BAB I

Perkembangan untuk melakukan pemantauan, pengendalian, maupun akuisisi data secara real-time. Di lingkungan tersebut, erkembangan Internet of Things (IoT) mendorong semakin banyak perangkat yang terhubung ke jaringan keterbatasan sumber daya pada mikrokontroler serta karakteristik jaringan nirkabel menuntut penggunaan protokol komunikasi yang efisien dari sisi latensi, jitter, throughput, dan keandalan pengiriman pesan. Berbagai studi menegaskan bahwa tidak ada satu protokol yang selalu unggul di semua skenario, sehingga pemilihan protokol perlu disesuaikan dengan kebutuhan aplikasi dan kondisi jaringan yang dihadapi. Pada lapisan aplikasi, protokol HTTP, MQTT, dan WebSocket merupakan tiga pendekatan yang banyak digunakan dalam implementasi IoT modern. HTTP menawarkan kompatibilitas tinggi dengan layanan web dan kemudahan integrasi, tetapi membawa overhead yang relatif besar dan pola komunikasi request response yang kurang ideal untuk pengiriman data periodik. MQTT dirancang sebagai protokol publish/subscribe yang ringan di atas TCP, sehingga banyak direkomendasikan untuk perangkat dengan sumber daya terbatas dan jaringan dengan bandwidth rendah. WebSocket, di sisi lain, menyediakan kanal komunikasi full-duplex yang persisten di atas koneksi TCP dengan mekanisme upgrade dari HTTP, sehingga memungkinkan server mendorong data secara langsung ke klien dengan latensi rendah[1].

Berbagai penelitian telah membandingkan performa protokol-protokol tersebut pada beragam skenario. Studi tentang Quality of Service (QoS) HTTP dan MQTT pada sistem smart metering, misalnya, menunjukkan bahwa MQTT cenderung memberikan delay dan jitter yang lebih rendah dibandingkan HTTP pada lalu lintas data yang sering dan berukuran kecil[2]. Studi lain membandingkan MQTT dan WebSocket pada aplikasi IoT berbasis mikrokontroler ESP8266 dengan meninjau metrik round-trip time, konsumsi memori, dan ukuran payload; temuannya menunjukkan bahwa WebSocket lebih sesuai untuk kebutuhan latensi sangat rendah, sedangkan MQTT tetap lebih menguntungkan dari sisi kesederhanaan implementasi serta ketersediaan dukungan ekosistem.

Kajian berikutnya mengevaluasi HTTP, MQTT over TCP, dan MQTT over WebSocket pada arsitektur digital twin berbasis ESP32 yang terintegrasi dengan sensor DHT22 menggunakan indikator latensi dan stabilitas koneksi; hasil evaluasi mengindikasikan MQTT over TCP menghasilkan latensi rata-rata terendah, sementara HTTP memiliki latensi tertinggi tetapi menawarkan kompatibilitas sistem yang lebih luas[3], [4].

Penelitian terkini juga menganalisis latensi WebSocket dibandingkan protokol industri seperti Modbus dan OPC UA pada integrasi digital twin, dan menegaskan bahwa WebSocket mampu memenuhi kebutuhan respon di bawah 100ms untuk sistem real-time tertentu[5].

Meskipun telah banyak dilakukan studi komparatif, beberapa keterbatasan masih dapat diidentifikasi. Pertama, sebagian besar penelitian hanya membandingkan dua protokol sekaligus, misalnya HTTP vs MQTT atau MQTT vs WebSocket, sehingga gambaran trade-off ketika ketiganya diterapkan pada perangkat dan jaringan yang sama belum sepenuhnya komprehensif[6].

Kedua, skenario pengujian umumnya berfokus pada aplikasi spesifik (misalnya smart metering, digital twin, atau push notification) dengan pola lalu lintas tertentu, sehingga pengaruh variasi frekuensi pengiriman data secara sistematis khususnya pada rentang 1-50hz yang relevan untuk banyak aplikasi pemantauan dan kontrol belum banyak dieksplorasi.

Ketiga, beberapa penelitian menyoroti latensi dan throughput, tetapi jarang mengukur metrik lain seperti jitter per paket, packet loss rate, serta penggunaan memori pada mikrokontroler sebagai indikator keberlanjutan sistem ketika beban trafik meningkat. Berdasarkan celah tersebut, penelitian ini bertujuan melakukan evaluasi kinerja protokol HTTP, MQTT, dan WebSocket pada node IoT berbasis ESP8266 dengan pendekatan eksperimen terkontrol. Paket data dikirimkan secara periodik pada sebelas frekuensi berbeda (1, 5, 10, 15, 20, 25, 30, 35, 40, 45, dan 50hz) selama 120 detik untuk setiap skenario.

