

Implementasi dan Analisis Performa Sistem *Monitoring* Suhu dan Kelembaban Kondisi Ruang *Server* pada Jaringan Berbasis Lora

Siti Zubaidah Effendi¹, Unan Yusmaniar Oktiawati^{1,*}

¹Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada;
siti.zubaidah.e@mail.ugm.ac.id

*Korespondensi: unan_yusmaniar@ugm.ac.id;

Abstract – A server room is a room used to store applications, data and network devices such as routers. The server room is the data center of a company or institution in which there are applications and databases that store all important and valuable information for the company or institution concerned, therefore the server room must always be in good temperature conditions because the devices work for 24 hours no stopping. If there is a significant increase in temperature, it can cause system performance to be disrupted or hardware damage occurs. The cooler in the server room is not optimal because the cooler is often constrained by frequent power outages. From these problems, a solution is needed to be able to monitor the system remotely so that changes in room temperature can be seen in real time. The Internet of Things (IoT) is the solution to this problem, by using LoRa, which then sends the data to the Telegram application that has been installed on the smartphone so that real-time temperature changes can be obtained.

Keywords : internet of thing, quality of service, temperature and humidity, raspberry pi, long range

Intisari – Ruang *server* adalah ruangan yang digunakan untuk menyimpan aplikasi, data dan perangkat jaringan seperti *router*. Ruang *server* merupakan pusat data suatu perusahaan atau lembaga yang di dalamnya terdapat aplikasi dan basis data yang menyimpan segala informasi penting dan bernilai bagi perusahaan atau lembaga yang bersangkutan, oleh karena itu ruang *server* harus selalu dalam kondisi suhu yang baik karena perangkat yang bekerja selama 24 jam tanpa henti. Apabila terjadi kenaikan suhu yang signifikan, dapat menyebabkan kinerja sistem menjadi terganggu atau terjadi kerusakan pada perangkat keras. Pendingin yang terdapat di ruang *server* dirasa belum maksimal karena pendingin tersebut sering kali terkendala dengan aliran listrik yang sering padam. Dari permasalahan tersebut diperlukan solusi untuk dapat memantau sistem dari jarak jauh sehingga dapat diketahui perubahan suhu ruang secara realtime. *Internet of Things (IoT)* menjadi solusi dari permasalahan ini, dengan menggunakan *LoRa* yang selanjutnya data akan dikirimkan ke aplikasi *Telegram* yang sudah dipasang pada *smartphone* sehingga dapat diperoleh perubahan suhu secara *real-time*.

Kata kunci : internet of things, quality of service, temperature and humidity, raspberry pi, long range

I. PENDAHULUAN

Internet of Things atau biasa dikenal dengan *IoT* merupakan salah satu perkembangan teknologi internet dan kemungkinan besar akan menjadi tren di masa depan. *Internet of Things (IoT)* memungkinkan pengguna untuk mengelola dan mengoptimalkan elektronik dan peralatan listrik yang menggunakan internet. Salah satu penerapan *Internet of Things* yaitu sistem *monitoring* atau *controlling* yang menggunakan sensor dan aktuator pada sebuah lingkungan dengan kondisi tertentu [1].

Ruang *server* adalah ruangan yang digunakan untuk menyimpan aplikasi, data dan perangkat jaringan seperti *router* dan *hub*, sehingga ruang *server* merupakan aset penting bagi suatu perusahaan atau lembaga yang menerapkan teknologi informasi sebagai penunjang dalam kegiatan sehari-hari. Hal ini karena pada ruang *server* terdapat aplikasi dan basis data yang menyimpan segala informasi penting yang bernilai bagi perusahaan atau lembaga yang bersangkutan, oleh karena itu ruangan *server* harus selalu dalam kondisi baik. Selain itu perangkat yang berada di ruang *server* harus bekerja selama 24 jam tanpa henti, sehingga perlu ditunjang dengan *monitoring* untuk memastikan perangkat bekerja dengan baik. Karena sifatnya yang tak henti-hentinya, sebuah pusat data membutuhkan pemantauan yang sangat baik dan perlindungan data dari kerusakan yang disebabkan dari kecelakaan ke infrastruktur jaringan.

