**PROJECT 2**

**NETWORK PROGRAMMING COURSE**



**Chatting Application**

**Group 5**

Hero Akbar Ahmadi NRP. 51151000008

Arij Nafi'atul Mashuda NRRP. 5115100013

Huda Fauzan Murtadho NRRP. 5115100022

Yolanda Wisdanita S. NRRP. 5115100023

Salma Nurkhafidoh NRRP. 5115100053

Alvin Mudhoffar NRRP. 5115100062

Djohan Prabowo NRRP. 5115100067

**Lecturer :**

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

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

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**SURABAYA 2017**

**Aplikasi Chat**

1. **Spesifikasi**
2. Fitur

* Multi-Client
* Server bisa meng-handle beberapa client sekaligus.
* Batas maisimal jumlah client yang online adalah sebanyak 10 dalam satu waktu.
* Login
  + User dapat masuk dengan memasukkan username dan password.
  + Misal user belum terdaftar akan memunculkan pesan “User belum terdaftar”.
* Chat antar client
* Chat antar client dapat dilakukan oleh 2 orang yang sedang online.
* Ketika salah satu client sedang offline maka akan dimunculkan pesan “koneksi terputus”.
* Simpan chat dalam database
  + Saat melakukan chat, chat tersebut akan di back-up secara otomatis ke server.
  + Ketika akan melakukan chat kembalidengan orang yang sama setelah koneksi terputus, chat terdahulu akan muncul.
* Enkripsi chat
  + Sebelum pesan dimasukkan ke database, akan dienkripsi dengan base64.
* User Online
  + Dengan fitur ini, user yang sedang online dapat terlihat.
* Logout
  + Client dapat memutusian koneksi dengan server.
* Broadcast
  + Dapat mengirimkan pesan ke semua client yang sedang online.
* Create Group Chat
  + Dapat membuat group chat.

1. **Code**

* **Client**

import socket

import sys

import base64

import select

import os

import json

BUFSIZE=100

server\_address = ('127.0.0.1', 5000)

client\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

client\_socket.connect(server\_address)

print >>sys.stderr, 'connecting to %s port %s' % server\_address

nama = ''

# tampilan awal

def printMenu():

print "--Cetingan Messenger--"

print "1. Login"

print "2. Register"

print "0. Keluar"

print "Pilihan : "

# tampilan setelah login

def printMenuMasuk(name):

print "Halo "+name+"!"

print "Selamat Datang di Cetingan Messenger"

print "------------------------------------"

print "1. List user online"

print "2. Private Chat"

print "3. Broadcast"

print "4. Group Chat"

print "0. Logout"

# group chat

def printGroupMenu():

print "-----Grup Chat Beranda-----"

print "1. Create Grup Chat"

print "2. List Grup Chat"

print "3. Grup Chat"

print "0. Kembali Ke Menu Utama"

def printGroupChat():

print "----------(Masukkan Nama Grup)-----------"

# mencetak pesan

def printPesan(pesan):

for i in xrange(0, len(pesan)+4):

sys.stdout.write("-")

print "\n| "+pesan+" |"

for i in xrange(0, len(pesan)+4):

sys.stdout.write("-")

print ''

def printBack():

print "(Masukkan apapun untuk kembali ke menu)"

def decrypt(msg)://dekripsi pesan

return base64.b64decode(msg)

def getChatHistory(sock)://ambil chat history

chats = json.loads(sock.recv(1024))

#print chats

print '>> Chat History <<\n'

for chat in chats:

print str(chat[0])+'\t:', decrypt(str(chat[2]))

print '>> Chat History END <<'

# chat room

def chatRoomRecv(name, sock):

print "-------------------------------------------------------------"

print "Chatting dengan", name

print "Mode: Menerima Pesan"

print "Petunjuk: Akan otomatis berubah ke mode mengirim pesan ketika "+name

print "sudah mengahiri pengiriman pesannya"

print "-------------------------------------------------------------"

while(1):

pesan = sock.recv(100)

sock.send(pesan)

if(pesan == '0'):

break

elif(pesan == '<<EXIT>>'):

return

else:

print name+':', pesan

print 'Chat Ended'

chatRoomSend(name, sock)

# menerima broadcast

def broadcastRoomRecv(sock):

print "-------------------------------------------------------------"

print "Mode: Menerima Broadcast"

print "-------------------------------------------------------------"

while(1):

pesan = sock.recv(100)

sock.send(pesan)

if(pesan == '0'):

break

elif(pesan == '<<EXIT>>'):

return

else:

print '>', pesan

print 'Chat Ended'

