SISTEM INTERNET OF THINGS DAN TRANSMISI DATA MENGGUNAKAN APLIKASI TELEGRAM PADA SISTEM KEAMANAN SEPEDA MOTOR

Abdul Kholik[1], Ibrahim [2], Reni Rahmadewi[3]
Teknik Elektro, Universitas Singaperbangsa Karawang[1]
abdul.kholik18097@student.unsika.ac.id

Submitted July 31, 2022; Revised August 3, 2022; Accepted August 4, 2022

Abstrak

Sepeda motor adalah alat transportasi kendaran beroda dua yang digerakan oleh sebuah mesin. Dengan adanya sepeda motor kita akan lebih cepat dan mudah mencapai lokasi tujuan kita. Hal itu membuat pengguna sepeda motor dijalan raya terlihat lebih banyak dari pada kendaraan lainnya. Akibat banyaknya kendaraan sepeda motor maka bertambahnya tindak kriminalitas pencurian sepeda motor. Faktor utama yang mendasari pencurian sepeda motor karena faktor ekonomi. Oleh karna itu, dibuat sistem keamanan sepeda motor berbasis IoT dengan aplikasi Telegram. Dimana sistem keamananya menggunakan GPS Quectel L80, *fingerprint* dan bisa juga mengontrol motor dengan aplikasi telegram. Hasil pengujian perintah pada sistem keamanan sepeda motor menggunakan aplikasi telegram didapat hasil waktu *delay* yang stabil pada setiap percobaan. Dimana nilai rata-rata *dela*y pada pengujian perintah *Engine ON* dan *Engine OFF* sebesar 5,01. Sedangkan nilai rata-rata *delay* pada pengujian *ADD* sidik jari sebesar 4,4 detik dan nilai rata-rata *delay* pada pengujian *Delete* sidik jari sebesar 2,3 detik. Pada penilaian *Quality of Service* dari hasil pengujian menggunakan *wireshark* dapat dilihat hasil rata-rata pada jaringan baik, karena memenuhi syarat standarisasi dengan nilai indeks 3.3. Dimana nilai rata-rata *delay* didapatkan sebesar 170,168865 ms, nilai rata-rata *throughput* didapatkan sebesar 17573,68780 *kilo bytes*/17k, dan nilai *packet loss* sebesar 0%.
Kata Kunci : Sepeda Motor, *IoT,* Telegram

Abstract

A motorcycle is a two-wheeled vehicle that is driven by an engine. With a motorcycle, it will be faster and easier to reach our destination. This makes motorcycle users on the road more visible than other vehicles. As a result of a large number of motorcycle vehicles, the crime of motorcycle theft has increased. The main factor underlying the theft of motorcycles is economic factors. Therefore, an IoT-based motorcycle security system was created with the Telegram application. Where the security system uses the Quectel L80 GPS, and fingerprint and can also control the motorbike with the telegram application. The results of testing the command on the motorcycle security system using the telegram application obtained a stable delay time in each experiment. Where the average value of delay in testing the Engine ON and Engine OFF commands is 5.01. While the average value of delay in the fingerprint ADD test is 4.4 seconds and the average delay value in the fingerprint Delete test is 2.3 seconds. In the Quality of Service assessment from the test results using Wireshark, it can be seen that the average results on the network are good because they meet the standardization requirements with an index value of 3.3. Where the average delay value is 170,168865 ms, the average throughput value is 17573,68780-kilo bytes/17k, and the packet loss value is 0%..
Key Words : *Motorcycle, IoT, Telegram*

1. PENDAHULUAN

Sepeda motor sudah tidak asing lagi di kalangan masyarakat indonesia yang mayoritas masyarakat indonesia sudah mempunyai sepeda motor. Sepeda motor adalah alat transportasi kendaran beroda dua yang digerakan oleh sebuah mesin. Masyarakat Indonesia menggunakan sepeda motor untuk mempermudah aktifitas yang dilakukan. Dengan adanya sepeda motor kita akan lebih cepat dan mudah mencapai lokasi tujuan kita. Selain itu, mengendarai sepeda motor sangat membantu jika kita akan menuju suatu tempat yang melewati kemacetan. Hal itu membuat pengguna sepeda motor dijalan raya terlihat lebih banyak dari pada kendaraan lainnya. Data menunjukan bahwa penjualan sepeda motor pada tahun 2021 yang tercatat oleh Kementrian Perindustrian (Kemenprin) sebanyak 5.057.516 unit untuk sepeda motor domestik dan sepeda motor ekspor terjual sebanyak 803.931 unit [1].

Akibat banyaknya penjualan sepeda motor maka bertambahnya tindak kriminalitas pencurian sepeda motor. Beberapa wilayah di indonesia menunjukan kasus pencurian sepeda motor masih banyak di jumpai. Faktor utama yang mendasari pencurian sepeda motor karena faktor ekonom [2]. Wilayah yang sering terjadi adanya pencurian sepeda motor salah satunya di Kabupaten Karawang, Jawa Barat. Banyak orang perantau menggunakan sepeda motor untuk melakukan aktifitas sehari-hari.

Deangan adanya pencurian sepeda motor di beberapa wilayah Kabupaten Karawang maka harus meningkatkan sistem keamanan sepeda motor. Sistem keamanan sepeda motor standar yang diberikan oleh pabrikan dirasa tidak cukup karena sistem tersebut sudah banyak diketahui orang. Penelitian tentang sistem keamanan sepeda motor telah banyak dilakukan dengan memanfaatkan mikrokontroler. Sebagian besar menggunakan teknologi SMS *gateway* dengan menggunakan modul SIM 800L [3]. Selain itu, pemanfaatan *Radio Frequency Identification* (RFID) untuk menghidupkan sepeda motor sehingga hanya pemilik yang mempunyai tag *reader* yang bisa menggunakannya [4]. Ada juga *fingerprint* digunakan sebagai saklar untuk menghidupkan motor sehingga hanya sidik jari yang terdaftar yang dapat menghidupkan motor [5].

