

Rancang Bangun Sistem Smart Parking Berbasis RFID dan ESP32 Dengan Komunikasi Data Menggunakan SIM7600G

Raihan Riyan Pratama¹, Andi Ahmad Dahlan², Ramiati³

^{1,2,3} Program Studi Teknik Telekomunikasi, Politeknik Negeri Padang

e-mail: raihanriyanp@gmail.com

Abstrak

Sistem parkir konvensional memiliki keterbatasan dalam proses identifikasi pengguna, pengendalian akses, dan pemantauan aktivitas parkir. Artikel ini membahas rancang bangun sistem Smart Parking berbasis RFID MFRC522 dan ESP32 dengan komunikasi data menggunakan modul SIM7600G melalui protokol Message Queuing Telemetry Transport (MQTT). Sistem dirancang untuk membaca UID kartu RFID, melakukan validasi pengguna melalui MQTT Broker pada Raspberry Pi, serta mengendalikan palang parkir menggunakan motor wiper 12 V dan driver BTS7960. Node-RED digunakan sebagai middleware untuk memproses data dan melakukan validasi terhadap database MySQL, sedangkan hasil aktivitas parkir ditampilkan melalui Web Admin dan aplikasi Android berbasis MIT App Inventor. Pengujian fungsional meliputi pembacaan RFID, konektivitas SIM7600G, komunikasi MQTT, pemrosesan Node-RED, mekanisme palang, sensor ultrasonik, website, dan aplikasi Android. Pengujian performa komunikasi dilakukan sebanyak 50 kali untuk setiap konfigurasi QoS 0, QoS 1, dan QoS 2 menggunakan Wireshark dengan parameter delay dan packet loss. Hasil pengujian menunjukkan delay rata-rata masing-masing sebesar 2,361 ms, 72,550 ms, dan 281,732 ms, sedangkan packet loss pada seluruh konfigurasi sebesar 0%. Pengukuran waktu respons sistem pada kondisi normal sebanyak lima kali menghasilkan rata-rata 13,51 detik dari proses tap RFID hingga palang kembali tertutup. Hasil tersebut menunjukkan bahwa sistem mampu menjalankan identifikasi, validasi, pengendalian palang, pencatatan data, dan monitoring sesuai rancangan.

Kata Kunci: *Smart Parking*, RFID, ESP32, SIM7600G, MQTT

Abstract

Conventional parking systems have limitations in user identification, access control, and parking activity monitoring. This article presents the design and development of a Smart Parking system based on MFRC522 RFID and an ESP32 microcontroller, with data communication using a SIM7600G module through the Message Queuing Telemetry Transport (MQTT) protocol. The system was designed to read RFID card UIDs, validate users through an MQTT Broker on a Raspberry Pi, and control a parking barrier using a 12 V wiper motor and BTS7960 driver. Node-RED was used as middleware to process data and validate users against a MySQL database, while parking activity was displayed through a Web Admin interface and an Android application developed with MIT App Inventor. Functional tests covered RFID reading, SIM7600G connectivity, MQTT communication, Node-RED processing, barrier operation, ultrasonic sensing, website monitoring, and Android monitoring. Communication performance was tested 50 times for each MQTT QoS configuration, namely QoS 0, QoS 1, and QoS 2, using Wireshark with delay and packet loss as the evaluation parameters. The average delays were 2.361 ms, 72.550 ms, and 281.732 ms for QoS 0, QoS 1, and QoS 2, respectively, while packet loss was 0% for all configurations. The system response time under normal conditions was measured five times and averaged 13.51 seconds from RFID tapping until the barrier returned to the closed

position. The system performs identification, validation, barrier control, data logging, and monitoring as designed.

Keywords : *Smart Parking, RFID, ESP32, SIM7600G, MQTT*

LATAR BELAKANG

Perkembangan mobilitas masyarakat meningkatkan kebutuhan terhadap fasilitas parkir yang cepat, aman, dan efisien. Sistem parkir yang masih manual atau semiotomatis dapat menimbulkan antrean, kesalahan pencatatan, keterlambatan identifikasi, dan kurang optimalnya pemanfaatan area parkir. Penerapan Internet of Things (IoT) memungkinkan proses identifikasi, pengendalian akses, pencatatan, dan pemantauan parkir dilakukan secara terintegrasi dan real-time. Kajian mengenai smart parking menunjukkan bahwa integrasi RFID, mikrokontroler, sensor, dan sistem monitoring dapat meningkatkan efisiensi pengelolaan parkir (Biyik et al., 2021; Elfaki et al., 2023; Aditya et al., 2023). RFID juga banyak digunakan karena mampu memberikan identifikasi pengguna tanpa kontak fisik, sedangkan ESP32 menyediakan kemampuan pemrosesan dan antarmuka yang sesuai untuk integrasi perangkat IoT (Atiqur, 2021; Simatupang et al., 2022).

