

Pengembangan Prototipe Sistem *Smart Parking* dengan Integrasi *Parking Lock* berbasis *Internet of Things*

TRIA ANANDA SURYANI, LINDAWATI*, MOHAMMAD FADHLI

Teknik Telekomunikasi Politeknik Negeri Sriwijaya, Indonesia

Email: trianandasy@gmail.com;

*Penulis Korespondensi

Received 21 Mei 2024 | *Revised* 6 Juli 2024 | *Accepted* 9 Juli 2024

ABSTRAK

Meningkatnya jumlah kendaraan roda empat di Indonesia, terutama di kota-kota besar, menimbulkan permasalahan dalam pengelolaan parkir seperti risiko kehilangan tiket kertas, kurangnya informasi real-time, dan pengelolaan area parkir yang tidak optimal. Penelitian ini mengembangkan sistem smart parking berbasis Internet of Things dengan mengintegrasikan parking lock menggunakan QR Code dan penggunaan sensor IR obstacle untuk mendeteksi ketersediaan slot parkir. Berdasarkan Pengujian yang telah dilakukan didapatkan hasil penggunaan sensor IR obstacle untuk deteksi slot parkir dengan akurasi 100%. Hasil pengujian kecepatan scan QR Code yaitu 1,44 detik hingga 5,03 detik, dan total rata-rata sebesar 2,88 detik. Total akurasi pada pengujian keakuratan data yang dipindai QR Code adalah 100%. Integrasi parking lock memastikan pengguna parkir sesuai slot yang dipesan dengan akurasi 100%.

Kata kunci: *Smart Parking, Parking Lock, Internet of Things, QR Code*

ABSTRACT

The increasing number of cars in Indonesia, particularly in big cities, has led to issues in parking management such as the risk of losing paper tickets, lack of real-time information, and inefficient management of parking areas. This research aims to develop an IoT-driven smart parking system by incorporating a QR code-enabled parking lock and utilizing IR obstacle sensors to accurately detect parking slot availability. The experimental results showed 100% accuracy in using IR obstacle sensors for parking space detection. The QR code scanning speed test results ranged from .44 seconds to 5.03 seconds, with an average of 2.88 seconds. The accuracy of QR code reading on scanned data was 100%. Parking lock integration ensures that users park in the designated space with 100% accuracy.

Keywords: *Smart Parking, Parking Lock, Internet of Things, QR Code*

1. PENDAHULUAN

Sebagai salah satu negara berkembang, meningkatnya jumlah penduduk di Indonesia menyebabkan peningkatan jumlah masyarakat yang menggunakan kendaraan roda empat. Kondisi ini dapat menimbulkan permasalahan terkait tempat parkir, terutama di pusat perbelanjaan dan lokasi lainnya **(Juwita, dkk, 2020) (Puspitasari, dkk, 2021)**. Tantangan terkait kebutuhan sistem parkir menjadi fenomena umum, khususnya di kota-kota besar di Indonesia. Dampaknya dapat dirasakan melalui kemacetan yang sering terjadi di area parkir **(Alsaedi & Jalal, 2023) (Arfianto, 2022)**.

Permasalahan umum dalam sistem parkir adalah penggunaan tiket kertas sebagai tanda terima parkir. Penggunaan tiket kertas dapat mengakibatkan pemborosan anggaran dan menciptakan dampak negatif terhadap lingkungan melalui limbah kertas yang dihasilkan. Dilihat dari aspek keamanan penggunaan tiket kertas kurang efisien, karena pengguna harus menyimpan tiket hingga meninggalkan area parkir **(Puspitasari, dkk, 2021) (Setiadi, dkk, 2017)**. Kendala lain dalam sistem parkir termasuk kesulitan mencari slot kosong karena kurangnya informasi *real-time* kepada pengguna **(Al Rubiye, dkk, 2020) (Javale, dkk, 2020)**. Selain itu, proses pembayaran tunai dengan sistem manual juga meningkatkan risiko kecurangan dan mengurangi efisiensi waktu serta kepercayaan pengguna **(Suyendra, dkk, 2020)**.

Berbagai penelitian sebelumnya telah dilakukan dengan bermacam teknik implementasi seperti pada penelitian **(Anitha, dkk, 2021)** menerapkan sistem IoT yang memanfaatkan mikrokontroler ATmega32 yang dikontrol oleh aplikasi Android dan penggunaan sensor IR untuk mendeteksi kendaraan. Namun, penelitian ini belum terfokus pada pengoptimalan akurasi dan responsivitas sensor IR dalam mendeteksi kendaraan secara tepat dan akurat. Selanjutnya penelitian **(Al Rubiye, dkk, 2020)** menggunakan Sensor PIRs untuk mendeteksi gerakan objek dan sensor LDR digunakan untuk mendeteksi cahaya dan memberikan informasi tentang slot parkir yang terisi dan kosong yang tersimpan pada repositori IoT *cloud*. Selain itu pada penelitian **(Patil, dkk, 2018)** mengaplikasikan Arduino Uno dan sensor ultrasonik HC-SR04 untuk mendeteksi keberadaan kendaraan di tempat parkir kemudian data dikirim ke Aplikasi menggunakan modul Bluetooth HC-05.

