**LAPORAN PRAKTIKUM PEMROGRAMAN JARINGAN**

****

**Disusun Oleh :**

**Nur Asiah (231401033)**

**Mata Kuliah : Pemrograman Jaringan**

**Dosen Pengampu : Ucok, S.Kom.,MT**

**FAKULTAS ILMU KOMPUTER**

**PROGRAM STUDI TEKNIK INFORMATIKA**

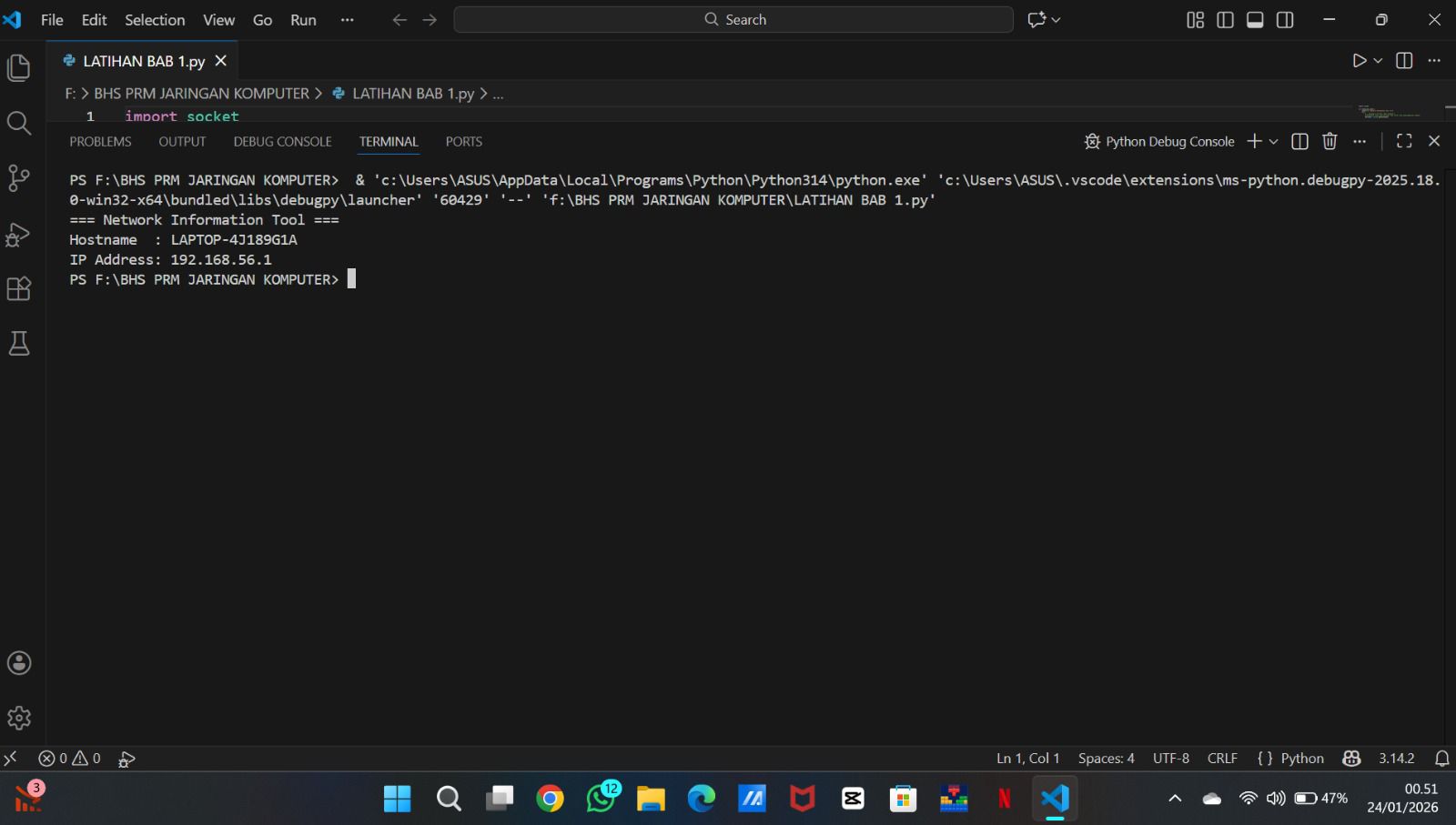
**UNIVERSITAS INDONESIA TIMUR**

**2026**

**Bab 1: Konsep Dasar Pemrograman Jaringan**

Pada gambar praktikum Bab 1 ini membahas konsep fundamental pemrograman jaringan yang menjadi dasar untuk memahami materi selanjutnya. Topik utamanya mencakup jaringan komputer, model client-server, identifikasi alamat IP dan port, serta bagaimana data ditransmisikan antar aplikasi.

**Hasil :**

****

# Bab 2: Socket API Dasar

# Pada Bab ini fokus pembahasan bergeser ke API socket, yaitu antarmuka pemrograman yang memungkinkan aplikasi melakukan komunikasi jaringan. Materi mencakup cara membuat socket di Python, mengikat socket ke alamat dan port tertentu, serta operasi dasar seperti mendengarkan koneksi, menerima data, dan mengirim data.

# Hasil :

# 

# 

# Bab 3: Protokol TCP (Aplikasi Chat)

Pada Bab ini memfokuskan pada membangun aplikasi chat sederhana menggunakan protokol TCP. Karena TCP menyediakan koneksi yang andal dan berurutan, bab ini menjelaskan bagaimana server TCP dapat mengelola koneksi client, menangani input pengguna, dan menyampaikan pesan secara dua arah. Pembaca belajar bagaimana menulis kode lengkap untuk client dan server yang bisa saling bertukar pesan, serta bagaimana menangani koneksi yang masuk dan keluar secara dinamis.

