**PROJECT 02**

**NETWORK PROGRAMMING COURSE**



**CHAT SERVER**

**Group 06**

Dwi Irsalina NRP. 5115100004

Neny Lukitasari NRP. 5115100018

Hafara Firdausi NRP. 5115100043

Muflich Kamil NRP. 5115100093

Nurotuz Zakiya NRP. 5115100095

Iqbal Mabruri NRP. 5115100120

Vincentius NRP. 5115100159

**Lecturer :**

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

**DEPARTMENT OF INFORMATICS, FACULTY OF INFORMATION TECHNOLOGY**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**SURABAYA 2017**

**DAFTAR ISI**

[1. Pengertian 4](#_Toc501108952)

[2. Implementasi 4](#_Toc501108953)

[a) Server 4](#_Toc501108954)

[b) Client 12](#_Toc501108955)

[3. Fitur 20](#_Toc501108956)

[2.1.1 Membuat Akun 20](#_Toc501108957)

[- Deskripsi Fitur : 20](#_Toc501108958)

[- Error Handling 21](#_Toc501108959)

[2.1.2 Masuk Akun 21](#_Toc501108960)

[- Deskripsi Fitur 21](#_Toc501108961)

[- Error Handling 22](#_Toc501108962)

[- Sub Menu 22](#_Toc501108963)

[2.1.2.1. Kirim Pesan Pribadi 22](#_Toc501108964)

[- Deskripsi fitur 22](#_Toc501108965)

[2.1.2.2. Kirim Pesan Broadcast 24](#_Toc501108966)

[- Deskripsi Fitur: 24](#_Toc501108967)

[2.1.2.3. Group Chat 26](#_Toc501108968)

[- Deskripsi Fitur 26](#_Toc501108969)

[2.1.2.4. Keluar Akun 27](#_Toc501108970)

[- Deskripsi Fitur 27](#_Toc501108971)

[2.1.2.5. Bantuan 27](#_Toc501108972)

[- Deskripsi Fitur 27](#_Toc501108973)

**DAFTAR GAMBAR**

[Gambar 3.1 Tampilan Awal Server 20](#_Toc501108974)

[Gambar 3.2 Tampilan Server bila ada Client Connect 20](#_Toc501108975)

[Gambar 3.3 Tampilan Client Saat Membuat Akun 21](#_Toc501108976)

[Gambar 3.4 Respon Server Saat Client Membuat Akun 21](#_Toc501108977)

[Gambar 3.5 Membuat Akun Cancel 21](#_Toc501108978)

[Gambar 3.6 Tampilan Client Username Telah Terpakai 21](#_Toc501108979)

[Gambar 3.7 Tampilan Client Masuk Akun 21](#_Toc501108980)

[Gambar 3.8 Respon Server Saat Client Masuk Akun 21](#_Toc501108981)

[Gambar 3.9 Server Dapat Menangani Lebih dari 1 Client 22](#_Toc501108982)

[Gambar 3.10 Cancel Masuk Akun 22](#_Toc501108983)

[Gambar 3.11 User Salah Memasukkan Password 22](#_Toc501108984)

[Gambar 3.12 Client 1 Mengirim Pesan ke Client 3 23](#_Toc501108985)

[Gambar 3.13 Client 3 Menerima Pesan dan Membalas ke Client 1 23](#_Toc501108986)

[Gambar 3.14 Client 1 Menerima Balasan dari Client 3 23](#_Toc501108987)

[Gambar 3.15 Perbandingan 3 Client (Client 2 Tidak Menerima Pesan) 24](#_Toc501108988)

[Gambar 3.16 Private Message Cancel 24](#_Toc501108989)

[Gambar 3.17 Tampilan Client Saat Client 2 Mengirim Pesan Broadcast 25](#_Toc501108990)

[Gambar 3.18 Respon Server Saat Mengirim Pesan Broadcast 25](#_Toc501108991)

[Gambar 3.19 Broadcast Message Cancel 25](#_Toc501108992)

[Gambar 3.20 Client 1 Membuat Group Chat 26](#_Toc501108993)

[Gambar 3.21 Client 1 Mengirim Pesan di Grup 26](#_Toc501108994)

[Gambar 3.22 Group Message Cancel 27](#_Toc501108995)

[Gambar 3.23 Tampilan Keluar Akun 27](#_Toc501108996)

# Pengertian

**Server** adalah sebuah sistem komputer yang menyediakan jenis layanan (service) tertentu dalam sebuah jaringan komputer, klien mungkin berbagi data, informasi, hardware dan software.

**Chatting** adalah suatu pesan instant ataupun *instant messaging* di sebuah teknologi jaringan komputer yang mengijinkan pemakainya untuk mengirimkan pesan ke pengguna lain yang tersambung dalam sebuah jaringan computer.

Maka dari itu dapat disimpulkan bahwa ***Chat Server*** merupakan server yang digunakan menangani chatting beberapa user.