Beberapa penelitian terbaru dalam pengelolaan parkir menunjukkan berbagai implementasi teknologi yang inovatif. Penelitian **(Florence, dkk, 2020)** menggunakan aplikasi Android untuk booking slot parkir dari jarak jauh dengan sensor RFID dan NodeMCU untuk kontrol akses. Penelitian lainnya **(Syahnas, dkk, 2023)** mengusulkan sistem parkir pintar dengan NodeMCU ESP8266 untuk memonitor status slot parkir melalui aplikasi Android, menggunakan *QR Code* dan *e-parking* untuk akses pintu otomatis. Penelitian **(Setiawan, dkk, 2020)** mengembangkan sistem *booking* dan *non-booking* dengan pembayaran menggunakan *QR Code*, namun integrasi *QR Code* saat ini hanya terbatas pada transaksi parkir. Namun, beberapa penelitian tersebut belum ada penerapan *QR Code* sebagai autentikasi *parking lock*. Dari beberapa permasalahan diatas, diperlukan suatu sistem manajemen parkir yang efektif untuk pengembangan sensor yang lebih akurat dalam mendeteksi kendaraan dan implementasi teknologi yang memudahkan pengguna dalam mencari slot parkir yang tersedia secara *real-time*. Strategi yang bisa digunakan untuk mencapai sasaran tersebut adalah memanfaatkan teknologi komunikasi *Internet of Things*.

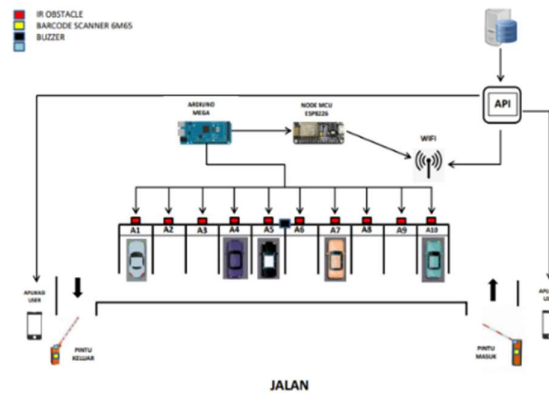
Berdasarkan uraian penjelasan dan penelitian terdahulu yang mengembangkan sistem *smart parking*, maka penelitian ini bertujuan untuk mengoptimalkan sistem *smart parking* dengan integrasi *parking lock* berbasis *Internet of Things*. Pengembangan sistem memastikan pengguna dapat parkir sesuai slot yang telah dipesan, dengan penerapan *QR Code* sebagai

otentikasi untuk *parking lock*. Implementasi sensor IR *obstacle* bertujuan meningkatkan akurasi dalam mendeteksi kendaraan di area parkir terfokus pada pengenalan objek.

2. METODE

2.1 Deskripsi Sistem

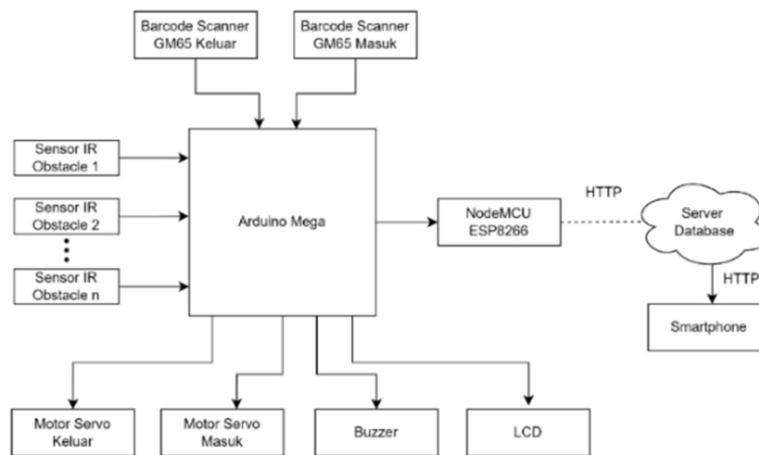
Penelitian ini ditujukan untuk mengembangkan prototipe sistem *smart parking* yang berfungsi untuk pengelolaan area parkir. Sistem ini secara *real-time* mampu mengirimkan data sensor IR *obstacle* dari setiap slot untuk mengetahui slot parkir terisi atau tidak. Pengembangan pada sistem ini yaitu integrasi *parking lock* yang menerapkan *QR Code* untuk verifikasi dan akses slot parkir. *QR Code* adalah modul yang membentuk pola khusus dan dapat menyimpan berbagai jenis informasi, seperti teks, URL, dan berbagai data lainnya (Suradi, dkk, 2023). *Parking lock* bertujuan untuk memastikan kesesuaian pengguna untuk parkir sesuai slot yang sudah di pesan. Jika kendaraan parkir pada slot yang salah, sistem akan mengaktifkan *buzzer* untuk menandakan kesalahan parkir. Ilustrasi sistem *smart parking* terdapat di Gambar 1.



Gambar 1. Sistem *Smart Parking*

2.2 Perancangan Perangkat Keras

Proses perancangan perangkat keras diawali dengan menyusun diagram skema rangkaian. Dalam penelitian ini, disusun skema desain untuk rangkaian komponen-komponen yang akan diaplikasikan pada perangkat yang sedang dikembangkan.

