

# Penerapan *IoT* (Internet Of Thing) Pada Sistem Monitoring Dan Kontrol Aquarium Berbasis *Web Service*

Muhammad Haikal<sup>1</sup>, Mursyidah<sup>2</sup>, Muhammad Nasir<sup>3\*</sup>

<sup>1,3</sup> *Jurusan Tekniknologi Informasi dan Komputer Politeknik Negeri Lhokseumawe  
Jln. B.Aceh Medan Km.280 Buketrata 24301 INDONESIA*

<sup>1</sup>mhaikal225@gmail.com

<sup>2</sup>mursyidah@pnl.ac.id

<sup>3\*</sup>muhammadnasir@pnl.ac.id

*Abstrak*— Monitoring dan kontrol dalam aquarium salah satunya yaitu memantau kondisi pakan ikan dan kondisi kekeruhan air pada aquarium. Metode biasa yang saat ini digunakan adalah dengan melakukan monitoring dan kontrol secara manual yaitu dengan datang langsung ke aquarium yang dilakukan oleh pemilik ikan hias itu sendiri. hal tersebut tentu timbul permasalahan yaitu dibutuhkan banyak tenaga dan waktu untuk menjaga dan merawat kesehatan air serta pakan aquarium setiap saat. Sehingga para pemilik tidak dapat meninggalkan aktivitas tersebut, yang menyebabkan kontrol dan pengamatan tidak efektif dan efisien lagi. Oleh karena itu dibangunnya sebuah penerapan Internet of Thing pada sistem monitoring dan kontrol aquarium berbasis web service. Dalam metode pengujiannya menggunakan pengujian respon terhadap alat dan website. Dari hasil pengujian Rata-rata bandwidth yang didapat pada pengamatan packet loss yaitu 10.73, pada delay 5.13 dan pada jitter 8.22, total bandwidth keseluruhan ialah 8.02918 pada saat proses pengiriman data informasi.

*Kata kunci*— *Raspberry Pi, monitoring dan kontrol, aquarium, webservice, MQTT*

*Abstract*— *Monitoring and control in the aquarium , one of which is monitoring the condition of fish feed and water turbidity conditions in the aquarium. The usual method currently used is to monitor and control manually, namely by coming directly to the aquarium which is carried out by the owner of the ornamental fish themselves. This of course arises a problem, namely it takes a lot of energy and time to maintain and care for the health of the water and aquarium feed at all times. So that the owners cannot leave the activity, which causes the control and observation to be ineffective and inefficient. Therefore, an application Internet of Thing -based aquarium monitoring and control system web service. In the test method using response testing of tools and websites. From the test results, the average bandwidth obtained by observing packet loss is 10.73, at delay 5.13 and at jitter 8.22, the total bandwidth is 8.02918 during the process of sending information data.*

*Keywords*— *Raspberry Pi, monitoring dan kontrol, aquarium, webservice, MQTT*

## I. PENDAHULUAN

Monitoring dan kontrol saat ini sudah semakin maju seiring dengan perkembangan teknologi di dunia. Penggunaan teknologi di bidang monitoring dan kontrol dapat dilakukan dengan kolaborasi perangkat keras (hardware) dan lunak (software) membentuk suatu sistem dalam melakukan pemantauan dan kontrol suatu daerah atau bidang kerja tertentu.

Monitoring dan kontrol dalam aquarium salah satunya yaitu memantau kondisi pakan ikan dan kondisi kekeruhan air pada aquarium. Metode biasa yang saat ini digunakan adalah dengan melakukan monitoring dan kontrol secara manual yaitu dengan datang langsung ke aquarium yang dilakukan oleh pemilik ikan hias itu sendiri. Dengan hal tersebut timbul permasalahan yakni diperlukan banyak tenaga dan waktu untuk menjaga dan merawat kesehatan air serta pakan aquarium setiap saat.

Internet Of Things (IoT) merupakan suatu penemuan yang dapat membantu menyelesaikan permasalahan yang ada dengan menggabungkan teknologi dan dampak sosial, apabila dilihat dari standarisasi secara teknik, IoT bisa dikatakan sebagai infrastruktur *global* dalam memenuhi kebutuhan informasi pada masyarakat [8].

*MQTT* (Message Queuing Telemetry Transport) protokol merupakan sebuah protokol yang berjalan diatas stack TCP/IP

Sehingga para pemilik tidak dapat meninggalkan aktivitas tersebut, yang mengakibatkan kontrol dan pengamatan tidak efektif dan efisien lagi. Hal inilah yang mengakibatkan banyaknya ikan hias mati pada sebuah aquarium.

