Implementasi MQTT Sebagai Protokol Komunikasi Pada Prototipe Sistem Monitoring Smart Building

1st Ferizki Putra Alfasa
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
ferizkiputraalfasa@student.telkomuniversity.ac.id

2nd Favian Dewanta
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
favian@telkomuniversity.ac.id

3rd Istikmal
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
istikmal@telkomuniversity.ac.id

***Abstrak* — *Smart building* merupakan salah satu teknologi yang sedang berkembang pesat pada masa ini. Smart building merupakan salah satu contoh implementasi dari IoT (Internet of Things), di mana konsepnya adalah mengintegrasikan teknologi informasi dan komunikasi dengan objek-objek fisik seperti bangunan, kendaraan dan masih banyak hal lain lagi, dan semua itu terhubung dengan akses internet. Salah satu masalah besar saat membangun smart building adalah keamanan serta penggunaan energi yang tidak terkontrol sehingga menimbulkan pemborosan energi yang sia-sia. Untuk itu pada penelitian ini dibuat prototipe sistem monitoring dan otomasi pada smart building dengan menggunakan MQTT (Message Queuing Telemetry Transport) protokol komunikasi yang di enkripsi sebagai keamanan data antara mikrokontroler yang terhubung dengan Raspyberry Pi sebagai server terhubung dengan sensor yang selalu mengirimkan data secara real time dan website monitoring akan menampilkan data suhu dan kelembaban. Pada sistem otomasi menggunakan Raspyberry Pi lainnya sebagai client yang akan terhubung dengan Raspyberry Pi sebagai server dengan dihubungkan melalui BACNet menggunakan ip address masing-masing perangkat lalu relay akan terhubung dengan client dan akan menyalakan lampu sebagai pengganti AC (Air Conditioning).**

***Kata kunci*— Smart Building, MQTT**

I. PENDAHULUAN

Internet of Thing (IoT) telah mengalami pekembangan yang sangat pesat pada masa ini. Secara garis besar IoT adalah sebuah gagasan di mana semua benda/perangkat bisa berkomunikasi satu sama lain dengan bantuan internet sebagai penghubungnya. Protokol komunikasi merupakan salah satu aspek yang penting dalam IoT. Terdapat banyak protokol komunikasi yang digunakan dalam dunia Iot, seperti MQTT (Message Queuing Telemetry Transport), CoAP (Constrained Application Protokol), AMQP(Advanced Message Queuing Protokol) dan HTTP (Hypertext Transfer Protokol)[1] [2]. MQTT menjadi salah satu protokol komunikasi yang paling banyak digunakan.

Penggunaan MQTT sebagai protokol komunikasi dalam Internet of Things (Iot) telah banyak digunakan sejak awal tahun 2010. Diciptakan pertama kali oleh Andy Stanford clark dan Srlen Nipper ditahun 1999 dengan tujuan sebagai protokol komunikasi yang ringan dan memiliki bandwidth yang kecil[3]. Besarnya bandwidth saat terjadinya komunikasi antar perangkat menjadi sebuah masalah utama dalam bidang IoT. MQTT menjadi populer digunakan dalam IoT sebagai protokol komunikasi dikarenakan alasan diatas. MQTT menggunakan sistem pub/sub untuk berkomunikasi satu sama lain. Pada MQTT juga menggunakan kriptografi AES, karena pada IoT kebanyakan menggunakan enkripsi dan dekripsi AES .

Karena dalam pembuatan sistem monitoring pada smart building proses pengiriman data terjadi setiap detik, maka dalam penelitian ini bertujuan untuk menilai bagaimana Quality of service (QoS) MQTT sebagai protokol komunikasi pada sistem monitoring data pada smart building.

II. KAJIAN TEORI

A. MQTT

MQTT merupakan salah satu protokol komunikasi yang paling populer digunakan saat ini untuk IoT. Sejak awal diciptakan, MQTT memang didesain sangat ringan dan low bandwidth [2]. MQTT bekerja dengan sistem publisher/subscriber. Beberapa hal yang harus di pahami mengenai MQTT antara lain:

1. Client

Dalam MQTT publisher dan subscriber di sebut sebagai client. Client bertugas atau berfungsi sebagai pengirim atau penerima data.