# Implementasi

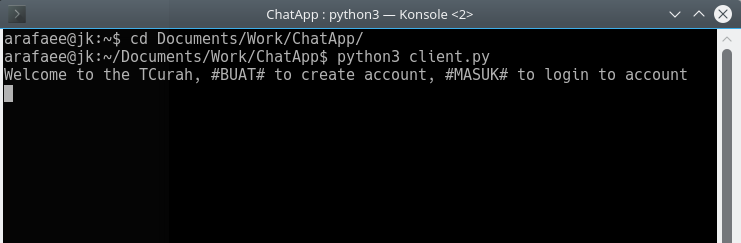
## Server

|  |
| --- |
|  |
|  | import socket  from threading import \* |
|  |  |
|  | #deklarasi server socket |
|  | server = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) |
|  |  |
|  | #Mengatur alamat ip dan port server |
|  | server\_address = ('localhost', 14000) |
|  | print ('starting up on %s port %s' % server\_address) |
|  | server.bind(server\_address) |
|  | server.listen(10) |
|  |  |
|  | list\_of\_clients=[] #socket object client yang terkoneksi ke server |
|  | user\_info=[] #informasi username dan password tiap pengguna |
|  | user\_ip = {} #socket object tiap pengguna |
|  | online\_users = [] |
|  | grup\_list = [] #daftar seluruh grup yang ada di server |
|  | grup\_member = {} #daftar anggota setiap grup |
|  |  |
|  | #fungsi pengecek apakah client disconnect |
|  | def connection\_check(message): |
|  | if not message.decode("utf-8"): |
|  | remove(client\_socket) |
|  | print("Connection to client lost") |
|  | return True |
|  |  |
|  | #fungsi yang melayani tiap client |
|  | def serve\_client(client\_socket, address): |
|  | client\_socket.sendall(b"Welcome to the TCurah, #BUAT# to create account, #MASUK# to login to account") |
|  | while True: |
|  | #menerima perintah dari client (buat akun atau login) |
|  | try: |
|  | message = client\_socket.recv(2048) |
|  | if connection\_check(message) : |
|  | break |
|  | print(message) |
|  | if message.decode('utf-8').startswith("#BUAT#"): |
|  | buat\_akun(client\_socket) |
|  | elif message.decode('utf-8').startswith("#MASUK#"): |
|  | masuk(client\_socket) |
|  | else : |
|  | client\_socket.sendall(b"Invalid command, try again") |
|  | except: |
|  |  |
|  | continue |
|  | finally: |
|  | message = '' |
|  |  |
|  | def buat\_akun(client\_socket) : |
|  | while True: |
|  | #menerima username dan password baru dengan delimiter spasi |
|  | client\_socket.sendall(b"Enter username (space) password, or #EXIT# to cancel") |
|  | message = client\_socket.recv(512) |
|  |  |
|  | if connection\_check(message): |
|  | return |
|  | elif message.decode("utf-8").startswith("#EXIT#"): |
|  | return |
|  | try: |
|  | new\_username, new\_password = message.decode('utf-8').split(" ") |
|  | except : |
|  | client\_socket.sendall(b"Invalid username or password\n") |
|  | continue |
|  | #Mengecek apakah username sudah terpakai |
|  | if any(new\_username in user[0] for user in user\_info) : |
|  | client\_socket.sendall(b"Username already exist\n") |
|  | continue |
|  | else: |
|  | break |
|  | #memasukkan username dan password baru ke list |
|  | user\_info.append([new\_username, new\_password]) |
|  | message\_to\_client = "Account " + user\_info[-1][0] + " created" |
|  | client\_socket.sendall(message\_to\_client.encode("utf-8")) |
|  | return |
|  |  |
|  | def masuk(client\_socket): |
|  |  |
|  | while True: |
|  | #menerima username dan password akun yang akan login dengan delimiter spasi |
|  | client\_socket.sendall(b"Enter username (space) password, or #EXIT# to cancel") |
|  | message = client\_socket.recv(512) |
|  | if connection\_check(message): |
|  | return |
|  | elif message.decode("utf-8").startswith("#EXIT#"): |
|  | return |
|  | try: |
|  | username, password = message.decode('utf-8').split(" ") |
|  | except: |
|  | client\_socket.sendall(b"Invalid username or password\n") |
|  | continue |
|  |  |
|  | #jika username dan password benar, simpan socket object tiap user |
|  | if [username, password] in user\_info: |
|  | user\_ip[username] = client\_socket |
|  | online\_users.append(username) |
|  | welcome\_message = "Welcome, " + username |
|  | client\_socket.sendall(welcome\_message.encode("utf-8")) |
|  | break |
|  | else: |
|  | client\_socket.sendall(b"Username or password incorrect\n") |
|  | continue |
|  |  |
|  | #menu utama setelah login |
|  | while True : |
|  | client\_socket.sendall(b"\n#PRIBADI# to send personal message, #GRUP# to send group message or manage groups, " |
|  | b"\n#BROADCAST# to send message to all online users, #KELUAR# to exit") |
|  | message = client\_socket.recv(512) |
|  | if connection\_check(message): |
|  | return |
|  | if message.decode('utf-8').startswith("#PRIBADI#"): #untuk mengirim pesan pribadi |
|  | pribadi(client\_socket, username) |
|  | elif message.decode('utf-8').startswith("#GRUP#"): #untuk mengelola dan mengirim pesan ke grup |
|  | grup(client\_socket, username) |
|  | elif message.decode('utf-8').startswith("#BROADCAST#"): #mengirim pesan ke semua pengguna yang online |
|  | broadcast(client\_socket, username) |
|  | elif message.decode("utf-8").startswith("#KELUAR#") : |
|  | remove(client\_socket) |