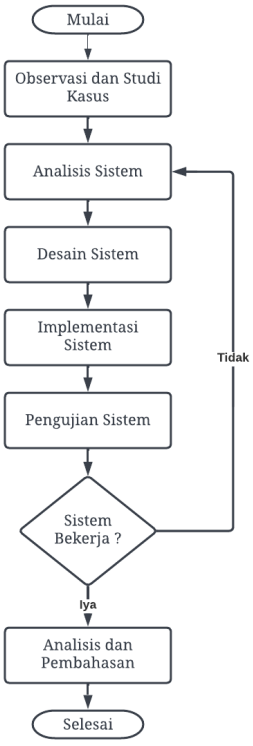
Dengan berkembangnya zaman masyarakat menggunakan teknologi dengan internet. Sehingga internet bisa digunakan dimana saja dan kapan saja untuk *Internet of Things* (IoT). *IoT* merupakan sebuah konsep dimana suatu objek ditanamkan teknologi dengan tujuan untuk berkomuniukasi, mengendalikan, dan bertukar data melalui perangkat lain yang terhubung dengan internet [6]. Dengan adanya *IoT* peneliti ingin menghubungkan sistem keamanan sepeda motor berbasis *IoT*.

Berdasarkan permasalahan diatas penulis ingin membuat sistem keamanan sepeda motor berbasis IoT menggunakan aplikasi Telegram. Dimana sistem keamananya menggunakan GPS Quectel L80, *fingerprint* dan bisa juga mengontrol motor dengan aplikasi telegram. GPS Quectel L80 untuk mencari posisi kendaraan motor secara *real time* yang terintegrasi dengan aplikasi telegram. Sensor *Fingerprint* dipilih karena memiliki keamanan yang cukup tinggi sehingga hanya bisa diakses oleh orang yang sidik jarinya tersimpan di *flash memory* sensor *fingerprint*. Sedangkan, aplikasi telegram berfungsi untuk monitoring dan sistem kontrol sepeda motor, contohnya bisa menghidupkan / mematikan motor, melihat titik kordinat motor itu berada, dan bisa menambahakan sidik jari di *flash memory* sensor *fingerprint* serta menghapus sidik jari.

2. METODE PENELITIAN

Pada penelitian ini menggunakan metode kuantitatif. Metode kuantitatif yaitu metode

yang menggunakan angka sebagai data. Data tersebut kemudian akan dikumpulkan dan dianalisis. Angka yang didapatkan adalah angka hasil pengujian terhadap alat yang telah dibuat. Kemudian data tersebut dianalisis yang menghasilkan kesimpulan dari permasalahan yang dibahas. Berikut adalah *flowchart* dari metode penelitian yang telah dilakukan.

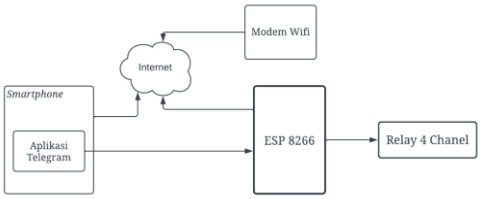


Gambar 1. Metode Penelitian

Pada Gambar 1 dijelaskan bahwa penelitian ini diawali dengan observasi dan studi kasus. Kemudian dilakukan analisis sistem untuk mengetahui permasalahan yang terdapat pada sistem. Selanjutnya dilakukan desain sistem berupa *hardware* dan *software*. Setelah itu dilakukan implementasi dari sistem yang telah dibuat sebelumnya. Tahap selanjutnya yaitu dilakukan pengujian keseluruhan sistem untuk mengetahui apakah sistem bekerja dengan baik atau tidak. Jika sistem bekerja dengan baik, berarti penelitian berhasil dan dilanjutkan ke tahap terakhir yaitu melakukan analisis dan pembahsan. Sebaliknya jika sistem tidak bekerja dengan baik atau sesuai dengan yang di inginkan kembail lagi melakukan analisis sistem tersebut agar bisa berhasil.

2.1 Perancangan Desain

Perancangan desain sistem ini dimulai dengan membuat diagram blok sistem yang menunjukan konsep desain sistem.



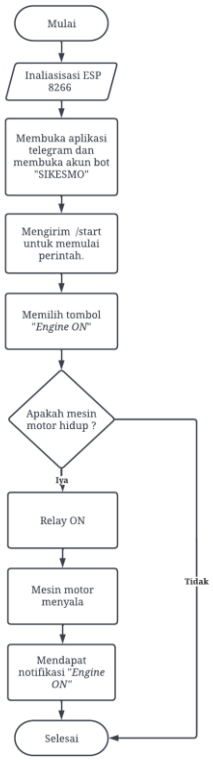
Gambar 2. Diagram Blok Sistem IoT Menggunakan Aplikasi Telegram

Dari Gambar 2. dapat kita lihat bahwa sistem kontrol dengan aplikasi telegram yang digunakan yaitu dengan Aplikasi Telegram, Relay 4 Channel, modem wifi dan mikrokontroler ESP 8266. Dalam aplikasi telegram terdapat bot akun yang sudah dibuat untuk terhubung dengan ESP 8266 agar bisa mengontrol Relay yang akan diperintahkan sesuai program.

