

LAPORAN PRAKTIKUM PEMROGRAMAN JARINGAN



Di Susun Oleh :

Suci Wardawani (231401005)

**Mata Kuliah : Bahasa Pemrograman Jaringan Komputer
(DELPHI)**

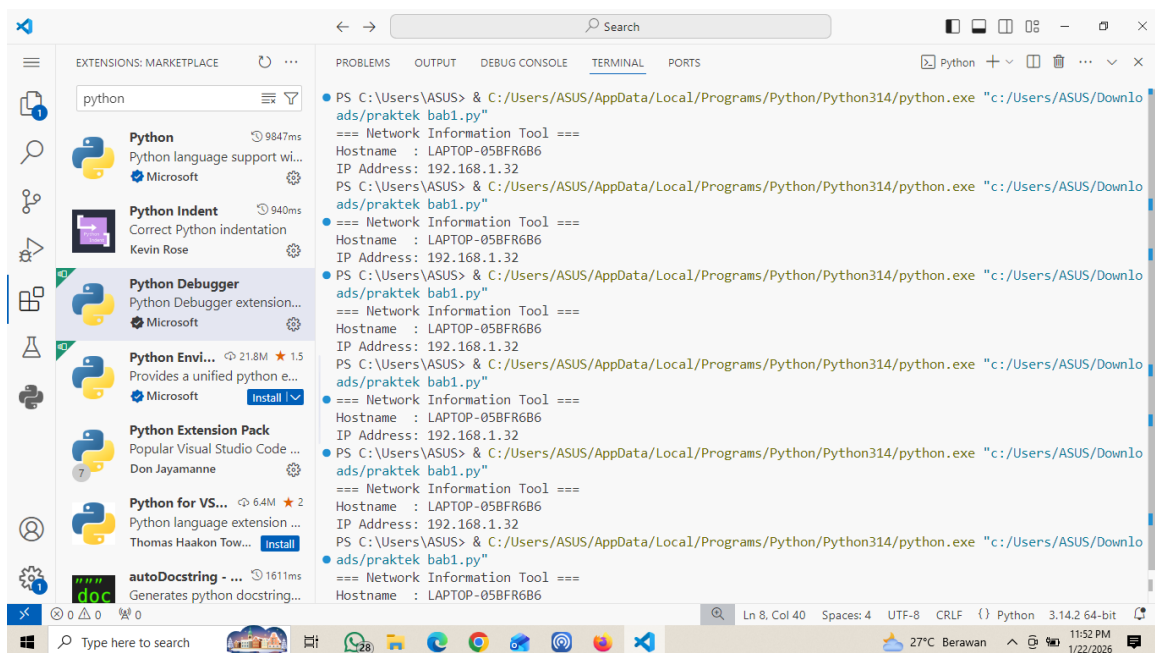
Dosen Pengampu : Ucok, S.Kom.,MT

**FAKULTAS ILMU KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA
UNIVERSITAS INDONESIA TIMUR 202**

Bab 1: Konsep Dasar Pemrograman Jaringan

Bab ini menjelaskan apa itu jaringan komputer dan pemrograman jaringan, termasuk bagaimana data bergerak dari satu komputer ke komputer lain lewat protokol TCP/IP, serta istilah-istilah penting seperti socket, port, IP address, *dan* protocol stack

Hasil :



The screenshot displays the Visual Studio Code interface. On the left, the 'EXTENSIONS: MARKETPLACE' sidebar is open, showing a search for 'python'. Several extensions are listed, including 'Python' by Microsoft, 'Python Indent' by Kevin Rose, 'Python Debugger' by Microsoft, 'Python Envi...' by Microsoft, 'Python Extension Pack' by Don Jayamanne, 'Python for VS...' by Thomas Haakon Tow..., and 'autoDocstring' by Microsoft. The main editor area shows a terminal window with the following output:

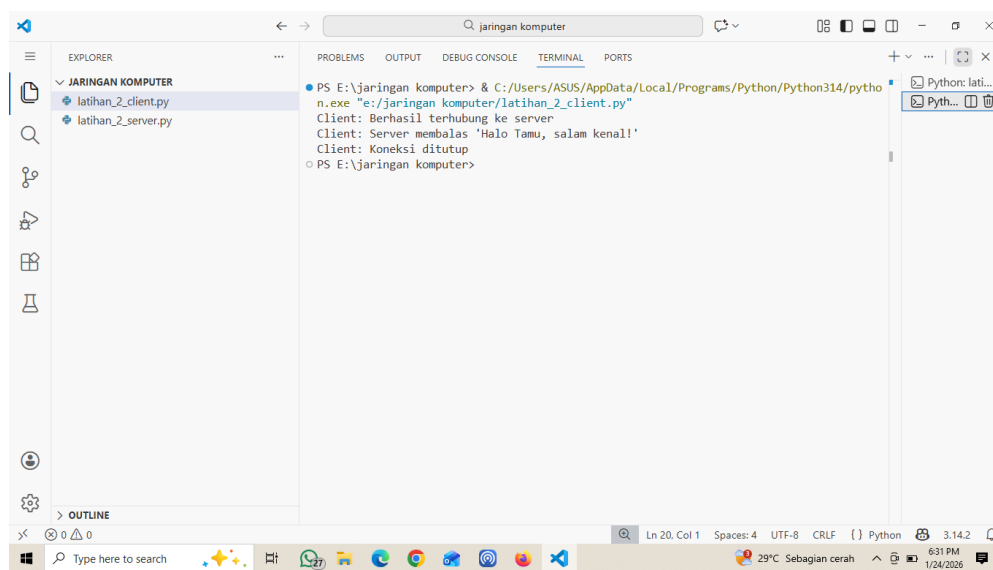
```
PS C:\Users\ASUS> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/ASUS/Downlo
ads/praktek bab1.py"
=== Network Information Tool ===
Hostname : LAPTOP-05BFR6B6
IP Address: 192.168.1.32
PS C:\Users\ASUS> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/ASUS/Downlo
ads/praktek bab1.py"
=== Network Information Tool ===
Hostname : LAPTOP-05BFR6B6
IP Address: 192.168.1.32
PS C:\Users\ASUS> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/ASUS/Downlo
ads/praktek bab1.py"
=== Network Information Tool ===
Hostname : LAPTOP-05BFR6B6
IP Address: 192.168.1.32
PS C:\Users\ASUS> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/ASUS/Downlo
ads/praktek bab1.py"
=== Network Information Tool ===
Hostname : LAPTOP-05BFR6B6
IP Address: 192.168.1.32
PS C:\Users\ASUS> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "c:/Users/ASUS/Downlo
ads/praktek bab1.py"
=== Network Information Tool ===
Hostname : LAPTOP-05BFR6B6
IP Address: 192.168.1.32
```

The status bar at the bottom indicates the current file is 'Ln 8, Col 40', using 'UTF-8' encoding, 'CRLF' line endings, and is a 'Python 3.14.2 64-bit' file. The system tray shows the date and time as '11:32 PM 1/22/2026' and the location as '27°C Berawan'.

Bab 2: Socket API Dasar

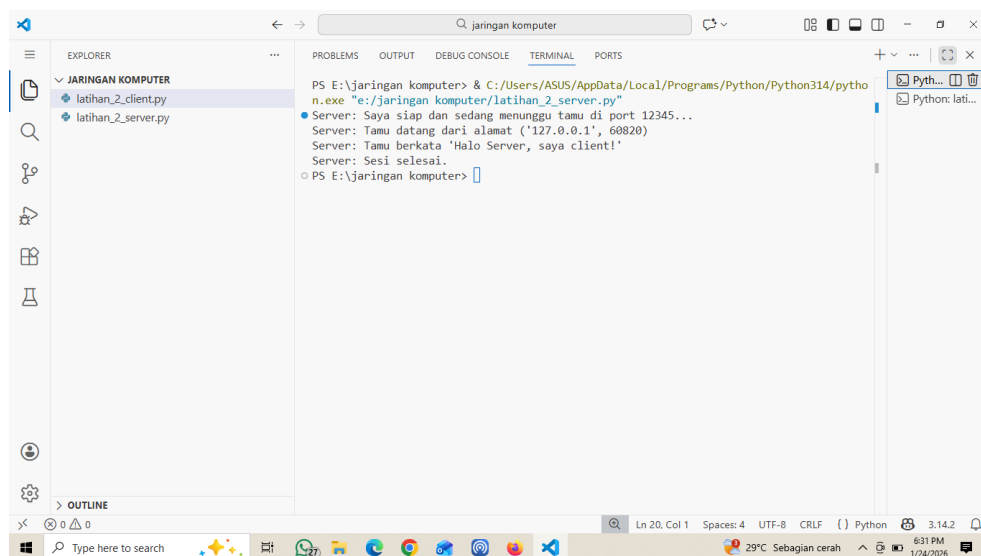
Bab ini memperkenalkan API socket yaitu antarmuka yang digunakan program untuk berkomunikasi lewat jaringan. Kamu akan belajar fungsi-fungsi dasar seperti membuat *socket*, bind, listen, connect, send/receive, dan bagaimana OS menangani koneksi. *Socket* sendiri adalah titik akhir komunikasi jaringan yang digunakan di seluruh aplikasi jaringan.

