

PERANCANGAN DAN IMPLEMENTASI PARKIR PINTAR MENGUNAKAN RASPBERRY PI MELALUI TELEGRAM

DESIGN AND IMPLEMENTATION SMART PARKING USING RASPBERRY PI VIA TELEGRAM

Irwanda Kurniawan¹, Dr. Rendy Munadi, Ir., M.T.², Nyoman Bogi Aditya Karna, S.T,
MSEE³

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

¹irwanda@student.telkomuniversity.ac.id, ²rendymunadi@telkomuniversity.ac.id,

³nyoman.bogi@gmail.com

Abstrak

Banyaknya kendaraan roda empat di Indonesia menjadi pokok permasalahan baru yang harus dihadapi di masa saat ini terbukti dengan data terakhir di tahun 2017 mencapai 138.556.669 kendaraan. Peningkatan kendaraan ini tiap tahunnya sebesar 12% pertahun. Penelitian ini membuat sistem informasi *smart parking* untuk pengguna khusus roda empat menggunakan mikrokontroler raspberry pi 3 model B yang terhubung ke aplikasi telegram *messenger* dengan memberikan informasi mengenai lahan parkir kepada pengguna android ataupun IOS. Perancangan sistem dilakukan meliputi proses penghubungan antara sensor ultrasonik HC-SR04, sensor *Light Dependent Resistor*(LDR), *Resistor*, *Light Emitting Diode*(LED) dengan mikrokontroler raspberry pi 3 model B, lalu raspberry pi 3 model B akan menggunakan MQTT *mosquitto* sebagai protocol untuk berkomunikasi dengan *Virtual Private Server* (VPS). Pada hasil perancangan sistem prototipe *smart parking* ini semua sensor dapat terkoneksi dan memberikan informasi ke pengguna melalui aplikasi telegram dengan akurasi informasi lahan parkir yang akurat. Dalam penelitian ini melakukan 5 percobaan. Dari hasil semua pengujian parameter *Quality of Service*(QoS) yang meliputi *delay*, *throughput*, *jitter*, dan *packet loss* dan dibandingkan dengan standar TIPHON bahwa hasilnya telah memenuhi standar. Sehingga usulan perancangan sistem prototipe ini telah dapat di implementasikan.

Kata Kunci: IoT, Raspberry pi 3 model B, *Smart Parking*,Telegram

Abstract

Many of four-wheeled vehicles in Indonesia has become a new problem that must be faced in the present, as evidenced by the latest data in 2017 reaching 138,556,669 vehicles. Increase in vehicles every year is 12%. This reasearch makes the smart parking information system for users of four wheels using a raspberry pi 3 model B microcontroller that is connected to the telegram messenger application by providing information about parking area to users of android or IOS. Design system carried out including the connecting process between HC-SR04 ultrasonic sensor, Light Dependent Resistor (LDR) sensor, Resistor, Light Emitting Diode (LED) with raspberry pi 3 model B microcontroller, raspberry pi 3 model B will use MQTT as a protocol for communicating with a Virtual Private Server (VPS). In the results of the design of the smart parking prototype system make all sensors can be connected and provide information to the user through a telegram application with accurate information from parking area . In this study conducted 5 experiments. From the results of all tests of Quality of Service(QoS) parameters which include delay, throughput, jitter, and packet loss and compared with the TIPHON standard and the results meet the standards. Therefore the proposed design of this prototype system can be implemented.

Keywords : IoT, Raspberry pi 3 model B, *Smart Parking*,Telegram

1. Pendahuluan

Pada era metropolitan ini tingkat konsumtif kehidupan orang-orang berlangsung sangat tinggi dengan efisiensi kerja yang maksimal sehingga mengakibatkan kebutuhan sekunder pun beralih fungsi menjadi kebutuhan primer seperti halnya kendaraan roda empat yang telah menjadi kebutuhan primer. Sehingga adanya peningkatan kendaraan roda empat di Indonesia, hal ini menjadi pokok permasalahan baru yang harus dihadapi di masa saat ini terbukti dengan data terakhir di tahun 2017 mencapai 138.556.669 kendaraan bermotor seperti mobil pribadi, bus, mobil

barang dan juga kendaraan sepeda motor, peningkatan kendaraan ini pun cukup tinggi tiap tahunnya yakni sebesar 12% per tahunnya[1].

