Mitigasi Topic Hijacking pada Protokol MQTT Melalui Topic-Level Authentication dan Authorization

Erlan Effendy[1*], Veri Arinal[2]
[1] Teknik Informatika, St Ilmu Komputer Cipta Karya Informatika
[2]Sistem Informasi, St Tinggi Ilmu Komputer Cipta Karya Informatika
Email: [1*]erlaneffendy78@gmail.com, [2]veriarinal@stikomcki.ac.id

Abstrak

Protokol Message Queuing Telemetry Transport (MQTT) merupakan standar komunikasi yang banyak digunakan dalam sistem Internet of Things (IoT), namun konfigurasi bawaannya sering kali mengabaikan aspek keamanan dengan mengizinkan akses anonim sehingga rentan terhadap serangan topic hijacking yang dapat mengancam integritas dan kerahasiaan data. Penelitian ini bertujuan untuk merancang, mengimplementasikan, dan mengevaluasi mekanisme mitigasi topic hijacking melalui penerapan topic-level authentication dan authorization berbasis Access Control List (ACL) dengan tetap menjaga efisiensi kinerja sistem. Metode penelitian yang digunakan adalah Research and Development (R&D) dengan pendekatan eksperimental, yang diimplementasikan menggunakan lingkungan terkontainerisasi berbasis Docker, simulasi perangkat IoT menggunakan Python, serta analisis lalu lintas jaringan menggunakan Wireshark. Hasil penelitian menunjukkan bahwa mekanisme keamanan yang diusulkan mampu mencegah seluruh skenario serangan yang diuji, termasuk unauthorized publish/subscribe dan eskalasi hak akses, dengan tingkat keberhasilan 100%, serta hanya menimbulkan peningkatan latensi rata-rata sebesar 0,05 ms (10,2%) dan penurunan throughput sebesar 3,24%. Temuan ini menunjukkan bahwa penerapan topic-level authentication dan authorization efektif dalam meningkatkan keamanan protokol MQTT tanpa memberikan dampak signifikan terhadap performa sistem, sehingga layak diterapkan pada berbagai lingkungan IoT.

Kata kunci: *Access Control List (ACL),* Autentikasi*, IoT (Internet of Things)*, Protokol *MQTT, Topic Hijacking*

Abstract

The Message Queuing Telemetry Transport (MQTT) protocol is widely adopted as a communication standard in Internet of Things (IoT) systems; however, its default configuration often neglects security aspects by allowing anonymous access, making it vulnerable to topic hijacking attacks that threaten data integrity and confidentiality. This study aims to design, implement, and evaluate a mitigation mechanism for topic hijacking through the application of topic-level authentication and authorization based on Access Control Lists (ACLs while maintaining minimal performance overhead. The research employs a Research and Development (R&D) methodology with an experimental approach, utilizing a Docker-based containerized environment, Python-based IoT device simulations, and network traffic analysis using Wireshark. Security evaluation results demonstrate that the proposed mechanism successfully prevents all tested unauthorized access scenarios, including unauthorized publish/subscribe actions and privilege escalation attempts, achieving a 100% attack prevention rate. From a performance perspective, the implementation introduces only a marginal average latency increase of 0.05 ms (10.2%) and a throughput reduction of 3.24%, indicating that the proposed solution effectively enhances MQTT security without significantly degrading system performance and is therefore suitable for deployment in both industrial and small-to-medium IoT environments.

Keywords*: Access Control List (ACL), Authentication, IoT (Internet of Things), MQTT Protocol, Topic Hijacking*

PENDAHULUAN

Perkembangan Internet of Things (IoT) telah mentransformasi berbagai sektor, mulai dari sistem keamanan gedung hingga layanan kesehatan cerdas(Putri & Arinal, 2025; Suharto et al., 2021). Dalam ekosistem ini, protokol *Message Queuing Telemetry Transport* (MQTT) telah menjadi standar komunikasi *de facto* karena karakteristiknya yang ringan dan efisien dalam penggunaan *bandwidth* (E M Q Technologies Inc, 2024; Usmani, 2023). Keunggulan MQTT telah dibuktikan dalam berbagai implementasi, seperti sistem pendeteksi banjir yang menghasilkan latensi lebih rendah dibandingkan protokol HTTP (Atmoko et al., 2017). Namun, efisiensi yang menjadi prioritas utama MQTT sering kali mengabaikan aspek keamanan, terutama pada konfigurasi bawaan (*default*) yang memungkinkan akses anonim (Safitri & Seto Priambodo, 2023).

