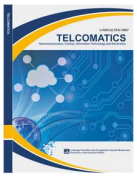
Available online at journal.uib.ac.id/index.php/telcomatics

TELCOMATICS

ISSN (online) 2541-5867



Perancangan dan Pengembangan Sistem Smart Parking Berbasis ESP32 dan Aplikasi Mobile Android Studio

Erick*[1], Maxi Yuvier[1], Nelson Fernando[1], Ng Marcelleno[1], Andik Yulianto[2]

[1]*Prodi Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Internasional Batam, Jalan Gajah Mada, Baloi Sei Ladi, Batam, Indonesia*

[2]*Prodi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Internasional Batam, Jalan Gajah Mada, Baloi Sei Ladi, Batam, Indonesia*

ARTICLE INFO

Keywords:
IoT, Smart Parking, ESP32, Android Studio, Distance Detection

Received: July 3, 2025
Revised: December 3, 2025
Accepted: December 20, 2025

***Corresponding author:**
E-mail: 2332027.erick@uib.edu (Erick)

DOI: 10.37253/telcomatics.v10i2.10867

ABSTRACT

The rapid growth in the number of vehicles has intensified challenges in parking management leading to traffic congestion and wasted time for drivers searching for available spots. Conventional parking systems, which often rely on manual processes, are inefficient and contribute to these problems. This research addresses these issues by designing and developing an Internet of Things (IoT) based smart parking system to provide real-time monitoring of parking availability. The system utilizes an ESP32 microcontroller, chosen for its integrated Wi-Fi and processing capabilities, as the core processing unit. For vehicle and user management, an RFID-RC522 sensor is used for vehicle entry authentication , while an HC-SR04 ultrasonic sensor detects vehicles at the exit. The backend architecture employs a multi-layered approach, using MQTT as the communication protocol with HiveMQ as the broker. System logic and data flows are managed using Node-RED, which communicates with a Flask server to interact with the database. The user interface is a mobile application developed using Android Studio. This application requires user authentication via a login page and provides a real-time dashboard that visually displays the status of parking slots—green for available and red for occupied. The successful implementation of this prototype demonstrates a functional end-to-end solution that effectively integrates hardware and software to create a more efficient and user-friendly parking management system.

PENDAHULUAN

Seiring berjalannya waktu, teknologi terus mengalami perkembangan yang pesat. Perkembangan-perkembangan tersebut telah membawa dampak yang signifikan dalam berbagai aspek kehidupan manusia, tidak terkecuali dalam manajemen efisiensi sistem parkir kendaraan bermotor. Teknologi-teknologi tersebut telah memberikan banyak solusi dalam meningkatkan efisiensi sistem manajemen parkir. Dalam pelaksanaan manajemen sistem parkir, terdapat beberapa tantangan yang harus dihadapi. Salah satu tantangan terbesarnya yaitu terjadinya kemacetan pada antrean masuk dan keluar kendaraan serta *monitoring* ketersediaan lahan parkir kendaraan [1]. Kondisi tersebut menimbulkan dampak yang buruk terhadap para pengguna kendaraan bermotor yang ingin memarkirkan kendaraan mereka dikarenakan kemacetan tersebut menyebabkan terbuangnya waktu secara sia-sia.

Pesatnya pertumbuhan jumlah kendaraan di berbagai kota di dunia juga mengakibatkan permasalahan kemacetan antrean parkir dan keterbatasan lahan parkir menjadi tantangan yang semakin kompleks. Di Indonesia, peningkatan jumlah kendaraan juga menjadi tantangan tersendiri, terutama di kota-kota besar seperti Jakarta, Surabaya, dan Denpasar [2]. Berdasarkan data Badan Pusat Statistik (BPS), jumlah kendaraan terus meningkat setiap tahun. Pada tahun 2021 menuju tahun 2022, jumlah kendaraan bermotor mengalami peningkatan sebesar 4,42% dari 141.992.573 unit ke 148.261.817 unit, sementara ketersediaan lahan parkir masih terbatas. Kendaraan yang berputar-putar mencari tempat parkir juga menjadi salah satu penyebab meningkatnya kemacetan. Selain itu, penggunaan sistem parkir menggunakan metode tekan tombol manual ketika memasuki pintu parkir dan proses pembacaan karcis parkir sewaktu keluar parkir dengan menggunakan metode *scan barcode* dinilai kurang efektif dan dapat meningkatkan kemacetan saat ingin keluar masuk parkiran [3].

Salah satu solusi yang dapat digunakan untuk menjawab tantangan tersebut yaitu penerapan teknologi *Internet of Things* (IoT). Sistem IoT memunculkan peluang dalam mempermudah manusia dalam mengelola kebutuhan hidup secara efisien. Dengan menerapkan teknologi IoT dalam manajemen sistem parkir, kemacetan antrean keluar masuk parkiran menjadi lebih tertib dan disiplin. Selain itu, teknologi IoT tersebut juga dapat menghadirkan sistem pemantauan ketersediaan jumlah lahan parkir yang kosong sehingga pemilik kendaraan dapat menghemat waktu dalam mencari parkiran kosong dalam gedung parkir [4].