Hasil :



The screenshot shows the Visual Studio Code interface with the Explorer pane on the left displaying a folder named 'JARINGAN KOMPUTER' containing two files: 'latihan_2_client.py' and 'latihan_2_server.py'. The Terminal pane on the right shows the output of running a Python script. The command executed is 'python e:/jaringan komputer/latihan_2_client.py'. The output shows the client successfully connecting to the server, receiving a response 'Halo Tamu, salam kenal!', and then closing the connection.

```
PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "e:/jaringan komputer/latihan_2_client.py"
Client: Berhasil terhubung ke server
Client: Server membalas 'Halo Tamu, salam kenal!'
Client: Koneksi ditutup
PS E:\jaringan komputer>
```



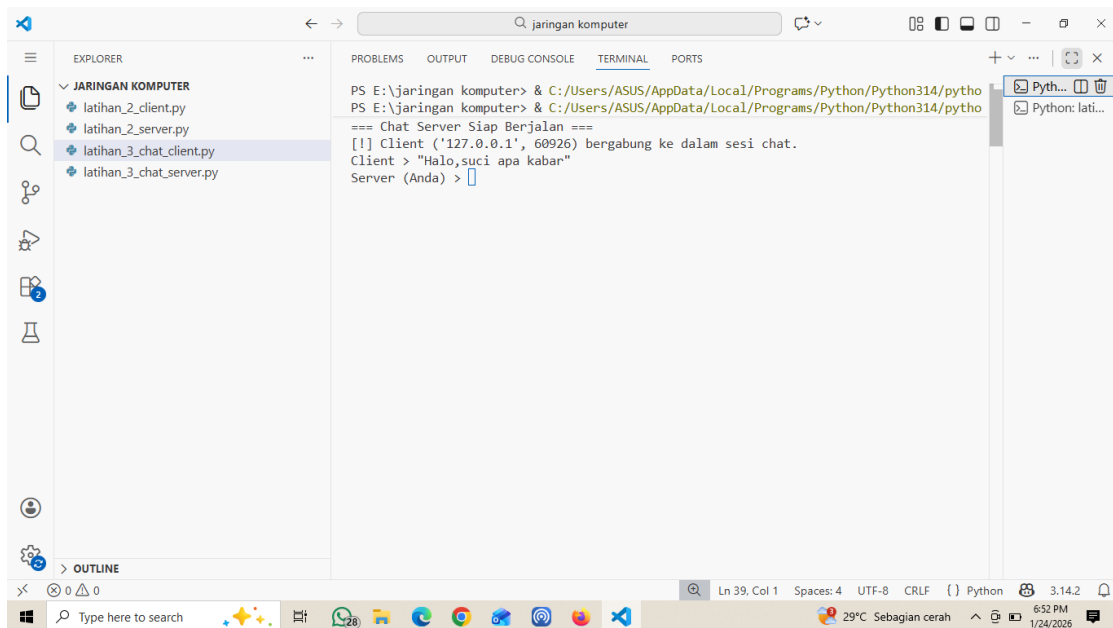
The screenshot shows the Visual Studio Code interface with the Explorer pane on the left displaying the same folder 'JARINGAN KOMPUTER' with files 'latihan_2_client.py' and 'latihan_2_server.py'. The Terminal pane on the right shows the output of running a Python script. The command executed is 'python e:/jaringan komputer/latihan_2_server.py'. The output shows the server listening on port 12345, receiving a connection from '127.0.0.1', receiving the message 'Halo Server, saya client!', and then closing the connection.

```
PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "e:/jaringan komputer/latihan_2_server.py"
Server: Saya siap dan sedang menunggu tamu di port 12345...
Server: Tamu datang dari alamat ('127.0.0.1', 60820)
Server: Tamu berkata 'Halo Server, saya client!'
Server: Sesi selesai.
PS E:\jaringan komputer>
```

Bab 3: Protokol TCP (Aplikasi Chat)

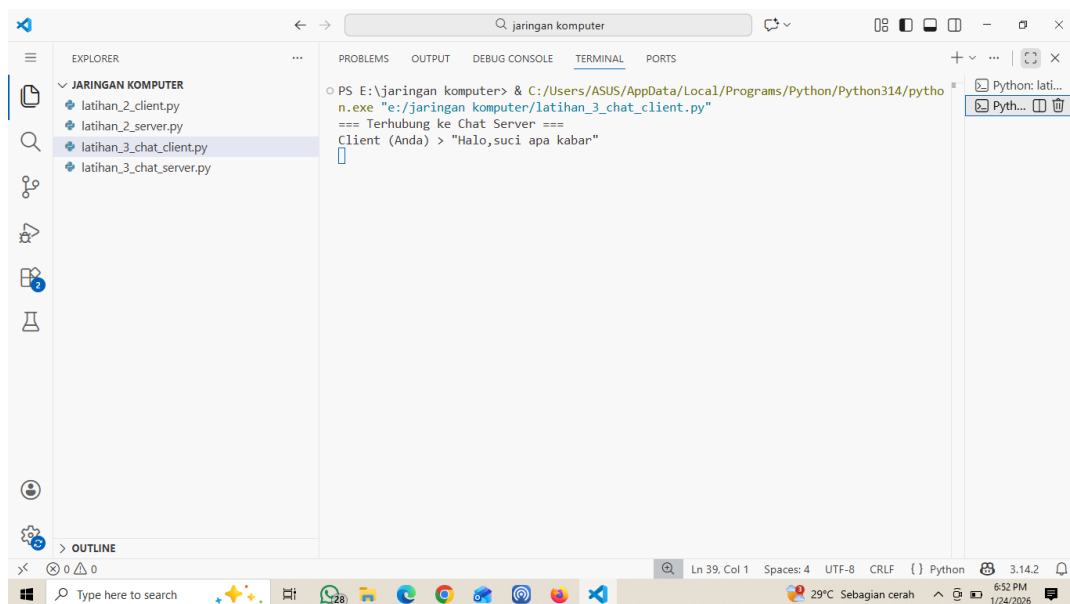
Fokus pada Transmission Control Protocol (TCP) protokol yang menjamin koneksi stabil antara dua aplikasi. Kamu akan membuat aplikasi chat sederhana yang berjalan di atas TCP, belajar tentang three-way handshake, pengiriman pesan, dan bagaimana memastikan tidak ada data yang hilang.

Hasil :



The screenshot shows the Visual Studio Code interface with a file explorer on the left containing a folder named 'JARINGAN KOMPUTER'. Inside this folder are four files: 'latihan_2_client.py', 'latihan_2_server.py', 'latihan_3_chat_client.py', and 'latihan_3_chat_server.py'. The 'TERMINAL' pane on the right displays the output of running a Python script. The commands executed are 'python latihan_3_chat_server.py' and 'python latihan_3_chat_client.py'. The output shows the server starting, a client connecting from '127.0.0.1', and the client sending the message 'Halo,suci apa kabar'.

```
PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "e:/jaringan komputer/latihan_3_chat_server.py"
PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "e:/jaringan komputer/latihan_3_chat_client.py"
=== Chat Server Siap Berjalan ===
[!] Client ('127.0.0.1', 60926) bergabung ke dalam sesi chat.
Client > "Halo,suci apa kabar"
Server (Anda) > 
```



The screenshot shows the Visual Studio Code interface with the same file explorer as the previous image. The 'TERMINAL' pane displays the output of running the chat client program. The command executed is 'python latihan_3_chat_client.py'. The output shows the client connecting to the chat server and sending the message 'Halo,suci apa kabar'.

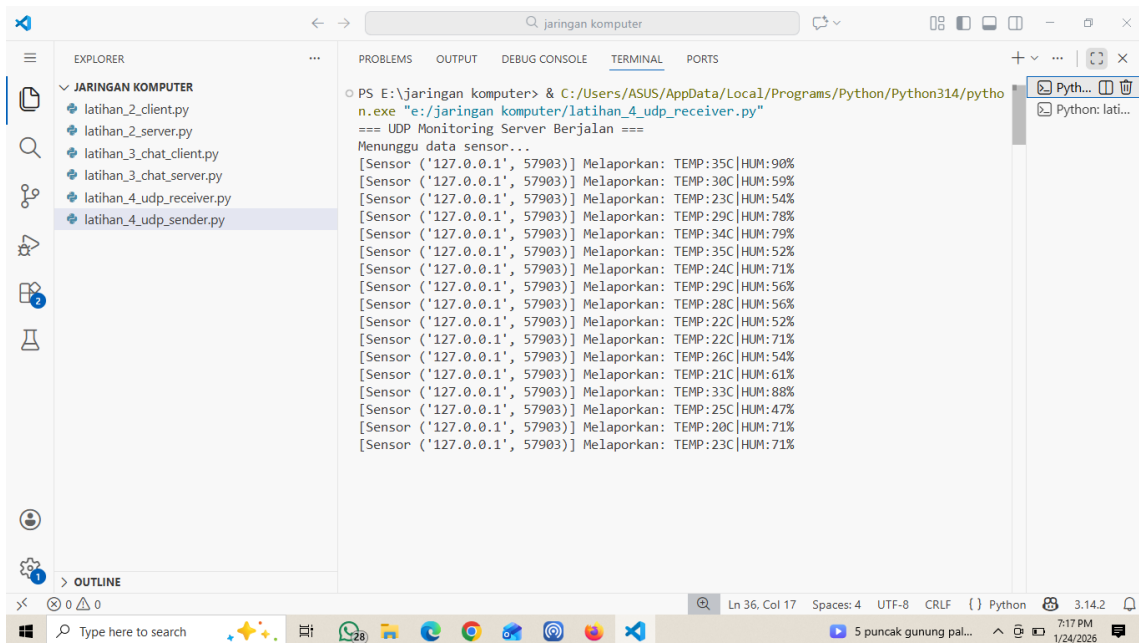
```
PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "e:/jaringan komputer/latihan_3_chat_client.py"
=== Terhubung ke Chat Server ===
Client (Anda) > "Halo,suci apa kabar"

```

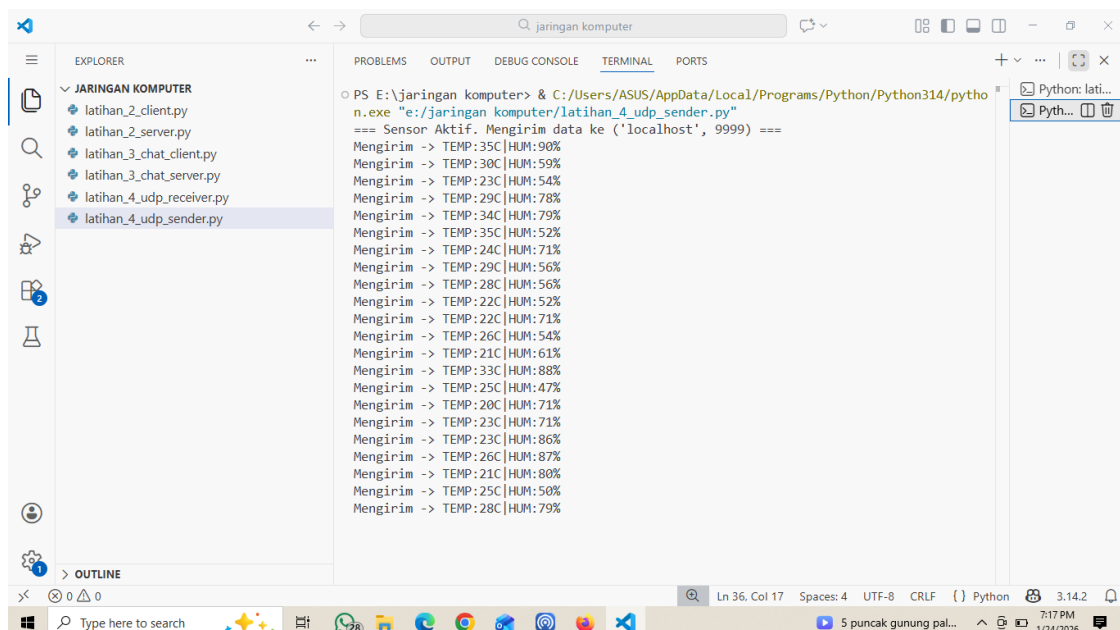
Bab 4: Protokol UDP (Streaming & Broadcasting)

Berbeda dari TCP, User Datagram Protocol (UDP) melakukan pengiriman data tanpa jaminan koneksi. Bab ini mengajarkan cara menggunakan UDP untuk aplikasi seperti streaming audio/video atau broadcast, di mana kecepatan sering lebih penting daripada akurasi.