# Hasil :

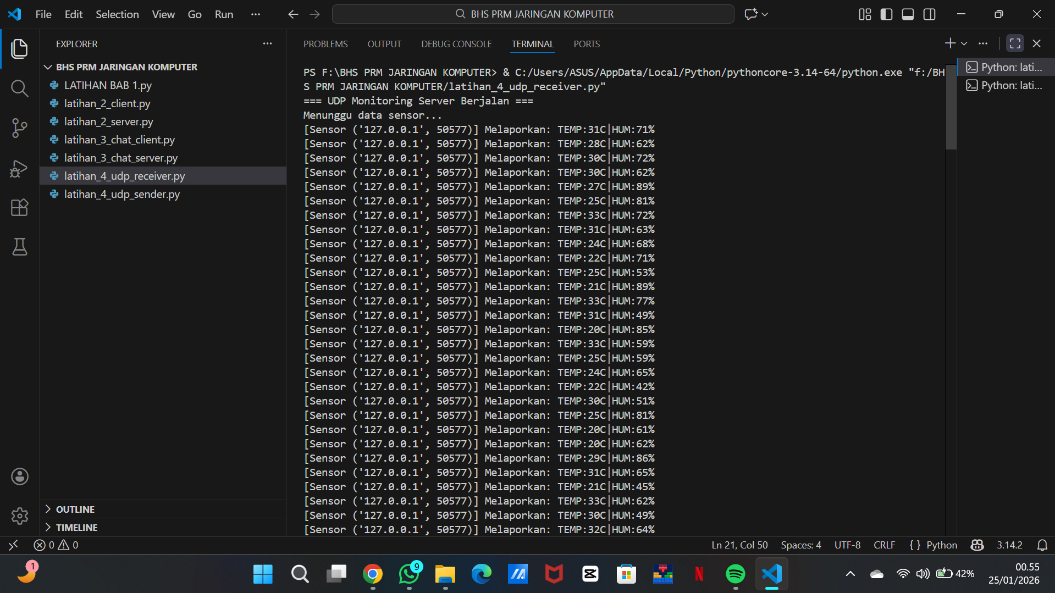
# 

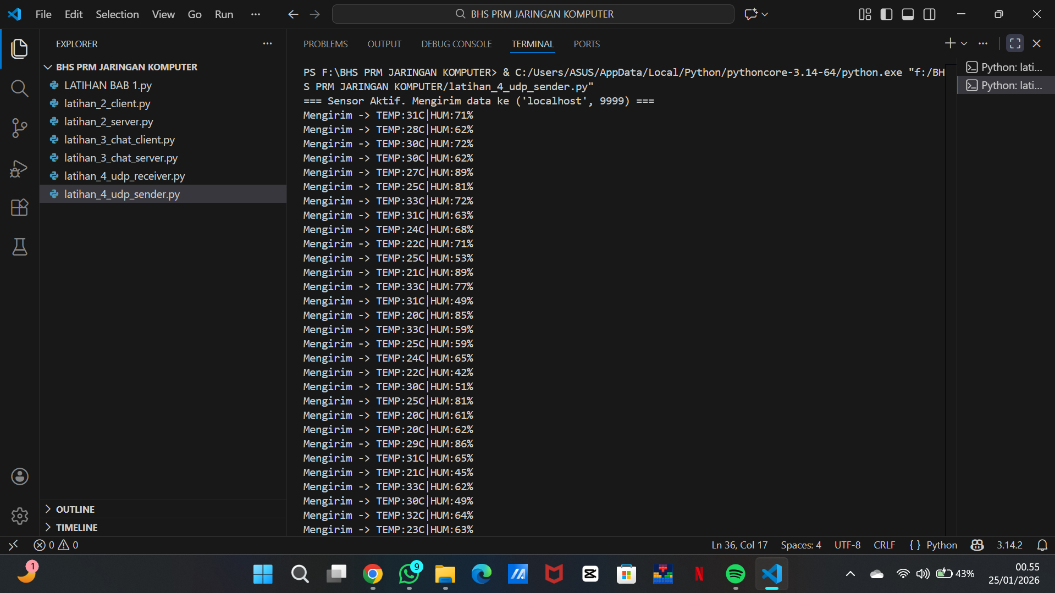
# 

**Bab 4: Protokol UDP (Streaming & Broadcasting)**

Pada Bab ini pembahasan beralih ke komunikasi UDP, yang bersifat tanpa koneksi. Bab ini menguraikan penggunaan UDP untuk aplikasi yang membutuhkan latensi rendah dan throughput tinggi, seperti streaming audio/video atau broadcast pesan. Pembaca akan dipandu memahami bagaimana membangun soket UDP, mengirim dan menerima paket data, serta menghadapi tantangan yang muncul karena UDP tidak menjamin pengiriman atau urutan paket.