Dalam penelitian ini, platform aplikasi *mobile* Android yang dirancang menggunakan *Android Studio* dipilih sebagai sarana untuk melakukan *monitoring* terhadap ketersediaan jumlah lahan parkir pada parkiran, baik parkiran yang sudah terisi maupun parkiran yang kosong.

Sistem ini dirancang dengan memperhatikan beberapa aspek penting, seperti kebutuhan manajemen efisiensi sistem parkir pada skala bisnis. Melalui penelitian ini, diharapkan mampu memberikan solusi terhadap pengelolaan sistem parkir

yang efisien. Keberhasilan implementasi sistem ini diharapkan dapat memberikan dampak positif terhadap efisiensi manajemen sistem parkir. Penelitian ini juga menunjukkan penerapan teknologi IoT dalam meningkatkan efisiensi terhadap berbagai aspek kehidupan manusia dan dapat digunakan sebagai inspirasi dalam mengembangkan teknologi sebagai solusi dalam menjawab berbagai permasalahan dalam kehidupan.

KAJIAN PUSTAKA

A. Penelitian Terdahulu

Penelitian [5] telah mengembangkan sistem *smart parking* berbasis IoT untuk meningkatkan mengatasi masalah sulitnya mencari tempat parkir di daerah yang ramai dengan menyediakan informasi ketersediaan tempat parkir secara *real-time*. Sistem ini menggunakan perangkat berupa mikrokontroler Arduino Mega 2560 sebagai unit pemrosesan data dan sensor Ultrasonic untuk memantau *slot* parkir yang tersedia yang kemudian data *slot* parkir tersebut akan dikirimkan ke tampilan *website* melalui koneksi internet menggunakan perangkat Ethernet Shield.

Pada penelitian [6], telah dikembangkan sistem pemantauan ketersediaan *slot* parkir secara *real-time* berbasis sensor Ultrasonic yang dimana data terkait ketersediaan *slot* tersebut akan dipancarkan pada LED Display yang diletakkan pada posisi yang dapat dijangkau oleh semua orang. Hasil penerapan sistem IoT pada penelitian tersebut menunjukkan tercapainya nilai efisiensi sebesar 94.23% dalam mendeteksi *slot* parkir yang tersedia dan akurasi sebesar 91.76% dalam mengosongkan area parkir yang ditempati.

Penelitian [7] berfokus dalam mengembangkan sistem *smart parking* berbasis IoT yang bertujuan untuk mempermudah dalam pencarian *slot* parkir serta melakukan pembayaran biaya parkir menggunakan aplikasi mobile via Paypal. Penelitian ini menerapkan perangkat berupa mikrokontroler Raspberry Pi 3 dan Python IDE untuk menjalankan program. Sensor Ultrasonic dan Pi Camera yang dilengkapi dengan kecerdasan buatan akan digunakan untuk melakukan monitoring ketersediaan *slot* parkiran dengan mendeteksi keberadaan kendaraan.

Penelitian [8] telah mengembangkan sistem *smart parking* berbasis IoT yang memberikan kemudahan dalam mencari *slot* parkir yang tersedia. Dalam penelitian ini, digunakan mikrokontroler Arduino dan sensor-sensor berupa Sensor IR dan RFID yang terhubung ke aplikasi mobile untuk menampilkan *slot* parkir yang tersedia. Melalui sistem ini, para pemilik kendaraan dapat melakukan *booking slot* parkir secara *online* melalui aplikasi mobile dan ketika sudah mencapai tempat parkiran tersebut, pemilik kendaraan yang melakukan *booking* tersebut dapat mengakses tempat parkir dengan menggunakan RFID tag. Kemudian saat pengendara ingin keluar dari tempat parkiran, jumlah biaya parkir yang harus dibayar akan dihitung menggunakan waktu penggunaan layanan melalui sensor IR dan pembayaran diproses menggunakan *in-app wallet* yang terhubung.

Penelitian [9] berfokus dalam mengembangkan sistem *smart parking* di area parkir jalanan. Penelitian ini memanfaatkan lingkungan Social IoT Lysis untuk membuat entitas virtual dari objek dunia nyata yang terlibat dalam sistem *smart parking* untuk area parkir di jalan. Penggunaan entitas virtual sosial memungkinkan untuk mengatasi masalah interoperabilitas di antara berbagai jenis perangkat IoT yang diterapkan di area yang berdekatan. Sensor magnetometer digunakan untuk mendeteksi keberadaan kendaraan secara otomatis di setiap tempat parkir dan data sensor dikumpulkan melalui konsentrator yang mencakup seluruh area parkir melalui Wi-Fi yang hemat daya.

