

## Rancang Bangun Pengamatan pH Air Menggunakan Protokol MQTT

Cahyo Karya Basuki<sup>1</sup>, Dany Primanita Kartikasari<sup>2</sup>, Achmad Basuki<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>cahyokarya06@gmail.com, <sup>2</sup>dany.jalin@ub.ac.id, <sup>3</sup>abazh@ub.ac.id

### Abstrak

pH air merupakan aspek terpenting dalam bidang perikanan, terutama pada pembudidayaan ikan dalam kolam. Masalah pemantauan pH, saat ini sangat banyak dibicarakan salah satunya adalah perubahan cuaca yang membuat pihak pengelola terkadang lupa ataupun malas melakukan pengecekan. Sehingga dibutuhkan peranan teknologi yang dapat membantu mengatasi masalah tersebut, untuk menyampaikan data pH air kepada pengelola dari jarak jauh. IoT atau *Internet of Things* merupakan sebuah konsep teknologi yang penerapannya dapat digunakan untuk membantu mengatasi masalah tersebut, dimana sistem kerjanya adalah menghubungkan benda – benda fisik dengan komunikasi internet. IoT memerlukan sebuah protokol untuk menjalankan transmisi data yang berfungsi sebagai transportasi data dari sensor kepada pengguna dalam sebuah jaringan. Protokol yang digunakan adalah MQTT (*Message Queuing Telemetry Transport*) dimana protokol tersebut diharapkan dapat bekerja dalam kondisi *low bandwidth* serta *unreliable network*. Mekanisme yang diterapkan pada sistem yang dibuat menggunakan adalah *publish-subscribe*. Dengan *sensor node* pada mikrokontroler arduino sebagai *publisher*. Kemudian mikrokomputer Raspberry Pi sebagai *server* yang berfungsi sebagai *broker*, *middleware* pengontrol waktu dan penyimpanan data. *Subscriber* sebagai peminta data mengadopsi dari bahasa *java* sebagai *interface* yang sederhana kepada pengguna agar mudah dibaca. Dari implementasi yang telah dibuat didapatkan hasil pengujian sistem yang sesuai harapan dan memperoleh kinerja protokol yang merujuk kepada nilai *average latency* dan *delay* dibawah 1 ms yaitu 0,31 ms dan 0,71 ms dengan lama waktu uji 3x24 jam.

**Kata kunci:** pH air, MQTT, *Publish Subscribe*, *Raspberry Pi*, *Middleware*.

### Abstract

*Water pH is the most important aspect in the field of fisheries, especially in fish cultivation in ponds. PH monitoring problems, currently very much talked about one of them is weather changes that make the management sometimes forget or are lazy to check. So that the role of technology is needed that can help overcome the problem, to convey water pH data to managers remotely. IoT or the Internet of Things is a technology concept whose application can be used to help overcome these problems, where the system works is to connect physical objects with internet communication. IoT requires a protocol to run data transmission that functions as data transport from sensors to users on a network. The protocol used is MQTT (Message Queuing Telemetry Transport) where the protocol is expected to work in low bandwidth and unreliable networks. The mechanism applied to the system used is publish-subscribe. With a sensor node on Arduino microcontroller as a publisher. Then the Raspberry Pi microcomputer is a server that functions as a broker, time control middleware and data storage. Subscriber as the data requester adopts from java language as a simple interface to the user so that it is easy to read. From the implementation that has been made, it is obtained the results of system testing that matches expectations and obtains protocol kinerja protokolnce which refers to the average latency and delay values below 1 ms, which are 0.31 ms and 0.71 ms with a test time of 3x24 hours.*

**Keywords:** *pH water*, MQTT, *Publish Subscribe*, *Raspberry Pi*, *Middleware*.

## 1. PENDAHULUAN

Dalam forum internasional oleh Kementerian Kelautan dan Perikanan (KKP) serta Masyarakat Akuakultur Indonesia (MAI). Mengatakan bahwa Indonesia merupakan negea terbesar kedua dalam produksi ikan berdasarkan data FAO 2016 (Mufid, A.2016). Namun, dari pernyataan tersebut masih banyak tantangan pada saat ini dalam pengembangan akuakultur. Banyaknya faktor yang mempengaruhi pembudidayaan ikan yang sedang berlangsung di Indonesia. Salah satu faktor tersebut adalah kualitas pH air yang digunakan, karena akan berdampak positif dan negatif pada keberlangsungan hidup ikan. Derajat pH air yang memenuhi syarat adalah 5-8,5 untuk budidaya ikan laut dan 6,5-7,5 untuk budidaya ikan tawar (Nasa, 2015).