|  | client\_socket.close() |
|  | client\_process.stop() |
|  | return |
|  | else: |
|  | client\_socket.sendall(b"Invalid command") |
|  | continue |
|  |  |
|  |
|  |
|  | def pribadi(client\_socket, username) : |
|  | while True: |
|  | client\_socket.sendall(b"Enter recipient username") |
|  | for user in online\_users: |
|  | if user != username: |
|  | user = "\n" + user |
|  | user = user.encode("utf-8") |
|  | client\_socket.sendall(user) |
|  | message = client\_socket.recv(512) |
|  | if connection\_check(message): |
|  | return |
|  | recipient = message.decode("utf-8") |
|  | if recipient in user\_ip : |
|  | sending\_message = "Sending message to " + recipient + " #EXIT# to exit" "\nenter message:" |
|  | client\_socket.sendall(sending\_message.encode("utf-8")) |
|  | break |
|  | elif message.decode("utf-8").startswith("#EXIT#"): |
|  | return |
|  | else: |
|  | client\_socket.sendall(b"Invalid recipient, try again\n") |
|  | continue |
|  |  |
|  | while True : |
|  | message = client\_socket.recv(2048) |
|  | if connection\_check(message): |
|  | return |
|  | elif message.decode("utf-8").startswith("#EXIT#"): |
|  | break |
|  | sent\_message = "Sent message : " + message.decode("utf-8") |
|  | client\_socket.sendall(sent\_message.encode("utf-8")) |
|  | message = username + "#>" + message.decode("utf-8") |
|  |  |
|  | recipient = user\_ip[recipient] |
|  | send\_message(message, recipient) |
|  |
|  |
|  |  |
|  | def grup(client\_socket, username) : |
|  |  |
|  | while True : |
|  | client\_socket.sendall(b"#BUATGRUP# to create new group, #CHATGRUP# to send group message," |
|  | b"\n#TAMBAHMEMBER# to add member to group, #EXIT# to return to main menu") |
|  | #menerima perintah dari client (buat grup atau chat di grup) |
|  | message = client\_socket.recv(512) |
|  |  |
|  | if connection\_check(message): |
|  | return |
|  |  |
|  | if message.decode('utf-8').startswith("#BUATGRUP#"): |
|  | client\_socket.sendall(b"Enter group name") |
|  | #create grup & nama grup |
|  | message = client\_socket.recv(512) |
|  | if connection\_check(message): |
|  | return |
|  | grup\_name = message.decode("utf-8") |
|  | grup\_list.append(grup\_name) |
|  | grup\_member[grup\_name] = [] |
|  | grup\_member[grup\_name].append(username) |
|  |  |
|  | client\_socket.sendall(b"Choose member (one per line), #END# to finish adding members\n") |
|  | #mencetak nama member yang bisa dimasukkan ke grup |
|  | for user in user\_info: |
|  | if user[0] != username: |
|  | user\_candidate = user[0] + "\n" |
|  | client\_socket.sendall(user\_candidate.encode("utf-8")) |
|  | while True: |
|  | #memasukkan member di dalam grup |
|  | message = client\_socket.recv(512) |
|  | if connection\_check(message): |
|  | return |
|  | member\_name = message.decode('utf-8') |
|  |  |
|  | if any(member\_name in user[0] for user in user\_info) and member\_name not in grup\_member[grup\_name]: |
|  | grup\_member[grup\_name].append(member\_name) |
|  | successful\_add = grup\_member[grup\_name][-1] + " successfully added" |
|  | client\_socket.sendall(successful\_add.encode("utf-8")) |
|  | continue |
|  |  |
|  | elif member\_name in grup\_member[grup\_name]: |
|  | client\_socket.sendall(b"Member already in group") |
|  | continue |
|  |  |
|  | elif message.decode('utf-8').startswith("#END#"): |
|  | break |
|  |  |
|  | else: |
|  | client\_socket.sendall(b"Invalid member or command") |
|  | continue |
|  |  |
|  | #mencetak member dari grup yang sudah dibuat |
|  | created\_group = "Group " + grup\_list[-1] + " created, with members :\n" |
|  | client\_socket.sendall(created\_group.encode("utf-8")) |
|  | for member in grup\_member[grup\_name]: |
|  | member\_name = member + "\n" |
|  | client\_socket.sendall(member\_name.encode("utf-8")) |
|  |  |
|  |  |
|  | elif message.decode("utf-8").startswith("#TAMBAHMEMBER#"): |
|  | client\_socket.sendall(b"Choose group") |
|  | my\_group = [] |
|  | # mencetak grup yang sudah tergabung |
|  | for grup2 in grup\_list: |
|  | if username in grup\_member[grup2]: |
|  | my\_group.append(grup2) |
|  | grup2 = "\n" + grup2 |
|  | client\_socket.sendall(grup2.encode("utf-8")) |
|  |  |
|  | message = client\_socket.recv(512) |
|  | if connection\_check(message): |
|  | return |
|  | message = message.decode("utf-8") |
|  |  |
|  | if message in my\_group: |
|  | selected\_grup = message |
|  | selected\_grup\_message = "Add member to group " + message + " #END# to finish or cancel\n" |
|  | client\_socket.sendall(selected\_grup\_message.encode("utf-8")) |
|  | for user in user\_info: |
|  | if user[0] != username and user[0] not in grup\_member[selected\_grup]: |
