

## Implementasi Smart Parking dengan Maps Pada *End User* di Pusat Perbelanjaan Menggunakan Metode *QR Code*

### *Implementation of Smart Parking with Maps on End User in Buildings Using the QR Code Methode*

Karyn Azzahra Palar<sup>1</sup>, Rendy Munadi<sup>2</sup>, Nyoman Bogi Aditya Karna<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>[karinpalar@student.telkomuniversity.ac.id](mailto:karinpalar@student.telkomuniversity.ac.id), <sup>2</sup>[rendymunadi@telkomuniversity.ac.id](mailto:rendymunadi@telkomuniversity.ac.id),

<sup>3</sup>[aditya.at@telkomuniversity.ac.id](mailto:aditya.at@telkomuniversity.ac.id)

---

#### Abstrak

Saat ini, penduduk di suatu daerah berkembang sangat pesat. Oleh sebab itu banyak diantara mereka yang memiliki kendaraan untuk memenuhi kebutuhan mereka. Banyak pengguna kendaraan mobil khususnya yang datang ke pusat perbelanjaan dan ingin belanja bulanan, namun sering terjadi di tempat parkir *mall/supermarket* yang penuh karena banyaknya orang yang berbelanja di tempat tersebut. Jadi masyarakat kesulitan mencari parkir yang kosong dan jalan ke tempat parkir yang kosong tersebut. Tujuan dari penelitian ini adalah dapat merancang dan mengimplementasikan *wireless sensor network* untuk *monitoring system* yang memberikan informasi parkir yang tersedia dan tidak tersedia serta *maps* yang dibutuhkan lalu diteruskan ke *user*. Hasil yang di dapat, *Prototype smart parking* yang telah dirancang dapat berfungsi dengan baik sesuai dengan yang diharapkan, sensor dapat mendeteksi objek kendaraan yang masuk ke area parkir dengan jarak maksimum 200 cm. Hasil *Throughput* adalah 639k bps. Hasil dari pengujian packet loss pada NodeMCU ke *Thingspeak* mendapatkan hasil yang sangat baik yaitu 0%. Hasil dari pengujian delay NodeMCU dengan sensor dan *Thingspeak* di dapatkan total delay 218ms dan rata rata delay 0.07 ms. Menurut standarisasi TIPHON QoS termasuk bagus dengan indeks 4.

**Kata kunci:** *Smart parking, Maps, QR Code.*

---

#### Abstract

*Currently, the population in an area is growing very rapidly. Therefore, many of them have vehicles to meet their needs. Many car users, especially those who come to shopping centers and want to shop monthly, but this often happens in the parking lot of malls / supermarkets that are full because of the large number of people who shop there. So people find it difficult to find an empty parking lot and get to the empty parking lot. The purpose of this research is to be able to design and implement a wireless sensor network for monitoring systems that provide available and unavailable parking information as well as the required maps and then forward it to the user. The results obtained, the smart parking prototype that has been designed can function properly as expected, the sensor can detect vehicle objects that enter the parking area with a maximum distance of 200 cm. The result of throughput is 639k bps. The results of packet loss testing at NodeMCU to Thingspeak get very good results, namely 0%. The results of the NodeMCU delay test with sensors and Thingspeak obtained a total delay of 218 ms and an average delay of 0.07 ms. According to the TIPHON QoS standard, it is quite good with an index of 4.*

**Keywords:** *Smart parking, Maps, QR Code.*

---

#### 1. Pendahuluan

Seiring dengan majunya zaman, Banyak pengguna kendaraan mobil khususnya yang datang ke pusat perbelanjaan dan ingin belanja bulanan, namun sering terjadi di tempat parkir *mall/supermarket* yang penuh karena banyaknya orang yang berbelanja di tempat tersebut. Jadi masyarakat kesulitan mencari parkir yang kosong dan jalan ke tempat parkir yang kosong tersebut.

