

**IMPLEMENTASI DAN ANALISIS PERFORMANSI SISTEM MONITORING
BENDUNGAN DENGAN MEDIA KOMUNIKASI INSTANT MESSAGING LINE
BERBASIS INTERNET OF THINGS**

*IMPLEMENTATION AND ANALYSIS OF DAM MONITORING SYSTEM
PERFORMANCE USING INSTANT MESSAGING LINE AS THE
COMMUNICATION MEDIA BASED ON INTERNET OF THINGS*

Gendhis Azzukhruf Dynastuti¹, Dr. Ir. Rendy Munadi., M.T.², Gustommy Bisono., S.T.,
M.T.³

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

¹ gazzukhrufd@gmail.com, ² rendymunadi@telkomuniversity.ac.id,

³ tommy.bisono@gmail.com

Abstrak

Perkembangan teknologi setiap waktunya berkembang sangat pesat, *Internet of Things* (IoT) hadir untuk memudahkan dan memenuhi kebutuhan manusia dalam melakukan kegiatannya. Dalam penelitian ini, dilakukan sebuah perancangan sistem monitoring bendungan menggunakan *microcontroller* yang telah terintegrasi dengan *chip wireless*. Sistem ini menciptakan teknologi *remote controlling* dengan jaringan internet sebagai penghubung antara *server* dan *client* menggunakan protokol MQTT. Sistem ini di desain untuk petugas bendungan dalam memonitor ketinggian *level* air menggunakan media komunikasi *instant messaging* yaitu berupa notifikasi. Dari hasil pengujian Sistem monitoring bendungan dapat berfungsi dengan baik. Serta *user* berhasil mendapatkan informasi mengenai ketinggian air dari aplikasi pesan instan. Dari pengujian dan analisis waktu pengukuran sensor menunjukkan waktu pengukuran pada jarak semakin tinggi dengan *error rate* 2% pada jarak terjauh, apabila jarak semakin jauh. Analisa *delay* pengiriman MQTT menunjukkan berdasarkan data *delay* rata-rata status berbahaya mengalami *delay* paling besar yaitu 448744,9608 microsecond, sedangkan nilai rata-rata *throughput* dari status siaga memiliki nilai yang paling tinggi yaitu 160,1392157 bps. *Memory Usage* dari Heroku *server* dengan maksimum penggunaan *memory* 23.1 MB. *Response time* pada Heroku *server* paling lama adalah 991 ms, sedangkan rata-ratanya adalah 863 ms dan yang tercepat adalah 703 ms. Pada pengujian dan Analisa keseluruhan sistem ini didapat rata-rata *delay* sebesar 220.377 *millisecond*, dimana dapat dikategorikan *Good* menurut standar ITU-T G.1010 dengan rentang waktu *delay* 150 – 300 ms. Sedangkan rata-rata *Throughput* yang diperoleh yaitu 227.9383 bps.

Kata Kunci : IoT, Wireless, mikrokontroler, pesan instan.

Abstract

The development of technology every time is growing very rapidly, the Internet of Things (IoT) is here to facilitate and fulfill human needs in carrying out their activities. In this study, a dam monitoring system was designed using a microcontroller that has been integrated with a wireless chip. This system creates remote controlling technology with an internet network as a connection between server and client using the MQTT protocol. This system is designed for dam officers to monitor water level heights using instant messaging communication media in the form of notifications. From the results of testing the dam monitoring system can function properly. And the user can get information about water level from the instant messaging application. From the testing and analysis of the measurement time the sensor shows the

measurement time at a higher distance with a 2% error rate at the farthest distance, if the distance is farther away. Analysis of MQTT delivery delay shows that based on the average data delay the dangerous status has the highest delay of 448744,9608 microseconds, while the average value of throughput from standby status has the highest value of 160.1392157 bps. Memory Usage from the Heroku server with a maximum memory usage of 23.1MB. The longest response time on the Heroku server is 991 ms, while the average is 863 ms and the fastest is 703 ms. In testing and analyzing the whole system, the average delay is 220,377 milliseconds, which can be categorized as Good according to the ITU-T G.1010 standard with a delay of 150 - 300 ms. While the average Throughput obtained is 227.9383 bps.

Key Word : IoT, Wireless, Microcontroller, Instant Messaging.