# mengirim pesan

def chatRoomSend(name, sock):

print "-------------------------------------------------------------"

print "Chatting dengan", name

print "Mode: Mengirim Pesan"

print "Petunjuk: "

print "- Masukkan 0 jika ingin mengahiri pengiriman pesan"

print " dan berubah menjadi mode menerima pesan"

print "- Masukkan <<EXIT>> untuk mengahiri percakapan"

print "-------------------------------------------------------------"

while(1):

pesan = raw\_input('> ')

sock.send(pesan)

if(pesan == '0'):

break

if(pesan == '<<EXIT>>'):

return

chatRoomRecv(name, sock)

# mengakhiri broadcast

def broadcastRoomSend(sock):

print "-------------------------------------------------------------"

print "Mode: Broadcast Mengirim Pesan"

print "Petunjuk: "

print "- Masukkan <<EXIT>> untuk mengahiri broadcast"

print "-------------------------------------------------------------"

while(1):

pesan = raw\_input('> ')

sock.send(pesan)

if(pesan == '<<EXIT>>'):

return

pesan = ''

cek=0

# registrasi

while(1):

while cek==0:

os.system('clear')

if (pesan != ''):

printPesan(pesan)

pesan = ''

printMenu()

pil=raw\_input()

client\_socket.send(pil)

#print pil

if pil=="2":

print "Masukkan username anda :"

msg = raw\_input()

client\_socket.send(msg)

print "Masukkan password anda :"

message = raw\_input()

client\_socket.send(message)

psn=client\_socket.recv(1000)

if(psn=="Sudah Ada"):

pesan = "Username Sudah Terdaftar!"

else:

nama=msg

pesan = 'Registrasi Berhasil!'

elif pil=="1":

print "Masukkan username anda :"

msg = raw\_input()

client\_socket.send(msg)

print "Masukkan password anda :"

message = raw\_input()

client\_socket.send(message)

psn=client\_socket.recv(1000)

if(psn=="Gagal Login!"):

pesan = "User belum terdaftar / password salah"

else:

global nama

nama=msg

pesan = "Berhasil Login!"

cek=1

elif pil=="0":

client\_socket.close()

sys.exit(0)

else:

print "Inputan salah"

# cek user online

while cek == 1:

os.system('clear')

if (pesan != ''):

printPesan(pesan)

pesan = ''

printMenuMasuk(nama)

selected = raw\_input()

grup=0

if(selected == "1"):

client\_socket.send(selected)

onuser = client\_socket.recv(1024)

print onuser

onuser = json.loads(onuser)

print onuser

os.system('clear')

print 'User Online:\n'

for index, user in enumerate(onuser):

print index+1, '->', user[1]

print ''

printBack()

raw\_input()

# private chat

if(selected == "2"):

client\_socket.send(selected)

tujuan = raw\_input('Tulis nama yang akan kamu chat: ')

client\_socket.send(tujuan)

action = raw\_input('Masukkan 1 untuk mengirim pesan atau 2 untuk menerima pesan: ')

client\_socket.send(action)

os.system('clear')

if(action == '1'):

getChatHistory(client\_socket)

chatRoomSend(tujuan, client\_socket)

if(action == '2'):

getChatHistory(client\_socket)

chatRoomRecv(tujuan, client\_socket)

# broadcast

if(selected == "3"):

client\_socket.send(selected)

action = raw\_input('Masukkan 1 untuk mengirim pesan atau 2 untuk menerima pesan: ')

client\_socket.send(action)

os.system('clear')

if(action == '1'):

broadcastRoomSend(client\_socket)

if(action == '2'):

broadcastRoomRecv(client\_socket)

# create group chat

if(selected == "4"):

client\_socket.send(selected)

grup=1

while grup == 1:

os.system('clear')

if (pesan != ''):

printPesan(pesan)

pesan = ''

printGroupMenu()

gc\_menu = raw\_input()

client\_socket.send(gc\_menu)

if(gc\_menu == "1"):

print "(Masukkan Nama Grup Yang Akan Dibuat)"

nama\_grup = raw\_input()

client\_socket.send(nama\_grup)

pesan = client\_socket.recv(100)

if(gc\_menu == "2"):

ongrup = client\_socket.recv(100)

# print onuser

ongrup = json.loads(ongrup)

print ongrup

os.system('clear')

print 'List Grup:\n'

for index, grupku in enumerate(ongrup):

print index+1, '->', grupku[1]

print ''

printBack()

raw\_input()

if(gc\_menu == "0"):

grup=0

break

# Logout

if(selected == "0"):

client\_socket.send(selected)

ret = client\_socket.recv(100)

if(ret == '1'):

pesan = 'Berhasil Logout!'

