Nama : Fariz Hidayat

Kelas : 4IA10

Npm : 51421595

PROJECT PJAR

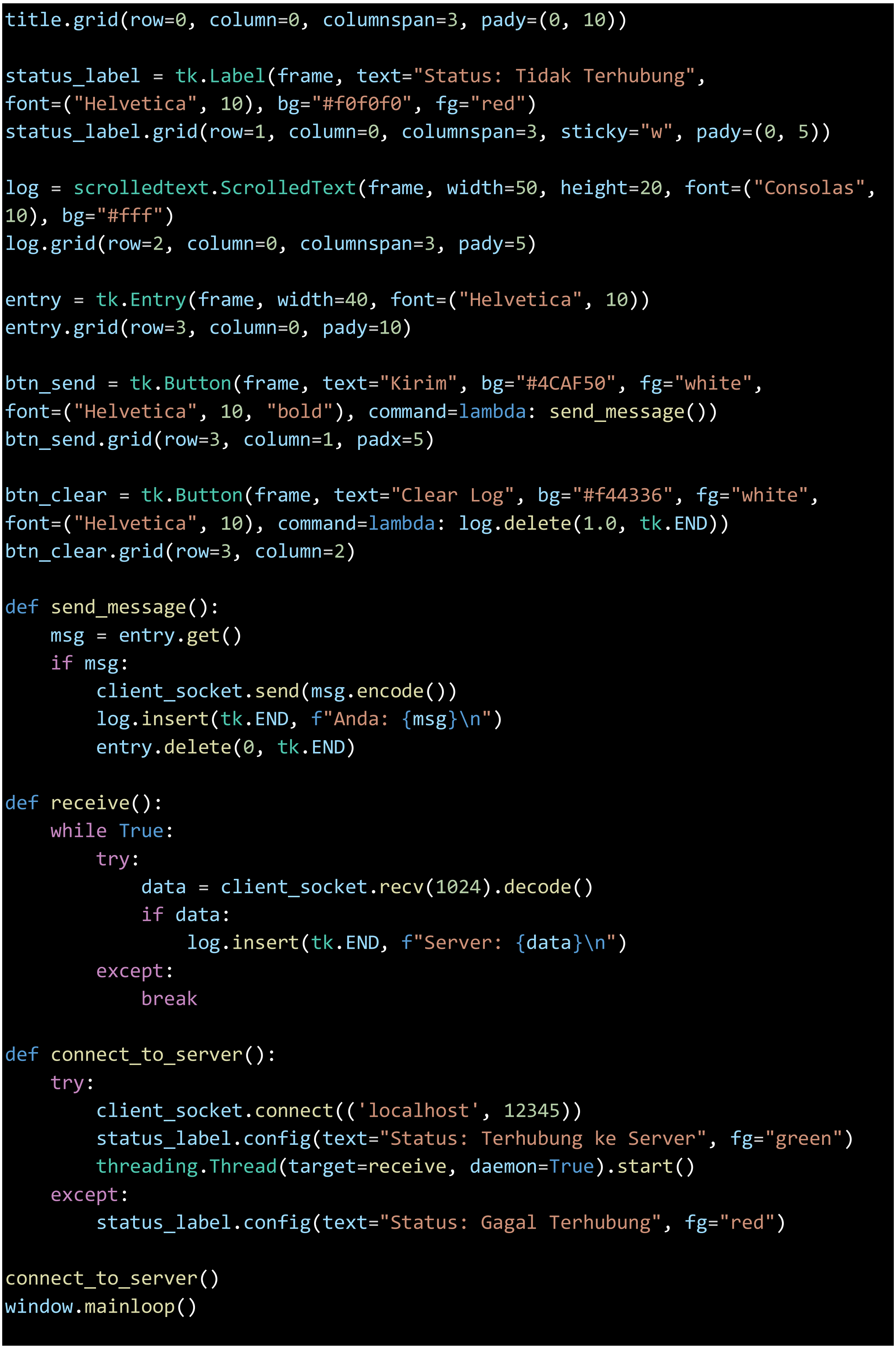
Source Code TCP Server

```
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))
```

