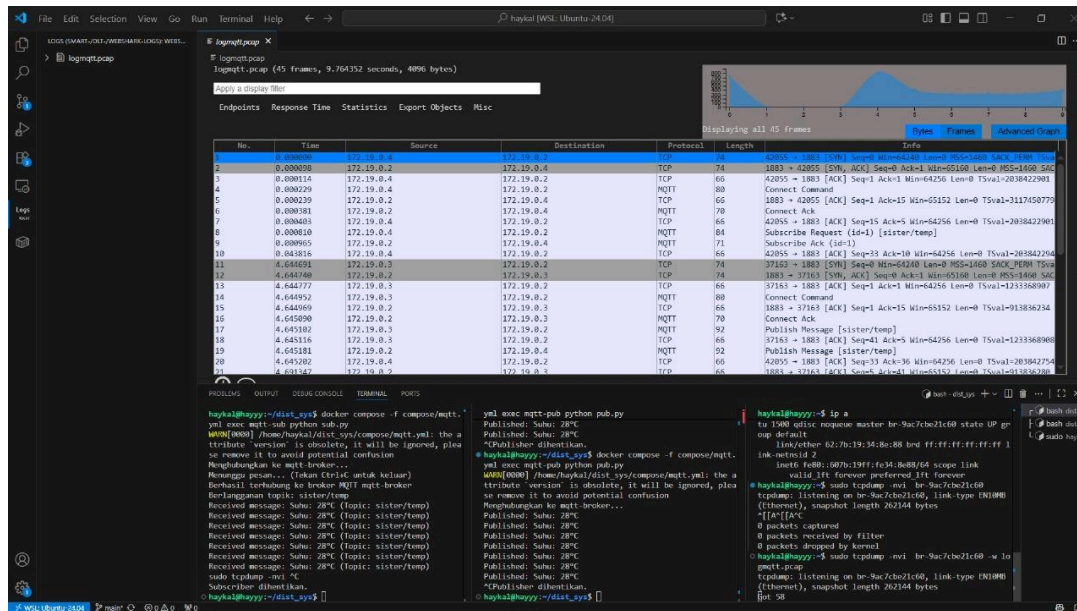
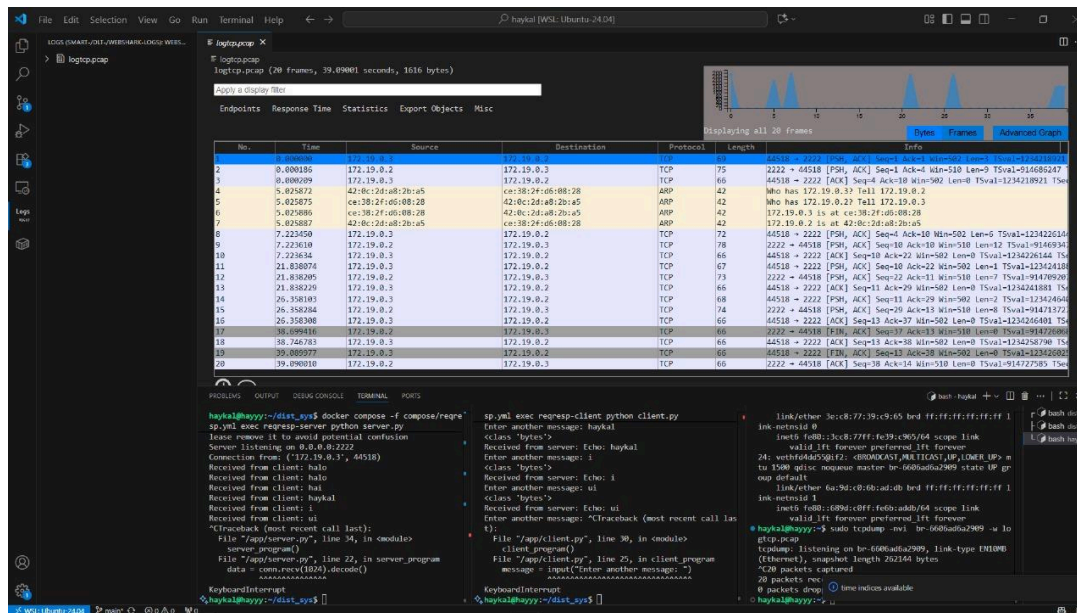