1. Pendahuluan

Perkembangan teknologi semakin harinya semakin maju, berbagai bidang keilmuan kini telah menggunakan teknologi yang memumpuni guna memudahkan masyarakat dalam melaksanakan pekerjaannya. Berbagai bidang ilmu apapun tidak lepas dari peran teknologi internet, salah satunya yaitu bendungan yang berguna untuk menampung dan menahan laju air dalam skala besar. Keadaan pintu air bendungan saat ini masih dikontrol secara manual melalui petugas bendungan yang turun langsung ke lokasi, sehingga setiap air meluap yang dapat berpotensi banjir petugas harus siap siaga di lokasi bendungan, dimana hal tersebut terlalu berisiko bagi petugas jika suatu saat terjadi cuaca buruk yang bisa membahayakan nyawanya. Maka dari itu dibutuhkan sistem monitoring keadaan bendungan jarak jauh. Penelitian sebelumnya pada jurnal *IJPLA* mengenai *A System for Remote System for Remote Monitoring and controlling of dam*. Pada jurnal tersebut pengaplikasian bendungan menggunakan komunikasi HTTPs yang membuat *server* dan *client* dapat berkomunikasi secara konstan satu sama lain. Dalam penelitian tersebut digunakan sensor – sensor yang berkomunikasi lewat *Bluetooth* untuk selanjutnya informasi dikirimkan *client* ke *server* dan disimpan dalam *database* yang terhubung ke *Web browser*, dimana pengguna dapat memonitoring dan mengontrol bendungan dari jarak jauh [1].

Pada penelitian ini mengaplikasikan penggunaan *sensor ultrasonik HC-SR04* yang akan diolah pada mikrokontroler yang sudah terhubung ke jaringan internet dan *instant messaging Line*, pada *instant messaging Line* tersebut petugas mendapatkan informasi berupa notifikasi mengenai ukuran ketinggian air pada bendungan beserta parameter aman, siaga dan berbahaya. Sehingga jika suatu saat muncul parameter berbahaya maka pintu air bendungan akan terbuka secara otomatis.

2. Dasar Teori

2.1 Internet of Things (IoT)

Menurut rekomendasi ITU-T Y.2060 [1], *Internet of Things* (IoT) didefinisikan sebagai sebuah penemuan yang mampu menyelesaikan permasalahan social yang ada melalui penggabungan teknologi, sementara definisi IoT jika ditinjau dari segi teknik standarisasi ialah sebagai infrastruktur global untuk memenuhi kebutuhan informasi masyarakat yang memungkinkan layanan canggih dengan kemampuan interkoneksi . Pendapat lain tentang IoT ialah dimasa yang akan datang komputer dapat mengambil peran lebih banyak dalam pekerjaan manusia seperti mengontrol peralatan elektronik dari jarak jauh melalui jaringan internet [2].

2.2 Instant Messaging

Instant Messaging merupakan metode komunikasi yang memungkinkan pengguna dapat membagikan informasi berbasis *digital*, seperti teks, suara dan video secara singkat atau instan. Adapun fitur tambahan yang bisa membuatnya semakin populer. Misalnya, pengguna bisa melakukan komunikasi dengan bermacam cara seperti, *video call*, bertukar foto, transfer file, berbagi lokasi atau berbicara langsung secara gratis melalui jaringan internet [3].

2.2.1 LINE Messenger

LINE messenger memiliki fitur keamanan ekstra yaitu Letter Sealing untuk mengamankan datanya. Letter Sealing menggunakan metode keamanan berbasis *end-to-end encryption*, dimana dapat melindungi data sedemikian rupa sehingga hanya dapat dibaca oleh pengirim dan penerima pesan [4].

2.3. Heroku

Heroku merupakan layanan cloud yang termasuk ke dalam *platform as a service* (PaaS). Heroku memungkinkan pengguna untuk melakukan pengembangan aplikasi, menjalankan aplikasi dan mengelola aplikasi tanpa perlu membangun dan melakukan perawatan infrastruktur. Bahasa pemrograman yang mendukung Heroku meliputi java, Node.js, Scala, Clojure, Python, PHP, Ruby, dsb.

2.4 MQTT

Message Queue Telemetry Transport (MQTT) merupakan sebuah protokol yang berjalan diatas protokol TCP/IP dan berbasis *open source*. Sistem umum MQTT membutuhkan dua komponen perangkat lunak utama, yaitu MQTT Client dan MQTT Broker. MQTT Client berperan sebagai penerima data dari MQTT Broker, dimana MQTT Broker berfungsi untuk menangani proses *publish* dan *subscribe* data.

