

IMPLEMENTASI SMART PARKING DENGAN MAPS PADA USER DI PUSAT PERBELANJAAN MENGGUNAKAN METODE *QR*CODE.

(Implementation of Smart Parking with Maps on Users

in Buildings Using the QRcode Method)

Resi Yulia Putri¹, Rendy Munadi², Nyoman Bogi Aditya Karna³

^{1,2,3}Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

¹Resiyuliaputri@student.telkomuniversity.ac.id, ²rendymunadi@telkomuniversity.ac.id
³aditya@telkomuniversity.ac.id

Abstrak

Smart Parking merupakan bagian dari pengembangan *smart city* yang dilakukan untuk meningkatkan kebutuhan fasilitas akan penggunaan tempat parkir. Penelitian ini menggunakan metode *QRcode* yang berisi informasi nomor slot parkir maps lokasi parkir. Dengan keterangan jika indikator lampu pada data berwarna redup menandakan parkir terisi dan untuk berwarna terang menandakan parkir kosong. Perancangan sistem ini membutuhkan fungsionalitas data dari sensor yang digunakan dikirim ke NodeMCUESP8266, dan diteruskan ke *ThingSpeak Platform* sebagai database yang digunakan oleh user dengan hasil delay pada pengujian QoS rata-rata 218,4621 ms. Menurut standarisasi TIPHON delay termasuk kategori Bagus dengan indeks 3. Dan untuk throughput rata rata nilai 639k bps. Berdasarkan standarisasi TIPHON Throughput termasuk ke dalam kategori sangat bagus dengan indeks 4. Dari hasil tersebut proses pengiriman jumlah data per satuan waktu NodeMCU menuju *ThingSpeak Platform* termasuk cepat.

Kata kunci : *Smart Parking, ThingSpeak Platfor, maps, QRcode.*

Abstract

Smart Parking is part of smart city development that is *carried* out to increase the need for facilities for the use of parking spaces. This study used *QRcode method* that contains parking slot number information maps parking location. With a description if the light indicator on the data is dimmed indicates the parking is filled and for light color indicates empty parking. Designing this system requires the functionality of data from the sensors used to be sent to NodeMCUESP8266, and forwarded to *ThingSpeak Platform* as a database used by users with delay results on QoS tests averaging 218.4621 ms. According to TIPHON delay standardization belongs to the Good category with an index of 3. And for the average throughput value of 639k bps. By tiphon throughput standards fall into a very good category with an index of 4. From these results the process of sending the amount of data per unit of Time NodeMCU to *ThingSpeak Platform* is included quickly.

Keywords: *Smart Parking, ThingSpeak Platfor, maps, QRcode.*

1. Pendahuluan

Teknologi modern terus berkembang sehingga manusia terus berusaha meningkatkan kualitas dan efektivitas dalam kehidupan. *Internet of Things* (IoT) menjadi salah satu bagian dari teknologi modern untuk memenuhi kebutuhan tersebut [1]. Setiap manusia membutuhkan efisiensi waktu, jaminan keamanan, dan kemudahan untuk dapat memenuhi tuntutan kebutuhan hidup. Aplikasi *smart city* dalam beberapa tahun terakhir telah menjadi populer dengan tujuan meningkatkan kualitas kehidupan orang-orang dikota [2]. Perkembangan konsep IoT telah banyak memungkinkan kemungkinan yang terjadi untuk aplikasi *smart city*.