2. Broker

Broker dalam MQTT sangat penting. Broker bertujuan untuk menerima semua pesan/data yang dikirim publisher dan mengirim semua pesan/data kepada subscriber.

3. Topik

Sebuah topik dalam MQQ merupakan point di mana kedua client bisa terhubung. Topik antara client harus sama agar broker bisa mengirimkan data kepada client yang sesuai.

4. Connection

MQTT membutuhkan TCP/IP untuk saling terhubung. Untuk standar port yang digunakan adalah 1883[2].

Perbandingan terhadap protokol komunikasi mana yang lebih baik digunakan dalam IoT telah dilakukan oleh beberapa orang, salah satunya adalah Yokotani dan Sasaki yang membandingkan antar MQTT dengan HTTP. Yokotani dan Sasaki[4] membandingkan kinerja MQTT dalam

kategori protokol berdasarkan arsitektur ICN dan HTTP dalam kategori protokol legacy dan mereka mengemukakan bahwa MQTT memiliki kinerja yang lebih baik. Eksperimen mereka mengonfirmasi bahwa MQTT bekerja lebih baik daripada HTTP dan menyimpulkan protokol berdasarkan arsitektur ICN lebih cocok untuk sistem IoT.

Selain perbandingan dengan HTTP, ada juga perbandingan dengan protokol AMQP, Luzuriaga bersama timnya[5] melakukan eksperimen terhadap protokol AMQP dan MQTT pada jaringan yang tidak stabil dan mobile untuk mengetahui niai dari packet loss, latency, jitter, dan nilai saturation boundarynya. Temuan dari penelitian ini adalah bahwa terjadi lonjakan pesan, pengiriman mengikuti urutan LIFO (last-input first-output) dalam kasus AMQP, tetapi dalam pengiriman paket MQTT tetap menjaga urutannya. AMQP lebih berorientasi pada keamanan daripada MQTT tetapi MQTT lebih hemat energi dibandingkan AMQP.

Berdasarkan perbandingan tersebut bisa diketahui bahwa MQTT masih memerlukan tambahan keamanan, dengan melakukan proses enkripsi pada program yang akan dibuat.

B. Kriptografi

Dalam kriptografi terdapat dua proses yang disebut enkripsi dan dekripsi. Enkripsi adalah proses penyandian plaint text menjadi chipper text, sedangkan dekripsi merupakan proses kebalikan dari enkripsi. Selain enkripsi dan dekripsi ada beberapa hal yang perlu diketahui dalam kriptografi:

1. Plaint Text: merupakan teks asli yang nantinya akan dienkripsi
2. Chiper Text: merupakan teks yang telah dienkripsi dengan bantuan key
3. Key: merupakan kata ataupun nilai yang dipakai untuk mengenkripsi plaint text.

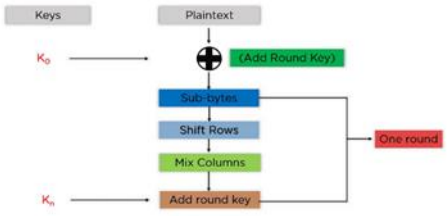
Proses enkripsi dan dekripsi sangat penting untuk menjaga kerahasiaan, integritas, dan ketersediaan data[6].

1. AES (Advenced Encryption Standard)

Algoritma AES mendukung beberapa variasi ukuran kunci yang digunakan, AES-128, AES-192, dan AES-256[7] . Perbedaan dari ketiga variasi di atas tentu saja merupakan panjang kunci yang digunakan, bisa dilihat pada Tabel 1. Semakin panjang kuncinya, makan semakin sulit juga datanya akan diserang. Algoritma AES seringkali diserang menggunakan metode “brute force attack”.

Tabel 1.
Perbandingan penggunaan bit blok pada AES

Bit Blok	Nb(Number of bit)	Nk(Number of key)		Nr(Number of rounds)
		Row	Column	
128	4	16	4	10
192	4	24	6	12
256	4	32	8	14



GAMBAR 1.
Alur kinerja AES [15]