Pemantauan kadar pH air sangatlah penting untuk menunjang keberhasilan sebuah produksi dalam pembudidayaan. Proses budidaya ikan pada kolam outdoor juga menjadi masalah pemantauan pH yang sedang berlangsung. Salah satu faktor perubahan pH adalah kondisi cuaca, misalnya terjadi hujan ataupun kontaminasi dari luar kolam. Sehingga pemantauan pH harus mendapatkan perhatian yang ekstra untuk mengetahui kondisi yang sebenarnya pada saat itu (Asdary, M. 2017). Salah satu cara untuk mengatasi masalah tersebut adalah dengan memanfaatkan teknologi untuk membantu pemantauan kualitas pH dalam segala kondisi. Pemanfaatan teknologi yang bisa diaplikasikan pada masalah tersebut adalah IoT atau *Internet of Things*. IoT sendiri merupakan sebuah gagasan pada setiap benda-benda fisik yang terhubung dengan internet dalam satuan bentuk pemantauan dan dapat dikendalikan atau yang lainnya secara realtime (Alsaadi, E. & Tubaishat, A., 2015). *Internet of Things* terklasifikasi menjadi 3 bagian yaitu, yang pertama "*Things oriented*" berupa benda fisik seperti sensor. Kedua "*Internet Oriented*" adalah transportasi untuk saling terhubung. Terakhir adalah "*Semantic Oriented*" bagian ini merupakan salah satu tantangan teknologi IoT yang saling berkaitan dengan jumlah *things* yang terlibat dalam sebuah konsep pengaplikasian (Atzori, L. et al., 2015).

Sistem pengamatan ini memerlukan alat yang berfungsi sebagai kontrol yang berupa mikrokontroler, mikrokomputer dan akuisisi data berupa sensor yang saling terhubung

dengan jaringan wireless. Oleh karena itu, dibutuhkan sebuah protokol komunikasi yang dapat memadai. Protokol MQTT (*Message Queuing Telemetry Transport*) merupakan sebuah protokol dengan komunikasi yang ringan berbasis publisher, subscriber dengan keunggulan pada daya proses dan penyimpanan kecil (Aziz, B., 2016). Selain itu ada hal mengenai sebuah fungsi komunikasi MQTT yang berperan sebagai client dan server. Terdapat alur komunikasi yang telah diatur untuk saling bertukar pesan dengan acknowledgment (ACK). Namun sebuah fakta membuktikan dengan suatu pendapat yang berbeda dengan tujuan yang sama, hanya saja penerapan tersebut merupakan penerapan tambahan dalam pembahasan tersebut, yang mana konsep tersebut memiliki efisiensi dalam sarana komunikasi WSN, tetapi memiliki tujuan untuk menyediakan interkoneksi antar WSN untuk mengetahui dari mana data tersebut dihasilkan (Weiss, B. et al., 2016).

Dari pembahasan tersebut dapat disimpulkan bahwa, penelitian ini akan melakukan sebuah implementasi untuk membantu proses kinerja dalam pembudidayaan ikan yang memfokuskan pada pH air. Dalam proses penelitian ini, peneliti mengambil peluang untuk memanfaatkan teknologi IoT, dengan bantuan protokol MQTT yang dapat digunakan dalam proses pengamatan pH Air, yang berfungsi sebagai sarana komunikasi antar *client* dan *server*. Dengan mekanisme *Publish* dan *Subscribe*, dimana terdapat node untuk sensor dengan mikrokontroler yang berfungsi sebagai *publisher*. *Publisher* akan mengirimkan hasil akuisisi kepada broker. Broker berperan sebagai kolektor data dengan bantuan komunikasi jaringan *access point* yang telah ditentukan. *Broker* ditempatkan pada mikrokomputer Raspberry Pi, yang memiliki fungsi untuk menyimpan hasil pengiriman data dari node sensor, dengan menggunakan bantuan database *mongoDB*. Proses penyimpanan data pada *mongoDB* akan diatur oleh *Pub\_Sub\_Middleware*. Terdapat *subscriber* yang bertugas untuk mendapatkan informasi pada *broker* sesuai topik yang sama dengan *interface*, yang juga sangat mudah dimengerti oleh pengguna

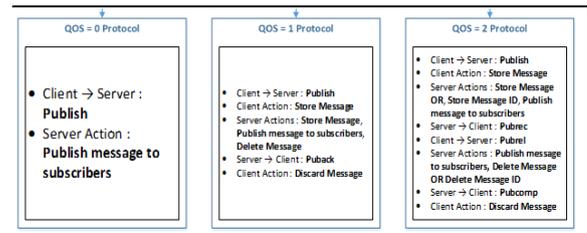
## 2. DASAR TEORI

### 2.1. Internet of Things

*Internet of Things (IoT)* merupakan sebuah penemuan yang saat ini sangat banyak diminati oleh para pengguna yang memanfaatkan kombinasi benda-benda fisik dan internet. IoT juga tergolong signifikan karena dapat membuat sebuah objek melakukan aktifitas dirinya sendiri secara digital dan menjadi sesuatu yang lebih besar dari objek itu sendiri. Arsitektur IoT terdiri dari jaringan, sistem yang kompleks dan juga keamanan yang ketat, jika ketiganya bisa dicapai maka kontrol otomatisasi dalam *Internet of Things* dan dapat berjalan dengan lancar. Komponen-komponen tersebut terdiri dari *identification, sensing, communication, computation, services* dan *semantics* (Mohammadi, m. et al., 2015).

