Nama : Nixon Castroman

NRP : 5025231024

Kelas : Pemrograman Jaringan D

Soal 1

(mesin1)

```
(base) jovyan@270a800520fe:~/work/progjar/progjar1$ python3 socket_info.py
timeout : None
timeout : 10.0
[(<AddressFamily.AF_INET: 2>, <SocketKind.SOCK_STREAM: 1>, 6, '', ('103.94.189.4', 80))]
```

20450	282.417433626	172.18.0.4	172.18.0.1	TCP	11650	6666 → 60664 [PSH, ACK] Seq=16784021 Ack=7141
20451	282.417505937	172.18.0.4	172.18.0.1	TCP	3382	6666 → 60664 [PSH, ACK] Seq=16795605 Ack=7141
20452	282.417577385	172.18.0.1	172.18.0.4	TCP	66	60664 → 6666 [ACK] Seq=71413 Ack=16798921 Win
20453	282.417811611	172.18.0.4	172.18.0.1	TCP	6005	6666 → 60664 [PSH, ACK] Seq=16798921 Ack=7141
20454	282.417919060	172.18.0.1	172.18.0.4	TCP	66	60664 → 6666 [ACK] Seq=71413 Ack=16804860 Win
20455	282.422318071	172.18.0.1	172.18.0.4	TCP	82	60664 → 6666 [PSH, ACK] Seq=71413 Ack=1680486
20456	282.464538234	172.18.0.4	172.18.0.1	TCP	72	6666 → 60664 [PSH, ACK] Seq=16804860 Ack=7142
20457	282.464655753	172.18.0.4	172.18.0.1	TCP	1282	[TCP segment of a reassembled PDU]
20458	282.464770747	172.18.0.1	172.18.0.4	TCP	66	60664 → 6666 [ACK] Seq=71429 Ack=16806082 Win
20459	282.466775869	172.18.0.1	172.18.0.4	TCP	82	60664 → 6666 [PSH, ACK] Seq=71429 Ack=1680608
20460	282.525237120	172.18.0.4	172.18.0.1	TCP	66	6666 → 60664 [ACK] Seq=16806082 Ack=71445 Win

(mesin2)

```
(base) jovyan@08bd1e100aa0:~/work/progjar/progjar1$ python3 socket_info.py
timeout : None
timeout : 10.0
[(<AddressFamily.AF_INET: 2>, <SocketKind.SOCK_STREAM: 1>, 6, '', ('103.94.189.4', 80))]
```

No.	Time	Source	Destination	Protocol	Length	Info
899	5.499570148	172.18.0.2	172.18.0.1	TCP	6512	6666 → 38672 [PSH, ACK] Seq=627610 Ack=4621 Win=50
900	5.499693391	172.18.0.1	172.18.0.2	TCP	66	38672 → 6666 [ACK] Seq=4621 Ack=634056 Win=512 Len
901	5.499716796	172.18.0.2	172.18.0.1	TCP	969	6666 → 38672 [PSH, ACK] Seq=634056 Ack=4621 Win=50
902	5.499766193	172.18.0.2	172.18.0.1	TCP	2533	6666 → 38672 [PSH, ACK] Seq=634959 Ack=4621 Win=50
903	5.499803487	172.18.0.1	172.18.0.2	TCP	66	38672 → 6666 [ACK] Seq=4621 Ack=637426 Win=512 Len
904	5.500293630	172.18.0.2	172.18.0.1	TCP	5879	6666 → 38672 [PSH, ACK] Seq=637426 Ack=4621 Win=50
905	5.500475156	172.18.0.1	172.18.0.2	TCP	66	38672 → 6666 [ACK] Seq=4621 Ack=643239 Win=512 Len
906	5.504328431	172.18.0.1	172.18.0.2	TCP	82	38672 → 6666 [PSH, ACK] Seq=4621 Ack=643239 Win=51
907	5.559442345	172.18.0.2	172.18.0.1	TCP	66	6666 → 38672 [ACK] Seq=643239 Ack=4637 Win=501 Len
908	5.624457437	172.18.0.1	172.18.0.2	TCP	78	38672 → 6666 [PSH, ACK] Seq=4637 Ack=643239 Win=51
909	5.624582702	172.18.0.2	172.18.0.1	TCP	66	6666 → 38672 [ACK] Seq=643239 Ack=4649 Win=501 Len