Hasil :



```
PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "e:/jaringan komputer/latihan_4_udp_receiver.py"
=== UDP Monitoring Server Berjalan ===
Menunggu data sensor...
[Sensor ('127.0.0.1', 57903)] Melaporkan: TEMP:35C|HUM:90%
[Sensor ('127.0.0.1', 57903)] Melaporkan: TEMP:30C|HUM:59%
[Sensor ('127.0.0.1', 57903)] Melaporkan: TEMP:23C|HUM:54%
[Sensor ('127.0.0.1', 57903)] Melaporkan: TEMP:29C|HUM:78%
[Sensor ('127.0.0.1', 57903)] Melaporkan: TEMP:34C|HUM:79%
[Sensor ('127.0.0.1', 57903)] Melaporkan: TEMP:35C|HUM:52%
[Sensor ('127.0.0.1', 57903)] Melaporkan: TEMP:24C|HUM:71%
[Sensor ('127.0.0.1', 57903)] Melaporkan: TEMP:29C|HUM:56%
[Sensor ('127.0.0.1', 57903)] Melaporkan: TEMP:28C|HUM:56%
[Sensor ('127.0.0.1', 57903)] Melaporkan: TEMP:22C|HUM:52%
[Sensor ('127.0.0.1', 57903)] Melaporkan: TEMP:22C|HUM:71%
[Sensor ('127.0.0.1', 57903)] Melaporkan: TEMP:26C|HUM:54%
[Sensor ('127.0.0.1', 57903)] Melaporkan: TEMP:21C|HUM:61%
[Sensor ('127.0.0.1', 57903)] Melaporkan: TEMP:33C|HUM:88%
[Sensor ('127.0.0.1', 57903)] Melaporkan: TEMP:25C|HUM:47%
[Sensor ('127.0.0.1', 57903)] Melaporkan: TEMP:20C|HUM:71%
[Sensor ('127.0.0.1', 57903)] Melaporkan: TEMP:23C|HUM:71%
```

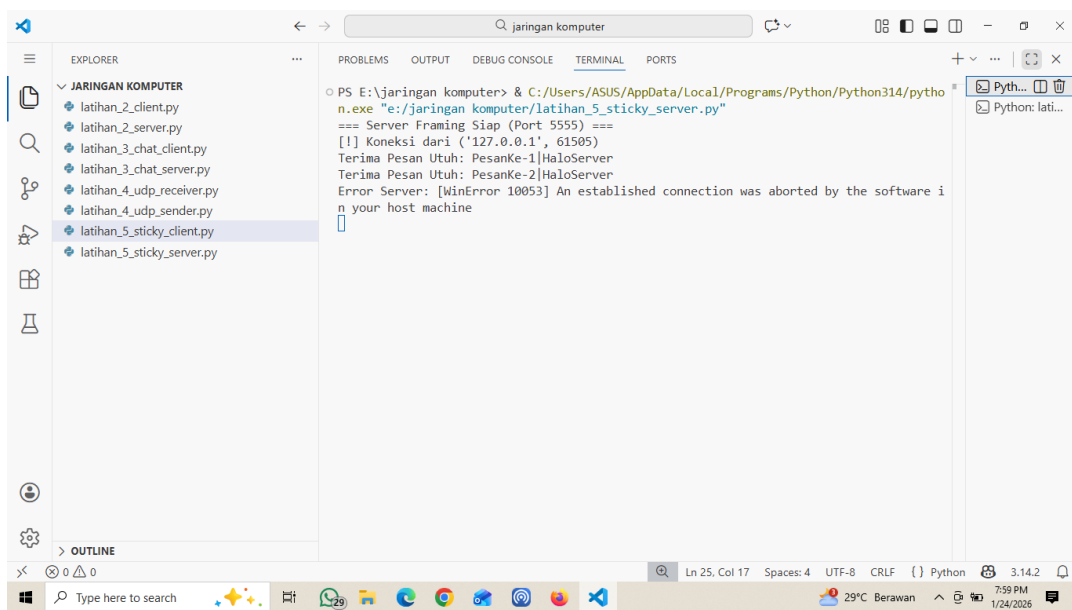


```
PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "e:/jaringan komputer/latihan_4_udp_sender.py"
=== Sensor Aktif. Mengirim data ke ('localhost', 9999) ===
Mengirim -> TEMP:35C|HUM:90%
Mengirim -> TEMP:30C|HUM:59%
Mengirim -> TEMP:23C|HUM:54%
Mengirim -> TEMP:29C|HUM:78%
Mengirim -> TEMP:34C|HUM:79%
Mengirim -> TEMP:35C|HUM:52%
Mengirim -> TEMP:24C|HUM:71%
Mengirim -> TEMP:29C|HUM:56%
Mengirim -> TEMP:28C|HUM:56%
Mengirim -> TEMP:22C|HUM:52%
Mengirim -> TEMP:22C|HUM:71%
Mengirim -> TEMP:26C|HUM:54%
Mengirim -> TEMP:21C|HUM:61%
Mengirim -> TEMP:33C|HUM:88%
Mengirim -> TEMP:25C|HUM:47%
Mengirim -> TEMP:20C|HUM:71%
Mengirim -> TEMP:23C|HUM:71%
Mengirim -> TEMP:23C|HUM:86%
Mengirim -> TEMP:26C|HUM:87%
Mengirim -> TEMP:21C|HUM:80%
Mengirim -> TEMP:25C|HUM:50%
Mengirim -> TEMP:28C|HUM:79%
```

Bab 5: Error Handling & Framing Data

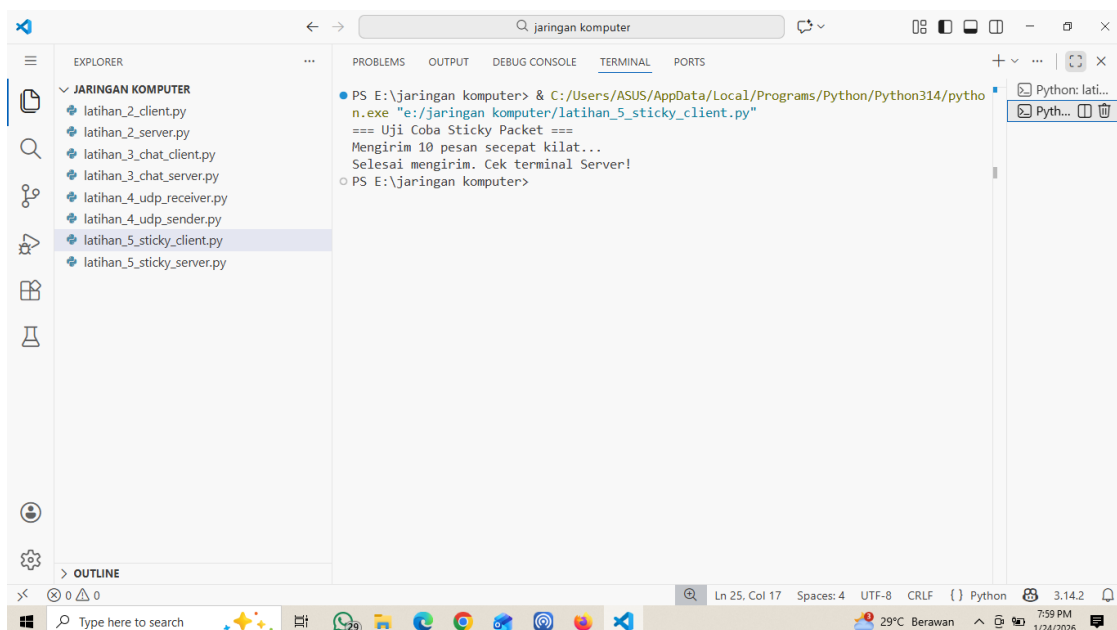
Materi ini membahas cara menangani kesalahan selama komunikasi jaringan dan bagaimana membingkai data supaya bisa dipahami oleh penerima. Topik seperti coding/decoding, pengecekan error, pengepakan paket, dan indikator batas data akan dibahas disini.