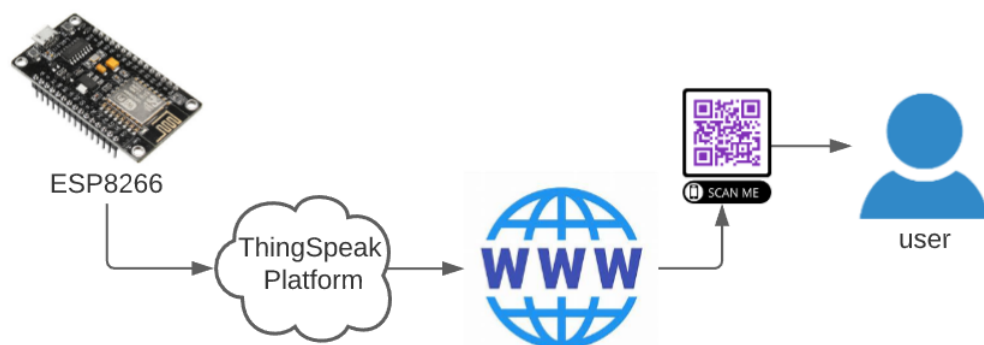
Mobil merupakan salah satu pendukung yang efisiensi dan efektivitas dalam menjalankan tuntutan tersebut, bahkan dalam sebuah keluarga memiliki lebih dari satu mobil. Kebebasan beregerak bagi seseorang yang memiliki mobil yaitu kebutuhan lahan parkir. Proses parkir mobil merupakan hal

yang umum terjadi seperti di pusat perbelanjaan yang membutuhkan keleluasaan tempat parkir, kenyamanan, dan keamanan dalam sistem parkir. Keterbatasan parkir hampir terjadi di setiap kota besar di dunia menyebabkan kemacetan lalu lintas. Sistem *smart parking* memperhatikan tentang keamanan dan kenyamanan, kondisi seperti ini dapat menyebabkan kegiatan parkir menjadi tidak efektif[3].

Dengan demikian sangat diperlukan untuk mengembangkan sistem *smart parking* yang akan membantu pengemudi untuk mengetahui informasi ketersediaan parkir dengan cepat dan mudah. Sistem *smart parking* ini meneruskan sistem pada NodeMCU ESP8266 ke *firebase ThingSpeak* sebagai file database untuk dapat di *update* secara *real-time* dalam sebuah website yang nanti data tersebut diperlukan oleh user berupa *QRcode* yang berisi informasi parkir yang kosong dan rute dalam bentuk *maps* menuju slot parkir yang tersedia. Untuk mengimplementasikan sistem tersebut user membutuhkan *smart phone* yang dapat mengakses *QRcode* agar dapat membaca dan menerima informasi mengenai slot parkir yang tersedia atau tidak, dan rute menuju tempat parkir yang tersedia.

