

## Tugas 2

1. Buatlah sebuah program time server dengan ketentuan sebagai berikut

- a) Membuka port di port 45000 dengan transport TCP (**server.py**)

```
def main():  
    svr = Server('localhost', 45000)  
    svr.start()  
  
class Server():  
    def __init__(self, ip, port):  
        self.ip = ip  
        self.port = port  
  
    def run(self):  
        self.my_socket.bind((self.ip, self.port))
```

- b) Server harus dapat melayani request yang concurrent, gunakan contoh multithreading (**server.py**)

```
class ProcessTheClient(threading.Thread):  
    def __init__(self, connection, address):  
        self.connection = connection  
        self.address = address  
        threading.Thread.__init__(self)  
  
    def run(self):  
        res=""  
        logging.warning("THIS IS THE RUN!")  
        while True:  
            try:  
                data = self.connection.recv(32)  
                if data:  
                    data = data.decode()  
                    if data[-2:] == '\r\n':  
                        now = datetime.datetime.now()  
                        now = now.time().strftime('%H:%M:%S\r\n')  
                        res = "JAM " + now  
                    self.connection.sendall(res.encode())  
                    res = ""  
            else:
```

```

        res = "ERR 404: {}".format(data)
        self.connection.sendall(res.encode())
        break
    self.connection.close()

except OSError as e:
    pass
self.connection.close()

```

- c) Request hanya dilayani dengan ketentuan
- Diawali dengan string "TIME dan diakhiri dengan karakter 13 dan karakter 10" (client.py)

```

# Create a TCP/IP socket
addr_fam = socket.AF_INET
stream = socket.SOCK_STREAM
sock = socket.socket(addr_fam, stream)

server_address = ('localhost', 45000)
logging.info(f"connecting to {server_address}")
sock.connect(server_address)

message = 'TIME\r\n'
logging.info(f"sending {message}")
sock.sendall(message.encode())

```

- d) Server akan merespon dengan jam dengan ketentuan
- Dalam bentuk string (UTF-8)

```

self.connection.sendall(res.encode())

```

- Diawali dengan "JAM<spasi><jam>"

```

res = "JAM " + now + '\r\n'

```

- <jam> berisikan info jam dalam format "hh:mm:ss" dan diakhiri dengan karakter 13 dan karakter 10

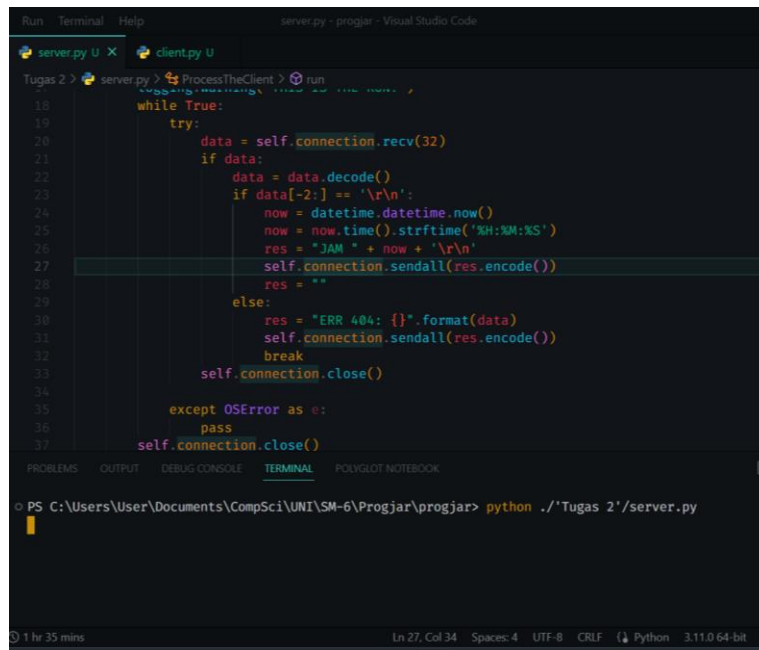
```

now = datetime.datetime.now()
now = time().strftime('%H:%M:%S')

```

## 2. Jalankan di lab environment

- Capturelah hasil eksekusi program server anda
  - Jalankan Server



The screenshot shows a Visual Studio Code editor with a file named `server.py` open. The code is a Python script for a simple server. It has a `while True:` loop that receives data from a client. If the data ends with `\r\n`, it logs the time and sends a response. Otherwise, it sends an error message. The terminal at the bottom shows the command `python ./Tugas 2/server.py` being executed.

```
Run Terminal Help server.py - progjar - Visual Studio Code
server.py U client.py U
Tugas 2 > server.py > ProcessTheClient > run
18 while True:
19     try:
20         data = self.connection.recv(32)
21         if data:
22             data = data.decode()
23             if data[-2:] == '\r\n':
24                 now = datetime.datetime.now()
25                 now = now.time().strftime('%H:%M:%S')
26                 res = "JAM " + now + '\r\n'
27                 self.connection.sendall(res.encode())
28                 res = ""
29             else:
30                 res = "ERR 404: {}".format(data)
31                 self.connection.sendall(res.encode())
32                 break
33             self.connection.close()
34         except OSError as e:
35             pass
36         self.connection.close()
37
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL POLYGLOT NOTEBOOK
PS C:\Users\User\Documents\CompSci\UNI\SM-6\Progjar\progjar> python ./Tugas 2/server.py
1 hr 35 mins Ln 27, Col 34 Spaces: 4 UTF-8 CRLF Python 3.11.0 64-bit
```

2. Request Valid

```
# Send data
message = 'TIME\r\n'
logging.info(f"sending {message}")
sock.sendall(message.encode())
```

3. Jalankan Client

```
C:\Users\User\Documents\CompSci\UNI\SM-6\Progjar
\progjar\Tugas 2>python client.py
INFO:root:connecting to ('localhost', 45000)
INFO:root:sending TIME

INFO:root:JAM 07:49:21

INFO:root:closing
```

4. Request Invalid

```
message = "AKSDMAIWJKENDALKJWN"
logging.info(f"sending {message}")
sock.sendall(message.encode())
```

5. Jalankan Client

```
C:\Users\User\Documents\CompSci\UNI\SM-6\Progjar
\progjar\Tugas 2>python client.py
INFO:root:connecting to ('localhost', 45000)
INFO:root:sending AKSDMAIWJKENDALKJWN
INFO:root:closing
```