2.5 Sensor Ultrasonik

Prinsip kerja sensor ini ialah saat sensor menerima *trigger* maka *transmitter* akan memancarkan gelombang ultrasonik dengan frekuensi 40 KHz kemudian *receiver* akan mendeteksi hasil gelombang pantul yang mengenai suatu objek. Saat transmitter mulai memancarkan gelombang ultasonik, pada saat yang sama sensor akan mulai menghitung waktu pengukuran dan akan berhenti saat *receiver* dapat menangkap gelombang pantul yang dihasilkan.

2.6 Mini Submersible pump

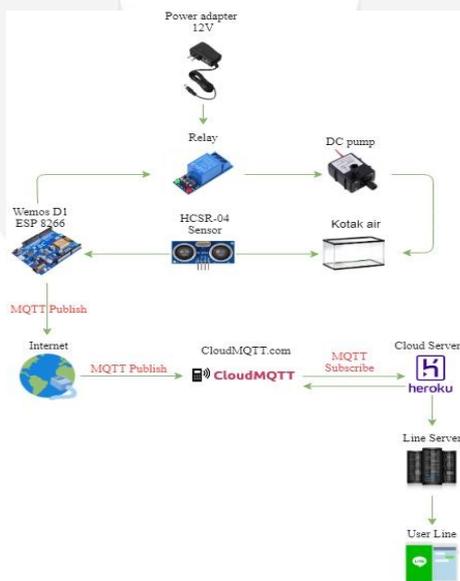
Mini Submersible Water Pump adalah motor pompa air celup dengan ukuran kecil. Pompa air mini ini digunakan untuk akuarium, kolam ikan, hidroponik, robotika atau proyek dalam pembuatan aplikasi yang berbasis mikrokontroler.

2.7 Wemos D1 ESP8266

WeMos D1 adalah *board* berbasis WiFi ESP8266 yang menggunakan tata letak Arduino dengan tegangan operasi 3,3V. *Board* ini dikendalikan oleh chip ESP8266 (prosesor 32-Bit) dan memiliki memori flash yang lebih besar dibandingkan dengan Arduino Uno. Terdiri dari 11 pin I/O digital dan 1 pin analog (input).

3. Perancangan Sistem

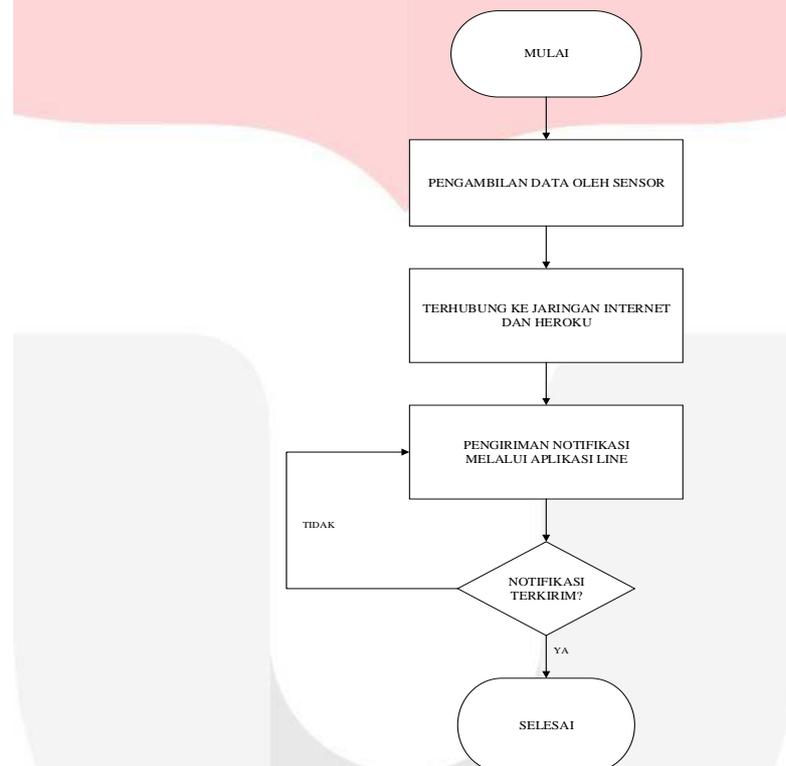
3.1. Diagram Blok Sistem



Gambar 3. 1 Diagram blok sistem.