Ancaman lingkungan paling umum terhadap ruang *server* adalah suhu, kelembaban, kebocoran air, human error dan pemadaman listrik. Banyak ancaman yang muncul seperti suhu dan kelembaban yang mempersulit pemantauan lingkungan pada ruang *server*. Suhu ruang adalah faktor

penunjang yang berpengaruh besar pada kinerja *server* dan perangkat jaringan lainnya karena jika suhu tidak mencapai rentang 18°C – 23°C maka kinerja kipas *server* akan bekerja ekstra untuk menstabilkan suhu *server*. Suhu yang terlalu tinggi mengakibatkan komponen pada perangkat seperti *harddisk* cepat rusak, sedangkan suhu yang terlalu rendah mengakibatkan udara mengalami pengembunan dan kelembaban menjadi tinggi sehingga terjadi hubungan arus pendek pada perangkat.

Berdasarkan permasalahan tersebut, penulis mengusulkan sebuah purwarupa sistem berbasis mikrokontroler yang dapat memantau suhu dan kelembaban. Sistem ini nantinya akan mengirimkan notifikasi pada *Telegram* ketika terjadi kenaikan atau penurunan suhu pada ruang *server*.

II. TINJAUAN PUSTAKA

Internet of Things atau yang dikenal dengan istilah *IoT* merupakan teknologi yang memungkinkan penggunaanya untuk mengelola dan mengoptimalkan elektronik dan peralatan listrik yang menggunakan internet. *Internet of Things* adalah segala sesuatu atau perangkat elektronik yang dapat berinteraksi secara langsung dengan pengguna yang digunakan untuk kebutuhan *monitoring* atau mengendalikan perangkat tersebut melalui internet.

Sistem *monitoring* adalah suatu sistem yang melakukan proses pemantauan secara terus menerus [2]. Sistem *monitoring* dibutuhkan dalam proses pemantauan keadaan suatu objek yang diamati guna mendapatkan informasi yang tepat waktu. *Monitoring* infrastruktur jaringan merupakan suatu hal yang sangat penting karena perangkat jaringan baik itu *server*, *router*, maupun perangkat jaringan lainnya dapat tetap terjaga kinerjanya. Pada penelitian yang dilakukan oleh [3] dibuat sebuah aplikasi *monitoring* yang dirancang dengan

menggunakan *Bot Telegram* dan diintegrasikan dengan *server OpenNMS* dan *Router Mikrotik* yang bertujuan agar pengguna dapat memantau *server* maupun perangkat jaringan dengan mudah.

Perangkat pada ruang *server* memiliki fungsi yang penting sehingga tidak boleh mengalami gangguan. Beberapa penyebab kinerja perangkat di ruang *server* terganggu adalah *overheat* dan kelembaban ruangan *server*. Kelembaban yang tinggi pada ruangan akan mempengaruhi usia perangkat [4]. Pada tahun 2019 dilakukan sebuah penelitian dengan membuat sebuah rancang bangun pengatur suhu dan kelembaban ruang *server* berbasis *IoT* dengan *DHT 11* sebagai sensor suhu dan *Arduino Uno* sebagai mikrokontrolernya, dan *Ethernet Shield* agar dapat terhubung pada internet. Dari penelitian ini dihasilkan bahwa sensor dapat memberikan informasi keadaan suhu di dalam ruangan dengan baik dan mikrokontroler yang menerima informasi tersebut akan mengatur sistem pendingin jika suhu dalam ruangan terlalu rendah atau tinggi [5].

Pada tahun 2019 juga dilakukan penelitian dengan membuat rancang bangun sistem *monitoring* suhu dan kelembaban untuk mendeteksi perubahan suhu dan kelembaban pada ruangan dengan menggunakan *NodeMCU ESP8266* sebagai kontrol utamanya, *DHT 11* sebagai sensor pendeteksi suhu dan kelembaban, selanjutnya data akan dikirim ke aplikasi *Telegram* yang sudah ter-install pada *smartphone*. Pada penelitian ini juga ditambahkan sensor gas *MQ-2* untuk mendeteksi adanya asap, jika lebih besar dari 500 *ppm* maka *buzzer* akan berbunyi. Dari penelitian ini didapatkan hasil bahwa pengambilan data suhu *DHT 11* dengan Termometer hampir sempurna karena perbedaan hasil tidak lebih dari 2°C atau masih dalam batas normal sesuai dengan ketentuan yang berlaku 20°C-25°C. Dengan kesalahan pembacaan 1°C-2°C. Sedangkan pengukuran kelembaban pada ruangan *server* menggunakan *DHT 11* dengan *Hygrometer* sebagai pembanding mendapatkan hasil selisih 1%-6%. Data hasil pembacaan *DHT 11* dan *MQ-2* dikirimkan ke *Telegram* menggunakan *NodeMCU ESP866* yang dapat terhubung dengan *wi-fi* dengan jarak maksimal dalam ruang *server* dengan jarak 20 meter [6].