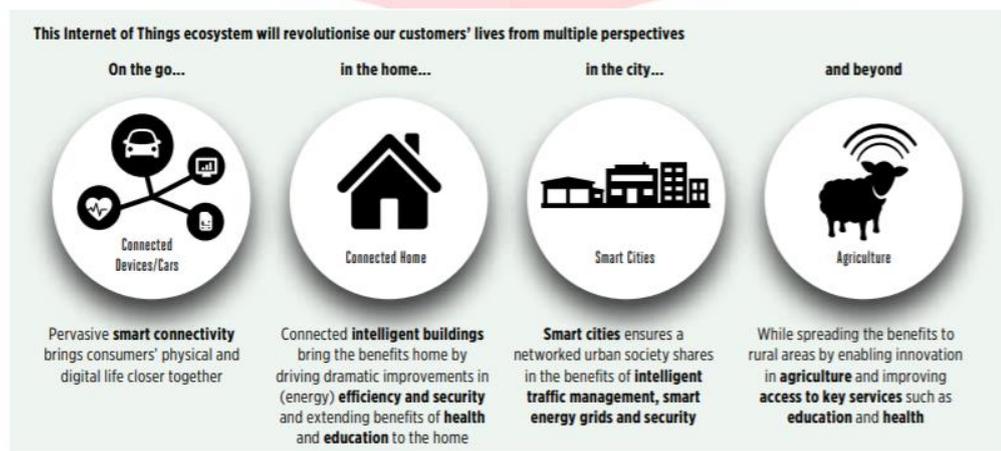
Dari beberapa masalah di atas, banyak pengembangan *smart parking* yang telah di kembangkan oleh penelitian-penelitian sebelumnya. Pada penelitian sebelumnya, membahas tentang aplikasi *smart parking* berbasis android. Aplikasi android yang digunakan yaitu android studio dan menggunakan bahasa pemrograman java. Mikrokontroler yang digunakan penelitian ini adalah Arduino dan menggunakan Bahasa C. Penelitian ini bertujuan untuk membantu pengendara mencari tempat parkir yang kosong. Penelitian ini memiliki kekurangan yaitu tidak terdapat rute sebagai petunjuk menuju tempat parkir yang kosong [1]. Penelitian selanjutnya membahas tentang pengendara diarahkan ke tempat parkir yang kosong dan setiap kendaraan yang masuk ke dalam diberikan kode akses untuk akses keluar kendaraan tersebut. Penelitian ini menggunakan Arduino Uno R3 dan Raspberry Pi (model 3 B) sebagai rancangan dan *prototype smart office system*. Hasil uji coba sistem ini menunjukkan tingkat keberhasilan 60% dari 10 plat nomor yang berbeda. Penelitian ini mengarahkan ke tempat parkir kosong menggunakan aplikasi smart parking system berbasis android yang terintegrasi dengan Arduino [2].

Dari beberapa penelitian di atas, peneliti perlu mengembangkan *smart parking* dengan membuat suatu alat menggunakan sensor ultrasonik yang dihubungkan dengan Arduino. Sensor ultrasonik dipasangkan pada basement parkir untuk mendeteksi parkir yang tersedia atau tidak tersedia, sehingga diperoleh data slot parkir yang disarankan. Penelitian ini melibatkan dua bagian yaitu *end user* dan *user*. Peneliti akan fokus ke *end user*, yaitu memastikan data parkir yang tersedia dan tidak tersedia serta mengambil data parkir yang disarankan. Data tersebut akan dikirimkan ke *website* agar user mengetahui informasi mengenai parkir yang tersedia dan tidak tersedia serta slot parkir yang disarankan dengan rute dan *maps* yang dibutuhkan.

## 2. Konsep Dasar

### 2.1 Internet Of Things (IoT)

Istilah IoT awalnya disarankan oleh Kevin Ashton pada tahun 1999 dan mulai terkenal melalui Auto-ID Center di MIT. Dan kini IoT menjadi salah satu tugas bagi seorang mahasiswa di sebuah perguruan tinggi [1]. IoT telah menjadi *buzzword* yang bagi para pengguna internet di Indonesia. IoT pertama kali diperkenalkan oleh Kevin Ashton pada tahun 1999. IoT memiliki pengertian internet yang telah berintegrasi ke komputer, handphone, dan peralatan elektronik lainnya untuk berinteraksi dengan objek lain, lingkungan maupun dengan peralatan komputasi cerdas lainnya melalui jaringan internet [3]. Beberapa penelitian awal mengenai konsep IoT salah satunya dilakukan oleh Neil Gershenfeld dari Massachuset Institute Technology (MIT) media Lab. Neil Gershenfeld menyebutnya 'Internet O', Konsep yang mengeksplor bagaimana objek dapat berkomunikasi dengan kita, bagaimana cara kita berkomunikasi dengan benda-benda tersebut, dan bagaimana benda-benda tersebut dapat berkomunikasi satu sama lain, sehingga kita dapat mengendalikan kehidupan menjadi lebih baik [3].



Gambar 2.1 Penerapan IoT [4]

### 2.2 Smart Parking

*Smart parking* merupakan suatu sistem parkir otomatis dan dapat di-monitoring melalui jarak jauh dengan menggunakan teknologi IoT sebagai media penghubung [5]. Di dalam *smart parking* yang dirancang oleh sistem terdapat *smart guides parking system*. *Smart guides parking system* merupakan sistem pemandu yang berada pada area parkir yang memungkinkan pemberian informasi dan petunjuk kepada pengemudi untuk memilih area parkir yang kosong. Pendeteksian kendaraan yang parkir menggunakan sensor ultrasonik. Sensor tersebut diletakkan di setiap slot parkir dan pada pintu masuk parkir untuk menghitung jumlah area parkir yang kosong [6].



