JRAEE
Journal of Robotics, Automation, and Electronics Engineering
Vol. 1, No. 1, 2023, pp. 45-56
https://journal.uny.ac.id/v3/jraee

MQTT and CoAP Communication Protocol Analysis in Internet of Things System for Strawberry Hydroponic Plants

Nilam Andi Safitri[a,1], Ardy Seto Priambodo[a,2,*]

[a] Department of Electrical and Electronics Engineering, Vocational Faculty, UNY

[1] nilamandi.2019@student.uny.ac.id; [2] ardyseto@uny.ac.id
* Corresponding Author

ARTICLE INFO

Article History
Received 27 July 2023
Revised 3 August 2023
Accepted 8 August 2023

Keywords
Hydroponic,
Internet of Things,
MQTT,
CoAP,
Messaging Protocol,

ABSTRACT

Strawberry is a fruit that is widely consumed and cultivated globally because of its richness in nutrients, vitamins, and minerals. However, several challenging factors that need to be faced in strawberry cultivation are temperature, humidity, lighting, etc. The author focuses on the application of IoT in *container* strawberries, especially in high-temperature advertisements. Where *Hardware, Back End* and *Front End* are important frameworks for developing the system. In developing a system that utilizes IoT technology optimally and efficiently, it is necessary to consider the lines of communication. Without an eligible communication, the device cannot be connected smoothly. In this study the authors will focus on *Back End* to analyze the performance of the two protocols MQTT and CoAP. The MQTT protocol is better used for IoT-based hydroponic plant automation systems because the MQTT protocol can configure messages/packets sent. MQTT can provide efficient and reliable communications in the IoT environment. In addition, the development of the *prototype* of strawberry cultivation using the *container* aims to allow strawberry farmers to control the plants without having to be near the plants.

Stroberi merupakan salah satu buah *subtropis* yang banyak dikonsumsi dan dibudidayakan secara global karena kekayaan nutrisi, vitamin, dan mineralnya. Namun, beberapa faktor tantangan yang perlu dihadapi dalam budidaya stoberi adalah seperti suhu, kelembapan, penyinaran, dll. Penulis berfokus pada penerapan IoT dalam *container* stoberi khususnya di iklam yang bersuhu tinggi. Dimana *Hardware, Back End* dan *Front End* merupakan kerangka penting untuk mengembangkan sistem tersebut. Dalam menggembangkan sistem yang memenfaatkan teknologi IoT yang optimal dan efisien, diperlukan pertimbangan terhadap jalur komunikasi. Tanpa adanya komunikasi yang eligible, maka perangkat tidak dapat terkonekai dengan lancar. Pada penelitian ini penulis akan berfokus pada *Back End* untuk menganalisis kinerja dua protokol MQTT dan CoAP. Protokol MQTT lebih baik digunakan untuk sistem otomasi tanaman hidroponik berbasis IoT karena protokol MQTT dapat mengkonfigurasi pesan/ packet yang terkirim. MQTT memiliki kemampuan untuk menyediakan komunikasi yang efisien dan handal dalam lingkungan IoT. Selain itu pengembangan *prototype* budidaya stoberi dengan *container* ini bertujuan agar para petani stoberi bisa mengontrol tanaman tanpa harus berada didekat tanaman.

1. Pendahuluan

Indonesia merupakan negara agraris penghasil komoditas pertanian dan memiliki potensi ekspor yang cukup besar. Untuk saat ini permintaan ekspor dan daya saing di pasar internasional dan do- mestik terbilang rendah. Sub sektor hortikultura salah satunnya buah-buahan memiliki peluang yang cukup besar untuk memasuki pasar internasional maupun lokal. Stroberi merupakan salah satu buah subtropis yang banyak dikonsumsi dan dibudidayakan secara global karena kekayaan nutrisi, vita- min, dan mineralnya. Berdasarkan pada Badan Pusat Statistik tentang produksi tanaman buah-buahan menyebutkan bahwa produksi stroberi terus meningkat dari tahun ke tahun. Produksi stroberi di In- donesia mencapai 9.860-ton pada 2021, jumlah tersebut meningkat 18,08% dibandingkan pada tahun sebelumnya yang hanya 8.350 ton. Namun, beberapa faktor tantangan yang perlu dihadapi dalam budidaya stroberi adalah seperti suhu, kelembapan, penyinaran, dll.

Atas dasar diatas, penulis mempunyai gagasan dalam pengembangan teknologi pada bidang budidaya stroberi. Teknologi yang penulis kembangkan berupa *monitoring* dan *controlling* terhadap kondisi tanaman stoberi berupa suhu dan kelembaban ruangan, EC, PH, Kelembapan tanah serta suhu larutan nutrusi. Semua parameter tersebut dapat mempengaruhi pertumbuhan dan hasil panen stroberi. *Monitoring* dan *controlling* ini berbasis IoT. Dimana *Hardware, Back End* dan *Front End* merupakan kerangka penting untuk mengembangkan sistem tersebut. Pada penelitian ini penulis akan berfokus pada *Back End.*

Dalam mengembangkan sistem yang memenfaatkan teknologi IoT yang optimal dan efisien, diperlukan pertimbangan atau perhatian terhadap parameter. Salah satu parameter tersebut ialah jalur komunikasi. Tanpa adanya komunikasi yang *eligible*, maka perangkat yang terhubung tidak dapat berinteraksi dengan lancar. Yang dapat menyebabkan sistem tidak berfungsi dengan baik dan tidak dapat melakukan pertukaran data. Komunikasi yang *eligible* sangat penting untuk memastikan bahwa IoT dapat berjalan dengan efisien.