Dan pada penelitian [10] dikembangkan sistem *smart parking* berbasis IoT yang berfokus untuk menyelesaikan permasalahan waktu yang terbuang dalam mencari parkir, menghindari kemacetan, dan memberikan pengalaman parkir yang nyaman bagi pengguna. Sistem ini menggunakan mikrokontroler Arduino Uno sebagai unit pemrosesan data yang terhubung ke sensor-sensor. Sistem ini juga menggunakan modul ESP8266 untuk melakukan koneksi ke jaringan internet dalam mengirimkan data ketersediaan parkir sehingga status parkir dapat dipantau melalui aplikasi mobile. Sensor IR dipilih untuk melakukan pendeteksian keberadaan kendaraan di setiap *slot* parkir. Melalui aplikasi mobile, pengguna dapat melakukan pemantauan terhadap ketersediaan *slot* parkir dan melakukan *booking* pada *slot* yang tersedia. Setelah melakukan *booking*, pengguna akan mendapat kode QR unik yang akan discan oleh penjaga untuk memverifikasi otentikasi pengguna dan kemudian membuka gerbang secara otomatis. Aplikasi mobile tersebut juga dilengkapi dengan sistem pembayaran menggunakan dompet digital

B. ESP32

ESP32 merupakan salah satu mikrokontroler yang dikembangkan oleh Espressif Systems dan merupakan penerus dari mikrokontroler ESP8266 seperti yang ditunjukkan pada Gambar 1. Mikrokontroler ini sangat cocok untuk diterapkan dalam perancangan sistem IoT. Modul ini mampu menghubungkan perangkat-perangkat IoT ke jaringan internet dikarenakan memiliki modul Wi-Fi dan Bluetooth yang sudah terintegrasi. Selain itu, mikrokontroler ini menggunakan sumber daya listrik yang rendah.

Tabel 1. Perbandingan Spesifikasi ESP32 dan Arduino Uno

Spesifikasi	ESP32	Arduino
CPU	Dual-Core 32-bit @ 160-240MHz	Single-Core 8-bit @ 16MHz
Flash Memory	4MB	32KB
Wi-FI	Terintegrasi	Tidak Terintegrasi
Bluetooth	Terintegrasi	Tidak Terintegrasi
Pin I/O	36	14
Tegangan	3.3V	5V

Gambar 1. Mikrokontroler ESP32

Dalam penelitian ini, mikrokontroler ESP32 dipilih dikarenakan mikrokontroler ini memiliki keunggulan dibandingkan dengan mikrokontroler Arduino Uno.

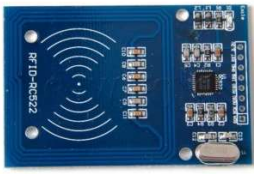
Mikrokontroler ini dilengkapi dengan fitur Wi-Fi 802.11 b/g/n dan Bluetooth v4.2 yang terintegrasi, prosesor Xtensa Dual-Core 32-bit LX6 yang memiliki kecepatan hingga 240MHz, serta memori Flash sebesar 4MB untuk menjalankan program dan data. Selain itu, ESP32 juga memiliki 48 Pin Input/Output (I/O) termasuk pin ADC, DAC, SPI, I2C dan UART. Perbandingan antara ESP32 dan Arduino Uno dapat dilihat pada Tabel 1.

C. RFID-RC522

Sensor RFID-RC522 merupakan perangkat sensor yang menggunakan gelombang radio untuk mengidentifikasi dan melacak objek secara otomatis. Sensor ini terdiri atas dua komponen utama (lihat Gambar 2), yaitu tag RFID yang memiliki identitas unik dan *reader* RFID yang membaca informasi dari tag. Spesifikasi umum dari perangkat ini dapat dilihat pada Tabel 2.

Tabel 2. Tabel 2. Spesifikasi RFID-RC522

Spesifikasi	ESP32
Chip	NXP MFRC522
Frekuensi Operasi	13.56 MHz
Tegangan	3.3V
Interface	SPI (*Serial Peripheral Interface*)
Jarak Baca	0 - 5cm
Kecepatan Transfer Data	± 10Mbit/s



Gambar 2. RFID-RC522

D. HC-SR04

Sensor HC-SR04 merupakan sensor yang menggunakan gelombang ultrasonik untuk mendeteksi keberadaan suatu objek dengan memperkirakan jarak dari sensor terhadap objek tersebut seperti ditunjukkan pada Gambar 3. Cara kerja dari sensor ini yaitu gelombang ultrasonik akan dipancarkan oleh perangkat piezoelektrik. Ketika gelombang ultrasonik menyentuh objek, maka objek akan memantulkan kembali gelombang tersebut. Gelombang pantulan akan ditangkap oleh sensor dan akan dihitung selisih antara waktu pengiriman gelombang dan waktu gelombang yang diterima. Berikut spesifikasi umum sensor HC-SR04 dapat dilihat pada Tabel 3.