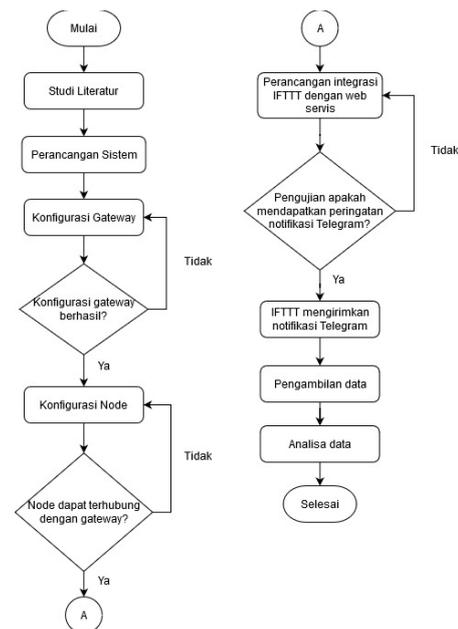
Salah satu perkembangan *IoT* adalah pengaplikasian *LoRa (Long Range)*. Pada *IoT*, *LoRa* digunakan dalam komunikasi M2M (*Machine to Machine*), contohnya adalah untuk pengembangan *Smart City*, dengan adanya *LoRa* sensor-sensor dapat berinteraksi langsung dengan manusia di mana saja dan kapan saja. Salah satu keunggulan teknologi *LoRa* adalah penggunaan daya rendah dan memiliki jangkauan komunikasi yang luas yaitu lebih dari 2 km. Namun pengiriman data ini tidak dapat langsung menuju ke *server*, sehingga diperlukan *gateway* sebagai penghubung antar perangkat di *node sensors* dengan *server*. Pada tahun 2019 dilakukan sebuah penelitian dengan melakukan pengembangan sistem *gateway* agar dapat menghubungkan komunikasi antara *node sensors* dengan *server* menggunakan modul komunikasi *LoRa* dan protokol *MQTT*. Hasil pengujian kinerja *successful rate gateway* dengan menggunakan variabel jarak, besar data, dan interval yang berbeda menunjukkan bahwa pada jarak 400 meter kinerja *gateway* dalam menerima dan meneruskan data ke pusat data lebih baik daripada jarak 200 meter [7].

Dalam penerapan *IoT* dibutuhkan suatu jaringan komunikasi yang sesuai dengan kebutuhan sistem. Salah satu

protokol yang sesuai dengan penerapan konsep *IoT* adalah protokol *Message Queue Telemetry Transport (MQTT)* yang sering digunakan dalam berbagai sistem yang menggunakan konsep *IoT* salah satunya sistem *monitoring*. Pada tahun 2018 dilakukan sebuah penelitian oleh [8] yang mengimplementasikan konsep *Internet of Things* pada sistem *monitoring* banjir menggunakan protokol *MQTT*. Berdasarkan hasil pengujian, diperoleh kesimpulan bahwa protokol *MQTT* dapat digunakan pada sistem *monitoring* banjir. Tingkat akurasi yang didapat dari pengujian sistem ini adalah 97,801% dengan standar deviasi yang diperoleh sebesar ± 0.0309 cm.

III. METODE PENELITIAN

Tahapan penelitian yang dilakukan dalam proyek akhir ini yang pertama adalah dilakukan studi literatur dari beberapa jurnal dan buku, kemudian mulai dilakukan perancangan dan analisis sistem. Selanjutnya dilakukan persiapan untuk perangkat keras dan perangkat lunak yang dibutuhkan. Dilanjutkan dengan konfigurasi *gateway* kemudian konfigurasi pada *node*. Kemudian dilakukan konfigurasi agar *node* dan *gateway* dapat terhubung kemudian dilanjutkan perancangan integrasi antara *IFTTT* dengan *web service*. Gambar 1 merupakan diagram alir metode penelitian yang dilakukan oleh penulis.



