BAB 1
USULAN GAGASAN

1.1 Deskripsi Umum Masalah

Pada zaman ini perkembangan di bidang Internet of Things (IoT) berkembang dengan sangat pesat, bahkan jumlah perangkat ini diperkirakan mencapai 75,44 miliar pada tahun 2025[1], oleh karena itu keamanan IoT menjadi aspek yang semakin penting akan tetapi hal ini sulit dicapai dikarenakan terbatasnya sumber daya yang mendukung keamanan perangkat IoT. Salah satu masalah yang ada dalam keamanan pembuatan IoT adalah kurangnya bagian *penetration testing* (pentest) dalam membangun dan mengamankan jaringan berbasis protokol MQTT (Message Queuing Telemetry Transport). MQTT adalah protokol komunikasi ringan yang sering digunakan dalam aplikasi Internet of Things (IoT) untuk pertukaran data antar perangkat yang memiliki sumber daya terbatas[2]. Namun, seperti protokol komunikasi lainnya, MQTT juga rentan terhadap berbagai ancaman keamanan. Serangan - serangan siber seperti man-in-the-middle (MITM), autentikasi *brute force*, dan kebocoran data adalah beberapa ancaman keamanan yang mengancam jaringan MQTT[3]. Oleh karena itu, untuk menjamin keamanan jaringan dan mencegah serangan, dilakukannya *Penetration Test* itu penting[4].

Ada lebih dari 377.000 *backend server* yang menggunakan MQTT, CoAP, dan XMPP yang masih banyak memiliki *vulnerabilities* [5]. Dengan banyaknya backend *server* yang masih memiliki *vulnerabilities* yang masih nampak, maka diperlukannya *Penetration Test* (pentest) untuk menguji coba perangkat IoT sebelum dapat diperjualbelikan. Dengan adanya masalah ini kami akan mencoba membuat sebuah prototipe yang dapat melakukan pentest untuk membantu *developer* dalam hal pengecekan keamanan perangkat IoT yang akan dibuat. Perangkat IoT pada umumnya beroperasi di *layer physical*, *layer network*, dan *layer application*. Setiap *layer* juga memiliki *kerentanannya* masing masing, contohnya pada *layer network* rentan terhadap DoS, sniffing, dan gangguan sinyal[6].

Selain itu, kompleksitas dalam menjaga keamanan IoT tidak hanya berasal dari banyaknya perangkat yang terhubung, tetapi juga dari keberagaman jenis perangkat dan jaringan yang digunakan. Perangkat IoT memiliki karakteristik yang sangat bervariasi, mulai dari perangkat dengan sumber daya yang sangat terbatas hingga yang memiliki kapabilitas tinggi. Hal ini membuat penerapan standar keamanan yang seragam menjadi

sulit dilakukan. Selain itu, sebagian besar perangkat IoT beroperasi tanpa intervensi manusia sehingga menambah tantangan dalam mengidentifikasi dan merespon ancaman keamanan secara cepat dan tepat. Serangan terhadap jaringan berbasis protokol MQTT dapat terjadi dalam waktu yang sangat singkat, sementara kemampuan perangkat dalam mendeteksi dan menangkal serangan tersebut seringkali terbatas[7]. Salah satu serangan yang terjadi adalah serangan malware "*WailingCrab*" yang terjadi di tahun 2023[8]. Kompleksitas ini memerlukan pendekatan yang komprehensif dalam pengujian keamanan, termasuk dalam penerapan *penetration testing* yang disesuaikan dengan karakteristik dan keterbatasan perangkat IoT itu sendiri.