Dari penelitian yang dilakukan terdapat beberapa tujuan diantaranya implementasi *smart parking* menggunakan raspberry pi 3 model B, membuat prototipe *smart parking* yang berguna untuk memberikan informasi tentang ketersediaan lahan parkir yang lebih spesifik, pengembangan *smart parking* dari lahan parkir konvensional agar dapat bekerja lebih efisien, menghasilkan sistem informasi yang efisien secara *realtime* menggunakan aplikasi telegram, memanfaatkan sensor Ultrasonik HC-SR04, sensor *Light Dependent Resistor*(LDR), *Light Emitting Diode*(LED) dan *Keypad* Matrik 4X4 yang berguna sebagai pendukung *smart parking*, serta membuat pengguna parkir lebih efisien dalam mencari lahan parkir yang tersedia.

2. Tinjauan Pustaka

2.1 Konsep Dasar *Internet of Things* (IoT)

Internet of Things (IoT) diperkenalkan oleh seorang ilmuwan Kevin Ashton pada tahun 1999. Walaupun diperkenalkan pada tahun 1999 belum ada terdapat sebuah kesepakatan global mengenai IoT. Namun pada umumnya konsep IoT merupakan sesuatu hal yang mengacu pada sebuah kemampuan yang dihubungkan melalui objek cerdas yang memungkinkan untuk dapat berinteraksi dengan objek lain, lingkungan maupun menggunakan peralatan teknologi yang cerdas melalui jaringan internet[2].

2.2 Raspberry pi

Raspberry pi merupakan papan elektronis dengan ukuran yang kecil yang memiliki kemampuan dan fungsi seperti komputer. Jika dihubungkan ke *monitor*, *keyboard*, *mouse* dan jaringan komputer maka kita akan dapat menggunakan layaknya sebuah komputer yang dapat melakukan pekerjaan seperti dokumen, melayani pemakaian internet dan bahkan dapat menjadikannya sebagai *web server*. Banyak percobaan telah dilakukan dikarenakan sangat efisien dalam proses pengerjaan dan juga raspberry pi sangat multifungsi dalam pengembangan setiap percobaan dilakukan[3].

2.3 Sensor Ultrasonik HC-SR04

Sensor ultrasonik HC-SR04 merupakan sensor jarak berbasis gelombang ultrasonik. Prinsip kerja sensor ini hampir sama dengan sonar kelelawar dimana untuk menentukan jarak dari terhadap sebuah objek. Sensor ini memiliki kelebihan deteksi jarak tanpa kontak dengan akurasi data yang tinggi[4]. Dan juga pembacaan yang stabil yang menghasilkan paket mudah di gunakan. Pendeteksian sensor ini pun memiliki rentang antara 2 cm sampai 400 cm tanpa terpengaruhi oleh material, sinar cahaya, ataupun warna dari objek yang di deteksi.

2.4 Sensor *Light Dependent Resistor* (LDR)

Sensor *Light Dependent Resistor*(LDR) merupakan jenis sensor yang nilainya dapat berubah akibat intensitas cahaya yang di terima oleh komponen tersebut[5]. Sensor ini memiliki sebuah semikonduktor yang memiliki dua elektroda pada permukaannya. saat ada cahaya terang maka elektron akan mengangkut muatan elektrik dan disaat cahaya redup maka elektron bebas dengan jumlah relatif yang kecil sehingga hanya sedikit elektron yang nantinya akan mengangkut muatan elektrik[5].

2.5 *Keypad* Matrik 4X4

Keypad matrik 4x4 terdiri dari 4 baris dan 4 kolom yang terdiri dari saklar-saklar *push button* yang disusun secara matrik yang berfungsi untuk dapat menginputkan data *keyword* seperti *password* yang nantinya akan di inputkan pada *keypad*[6]. Proses untuk membaca penekanan tombol matrik ini pun dilakukan dengan cara bertahap dari kolom pertama sampai kolom ke empat begitupun dengan baris dilakukan dengan bertahap dari kolom pertama ke kolom keempat. Kemudian data pembacaan scanning tadi diolah sebagai pembacaan data penekanan tombol *keypad*. Sehingga tiap tombol penekanan pada *keypad* akan menghasilkan data *keyword*[7].

2.6 *Light Emitting Diode*(LED)

Light Emitting Diode(LED) adalah perangkat elektronika yang dapat memancarkan cahaya spasial ketika diberikan tegangan pada frekuensi yang relatif sempit[8]. *Light Emitting Diode*(LED) merupakan dioda yang terbuat dari bahan semikonduktor. Karakteristik *Light Emitting Diode*(LED) berbeda-beda berdasarkan warna yang dihasilkan[8].

2.7 *Message Queue Telemetry Transport*(MQTT)

Message Queue Telemetry Transport (MQTT) merupakan protokol yang bekerja diatas protokol TCP/IP. MQTT ini bersifat *lightweight message* sehingga disebut juga dengan protokol yang mempunyai ukuran paket yang kecil pada header yang hanya memiliki ukuran sebesar 2bytes

untuk setiap jenis data serta membutuhkan catu daya yang relatif kecil dan pada dasar protokol MQTT ini dipilih karena menunjang *Internet of Things* (IoT) dan tepat sebagai penghubung antara *machine-to-machine* (M2M) yang berada pada layer aplikasi. Selain itu protokol ini juga menggunakan sistem *publish/subscribe* dalam prinsip kerjanya[9].