Solusi dari permasalahan tersebut adalah dibangunnya sebuah penerapan Internet of Thing pada sistem monitoring dan kontrol aquarium berbasis *web service*.

Penerapan teknologi ini berfungsi untuk melakukan pengontrolan pakan ikan dan memonitoring kondisi kekeruhan air menggunakan antarmuka yang dijalankan melalui website menggunakan *device* sehingga dapat memudahkan pemilik aquarium dalam memelihara ikan hias. Berdasarkan latar belakang diatas, peneliti tertarik untuk membangun Penerapan IoT (Internet Of Thing) pada Sistem Monitoring dan Kontrol Aquarium Berbasis *Web service*

dan dirancang khusus untuk machine to machine yang tidak memiliki alamat khusus. Maksud dari kata tidak memiliki alamat khusus ini seperti halnya sebuah arduino, raspberry Pi atau *device* lain yang tidak memiliki alamat.khusus [10].

*Rest* adalah gaya arsitektur dalam merancang layanan web, di mana desain *rest* memiliki sumber daya yang dapat diakses melalui satu *URL HTTP* yang unik. *Rest* tersebut memungkinkan klien untuk membuat permintaan melalui

*protocol HTTP* dengan mudah menggunakan *URL* pencarian. Setiap *URL* tersebut mengacu pada sekumpulan program yang akan dijalankan dan mengembalikan pesan ke pengirim perintah[11].

Pesan yang nantinya diterima dari server sebagai kode sukses atau gagal yaitu *HTTP* di header, hasil pesan berasal dari penanganan program itu sendiri. Kode *HTTP* berikut biasanya digunakan saat menggunakan *Rest API*.

*QoS* adalah metode pengukuran untuk mengetahui kualitas jaringan dengan mengukur atribut kinerja yang telah diproses, dirincikan dan dikumpulkan guna mengetahui karakteristik dan sifat dari satu pelayanan. *QoS* memiliki tujuan akhir berupa memberikan *network service* yang lebih baik dan meningkatkan *loss* karakteristik[12].

*Web service* adalah kumpulan protokol dan standar terbuka yang digunakan untuk bertukar data antara aplikasi atau sistem. Aplikasi perangkat lunak yang ditulis dalam berbagai bahasa pemrograman dan berjalan di berbagai *platform* dapat menggunakan *web service* untuk bertukar data melalui jaringan komputer seperti Internet dengan cara yang mirip dengan komunikasi antar proses pada satu komputer [13].

*Laravel* adalah salah satu *framework PHP* yang membantu proses pengembangan website yang dapat digunakan secara gratis. *Laravel* dikembangkan oleh programmer asal amerika yang bernama Taylor Otwell dan diluncurkan sejak tahun 2011, kemudian mengalami pertumbuhan yang cukup eksponensial.

*XAMPP* merupakan sebuah *software web server apache* yang di dalamnya udah tersedia *database server mysql* dan support php programming. *XAMPP* merupakan *software* yang mudah digunakan, gratis dan mendukung instalasi di *Linux* dan *Windows*. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia *Apache Web Server, MySQL Database Server, PHP Support (PHP 4 dan PHP 5)* dan beberapa modul lainnya

Hanya bedanya kalau yang versi untuk *Windows* sudah dalam bentuk instalasi grafis dan yang *Linux* dalam bentuk *file* terkompresi *tar.gz*. Kelebihan lain yang berbeda dari versi untuk *Windows* adalah memiliki fitur untuk mengaktifkan sebuah *server* secara grafis, sedangkan *Linux* masih berupa perintah-perintah di dalam *console* [16].

*Database* digunakan untuk mengelompokkan data serta informasi agar data lebih mudah dimengerti, *database* juga berfungsi untuk mencegah duplikasi serta inkonsistensi data yang mungkin terjadi apabila data dan informasi disimpan dalam jumlah yang besar [17].

*Database* merupakan sekumpulan data yang saling berhubungan dan disimpan dalam komputer secara sistematis dan mempunyai arti secara implisit serta dapat diolah dan diperiksa. Contohnya: *database* untuk universitas akan berisi informasi-informasi berupa nama mahasiswa, fakultas, mata kuliah, serta ruangan kuliah. Dalam proses pengolahan *database*, dibutuhkan *software* yang dapat membantu untuk menyimpan dan mengambil data dari *database*, yang sering disebut dengan *Database Management System (DBMS)* atau biasa dikenal dengan manajemen sistem basis data. *DBMS* merupakan *tools* yang bisa digunakan untuk membuat serta mengatur data dengan ukuran data yang cukup besar dengan proses yang efisien.

