

Simple-Socket-Chat

Anggota Kelompok:

IBRAHIM TAMTAMA ADI 05111640000018

AGUEL SATRIA WIJAYA 05111640000056

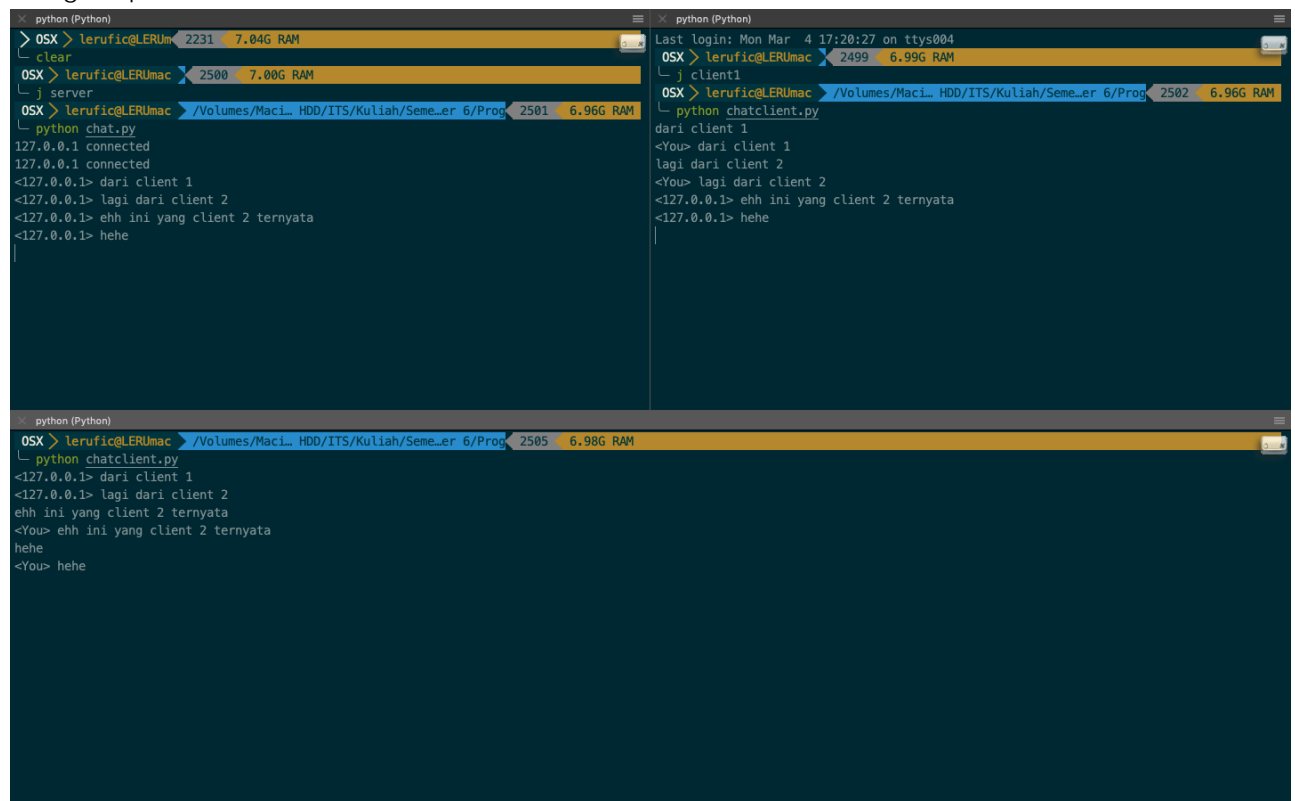
Fungsionalitas

Program ini harus bisa:

- Mengirim pesan biasa dan di broadcast ke semua clients
- Bisa mengirim file dengan perintah SEND minimal 3MB