Tabel 3. Spesifikasi Umum Sensor HC-SR04

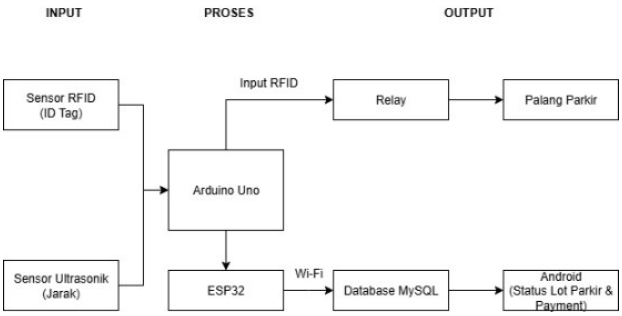
Spesifikasi	ESP32
Tegangan	5V (DC)
Jangkauan Pengukuran	2 – 400cm
Frekuensi	40 kHz
Interface	SPI (Serial Peripheral Interface)
Sudut Ukur	< 15 derajat
Akurasi	± 3mm

Gambar 3. Sensor HC-SR04

METODE PENELITIAN

A. Diagram Blok Sistem

Diagram blok sistem yang dirancang untuk sistem *smart parking* berbasis IoT ditunjukkan pada Gambar 4. Sistem IoT yang telah dirancang terdiri 4 lapisan yang memiliki fungsi dan peranan masing-masing. Lapisan-lapisan tersebut yaitu *Perception Layer*, *Communication Layer*, *Data Processing Layer*, dan *Application Layer*.



Gambar 4. Diagram Blok Sistem *Smart Parking* Berbasis IoT

1. Perception Layer

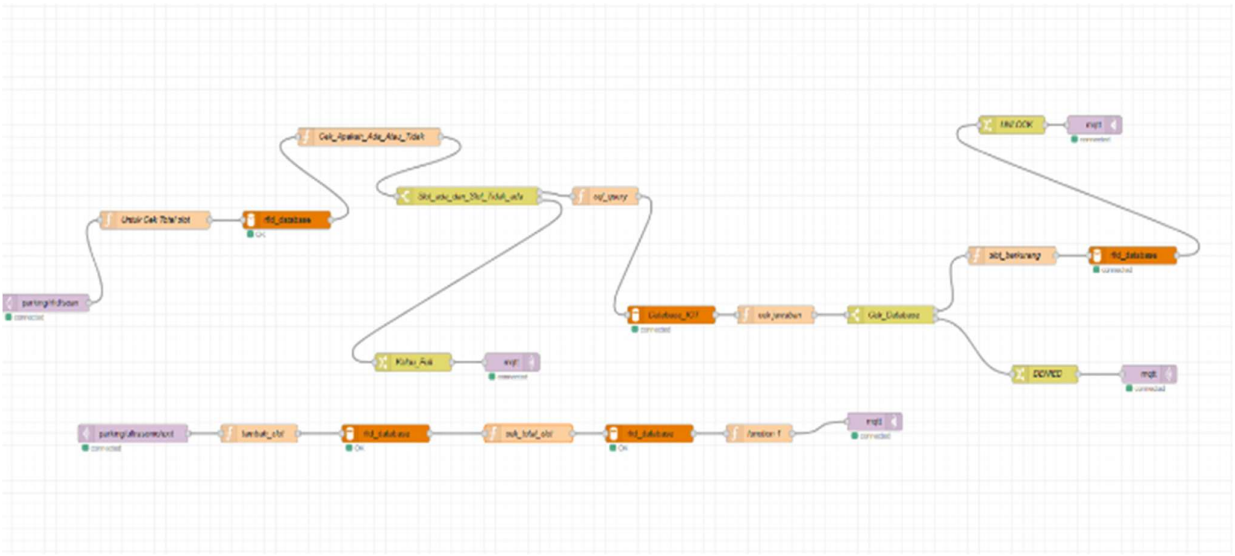
Merupakan lapisan yang memiliki peran dalam mengambil data dari lingkungan sekitar melalui sensor dan mengirimkannya kepada *layer* yang lebih tinggi untuk diolah. Pada penelitian, sensor yang digunakan yaitu sensor RFID-RC522 untuk mengambil input dari Tag RFID dan sensor HC-SR04 yang menggunakan gelombang ultrasonik untuk mendeteksi keberadaan objek berupa kendaraan yang ingin keluar. Selain itu, juga digunakan aktuator berupa motor servo yang akan terbuka sebesar 90 derajat.

2. Communication Layer

Penelitian ini menggunakan Modul Wi-FI yang terintegrasi pada mikrokontroler ESP32 untuk melakukan komunikasi antar *layer*. Protokol komunikasi yang digunakan berupa protokol MQTT berbasis HiveMQ untuk menerima data dan menampilkannya pada mobile app.

