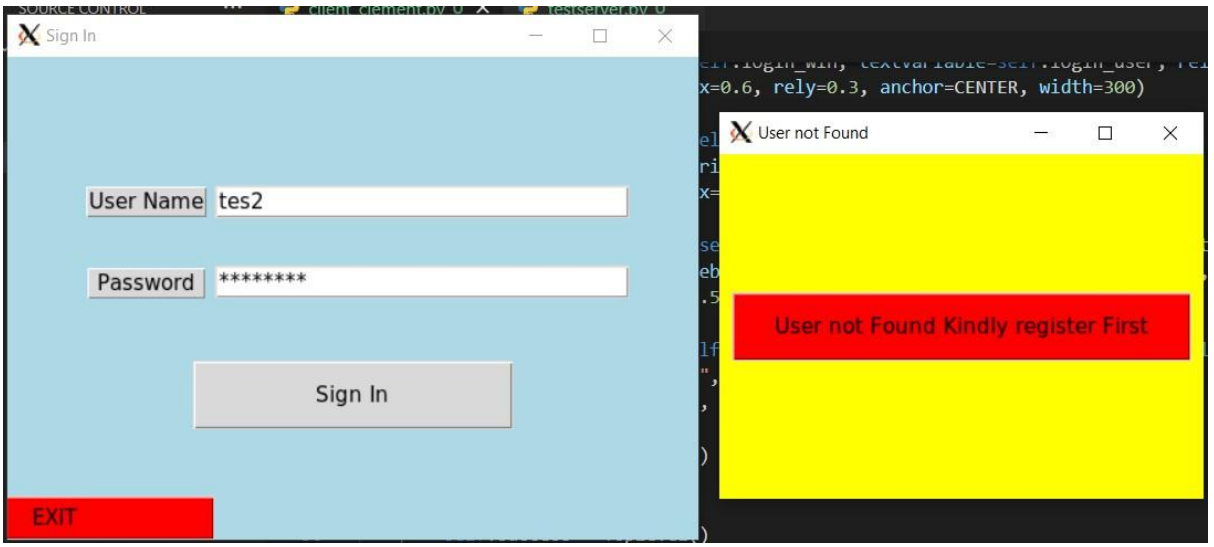


- Halaman berhasil *login*

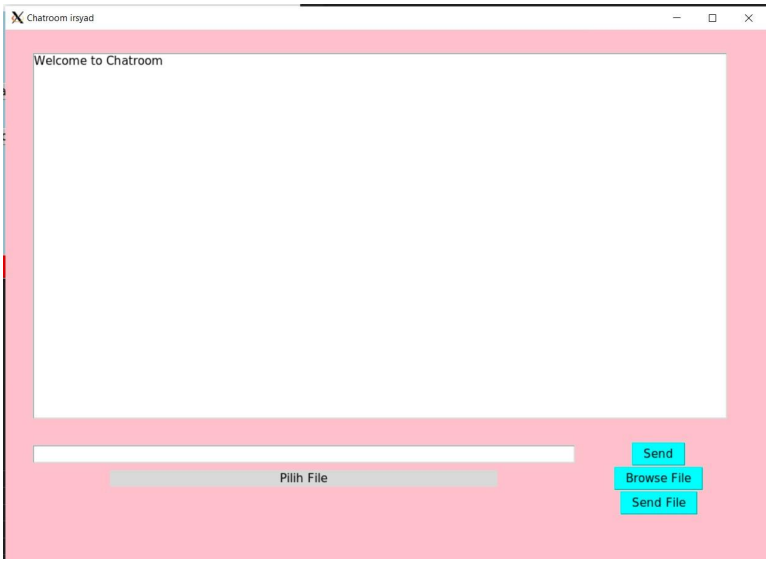
- Halaman salah *password*

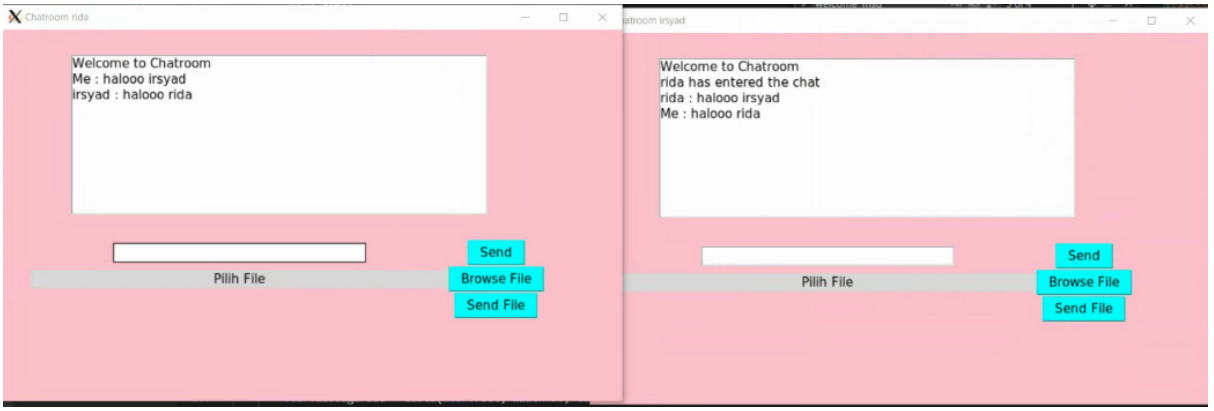


- Halaman *user* tidak ditemukan

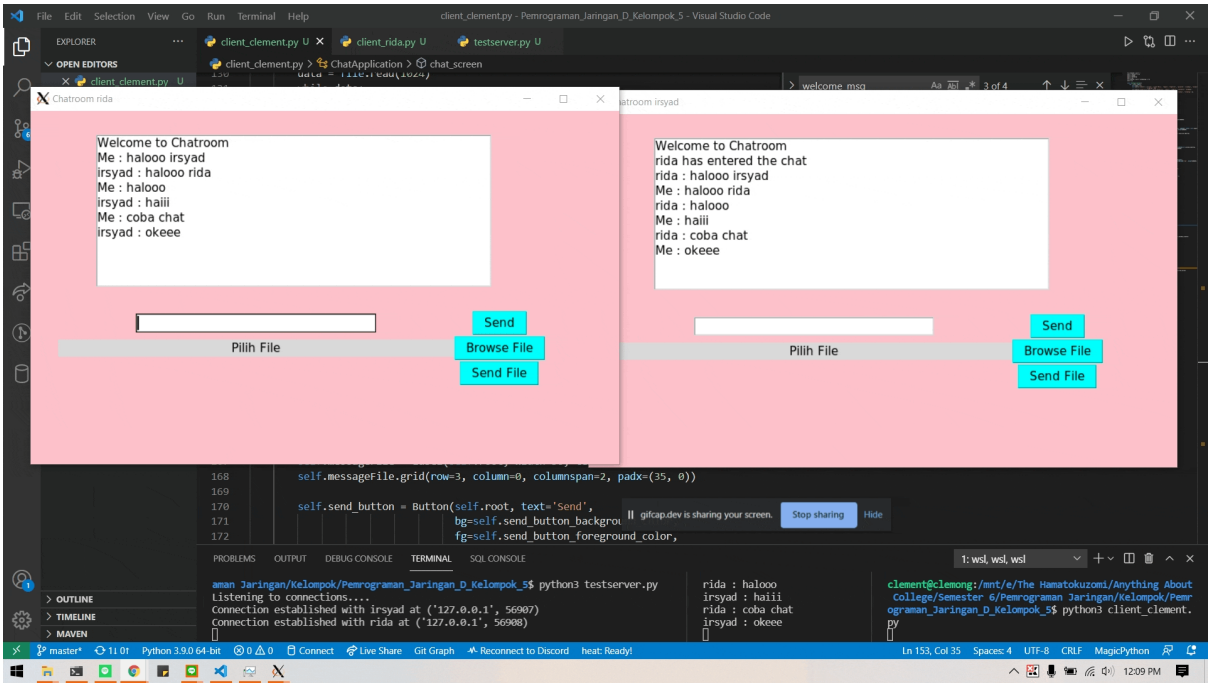


- Halaman *chat*





- Chat



LINK REPOSITORY KELOMPOK :
https://github.com/prolifel/Pemrograman_Jaringan_D_Kelompok_5.git

LINK REPOSITORY INDIVIDU :

1) Rida Adila 05111840000002 :
   https://github.com/ridaadila/Pemrograman_Jaringan_D/tree/master/progjar4c/Tugas%20Kelompok%20Chat%20Server%204C
2) Clement Prolifel P. 05111840000013 :
   https://github.com/prolifel/Pemrograman_Jaringan_D/tree/master/progjar4c/Tugas%20Chat%20Server%204C
3) Irsyadhani Dwi S. 05111840000022 :

   https://github.com/irsyadhani/Pemrograman_Jaringan_D/tree/master/progjar4c/Tugas%20Kelompok%20Chat%20Server%204C