# Protokol MQTT



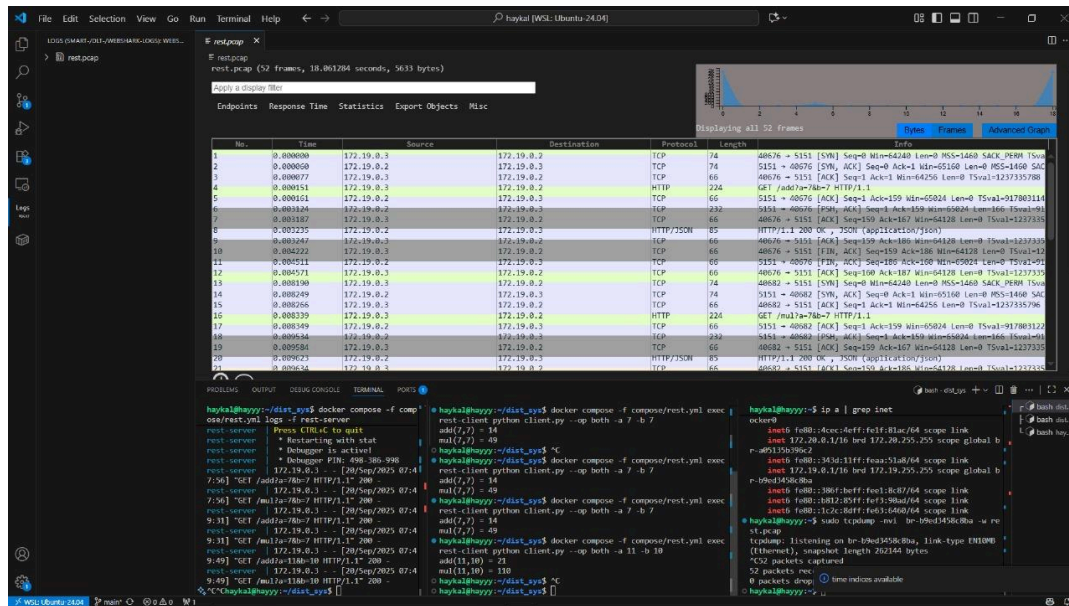
Wireshark sedang menangkap traffic data dari protokol MQTT. Terlihat adanya paket-paket yang berurutan, seperti CONNECT untuk memulai koneksi, SUBSCRIBE untuk berlangganan sebuah topik informasi, dan PUBLISH untuk mengirimkan pesan. Rangkaian paket yang terstruktur ini menunjukkan sebuah pola komunikasi Publish-Subscribe yang terkoordinasi melalui sebuah perantara (broker).

# Protokol TCP



Wireshark menangkap sesi komunikasi yang menggunakan protokol dasar TCP. Hanya terlihat paket-paket fundamental TCP dengan penanda [PSH, ACK], yang menandakan adanya proses pengiriman data dan konfirmasi penerimaan. Karena tidak ada protokol spesifik lain yang terdeteksi, ini mengindikasikan sebuah model komunikasi Client-Server yang bersifat langsung atau point-to-point.

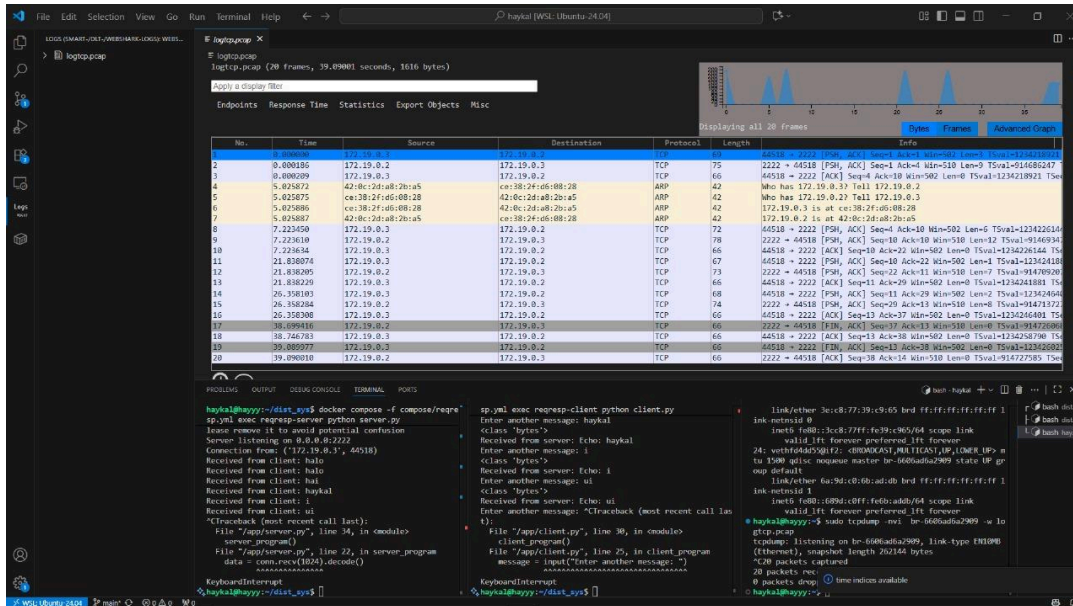
## Protokol Rest



Wireshark menangkap sesi komunikasi yang menggunakan protokol dasar TCP. Hanya terlihat paket-paket fundamental TCP dengan penanda [PSH, ACK], yang menandakan adanya proses pengiriman data dan konfirmasi penerimaan. Karena tidak ada protokol spesifik lain yang terdeteksi, ini mengindikasikan sebuah model komunikasi Client-Server yang bersifat langsung atau point-to-point.

