

Laporan Praktikum Pemrograman Jaringan

Single Thread Chatting App

Nama : Aditya Aulia Rohman

NIM : 1203210080

Kelas : IF-01-01

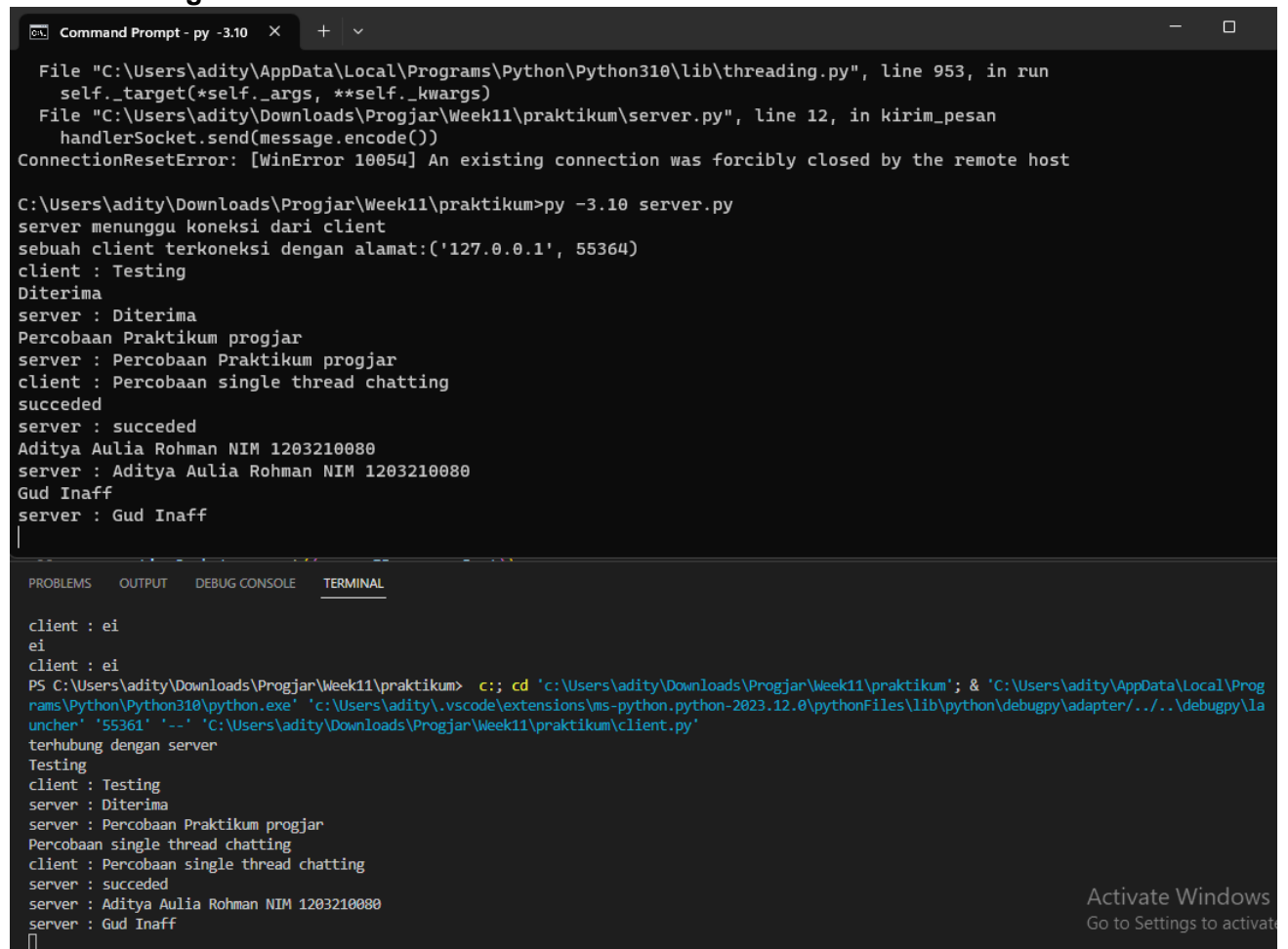
Code server:

```
server.py > ...
1  import socket
2  from threading import Thread
3
4  # konfigurasi server
5  listenerSocket = socket.socket()
6  serverIP = "0.0.0.0"
7  serverPort = 2222
8
9  def kirim_pesan(handlerSocket: socket.socket):
10     while True:
11         message = input()
12         handlerSocket.send(message.encode())
13         print("server : {}".format(message))
14
15  def terima_pesan(handlerSocket: socket.socket):
16     while True:
17         message = handlerSocket.recv(1024)
18         print("client : {}".format(message.decode('utf-8')))
19
20  # binding socket dengan IP dan port
21  listenerSocket.bind((serverIP, serverPort))
22
23  # listener socket siap menerima koneksi
24  listenerSocket.listen(0)
25  print("server menunggu koneksi dari client")
26
27  # listener socket menunggu koneksi dari client, line di bawah ini bersifat 'blocking'
28  # artinya, programnya terhenti di sini sampai ada koneksi ke listenerSocket
29  handler, addr = listenerSocket.accept()
30
31  # jika sudah ada koneksi dari client, maka program akan jalan ke line ini
32  print("sebuah client terkoneksi dengan alamat:{}".format(addr))
33  t1 = Thread(target=kirim_pesan, args=(handler,))
34  t2 = Thread(target=terima_pesan, args=(handler,))
35
36  t1.start()
37  t2.start()
38  t1.join()
39  t2.join()
```