2.2 Perancangan Sistem

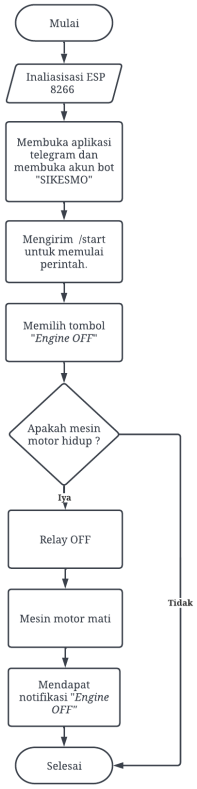
Pada perancangan implementasi sistem menjelaskan penerapan desain dari sistem yang dibuat. Implementasi sistem ini bertujuan untuk mengetahui apakah alat yang dibuat sudah sesuai dengan desain perancangan sistem atau masih ada perbaikan. Berikut ini merupakan

flowchart implementasi dapat dilihat pada Gambar dibawah ini.



Gambar 3. *Flowchart* Perintah *Engine ON* pada Aplikasi Telegram

Gambar 3. merupakan *flowchart* dari Perintah *Engine ON* pada aplikasi Telegram. Perintah *Engine ON* berfungsi untuk menyalakan motor dengan jarak jauh. Cara kerja nya yaitu saat menekan tombol "*Engine ON*", maka perintah terkirim ke mikrokontroler. Kemudian mikrokontroler diolah menjadi output relay menjadi *ON*, maka mesin motor menyala. Setelah itu, akan mendapatkan notifikasi pada telegram berupa "*Engine ON*".

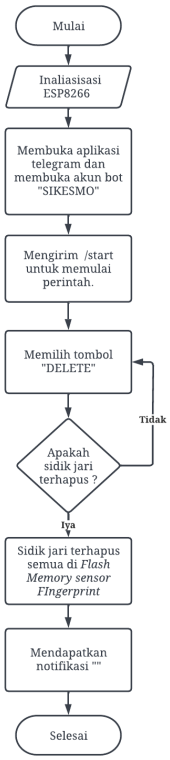


Gambar 4. *Flowchart* Perintah *Engine OFF* pada Aplikasi Telegram

Gambar 4. merupakan *flowchart* dari Perintah *Engine OFF* pada aplikasi Telegram. Perintah *Engine OFF* berfungsi untuk mematikan motor dengan jarak jauh. Cara kerja nya sama dengan perintah *Engine ON* yaitu saat menekan tombol "*Engine OFF*", maka perintah terkirim ke mikrokontroler. Kemudian mikrokontroler diolah menjadi output relay menjadi *OFF*, maka mesin motor mati. Setelah itu, akan mendapatkan notifikasi pada telegram berupa "*Engine OFF*".

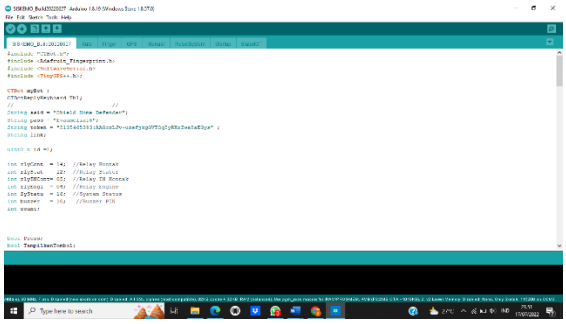
Gambar 5. *Flowchart* Perintah *ADD* Sidik Jari Pada Aplikasi Telegram

Gambar 5. merupakan *flowchart* dari Perintah *ADD* sidik jari pada aplikasi Telegram. Perintah *ADD* berfungsi untuk menambahkan sidik jari yang belum tersimpan. Cara kerja nya yaitu saat menekan tombol "*ADD*", maka perintah terkirim ke mikrokontroler. Kemudian mikrokontroler akan memerintahkan sensor fingerprint untuk *enroll* dalam pendafatarn sidik jari. Setelah mendaftarkan sidik jari maka akan mendapatkan notifikasi "*Stored*" pada aplikasi telegram.



Gambar 6. *Flowchart* Perintah *Delete* Sidik Jari Pada Aplikasi Telegram

Gambar 6. merupakan *flowchart* dari Perintah *Delete* sidik jari pada aplikasi Telegram. Perintah *Delete* berfungsi untuk menghapus sidik jari yang sudah tersimpan. Cara kerja nya yaitu saat menekan tombol "*Delete*", maka perintah terkirim ke mikrokontroler. Kemudian mikrokontroler akan memerintahkan sensor fingerprint untuk mengahpus sidik jari yang sudah tersimpan di flash memory. Setelah menghapus sidik jari maka akan mendapatkan notifikasi "*Fingerprint Deleted*" pada aplikasi telegram.

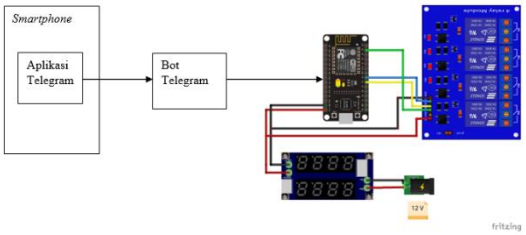


Gambar 7. *Source Code* Pada Arduino IDE

Pada Gambar 7. merupakan implementasi program yang diterapkan pada bot telegram agar terhubung dengan ESP 8266. Setelah terhubung dengan bot telegram, bot ini dapat memerintahkan atau tracking sepeda motor yang sudah di pasang pada sistem keamanan sepeda motor.

2.3 Rangkain Sistem

Rangkaian skematik merupakan sebuah rangkaian yang menunjukkan hubungan antara komponen – komponen yang digunakan pada sistem. Berikut rancangan skematik yang dibuat menggunkan aplikasi fritzing.



Gambar 8. Rangkaian Skematik Sistem IoT Menggunakan Aplikasi Telegram

Gambar 8. merupakan rangkain skematik pada sistem IoT dengan aplikasi telegram yang terdiri dari Relay 4 Chanel dan ESP 8266. Pada tabel 1. dan 2. menjelaskan konfigurasi pin masing – masing komponen.