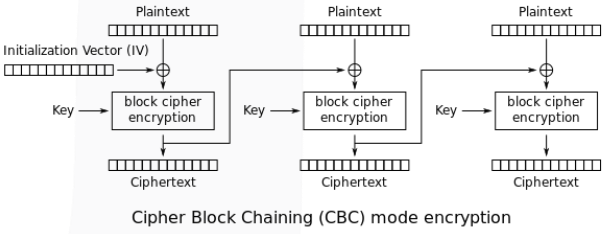
Penjelasan dari Gambar 1 adalah proses enkripsi pada AES menerima teks sehingga diproses pada algoritma menyesuaikan dengan bit yang digunakan, kemudian hasil tiap enkripsi pada plaintext melakukan XOR antar state sehingga di menghasilkan pesan baru tak bermakna. Proses enkripsi algoritma AES terdiri dari empat jenis transformasi byte: SubBytes, ShiftRows, MixColumns, dan AddRoundKey. Pada awal proses enkripsi, input yang telah disalin ke dalam keadaan akan mengalami transformasi byte AddRoundKey. Setelah itu, keadaan akan berulang kali mengalami transformasi SubBytes, ShiftRows, MixColumns, dan AddRoundKey sebanyak Nr. Proses ini dalam algoritma AES disebut sebagai fungsi putaran [14].

2. CBC (Cipher Block Chaining))

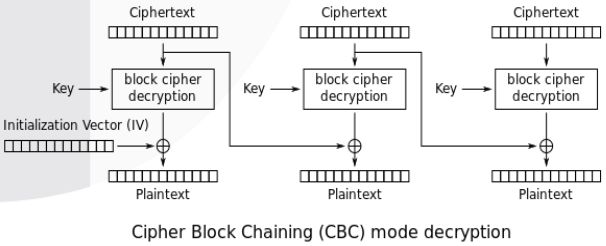
Cipher Block Chaining (CBC)[8] merupakan salah satu operasi mode chipper block yang populer di mana setiap teks asli (plaintext) di-XOR dengan blok sandi sebelumnya sebelum dienkripsi. Oleh karena itu, setiap sandi blok tergantung pada semua blok teks asli hingga tahap tersebut. Pesan teks asli M dibagi menjadi t blok bit-n M i dan sandi blok C i diberikan sebagai:

$$C_i = E_k(M \oplus C_{i-1}),\ i= 1, 2, \ldots t. \quad (1)$$

Kriptografi AES mode CBC menjadi algoritma yang melibatkan nilai Inisialisasi Vektor (IV) pada blok Cipher. IV berdasarkan ukuran pada setiap blok masukan plaintext-nya. Pada rangkaian bit pada plaintext akan dibagi menjadi blok yang sama hingga memiliki ukuran yang serupa [9].



GAMBAR 2.
Enkripsi mode CBC [16]



GAMBAR 3.
Dekripsi mode CBC [16]

3. OFB (Output Feedback)

OFB merupakanan salah satu mode operasi yang digunakan kriptografi blok untuk mengenkrispi data. OFB memungkinkan enkripsi blok demi blok, sehingga lebih cocok untuk mengenkripsi data dengan ukuran yang lebih besar. Pada dasarnya, OFB mengubah enkripsi blok menjadi aliran kunci(keystream) yang kemudian digunakan untuk melakukan XOR dengan plaintext untuk menghasilkan

chipertext[10] . Enkripsi dan dekripsi OFB dapat dirumuskan sebagai berikut :

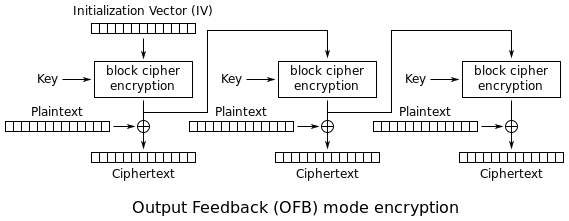
$$C_i = V_i \oplus B_i \quad (2)$$

$$B_i = V_i \oplus C_i \quad (3)$$

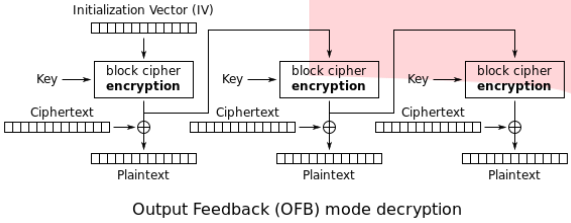
V_i di atas melabangkang initialization vector, untuk mencari initialization vector dapat menggunakan rumus berikut :

$$V_i = E_K(V_{i-1}) \quad (4)$$

Gambaran bagaimana proses kerja enkripsi dan dekripsi mode OFB dapat dilhat pada Gambar 4 dan Gambar 5.