2. Material dan Metodologi

A. Desain Sistem



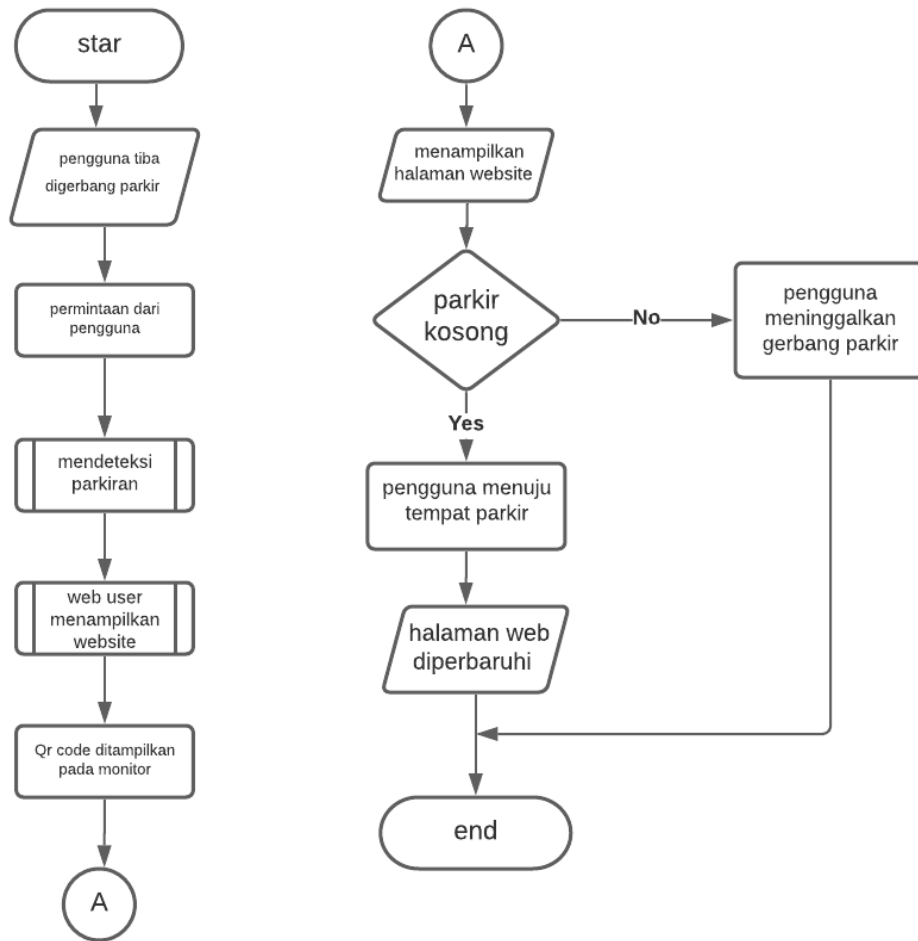
Gambar 1. Desain sistem

pada penelitian ini meliputi perancangan NodeMCU yang dihubungkan dengan ThingSpeak menggunakan bahasa komunikasi arduino IDE. Perancangan ThingSpeak sebagai *database*, dan perancangan *website* untuk menampilkan data melalui *scan QR Code* pada *smart phone* pengguna. Data diolah dalam bentuk *QR Code* berisi informasi mengenai lokasi nomor parkir yang tersedia beserta *maps* menuju tempat parkir tersebut dan *code* tersebut di-*update* pada monitor sehingga *QR Code* tersebut dapat digunakan oleh *user*. Sistem ini diprogram menggunakan Arduino IDE dan data yang diperoleh disimpan dalam *database* yang telah dibuat.

B. Flowchart Sistem

Proses kerja sistem *software* dimulai dengan menghubungkan NodeMCU ke jaringan internet. Koneksi tersebut dibutuhkan agar terhubung dengan *website* sehingga dapat mengirimkan data menuju basis data. Pengguna perlu menghubungkan *smart phone* dengan *access point* sistem parkir. Pengguna yang ingin parkir dapat mengakses *website* untuk mendapatkan informasi parkir yang tersedia dan *maps* lokasi parkir tersebut dengan melakukan *scan QR Code*. *Website* menampilkan informasi-informasi yang dapat memudahkan pengguna dalam menggunakan fasilitas *smart parking* yang telah disediakan. Setelah membuka *website* pengguna dapat melihat informasi parkir yang tersedia dan kondisi lahan parkir. Indikator pada *website* terdapat lampu berwarna merah saat parkir penuh dan berwarna terang pada saat parkir kosong atau tersedia.