**Hasil :**

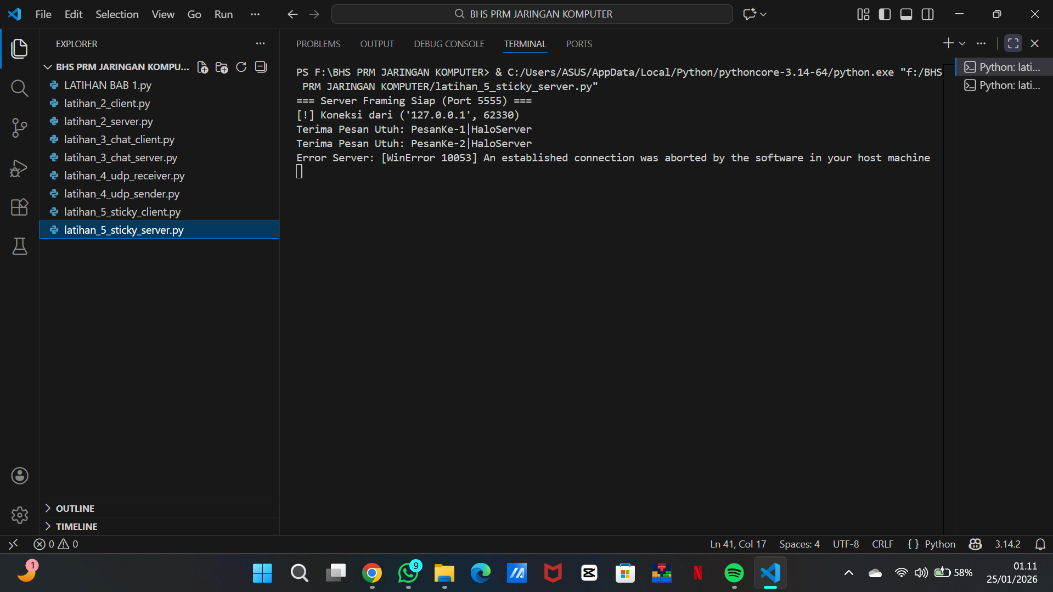
****

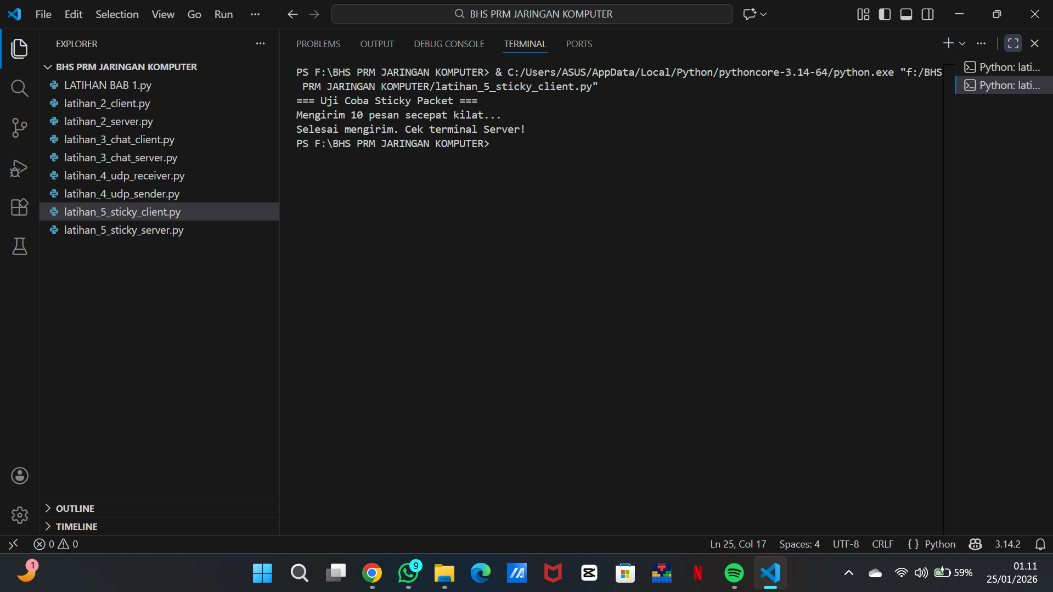
****

**Bab 5: Error Handling & Framing Data**

Pada Bab ini membahas teknik-teknik penting untuk membentuk data (framing) dan menangani kesalahan jaringan. Karena data yang berjalan di jaringan belum tentu utuh atau datang dengan format yang konsisten, penanganan framing membantu memastikan bahwa pesan dapat dipisahkan dan diinterpretasi dengan benar oleh penerima.

**Hasil :**

****

****

# Bab 6: Concurrency Part I – Threading

Pada Bab ini menjelaskan bagaimana setiap koneksi client dapat ditangani di thread terpisah sehingga server dapat melayani banyak client tanpa blocking satu sama lain. Selain itu dibahas pula bagaimana sinkronisasi sumber daya bersama dilakukan agar tidak terjadi konflik antar thread, serta strategi untuk memaksimalkan performa server pada aplikasi dunia nyata.