*Database* digunakan untuk mengelompokkan data serta informasi agar data lebih mudah dimengerti, *database* juga berfungsi untuk mencegah duplikasi serta inkonsistensi data yang mungkin terjadi apabila data dan informasi disimpan dalam jumlah yang besar. Dalam proses penyimpanan *database* yang sistematis dapat menjaga kualitas data dan informasi yang akan diakses, *Database* sangat berguna untuk mekanisme penyimpanan data, khususnya dalam konsep big data.

*Operational database* digunakan untuk memperbarui data secara real-time. Tidak hanya sekedar melihat datanya saja, tipe basis data ini memungkinkan pengguna untuk memodifikasi data seperti menambah, mengubah, dan menghapus data secara *real-time*.

*Relational database* adalah jenis *database* yang menyimpan dan menyediakan akses ke data yang terkait satu sama lain. *Relational database* didasarkan pada model relasional, cara yang intuitif dan mudah untuk merepresentasikan data dalam tabel. Dalam *database* relasional, setiap baris dalam tabel adalah catatan dengan ID unik yang disebut kunci. Kolom tabel menyimpan atribut data, dan setiap record memiliki nilai untuk setiap atribut.

*Distributed Database* adalah kumpulan *database* terintegrasi yang didistribusikan secara fisik di seluruh situs dalam jaringan komputer. Untuk membentuk sistem *database* terdistribusi

Salah satu tujuan utama digunakannya *DBMS* adalah untuk menghindari kerumitan dalam proses pengolahan data yang mempunyai ukuran yang cukup besar serta dapat meningkatkan efektivitas dalam prosesnya dalam proses penyimpanan *database* yang sistematis dapat menjaga kualitas data dan informasi yang akan diakses, *Database* sangat berguna untuk mekanisme penyimpanan data, khususnya dalam konsep *big data*.

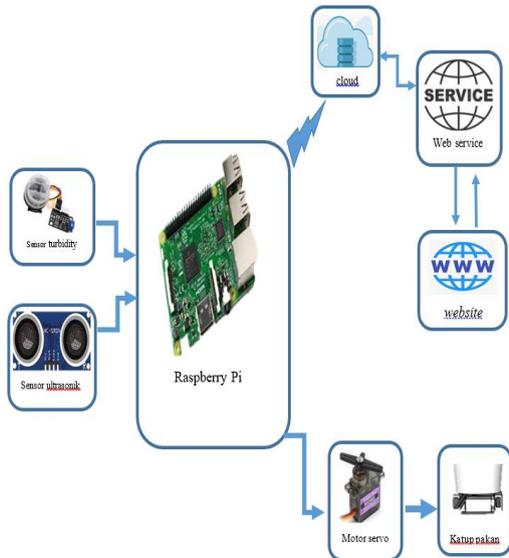
## II. METODOLOGI PENELITIAN

### A. Perancangan Sistem

Dalam proses perancangan Sistem Monitoring dan Kontrol aquarium berbasis *Raspberry Pi* terdapat beberapa hal yang perlu dikerjakan seperti merangkai struktur kabel yang antara modul sensor dan mikrokontroler, serta merancang sebuah website yang akan digunakan untuk memantau sistem, membuat pengkodean dari sisi hardware dan membuat jembatan yang menghubungkan antara *hardware* dan website pengontrolnya.

### B. Rancangan Sistem

Perancangan sistem digunakan untuk menjelaskan gambaran mengenai perancangan sistem yang akan dibuat. Blok diagram sistem secara keseluruhan menjelaskan bagaimana penerapan *Internet of Things* pada Sistem Monitoring dan Kontrol *Aquarium* Berbasis *Web service* dapat bekerja.