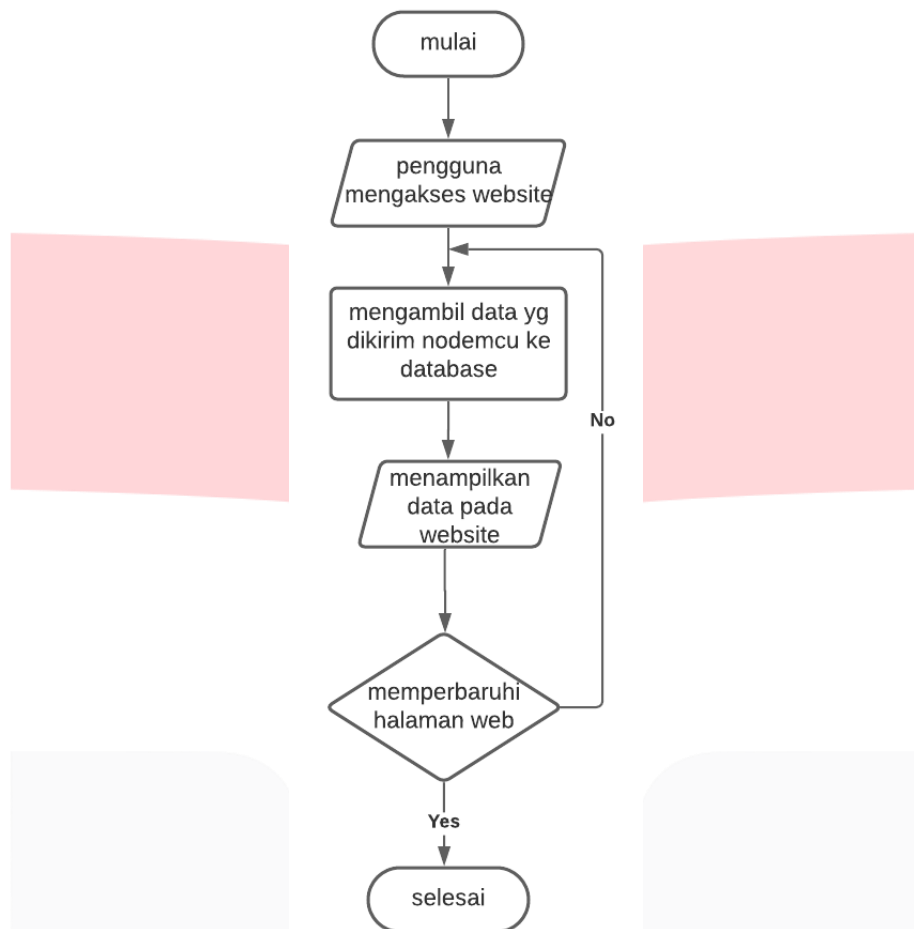
Ketika slot parkir tersedia, pengguna dapat memilih dan menempati tempat parkir yang tertera pada *maps* parkir tersebut. Pada *website* ditampilkan secara keseluruhan informasi parkir yang tersedia atau penuh. Hal tersebut dilakukan agar mempermudah pengguna memilih lokasi parkir yang terdekat dari pintu masuk yang sudah tertera pada *maps* parkir tersebut.



Gambar 2. Flowchart sistem

C. Flowchart Web User

Perancangan perangkat lunak web sesuai dengan Gambar 10.3. Proses diawali dengan adanya pengguna untuk mengakses website informasi parkir. Data yang didapatkan oleh NodeMCU dikirimkan ke *database*, lalu diproses oleh thingSpeak *platform*. *Website* ditampilkan pada browser pengguna sesuai dengan data yang ada di dalam *database*. Apabila data dalam *database* sewaktu-waktu berubah, maka *website* akan memperbarui halaman web tersebut dan menampilkan data terbaru.



Gambar 3. Flowchart web user

D. Pengujian Sistem

1. Pengujian User pada Sistem Prototipe Smart Parking

Pengujian dilakukan dengan memindai *QR Code* yang pada *website* mendapatkan informasi mengenai nomor slot parkir dan *maps* parkir. Pengujian sistem secara keseluruhan dilakukan dengan menyebarkan kuisioner kepada pengguna *smart parking*. Kuisioner berisi 5 pertanyaan. Pertanyaan-pertanyaan yang diberikan berkaitan dengan sistem yang diperlukan *user* secara keseluruhan dan mengenai pemahaan pengguna. Nilai dari data kuisioner ada 5 yaitu mulai dari 1 dengan bobot kurang hingga 5 dengan bobot sangat baik.