Untuk menentukan jalur komunikasi yang efisien, diperlukan analisis terhadap messaging protokol yang akan diimplementasikan pada sistem tersebut. Oleh sebab itu, penulis akan melakukan analisis terhadap metode messaging protokol *Constrained Application Protocol (CoAP)* dan *Message Queue Telemetry Transport (MQTT)* pada tanaman hidroponik stoberi berbasis IoT. Analisis ini bertujuan untuk menguji fitur dan kemampuan dari kedua protokol tersebut.

Protokol MQTT lebih baik digunakan untuk sistem otomasi tanaman hidroponik stroberi berbasis IoT karena protokol MQTT dapat mengkonfigurasi pesan/ packet yang terkirim. Jika jaringan inter-net tidak stabil maka MQTT menyimpan pesan yang belum diproses oleh server dan akan terkirim ulang jika internet pada jaringan yang baik. Selain itu, MQTT menyediakan label pesan dimana san-gat dibutuhkan dalam sistem ini sebagai konfirmasi apakah pesan sudah terkirim atau belum. MQTT memiliki kemampuan untuk menyediakan komunikasi yang efisien dan handal dalam lingkungan IoT. Sedangkan protokol CoAP kurang layak untuk sistem hidroponik stroberi dikarenakan tidak mem- berikan dukungan dalam hal pemberian label pesan dengan jenis atau metadata lain untuk membantu user memahaminya. Tetapi, jika memiliki *bandwidth* yang kecil, protokol CoAP mampu diunggulkan karena dapat mengirimkan pesan lebih cepat.

Selain itu pengembangan *prototype* budidaya stroberi dengan *container* ini bertujuan agar para petani stroberi bisa mengontrol tanaman tanpa harus berada didekat tanaman. Cukup menggunakan *smartphone* yang sudah terintegrasi pada perangkat *controller* yang sudah dipasang di dalam *container* stroberi sehingga proses budidaya stroberi lebih cepat, efisien dan mendapatkan hasil panen yang lebih baik serta dapat memenuhi permintaan ekspor dan daya saing di pasar internasional maupun domestik.

2. Pendekatan Pemecahan Masalah

2.1. Message Queuing Telemetry Transport (MQTT)

Protokol berlangganan berbasis topik, seperti MQTT, biasanya menyajikan arsitektur fisik yang terdiri dari tiga jenis node yaitu *publisher, subscriber* dan *broker*.

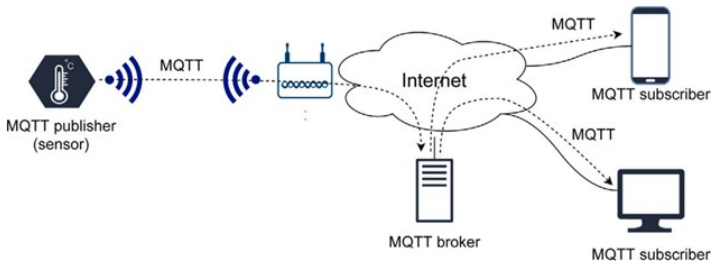


Fig. 1. MQTT system overview

- MQTT *Publisher* yaitu setiap *client* yang membuat/ mengirimkan pesan yang terkait dengan topik apapun (sensor) ke *broker*.
- MQTT *Subscriber* yaitu setiap *client* yang akan meminta/ menerima informasi berlangganandari *broker* sesuai dengan topik yang di *subscribe*.
- MQTT *Broker* yaitu perangkat *server* yang bertanggung jawab untuk menerima pesan dari *publisher* dan meneruskan ke *subscriber* sesuai dengan topik.

2.2. Constrained Application Protocol (CoAP)

Bedasarkan model *server/ client*, CoAP mengandalkan REST sehingga dapat meningkatkan kemampuan interoperanya.

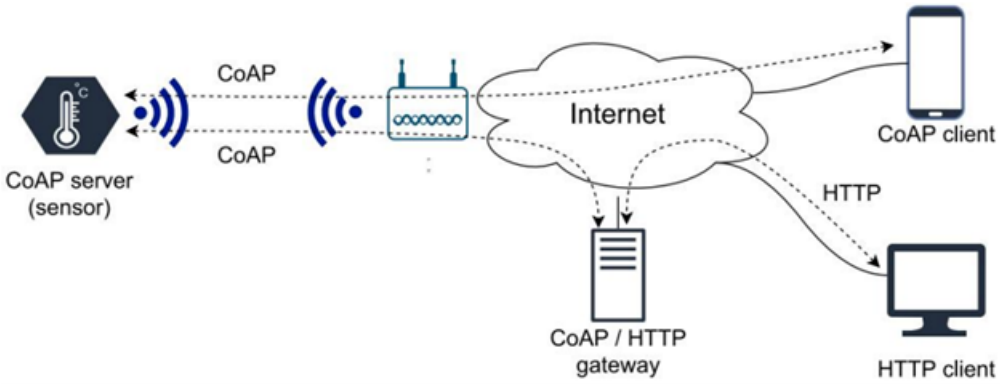


Fig. 2. CoAP system overview

Dalam jaringan CoAP memiliki 2 jenis node yang terdapat pada gambar 2.10. Server CoAP, biasanya perangkat yang dibatasi (*sensor dan actuator*) yang dapat dikontrol menggunakan REST API dan *client* CoAP. Dimana perangkat yang akan mengambil data atau meminta tindakan dari *server*. *client* HTTP juga dapat Ber Komunikasi dengan *server* CoAP menggunakan protokol CoAP.