Sebagian implementasi smart parking masih menggunakan Wi-Fi atau komunikasi jarak pendek sebagai media konektivitas. Pendekatan tersebut sesuai untuk lingkungan dengan infrastruktur jaringan lokal yang tersedia, tetapi dapat menjadi keterbatasan ketika perangkat ditempatkan pada area terbuka atau lokasi yang tidak memiliki akses Wi-Fi yang stabil. Beberapa sistem yang menggabungkan ESP32, RFID, sensor ultrasonik, dan MQTT telah menunjukkan kemampuan monitoring real-time, tetapi belum seluruhnya memanfaatkan jaringan seluler sebagai jalur komunikasi utama (Sari et al., 2024; Yuvier et al., 2025; Nafisy et al., 2025). Penggunaan komunikasi seluler melalui SIM7600G memberikan alternatif konektivitas berbasis jaringan 4G LTE sehingga perangkat lapangan dapat terhubung ke server melalui internet tanpa bergantung pada jaringan Wi-Fi lokal.

Berdasarkan kondisi tersebut, sistem pada artikel ini dirancang menggunakan RFID MFRC522 dan ESP32 sebagai perangkat lapangan, SIM7600G sebagai media komunikasi seluler, serta MQTT sebagai protokol pertukaran data. Raspberry Pi digunakan sebagai server yang menjalankan MQTT Broker, Node-RED, dan MySQL. Data UID hasil pembacaan RFID dikirim ke server untuk divalidasi, kemudian hasil validasi digunakan untuk menentukan akses palang. Sistem juga dilengkapi sensor ultrasonik sebagai pengaman ketika kendaraan masih berada di bawah palang serta sensor inframerah sebagai pembatas posisi palang. Informasi aktivitas dan kondisi parkir selanjutnya ditampilkan melalui Web Admin dan aplikasi Android.

METODE

Metode yang digunakan berupa perancangan, implementasi, dan pengujian prototipe Smart Parking. Tahapan pekerjaan meliputi perancangan arsitektur sistem, perakitan perangkat keras, pengembangan perangkat lunak, integrasi, serta pengujian fungsional dan performa komunikasi. Arsitektur sistem menghubungkan node lapangan dengan server melalui jaringan seluler 4G LTE. Pada sisi lapangan, ESP32 membaca UID dari RFID MFRC522 dan mengendalikan aktuator palang. SIM7600G menyediakan koneksi data seluler dan menjadi jalur komunikasi MQTT. Pada sisi server, Raspberry Pi menjalankan MQTT Broker Mosquitto sebagai perantara publish-subscribe, Node-RED sebagai middleware pemrosesan, dan MySQL sebagai basis data. Web Admin dan aplikasi Android digunakan sebagai antarmuka monitoring.

Perangkat keras utama terdiri atas ESP32 DevKit V1, RFID MFRC522, SIM7600G, sensor ultrasonik JSN-SR04T, dua sensor inframerah E18-D80NK, driver motor BTS7960, motor wiper 12 V, LCD I2C 16×2, serta catu daya. RFID dihubungkan melalui SPI, LCD melalui I2C, SIM7600G melalui UART, sensor ultrasonik melalui pin trigger dan echo, sedangkan driver BTS7960 dikendalikan menggunakan sinyal PWM. Sensor inframerah

dipasang pada posisi palang 90° dan 0° sebagai limit sensor. Sensor ultrasonik digunakan untuk mendeteksi kendaraan di bawah palang dan menunda proses penutupan ketika objek masih terdeteksi. Konfigurasi perangkat keras tersebut direalisasikan menjadi satu node lapangan dan diintegrasikan dengan sistem mekanik palang.

Alur kerja dimulai ketika kartu RFID ditap. ESP32 membaca UID dan membentuk payload, kemudian mengirimkannya pada topic smartparking/rfid melalui SIM7600G. MQTT Broker meneruskan data kepada Node-RED. UID dicocokkan dengan data pada tabel data_pelanggan MySQL. Jika UID valid, Node-RED mengirimkan status VALID pada topic smartparking/result dan ESP32 menggerakkan palang menuju posisi terbuka. Setelah kendaraan melewati area deteksi, palang ditutup hingga mencapai limit sensor. Jika UID tidak valid, palang tetap tertutup. Data transaksi, status parkir, dan informasi pengguna disimpan pada database dan dapat dipantau melalui Web Admin serta aplikasi Android.

Pengujian fungsional meliputi pembacaan RFID, konektivitas SIM7600G, komunikasi MQTT, pemrosesan Node-RED, mekanisme palang, sensor ultrasonik, website, aplikasi Android, dan integrasi sistem. Pengujian performa komunikasi dilakukan menggunakan Wireshark. Masing-masing QoS diuji sebanyak 50 kali tap RFID. Delay dihitung dari selisih timestamp pesan publish UID dan pesan hasil validasi, sedangkan packet loss dihitung dari perbandingan jumlah paket yang dikirim dan diterima. Persamaan yang digunakan adalah:

$$Delay = t_{\text{terima}} - t_{\text{kirim}}$$

$$PL = \frac{\text{Jumlah Paket Dikirim} - \text{Jumlah Paket Diterima}}{\text{Jumlah Paket Dikirim}} \times 100\%$$

Pengujian waktu respons dilakukan lima kali pada kondisi normal, mulai dari tap kartu RFID sampai palang kembali ke posisi tertutup. Data hasil pengukuran kemudian direkapitulasi untuk memperoleh nilai minimum, maksimum, dan rata-rata delay serta nilai packet loss.