cek = 0

try:

# menutup chat

while True:

socket\_list = [sys.stdin, client\_socket]

read\_sockets, write\_sockets, error\_sockets = select.select(socket\_list , [], [])

for sock in read\_sockets:

if sock == client\_socket:

data = sock.recv(4096)

if not data :

print "koneksi mati"

client\_socket.close()

exit()

else:

print data

else:

print nama

msg = raw\_input()

if msg=="close":

client\_socket.send(msg)

client\_socket.close()

exit()

msg = "\r"+"<"+str(nama)+">"+msg

client\_socket.send(msg)

except KeyboardInterrupt:

client\_socket.close()

sys.exit(0)

* **Server**

import socket

import sys

import mysql.connector

from threading import Thread

import select

import json

import base64

current\_user = []

listSocketUsername = {}

listNumberUsername = {}

class clienthandler(Thread):

def \_\_init\_\_(self,client,number):

global sockets

global idPort

Thread.\_\_init\_\_(self)

self.\_client = client

self.\_number = number

def run(self):

global listNumberUsername

cekk=0

# registrasi

while (1):

while cekk==0:

pil = self.\_client.recv(100)

#print pil

if pil=="2":

print "Register"

usr = self.\_client.recv(100)

pwd = self.\_client.recv(100)

print usr

print enkripsi(pwd)

a=cekusr(usr)

if a==0:

print "Username sudah ada"

self.\_client.send("Sudah Ada")

else:

print "Register berhasil"

self.\_client.send("Belum Ada")

masukdb(usr,pwd)

# login

elif pil=="1":

print "Login"

usr = self.\_client.recv(100)

pwd = self.\_client.recv(100)

print usr

print enkripsi(pwd)

a=cekpwd(usr,pwd,self.\_client)

if a==0:

print "Login gagal"

self.\_client.send("Gagal Login!")

else:

print "Login berhasil"

listNumberUsername[str(self.\_number)] = usr

self.\_client.send("Berhasil Login!")

cekk=1

elif pil=="0":

print "Client Disconnected"

self.\_client.close()

while cekk == 1:

men\_grp = 0

grup = 0

pill = self.\_client.recv(100)

# cek user online

if pill == "1":

data = getUserOnline()

data = json.dumps(data)

print data

self.\_client.send(data)

if pill == "2":

lawan = self.\_client.recv(100)

action = self.\_client.recv(100)

global listSocketUsername

sock\_lawan = listSocketUsername[lawan]

for index, sock\_l in enumerate(listSocketUsername):

print index+1, '->', sock\_l

# private chat antar client

if(action == '1'):

chats = getChatHistory(listNumberUsername[str(self.\_number)], lawan)

print chats

chats = json.dumps(chats)

self.\_client.send(chats)

chat(lawan, self.\_client, sock\_lawan, self.\_number)

print "RETURNED!"

elif(action == '2'):

chats = getChatHistory(listNumberUsername[str(self.\_number)], lawan)

print chats

chats = json.dumps(chats)

self.\_client.send(chats)

terimaChat(lawan, self.\_client, sock\_lawan, self.\_number)

# broadcast

if pill == "3":

action = self.\_client.recv(100)

if(action == '1'):

broadcastChat(self.\_client, self.\_number)

print "RETURNED!"

elif(action == '2'):

broadcastTerimaChat(self.\_client, self.\_number)

# membuat grup chat

if pill == "4":

men\_grp = 1

while men\_grp == 1:

men = self.\_client.recv(100)

if men == "1":

nama\_grup = self.\_client.recv(100)

print nama\_grup

createGroup(nama\_grup)

self.\_client.send('Berhasil Membuat Grup')

elif men == "2":

data = getGroup()

data = json.dumps(data)

print data

self.\_client.send(data)

elif men == "0":

men\_grp = 0

break

# logout

if pill == "0":

a = doLogout()

cekk = 0

self.\_client.send('1')

sockets=[]#buat kumpulan client

idPort = []

sock=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM)

server\_address=('localhost',5000)

print >> sys.stderr,'starting up on %s port %s'%server\_address

sock.bind(server\_address)

sock.setsockopt(socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1)

sock.listen(1)

sockets.append(sock)

# masuk ke database

def masukdb(user,passw):

#mysql.connector.connect(host='localhost',database='fp',user='root',password='')

#sudo dpkg -i mysql-connector-python\_2.0.5-1ubuntu16.04\_all.deb install mysql

cnx = mysql.connector.connect(host='localhost',database='fp',user='fp',password='fp')

cursor = cnx.cursor()

add\_user = ("INSERT INTO user "

