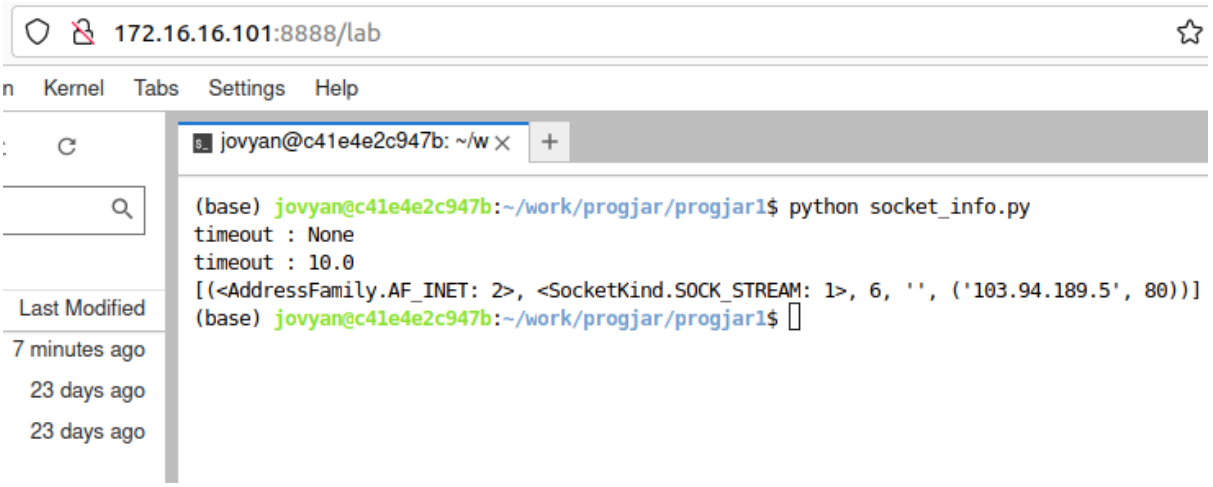
Antonio Taifan Montana
5025201219

TUGAS 1 PEMROGRAMAN JARINGAN (C)

[Shagdovala159/Tugas-1-Pemrograman-Jaringan-C (github.com)](https://github.com/Shagdovala159/Tugas-1-Pemrograman-Jaringan-C)

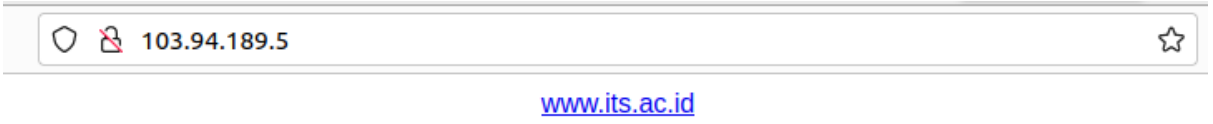
jalankan socket_info.py di mesin-1 dan mesin-2, capturelah hasilnya

Pada Mesin 1 :



Pada Mesin 2 :