3. Data Processing Layer

Lapisan ini merupakan bagian *backend*, tempat data dari perangkat diolah, dianalisis, dan menjadi dasar pengambilan keputusan. Dalam arsitektur perangkat tersebut, HiveMQ Cloud berperan sebagai MQTT Broker terkelola yang menjadi perantara pesan andal antara perangkat ESP32 dan platform pemrosesan, secara efektif memisahkan (*decoupling*) logika



Gambar 5. Process Flow Node-RED

perangkat keras dari logika bisnis. Data yang masuk melalui broker kemudian diterima oleh Node-RED, sebuah platform *middleware* visual yang menjadi inti dari logika sistem. Alur kerja (*flow*) di Node-RED dirancang untuk menerima data UID, memprosesnya dengan menerapkan logika bisnis seperti validasi pengguna dan pengecekan ketersediaan *slot*, lalu mempublikasikan perintah kembali ke ESP32. Untuk tugas-tugas yang memerlukan interaksi dengan database, seperti validasi UID atau pembaruan data slot parkir, Node-RED dan ESP32 berkomunikasi dengan Server Flask (Python). Server *backend* ini menyediakan *endpoint* API yang diperlukan untuk mengakses dan mengelola data pengguna serta status parkir di dalam database sistem. Diagram proses Node-RED ditunjukkan pada Gambar 5.

4. Application Layer

Data-data yang telah diproses dalam *data processing layer* akan diterima dan disimpan pada database. HiveMQ kemudian akan membaca isi pada database dan menampilkannya secara *real-time* pada aplikasi mobile yang dirancang menggunakan Android Studio. Dalam aplikasi tersebut, pengguna dapat memantau jumlah *slot* parkir yang tersedia.

B. Cara Kerja Sistem

Alur proses kerja dari sistem *smart parking* berbasis IoT ini dimulai dengan proses inisialisasi ketika perangkat pertama dinyalakan. Setelah proses inisialisasi selesai, maka sistem akan masuk ke tahapan *loop* untuk menjalankan fungsi-fungsi utamanya secara terus-menerus.

Ketika sensor RFID menerima input dari Tag RFID, ESP32 akan membaca ID unik dari Tag tersebut. Untuk menghindari pengiriman berulang, sistem akan mengabaikan kartu yang sama jika dipindai lagi dalam waktu singkat. ID yang valid kemudian dikirim ke server melalui jaringan WiFi menggunakan protokol MQTT. Server Node-RED akan memeriksa apakah ID tersebut terdaftar di database serta memeriksa apakah slot parkir masih tersedia. Jika terdaftar maka motor servo akan terbuka sebesar 90 derajat selama 3 detik sebelum servo tertutup kembali dan *slot* parkiran yang terdapat pada aplikasi mobile akan berkurang 1. Ketika kendaraan ingin keluar dari tempat parkir, sensor ultrasonik akan mendeteksi keberadaan objek kendaraan dan servo motor akan terbuka sebesar 90 derajat selama 3 detik. Setelah itu, servo akan tertutup kembali dan *slot* parkir pada aplikasi mobile akan bertambah 1.

C. Keamanan Sistem

Dalam sistem IoT, keamanan merupakan salah satu aspek yang penting. Dalam implementasi sistem *smart parking* berbasis IoT akan dibutuhkan untuk melindungi data dan perangkat-perangkat dari ancaman. Aplikasi mobile berbasis Android Studio yang dirancang akan menggunakan fitur *login* untuk mengidentifikasi pengguna yang ingin memantau jumlah *slot* parkir pada parkiran. Data-data identifikasi tersebut akan disimpan pada database

D. Implementasi

Implementasi sistem IoT pada proyek *smart parking* ini menggunakan protokol MQTT (*Message Queuing Telemetry Transport*) sebagai protokol komunikasi utama dalam proses pengiriman dan penerimaan data antar perangkat dan sistem. Dalam sistem ini, HiveMQ digunakan sebagai broker MQTT yang mengatur lalu lintas data dari dan ke perangkat, sedangkan Node-RED digunakan sebagai platform visual untuk memproses, mengelola, dan memvisualisasikan alur data. Dalam konteks komunikasi MQTT, sistem ini menggunakan konsep topik sebagai saluran pertukaran pesan antar entitas. Setiap topik didefinisikan berdasarkan fungsi spesifik yang dibutuhkan oleh perangkat dan antarmuka pengguna.

Pada penelitian ini, beberapa topik MQTT utama didefinisikan untuk menyimpan dan meneruskan data yang akan diproses. Data yang telah diproses ini kemudian digunakan untuk mendukung fungsionalitas berbagai fitur

pada *Application Layer*. Setiap topik memiliki fungsi tertentu seperti yang ditunjukkan pada Tabel 4.

Tabel 4. Topik-Topik MQTT

Topik MQTT	Tipe Data	Deskripsi
Parking/rfid/scan	Integer	Berisi data UID kartu yang dipindai di gerbang masuk. Dipublikasikan oleh ESP32 ke Node-RED.
Parking/ultrasonic/exit	String	Berisi sinyal dari sensor ultrasonik saat kendaraan terdeteksi di gerbang keluar.
Parking/servo/control	String	Berisi perintah kontrol motor servo (misal: "UNLOCK","DENIED"). Dikirim oleh Node-RED ke ESP32.
parking/device/status	String	Berisi pesan status sebagai umpan balik ke pengguna (misal: "PENUH", "DITOLAK").

