

ANALISIS QOS (*QUALITY OF SERVICE*) SISTEM MONITORING PINTAR MITIGASI PENULARAN COVID-19 BERBASIS IOT

Depandi Enda¹, Muhammad Asep Subandri², Supria³

Jurusan Teknik Informatika, Politeknik Negeri Bengkalis, Indonesia

¹depandienda@polbeng.ac.id, ²msubandri@polbeng.ac.id, ³phiya@polbeng.ac.id

Abstrak

Sistem *monitoring* pintar mitigasi COVID-19 adalah sistem *monitoring* berbasis teknologi IoT yang menggunakan beberapa sensor untuk merekam keadaan di sekitar lokasi pemantauan dan memberikan rekomendasi kepada pengguna berdasarkan data yang diambil disekitar area pemantauan. Salah satu aspek yang harus diperhatikan dalam membangun sistem *monitoring* berbasis IoT adalah kualitas layanan jaringan (*Quality of Service*). Kualitas layanan jaringan digunakan sebagai tolak ukur untuk memastikan perangkat IoT dapat berkomunikasi dengan baik didalam suatu jaringan nirkabel. Penelitian ini bertujuan untuk melakukan analisis kualitas layanan jaringan pada sistem *monitoring* pintar mitigasi penularan COVID-19 berbasis IoT. Adapun aspek yang diukur adalah *delay*, *packet lost*, *jitter* dan *throughput*, dimana pengukuran parameter ini berguna untuk mengetahui tingkat performa *sensor node* (ESP8266 dan ESP32) yang digunakan pada sistem *monitoring* dalam menangani permintaan layanan API ke *web server*. Uji coba dilakukan sebanyak 10 kali, dengan durasi masing-masing uji coba 100 detik, jumlah paket yang dikirimkan 40 dan besar setiap paket 7200 *bytes*. Berdasarkan hasil uji coba pengiriman data secara simultan, ESP8266 lebih baik daripada ESP32 dengan rata-rata selisih *delay* 48.06 milidetik, *packet lost* 0%, *jitter* 0.44 milidetik dan *throughput* 2.18 *bytes*/detik.

Kata kunci : QoS, Sistem *Monitoring*, Mitigasi, COVID-19, IoT.

1. Pendahuluan

Coronavirus Disease 2019 (COVID-19) adalah sebuah pandemi global yang menjadi krisis kesehatan dunia diakhir tahun 2019. COVID-19 dapat menyebar antar individu dengan cepat melalui cairan *droplet* yang dikeluarkan melalui batuk atau bersin dan media perantara lainnya yang telah terkontaminasi oleh virus. Oleh karena itu langkah mitigasi dan penanganan yang tepat dalam menghentikan penyebaran virus ini sangatlah diperlukan.

IoT menjadi salah satu teknologi yang dapat dimanfaatkan untuk mitigasi COVID-19. Kumar et al., 2020 pada penelitiannya yang berjudul peran IoT untuk mencegah penyebaran COVID-19, menyebutkan bahwa penggunaan sensor pintar yang terhubung ke *cloud* untuk mengukur dan merekam suhu tubuh akan membantu mengidentifikasi individu yang terinfeksi. Sehingga dapat memutus mata rantai penyebaran COVID-19. Enda & Subandri, 2020, mengusulkan sistem *monitoring* pintar berbasis IoT untuk meminimalisir resiko penyebaran COVID-19 di lokasi padat publik. Sistem ini dapat memberikan informasi yang berguna bagi masyarakat sebelum memutuskan untuk berpergian ke tempat-tempat padat publik. Informasi tersebut diantaranya suhu, kelembaban, tingkat kebisingan yang merupakan indikator resiko penyebaran COVID-19.

Sistem berbasis IoT pada dasarnya membutuhkan konektivitas dan ketersediaan yang tinggi pada sebuah layanan jaringan. Semakin banyaknya perangkat IoT yang saling terhubung bersama untuk bertukar informasi melalui *internet* maka kebutuhan akan standar kualitas layanan jaringan (*Quality of Service*) harus diperhatikan dan merupakan salah satu aspek penting yang perlu dikaji (Tabbasum et al., 2018).

Beberapa penelitian lainnya yang membahas tentang kualitas layanan jaringan pada sistem berbasis IoT yaitu analisa perbandingan protokol MQTT dengan HTTP pada *platform* IoT Patriot, hasil penelitian menunjukkan bahwa protokol MQTT dapat lebih dipercaya berjalan pada *bandwidth* rendah atau latensi tinggi jika dibandingkan dengan protokol HTTP (Windryani et al., 2019). Sebuah kajian studi mengenai arsitektur kualitas layanan jaringan dan implementasinya dalam lingkungan IoT juga telah dibahas. Kajian ini membagi tiga *layer* arsitektur pada sistem jaringan IoT yaitu *layer* aplikasi, jaringan dan persepsi. Pada setiap *layer* ini akan dikelompokkan parameter kualitas layanan jaringannya (Bhaddurgatte & BP, 2015). Studi lainnya yang juga membahas tentang kualitas layanan jaringan pada sistem IoT adalah pemetaan sistematis pendekatan kualitas jaringan IoT. Kajian ini membahas tentang objek yang sering diteliti pada layanan kualitas jaringan IoT, yaitu *layer* arsitektur, faktor kualitas layanan jaringan dan jenis penelitian

apa yang sering dibahas pada area tersebut (White et al., 2017).