# Hasil :

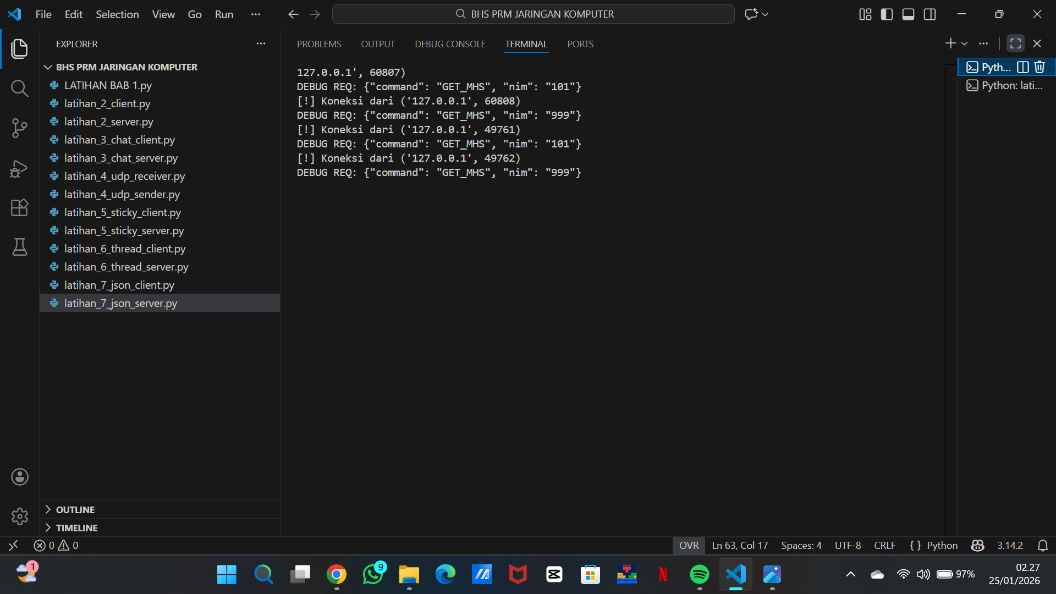
# 

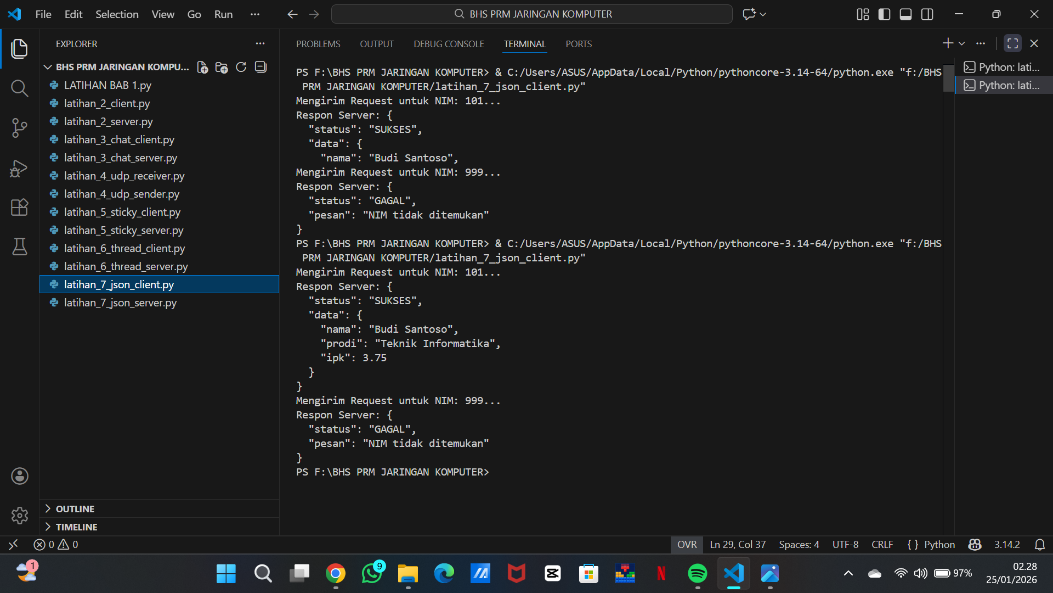
# 

**Bab 7: Serialisasi Data (JSON & Pickle)**

Pada Bab ini mengulas pentingnya serialisasi data ketika mengirimkan struktur data yang kompleks melalui jaringan. Dua teknik yang umum dibahas adalah penggunaan format JSON (teks) dan Pickle (biner Python). JSON cocok untuk interoperabilitas dan dibaca manusia, sedangkan Pickle lebih efisien untuk objek Python namun kurang kompatibel di luar Python.

**Hasil :**

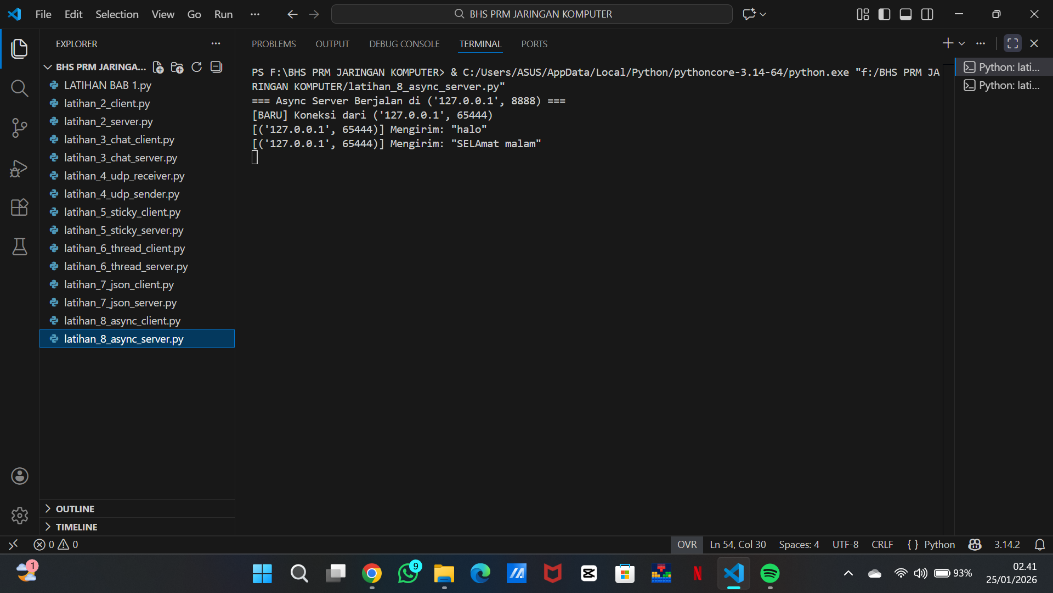
****

****

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

Pada Bab ini memperkenalkan konsep I/O asinkron menggunakan event loop. Teknik ini memungkinkan operasi jaringan berjalan tanpa blocking thread utama, sehingga server bisa mengelola banyak koneksi sekaligus secara efisien. Biasanya diperkenalkan pada modul seperti asyncio di Python, memahami bagaimana coroutine bekerja, serta bagaimana event loop memicu tugas-tugas asinkron sesuai dengan event yang terjadi di jaringan.