Tabel 1. Ringkasan pengujian fungsional sistem

Bagian yang diuji	Hasil	Status
RFID MFRC522	UID terdaftar dan tidak terdaftar terbaca serta tervalidasi	Berhasil
SIM7600G	Modul terhubung ke jaringan dan layanan data	Berhasil
MQTT	UID dikirim dan respons validasi diterima pada QoS 0-2	Berhasil
Node-RED/MySQL	Validasi UID, log, dan status parkir diproses	Berhasil
Palang + sensor	Buka/tutup dan interlock keselamatan bekerja	Berhasil
Web Admin + Android	Data monitoring diperbarui sesuai kondisi sistem	Berhasil

Pengujian monitoring dilakukan setelah data transaksi berhasil diproses oleh server. Web Admin digunakan untuk melihat informasi UID, identitas pengguna, waktu transaksi, status akses, dan kondisi parkir, sedangkan aplikasi Android digunakan untuk menampilkan total slot, slot tersedia, kendaraan di dalam, dan status koneksi. Dengan demikian, pengujian tidak hanya menilai keberhasilan komunikasi antara ESP32 dan server, tetapi juga memastikan hasil pemrosesan dapat diteruskan hingga lapisan antarmuka pengguna.

Pengujian mekanisme palang dilakukan dengan mempertimbangkan dua kondisi utama, yaitu akses pengguna dan kondisi keselamatan. UID yang terdaftar digunakan untuk memicu pembukaan palang, sedangkan UID yang tidak terdaftar digunakan untuk memastikan palang tetap tertutup. Setelah palang terbuka, sensor ultrasonik digunakan untuk mendeteksi keberadaan kendaraan di bawah palang. Selama kendaraan masih terdeteksi pada kondisi pengujian, perintah penutupan ditahan. Setelah area dinyatakan aman, palang dapat bergerak menuju posisi tertutup dan dihentikan oleh sensor inframerah pada posisi sekitar 0°.

Pada pengujian komunikasi, data UID dikirim oleh ESP32 melalui SIM7600G menuju MQTT Broker pada Raspberry Pi. Node-RED menerima pesan pada topic smartparking/rfid, memproses UID, kemudian melakukan pencocokan dengan tabel data_pelanggan pada MySQL. Hasil pencocokan dikirim kembali melalui topic smartparking/result. Alur tersebut digunakan sebagai dasar pengukuran delay komunikasi sekaligus untuk memastikan bahwa data yang dikirim dari perangkat lapangan dapat diproses dan menghasilkan respons yang sesuai.

Untuk memastikan setiap bagian sistem bekerja sesuai fungsi yang dirancang, pengujian dilakukan secara bertahap mulai dari pengujian komponen, pengujian komunikasi, pengujian proses validasi, hingga pengujian integrasi. Pengujian komponen berfokus pada kemampuan RFID MFRC522 membaca UID, kemampuan SIM7600G memperoleh layanan jaringan dan data, serta kemampuan sensor dan aktuator merespons kondisi masukan. Setelah masing-masing bagian berfungsi, seluruh subsistem diintegrasikan untuk melihat kesinambungan proses dari identifikasi pengguna sampai pengendalian palang dan pencatatan transaksi.

Tabel 2. Skenario pengujian sistem dan parameter evaluasi

Parameter	Metode pengujian	Jumlah pengujian	Indikator
RFID MFRC522	Tap kartu terdaftar dan tidak terdaftar Pemeriksaan AT Command, SIM, sinyal, registrasi, dan layanan data	Observasi fungsional	UID terbaca dan status akses sesuai
SIM7600G	Pengiriman UID dan penerimaan respons validasi	Observasi fungsional	Modul siap komunikasi data
MQTT QoS	Pencocokan UID dan pencatatan transaksi	50 kali/QoS	Delay dan packet loss
Node-RED/MySQL	Akses valid/tidak valid dan gerakan buka-tutup	Observasi integrasi	Validasi dan log sesuai
Palang	Objek pada 25 cm, 35 cm, tanpa objek, dan >100 cm	Observasi fungsional	Posisi palang sesuai kondisi
Sensor keselamatan	Pemeriksaan Web Admin dan Android	Observasi fungsional	Penutupan tertahan/diizinkan sesuai kondisi
Monitoring	Tap RFID sampai palang kembali tertutup	Observasi integrasi	Data dan status diperbarui
Waktu respons		5 kali	Nilai minimum, maksimum, rata-rata

HASIL DAN PEMBAHASAN

Hasil implementasi menunjukkan bahwa seluruh subsistem dapat diintegrasikan menjadi prototipe Smart Parking. RFID MFRC522 berhasil membaca UID kartu terdaftar maupun tidak terdaftar. Pada pengujian fungsional, UID 0E403A038CB738 dan 69FE2EC2