### 2.2. MQTT

Protokol *Message Queuing Telemetry Transport* atau yang bisa disebut MQTT merupakan sebuah protokol komunikasi data M2M atau *Machine to Machine* yang berada pada layer aplikasi. MQTT berkomunikasi dengan mengirimkan data pesan yang memiliki *header* berukuran kecil yaitu hanya sebesar *2bytes* untuk setiap jenis data, sehingga dapat bekerja di dalam lingkungan yang terbatas sumber daya seperti kecilnya *bandwidth* dan terbatasnya sumber daya listrik. protokol MQTT menggunakan metode *publish/subscribe* untuk metode komunikasi. *Publish/Subscribe* sendiri adalah sebuah pola pertukaran pesan di dalam komunikasi jaringan dimana, pengirim data disebut dengan *Publisher* dan penerima data disebut dengan *Subscriber*. Pada setiap MQTT *publish* mengirim dengan salah satu dari tiga tingkatan. Tingkatan ini terkait dengan jaminan yang berbeda sehubungan pada kendala pengiriman pesan. Baik pada sisi *client* dan *broker* yang dapat memberikan tepatan dan meningkatkan kehandalan jika terjadi kegagalan dalam jaringan, *reload* aplikasi dan keadaan tidak terduga. MQTT mengandalkan *TCP*, yang memiliki kehandalan tersendiri, secara *history* tingkat *QoS* dibutuhkan untuk mengatasi kehilangan data pada jaringan *TCP* yang lebih lama dan tidak dapat diandalkan. Perbedaan *QoS* MQTT dapat dilihat pada Gambar 1.



Gambar 1. MQTT QoS

## 3. METODOLOGI

### • Studi Literatur

Studi literatur merupakan sebuah landasan pemahaman dari dasar teori yang terkait dengan refrensi serta pokok bahasan yang akan digunakan dalam penelitian.

### • Analisis Kebutuhan

Analisis kebutuhan ditunjukkan untuk melakukan sebuah analisis yang diperlukan dalam pembangunan sistem, dari segi *hardware* maupun *software*.

### • Perancangan

Perancangan merupakan tahapan untuk melakukan Gambaran alur kerja pada sistem untuk pedoman mempermudah tahap implmenetasi sistem. Perancangan pada penelitian ini dibagi ke dalam empat komponen, yakni: perancangan sensor node, perancangan broker middleware, perancangan database, dan perancangan subscriber.

### • Pengujian

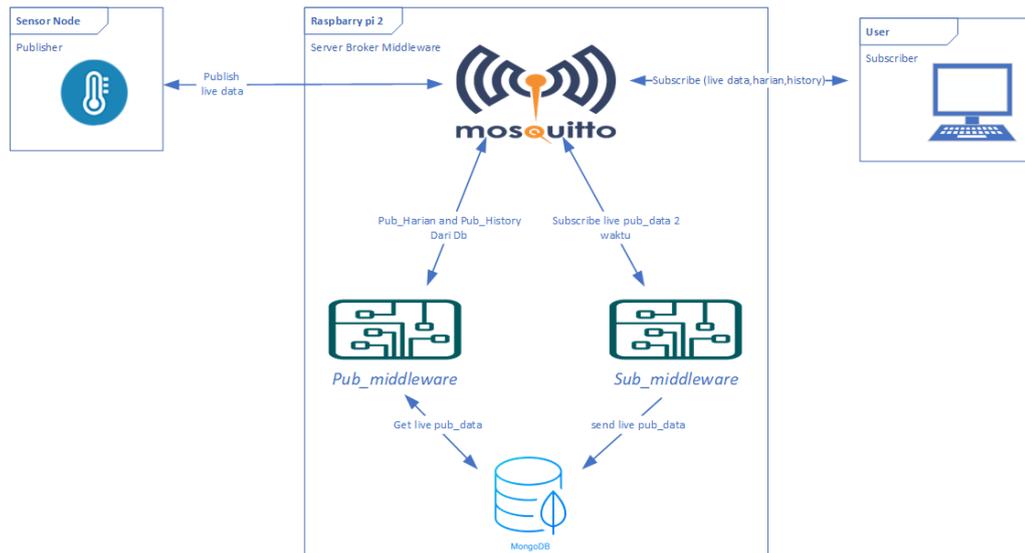
Pengujian dilakukan dengan beberapa skenario uji untuk mengetahui dampak dari *middleware* yang telah dibangun dalam sistem. Pengujian dalam penelitian ini dibagi menjadi dua bagian, yaitu fungsional sistem dan kinerja perotokol.