HASIL

1. Mengirim pesan biasa berhasil

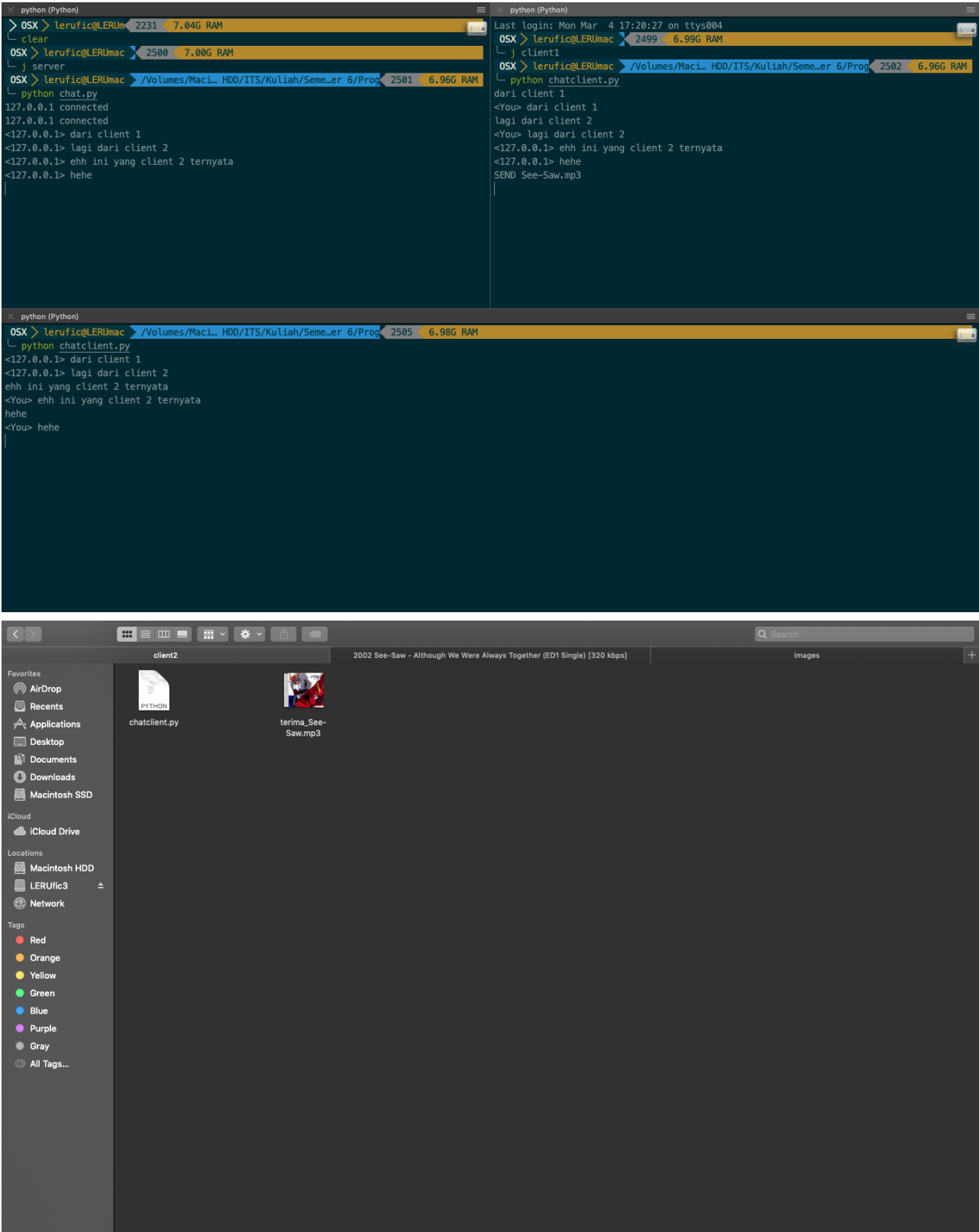


```
python (Python)
OSX > lerufic@LERUmac 2231 7.04G RAM
clear
OSX > lerufic@LERUmac 2500 7.00G RAM
j server
OSX > lerufic@LERUmac /Volumes/Mac... HDD/ITS/Kuliah/Seme...er 6/Prog 2501 6.96G RAM
python chat.py
127.0.0.1 connected
127.0.0.1 connected
<127.0.0.1> dari client 1
<127.0.0.1> lagi dari client 2
<127.0.0.1> ehh ini yang client 2 ternyata
<127.0.0.1> hehe

python (Python)
Last login: Mon Mar 4 17:20:27 on ttys004
OSX > lerufic@LERUmac 2499 6.99G RAM
j client1
OSX > lerufic@LERUmac /Volumes/Mac... HDD/ITS/Kuliah/Seme...er 6/Prog 2502 6.96G RAM
python chatclient.py
dari client 1
<You> dari client 1
lagi dari client 2
<You> lagi dari client 2
<127.0.0.1> ehh ini yang client 2 ternyata
<127.0.0.1> hehe

python (Python)
OSX > lerufic@LERUmac /Volumes/Mac... HDD/ITS/Kuliah/Seme...er 6/Prog 2505 6.98G RAM
python chatclient.py
<127.0.0.1> dari client 1
<127.0.0.1> lagi dari client 2
ehh ini yang client 2 ternyata
<You> ehh ini yang client 2 ternyata
hehe
<You> hehe
```