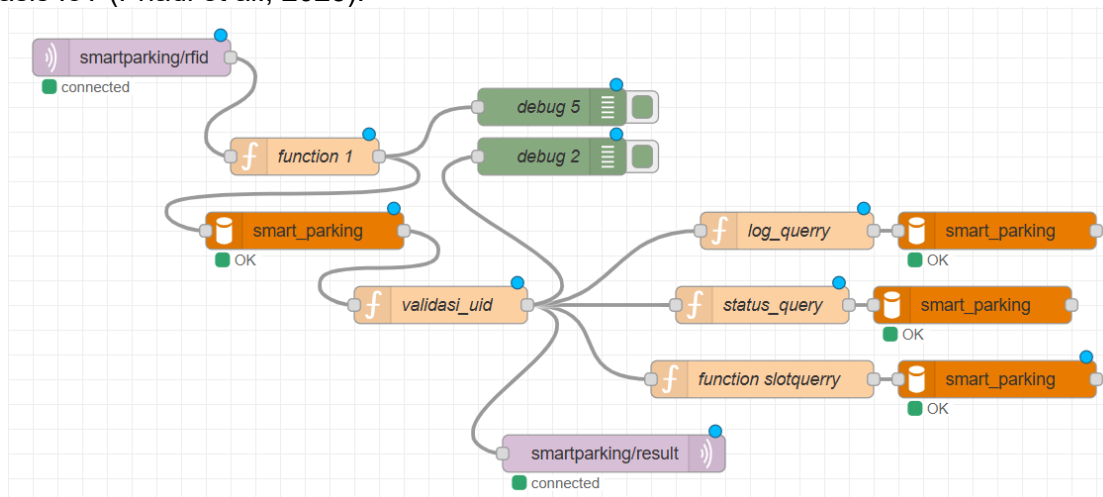
berhasil dibaca dan divalidasi sebagai pengguna terdaftar sehingga palang terbuka, sedangkan UID 3352AB0F terbaca tetapi tidak ditemukan pada basis data sehingga sistem memberikan status INVALID dan palang tetap tertutup. Hasil ini menunjukkan bahwa proses identifikasi dan validasi dapat digunakan sebagai dasar pengendalian akses.

Pengujian SIM7600G menggunakan AT Command menunjukkan respons modul OK, status kartu SIM READY, nilai kekuatan sinyal 25,99, registrasi jaringan 0,1, dan koneksi layanan data 1. Hasil tersebut menunjukkan bahwa modul telah terhubung ke jaringan dan siap digunakan untuk komunikasi data. Pertukaran pesan MQTT juga berhasil dilakukan pada QoS 0, QoS 1, dan QoS 2. Pada sisi server, Node-RED berhasil menerima UID, melakukan validasi terhadap MySQL, memperbarui status parkir, menambahkan log transaksi, dan mengirimkan hasil validasi kembali ke ESP32.

Mekanisme palang bekerja sesuai rancangan. Ketika UID valid, motor wiper menggerakkan palang menuju posisi terbuka dan berhenti pada sudut sekitar 90° berdasarkan sensor inframerah. Ketika kendaraan masih terdeteksi, palang dipertahankan dalam keadaan terbuka. Pengujian sensor ultrasonik menunjukkan bahwa objek pada jarak 25 cm dan 35 cm membuat palang tetap terbuka, sedangkan kondisi tanpa objek atau jarak lebih dari 100 cm memungkinkan palang menutup. Sensor inframerah pada posisi tertutup menghentikan motor pada sudut sekitar 0°.

Web Admin dan aplikasi Android juga berhasil menampilkan kondisi sistem. Website menampilkan UID, identitas pengguna, waktu transaksi, status akses, serta informasi kondisi parkir secara real-time. Aplikasi Android menampilkan total slot, slot tersedia, kendaraan di dalam, dan status koneksi. Dengan demikian, sistem tidak hanya mengendalikan akses, tetapi juga menyediakan lapisan monitoring bagi administrator.

Integrasi perangkat menunjukkan bahwa ESP32, SIM7600G, driver BTS7960, sensor, dan rangkaian pendukung dapat dioperasikan sebagai satu node. Pada sisi mekanik, dua sensor inframerah berfungsi sebagai pembatas posisi sekitar 90° dan 0°, sedangkan JSN-SR04T digunakan sebagai sensor keselamatan untuk mencegah palang menutup ketika kendaraan masih terdeteksi. Penggunaan sensor ultrasonik sebagai bagian dari mekanisme parkir juga sejalan dengan pemanfaatan sensor jarak pada sistem parkir berbasis IoT (Priadi et al., 2025).