Hasil :



The screenshot shows the Visual Studio Code interface with the Explorer pane on the left displaying a folder named 'JARINGAN KOMPUTER'. The file list includes: latihan_2_client.py, latihan_2_server.py, latihan_3_chat_client.py, latihan_3_chat_server.py, latihan_4_udp_receiver.py, latihan_4_udp_sender.py, latihan_5_sticky_client.py (selected), and latihan_5_sticky_server.py. The Terminal pane on the right shows the following output:

```
PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "e:/jaringan komputer/latihan_5_sticky_server.py"
=== Server Framing Siap (Port 5555) ===
[!] Koneksi dari ('127.0.0.1', 61505)
Terima Pesan Utuh: PesanKe-1|HaloServer
Terima Pesan Utuh: PesanKe-2|HaloServer
Error Server: [WinError 10053] An established connection was aborted by the software in your host machine
```



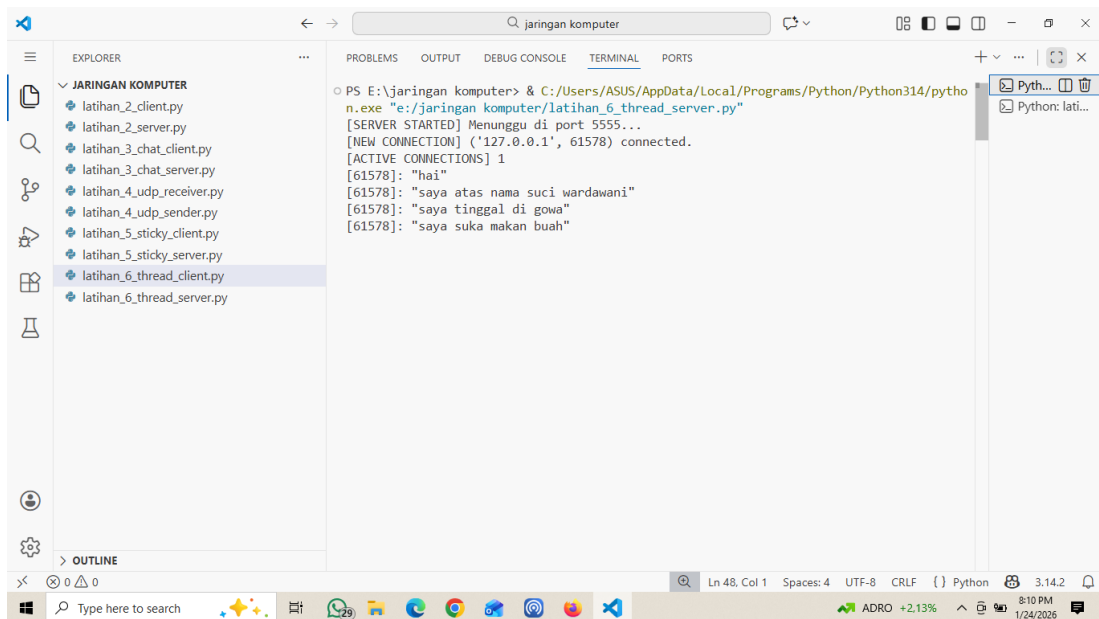
The screenshot shows the Visual Studio Code interface with the Explorer pane on the left displaying the same folder 'JARINGAN KOMPUTER' and file list. The Terminal pane on the right shows the following output:

```
PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "e:/jaringan komputer/latihan_5_sticky_client.py"
=== Uji Coba Sticky Packet ===
Mengirim 10 pesan secepat kilat...
Selesai mengirim. Cek terminal Server!
PS E:\jaringan komputer>
```

Bab 6: Concurrency Part I – Threading

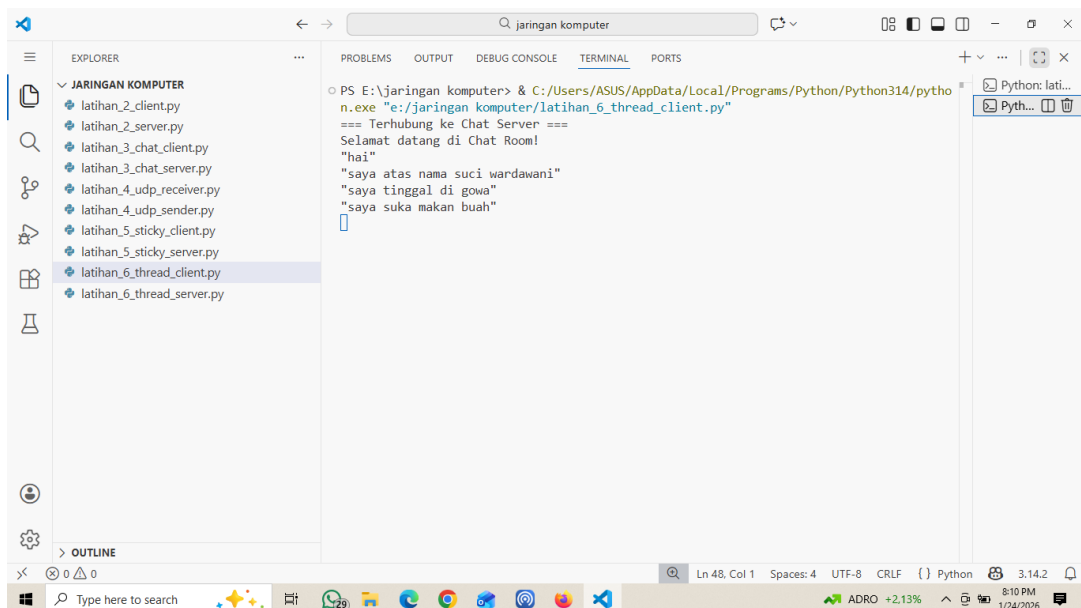
Di jaringan, sering diperlukan program yang bisa menangani banyak koneksi sekaligus. Bab ini memperkenalkan threading (multi-utasan), yaitu teknik untuk membuat program bisa menjalankan banyak tugas jaringan secara bersamaan.

Hasil :



The screenshot shows the Visual Studio Code interface with the Explorer pane on the left displaying a folder named 'JARINGAN KOMPUTER'. The file list includes: latihan_2_client.py, latihan_2_server.py, latihan_3_chat_client.py, latihan_3_chat_server.py, latihan_4_udp_receiver.py, latihan_4_udp_sender.py, latihan_5_sticky_client.py, latihan_5_sticky_server.py, latihan_6_thread_client.py (selected), and latihan_6_thread_server.py. The Terminal pane on the right shows the command prompt output for running the server script:

```
PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "e:/jaringan komputer/latihan_6_thread_server.py"
[SERVER STARTED] Menunggu di port 5555...
[NEW CONNECTION] ('127.0.0.1', 61578) connected.
[ACTIVE CONNECTIONS] 1
[61578]: "hai"
[61578]: "saya atas nama suci wardawani"
[61578]: "saya tinggal di gowa"
[61578]: "saya suka makan buah"
```



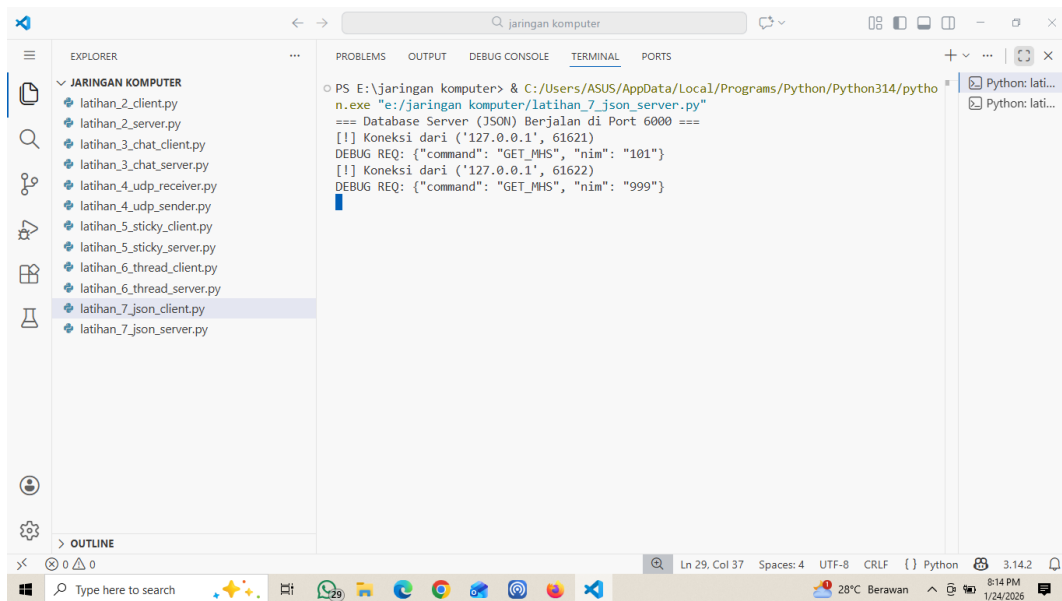
The screenshot shows the Visual Studio Code interface with the Explorer pane on the left displaying the same file list as the previous image. The Terminal pane on the right shows the command prompt output for running the client script:

```
PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "e:/jaringan komputer/latihan_6_thread_client.py"
=== Terhubung ke Chat Server ===
Selamat datang di Chat Room!
"hai"
"saya atas nama suci wardawani"
"saya tinggal di gowa"
"saya suka makan buah"
```

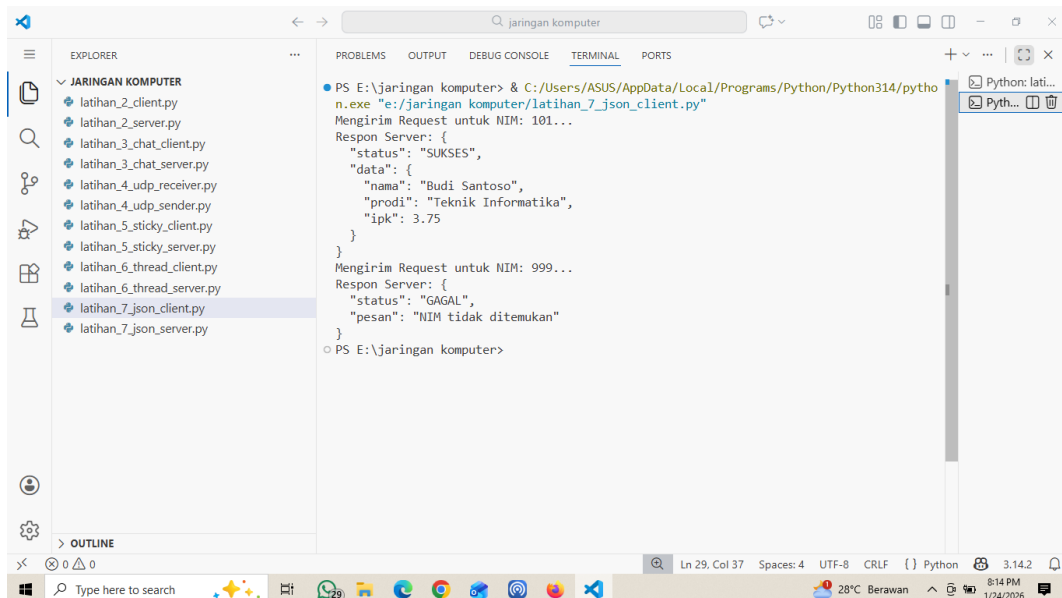
Bab 7: Serialisasi Data (JSON & Pickle)

Serialisasi adalah cara mengubah struktur data menjadi format yang bisa dikirim lewat jaringan, misalnya JSON atau Pickle. Bab ini menjelaskan cara serialisasi/ deserialisasi data di jaringan agar komunikasi lebih efisien.