GAMBAR 4.
Enkripsi OFB [17]



GAMBAR 5.
Dekripsi OFB [18]

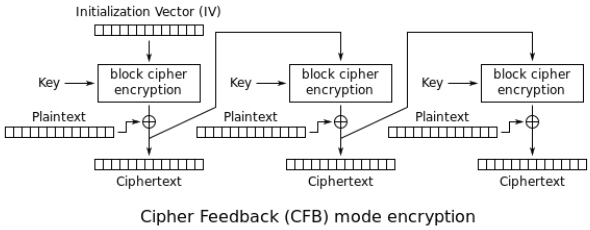
4. CFB (Cipher Feedback)

CFB adalah salah satu mode enkripsi yang digunakan dalam algoritma enkripsi blok seperti AES (Advanced Encryption Standard). Mode ini mengubah enkripsi blok menjadi aliran bit, yang memungkinkan enkripsi data dalam bentuk yang lebih kecil daripada ukuran blok cipher [11]. Enkripsi dan dekripsi mode CFB bisa dirumuskan dengan cara berikut :

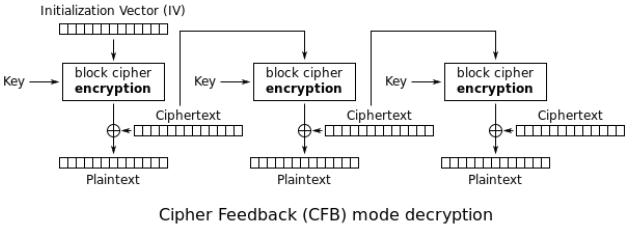
$$C_i = E_K(C_{i-1}) \oplus B_i \quad (5)$$

$$B_i = E_K(C_{i-1}) \oplus (C_i) \quad (6)$$

Kelebihan dari CFB adalah ukuran blok ciphertext dapat lebih kecil daripada ukuran blok cipher, sehingga mode ini cocok untuk enkripsi data yang lebih kecil daripada ukuran blok cipher. Namun, CFB tidak mendukung operasi paralel seperti CTR atau CBC, sehingga efisiensi enkripsi paralelnya terbatas. Gambaran bagaimana proses kerja enkripsi dan dekripsi mode CFB dapat dilhat pada Gambar 6 dan Gambar 7.



GAMBAR 6.
Enkripsi CFB [19]



GAMBAR 7.
Dekripsi CFB [20]

C. Qos

Quality of Services (QoS) merupakan mekanisme pada jaringan yang menentukan bahwa aplikasi-aplikasi atau layanan dapat beroperasi sesuai dengan standar kualitas layanan yang telah diterapkan[12] [13]. Berikut merupakan Parameter-parameter Quality of Services (QoS):

1. *Throughput*

Throughput merupakan kecepatan transfer data. Throughput adalah jumlah total kedatangan paket yang sukses diamati pada tujuan selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut[13]. Berikut merupakan persamaan untuk perhitungan throughput :

$$\text{Throughput} = \left(\frac{\text{Jumlah Byte}}{\text{Time Span}}\right) \times 8 \quad (7)$$

2. *Packet Loss*

Packet loss adalah banyaknya paket yang gagal mencapai tempat tujuan paket tersebut dikirim[13] . Persamaan untuk perhitungan packet loss :

$$Packet\ Loss = \left(\frac{\text{Paket dikirim} - \text{Paket diterima}}{paket\ dikirim}\right) \times 100 \quad (8)$$

3. *Delay*

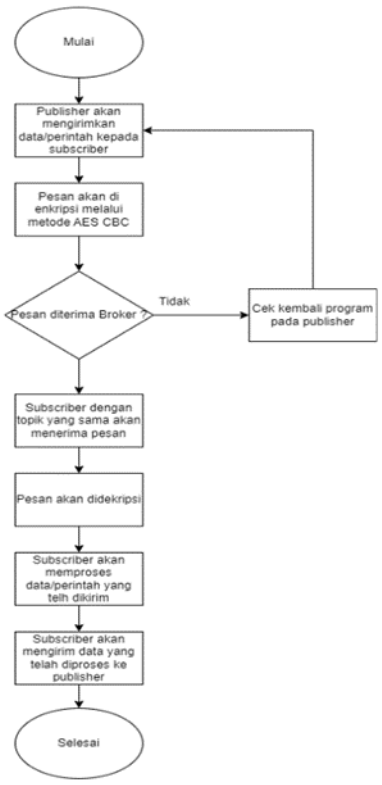
Delay adalah waktu yang dibutuhkan sebuah data untuk menempuh jarak dari asal ke tujuan[13]. Delay dapat dipengaruhi oleh jarak media fisik, kongesti atau waktu lama proses yang lama. Berikut merupakan persamaan untuk delay:

$$Delay = \left(\frac{\text{Total Delay}}{Total\ paket\ diterima}\right) \quad (9)$$

III. METODE

A. Rancangan Penelitian

Berikut merupakan diagram alir dari rancangan sistem MQTT yang dikembangkan pada penelitian ini:



GAMBAR 8.
Diagram alir sistem MQTT

Gambar 8 menunjukkan bahwa sistem MQTT yang dirancang telah dilengkapi dengan mode enkripsi dan dekripsi AES CBC. Proses enkripsi dan dekripsi bertujuan untuk menjadikan pengiriman data dari Raspberry Pi versi 4 ke Website monitoring menjadi lebih aman. Peran MQTT yang dirancang di sini adalah sebagai protokol komunikasi antara Raspberry Pi versi 4 dengan website yang telah dibuat.

B. Prosedur Penelitian

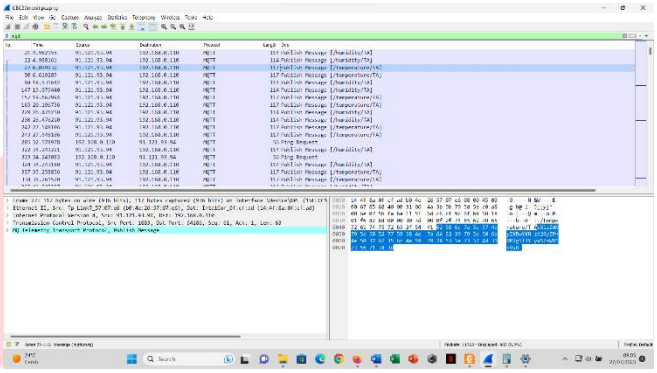
Penelitian mengenai MQTT ini dilakukan melalui beberapa langkah prosedur berikut:

1. Tahap pertama adalah membuat program MQTT menggunakan bahasa pemrograman Python dengan parameter-parameter spesifik, termasuk topik khusus untuk memungkinkan komunikasi antara publisher dan subscriber. Selain itu, pemilihan broker juga menjadi penting, dan dalam penelitian ini, broker yang digunakan adalah test.mosquitto.org. Program juga dilengkapi dengan enkripsi dan dekripsi untuk keamanan data.
2. Setelah program selesai dibuat, tahap pengujian dimulai dengan mengirimkan data dari publisher ke subscriber.
3. Jika pesan berhasil terkirim, prosedur selanjutnya adalah melakukan penangkapan paket (capture) menggunakan aplikasi Wireshark. Tujuan dari prosedur ini adalah untuk memastikan bahwa pengiriman data benar-benar menggunakan protokol komunikasi MQTT.
4. Setelah berhasil mengidentifikasi protokol MQTT pada Wireshark, tahap selanjutnya adalah menghitung nilai Quality of Service (QoS).
5. Data yang tertampil pada aplikasi Wireshark dicatat, dan dilakukan proses perhitungan manual menggunakan rumus yang sesuai dengan kajian teori.

Dengan prosedur ini, penelitian dapat mengevaluasi kinerja MQTT sebagai protokol komunikasi pada sistem monitoring data dalam smart building.

C. Cara Perolehan Data

Data parameter QoS (*throughput, packet loss,* dan *delay*) diperoleh dari proses *capture* data melalui aplikasi Wireshark sesuai dengan Gambar 9. Data yang terlihat pada Wireshark mencakup jumlah paket, waktu, rata-rata besar paket, rata-rata bytes/s, dan rata-rata bits/s. Dengan data tersebut, parameter QoS dapat dihitung secara manual sesuai dengan rumus yang sesuai dengan teori yang dikaji.