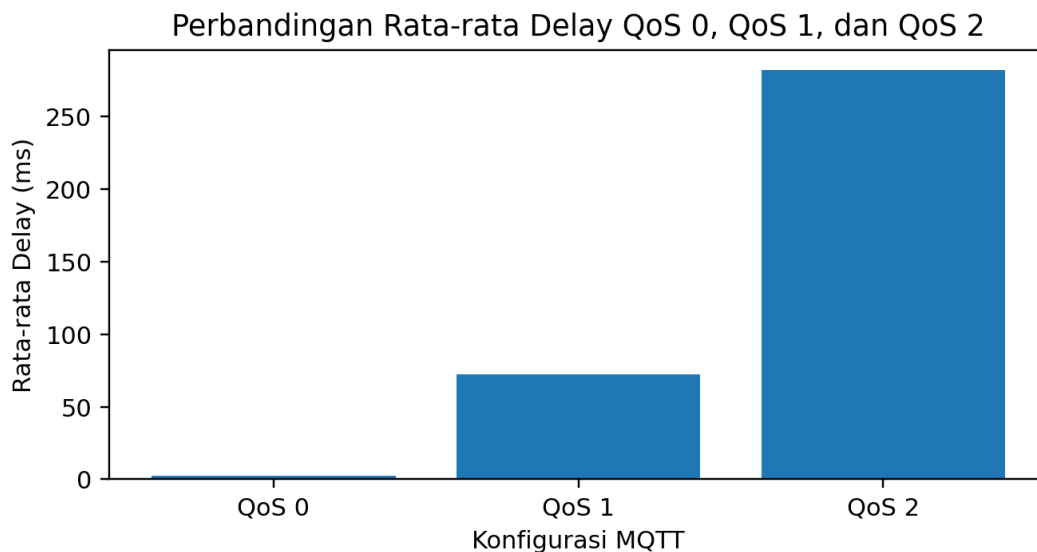
Gambar 1. Flow Node-RED untuk validasi dan pengolahan data

Flow Node-RED menjadi penghubung antara MQTT Broker, basis data, dan layanan monitoring. Data UID yang diterima diproses untuk pencarian pada tabel data_pelanggan. Jika data ditemukan, hasil validasi diteruskan kembali melalui MQTT; aktivitas transaksi dan status kapasitas parkir juga diperbarui. Penggunaan Node-RED sebagai middleware mendukung integrasi aliran data berbasis IoT secara visual dan terstruktur (Dewanto et al., 2024).

Tabel 3. Hasil pengukuran delay komunikasi MQTT

QoS	Minimum (ms)	Maksimum (ms)	Rata-rata (ms)
0	1,776	4,336	2,361
1	64,520	89,038	72,550
2	230,583	746,378	281,732

Pengukuran delay dilakukan 50 kali untuk setiap konfigurasi QoS. Hasil rekapitulasi ditunjukkan pada Tabel 1. QoS 0 menghasilkan delay rata-rata paling rendah, sedangkan QoS 2 menghasilkan delay rata-rata paling tinggi. Perbedaan tersebut berkaitan dengan mekanisme pertukaran paket pada setiap level QoS. QoS 0 mengirimkan pesan tanpa konfirmasi, QoS 1 menambahkan Publish Ack, sedangkan QoS 2 melibatkan rangkaian Publish, Publish Received, Publish Release, dan Publish Complete. Dengan bertambahnya tahapan konfirmasi, waktu yang dibutuhkan untuk menyelesaikan pertukaran pesan juga meningkat.

**Gambar 2. Perbandingan rata-rata delay QoS 0, QoS 1, dan QoS 2**

Jika dibandingkan secara relatif, rata-rata delay QoS 1 sekitar 30,7 kali lebih besar daripada QoS 0, sedangkan QoS 2 sekitar 119,3 kali lebih besar daripada QoS 0. Perbandingan ini memperlihatkan bahwa tambahan mekanisme konfirmasi pada MQTT memberikan konsekuensi yang nyata terhadap waktu pertukaran pesan pada konfigurasi sistem yang diuji. Namun, seluruh konfigurasi tetap menghasilkan packet loss 0%, sehingga kenaikan delay tersebut tidak diikuti oleh kehilangan paket pada pengujian 50 kali pengiriman. Dari sisi kebutuhan aplikasi, QoS 0 lebih sesuai ketika prioritas utama adalah respons cepat, sedangkan QoS 1 atau QoS 2 dapat dipertimbangkan ketika mekanisme konfirmasi pesan lebih diutamakan.

Seluruh konfigurasi menghasilkan packet loss sebesar 0% karena 50 paket yang dikirim pada masing-masing QoS berhasil diterima oleh MQTT Broker. Dengan demikian, pada kondisi pengujian yang dilakukan, packet loss tidak menjadi pembeda performa. Perbedaan utama terlihat pada delay, sehingga pemilihan QoS perlu mempertimbangkan kebutuhan antara kecepatan respons dan mekanisme keandalan pengiriman.

Berdasarkan klasifikasi delay TIPHON yang digunakan dalam laporan, nilai rata-rata QoS 0 dan QoS 1 berada pada kategori sangat baik karena kurang dari 150 ms. QoS 2 memiliki rata-rata 281,732 ms sehingga berada pada kategori baik. Meskipun demikian, nilai maksimum QoS 2 mencapai 746,378 ms, yang menunjukkan adanya variasi waktu tunda pada sebagian pengujian. Hasil ini memperlihatkan bahwa peningkatan level QoS memberikan mekanisme konfirmasi yang lebih lengkap, tetapi pada sistem dan kondisi jaringan yang diuji juga meningkatkan waktu pertukaran data.