Jika IP Address itu diakses maka akan merujuk ke www.its.ac.id

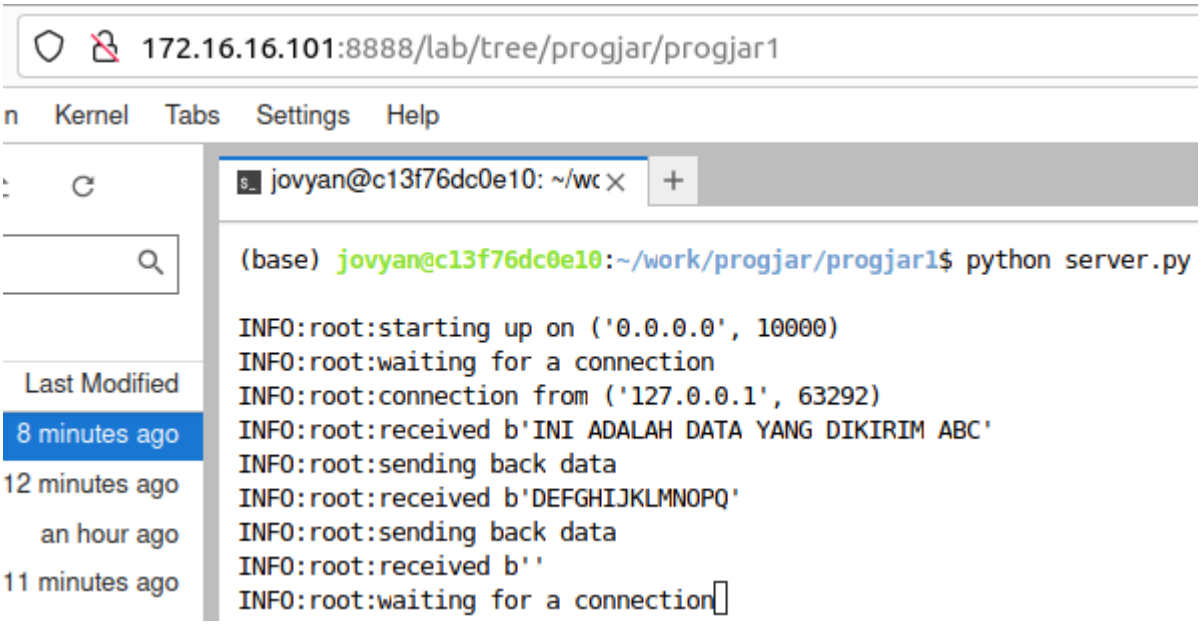


www.its.ac.id

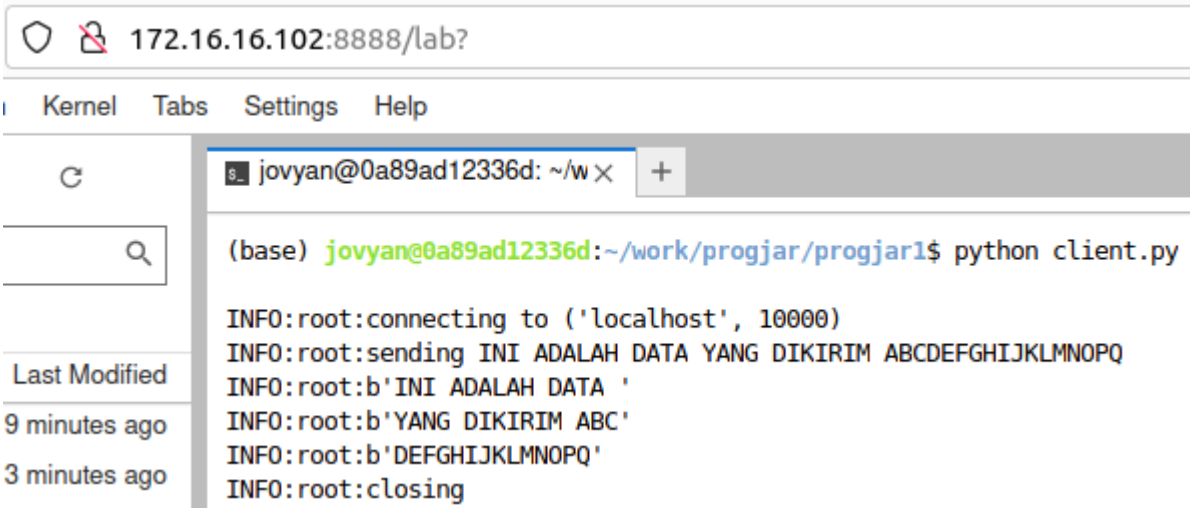
Antonio Taifan Montana
5025201219

jalankan server.py di mesin-1 dan client.py di mesin-2, sesuaikan isi program, pastikan komunikasi dapat dilakukan, capturelah hasilnya

Pada mesin 1 :



Pada mesin 2 :



Dapat terlihat bahwa komunikasi dapat terjalin karena server dan client menggunakan port localhost yang sama. Client akan mengirimkan datanya secara bertahap untuk mencegah buffer overflow. Namun, hal ini tidak terlalu kritis karena saya mengubah ukuran buffer server menjadi lebih besar daripada buffer client.

