

## Pengembangan Sistem Parkir Pintar Berbasis IoT

### *IoT-Based Smart Parking System Development*

Tito Nursyahbani<sup>1</sup>, Rendy Munadi<sup>2</sup>, Nyoman Bogi Karna<sup>3</sup>

<sup>1,2,3</sup> Universitas Telkom, Bandung

<sup>1</sup> titonursyahbani@telkomuniversity.ac.id, <sup>2</sup> rendymunadi@telkomuniversity.co.id  
, <sup>3</sup> aditya.at@telkomuniversity.ac.id

---

#### Abstrak

Parkiran merupakan tempat untuk meletakkan dan menitipkan kendaraan. Seiring pertumbuhan jumlah kendaraan, lahan parkir yang tersedia di tempat umum (pusat perbelanjaan, bandar udara, stasiun, dan lain-lain) semakin terbatas kapasitasnya. Untuk mempermudah pengemudi mencari lahan parkir, dibutuhkan sebuah sistem dan alat yang dapat membantu pengemudi. Oleh karena itu, penelitian ini merancang aplikasi dan alat yang bertujuan untuk mengatur pemesanan tempat parkir agar mempermudah pengemudi untuk mendapatkan lahan parkir. Perancangan ini meliputi *hardware* dan koneksi *Internet of Things* dengan aplikasi yang dibuat. Alat dirancang menggunakan Arduino Uno, ESP32 CAM, *buzzer* dan motor servo. Komponen tersebut terletak pada palang pintu otomatis. Untuk membuat aplikasi sistem parkir dibutuhkan aplikasi Android Studio dan penyimpanan *database* sistem ini menggunakan web *database* yaitu Firebase. Dari hasil pengujian, sistem *smart parking* memiliki kecepatan *rate transfer (Throughput)* sebesar 61292 bps, Total *delay* ESP32 CAM dengan Firebase sebesar 193.762 ms. . Penelitian ini, diharapkan *smart parking* dapat membantu para pengemudi untuk mendapatkan slot parkir dengan mudah.

**Kata kunci :** *Internet of Things, Smart parking, ESP32 CAM*

---

#### Abstract

The parking area is a site for placing and entrusting vehicles. With the growing number of vehicles, the available parking slot in public areas (shopping mall, airport, station, and others) is becoming rare in terms of capacity. To facilitate the driver in finding a parking slot, a system and its corresponding appliance are needed to solve the problem. Therefore, this research is designing the application and tools to arrange parking slot reservations to make the driver easier for getting the parking slot. This architecture includes hardware and Internet of Things connection with the created apps. The tools are created using Arduino Uno, ESP32 CAM, *buzzer*, and servo motor. The components are located at the automatic doorstop. To initiate the parking system application, Android Studio and database storage system using the web database, namely Firebase, are needed. From the testing stage, the smart parking system has a *Throughput* rate of 61292 bps. The delay total of ESP32 CAM with Firebase is 193.762 ms. This research is expected that *smart parking* is able to help the drivers to find parking slots easier.

**Keywords:** *Internet of Things, Smart parking, ESP32 CAM*

---

#### 1. Pendahuluan

Sistem parkir yang memadai diperlukan seiring dengan pertumbuhan jumlah kendaraan bermotor yang mengalami peningkatan setiap tahunnya. Sistem parkir yang diterapkan pada saat ini dinilai tidak efisien sehingga mengakibatkan kemacetan, polusi udara, dan meningkatkan penggunaan Bahan Bakar Minyak (BBM). Untuk mengatasi masalah tersebut, dibutuhkan sistem *smart parking* yang efektif dalam menemukan lahan parkir yang tersedia[1].

Sekarang ini, sudah ada beberapa sistem *smart parking* yang diterapkan kota besar. Sebagai contoh, pengaplikasian *smart parking* sistem di salah satu pusat perbelanjaan di Surabaya. Tiap lahan parkir dilengkapi oleh sensor yang dapat mendeteksi ada atau tidaknya kendaraan yang terparkir. Apabila lahan parkir terisi, maka sensor lampu yang ada akan berwarna merah dan apabila lahan parkir belum terisi maka sensor lampu yang ada akan berwarna hijau. Sehingga pengemudi

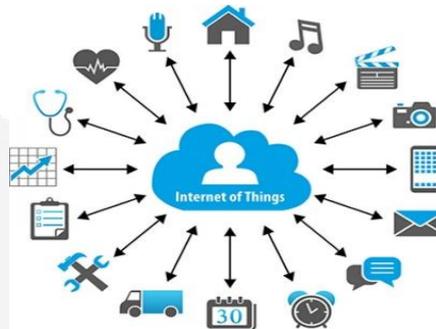
dapat melihat warna lampu tersebut dari jarak jauh. Namun demikian, sistem ini kurang efektif hal ini disebabkan oleh pengemudi harus tetap berkeliling mencari lahan parkir yang tersedia[2].