2.8 Aplikasi Telegram Messenger

Telegram adalah sebuah aplikasi yang memungkinkan pengguna atau *user* untuk mengirimkan pesan, gambar, data, *file* dan video tanpa menetapkan besarnya ukuran *file* yang di kirimkan secara aman yang menggunakan sistem *end-to-end*. Telegram merupakan aplikasi yang ringan, cepat dan tidak ada iklan dan gratis dalam penggunaannya dengan telegram kita bisa membuat grup yang isinya bisa mencapai maksimum 30.000 pengguna[10].

2.8.1 Telegram Bot

Bot merupakan kependekan dari intelegensi robot. Berfungsi untuk memudahkan tugas manusia. Aplikasi telegram adalah salah satu aplikasi yang mendukung adanya penggunaan bot ini. Dengan adanya bot ini akan memudahkan kita dalam membuat aplikasi pesan khususnya dengan sistem asisten yang akan menggantikan tugas dari moderasi di dalam sebuah grup[10].

2.9 Python[11]

Python adalah Bahasa pemrograman yang mendukung *object-oriented programming* dan *functionalitas programming*. Adapun fitur-fitur dari bahasa pemrograman python yang membedakan dari bahasa pemrograman lain:

1. Memiliki library yang sangat luas, dalam distribusi python telah di sediakan juga modul yang siap pakai sehingga dapat memudahkan programming.
2. Memiliki tata Bahasa yang mudah untuk di pelajari.
3. Berorientasi objek.
4. Memiliki aturan layout kode dari sumber yang dapat memudahkan pengecekan dan pembacaan.
5. Memiliki sistem pengelolaan memori otomatis.

2.10 Virtual Private Server (VPS)

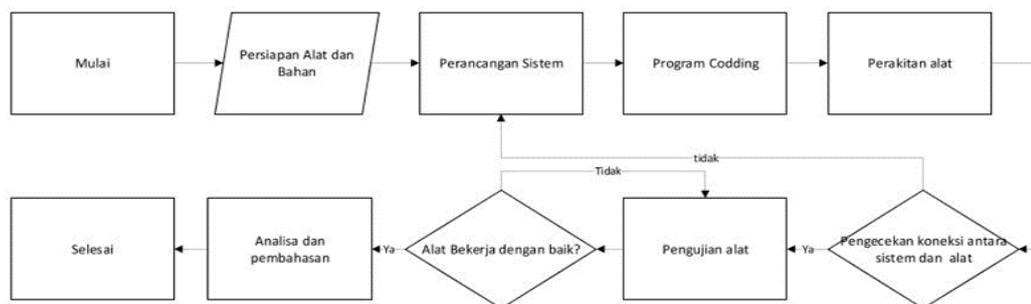
Virtual Private Server (VPS) merupakan sebuah teknologi bentuk virtual dari *server*. Dimana sebuah *physical server* yang dibentuk atau di bagi menjadi *virtual private server* sehingga satu buah *server* dapat menampung beberapa beberapa *server* secara bersamaan. VPS ini secara logis dapat bekerja terpisah meskipun secara fisik terletak pada hardware yang sama[12]. Setiap VPS memiliki *full root, access*, sistem operasi dan pengaturan terhadap *script, user* dan pemrosesan *server* termasuk *resources server* seperti CPU dan RAM[13].

3. Perancangan Sistem

Adapun tahapan yang telah dilakukan dalam penelitian prototipe *smart parking* ini meliputi perancangan perangkat yaitu sensor ultrasonik HC-SR04, sensor *Light Dependent Resistor*(LDR), *Resistor, Light Emitting Diode*(LED) dan *keypad* matrik 4x4 yang dihubungkan ke mikrokontroler raspberry pi 3 model B. *Virtual Private Server* (VPS) dihubungkan dengan mikrokontroler raspberry pi 3 model B menggunakan protokol MQTT *broker mosquitto*, dan juga *Virtual Private Server* (VPS) dihubungkan dengan aplikasi bot telegram *messenger*.