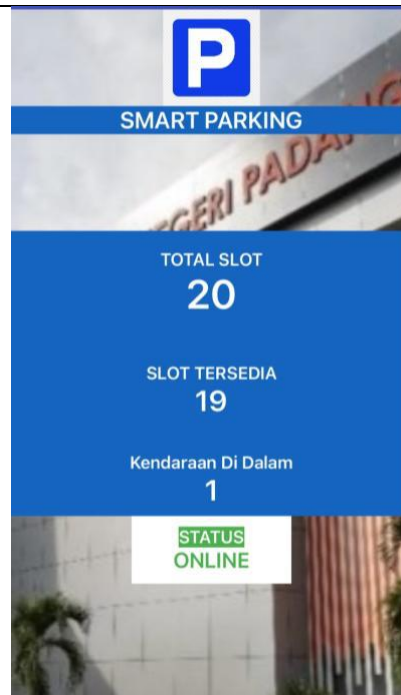
Waktu respons sistem diukur dari tap kartu RFID hingga palang kembali tertutup pada kondisi normal. Lima percobaan menghasilkan waktu 14,33; 13,58; 13,68; 12,51; dan 13,46 detik, dengan rata-rata 13,51 detik. Variasi tersebut tidak hanya dipengaruhi komunikasi seluler dan proses validasi server, tetapi juga waktu pembukaan dan penutupan mekanisme palang. Secara keseluruhan, hasil pengujian menunjukkan bahwa sistem mampu menjalankan rangkaian identifikasi, komunikasi, validasi, pengendalian mekanis, pencatatan, dan monitoring secara terintegrasi.

Tabel 4. Rekapitulasi packet loss

QoS	Paket dikirim	Paket diterima	Packet Loss (%)
0	50	50	0
1	50	50	0
2	50	50	0

Tabel 5. Waktu respons sistem pada kondisi normal

Percobaan	Waktu Respons (detik)
1	14,33
2	13,58
3	13,68
4	12,51
5	13,46
Rata-rata	13,51

**Gambar 3. Antarmuka aplikasi Android monitoring Smart Parking**

Aplikasi Android mengambil data dari server melalui layanan web dan menampilkan total slot, slot tersedia, kendaraan di dalam, serta status koneksi. Hasil pengujian menunjukkan bahwa informasi yang ditampilkan sesuai dengan data sistem dan dapat diperbarui secara real-time. Implementasi antarmuka mobile memperluas akses monitoring tanpa mengubah fungsi utama node lapangan.

1003 6.628	114.10.95.60	103.252.189.185	MQTT	99 Publish Message [smartparking/rfid]
1004 6.629	103.252.189.185	103.252.189.185	MQTT	111 Publish Message [smartparking/rfid]
1009 6.631	103.252.189.185	103.252.189.185	MQTT	100 Publish Message [smartparking/result]
1013 6.631	103.252.189.185	114.10.95.60	MQTT	88 Publish Message [smartparking/result]
1453 10.083	103.252.189.185	103.252.189.185	MQTT	74 Ping Request
1455 10.083	103.252.189.185	103.252.189.185	MQTT	74 Ping Response
2615 18.971	114.10.95.60	103.252.189.185	MQTT	99 Publish Message [smartparking/rfid]
2616 18.971	103.252.189.185	103.252.189.185	MQTT	111 Publish Message [smartparking/rfid]
2621 18.973	103.252.189.185	103.252.189.185	MQTT	100 Publish Message [smartparking/result]
2625 18.974	103.252.189.185	114.10.95.60	MQTT	88 Publish Message [smartparking/result]
3578 25.420	114.10.95.60	103.252.189.185	MQTT	66 Ping Request
3579 25.420	103.252.189.185	114.10.95.60	MQTT	62 Ping Response
3860 28.129	114.10.95.60	103.252.189.185	MQTT	78 Publish Message [smartparking/rfid]
3862 28.129	103.252.189.185	103.252.189.185	MQTT	111 Publish Message [smartparking/rfid]
3866 28.131	103.252.189.185	103.252.189.185	MQTT	100 Publish Message [smartparking/result]
3869 28.131	103.252.189.185	114.10.95.60	MQTT	88 Publish Message [smartparking/result]
5325 37.603	114.10.95.60	103.252.189.185	MQTT	78 Publish Message [smartparking/rfid]
5327 37.603	103.252.189.185	103.252.189.185	MQTT	111 Publish Message [smartparking/rfid]
5331 37.605	103.252.189.185	103.252.189.185	MQTT	100 Publish Message [smartparking/result]
5336 37.606	103.252.189.185	114.10.95.60	MQTT	88 Publish Message [smartparking/result]
8850 64.069	114.10.95.60	103.252.189.185	MQTT	78 Publish Message [smartparking/rfid]
8852 64.069	103.252.189.185	103.252.189.185	MQTT	111 Publish Message [smartparking/rfid]
8856 64.071	103.252.189.185	103.252.189.185	MQTT	100 Publish Message [smartparking/result]
8858 64.071	103.252.189.185	114.10.95.60	MQTT	88 Publish Message [smartparking/result]

Gambar 4. Capture paket MQTT pada konfigurasi QoS 0.

Capture Wireshark memperlihatkan pertukaran paket pada topic smartparking/rfid dan smartparking/result. Pada QoS 0, pesan dikirim tanpa mekanisme konfirmasi, sehingga jumlah tahapan pertukaran paket lebih sedikit dibandingkan QoS 1 dan QoS 2. Pada QoS 1 ditemukan Publish Ack sebagai konfirmasi, sedangkan QoS 2 memperlihatkan rangkaian Publish, Publish Received, Publish Release, dan Publish Complete. Karakteristik tersebut menjelaskan peningkatan delay ketika level QoS dinaikkan (Almheiri & Maamar, 2021; Saha et al., 2024).