Kumpulan topik pada Tabel 4 diatur pada broker HiveMQ dan digunakan untuk menyimpan dan mengirim data antar-lapisan. Topik yang berisi data dari sensor (parking/rfid/scan, parking/ultrasonic/exit) akan dipublikasikan oleh ESP32 dan di-subscribe oleh Node-RED. Sebaliknya, topik yang berisi perintah (parking/servo/control, parking/device/status) akan dipublikasikan oleh Node-RED dan di-*subscribe* oleh ESP32.

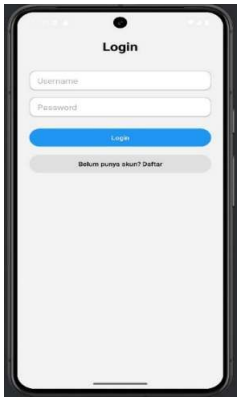
Setiap kali terjadi pembaruan data pada virtual yang di-*subscribe*, ESP32 akan menerima data baru secara *real-time*. Data yang diterima kemudian diproses dan digunakan dalam berbagai logika yang mendukung fitur-fitur yang ada pada penelitian ini. Semua proses yang dapat dilakukan oleh *user* tersebut mengakibatkan terjadinya hal berikut:

1. Membuka Palang Servo
   Berdasarkan data UID yang diterima dari topik *parking/rfid/scan*, Node-RED akan melakukan validasi. Jika valid dan *slot* tersedia, Node-RED mengirimkan perintah "UNLOCK" ke topik *parking/servo/control*. ESP32 yang menerima perintah ini akan mengendalikan motor servo untuk membuka palang parkir.
2. Deteksi Kendaraan dan Kontrol Gerbang Keluar
   Ketika sensor ultrasonik mendeteksi kendaraan, ESP32 langsung membuka palang gerbang dan secara bersamaan mempublikasikan pesan ke topik *parking/ultrasonic/exit*. Node-RED yang menerima pesan ini akan melakukan *update* slot parkir ke dalam database.
3. Pembaruan Status Kapasitas Parkir
   Setiap kali terjadi proses masuk atau keluar, Node-RED akan menghitung ulang jumlah slot yang tersedia dan dapat mempublikasikan nilai terbaru ke topik terpisah (tidak tercakup dalam kode ESP32) untuk ditampilkan pada *dashboard* di *Application Layer*.
4. Notifikasi
   Pembaruan data pada topik *parking/device/status* memungkinkan sistem memberikan umpan balik langsung. Jika parkir penuh, Node-RED akan membuat log bahwa parkir tersedia adalah [0] Jika kartu tidak terdaftar, pesan "DITOLAK" dikirim, dan saat *slot* parkir penuh lampu LED akan berkelip dengan cepat. ESP32 menggunakan data ini untuk mengontrol LED, memberikan notifikasi visual yang jelas kepada pengguna di lokasi. Serta *user* dapat memantau dari aplikasi mobile untuk mengetahui jumlah dari *slot* parkir tersisa.

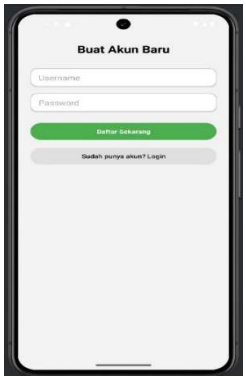
HASIL DAN PEMBAHASAN

Hasil dari penelitian ini merupakan sebuah aplikasi mobile yang berfungsi sebagai sistem *monitoring* parkir. Aplikasi ini dirancang dan dikembangkan secara spesifik untuk platform Android menggunakan Android Studio, yang merupakan IDE resmi dari Google. Fungsi utama dari aplikasi ini adalah memberikan kemudahan bagi pengguna untuk memantau ketersediaan *slot* parkir secara *real-time*. Berbeda dengan sistem konvensional yang mengharuskan pengemudi untuk berputar-putar mencari tempat kosong, aplikasi ini menyajikan informasi akurat langsung ke pengguna secara *online*.

Ketika pengguna membuka aplikasi tersebut, pengguna akan diarahkan ke halaman *login* yang ditunjukkan pada Gambar 6. untuk melakukan otentikasi terlebih dahulu sebelum masuk ke tampilan *dashboard monitoring*. Pengguna yang sudah terdaftar hanya perlu memasukkan *username* dan *password* mereka untuk diverifikasi oleh sistem dan mendapatkan akses ke *dashboard* utama. Jika pengguna belum memiliki akun yang terdaftar, maka pengguna dapat melakukan pendaftaran akun baru pada halaman *register* seperti pada Gambar 7. dengan memasukkan *username* dan *password* sesuai pilihan pengguna. Setelah pendaftaran berhasil, kredensial ini akan disimpan secara aman di *database* sistem dan dapat langsung digunakan untuk melakukan proses *login*.