Melihat dari permasalahan yang ada pada sistem sistem sebelumnya, maka penulis mengusulkan untuk membuat aplikasi *Smart Parking* berbasis IoT yang memanfaatkan teknologi Aplikasi android, dan ESP32CAM. Aplikasi ini diharapkan dapat mempermudah para pengemudi untuk mengecek dan mendapatkan informasi tentang lahan parkir yang tersedia dilokasi parkir tanpa membuat kemacetan, membuang banyak waktu, dan membuang bahan bakar untuk mencari dan memesan lahan parkir kosong. Aplikasi ini diakses menggunakan *smartphone*.

## 2. Dasar Teori

### 2.1 Internet of Things

*Internet of Things* atau yang sering disebut IoT adalah konsep objek sehari-hari yang terhubung ke internet dan mampu mengidentifikasi ke perangkat lain. *Internet of Things* terdiri atas dua bagian utama yaitu Internet adalah sistem jaringan komputer yang saling berhubungan menggunakan paket *protocol* internet standar (TCP/IP) untuk melayani pengguna serta Internet yang mengatur konektivitas dan *Things* yang berarti objek atau perangkat[3]. Prinsip dari IoT adalah media yang membantu mempermudah kegiatan manusia sehari-hari. Teknologi sering diterapkan dalam bidang pertanian, energi, lingkungan, otomatisasi rumah, pemerintah, individu, dan kesehatan[4]. Teknologi IoT ini jika terealisasi akan sangat mempermudah kegiatan manusia sehari-hari tanpa harus mengoprasikannya satu persatu, alat atau mesin yang memiliki teknologi IoT akan bekerja dengan sendirinya.



Gambar 2. 1 *Internet of Things*

### 2.2 Smart Parking

*Smart Parking* atau parkir pintar adalah bagian dari internet yang menggunakan sensor yang berkomunikasi dengan menggunakan *remote* melalui internet dan berbagi informasi dengan menggunakan *protocol* komunikasi[5]. *Smart Parking* ini merupakan sistem monitoring dan keamanan untuk akses parkir, dengan adanya smart parking akan sangat membantu para pengemudi menemukan tempat parkir yang tersedia serta menjaga keamanan kendaraan. *Smart Parking* juga merupakan salah satu dari banyak teknologi yang diimplementasikan pada konsep *smart city*[6].



Gambar 2. 2 *Smart Parking*

### 2.3 QR Code

*The Quick -Response Code* atau yang biasa dikenal dengan *QR-Code* adalah sebuah kode batang (barcode) dua-dimensi dengan pola berwarna monokrom. *QR-code* yang telah terstandarisasi terdapat 40 versi. *QR-code* telah banyak digunakan dalam berbagai aplikasi untuk menyimpan dan mengambil data secara *real-time*, seperti media sosial, *e-commerce*, dan lainnya setelah diusulkan oleh perusahaan Denso di Jepang. Pada berbagai aplikasi tersebut, kode batang dua dimensi dapat digunakan sebagai alat untuk mendapatkan data yang tersimpan didalam server yang terkoneksi internet dengan kualitas yang tinggi. *QR-code* yang adalah sebagaimana standar paten publik, maka *QR-code* dapat dibaca oleh berbagai standar pembaca *QR-code*[7].



Gambar 2. 3 Contoh QR-Code

### 2.4 ESP32 CAM

ESP32 CAM merupakan kamera pembaca *QR-code*, dilengkapi koneksi WiFi dan *Bluetooth* yang *Low* serta slot MicroSD. Membuat pengguna dapat membuat sistem yang berkonsep *Internet of Things*. ESP32 CAM sangat mudah diprogram menggunakan Arduino IDE[8].

