

Desain dan Implementasi Middleware Layer Untuk Kendali Mobil Listrik Jarak Jauh Dengan Protokol HTTP Dan MQTT

Oleh: Rustam Asnawi, PhD., Prof. Samsul Hadi, Prof. Moh. Khairudin, Dr. Sarwo Pranoto, Dr. Herlambang Sigit Pramono

ABSTRAK

Penelitian tentang kendaraan listrik saat ini terus berproses dan semakin banyak penemuan baru terkait perkembangan teknologi pada kendaraan listrik. Penggunaan kendaraan listrik jarak jauh memerlukan sistem kontrol yang andal dan efisien. Lapisan middleware, dapat diterapkan sebagai penghubung antara kendaraan listrik dan sistem kendali jarak jauh sehingga antara pengguna dan sistem mobil listrik yang dikendalikan lebih aman dan efisien karena menerapkan sistem monitoring dan control cerdas berbasis protocol HTTP dan/atau MQTT di middleware layer.

Tujuan perancangan dan implementasi lapisan middleware untuk kontrol mobil listrik jarak jauh dengan protokol HTTP dan MQTT adalah untuk menciptakan sistem kontrol yang lebih aman, andal, dan efisien. Lapisan middleware ini mencakup sistem pemantauan, kontrol, dan komunikasi, di antara komponen lainnya. Protokol HTTP digunakan untuk Komunikasi data menggunakan cara permintaan dan tanggapan antara klien dan server, sedangkan MQTT digunakan untuk mengirim data dalam format publish dan subscribe.

Secara umum penelitian ini melalui beberapa tahapan. **Tahap pertama, Identifikasi Tujuan Penelitian.** Penelitian ini bertujuan untuk mengkaji efektivitas desain dan implementasi dari sistem middleware dalam kendali jarak jauh kendaraan listrik dengan menggunakan protokol HTTP dan MQTT. Tujuan lain, untuk mengidentifikasi kelebihan dan kekurangan dari masing-masing protokol dan membandingkan kinerjanya. **Tahap kedua, Desain Penelitian.** Penelitian ini menggunakan desain penelitian kuantitatif dengan pendekatan studi kasus. Pengujian dilakukan dengan cara mengirimkan beberapa perintah dari pengontrol jarak jauh ke kendaraan listrik (dalam penelitian ini menggunakan simulasi modul luaran seperti relay, dan lampu) dan mengamati respons yang diberikan oleh kendaraan. **Tahap ketiga, Pengumpulan Data.** Pengumpulan data dilakukan dengan menguji respons sistem middleware terhadap perintah-perintah yang dikirimkan dari pengontrol jarak jauh. **Tahap keempat, Analisis Data.** Data dianalisis menggunakan teknik analisis deskriptif. Analisis data juga akan dilakukan untuk membandingkan kinerja masing-masing protokol dan menentukan kelebihan dan kekurangannya. **Tahap kelima, Interpretasi dan Pelaporan Hasil Penelitian.** Hasil penelitian akan disajikan dalam bentuk laporan penelitian yang dilengkapi dengan tabel, grafik, dan narasi untuk memudahkan pemahaman. **Tahap keenam, Evaluasi Hasil Penelitian.** Evaluasi juga akan dilakukan untuk menilai kekurangan dan kelebihan dari desain penelitian dan metode yang digunakan sehingga dapat memberikan masukan untuk penelitian selanjutnya.

Pengujian dan analisis unjukkerja penggunaan berbagai server HTTP dan MQTT Protocol berdasarkan data latency telah dilakukan. Server HTTP digunakan thingspeak dan personal website. Hasil pengujian kedua server http tersebut menunjukkan bahwa server Thingspeak lebih variative nilai latensinya dan cenderung lebih besar latensinya dibandingkan dengan webserver pribadi. Sementara itu untuk pengajian protocol MQTT menggunakan Server MQTT HiveMQ, EMQX, dan Adafruit menunjukkan bahwa di ketiga server MQTT tersebut stabilitas, responsivitas cepat, serta konsistensi latency rendah dan mampu memberikan layanan komunikasi yang handal dan efisien.

Penelitian selanjutnya adalah menganalisis, mendesain dan mengimplementasi REST API untuk mendukung penerapan protocol V2X. Ada lima API utama yang berhasil dikembangkan. API untuk membaca dan menampilkan seluruh data dari sebuah tabel yang ada di database serta menampilkannya dalam bentuk json dan text. API untuk membaca data terakhir dari sebuah tabel yang ada di database serta menampilkannya dalam bentuk json dan text. API untuk menyimpan data baru ke sebuah tabel yang ada di database. Hal ini untuk mengendalikan actuator yang ada di kendaraan listrik: seperti lampu beam, lampu sein, rem dll. API untuk menghapus sebuah data yang ada di sebuah tabel dalam database. API untuk menghapus semua data yang ada di sebuah tabel dalam database.

Pengujian dilakukan dengan simulasi pengiriman data hasil akuisisi data sensor-sensor oleh ESP32. Selanjutnya ESP32 akan mengirimkan data hasil akuisisi sensor-sensor tadi ke server database MySQL menggunakan sintak API yang ada dalam REST API tersebut. Hasilnya akan diperoleh data terakhir yang dikirim ke tabel sensor1 hingga sensor10 benar sudah masuk dan dapat ditampilkan di browser dalam format json. Adapun rata-rata lama waktu yang dibutuhkan untuk mengerjakan instruksi API tersebut sekitar 1 detik.

Kata Kunci: *middleware application, REST, API, ESP32 mikrokontroler*