|  | user\_candidate = user[0] + "\n" |
|  | client\_socket.sendall(user\_candidate.encode("utf-8")) |
|  |  |
|  | while True: |
|  | #memasukkan member di dalam grup |
|  | message = client\_socket.recv(512) |
|  | if connection\_check(message): |
|  | return |
|  | member\_name = message.decode('utf-8') |
|  |  |
|  | if any(member\_name in user[0] for user in user\_info) and member\_name not in grup\_member[selected\_grup]: |
|  | grup\_member[selected\_grup].append(member\_name) |
|  | successful\_add = grup\_member[selected\_grup][-1] + " successfully added" |
|  | client\_socket.sendall(successful\_add.encode("utf-8")) |
|  | continue |
|  |  |
|  | elif member\_name in grup\_member[selected\_grup]: |
|  | client\_socket.sendall(b"Member already in group") |
|  | continue |
|  |  |
|  | elif message.decode('utf-8').startswith("#END#"): |
|  | break |
|  |  |
|  | else: |
|  | client\_socket.sendall(b"Invalid member or command") |
|  | continue |
|  |  |
|  | created\_group = "Group " + grup\_list[-1] + " updated, with members :" |
|  | client\_socket.sendall(created\_group.encode("utf-8")) |
|  | for member in grup\_member[selected\_grup]: |
|  | member\_name = "\n" + member |
|  | client\_socket.sendall(member\_name.encode("utf-8")) |
|  |  |
|  | elif message.decode("utf-8").startswith("#CHATGRUP#"): |
|  | #fungsi chat grup |
|  | client\_socket.sendall(b"Choose group recipient, #EXIT# to cancel") |
|  | my\_group = [] |
|  | #mencetak grup yang sudah tergabung |
|  | for grup2 in grup\_list: |
|  | if username in grup\_member[grup2] : |
|  | my\_group.append(grup2) |
|  | grup2 = "\n" + grup2 |
|  | client\_socket.sendall(grup2.encode("utf-8")) |
|  |  |
|  | while True : |
|  | #memilih salah satu grup yang tergabung untuk dikirimi pesan |
|  | message = client\_socket.recv(512) |
|  | if connection\_check(message): |
|  | return |
|  | message = message.decode("utf-8") |
|  |  |
|  | if message in my\_group: |
|  | selected\_grup = message |
|  | selected\_grup\_message = "Sending message to " + message + " #EXIT# to exit\nenter message :" |
|  | client\_socket.sendall(selected\_grup\_message.encode("utf-8")) |
|  | while True: |
|  | #menerima pesan untuk dikirimkan ke client lain dalam grup |
|  | message = client\_socket.recv(2048) |
|  | if connection\_check(message): |
|  | return |
|  | message = message.decode("utf-8") |
|  | if message.startswith("#EXIT#"): |
|  | return |
|  | message = username+" ("+selected\_grup+") " +"#>"+message |
|  | send\_list = [] |
|  | #membuat daftar user yang akan dikirimi dalam grup |
|  | for recipients in grup\_member[selected\_grup] : |
|  | if recipients in user\_ip and recipients != username: |
|  | send\_list.append(user\_ip[recipients]) |
|  | #mengirim pesan ke seluruh user yang ada di daftar kirim |
|  | for recipient in send\_list: |
|  | send\_message(message, recipient) |
|  |  |
|  |  |
|  | elif message.startswith("#EXIT#"): |
|  | break |
|  | else: |
|  | client\_socket.sendall(b"Invalid group name") |
|  | continue |
|  |  |
|  | elif message.decode("utf-8").startswith("#EXIT#"): |
|  | return |
|  |  |
|  | else : |
|  | client\_socket.sendall(b"Invalid Command\n") |
|  | continue |
|  |  |
|  |  |
|  | def send\_message(message, recipient): |
|  | try : |
|  | recipient.sendall(message.encode("utf-8")) |
|  | except : |
|  | recipient.close() |
|  | remove(recipient) |
|  |  |
|  |  |
|  | def broadcast(client\_socket, username): |
|  | client\_socket.sendall(b"Enter broadcast message, #EXIT# to cancel") |
|  | message = client\_socket.recv(2048) |
|  | if connection\_check(message): |
|  | return |
|  | message = message.decode("utf-8") |
|  | if message.startswith("#EXIT#") : |
|  | return |
|  | message = username +"(broadcast)" + "#>" + message |
|  | print(message) |
|  | for clients in list\_of\_clients: |
|  | if clients != client\_socket: |
|  | try: |
|  | clients.send(message.encode("utf-8")) |
|  | except: |
|  | clients.close() |
|  | remove(clients) |
|  |
|  |
|  | def remove(connection): |
|  | if connection in list\_of\_clients: |
|  | list\_of\_clients.remove(connection) |
|  | for clients in user\_info: |
|  | if user\_ip[clients[0]] == connection: |
|  | del user\_ip[clients[0]] |
|  | if clients[0] in online\_users: |
|  | online\_users.remove(clients[0]) |
|  |  |
|  |  |
|  | #jika ada client yang connect |
|  | client\_socket, address = server.accept() |
|  |  |
|  | list\_of\_clients.append(client\_socket) |
|  | print (address[0] + " connected") |
|  |  |
|  | client\_process = Thread(target = serve\_client,args = (client\_socket,address)) |
|  | client\_process.start() |
|  |  |
|  |  |
|  | client\_socket.close() |
|  | server.close() |