Code client:

```
client.py > ...
1  from email import message
2  import socket
3  from threading import Thread
4
5  # konfigurasi client dan server
6  connectionSocket = socket.socket()
7  serverIP = "127.0.0.1"
8  serverPort = 2222
9
10 def kirim_pesan(handlerSocket: socket.socket):
11     while True:
12         message = input()
13         handlerSocket.send(message.encode())
14         print("client : {}".format(message))
15
16 def terima_pesan(handlerSocket: socket.socket):
17     while True:
18         message = handlerSocket.recv(1024)
19         print("server : {}".format(message.decode("utf-8")))
20
21 # menghubungi server
22 connectionSocket.connect((serverIP, serverPort))
23 print("terhubung dengan server")
24
25 t1 = Thread(target=terima_pesan, args=(connectionSocket,))
26 t2 = Thread(target=kirim_pesan, args=(connectionSocket,))
27
28 t1.start()
29 t2.start()
30 t1.join()
31 t2.join()
```

Hasil Running:



```
File "C:\Users\adity\AppData\Local\Programs\Python\Python310\lib\threading.py", line 953, in run
  self._target(*self._args, **self._kwargs)
File "C:\Users\adity\Downloads\Progjar\Week11\praktikum\server.py", line 12, in kirim_pesan
  handlerSocket.send(message.encode())
ConnectionResetError: [WinError 10054] An existing connection was forcibly closed by the remote host

C:\Users\adity\Downloads\Progjar\Week11\praktikum>py -3.10 server.py
server menunggu koneksi dari client
sebuah client terkoneksi dengan alamat>('127.0.0.1', 55364)
client : Testing
Diterima
server : Diterima
Percobaan Praktikum progjar
server : Percobaan Praktikum progjar
client : Percobaan single thread chatting
succeeded
server : succeeded
Aditya Aulia Rohman NIM 1203210080
server : Aditya Aulia Rohman NIM 1203210080
Gud Inaff
server : Gud Inaff

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

client : ei
ei
client : ei
PS C:\Users\adity\Downloads\Progjar\Week11\praktikum> c:: cd 'c:\Users\adity\Downloads\Progjar\Week11\praktikum'; & 'C:\Users\adity\AppData\Local\Programs\Python\Python310\python.exe' 'c:\Users\adity\.vscode\extensions\ms-python.python-2023.12.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher' '55361' '--' 'C:\Users\adity\Downloads\Progjar\Week11\praktikum\client.py'
terhubung dengan server
Testing
client : Testing
server : Diterima
server : Percobaan Praktikum progjar
Percobaan single thread chatting
client : Percobaan single thread chatting
server : succeeded
server : Aditya Aulia Rohman NIM 1203210080
server : Gud Inaff
```

Dari Percobaan program singlethread chatting app berjalan dengan sukses, kedua sisi dapat mengirim pesan ke sesama lain.

Pada server side pertama dibuat socket listener dengan `socket.socket()`, lalu di bind ke IP dan port yang sudah ditentukan, kemudian server `listen()` untuk membiarkan koneksi masuk dari client, lalu server akan menunggu koneksi dari client dengan `listenerSocket.accept()`, saat ada koneksi masuk server print alamat client yang terhubung, lalu dijalankan fungsi `kirim_pesan` dan `terima_pesan` yang diloop selama koneksi masih terhubung.

Pada client side pertama dibuat socket dengan `socket.socket()`, lalu menghubungi server dengan IP dan port yang telah ditentukan, setelah terhubung dijalankan fungsi `kirim_pesan` dan `terima_pesan` yang diloop selama koneksi masih terhubung.