Tabel 1. Konfigurasi Pin ESP8266

Pin ESP 8266	Keluaran Pin
VIN	5 V *Power Supply*
GND	GND *Power Supply*

Tabel 2. Konfigurasi Pin Relay 4 Channel

Pin Relay 4 Channel	Keluaran Pin
IN1	GPIO14 (D5)
IN2	GPIO12 (D6)
IN3	GPIO05 (D1)
IN4	-
VCC	5 V *Power Supply*
GND	GND *Power Supply*

3. HASIL DAN PEMBAHASAN

Setelah selesai melakukan perancangan dan pembuatan alat, maka dapat dihubungkan antara perancangan dan implementasi sistem sehingga menjadi sebuah sistem kesatuan yang dapat kita lihat pada Gambar 9. Kemudian dilakukan dengan pengujian dan analisa data pada alat yang sudah dibuat.

Gambar 9. Penempatan Alat Pada Motor

3.1 Pengujian Aplikasi Telegram

Pada pengujian aplikasi telegram, dilakukan pengujian sebanyak 2 macam. Pengujian waktu respon perintah *Engine ON* dan *Engine OFF*, dan pengujian perintah delete sidik jari. Semua pengujian mencari respon waktu/delay dari yang diperintahkan oleh bot telegram. Pada pengujian pertama dilakukan dengan 30 kali percobaan dengan jarak minimal 1 kilo meter hingga 10 kilo meter pada pengujian perintah *Engine ON* dan *Engine OFF*. Karena di perangkat *hardware* terpasang jaringan internet yang akan selalu menghidupkan sistem, agar terhubung

dengan aplikasi telegram. Sedangkan pengujian perintah *ADD* dan *Delete* sidik jari dilakukan sebanyak 20 kali.

3.1.1 Pengujian Perintah *Engine ON* dan *Engine OFF*

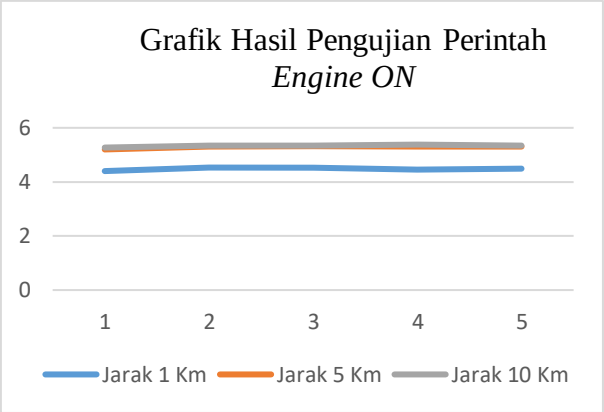
Pada pengujian ini dilakukan sebanyak 30 kali percobaan. Dimana pengujian ini melakukan perintah *Engine ON* dan *Engine OFF* dengan jarak yang berbeda-beda. Pengujian ini dilakukan dengan perintah chat bot yang sudah terhubung dengan mikrokontroler. Pengujian ini menghasilkan waktu delay berapa detik saat mengontrol motor dengan jarak yang berbeda-beda menggunakan aplikasi telegram.

Tabel 3. Hasil Pengujian Pengujian *Engine ON* dan *Engine OFF*

No.	Perintah	Jarak (Km)	*Delay* (detik)	Kondisi
1	*Engine ON*	1	4,49	Mesin motor menyala
2	*Engine ON*	1	5,20	Mesin motor menyala
3	*Engine ON*	1	5,10	Mesin motor menyala
4	*Engine ON*	1	5,25	Mesin motor menyala
5	*Engine ON*	1	5,18	Mesin motor menyala
6	*Engine ON*	5	5,20	Mesin motor menyala
7	*Engine ON*	5	5,30	Mesin motor menyala
8	*Engine ON*	5	5,24	Mesin motor menyala
9	*Engine ON*	5	5,26	Mesin motor menyala
10	*Engine ON*	5	5,30	Mesin motor menyala
11	*Engine ON*	10	5,34	Mesin motor menyala
12	*Engine ON*	10	5,26	Mesin motor menyala
13	*Engine ON*	10	5,30	Mesin motor menyala
14	*Engine ON*	10	5,29	Mesin motor menyala
15	*Engine ON*	10	5,35	Mesin motor menyala
16	*Engine OFF*	1	4,40	Mesin motor mati
17	*Engine OFF*	1	4,53	Mesin motor mati
18	*Engine OFF*	1	4,52	Mesin motor mati
19	*Engine OFF*	1	4,44	Mesin motor mati
20	*Engine OFF*	1	4,48	Mesin motor mati
21	*Engine OFF*	5	5,20	Mesin motor mati
22	*Engine OFF*	5	5,30	Mesin motor mati
23	*Engine OFF*	5	5,32	Mesin motor mati

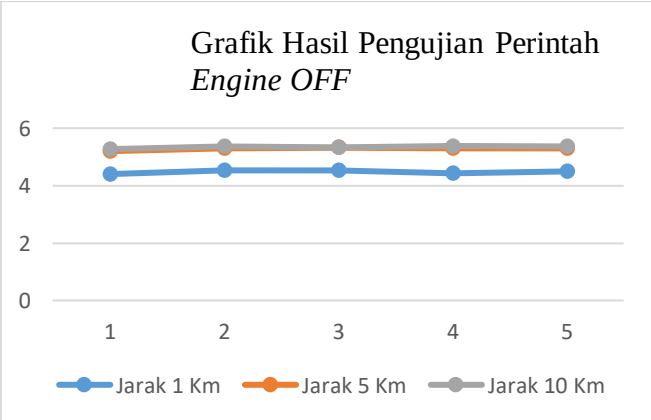
No.	Perintah	Jarak (Km)	*Delay* (detik)	Kondisi
24	*Engine OFF*	5	5,30	Mesin motor mati
25	*Engine OFF*	5	5,31	Mesin motor mati
26	*Engine OFF*	10	5,27	Mesin motor mati
27	*Engine OFF*	10	5,35	Mesin motor mati
28	*Engine OFF*	10	5,33	Mesin motor mati
29	*Engine OFF*	10	5,38	Mesin motor mati
30	*Engine OFF*	10	5,35	Mesin motor mati