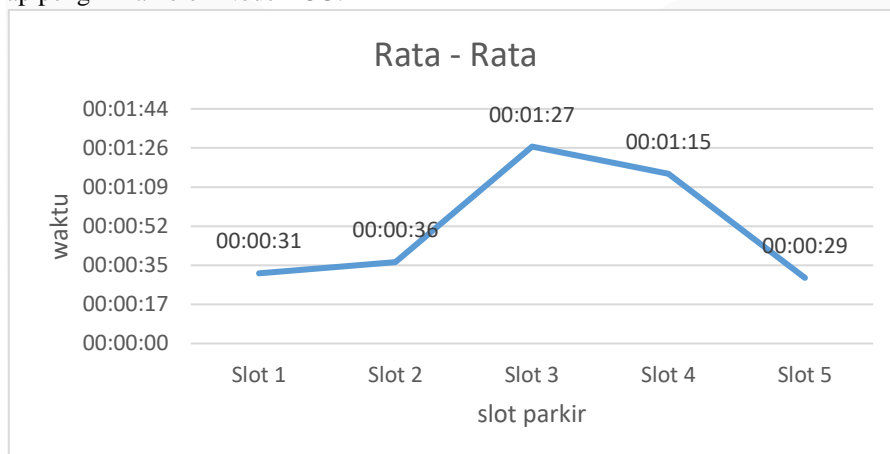
Hasil rata-rata yang didapat dari kuisioner yang diisi oleh 61 pengguna yang berbeda. Hasil perhitungan yang diperoleh menyatakan bahwa kinerja sistem sudah cukup baik sebesar 72%. Tingkat pemahaman pengguna dan kepuasan pengguna dengan sistem ini sudah cukup baik dengan nilai 64% dan 62%.



Gambar 4. Grafik hasil kuisiner

2. Pengujian Kinerja Pengujian Pengiriman Data NodeMCU

Pengujian dilakukan dengan menghubungkan data pada NodeMCU dan dilakukan pemograman untuk membaca jarak melalui *software* Arduino IDE. Pengujian sebanyak 8 kali pengiriman data oleh NodeMCU pada masing-masing slot, dilakukan dengan melihat jeda waktu setiap pengiriman oleh NodeMCU.



Gambar 5. Grafik rata-rata kinerja pengiriman data NodeMCU

Keterangan waktu yang digunakan yaitu, dengan rata-rata diperoleh pada slot satu perubahan waktu saat kendaraan mengisi slot parkir pada pukul 08 lewat 48 menit 12 detik, data terbaca pada *database* pada pukul 08 lewat 48 menit 43 detik, dengan kesimpulan lama waktu data terbaca pada thingspeak untuk slot satu selama 31 detik. Dari hasil pengujian dapat disimpulkan pada slot 3 mengalami perubahan waktu paling lama dengan rata-rata sebesar 2 menit 27 detik. Sedangkan pada slot 5 mengalami perubahan waktu paling cepat dengan rata-rata sebesar 29 detik. Hal ini dapat terjadi karena jarak dan jadwal yang diukur oleh sensor pada tiap slot berbeda.

3. Pengujian Pembacaan Database

Tabel 1. Pengujian pembacaan *database*

Keterangan	Percobaan pembacaan data	Persentase percobaan data
Slot 1	1-8	100%
Slot 2	1-8	100%

Slot 3	1-8	100%
Slot 4	1-8	100%
Slot 5	1-8	100%

Data hasil percobaan pembacaan *database* pada *website*. Percobaan dilakukan sebanyak 8 kali dengan tujuan untuk dapat memastikan kesesuaian data yang ditampilkan pada *website* oleh *database*. Setiap slot berhasil dilakukan percobaan pembacaan *database* dengan baik, dan tingkat kesesuaian data pada sistem sebesar 100%

3. Hasil

A. Tampilan Website



Gambar 6. Tampilan *website*

Bentuk tampilan *website* dengan kondisi slot parkir yang ditampilkan. Hal ini ditunjukkan dengan keterangan warna pada indikator lampu. Pada gambar dapat dilihat slot A3 merupakan bentuk tampilan *website* dengan kondisi parkir yang sebagian terisi ditandai dengan warna indikator lampu yang redup. Untuk slot A1, A2, A4, A5 menunjukkan kondisi parkir kosong, ditunjukkan dengan keterangan indikator lampu berwarna terang.

B. Hasil Pengujian QoS

Penulis melakukan pengujian untuk mencari nilai pada setiap parameter QoS, berikut hasil uji coba yang penulis lakukan.