Gambar 2.2 Smart Parking [7]

### 2.3 Sensor Ultrasonik HC-SR04

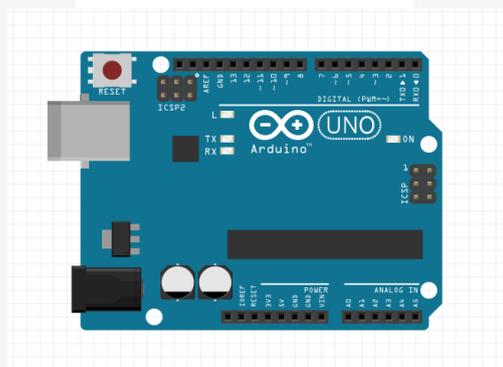
Sensor ultrasonik adalah sensor yang bekerja berdasarkan prinsip pantulan gelombang suara dan digunakan untuk mendeteksi keberadaan suatu objek tertentu di depannya. Frekuensi kerjanya berada pada daerah gelombang suara di atas 40 KHz hingga 400 KHz. Sensor ultrasonik terdiri dari dua unit, yaitu unit pemancar dan unit penerima [8][9].



Gambar 2.3 Sensor Ultrasonik HC-SR04

### 2.4 Arduino

Arduino adalah kit elektronik atau papan rangkaian elektronik *open source* yang di dalamnya terdapat komponen utama yaitu sebuah *chip* mikrokontroler dengan jenis AVR dari perusahaan Atmel. Mikrokontroler itu sendiri adalah *chip* atau (*integrated circuit*) IC yang bisa diprogram menggunakan komputer. Tujuan menanamkan program pada mikrokontroler adalah agar rangkaian elektronik dapat membaca input, memproses *input* tersebut dan kemudian menghasilkan *output* sesuai yang diinginkan. Jadi mikrokontroler bertugas sebagai 'otak' yang mengendalikan input, proses dan output sebuah rangkaian elektronik. Mikrokontroler ada pada perangkat elektronik di sekeliling kita dan juga dipakai untuk keperluan mengendalikan robot, baik robot mainan maupun robot industri misalnya handphone, MP3 *player*, DVD, televisi, AC, dll. Bahasa pemrograman Arduino adalah bahasa C. Tetapi bahasa ini sudah dipermudah menggunakan fungsi-fungsi yang sederhana sehingga pemula pun bisa mempelajarinya dengan cukup mudah [1].



Gambar 2.4 ArduinoUno R3

### 2.5 NodeMCU

NodeMCU adalah sebuah *board* elektronik yang berbasis *chip* ESP8266 dengan kemampuan menjalankan fungsi mikrokontroler dan bisa dikoneksikan dengan internet. Terdapat beberapa pin I/O sehingga dapat dikembangkan menjadi sebuah aplikasi monitoring maupun controlling pada proyek IoT [10].



Gambar 2.5 NodeMCU ESP8266 Versi 1.0

## 2.6 Quality of Service (QoS)

Pengujian sistem parkir akan dilakukan dengan mengukur *Quality of Service* (QoS). Parameter yang digunakan dalam pengujian sistem ini adalah sebagai berikut:

### 2.6.1 Delay

Merupakan waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik, *congesti* atau juga waktu proses yang lama [11].

### 2.6.2 Throughput

Yaitu kecepatan (rate) transfer data efektif yang diukur dalam bit per second (bps). *Throughput* adalah jumlah total kedatangan paket yang sukses yang diamati pada tujuan selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut [11].

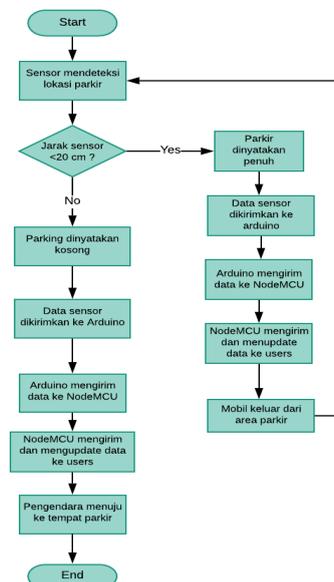
### 2.6.3 Packet Loss

*Packet loss* yaitu jumlah total paket yang hilang karena *collision* dan *congestion* pada jaringan.