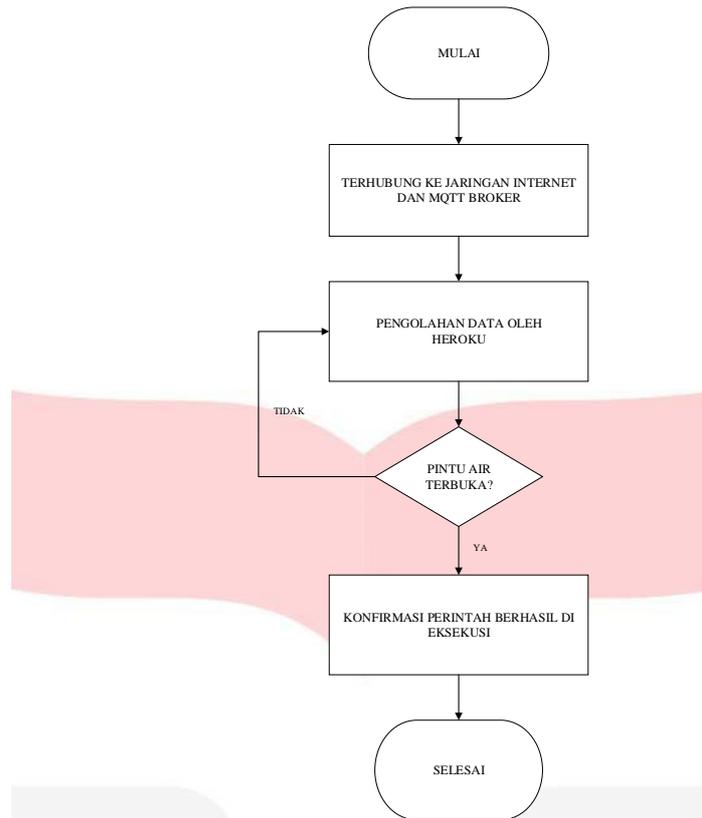
Pada gambar 3.1 menunjukan diagram blok sistem monitoring bendungan menggunakan *instant messaging* line. Wemos D1 ESP8266 sebagai mikrokontroler mengirimkan data ke server Heroku melalui protocol MQTT. Broker MQTT yang menghubungkan server Heroku dengan wemos menggunakan cloudmqtt sehingga prosesnya wemos mengirimkan terlebih dahulu data ke cloudmqtt dalam bentuk MQTT publish, sedangkan dari sisi Heroku melakukan MQTT subscribe ke cloud MQTT. Apabila terdapat pesan dari wemos yang ditujukan ke Heroku, kemudian mengirimkan pesan tersebut ke line server untuk diteruskan ke channel line dengan ID penerima tertentu. Dari channel line tersebut output yang dikeluarkan berupa peringatan atau notifikasi level air pada bendungan, yaitu peringatan Aman, Siaga dan Bahaya yang nantinya informasi tersebut akan disalurkan kepada *user*. *User* yang dimaksud disini adalah petugas bendungan. Selain itu pintu air pada bendungan juga dapat terbuka secara otomatis saat peringatan menunjukkan parameter bahaya dimana level air sudah mencapai titik tertinggi.

3.2 Diagram Alir Sistem



Gambar 3.2 Diagram alir pengiriman data ke line

Berdasarkan gambar 3.2 proses kerja diawali dengan pengambilan data lewat sensor ultrasonik dengan parameter ketinggian yang berbeda-beda. Deteksi jarak sensor ke permukaan air $\leq 5\text{cm}$ dapat dikatakan Bahaya, pada ketinggian $\leq 10\text{cm}$ jarak sensor ke permukaan air dikatakan Siaga dan ketinggian air $\geq 10\text{cm}$ jarak sensor ke permukaan air dapat dikatakan bahwa bendungan berada pada titik Aman. Setelah data ketinggian air didapat dan diolah pada Wemos D1 ESP8266 selanjutnya akan di proses ke *Cloud server* Heroku untuk menampung dan memproses semua *request* yang dikirim oleh Wemos yang terhubung ke line bot server untuk mengirimkan notifikasi peringatan bahaya.