Parameter uji yang digunakan dalam pengujian kinerja protokol adalah *latency*, dan *delay*. Nilai dari parameter uji didapatkan ketika pengguna melakukan *publish* data sebanyak 100, 200, dan 300 secara bersamaan dan melakukan pengamatan dalam 3x24 jam.

## 4. ANALISIS KEBUTUHAN

Analisis kebutuhan merupakan tahap pengumpulan informasi tentang sistem yang dibangun. Pemantapan fitur dan hal - hal apa saja yang diperlukan sehingga sistem dapat bekerja. Kebutuhan pada rancang bangun ini dibagi mendi tiga bagian, yaitu *sensor node, server, susbcriber*.

Kebutuhan sistem pada server middleware ini dibagi menjadi beberapa bagian antara lain:



Gambar 2. Gambaran Umum Sistem

- Sistem dapat melakukan layanan pengambilan data secara *realtime*.
- Sistem dapat menyimpan data dari sensor dengan syarat waktu yang telah ditentukan.
- Sistem dapat berinteraksi dengan database.
- Sistem dapat memberikan layanan informasi data yang diperlukan oleh *subscriber*.

## 5. PERANCANGAN

### 5.1. Deskripsi Umum Sistem

Sistem yang dibuat meliputi tiga komponen utama, yaitu *sensor node*, *server* dan *subscriber*. Pertama *sensor node*, perangkat yang bertugas sebagai pengambil data dari objek kemudian melakukan *publish* data kepada *server*. *Sensor node* terdiri dari sebuah sensor pH Probe Meter yang terhubung langsung dengan *Microkomputer Arduino WeMos D1*. Kedua *server*, merupakan penengah antara data objek dengan pengguna. Dalam sistem *server* mencakupi sistem MQTT *broker*, *pub\_sub\_middleware* dan *database mongoDB* yang berjalan pada sebuah *device mikrokomputer* yang terhubung pada jaringan. Terakhir yaitu *subscriber*, komponen ini berada langsung pada pengguna untuk melakukan permintaan data yang sedang berlangsung pada objek. Untuk memperjelas Gambaran umumnya terdapat dapat dilihat pada Gambar 2.

Proses berjalannya sistem berlangsung dari *sensor node* yang melakukan koneksi kepada *access pont* dan melakukan koneksi kepada *server* dengan inialisasi *username password*

*broker*. Karena jika tidak *sensor node* tidak akan menjalin komunikasi dengan *server*. Setelah berhasil terhubung, *sensor node* melakukan translasi data sensor pH untuk dijadikan data *Json*. Kemudian melakukan *publish* kepada *broker*.

Kemudian *server* yang telah terhubung dengan *sensor node* menerima *publish* data yang berisi Id sensor, topik dan pH air. Didalam *server* terdapat fitur *pub\_sub\_middleware* yang aktif untuk mengontrol proses penyimpanan ke database dan penginformasian data ke *subscriber*. *Pub\_sub\_middleware* terbagi menjadi dua fungsi, yaitu pertama *subscriber\_middleware* bertugas untuk menyimpan data *publish* sensor dengan syarat waktu yang telah ditentukan. Kedua *publish\_middleware* untuk selalu memberikan informasi tentang data yang telah berhasil disimpan oleh *subscribe\_middleware*. *Pub\_middleware* melakukan publikasi dengan dua jalur yaitu untuk “harian”, yang memprioritaskan 2 ataupun 4 data terakhir untuk dikirim dan “histori” yang melakukan keseluruhan data yang dikirim pada database.

Terakhir adalah *subscriber*, pengguna dapat melakukan *request* kepada *server* untuk melakukan *subscribe* yang telah diperoleh dari *sensor node* melalui *broker middleware*. Pengguna tidak hanya mendapatkan data *realtime* saja, akan tetapi bisa melakukan *request* data dari pencatatan waktu *middleware* yaitu dengan menekan fitur “haria” dan “histori” seperti yang dijelaskan pada tahap *server* sebelumnya.

### 5.2. Perancangan Sensor Node

Sensor node berfungsi sebagai komponen *publisher* yang mengambil data pH langsung yang nantinya akan dikirim kepada user. Komponen sensor node bekerja dengan konfigurasi yang telah dirancang dan disematkan *code* yang berada di *mikrokontroller Arduino*. Gambaran alur tersedia pada Gambaran alur kerja sensor node publisher pada Gambar 3.