Gambar 6. Tampilan *User Inteface* (UI) Halaman *Login*



Gambar 7. Tampilan User Interface (UI) Halaman Register

Setelah melakukan pengisian kredensial pada halaman *login*, pengguna kemudian akan diarahkan ke halaman utama dari aplikasi mobile tersebut yaitu halaman *dashboard monitoring* seperti pada Gambar 8. Pada halaman tersebut, pengguna dapat disajikan tampilan berupa logo dari aplikasi mobile, teks yang menampilkan jumlah ketersediaan *slot* parkir, serta *image* mobil yang merepresentasikan jumlah *slot* parkir maksimal yang terdapat pada parkiran tersebut.

Image mobil tersebut memiliki 2 status, yaitu status *image* yang berwarna hijau dan *image* yang berwarna merah. Ketika status mobil tersebut berwarna hijau, maka menandakan bahwa parkiran tersebut masih memiliki *slot* parkir yang kosong. Bertolak belakang dengan status yang berwarna hijau, jika status mobil berwarna merah maka menandakan bahwa *slot* parkir tersebut sudah terisi oleh kendaraan lain. Status warna *image* tersebut akan berubah mengikuti data yang diambil oleh API terhadap *database slot* parkir tersebut.

Aplikasi mobile ini bekerja dengan cara melakukan panggilan API secara periodik ke *endpoint* 192.168.1.7:5000/status_parkir yang disediakan oleh server Flask. Setiap kali ada kendaraan yang masuk atau keluar, server Flask memperbarui status di database, sehingga data yang disajikan oleh API selalu yang terbaru. Dengan demikian, pengguna dapat mengetahui kondisi *slot* parkir sebelum tiba di lokasi.

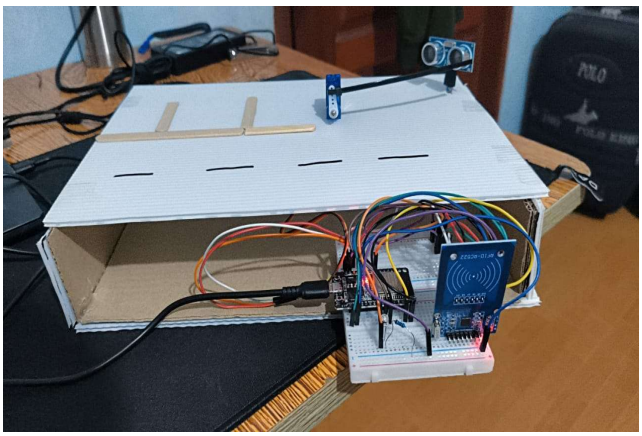
Untuk kebutuhan *monitoring* teknis dan *debugging*, administrator dapat memanfaatkan debug *window* pada Node-RED yang menampilkan log aktivitas sistem secara *real-time* pada Gambar 9. Melalui log ini, setiap kejadian dapat dipantau, mulai dari kartu RFID yang dipindai, hasil validasi dari server, hingga perintah yang dikirimkan kembali ke perangkat keras.

Gambar 8. Tampilan User Interface (UI) Dashboard Monitoring

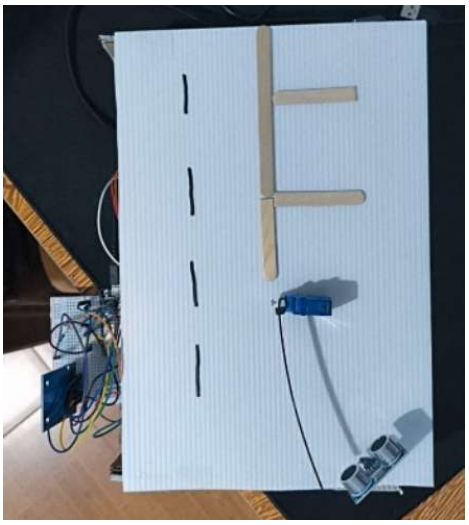
Gambar 9. Tampilan Log *Backend* (Flask API)

Logika pemrosesan di Node-RED menerima data UID dari topik parking/rfid/scan, kemudian mengirimkan permintaan HTTP ke server Flask untuk validasi. Berdasarkan respons server, Node-RED akan menerbitkan pesan "BUKA" ke topik parking/servo/control jika kartu valid dan parkir tersedia, atau pesan "UNLOCK/DENIED" ke topik parking/device/status jika sebaliknya.

Adapun hasil perangkat fisik sistem *smart parking* IoT yang telah dirancang dapat dilihat pada Gambar 9. dan Gambar 10.



Gambar 10. Tampilan Depan Perangkat *Smart Parking*



Gambar 11. Tampilan Atas Perangkat *Smart Parking*

Perangkat dirancang menyerupai miniatur simulasi parkiran dengan menggunakan *infraboard* sebagai fondasinya sepertin pada Gamabar 11. Perangkat-perangkat yang berhubungan dengan pemrosesan data seperti mikrokontroler dan kabel-kabel akan diletakkan pada bagian bawah miniatur. Pada landasan atas telah disusun seperti parkiran pada umumnya yang dimana terdapat palang parkir menggunakan motor servo yang telah diikat dengan penyegel kantong plastik agar terlihat seperti palang, slot parkiran menggunakan stik kayu, dan sensor HC-SR04 untuk mendeteksi kendaraan yang

ingin keluar. Hasil pengujian dari sistem IoT ini dapat dilihat pada Tabel 5.