Spesifikasi:

- Kamera OV2640 2 megapiksel
- WIFI: 802.11 b / g / n / e / i (hingga 150 Mbps)
- Bluetooth v4.2BR / EDR dan standar BLE
- Kartu MicroSD maksimum 4 GB
- Tegangan 5V
- Tegangan pemrograman 3.3V
- Dimensi 40.5mm x27mm x4.5mm
- CPU 32-bit dual-core berdaya rendah untuk prosesor aplikasi
- Frekuensi utama hingga 240MHz, daya komputasi hingga 600 DMIPS
- Mendukung beberapa mode tidur
- Tertanam Lwip dan FreeRTOS



Gambar 2. 4 ESP32 CAM

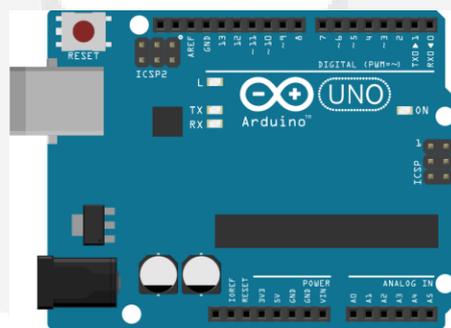
### 2.5 Arduino Uno

Proses pengontrolan gerakan diperlukan mikrokontroler, mikrokontroler yang digunakan adalah Arduino Uno. Arduino Uno adalah board mikrokontroler berbasis ATmega328 (*datasheet*). Memiliki 14 pin *input* dari *output* digital dimana 6 pin input tersebut dapat digunakan sebagai *output* PWM dan 6 pin *input* analog, 16 MHz osilator kristal, koneksi USB, *jack power*, ICSP header, dan tombol reset[9]. Arduino uno menyediakan 20 pin I/O, yang terdiri dari 6 pin *input* analog dan 14 pin digital input/output untuk pin 6 analog sendiri bisa difungsikan sebagai *output* digital jika dibutuhkan *output* tambahan selain 14 pin yang telah tersedia.

Spesifikasi Arduino Uno[10]:

**Tabel 2. 1** Spesifikasi Arduino Uno

Spesifikasi	Keterangan
Mikrokontroler	ATmega328P
Tegangan	5 Volt dan 3.3 Volt
Tegangan Masukan	7-12 Volt
Tegangan Masukan	6-20 Volt
Pin Digital I/O	14 (6 sebagai output pwm)
Pin PWN	6
Arus DC per I/O Pin	20 mA
Arus DC pin 3.3 V	50 mA
Flash Memory	32 KB (Atmega328P), 0.5 KB digunakan bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g



**Gambar 2. 5** Arduino Uno

## 2.6 Android Studio

Dalam pengembangan aplikasi android, dibutuhkan suatu sistem yang dapat membantu pengembangan. Android studio adalah sistem pengembangan aplikasi android baru berdasarkan IntelliJ IDEA. Android studio ini menyediakan alat pengembangan Android yang terintegrasi untuk pengembangan dan debugging.

Android Studio memiliki fitur-fitur antara lain[11]:

1. Emulator yang cepat dan kaya fitur

2. *Framework* dan fitur pengujian yang lengkap
3. Fitur lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya
4. Dukungan C++ dan NDK
5. Dukungan bawaan untuk *Google Cloud Platform*, yang memudahkan integrasi *Google Cloud Messaging* dan *App Engine*.



**Gambar 2. 6** Android Studio

## 2.7 Firebase

Firebase *database* merupakan penyimpanan data basis data non SQL yang memungkinkan untuk menyimpan beberapa tipe data. Tipe data itu antara lain *String*, *long*, dan *Boolean*. Firebase menawarkan dua solusi database berbasis cloud yang dapat diakses client, yaitu[12]:

1. *Realtime Database* adalah *database* asli Firebase. Solusi ini berlatensi rendah dan efisien yang membutuhkan status sinkronisasi secara *real-time*
2. *Cloud Firestore* adalah database unggulan baru dari Firebase untuk pengembangan aplikasi seluler. Cloud Firestore juga memiliki fitur yang lebih kaya, cepat, dan lebih baik dari *Realtime Database*.