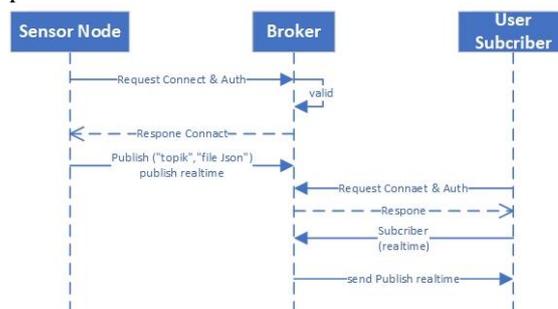
Gambar 3. Alur Kerja Sensor Node Publisher

### 5.3 Perancangan Server Broker

Pada *Server Broker Middleware*, perancangan perangkat keras menggunakan komponen *mikrokomputer Raspberry Pi 2 model B*. Sedangkan perancangan perangkat lunak menggunakan sistem operasi *Rasbian Jessie, Mosquitto, Python, MongoDB Database*. Alur kerja dari *Server Broker* terdiri dari beberapa tahapan yang berlangsung di dalamnya. Agar dapat bekerja seperti yang diinginkan pengaktifan *Broker Mosquitto* dan *PubSub\_Middleware* harus dilakukan terlebih dahulu. Setelah aktif, sistem dapat bekerja dengan alur yang telah dirancang yaitu dapat menjadi jembatan antara sensor node dan *subscriber* untuk *live data*. *Middlware* dapat menyimpan *live data* dengan sistem *sub\_middleware* yang menjadi jembatan untuk *database MongoDB*. *Middleware* juga dapat melakukan *publish* *live data* dengan sistem *pub\_middleware* dari data *live* yang telah disimpan.

#### a. Alur kerja Server Broker Real Time

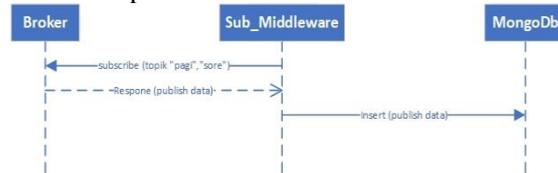
*Server broker* yang melakukan aktifitas penerimaan data dari sensor secara *real time*. *Real time* disini dimaksudkan penerimaan data *publisher* dari sensor yang langsung diteruskan kepada *subscriber*. Tujuan dari fitur ini adalah untuk memudahkan user memantau kondisi pH Air diluar waktu yang ditentukan. Gambaran alur kerja server broker real time dan live data pada Gambar 4.



Gambar 4. Alur Kerja Real Time Data

#### b. Alur kerja Server Broker Subscribe Middleware to Database.

*Server broker* yang melakukan aktifitas penyimpanan data dari sensor dengan fitur *sub\_Middleware ke Database*. Fitur ini dilakukan oleh *sub\_Middleware* dengan topik dan waktu yang telah ditentukan. Kemudian data yang telah didapatkan oleh *sub\_Middleware* akan melakukan *insert* data pada *mongoDB* sebagai *database* yang digunakan. Tujuan dari fitur ini adalah untuk memudahkan user menyimpan data yang dari sensor untuk nantinya bisa di buat sebagai acuan jika terjadi sesuatu. Gambaran alur kerja *sub\_Middleware to Database* pada Gambar 5.



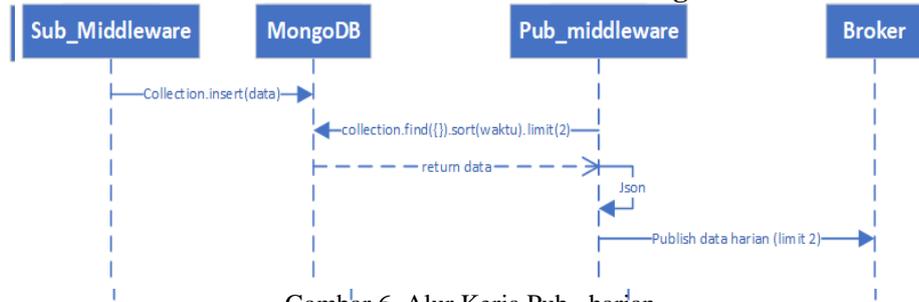
Gambar 5. Alur Kerja Subscribe Middleware ke database