Pada pengujian perintah melakukan perintah *Engine ON* dan *Engine OFF* menggunakan aplikasi telegram dilakukan sebanyak 30 kali dengan jarak yang berbeda yaitu 1, 5, dan 10 kilo meter. Pada setiap jarak dilakukan 5 percobaan disetiap perintah. Pada Tabel 3. merupakan hasil dari pengujian yang dilakukan. Untuk rata-rata delay pada jarak 1 kilo meter dengan perintah *Engine ON* sebesar 4,47 detik. Sedangkan pada jarak 5 kilo meter rata-rata delay sebesar 5,26 detik, dan pada jarak 10 kilo meter sebesar 5,30 detik. Pada perintah *Engine OFF* rata-rata delay pada jarak 1 kilo meter sebesar 4,47 detik. Sedangkan pada jarak 5 kilo meter rata-rata delay sebesar 5,28 detik, dan pada jarak 10 kilo meter rata-rata delay sebesar 5,33 detik. Dengan demikian dari ketiga jarak dengan perintah *Engine ON* dan *Engine OFF* memiliki nilai rata-rata delay sebesar 5,01 detik.



Gambar 10. Grafik Pengujian Delay dengan Perintah *Engine ON*

Pada Gambar 10. ditampilkan dari hasil pengujian pengulangan perintah *Engine ON* pada *chat* bot telegram dengan jarak berbeda yang dilakukan dengan 5 kali pengujian pada setiap jarak. Dapat dilihat pada Gambar 3.1 pada jarak 1 kilo meter hasil pengujian terlihat stabil berada pada nilai 4 detik lebih. Sedangkan untuk jarak 5 dan 10 kilo meter juga sama hasil pengujiannya terlihat stabil berada pada nilai 5 detik lebih. Dari kedua perintah tersebut bahwa pengujiannya menghasilkan nilai yang stabil dan tidak berbeda jauh dengan yang lainnya, dikarenakan internet yang terhubung lancar dan baik.



Gambar 11. Grafik Pengujian Delay dengan Perintah *Engine OFF*

Pada Gambar 11 ditampilkan dari hasil pengujian pengulangan perintah *Engine OFF* pada *chat* bot telegram dengan jarak berbeda yang dilakukan dengan 5 kali

pengujian pada setiap jarak. Dapat dilihat pada Gambar 11. pada jarak 1 kilo meter hasil pengujian terlihat stabil berada pada nilai 4 detik lebih. Sedangkan untuk jarak 5 dan 10 kilo meter juga sama hasil pengujiannya terlihat stabil berada pada nilai 5 detik lebih. Dari kedua perintah tersebut bahwa pengujiannya menghasilkan nilai yang stabil dan tidak berbeda jauh dengan yang lainnya, dikarenakan internet yang terhubung lancar dan baik.

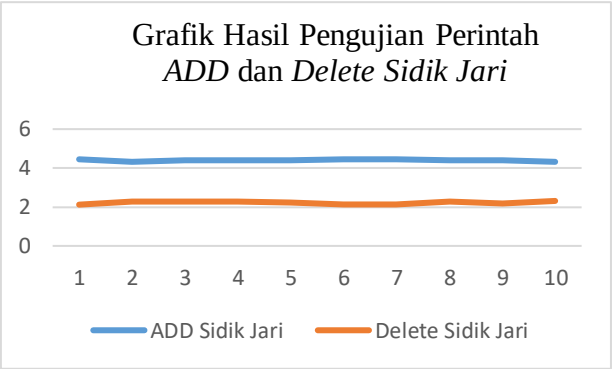
3.1.2 Pengujian Perintah *ADD* dan *Delete* Sidik Jari

Pada pengujian ini dilakukan sebanyak 20 kali percobaan. Dimana pengujian ini melakukan menambahkan sidik jari dan menghapus sidik jari yang sudah tersimpan. Pengujian ini dilakukan dengan perintah chat bot yang sudah terhubung dengan mikrokontroler. Pengujian ini melibatkan selisih waktu respon dan kondisi apakah sesuai dengan perintah.

Tabel 4. Hasil Pengujian Perintah *ADD* dan *Delete* Sidik Jari

`No.	Perintah	*Delay* (Detik)	Kondisi
1	*Add* sidik jari	4,45	Berhasil
2	*Add* sidik jari	4,32	Berhasil
3	*Add* sidik jari	4,40	Berhasil
4	*Add* sidik jari	4,42	Berhasil
5	*Add* sidik jari	4,41	Berhasil
6	*Add* sidik jari	4,43	Berhasil
7	*Add* sidik jari	4,45	Berhasil
8	*Add* sidik jari	4,41	Berhasil
9	*Add* sidik jari	4,40	Berhasil
10	*Add* sidik jari	4,32	Berhasil
11	*Delete* sidik jari	2,12	Berhasil
12	*Delete* sidik jari	2,30	Berhasil
13	*Delete* sidik jari	2,29	Berhasil
14	*Delete* sidik jari	2,30	Berhasil
15	*Delete* sidik jari	2,22	Berhasil
16	*Delete* sidik jari	2,15	Berhasil
17	*Delete* sidik jari	2,13	Berhasil
18	*Delete* sidik jari	2,27	Berhasil
19	*Delete* sidik jari	2,20	Berhasil
20	*Delete* sidik jari	2,31	Berhasil

Pada pengujian melakukan perintah *ADD* dan *Delete* Sidik Jari menggunakan aplikasi telegram dilakukan sebanyak 20 kali dengan di bagi 2 perintah. Pada tabel 4. merupakan hasil dari pengujian perintah *ADD* dan *Delete* sidik jari. Pada pengujian perintah yang pertama yaitu *ADD* sidik jari menghasilkan rata-rata delay bernilai 4,4 detik. Pada pengujian perintah *Delete* sidik jari menghasilkan rata-rata delay bernilai 2,3 detik.