## Client

|  |
| --- |
|  |
|  | import socket  import select |
|  | import sys |
|  | from threading import \* |
|  |
|  |
|  | client = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) |
|  | server\_address = ("localhost", 14000) |
|  | client.connect(server\_address) |
|  |
|  |
|  | thread\_stopped = 0 |
|  |
|  | def receive(client): |
|  | while True: |
|  | message = client.recv(2048) |
|  | if not message: |
|  | print("Disconnected from server\n") |
|  | global thread\_stopped |
|  | thread\_stopped = 1 |
|  | Thread.stop() |
|  | message = message.decode("utf-8") |
|  | try: |
|  | sender, message = message.split("#>") |
|  | print("<"+sender+"> " + message + "\n") |
|  | except: |
|  | print(message) |
|  |  |
|  | Thread(target=receive,args=(client,)).start() |
|  |  |
|  | while True: |
|  | usercmd = input("") |
|  | if thread\_stopped == 1 or usercmd == "#KELUAR#": |
|  | Thread.stop() |
|  | break |
|  | client.sendall(usercmd.encode('utf-8')) |
|  | usercmd = "" |
|  |  |
|  |  |
|  | client.close() |

# Fitur



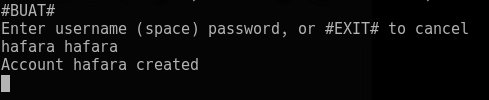
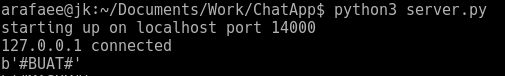
Gambar . Tampilan Awal *Server*



Gambar . Tampilan *Server* bila ada *Client* Connect

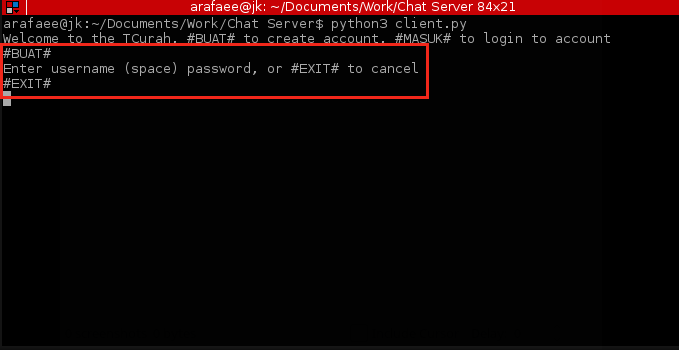
## Membuat Akun

### Deskripsi Fitur :

 Fitur untuk membuat akun baru.

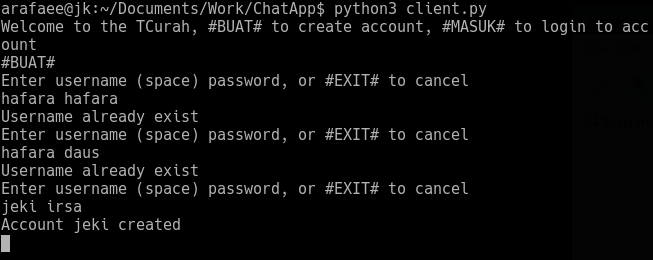
Gambar . Tampilan *Client* Saat Membuat Akun

Gambar . Respon *Server* Saat *Client* Membuat Akun



Gambar . Membuat Akun Cancel

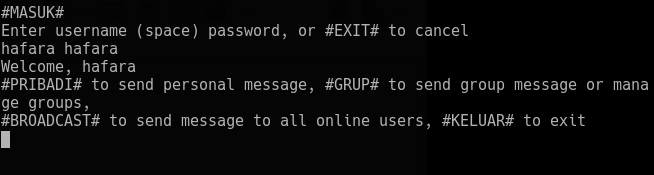
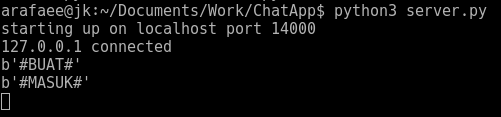
### Error Handling

Bila *username* telah dipakai akan mengeluarkan notifikasi “ Username already exist” 

Gambar . Tampilan *Client* Username Telah Terpakai

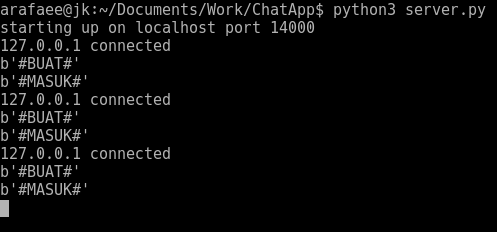
## Masuk Akun

### Deskripsi Fitur

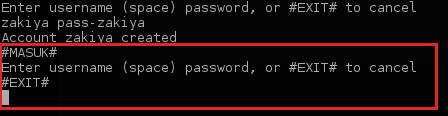
Fitur untuk masuk ke dalam akun TCurah.