Saat ini, solusi-solusi yang tersedia untuk mengamankan jaringan IoT berbasis MQTT masih terbatas dan banyak yang bersifat reaktif, yaitu hanya merespon setelah serangan terjadi. Beberapa metode keamanan seperti enkripsi data, autentikasi dua faktor, dan penerapan *firewall* memang dapat memberikan lapisan perlindungan tambahan, namun belum cukup efektif untuk menangkal serangan yang lebih canggih seperti serangan man-in-the-middle atau brute force[9]. Selain itu, karena keterbatasan sumber daya perangkat IoT, penerapan solusi-solusi keamanan yang ada seringkali mengurangi efisiensi perangkat dalam menjalankan fungsinya. Oleh karena itu, diperlukan solusi yang lebih proaktif, seperti pengembangan prototipe alat *penetration testing* yang tidak hanya dapat mengidentifikasi *vulnerabilities* lebih awal, tetapi juga mampu beroperasi secara efisien pada perangkat dengan sumber daya terbatas.

1.2 Analisa Masalah

Pengembangan *software pentesting* untuk IoT merupakan langkah penting dalam otomasi dalam pembuatan perangkat IoT dan meningkatkan keamanan teknologi IoT yang terus berkembang. Namun, ada tantangan kompleks dan kritis muncul yang perlu dianalisis secara mendalam. Beberapa masalah utama dalam pengembangan perangkat lunak ini mencakup berbagai aspek yang perlu dipahami dan diselesaikan. Dalam analisis ini, kami akan meninjau aspek - aspek yang relevan dengan disiplin ilmu masing-masing, serta memastikan keamanan dan kepatuhan teknologi tersebut. Berikut adalah aspek - aspek yang akan dikaji :

1.2.1. Aspek Teknis

- Berkaitan dengan pengembangan *software* yang mampu untuk melakukan *scan* dan memeriksa kerentanan sebuah perangkat IoT. Mulai dari memeriksa

port yang terbuka, mencoba masuk kedalam sistem, melakukan *sniffing* data yang dikirim ke *broker*.

- Penggunaan protokol MQTT sebagai sarana komunikasi perangkat IoT dikarenakan pemakaian dayanya yang rendah, namun tidak memiliki perlindungan lebih. bisa menggunakan *port* TLS/SSL namun akan memakan lebih banyak daya, sehingga tidak bisa diterapkan di setiap perangkat IoT [7].
- Beberapa perangkat IoT tidak berisi data yang cukup penting atau tidak memiliki efek yang signifikan sehingga tidak memerlukan pengamanan lebih lanjut.

1.2.2. Aspek Keamanan

- Apabila terjadi peretasan perangkat IoT, terdapat kemungkinan adanya data yang bocor tergantung dari perangkat IoT yang diretas. Seperti rekaman aktifitas rumah apabila perangkat IoT berupa kamera pengawas, hal ini dapat membahayakan keluarga yang menghuni rumah tersebut[10].

1.2.3. Aspek Hukum

- Pasal 46 ayat (1) UU ITE menyebutkan setiap orang dengan sengaja dan tanpa hak atau melawan hukum mengakses komputer dan/atau sistem elektronik milik orang lain dengan cara apapun diancam pidana penjara paling lama 6 tahun dan/atau denda paling banyak Rp600 juta.
- Penting untuk memastikan bahwa *software pentest* yang dikembangkan mematuhi hukum terkait privasi data, seperti GDPR (General Data Protection Regulation) atau UU Perlindungan Data lainnya, terutama jika melibatkan data pengguna saat pengujian.[11]

1.2.4. Aspek Sosial

- Dampak sosial dari serangan siber terhadap perangkat IoT mencakup berbagai risiko dan tantangan yang dapat mempengaruhi masyarakat. Salah satu dampak utamanya adalah ancaman terhadap keamanan individu dan privasi. Sebagai contoh, data yang dihasilkan oleh perangkat IoT seperti *smart meters* dapat digunakan untuk membuat profil pengguna secara rinci, termasuk aktivitas harian dan kebiasaan mereka, tanpa sepengetahuan mereka. Hal ini

dapat membahayakan privasi pengguna, mengingat data tersebut dikumpulkan dan diproses secara otomatis dan kontinu.