Gambar 12. Grafik Selisih Waktu Respon Pengiriman Perintah

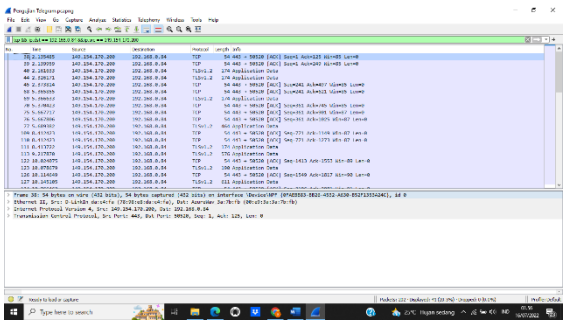
Pada Gambar 12 ditampilkan dari hasil pengujian pengulangan perintah chat bot telegram pada 2 kondisi berbeda yang dilakukan dengan 10 kali pengujian pada setiap perintah. Dapat dilihat pada Gambar 12. perintah *ADD* sidik jari hasil pengujian terlihat stabil berada pada nilai 4 detik lebih. Untuk perintah Delete sidik jari juga sama hasil pengujiannya terlihat stabil berada pada nilai 2 detik lebih. Dari kedua perintah tersebut bahwa pengujiannya menghasilkan nilai yang stabil dan tidak berbeda jauh dengan yang lainnya, dikarenakan internet yang terhubung bagus.

3.2 Pengujian Transmisi Data Pada Apliaksi Telegram

3.2.1 Pengujian *Delay*

Delay merupakan waktu yang dibutuhkan untuk sebuah paket yang dikirimkan dari suatu komputer ke komputer yang dituju. *Delay* dalam sebuah proses transmisi paket dalam sebuah jaringan komputer disebabkan karena adanya antrian yang panjang, atau mengambil rute lain untuk menghindari kemacetan pada routing [7].

Berikut ini adalah gambar dan tabel dengan melakukan pengiriman input data dari telegram web menggunakan *software* wireshark.



Gambar 13. Tampilan Wireshark Pengujian Delay

Tabel 5. Hasil Pengujian *Delay*

Time	Source	Destination	Time delta from previous displayed frame
2.135	149.154.170.200	192.168.0.84	0
2.139	149.154.170.200	192.168.0.84	0,004474
2.161	149.154.170.200	192.168.0.84	0,021674
2.326	149.154.170.200	192.168.0.84	0,164538
2.373	149.154.170.200	192.168.0.84	0,047643
5.365	149.154.170.200	192.168.0.84	0
5.366	149.154.170.200	192.168.0.84	0,000778
5.370	149.154.170.200	192.168.0.84	0,00379
5.667	149.154.170.200	192.168.0.84	0,297294
5.667	149.154.170.200	192.168.0.84	0,000089
5.689	149.154.170.200	192.168.0.84	0,021576
8.412	149.154.170.200	192.168.0.84	0
8.412	149.154.170.200	192.168.0.84	0
8.413	149.154.170.200	192.168.0.84	0,001299
9.217	149.154.170.200	192.168.0.84	0,804148
10.024	149.154.170.200	192.168.0.84	0,806205
10.078	149.154.170.200	192.168.0.84	0,054604
10.114	149.154.170.200	192.168.0.84	0,03597
10.145	149.154.170.200	192.168.0.84	0,030456
10.766	149.154.170.200	192.168.0.84	0,621357
10.787	149.154.170.200	192.168.0.84	0,020659
11.451	149.154.170.200	192.168.0.84	0,664669
11.451	149.154.170.200	192.168.0.84	0
11.731	149.154.170.200	192.168.0.84	0,27993
11.743	149.154.170.200	192.168.0.84	0,011761
14.129	149.154.170.200	192.168.0.84	0
14.493	149.154.170.200	192.168.0.84	0,364365
14.537	149.154.170.200	192.168.0.84	0,04408
16.009	149.154.170.200	192.168.0.84	0
16.812	149.154.170.200	192.168.0.84	0,803006
16.815	149.154.170.200	192.168.0.84	0,003243
16.828	149.154.170.200	192.168.0.84	0,012624
16.861	149.154.170.200	192.168.0.84	0,033306
16.861	149.154.170.200	192.168.0.84	0
16.877	149.154.170.200	192.168.0.84	0,015952
16.911	149.154.170.200	192.168.0.84	0,034248
16.911	149.154.170.200	192.168.0.84	0
16.948	149.154.170.200	192.168.0.84	0,036609
17.524	149.154.170.200	192.168.0.84	0,576192
17.549	149.154.170.200	192.168.0.84	0,025066
17.557	149.154.170.200	192.168.0.84	0,007896

Untuk mencari rata-rata *delay* maka menggunakan rumus sebagai berikut:

$$\text{Rata-rata } Delay = \frac{Total\ Delay}{Total\ Paket\ yang\ diterima}$$

$$\text{Rata-rata } Delay = \frac{6{,}8067546}{40} = 0{,}170168865 \text{ s.}$$

Rata-rata *Delay* = 0,170168865 s x 1000 = 170,168865 ms.