**Gambar 2. 7** Firebase

## 2.8 Paramater *Quality of Service* (QoS)

### 2.8.1 *Throughput*

*Throughput* merupakan kecepatan rate transfer data efektif, yang diukur dalam bps. *Throughput* merupakan jumlah total kedatangan paket yang sukses yang diamati pada destinasi selama interval waktu tertentu dibagi durasi interval waktu tersebut[13].

$$\textit{Throughput} = \frac{\textit{Data diterima}}{\textit{Waktu pengiriman data}}$$

### 2.8.2 *Delay*

*Delay* adalah total waktu tunda suatu paket yang dikirimkan[13].

$$\textit{Delay} = \frac{\textit{Total delay}}{\textit{Total paket diterima}}$$

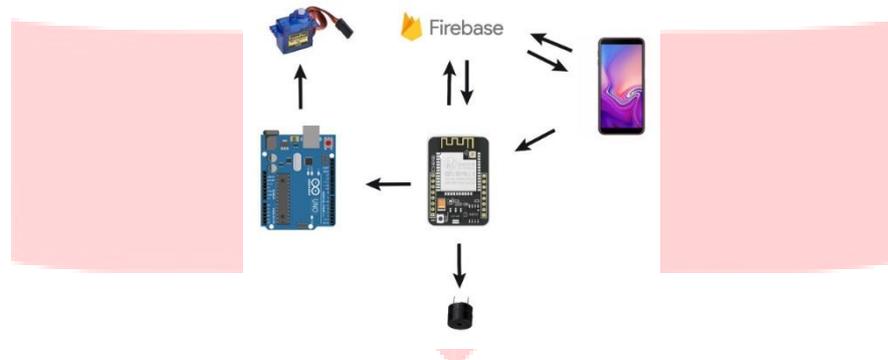
## 3. Perancangan dan Simulasi Sistem

### 3.1 Desain Sistem

Pada sistem parkir sekarang menggunakan sistem parkir yang kurang efektif dan masih membuat para pengemudi membuang banyak waktu untuk mencari lahan parkir yang tersedia, dengan perkembangan teknologi maka dibutuhkan suatu aplikasi yang dapat memudahkan para pengemudi untuk mendapatkan lahan parkir. Aplikasi *smart parking* dan *Internet of Things* dapat memudahkan kegiatan pengemudi. Desain sistem Tugas Akhir ini menggunakan komponen ESP32 CAM yang terhubung dengan mikrokontroler Arduino Uno dan aplikasi android untuk membuka palang pintu masuk dan keluar parkir menggunakan servo. Aplikasi android dibuat dengan

menggunakan Android Studio, aplikasi android ini terhubung dengan Firebase dan dari aplikasi android dapat memberikan database kepada admin. Pada aplikasi android untuk user dapat membuat pesanan lahan parkir dan mendapat QR-code yang akan di scan pada ESP32CAM yang berada di palang pintu masuk dan keluar yang berfungsi untuk membuka palang tersebut, kemudian user langsung menuju tempat parkir yang telah dipesan.

### 3.2 Diagram Blok

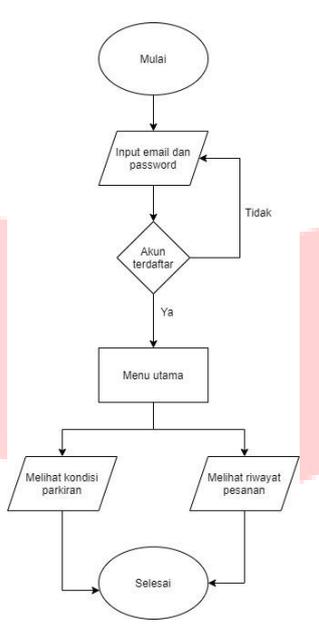
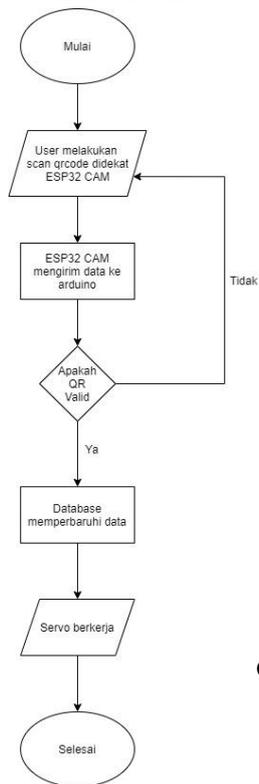