"(nama, password) "

"VALUES (%s, %s)")

data\_user = (user, passw)

cursor.execute(add\_user, data\_user)

cnx.commit()

print "Berhasil Mendaftarkan User"

cursor.close()

cnx.close()

# cek user dari database

def cekusr(usr):

a=[]

cnx = mysql.connector.connect(host='localhost',database='fp',user='fp',password='fp')

cursor = cnx.cursor(buffered=True)

add\_user = "SELECT \* FROM user WHERE nama=%s"

cursor.execute(add\_user,(usr,))

data = cursor.fetchall()

for row in data:

a.append(row)

cnx.commit()

cursor.close()

cnx.close()

if len(a)>0:

return 0

else:

return 1

# cek username dan password dari database

def cekpwd(usr,pwd,sockclient):

#print usr

#print pwd

global current\_user

a=[]

cnx = mysql.connector.connect(host='localhost',database='fp',user='fp',password='fp')

cursor = cnx.cursor(buffered=True)

add\_user = "SELECT \* FROM user WHERE nama=%s and password=%s"

cursor.execute(add\_user,(usr,pwd,))

data = cursor.fetchall()

for row in data:

a.append(row)

cnx.commit()

if len(a)>0:

cursor = cnx.cursor(buffered=True)

change\_status = "UPDATE user SET status='1' WHERE id=%d"

cursor.execute(change\_status % (a[0][0]))

cnx.commit()

current\_user = a

#print 'a = ', a

#print 'cur = ', current\_user

global listSocketUsername

listSocketUsername[str(a[0][1])] = sockclient

#print listSocketUsername

cursor.close()

cnx.close()

return 1

else:

cursor.close()

cnx.close()

return 0

# cek user online dari database

def getUserOnline():

global current\_user

a=[]

cnx = mysql.connector.connect(host='localhost',database='fp',user='fp',password='fp')

cursor = cnx.cursor(buffered=True)

query = "SELECT \* FROM user WHERE status = 1"

cursor.execute(query)

data = cursor.fetchall()

for row in data:

a.append(row)

cnx.commit()

return a

# mengambil data grup dari database

def getGroup():

global current\_user

a=[]

cnx = mysql.connector.connect(host='localhost',database='fp',user='fp',password='fp')

cursor = cnx.cursor(buffered=True)

query = "SELECT \* FROM grup"

cursor.execute(query)

data = cursor.fetchall()

for row in data:

a.append(row)

cnx.commit()

return a

# logout

def doLogout():

global current\_user

cnx = mysql.connector.connect(host='localhost',database='fp',user='fp',password='fp')

cursor = cnx.cursor(buffered=True)

query = "UPDATE user SET status='0' WHERE id=%d"

cursor.execute(query % (current\_user[0][0]))

cnx.commit()

print current\_user, "logged out"

cursor.close()

cnx.close()

return 1

# chat dan enkripsi pesan

def chat(lawan, sock, sock\_lawan, thread\_number):

global listNumberUsername

while(1):

#print 'masuk fungsi chat'

pengirim = listNumberUsername[str(thread\_number)]

#print pengirim

pesan = sock.recv(100)

#print pesan

sock\_lawan.send(pesan)

if(pesan == '0'):

break

if(pesan == '<<EXIT>>'):

return

else:

insertChat(pengirim, lawan, pesan)

print 'sent', enkripsi(pesan), 'to', sock\_lawan

terimaChat(lawan, sock, sock\_lawan, thread\_number)

# broadcast

def broadcastChat(sock, sock\_lawans, thread\_number):

global listNumberUsername

global listSocketUsername

while(1):

#print 'masuk fungsi chat'

pengirim = listNumberUsername[str(thread\_number)]

#print pengirim

pesan = sock.recv(100)

#print pesan

#iterasi

for index, sock\_lawan in enumerate(listSocketUsername):

sck\_lwn = listSocketUsername(sock\_lawan)

sck\_lwn.send(pesan)

if(pesan == '0'):

break

if(pesan == '<<EXIT>>'):

return

else:

#iterasi

for index, lwn in enumerate(listSocketUsername):

insertChat(pengirim, lwn, pesan)

print 'sent', enkripsi(pesan), 'to', lwn

#terimaChat(lawan, sock, sock\_lawan, thread\_number)

# mode menerima pesan

def terimaChat(lawan, sock, sock\_lawan, thread\_number):

while(1):

pesan = sock.recv(100)

print 'FROM CLIENT ->', pesan

if(pesan == '0'):

break

if(pesan == '<<EXIT>>'):

return

else:

print lawan+':', pesan

# menerima broadcast pesan

def broadcastTerimaChat(sock, thread\_number):

while(1):

pesan = sock.recv(100)

print 'FROM CLIENT ->', pesan

if(pesan == '0'):

break

if(pesan == '<<EXIT>>'):

return

else:

print ':', pesan

def enkripsi(pesan):

return base64.b64encode(pesan)