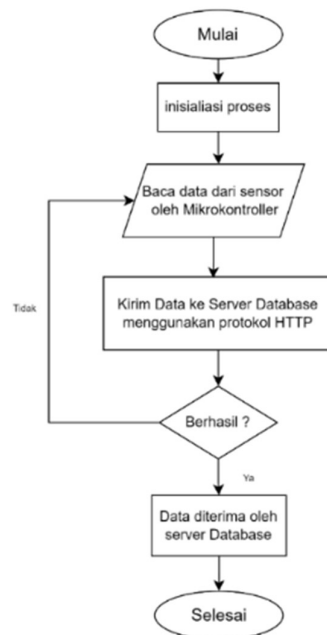


Gambar 2. Diagram Blok Sistem Perangkat Keras

Gambar 2 menampilkan sistem diagram blok perangkat keras yang dirancang untuk sistem *smart parking* terdiri dari beberapa mikrokontroler seperti Sensor IR *Obstacle* untuk mendeteksi slot parkir terisi atau tidak dan mengirim data ke Arduino Mega 2560, yang kemudian mengirimkannya ke NodeMCU ESP8266. NodeMCU berfungsi sebagai *gateway* yang mengumpulkan dan mengirimkan data ke server *database* menggunakan protokol *Hypertext Transfer Protocol* (HTTP). Kemudian data yang tersimpan di dalam server akan dikirimkan ke *smartphone* melalui API. Input Selanjutnya, ketika pengguna memindai *QR Code* pada pintu masuk, data akan dikelola di Arduino Mega 2560. Motor servo akan membuka portal, dan tampilan pada LCD akan memberikan informasi terkait. *Barcode scanner* GM65 telah diatur dengan metode *parking lock*.

2.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada sistem *smart parking* berfungsi untuk pengiriman data sensor melalui node ke database server menggunakan protokol HTTP. Program perangkat lunak dibuat berdasarkan desain perangkat keras untuk mengumpulkan data sensor, komunikasi data ke *gateway*, pengelolaan data di server, autentikasi dan kontrol akses. Pada Gambar 4 merupakan diagram alur dari rancangan perangkat lunak yang mengatur Proses transfer data dari sensor ke server *database*.

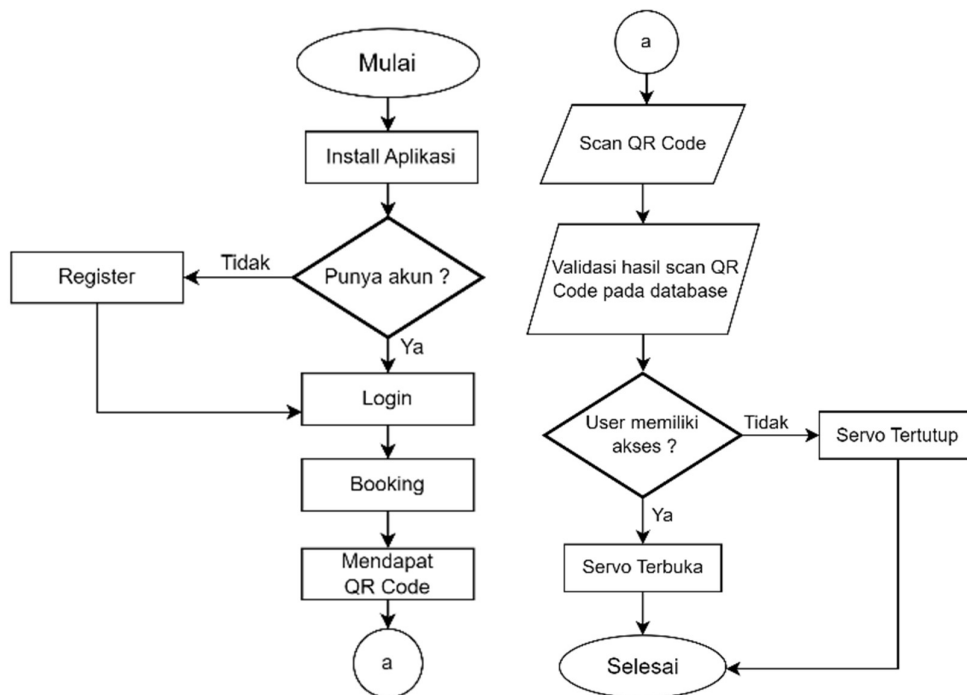


Gambar 3. Pengiriman Data Sensor ke Server *Database*

Data dari sensor IR *obstacle* dikumpulkan dan diproses oleh Arduino Mega. Selanjutnya, data tersebut dikirimkan melalui komunikasi serial ke NodeMCU ESP8266. NodeMCU berperan sebagai *gateway* yang menerima data dari Arduino Mega dan kemudian mengirimkannya ke server *database* menggunakan protokol HTTP. HTTP berperan sebagai protokol yang mengatur komunikasi antara klien dan server (Batubara, dkk, 2020). NodeMCU bertindak sebagai klien yang mengirimkan permintaan HTTP (*HTTP request*) ke server melalui URL yang terbentuk dari nilai-nilai yang diterima melalui komunikasi serial. Setelah koneksi WiFi terhubung, NodeMCU menggunakan objek *HTTPClient* untuk melakukan permintaan *GET* ke server. *Response* HTTP yang diterima berupa data JSON, kemudian diproses dan dikirimkan ke fungsi *accessDataFromJson()* untuk pengolahan lebih lanjut sesuai dengan format JSON dari

server. Perancangan program pengiriman data sensor ke server dilakukan melalui *software* Arduino IDE (*Integrated Development Environment*).