Gambar 3. 1 Diagram blok

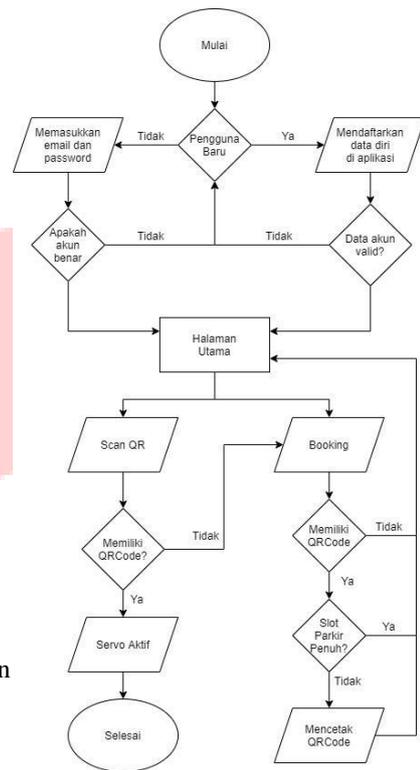
Sistem dimulai saat user atau pengemudi melakukan registrasi dan login aplikasi android. Saat pengemudi telah login, pengemudi bisa langsung memilih dan pesan lahan parkir yang tersedia. Aplikasi mengirimkan QR-code pada pengemudi, pengemudi men-scan QR-code tersebut pada ESP32CAM. Saat QR-code telah di scan Arduino Uno memerintahkan servo untuk membuka palang pintu.

Pada aplikasi android selain memiliki login *user*, aplikasi android juga memiliki fitur login admin berfungsi untuk mengecek data parkir yaitu riwayat parkir, parkir *on going* dan persentase parkir yang terhubung dengan Firebase. Fitur admin juga dilengkapi pengaturan lokasi parkir yang dapat diubah. Aplikasi android, sistem dimulai dengan mengirim data ke Firebase, setelah itu log data dikirimkan ke Arduino Uno. Log data pada Firebase dapat mengontrol servo untuk membuka dan menutup palang pintu.

### 3.3 Flowchart Sistem



Gambar 3. 2 Flowchart Admin



Gambar 3. 4 Flowchart User

Gambar 3. 3 Flowchart Alat

### 3.4 Komponen dan Pendukung Sistem

Untuk menjalankan seluruh sistem dibutuhkan komponen-komponen sebagai berikut:

Tabel 3. 1 Komponen Sistem

Komponen	Fungsi
Arduino Uno	Mikrokontroller sistem
ESP32 CAM	Pembaca QR-code
Servo	Palang pintu otomatis

### 3.5 Aplikasi dan Software

Tugas Akhir ini membutuhkan aplikasi dan *software* untuk membuat program aplikasi *Smart Parking System* yang di dalamnya terdapat *user* dan *admin*. Berikut ini adalah aplikasi dan *software* pendukung Tugas Akhir:

Tabel 3. 2 Komponen Pendukung Aplikasi

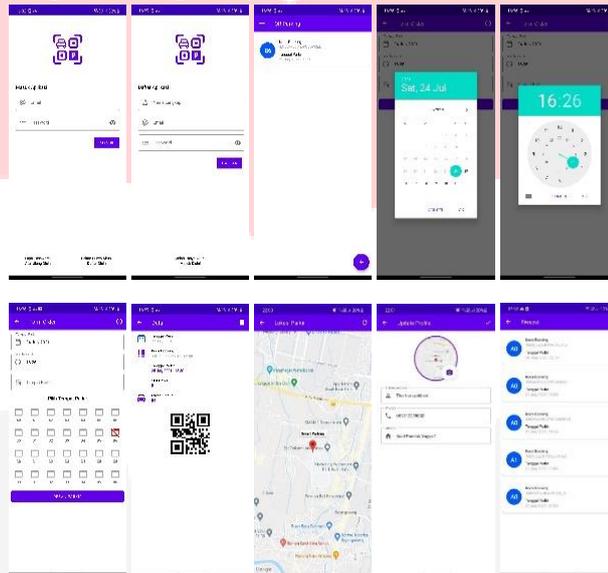
Software	Fungsi
Visual Studio Code	Software pembuatan sistem
Android Studio	Software pembuatan aplikasi
Java	Bahasa pemrograman aplikasi
C++	Bahasa pemrograman sistem