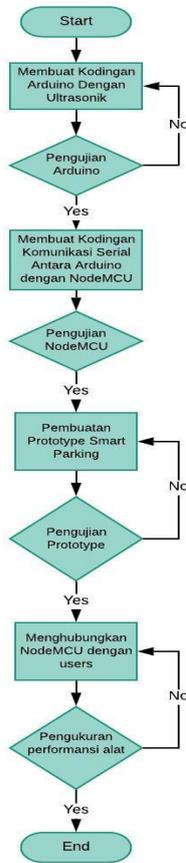
## 3. Perancangan Sistem

### 3.1 Desain Sistem

Pada Tugas Akhir ini, akan dirancang *smart parking* sistem untuk memudahkan pengguna parkir dalam mendapatkan parkir yang kosong serta *maps* untuk menuju ke parkir kosong tersebut. Pada Tugas Akhir ini menggunakan 2 mikrokontroler. Mikrokontroler digunakan untuk desain sistem portal *smart parking* dan desain sistem *smart parking*. Desain sistem portal *smart parking* dibuat dengan menggunakan motor servo dan ultrasonik HC-SR04 sebagai sensornya. Portal digunakan sebagai pintu masuk dan pintu keluar pengguna parkir. Keduanya diprogram menggunakan Arduino IDE. Desain sistem *smart parking* bagian end user menggunakan 5 sensor ultrasonik sebagai slot parkir. Sensor ultrasonik akan dihubungkan ke Arduino, dan Arduino berhasil membaca data jarak sensor ultrasonik. Arduino mengirim data tersebut ke NodeMCU. Selanjutnya NodeMCU mengirim data yang didapat ke *users*. Sistem yang akan dibangun adalah sebuah fitur dari *smart parking system* yaitu fitur penentuan slot parkir yang disarankan serta rute dan *maps* yang dibutuhkan. Dengan menggunakan sistem ini, *user* atau pengemudi akan mendapatkan lokasi pasti untuk parkir kendaraan dan tidak kesulitan mencari rute parkir. Sistem mendapatkan informasi dari sensor ultrasonik yang dipasang di area parkir. Sistem diprogram melalui Arduino dan data yang diperoleh disimpan di database yang telah dibuat. Sistem terdiri dari 2 bagian, yaitu sistem *end user* dan sistem *users*. Sistem *end user* dikerjakan oleh Karyn Azzahra Palar dan sistem *users* dikerjakan oleh Resi Yulia Putri. Pembatasan *end user* dan *users* pada sistem ini yaitu bagian *end user* akan fokus kepada semua yang berhubungan dengan alat-alat yang dibutuhkan oleh sistem. Hal ini dimulai dengan pengendara masuk dan melakukan scan QR Code, misal pengendara mendapatkan informasi slot parkir yang kosong ada di titik A dan ada *maps* yang menunjukkan ke arah parkir A tersebut, pengendara akan memarkirkan mobil di tempat yang kosong tersebut, maka sensor akan membaca ada mobil yang parkir. Hal tersebut menandakan bahwa parkir A sudah penuh dan data itu akan dikirimkan dan disimpan oleh Arduino. Arduino akan mengirimkan data tersebut kepada NodeMCU, lalu NodeMCU akan mengirimkan dan memperbarui data yang akan disimpan pada *database*. Di saat kendaraan akan keluar dari tempat parkir, sensor akan mendeteksi tidak ada kendaraan di parkir A. Selanjutnya pengendara melakukan *scan* sehingga menandakan kembali bahwa parkir A sudah kosong. Data itu diperbarui dan dikirimkan ke NodeMCU melalui Arduino agar *user* mendapatkan data terbaru.



Gambar 3.1 Flowchart Sistem



Gambar 3.2 *Flowchart* Perencanaan

**3.2 Desain hardware dan software sistem**

Pada tugas akhir ini, pendataan kebutuhan hardware dapat dilihat pada Tabel 3.1.

Tabel 3.1 Kebutuhan *Hardware*

Nama komponen	Jumlah
Sensor ultrasonik	7
Arduino Uno	2
Kabel Jumper	< 15
Laptop	2
Smartphone android	2
Kabel USB Arduino	2
Nodemcu	1
Motor Servo	2
Project Board	1

Pada tugas akhir ini, *software* yang digunakan yaitu Arduino IDE 1.8.13. Dengan menggunakan Arduino Uno dan sensor ultrasonik, maka diperlukan software agar hardware dapat bekerja. Arduino IDE dapat mengontrol cara kerja dari sensor ultrasonik tersebut, seperti menyambungkan antara sensor dan database, membaca masukan dari objek. Semua terprogram melalui Arduino IDE. Arduino IDE menggunakan bahasa C dalam pemogramannya.