Kondisi tanpa proteksi ini menimbulkan kerentanan kritis berupa *topic hijacking*, di mana penyerang tidak berwenang dapat memublikasikan data palsu atau mencuri informasi sensitif dengan mengakses topik yang seharusnya dibatasi (Rupali Atul Mahajan, 2024). Dalam konteks industri, manipulasi data ini dapat menyebabkan keputusan operasional yang salah atau kerusakan peralatan fisik. Penelitian terdahulu telah mengusulkan berbagai solusi, seperti penggunaan *Transport Layer Security* (TLS) untuk enkripsi. Meskipun efektif mencegah penyadapan, penggunaan TLS secara signifikan meningkatkan ukuran paket dan latensi sistem (Dian Rachmadini et al., 2020). Pendekatan lain menggunakan *JSON Web Token* (JWT) berhasil menangkal serangan *Man-in-the-Middle*, namun belum mengatur kontrol akses di tingkat topik secara granular (Mangkurat, 2022). Demikian pula, konfigurasi dasar pada broker Mosquitto sering kali belum mengoptimalkan mekanisme otorisasi yang spesifik untuk memitigasi serangan pembajakan topik (Irwansyah et al., 2025).

Berdasarkan tinjauan tersebut, teridentifikasi celah penelitian yang signifikan: belum adanya kajian komprehensif yang membahas implementasi kontrol akses tingkat topik melalui *Access Control List* (ACL) yang disertai dengan validasi keamanan sistematis dan analisis dampak kinerjanya. Otorisasi melalui ACL merupakan mekanisme krusial yang memungkinkan administrator mendefinisikan izin *fine-grained* (siapa boleh melakukan apa terhadap topik mana) untuk menjembatani kesenjangan antara riset teoritis dan kebutuhan praktis industri(Sahmi et al., 2021).

Penelitian ini bertujuan untuk mengisi celah tersebut dengan merancang, mengimplementasikan, dan mengevaluasi mekanisme autentikasi serta otorisasi tingkat topik berbasis ACL pada broker Mosquitto. Target perubahan yang diharapkan adalah terciptanya sistem IoT yang mampu mencapai tingkat pencegahan serangan (*attack prevention rate*) di atas 90% dengan tetap menjaga beban latensi di bawah 50 ms dan penurunan *throughput* kurang dari 20%. Melalui pendekatan *Research and Development* (R&D) berbasis kontainerisasi Docker, penelitian ini diharapkan memberikan panduan praktis untuk pengembangan IoT yang aman tanpa mengorbankan performa sistem *real-time*.

TINJAUAN PUSTAKA

Protokol *Message Queuing Telemetry Transport* (MQTT) mengadopsi model komunikasi *publish/subscribe* yang memisahkan pengirim dan penerima melalui entitas perantara berupa broker (Rusnawati & Hariyati, 2022). Meskipun efisien untuk perangkat dengan sumber daya terbatas, konfigurasi standar MQTT sering kali mengizinkan akses anonim, yang menciptakan vektor serangan berupa *topic hijacking*. Fenomena ini mencakup *unauthorized publishing* yang mengancam integritas data melalui injeksi pesan palsu, serta *unauthorized subscribing* yang mengakibatkan kebocoran data sensitif. Keamanan pada level aplikasi menjadi krusial karena tanpa enkripsi atau otorisasi, seluruh kredensial dan *payload* terekspos dalam bentuk teks biasa (*plaintext*) di tingkat paket jaringan (Dewanta et al., 2022).

Mitigasi terhadap ancaman tingkat topik memerlukan mekanisme otorisasi yang lebih granular daripada sekadar autentikasi koneksi. *Access Control List* (ACL) merupakan instrumen yang memungkinkan penegakan prinsip *Least Privilege*, di mana setiap entitas hanya diberikan hak akses minimal yang diperlukan untuk fungsinya. Implementasi ACL pada broker seperti Mosquitto menerapkan kebijakan *default deny*, di mana seluruh upaya akses ditolak secara eksplisit kecuali telah didefinisikan dalam aturan otorisasi. Berbeda dengan metode enkripsi kompleks yang sering kali membebani performa, ACL menawarkan

proteksi pada level broker dengan *overhead* yang lebih rendah, menjadikannya solusi ideal untuk sistem *real-time*(Arifin et al., 2024).

Upaya peningkatan keamanan MQTT telah dieksplorasi melalui berbagai pendekatan. (Mangkurat, 2022) mengusulkan penggunaan *JSON Web Token* (JWT) dan TLS untuk mengatasi ketiadaan autentikasi pada ribuan server MQTT publik, namun solusi tersebut belum menjangkau kontrol akses granular di tingkat topik. (Rupali Atul Mahajan, 2024) mengembangkan algoritma simetris untuk enkripsi data, namun tetap menyisakan celah di mana pengguna yang terautentikasi dapat mengakses topik di luar wewenangnya karena ketiadaan mekanisme otorisasi.

