

BAB I

PENDAHULUAN

1.1 Latar Belakang

Message Queue Telemetry Transport (MQTT) adalah protokol komunikasi data yang bersifat lightweight message, sehingga MQTT dapat bekerja dilingkungan dengan sumberdaya yang terbatas seperti kecilnya bandwidth dan terbatasnya sumber daya listrik, protokol ini juga dapat menjamin terkirimnya sebuah pesan walaupun koneksi terputus. Selain MQTT ada juga protokol HTTP yang memiliki kemampuan kurang lebih sama dengan MQTT akan tetapi protokol HTTP masih memiliki kekurangan seperti halnya penggunaan bandwidth yang cukup besar dan ukuran paket yang besar sehingga tidak reliable untuk berjalan pada sistem yang memiliki bandwidth rendah atau latency yang tinggi. Berdasarkan kekurangan protokol HTTP tersebut, implementasi protokol MQTT server sangat dibutuhkan untuk menunjang pengembangan IoT Platform. MQTT merupakan protokol komunikasi yang sangat sederhana dan ringan. Protokol MQTT juga didesain untuk alat berkemampuan terbatas, bandwidth yang rendah, latency yang tinggi dan jaringan yang kurang dapat diandalkan[1].

Hasil pengujian QoS dengan parameter delay diperoleh rata-rata delay MQTT QoS 0 sebesar 0,0017s, QoS 1 sebesar 0,0628805s, MQTT QoS 2 sebesar 0,16987s dan HTTP sebesar 0,124591s. Packet loss yang di dapatkan sebesar 0% untuk MQTT QoS 1, QoS 2 dan HTTP sedangkan QoS 0 sebesar 13,3333%, nilai throughput protokol MQTT lebih rendah +/-324,7943 Bytes/s dibandingkan protokol HTTP sehingga protokol MQTT dapat lebih reliable berjalan pada keadaan bandwidth rendah atau latency tinggi dibandingkan protokol HTTP.

Akan tetapi protocol MQTT memiliki banyak platform seiring dengan berkembangnya teknologi, berbagai keuntungan yang ditawarkan dari berbagai platform MQTT tersebut mencakup seperti pengiriman data yang instan, dan penyimpanan yang besar menjadi keuntungan dari berbagai platform yang berbeda. Akan tetapi dari banyaknya layanan Broker MQTT, kebanyakan orang masih belum mengetahui broker mana yang paling reliabel dan dapat mengirim data dengan efisien, oleh karena itu dari permasalahan tersebut pengembang ingin melakukan analisis pada dua platform MQTT yang berbeda terkait performa dari broker mana yang paling andal dalam mengatasi tantangan

pengiriman data IoT, memastikan kualitas layanan yang tinggi, serta memenuhi harapan pengguna.

Pada penelitian sebelumnya yang dikutip oleh buku Software Architecture Bahwa HiveMQ melakukan tes kinerja pada AWS menggunakan scenario fan-in dan fan-out. Misalnya, dalam skenario fan-in dengan QoS1, mereka mencapainya hingga 60K msg/s pada CPU 8 core[2]. ScaleAgent membandingkan JoramMQ, Apollo, Mosquitto, dan RabbitMQ menghasilkan 44K msg/s dan menyimpulkan bahwa Broker JoramMQ memiliki kinerja terbaik[2]. Dan ada juga penelitian sebelumnya dari Biswajeeban Mishra melakukan sebuah test yang menggunakan metode QoS dengan mengurutkan broker sesuai dengan server ke waktu pengiriman pesan klien maka dapat menghasilkan Message Delivery Time dengan hasil QoS 0 Mosquitto menghabiskan waktu paling sedikit untuk menyampaikan pesan ke klien dengan 656msg/s, lalu datang HiveMQ 1,006.48msg/s dan kemudian MQTT Server 1,006.67msg/s . Pada tingkat QoS 1 MQTT Server berada di urutan satu dengan 1,000.41msg/s, Mosquitto berada di urutan kedua dengan 1.006,77, dan HiveMQ berada di posisi ketiga 1,006.97. Pada tingkat QoS 2 Mosquitto kembali menduduki puncak daftar 936.7msg/s, HiveMQ berada di posisi kedua 1,006.80msg/s, MQTT Server berada di tempat ketiga dengan 1,011.86msg/s.[3].

Untuk mencari hasil pengujian yang lebih maksimal dari penelitian sebelumnya digunakan sebuah VPS yang merupakan sebuah teknologi virtualisasi dari sebuah physical server. VPS dapat berjalan mandiri yang memiliki full root access, system operasi, konfigurasi, user, pemrosesan CPU, dan RAM nya masing – masing sehingga VPS tidak mempengaruhi VPS lain dalam satu physical server[4]. Penggunaan VPS ini digunakan upaya untuk menganalisa performa pada protocol MQTT yang pada penelitian sebelumnya menghasilkan hasil performa yang tidak terlalu jauh satu sama lain. Oleh karena itu pada penelitian ini VPS sangat penting karena seluruh proses pengujian broker MQTT dan pengiriman data akan terus dilakukan oleh VPS demi menemukan performa yang teruji baik pada VPS.