Saya menjalankan program **socket_info.py** di mesin1 dan mesin2. Program ini membuka koneksi socket menggunakan protokol TCP ke alamat **IP 103.94.189.4** pada port **80** dengan timeout **10 detik**. Saya melakukan capture lalu lintas jaringan menggunakan Wireshark pada interface **eth0**. Hasil capture menunjukkan adanya paket TCP yang saling bertukar data antara port lokal dan port **6666** pada IP masing-masing mesin. Paket-paket ini mengandung flag **PSH** dan **ACK** yang menandakan data sedang dikirim dan diterima secara aktif. Hal ini membuktikan bahwa program socket berhasil melakukan koneksi dan komunikasi data menggunakan TCP.

Soal 2

Kita mengubah server address **client.py** menjadi ip address mesin 1 agar bisa berkomunikasi.

```
server_address = ('172.16.16.101', 10000)
```

(mesin1)

```
(base) jovyan@270a800520fe:~/work/progjar/progjar1$ python3 server.py
INFO:root:starting up on ('0.0.0.0', 10000)
INFO:root:waiting for a connection
INFO:root:connection from ('172.16.16.102', 38090)
INFO:root:received b'INI ADALAH DATA YANG DIKIRIM ABC'
INFO:root:sending back data
INFO:root:received b'DEFGHIJKLMNOPQ'
INFO:root:sending back data
INFO:root:received b''
INFO:root:waiting for a connection
```

	Time	Source	Destination	Protocol	Length	Info
2	0.000020280	172.16.16.101	172.16.16.102	TCP	74	10000 → 57278 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 M
3	0.000090976	172.16.16.102	172.16.16.101	TCP	66	57278 → 10000 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=
4	0.000214865	172.16.16.102	172.16.16.101	TCP	112	57278 → 10000 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=46
5	0.000220826	172.16.16.101	172.16.16.102	TCP	66	10000 → 57278 [ACK] Seq=1 Ack=47 Win=65152 Len=0 TSval=
6	0.000968662	172.16.16.101	172.16.16.102	TCP	98	10000 → 57278 [PSH, ACK] Seq=1 Ack=47 Win=65152 Len=32
7	0.001054006	172.16.16.102	172.16.16.101	TCP	66	57278 → 10000 [ACK] Seq=47 Ack=33 Win=64256 Len=0 TSval
8	0.001258350	172.16.16.101	172.16.16.102	TCP	80	10000 → 57278 [PSH, ACK] Seq=33 Ack=47 Win=65152 Len=1
9	0.001286264	172.16.16.102	172.16.16.101	TCP	66	57278 → 10000 [ACK] Seq=47 Ack=47 Win=64256 Len=0 TSval
10	0.001529974	172.16.16.102	172.16.16.101	TCP	66	57278 → 10000 [FIN, ACK] Seq=47 Ack=47 Win=64256 Len=0
11	0.001730507	172.16.16.101	172.16.16.102	TCP	66	10000 → 57278 [FIN, ACK] Seq=47 Ack=48 Win=65152 Len=0
12	0.001765996	172.16.16.102	172.16.16.101	TCP	66	57278 → 10000 [ACK] Seq=48 Ack=48 Win=64256 Len=0 TSval

(mesin2)

```
(base) jovyan@08bd1e100aa0:~/work/progjar/progjar1$ python3 client.py
INFO:root:connecting to ('172.16.16.101', 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'DEFGHIJKLMNOPQ'
INFO:root:closing
```

1. **Topologi**:
    - Mesin1 (Server): **IP 172.16.16.101**
    - Mesin2 (Client): **IP 172.16.16.102**
    - Interface: **eth1** (jaringan antar mesin)

- Port: **10000 TCP**

2. **Proses Koneksi**:
   - Client menginisiasi koneksi ke Server pada port **10000**
   - Terjadi handshake **TCP (SYN, SYN-ACK, ACK)** berhasil
   - Client mengirim pesan **"INI ADALAH DATA…"**
   - Server menerima data dan menutup koneksi dengan benar
3. **Analisis Wireshark**:
   - Paket data tampil pada interface eth1 sesuai IP dan port
   - Tidak ada paket retransmission, berarti koneksi stabil
   - Durasi koneksi singkat dan efisien
4. **Kesimpulan**:
   Komunikasi client-server berjalan lancar dan sesuai skenario tugas pemrograman jaringan.