Gambar 3.3 Diagram alir pembukaan pintu air

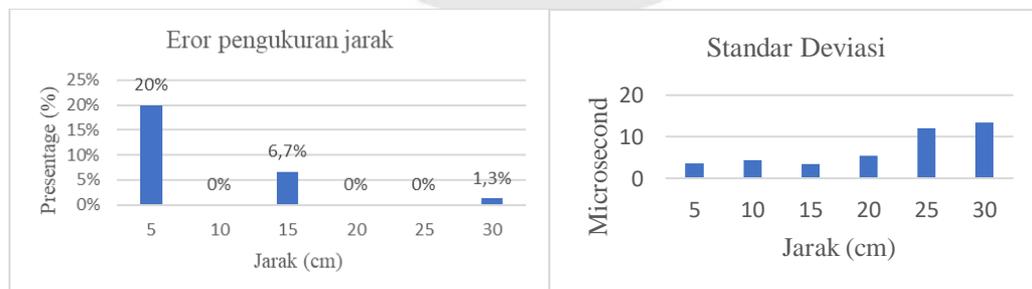
Berdasarkan gambar 3.3 proses pengiriman data mikrokontroler diharuskan sudah terkoneksi dengan internet dan dengan MQTT Broker. Ketika sudah terkoneksi, Wemos berada pada kondisi *idle* dan akan terus berada pada kondisi *idle* sampai mencapai kondisi dimana terdapat data yang diterima. Data tersebut berisikan perintah yang harus dieksekusi oleh Wemos untuk membuka pintu air. Jika pintu air terbuka, maka proses *request* dari *user* akan dikonfirmasi. Namun, jika pintu air tidak terbuka, maka wemos akan berada pada kondisi *idle* sampai data diterima.

4. Hasil Pengujian dan Analisis

4.1.1 Pengujian dan Analisa pengukuran jarak

semakin jauh jarak dari sensor ke objek maka dibutuhkan waktu yang relatif lebih lama dalam penyampaian informasi. Dimana jarak terjauh dari sensor yaitu pada jarak 30 cm dibutuhkan waktu 2267,8 μs dan jarak terdekat dari sensor yaitu 5 cm dibutuhkan waktu 854,7 μs .

4.1.2 Pengujian dan Analisa waktu pengukuran

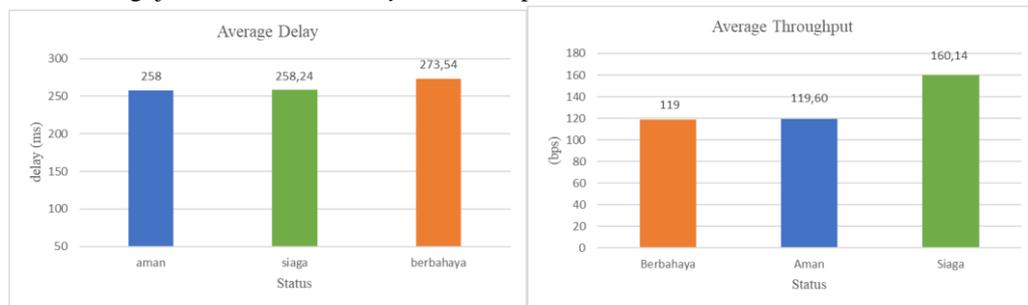


Gambar 4.1 Grafik waktu pengukuran Sensor Ultrasonik dan Standar Deviasi

Pada pengukuran tersebut terlihat maksimum eror berada pada pengukuran 5 cm jarak sensor ke objek, pada jarak 10 cm, 20 cm dan 25 cm tidak ditemukan kesalahan pengukuran. Sedangkan pada pengukuran 15 cm terdapat eror sebesar 6,7% dan pada pengukuran 30 cm terdapat eror 1,3%. Pada grafik standar deviasi terlihat bahwa adanya deviasi waktu dalam μs yang semakin menyimpang pada pengukuran jarak yang semakin jauh dari objek. Dalam hal ini jarak paling jauh dari pengukuran yang diambil yaitu 30 cm membutuhkan waktu 13,389 μs .