#### c. Alur Kerja Server Broker Pub\_Middleware Harian.

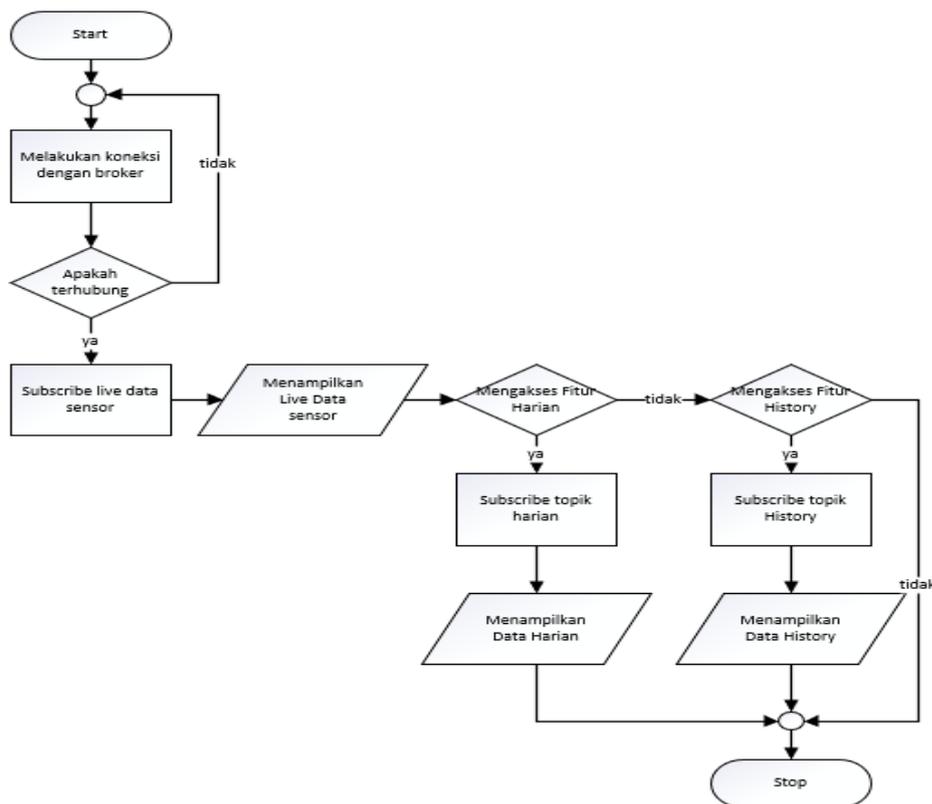
*Server broker* yang melakukan aktifitas pengiriman data dari database dengan topik "harian" berupa *publisher*. Pada fitur ini piranti *pub\_Middleware* melakukan pengambilan data dari database dengan limit data yang bisa diambil. *Pub\_Middleware* melakukan *collection.find({}).sort(waktu).limit(2)* yang artinya hanya mengambil data dari waktu dan batasan data yang bisa di ambi yaitu 2. Batasan

tersebut bisa dirubah tergantung kebutuhan sensor yang akan menjadi *publisher*. Tujuan dari fitur ini adalah untuk memudahkan user melihat data yang diambil dengan waktu yang ditentukan. Karena fitur ini mengambil data dari database yang telah disubscribe oleh *sub\_Middleware*. Gambaran alur kerja *Pub\_Middleware* harian pada Gambar 6..

d. Alur Kerja Server Broker *Pub\_Middleware History*



Gambar 6. Alur Kerja *Pub\_middleware* harian  
 Gambar 7. Alur Kerja *Pub\_Middleware History*



Gambar 8. Alur Kerja Subscriber

*Server broker* yang melakukan aktifitas pengiriman data dari *database* dengan topik “history” berupa *publisher*. Alur kerja yang diterapkan sama dengan *Pub\_Middleware* harian tetapi memiliki perbedaan. Pada *Pub\_Middleware history* pengambilan data dari *database* menggunakan *collection.find({})* dimana perintah tersebut untuk mengambil seluruh data tanpa limit yang berada di database

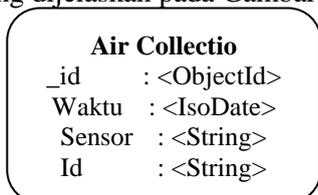
dan kemudian dipublish. Tujuan dari fitur ini adalah untuk memudahkan user mengetahui masukan data dari sensor yang telah dipublish selama kurun waktu yang lama. Dengan demikian user dapat melakukan peninjauan dikemudian hari dengan hasil data yang telah tersimpan dari hari hari yang lalu. Gambaran Alur kerja *server broker Pub\_Middleware history* pada Gambar 7.

5.4 Perancangan Subscriber

Pengguna dapat melihat atau memantau hasil pengambilan data dari *sensor node*. User akan melihat live data yang berupa *realtime* dari *publisher*, melihat data yang telah dirancang untuk disimpan pada 2 waktu yaitu pagi dan sore, dan user dapat melihat hasil dari keseluruhan data dalam waktu yang telah berlalu atau *History*. Gambaran alur kerja subscriber pada Gambar 8.