Adapun standar *latency*/*delay* menurut TIPHON adalah sebagai berikut [8]:

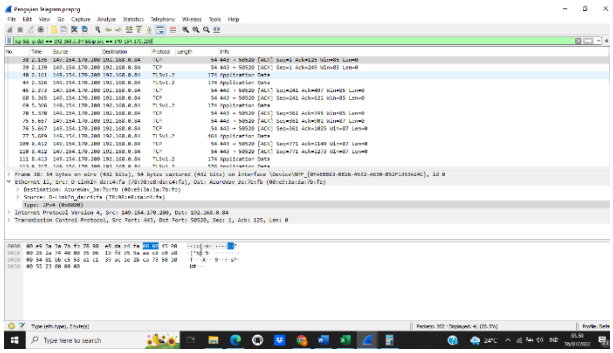
Tabel 6. Standarisasi *Delay*

Kategori Latency	Besar Delay	Indeks
Sangat Bagus	< 150 ms	4
Bagus	150 – 300 ms	3
Sedang	300 – 450 ms	2
Buruk	>450 ms	1

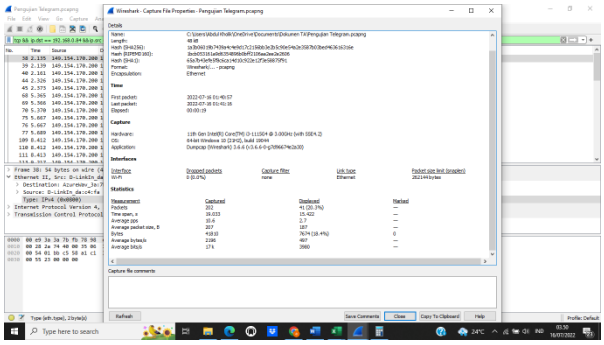
Dari hasil pengujian bahwa rata-rata *delay* sebesar 170,168865 ms, maka kategori latency dan indeks dilihat dari Tabel 6. menghasilkan Kategori *latency*/delay bagus dan Indeks dinilai 3.

3.2.2 Pengujian *Throughput*

Throughput merupakan suatu kecepatan (rate) dalam melakukan transfer data efektif, yang diukur dalam *bit per second* (bps) [9].



Gambar 14. Tampilan Wireshark TCP



Gambar 15. Tampilan Wireshark Properties TCP

Untuk mencari *throughput* dapat dihitung menggunakan rumus sebagai berikut :

$$Throughput = \frac{data\ yang\ dikirim\ (byte)}{waktu\ pengiriman\ data}$$

$$Throughput = \frac{41810}{19{,}033} = 2196{,}71097567383$$

byte/s

Throughput = 2196,71097567383 x 8

Throughput = 17573,68780 kilo bytes = 17k

Adapun standard *throughput* menurut TIPHON adalah sebagai berikut [8]:

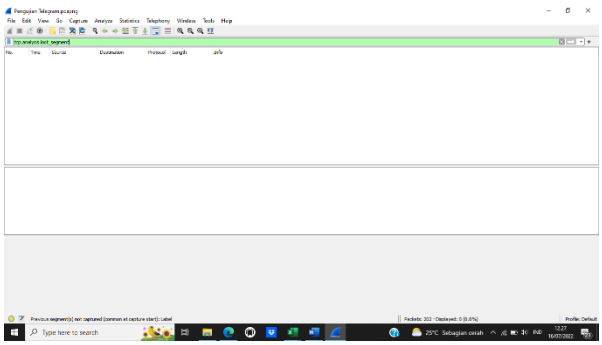
Tabel 7. Standarisasi *Throughput*

Kategori *Throughput*	Besar *bytes*	Indeks
Sangat buruk	0 – 338 kbps	0
Buruk	338 – 700 kbps	1
Sedang	700 – 1200kbps	2
Bagus	1200 kbps – 2,1 Mbps	3
Sangat bagus	> 2,1 Mbps	4

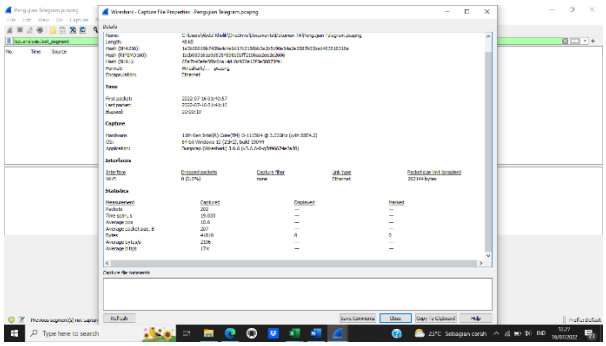
Dari hasil pengujian *throughput* didapatkan 17000 kbps, maka standarisasi *throughput* nya dilihat pada Tabel 7. adalah kategorinya bagus dan indeks nya bernilai 3.

3.2.3 Pengujian *Packet Loss*

Packet loss merupakan sebuah kondisi dimana ketika paket data yang ditransmisikan gagal tiba ditujuan atau kehilangan**.**



Gambar 16. Tampilan Wireshark Pengujian *Paket Loss*



Gambar 17. Tampilan Wireshark Properties *Packet Loss.*

Dapat dilihat pada Gambar 16. bahwa tidak terdapat *packet loss*. Selain melihat dari gambar diatas bisa juga di hitung

menggunakan rumus. Untuk rumus yang digunakan sebagai berikut:

$$Packet\ Loss = \frac{(Paket\ data\ terkirim - Paket\ data\ diterma)\ X\ 100\%}{Paket\ data\ terkirim}$$

$$Packet\ Loss = \frac{(202-202)\ x\ 100\%}{202}$$

$$Packet\ Loss = \frac{0\ x\ 100\%}{202} = 0\%$$

Hasil dari perhitungan dan dari gambar 17. *packet loss* menghasilkan nilai yang sama yaitu 0%. Adapun standar packet loss menurut TIPHON adalah sebagai berikut [8]:

Tabel 8. Standarisasi Packet Loss.