4.2 Pengujian dan Analisa Pengiriman MQTT

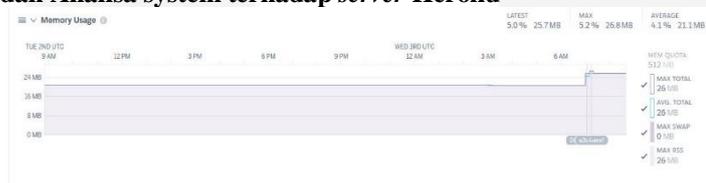
4.2.1 Pengujian dan Analisa Delay Pada Setiap status



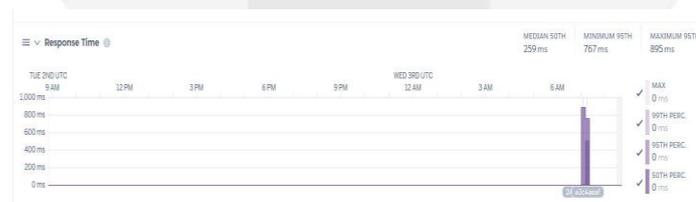
Gambar 4.2 Grafik Delay rata-rata dan Throughput rata-rata.

Berdasarkan gambar 4.2 mengenai grafik *delay* rata-rata, status berbahaya mengalami *delay* yang paling besar yaitu mencapai 273,54 ms dibandingkan status aman yaitu 258,24 ms dan status siaga yaitu 258 ms. Berdasarkan grafik diatas *throughput* adalah berapa banyaknya *byte* yang mampu dikirim pada waktu tertentu. Berdasarkan pengujian ini, nilai rata-rata *throughput* dari status siaga memiliki nilai yang paling tinggi yaitu 160,14 bps.

4.3 Pengujian dan Analisa system terhadap server Heroku



Gambar 4.3 Grafik Memory Usage



Gambar 4.4 Grafik Response Time



Gambar 4.5 Grafik Dyno Load

Pada pengujian server digunakan Heroku metrics, *Memory Usage* dari Heroku server sangat kecil dengan maksimum penggunaan memory 23.1 MB sedangkan rata-rata 2.6 MB. Hal ini mengacu pada proses deploy dan restart yang membutuhkan kinerja memory lebih tinggi. *Response time* pada Heroku server paling lama adalah 991 ms, sedangkan rata-ratanya adalah 863 ms dan yang tercepat adalah 703 ms. *Response time* tersebut dapat dikategorikan baik karena memiliki waktu kurang dari 1 detik. Adapun *response time* terlama yaitu terjadi saat proses deploy. Beban pada CPU maksimum diperoleh 0.31 task CPU yang dapat dieksekusi paling tinggi, sedangkan rata-rata beban yang dapat dieksekusi adalah 0.05 task CPU.

4.4 Pengujian dan Analisis Reliability dan Availability

Pada pengujian Reliability diketahui bahwa dari 30 kali pengiriman pesan yang dikirim setiap 3 detik, data tersebut berhasil dikirim. Artinya Reliability pada kondisi tersebut adalah 100%. Dengan keseluruhan data diterima maka sistem pengiriman MQTT dari wemos ke server Heroku memiliki reliability yang baik. Selanjutnya adalah pengukuran Availability. Penulis melakukan pengiriman informasi dari wemos ke server Heroku dimana pengujian yang dilakukan adalah dengan mengirimkan pesan sebanyak 44 kali dengan waktu yang random. Pada pengujian availability menunjukkan bahwa Heroku memiliki ketersediaan yang baik terlihat dari hasil pengiriman data dengan waktu yang random, server mampu menerima data dengan baik dan 100% diterima.

4.5 Pengujian dan Analisis Performansi jaringan Wemos – server - line

Pada pengujian dan Analisis sistem ini ditujukan untuk melihat kemampuan sistem yang telah dibuat saat traffic masuk ke Line App dari server Heroku dan wemos dengan menggunakan wireshark. Pada pengujian dan Analisis keseluruhan sistem ini didapat rata-rata delay sebesar 220,377 ms, sedangkan rata-rata Throughput yang diperoleh yaitu 227,938 bps. Nilai ini diperoleh cukup baik, dan data tidak terlalu besar karena hanya berupa karakter text.