GAMBAR 9.
Capture MQTT

IV. HASIL DAN PEMBAHASAN

Setelah mendapatkan data dari menu "Capture File Data", langkah selanjutnya adalah melakukan perhitungan untuk menghitung Quality of Service (QoS) guna mendapatkan nilai throughput, packet loss, dan delay. Penelitian dilakukan selama 3 sesi, di mana sesi 1 dilakukan selama 1 menit, sesi 2 dilakukan selama 2 menit, dan sesi 3 dilakukan selama 3 menit, dan jumlah masing-masing pengambilan data dilakukan sebanyak 30 kali. Rata- rata setiap data yang dikirim sebesar 134 Byte dan dengan menggunakan internet dari Wifi dengan bandwith maksimal mencapai 40 Mbps. Dalam proses pengujian, standar yang digunakan pada pengujian ini adalah mengikuti standar ITU-T G1010.

TABEL 2.
Standar ITU-T G1010

Medium	Application	Degree of symmetry	Typical amount of data	Key performance parameters and target values		
				One-way delay (Note)	Delay variation	Information loss
Data	Web-browsing – HTML	Primarily one-way	~10 KB	Preferred < 2 s /page Acceptable < 4 s /page	N.A.	Zero
Data	Bulk data transfer/retrieval	Primarily one-way	10 KB-10 MB	Preferred < 15 s Acceptable < 60 s	N.A.	Zero

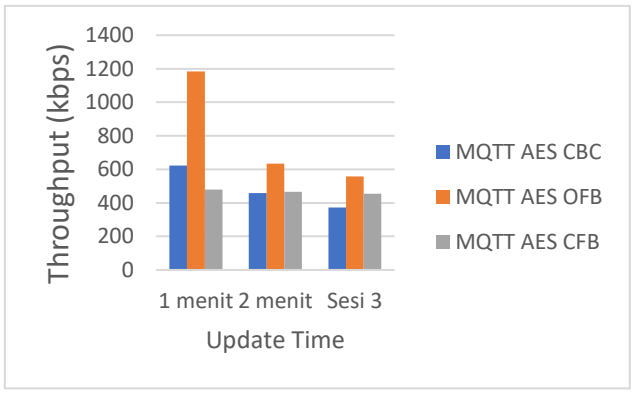
Tabel 2 merupakan standar ITU-T G.1010 sebagai acuan atau taget yang ingin dicapai pada penelitian ini. Berikut adalah hasil dari nilai-nilai parameter QoS setelah dilakukan perhitungan:

A. Throughput

Merupakan analisis jaringan untuk mengetahui jumlah data yang yang berhasil di transfer. Berikut adalah persamaan untuk mendapatkan nilai *throughput* :

$$\text{Throughput} = \left(\frac{\text{Jumlah Byte}}{\text{Time Span}}\right) \times 8$$

Dengan menggunakan persamaan tersebut nilai troughput dapat ditentukan. Berikut merupakan hasil troughput dari masing-masing sesi yang telah diuji :



GAMBAR 10.
Troughput protokol MQTT

Gambar 10 merupakan rata-rata hasil dari throughput yang didapatkan dari pengujian selama 3 sesi dengan jumlah pengambilan data sebanyak 30 kali.

B. Packet Loss

Merupakan paket data yang dikirim dari sumber data ke sumber penerima. *Packet loss* dapat menyebabkan penurunan kecepatan data, penundaan pengiriman data, dan masalah pada *website.* Untuk nilai *packetloss* pada MQTT adalah 0 %. Untuk menghitung packetloss menggunakan persamaan berikut :

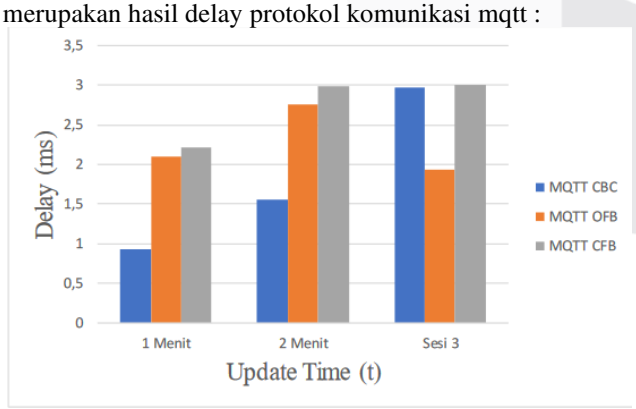
$$Packet\ Loss = \left(\frac{\text{Paket dikirim} - \text{Paket diterima}}{paket\ dikirim}\right) \times 100$$