2. Berhasil mengirim file dengan nama **See-Saw.mp3** (11.5MB) dan muncul pada client2 dengan nama **terima_See-Saw.mp3**. Client 1 dan 2 juga masih bisa saling berkomunikasi tanpa ada error.



```
python (Python)
OSX > lerufic@LERUmac 2231 7.04G RAM
└─ clear
OSX > lerufic@LERUmac 2500 7.00G RAM
└─ j server
OSX > lerufic@LERUmac /Volumes/Mac... HDD/ITS/Kuliah/Seme_er 6/Prog 2501 6.96G RAM
└─ python chat.py
127.0.0.1 connected
127.0.0.1 connected
<127.0.0.1> dari client 1
<127.0.0.1> lagi dari client 2
<127.0.0.1> ehh ini yang client 2 ternyata
<127.0.0.1> hehe
Berhasil gan
<127.0.0.1> Berhasil gan
<127.0.0.1> Berhasil gan
Mantap
<127.0.0.1> Mantap
<127.0.0.1> Mantap

python (Python)
Last login: Mon Mar 4 17:20:27 on ttys004
OSX > lerufic@LERUmac 2499 6.99G RAM
└─ j client1
OSX > lerufic@LERUmac /Volumes/Mac... HDD/ITS/Kuliah/Seme_er 6/Prog 2502 6.96G RAM
└─ python chatclient.py
dari client 1
<You> dari client 1
lagi dari client 2
<You> lagi dari client 2
<127.0.0.1> ehh ini yang client 2 ternyata
<127.0.0.1> hehe
SEND See-Saw.mp3
Berhasil gan
<You> Berhasil gan
<127.0.0.1> Berhasil gan
Mantap
<You> Mantap
<127.0.0.1> Mantap

python (Python)
OSX > lerufic@LERUmac /Volumes/Mac... HDD/ITS/Kuliah/Seme_er 6/Prog 2505 6.98G RAM
└─ python chatclient.py
<127.0.0.1> dari client 1
<127.0.0.1> lagi dari client 2
ehh ini yang client 2 ternyata
<You> ehh ini yang client 2 ternyata
hehe
<You> hehe
Berhasil gan
<You> Berhasil gan
<127.0.0.1> Berhasil gan
Mantap
<You> Mantap
<127.0.0.1> Mantap
```

