Nama : Fariz Hidayat

Kelas : 4IA10

Npm : 51421595

**PROJECT PJAR**

**Source Code TCP Server**

import socket

import threading

import tkinter as tk

from tkinter import scrolledtext

def start\_server():

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

    server\_socket.bind(('localhost', 12345))

    server\_socket.listen(1)

    log.insert(tk.END, "Menunggu koneksi client...\n")

    status\_label.config(text="Status: Menunggu koneksi...", fg="orange")

    conn, addr = server\_socket.accept()

    log.insert(tk.END, f"Terhubung dengan {addr}\n")

    status\_label.config(text=f"Status: Terhubung dengan {addr}", fg="green")

    def handle\_client():

        while True:

            try:

                data = conn.recv(1024).decode()

                if not data:

                    break

                log.insert(tk.END, f"Client: {data}\n")

                conn.send(f"Pesan diterima: {data}".encode())

            except:

                break

    threading.Thread(target=handle\_client, daemon=True).start()

# GUI

window = tk.Tk()

window.title("TCP Server")

window.configure(bg="#f0f0f0")

window.resizable(False, False)

frame = tk.Frame(window, bg="#f0f0f0", padx=10, pady=10)

frame.pack()

title = tk.Label(frame, text="TCP Server", font=("Helvetica", 16, "bold"), bg="#f0f0f0", fg="#333")

title.grid(row=0, column=0, columnspan=2, pady=(0, 10))

status\_label = tk.Label(frame, text="Status: Belum berjalan", font=("Helvetica", 10), bg="#f0f0f0", fg="red")

status\_label.grid(row=1, column=0, columnspan=2, sticky="w")

log = scrolledtext.ScrolledText(frame, width=50, height=20, font=("Consolas", 10), bg="#fff")

log.grid(row=2, column=0, columnspan=2, pady=5)

btn\_start = tk.Button(frame, text="Mulai Server", bg="#2196F3", fg="white", font=("Helvetica", 10, "bold"), command=lambda: threading.Thread(target=start\_server).start())

btn\_start.grid(row=3, column=0, pady=10)

btn\_clear = tk.Button(frame, text="Clear Log", bg="#f44336", fg="white", font=("Helvetica", 10), command=lambda: log.delete(1.0, tk.END))

btn\_clear.grid(row=3, column=1, pady=10)

window.mainloop()

**Source Code TCP Client**

import socket

import threading

import tkinter as tk

from tkinter import scrolledtext

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

# GUI

window = tk.Tk()

window.title("TCP Client")

window.configure(bg="#f0f0f0")

window.resizable(False, False)

frame = tk.Frame(window, bg="#f0f0f0", padx=10, pady=10)

frame.pack()

title = tk.Label(frame, text="TCP Client", font=("Helvetica", 16, "bold"), bg="#f0f0f0", fg="#333")

title.grid(row=0, column=0, columnspan=3, pady=(0, 10))

status\_label = tk.Label(frame, text="Status: Tidak Terhubung", font=("Helvetica", 10), bg="#f0f0f0", fg="red")

status\_label.grid(row=1, column=0, columnspan=3, sticky="w", pady=(0, 5))

log = scrolledtext.ScrolledText(frame, width=50, height=20, font=("Consolas", 10), bg="#fff")

log.grid(row=2, column=0, columnspan=3, pady=5)

entry = tk.Entry(frame, width=40, font=("Helvetica", 10))

entry.grid(row=3, column=0, pady=10)

btn\_send = tk.Button(frame, text="Kirim", bg="#4CAF50", fg="white", font=("Helvetica", 10, "bold"), command=lambda: send\_message())

btn\_send.grid(row=3, column=1, padx=5)

btn\_clear = tk.Button(frame, text="Clear Log", bg="#f44336", fg="white", font=("Helvetica", 10), command=lambda: log.delete(1.0, tk.END))

btn\_clear.grid(row=3, column=2)

def send\_message():

    msg = entry.get()

    if msg:

        client\_socket.send(msg.encode())

        log.insert(tk.END, f"Anda: {msg}\n")

        entry.delete(0, tk.END)

def receive():

    while True:

        try:

            data = client\_socket.recv(1024).decode()

            if data:

                log.insert(tk.END, f"Server: {data}\n")

        except:

            break

def connect\_to\_server():

    try:

        client\_socket.connect(('localhost', 12345))

        status\_label.config(text="Status: Terhubung ke Server", fg="green")

        threading.Thread(target=receive, daemon=True).start()

    except:

        status\_label.config(text="Status: Gagal Terhubung", fg="red")

connect\_to\_server()

window.mainloop()