C. Delay

Merupakan waktu yang diperlukan untuk sebuah paket data dari pengirim ke penerima tanpa hambatan, pada website monitoring pengujian dilakukan dengan cara mendapatkan nilai dari aplikasi software wireshark dan dihitung menggunakan persamaan sebagai berikut :

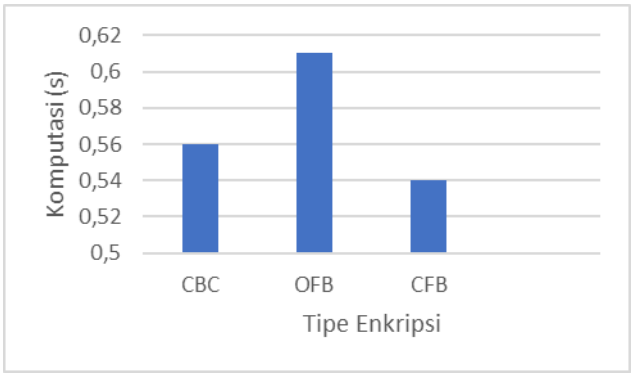
$$Delay = \left(\frac{\text{Total Delay}}{Total\ paket\ diterima}\right)$$

Pengujian delay dilakukan dua kali, pertama untuk melihat seberapa besar delay untuk protokol komunikasinya lalu yang kedua bagaimana delay komputasi nya. Berikut merupakan hasil delay protokol komunikasi mqtt :



GAMBAR 11.
Delay protokol MQTT

Gambar 11 menunjukkan hasil delay proses komunikasi dengan menggunakan MQTT, delay mengalami kelonjakan pada menit kedua dan ketiga karena jaringan internet yang digunakan tidak stabil. Untuk delay masing-masing proses enkripsi dekripsi adalah sebagai berikut :



GAMBAR 12.
Delay komputasi

Gambar 12 menunjukkan bahwa proses enkripsi dan dekripsi tercepat dicapai oleh mode CFB lalu CBC dan yang terakhir OFB. Proses perhitungan delay komputasi menggunakan fungsi time span pada program yang telah dibuat. Kemudian diambil sebanyak 30 data lalu dirata-ratakan.

V. KESIMPULAN

Protokol MQTT merupakan protokol yang umum digunakan dalam dunia IoT sebagai protokol komunikasi antara mesin ke mesin (M2M) di dunia IoT. Meskipun umum digunakan, protokol MQTT tidak dilindungi dengan mekanisme keamanan tertentu selain menggunakan TLS(Transport Layer Security).

Penelitian ini bertujuan untuk menganalisis kinerja protokol MQTT sebagai protokol komunikasi yang aman dengan mengimplementasikan proses enkripsi dan dekripsi AES tambahan, sehingga proses komunikasi secara end-to-end dapat berjalan dengan aman. Penelitian berfokus pada perbandingan kinerja MQTT jika disandingkan dengan AES CBC, OFB dan CFB. Beberapa parameter yang diteliti adalah troughput, packetloss, dan delay. Penelitian dibagi menjadi 3 sesi, di mana tiap sesi memiliki rentang waktu penelitian yang berbeda beda, dan rata-rata jumlah paket yang dikirimkan adalah sebesar 134 Byte. Dari hasil penelitian dapat dilihat bahwa nilai parameter Qos dari protokol MQTT berhasil mencapai target sesuai dengan standar ITU-T G1010 dan berdasarkan hasil penelitian dapat disimpulkan bahwa pengunaan protokol MQTT sebagai protokol komunikasi dengan tambahan enkripsi dekripsi CBC mendapatkan nilai yang memuaskan untuk setiap parameternya dan paling stabil dibandingkan dengan yang lainnya, meskipun nilai parameter QoSnya tidak selalu terbaik.

REFERENSI

[1] A. Amrullah, M. U. Al Rasyid, and I. Winarno, “Implementasi Dan Analisis Protokol komunikasi IOT untuk Crowdsensing Pada Bidang kesehatan,” *INOVTEK Polbeng - Seri Informatika*, vol. 7, no. 1, p. 122, 2022. doi:10.35314/isi.v7i1.2365

[2] B. Mishra and A. Kertesz,”The use of MQTT in M2M and IoT systems: A survey,” *IEEE Access*, vol. 8, pp. 201071–201086, 2020.