Gambar . Tampilan *Client* Masuk Akun

Gambar . Respon *Server* Saat *Client* Masuk Akun

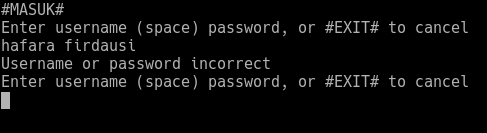


Gambar . Server Dapat Menangani Lebih dari 1 *Client*



Gambar . Cancel Masuk Akun

### Error Handling

Jika memasukkan password yang salah “ Username or password incorrect”

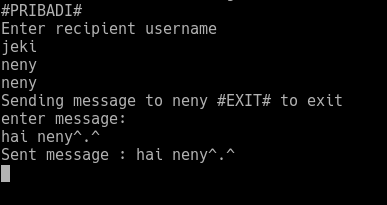
Gambar . User Salah Memasukkan Password

### Sub Menu

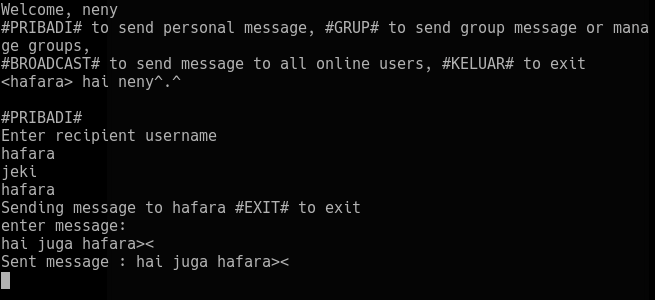
#### Kirim Pesan Pribadi

##### Deskripsi fitur

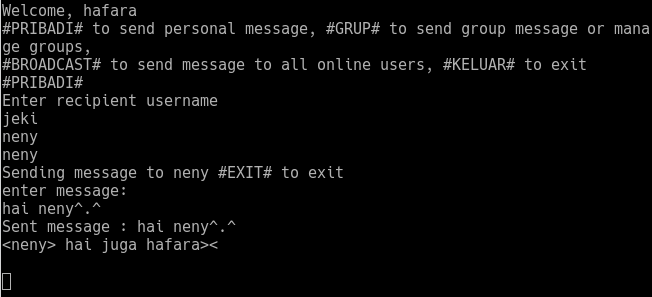
Fitur untuk mengirim pesan pribadi / *personal chat* antar pengguna yang telah masuk.



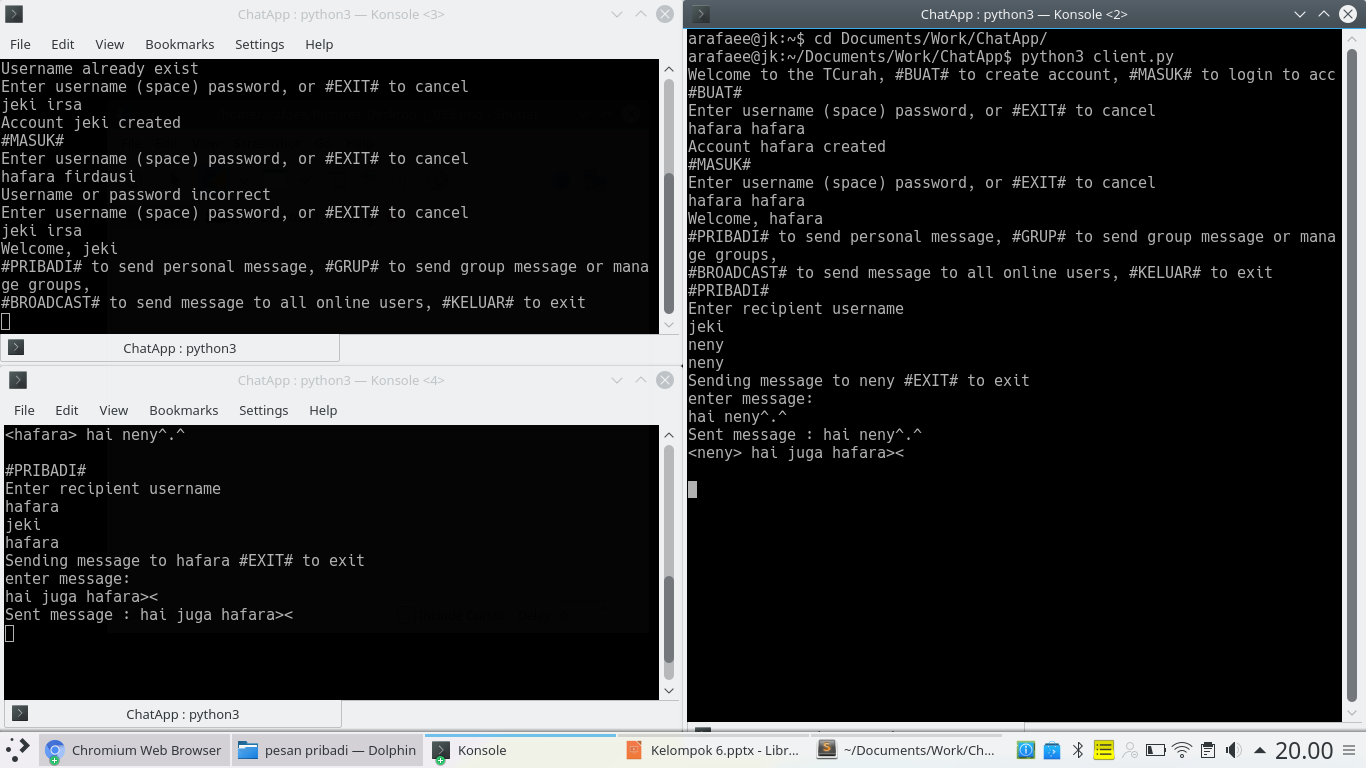
Gambar . *Client 1* Mengirim Pesan ke *Client 3*