Penelitian lain oleh (Irwansyah et al., 2025) mendokumentasikan konfigurasi dasar broker pada perangkat *single-board computer* dengan tingkat keberhasilan pengiriman pesan sebesar 78,6%, namun belum mengoptimalkan fitur ACL untuk mitigasi serangan aktif. Sementara itu, (Salim et al., 2024) memvalidasi efektivitas *Elliptic Curve Cryptography* (ECC) dalam menjaga kerahasiaan data dengan latensi rata-rata 54,1 ms. Meskipun berbagai riset telah membahas aspek enkripsi dan autentikasi, masih terdapat kesenjangan signifikan dalam literatur mengenai analisis perilaku dan dampak performa dari implementasi ACL yang komprehensif untuk memitigasi *topic hijacking*. Penelitian ini mengisi celah tersebut dengan menyediakan validasi empiris mengenai *silent drop* pada mekanisme ACL dan karakterisasi kinerjanya dalam lingkungan industri yang tersimulasi.

METODE

Penelitian ini menggunakan pendekatan Research and Development (R&D) dengan metodologi eksperimental untuk merancang dan mengevaluasi mekanisme otorisasi tingkat topik pada protokol MQTT. Pendekatan ini dipilih karena penelitian tidak hanya menganalisis kerentanan yang ada, tetapi juga mengembangkan solusi konkret dan mengevaluasinya secara empiris untuk menjembatani kesenjangan antara riset teoretis dan praktik industri.

Gambar 1. Diagram Alur Tahapan R&D

Tahapan Penelitian Proses penelitian dilaksanakan melalui enam tahapan utama yang sistematis:

1. Analisis Kerentanan: Melakukan pemodelan ancaman (*threat modeling*) untuk mengidentifikasi vektor serangan *topic hijacking* dan menetapkan *baseline* pada sistem tanpa pengamanan.
2. Perancangan Solusi: Mendesain lapisan keamanan ganda yang terdiri dari *Authentication Layer* (PBKDF2-SHA512) dan *Authorization Layer* menggunakan *Access Control List* (ACL) dengan prinsip *Least Privilege*.
3. Implementasi Teknis: Membangun lingkungan eksperimental menggunakan kontainerisasi Docker untuk memastikan hasil yang dapat direproduksi (*reproducible*).
4. Pengujian dan Validasi: Melakukan uji penetrasi keamanan, pengukuran metrik performa, dan analisis paket jaringan.
5. Analisis dan Evaluasi: Mengevaluasi metrik keberhasilan berdasarkan *attack prevention rate*, beban latensi, dan degradasi *throughput*.
6. Dokumentasi: Menyusun panduan implementasi praktis dan hasil benchmark sebagai referensi deployment produksi.

Lingkungan Eksperimental Solusi diimplementasikan pada broker Eclipse Mosquitto versi 2.0.18. Arsitektur sistem dirancang menggunakan *multi-container* yang terisolasi dalam jaringan virtual Docker untuk meminimalkan latensi eksternal.

Perangkat IoT disimulasikan menggunakan skrip Python dengan pustaka paho-mqtt yang bertindak sebagai *publisher* dan *subscriber*.

Rancangan Pengujian Validasi solusi dilakukan melalui tiga kategori pengujian utama untuk menjawab pertanyaan penelitian:

1. Pengujian Keamanan: Dilakukan *penetration testing* terhadap lima kasus uji (*test cases*), mencakup koneksi tanpa kredensial, publikasi/langganan tidak sah, akses lintas perangkat, dan eskalasi hak akses. Kriteria keberhasilan ditetapkan pada *attack prevention rate* > 90%.
2. Pengujian Kinerja: Mengukur *Round-Trip Time* (RTT) untuk 100 pesan dan *throughput* selama 30 detik publikasi berkelanjutan. Pengujian membandingkan tiga skenario:

Baseline (S1), *Auth-only* (S2), dan *Full Security* (S3).
3. Analisis Protokol: Menggunakan Wireshark untuk memverifikasi perilaku *silent drop* pada ACL dan menganalisis kode kembalian (*return code*) pada level paket MQTT.

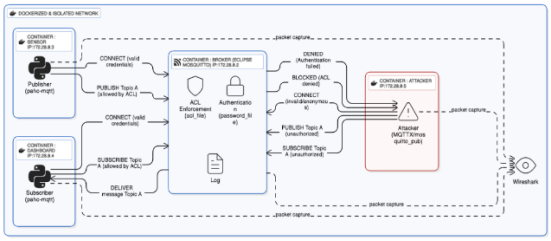
HASIL DAN PEMBAHASAN

1. Implementasi dan Lingkungan Eksperimental

Bagian ini mendokumentasikan spesifikasi teknis dan desain arsitektur yang digunakan untuk memvalidasi mekanisme *topic-level authentication* dan *authorization*. Penggunaan teknologi kontainerisasi menjadi inti dari pengembangan ini untuk memastikan isolasi lingkungan dan konsistensi hasil pengujian.

a. Arsitektur Kontainerisasi dan Lingkungan Teknis

Penelitian ini membangun ekosistem IoT tersimulasi menggunakan **Docker Engine 28.5.1** dan **Docker Compose 2.40.2** sebagai alat orkestrasi. Arsitektur sistem dirancang dalam jaringan virtual *bridge* terisolasi (subnet 172.28.0.0/16) untuk meminimalkan interferensi latensi eksternal dari jaringan *host*. Lingkungan ini terdiri dari empat komponen utama: sebuah broker Eclipse Mosquitto (v2.0.18), satu kontainer sensor IoT, satu kontainer dasbor monitoring, dan satu kontainer penyerang (*attacker*) yang dikembangkan menggunakan Python 3.11 dengan pustaka *paho-mqtt*.