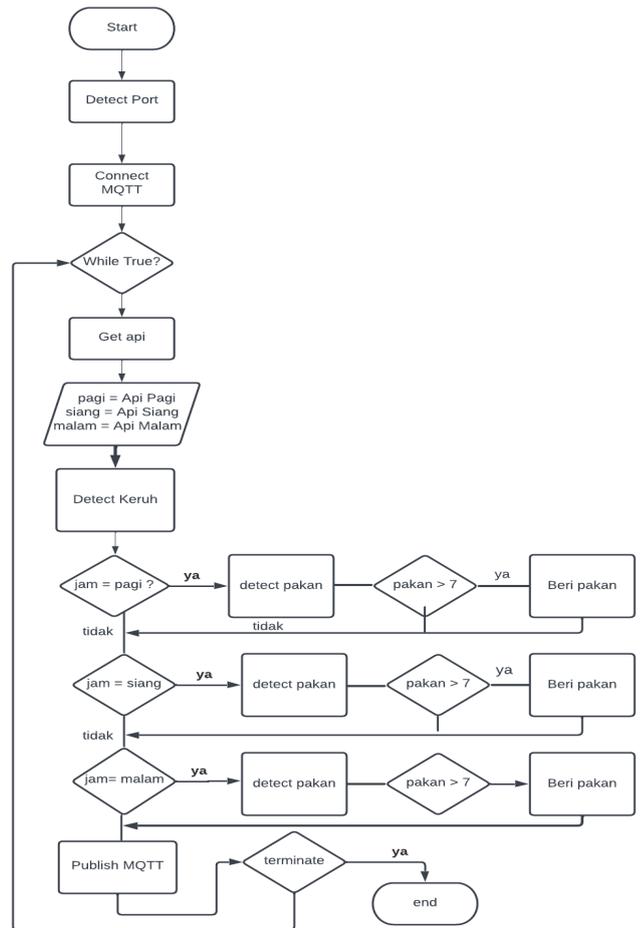


Gambar 1. Rancangan arsitektur jaringan

Berdasarkan rancangan sistem keseluruhan pada gambar tersebut bahwasanya menggambarkan konesifikasi antara alat yang akan digunakan pada *aquarium* dan *website*. Data atau informasi dari sensor yang dikirimkan melalui *raspberry pi* nantinya akan tersimpan pada sebuah *cloud* yang digunakan sebagai tempat penyimpanan atau pengambilan suatu data, yang dimana data yang tersimpan pada *cloud* akan di akses melalui *website*. Untuk mendukung interaksi antara *cloud* dan *website* maka digunakan yang namanya *web service* yang berfungsi sebagai jalur penghubung antara *cloud* dan *website* dalam sebuah jaringan, yang dimana Jaringan yang akan digunakan biasanya berupa *HTTP*.

C. Rancangan Flowchart Sistem

*Flowchart* sistem digunakan untuk menggambarkan fungsional terhadap *system* yang akan berjalan. Gambar 2. Menunjukkan *flowchart* sistem yang akan dibuat pada penelitian ini.

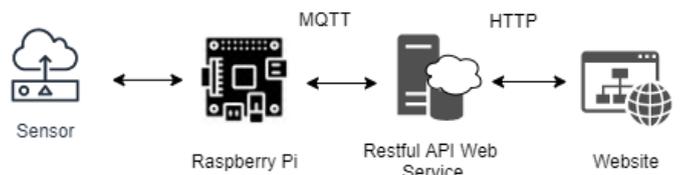


Gambar 2. Rancangan flowchart sistem

Pada *flowchart* gambar 2 Alur kerja Sistem, dapat dilihat proses pertama yaitu melakukan deteksi port yang digunakan oleh sensor kemudian alat akan mengkoneksikan ke MQTT jika *while true*, maka alat akan mengambil data dari API yang berupa variabel pagi siang dan malam. Alat akan mendeteksi keruh menggunakan sensor *turbidity* kemudian akan dilakukan pengecekan jam apakah jam sesuai dengan data pagi, siang dan malam. Jika data sama maka akan melakukan pendeteksian pakan apakah pakan cukup atau tidak dengan acuan 7cm untuk pakan kosong, data akan dipublish ke MQTT dan akan dilakukan pengecekan apakah program dihentikan atau tidak, jika tidak dihentikan akan dilakukan looping kembali mulai dari pengambilan API.

D. Blok Diagram Web service

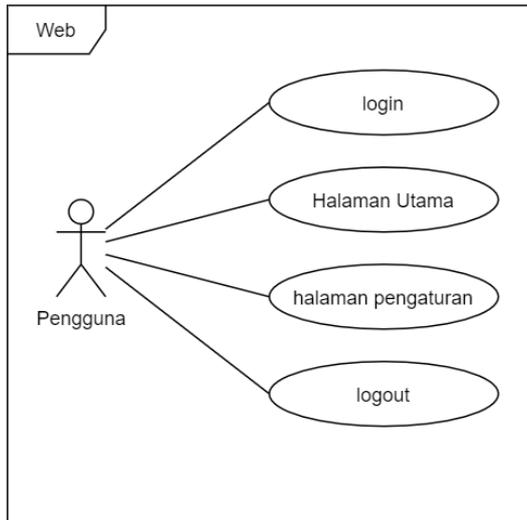
Pada bagian ini merupakan diagram *webservice* yang akan digunakan pada saat pertukaran data antara alat dan *website*, agar dapat beroperasi secara efektif dan efisien. Berikut keterangan blok diagram *web service* yang digunakan untuk penelitian ini dapat dilihat pada gambar 3.