Gambar 1. Bagan Alir Proyek Akhir

A. Alat dan Bahan

Dalam melaksanakan penelitian ini, dibutuhkan perangkat keras dan perangkat lunak guna menunjang penelitian yang akan dilakukan. Adapun perangkat tersebut diantaranya sebagai berikut:

1) Perangkat keras

- 2 Unit *Raspberry Pi 3 B+*
- 2 Unit *LoRa Dragino GPS/HAT*
- 1 Sensor Suhu *DHT11*
- Kabel *Jumper*
- *Android*

- PC/Laptop
- Power Supply/Adaptor 5V

2) Perangkat lunak

- Raspbian
- Python
- PosgreSQL
- Telegram

B. Pengujian dan Pengambilan Data

Pengujian *Quality of Service (QoS)* dilakukan untuk menilai kualitas pengiriman data dari *node* ke *gateway* hingga sampai ke pengguna untuk membuktikan fitur pemantauan bersifat *real-time*. Parameter yang diuji adalah *Delay*, *Packet Loss* dan *Packet Delivery*.

Nilai parameter-parameter *QoS* didapat dari 2 skenario yang akan dijalankan dengan 10 perulangan percobaan pada tiap skenario sehingga total terdapat 20 percobaan. Skenario berupa pengiriman data dengan interval waktu yang berbeda-beda pada jaringan yang sama. Skenario 1 dilakukan pengiriman data sebanyak 100 paket data dan antara satu paket data dengan paket data berikutnya diberikan jeda waktu selama 5 detik, sehingga dapat diasumsikan bahwa pengiriman data dilakukan setiap 5 detik. Skenario 2 dilakukan dengan mengirim paket data sebanyak 100 paket data dan diberikan jeda waktu selama 10 detik antara pengiriman satu paket data dengan paket data berikutnya sehingga dapat dikatakan bahwa dilakukan pengiriman paket data setiap 10 detik. Berikut Tabel 1 yang menunjukkan skenario pengujian.

Tabel 1. Skenario Pengujian

Skenario	Eksperimen
Skenario 1	Pengiriman data per 5 detik
Skenario 2	Pengiriman data per 10 detik

1) Pengambilan Nilai *Delay*

Nilai *delay* diperoleh dari perhitungan waktu paket data ketika diterima dikurangi dengan waktu paket data dikirimkan atau untuk lebih jelasnya dengan menggunakan persamaan berikut:

$$Delay = Waktu_{penerimaan\ paket} - Waktu_{pengiriman\ paket} \tag{1}$$

Waktu pengirim dan penerimaan paket data diketahui dari hasil rekaman trafik data yang telah dibuat dan ditampilkan pada *LoRa* yang kemudian diolah pada *Excel* hingga menjadi seperti pada Gambar 2.

	A	B	C	D	E	F	G	H	I
	No.	Gateway	Node	Gateway	Node	Gateway	Node	Gateway	Node
1	1	0.370888	Event EV_TXCOMPLETE	0.397989	Event EV_TXCOMPLETE	0.379923	Event EV_TXCOMPLETE	0.3616	Event EV_TXCOMPLETE
2	2	0.352567	Event EV_TXCOMPLETE	0.333745	Event EV_TXCOMPLETE	0.353383	Event EV_TXCOMPLETE	0.387393	Event EV_TXCOMPLETE
3	3	0.353897	Event EV_TXCOMPLETE	0.381344	Event EV_TXCOMPLETE	0.355041	Event EV_TXCOMPLETE	0.357209	Event EV_TXCOMPLETE
4	4	0.362058	Event EV_TXCOMPLETE	0.358937	Event EV_TXCOMPLETE	0.340041	Event EV_TXCOMPLETE	0.366125	Event EV_TXCOMPLETE
5	5	0.3599	Event EV_TXCOMPLETE	0.367385	Event EV_TXCOMPLETE	0.349536	Event EV_TXCOMPLETE	0.358334	Event EV_TXCOMPLETE

Gambar 2. Nilai *Delay*

2) Pengambilan Nilai *Packet Loss*

Packet Loss yang diambil pada pengujian ini merupakan paket data yang hilang ketika proses pengiriman data.