**Source Code UDP Server**

import socket

import threading

import tkinter as tk

from tkinter import scrolledtext

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

server\_socket.bind(('localhost', 12346))

def receive():

    status\_label.config(text="Status: Menunggu pesan dari client...", fg="orange")

    while True:

        data, addr = server\_socket.recvfrom(1024)

        msg = data.decode()

        log.insert(tk.END, f"Client: {msg}\n")

        server\_socket.sendto(f"Pesan diterima: {msg}".encode(), addr)

        status\_label.config(text=f"Status: Terhubung dengan {addr}", fg="green")

# GUI

window = tk.Tk()

window.title("UDP Server")

window.configure(bg="#f0f0f0")

window.resizable(False, False)

frame = tk.Frame(window, bg="#f0f0f0", padx=10, pady=10)

frame.pack()

title = tk.Label(frame, text="UDP Server", font=("Helvetica", 16, "bold"), bg="#f0f0f0", fg="#333")

title.grid(row=0, column=0, columnspan=2, pady=(0, 10))

status\_label = tk.Label(frame, text="Status: Belum berjalan", font=("Helvetica", 10), bg="#f0f0f0", fg="red")

status\_label.grid(row=1, column=0, columnspan=2, sticky="w")

log = scrolledtext.ScrolledText(frame, width=50, height=20, font=("Consolas", 10), bg="#fff")

log.grid(row=2, column=0, columnspan=2, pady=5)

btn\_start = tk.Button(frame, text="Mulai Server", bg="#2196F3", fg="white", font=("Helvetica", 10, "bold"), command=lambda: threading.Thread(target=receive).start())

btn\_start.grid(row=3, column=0, pady=10)

btn\_clear = tk.Button(frame, text="Clear Log", bg="#f44336", fg="white", font=("Helvetica", 10), command=lambda: log.delete(1.0, tk.END))

btn\_clear.grid(row=3, column=1, pady=10)

window.mainloop()

**Source UDP Client**

import socket

import threading

import tkinter as tk

from tkinter import scrolledtext

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

server\_addr = ('localhost', 12346)

# GUI

window = tk.Tk()

window.title("UDP Client")

window.configure(bg="#f0f0f0")

window.resizable(False, False)

frame = tk.Frame(window, bg="#f0f0f0", padx=10, pady=10)

frame.pack()

title = tk.Label(frame, text="UDP Client", font=("Helvetica", 16, "bold"), bg="#f0f0f0", fg="#333")

title.grid(row=0, column=0, columnspan=3, pady=(0, 10))

status\_label = tk.Label(frame, text="Status: Siap mengirim", font=("Helvetica", 10), bg="#f0f0f0", fg="green")

status\_label.grid(row=1, column=0, columnspan=3, sticky="w", pady=(0, 5))

log = scrolledtext.ScrolledText(frame, width=50, height=20, font=("Consolas", 10), bg="#fff")

log.grid(row=2, column=0, columnspan=3, pady=5)

entry = tk.Entry(frame, width=40, font=("Helvetica", 10))

entry.grid(row=3, column=0, pady=10)

btn\_send = tk.Button(frame, text="Kirim", bg="#4CAF50", fg="white", font=("Helvetica", 10, "bold"), command=lambda: send\_message())

btn\_send.grid(row=3, column=1, padx=5)

btn\_clear = tk.Button(frame, text="Clear Log", bg="#f44336", fg="white", font=("Helvetica", 10), command=lambda: log.delete(1.0, tk.END))

btn\_clear.grid(row=3, column=2)

def send\_message():

    msg = entry.get()

    if msg:

        client\_socket.sendto(msg.encode(), server\_addr)

        log.insert(tk.END, f"Anda: {msg}\n")

        entry.delete(0, tk.END)

def receive():

    while True:

        try:

            data, \_ = client\_socket.recvfrom(1024)

            log.insert(tk.END, f"Server: {data.decode()}\n")