Antonio Taifan Montana
5025201219

jalankan kembali soal nomor 3, namun kali ini rubahlah komunikasi agar berjalan di port 32444, kirimkan isi sebuah file, dan capturelahh hasilnya

Parameter instasiasi socket diubah nilai port agar sesuai. Selain itu dibuatkan file pengujian bernama testing.txt pada working directory. Client membaca dari file bernama testing.txt dengan perubahan kode seperti berikut:

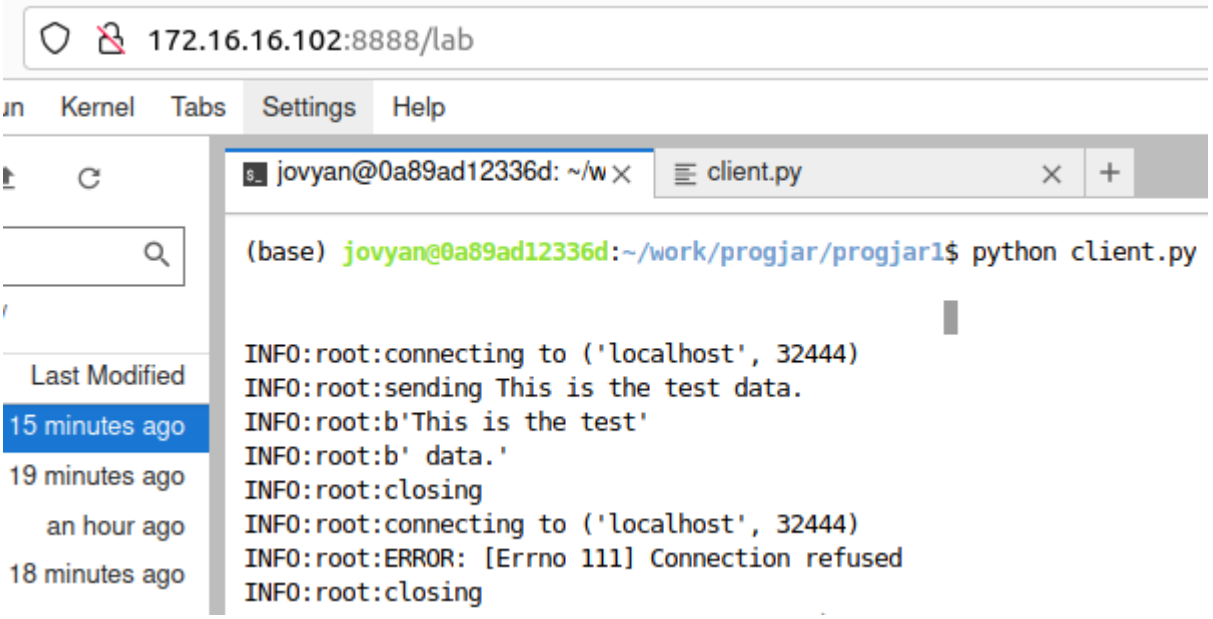
```
# Send data
f = open("testing.txt", "r")
message = f.read()
```

Berikut output terminal server.py:

```
(base) jovyan@c13f76dc0e10:~/work/progjar/progjar1$ python server.py


INFO:root:starting up on ('0.0.0.0', 32444)
INFO:root:waiting for a connection
INFO:root:connection from ('127.0.0.1', 63937)
INFO:root:received b'This is the test data.'
INFO:root:sending back data
INFO:root:received b''
INFO:root:waiting for a connection
INFO:root:ERROR: timed out
INFO:root:closing
```

Berikut output client.py:



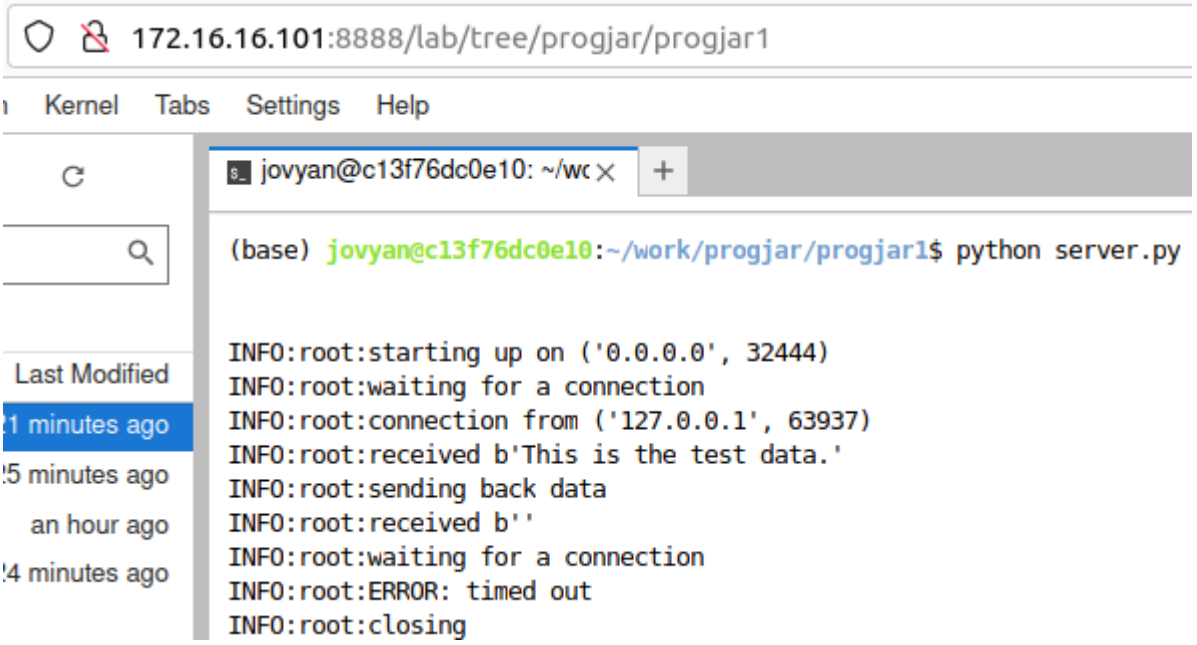
Antonio Taifan Montana
5025201219

Jalankan client di mesin-3 dan mesin-4 dengan server berada di mesin-1, jalankan client secara bersamaan, apakah yang terjadi ? capturelah hasilnya

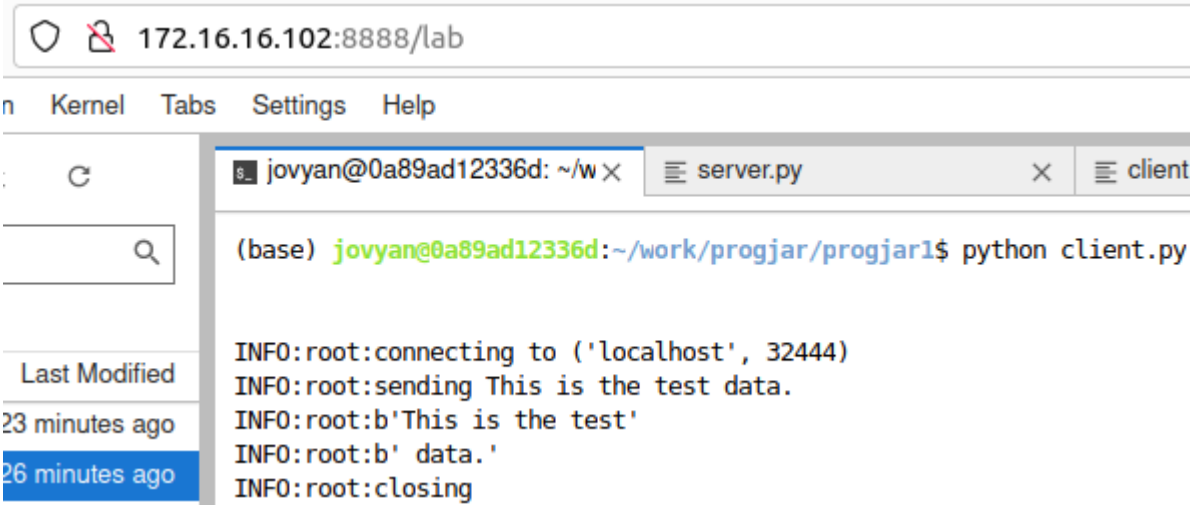
Ditambahkan fungsi sleep() untuk memberi waktu menjalankan beberapa client seperti berikut:

```
while True:
    # Wait for a connection
    logging.info("waiting for a connection")
    connection, client_address = sock.accept()
    logging.info(f"connection from {client_address}")
    import time
    time.sleep(10)
```