3) Pengambilan Nilai *Packet Delivery*

Packet Delivery yang diambil pada pengujian ini adalah paket data yang dikirimkan dari *node* ke *gateway* dalam 100 data yang dikirimkan.

IV. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan purwarupa sistem pemantau suhu dan kelembaban ruang *server* yang terdiri dari dua buah *node* yaitu *node sensors* dan *node gateway*. *Node sensors* berfungsi untuk membaca suhu dan kelembaban pada rak *server*. *Node gateway* berfungsi untuk menerima hasil pembacaan dari *node sensors* yang selanjutnya akan dikirimkan ke *server*. Gambar 3 adalah tampilan *node gateway* yang komponen-komponennya sudah dirakit dan siap untuk digunakan. Sedangkan Gambar 4 merupakan *node sensors* yang berfungsi untuk membaca suhu dan kelembaban ruangan dimana pada *node* tersebut terdapat sensor *DHT 11* untuk membaca suhu dan kelembaban. *Node sensors* juga akan mengirimkan data ke *node gateway* ketika suhu dan kelembaban ruangan melebihi dari batas yang telah ditetapkan.



Gambar 3. *Node Gateway*



Gambar 4. *Node Sensors*

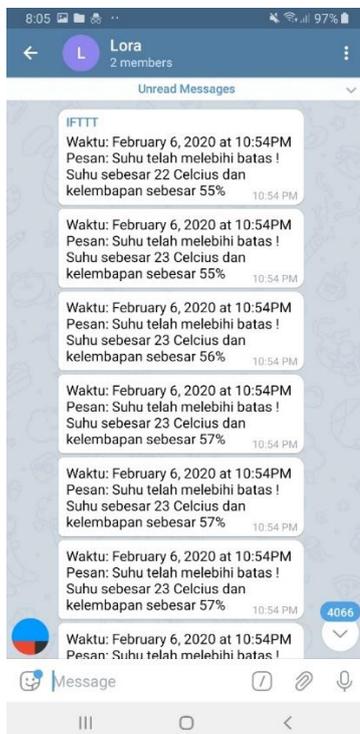
Hasil pembacaan suhu dan kelembaban yang melebihi batas yang telah ditentukan akan dikirimkan melalui notifikasi pada aplikasi *Telegram*.



Gambar 5. Telegram IFTTT yang menunjukkan suhu dan kelembapan melebihi batas yang telah ditentukan

A. Hasil Pengujian Sistem

Penulis melakukan pengujian sensor *DHT11* dengan variasi jeda waktu pengiriman data. Variasi jeda tersebut sebesar 5 detik dan 10 detik pada setiap percobaan. Setiap variasi jeda tersebut dilakukan sebanyak 10 kali percobaan. Sistem pemantauan suhu dan kelembapan yang telah dibuat sudah terintegrasi dengan sistem notifikasi melalui *Telegram*. Jika suhu dan kelembapan melebihi batas yang sudah ditentukan, maka akan mengirimkan notifikasi melalui *Telegram* seperti pada Gambar 6.



Gambar 6. Notifikasi Telegram

B. Hasil Pengujian Delay

Pengujian *delay* dilakukan di Gedung Sekip Unit IV tepatnya di ruang 128. Pengujian dilakukan dengan 2 skenario yaitu diberikan jeda 5 detik dan 10 detik. Dalam mencari nilai *delay*, tidak dapat dilakukan dengan satu kali percobaan saja. Oleh karena itu penulis melakukan 10 kali percobaan untuk masing-masing skenario yaitu 10 percobaan untuk skenario jeda 5 detik dan 10 percobaan pada skenario 10 detik. Setiap percobaan dilakukan pengiriman data sebanyak 100 data, setelah selesai melakukan percobaan dicari nilai rata-ratanya. Hal tersebut dilakukan agar data yang didapat *valid*. Rata-rata nilai *delay* dari hasil percobaan pada setiap skenario termasuk dalam kategori sangat bagus karena nilainya kurang dari 150 ms. Tabel 2 menunjukkan indeks dan kategori *delay* yang menentukan kualitas layanan menurut *Telecommunication Internet Protocol Harmonization Over Network (TIPHON)*.