1. Throughput

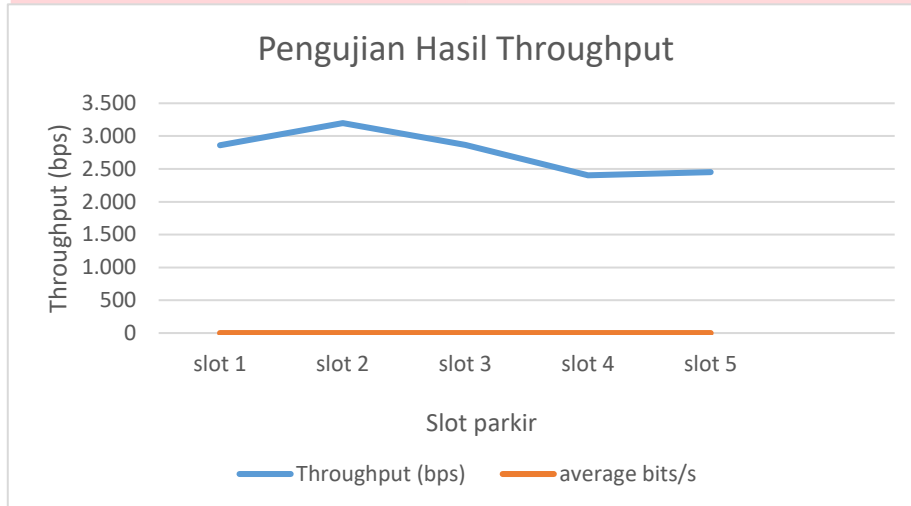
Throughput adalah kecepatan transfer data efektif atau didefinisikan sebagai ukuran keberhasilan secara aktual di dalam pengiriman paket data pada jaringan komputer oleh suatu perangkat [4].

$$\text{Throughput} : \frac{\text{Packed received (kb)}}{\text{Time transmitted (s)}}$$

Gambar 7. Rumus throughput

Tabel 2. Standar throughput TIPHON

Throughput (bps)	Indeks	Kategori
>100	4	Sangat Bagus
75	3	Bagus
50	2	Sedang
<25	1	Buruk



Gambar 8. Grafik hasil uji coba mencari nilai throughput

Hasil pengujian *throughput* ThingSpeak yang digunakan diperoleh rata-rata nilai dari masing-masing slot sebesar 639k bps/s. Berdasarkan standarisasi TIPHON *throughput* termasuk ke dalam kategori sangat bagus dengan indeks 4. Dari hasil tersebut, proses pengiriman jumlah data per satuan waktu NodeMCU menuju ThingSpeak *platform* termasuk cepat.

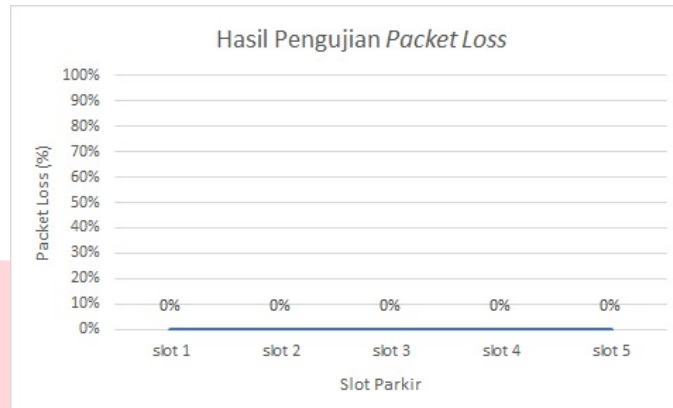
2. Packet Loss

Packet Loss pada kegagalan suatu transmisi dapat disebabkan oleh beberapa faktor seperti *buffer*, *Overload Traffic*, *congestion* dan *overflow*.

$$Packet Loss = \frac{(\text{Paket data dikirim} - \text{paket data diterima})}{\text{Paket data dikirim}} \times 100$$

Tabel 3. Standar *packet loss* TIPHON

Packet Loss (%)	Indeks	Kategori degradasi
0	4	Sangat Bagus
3	3	Bagus
15	2	Sedang
25	1	Buruk



Gambar 9. Grafik hasil pengujian *packet loss*

Hasil dari pengujian *packet loss* menunjukkan hasil yang sangat baik, dari jumlah paket data yang terkirim sehingga diperoleh *packet loss* secara keseluruhan maupun rata-ratanya 0% dan berdasarkan standarisasi TIPHON merupakan kategori yang sangat bagus.

