**“APLIKASI CHATTING CLIENT SOCKET MENGGUNAKAN PYTHON”**

Tugas: Pemrograman Jaringan

Dosen Pengampu:

Muhammad Priyono Tri Sulistyanto, S.T, M.Eng



Dibuat Oleh:

(Fesa Dwi Cahyo 170403010069)

**TEKNIK INFORMATIKA**

**UNIVERSITAS KANJURUHAN MALANG**

**MALANG**

**2019**

# **A. CLIENT**

import socket, select, string, sys //import libraray yang akan dipakai, socket dll

#Helper function (formatting)

def display() :

you="\33[33m\33[1m"+" You: "+"\33[0m"

sys.stdout.write(you)

sys.stdout.flush()

def main(): //fungsi utama

if len(sys.argv)<2: //fungsi if untuk

host = raw\_input("Enter host ip address: ") //

else:

host = sys.argv[1]

port = 5001 //client dengan port 5001

#asks for user name

//client diminta untuk memasukkan username baru.

name=raw\_input("\33[34m\33[1m CREATING NEW ID:\n Enter username: \33[0m")

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

s.settimeout(2)

# Untuk mengkoneksikan hostnya.

try :

s.connect((host, port)) //koneksi ke host dan port yang sudah disetting.

# Jika koneksi gagal, maka akan menampilkan pesan ini.

except :

print "\33[31m\33[1m Can't connect to the server \33[0m"

#Sistem akan ter-close jika koneksi gagal.

sys.exit()

#if connected

#mengirimkan nama dan menampilkan namanya.

s.send(name)

display()

while 1:

socket\_list = [sys.stdin, s]

# Mengambil list sockets yang ready.

rList, wList, error\_list = select.select(socket\_list , [], [])

#perulangan untuk setiap socket

for sock in rList:

#Pesan masuk dari server

#Membaca data yang masuk dari server

if sock == s:

data = sock.recv(4096)

#Jika tidak ada data, maka akan memberikan pesan disconnected dan sistem keluar

if not data :

print '\33[31m\33[1m \rDISCONNECTED!!\n \33[0m'

sys.exit()

#Jika ada data, maka user bisa menulis pesan dan pesan akan dimunculkan.

else :

sys.stdout.write(data)

display()

#user entered sebuah pesan

#variabel msg akan menyimpan data dan mengirimkannya, kemudian menampilkan pesan yang sudah diketik.

else :

msg=sys.stdin.readline()

s.send(msg)

display()

#fungsi untuk memanggil file main ini.

if \_\_name\_\_ == "\_\_main\_\_":

main()

# **B. SERVER**

import socket, select

#Berfungsi untuk mengirim pesan ke semua klien yang terhubung

def send\_to\_all (sock, message):

# Pesan tidak diteruskan ke server dan pengirim itu sendiri

for socket in connected\_list:

if socket != server\_socket and socket != sock :

try :

socket.send(message)

except :

# jika koneksi tidak tersedia maka koneksi list akan dihapus.

socket.close()

connected\_list.remove(socket)

if \_\_name\_\_ == "\_\_main\_\_":

name=""

# kamus untuk menyimpan alamat yang sesuai dengan nama pengguna

record={}

# Daftar untuk melacak deskriptor soket

connected\_list = []

buffer = 4096

port = 5001

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

server\_socket.bind(("localhost", port))

server\_socket.listen(10) # dengarkan paling banyak 10 koneksi sekaligus

# Tambahkan soket server ke daftar koneksi yang dapat dibaca

connected\_list.append(server\_socket)

print "\33[32m \t\t\t\tSERVER WORKING \33[0m"

while 1:

# Dapatkan daftar soket yang siap dibaca pilih

rList,wList,error\_sockets = select.select(connected\_list,[],[])

for sock in rList:

#Koneksi baru

if sock == server\_socket:

# Menangani kasus di mana ada koneksi baru yang diterima melalui server\_socket

sockfd, addr = server\_socket.accept()

name=sockfd.recv(buffer)

connected\_list.append(sockfd)

record[addr]=""

#print rekam dan daftar sambungan ", rekam, connected\_list

# jika diulangi username

if name in record.values():

sockfd.send("\r\33[31m\33[1m Username already taken!\n\33[0m")

del record[addr]

connected\_list.remove(sockfd)

sockfd.close()

continue

else:

# tambahkan nama dan alamat

record[addr]=name

print "Client (%s, %s) connected" % addr," [",record[addr],"]"

sockfd.send("\33[32m\r\33[1m Welcome to chat room. Enter 'tata' anytime to exit\n\33[0m")

send\_to\_all(sockfd, "\33[32m\33[1m\r "+name+" joined the conversation \n\33[0m")

# Beberapa pesan masuk dari klien

else:

# Data dari client

try:

data1 = sock.recv(buffer)

#print "sock is: ",sock

data=data1[:data1.index("\n")]

#print "\ndata received: ",data

#mendapatkan addr dari client mengirimkan pesan

i,p=sock.getpeername()

if data == "tata":

msg="\r\33[1m"+"\33[31m "+record[(i,p)]+" left the conversation \33[0m\n"

send\_to\_all(sock,msg)

print "Client (%s, %s) is offline" % (i,p)," [",record[(i,p)],"]"

del record[(i,p)]

connected\_list.remove(sock)

sock.close()

continue

else:

msg="\r\33[1m"+"\33[35m "+record[(i,p)]+": "+"\33[0m"+data+"\n"

send\_to\_all(sock,msg)

# keluarkan pengguna secara tiba-tiba

except:

(i,p)=sock.getpeername()

send\_to\_all(sock, "\r\33[31m \33[1m"+record[(i,p)]+" left the conversation unexpectedly\33[0m\n")

print "Client (%s, %s) is offline (error)" % (i,p)," [",record[(i,p)],"]\n"

del record[(i,p)]

connected\_list.remove(sock)

sock.close()

continue

server\_socket.close()