## 4. Hasil Pengujian

### 4.1 Implementasi Aplikasi

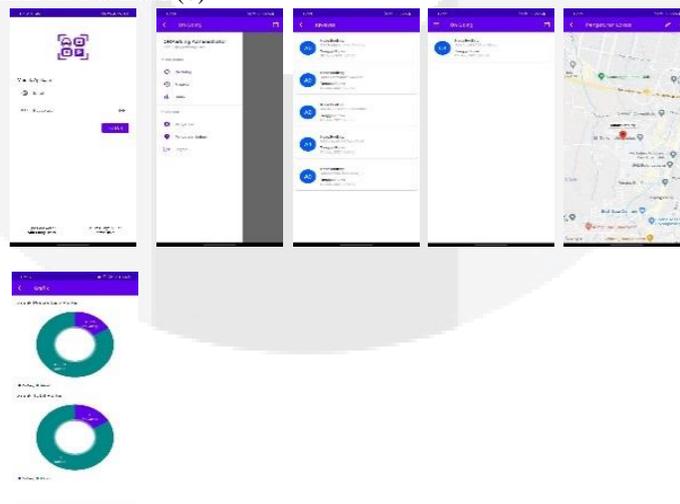
#### 4.1.1 Aplikasi untuk User

Pada menu login, *user* memasukkan *username* dan *password* (1). Jika *user* belum memiliki akun, aplikasi terdapat menu daftar untuk *user* (2). Setelah login, terdapat menu pembuatan dan hasil pemesanan (3). Di dalam menu pembuatan pesanan terdapat pemilihan tanggal (4), waktu (5) dan lahan parkir yang ingin dipesan (6). Setelah membuat pesanan, aplikasi memberikan detail pesanan dan QR-Code (7). Aplikasi ini terdapat fitur yaitu Maps untuk mengarahkan *user* ke tempat parkir (8), *Update Profile* untuk mengisi data *user* (9) dan riwayat untuk menampilkan data parkir yang selesai (10).



#### 4.1.2 Aplikasi untuk Admin

Pada menu login, admin memasukkan *username* dan *password* (1). Admin memiliki beberapa fitur yaitu *on going*, riwayat, grafik dan pengaturan lokasi (2). Menu fitur riwayat, untuk menampilkan data parkir yang telah selesai (3). Menu fitur *on going*, untuk menampilkan data parkir yang sedang berjalan (4). Menu fitur pengaturan lokasi, untuk mengubah atau menambah lokasi parkir (5). Menu fitur grafik, untuk menampilkan total grafik pengguna parkir yang sedang berjalan atau selesai (6).



## 4.2 Pengujian Hardware

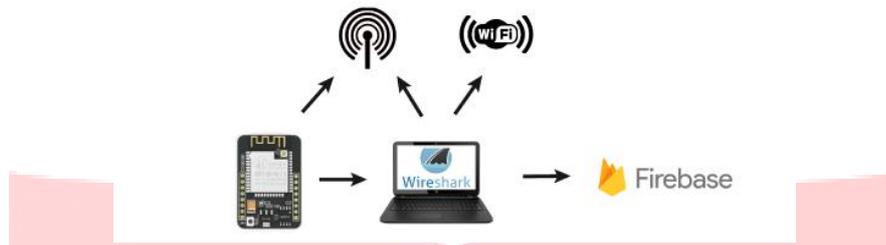


Gambar 4. 1 Desain Hardware

Tabel 4. 1 Pengujian Hardware

Hardware	Indikator	Skenario	Status
Arduino	Lampu indikator pada Arduino Uno menyala dan dapat menjalankan program	Menghubungkan ke laptop dan <i>running</i> program	Berhasil
ESP32CAM dan Buzzer	ESP32CAM membaca QR-code	ESP32CAM membaca QR-code yang di generate aplikasi, jika QR-code terdaftar maka buzzer akan berbunyi dua kali	Berhasil
	ESP32CAM membaca QR-code	ESP32CAM membaca QR-code yang salah atau tidak terdaftar pada aplikasi, maka buzzer akan berbunyi panjang satu kali	Berhasil
Motor Servo	Palang Pintu Terbuka	Motor servo berputar ke sudut 90°	Berhasil
	Palang Pintu Tertutup	Motor servo berputar ke sudut 0°	Berhasil