Hasil :



```
PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python
n.exe "e:/jaringan komputer/latihan_7_json_server.py"
=== Database Server (JSON) Berjalan di Port 6000 ===
[!] Koneksi dari ('127.0.0.1', 61621)
DEBUG REQ: {"command": "GET_MHS", "nim": "101"}
[!] Koneksi dari ('127.0.0.1', 61622)
DEBUG REQ: {"command": "GET_MHS", "nim": "999"}
```

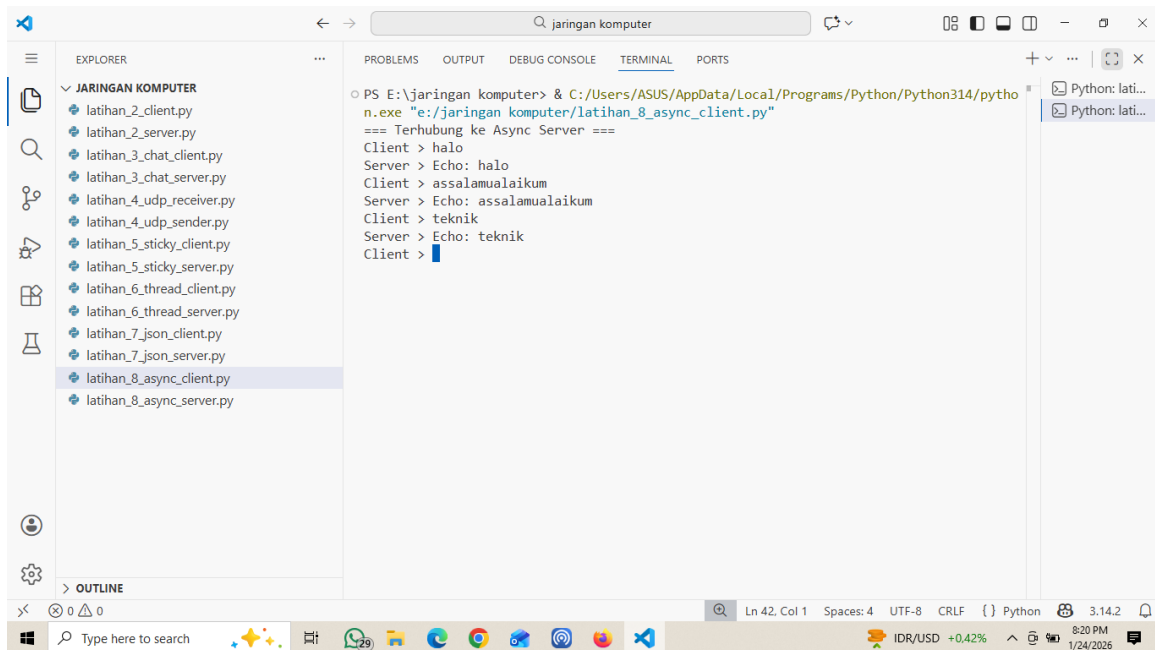


```
PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/pytho
n.exe "e:/jaringan komputer/latihan_7_json_client.py"
Mengirim Request untuk NIM: 101...
Respon Server: {
  "status": "SUKSES",
  "data": {
    "nama": "Budi Santoso",
    "prodi": "Teknik Informatika",
    "ipk": 3.75
  }
}
Mengirim Request untuk NIM: 999...
Respon Server: {
  "status": "GAGAL",
  "pesan": "NIM tidak ditemukan"
}
PS E:\jaringan komputer>
```


Bab 8: Asynchronous I/O (Concurrency Part II)

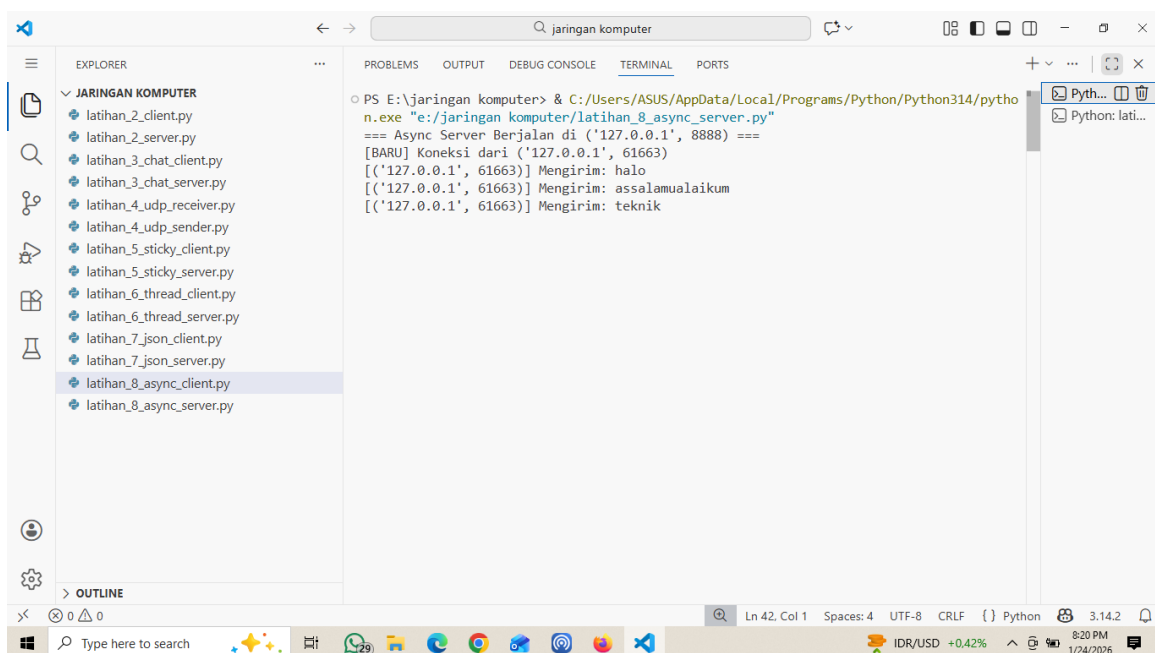
Selain threading, ada pendekatan asynchronous *I/O* yang lebih efisien untuk menangani banyak koneksi tanpa memblokir program. Bab ini menjelaskan cara kerja asynchronous, loop event, dan bagaimana menulis server/client yang non-blokir.

Hasil :



The screenshot shows the Visual Studio Code interface with the Explorer pane on the left displaying a folder named 'JARINGAN KOMPUTER'. The file 'latihan_8_async_client.py' is selected. The Terminal pane on the right shows the command prompt output for running the client program. The output indicates a successful connection to the server and the receipt of three echo responses.

```
PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "e:/jaringan komputer/latihan_8_async_client.py"
=== Terhubung ke Async Server ===
Client > halo
Server > Echo: halo
Client > assalamualaikum
Server > Echo: assalamualaikum
Client > teknik
Server > Echo: teknik
Client > 
```



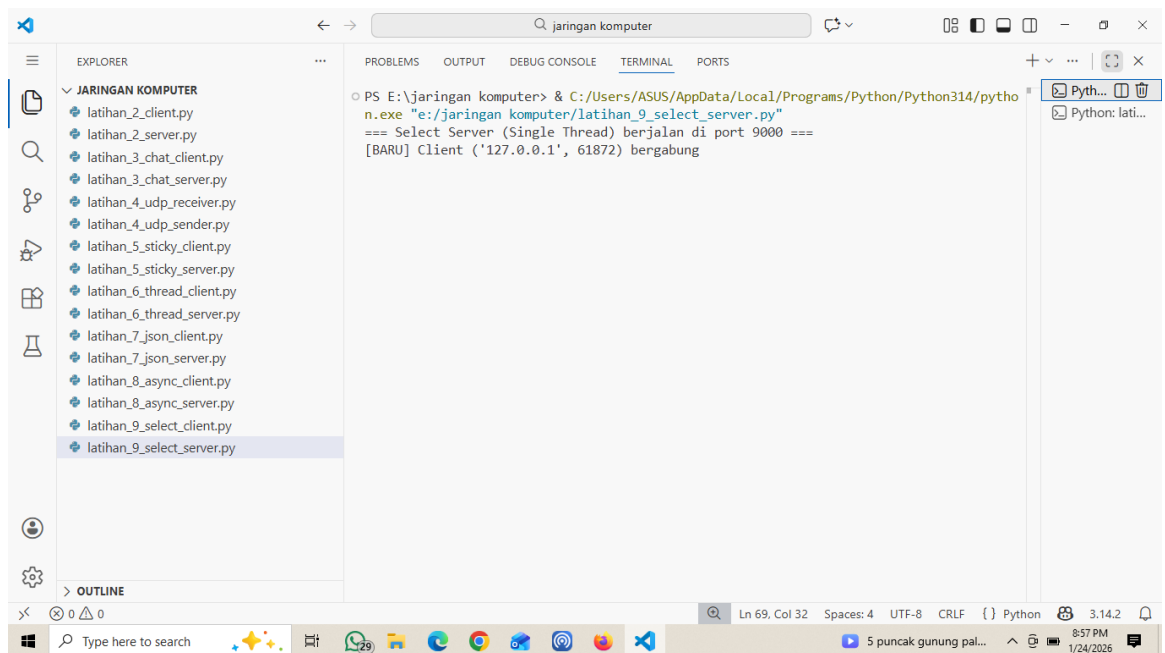
The screenshot shows the Visual Studio Code interface with the Explorer pane on the left displaying a folder named 'JARINGAN KOMPUTER'. The file 'latihan_8_async_server.py' is selected. The Terminal pane on the right shows the command prompt output for running the server program. The output indicates the server is running on port 8888 and has received three connections from the client, each with an echo response.

```
PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "e:/jaringan komputer/latihan_8_async_server.py"
=== Async Server Berjalan di ('127.0.0.1', 8888) ===
[BARU] Koneksi dari ('127.0.0.1', 61663)
[('127.0.0.1', 61663)] Mengirim: halo
[('127.0.0.1', 61663)] Mengirim: assalamualaikum
[('127.0.0.1', 61663)] Mengirim: teknik
```