Dari beberapa penelitian sebelumnya yang telah dibahas maka pada penelitian ini akan dibahas kualitas layanan jaringan pada layer jaringan (Bhaddurgatte & BP, 2015). *Layer* ini adalah *layer* yang cukup menjadi perhatian pada arsitektur kualitas layanan jaringan berbasis IoT (White et al., 2017). Dimana pada *layer* ini juga akan menentukan kinerja dan performa layanan jaringan khususnya pada *platform* IoT.

Penelitian ini berfokus pada pengujian dan analisis kualitas layanan jaringan pada sistem *monitoring* berbasis IoT. Adapun sistem yang diuji adalah sistem *monitoring* pintar mitigasi penularan COVID-19 berbasis IoT yang merupakan sistem yang telah dirancang pada penelitian sebelumnya (Enda & Subandri, 2020). Adapun parameter layanan kualitas jaringan yang diuji adalah *delay*, *packet lost*, *jitter* dan *throughput*. Dimana pengujian dilakukan menggunakan kakas bantu *wireshark* dalam mengamati lalu lintas jaringan pada sistem yang diuji untuk menganalisis layanan kualitas layanan jaringan antara *sensor node* (ESP8266 dan ESP32) ke unit *base station*. Dengan adanya pengujian dan analisis kualitas layanan jaringan pada sistem yang diuji dapat memberikan gambaran apabila sistem ini diimplementasikan pada kondisi lingkungan sesungguhnya dan memberikan saran maupun pertimbangan dalam memilih perangkat *sensor node* mana yang lebih baik untuk diterapkan.

2. Tinjauan Pustaka

2.1 Kajian Terdahulu

IoT adalah teknologi baru yang saat ini hadir di sebagian besar proses dan perangkat yang memungkinkan untuk meningkatkan kualitas hidup manusia dan memfasilitasi akses ke informasi dan layanan tertentu (Villamil et al., 2020). IoT juga dapat berperan untuk melakukan tugas tertentu menggunakan beberapa perangkat IoT dan bekerja secara otomatis tanpa campur tangan manusia (Hassija et al., 2019). Salah satu penerapannya adalah dibidang kesehatan. Dibidang kesehatan, perangkat IoT dapat digunakan untuk memonitoring status kesehatan pasien secara *realtime* seperti tekanan darah, detak jantung dan tingkat glukosa (Kumar et al., 2020). Beberapa manfaat yang diperoleh dari penerapan aplikasi IoT yaitu untuk memberikan perawatan yang terbaik kepada pasien, mengurangi resiko kesalahan manusia, mengurangi biaya, meningkatkan diagnosis dan pengawasan yang efektif (Singh et al., 2020).

Beberapa peneliti telah berkontribusi dalam memerangi pandemi COVID-19 seperti penelitian yang dilakukan oleh Mohammed et al. all mengembangkan sebuah helm berbasis teknologi IoT untuk mendeteksi dan mendiagnosis orang yang terpapar COVID-19 menggunakan *thermal camera*

(Mohammed et al., 2020). Penelitian berikutnya menghasilkan sebuah model untuk mengidentifikasi gejala COVID-19 menggunakan aplikasi kecerdasan buatan (Vaishya et al., 2020). Aplikasi berbasis IoT nCapp juga telah dikembangkan untuk mendiagnosis lebih dini COVID-19 dan memandu tata cara pengobatannya. nCapp bekerja berdasarkan data, kuisioner dan hasil pemeriksaan yang ada untuk mendiagnosis pengguna secara otomatis sebagai dikonfirmasi, dicurigai, atau dicurigai terinfeksi COVID-19 (Bai et al., 2020).

Sebagai upaya mitigasi penularan COVID-19 di tempat padat publik telah dibangun sistem *monitoring* pintar berbasis IoT untuk melakukan *monitoring* keadaan lingkungan sekitar lokasi padat publik seperti tempat perbelanjaan, kampus, pelabuhan dan tempat lainnya (Enda & Subandri, 2020).

Aspek kritis dari sebuah sistem berbasis IoT adalah kualitas layanan jaringan. Kualitas layanan jaringan sangat menentukan ketersediaan dan kecepatan konektivitas antar perangkat IoT. Sehingga apapun sistem berbasis IoT yang dibangun jika tidak memenuhi standar layanan kualitas jaringan yang memadai maka akan sulit untuk diimplementasikan dalam skala yang besar (Tabbasum et al., 2018). Terlebih lagi jika sistem berbasis IoT diterapkan untuk kebutuhan yang kritis misalnya di bidang kesehatan untuk menangani pandemi COVID-19. Oleh karena itu, butuh pengujian dan analisis kualitas layanan jaringan pada sistem berbasis IoT.