Penjelasan Code

Server (chat.py)

```
def clientthread(conn, addr):
    while True:
        try:
            message = conn.recv(2048)
            if message:
                if "SEND" in message:
                    messagex = message.split('\n')
                    broadcast(messagex[0], conn)

                    messages = message.split(' ')

                    messages = messages[1].split('\n')
                    with open("terima_"+messages[0],
'wb') as file:

                        while True:
                            data = conn.recv(2048)
                            if '//AGUEL//TAMTAM//' in
                                break
                            file.write(data)
                            broadcast(data, conn)

                        else:
                            messages = message.split('\n')
```

```

        print "<" + addr[0] + "> " +
messages[0]

        message_to_send = "<" + addr[0] +
"> " + messages[0]
        broadcast(message_to_send, conn)
    else:
        remove(conn)
except:
    continue

```

Pertama server akan menerima pesan dari client. Pesan jika pesan ada isinya maka akan dilanjutkan ke tahap berikutnya namun jika tidak maka data client akan dihapus dari list_of_clients.

Pada tahap selanjutnya pesan akan dibedakan menjadi 2 yaitu pesan dengan kata **SEND** didalamnya dan pesan tanpa SEND.

Pesan dengan SEND akan menerima pesan yang pertama yaitu adalah pesan yang berisi commandnya. Dari pesan tersebut akan diambil nama filenya lalu dibuka dan diisi dengan data yang diterima berikutnya yaitu data file yang dikirim. Kami menggunakan wb karena data yang server terima berupa tulisan binary. Server akan mencari string **//AGUEL//TAMTAM//** jika ada string tersebut artinya file sudah terkirim semua dan berikutnya keluar dari while. Semua pesan yang diterima oleh server akan dikirimkan kembali ke client-client untuk dibuat filenya pada masing-masing client.

Pesan yang tidak memiliki send akan dikirim langsung ke semua client karena hanya berupa pesan biasa.

Client (chatclient.py)

```

while True:
    sockets_list = [sys.stdin, server]
    read_sockets, write_socket, error_socket =
select.select(sockets_list, [], [])

    for socks in read_sockets:
        if socks == server:
            message = socks.recv(2048)
            if message:
                if "SEND" not in message:
                    print message
                if "SEND" in message:
                    messages = message.split(' ')
                    messages = messages[1].split('\n')
                    with open("terima_"+messages[0],
'wb') as file:

                        while True:
                            data = socks.recv(2048)
                            if '//AGUEL//TAMTAM//' in
                                break
                            file.write(data)
            else:

```

```
message = sys.stdin.readline()
if "SEND" in message:
    server.send(message)
    messages = message.split(' ')
    messages = messages[1].split('\n')
    file = open(messages[0], 'rb')
    line = file.read(2048)
    while(line):
        server.send(line)
        line = file.read(2048)
    file.close()
    server.send("//AGUEL//TAMTAM//")

else:
    server.send(message)
    sys.stdout.write("<You> ")
    sys.stdout.write(message)
    sys.stdout.flush()

server.close()
```

Pada clients socket dibagi menjadi 2 hal yaitu socket yang mengirim dan socket yang menerima pesan. Pada socks yang menerima pesan, pesan akan dibagi jadi 2 sama seperti pada server yaitu pesan dengan SEND dan tanpa SEND. Pesan tanpa SEND akan langsung di print dan pesan dengan SEND akan diproses untuk dibuat filenya. Algoritma yang digunakan ketika pembentukan file adalah sama dengan yang ada di server.

Pada socks yang mengirim pesan juga dibagi jadi 2 yaitu pesan dengan SEND dan tanpa SEND. Pesan dengan send akan dibaca ngan rb (secara binary) dan dikirim perline ke server. Jika sudah selesai maka akan diakhiri dengan string *//AGUEL//TAMTAM//* dan dikirimkan ke server agar server tahu bahwa proses pengiriman telah selesai. Sedangkan pesan tanpa SEND akan langsung dikirimkan secara langsung ke server.