Bab 9: I/O Multiplexing (select & poll)

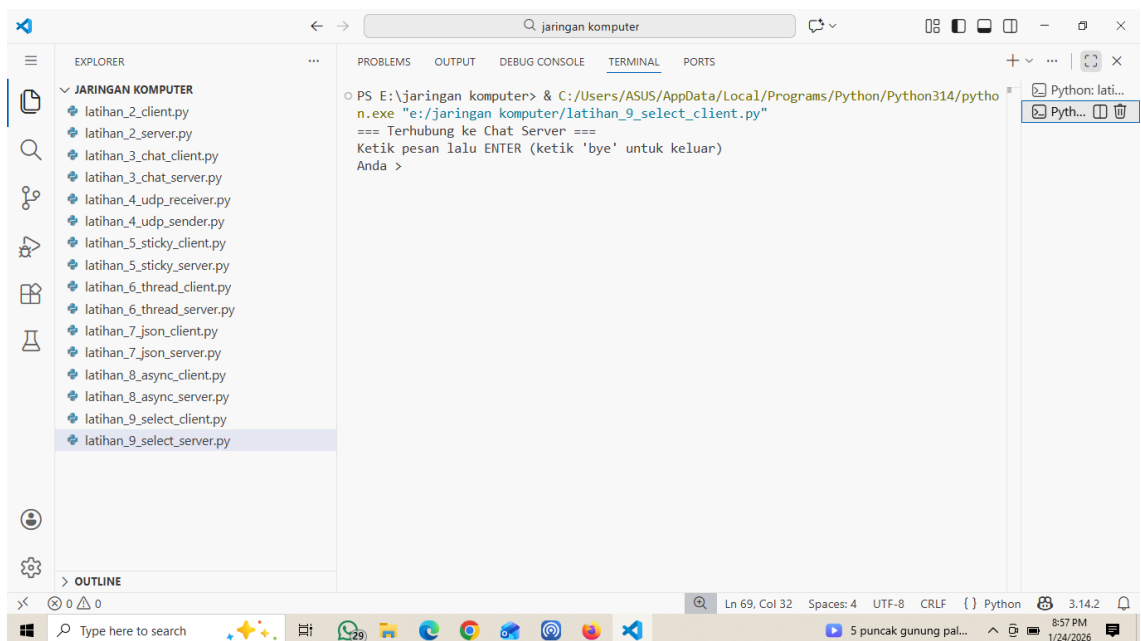
Bab lanjutan tentang manajemen koneksi: I/O multiplexing seperti `select()` dan `poll()` memungkinkan satu program memantau banyak koneksi tanpa harus membuat banyak thread. Ini penting di server berskala besar.

Hasil :



The screenshot shows the Visual Studio Code interface with the Explorer pane on the left displaying a project named 'JARINGAN KOMPUTER'. The file list includes various Python files for client and server exercises. The Terminal pane on the right shows the command prompt output for running the 'latihan_9_select_server.py' file. The output indicates that the server is running on port 9000 and a client has connected.

```
PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python n.exe "e:/jaringan komputer/latihan_9_select_server.py"
=== Select Server (Single Thread) berjalan di port 9000 ===
[BARU] Client ('127.0.0.1', 61872) bergabung
```



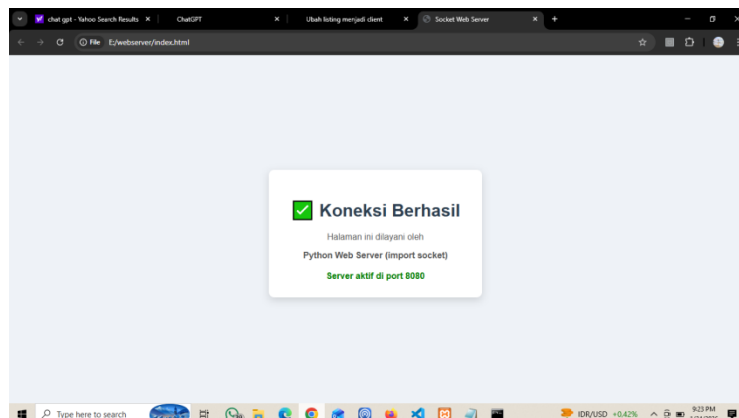
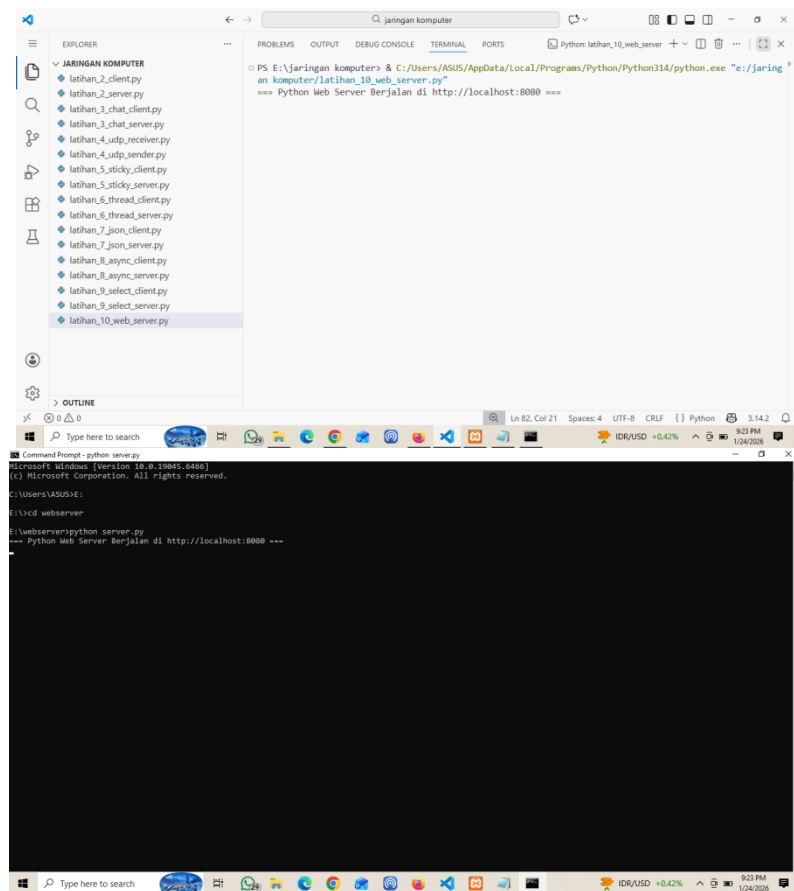
The screenshot shows the Visual Studio Code interface with the Explorer pane on the left displaying the same project. The Terminal pane on the right shows the command prompt output for running the 'latihan_9_select_client.py' file. The output indicates that the client is connected to the chat server and prompts the user to enter a message.

```
PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python n.exe "e:/jaringan komputer/latihan_9_select_client.py"
=== Terhubung ke Chat Server ===
Ketik pesan lalu ENTER (ketik 'bye' untuk keluar)
Anda >
```

Bab 10: Protokol HTTP & Web Server

Bab ini fokus pada HTTP, protokol utama di web. Kamu akan belajar membuat server HTTP sederhana yang bisa menerima request dari klien (misalnya browser) dan mengirim response seperti halaman web.

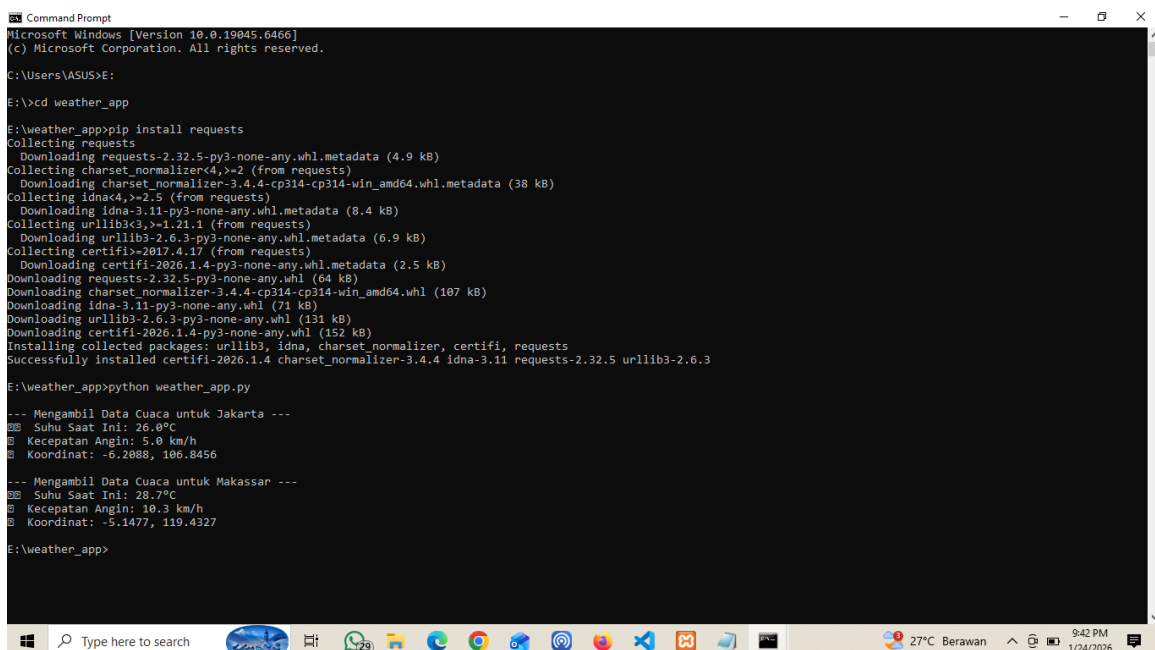
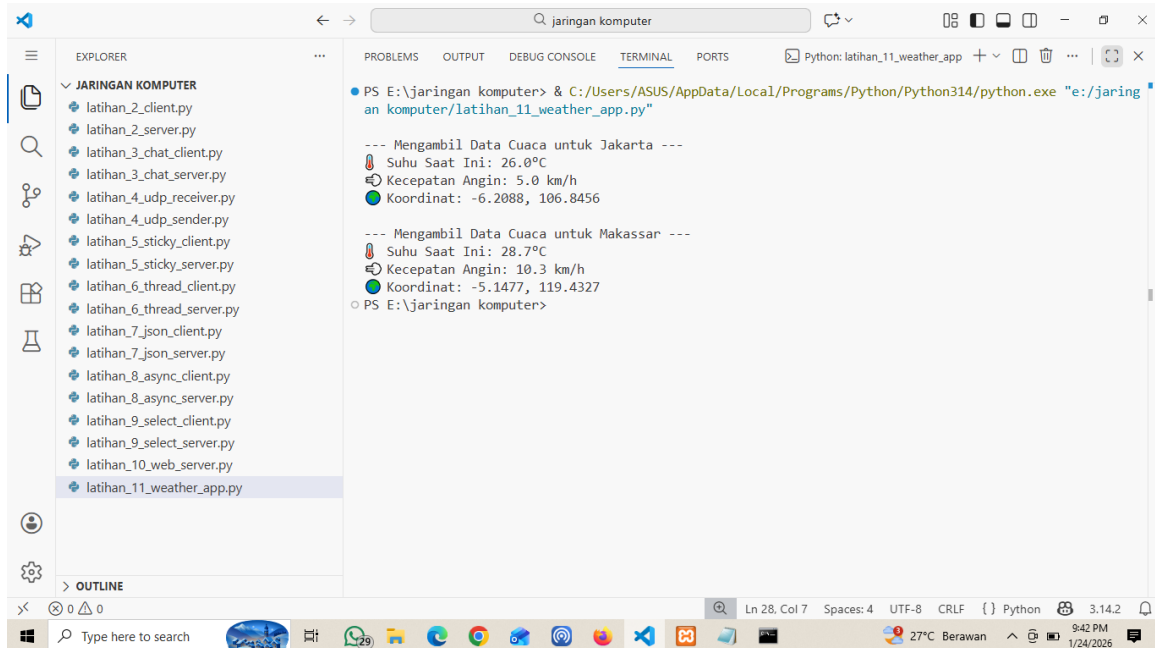
Hasil :