#### 4. Skenario dan Hasil Pengujian



Gambar 4.1 Desain *Prototype* Alat *Smart Parking*

Pada pelaksanaannya, pengujian dilakukan menggunakan laptop dengan spesifikasi *hardware* seperti pada tabel 4.1

Tabel 4.1 Spesifikasi Laptop yang Digunakan

<b>Jenis</b>	Asus vivobook ultra A412
<b>Prosesor</b>	Intel core i7-8565U
<b>OS</b>	Windows 10
<b>RAM</b>	4 GB
<b>Dual Storage</b>	256 GB SATA SSD + 1 TB HDD

Sementara itu, spesifikasi *software* yang digunakan terdapat pada tabel 4.2.

Tabel 4.2 Spesifikasi *Software* yang Digunakan

<b>Bahasa Pemrograman</b>	Arduino IDE 1.8.13
<b>Software Testing Data</b>	Wireshark

##### 4.1 Pengujian *Hardware*

Pengujian ini dilakukan untuk mengetahui fungsi-fungsi dari perangkat yang digunakan dapat bekerja dengan baik, seperti Arduino Uno, sensor ultrasonik, motor servo, dan NodeMCU. Hasil dari pengujian tersebut dapat dilihat pada Tabel 4.1.

Tabel 4.1 Hasil Pengujian *Hardware*

Hardware	Skenario	Indikator	Status
Arduino	Menghubungkan ke laptop dan <i>running</i> program	Lampu indikator Arduino menyala dan dapat menjalankan program yang di <i>running</i>	Berhasil
Sensor Ultrasonik	Memasukan kendaraan roda empat ke area parkir slot 1	Sensor dapat mendeteksi adanya kendaraan	Berhasil

	Memasukan kendaraan roda empat ke area parkir slot 2	Sensor dapat mendeteksi adanya kendaraan	Berhasil
	Memasukan kendaraan roda empat ke area parkir slot 3	Sensor dapat mendeteksi adanya kendaraan	Berhasil
	Memasukan kendaraan roda empat ke area parkir slot 4	Sensor dapat mendeteksi adanya kendaraan	Berhasil
	Memasukan kendaraan roda empat ke area parkir slot 5	Sensor dapat mendeteksi adanya kendaraan	Berhasil
	Kendaraan roda empat keluar dari area parkir slot 1	Sensor dapat mendeteksi tidak adanya kendaraan	Berhasil
	Kendaraan roda empat keluar dari area parkir slot 2	Sensor dapat mendeteksi tidak adanya kendaraan	Berhasil
	Kendaraan roda empat keluar dari area parkir slot 3	Sensor dapat mendeteksi tidak adanya kendaraan	Berhasil
	Kendaraan roda empat keluar dari area parkir slot 4	Sensor dapat mendeteksi tidak adanya kendaraan	Berhasil
	Kendaraan roda empat keluar dari area parkir slot 5	Sensor dapat mendeteksi tidak adanya kendaraan	Berhasil
Motor Servo	Pintu Portal Terbuka	Poros motor servo berputar ke sudut 90°	Berhasil
	Pintu Portal Tertutup	Poros motor servo berputar ke sudut 0°	Berhasil
NodeMCU	Data Arduino dikirim ke nodemcu	Jarak sensor di Arduino sama dengan jarak yang diterima NodeMCU	Berhasil
	Data NodeMCU dikirim ke thingspeak	Jarak sensor di	Berhasil

		NodeMCU sama dengan jarak yang diterima di thingspeak	
--	--	---	--

Untuk membuktikan *hardware* yang digunakan dapat berfungsi dengan baik dan dapat terhubung antara satu dengan yang lainnya, dilakukan pengujian dengan cara menghubungkan semua perangkat pada laptop, dilanjutkan dengan melakukan *upload program* yang telah dibuat. Pada Tabel 4.1 dapat dilihat bahwa semua perangkat yang digunakan pada Tugas Akhir ini dapat berfungsi sesuai dengan yang diharapkan karena semua perangkat memiliki status “berhasil”.

**4.2 Pengujian Jarak Sensor Ultrasonik**

Pengujian ini dilakukan untuk mengetahui sensor ultrasonik dapat membaca jarak dengan baik dan benar ketika ada benda yang didekatkan ke sensor. Pengujian ini dilakukan pada kelima sensor ultrasonik. Hasil pengujian ini dapat dilihat pada Tabel 4.2.