## Protoko ZMQ

Clientserver

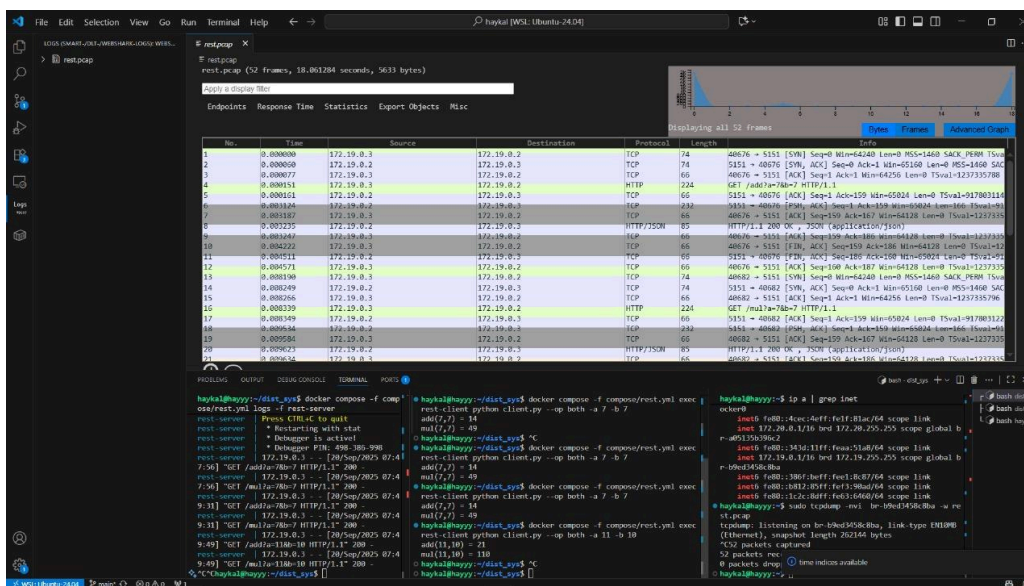


Pola ini digunakan untuk menghubungkan sekumpulan klien ke sekumpulan layanan (server). Cara kerjanya sangat terstruktur:

- Client akan mengirimkan sebuah pesan permintaan (Request) menggunakan soket REQ.
- Server akan menerima permintaan tersebut dengan soket REP, memproses nya, lalu wajib mengirimkan pesan balasan (Reply).

Komunikasi ini bersifat sinkron, artinya setelah klien mengirim permintaan, ia harus menunggu balasan dari server sebelum bisa mengirim permintaan berikutnya. Pola ini cocok digunakan untuk model komunikasi seperti RPC

## Push Pull





- Sebuah node pengirim dengan soket PUSH akan "mendorong" pesan atau tugas ke beberapa node penerima.
- Setiap node penerima dengan soket PULL akan "menarik" tugas yang masuk.

ZMQ secara otomatis akan mendistribusikan pesan dari PUSH secara adil (bergiliran) ke setiap soket PULL yang terhubung. Ini sangat efektif untuk load balancing atau menyebar beban kerja ke sekumpulan worker.

Wireshark network packet capture showing a sequence of TCP connections. The packet list displays 58 frames, all of which are TCP segments. The packet details pane shows the structure of a TCP segment, including the header and options. The packet bytes pane shows the raw data of the segment. The status bar at the bottom indicates that 58 packets were received and 0 packets were dropped.

- Publisher (PUB) akan menyiarkan pesan yang diberi "topik" tertentu. Publisher tidak perlu tahu siapa saja yang akan menerima pesannya.
- Subscriber (SUB) dapat memilih untuk hanya menerima pesan dari topik-topik yang ia minati. Jika sebuah pesan tidak sesuai dengan topik yang diikuti, pesan itu akan diabaikan.

Ini adalah pola komunikasi satu-ke-banyak yang sangat efisien untuk menyebarkan informasi atau pembaruan status ke semua bagian sistem yang tertarik tanpa harus terhubung satu per satu.