**Hasil :**

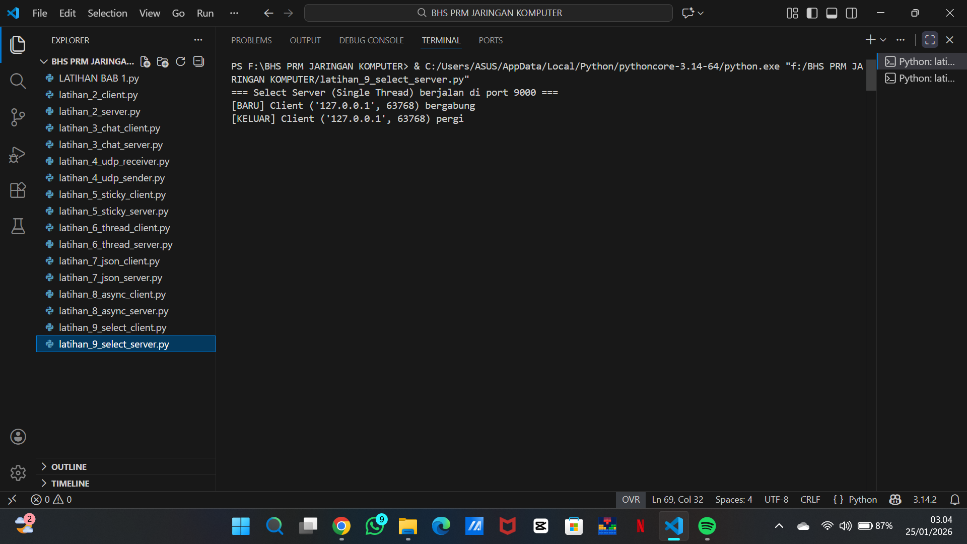
****

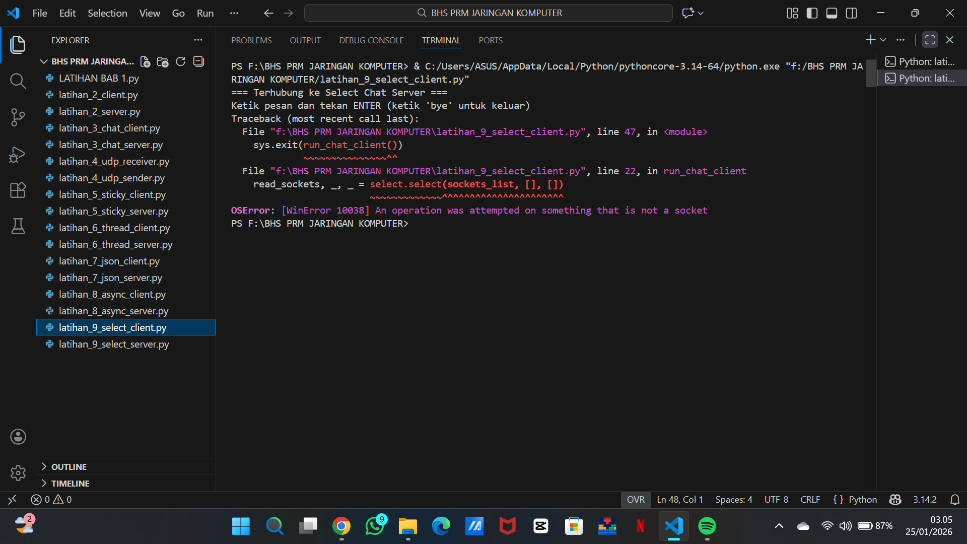
****

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

Pada Bab ini menjelaskan teknik multiplexing, yakni memantau banyak socket sekaligus tanpa harus menggunakan thread untuk setiap koneksi. Fungsi seperti select() dan poll() memungkinkan program menunggu event pada sekumpulan soket dan merespon hanya ketika ada data yang tersedia. Pendekatan ini sangat efisien untuk server single-threaded yang harus menangani banyak koneksi, dan bab ini memandu memahami bagaimana menggunakan mekanisme tersebut dengan benar untuk meningkatkan kemampuan pemrosesan aplikasi jaringan.

**Hasil :**

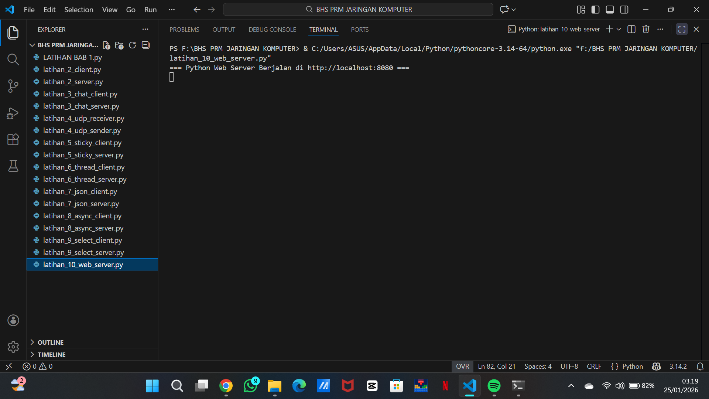
****

****

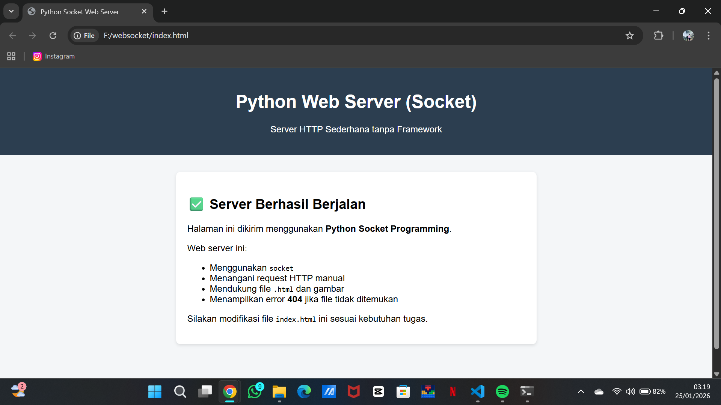
**Bab 10: Protokol HTTP & Web Server**

Pada Bab ini memberikan pembahasan mendalam tentang bagaimana HTTP bekerja serta bagaimana merancang web server sederhana menggunakan Python. Materinya mencakup format permintaan dan respon HTTP, status code, header, serta cara menyajikan file HTML dan resource lain ke browser. Pembaca belajar cara memproses request dari browser, membentuk respon yang sesuai, serta menangani berbagai format konten seperti teks, gambar, atau file lainnya.

**Hasil :**

****

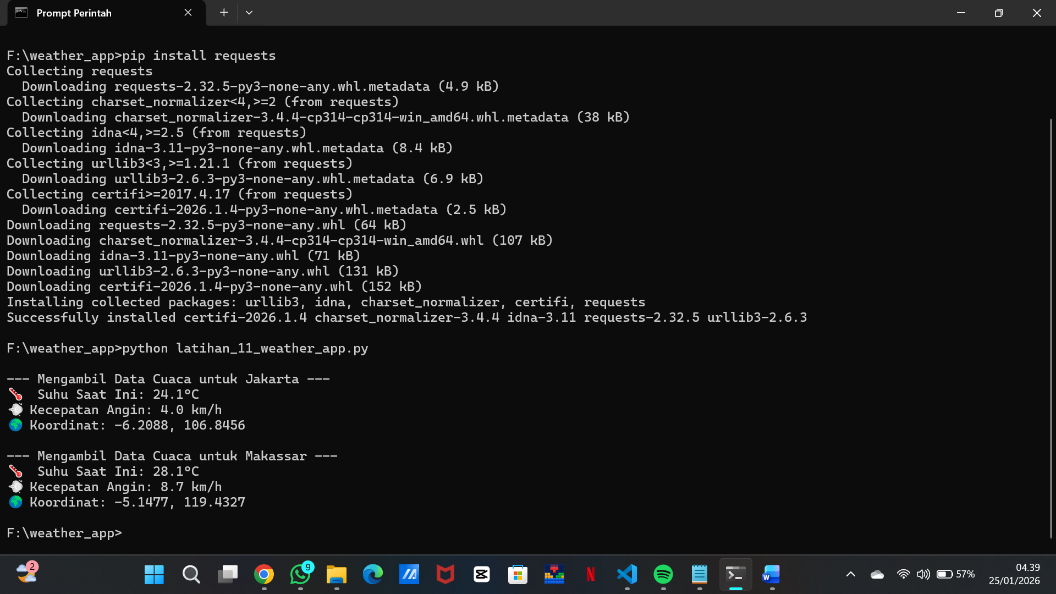
****

****

**Bab 11: REST API & Web Services**

Pada Bab ini memperkenalkan konsep REST API, pendekatan modern untuk membangun layanan web yang bisa diakses melalui HTTP dengan format pertukaran data JSON. Bab ini dibimbing bagaimana menyusun API server yang dapat diakses oleh client lain, serta bagaimana client mengambil atau mengirim data ke server melalui permintaan REST.