3. Rancangan Penelitian

Pada bab ini dijelaskan langkah- langkah yang dilakukan untuk menyelesaikan masalah pada penelitian ini. Langkah- Langkah tersebut meliputi Studi Literatur, Analisis Kebutuhan, Arsitektur Sistem Perancangan, Implementasi, Pengujian dan Analisis, Pengambilan Kesimpulan. Rancangan penelitian yang digunakan pada penelitian ini digambarkan flowchart pada gambar 3.

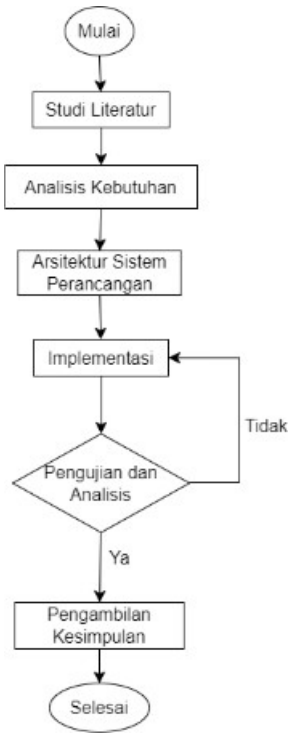


Fig. 3. Rancangan penelitian

Studi Literatur ialah kegiatan yang dilakukan untuk menentukan objek penelitian yang sesuai dengan topik yang diambil yang berfungsi sebagai referensi untuk penelitian yang akan dilakukan. Studi literatur digunakan untuk mendukung penelitian dalam penyelesaian masalah agar tercapainya tujuan penelitian. Teori pendukung dapat diperoleh dari jurnal, buku dan penelitian yang sudah dilakukan sebelumnya. Studi literatur dalam penelitian ini yaitu *Internet of Things* (IoT), protokol CoAP, protokol MQTT.

Analisis kebutuhan didapatkan dari pembahasan studi literatur untuk perancangan kebutuhan dalam melakukan penelitian. Kebutuhan yang diperlukan dalam penelitian terbagi menjadi dua yaitu kebutuhan perangkat keras *(hardware)* dan kebutuhan perangkat lunak *(software)*. Kebutuhan Perangkat Keras *(Hardware)*

- Laptop
- Raspberry Pi, Pada penelitian ini penulis menggunakan Raspbery Pi model 3B+ sebagai perangkatyang difungsikan sebagai *server*.

Kebutuhan Perangkat Lunak *(Software)* Perangkat lunak yang dibutuhkan dalam implementasi sistem dalam penelitian ini yaitu:

- *VNC Viewer*, Pada penelitian ini VNC *Viewer* digunakan untuk mengakses raspberry pi dalam *Local Area Network* (LAN) dengan jarak dekat ataupun jarak jauh.
- *Mosquitto*, *Mosquitto* merupakan sebuah MQTT broker *open source* dan dalam penelitian ini digunakan sebagai perantara pengirim dan penerima pesan pada jaringan MQTT.
- Python, Python merupakan bahasa pemrograman interpretative dengan perancangan yang berfokus pada tingkatan pembacaan kode dan dalam penelitian kali ini python digunakan sebagai ba- hasa pemrograman dengan komunikasi menggunakan MQTT lokal dan mengirimkan data hasil pemrosesannya ke *cloud*.
- Ubuntu, Pada penelitian ini ubuntu digunakan sebagai pengganti raspberry pi. Raspberry dan ubuntu sama-sama berbasis linux. Ubuntu yang dipakai adalah jenis WSL, merupakan aplikasi windows yang dapat menjalankan paket-paket aplikasi dari linux. Ubuntu digunakan sebagai pengembangan program CoAP sebelum diterapkan ke *server* (Raspberry pi).

Arsitektur sistem perancangan merupakan gambaran dari *project* yang akan diimplementasikan

pada penelitian tugas akhir. Skema project ini akan dibagi menjadi sub tiga bagian diantaranya *hardware,back end,* dan *front end*. Dimana masing-masing pembagian *project*-nya untuk *hardware* sendiri nantinya akan mengoperasikan sistem dari sisi *hardware* untuk mendapatkan data sensor menggu- nakan *mikrokontroller*.

Impelentasi adalah proses pembuatan sistem yang disesuaikan dengan perancangan yang telah dibuat sebelumnya dan berpacu pada studi literatur. Dalam penelitian ini langkah pertama dalam proses implementasi yaitu multi-protokol sesuai dengan arsitektur yang sudah dibuat kemudian men-jalankannya di Raspberry Pi. Lalu mengimplementasikannya dalam topologi jaringan untuk proses pengujian.

Pengujian dilakukan bertujuan untuk mengetauhi apakah implementasi yang dikembangkan su-dah sesuai dengan kebutuhan fungsional untuk kinerja dalam hal *monitoring* pengiriman data dari MQTT ke *server*, *monitoring* pengiriman data dari CoAP ke *server*. Dari data hasil pengujian akan dianalisis untuk mengetauhi tingkat in- teroperabilitas dan kinerja sistem dalam pengiriman data yang dihadapkan dengan beberapa kondisijaringan.

