

## KELOMPOK 20

Anggota :

1. Zahra Fayyadiyah - 5025201133
2. Khariza Azmi Alfajira Hisyam - 5025201044

### Screenshot kode program

server.py

```
import socket
import select
import sys
import threading

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
ip_address = '127.0.0.1'
port = 8081
server.bind((ip_address, port))
server.listen(100)
list_of_clients = []

def clientthread(conn, addr):
    while True:
        try:
            message = conn.recv(2048).decode()
            if message:
                if message:
                    result = message + str(eval(message))
                    print('<' + addr[0] + '>' + result)
                    message_to_send = '<' + addr[0] + '>' + result
                    broadcast(message_to_send, conn)
                else:
                    remove(conn)
        except:
            continue

def broadcast(message, connection):
    for clients in list_of_clients:
        if clients != connection:
            try:
                clients.send(message.encode())
            except:
                clients.close()
                remove(clients)
```

```

def remove(connection):
    if connection in list_of_clients:
        list_of_clients.remove(connection)

while True:
    conn, addr = server.accept()
    list_of_clients.append(conn)
    print(addr[0] + ' connected')
    threading.Thread(target=clientthread, args=(conn, addr)).start()

conn.close()

```

#### client.py

```

import socket
import select
import sys
from threading import Thread

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
ip_address = '127.0.0.1'
port = 8081
server.connect((ip_address, port))

def send_msg(sock):
    while True:
        data = sys.stdin.readline()
        sock.send(data.encode())
        sys.stdout.write('<You>')
        sys.stdout.write(data)
        sys.stdout.flush()

def recv_msg(sock):
    while True:
        data = sock.recv(2048)
        sys.stdout.write(data.decode())

Thread(target=send_msg, args=(server,)).start()
Thread(target=recv_msg, args=(server,)).start()

while True:
    socket_list = [server]
    read_socket, write_socket, error_socket =
select.select(socket_list, [], [])

```

```

for socks in read_socket:
    if socks == server:
        sys.stdout.write("Pesan terkirim ke client 2 dan client 3")
        recv_msg(socks)
    else:
        print("Menerima pesan dari client 1.")
        send_msg(socks)

server.close()

```

Penjelasan :

Client membuka dua thread. yang pertama untuk menunggu recv dan 1 lagi untuk menunggu input. Apabila user sudah memasukkan input, dia akan mengirimkan data input ke server. Pada server, data akan di evaluasi menjadi operasi aritmatika, sehingga server bisa menghitung hasil dari inputan user. Server mengirimkan hasil yang sudah diproses kepada client yang masih aktif, sehingga client yang lain dapat menerima hasil yang telah diproses oleh server.

### Screenshot eksekusi program

```

127.0.0.1 connected
127.0.0.1 connected
<127.0.0.1>3+5-3
5
<127.0.0.1>3-4-6
-7
<127.0.0.1>3-8+9
4
[]

PS D:\PROGJAR\Pertemuan4\challenge1> python client.py
3+5-3
<You>3+5-3
3-4-6
<You>3-4-6
<127.0.0.1>3-8+9
4
[]

PS D:\PROGJAR\Pertemuan4> cd challenge1
PS D:\PROGJAR\Pertemuan4\challenge1> python client.p
y
<127.0.0.1>3+5-3
5<127.0.0.1>3-4-6
-7
<You>
3-8+9
<You>3-8+9
[]

```

Ketika client mengirimkan operasi ke server maka server akan memberikan jawaban kemudian mengirimkan ke client satunya