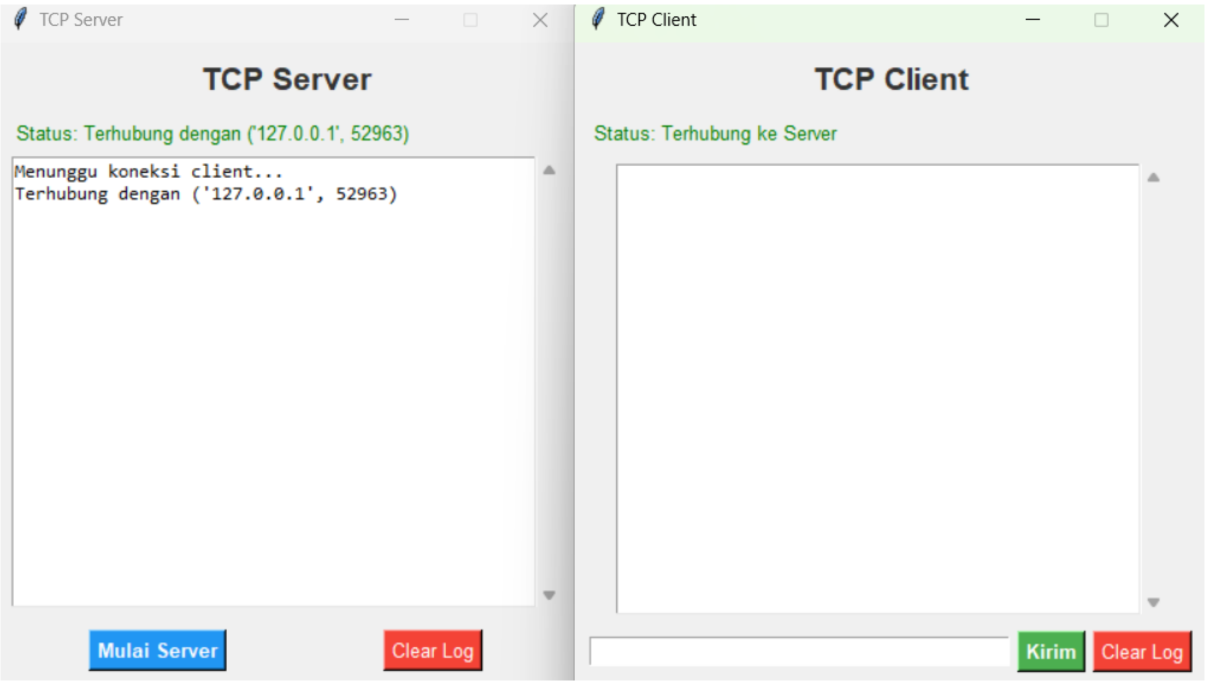
```
btn_clear.grid(row=3, column=1, pady=10)

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()
```

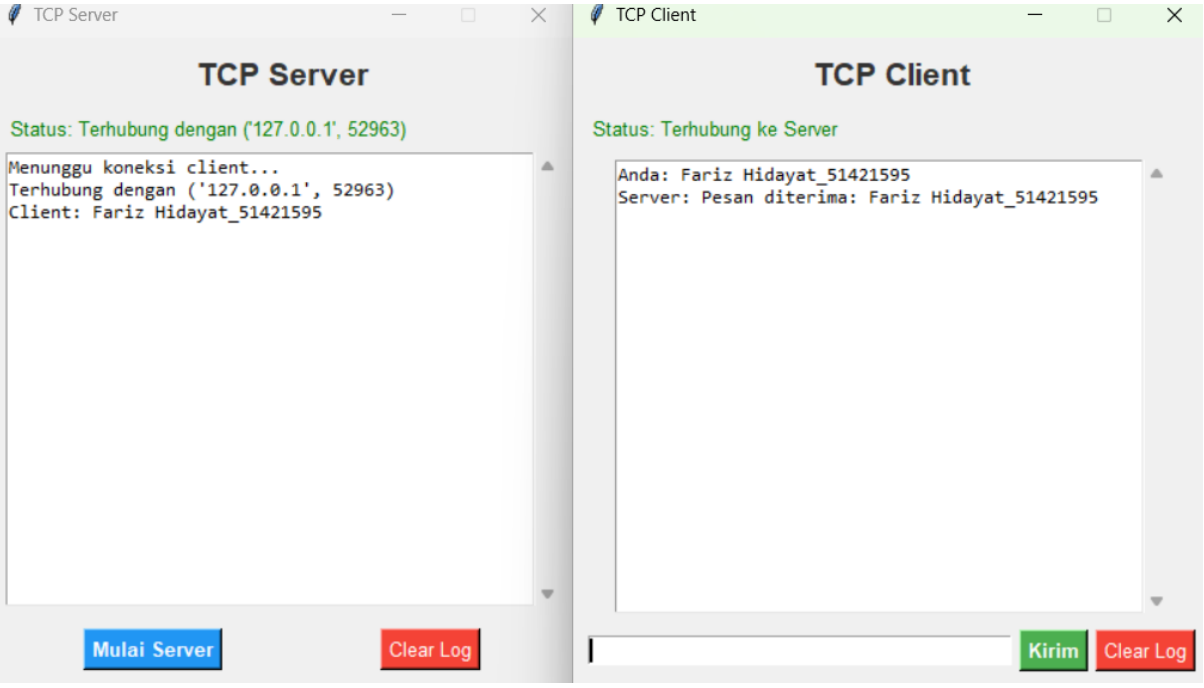
Output



Status: Terhubung dengan ('127.0.0.1', 52963)

```
Menunggu koneksi client...
Terhubung dengan ('127.0.0.1', 52963)
```