Beberapa penelitian yang terkait dengan pengujian dan analisis kualitas layanan jaringan yaitu pengembangan sebuah kakas bantu pengujian API otomatis dan studi literatur tentang pengujian API otomatis. Pengujian API otomatis ini bersifat krusial yang memastikan fungsionalitas, reliabilitas, keamanan dan pengiriman logika bisnis sebuah sistem berjalan dengan baik (Isha et al., 2018). Analisa QoS pada Implementasi IPV4 dan IPV6 dengan Teknik *Tunneling*, penelitian ini menggunakan beberapa parameter untuk mengukur kualitas layanan jaringan yaitu kecepatan *upload*, *download*, *delay*, *packet lost*, *throughput* dan juga *jitter* (Novianti & Widiatoro, 2017). Serta analisis QoS pengukuran *delay*, *jitter*, *packet lost* dan *throughput* untuk mendapatkan kualitas kerja radio *streaming* yang baik (Hasanul Fahmi, 2018).

2.2 Landasan Teori

1. Modul WiFi ESP8266 dan ESP32

Modul WiFi ESP8266 adalah papan pengembangan WiFi yang menggunakan *chip* ESP8266-12E sebagai pengendali utama. Modul ini dapat digunakan sebagai perangkat komunikasi dalam jaringan berbasis IoT yang dapat terhubung ke sensor-sensor layaknya Arduino. Sedangkan ESP32 memiliki fungsi yang sama dengan ESP8266, perbedaannya modul ESP32 memiliki fungsi

tambahan berupa modul komunikasi Bluetooth Low Energy 4.0.

2. Sistem Monitoring Pintar Mitigasi Penularan COVID-19 Berbasis IoT

Sistem *monitoring* pintar mitigasi penularan COVID-19 adalah sistem *monitoring* berbasis teknologi IoT yang menggunakan beberapa sensor untuk merekam data keadaan di sekitar lokasi pemantauan dan memberikan rekomendasi kepada pengguna berdasarkan data yang diambil disekitar area pemantauan. Sistem monitoring ini terdiri dari beberapa sub unit sistem yang saling terintegrasi. Adapun unit-unit tersebut adalah sub unit *sensor node*, *web server*, *web service* (RESTful API), *database service* (MySQL database), sistem rekomendasi pengguna (*user recommendation system*) dan aplikasi *client* berbasis web yang dapat diakses di perangkat laptop, komputer desktop maupun perangkat *mobile* sebagai antarmuka sistem kepada pengguna akhir atau *end user*.

3. Parameter Kualitas Layanan Jaringan

Adapun parameter kualitas layanan jaringan yang diteliti untuk setiap skenario pengujian yang dilakukan adalah parameter *delay*, *packet loss*, *jitter* dan *throughput*. Nilai *delay* dapat diukur melalui rumus berikut:

$$delay = \frac{\text{waktu pengiriman paket pertama hingga paket terakhir}}{\text{jumlah keseluruhan paket}} \dots (1)$$

Standar yang digunakan untuk mengelompokkan nilai *delay* yaitu TIPHON (*Telecommunications and Internet Protocol Harmonization Over Network*) dengan pengelompokan nilai seperti tabel berikut (ETSI, 1999):

Tabel 1. Performansi Jaringan Berdasarkan Delay

Kategori Degradasi	Delay (ms)
Sangat Bagus	< 150
Bagus	150 s/d 300
Sedang	300 s/d 450
Buruk	> 450

Presentase *packet lost* dapat dihitung dengan rumus berikut:

$$packet_lost = \frac{\text{jumlah paket yang dikirim} - \text{jumlah paket yang diterima}}{\text{jumlah paket yang dikirim}} \dots (2)$$

Standar TIPHON untuk menentukan kriteria *packet lost* dapat dilihat pada tabel berikut:

Tabel 2. Performansi Jaringan Berdasarkan Packet Lost

Kategori Degradasi	Packet Lost (%)
Sangat Bagus	0
Bagus	3
Sedang	15
Buruk	25

Jitter dapat diukur menggunakan rumus berikut:

$$jitter = \frac{\text{total variasi delay}}{\text{paket yang diterima} - 1} \dots (3)$$

Terdapat empat klasifikasi nilai *jitter* menurut TIPHON seperti pada tabel berikut:

Tabel 3. Performansi Jaringan Berdasarkan Jitter

Kategori Degradasi	Jitter (ms)
Sangat Bagus	0
Bagus	0 s/d 75
Sedang	76 s/d 125
Buruk	125 s/d 255

Terakhir adalah *throughput* atau kecepatan transfer data yang diukur dalam satuan *bytes* per detik. *Throughput* dapat dihitung menggunakan rumus berikut:

$$throughput = \frac{\text{rata-rata data yang dikirim tiap detik}}{\text{waktu pengiriman paket pertama hingga paket terakhir}} \dots (4)$$