Gambar 2. Diagram Arsitektur Lingkungan Eksperimental Docker

Spesifikasi perangkat keras yang digunakan mencakup prosesor Intel i5-1235U dengan RAM 16GB, yang memberikan kapasitas komputasi yang memadai untuk menjalankan simulasi beban kerja frekuensi tinggi tanpa mengalami *bottleneck* pada level *hardware*.

b. Penegakan Lapisan Autentikasi dan Otorisasi

Implementasi keamanan dilakukan melalui modifikasi pada file konfigurasi *mosquitto.conf*. Berbeda dengan konfigurasi *default* yang sering kali mengabaikan aspek keamanan, sistem ini secara eksplisit menonaktifkan akses anonim melalui perintah allow_anonymous false, memaksa setiap entitas untuk melakukan verifikasi kredensial sebelum diberikan akses ke broker.
Lapisan autentikasi menggunakan mekanisme *hashing* PBKDF2-SHA512 dengan iterasi tinggi untuk melindungi file *password* dari serangan *brute-force*. Setelah fase autentikasi berhasil, broker menerapkan lapisan otorisasi melalui *Access Control List* (ACL). Mekanisme ACL ini mengikuti prinsip *default deny*, di mana semua upaya akses ditolak secara otomatis kecuali jika diizinkan secara eksplisit dalam aturan otorisasi. Strategi ini sangat krusial untuk memitigasi *topic hijacking* karena membatasi ruang gerak setiap *user* hanya pada topik yang relevan dengan fungsinya.

Tabel 1. Matriks Izin Akses per User

User	Topic Pattern	Permission
admin	#	readwrite
sensor_device_01	sensor/temperature/room1	write
sensor_device_01	sensor/humidity/ room1	write
sensor_device_01	config/sensor/ room1	read
monitoring_app	sensor/#	read
monitoring_app	status/#	read
attacker	-	-

Matriks izin tersebut menunjukkan penerapan prinsip *Least Privilege*. Sebagai contoh, *monitoring_app* hanya diberikan izin *read* pada hirarki topik sensor/# dan tidak memiliki izin *write*, sehingga mencegah potensi manipulasi data meskipun akun tersebut terkompromi. Hal ini menjawab kesenjangan dalam penelitian Mangkurat (2022) yang sebelumnya hanya fokus pada autentikasi koneksi tanpa pembatasan granular pada level topik.

2. Evaluasi Efektivitas Mitigasi *Topic Hijacking*

Evaluasi keamanan dilakukan untuk mengukur kemampuan sistem dalam memblokir berbagai vektor serangan melalui pendekatan komparatif terhadap tiga skenario konfigurasi: *Baseline* (S1), *Auth-Only* (S2), dan *Full Security* (S3). Setiap skenario diuji menggunakan lima kasus uji (*test cases*) yang mencakup spektrum ancaman *topic hijacking* dari tingkat koneksi hingga pergerakan lateral.

a. Analisis Komparatif Pencegahan Serangan

Hasil pengujian menunjukkan perbedaan signifikan dalam tingkat perlindungan yang diberikan oleh masing-masing skenario. Skenario S1 (*baseline*) mencatatkan tingkat pencegahan 0%, di mana sistem sepenuhnya terekspos terhadap manipulasi data dan penyadapan. Penambahan lapisan autentikasi pada skenario S2 hanya meningkatkan proteksi menjadi 20% karena hanya mampu memitigasi koneksi tanpa kredensial (TC-1), namun gagal mencegah pengguna yang terautentikasi untuk mengakses topik di luar wewenangnya.

Sebaliknya, skenario S3 (*Full Security*) yang mengintegrasikan ACL berhasil mencapai *attack prevention rate* sebesar 100%. Hal ini membuktikan bahwa otorisasi granular adalah komponen wajib dalam keamanan MQTT, karena autentikasi saja tidak cukup untuk mencegah pembajakan topik.