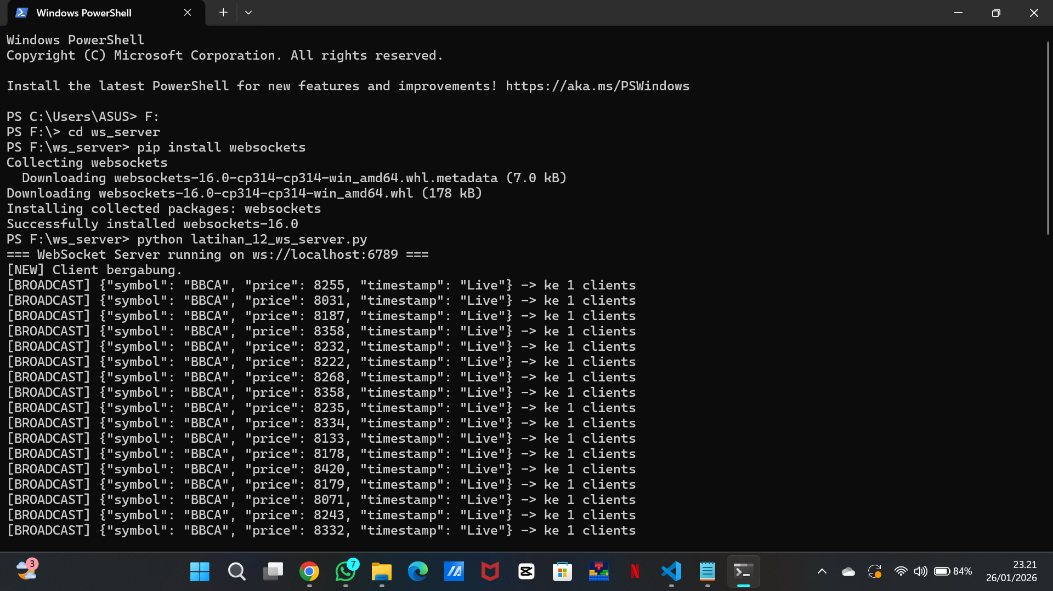
**Hasil :**

****

**Bab 12: Real-time Communication** **(WebSocket)**

Pada Bab ini membahas bagaimana WebSocket berbeda dari HTTP biasa karena koneksinya yang persisten dan full-duplex, sehingga cocok untuk aplikasi chat, game, atau notifikasi langsung. Bab ini juga mempelajari cara membuka koneksi WebSocket, menangani pesan masuk/keluar, serta menjaga koneksi tetap aktif selama sesi berlangsung.