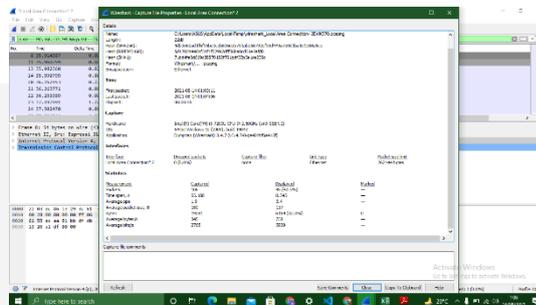
### 4.3 Pengujian *Quality of Service* (QoS)



Gambar 4. 2 Skenario pengujian

Skenario pengujian menggunakan Wireshark ini untuk mengetahui kinerja jaringan ESP32 CAM mengirim data ke Firebase sebagai web database yang digunakan. Cara pengujian, laptop connect Wi-Fi kemudian mengaktifkan mobile hotspot untuk koneksi ESP32 CAM dan didapatkan IP ESP32 CAM adalah 192.168.137.245 dan IP tujuan adalah 35.201.97.85. Cara untuk filter paket yang dikirimkan ke Firebase pada wireshark adalah `ip.src==192.168.137.245 && ip.dst==35.201.97.85`. Informasi yang terdapat pada wireshark [FIN, ACK] menandakan bahwa paket berhasil terkirim ke IP tujuan.

#### 4.3.1 *Throughput*



Gambar 4. 3 Pegujian *Throughput* menggunakan Wireshark

*Throughput* yang diobservasi dalam pengujian ini ialah *Throughput* antara ESP32 CAM dan Firebase. Pengujian ini menggunakan aplikasi Wireshark dengan cara scan QR-code menggunakan ESP32 CAM, filter paket yang terkirim ke Firebase dengan cara `ip.src==192.168.137.245 && ip.dst==35.201.97.85` dan terdapat informasi [FIN, ACK] kemudian untuk melihat hasil *Throughput* dengan cara memilih statistics lalu capture. Pengujian dilakukan sebanyak 3 (tiga) kali dengan 10 percobaan setiap pengujiannya dan waktu pengujian pada pagi, siang dan malam. Hasil *Throughput* adalah 61292 bps. Menurut standarisasi TIPHON dengan hasil yang didapatkan, maka pengiriman total data tiap-tiap satuan waktu dari ESP32 CAM menuju Firebase tergolong kategori sangat baik dengan indeks 4. Hasil pengujian *Throughput* bisa diamati dari tabel 4.2.

Tabel 4. 2 Hasil Pengujian *Throughput*

Pengujian	Percobaan	<i>Throughput</i>
Pengujian 1 (Malam)	1 - 10	56200 bps
Pengujian 2 (Siang)	1 - 10	62710 bps
Pengujian 3 (Pagi)	1 - 10	64966 bps
Rata-Rata		61292 bps

#### 4.3.2 *Delay*

Pengujian ini dilaksanakan guna mengetahui kekuatan jaringan yang dipakai sebab jaringan yang baik ialah jaringan yang memiliki angka *delay* yang rendah. Untuk melihat dan mendapatkan hasil pengujian dengan cara memilih file, *export packet dissections* kemudian pilih As CSV, kemudian akan didapatkan hasil pengujian *delay*. Pengujian *delay* ESP32 CAM ke Firebase dilakukan sebanyak 3 (tiga) kali dengan 10 percobaan setiap pengujian dan waktu pengujian pada pagi, siang dan malam. Hasil *delay* 193.762 ms. Menurut standarisasi TIPHON berdasarkan hasil didapatkan, maka waktu pengiriman data dari ESP32 CAM menuju Firebase termasuk kategori baik dengan indeks 3. Hasil bisa diamati pada tabel 4.3.