**Tabel 4.2 Hasil Pengujian jarak sensor ultrasonik**

Jarak (cm)	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5
4	Oke	Oke	Oke	Oke	Oke
8	Oke	Oke	Oke	Oke	Oke
12	Oke	Oke	Oke	Oke	Oke
16	Oke	Oke	Oke	Oke	Oke
20	Oke	Oke	Oke	Oke	Oke
24	Oke	Oke	Oke	Oke	Oke
28	Oke	Oke	Oke	Oke	Oke
32	Oke	Oke	Oke	Oke	Oke
36	Oke	Oke	Oke	Oke	Oke
40	Oke	Oke	Oke	Oke	Oke
44	Oke	Oke	Oke	Oke	Oke
48	Oke	Oke	Oke	Oke	Oke
52	Oke	Oke	Oke	Oke	Oke
56	Oke	Oke	Oke	Oke	Oke
60	Oke	Oke	Oke	Oke	Oke
64	Oke	Oke	Oke	Oke	Oke
68	Oke	Oke	Oke	Oke	Oke
72	Oke	Oke	Oke	Oke	Oke
76	Oke	Oke	Oke	Oke	Oke
80	Oke	Oke	Oke	Oke	Oke
84	Oke	Oke	Oke	Oke	Oke
88	Oke	Oke	Oke	Oke	Oke
92	Oke	Oke	Oke	Oke	Oke

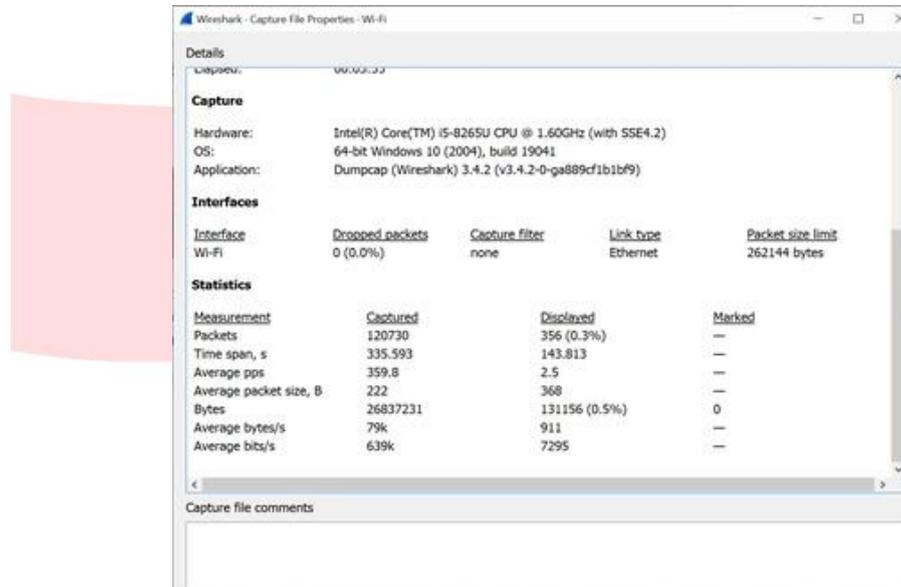
96	Oke	Oke	Oke	Oke	Oke
100	Oke	Oke	Oke	Oke	Oke
104	Oke	Oke	Oke	Oke	Oke
108	Oke	Oke	Oke	Oke	Oke
112	Oke	Oke	Oke	Oke	Oke
116	Oke	Oke	Oke	Oke	Oke
120	Oke	Oke	Oke	Oke	Oke
124	Oke	Oke	Oke	Oke	Oke
128	Oke	Oke	Oke	Oke	Oke
132	Oke	Oke	Oke	Oke	Oke
136	Oke	Oke	Oke	Oke	Oke
140	Oke	Oke	Oke	Oke	Oke
144	Oke	Oke	Oke	Oke	Oke
148	Oke	Oke	Oke	Oke	Oke
152	Oke	Oke	Oke	Oke	Oke
156	Oke	Oke	Oke	Oke	Oke
160	Oke	Oke	Oke	Oke	Oke
164	Oke	Oke	Oke	Oke	Oke
168	Oke	Oke	Oke	Oke	Oke
172	Oke	Oke	Oke	Oke	Oke
176	Oke	Oke	Oke	Oke	Oke
180	Oke	Oke	Oke	Oke	Oke
184	Oke	Oke	Oke	Oke	Oke
188	Oke	Oke	Oke	Oke	Oke
192	Oke	Oke	Oke	Oke	Oke
196	Oke	Oke	Oke	Oke	Oke
200	Oke	Oke	Oke	Oke	Oke
204	Tidak Oke				
208	Tidak Oke				
212	Tidak Oke				
216	Tidak Oke				

Pengujian dilakukan untuk mengetahui seberapa jauh jarak sensor dapat mendeteksi objek. Pada tabel 4.2 dapat dilihat bahwa pengujian dilakukan pada kelipatan 4 cm. sensor ultrasonik dapat membaca jarak dengan benar pada jarak 4-200 cm. Ketika mencapai jarak >200 cm, sensor tidak dapat membaca jarak dengan baik. Hal ini terjadi karena pemrograman yang dilakukan telah diatur untuk maximum range nya adalah 200 cm, jadi sensor ultrasonik dapat membaca hingga jarak 200 cm.

