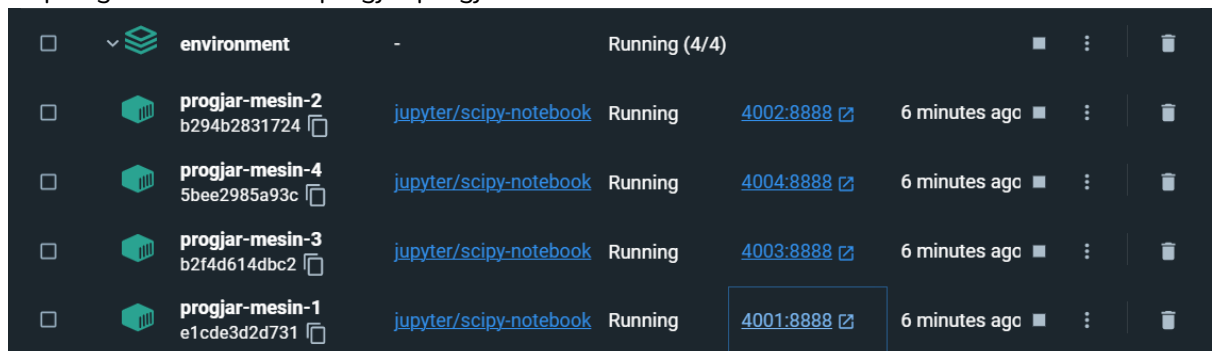


Tugas 1

1. Jalankan lab sesuai panduan, dan gunakan contoh program di <https://github.com/rm77/progjar/progjar1>



Menjalankan docker container menggunakan docker desktop. Namun, pada akhirnya saya memutuskan untuk menjalankan pengujian server pada localhost windows. Saya menggunakan VS Code dan Terminal Windows untuk menjalankan program python server dan client.

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

```
timeout : None
timeout : 10.0

[(<AddressFamily.AF_INET: 2>, <SocketKind.SOCK_STREAM: 1>, 6, '',
('103.94.189.5', 80)),
(<AddressFamily.AF_INET6: 10>, <SocketKind.SOCK_STREAM: 1>, 6, '',
('::ffff:103.94.189.5', 80, 0, 0))]
```

Program socket_info.py merupakan program yang mencetak informasi mengenai parameter socket. Parameter digunakan untuk menginstansiasikan socket. IP Address yang didapatkan merupakan IP dari domain www.its.ac.id dengan port 80. Karena akan menggunakan internet, address family yang digunakan AF_INET. Selain itu, karena menggunakan TCP, digunakan tipe socket SOCK_STREAM.

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

server.py

```
C:\Users\User\Documents\CompSci\UNI\SM-6\Progjar\progjar\progjar1>python server.py

INFO:root:starting up on ('0.0.0.0', 10000)
```

```
INFO:root:waiting for a connection
INFO:root:connection from ('127.0.0.1', 63292)
INFO:root:received b'INI ADALAH DATA YANG DIKIRIM ABC'
INFO:root:sending back data
INFO:root:received b'DFGHIJKLMNOPQ'
INFO:root:sending back data
INFO:root:received b''
INFO:root:waiting for a connection
```

client.py

```
C:\Users\User\Documents\CompSci\UNI\SM-6\Progjar\progjar\progjar1>python client.py

INFO:root:connecting to ('localhost', 10000)
INFO:root:sending INI ADALAH DATA YANG DIKIRIM ABCDEFGHIJKLMNOPQ
INFO:root:b'INI ADALAH DATA '
INFO:root:b'YANG DIKIRIM ABC'
INFO:root:b'DFGHIJKLMNOPQ'
INFO:root:closing
```

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.

4. jalankan kembali soal nomor 3, namun kali ini rubahlah komunikasi agar berjalan di port 32444, kirimkan isi sebuah file, dan capturelah 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()
# message = 'INI ADALAH DATA YANG DIKIRIM ABCDEFGHIJKLMNOPQ'
```

Berikut output terminal server.py:

```
C:\Users\User\Documents\CompSci\UNI\SM-6\Progjar\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:

```
C:\Users\User\Documents\CompSci\UNI\SM-6\Progjar\progjar\progjar1>python server.py

INFO:root:connecting to ('localhost', 32444)
INFO:root:sending This is the test data.
INFO:root:b'This is the test'
INFO:root:b' data.'
INFO:root:closing
```

5. 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

```
C:\Users\User\Documents\CompSci\UNI\SM-6\Progjar\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
```

client.py

```
C:\Users\User\Documents\CompSci\UNI\SM-6\Progjar\progjar\progjar1>python server.py

INFO:root:connecting to ('localhost', 32444)
INFO:root:sending This is the test data.
INFO:root:b'This is the test'
INFO:root:b' data.'
INFO:root:closing
```

Output dengan dua client:

server.py

```
C:\Users\User\Documents\CompSci\UNI\SM-6\Progjar\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
```

client.py (Client pertama)

```
C:\Users\User\Documents\CompSci\UNI\SM-6\Progjar\progjar\progjar1>python server.py

INFO:root:connecting to ('localhost', 32444)
INFO:root:sending This is the test data.
INFO:root:b'This is the test'
INFO:root:b' data.'
INFO:root:closing
```

client.py (Client kedua)

```
C:\Users\User\Documents\CompSci\UNI\SM-6\Progjar\progjar\progjar1>python server.py

INFO:root:connecting to ('localhost', 32444)
INFO:root:sending This is the test data.
INFO:root:b'This is the test'
INFO:root:b' data.'
INFO:root:closing
```

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.