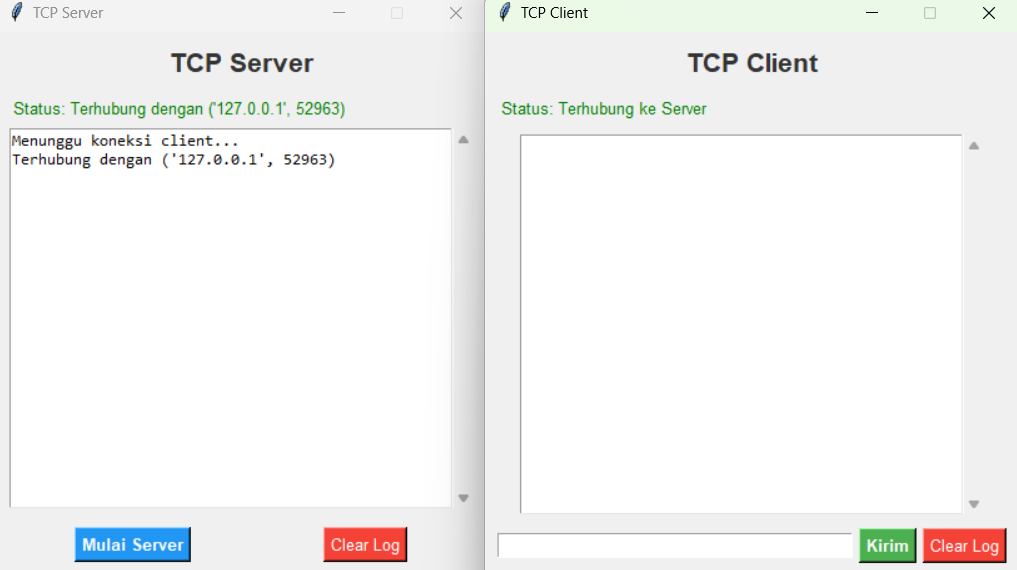
        except:

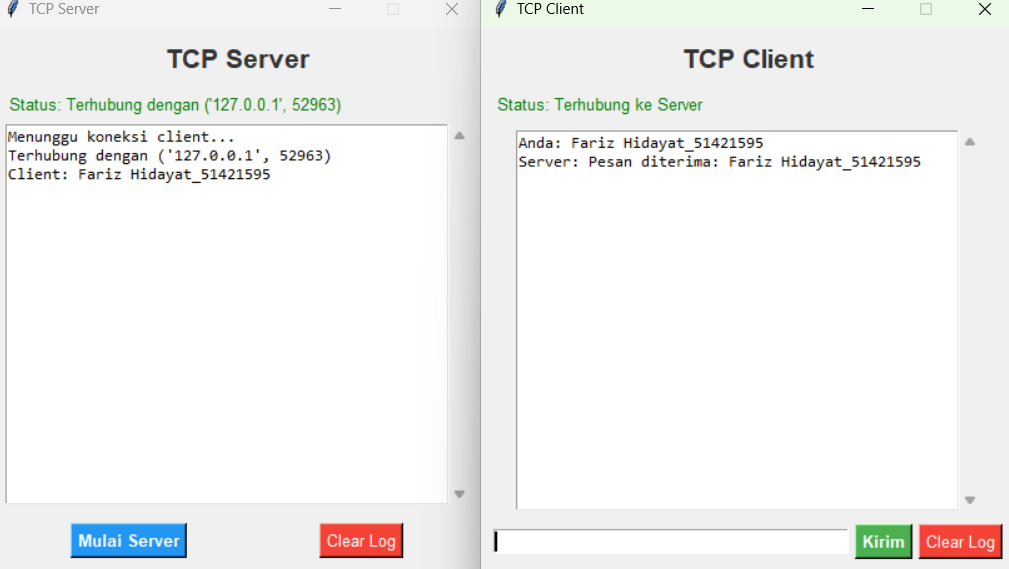
            break

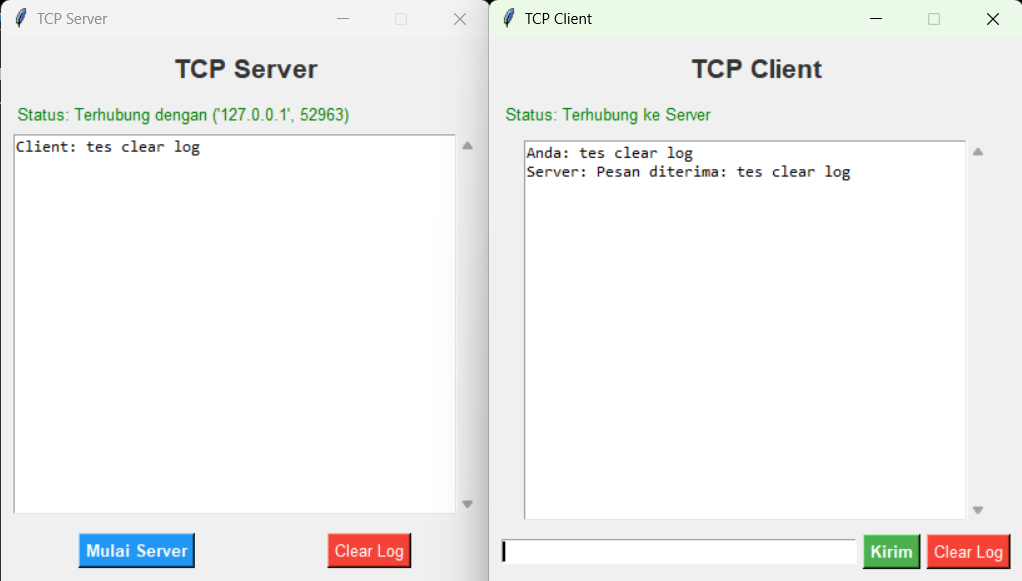
threading.Thread(target=receive, daemon=True).start()

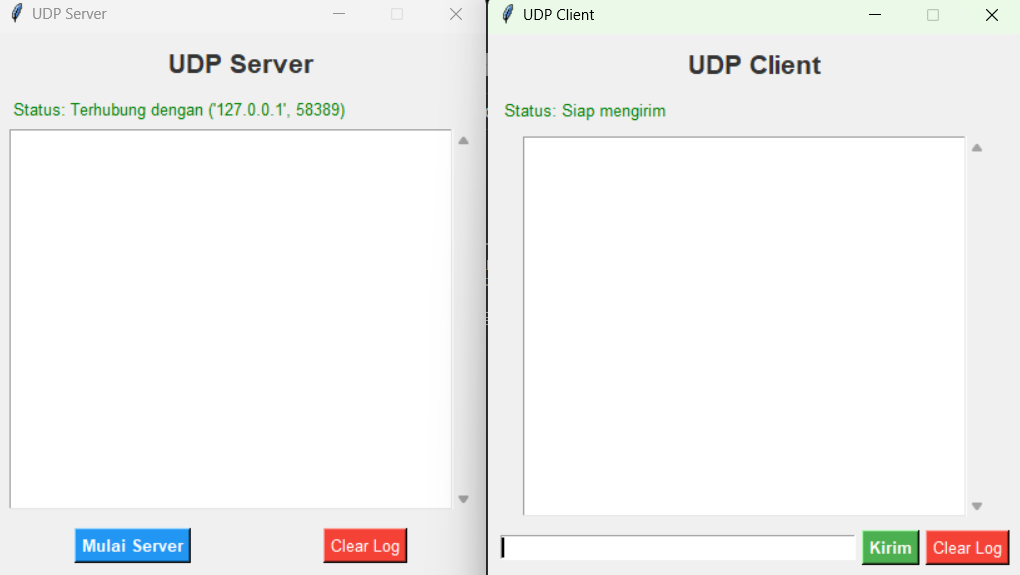
window.mainloop()