SIMPULAN

Sistem Smart Parking berhasil dirancang dan diimplementasikan menggunakan RFID MFRC522, ESP32, SIM7600G, MQTT, Raspberry Pi, Node-RED, dan MySQL. Sistem mampu melakukan identifikasi serta validasi pengguna, mengendalikan palang, mendeteksi kendaraan untuk keselamatan, mencatat transaksi, dan menyediakan monitoring melalui Web Admin serta aplikasi Android. Pengujian komunikasi sebanyak 50 kali pada setiap QoS menghasilkan delay rata-rata 2,361 ms pada QoS 0, 72,550 ms pada QoS 1, dan 281,732 ms pada QoS 2, dengan packet loss 0% pada seluruh konfigurasi. QoS 0 memberikan respons komunikasi tercepat, sedangkan QoS 2 memberikan mekanisme konfirmasi paling lengkap dengan delay tertinggi. Waktu respons sistem pada kondisi normal memiliki rata-rata 13,51 detik. Pengembangan berikutnya disarankan menguji sistem pada variasi kondisi jaringan seluler yang lebih luas dan menggunakan mekanisme aktuator palang yang lebih sesuai untuk operasi jangka panjang.

UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih kepada Politeknik Negeri Padang, dosen pembimbing, serta pihak-pihak yang telah memberikan dukungan dalam proses perancangan, implementasi, pengujian, dan penyelesaian sistem Smart Parking ini.

DAFTAR PUSTAKA

- Aditya, A., Anwarul, S., Tanwar, R., & Koneru, S. K. V. (2023). An IoT Assisted Intelligent Parking System (IPS) For Smart Cities. *Procedia Computer Science*, 218, 1045–1054. <https://doi.org/10.1016/j.procs.2023.01.084>
- Alamin, M. M., Firmansyah, A. R., Bittuqoh, A., Adzimi, C. B., Wahyudi, M. I., & Z. M. AT. (2025). Pengukuran Performa Jaringan Internet Menggunakan Quality Of Service Dengan Wireshark. *Nusantara Computer And Design Review*, 3(1), 9–14. <https://doi.org/10.55732/Ncdr.V3i1.1633>
- Almheiri, A., & Maamar, Z. (2021). IoT Protocols – MQTT Versus Coap. *Proceedings Of The 4th International Conference On Networking, Information Systems & Security*, 1–5. <https://doi.org/10.1145/3454127.3456594>

- Anusha, T., & Pushpalatha, M. (2022). Efficient Communication Model For A Smart Parking System With Multiple Data Consumers. *Smart Cities*, 5(4), 1536–1553. <https://doi.org/10.3390/Smartcities5040078>
- Atiqur, R. (2021). Radio Frequency Identification Based Smart Parking System Using Internet Of Things. *IAES International Journal Of Robotics And Automation*, 10(1), 41–50. <https://doi.org/10.11591/ijra.V10i1.Pp41-50>
- Bhardwaj, P. (2025). An Iot-Driven Smart Parking Framework With RFID-Based Vehicle Authentication. *International Journal Of Science And Technology*, 16(3). <https://doi.org/10.71097/IJSAT.V16.I3.8350>
- Bibi, N., Iqbal, F., Akhtar, S., Anwar, R., & Bibi, S. (2021). A Survey Of Application Layer Protocols Of Internet Of Things. *International Journal Of Computer Science And Network Security*, 21, 301–311. <https://doi.org/10.22937/IJCSNS.2021.21.11.41>
- Biyik, C., Et Al. (2021). Smart Parking Systems: Reviewing The Literature, Architecture And Ways Forward. *Smart Cities*, 4(2), 623–642. <https://doi.org/10.3390/Smartcities4020032>
- Chesuh, L. N., Fernández-Díaz, R. Á., Alija-Pérez, J. M., Benavides-Cuellar, C., & Alaiz-Moreton, H. (2024). Improve Quality Of Service For The Internet Of Things Using Blockchain & Machine Learning Algorithms. *Internet Of Things*, 26, 101123. <https://doi.org/10.1016/J.Iot.2024.101123>
- Dewanto, W. K., Arifianto, A. S., Rakhmad, H., Putranto, H. A., & Firmansyah, M. H. (2024). Node-RED Dan Robotik Pada Sistem Penyiraman Otomatis Berbasis Iot. *Jurnal Teknologi Informasi Dan Multimedia*, 6(3), 354–367.
- Elfaki, A. O., Messoudi, W., Bushnag, S., Abuzneid, S., & Alhmiedat, T. (2023). A Smart Real-Time Parking Control And Monitoring System. *Sensors*, 23(24), 9741. <https://doi.org/10.3390/S23249741>
- Kuchekar, A., Saware, P., Sawant, M., & K. R. P. (2026). Automatic Car Parking System By Using ESP 32. *International Journal Of Science, Strategy, Management And Technology*, 2(4), 1–9. <https://doi.org/10.55041/Ijsmt.V2i4.041>
- Messina, F., Russo, M., Santoro, C., Santoro, F. F., & Tudisco, A. (2025). Evaluation Of A Lightweight Iot Protocol For Intelligent Parking Management In Urban Environments. *Applied Sciences*, 15(17), 9621. <https://doi.org/10.3390/App15179621>
- Morata, J. A. P., Novilla, K. V. S., Lapaña, L. F., Estimada, J. G., C. E. G. MIT, & R. S. P. MIT. (2025). SEAIT Smart Parking System: Enhancing Efficiency And Security In Campus Vehicle Management. *International Journal Of Innovative Science And Research Technology*, 893–903. <https://doi.org/10.38124/Ijisrt/25may641>
- Nafisy, Y., Zain, S. G., & Putra, K. P. (2025). Smart Parking System Based On Internet Of Things Technology For Realtime Parking Slot Monitoring. *Journal Of Embedded Systems, Security And Intelligent Systems*, 6(3), 404–414. <https://doi.org/10.59562/Jessi.V6i3.9716>
- Novianti, N., Chofyfa, A., & Caesaria, S. R. (2026). Edukasi Dasar Arsitektur TCP/IP Dan Protokol Jaringan Pada Siswa Sekolah Menengah Atas. *TIARA–Jurnal Pengabdian Teknologi Informasi Dan Rekayasa Aplikasi*, 1(2), 75–85.
- Priadi, A. R., Safitri, R. A., Pratama, T. B., & Latifa, U. (2025). Comparison Of Accuracy And Precision Of Distance Readings On HC-SR04, JSN-SR04T, And A02YYUW Ultrasonic Sensors. *TESLA: Jurnal Teknik Elektro*, 27(1), 19–29. <https://doi.org/10.24912/Tesla.V27i1.33372>
- Rahmadila, C. D. (2025). Optimasi Kualitas Jaringan 4G LTE Menggunakan Metode Physical Tuning. *Telekontran: Jurnal Ilmiah Telekomunikasi, Kendali Dan Elektronika Terapan*, 12(2). <https://doi.org/10.34010/Telekontran.V12i2.12310>
- Rawat, B., Purnama, S., & Mulyati, M. (2021). Mysql Database Management System (DBMS) On FTP Site LAPAN Bandung. *International Journal Of Cyber And IT Service Management*, 1(2), 173–179. <https://doi.org/10.34306/Ijcitsm.V1i2.47>