3. Metode Penelitian

Tahapan prosedur penelitian ini dimulai dengan mempersiapkan lingkungan pengujian dan skenario pengujian terlebih dahulu. Selanjutnya, mempersiapkan topologi jaringan sistem *monitoring* yang diuji, melakukan konfigurasi yang dibutuhkan untuk melakukan pengujian pada kaskas bantu *wireshark*, melakukan analisis paket data di dalam jaringan menggunakan parameter yang telah ditentukan dan terakhir menarik kesimpulan dari hasil pengujian yang telah diperoleh. Adapun prosedur penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Prosedur Penelitian

Untuk memberikan gambaran yang jelas mengenai prosedur penelitian akan diuraikan pada pembahasan berikut:

3.1. Lingkungan Pengujian

Daftar spesifikasi perangkat keras dan perangkat lunak yang digunakan untuk melakukan pengujian ini adalah:

1. Perangkat Keras (Hardware)

- a) Laptop dengan spesifikasi :

- CPU : Intel Core i3 CPU M 380 @2.53 GHz
 - Penyimpanan : Harddisk 500 GB
 - RAM : 6 GB
- b) *Sensor Node 1* dengan spesifikasi:
- Modul WiFi ESP8266
 - Sensor DHT-22
 - Sensor Suara KY-037
 - Modul GPS Neo-7M
 - LCD I2C 128x64
- c) *Sensor Node 2* dengan spesifikasi:
- Modul WiFi ESP32
 - Sensor DHT-22
 - Sensor Suara KY-037
 - Modul GPS SIM808
 - LCD I2C 128x64
- d) Mini PC (*Gateway*) dengan spesifikasi:
- Raspberry Pi 4 Model B
 - Penyimpanan : SD Card 32 GB
 - RAM : 8GB
- e) *Wireless Access Point + Router* (TP-Link 300 Mbps)

2. Perangkat Lunak (*Software*)

- a) Arduino IDE versi 1.8.13
- b) Wireshark versi 3.2.5

Data yang digunakan untuk menguji kualitas layanan jaringan adalah data sensor dalam format *JSON Object* yang dikirim dari *sensor node* ke unit *base station* menggunakan protokol HTTP. Format data sensor yang dikirim adalah seperti berikut:

```
{
  "device_id":<string_value>,
  "latitude":<float_value>,
  "longitude": <float_value>,
  "noise_level":<integer_value>,
  "temperature":<float_value>,
  "humidity":<float_value>
}
```

Device id adalah id perangkat yang digunakan sebagai tanda pengenal *sensor node*, *latitude* dan *longitude* adalah data posisi dimana sensor node dipasang dan data ini di peroleh melalui modul GPS. Selanjutnya *noise_level* atau tingkat kebisingan, diperoleh dari instalasi sensor suara yang di pasang pada unit *sensor node*. Begitu juga dengan data *temperature* dan *humidity* yang diperoleh melalui sensor DHT-22.

3.2. Skenario Pengujian

Terdapat dua skenario pengujian yang digunakan untuk menguji kualitas layanan jaringan yaitu:

1. Pengujian kualitas layanan jaringan di perangkat *sensor node* pertama (ESP8266) dan perangkat *sensor node* kedua (ESP32) yang dilakukan secara terpisah satu persatu.

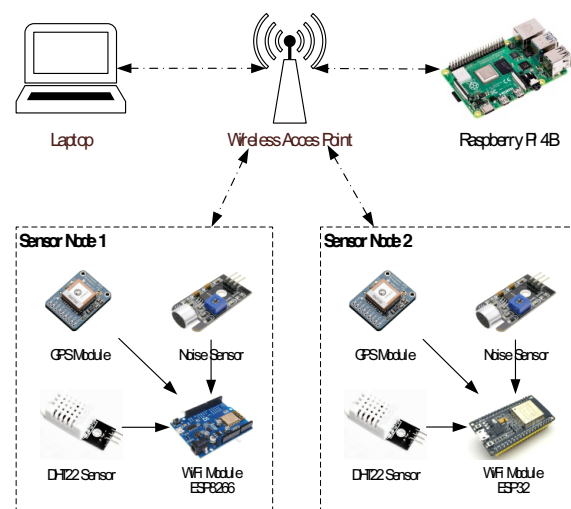
2. Pengujian kualitas layanan jaringan di kedua perangkat *sensor node* (ESP8266 dan ESP32) secara simultan pada satu waktu.

Jumlah percobaan yang dilakukan untuk masing-masing skenario pengujian berjumlah 10 kali percobaan, dengan lama waktu masing-masing percobaan adalah 100 detik.

Pengujian dilakukan dengan menggunakan bantuan kaskas bantu *wireshark* dalam mengamati lalu lintas jaringan antara *sensor node* ke unit *base station*.