Gambar 3. Blok Diagram Web Service

Perancangan *Use Case Diagram*

*Use case diagram* dibuat guna melihat cara kerja sistem yang berjalan, terutama pada pengguna. hanya terdapat aktor pengguna yang akan digunakan untuk membangun sistem ini. Pengguna aktor ini dapat melakukan pemantauan data sensor dari sistem monitoring yang telah dibuat. Perancangan *use case* dapat dilihat pada Gambar 4 sebagai berikut.



Gambar 4. Perancangan *Use Case*

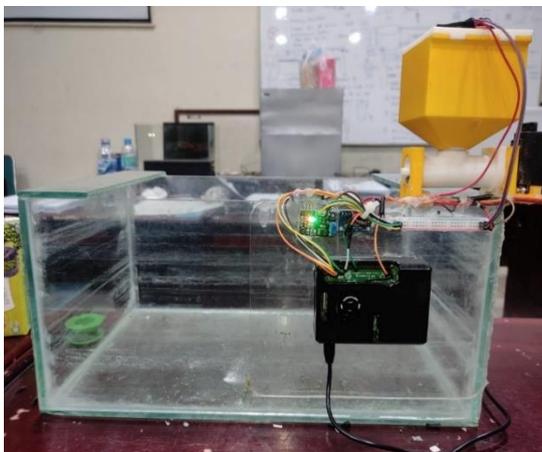
E. Rancangan Sistem (*Software*)

Rancangan sistem dibangun dengan tujuan untuk mengetahui gambaran perihal sistem yang akan dibangun. Pada penelitian “Implementasi Web Service Pada Sistem Akuakultur Berbasis Internet of things”, Adapun pada perancangan ini terdapat blok diagram sistem, use case diagram dan user interface.

III. HASIL DAN PEMBAHASAN

I. *Implementasi Rangkaian Perangkat*

Pemasangan rangkaian pada perangkat dilakukan untuk mengetahui apakah sistem memiliki kesalahan rangkaian atau tidak, pada tampilan rangkaian alat dapat dilihat pada Gambar 5 sebagai berikut.



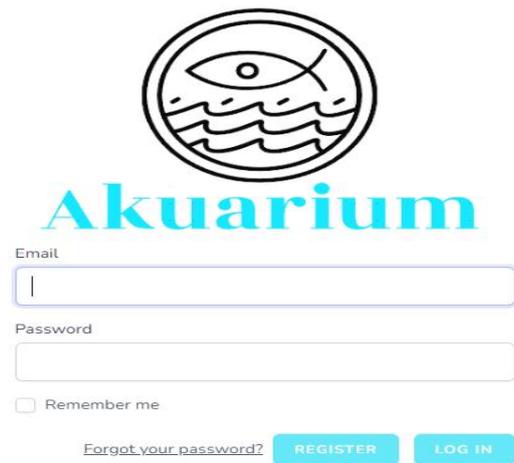
Gambar 5. Pemasangan rangkaian pada perangkat

II. Implementasi *Interface*

Pada implementasi interface akan dibahas bagaimana prosedur dan fungsi yang terdapat pada aplikasi monitoring aquarium yang akan di terapkan pada *device*. Pengujian sistem ini bertujuan untuk mengetahui sejauh mana aplikasi berhasil berjalan sehingga dapat menjalankan serta mengetahui kelebihan dan kekurangan dari aplikasi

1. Tampilan *Login*

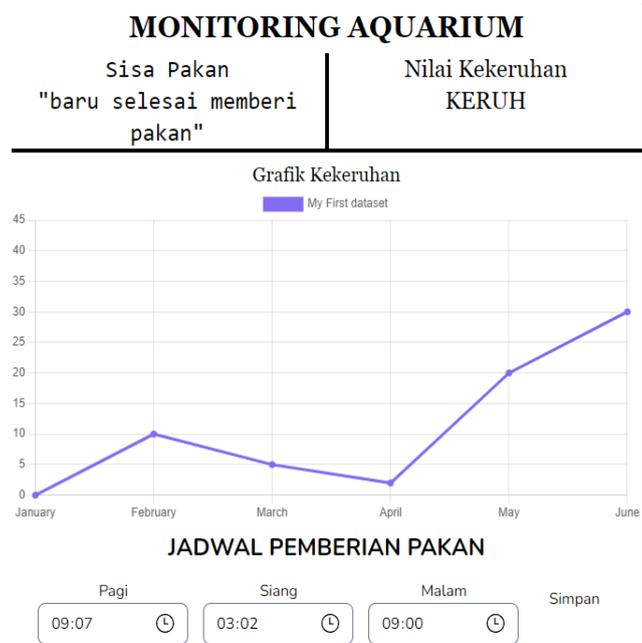
Halaman *login* merupakan halaman yang dipakai untuk mengakses sistem monitoring *aquarium*. Tampilan halaman *login* dapat dilihat pada Gambar 6 berikut