Tabel 2. Perbandingan Hasil Security Testing Antar Skenario

Test Case	S1: Baseline	S2: Auth-Only	S3: Full Security
TC-1: Koneksi tidak terotorisasi	Tidak Aman	Aman	Aman
TC-2: Unauthorized Publish	Tidak Aman	Tidak Aman	Aman
TC-3: Unauthorized Subscribe	Tidak Aman	Tidak Aman	Aman
TC-4: Privilege Escalation	Tidak Aman	Tidak Aman	Aman
TC-5: Cross-Device Access	Tidak Aman	Tidak Aman	Aman
Attack Prevention Rate	0%	20%	100%

b. Validasi Mekanisme Pertahanan Berlapis (S3)

Keberhasilan skenario S3 didasarkan pada penegakan aturan keamanan pada dua level yang berbeda:

Pencegahan pada Lapisan Autentikasi (TC-1): Pada kasus uji koneksi tanpa kredensial, broker secara eksplisit menolak paket *CONNECT* dengan mengirimkan *CONNACK Return Code 5*. Analisis paket menunjukkan bahwa koneksi diputus pada level aplikasi sebelum penyerang dapat melakukan operasi *publish* atau *subscribe*.

Pencegahan pada Lapisan Otorisasi (TC-2 & TC-4): Untuk serangan publikasi tidak sah (TC-2) dan eskalasi hak akses (TC-4), broker berhasil memisahkan fase autentikasi dan otorisasi. Meskipun penyerang terhubung dengan kredensial valid, setiap upaya *publish* ke topik sensitif seperti sensor/temperature/room1 ditolak oleh ACL. Hal ini memvalidasi efektivitas prinsip *Least Privilege* di mana akun pemantauan (*read-only*) tidak dapat digunakan untuk menyuntikkan data palsu.

Gambar 3. Log Broker Penolakan Publish

c. Isolasi Topik dan Mitigasi Pergerakan Lateral (TC-5)

Salah satu temuan krusial dalam pengujian TC-5 adalah kemampuan ACL dalam melakukan isolasi topik antar perangkat (*topic isolation*). Dalam skenario industri, sensor yang memiliki keamanan fisik rendah seringkali menjadi titik awal pergerakan lateral. Hasil pengujian membuktikan bahwa sensor yang terkompromi di Room 1 tidak dapat mengirimkan perintah ke sistem kontrol pintu (control/door/lock). Implementasi ACL secara efektif membatasi dampak serangan hanya pada area yang terotorisasi, sehingga melindungi infrastruktur kritis dari manipulasi melalui perangkat IoT yang tingkat keamanannya lebih rendah.

d. Karakteristik Penolakan Akses (*Silent Drop*)

Berbeda dengan penolakan koneksi yang bersifat eksplisit, mekanisme ACL pada operasi *publish* dan *subscribe* (TC-3) menerapkan perilaku ***silent drop***. Pada level

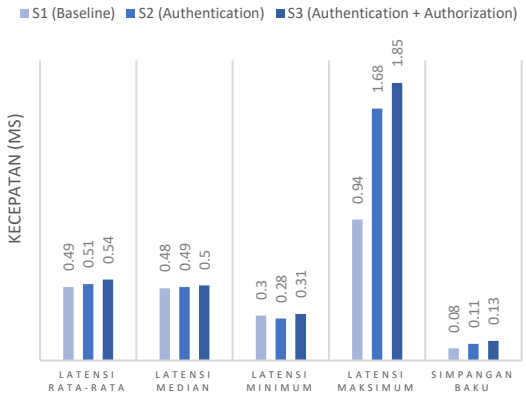
protokol, broker menerima paket dari penyerang namun tidak meneruskannya ke pelanggan lain tanpa memberikan notifikasi kesalahan kembali ke pengirim. Karakteristik ini secara strategis efektif untuk mempersulit upaya pengintaian (*reconnaissance*) karena penyerang tidak mendapatkan umpan balik langsung mengenai struktur otorisasi sistem.

3. Analisis Beban Performa dan *Trade-off* Keamanan

Evaluasi kinerja dilakukan untuk menguantifikasi beban komputasi yang ditimbulkan oleh lapisan autentikasi dan otorisasi terhadap efisiensi komunikasi protokol MQTT. Pengujian ini membandingkan metrik latensi dan *throughput* pada tiga skenario: *Baseline* (S1), Hanya Autentikasi (S2), dan Keamanan Penuh (S3).

a. Analisis Latensi (*Round-Trip Time*)

Berdasarkan hasil pengukuran terhadap 100 pesan, skenario *Baseline* (S1) menunjukkan latensi rata-rata sebesar 0,49 ms. Implementasi lapisan autentikasi pada skenario S2 menyebabkan peningkatan latensi sebesar 0,02 ms (4,1%), sementara penambahan ACL pada skenario S3 menghasilkan total latensi sebesar 0,54 ms. Secara akumulatif, beban latensi untuk keamanan penuh adalah sebesar 0,05 ms atau 10,2% dibandingkan kondisi tanpa pengamanan.