3.3 Topologi Jaringan Sistem Monitoring Pintar

Topologi jaringan sistem monitoring pintar dapat dilihat pada gambar berikut:



Gambar 2. Topologi Jaringan Sistem Yang Diuji

Pada Gambar 2 terdapat 2 buah *sensor node* (ESP8266 dan ESP32) sistem monitoring yang akan diuji. Kedua *sensor node* ini akan melakukan *request* ke unit *base station* (Raspberry Pi 4B) untuk menambahkan data melalui sebuah *web service* (Restful API). Lalu lintas data akan di pantau menggunakan kaskas bantu *wireshark* yang terinstall di laptop. Semua perangkat baik laptop, Raspberry Pi 4B, *sensor node* ESP8266 dan *sensor node* ESP32 mengakses satu jaringan yang sama menggunakan *wireless access point*.

3.4. Konfigurasi Lalu Lintas Jaringan

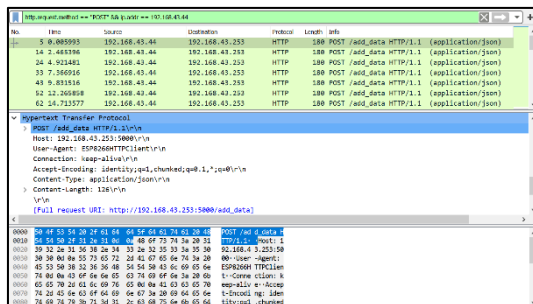
Konfigurasi lalu lintas jaringan dilakukan pada kaskas bantu analisis jaringan *wireshark*. *Wireshark* sejatinya akan memonitor setiap lalu lintas yang ada dalam suatu jaringan. Oleh karena itu perlu sedikit konfigurasi berupa aturan penyaringan untuk menampilkan lalu lintas jaringan hanya pada alamat ip dan metode *request* tertentu. Caranya adalah memasukkan aturan penyaringan pada *textfield display filter* seperti berikut:

```
http.request.method == "POST" && ip.addr == 192.168.43.44
```

Maka hasilnya akan menampilkan lalu lintas jaringan hanya pada alamat sumber ip 192.168.43.44 dengan metode *request* POST.

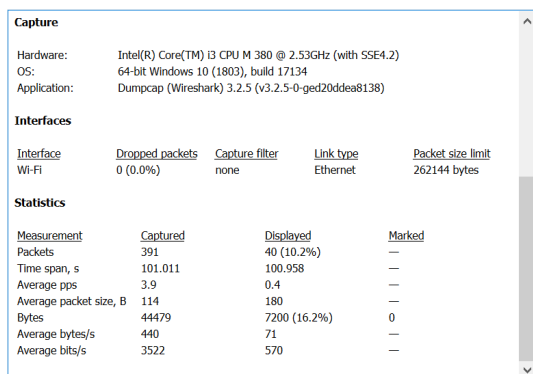
4. Hasil dan Pembahasan

Analisis kualitas layanan jaringan dilakukan pada ketiga skenario yang telah ditentukan. Lalu lintas jaringan harus disaring agar hanya menampilkan dan memproses data yang kita butuhkan untuk proses analisis. Proses *capture* data pada skenario pertama percobaan 1 perangkat *sensor node* ESP8266 dapat dilihat pada gambar berikut:



Gambar 3. Proses Pengujian Pada Sensor Node 1

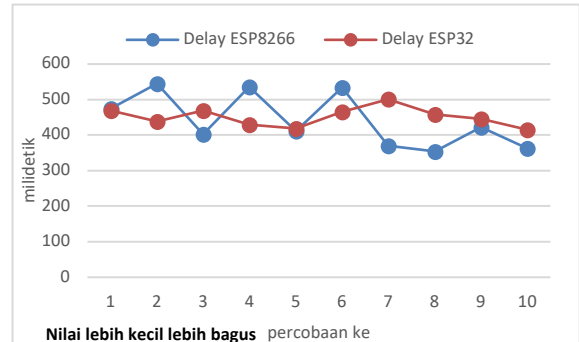
Pada Gambar 3 menunjukkan bahwa data telah disaring dan hanya diproses pada sumber ip 192.168.43.44 dan jenis metode *request* POST. Jenis data yang di terima menggunakan protokol komunikasi HTTP dengan *request* path: /add_data dan ukuran data yang dikirim sebesar 180 bytes.



Gambar 4. Hasil Capture Detail Percobaan 1

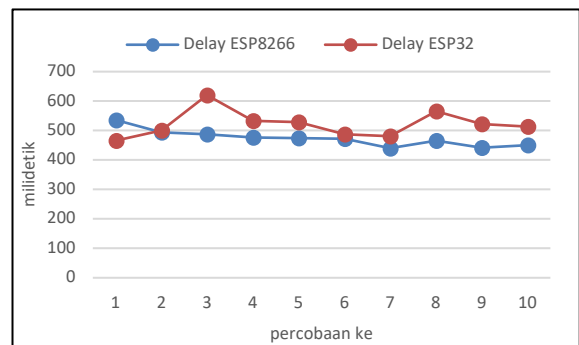
Informasi statistik yang diperlukan untuk menghitung *delay*, *packet lost*, *jitter* dan *throughput* antara lain adalah jumlah paket, rentang waktu, jumlah ukuran data (*bytes*) dan waktu *delay*. Hasil percobaan 1 pada Gambar 4 menunjukkan jumlah paket yang dikirim sebanyak 40 paket, dengan rentang waktu pengiriman 100,958 detik dan total ukuran data yang terkirim sebanyak 7200 *bytes*.