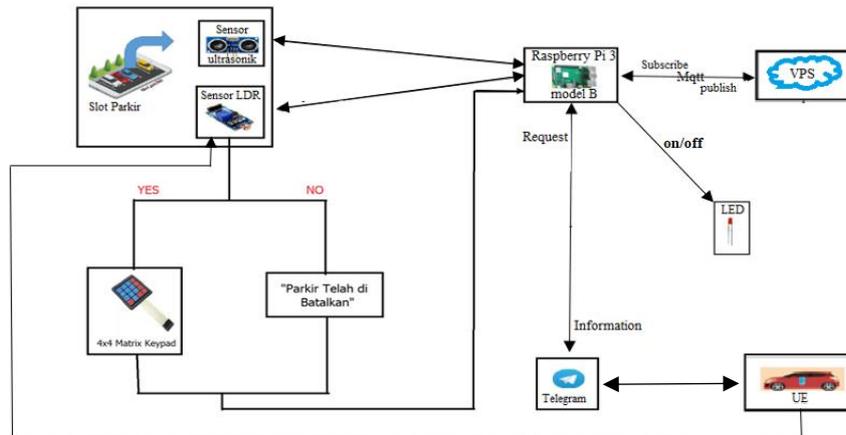
3.1. Perancangan Sistem

Perancangan sistem yang telah dilakukan meliputi proses perancangan persiapan alat dan bahan, perancangan sistem yang akan dilakukan, melakukan perakitan alat, melakukan pengecekan koneksi antara sistem dan alat yang telah di rakit, serta melakukan pengujian alat serta menganalisa *Quality of Service*(QoS) yang di dapatkan dari hasil prototipe *smart parking* ini.



Gambar 3.1 Diagram blok perancangan sistem

Adapun perancangan kerja sistem pada prototipe *smart parking* yang telah di lakukan sebagai berikut:



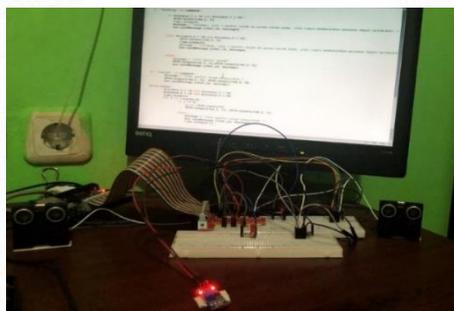
Gambar 3.2 Perancangan Kerja Sistem

Pada perancangan perangkat prototipe *smart parking* ini akan memperhatikan nilai hasil dari analisa *Quality of Service(QoS)* berupa *delay*, *throughput*, *jitter* dan *packet loss*. Dimana nantinya akan di bandingkan dengan standar TIPHON (*Telecommunication and Internet Protocol Harmonization Over Network*).

Tabel 3.1 Standar TIPHON[14]

Delay (ms)		Throughput (bps)			Jitter (ms)		Packet Loss (%)	
Kategori	Kualitas	Kategori	Indeks	Kualitas	Kategori	Kualitas	Kategori	Kualitas
0 - 150	Baik	100	4	Sangat Bagus	0 - 20	Baik	0 - 1 %	Baik
150 - 400	Dapat Diterima	75	3	Bagus	20 - 50	Dapat Diterima	1 - 5%	Dapat Diterima
>400	Buruk	50	2	Kurang Bagus	>50	Buruk	>10%	Tidak Dapat diterima
		<25	1	Jelek				

3.2 Perancangan Perangkat keras



Gambar 3.3 Perancangan perangkat keras

Perancangan sistem yang akan dilakukan meliputi proses penghubungan antara sensor ultrasonik HC-SR04, sensor *Light Dependent Resistor(LDR)*, *Resistor*, *Light Emitting Diode(LED)* dengan mikrokontroler raspberry pi 3 model B, lalu raspberry pi 3 model B akan menggunakan MQTT sebagai protocol untuk berkomunikasi dengan *Virtual Private Server (VPS)* untuk memberikan informasi ketersediaan lahan parkir yang dapat di akses melalui aplikasi telegram secara *realtime*. Pada perancangan dan implementasi prototipe *smart parking* bertujuan agar pengguna dapat lebih efisien untuk mendapatkan informasi lahan parkir secara *realtime*.

3.3 Perancangan Perangkat Lunak

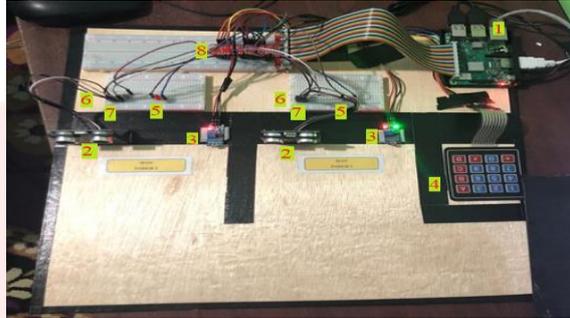
Pada perancangan perangkat lunak ini menjelaskan bagaimana proses koneksi menggunakan protokol MQTT *mosquitto broker* yang bekerja untuk proses pengiriman data antara raspberry pi dengan VPS. Dalam penerapan pada sistem *smart parking* ini, dilakukan pertimbangan syarat minimum suatu sistem dalam *Internet of Things(IoT)* agar berjalan efektif diantaranya yaitu secara *realtime*.