3.1. Perancangan Alur Komunikasi

Perancangan alur sistem menggunakan server multi-protokol yang dikembangkan dapat dilihat pada gambar 4.

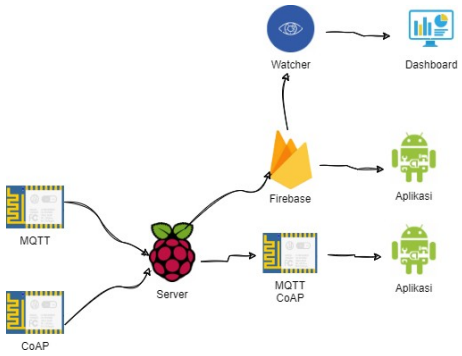


Fig4 Arsitektur sistem perancangan alur komunikasi sistem

Penjelasan pada gambar 4 yaitu :

a. Perangkat sensor *subscribe* data ke server.
b. MQTT mengirimkan data sensor *atlas scientific, DHT* dan *actuator* setiap 10 detik sekali.
c. CoAP mengirimkan data sensor *atlas scientific, DHT* dan *actuator* setiap 10 detik sekali.
d. Aplikasi menerima data yang telah dikirimkan secara lokal.
e. Server juga mengirim data ke *cloud firebase* untuk diterima aplikasi.
f. Program *weatcher* bertugas menerima data dan menyimpan data sensor ke dalam basis dataMySQL untuk analisis data yang akan ditampilkan di *dashboard*.
g. Aplikasi dapat diakses menggunakan *smartphone* dan *dashboard* munggunakan *browser*.

4. Hasil dan Pembahasan

4.1. Implementasi Pengembangan Alur Komunikasi Sistem

Pada pengujian kali ini keseluruhan alat bekerja dengan cara dihubungkan pada kesatuan jaringan WiFi yang sama antara sensor *(hardware), server (software)*, dan aplikasi. Selama pengujian hard- ware dinyalakan selama 3 jam dengan pengambilan data selama 5 menit. Ketika alat dinyalakan sensor akan membaca nutrisi tanaman, kondisi sekitar tanaman hidroponik stoberi. Data yang su- dah didapatkan akan dikirimkan ke *server* (Raspberry Pi), data tersebut diolah oleh server menjadi tipe data JSON dan dikirimkan pada protokol MQTT, CoAP dan *Cloud (firebase)*. Data yang akan dikirimkan sensor akan diterima aplikasi melalui dua protokol yang berbeda, yaitu komunikasi lokal dengan MQTT, *cloud* dengan *firebase*. *Server* akan melakukan *subscribe* data

sensor lalu mengir- imkannya ke *cloud firebase* untuk diterima aplikasi dan akan ditampilkan berupa data sensor, lalu terdapat *dashboard* yang akan menampilkan data sensor secara *realtime*.

Fig. 5. Pengembangan sistem otomasi hidroponik stroberi

4.2. Evaluasi Kinerja Pengembangan Prototype

Bedasarkan implementasi *prototype* yang sudah dibuat dalam kondisi nyata untuk pertanian stroberi,*prototype* mampu melakukan pengontrolan terhadap pemberian air, nutrisi secara otomatis dan dapat dilakukan pemonitoran menggunakan aplikasi secara berkala serta dapat memudahkan para petani hidroponik dengan bantuan teknologi IoT.

4.3. Evaluasi Kinerja Protokol MQTT dan CoAP

Berikut ditampilkan grafik komparasi *packet loss, delay, CPU Usage dan Battery Usage* dari tabel yang menunjukkan hasil uji coba pada jaringan MQTT dan CoAP.

Evaluasi *Packet Loss* Protokol MQTT dan CoAP

Pada gambar 6 akan ditampilkan perbandingan *packet loss* antara MQTT dan CoAP dengan topik/URI Actuator dan variasi waktu sebesar 10, 20, 30, 40, 50, dan 60 detik (s). Untuk topik/

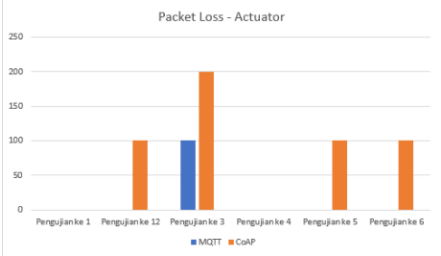


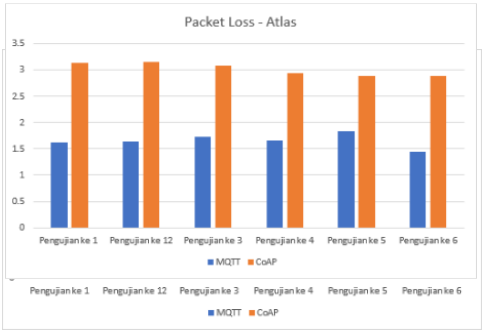
Fig. 6. Grafik packet loss sensor actuator

URI *Actuator* pada variasi waktu 10, 20, 30, 40, 50, 60 detik (s) dan adanya 6 parameter dengan total 30 parameter. Didapatkan hasil jika pada protokol MQTT hasil % error yaitu terdapat pada pengujian ke- 3 yaitu pada parameter pompa pH dimana saat kondisi sudah dimatikan namunhasil data yang diterima masih dalam kondisi menyala. Dalam hal itu, *packet* yang error dengan perhitungan (total parameter sebanyak 30 dibagi %error 100) dan didapatkan hasil yaitu 3.33%dengan kategori kondisi bagus. Sedangkan untuk protokol CoAP hasil % error didapatkan padapengujian ke- 2, 3, 5, dan 6. packet yang error dengan perhitungan (30 dibagi 500 yaitu 0.06) dan didapatkan hasil yaitu sebesar 6% dengan kategori kondisi bagus.