Status: Terhubung ke Server



TCP Server

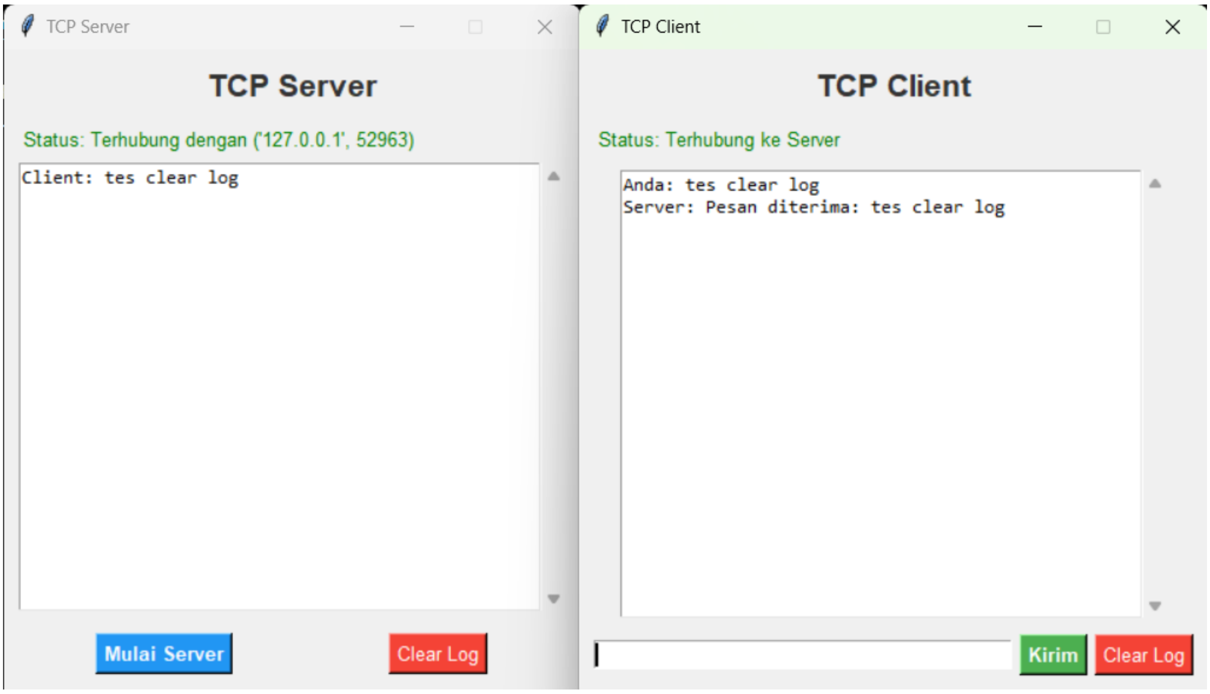
Status: Terhubung dengan ('127.0.0.1', 52963)

```
Menunggu koneksi client...
Terhubung dengan ('127.0.0.1', 52963)
Client: Fariz Hidayat_51421595
```

TCP Client

Status: Terhubung ke Server

```
Anda: Fariz Hidayat_51421595
Server: Pesan diterima: Fariz Hidayat_51421595
```