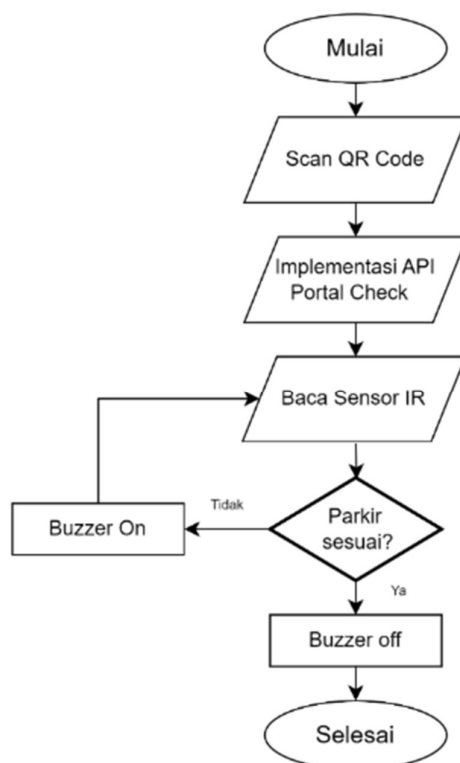
Perancangan mekanisme user dalam penelitian ini bertujuan sebagai penghubung antara mikrokontroler, aplikasi, dan server. Program dikembangkan pada aplikasi berfungsi untuk *men-generte QR Code* dan memberikan akses ke sistem parkir. Informasi dari mikrokontroler akan dikirimkan ke server menggunakan protokol HTTP. Server yang terhubung dengan *database booking* menerima data dari NodeMCU dan memvalidasinya. Proses validasi yaitu pengecekan data *QR Code* sesuai dengan data *booking* yang ada di *database*. Jika data valid maka servo akan terbuka jika tidak maka servo akan tertutup. Pada Gambar 4 menunjukkan mekanisme user.



Gambar 4. Mekanisme User untuk Akses Sistem Parkir Menggunakan *QR Code*

2.4 Perancangan Penerapan *Parking Lock*

Perancangan *parking lock* ini terkait dengan proses awal pemindaian *QR Code*, dimana *QR Code* telah diinisialisasi sesuai dengan slot yang telah dipesan oleh pengguna. Apabila *user* tidak parkir sesuai dengan slot yang telah dipesan, sistem akan memberikan *respons* berupa aktivasi *buzzer* sebagai indikator ketidaksesuaian tersebut. Pada Gambar 5 merupakan Integrasi *Parking Lock*.



Gambar 5. Integrasi *Parking Lock*

Perancangan *parking* proses dari sistem portal *check* yang memanfaatkan *QR Code* sebagai mekanisme akses ke dalam sistem parkir. Untuk menjadikan portal *check* ini berfungsi, diperlukan pembuatan perintah dalam Arduino IDE dengan menggunakan Arduino Mega sebagai mikrokontroler. Proses ini melibatkan komunikasi serial untuk membaca data dari *QR Code*, dimana data yang terbaca akan dikirimkan melalui API portal *check* ke aplikasi. Selanjutnya, penerapan *parking lock* menerapkan user harus parkir sesuai dengan slot yang telah dipesan.

2.5 Pengujian pada Sistem *Smart Parking*

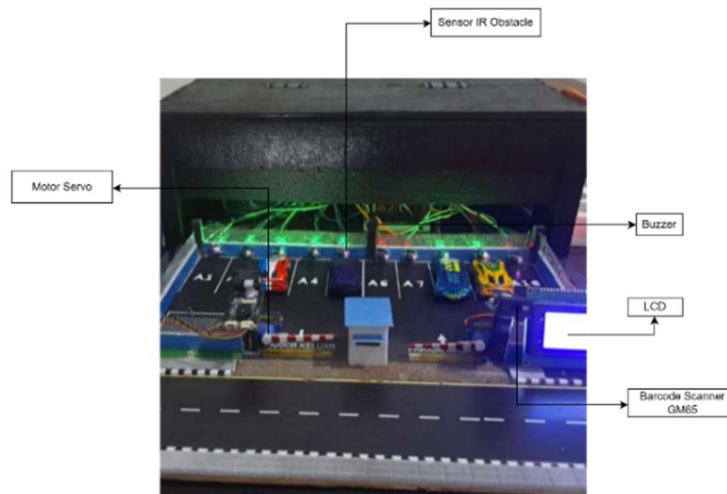
Pada penelitian ini, terdapat 3 macam pengujian kinerja sistem secara keseluruhan. Dalam pengujian ini, perangkat keras dan perangkat lunak yang sudah dipersiapkan digunakan dalam pengujian sistem. Data pengujian perangkat keras akan dikumpulkan menggunakan alat sistem parkir. Tes kinerja dari sistem parkir akan menampilkan informasi dengan mempertimbangkan parameter yang diuji sebagai berikut:

1. Akurasi dalam mendeteksi status slot parkir. Pengujian menilai keakuratan deteksi slot parkir pada alat dalam kondisi sebenarnya.
2. Kecepatan dan akurasi pemindaian *QR Code* Pengujian dilakukan dengan melihat kecepatan dan keakuratan pemindaian *QR Code*
3. Akurasi deteksi *parking lock* Pengujian dilakukan menilai keakuratan *parking lock* yaitu pengguna untuk dapat parkir sesuai slot yang dipesan.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Implementasi

Pada penerapan perangkat keras terdapat beberapa mikrokontroler utama yaitu sensor IR obstacle, Arduino Mega 2560, dan NodeMcu. Sistem terintegrasi untuk mengambil data dari sensor IR dan *scanner* GM65. sensor IR *obstacle* dapat secara efektif mendeteksi kendaraan yang memasuki slot parkir dengan memanfaatkan pantulan cahaya inframerah dari objek di depannya. Arduino bertugas mengumpulkan data dalam bentuk analog, kemudian mengonversinya menjadi data digital. NodeMcu berperan sebagai pengendali yang menerima, mengelola, dan menampilkan data yang berasal dari sensor node. Gambar 6 merupakan hasil *prototype* sistem *smart parking*.



Gambar 6. Prototype Smart Parking

Dalam sistem ini, proses implementasi perangkat keras mencakup akses ke sistem parkir menggunakan scan *QR Code*. Barcode scanner membaca data QR code dan memvalidasi pengecekan data *QR Code* sesuai dengan data booking yang ada di *database*. Jika *QR Code* memiliki validasi data yang sesuai maka servo portal terbuka. Implementasi *scan QR Code* pada portal sistem parkir tampak pada Gambar 7.



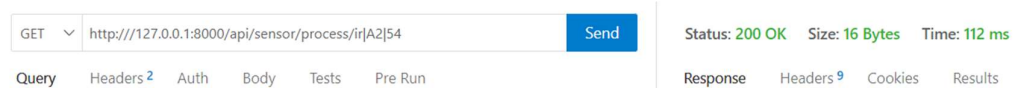
Gambar 7. Implementasi Scan QR Code pada Portal Sistem Parkir

Pada sistem kendali servo, servo terhubung dengan Arduino yang berfungsi untuk menerima data dari *barcode scanner*. Arduino kemudian mengelola proses validasi data pemindaian *QR Code* dengan mencocokkan data di *database*. Jika data pemindaian valid, Arduino akan mengirimkan sinyal untuk membuka servo jika data tidak valid servo akan tetap tertutup.



Gambar 8. (a) Servo Portal *Terbuka* (b) Servo Portal *Tertutup*

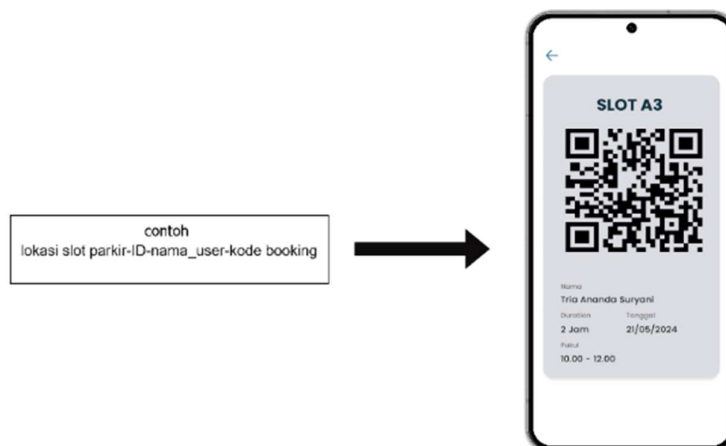
Dalam implementasi perangkat lunak, proses pengiriman data sensor ke server menggunakan protokol HTTP. Sistem komunikasi antara node sensor dan server diimplementasikan dengan menguji fungsi API melalui permintaan pada *endpoint*. Pengujian API yang telah dibuat bertujuan untuk memverifikasi koneksi antara node dengan server agar berfungsi sesuai harapan. Proses pengujian ini menggunakan API untuk mengirim atau meminta data. Pengujian API dilakukan di *Visual Studio Code* menggunakan *Thunder Client*.



Gambar 9. API Pengiriman Data Sensor ke Server

Gambar 7 memperlihatkan *respons* kode 200 dari API, yang mengindikasikan bahwa permintaan telah berhasil diproses oleh API. Pengiriman data memerlukan waktu 112 ms, dengan ukuran data sebesar 16 *byte*. Hasil ini menunjukkan komunikasi antara node dan server berjalan lancar dan dapat melakukan pengiriman dan penerimaan data sensor melalui API.

Implementasi sistem *smart parking* memiliki antarmuka aplikasi berbasis android untuk akses masuk dan keluar sistem parkir menggunakan *QR Code* dari pemesan di aplikasi dan integrasi *parking lock*. *QR Code* ini dibuat dengan menghasilkan data unik berdasarkan kode *booking* pengguna. Data yang di-*generate QR Code* mencakup informasi seperti lokasi slot parkir, ID, nama, dan kode *booking* yang sesuai dengan pemesanan Gambar 8, terlihat proses untuk memperoleh kode akses ke *QR Code*.













Gambar 10. Tampilan Halaman *QR Code*

Kode pada *QR Code* akan selalu bervariasi. Hal ini disebabkan informasi yang tersimpan dalam *QR Code* menampilkan informasi terbaru mengenai status akses, data pengguna dan detail pemesanan terkait.