Bab 11: REST API & Web Services

Selanjutnya kamu akan belajar membangun REST API antarmuka yang dipakai aplikasi lain untuk berkomunikasi dengan server (misalnya aplikasi mobile yang memakai data dari server). REST API merupakan dasar banyak layanan web modern.

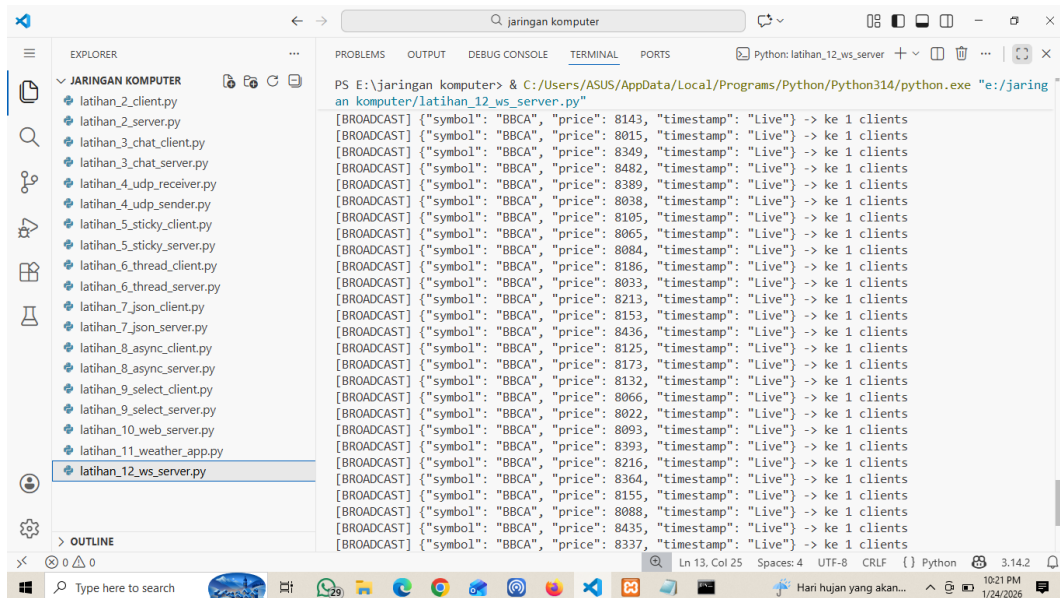
Hasil :



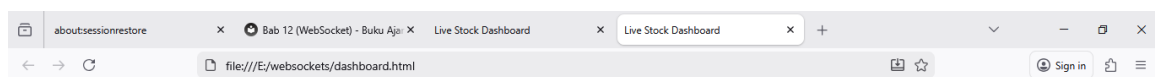
Bab 12: Real-time Communication (WebSocket)

WebSocket adalah protokol yang memungkinkan komunikasi dua arah secara real-time di atas koneksi TCP (persistent connection). Dengan WebSocket, server dan klien dapat bertukar data tanpa harus setiap kali membuat koneksi baru.

Hasil :



```
PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "e:/jaringan komputer/latihan_12_ws_server.py"
[BROADCAST] {"symbol": "BBCA", "price": 8143, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8015, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8349, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8482, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8389, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8038, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8105, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8065, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8084, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8186, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8033, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8213, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8153, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8436, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8125, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8173, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8132, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8066, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8022, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8093, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8393, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8216, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8364, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8155, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8088, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8435, "timestamp": "Live"} -> ke 1 clients
[BROADCAST] {"symbol": "BBCA", "price": 8337, "timestamp": "Live"} -> ke 1 clients
```



Pantauan Saham BBCA (Real-time)

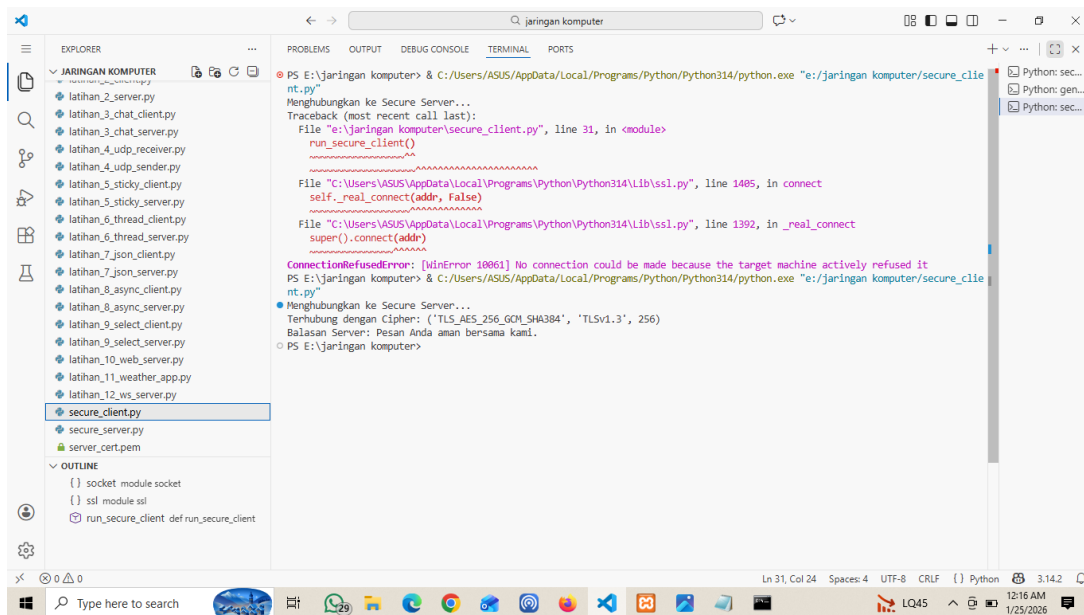
Rp 8466
Connected to Server 



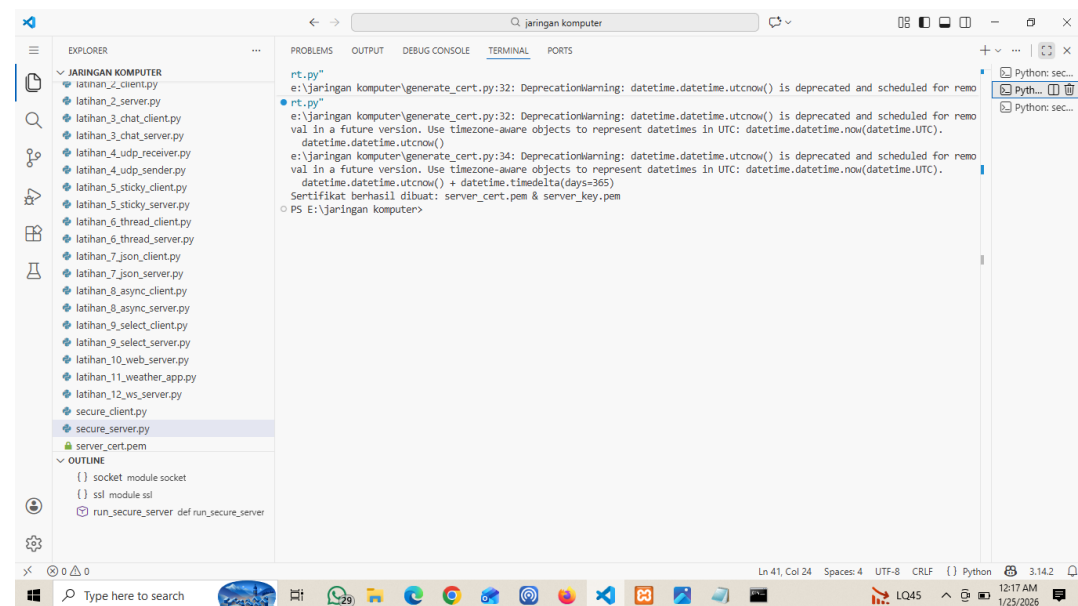
Bab 13: Keamanan Jaringan (Network Security)

Di bab ini kamu akan mempelajari teknik-teknik keamanan komunikasi jaringan misalnya enkripsi, otentikasi, SSL/TLS, dan bagaimana melindungi aplikasi dari serangan jaringan. Ini penting supaya aplikasi yang kamu buat tidak rentan diretas.