Gambar 4. Grafik Perbandingan Latensi

Peningkatan latensi pada skenario S3 disebabkan oleh dua proses komputasi utama: verifikasi kredensial menggunakan algoritma PBKDF2-SHA512 pada fase autentikasi dan mekanisme *pattern matching* pada aturan ACL untuk setiap paket *publish* yang dikirimkan. Meskipun terjadi peningkatan, nilai latensi sub-milidetik ini jauh di bawah ambang batas toleransi sistem *real-time* yang ditetapkan dalam kriteria keberhasilan (<50 ms).

b. Analisis Kapasitas *Throughput*

Pengukuran *throughput* dilakukan untuk mengevaluasi resiliensi sistem dalam menangani beban kerja frekuensi tinggi selama 30 detik. Hasil pengujian menunjukkan bahwa skenario S3 mampu memproses 549,80 pesan per detik, mengalami penurunan minimal sebesar 3,24% dibandingkan skenario *baseline* (568,20 msg/s).

Tabel 3. Analisis Penurunan Throughput

Perbandingan	Penurunan Throughput	Persentase Penurunan	Status vs Kriteria
S2 vs S1 (Dampak autentikasi)	-2.12 msg/s	-0.37%	< 20%
S3 vs S1 (Dampak penuh)	18.40 msg/s	3.24%	< 20%
S3 vs S2 (Dampak ACL)	20.52 msg/s	3.60%	-

Penurunan yang sangat rendah ini mengonfirmasi bahwa broker Mosquitto tetap efisien dalam melakukan pemeriksaan izin ACL tanpa menyebabkan kemacetan antrean pesan (*message queuing bottleneck*) yang signifikan. Tingkat keberhasilan pengiriman pesan pada skenario keamanan penuh tetap stabil pada angka 96,0%, yang menunjukkan keandalan sistem dalam kondisi beban operasional yang berkelanjutan.

c. Evaluasi *Trade-off* dan Kelayakan Implementasi

Analisis *trade-off* menunjukkan rasio manfaat-biaya yang sangat menguntungkan. Dengan "biaya" berupa beban latensi 10,2% dan penurunan *throughput* 3,24%, sistem memperoleh "manfaat" berupa pencegahan serangan *topic hijacking* sebesar 100%. Hal ini jauh lebih efisien dibandingkan solusi enkripsi TLS konvensional yang sering kali menambahkan beban *overhead* antara 15% hingga 25% (Rachmadini et al., 2020).

Tabel 4. Estimasi Kapasitas Deployment

Jumlah Perangkat	Frekuensi	Total msg/s	Kelayakan
10-50	1 msg/s	10-50 msg/s	Sangat layak

50-200	1 msg/s	50-200 msg/s	Layak / Perlu evaluasi
100-500	1 msg/s	100-500 msg/s	Layak / Marginal / Perlu optimasi
500+	1 msg/s	500+ msg/s	Layak / Perlu clustering / Perlu optimasi

Berdasarkan capaian tersebut, sistem ini dinilai sangat layak untuk berbagai aplikasi IoT. Untuk pemantauan lingkungan (suhu/kelembapan) yang memiliki toleransi latensi 1-5 detik, sistem ini memberikan margin keamanan yang sangat besar. Bahkan untuk aplikasi kontrol industri yang membutuhkan latensi di bawah 50 ms, latensi aktual sebesar 0,54 ms menjamin responsivitas sistem tetap optimal di lingkungan produksi.

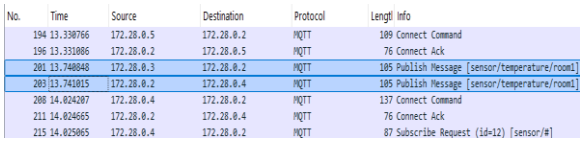
4. Analisis Karakteristik Protokol dan Mekanisme *Enforcement*

Analisis pada tingkat paket menggunakan Wireshark dilakukan untuk memvalidasi bagaimana mekanisme keamanan diimplementasikan dalam struktur protokol MQTT dan untuk memahami perbedaan perilaku antara lapisan autentikasi dan otorisasi.

a. Perbedaan Respons Lapisan Keamanan

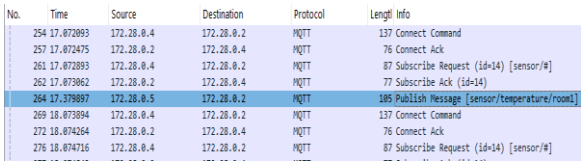
Hasil analisis menunjukkan adanya perbedaan fundamental dalam cara broker Mosquitto memberikan umpan balik terhadap pelanggaran keamanan. Pada **lapisan autentikasi (TC-1)**, broker memberikan respons eksplisit melalui paket *CONNACK* dengan *Return Code 5* (koneksi ditolak karena tidak terotorisasi). Hal ini sesuai dengan spesifikasi standar protokol MQTT untuk memberitahu klien bahwa kredensial yang diberikan tidak valid.