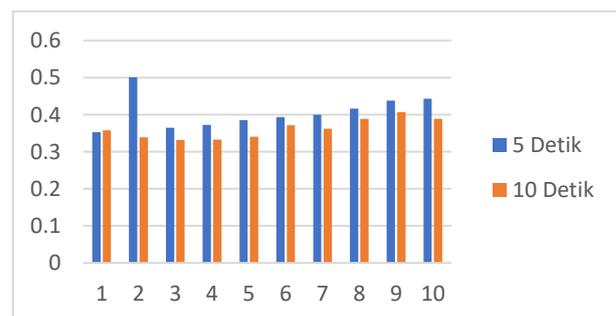
Tabel 2. Performa Jaringan Berdasarkan Delay

Kategori	Delay
Sangat Bagus	< 150ms
Bagus	150ms s/d 300ms
Sedang	300ms s/d 450ms
Jelek	>450 ms

Hasil pengukuran *delay* dapat dilihat pada Tabel 3 dan didapatkan nilai rata-rata setelah sepuluh kali percobaan, pada *delay* 5 detik didapat nilai rata-rata 0.4068275 dan pada *delay* 10 detik didapat nilai rata-rata 0.3621856.

Tabel 3. Hasil Pengukuran Delay

Percobaan Ke-n	Delay n (detik)	
	5	10
1	0.353082	0.357532
2	0.500792	0.338959
3	0.364898	0.331779
4	0.372503	0.333313
5	0.385375	0.340870
6	0.393357	0.371652
7	0.400360	0.362293
8	0.416657	0.389247
9	0.437990	0.407397
10	0.443261	0.388814
Rata-Rata	0.4068275	0.3621856



Gambar 7. Grafik Pengujian Delay

C. Hasil Pengujian *Packet Loss*

Dalam pengambilan data *packet loss* dilakukan dua skenario, skenario pertama dilakukan pengiriman data dari *node* menuju *gateway* sebanyak 100 data dan antara pengiriman satu data dengan data berikutnya diberikan jeda waktu selama 5 detik. Skenario kedua juga dilakukan pengiriman data dari *node* menuju *gateway* sebanyak 100 data dan antara pengiriman satu data dengan data berikutnya diberikan jeda waktu selama 10 detik. Pada dua skenario tersebut dilakukan sebanyak 10 percobaan pada setiap skenario sehingga total terdapat 20 percobaan dengan 10 percobaan pada skenario jeda 5 detik dan 10 percobaan pada skenario jeda 10 detik. Data *packet loss* didapat ketika *node* mengirimkan data ke *gateway* hingga muncul *Event EV_TXCOMPLETE* seperti pada Gambar 8 berikut

```
Event EV_TXCOMPLETE, time: 3264
[3e91612] (1579951205) Sat Jan 25 18:20:05 2020
65607194: freq=923200000
Event EV_TXCOMPLETE, time: 3283
[3ef30b6] (1579951225) Sat Jan 25 18:20:25 2020
66007225: freq=923200000
Event EV_TXCOMPLETE, time: 3304
[3f54b4b] (1579951245) Sat Jan 25 18:20:45 2020
```

Gambar 8. Hasil Pengiriman Data di *Node*

Setelah dilakukan masing-masing 10 percobaan di tiap skenario, *packet loss* yang didapat bernilai 0% pada setiap skenario atau dapat dikatakan bahwa tidak ada paket yang hilang selama pengiriman data berlangsung. Nilai 0% tersebut didapat dari persamaan berikut ini:

$$\text{Packet Loss Rasio (PLR) \%} = \frac{\text{Paket Data Hilang}}{\text{Paket Data Dikirim}} \times 100\%$$

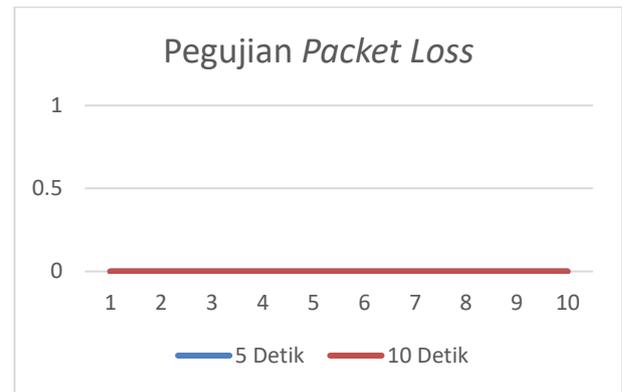
$$\text{Packet Loss Rasio (PLR) \%} = \frac{0}{100} \times 100\%$$

$$= 0\%$$

Hal ini dapat dilihat pada Tabel 4 berikut:

Percobaan Ke-n	<i>Packet Loss</i> Skenario n Detik	
	5	10
1	0%	0%
2	0%	0%
3	0%	0%
4	0%	0%
5	0%	0%
6	0%	0%
7	0%	0%
8	0%	0%
9	0%	0%
10	0%	0%

Dari Tabel 4 dapat digambarkan grafik seperti pada Gambar 9.