- Selain itu, serangan terhadap sistem siber yang mengendalikan perangkat IoT dapat mengakibatkan kerusakan fisik yang nyata, bahkan mengancam keselamatan manusia. Serangan siber ini juga dapat memperburuk ketimpangan sosial, misalnya dengan menciptakan jurang digital antara mereka yang memiliki akses ke teknologi canggih dan mereka yang tidak[12].

1.2.5. Aspek Ekonomi

- Dari perspektif ekonomi, serangan siber pada perangkat Internet of Things (IoT) dapat mengakibatkan kerugian finansial yang signifikan bagi perusahaan. Biaya pemulihan dari serangan siber, seperti investigasi, pemulihan data, pemberian kompensasi, hingga denda hukum, membutuhkan biaya yang sangat besar. Menurut *IBM Report: Escalating Data Breach Disruption Pushes Costs to New*, rata-rata biaya global dari satu pelanggaran data telah mencapai sekitar $4,88 juta, atau meningkat 10% dari tahun sebelumnya. Faktor-faktor seperti ketidakpatuhan terhadap peraturan, lingkungan IoT yang rentan, dan serangan rantai pasok juga berkontribusi meningkatkan biaya pelanggaran ini secara signifikan[13].
- Selain itu, perusahaan yang mengalami serangan siber berpotensi menghadapi kerugian reputasi yang mengarah pada hilangnya pelanggan, serta biaya tambahan dari kompensasi pelanggan. *IBM's Cost of a Data Breach Report,* sekitar 63% perusahaan bahkan harus menaikkan harga produk atau jasa mereka untuk menutup biaya pelanggaran[14].

Berdasarkan analisis pada aspek-aspek diatas. *Penetration testing* pada perangkat IoT merupakan hal yang sangat penting, karena pengujian ini membantu mengidentifikasi celah keamanan sebelum dapat dimanfaatkan oleh penyerang. Dengan melakukan *penetration testing* secara rutin, perusahaan dapat menemukan dan memperbaiki kelemahan di perangkat IoT mereka, sehingga mengurangi risiko penyerangan siber dan potensi kerugian finansial yang signifikan. Investasi dalam *penetration testing* dapat menjadi langkah proaktif yang tidak hanya melindungi aset digital, tetapi juga meningkatkan kepercayaan pelanggan dan reputasi perusahaan.

1.3 Analisa Solusi yang Ada

Saat ini sudah ada beberapa *software* yang dapat melakukan *penetration testing* pada protokol MQTT di perangkat IoT. Berikut adalah beberapa contoh *software* tersebut:

1.3.1 Solusi yang Ada

- **Nmap**

  Nmap merupakan singkatan dari *Network Mapper*, nmap dianggap sebagai pemindai jaringan yang paling bagus pada saat ini karena kemampuannya untuk mengelabui target sehingga menganggap bahwa pemindaian yang dilakukan bukan hanya berasal dari satu sumber Alamat IP[15].

  Nmap menggunakan paket IP mentah dengan cara yang inovatif untuk menentukan *host* mana yang tersedia di jaringan, layanan apa (nama aplikasi dan versi) yang ditawarkan oleh *host* tersebut, sistem operasi (dan versi OS) yang mereka jalankan, jenis pemfilteran paket/*firewall* yang digunakan, dan berbagai karakteristik lainnya[16].

- **Wireshark**

  Wireshark merupakan sebuah *tools* yang dirancang pada tahun 1998 oleh Gerald Combs, Bahasa utama yang digunakan dalam pengembangannya adalah C dan C++. Wireshark dianggap sebagai salah satu alat penetrasi jaringan internal. Wireshark sendiri bekerja dengan memantau lalu lintas yang terdapat dalam sebuah jaringan. Wireshark dapat memantau dan mengevaluasi paket MQTT yang dikirimkan antara perangkat IoT dan *broker*, apabila *broker* tidak menggunakan enkripsi tambahan seperti TLS/SSL, data yang dikirimkan akan berupa *plaintext* yang dapat dilihat langsung melalui wireshark[17]. Wireshark sendiri lebih berfokus pada analisis dan kurang memiliki fitur untuk melakukan eksploitasi secara langsung, sehingga memerlukan dukungan *tools* lainnya.