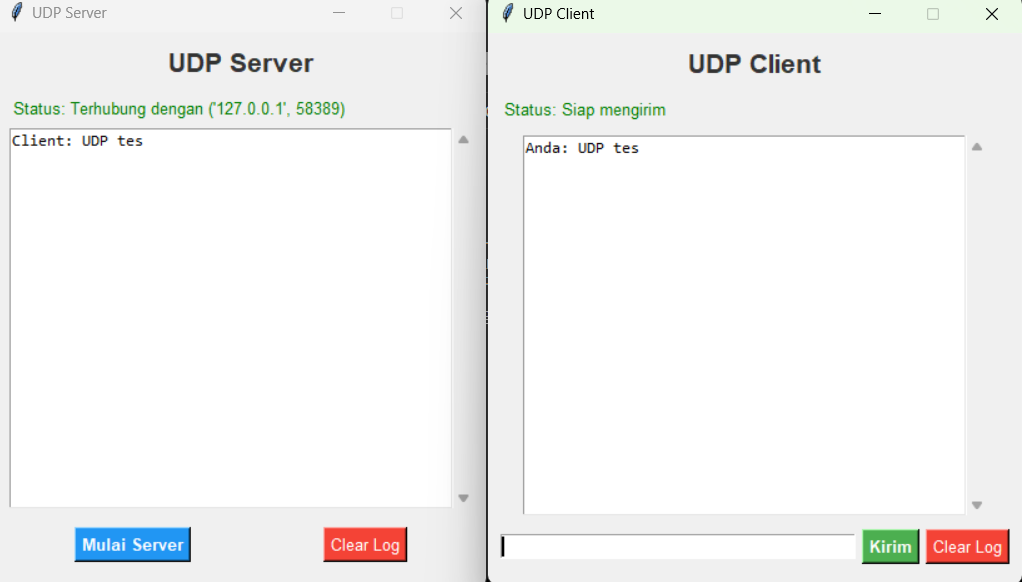
**Output**

****

****

****

****

****

**Pejelasan Source Code**

1. **TCP Client**

**Komponen Utama:**

python

SalinEdit

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

client\_socket.connect(('localhost', 12345))

* Membuat socket TCP.
* Terhubung ke server di localhost:12345.

**Fungsi Kirim:**

python

SalinEdit

def send\_message():

msg = entry.get()

if msg:

client\_socket.send(msg.encode())

log.insert(tk.END, f"Anda: {msg}\n")

entry.delete(0, tk.END)

* Mengambil teks dari entry box.
* Mengirimkannya ke server.
* Menampilkan pesan di log.

**Fungsi Terima:**

python

SalinEdit

def receive():

while True:

data = client\_socket.recv(1024).decode()

if data:

log.insert(tk.END, f"Server: {data}\n")

* Terus mendengarkan balasan dari server.
* Menampilkan ke GUI jika ada pesan masuk.

1. **TCP Server**

**Komponen Utama:**

python

SalinEdit

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

server\_socket.bind(('localhost', 12345))

server\_socket.listen(1)

* Membuat socket TCP.
* Server mendengarkan koneksi masuk di port 12345.

**Menerima Client:**

python

SalinEdit

conn, addr = server\_socket.accept()

* Menunggu koneksi dari satu client.

**Fungsi handle\_client:**

python

SalinEdit

def handle\_client():

while True:

data = conn.recv(1024).decode()

if not data:

break

log.insert(tk.END, f"Client: {data}\n")

conn.send(f"Pesan diterima: {data}".encode())

* Terima data dari client.
* Tampilkan di GUI.
* Kirim respons balik.

1. **UDP Client**

**Setup:**

python

SalinEdit

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

server\_addr = ('localhost', 12346)

* Socket UDP tidak membuat koneksi.
* Siap mengirim ke alamat tujuan.

**Kirim Pesan:**

python

SalinEdit

client\_socket.sendto(msg.encode(), server\_addr)

* Kirim langsung ke server tanpa koneksi.

**Terima Pesan:**

python

SalinEdit

data, \_ = client\_socket.recvfrom(1024)

* Menerima pesan balasan dari server.

1. **UDP Server**

**Setup:**

python

SalinEdit

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

server\_socket.bind(('localhost', 12346))

* Socket UDP menunggu pesan di port 12346.

**Menerima & Membalas:**

python

SalinEdit

data, addr = server\_socket.recvfrom(1024)

msg = data.decode()

server\_socket.sendto(f"Pesan diterima: {msg}".encode(), addr)

* Terima pesan dari client.
* Kirim balasan ke pengirim.

1. **Perbedaan TCP dan UDP**

| **Aksi** | **TCP** | **UDP** |
| --- | --- | --- |
| Membuat koneksi awal | connect() | Tidak ada (sendto) |
| Terima data | recv() | recvfrom() |
| Kirim data | send() | sendto() |
| Sinkronisasi | Ya (harus konek dulu) | Tidak perlu koneksi |