Kategori *Paket Loss*	Besar *Paket Loss*	Indeks
Sangat bagus	0%	4
Bagus	3%	3
Sedang	15%	2
Buruk	>25%	1

Dari hasil pengujian dan perhitungan rumus didapatkan persentasi nilai *packet loss* sebesar 0%, maka standarisasi *packet loss* jika dilihat pada Tabel 8. adalah kategori nya sangat bagus dan indeks nya bernilai 4.

3.2.4 *Quality of Service*

Quality of Service (QoS) merupakan suatu metode pengukuran seberapa baiknya jaringan yang dipakai [7]. Pada pengujian transmisi data menggunakan software wireshark, dapat disimpulkan sebagai berikut:

Tabel 9. Hasil *Quality of Service*

No.	Parameter	Indeks	Kategori
1	*Delay*	3	Bagus
2	*Throughput*	3	Bagus
3	*Packet Loss*	4	Sangat bagus
Rata - Rata		3.3	Bagus

Setelah dikelompokan dalam Tabel 9. maka nilai QoS secara keseluruhan dari pengujian pertama hingga pengujian ketiga dapat dilihat bahwa nilai rata-rata QoS pada jaringan bagus karena memenuhi syarat standarisasi dengan nilai rata-rata indeks 3,3.

4. SIMPULAN

Dari hasil pembahasan sebelumnya, terdapat beberapa kesimpulan yang dapat diambil antara lain:

1. Dari hasil pengujian perintah pada sistem keamanan sepeda motor menggunakan aplikasi telegram didapat hasil waktu delay yang stabil pada setiap percobaan. Dimana nilai rata-rata delay pada pengujian perintah Engine ON dan Engine OFF sebesar 5,01. Sedangkan nilai rata-rata delay pada pengujian ADD sidik jari sebesar 4,4 detik dan nilai rata-rata delay pada pengujian Delete sidik jari sebesar 2,3 detik.
2. Dari hasil pengujian Quality of Service nilai QoS secara keseluruhan dari pengujian *Throughput, Delay*, dan *Packet Loss* dapat dilihat hasil rata-rata pada jaringan baik, karena memenuhi syarat standarisasi dengan nilai rata-rata indeks 3,3. Dimana nilai rata-rata *delay* didapatkan sebesar 170,168865 ms, nilai rata-rata *throughput* didapatkan sebesar 17573,68780 kilo bytes/17k, dan nilai *packet loss* sebesar 0%.

DAFTAR PUSTAKA

[1] "Statistic Distribution - AISI." https://www.aisi.or.id/statistic/ (accessed Jun. 23, 2022).

[2] V. F. Setiawan and A. Ma, "Sistem Keamanan Sepeda Motor (SIKESEM) Menggunakan Kamera dan GPS Berbasis Internet of Things," *J. Jtev Tek. Elektro*, vol. 8, no. 1, pp. 57–66, 2022.

[3] A. Surahman, A. Tri Prastowo, and L. Ashari Aziz, "Rancang Alat Keamanan Sepeda Motor Honda Beat Berbasis Sim Gsm Menggunakan Metode Rancang Bangun," *J. Univ. Teknokr. Indones.*, vol. 3, no. 1, pp. 17–24, 2022.

[4] F. F. Musyafa, S. Pamuji, and H. Nasrullah, "Sistem Keamanan Sepeda Motor Mio Gt Berbasis Arduino Uno Dan Rfid," *Auto Tech J. Pendidik. Tek. Otomotif Univ. Muhammadiyah Purworejo*, vol. 16, no. 2, pp. 174–186, 2021, doi: 10.37729/autotech.v16i2.1253.

[5] Raju Rizkyana and Awang Surya, "Sistem Keamanan Sepeda Motor Dengan Mengganti Saklar Starter Menggunakan Fingerprint," *JTTM J. Terap. Tek. Mesin*, vol. 2, no. 1, pp. 43–51, 2021, doi: 10.37373/jttm.v2i1.90.

[6] A. Tri Wibowo, I. Salamah, and A. Taqwa, "Rancang Bangun Sistem Keamanan Sepeda Motor Berbasis Iot (Internet of Things)," *J. Fasilkom*, vol. 10, no. 2, pp. 103–112, 2020, doi: 10.37859/jf.v10i2.2083.

[7] Hasanul Fahmi, "Analisis Qos (Quality of Service) Pengukuran Delay, Jitter, Packet Lost Dan Throughput Untuk Mendapatkan Kualitas Kerja Radio Streaming Yang Baik," *J. Teknol. Inf. dan Komun.*, vol. 7, no. 2, pp. 98–105, 2018.

[8] M. Purwahid and J. Triloka, "Analisis Quality of Service (QOS) Jaringan Internet Untuk Mendukung Rencana Strategis Infrastruktur Jaringan Komputer Di SMK N I Sukadana," *Jtksi*, vol. 2, no. 3, pp. 100–109, 2019, [Online]. Available: https://ojs.stmikpringsewu.ac.id/index.php/jtksi/article/view/778/

[9] E. B. Wagiu, A. Butar-butar, and J. I. Sihotang, "Analisis QoS (Quality of Service) Pada Jaringan Internet (Studi Kasus: Universitas Advent Indonesia)," *TeIKa*, vol. 9, no. 01, pp. 31–41, 2019, doi: 10.36342/teika.v9i01.789.