- **MQTT-PWN**

  MQTT-PWN merupakan salah satu *software open-source* yang dapat digunakan untuk melakukan *penetration testing* pada protokol MQTT. MQTT-PWN mengkombinasikan daftar yang lengkap, beberapa fungsi pendukung, serta modul eksploitasi yang dikemas dalam antarmuka *command-line* yang mudah dipahami dan digunakan[18].

- **MQTTSA**

  MQTT Security Assistant (MQTTSA) adalah alat yang dirancang untuk meningkatkan kesadaran keamanan para pengembang IoT dengan secara otomatis menilai kesalahan konfigurasi di lingkungan berbasis MQTT dan dengan memberikan laporan potensi kerentanan dan tindakan mitigasi pada tingkat detail yang berbeda - dari deskripsi bahasa alamiah hingga cuplikan kode yang bisa dipotong dan ditempelkan pada penerapan yang sebenarnya.

  MQTTSA bekerja dalam 2 langkah, pertama mendeteksi *vulnerability* dalam *broker* MQTT dengan melakukan inisiasi serangan untuk mengekspos *vulnerability* yang diketahui. MQTTSA kemudian membuat sebuah laporan berupa pdf yang berisi serangkaian tindakan yang dapat dilakukan untuk pencegahan serangan terhadap kerentanan yang ada[19].

- **Moxie**

  Moxie merupakan *script bash* yang didesain untuk melakukan penetrasi pada protokol MQTT perangkat IoT. Moxie memiliki beberapa fitur yaitu :

  - **Pemeriksaan Layanan MQTT**: Memeriksa apakah layanan MQTT dapat diakses pada alamat IP dan *port* tertentu.
  - **Pemindaian Lanjutan**: Melakukan pemindaian lanjutan menggunakan Nmap untuk mengumpulkan informasi terperinci tentang layanan MQTT.
  - **Serangan Brute-force**: Mencoba melakukan serangan *brute-force* terhadap autentikasi layanan MQTT menggunakan daftar kata nama pengguna dan kata sandi yang disediakan.

- **MQTTack**

  MQTTack merupakan sebuah *software/tools* berbasis *script bash* yang bertujuan untuk melakukan pengujian keamanan pada protokol MQTT. MQTTack hanya dapat digunakan terhadap *port* standard MQTT, yaitu pada *port* 1883[20].

1.3.2 Perbandingan *Software*

Tabel 1.1 Perbandingan fitur software yang ada

Fitur	MQTT-PWN	MQTTSA	Moxie	MQTTack
Dukungan Protokol Lain	Tidak	Tidak	Ya (CoAP, AMQP)	Tidak
Serangan Brute Force Credential	Ya	Tidak	Ya	Ya
DoS attack	Ya	Tidak	Ya	Ya
Wildcard Testing	Eksploitasi	Analisis	Eksploitasi	Ya
MITM	Terbatas	Tidak	Ya	Tidak
Analisis TLS	Tidak	Ya	Terbatas	Tidak
Port Scanning	Tidak	Tidak	Terbatas	Tidak
Fuzzing	Tidak	Tidak	Ya	Ya
User Interface (GUI)	Tidak	Tidak	Tidak	Tidak
Dokumen Laporan	Tidak	Ya	Tidak	Tidak