Sebaliknya, pada **lapisan otorisasi ACL (TC-2)**, broker menerapkan mekanisme **"silent drop"**. Analisis paket menunjukkan bahwa meskipun paket *PUBLISH* dari penyerang diterima secara fisik oleh broker, paket tersebut tidak diteruskan (*forwarding*) ke pelanggan (*subscriber*) yang sah.



Gambar 5. Hasil Tangkapan Paket Jaringan MQTT pada Publish yang Terotorisasi di Wireshark

Paket *PUBLISH* terkirim ke kedua arah, arah *sensor* (172.16.0.3) ke *broker* (172.16.0.2) dan juga *broker* (172.16.0.2) ke (172.16.0.4) *dashboard.*



Gambar 6. Hasil Tangkapan Paket Jaringan MQTT pada Publish yang Tidak Terotorisasi di Wireshark

Paket *PUBLISH* hanya pada arah *attacker* (172.16.0.5) ke *broker* (172.16.0.2) dan paket *broker* (172.16.0.2) ke (172.16.0.4) *dashboard* tidak terkirim.

b. Signifikansi Strategis *Silent Drop*

Penerapan *silent drop* memberikan dua keuntungan utama bagi keamanan sistem, yaitu mencegah pengintaian (*reconnaissance*) dengan tidak mengirimkan notifikasi kesalahan kembali ke pengirim sehingga penyerang tidak memperoleh umpan balik mengenai struktur otorisasi dan topik yang dilindungi oleh ACL, serta mengoptimalkan kinerja broker karena tidak perlu mengonsumsi sumber daya tambahan untuk memproses dan mengirim paket balasan kesalahan pada setiap pelanggaran akses di tingkat topik.

**c. Perilaku ACL pada Operasi *Subscribe* **

Analisis paket terhadap pengujian **TC-3** mengungkap karakteristik unik Mosquitto dalam menangani langganan tidak sah. Broker tetap mengirimkan paket *SUBACK* (berhasil berlangganan) meskipun pengguna tidak memiliki izin baca (*read*) pada ACL. Namun, *enforcement* keamanan tetap terjadi secara efektif pada fase **distribusi pesan (*message delivery phase*)**. Meskipun penyerang terdaftar sebagai *subscriber*, broker secara konsisten memblokir pengiriman data sensitif ke alamat IP penyerang. Pendekatan ini merupakan bentuk optimasi untuk menghindari *overhead* pemeriksaan ACL yang berulang pada setiap permintaan *subscribe* yang masuk.

Tabel 5. Ringkasan Temuan Analisis Protokol

Aspek	Temuan	Implikasi Keamanan
Kerentanan Baseline	Kredensial dan data terlihat plaintext	Sistem tanpa enkripsi sangat rentan terhadap penyadapan
Penolakan Autentikasi	CONNACK Return Code = 5	Autentikasi bekerja pada level aplikasi MQTT
Enforcement ACL (Publish)	Silent drop, tidak ada forwarding	ACL memblokir unauthorized publish secara efektif
Enforcement ACL (Subscribe)	SUBACK berhasil, delivery blocked	Pemeriksaan ACL pada message delivery, bukan subscribe request

KESIMPULAN

Penelitian ini menyimpulkan bahwa implementasi mekanisme keamanan berlapis yang mengintegrasikan autentikasi tingkat perangkat dan otorisasi tingkat topik melalui *Access Control List* (ACL) terbukti sangat efektif dalam memitigasi ancaman *topic hijacking* pada sistem IoT berbasis MQTT. Melalui konfigurasi keamanan penuh, sistem mampu memberikan perlindungan komprehensif terhadap berbagai vektor serangan, termasuk upaya publikasi data palsu, penyadapan informasi sensitif, hingga eskalasi hak akses yang sering mengeksploitasi kelemahan pada konfigurasi bawaan broker.

Temuan utama menunjukkan bahwa penggunaan ACL mampu memberikan kontrol akses yang granular dan terperinci, memastikan setiap entitas hanya berinteraksi dengan topik yang sesuai dengan fungsinya berdasarkan prinsip hak akses minimal (*least privilege*). Dari aspek kinerja, penambahan lapisan keamanan ini tidak menghambat operasional sistem secara signifikan, di mana beban latensi tambahan dan penurunan kapasitas pengiriman pesan tetap berada jauh di bawah ambang batas toleransi sistem waktu nyata, sehingga sangat layak untuk diimplementasikan pada lingkungan produksi industri maupun usaha kecil menengah. Analisis paket data juga mengungkap keunggulan strategis mekanisme *silent drop* pada ACL yang dapat mempersulit upaya pengintaian oleh pihak luar.