Gambar 9. Grafik Pengujian *Packet Loss*

Setelah dilakukan percobaan pada masing-masing skenario dan menghasilkan nilai *packet loss* 0% maka dapat disimpulkan bahwa *packet loss* untuk pengiriman data dari *node* menuju *gateway* termasuk kategori sangat bagus menurut *TIPHON*. Hal ini dapat dilihat pada Tabel 5 terdapat empat kategori penurunan performansi jaringan berdasarkan nilai *packet loss* menurut *TIPHON*.

Tabel 5. Performa Jaringan Berdasarkan *Packet Loss*

Kategori	<i>Packet Loss</i>
Sangat Bagus	0%
Bagus	3%
Sedang	15%
Jelek	>25%

D. Hasil Pengujian *Packet Delivery*

Packet Delivery yang diambil pada pengujian ini adalah paket data yang dikirimkan oleh *node sensors* dan berhasil sampai ke *node gateway* dalam satu kali pengiriman. Sama halnya dengan pengujian *packet loss*, dilakukan 2 skenario percobaan dengan masing-masing skenario dilakukan percobaan sebanyak 10 kali. Dalam satu kali percobaan dikirimkan paket data sebanyak 100 data. Perbedaan antara skenario satu dan skenario dua adalah interval waktu pengiriman data. Skenario satu diberikan jeda waktu pengiriman paket data antara satu paket data dengan paket data berikutnya selama 5 detik sedangkan skenario dua diberikan jeda waktu pengiriman selama 10 detik. Nilai *packet delivery* didapatkan dari persamaan:

$$\text{Packet Loss Rasio (PLR) \%} = \frac{\text{Paket Data Hilang}}{\text{Paket Data Dikirim}} \times 100\%$$

$$\text{Packet Loss Rasio (PLR) \%} = \frac{100}{100} \times 100\%$$

$$= 100\%$$

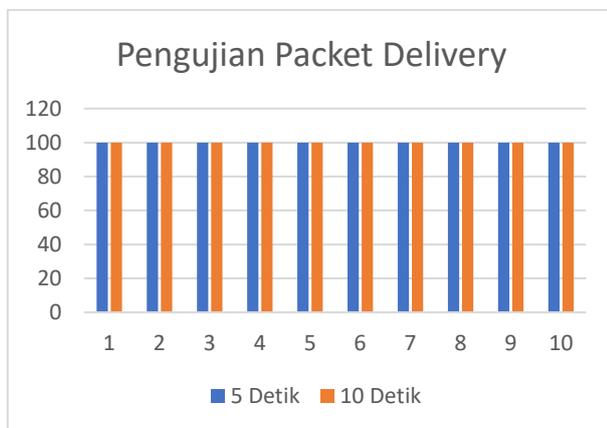
(3)

Hasil dari percobaan kedua skenario tersebut dapat dilihat pada Tabel 6 berikut:

Tabel 6. Hasil Pengukuran *Packet Delivery*

Percobaan Ke-n	<i>Packet Delivery</i> Skenario n	
	Detik	
	5	10
1	100%	100%
2	100%	100%
3	100%	100%
4	100%	100%
5	100%	100%
6	100%	100%
7	100%	100%
8	100%	100%
9	100%	100%
10	100%	100%

Dari Tabel 6 di atas kemudian dapat digambarkan grafik seperti pada Gambar 10 berikut:

Gambar 10. Grafik Pengujiian *Packet Delivery*

Dari hasil pengujian yang telah dilakukan didapat nilai *packet delivery* sebesar 100% atau dapat disimpulkan bahwa tidak ada paket yang hilang selama pengiriman data. Menurut *TIPHON* performansi jaringan berdasarkan nilai dari *packet delivery* dari dua skenario yang telah dijalankan dapat dikategorikan sebagai sangat bagus, hal ini dapat dilihat pada Tabel 7.