Waktu *delay* yang dihasilkan pada skenario pengujian pertama dapat dilihat pada grafik berikut:



Gambar 5. Grafik Delay Skenario 1

Rata-rata waktu *delay* pada pengujian skenario 1 untuk *sensor node* ESP8266 adalah sebesar 441 milidetik. Sedangkan waktu *delay* rata-rata pada *sensor node* ESP32 adalah sebesar 450 milidetik. Sehingga, jika kita lihat berdasarkan hasil grafik *delay* pada Gambar 5 menunjukkan waktu *delay* pengiriman paket data pada kedua sensor node hampir sama dan tidak menunjukkan perbedaan yang signifikan (hanya terkait 9 milidetik lebih unggul perangkat ESP8266) dan termasuk pada kategori sedang jika dikategorikan menurut standar TIPHON. Sedangkan pada skenario pengujian kedua, dimana dua *sensor node* mengirim data ke *base station* secara simultan pada satu waktu menunjukkan hasil pengujian seperti grafik berikut:



Gambar 6. Grafik Delay Skenario 2

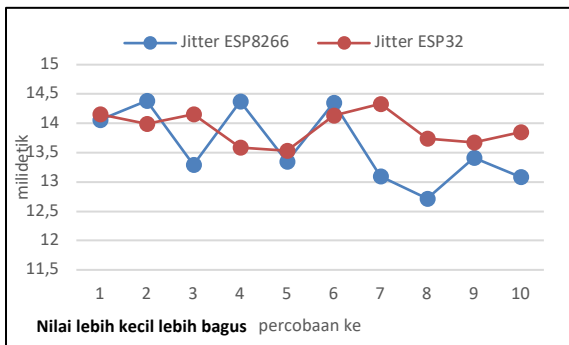
Pada Gambar 6 diperoleh rata-rata waktu *delay* untuk *sensor node* ESP8266 adalah sebesar 473,68 milidetik dan waktu *delay* untuk *sensor node* ESP32 adalah sebesar 521,74 milidetik. Menurut rata-rata waktu *delay* yang diperoleh, menunjukkan waktu *delay* untuk kedua *sensor node* berada pada kategori buruk jika *sensor node* bekerja secara simultan dan *delay* yang dihasilkan juga tidak lebih baik dari skenario pengujian pertama. Analisis *packet lost* pada kedua skenario pengujian dapat dilihat pada tabel berikut:

Tabel 4. Hasil Pengujian *Packet Lost*

Perobaan ke	Skenario 1		Skenario 2	
	ESP8266 (%)	ESP32 (%)	ESP8266 (%)	ESP32 (%)
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0

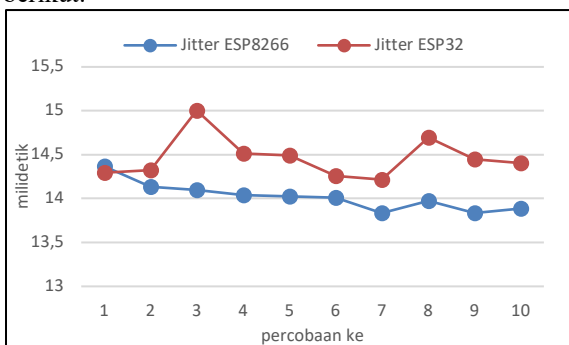
Berdasarkan hasil pengujian *packet lost* pada Tabel 4 diatas diperoleh rata-rata *packet lost* pada kedua skenario pengujian yaitu sebesar 0% atau berada pada kategori sangat bagus. Hal ini dikarenakan skalabilitas pengujian masih kecil, hanya menggunakan 2 perangkat *sensor node*, sehingga tidak mewakili jika pengujian dengan skalabilitas yang besar.

Berikutnya adalah waktu *jitter*, semakin kecil nilai *jitter* yang diperoleh maka akan semakin baik kualitas layanan jaringan. Hasil pengujian pada skenario pertama dari parameter *jitter* dapat dilihat pada grafik berikut:



Gambar 7. Grafik *Jitter* Skenario 1

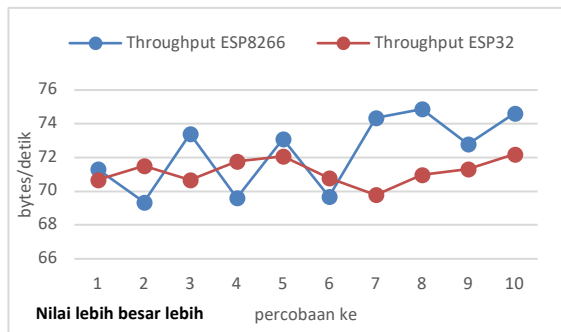
Berdasarkan grafik pada Gambar 7 diperoleh hasil rata-rata waktu *jitter* pada *sensor node* ESP8266 adalah sebesar 13,61 milidetik dan waktu *delay* untuk *sensor node* ESP32 adalah sebesar 13,91 milidetik. Waktu *jitter* yang di hasilkan pada kedua *sensor node* ini termasuk dalam kategori bagus. Selanjutnya untuk pengukuran waktu *jitter* pada skenario pengujian kedua dapat dilihat pada grafik berikut:



Gambar 8. Grafik *Jitter* Skenario 2

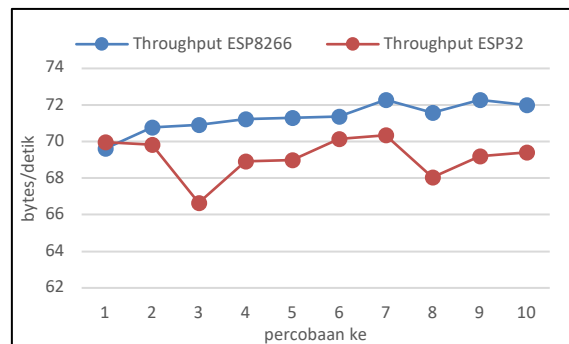
Rata-rata waktu *jitter* untuk *sensor node* ESP8266 adalah sebesar 14,02 milidetik dan *sensor node* ESP32 adalah sebesar 14,46 milidetik. Rata-rata waktu pada kedua *sensor node* ini termasuk kedalam kategori bagus. Hal ini juga menunjukkan bahwa hasil rata-rata waktu *jitter* pada skenario pertama lebih baik karena pengujian kedua sensor tidak diuji secara bersamaan.

Aspek terakhir yang dianalisis adalah nilai kecepatan pengiriman data (*throughput*). Adapun rata-rata nilai *throughput* yang dihasilkan pada skenario pengujian pertama dapat dilihat pada grafik berikut:



Gambar 9. Grafik *Throughput* Skenario 1

Throughput pada *sensor node* ESP8266 yaitu sebesar 72,31 bytes/detik dan pada *sensor node* ESP32 memiliki nilai rata-rata *throughput* sebesar 71,18 bytes/detik. Hal ini menunjukkan bahwa *throughput sensor node* ESP8266 lebih baik daripada *sensor node* ESP32. Kemudian untuk hasil nilai *throughput* pada skenario pengujian kedua dapat dilihat pada grafik berikut:



Gambar 10. Grafik *Throughput* Skenario 2

Dari hasil pengujian *throughput* skenario kedua pada Gambar 10 diperoleh rata-rata *throughput* untuk perangkat *sensor node* ESP8266 adalah sebesar 71,33 bytes/detik dan untuk perangkat *sensor node* ESP32 adalah sebesar 69,15 bytes/detik. Jika di bandingkan dengan rata-rata pengujian *throughput* pada skenario pertama maka nilai *throughput* pada skenario pertama lebih baik.

5. Kesimpulan dan Saran

5.1 Kesimpulan

Uji coba dilakukan dengan 10 kali percobaan, masing-masing percobaan diberikan durasi 100 detik dan paket yang dikirim sebanyak 40 paket dengan total jumlah ukuran data yang terkirim sebesar 7200 bytes. Pada skenario pertama menunjukkan bahwa perangkat *sensor node* ESP8266 memiliki *delay* 9 milidetik lebih baik dibandingkan dengan perangkat *sensor node* ESP32. Sedangkan untuk nilai *jitter* pada perangkat *sensor node* ESP8266 lebih baik 0.3 milidetik dan *throughput* lebih cepat 1.13 bytes/detik. Pada percobaan skenario kedua juga menunjukkan bahwa perangkat *sensor node* ESP8266 memiliki *delay* 48.06 milidetik lebih baik dibandingkan dengan perangkat ESP32. Sedangkan untuk nilai *jitter* pada perangkat *sensor node* ESP8266 lebih baik 0.44 milidetik dan *throughput* lebih cepat 2.18 bytes/detik. Sedangkan rata-rata *packet lost* dari kedua skenario percobaan yaitu sebesar 0% pada kedua perangkat *sensor node*.

5.2 Saran

Perangkat ESP8266 lebih baik dari ESP32 yang dapat digunakan untuk mengirim data sensor menggunakan protokol HTTP pada jaringan berbasis IoT.

Ucapan Terima Kasih

Peneliti mengucapkan terimakasih kepada Pusat Penelitian dan Pengabdian Masyarakat Politeknik Negeri Bengkalis yang telah mendukung dan membiayai penelitian ini.