Meskipun studi terdahulu telah banyak mengevaluasi protokol IoT, sebagian besar pengujian dilakukan pada kondisi trafik rendah atau frekuensi statis yang tidak merepresentasikan dinamika aplikasi nyata. Kesenjangan penelitian yang diisi oleh studi ini terletak pada analisis perilaku protokol saat menghadapi eskalasi beban trafik pada perangkat dengan sumber daya terbatas (ESP8266).

Secara ilmiah, pengujian dengan variasi frekuensi 1-50hz krusial untuk memetakan fenomena non-linear pada metrik QoS seperti lonjakan jitter dan packet loss ketika offered load mendekati kapasitas pemrosesan maksimal perangkat. Secara praktis, rentang 1-50hz mencakup spektrum kebutuhan aplikasi dari pemantauan lingkungan (frekuensi rendah) hingga sensor industri/medis (frekuensi tinggi). Tanpa pemahaman mengenai titik degradasi kinerja pada rentang frekuensi ini, pengembang berisiko menerapkan protokol yang tampak stabil di laboratorium namun gagal menjaga sinkronisasi data pada implementasi IoT nyata yang memerlukan presisi waktu tinggi[7].

Di sisi server, aplikasi pencatat berbasis HTTP, broker MQTT, dan server WebSocket yang dikembangkan khusus melakukan logging terstruktur ke berkas CSV. Parameter yang diukur meliputi akurasi interval pengiriman, latency end-to-end dalam bentuk round-trip time (RTT), jitter sebagai deviasi standar RTT, throughput dalam byte per detik dan pesan per detik, packet loss rate, serta nilai free heap memory pada ESP8266. Secara umum, kontribusi utama penelitian ini adalah: (1) menyajikan rancangan testbed dan metodologi pengukuran RTT langsung di tingkat aplikasi untuk tiga protokol IoT populer pada perangkat ESP8266 yang sama; (2) memberikan analisis kuantitatif pengaruh variasi frekuensi pengiriman 1-50hz terhadap latency, jitter, throughput, dan packet loss masing-masing protokol; (3) mengevaluasi penggunaan memori sebagai indikator efisiensi implementasi protokol pada perangkat IoT.

Dalam implementasi IoT di lapangan, perbedaan kinerja protokol pada metrik QoS (latency, jitter, throughput, keandalan pengiriman, dan dampaknya terhadap sumber daya perangkat) tidak hanya bersifat teknis, tetapi menentukan keberhasilan operasional langsung pada mutu layanan aplikasi. Pada sistem smart

healthcare, misalnya pemantauan detak jantung atau parameter fisiologis yang ditampilkan ke dashboard secara kontinu, konsistensi waktu kedatangan data (jitter rendah) menjadi krusial karena fluktuasi delay dapat mengganggu interpretasi tren dan memengaruhi ketepatan notifikasi[8].

Pada smart agriculture, data lingkungan seperti kelembapan tanah, suhu, atau pH umumnya dikirim periodik dengan frekuensi rendah menengah; konsekuensi praktisnya adalah protokol yang stabil pada trafik periodik dan mudah diintegrasikan ke layanan web tetap relevan, sementara efisiensi bandwidth menjadi pertimbangan ketika node tersebar luas[9].

Pada industrial monitoring dan kontrol, terutama ketika data sensor atau status aktuator perlu dipantau mendekati real-time, kombinasi latency rendah dan jitter kecil berkontribusi terhadap respons yang “halus” pada kontrol dan sinkronisasi tampilan pemantauan, sehingga pemilihan protokol menjadi determinan kualitas operasi sistem. Kebutuhan-kebutuhan ini selaras dengan argumen bahwa pemilihan protokol harus disesuaikan dengan karakteristik aplikasi dan kondisi jaringan, serta tidak ada protokol yang selalu unggul pada semua skenario[10].

Temuan ini diharapkan dapat menjadi acuan praktis dalam pemilihan protokol komunikasi yang paling sesuai dengan kebutuhan aplikasi IoT, khususnya ketika harus menyeimbangkan antara persyaratan latensi, kestabilan waktu, dan keterbatasan sumber daya perangkat[11]..