Soal 3

```
server_address = ('172.16.16.101', 32444)
```

Kita mengubah port server address pada client dan juga server ke **32444**, sesuai arahan soal

Lalu kita mengubah kode agar bisa mengirimkan isi file

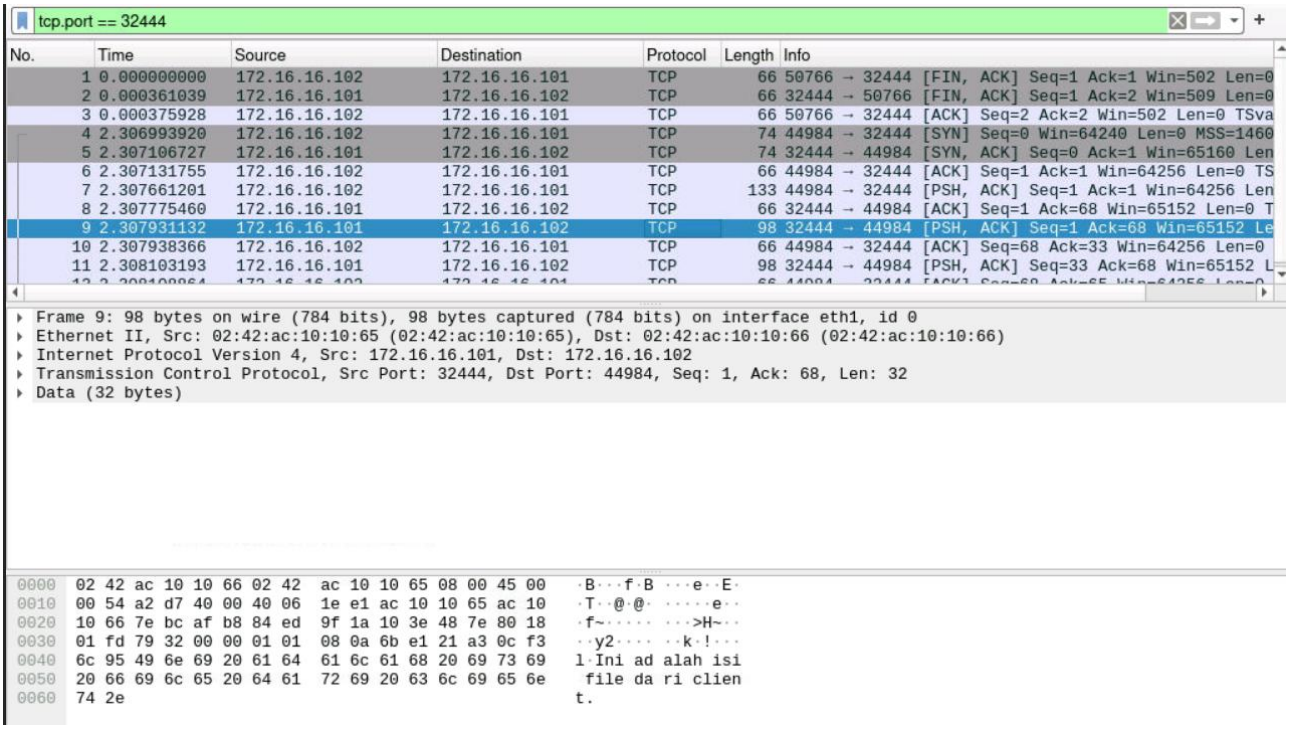
```python
import socket
import logging

logging.basicConfig(level=logging.INFO)

try:
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_address = ('172.16.16.101', 32444)  # IP Server
    logging.info(f"connecting to {server_address}")
    sock.connect(server_address)

    # Baca isi file yang mau dikirim
    filename = 'data.txt'
    with open(filename, 'rb') as f:
        message = f.read()

    logging.info(f"sending file content ({len(message)} bytes)")
    sock.sendall(message)

    # Terima balasan dari server
    amount_received = 0
    while True:
        data = sock.recv(32)
        if not data:
            break
        amount_received += len(data)
        logging.info(f"received: {data}")

except Exception as ee:
    logging.info(f"ERROR: {str(ee)}")
finally:
    logging.info("closing")
    sock.close()
```