5. Kesimpulan dan Saran

Kesimpulan

1. Sistem monitoring aquarium yang terdiri dari Wemos D1 ESP8266, Sensor Ultrasonik, cloud server Heroku dan pesan instan dapat berfungsi dengan baik. Serta user dapat mendapatkan informasi mengenai ketinggian air dari aplikasi pesan instan.
2. Pompa air dapat mengeluarkan air secara otomatis saat ketinggian air berada pada status berbahaya.
3. Pengujian dan analisis pengukuran jarak sensor ultrasonik menunjukkan bahwa maksimum error terjadi pada jarak 5 cm dari objek yaitu mencapai 20%. Hal ini diakibatkan oleh kemampuan pengukuran HCSR 04 yang memiliki jarak minimum 2 cm sehingga dapat mengalami kesalahan sekitar 20% pada jarak 5 cm.
4. Pengujian dan analisis waktu pengukuran sensor menunjukkan bahwa waktu pengukuran terhadap jarak semakin lama apabila jarak dari sensor ke objek semakin jauh, pada pengujian ini jarak terjauh dari sensor yaitu 30 cm memakan waktu 2267,8 μ s dan jarak 5 cm memakan waktu 854,7 μ s.
5. Berdasarkan pengujian dan analisis delay pengiriman MQTT menunjukkan bahwa delay rata-rata status berbahaya mengalami delay paling besar yaitu 273,537 ms dibandingkan status aman yaitu 258,016 ms, dimana dapat dikategorikan Good menurut standar ITU-T G.1010 dengan rentang waktu delay 150 – 300 ms. Hal tersebut bisa terjadi karena jumlah byte yang dikirim sedikit berbeda antara kata-kata status berbahaya, aman dan siaga, dalam hal ini berbahaya memiliki panjang byte yang lebih banyak dibandingkan status lainnya.
6. Berdasarkan pengujian dan analisis throughput pengiriman MQTT, nilai rata-rata throughput dari status siaga memiliki nilai yang paling tinggi yaitu 160,139 bps.
7. Pada pengujian server Heroku menggunakan Heroku metrics maksimum penggunaan memori dari Heroku sebesar 26,8 MB sedangkan rata-rata penggunaan memori adalah 2,6 MB. Hal ini mengacu pada proses deploy dan restart yang membutuhkan kinerja memori lebih tinggi.
8. Response time pada Heroku server memiliki respon paling lama adalah 767 ms, sedangkan rata-rata waktu respon adalah 259 ms dan yang tercepat adalah 895 ms. Response time tersebut dapat dikategorikan baik karena memiliki waktu kurang dari 1 detik. Adapun response time terlama yaitu terjadi saat proses deploy.

9. Beban pada CPU maksimum diperoleh 0.5 *task* CPU yang dapat dieksekusi paling tinggi, sedangkan rata-rata beban yang dapat dieksekusi adalah 0.07 *task* CPU
10. Pada pengujian dan analisis keseluruhan sistem didapat rata-rata *delay* sebesar 220.377 ms, dimana dapat dikategorikan *Good* menurut *standar ITU-T G.1010* dengan rentang waktu *delay* 150 – 300 ms. Sedangkan rata-rata *Throughput* yang diperoleh yaitu 227,938 bps.
11. Nilai *Reliability & Availability* pada saat pengiriman informasi dari wemos ke Heroku memiliki nilai 100% dengan skenario yang telah ditentukan.

Saran

1. Untuk penelitian berikutnya diharapkan melakukan perbandingan penggunaan protokol.
2. Melakukan penambahan fitur pada aplikasi *Line Messenger*
3. Menambah parameter pengujian.
4. Melakukan implementasi pada objek lain.

Daftar Pustaka

- [1] ITU-T, "Overview of the Internet of things," in *ITU-T Y.2060*, 2012.
- [2] A. Junaidi, "INTERNET OF THINGS, SEJARAH, TEKNOLOGI DAN PENERAPANNYA," *Jurnal Ilmiah Teknologi Informasi Terapan : REVIEW*, vol. 1, p. 3, 2015.
- [3] B. I. A. Barry and F. M. Tom, "INSTANT MESSAGING: STANDARDS, PROTOCOLS, APPLICATIONS AND RESEARCH DIRECTION," *ResearchGate*, vol. 7, 2015.
- [4] LINE, [Online]. Available: <https://help.line.me/line/?contentId=50000087>.
- [5] E. I. Team, "Espressif Inc.," 2018. [Online]. Available: www.espressif.com. [Accessed 24 October 2018].
- [6] STMicroelectronics, "SparkFun Electronics," 2000. [Online]. Available: www.sparkfun.com. [Accessed 15 October 2018].