4. Simulasi Dan Analisis

Pada simulasi dan analisis ini akan menjelaskan mengenai implementasi dan persiapan pengujian sistem pada *smart parking* berupa akurasi sensor ultrasonik HC-SR04, sensor *Light Dependent Resistor*(LDR) yang telah digunakan di prototipe *smart parking*. Dan menjelaskan hasil analisis parameter *Quality of Service*(QoS) berupa *delay*, *throughput*, *jitter* dan *packet loss*, selanjutnya akan dibandingkan dengan standart nilai QoS dari TIPHON.

4.1 Implementasi Sistem

Pada penelitian ini implementasi sistem telah dilakukan dengan pembuatan sebuah prototipe *smart parking* yang dapat sebagai gambaran area parkir yang sesungguhnya. ukuran dari prototipe *smart parking* ini yaitu 40 cm x 40 cm.



No	Nama Perangkat
1.	Raspberry Pi 3 model B
2.	Sensor Ultrasonik HC-SR04
3.	Sensor <i>Light Dependent Resistor</i> (LDR)
4.	<i>Keypad</i> Matrik 4X4
5.	<i>Light Emitting Diode</i> (LED)
6.	Resistor 1K
7.	Resistor 10K
8.	Kabel <i>Jumper</i>

Gambar 4.1 Implementasi sistem

4.2 Pengujian Sensor Ultrasonik HC-SR04

Pada pengujian sensor ultrasonik HC-SR04 ini dilakukan pengujian berupa perubahan jarak dari objek yang berada di depan sensor ultrasonik HC-SR04, jarak tersebut sebesar :5 cm, 10 cm, 15 cm, 20 cm, dan 25 cm. Pengujian ini akan membandingkan nilai antara jarak yang telah peneliti tetapkan, dengan nilai error yang di dapatkan oleh sensor. Dimana tujuannya agar kita dapat mengetahui bagaimana ketepatan nilai dari sensor ultrasonik HC-SR04 ini bekerja.

Dengan rumus :

$$\text{Nilai error \%} = \frac{A}{B} \times 100 \%$$

Keterangan :

A = Jumlah sample dengan nilai yang tidak sesuai jarak

B = Jumlah total sample yang diambil

Tabel 4.2 Hasil pengujian sensor ultrasonik HC-SR04

Jarak yang di tetapkan untuk kedua sensor ultrasonik HC-SR04	Nilai yang terbaca oleh sensor ultrasonik HC-SR04	
	Sensor lahan parkir 1 (Nilai <i>Error</i>)	Sensor lahan parkir 2 (Nilai <i>Error</i>)
5 cm	0.1 %	0%
10 cm	0 %	0.1 %
15 cm	0.2 %	0 %
20 cm	0 %	0.2 %
25 cm	0 %	0.1 %
Rata-rata <i>Error</i>	0.06%	0.08%

Pada tabel 4.2 hasil pengujian sensor ultrasonik HC-SR04 ini telah dilakukan dalam 1 kali percobaan dengan mengambil 5 percobaan dengan nilai rata-rata *error* sebesar 0.06% untuk sensor di lahan parkir 1, dan rata-rata *error* untuk lahan parkir 2 sebesar 0.08%. Dapat di simpulkan bahwa akurasi pengukuran jarak oleh sensor ultrasonik HC-SR04 yang terdapat di lahan parkir 1 dan lahan parkir 2 *smart parking* ini telah berjalan dengan akurat untuk sistem deteksi *smart parkir*.

4.3 Pengujian Sensor *Light Dependent Resistor* (LDR)

Pada pengujian sensor *Light Dependent Resistor*(LDR) ini akan dilakukan pengujian dengan seberapa jauh jarak cahaya yang dapat diterima oleh sensor *Light Dependent Resistor*(LDR) dengan membandingkan antara jarak cahaya yang telah peneliti tetapkan, dengan nilai yang di dapatkan oleh sensor. Adapun pada pengujian ini telah dilakukan dengan 1 percobaan dengan jarak pancaran cahaya yang akan di lakukan yaitu sebesar 5 cm, 10 cm, 15 cm, dan 25 cm.