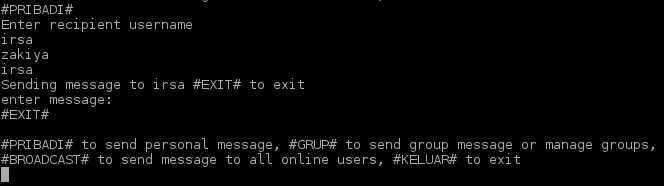
Gambar . *Client 3* Menerima Pesan dan Membalas ke *Client 1*



Gambar . *Client 1* Menerima Balasan dari *Client 3*



Gambar . Perbandingan 3 *Client (Client 2* Tidak Menerima Pesan)

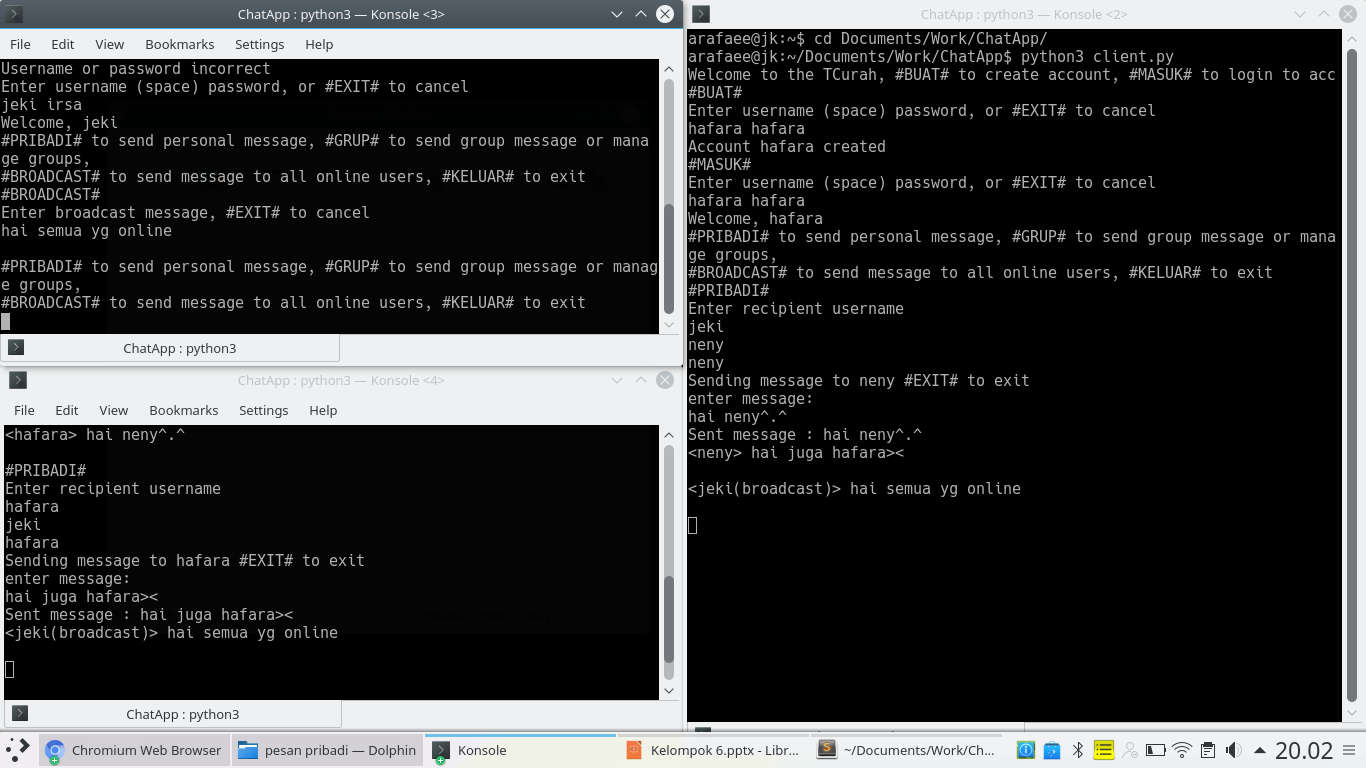


Gambar . Private Message Cancel

#### Kirim Pesan Broadcast

##### Deskripsi Fitur:

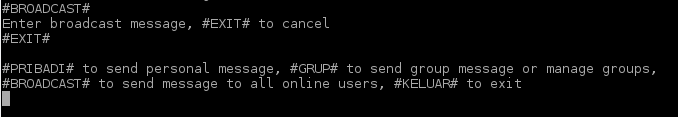
Fitur untuk mengirim pesan broadcast (ke semua pengguna).