### 4.3 Pengujian QoS

Pengujian ini dilakukan untuk mengetahui kualitas jaringan saat melakukan pengiriman dan penerimaan data dalam sebuah sistem. Parameter yang diuji adalah *throughput*, *delay*, dan *packet loss*.

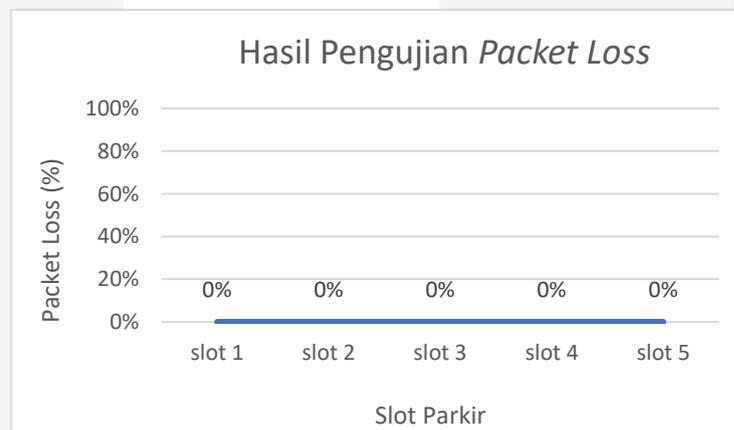
#### 4.3.1 Pengujian Throughput



Gambar 4.2 Hasil Pengujian *Throughput*

*Throughput* yang diamati dalam pengujian ini adalah *Throughput* antara NodeMCU dengan *Thingspeak*. Menggunakan aplikasi *wireshark* didapatkan hasil *Throughput* adalah 639k bps. Menurut standarisasi TIPHON dengan hasil yang didapatkan, maka pengiriman jumlah data per satuan waktu dari NodeMCU ke *Thingspeak* termasuk *Throughput* dalam kategori sangat bagus dengan indeks 4.

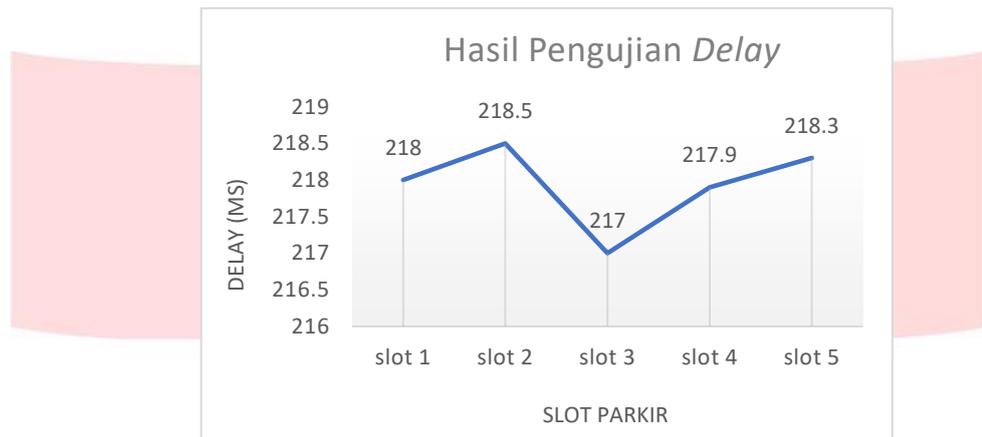
#### 4.3.2 Pengujian Packet Loss



Gambar 4.3 Hasil Pengujian *Packet Loss*

Pengujian ini dilakukan untuk mengetahui jumlah total paket yang hilang dapat terjadi karena *collision* dan *congestion* pada jaringan. Pengujian dilakukan menggunakan aplikasi *wireshark*. Hasil dari pengujian *packet loss* pada NodeMCU ke *Thingspeak* mendapatkan hasil yang sangat baik karena nilai *packet loss*nya yaitu 0%. Artinya tidak ada jumlah paket yang hilang dikarenakan *collision* atau *congestion*.

### 4.3.3 Pengujian Delay



Gambar 4.4 Hasil Pengujian Delay

Hasil dari pengujian *delay* NodeMCU dengan sensor dan Thingspeak di dapatkan total delay 218 ms dan rata-rata delay 0.07 ms. Menurut standarisasi TIPHON. Dengan hasil yang di dapat maka waktu pengiriman data dari NodeMCU menuju *database Thingspeak* adalah bagus.