Pada gambar 7 akan ditampilkan perbandingan *packet loss* antara MQTT dan CoAP dengan topik/URI Atlas dan variasi waktu sebesar 10, 20, 30, 40, 50, dan 60 detik (s). Untuk topik/URI Atlas pada variasi waktu 10, 20, 30, 40, 50, 60 detik (s) dan adanya 6 parameter dengan total 12 parameter. Didapatkan hasil jika pada protokol MQTT hasil % error yaitu terdapat pada semua pengujian sensor pH dimana data yang diterima terdapat selisih sedikit dengan data yang

dikirimkan. Dalam hal itu, *packet loss* yang didapatkan pada protokol MQTT yaitu 9.92% dengan kategori kondisi bagus. Sedangkan untuk protokol CoAP hasil % error terdapat pada semua pengujian baik pH maupun EC dan didapatkan sebesar 18.07% dengan kategori kondisi sedang.

Fig. 7. Grafik packet loss Atlas



Pada gambar 8 akan ditampilkan perbandingan *packet loss* antara MQTT dan CoAP dengan topik/URI DHT dan variasi waktu sebesar 10, 20, 30, 40, 50, dan 60 detik (s). Untuk topik/

Fig. 8. Grafik packet loss sensor DHT

URI DHT pada variasi waktu 10, 20, 30, 40, 50, 60 detik (s) dan adanya 6 parameter dengan total 12 parameter. Didapatkan hasil jika pada protokol MQTT hasil % error yaitu terdapat pada semua pengujian sensor *humidity* dimana data yang diterima terdapat selisih sedikit den- gan data yang dikirimkan. Dalam hal itu, *packet loss* yang didapatkan pada protokol MQTT yaitu 1.18% dengan kategori kondisi sangat bagus. Sedangkan untuk protokol CoAP hasil % error terdapat pada semua pengujian baik *humidity* maupun *temperature* dan didapatkan sebesar 43.48% dengan kategori kondisi jelek.

Dari hasil rata-rata *packet loss* diatas, terlihat bahwa protokol MQTT dan CoAP memiliki besaran paket yang jauh berbeda. Dimana terbukti jika % error paket MQTT lebih ringan dibadingkan dengan CoAP dan CoAP lebih besar dibandingkan MQTT. Hal ini terjadi karena CoAP merupakan transmisi komunikasi berjenis *connectionless* yang menyebabkan paket akan terus dikirimkan meskipun penerima tidak menerima pesan sebelumnya. Variasi *packet loss* pada

protokol CoAP menunjukkan bahwa tetap akan ada pesan yang terkirim meskipun banyak pesan yang tidak sampai kepada penerima. Sedangkan protokol MQTT komunikasi berjenis *con*nectionful, sehingga paket akan dipastikan untuk sampai ke penerima. Apabila belum sampai maka server MQTT akan menyimpan pesan. Hal ini yang menyebabkan sekidit adanya *packet loss* pada MQTT.

Evaluasi *Delay* Protokol MQTT dan CoAP

Pada gambar 9 akan ditampilkan perbandingan *delay* antara MQTT dan CoAP dengan topik/URI *Actuator* dan variasi waktu sebesar 10, 20, 30, 40, 50, dan 60 detik (s). Untuk topik/ URI Actuator pada variasi waktu 10, 20, 30, 40, 50, 60 detik (s) dan adanya 6 parameter dengan total30 parameter

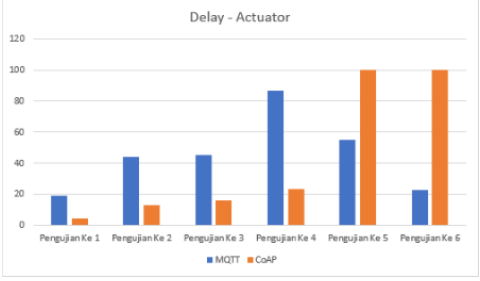


Fig. 9. Grafik delay actuator

Didapatkan hasil rata-rata dengan menjumlah semua total *delay* protokol MQTTdari data diatas adalah (19.01 + 21.15 + 22.99 + 24.97 + 20.22 + 31.44 + 31.44 + 23.74 + 26.40 + 28.49 + 22.82) : 60 = 4.5445 *second*. Sedangkan untuk protokol CoAP didapatkan hasil (04.68 + 07.02 + 05.62 + 09.75 + 06.44 + 09.60 + 12.39 + 13.85 + 15.07 + 10.17 + 13.27) : 60 = 1.7977 *second*.
Pada gambar 10 akan ditampilkan perbandingan *delay* antara MQTT dan CoAP dengan topik/URI Atlas dan variasi waktu sebesar 10, 20, 30, 40, 50, dan 60 detik (s). Untuk topik/ URI Atlas