Gambar 6. Perancangan *Use Case*

2. Halaman Awal

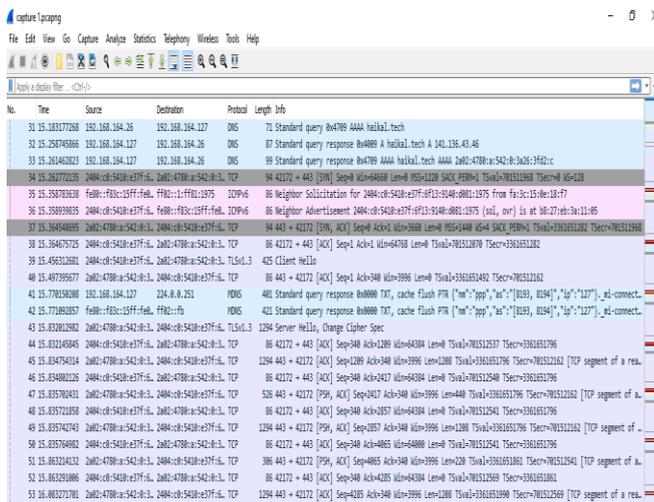
Halaman awal merupakan halaman yang digunakan untuk menyajikan data informasi terkait nilai kekeruhan pemberian pakan ikan pada aquarium. Tampilan halaman awal dapat dilihat pada Gambar 7 berikut



Gambar 7. Grafik Pemberian Pakan Ikan

### III. Pengujian Qos Pada Saat pengiriman Data

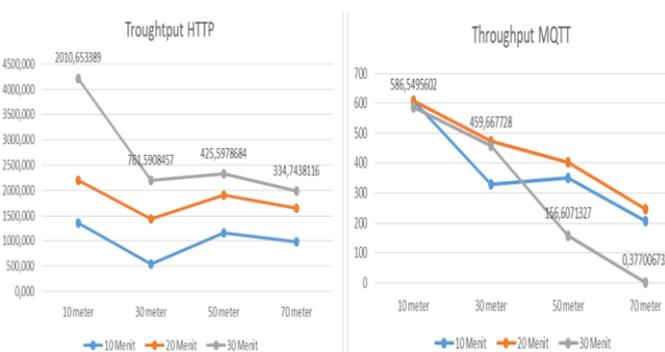
*throughput*, *packet loss*, *delay*, dan *jitter* adalah parameter yang akan diukur pada tahap pada saat pengiriman data informasi yang berisi tentang pemberian pakan dan kekeruhan air pada aquarium. Metode yang dilakukan pada pengujian kinerja pengiriman data ialah menggunakan aplikasi *wireshark* dengan melakukan *capture* data. Dari hasil *capture* data tersebut dapat dilihat perbandingan kualitas pengiriman data dari masing-masing protokol tersebut baik menggunakan protokol *MQTT* atau *HTTP*, Hasil *capture wireshark* dapat dilihat pada Gambar 8 sebagai berikut.



Gambar 8. Hasil *capture wireshark*

### IV. Pengujian Throughput

Pengujian *throughput* dilakukan untuk mengetahui jumlah total kedatangan paket yang berhasil dikirim ke tujuan selama interval waktu tertentu dengan durasi interval waktu tersebut. Hasil pengamatan pengujian dapat dilihat pada grafik Gambar 9 berikut.

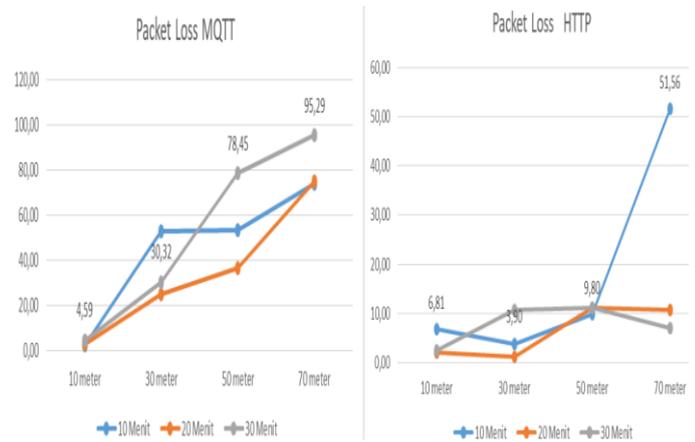


Gambar 9. Hasil pengamatan pengujian *Throughput*

Pada Gambar 9 menjelaskan Hasil pengujian *throughput* pada protokol *MQTT* menunjukkan bahwa ada beberapa penurunan *throughput*. Dalam waktu pengujian kurang lebih 10 menit, terjadi penurunan dari 10 meter menjadi 30 meter, kemudian angka tersebut naik lagi pada 50 meter..