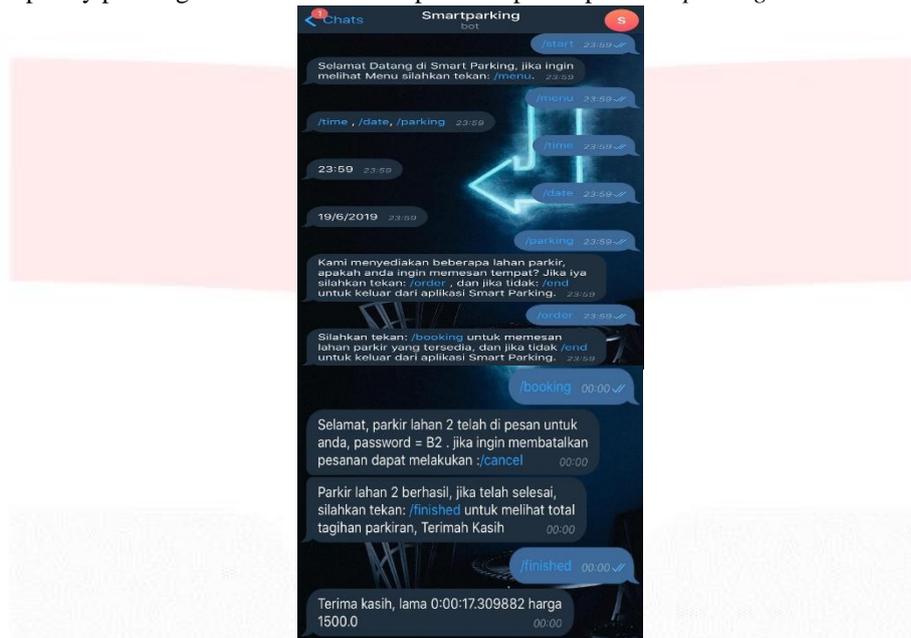
Tabel 4.3 Hasil pengujian sensor *Light Dependent Resistor*(LDR)

Jarak yang di tetapkan	sensor <i>Light Dependent Resistor</i> (LDR)	
	Lahan 1	Lahan 2
5 cm	Terdeteksi	Terdeteksi
10 cm	Terdeteksi	Terdeteksi
15 cm	Terdeteksi	Terdeteksi
20 cm	Terdeteksi	Terdeteksi
25 cm	Terdeteksi	Terdeteksi

Pada tabel 4.3 hasil pengujian sensor *Light Dependent Resistor*(LDR) didapatkan hasil terdeteksi untuk setiap jarak yang telah di tetapkan. Sehingga dapat disimpulkan bahwa sensor LDR bekerja dengan sangat baik untuk mendeteksi adanya sorotan cahaya di setiap lahan parkir *smart parking*.

4.4 Pengujian Aplikasi Telegram Messenger

Pengujian aplikasi telegram bot ini akan di lakukan dengan melakukan pengecekan apakah command bot telah dapat berjalan dengan baik dan apakah telah ada balasan dari *command* tersebut dari raspberry pi mengenai informasi lahan parkir di prototipe *smart parking*.



Gambar 4.2 Bot aplikasi telegram messenger

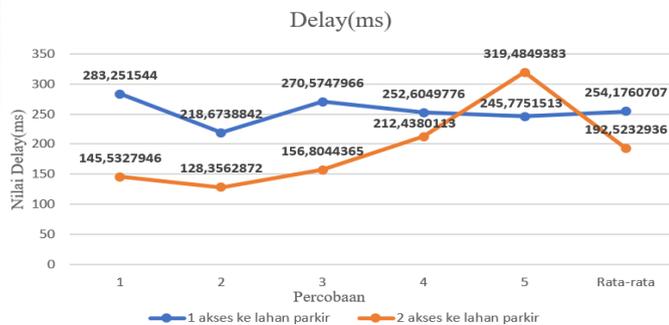
Pada pengujian ini kita dapat melihat bahwa telegram bot messenger telah dapat terkoneksi dengan baik dengan raspberry pi melalui VPS dengan memakai protokol MQTT broker mosquitto.

4.5 Hasil Pengujian Sistem Quality of Service(QoS)

Analisis pengujian sistem berupa performansi prototipe *smart parking* ini peneliti telah melakukan 5 percobaan dengan tujuan untuk dapat menganalisa nilai *Quality of Service*(QoS) seperti *delay*, *throughput*, *jitter* dan *packet loss*

4.5.1 Pengujian Delay

Analisis Proses pengujian *delay* ini peneliti telah melakukan 5 percobaan dengan satu akses dan dua akses untuk memesan lahan parkir sehingga didapatkan data sebagai berikut:



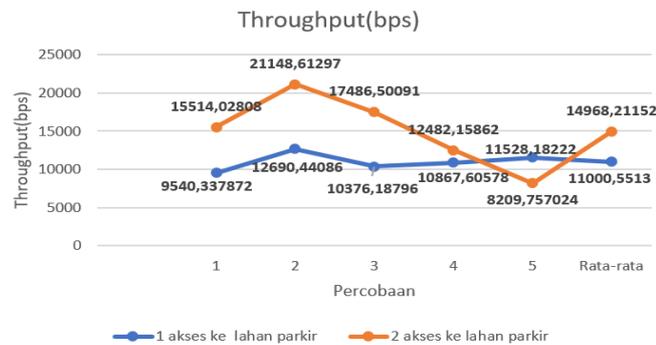
Gambar 4.3 Grafik hasil rata-rata delay

Dari gambar 4.3 menunjukkan waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan, Saat satu akses memesan 1 lahan parkir maka rata-rata *delay* yang dihasilkan sebesar 254,1760707 ms dan saat dua akses mencoba memesan 2 lahan parkir maka rata-rata *delay* yang di

hasilkan sebesar 192,5232936 ms. Standar deviasi yang dihasilkan untuk satu akses ke lahan parkir sebesar 22,12577856 dan dua akses ke lahan parkir sebesar 69,45099519. Sehingga dapat disimpulkan bahwa nilai *delay* yang didapatkan dan dibandingkan dengan standar *delay* dari TIPHON menghasilkan parameter yang dapat diterima (150 - 400 ms) untuk sebuah pengujian *delay*. Dan faktor yang mempengaruhi nilai *delay* yang di dapatkan yaitu media fisik yang tidak stabil.

4.5.2 Pengujian *Throughput*

Proses pengujian *throughput* ini peneliti telah melakukan 5 percobaan dengan satu akses dan dua akses untuk memesan lahan parkir sehingga didapatkan data sebagai berikut:

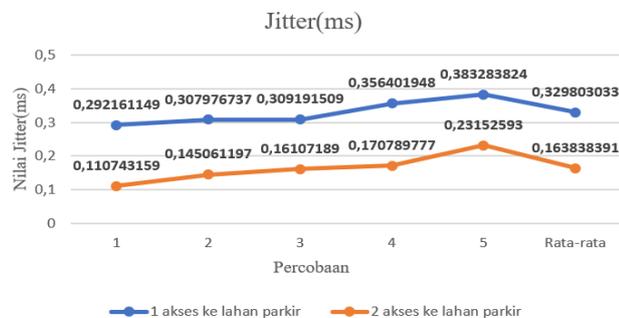


Gambar 4.4 Grafik hasil rata-rata *throughput*

Dari gambar 4.4 menunjukkan bahwa *throughput* untuk satu akses saat memesan 1 lahan parkir sebesar 11000,5513 bps dan untuk dua akses saat memesan 2 lahan parkir sebesar 14968,21152 bps. Standar deviasi yang dihasilkan untuk satu akses ke lahan parkir sebesar 1065,255647 dan dua akses ke lahan parkir sebesar 4397,595376. Sehingga dapat disimpulkan bahwa nilai *throughput* yang didapatkan telah memenuhi standar *throughput* dari TIPHON yang menghasilkan nilai sangat bagus 100 bps dengan nilai indeks 4. Dan faktor yang mempengaruhi nilai *throughput* ini diakibatkan oleh waktu pengiriman pada destinasi saat pengiriman paket yang sukses dari sumber.

4.5.3 Pengujian *Jitter*

Proses pengujian *jitter* ini peneliti telah melakukan 5 percobaan dengan satu akses dan dua akses untuk memesan lahan parkir sehingga didapatkan data sebagai berikut:

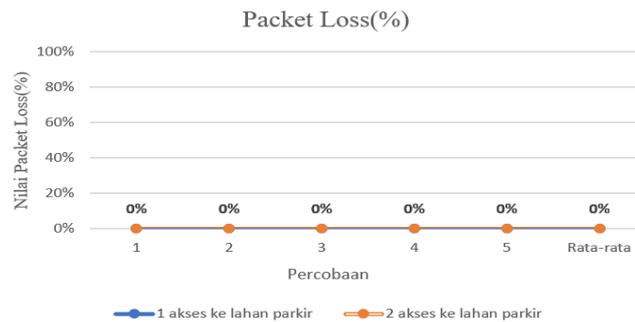


Gambar 4.5 Grafik hasil rata-rata *jitter*

Dari gambar 4.5 menunjukkan bahwa hasil pengujian dengan nilai untuk rata rata *jitter* saat satu akses memesan 1 lahan parkir yaitu sebesar 0,329803033 ms dan saat dua akses mencoba memesan 2 lahan parkir yaitu sebesar 0,163838391 ms. Standar deviasi yang dihasilkan untuk satu akses ke lahan parkir sebesar 0,034309855 dan dua akses ke lahan parkir sebesar 0,039520286. Sehingga dapat disimpulkan bahwa sesuai dengan standar nilai *jitter* pada TIPHON dengan indeks baik (0 – 20 ms). Dan faktor yang mempengaruhi nilai *jitter* yaitu variasi *delay* dan *latency* saat pengiriman data.