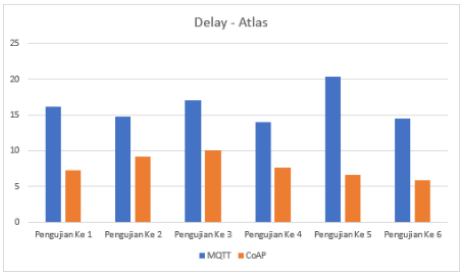


Fig. 10. Grafik aelay Atlas

pada variasi waktu 10, 20, 30, 40, 50, 60 detik (s) dan adanya 6 parameter dengan total 30 parameter. Didapatkan hasil rata-rata dengan menjumlah semua total *delay* protokol MQTT dari data diatas adalah (16.11 + 14.71 + 17.07 + 14.02 + 20.33 + 14.49) : 60 = 1.6122 *second*. Sedangkan untuk protokol CoAP didapatkan hasil (07.25 + 09.21 + 10.06 + 07.63 + 06.59 + 05.86) : 60 = 0.7767 *second*.

Pada gambar 11 akan ditampilkan perbandingan *delay* antara MQTT dan CoAP dengan topik/URI DHT dan variasi waktu sebesar 10, 20, 30, 40, 50, dan 60 detik (s). Untuk topik/ URI Actuator pada variasi waktu 10, 20, 30, 40, 50, 60 detik (s) dan adanya 6 parameter dengan total 30 parameter.

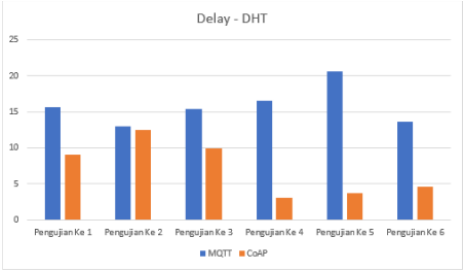


Fig. 11. Grafik delay DHT

Didapatkan hasil rata-rata dengan menjumlah semua total delay protokol MQTT dari data diatas adalah (15.60 + 13.01 + 15.42 + 16.50 + 20.53 + 13.60) : 60 = 1.5777 *second*. Sedangkan untuk protokol CoAP didapatkan hasil (09.10 + 12.47 + 09.92 + 03.07 + 03.67 + 04.56) : 60 = 1.4263 *second*.

Dari grafik *delay* diatas, terlihat dengan jelas bahwa pada protokol MQTT dan CoAP memi-liki variasi nilai yang berbeda. Rata-rata *delay* dari variasi topik yang berbeda pada pengujiansistem yaitu MQTT lebih besar dibandingkan CoAP. Rata-rata *delay* yang dihasilkan CoAP lebih stabil dibandingkan MQTT. Hal tersebut terjadi karena *client response* CoAP akan terus mengirimkan *response* walaupun *client receiver* belum menerima pesan tersebut. Sedangkan protokol MQTT, *publisher* akan menunggu acknowledge dari *client* penerima sebelum mengir- imkan pesan berikutnya.

Evaluasi *CPU Usage* Protokol MQTT dan CoAP

Pada gambar 12 akan ditampilkan *CPU usage* protokol MQTT dengan topik/URI Actuator variasi waktu sebesar 10, 20, 30, 40, 50, dan 60 detik (s).

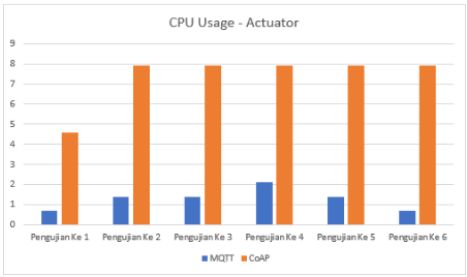


Fig. 12. Grafik *CPU usage* sensor actuator

Dari grafik pengggunaan CPU untuk protokol MQTT memiliki nilai terbesar yaitu 0.7%. Rata - rata untuk penggunaan CPU protokol MQTT sensor *actuator* sebesar 7.7%.Hal tersebut terjadi jika kondisi dinyalakan maka CPU akan bekerja, dan jika actuator dimatikan maka penggunaan CPU 0.0%. Sedangkan untuk protokol CoAP yaitu sebesar 44.1%.

Pada gambar 13 akan ditampilkan *CPU usage* protokol MQTT dengan topik/URI Atlas variasi waktu sebesar 10, 20, 30, 40, 50, dan 60 detik (s).

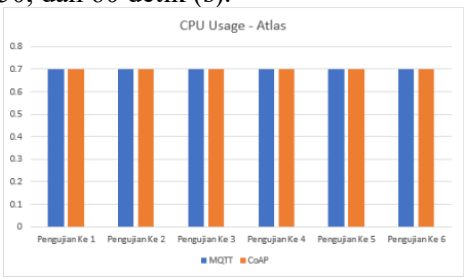


Fig. 13. Grafik *CPU usage* sensor Atlas

Dari grafik pengggunaan CPU untuk protokol MQTT dan CoAP memiliki hasil yang konstan yaitu 0.7%. Sedangkan untuk rata-rata penggunaan CPU protokol MQTT dan CoAP yaitu 4.2%.

Pada gambar 14 akan ditampilkan *CPU usage* protokol MQTT dengan topik/URI DHT variasi waktu sebesar 10, 20, 30, 40, 50, dan 60 detik (s).