3.2 Hasil Pengujian

Tabel 1 menampilkan hasil pengujian akurasi dalam mendeteksi slot parkir. Tabel tersebut menampilkan data pengujian kondisi *real* dan kemampuan sistem dalam mendeteksi status slot parkir terfokus pada akurasi komunikasi data antara sensor IR, Arduino, NodeMCU, dan server. Setiap kolom dalam tabel mencakup informasi seperti kondisi *real*, alat mendeteksi, Tampilan LCD, dan Akurasi. Dalam pengujian ini, terdapat 10 slot yang diuji dengan akun pengguna yang berbeda. Nilai akurasi pada pengujian deteksi slot parkir adalah 100%.

Tabel 1. Pengujian Akurasi Deteksi Slot Parkir

No	Kondisi Real	Alat Mendeteksi	Tampilan LCD	Akurasi
1.	A1	A1		100%
2.	A2	A2		100%
3.	A3	A3		100%
4.	A4	A4		100%
5.	A5	A5		100%
6.	A6	A6		100%
7.	A7	A7		100%
8.	A8	A8		100%
9.	A9	A9		100%
10.	A10	A10		100%

Untuk menghitung akurasi deteksi slot parkir berdasarkan data pada tabel dapat menggunakan rumus akurasi berikut:

$$Akurasi = \left(\frac{\text{Jumlah deteksi benar}}{\text{Jumlah total deteksi}} \right) \times 100\% \quad (1)$$

semua kondisi real (A1 sampai A10) terdeteksi dengan benar oleh alat. Jadi, jumlah deteksi benar adalah 10, dan jumlah total deteksi juga 10. Berikut hasil perhitungan akurasi:

$$Akurasi = \left(\frac{10}{10} \right) \times 100\% = 100\% \quad (2)$$

Tabel 2. Pengujian Deteksi dan Pengiriman Data Sensor Slot Parkir

No	Slot Parkir Terisi	Jarak deteksi Sensor	Sensor Point	Acces Point	Data ke Server	Verifikasi Data ke server
1.	A1	2,5 cm	Terdeteksi	Terkirim	Diterima	Sesuai
2.	A2	2.1 cm	Terdeteksi	Terkirim	Diterima	Sesuai
3.	A3	1.3 cm	Terdeteksi	Terkirim	Diterima	Sesuai
4.	A4	1.6 cm	Terdeteksi	Terkirim	Diterima	Sesuai
5.	A5	1.4 cm	Terdeteksi	Terkirim	Diterima	Sesuai
6.	A6	1.2 cm	Terdeteksi	Terkirim	Diterima	Sesuai
7.	A7	3.3 cm	Terdeteksi	Terkirim	Diterima	Sesuai
8.	A8	2.7 cm	Terdeteksi	Terkirim	Diterima	Sesuai
9.	A9	1.2 cm	Terdeteksi	Terkirim	Diterima	Sesuai
10.	A10	1.9 cm	Terdeteksi	Terkirim	Diterima	Sesuai

Tabel 2 merupakan hasil pengujian deteksi dan pengiriman data sensor slot parkir. Tabel ini menunjukkan pengujian slot parkir (A1 hingga A10) yang mendeteksi kendaraan pada jarak yang bervariasi. Data yang terdeteksi oleh sensor berhasil dikirim dan diterima oleh server, serta diverifikasi dengan benar. Ini menunjukkan bahwa sistem deteksi dan komunikasi antara sensor dan server berfungsi dengan baik. Pada Gambar 8, terlihat informasi mengenai jarak deteksi sensor IR untuk setiap slot parkir.

Pengujian kecepatan dan akurasi pemindaian *QR Code*. dilakukan untuk memastikan bahwa sistem pemindaian *QR Code* berfungsi dengan optimal dalam hal waktu yang diperlukan untuk kecepatan dan keakuratan data yang dipindai. ini melibatkan penggunaan *barcode scanner* GM65 yang membaca *QR Code*. Dapat dilihat pada Tabel 3 pengujian kecepatan Scan *QR Code*.

Pengujian kecepatan scan QR Code dilakukan dengan menggunakan stopwatch untuk mencatat waktu yang dibutuhkan dari saat QR Code dipindai oleh perangkat hingga saat motor servo pintu terbuka sepenuhnya. Pengujian dilakukan sebanyak 30 kali untuk memastikan hasil yang konsisten. Hasil pengujian menunjukkan rentang waktu dari 1,44 detik hingga 5,03 detik untuk proses scan QR Code, dengan rata-rata keseluruhan sebesar 2,88 detik. Pada Tabel 4 merupakan tabel keakuratan data yang dipindai dari *QR Code* untuk empat komponen yang berbeda yaitu *user*, *kode booking*, lokasi, dan *id_user*. Total akurasi pada pengujian keakuratan data yang dipindai *QR Code* adalah 100%.