Tabel 7. Performa Jaringan Berdasarkan *Packet Delivery*

Kategori	<i>Packet Delivery</i>
Sangat Bagus	100%
Bagus	97%
Sedang	85%
Jelek	>75%

V. KESIMPULAN

Berdasarkan analisis pengujian yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

- Sistem yang dibuat berupa alat pendeteksi suhu dan kelembaban pada ruang *server* dengan menggunakan sensor *DHT 11*, ketika sensor mendeteksi adanya perubahan suhu yang terjadi, *Raspberry Pi* dan *LoRa* bekerja dengan mengirimkan notifikasi berupa pesan pada aplikasi *Telegram*.
- Sistem dapat mengintegrasikan sensor suhu dan kelembaban dengan mikrokontroler *Raspberry Pi* menggunakan *LoRa* secara *real time* sehingga ketika

terjadi perubahan suhu *LoRa node* mengirimkan paket kepada *LoRa gateway* yang selanjutnya akan diteruskan ke *server* sampai pada akhirnya terdapat pesan dari *Telegram* berupa notifikasi.

- Performa kinerja *LoRa* pada parameter *packet loss* dapat dikatakan sangat bagus dengan persentase *packet loss* penelitian 0%.
- Kinerja dari sistem yang telah diuji terdapat kendala jika dilakukan pengujian terhadap jarak antara *node sensors* dan *gateway*, kendala tersebut adalah *gateway* mampu menerima paket data yang dikirimkan oleh *node* pada satu hingga tiga percobaan pertama namun pada percobaan berikutnya *gateway* tidak menerima paket data yang dikirimkan.

Berikut adalah saran yang dapat digunakan untuk mengembangkan penelitian selanjutnya:

- Menerapkan protokol komunikasi nirkabel lain. Diharapkan kedepannya dapat dilakukan komparasi kinerja berbagai protokol yang beragam.
- Jumlah *gateway* yang digunakan pada penelitian ini hanyalah satu. Diharapkan untuk kedepannya dapat menggunakan jumlah *gateway* lebih dari satu.
- Menerapkan serangan untuk mengetahui efeknya terhadap kinerja *LoRa*.

DAFTAR PUSTAKA

- [1] H. Rochman, R. Primananda and H. Nurwasito, "Sistem Kendali Berbasis Mikrokontroler Menggunakan Protokol MQTT pada Smarthome," *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2017.
- [2] Mudjahidin and N. D. P. Putra, "Rancang Bangun Sistem Informasi Monitoring Perkembangan Proyek Berbasis Web Studi Kasus di Dinas Bina Marga dan Pemantusan," *Jurnal teknik Industri*, 2012.
- [3] R. J. Putra, N. P. Sastra and D. M. Wiharta, "Pengembangan Komunikasi Multikanal untuk Monitoring Infrastruktur Jaringan Berbasis Bot Telegram," *Jurnal SPEKTRUM*, 2018.
- [4] S. Taftazanie, A. B. Prasetijo and E. D. Widiyanto, "Aplikasi Pemantau Perangkat Jaringan Berbasis Web Menggunakan Protokol SNMP dan Notifikasi SMS," *Jurnal Teknologi dan Sistem Komputer*, 2017.
- [5] M. I. Z., J. Endri and Sarjana, "Rancang Bangun Pengatur Suhu dan Kelembaban Ruang Server Berbasis IoT," *Prosiding SENIATI*, 2019.
- [6] G. Santoso, S. Kristiyana, S. Hani and A. M. Mujahidin, "Rancang Bangun Sistem Monitoring Suhu dan Kelembaban pada Ruang Server Berbasis IoT (Internet of Things)," *Jurnal Teknologi Technoscientia*, 2019.
- [7] H. Arijuddin, A. Bhawiyuga and K. Amron, "Pengembangan Sistem Perantara Pengiriman Data Menggunakan Modul Komunikasi LoRa dan Protokol MQTT Pada Wireless Sensor Network," *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2019.
- [8] C. P. R. & A. K. Hasiholan, "Implementasi Konsep Internet of Things pada Sistem Monitoring Banjir menggunakan Protokol MQTT," *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 2018.