Gambar . Tampilan *Client* Saat *Client 2* Mengirim Pesan Broadcast



Gambar . Respon *Server* Saat Mengirim Pesan Broadcast

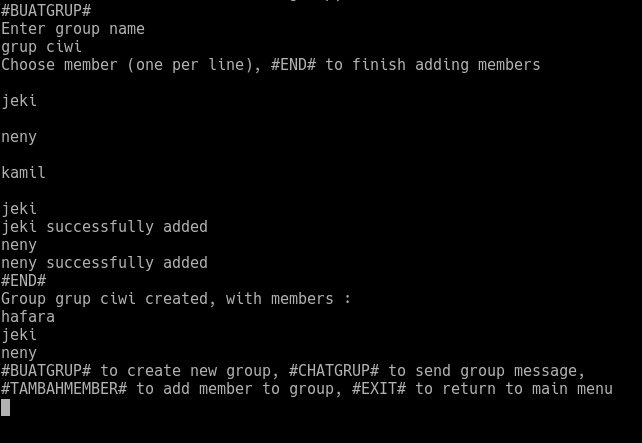


Gambar . Broadcast Message Cancel

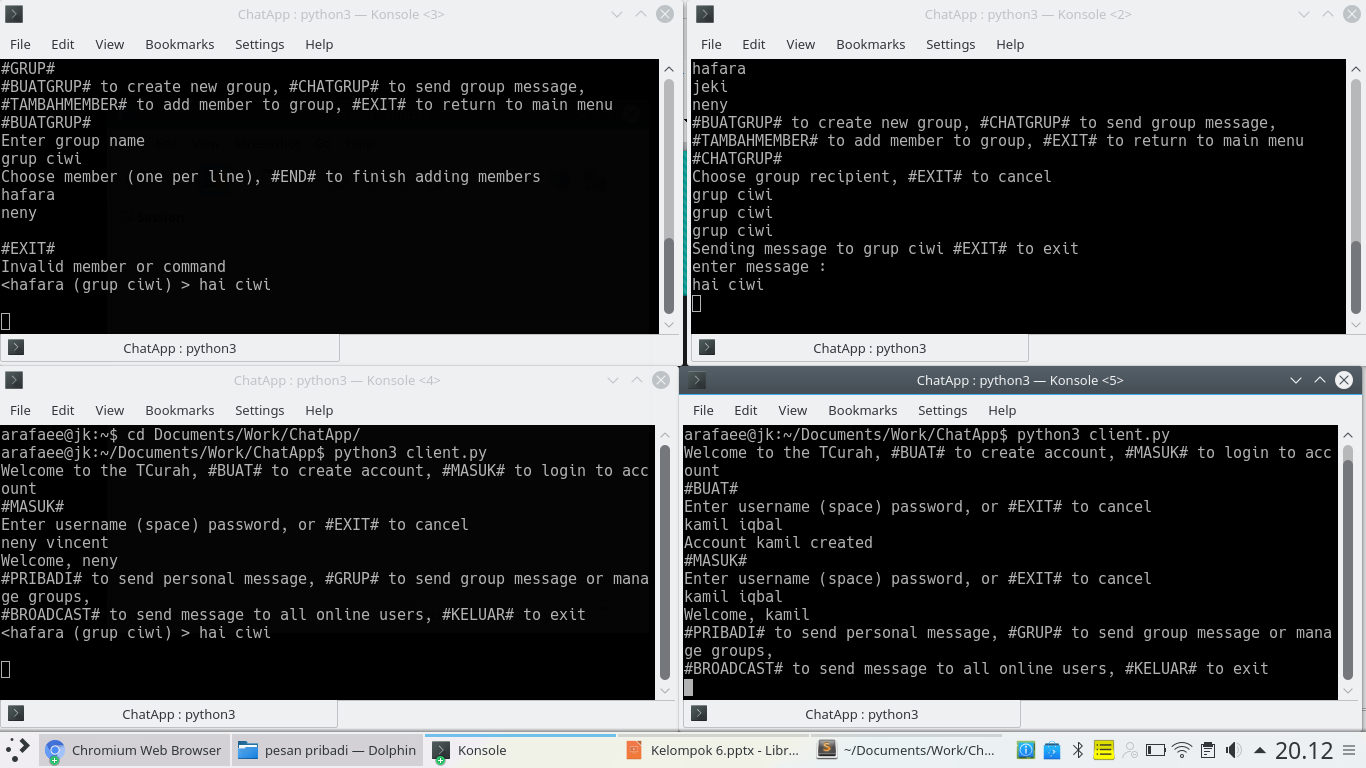
#### *Group Chat*

##### Deskripsi Fitur

Fitur untuk membuat *Group Chat,* menambahkan member yang tergabung dalam grup tersebut, dan mengirim pesan kepada semua member grup.



Gambar . *Client 1* Membuat *Group Chat*



Gambar . *Client 1* Mengirim Pesan di Grup

Semua *Client* yang Tergabung pada *Group Chat* Menerima Pesan, sedangkan Kamil (*Client 4)* tidak, karena Kamil Tidak Tergabung pada *Group Chat* tersebut

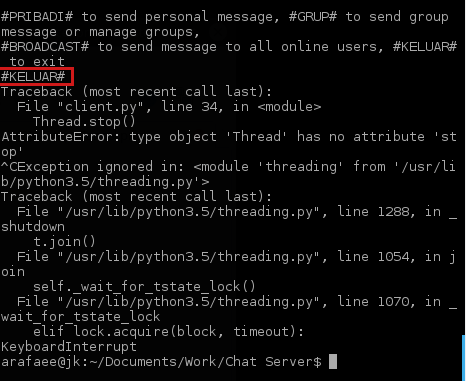


Gambar . Group Message Cancel

#### Keluar Akun

##### Deskripsi Fitur

Fitur untuk keluar akun



Gambar . Tampilan Keluar Akun

#### Bantuan

##### Deskripsi Fitur

Fitur untuk menampilkan deskripsi tiap menu. Menu bantuan tidak dibuat sebagai menu tersendiri, tetapi tiap memilih menu/menjalankan suatu perintah, terdapat petunjuk pada layer *client* seperti fitur-fitur di atas.