(mesin1)

```
(base) jovyan@270a800520fe:~/work/progjar/progjar1$ python3 ser
ver.py
INFO:root:starting up on ('0.0.0.0', 32444)
INFO:root:waiting for a connection
INFO:root:connection from ('172.16.16.102', 50766)
INFO:root:received b'Ini adalah isi file dari client.'
INFO:root:sending back data
INFO:root:received b'\nFile ini akan dikirim via socke'
INFO:root:sending back data
INFO:root:received b't.\n'
INFO:root:sending back data
```

(mesin2)

```
(base) jovyan@08bd1e100aa0:~/work/progjar/progjar1$ python3 cli
ent.py
INFO:root:connecting to ('172.16.16.101', 32444)
INFO:root:sending file content (67 bytes)
INFO:root:received: b'Ini adalah isi file dari client.'
INFO:root:received: b'\nFile ini akan dikirim via socke'
INFO:root:received: b't.\n'
```



1. Perubahan Konfigurasi Port

- Port default pada program sebelumnya diubah menjadi **32444**, sesuai instruksi soal.
- Port **32444** digunakan oleh TCP untuk mentransmisikan file antara client dan server.

2. Isi File Terkirim

- Isi file (**data.txt**) berhasil dibaca oleh client dan dikirim ke server.
- Server menerima isi file dan mengirim balik (echo) ke client.
- Client menampilkan isi file yang diterima kembali, sesuai log di terminal.

3. Analisis Wireshark

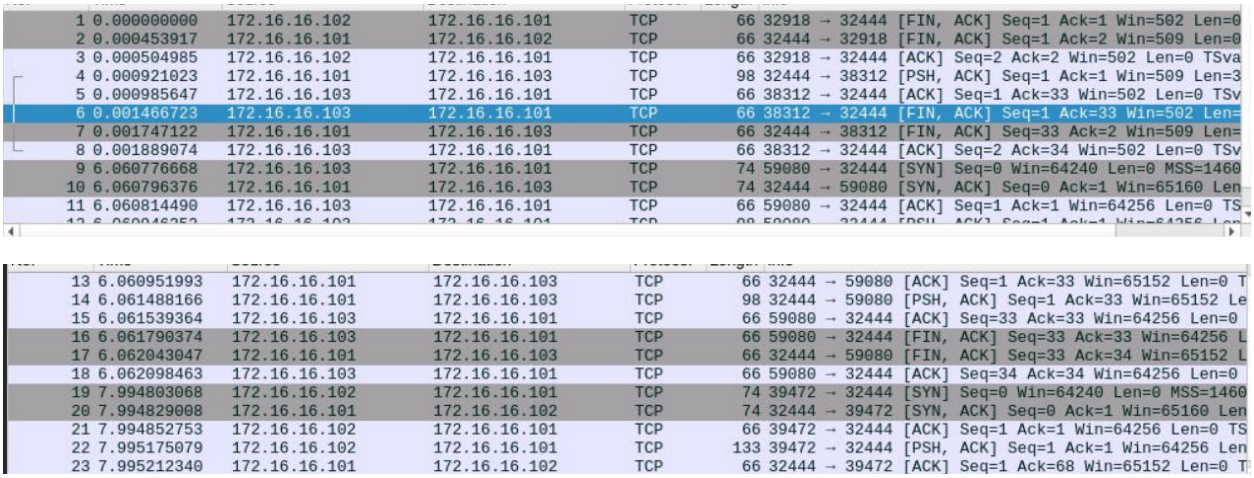
Dari tangkapan Wireshark:

- Filter **tcp.port == 32444** digunakan untuk memantau lalu lintas data di port tersebut.
- Terlihat adanya komunikasi dari:
    - **Source:** 172.16.16.101 (client)
    - **Destination:** 172.16.16.102 (server)
- Di frame yang disorot:
    - Terdapat payload (Data: 32 bytes), yang menunjukkan bagian isi file terkirim.
    - Baris hex (49 6e 69 20 61 64 61 6c 61 68 20 69 73 69...) terjemahannya: **Ini adalah isi dari client**

4. Protokol dan Flags

- Terdapat beberapa paket dengan **PSH, ACK** flags:
    - Artinya data dikirim langsung tanpa buffering tambahan (push).
- Paket dengan flag **FIN, ACK** menandakan koneksi TCP ditutup secara normal.