**Hasil :**

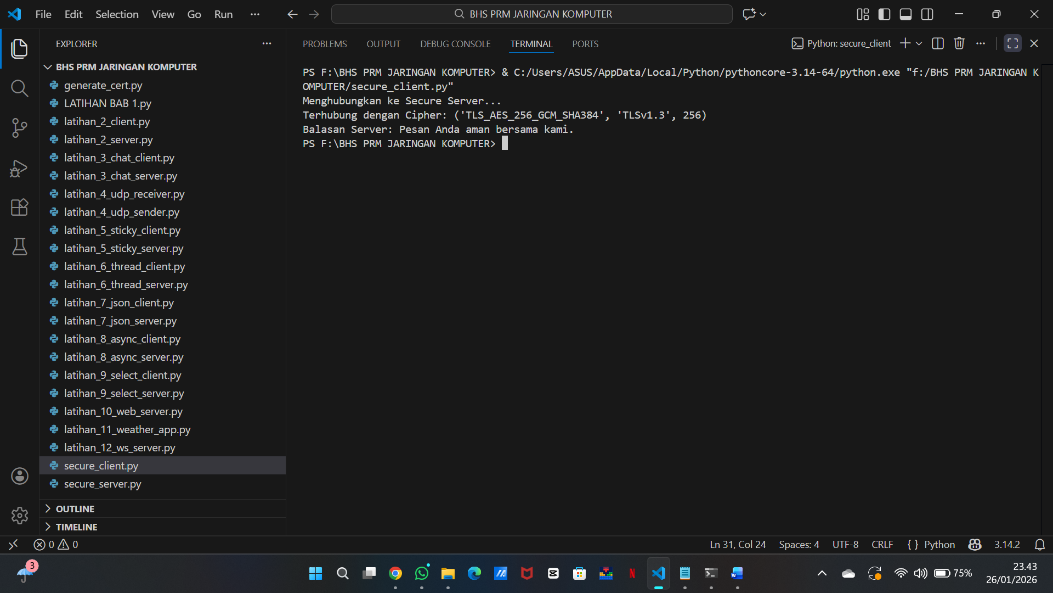
****

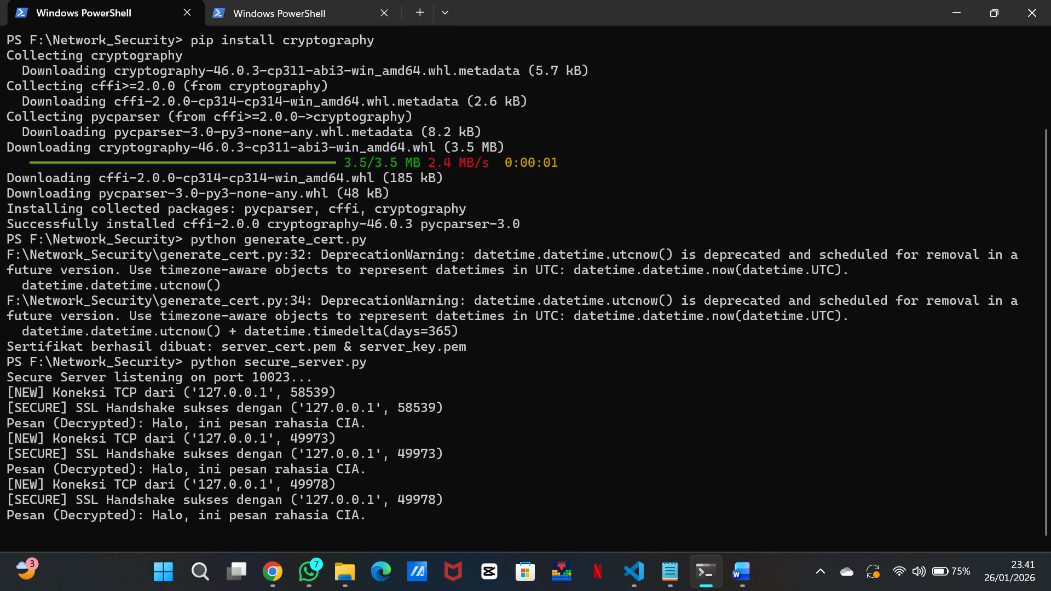
****

**Bab 13: Keamanan Jaringan (Network Security)**

Pada Bab ini menjelaskan berbagai konsep penting seperti enkripsi data, penggunaan SSL/TLS untuk koneksi aman, autentikasi client/server, serta proteksi terhadap serangan umum seperti DoS atau spoofing. Bab ini juga diajarkan praktik terbaik untuk menulis program jaringan yang aman dari eksploitasi, termasuk validasi input, penggunaan sertifikat digital, dan teknik mitigasi risiko agar aplikasi lebih tahan terhadap ancaman keamanan.

**Hasil :**

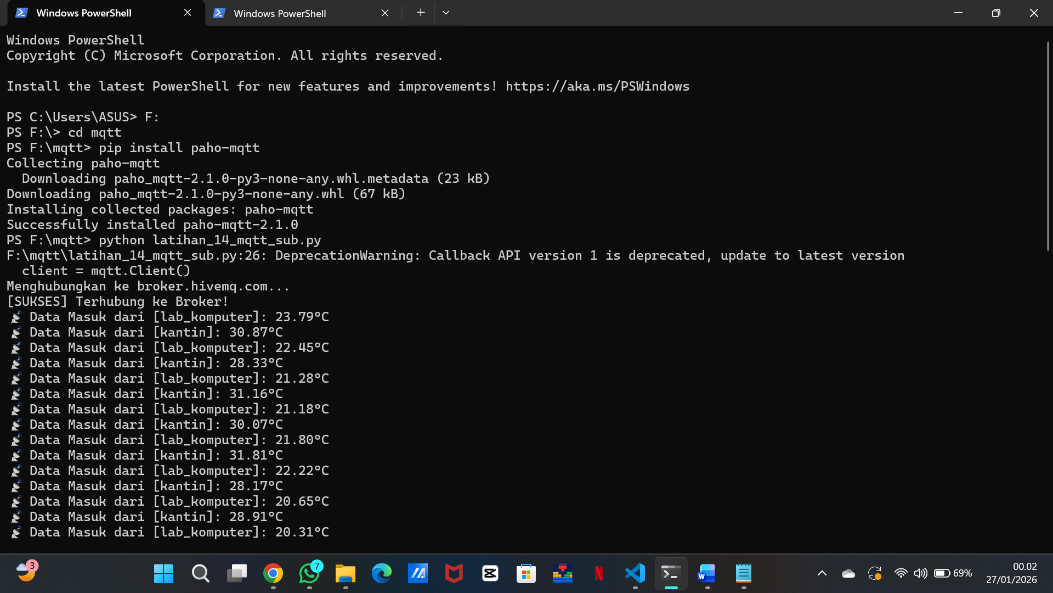
****

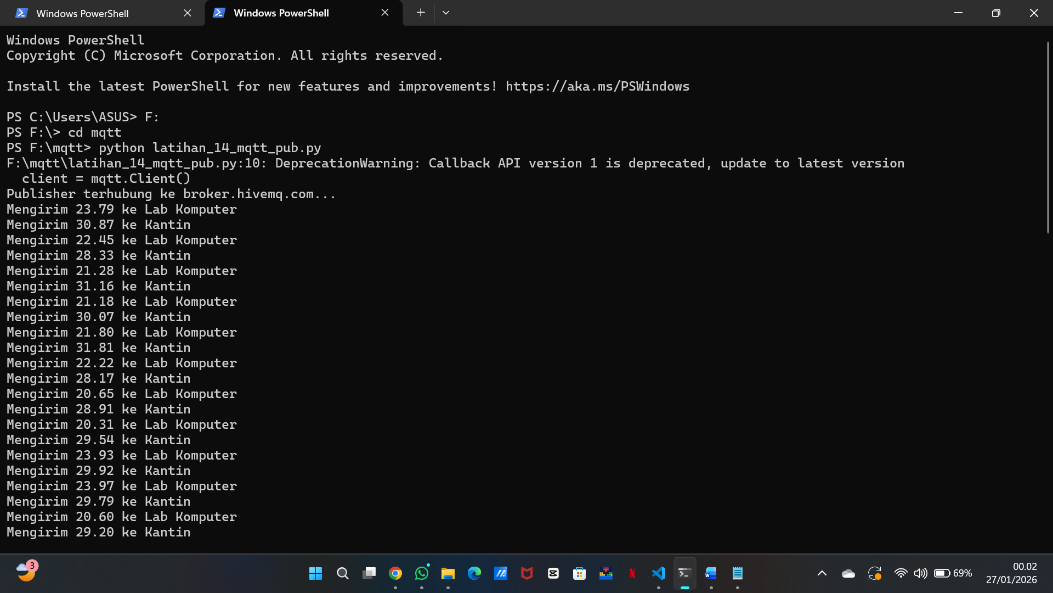
****

**Bab 14: Arsitektur Sistem Terdistribusi & IoT (MQTT)**

Pada Bab ini memperkenalkan protokol ringan seperti MQTT yang dirancang untuk komunikasi efisien antar perangkat resource-limited. Pembaca belajar cara kerja pola publish/subscribe MQTT, penggunaan broker MQTT, serta bagaimana perangkat dan aplikasi berbasis IoT dapat saling bertukar data secara efisien dalam sistem terdistribusi.