## 5. Kesimpulan

Berdasarkan hasil simulasi dan analisis penelitian yang telah dilakukan sebelumnya, dapat ditarik kesimpulan sebagai berikut:

1. *Prototype smart parking* yang telah dirancang menggunakan *wireless sensor network* untuk *monitoring system* yang memberikan informasi parkir tersedia, tidak tersedia dan slot parkir yang disarankan serta rute dan *maps* yang dibutuhkan dapat berfungsi dengan baik sesuai dengan yang diharapkan.
2. Pada pengujian jarak jangkauan maksimum dengan menghubungkan Arduino dan sensor ultrasonik didapatkan jarak maksimum yang dapat terbaca oleh sensor ultrasonik yaitu 200 cm.
3. Hasil *Throughput* adalah 639k bps. Menurut standarisasi TIPHON, *Throughput* termasuk bagus dengan indeks 4. Dengan hasil yang di dapatkan maka pengiriman jumlah data per satuan waktu dari NodeMCU ke Thingspeak adalah bagus.
4. Hasil dari pengujian *packet loss* pada NodeMCU ke *Thingspeak* mendapatkan hasil yang sangat baik karena nilai *packet loss*nya yaitu 0%. Artinya tidak ada jumlah paket yang hilang dikarenakan *collision* atau *congestion*.
5. Hasil dari pengujian *delay* NodeMCU dengan sensor dan *Thingspeak* di dapatkan total delay 218 ms dan rata-rata delay 0.07 ms. Menurut standarisasi TIPHON. Dengan hasil yang di dapat maka waktu pengiriman data dari NodeMCU menuju *database Thingspeak* adalah bagus.

**Referensi :**

- [1] W. Hilmy, A. Yoana, and P. Damanik, "Aplikasi Mobile Smart Parking pada Basement Bertingkat Menggunakan Sensor Ketinggian Smart Parking Mobile Application for Storey Basement using Height-sensor."
- [2] D. Susandi, W. Nugraha, and S. F. Rodiyansyah, "Perancangan Smart Parking System pada Prototype Smart Office Berbasis Internet of Things," *Tek. Ind. dan Tek. Inform. Univ. Majalengka*, no. November, pp. 1–2, 2017.
- [3] P. V. Ertyan, P. Pangaribuan, and A. S. Wibowo, "Sistem monitoring dan mengontrol aquarium dalam pemeliharaan ikan hias jarak jauh," vol. 6, no. 2, pp. 3102–3108, 2019.
- [4] GSMA Association, "Understanding the Internet of Things (IoT)," *Gsma Connect. Living*, no. July, p. 15, 2014.
- [5] E. S. Wahyuningtyas, I. R. Munadi, S. S. Si, S. T. Telekomunikasi, F. T. Elektro, and U. Telkom, "Aplikasi Smart Parking Berbasis Android Menggunakan Sensor Radio Frequency Identification ( Rfid ) Di Universitas Telkom Application of Smart Parking By Android Using Radio Frequency Indentification ( Rfid ) in Telkom University," vol. 6, no. 2, pp. 3620–3627, 2019.
- [6] P. Hendriantoro, E. Ariyanto, A. G. P. S, F. Informatika, U. Telkom, and W. Network, "PROTOTIPE SMART GUIDES PARKING SYSTEM MENGGUNAKAN METODE EVENT-BASED BERBASIS WIRELESS NETWORK PROTOTYPE SMART GUIDES PARKING SYSTEM USING EVENT-BASED METHOD," pp. 1–9.
- [7] A. Mohammad, "Smart Parking System," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 6, no. 5, pp. 81–83, 2018, doi: 10.22214/ijraset.2018.5011.
- [8] V. N. April, "Jurnal Teknik Mesin Volume 21 - No.1 - April 2006," vol. 21, no. 1, 2006.
- [9] I. Kurniawan *et al.*, "Menggunakan Raspberry Pi Melalui Telegram Design and Implementation Smart Parking Using," vol. 6, no. 2, pp. 4463–4470, 2019.
- [10] N. Hidayati, L. Dewi, M. F. Rohmah, and S. Zahara, "Prototype Smart Home Dengan Modul NodeMCU ESP8266 Berbasis Internet of Things (IoT)," *Tek. Inform. Univ. Islam Majapahit*, pp. 1–9, 2018.
- [11] K. WAHID, "Desain dan Implementasi Sistem Reservasi pada Smart Parking," vol. 6, no. 3, pp. 10186–10194, 2019.