### V. Pengujian Packet loss

Selama proses transmisi data, akan dilakukan pengujian *packet loss* untuk mengetahui kondisi jumlah total paket yang hilang. Dalam pengujian protokol *MQTT*, *packet loss* paling kecil terjadi pada jarak 10 meter, yang hanya terjadi antara 2 dan 4 persen dari waktu. Pada Gambar 10 dapat dilihat Hasil dari pengamatan pengujian pada *packet loss*.

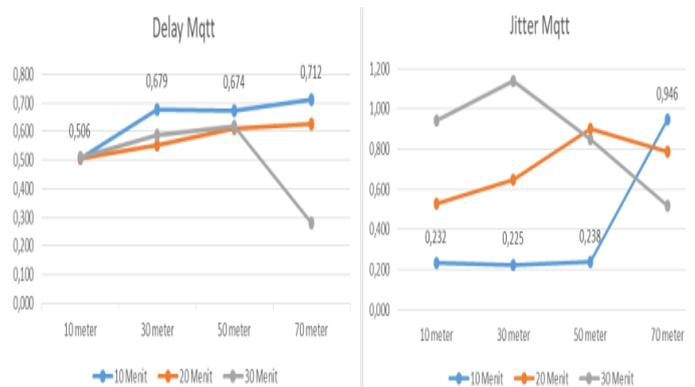


Gambar 10. Pengujian *Packet Loss*

Sebagian besar waktu, hal-hal yang dapat menyebabkan *packet loss* disebabkan oleh gangguan pada sinyal wifi

### VI. Pengujian Delay Dan Jitter

Pengujian *delay (Latency)* dilakukan untuk mengetahui berapa lama waktu yang dibutuhkan data pada saat proses pengiriman data dari asalnya ke tujuannya selama proses transmisi. Hasil rata-rata dari pengujian *delay* pada Protokol *MQTT* adalah sama, namun penundaan meningkat seiring bertambahnya jarak, tetapi laju penundaan berkurang sekitar 30 menit dan 70 meter, sehingga terdapat perbedaan. pada pengujian *Jitter* menghasilkan berbagai hasil. pengujian *Delay* dan *Jitter* dapat dilihat pada Gambar 11 sebagai berikut.



Gambar 11. Pengujian *Delay dan Jitter*

Dapat dilihat pada Gambar 8 pada pengujian *Jitter* menghasilkan berbagai hasil, dengan angka tetap stabil di 0,2 selama kira-kira 10 menit pada jarak 10 meter hingga 50 meter dan meningkat menjadi 70 meter pada

jarak 70 meter, Delay dapat terjadi karena disebabkan oleh jarak dan media fisik atau juga waktu proses pengiriman data yang lama.

VII. *Pengujian Sensor Turbidty Pada Kekeruhan Air*

Pada pengujian ini akan dilakukan 2 kondisi yaitu pada air jernih dan air keruh yang dimana sensor turbidty akan dimasukan kedalam air untuk mengetahui tiap-tiap nilai kekeruhan pada suatu air. Pengujian pertama yaitu akses sensor pada air jernih untuk mengetahui nilai sensor pada saat kondisi air jernih, pengujian kedua akses sensor pada air keruh untuk mengetahui nilai sensor pada saat kondisi air keruh, Kedua Pengujian kondisi pada air menggunakan sensor *turbidty* dapat dilihat pada Gambar 12 dan 13 seperti berikut.



Gambar 12. pengujian pada air jernih

Pengujian dilakukan di area CAP 1 dengan menghubungkan pada SSID dan menjalankan aplikasi inSSIDer untuk mengetahui kekuatan sinyal yang didapat. Selama pengujian *client* tidak berpindah tempat selama 5 menit untuk mengetahui apakah ada perubahan sinyal selama berada di area CAP 1.



Gambar 13. pengujian pada air keruh

Data yang telah diperoleh dikumpulkan dalam sebuah tabel pengujian. Tabel data pengujian *turbidty* yang diperoleh ditunjukkan pada Tabel 3.