Hasil :



The screenshot shows the Visual Studio Code interface with the Explorer pane on the left displaying a project named 'JARINGAN KOMPUTER'. The file 'secure_client.py' is selected. The Terminal pane on the right shows the output of running a Python script. The script attempts to connect to a secure server but fails with a 'ConnectionRefusedError: [WinError 10061] No connection could be made because the target machine actively refused it'. The error message is displayed in red text. The terminal also shows the command used to run the script: 'PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "e:/jaringan komputer/secure_client.py"'.

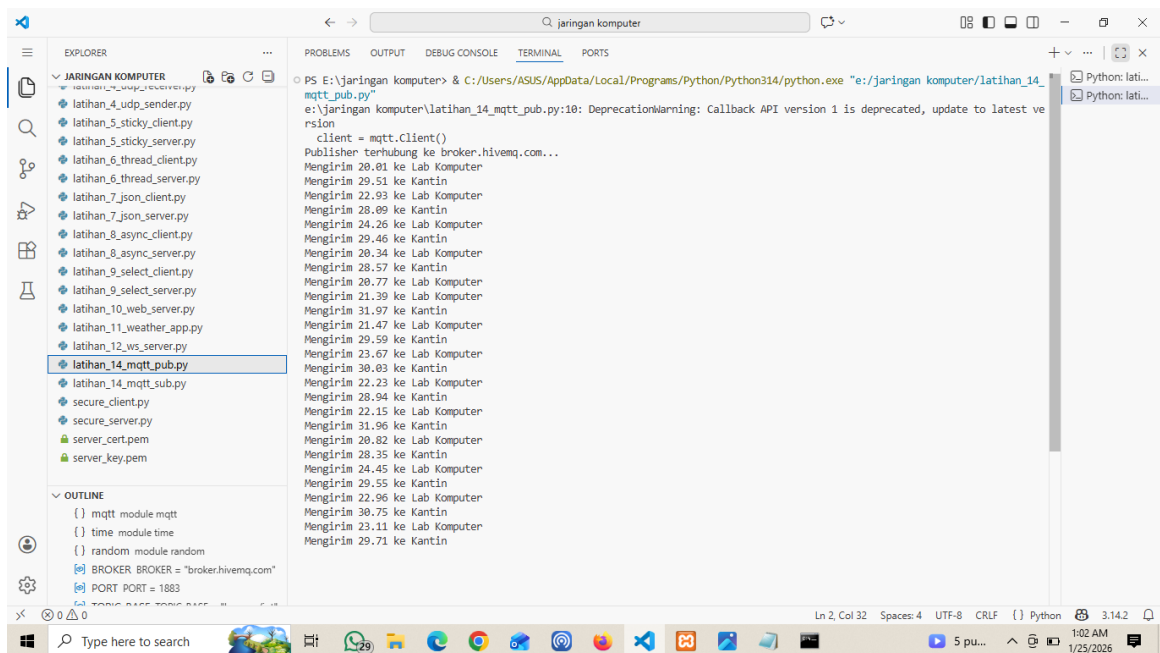


The screenshot shows the Visual Studio Code interface with the Explorer pane on the left displaying the same project. The file 'secure_server.py' is selected. The Terminal pane on the right shows the output of running a Python script. The script successfully generates a certificate, displaying the following output: 'e:\jaringan komputer\generate_cert.py:32: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).', 'e:\jaringan komputer\generate_cert.py:34: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).', 'datetime.datetime.utcnow() + datetime.timedelta(days=365)', 'Sertifikat berhasil dibuat: server_cert.pem & server_key.pem'. The terminal also shows the command used to run the script: 'PS E:\jaringan komputer>'.

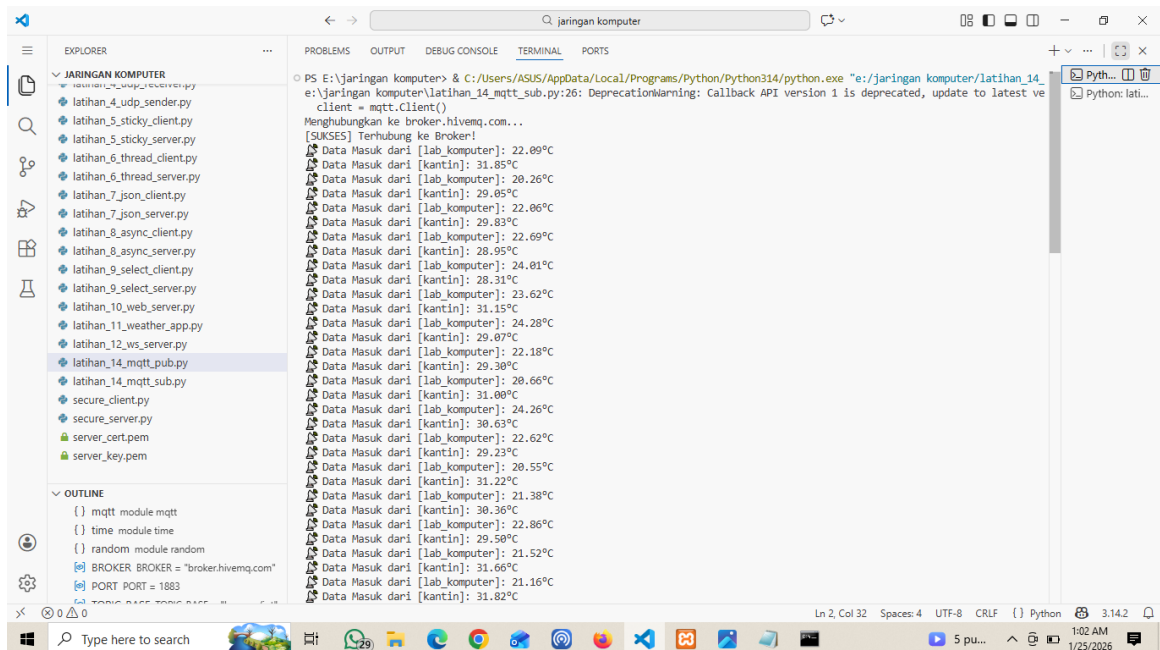
Bab 14: Arsitektur Sistem Terdistribusi & IoT (MQTT)

Materi ini memperkenalkan komunikasi terdistribusi dan protokol untuk IoT seperti MQTT protokol ringan yang banyak digunakan perangkat IoT karena efisiensi data dan bandwidth.

Hasil :



```
PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "e:/jaringan komputer/latihan_14_mqtt_pub.py"
e:\jaringan komputer\latihan_14_mqtt_pub.py:10: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt.Client()
Publisher terhubung ke broker.hivemq.com...
Menginim 20.01 ke Lab Komputer
Menginim 29.51 ke Kantin
Menginim 22.93 ke Lab Komputer
Menginim 28.09 ke Kantin
Menginim 24.26 ke Lab Komputer
Menginim 29.46 ke Kantin
Menginim 20.34 ke Lab Komputer
Menginim 28.57 ke Kantin
Menginim 20.77 ke Lab Komputer
Menginim 21.39 ke Lab Komputer
Menginim 31.97 ke Kantin
Menginim 21.47 ke Lab Komputer
Menginim 29.59 ke Kantin
Menginim 23.67 ke Lab Komputer
Menginim 30.03 ke Kantin
Menginim 22.23 ke Lab Komputer
Menginim 28.94 ke Kantin
Menginim 22.15 ke Lab Komputer
Menginim 31.96 ke Kantin
Menginim 20.82 ke Lab Komputer
Menginim 28.35 ke Kantin
Menginim 24.45 ke Lab Komputer
Menginim 29.55 ke Kantin
Menginim 22.96 ke Lab Komputer
Menginim 30.75 ke Kantin
Menginim 23.11 ke Lab Komputer
Menginim 29.71 ke Kantin
```



```
PS E:\jaringan komputer> & C:/Users/ASUS/AppData/Local/Programs/Python/Python314/python.exe "e:/jaringan komputer/latihan_14_mqtt_sub.py"
e:\jaringan komputer\latihan_14_mqtt_sub.py:26: DeprecationWarning: Callback API version 1 is deprecated, update to latest version
  client = mqtt.Client()
Menghubungkan ke broker.hivemq.com...
[SUCCESS] Terhubung ke Broker!
Data Masuk dari [lab komputer]: 22.09°C
Data Masuk dari [kantin]: 31.85°C
Data Masuk dari [lab komputer]: 20.26°C
Data Masuk dari [kantin]: 29.05°C
Data Masuk dari [lab komputer]: 22.06°C
Data Masuk dari [kantin]: 29.83°C
Data Masuk dari [lab komputer]: 22.69°C
Data Masuk dari [kantin]: 28.95°C
Data Masuk dari [lab komputer]: 24.01°C
Data Masuk dari [kantin]: 28.31°C
Data Masuk dari [lab komputer]: 23.62°C
Data Masuk dari [kantin]: 31.15°C
Data Masuk dari [lab komputer]: 24.28°C
Data Masuk dari [kantin]: 29.07°C
Data Masuk dari [lab komputer]: 22.18°C
Data Masuk dari [kantin]: 29.30°C
Data Masuk dari [lab komputer]: 20.66°C
Data Masuk dari [kantin]: 31.00°C
Data Masuk dari [lab komputer]: 24.26°C
Data Masuk dari [kantin]: 30.63°C
Data Masuk dari [lab komputer]: 22.62°C
Data Masuk dari [kantin]: 29.23°C
Data Masuk dari [lab komputer]: 20.55°C
Data Masuk dari [kantin]: 31.22°C
Data Masuk dari [lab komputer]: 21.38°C
Data Masuk dari [kantin]: 30.36°C
Data Masuk dari [lab komputer]: 22.86°C
Data Masuk dari [lab komputer]: 29.50°C
Data Masuk dari [lab komputer]: 21.52°C
Data Masuk dari [kantin]: 31.66°C
Data Masuk dari [lab komputer]: 21.16°C
Data Masuk dari [kantin]: 31.82°C
```

Bab 15: Penutup & Proyek Akhir (Capstone Project)

Tugas proyek ini bertujuan untuk membuat satu aplikasi terintegrasi yang menerapkan konsep sistem terdistribusi, di mana aplikasi tidak berjalan dalam satu program saja, melainkan terdiri dari beberapa komponen terpisah yang saling berkomunikasi melalui jaringan.

Bab terakhir menyatukan semua konsep lewat proyek akhir. Kamu akan menerapkan teknik socket, concurrency, protokol web, dan keamanan untuk membuat aplikasi jaringan nyata sebagai portofolio

Hasil :

[illegible]