# masukkan chat ke database

def insertChat(dari, ke, pesan):

pesan = enkripsi(pesan)

cnx = mysql.connector.connect(host='localhost',database='fp',user='fp',password='fp')

cursor = cnx.cursor(buffered=True)

query = "INSERT INTO pesan (nama\_pengirim, nama\_penerima, pesan) VALUES ('%s', '%s', '%s')"

print query % (dari, ke, pesan)

cursor.execute(query % (dari, ke, pesan))

cnx.commit()

cursor.close()

cnx.close()

return 1

# mengambil chat history dari database

def getChatHistory(pengirim, penerima):

chats = []

cnx = mysql.connector.connect(host='localhost',database='fp',user='fp',password='fp')

cursor = cnx.cursor(buffered=True)

query = """SELECT nama\_pengirim, nama\_penerima, pesan FROM pesan

WHERE nama\_pengirim='%s' AND nama\_penerima='%s'

OR nama\_pengirim='%s' AND nama\_penerima='%s' ORDER BY created\_at"""

print query % (pengirim, penerima, penerima, pengirim)

cursor.execute(query % (pengirim, penerima, penerima, pengirim))

data = cursor.fetchall()

for row in data:

chats.append(row)

cnx.commit()

cursor.close()

cnx.close()

return chats

# memasukkan grup yang dibuat ke database

def createGroup(nama\_grup):

cnx = mysql.connector.connect(host='localhost',database='fp',user='fp',password='fp')

cursor = cnx.cursor()

query = "INSERT INTO grup (nama) VALUES ('%s')"

cursor.execute(query % (nama\_grup))

cnx.commit()

cursor.close()

cnx.close()

# broadcast pesan

def broadcast(sockx,message):

for skt in sockets:

if skt!=sock and skt!=sockx:

try :

print "pesan berhasil dikirim"

skt.send(message)

except:

print "tidak berhasil dikirim"

skt.close()

sockets.remove(skt)

input\_socket=[sock]

try:

while True:

handlers = []

read\_sockets,write\_sockets,error\_sockets = select.select(sockets,[],[])

for sockx in read\_sockets:

if sock==sockx:

client, address = sock.accept()

if len(sockets) > 2:

sockets.remove(sock)

sockx.close()

else:

sockets.append(client)

print 'client connected from: ', address[0],'with id : ', address[1]

idPort.append(address[1])

newthread = clienthandler(client,address[1])

handlers.append(newthread)

newthread.daemon = True

newthread.start()

else:

try:

datax = sockx.recv(1024)

if datax=="close":

print "mati"

sockx.close()

sockets.remove(sockx)

continue

elif datax:

print "hidup"

broadcast(sockx,datax)

print datax

except:

print "mati"

print sockx

sockx.close()

sockets.remove(sockx)

continue

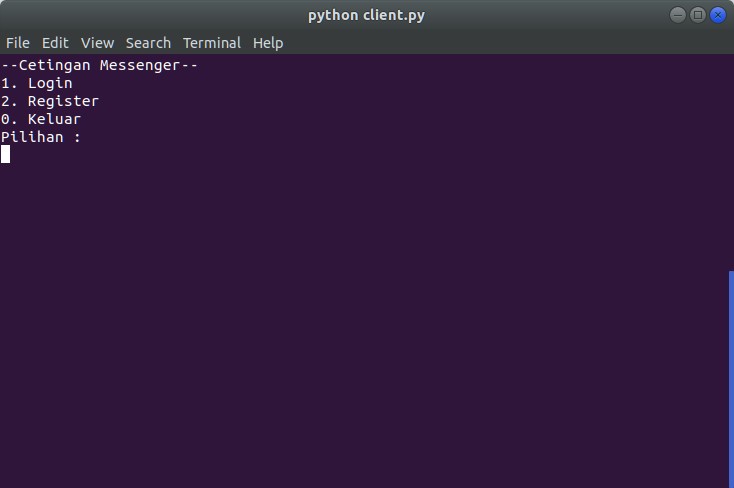
except KeyboardInterrupt:

sock.close()