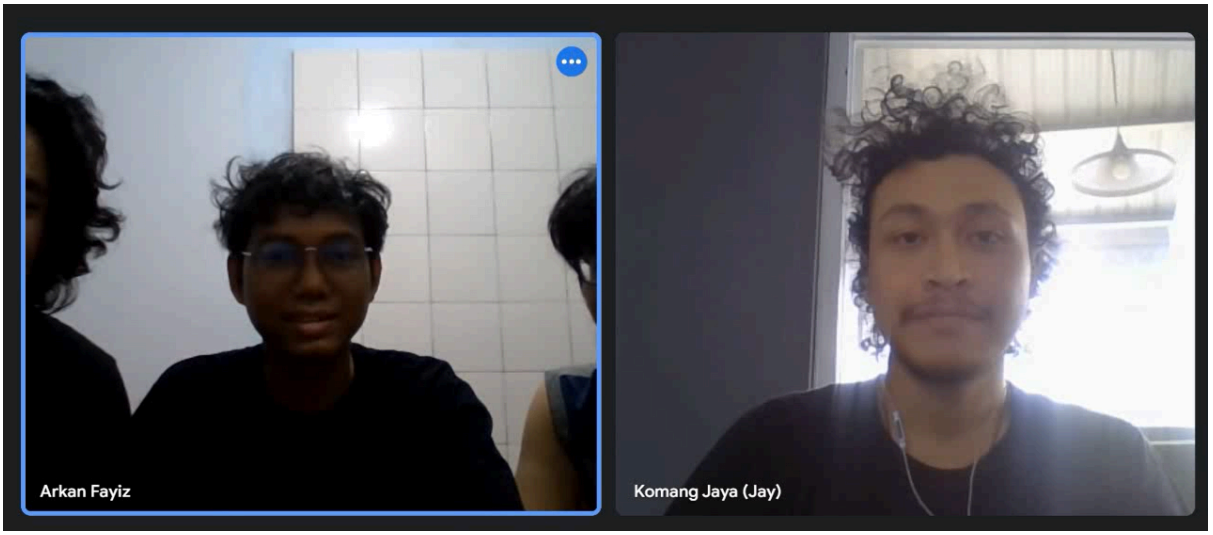
1.4 Kesimpulan

Dengan semakin banyaknya perangkat IoT yang menyebar luas di masyarakat, serangan kepada perangkat perangkat IoT semakin rentan dan semakin besar pula skala kerugian yang dapat dihasilkan dari serangan siber. Banyaknya cara untuk melakukan serangan siber seperti MITM, *brute-force*, dan *replay attack* ke perangkat IoT dan dengan sedikitnya keamanan pada perangkat IoT maka diperlukannya *pentesting* perangkat IoT oleh *developer* sebelum perangkat tersebut diperjualbelikan. Aplikasi yang akan kami kembangkan akan sangat membantu *developer* dalam mengamankan

perangkat IoT dan dengan diberlakukannya otomasi pada *software pentesting* tersebut maka *developer* juga dapat menghemat tenaga dan waktu.

1.5 Dokumentasi Wawancara

Wawancara dengan user yang merupakan developer IoT tentang masalah-masalah keamanan dan cara kerja IoT dari perspektif developer.



Gambar 1.1. Dokumentasi Wawancara dengan user

Berdasarkan wawancara dengan klien, aplikasi *pentest* yang dirancang akan difokuskan untuk menguji kerentanan pada perangkat IoT yang menggunakan protokol MQTT. Aplikasi ini diharapkan dapat bekerja secara otomatis dalam mendeteksi broker MQTT, mengidentifikasi celah keamanan seperti autentikasi lemah, port terbuka, dan kesalahan konfigurasi, dengan cara melakukan beberapa uji serangan serta dapat menghasilkan laporan yang terstruktur. Kemudian aplikasi juga diminta untuk memiliki otomasi tertentu, jadi klien hanya perlu memasukkan beberapa konfigurasi kemudian *software* akan berjalan secara otomatis melakukan proses.

Klien juga menekankan pentingnya ada *interface* yang sederhana dan mudah dipahami, sehingga dapat digunakan oleh *user* dengan pengalaman teknis yang baru. Selain itu, klien juga meminta dokumentasi yang jelas dan lengkap yang diperlukan untuk memandu penggunaan aplikasi. Dengan desain ini, aplikasi diharapkan mampu memenuhi kebutuhan klien dalam memastikan keamanan perangkat IoT yang menggunakan protokol MQTT.