Soal 4



1	0.000000000	172.16.16.102	172.16.16.101	TCP	66	32918 → 32444 [FIN, ACK] Seq=1 Ack=1 Win=502 Len=0
2	0.000453917	172.16.16.101	172.16.16.102	TCP	66	32444 → 32918 [FIN, ACK] Seq=1 Ack=2 Win=509 Len=0
3	0.000504985	172.16.16.102	172.16.16.101	TCP	66	32918 → 32444 [ACK] Seq=2 Ack=2 Win=502 Len=0 TSva
4	0.000921023	172.16.16.101	172.16.16.103	TCP	98	32444 → 38312 [PSH, ACK] Seq=1 Ack=1 Win=509 Len=3
5	0.000985647	172.16.16.103	172.16.16.101	TCP	66	38312 → 32444 [ACK] Seq=1 Ack=33 Win=502 Len=0 TSv
6	0.001466723	172.16.16.103	172.16.16.101	TCP	66	38312 → 32444 [FIN, ACK] Seq=1 Ack=33 Win=502 Len=
7	0.001747122	172.16.16.101	172.16.16.103	TCP	66	32444 → 38312 [FIN, ACK] Seq=33 Ack=2 Win=509 Len=
8	0.001889074	172.16.16.103	172.16.16.101	TCP	66	38312 → 32444 [ACK] Seq=2 Ack=34 Win=502 Len=0 TSv
9	6.060776668	172.16.16.103	172.16.16.101	TCP	74	59080 → 32444 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
10	6.060796376	172.16.16.101	172.16.16.103	TCP	74	32444 → 59080 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len
11	6.060814490	172.16.16.103	172.16.16.101	TCP	66	59080 → 32444 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TS
13	6.060951993	172.16.16.101	172.16.16.103	TCP	66	32444 → 59080 [ACK] Seq=1 Ack=33 Win=65152 Len=0 T
14	6.061488166	172.16.16.101	172.16.16.103	TCP	98	32444 → 59080 [PSH, ACK] Seq=1 Ack=33 Win=65152 Le
15	6.061539364	172.16.16.103	172.16.16.101	TCP	66	59080 → 32444 [ACK] Seq=33 Ack=33 Win=64256 Len=0
16	6.061790374	172.16.16.103	172.16.16.101	TCP	66	59080 → 32444 [FIN, ACK] Seq=33 Ack=33 Win=64256 L
17	6.062043047	172.16.16.101	172.16.16.103	TCP	66	32444 → 59080 [FIN, ACK] Seq=33 Ack=34 Win=65152 L
18	6.062098463	172.16.16.103	172.16.16.101	TCP	66	59080 → 32444 [ACK] Seq=34 Ack=34 Win=64256 Len=0
19	7.994803068	172.16.16.102	172.16.16.101	TCP	74	39472 → 32444 [SYN] Seq=0 Win=64240 Len=0 MSS=1460
20	7.994829008	172.16.16.101	172.16.16.102	TCP	74	32444 → 39472 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len
21	7.994852753	172.16.16.102	172.16.16.101	TCP	66	39472 → 32444 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TS
22	7.995175079	172.16.16.102	172.16.16.101	TCP	133	39472 → 32444 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len
23	7.995212340	172.16.16.101	172.16.16.102	TCP	66	32444 → 39472 [ACK] Seq=1 Ack=68 Win=65152 Len=0 T

1. **Topologi:**
   - Server: **172.16.16.101 (mesin-1)**
   - Client 1: **172.16.16.102 (mesin-2)**
   - Client 2: **172.16.16.103 (mesin-3)**
2. **Eksperimen:**
   - Menjalankan kedua client secara bersamaan menuju server.
3. **Hasil Wireshark:**
   - Tampak dua ***3-way handshake***:
     - Mesin-3 ke Mesin-1 **(baris 9–11)**
     - Mesin-2 ke Mesin-1 **(baris 19–21)**
4. **Analisis:**
   - Koneksi TCP dapat dilakukan secara paralel oleh dua client ke satu server.
   - Server dapat menangani dua koneksi simultan, terlihat dari port tujuan **(32444)** yang sama, tapi port sumber berbeda (**39472** dan **38312**).
   - Ini menunjukkan **TCP mendukung multiple simultaneous connections** ke port yang sama, dibedakan oleh IP dan port sumber.