Tabel 3. Pengujian Kecepatan *Scan QR Code*

No	Waktu Scan	Qr Code yang Dibaca	Kecepatan (s)
1.	14:01	6658415cd665841cd8243	2,69
2.	14:02	665321sd8a26966571774	2,18
3.	14:12	269665841cd8a26966571	1,44
4.	14:19	665841cd8a26966571774	3,11
5.	14:27	9cd665841cd8243555245	3,10
6.	14:32	269665841cd8a26966571	2,89
7.	14:41	5245789cd665841cd8243	1,92
8.	14:48	665841cd8243555245789	2,04
9.	14:45	841cd8243555245789cd6	3,30
10.	14:56	665841cd8243555245789	2,68
11.	15:06	841cd8243555245789cd6	2,36
12.	15:15	269665841cd8a26966571	3,11
13.	15:19	5245789cd665841cd8243	2,16
14.	15:27	665841cd8a26966571774	2,59
15.	15:29	665841cd8a26966571774	3,30
16.	15:33	665841cd8243555245789	2,74
17.	15:38	665841cd8a26966571774	2,79
18.	15:42	665841dcd665841cd8243	2,83
19.	15:48	665841cd8a26966571774	2,80
20.	15:55	665841cd8243555245789	2,78
21.	16:06	665841cd8a26966571774	4,15
22.	16:12	66584218cda6691090027	2,16
23.	16:13	665842575dd6b79017118	2,83
24.	16:19	665842ce420a659806672	3,53
25.	16:27	665842f91286885783420	5,03
26.	16:32	66584320dd5ab58941150	3,15
27.	16:46	66584342b2a0159196504	2,90
28.	16:49	66584368bb14474176334	2,67
29.	16:52	665843a9b706185549708	3,14
30.	16:55	6658440070de008038541	2,83

Tabel 4. Pengujian keakuratan data yang dipindai *QR Code*

No	Komponen QR Code	Total QR di uji	Jumlah Data Terbaca benar	Akurasi
1.	User	30	30	100%
2.	Kode booking	30	30	100%
3.	Lokasi	30	30	100%
4.	id_user	30	30	100%

Akurasi keakuratan yang didapat pada saat pemindaian *QR Code* mempengaruhi hasil dari implementasi *parking lock*, karena komponen yang di pindai oleh *QR Code* digunakan di sistem untuk menyesuaikan data penerapan *parking lock*. Pengujian keakuratan *parking lock* terdapat di Tabel 5.

Tabel 5. Pengujian keakuratan *Parking Lock*

No	Ketentuan Parkir	Kondisi Real Parkir	Kondisi Buzzer	Akurasi	ket
1.	A1, A5	A1, A5	Off	100%	sukses
2.	A2, A3, A1	A2, A3, A1	Off	100%	sukses
3.	A10, A9	A10, A9	Off	100%	sukses
4.	A10, A6, A7	A10, A5, A7	On	100%	sukses
5.	A2, A6	A2, A5	On	100%	sukses
6.	A9, A1	A9, A1	Off	100%	sukses
7.	A8, A4	A8, A2	On	100%	sukses
8.	A10, A9, A8	A10, A9, A8	Off	100%	sukses
9.	A2, A8, A7	A2, A8, A5	On	100%	sukses
10.	A8, A3	A8, A4	On	100%	sukses
11.	A3, A10	A3, A10	Off	100%	sukses
12.	A9, A1, A6,	A9, A1, A6,	Off	100%	sukses
13.	A10, A2	A10, A3	On	100%	sukses
14.	A3, A4	A3, A1	On	100%	sukses
15.	A9, A5, A4	A9, A5, A4	Off	100%	sukses
16.	A1, A2, A9	A1, A2, A9	Off	100%	sukses
17.	A7, A8	A7, A8	Off	100%	sukses
18.	A8, A2	A8, A9	On	100%	sukses
19.	A5, A4, A3	A5, A7, A3	On	100%	sukses
20.	A7, A8, A3	A7, A8, A3	Off	100%	sukses
21.	A5, A6, A7	A5, A6, A7	Off	100%	sukses
22.	A7, A6, A8	A7, A6, A8	Off	100%	sukses
23.	A3, A2, A1	A3, A5, A1	On	100%	sukses
24.	A10, A7	A9, A7	On	100%	sukses
25.	A9, A1	A9, A1	Off	100%	sukses
26.	A2, A9	A5, A9	On	100%	sukses
27.	A10, A9	A10, A9	Off	100%	sukses
28.	A1, A7	A2, A7	On	100%	sukses
29.	A6, A2	A6, A2	Off	100%	sukses
30.	A2, A3, A5	A2, A3, A5	Off	100%	sukses

Tabel ini digunakan untuk mengevaluasi kemampuan sistem *parking lock* dalam mengelola dan merespons kondisi parkir berdasarkan informasi yang terkandung dalam *QR Code*. Akurasi yang didapatkan yaitu 100% untuk setiap pengujian, sistem *parking lock* menunjukkan kehandalannya dalam mengelola berbagai skenario parkir yang berbeda untuk mengoptimalkan sistem *smart parking* yang dirancang.