**Tabel 4. 3 Hasil Pengujian Delay**

Pengujian	Percobaan	Delay
Pengujian 1 (Malam)	1 – 10	237.803 ms
Pengujian 2 (Siang)	1 – 10	180.184 ms
Pengujian 3 (Pagi)	1 – 10	163.298 ms
Rata-Rata		193.762 ms

## 5 Kesimpulan dan Saran

### 5.1 Kesimpulan

1. Aplikasi dan alat yang sudah dibuat untuk monitoring system yang akan memberikan informasi tentang slot parkir yang tersedia, tidak tersedia dan lokasi parkir akan diarahkan menggunakan maps yang ada didalam aplikasi yang bisa di direct menggunakan google maps.
2. Parameter pengujian Throughput ESP32 CAM menuju Firebase mendapatkan hasil 61292 bps. Hasil Throughput termasuk kategori sangat baik dengan indeks 4, menurut standarisasi TIPHON.
3. Parameter pengujian Delay ESP32 CAM menuju Firebase mendapatkan hasil 193.762 ms. Hasil Delay termasuk kategori baik dengan indeks 3, menurut standarisasi TIPHON.

### Saran

1. QR-code bisa digantikan dengan nomor polisi kendaraan pengguna. Pengguna memasukkan nomor polisi kendaraan ke dalam aplikasi. Jadi, nomor polisi tersebut digunakan sebagai akses untuk membuka palang pintu otomatis.
2. Menambahkan server untuk memberikan kadaluarsa pesanan.
3. Menambahkan fitur payment method pada aplikasi.
4. Fungsi filter riwayat pada aplikasi.
5. Melihat posisi parkir user sesuai atau tidak dengan yang terpesan pada aplikasi.
6. Membuat aplikasi agar saat sinyal offline tetap bisa digunakan.

### Referensi:

- [1] D. Jenderal *et al.*, "Rencana Strategis 2015-2019."
- [2] G. R. Pradana, "Smart Parking Berbasis Arduino Uno," *Univ. Negeri Yogyakarta*, no. 12507134001, pp. 1–9, 2015.
- [3] A. N. Baharsyah, "Pengertian Internet of Things." 2019, [Online]. Available: <https://www.jagoanhosting.com/blog/pengertian-internet-of-things-iot/>.
- [4] Hosteko, "Internet of Things Pengertian Prinsip dan Contoh." [Online]. Available: <https://hosteko.com/blog/internet-of-things-iot-pengertian-prinsip-dan-contoh>.
- [5] Aknovia, "Smart Parking." 2018, [Online]. Available: <https://garudacyber.co.id/artikel/207-apa-itu-smart-parking>.
- [6] J. J. Barriga *et al.*, "Smart parking: A literature review from the technological perspective," *Appl. Sci.*, vol. 9, no. 21, 2019, doi: 10.3390/app9214569.
- [7] L. F. Freitas, A. R. Nogueira, and M. E. V. Melgar, "Data Validation System Using QR Code and Meaningless Reversible Degradation," *Int. Conf. Appl. Electron.*, vol. 2019-September, pp. 1–4, 2019, doi: 10.23919/AE.2019.8867027.
- [8] Dewantoro. D. W, "Rancang Bangun Lengan Robot Pemilah Barang Berdasarkan Berat

dengan Pemanfaatan Internet Of Things (IoT) Sebagai Kontrol Dan Monitoring Jarak Jauh,” *Semin. Has. Elektro S1 ITN Malang*, vol. 21, no. 1, pp. 1–9, 2020.

- [9] F. T. Elektro and U. Telkom, “MIKROKONTROLLER ARDUINO UNO DESIGN OF AUTOMATIC CLOTHESLINE DRIVE SYSTEM BASED THE ARDUINO.”
- [10] Il. Media, “Pengertian Arduino Uno.” [Online]. Available: <https://ilearning.me/sample-page-162/arduino/pengertian-arduino-uno/>.
- [11] A. S. Developers, “Mengenal Android Studio.” Android Studio, [Online]. Available: <https://developer.android.com/studio/intro?hl=id>.
- [12] G. Developer, “Firebase.” [Online]. Available: <https://firebase.google.com>.
- [13] R. Wulandari, “ANALISIS QoS (QUALITY OF SERVICE) PADA JARINGAN INTERNET (STUDI KASUS : UPT LOKA UJI TEKNIK PENAMBANGAN JAMPANG KULON – LIPI),” *J. Tek. Inform. dan Sist. Inf.*, vol. 2, no. 2, pp. 162–172, 2016, doi: 10.28932/jutisi.v2i2.454.