- Rostami, M., & Goli-Bidgoli, S. (2024). An Overview Of Qos-Aware Load Balancing Techniques In SDN-Based Iot Networks. *Journal Of Cloud Computing*, 13(1), 89. <https://doi.org/10.1186/S13677-024-00651-7>
- Saha, N., Paul, P., Ji, K., & Harik, R. (2024). Performance Evaluation Framework Of MQTT Client Libraries For Iot Applications In Manufacturing. *Manufacturing Letters*, 41, 1237–1245. <https://doi.org/10.1016/J.Mfglet.2024.09.150>
- Sari, D. N., Priyopradono, B., & Akhir, A. F. P. (2024). Analisis Dan Perancangan Sistem Smart Parking Berbasis RFID Dengan Penggunaan Kartu Tanda Mahasiswa (KTM) Untuk Mengurangi Risiko Pencurian Kendaraan. *Jurnal Teknologi Dan Sistem Informasi Bisnis*, 6(4), 791–798. <https://doi.org/10.47233/Jteksis.V6i4.1561>
- Setyawan, F. A., Sulistiyanti, S. R., Purwiyanti, S., Fitriawan, H., & Adnan, A. R. (2021). Monitoring And Control System With A Client-Server Model Based On Internet Of Things (Iot). *IJUM Engineering Journal*, 22(1), 93–102. <https://doi.org/10.31436/IJUMej.V22i1.1596>
- Shaharudin, N., Hasan, M. Z., & Noor, S. M. (2021). Direct Current (DC) Motor Speed And Direction Controller. *Journal Of Physics: Conference Series*, 2129(1), 012035. <https://doi.org/10.1088/1742-6596/2129/1/012035>
- Simatupang, J. W., Lubis, A. M., & Vincent. (2022). Iot-Based Smart Parking Management System Using ESP32 Microcontroller. 2022 9th International Conference On Electrical Engineering, Computer Science And Informatics, 305–310. <https://doi.org/10.23919/EECSI56542.2022.9946608>
- Tobi, M. H. (2022). Design Of Automatic Parking Access System Based On Internet Of Things (Iot). *Brilliance Research Of Artificial Intelligence*, 2(2), 62–65. <https://doi.org/10.47709/Brilliance.V2i2.1561>
- Ulum, M. B., & Badri, F. (2023). Sistem Monitoring Cuaca Dan Peringatan Banjir Berbasis Iot Dengan Menggunakan Aplikasi MIT App Inventor. *JITET (Jurnal Informatika Dan Teknik Elektro Terapan)*, 11(3), 319–328.
- Yuvier, M., Fernando, N., Marcelleno, N., & Yulianto, A. (2025). Perancangan Dan Pengembangan Sistem Smart Parking Berbasis ESP32 Dan Aplikasi Mobile Android Studio. *Telcomatics*, 10(2). <https://doi.org/10.37253/Telcomatics.V10i2.10867>