4.5.4 Pengujian *Packet Loss*

Pada proses pengujian *packet loss* ini peneliti telah melakukan 5 percobaan, dengan data yang dihasilkan pada tiap percobaan yaitu sebagai berikut:



Gambar 4.6 Grafik hasil rata-rata *packet loss*

Dari gambar 4.6 menunjukkan bahwa dalam 5 percobaan menghasilkan nilai rata-rata pengujian untuk *packet loss* rata rata sebesar 0% untuk satu maupun dua akses lahan parkir. Hal ini telah memenuhi standar dari TIPHON dengan indeks baik (0% – 1%). Hal ini membuktikan bahwa tidak ada paket yang hilang dikarenakan tidak terjadinya *collision* dan *congestion* disaat pengiriman data.

5. Kesimpulan

Pada hasil perancangan sistem prototipe *smart parking* ini semua sensor dapat terkoneksi dan memberikan informasi ke pengguna melalui aplikasi telegram dengan dengan akurasi informasi lahan parkir yang akurat. Dalam menganalisa *Quality of Service(QoS)* berupa *delay*, *throughput*, *jitter* dan *packet loss* dilakukan dalam 5 percobaan. Dalam percobaan *delay* menghasilkan rata-rata *delay* saat akses memesan 1 lahan parkir yaitu sebesar 254,1760707 ms dan saat dua akses memesan 2 lahan parkir yaitu sebesar 192,5232936 ms. Untuk percobaan *throughput* menghasilkan nilai saat akses memesan 1 lahan parkir yaitu sebesar 11000,5513 bps dan saat dua akses memesan 2 lahan parkir yaitu sebesar 14968,21152 bps. Untuk percobaan *jitter* menghasilkan nilai saat akses memesan 1 lahan parkir yaitu sebesar 0,329803033 ms dan saat dua akses memesan 2 lahan parkir yaitu sebesar 0,163838391 ms. Dan untuk percobaan *packet loss* menghasilkan 0% saat satu atau dua akses mencoba memesan lahan parkir yang tersedia. Sehingga dapat disimpulkan dari parameter *Quality of Service(QoS)* yang diuji telah memenuhi standar TIPHON.

Daftar Pustaka:

- [1] PUSAT BADAN STATISTIK, “Perkembangan Kendaraan Bermotor di Indonesia Menurut Jenis,” *BADAN PUSAT STATISTIK*, 2017. [Online]. Available: <https://www.bps.go.id/linkTableDinamis/view/id/1133>.
- [2] E. D. Meutia, “Internet of Things – Keamanan dan Privasi,” *Semin. Nas. dan Expo Tek. Elektro 2015*, pp. 85–89, 2015.
- [3] A. Kadir, *Dasar Raspberry Pi*, Ed.1. Andi Yogyakarta, 2017.
- [4] P. Khoenkaw, “A Software Based Method for Improving Accuracy of Ultrasonic Range Finder Module,” no. 1, pp. 3–6, 2017.
- [5] S. Aldeen, S. Alkadhim, and S. K. Aboud, “Light sensor to switch on a light or any device ldr,” no. December, 2018.
- [6] S. Pengaman, B. Di, and R. Kantor, “Jurnal ICT Penelitian dan Penerapan Teknologi.”
- [7] O. Code, “4x4 Matrix Keypad,” pp. 1–8, 2008.
- [8] G. Rg *et al.*, “The use of light-emitting diodes (LED) in commercial layer production □,” 2013.
- [9] T. Budioko, “Sistem monitoring suhu jarak jauh berbasis internet of things menggunakan protokol MQTT,” *Semin. Nas. Ris. Teknol. Inf.*, vol. 1, no. 30 July, pp. 353–358, 2016.
- [10] Telegram.org, “Telegram,” *Telegram*. [Online]. Available: <https://telegram.org/faq#q-what-is-telegram-what-do-i-do-here>. [Accessed: 03-Nov-2018].
- [11] jubilee enterprise, *Python Untuk Programmer pemula*, 1st ed. Jakarta: PT. Elex Media Komputindo, 2019.
- [12] E. Warni and A. A. Sabri, “Memory Sharing Management on Virtual Private.”
- [13] C. Eckart, *Virtual Private Server (VPS) in Web Hosting Solutions*. 2018.
- [14] I. Protocol, “Standart of QoS (Quality of Service) TIPHON,” *Etsi*, vol. 2, pp. 1–72, 2002.