Tabel 5. Hasil Pengujian Fungsional Log Sistem Palang Parkir

Waktu (Jadwal)	Aksi	Toleransi Waktu	Note
2 Juli 21:49:29	Cek Total *Slot*	2 Detik	Cek total *slot* yang tersedia di dalam *database*.
2 Juli 21:49:29	Cek UID *RFID Card*	3 Detik	Cek *unique* id yang ada pada RFID Card apakah terdaftar atau tidak.
2 Juli 21:49:29	Hasil Cek *Database*	1 Detik	Cek status unique id didalam database apakah diizinkan atau tidak.
2 Juli 21:49:29	Ultrasonic Sensor	3 Detik	Terdeteksi mobil keluar dan memunculkan sisa *slot*.

KESIMPULAN

Tabel hasil pengujian menunjukkan data terkait waktu kejadian, aksi yang dilakukan, waktu respons sistem, toleransi waktu, dan catatan terkait pengujian. Secara keseluruhan, sistem menunjukkan hasil yang sangat baik dengan toleransi waktu respons yang berkisar antara 1 hingga 2 detik untuk proses yang melibatkan validasi server, yang dianggap sangat memadai untuk sistem palang parkir.

Dalam analisis hasil, akurasi logika sistem terbukti berjalan sesuai dengan skenario yang dirancang. Motor servo selalu mengeksekusi perintah dengan benar, dan umpan balik visual melalui LED juga berjalan sesuai kondisi. Efektivitas sistem terbukti dengan kemampuannya untuk menjalankan tugas otentikasi dan kontrol secara otomatis dengan latensi yang rendah dan kinerja mekanisme yang stabil.

REFERENCES

[1] A. Khanna and R. Anand, "IoT based smart parking system," in *2016 International Conference on Internet of Things and Applications (IOTA)*, 2016, pp. 266–270. doi: 10.1109/IOTA.2016.7562735.

[2] A. Agung Permana Putra, I. Putu Satwika, and M. Kom, "SMART PARKING DALAM MENUNJANG IMPLEMENTASI SMART CITY DI KOTA DENPASAR," *SMART TECHNO (Smart Technology, Informatic, and Technopreneurship)*, vol. 04, no. 02, pp. 2541–0679, 2022.

[3] B. Kurniawan, E. B. Setiawan, and R. Hartono, "PERBAIKAN SISTEM PARKIR KENDARAAN BERMOTOR DI LINGKUNGAN UNIVERSITAS KOMPUTER INDONESIA DENGAN MENGGUNAKAN RFID DAN DATABASE," *Majalah Ilmiah UNIKOM*, vol. 12, no. 2, pp. 125–134.

[4] T. N. Pham, M. F. Tsai, D. B. Nguyen, C. R. Dow, and D. J. Deng, "A Cloud-Based Smart-Parking System Based on Internet-of-Things Technologies," *IEEE Access*, vol. 3, pp. 1581–1591, 2015, doi: 10.1109/ACCESS.2015.2477299.

[5] S. Rahman and P. Bhoumik, "IoT Based Smart Parking System," *International Journal of Advances in Computer and Electronics Engineering*, vol. 4, no. 1, pp. 11–16, Jan. 2019, Accessed: Jun. 30, 2025. [Online]. Available: https://www.researchgate.net/profile/Saidur-Rahman-7/publication/329686583_IoT_Based_Smart_Parking_System/links/5c153f0aa6fdcc494ff7b9be/IoT-Based-Smart-Parking-System.pdf

[6] H. Mohapatra and A. K. Rath, "An IoT based efficient multi-objective real-time smart parking system," *International Journal of Sensor Networks*, vol. 37, no. 4, pp. 219–232, 2021, doi: 10.1504/IJSNET.2021.119483.

[7] M. M. S. Ismail *et al.*, "IoT Based Smart Parking System," *J Phys Conf Ser*, vol. 1424, no. 1, pp. 012–021, Dec. 2019, doi: 10.1088/1742-6596/1424/1/012021.

[8] Y. Agarwal, P. Ratnani, U. Shah, and P. Jain, "IoT based smart parking system," *Proceedings - 5th International Conference on Intelligent Computing and Control Systems, ICICCS 2021*, pp. 464–470, May 2021, doi: 10.1109/ICICCS51141.2021.9432196.

[9] A. Floris, S. Porcu, L. Atzori, and R. Girau, "A Social IoT-based platform for the deployment of a smart parking solution," *Computer Networks*, vol. 205, p. 108756, Mar. 2022, doi: 10.1016/J.COMNET.2021.108756.

[10] H. Tanti, P. Kasodariya, S. Patel, and D. H. Rangrej, "Smart Parking System based on IOT; Smart Parking System based on IOT," *International Journal of Engineering Research & Technology*, vol. 9, no. 5, pp. 73–77, May 2020, Accessed: Jul. 02, 2025. [Online]. Available: www.ijert.org