### 5.5 Perancangan Database MongoDB

sistem ini menggunakan *database MongoDB* yang berbasis dokumen yang tentunya harus disesuaikan terhadap data yang akan disimpan dan *support file Json*. Dalam proses identifikasi memerlukan satu dokumen yang dimana dalam sistem ini bernama “*air*” untuk menyimpan informasi dari *sensor node* seperti yang dijelaskan pada Gambar 9.



Gambar 9. Skema Collection

### 5.6 Lingkungan Pengujian

Tahap ini merupakan perancangan dengan suasana lingkungan. Dimana penempatan setiap *device* ditempatkan pada posisi yang telah diatur seperti pada rancangan pengujian cakupan area dengan lingkungan atau suasana yang sebenarnya. Dimana posisi sensor node berada pada ruang terbuka dengan cakupan area 5 meter dihari ke1, 10 meter dihari ke 2 dan 15 meter dihari ke 3. Tujuan dari penempatan sensor node diluar ruangan agar mengetahui nilai keaslian dari parameter uji yang telah dicantumkan. Kemudian untuk *access point* ditempatkan pada kisi-kisi ruangan agar dapat memancarkan sinyal untuk menjembatani *sensor node, server, subscriber*. Untuk penempatan *server* dan *subscriber* ditempatkan didalam ruangan dengan jarak masing-masing 5 meter dari *access point*.

## 6. HASIL PENGUJIAN KINERJA PROTOKOL

pengujian kinerja protokol dilihat dari 2 bagian yaitu hasil pengujian kinerja protokol protokol dan hasil pengujian jarak cakupan area. Pertama hasil pengujian kinerja protokol protokol yang dilihat besaran nilai yang didapat dari *latency*. Kedua hasil pengujian kinerja protokol dilihat dari besaran nilai *delay* selama 3x24 jam.

### a. Pengujian Kinerja protokol Server Middleware

Pengujian dilakukan untuk melihat nilai *latency* dengan skenario banyaknya jumlah *publisher* untuk melakukan pengiriman data dalam waktu yang bersamaan. Skenario yang digunakan adalah 100, 200 dan 300 *publisher*. Pengujian dilakukan 2 kali pada siang hari dan

malam hari. Hasil pengujian dapat dilihat pada Tabel 1.

Tabel 1. Hasil Pengujian Kinerja protokol Protokol

Banyaknya sensor node	Siang Hari pukul 12:00		Malam Hari pukul 23:59	
	Average Latency (ms)	Error Rate (%)	Average Latency (ms)	Error Rate (%)
100	0,05	0	0,12	0
200	0,135	0	0,12	0
300	0,1533	0	0,1733	0

Mengacu pada hasil yang didapat pada Tabel 1 pada uji siang hari *latency* yang diperoleh semakin banyak perangkat *sensor node* yang digunakan maka nilai yang didapat semakin tinggi. Berbeda dengan uji malam hari dimana *Latency* yang diperoleh memiliki 2 nilai yang sama pada jumlah perangkat 100 dan 200 yaitu 0,12ms namun tidak pada perangkat 300 yang jauh lebih tinggi. Dengan nilai *Error rate* yang sama yaitu 0%.

### b. Pengujian Cakupan Area

Pengujian ini bertujuan untuk mengetahui cakupan area yang dilandasi dari lokasi sebenarnya. Terdapat 3 jarak yang digunakan untuk melakukan uji cakupan area yaitu 5 meter, 10 meter dan 15 meter. Setiap jarak dilakukan pengujian selama 24 untuk mengetahui rata-rata *delay* yang diperoleh. Berikut hasil pengujian jarak yang telah dilakukan pada Tabel 2. Mengacu pada hasil pengujian cakupan area Tabel 2. Pada cakupan area 5 meter pada hari ke-1 dapat dilihat bahwa nilai *delay* yang diperoleh selama 24 jam adalah 0,42ms. Kemudian hasil tersebut selanjutnya difilter menjadi per-12 Jam sehingga diperoleh hasil 0,48ms pada 12 Jam (06:00-17:59) dan 0,33ms pada 12 Jam (18:00-05:59). Pada hari ke-1 kondisi cuaca pada siang hari cerah dengan suhu 34°C dan pada malam dengan suhu 24°C yang berangin.

Pada cakupan area 10 meter pada hari ke-2 dapat dilihat bahwa nilai *delay* yang diperoleh selama 24 jam adalah 1,01ms. Kemudian hasil tersebut selanjutnya difilter menjadi per-12 jam sehingga diperoleh hasil 0,90ms pada 12 Jam (06:00-17:59) dan 1,12ms pada 12 Jam (18:00-05:59). Pada hari ke-1 kondisi cuaca pada siang hari cerah mendung dengan suhu 34°C dan pada malam berawan dengan suhu 23°C yang berangin.