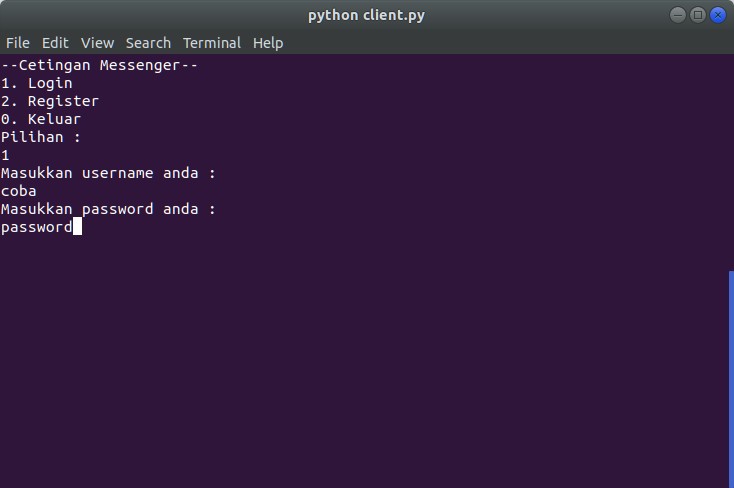
sys.exit(0)

1. **Screenshot Program**

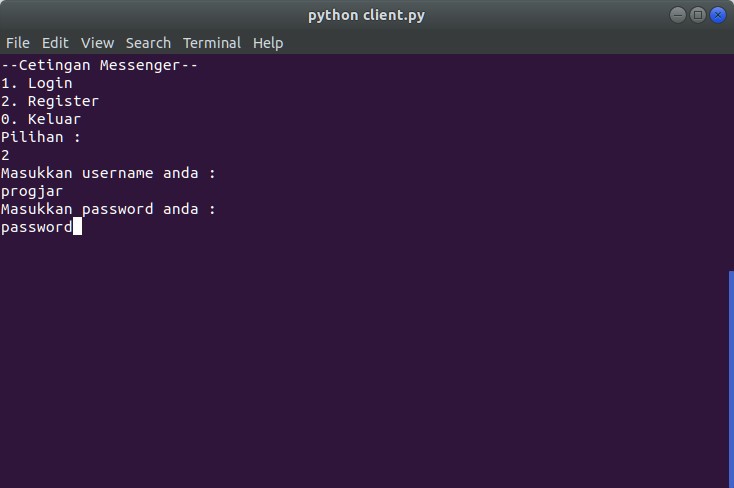
* Tampilan Awal



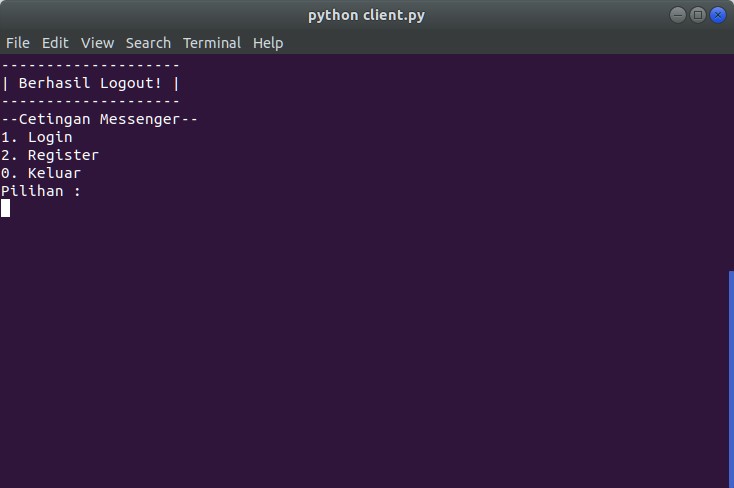
* Login



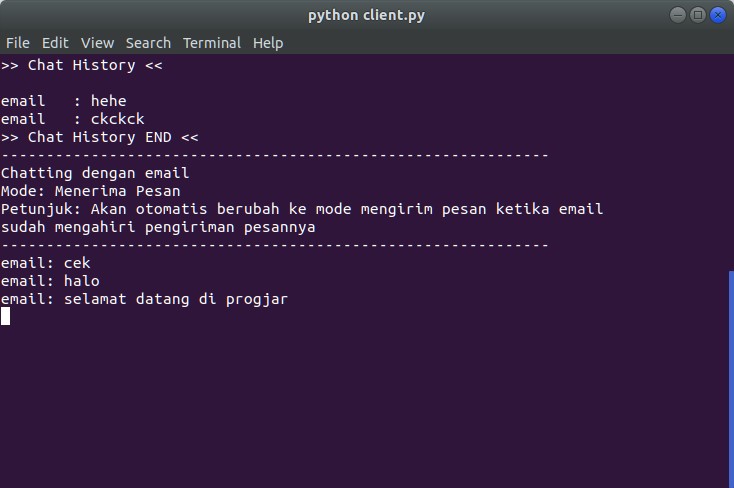
* Register

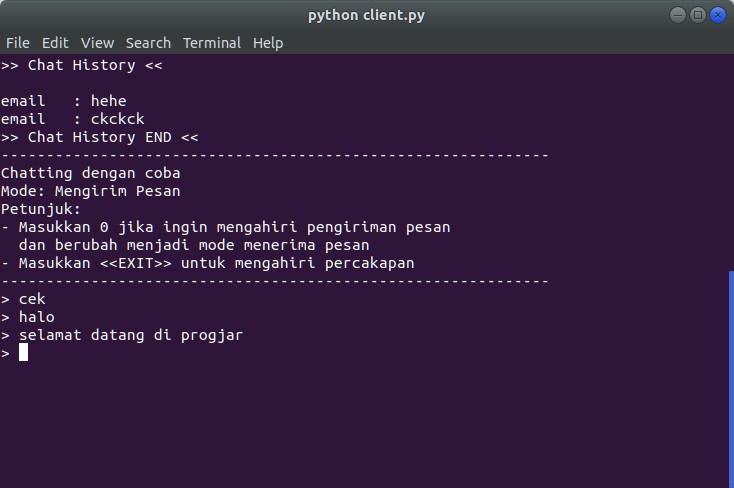


* Logout

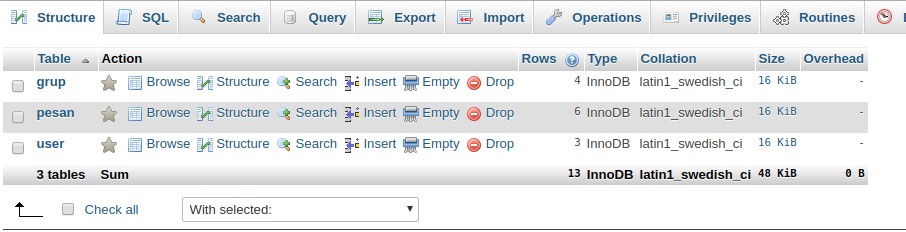


* Chat antar client





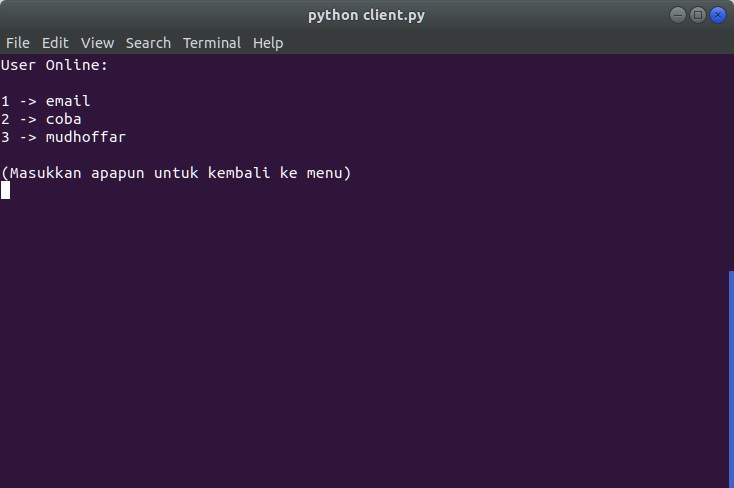
* Simpan chat dalam database