4. KESIMPULAN

Dalam penelitian ini, telah dirancang sebuah sistem *smart parking* dengan mengoptimalkan integrasi *parking lock* menggunakan autentikasi *QR code* berbasis *Internet of Things*. Teknologi IOT beroperasi dengan mengumpulkan data dari sensor dan transmisi data nirkabel yang dapat diandalkan antara node sensor. Sistem ini sangat bergantung pada koneksi internet yang stabil dan andal untuk transmisi data. Pengumpulan data dan pengujian dilakukan menggunakan prototipe sistem dalam lingkungan yang terkendali. Penggunaan sensor IR obstacle untuk mendeteksi ketersediaan kendaraan di slot parkir dengan memanfaatkan pantulan cahaya inframerah dari objek di depannya menunjukkan akurasi 100%. Pengujian kecepatan *scan QR Code* adalah 1,44 detik hingga 5,03 detik, dengan rata-rata 2,88 detik. Total akurasi pada pengujian keakuratan data yang dipindai *QR Code* adalah 100%. Sistem *smart parking* ini tidak hanya meningkatkan efisiensi pencarian slot parkir, tetapi juga

meminimalkan risiko kesalahan parkir dengan integrasi *parking lock* dengan hasil pengujian keakuratan sebesar 100%.

DAFTAR RUJUKAN

- Al Rubiye, A. E. H., Qasim, H. H., Alwash, S. A., Fadhel, M. M., Audah, L., Ibrahim, H. H., Hamed, O. S., Saeed, H. A., Hussein, A. I., & Hamza, M. I. (2020). Design and implement WSN/IOT smart parking management system using microcontroller. *International Journal of Electrical and Computer Engineering*, *10*(3), 3108–3115. <https://doi.org/10.11591/ijece.v10i3.pp3108-3115>
- Alsaedi, N., & Jalal, A. S. A. (2023). Efficient automated car parking system based modified internet of spatial things in smart cities. *Indonesian Journal of Electrical Engineering and Computer Science*, *31*(2), 1164–1170. <https://doi.org/10.11591/ijeecs.v31.i2.pp1164-1170>
- Anitha, G., Krupasree, P., Kalaivani, K., & Nadheera, M. (2021). Embedded IoT Car Parking and Billing System. *2021 7th International Conference on Advanced Computing and Communication Systems, ICACCS 2021*, (pp. 797–801). <https://doi.org/10.1109/ICACCS51430.2021.9441757>
- Apriano, F. S., Asep, M., & Hafidudin. (2023). Perencanaan Aplikasi Android Untuk Ketersediaan Slot Parkir dan Simulasi E-Payment Dalam Sistem Smart Parking Berbasis IoT. *e-Proceeding of Applied Science*, *9*(3), (pp. 1150).
- Arfianto, M. R. (2022). Analisis Desain User Interface pada Aplikasi Pencari Parkir Mobil. *Jurnal Desain Produk dan Desain Komunikasi Visual*, *1*(1).
- Batubara, O. M., Ichsan, M. I. H., & Setiawan, E. (2020). Pengembangan Sistem Monitoring Kanal Air sebagai Sarana Penanggulangan Banjir. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, *4*(8), 2554–2561. <http://j-ptiik.ub.ac.id>
- Florence, S. Metilda., Mohan, U., Fancy, C., & Saranya, G. (2020). A study of remotely booking slot for vehicle using Internet of Things. *International Journal of Electrical and Computer Engineering*, *10*(5), 5392–5399. <https://doi.org/10.11591/IJECE.V10I5.PP5392-5399>
- Javale, D., Pahade, A., Singal, R., Potdar, A., & Patil, M. (2020). *Smart Detection of Parking Rates and Determining the Occupancy*, (pp. 381–390). https://doi.org/10.1007/978-981-15-0790-8_37
- Juwita, P. S., Fadhil, R., Damayanti, T. N., & Ramadan, D. N. (2020). Smart parking management system using SSGA MQTT and real-time database. *Telkomnika (Telecommunication Computing Electronics and Control)*, *18*(3), 1243–1251. <https://doi.org/10.12928/TELKOMNIKA.v18i3.14869>

- Patil, R. M., Vinay, N. R., & Pratiba, D. (2018). Application-based smart parking system using CAN bus. *Indonesian Journal of Electrical Engineering and Computer Science*, *12*(2), 759–764. <https://doi.org/10.11591/ijeecs.v12.i2.pp759-764>
- Puspitasari, D., Noprianto, Hendrawan, M. A., & Asmara, R. A. (2021). Development of smart parking system using internet of things concept. *Indonesian Journal of Electrical Engineering and Computer Science*, *24*(1), 611–620. <https://doi.org/10.11591/ijeecs.v24.i1.pp611-620>
- Setiadi, H., Priyandari, Y., & Cahyono, S. I. (2017). Implementation of Parking System Based on Radio Frequency Identification (RFID) at the Faculty of Engineering Sebelas Maret University Sukmaji Indro Cahyono. *ITSMART: Jurnal Teknologi dan Informasi*, *6*(1).
- Setiawan, J. C., Lim, R. M. E., & Andjarwirawan, J. M. Eng. (2020). Implementasi Internet of Things Untuk Parkir Mobil Dengan Pembayaran Menggunakan QR Code. *Jurnal Infra - Petra Christian University*, *8*.
- Suradi, A. A. M., Mushaf, Rasyid, M. F., M, A. R. J., & Bahtiar, A. (2023). Penerapan Teknologi QR Code Pada Sistem Informasi Parkir Berbasis Android. *Jurnal Sistem Informasi Dan Teknologi Informasi*, *12*(1).
- Syahnas, A., Mulyana, A., & Hafidudin. (2023). Perancangan Dan Realisasi Prototype Perangkat Keras Sistem Smart Parking Berbasis IoT. *eProceedings of Applied Science*, (pp. 171) .