Selain itu, penelitian selanjutnya dapat mengkaji penggunaan mekanisme *caching* aturan ACL di memori untuk mempercepat proses verifikasi otorisasi serta pengintegrasian fitur *TLS session resumption* guna mengurangi beban komputasi pada koneksi yang terjadi secara berulang. Eksplorasi mengenai penggabungan mekanisme otorisasi granular ini dengan enkripsi pada lapisan transportasi (TLS/SSL) juga menjadi krusial untuk memberikan proteksi menyeluruh terhadap penyadapan di tingkat jaringan tanpa mengabaikan efisiensi sumber daya pada perangkat dengan keterbatasan komputasi. Terakhir, pengembangan metodologi pengujian pada perangkat keras fisik di lingkungan industri yang lebih kompleks akan memberikan validasi tambahan terhadap reliabilitas mekanisme *silent drop* dan efektivitas isolasi topik yang telah dirancang dalam penelitian ini.

DAFTAR PUSTAKA

Arifin, S., Nugraha, A. W., Mukti, F. S., & Jatmika, S. (2024). MQTT Broker Optimization: Comparative Analysis of Round Robin and Least Response Time. *Jurnal Nasional Teknik Elektro*, *3*, 127–136.

Atmoko, R. A., Riantini, R., & Hasin, M. K. (2017). IoT real time data acquisition using MQTT protocol. *Journal of Physics: Conference Series*, *853*(1). https://doi.org/10.1088/1742-6596/853/1/012003

Dewanta, F., Yustiarini, B. Y., & Indrakusumo Radityo Harsritanto, B. (2022). A study of secure communication scheme in MQTT: TLS vs AES cryptography. *Jurnal Infotel*, *14*(4), 269–276. https://doi.org/10.20895/infotel.v14i4.807

Dian Rachmadini, Ira Puspasari, & Jusak. (2020). Implementasi dan Analisis Fitur Keamanan Protokol MQTT pada Telehealthcare. *Journal of Technology and Informatics (JoTI)*, *2*(1), 10–19. https://doi.org/10.37802/joti.v2i2.117

E M Q Technologies Inc. (2024). Mastering MQTT: Your Ultimate Tutorial for MQTT. *Www.Emqx.Com*, 1–9. https://www.emqx.com/en/resources/your-ultimate-tutorial-for-mqtt

Irwansyah, I., Wahyuni Eka Sari, Agus Triyono,

Yhunike Widiya Pratama, Muhammad Bagus, Bintang Tmur, & Susanna, S. (2025). Rancang bangun Server Message Queuing Telemetry Transport (MQTT) Internet of Things (IoT) Berbasis Eclipse. *JSAI (Journal Scientific and Applied Informatics)*, *8*(1), 9–22. https://doi.org/10.36085/jsai.v8i1.7403

Mangkurat, C. A. (2022). *RANCANG BANGUN SISTIM KEAMANAN PERANGKAT IoT DENGAN METODE AUTENTIKASI MENGGUNAKAN JSON WEB TOKEN PADA PROTOKOL MQTT*.

Putri, S. E., & Arinal, V. (2025). *Pengembangan Sistem Monitoring Kualitas Air dan Kendali Pakan Otomatis untuk Budidaya Ikan Berbasis Internet of Things*. *6*(3), 1884–1892.

Rupali Atul Mahajan. (2024). Enhancing MQTT Security in the Internet of Things with an Enhanced Symmetric Algorithm. *Journal of Electrical Systems*, *20*(1s), 126–137. https://doi.org/10.52783/jes.758

Rusnawati, R. D., & Hariyati, T. S. (2022). Implementasi Internet of Things Pada Layanan Kesehatan. *Journal of Innovation Reseach and Knowledge*, *3471*(8), 569–574.

Safitri, N. A., & Seto Priambodo, A. (2023). MQTT and CoAP Communication Protocol Analysis in Internet of Things System for Strawberry Hydroponic Plants. *Journal of Robotics, Automation, and Electronics Engineering*, *1*(1), 45–56. https://journal.uny.ac.id/v3/jraee

Sahmi, I., Abdellaoui, A., Mazri, T., & Hmina, N. (2021). MQTT-PRESENT: Approach to secure internet of things applications using MQTT protocol. *International Journal of Electrical and Computer Engineering*, *11*(5), 4577–4586. https://doi.org/10.11591/ijece.v11i5.pp4577-4586

Salim, A. N., Sutabri, T., Negara, E. S., & Herdiansyah, M. I. (2024). Communication Security in the MQTT Protocol for Monitoring Internet of Things Devices Using Methods Elliptic Curve Cryptography. *Jurnal Teknik Informatika (JUTIF)*, *5*(2), 377–387.

Suharto, K., Fajar, M., & Hasniati. (2021). Penerapan Protokol Mqtt-Sn Pada Sistem Keamanan Rumah Berbasis Mikrokontroller Nodemcu. *Jurnal Ilmu Komputer*, *16*(No.1), 67–78. https://jurnal.kharisma.ac.id/kharismatech/article/view/152

Usmani, M. F. (2023). MQTT Protocol for the IoT - Review Paper. *ResearchGate*, *May 2021*. https://doi.org/10.13140/RG.2.2.26065.10088