Tabel 2. Hasil Pengujian Delay dengan Jarak

Jarak	Average Delay (ms)		
	24 Jam	12 Jam (06:00-17:59)	12 Jam (18:00-05:59)
5 Meter	0,42	0,48	0,33
10 Meter	1,01	0,90	1,12
15 Meter	0,82	0,83	0,80

Pada cakupan area 15 meter pada hari ke-3 dapat dilihat bahwa nilai *delay* yang diperoleh selama 24 jam adalah 0,82ms. Kemudian hasil tersebut selanjutnya difilter menjadi per-12 Jam sehingga diperoleh hasil 0,83ms pada 12 Jam (06:00-17:59) dan 0,80ms pada 12 Jam (18:00-05:59). Pada hari ke-1 kondisi cuaca pada siang hari cerah berawan dengan suhu 34°C dan pada malam berawan dengan suhu 24°C yang berangin.

Dari kedua hasil uji pada cakupan area untuk kebutuhan sistem yang dibuat dapat ditarik kesimpulan bahwa perubahan jarak dan perubahan cuaca berpengaruh pada proses pertukaran data yang terjadi antara *client* dan *server* pada sistem.

### 7. KESIMPULAN

Dari hasil implementasi yang telah dibuat disimpulkan bahwa permasalahan yang terjadi pada pemantauah pH air dapat memanfaatkan bantuan teknologi IoT dengan protokol MQTT. Sehingga dapat melakukan pemantauan dengan segala kondisi cuaca maupun kondisi air yang berubah karena faktor lain. Pada sistem yang dibuat penerapan *pub\_sub\_middleware* untuk melakukan penyimpanan data serta melakukan informasi kepada pengguna berjalan sesuai harapan. Sistem tersebut telah diuji dengan melihat kinerja protokol yang mampu menangani hingga 300 pengguna dalam waktu bersamaan dengan menghasilkan rata-rata *latency* dibawah 1 ms yaitu 0,31 ms. Sistem ini juga diuji dengan kinerja protokol yang dilihat dari cakupan areanya. Dengan skenario 5, 10 dan 15 meter dalam kurun waktu 3x24 kinerja protokol tersebut menghasilkan rata-rata *delay* dibawah 1 ms yaitu 0,17 ms.

Dengan demikian penambahan fitur *pub\_sub\_middleware* juga memiliki fungsi khusus yang berperan sebagai pengontrol dan perantara antara pengguna dan perangkat lot. Tetapi penambahan fitur ini tidak membuat kinerja perangkat IoT maupun protokol mendapatkan mengalami permasalahan yang signifikan.

### 8. DAFTAR PUSTAKA

- Adi, H. K., P, E. S. & Amron, K., 2016. PENGUMPULAN DATA MENGGUNAKAN METODE PUBLISH SUBSCRIBE PADA NODE SENSOR DALAM WIRELESS MESH NETWORK. Volume 8.
- Alsaadi, E. & Tubaishat , A., 2015. Internet of Things: Features, Challenges, and Vulnerabilities. *International Journal of Advanced Computer Science and Information Technology (IJACSIT)*., Volume 04.
- Asdary, M., 2017. Permasalahan pemantauan pH air dan pemanfaatan teknologi IoT. [Cahyo karya]. Balai Perikanan Budidaya Air Payau (BPBAP Situbondo). 27 Juli 2017, 10:23.
- Atzori, L., Iera, A. & Morabito, G., 2015. The Internet of Things: A survey. *e-journal*, 54 (2010)(15), pp. 2787-2805.
- Aziz, B., 2016 . A formal model and analysis of an IoT protocol. 36(P1), pp. 49-57.
- Barata, D., Louzada, G., Carreiro, A. & Damasceno, A., 2013. System of acquisition, transmission, storage and visualization of Pulse Oximeter and ECG data using Android and MQTT.
- Manyika, J. et al., 2015. *THE INTERNET OF THINGS: MAPPING THE VALUE BEYOND THE HYPE*, s.l.: McKinsey&Company.
- Mohammadi, m. et al., 2015. Internet of Things: A Survey on Enabling Technologies, Protocols and Applications.
- Mufid, A., 2016. *INDUSTRI AKUAKULTUR UNTUK PERTUMBUHAN EKONOMI BERKELANJUTAN*. [Online] Available at: <http://aquaculture-mai.org/archives/1643> [Diakses pada 26 12 2016].
- Nasa, D. S., 2015. *Panduan Pengelolaan Air Budidaya Ikan*. [Online] Available at: <http://www.viternaplus.com/2015/09/panduan-pengelolaan-air-budidaya-ikan.html> [Diakses pada 17 03 2017].
- Weiss, B. et al., 2016. A Publish/Subscribe Messaging System For Wireless Sensor Communication.