TABEL I  
PENGUJIAN  
TURBIDTY

Pengujian	Air Jernih	Air Keruh
1	2.95	60.88
2	3.40	63.62
3	3.65	65.44
4	3.55	64.88
5	3.39	62.30

Hasil pengujian yang diperoleh menjadi tolak ukur keberhasilan sistem yang dibangun. Hasil pengujian dibandingkan dengan analisis awal sebelum *system* baru yang dibangun.

VIII. *Data hasil Pengujian Pakan*

Pada Tabel 1 merupakan tabel uji kerja dari alat pemberi pakan ikan otomatis, yang dimana Tujuan dari pengujian ini adalah untuk melihat seberapa baik alat bekerja saat memberi makan ikan. Pengujian kerja dilakukan sesuai dengan keluaran derajat motor servo mikrokontroler, dan hasil pengujian tersebut akan dimasukkan ke dalam tabel berdasarkan hasil percobaan. Pengujian dilakukan selama 3 kali percobaan dengan waktu pukul 08:00, 14:00, 20:00. Tabel uji kerja pemberi pakan ikan dapat dilihat pada Tabel 2 sebagai berikut.

TABEL II  
TABEL UJI KERJA PEMBERI  
PAKAN IKAN

Pen guji an	waktu	Derajat motor servo	Pakan ikan yang jatuh	Waktu delay respon
1	08:00	360 deravjavn	3,7 gravm	2 detik
2	14:00	360 deravjavn	2,8 gravm	2 detik
3	20:00	360 deravjavn	3,7 gravm	2 detik

Dari pengujian 1 sampai 3 yang dapat dilihat pada pada Tabel 2 diatas terjadi perbedaan takaran pakan yang keluar dan perbedaan *respon delay* pada saat pemberian pakan dilakukan, hal tersebut dapat terjadi karena lambatnya sinyal pada saat proses pengiriman data pada alat.

IV. KESIMPULAN

Berdasarkan hasil dan pembahasan pada penelitian yang telah dilakukan dapat disimpulkan bahwa:

1. Rata-rata *bandwidth* yanag didapat pada pengamatan packet loss yaitu 10.73, pada delay 5.13 dan pada jitter 8.22, total bandwidth keseluruhan ialah 8.029 pada saat proses pengiriman data informasi.
2. Dalam pengujian *turbidty* disimpulkan bahwa semakin kecil nilai NTU (*Nephelometric Turbidity Unit*) maka semakin bersih air dan apabila nilai NTU lebih dari 25 NTU bisa dikategorikan air keruh.
3. Dari pengujian pakan terjadi perbedaan takaran pakan yang keluar dan perbedaan

respon delay pada saat pemberian pakan dilakukan, hal tersebut dapat terjadi karena lambatnya sinyal pada saat proses pengiriman data pada alat.

4. protokol MQTT mempunyai kualitas lebih tinggi dengan ukuran paket data dan delay yang lebih kecil dibandingkan protokol HTTP. Protokol MQTT dan HTTP,

#### REFERENSI

- [1] K. Indartono, B. A. Kusuma, dan A. P. Putra, "Perancangan Sistem Pemantau Kualitas Air Pada Budidaya Ikan Air Tawar," *J. Inf. Syst. Manag.*, vol. 1, no. 2, pp. 11–17, 2020, doi: 10.24076/joism.2020v1i2.23.
- [2] Admin. Dkpp, "Potensi Produksi Perikanan di Indonesia," 2018, [Online] <https://dkpp.bulelengkab.go.id/informasi/detail/artikel/potensi-produksi-perikanan-di-indonesia-26> (diakses 9 Agustus 2022).
- [3] M. S. Dr. Ir. Sri Rejeki, M. S. Restiana Wisnu Aryati, S.Pi, dan M. P. Lestari Laksmi Widowati, S.Pi, Pengantar Akuakultur. Semarang: *Undip Press Semarang*, 2019, ISBN: 978-979-097-517-0.
- [4] W. DEWANTORO dan M. B. Ulum, "Rancang Bangun Sistem Monitoring Kualitas Air Pada Budidaya Ikan Hias Air Tawar Berbasis Iot (Internet of Things)," *J. Komputasi*, vol. 9, no. 2, pp. 67–75, 2021.
- [5] Husna. R, Nasir. M, dan Hidayat. H. T., "Rancang Bangun Prototype Jemuran Berbasis Iot (Internet Of Things)," *J. Teknol. Rekayasa Inf. dan Komput.*, vol. 3, no. 2, 2020, ISSN: 2581–2882