Output dengan satu client:
server.py

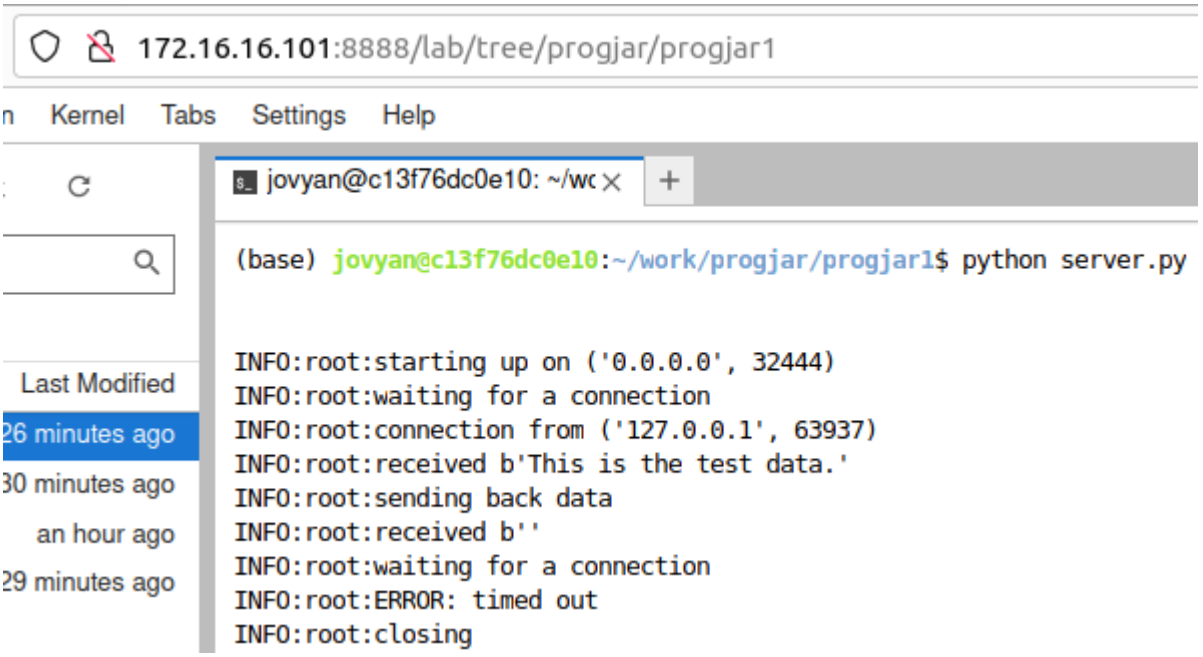


Client.py

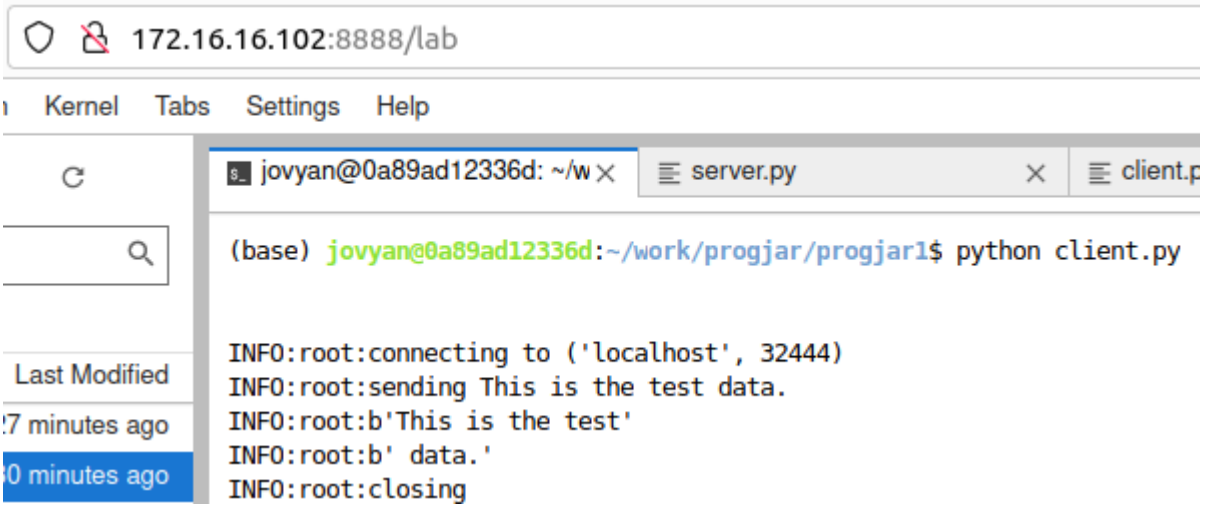


Antonio Taifan Montana
5025201219

Output dengan dua client:
Server.py

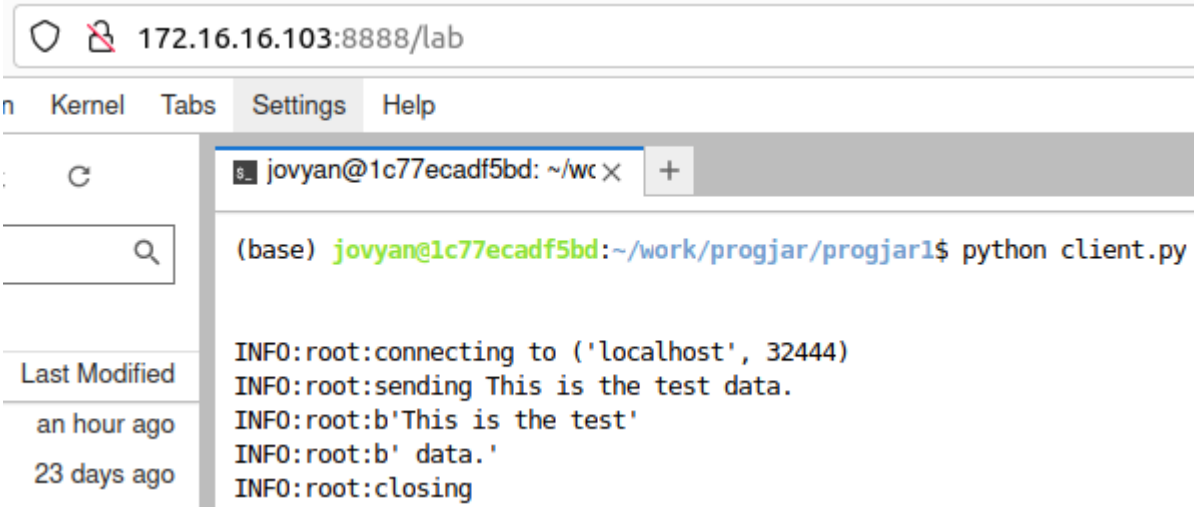


client.py (Client pertama)



Antonio Taifan Montana
5025201219

client.py (Client kedua)



Karena socket hanya listen untuk 1 koneksi, maka request kedua tidak diterima sampai koneksi client pertama putus. Meskipun kedua client pada akhirnya berhasil mendapatkan data, mereka harus melakukannya secara sekuensial, tidak bisa secara konkuren.
Karena socket hanya listen untuk 1 koneksi, maka request kedua tidak diterima sampai koneksi client pertama putus. Meskipun kedua client pada akhirnya berhasil mendapatkan data, mereka harus melakukannya secara sekuensial, tidak bisa secara konkuren.