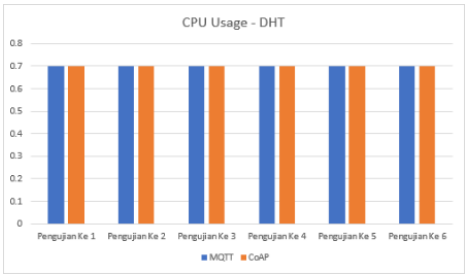


Fig. 14. Grafik *CPU usage* sensor DHT

Dari grafik pengggunaan CPU untuk protokol MQTT dan CoAP memiliki hasil yang konstan yaitu 0.7%. Sedangkan untuk rata-rata penggunaan CPU protokol MQTT dan CoAP yaitu 4.2%.

Presentase rata-rata total CPU usage pada pengujian diatas yaitu protokol CoAP lebih besar dibandingkan MQTT. Rata- rata penggunaan CPU yang dihasilkan MQTT lebih kecil, karena penggunaan CPU hanya dijalankan untuk menjalankan broker MQTT yang bertugas melakukan subscribe dan CPU akan memproses penguraian pesan MQTT. Sedangkan penggunaan CPU CoAP digunakan untuk menjalankan server dan client secara bersamaan, termasuk pengiriman dan penerimaan permintaan serta tanggapan CoAP (acknowledge).

Evaluasi *Battery Usage* Protokol MQTT dan CoAP

Pada gambar 15 akan ditampilkan perbandingan *battery usage* antara MQTT dan CoAP dengan topik/URI *Actuator, Atlas, dan DHT* dengan waktu pengujian selama 5 menit.

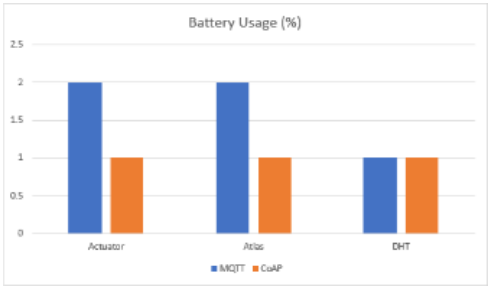


Fig. 15. Grafik battery usage actuator, Atlas, DHT

Dari grafik penggunaan *battery* diatas, pada protokol MQTT dan CoAP memiliki nilai yang hampir sama. Penggunaan battery MQTT dengan jumlah nilai 5%. Sedangkan penggunaan *battery* CoAP sebesar 3%. Presentase rata-rata battery usage dari variasi topik/ URI dengan variasi waktu yang berbeda pada pengujian sistem yaitu protokol MQTT lebih besar daripada protokol CoAP. Rata-rata penggunaan battery protokol CoAP lebih kecil, karena CoAP memi- liki overhead yang lebih rendah dan ukuran paket yang lebih kecil daripada MQTT, serta meng- gunakan UDP sebagai transport layer sehingga hanya membutuhkan percakapan satu arah.

4.4. Evaluasi Literatur Protokol MQTT dan CoAP

Pada sub bab kali ini penulis akan melakukan komparasi protokol bedasarkan literatur protokol MQTT dan CoAP. Beberapa parameter yang akan dikomparasikan yaitu pola komunikasi, jenis pro- tokol dan keamanan protokol, serta kehandalan protokol.

Pola Komunikasi MQTT dan CoAP

Protokol MQTT dan CoAP memiliki paradigma komunikasi yang berbeda. Protokol MQTT menggunakan arsitektur komunikasi *publish-subscribe*, dimana *publisher* mengirim pesan den- gan topik tertentu dan *subscriber* dapat mengenerate topik tersebut sesuai dengan topik yang ingin diterima sehingga sangat memungkinkan kamunikasi yang fleksibel antara *publisher* dan*subscriber*. Tetapi MQTT memerlukan broker yang berperan sebagai perantara antara *pub- lisher* dan *subscriber* (percakapan dua arah).

Sedangkan protokol CoAP menggunakan pola komunikasi RESTful (*Representational State Transfer*) dengan fitur *request-respons*. *Client* akan mengirim *request* ke *server* untuk mengak- ses/ mengirim pesan dan server melakukan respon yang berisi pesan yang diminta (percakapansatu arah).

Jenis Protokol Aplikasi MQTT dan CoAP

MQTT menggunakan TCP (*Transmission Control Protocol*) sebagai protokol *transport* yang memastikan pengiriman pesan dapat diandalkan. Koneksi TCP memastikan pesan yang akan dikirimkan berurutan dan tidak ada pesan yang hilang. Namun penggunaan TCP dalam MQTT memiliki *overhead* yang tinggi. Sedangkan CoAP menggunakan UDP (*User Datagram proto- col*) sebagai protokol *transport* yang sangat ringan sehingga dapat mengurangi *overhead* ko- munikasi. Tetapi UDP tidak sehandal TCP dikarenakan UDP tidak terjamin bahwa pesan akan sampai ke penerima. Oleh karena itu, MQTT diekspetasikan untuk lebih andal daripada CoAP. Tetapi MQTT memiliki nilai *overhead* yang lebih besar dibandingan CoAP.

Keamanan Protokol MQTT dan CoAP

Pada aspek keamaan, kedua protokol tersebut menyajikan keamaan yang sama. MQTT dikem- bangkan dengan *enkripsi* SSL (*Secure Sockets Layer*)/ TLS (*Transport Layer Security*) yang mendukung protokol TCP yang dirancang untuk keamanan yang lebih tinggi dalam kean- dalan

pengiriman data. Sedangkan CoAP dikembangkan enkripsi dengan metode DTLS (*Data- gram Transport Layer Security*) yang mendukung protokol UDP yang dirancang khusus untuklingkungan jaringan yang tidak andal.