3. Delay

Delay adalah waktu yang dibutuhkan data untuk menempuh jarak dari gerbang masuk parkir ke tempat parkir yang tersedia. *Delay* dapat dipengaruhi oleh jarak, media, *congesti* atau juga proses waktu yang lama [4].

$$\text{Delay} = \text{Waktu Paket Diterima} - \text{Waktu Paket Dikirim}$$

Tabel 4. Standar *delay* TIPHON

Besar Delay (ms)	Indeks	Kategori Latensi
<150	4	Sangat Bagus
120 s/d 300	3	Bagus
300 s/d 450	2	Sedang
>450	1	Buruk

Berikut hasil uji coba mencari nilai delay:



Gambar 10. Grafik hasil pengujian *delay*

4. Kesimpulan dan Saran

4.1 Kesimpulan

1. Menurut hasil pengujian situs web dan komponen, sistem bekerja dengan baik.
2. Dalam memantau hasil pengujian, NodeMCU mengirimkan data secara *real-time*, sehingga ditampilkan di situs web dan disimpan di *database* ThingSpeak
3. *QR Code* memberikan informasi yang terbaru yang dihubungkan dengan web, dengan parameter pengujian pada kuisisioner yang disebar ke pengguna sehingga Tingkat pemahaman pengguna dan kepuasan pengguna dengan sistem ini sudah cukup baik dengan nilai 64% dan 62%.
4. *Maps* telah sesuai dengan contoh peta parkir yang ada pada BIP, dengan melakukan parameter pengujian menyebarkan kuesioner ke pengguna sehingga diperoleh kepuasan pengguna dengan adanya maps sebesar 85 %
5. Pada pengujian *throughput* antara NodeMCU dengan *thingSpeak* pada saat pengiriman data dilakukan untuk 5 sensor yang digunakan diperoleh nilai sebesar 639k bps dengan indeks 4, berdasarkan standarisasi TIPHON termasuk kategori sangat bagus.
6. Pada pengujian *delay* NodeMCU dengan sensor ultrasonik menuju firebase thingspeak diperoleh rata-rata nilai 218,4621 ms.
7. Pengujian *packet loss* pada sistem yang telah dirancang sangat bagus karena tidak terdapat paket yang hilang pada saat pengiriman dan penerimaan data sehingga hasil keseluruhan paket loss 0%.

4.2 Saran

1. *Maps* bisa dikembangkan dengan menggunakan google maps untuk lebih mempermudah menemukan lokasi parkir dengan adanya navigasi pada google *maps*
2. Membuat tampilan web lebih menarik.
3. Dapat diimplementasikan pada lahan parkir berskala besar dan luas.

Referensi:

- [1] 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.
- [2] I. Aydin, M. Karakose, and E. Karakose, "A navigation and reservation based smart parking platform using genetic optimization for smart cities," *ICSG 2017 - 5th Int. Istanbul Smart Grids Cities Congr. Fair*, pp. 120–124, 2017, doi: 10.1109/SGCF.2017.7947615.
- [3] 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."
- [4] U. Diffserv, A. Mulyana, T. A. Riza, F. I. Terapan, and T. Play, "ANALISIS PERFORMANSI QUALITY OF SERVICE (QOS) DENGAN METODE DIFFSERV DAN INTSERV."
- [5] K. Wahid, N. B. Karna, and V. Susanti, "DESAIN DAN IMPLEMENTASI SISTEM RESERVASI PADA SMART PARKING DESIGN AND IMPLEMENTATION RESERVATION SYSTEM IN SMART PARKING," vol. 6, no. 3, pp. 10186–10194, 2019.