[3] E. Pereira, R. Pinto, J. Reis, and G. Gonçalves, “MQTT-RD: A mqtt based resource discovery for machine

to machine communication," *Proceedings of the 4th International Conference on Internet of Things, Big Data and Security*, 2019. doi:10.5220/0007716201150124
[4] T. Yokotani and Y. Sasaki, "Comparison with HTTP and MQTT on required network resources for IOT," *2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC)*,2016.doi:10.1109/iccerec.2016.7814989
[5] Luzuriaga, J. *et al.* (2015) 'A comparative evaluation of AMQP and MQTT protocols over unstable and mobile networks', *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*,2015. doi:10.1109/ccnc.2015.7158101.
[6] B. Kaushik, V. Malik, and V. Saroha, "A review paper on data encryption and decryption," *International Journal for Research in Applied Science and Engineering Technology*, vol. 11, no. 4, pp. 1986–1992, 2023. doi:10.22214/ijraset.2023.50101
[7] M. A. Sadikin and R. W. Wardhani, "Implementation of RSA 2048-bit and AES 256-bit with digital signature for Secure Electronic Health Record Application," *2016 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, 2016. doi:10.1109/isitia.2016.7828691
[8] F. Cohen, "Managing network security: The limits of cryptography," *Network Security*, vol. 1999, no. 11, pp. 7–11, 1999. doi:10.1016/s1353-4858(00)80003-3
[9] E. R. Febrianto and E. A. Sarwoko, "Kriptografi Citra Digital Menggunakan Algoritma Hill cipher dan Affine Cipher Berbasis Android," *Jurnal Masyarakat Informatika*, vol. 10, no.2,pp.11–21,2019.doi:10.14710/jmasif.10.2.31495
[10] M. J. Dworkin, "*Recommendation for block cipher modes of operation" :*, 2001. doi:10.6028/nist.sp.800-38a
[11] S. Mister and R. Zuccherato, "An attack on CFB mode encryption as used by openpgp," *Selected Areas in Cryptography*, pp. 82–94, 2006. doi:10.1007/11693383_6
[12] ETSI, "Telecommunications and Internet Protokol Harmonization Over Networks (TIPHON); General aspects of Quality of Service (QoS)," 1999.
[13] ITU-T G.1010, "Series G: Transmission Systems And Media, Digital Systems And Networks Quality of service and performance," 2001.
[14] B. Y. Yustiarini, F. Dewanta, and H. H. Nuha, "A comparative method for securing internet of things (IOT) devices: AES vs Simon-Speck encryptions," *2022 1st International Conference on Information System & Information Technology (ICISIT)*, 2022. doi:10.1109/icisit54091.2022.9872666
[15] B. K. Jena, "What is AES encryption and how does it work? - simplilearn," Simplilearn.com, http://www.simplilearn.com/tutorials/cryptography-tutorial/aes-encryption (accessed Aug. 21, 2023).
[16] K. Informatika, "Perhitungan Manual Algoritma Cipher Block Chaining Untuk Kriptografi Teks," Kita Informatika,http://www.kitainformatika.com/2019/05/perhitungan-manual-algoritma-chiper.html (accessed Aug. 21, 2023).
[17] A. Z. Mustafeez,"What is OFB?," Educative, http://www.educative.io/answers/what-is-ofb (accessed Aug. 21, 2023).
[18] L. Leli,"Cryptography Stack Exchange, https://crypto.stackexchange.com/questions/99304/ofb-mode-discussion (accessed Aug. 21, 2023).
[19] A. Z. Mustafeez,"What is Cfb?," Educative, http://www.educative.io/answers/what-is-cfb (accessed Aug. 21, 2023).
[20] D. Zapp,"Is it possible to decrypt the 2nd byte of AES-256-CFB-8 ciphertext without decrypting the 1st byte?," Cryptography Stack Exchange, https://crypto.stackexchange.com/questions/42400/is-it-possible-to-decrypt-the-2nd-byte-of-aes-256-cfb-8-ciphertext-without-decry (accessed Aug. 21, 2023).