Daftar Pustaka

- Bai, L., Yang, D., Wang, X., Tong, L., Zhu, X., Zhong, N., Bai, C., Powell, C. A., Chen, R., Zhou, J., Song, Y., Zhou, X., Zhu, H., Han, B., Li, Q., Shi, G., Li, S., Wang, C., Qiu, Z., ... Tan, F. (2020). Chinese experts' consensus on the Internet of Things-aided diagnosis and treatment of coronavirus disease 2019 (COVID-19). *Clinical EHealth*, 3, 7–15. <https://doi.org/10.1016/j.ceh.2020.03.001>
- Bhaddurgatte, R. C., & BP, V. K. (2015). A Review: QoS Architecture and Implementations in IoT Environment. *Research & Reviews: Journal of Engineering and Technology*, 2016, 6–12. <http://www.rroj.com/open-access/a-review-qos-architecture-and-implementations-in-iot-environment.php?aid=63427>
- Enda, D., & Subandri, M. A. (2020). PERANCANGAN SISTEM MONITORING PINTAR UNTUK MITIGASI PENULARAN COVID-19 DI AREA PADAT PUBLIK BERBASIS IOT. *Teknik Informatika, Politeknik Negeri Bengkalis, Jl. Bathin Alam, Bengkalis, 28711 PENDAHULUAN*
- Berbagai cara bisa dilakukan sebagai langkah mitig. 6(1), 861–868. <https://proceeding.isas.or.id/index.php/sentrino/article/view/560/239>
- ETSI. (1999). Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); General aspects of Quality of Service (QoS). In *Etsi Tr 101 329 V2.1.1* (Vol. 1). http://www.etsi.org/deliver/etsi_tr/101300_101399/101329/02.01.01_60/tr_101329v020101p.pdf
- Hasanul Fahmi. (2018). Analisis Qos (Quality of Service) Pengukuran Delay, Jitter, Packet Lost Dan Throughput Untuk Mendapatkan Kualitas Kerja Radio Streaming Yang Baik. *Jurnal Teknologi Informasi Dan Komunikasi*, 7(2), 98–105.
- Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P., & Sikdar, B. (2019). A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures. *IEEE Access*, 7, 82721–82743. <https://doi.org/10.1109/ACCESS.2019.2924045>
- Isha, Sharma, A., & Revathi, M. (2018). Automated API Testing. *Proceedings of the 3rd International Conference on Inventive Computation Technologies, ICICT 2018*, 788–791. <https://doi.org/10.1109/ICICT43934.2018.9034254>
- Kumar, K., Kumar, N., & Shah, R. (2020). Role of IoT to avoid spreading of COVID-19. *International Journal of Intelligent Networks*, 1(July), 32–35. <https://doi.org/10.1016/j.ijin.2020.05.002>
- Mohammed, M. N., Syamsudin, H., Al-Zubaidi, S., Sairah, A. K., Ramli, R., & Yusuf, E. (2020). Novel covid-19 detection and diagnosis system using iot based smart helmet. *International Journal of Psychosocial Rehabilitation*, 24(7), 2296–2303. <https://doi.org/10.37200/IJPR/V24I7/PR270221>
- Novianti, T., & Widiantoro, A. (2017). Analisa QOS (Quality of Services) pada Implementasi IPV4 dan IPV6 dengan Teknik Tunneling. *Rekayasa*, 9(2), 76. <https://doi.org/10.21107/rys.v9i2.3343>
- Singh, R. P., Javaid, M., Haleem, A., & Suman, R. (2020). Internet of things (IoT) applications to fight against COVID-19 pandemic. *Diabetes and Metabolic Syndrome: Clinical Research and Reviews*, 14(4), 521–524. <https://doi.org/10.1016/j.dsx.2020.04.041>
- Tabbasum, M., Tikoicina, K. M., & Huda, E. (2018). Comparative Analysis of Queuing Algorithms and QoS Effects on the IoT Network Traffic. *8th IEEE International Conference on Control*

- System, Computing and Engineering (ICCSCE 2018)*, 6, 3–7.
- Vaishya, R., Javaid, M., Khan, I. H., & Haleem, A. (2020). Artificial Intelligence (AI) applications for COVID-19 pandemic. *Diabetes and Metabolic Syndrome: Clinical Research and Reviews*, 14(4), 337–339. <https://doi.org/10.1016/j.dsx.2020.04.012>
- Villamil, S., Hernández, C., & Tarazona, G. (2020). An overview of internet of things. *Telkomnika (Telecommunication Computing Electronics and Control)*, 18(5), 2320–2327. <https://doi.org/10.12928/TELKOMNIKA.v18i5.15911>
- White, G., Nallur, V., & Clarke, S. (2017). Quality of service approaches in IoT: A systematic mapping. *Journal of Systems and Software*, 132, 186–203. <https://doi.org/10.1016/j.jss.2017.05.125>
- Windryani, N. P., K, N. B. A., & Mayasari, R. (2019). Analisa Perbandingan Protokol Mqtt Dengan Http Pada Iot Platform Patriot. *E-Proceeding of Engineering: Vol.6, No.2 Agustus 2019*, 6(2), 3192–3199. <https://librarye proceeding.telkomuniversity.ac.id/index.php/engineering/article/view/9468>