Kehandalan Protokol MQTT dan CoAP

MQTT menghadirkan fitur QoS (*Quality of Service*) yaitu QoS 0, QoS 1, QoS 2. Sedan- gkan keandalan CoAP didasari dengan pesan NON/CON. MQTT level QoS 0 dan pesan NON tidak memberikan jaminan keandalan dalam berkomunikasi. MQTT level QoS 1 dan pesan CON memberikan jaminan keandalan bedasarkan ACK dalam berkomunikasi. Sedangkan pada MQTT QoS 2 menjamin pesan tidak akan terduplikat pada sisi penerima. Hal tersebut tidak dimiliki oleh CoAP yang memberikan fleksibelitas kepada para user dan membuat MQTT lebih cocok digunakan pada aplikasi yang tidak mentoleransi pesan duplikat.

4.5. Analisis Keseluruhan Literatur Protokol MQTT dan CoAP

Pada bagian ini merupakan kesimpulan dari studi literatur yang telah dilakukan.

Table. 1. Parameter protokol MQTT dan CoAP

Parameter	MQTT	CoAP
Pola Komunikasi	Publish – Subscribe	RESTful
Kualitas Pengiriman Pesan	Terstruktur	Berantakan/Tidak terlacak
Internet Protokol	TCP	UDP
Keamanan	SSL/TLS	DTLS
Pengembangan Protokol	Stabil	Masih Berkembang

5. Kesimpulan

Berdasarkan hasil pengujian IoT menggunakan metode MQTT dan CoAP, dapat disimpulkan bahwa keseluruhan alat dapat bekerja dengan cara dihubungkan pada kesatuan jaringan WiFi yang sama antara sensor (*hardware*), *server*, dan aplikasi. Sehingga dapat melakukan *monitoring* dan *controlling* terhadap hidroponik stroberi secara efektif. MQTT lebih cocok digunakan dalam sistem *many-to-many*, dimana pesan disampaikan an- tara beberapa *client* melalui *broker*. MQTT menggunakan prinsip percapakan dua arah karena memisahkan *publisher* dan *subscriber* dengan membiarkan *client* menerbitkan dan meminta *broker* untuk memutuskan dan penyalinan pesan. Sedangkan, CoAP cocok dipakai dalam sistem *one-to-one*, dimana pesan disampaikan antara *client* dan *server*. CoAP menggunakan prinsip percakapan satu arah karena menggabungkan *server* dan PUT serta GET terpisah. Protokol CoAP tidak memberikan dukungan dalam hal pemberian label pesan dengan jenis atau metadata lain untuk membantu *user* memahaminnya. Sehingga *user* tidak mengetauhi pesan mana yang berhasil terkirim, karena berupa kode *enskripsi*. Protokol MQTT lebih unggul dibandingkan protokol CoAP di dalam aspek *packet loss*, peng- gunaan CPU, dan konsistensi pesan yang terkirim. Sedangkan protokol CoAP lebih unggul dalam aspek *delay* dan penggunaan *battery*.

Keunggulan protokol MQTT akan lebih maksimal apabila kondisi *bandwidth* pada sistem yang akan diimplementasikan lebih besar. Dengan *bandwidth* yang besar, protokol MQTT dapat mengirimkan pesan dengan jumlah yang banyak dan waktu yang cepat. Tetapi jika memi- liki keterbatasan *bandwidth* maka protokol CoAP lebih cocok diimplementasikan pada sistemtersebut. Karena protokol CoAP dapat mengirimkan *bandwidth* yang kecil, protokol CoAP dapat mengirimkan pesannya lebih cepat. Selain itu, *prototype* mampu melakukan pengontrolan terhadap pemberian air, nutrisi secara otomatis pada tanaman hidroponik stoberi dengan menggunakan protokol MQTT dan dapat dilakukan pemonitoran menggunakan aplikasi secara berkala.

Kedepannya perlu adanya pengembangan lebih dalam untuk protokol CoAP pada metode PUT agar pesan yang diterima terdapat label pesan yang berhasil dikirimkan. Sehingga *user* dapat mengetauhidengan mudah pesan mana yang sudah terkirimkan.

References

[1] I. of Electrical, E. Engineers, and I. A. C. Singapore. (2014). IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP): Singapore.
[2] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso- Zarate. (2015). "A survey on application layer protocols for the internet of things".
[3] H. Anwari, E. S. Pramukantoro, and M. H. Hanafi. (2017). "Pengembangan iot middleware berbasis event-based dengan protokol komunikasi coap, mqtt dan websocket," pp. 1560–1567.
[4] N. Sharma, S. Acharya, K. Kumar, N. Singh, and O. Chaurasia. (2018). "Hydroponics as an advanced technique for vegetable production: An overview," Journal of Soil and Water Conservation, vol. 17, p. 364.
[5] D. Silva, L. I. Carvalho, J. Soares, and R. C. Sofia. (2021). "A Performance Analysis of Internet of Things Networking," Applied Sciences, vol. 11, no. 4879, pp. 1–30.
[6] V. Seoane, C. Garcia-Rubio, F. Almenares, and C. Campo. (2021). "Performance evaluation of CoAP and MQTT with security support for IoT environments," Computer Networks, vol. 197, no. April, p. 108338.