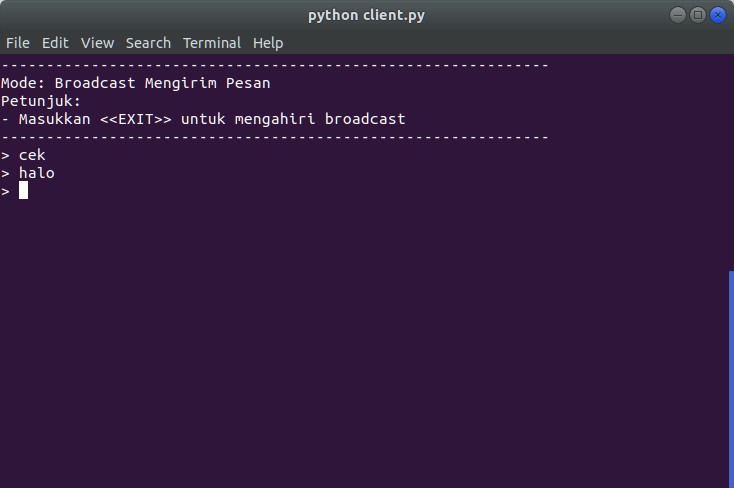
* Enkripsi chat

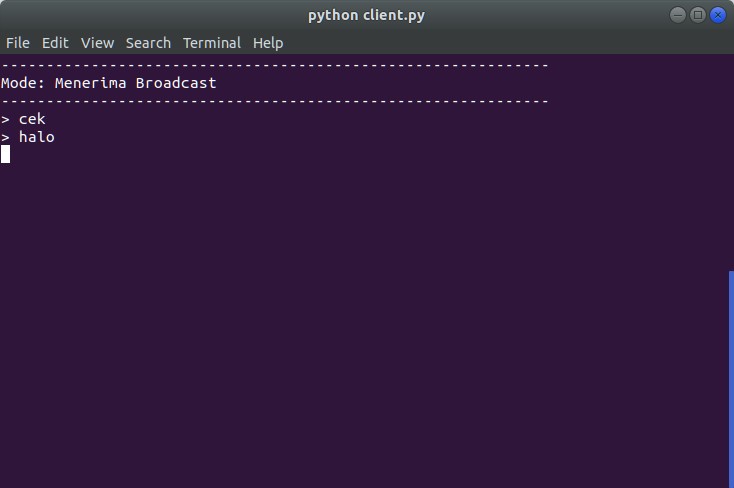


* User Online

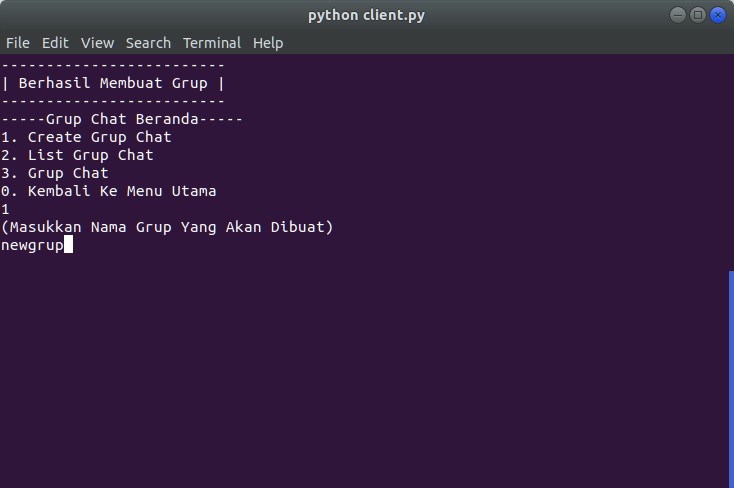


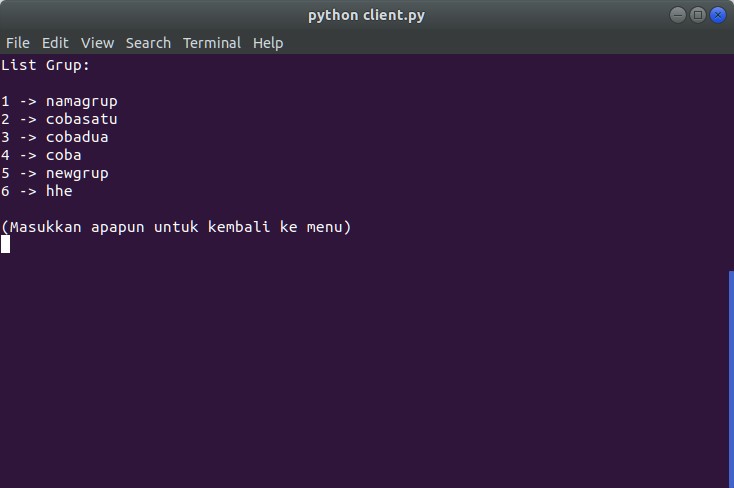
* Broadcast





* Create Group Chat





1. **Pembagian tiap kelompok**

|  |  |  |
| --- | --- | --- |
| **NRP** | **Nama** | **Tugas** |
| 5115100006 | Hero Akbar Ahmadi | Multi-client, Chat antar client |
| 5115100013 | Arij Nafi’atul M. | Login, Dokumentasi Laporan |
| 5115100022 | Huda Fauzan Murthado | Simpan chat dalam database, Chat antar client |
| 5115100023 | Yolanda Wisdanita S. | User Online, Dokumentasi Laporan |
| 5115100053 | Salam Nurkhafidoh | Logout, Dokumentasi Laporan |
| 5115100062 | Alvin Mudhoffar | Broadcast, Enkripsi Chat antar client |
| 5115100067 | Djohan Prabowo | Create Group Chat, Chat antar client |