**Hasil :**

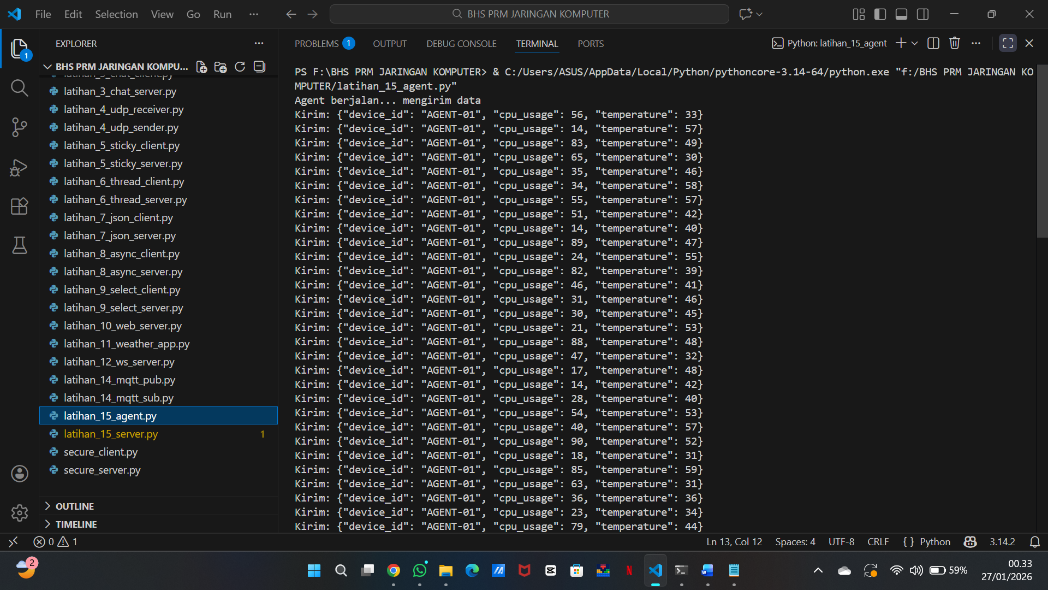
****

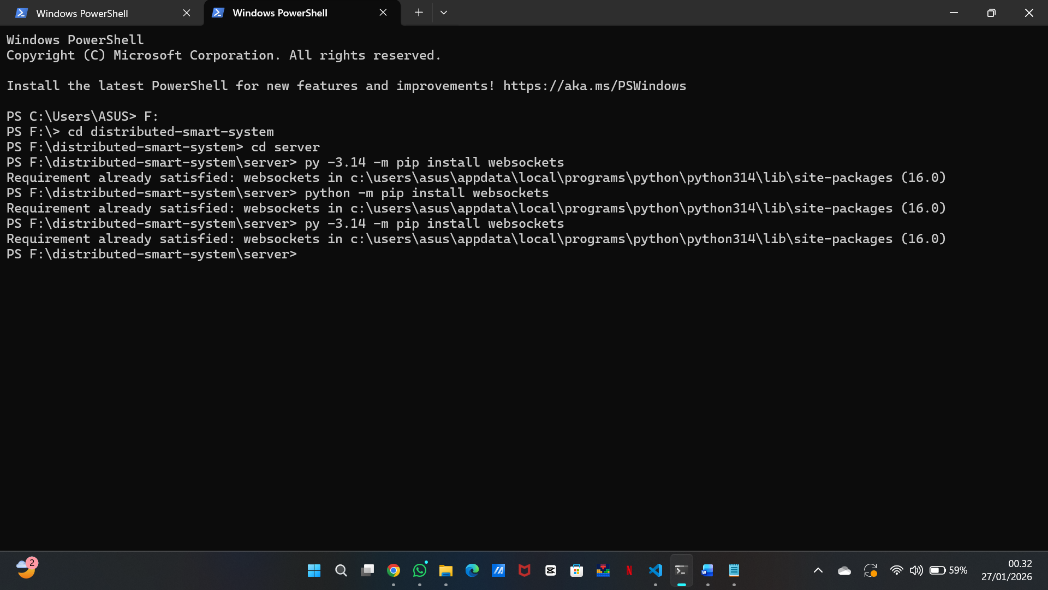
****

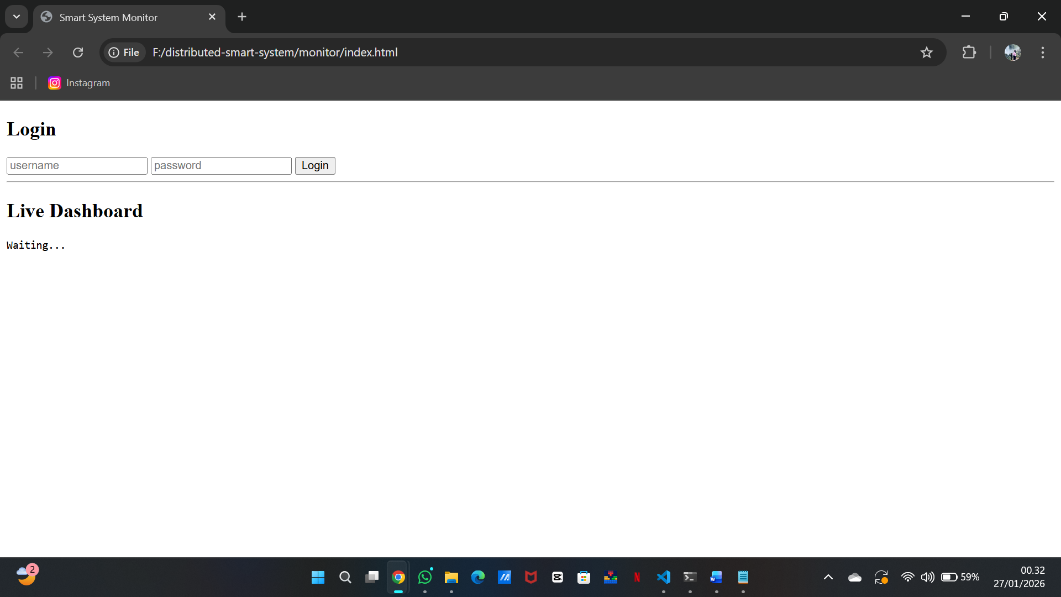
**Bab 15: Penutup & Proyek Akhir (Capstone Project)**

Bagian terakhir ini adalah proyek akhir yang mensintesis semua konsep yang telah dipelajari dari bab-bab sebelumnya. Pembaca diwajibkan merancang dan membangun sebuah aplikasi jaringan end-to-end yang mencakup berbagai teknik seperti socket programming, concurrency, protokol HTTP/REST, dan mungkin WebSocket atau IoT. Tujuannya adalah memastikan pembaca tidak hanya memahami teori, tapi juga mampu mengimplementasikan solusi nyata yang kompleks dan berfungsi penuh.

**Hasil :**

****

****

****