TCP Server

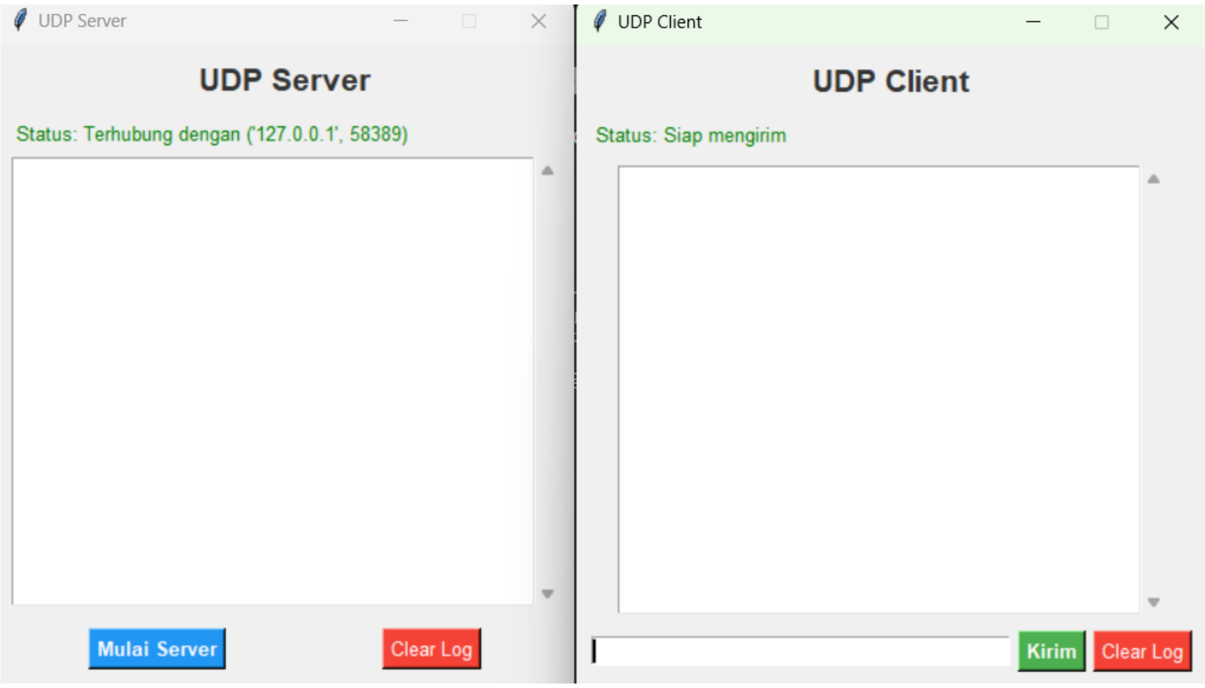
Status: Terhubung dengan ('127.0.0.1', 52963)

```
Client: tes clear log
```

TCP Client

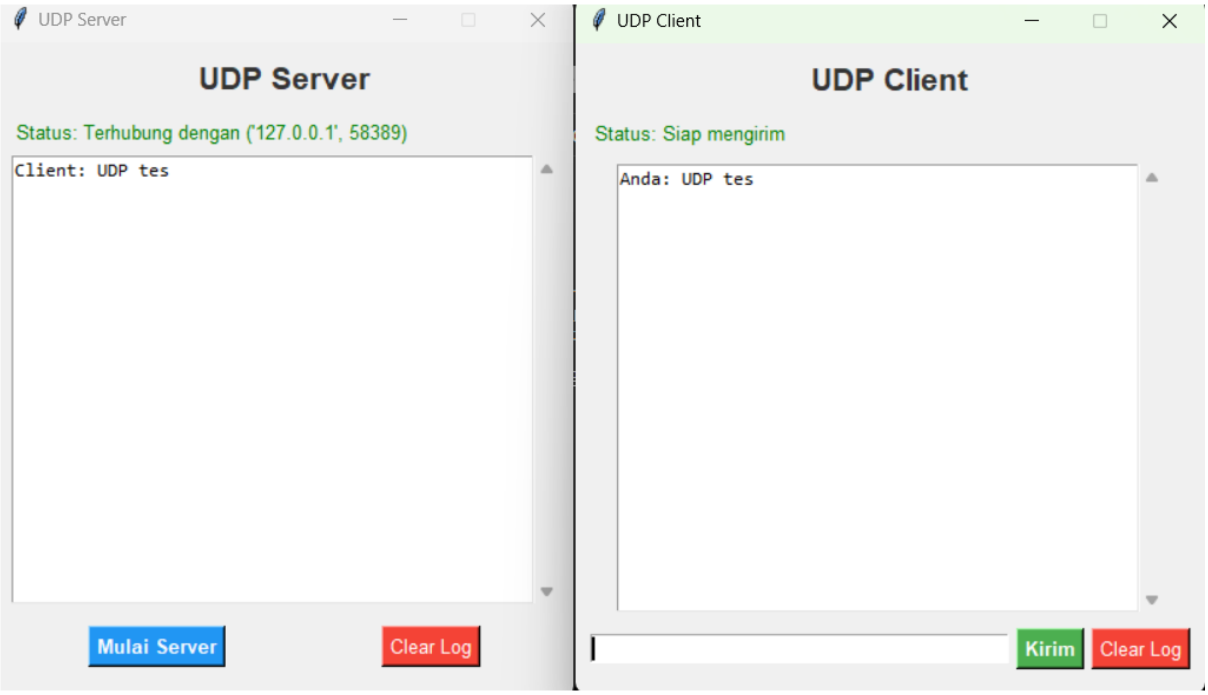
Status: Terhubung ke Server

```
Anda: tes clear log
Server: Pesan diterima: tes clear log
```



Status: Terhubung dengan ('127.0.0.1', 58389)

Status: Siap mengirim



Status: Terhubung dengan ('127.0.0.1', 58389)

```
Client: UDP tes
```

Status: Siap mengirim

```
Anda: UDP tes
```

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.

2. 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.

3. 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.

4. 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.

5. 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