Dengan menggunakan analisis QoS (Quality Of Service) pengembang dapat menganalisa protokol MQTT yang reliabel, dengan memahami dan mengukur parameter QoS (Quality Of Service) dapat memastikan pengiriman layanan yang berkualitas dan pengalaman yang baik bagi pengguna pada VPS secara keseluruhan[5]. Analisis QoS (Quality Of Service) memberikan alat untuk mengukur parameter secara objektif dan membuat keputusan berdasarkan data yang terukur seperti Latency, Throughput, Reabilitas, Kecepatan Respon dan Keandalan[6][7][2][8]

Dengan adanya Analisis QoS (Quality Of Service) pada 2 Platform MQTT yang berbeda dan perbandingan dari penelitian sebelumnya sebagai solusi, penelitian ini diharapkan dapat menemukan protokol MQTT yang reliabel dan dapat mendukung pengiriman data IoT pada VPS. Penelitian ini bertujuan untuk mengidentifikasi platform yang paling andal dalam mengatasi tantangan pengiriman data IoT, memastikan kualitas layanan yang tinggi, serta memenuhi harapan pengguna. Dengan fokus pada aspek-aspek krusial seperti Latency, Throughput, Reabilitas, Kecepatan Respon, dan Keandalan, upaya ini mendukung pengembangan aplikasi IoT yang dapat diandalkan dalam lingkungan yang semakin terkoneksi.

1.2 Identifikasi Masalah

Telah dilakukan identifikasi masalah dengan cara observasi, maka penulis dapat mengidentifikasi masalah ke dalam beberapa poin, yaitu sebagai berikut :

1. Bagaimana cara untuk menemukan protocol MQTT yang paling reliabel terkait tentang interaksi MQTT dalam lingkungan pengiriman data IoT?.
2. Bagaimana cara mengukur kinerja MQTT Kontainer Brocker pada VPS?.

1.3 Maksud dan Tujuan

Penelitian ini bertujuan meningkatkan pemahaman terkait interaksi antara MQTT Broker, dan infrastruktur VPS dalam mendukung pengiriman data IoT. Dengan Point berikut :

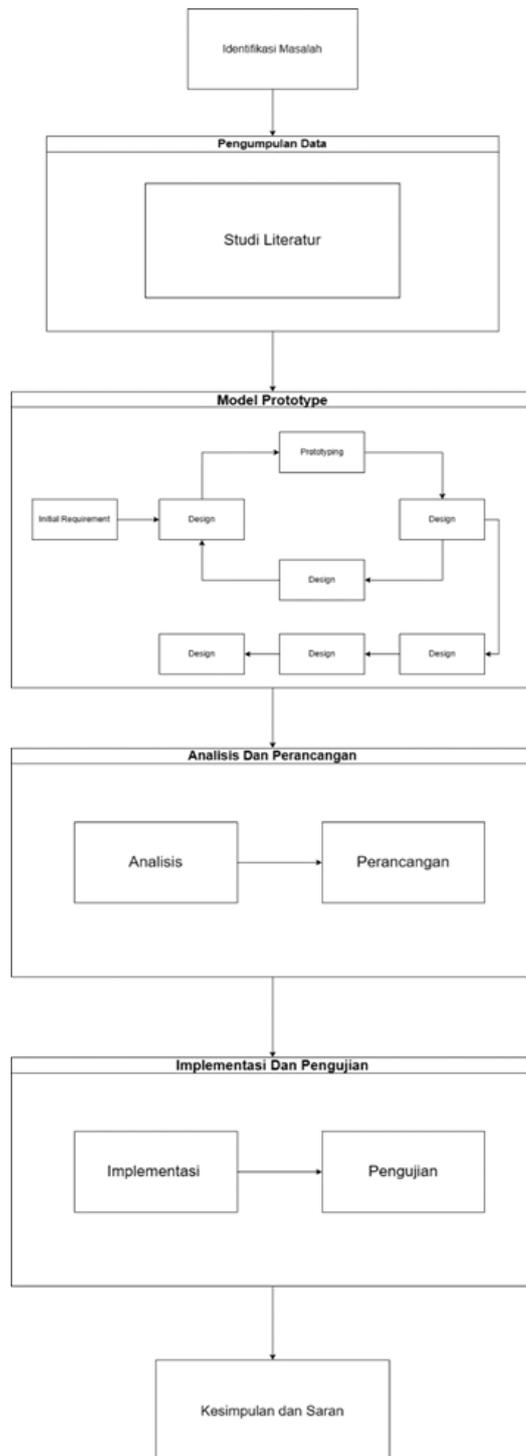
1. Mengimplementasikan Quality of Service untuk mencari MQTT Broker yang paling reliabel
2. Memberikan hasil pengujian yang dapat memberikan hasil pengukuran kinerja MQTT broker pada VPS untuk mencari hasil yang lebih maksimal

1.4 Batasan Masalah

Agar penelitian ini terarah dan sesuai, Adapun batasan masalah dalam penelitian ini sebagai berikut:

1. Pengujian pada kinerja MQTT Broker dalam pengiriman data IoT ke VPS.
2. Penyedia MQTT yang digunakan adalah Mosquitto dan HiveMQ.
3. Pengujian Delay, Throughput, dan Packet Loss menggunakan Wireshark
4. Waktu yang digunakan untuk pengujian 5 menit
5. Analisis akan dilakukan menggunakan Quality of Service

1.5 Metodologi Penelitian



Gambar 1. 1 Metodologi Penelitian

Berdasarkan gambar 1.1 diatas, menjelaskan terkait dengan tahapan atau alur dari penelitian dengan detail sebagai berikut[9] :

1. Identifikasi Masalah

Identifikasi masalah merupakan sebuah proses untuk menentukan dan memahami masalah yang ada. Identifikasi masalah dalam penelitian, dapat membantu peneliti untuk mencari topik penelitian dengan melakukan analisis data dan informasi untuk mengidentifikasi permasalahan yang mendasar dan mengkategorikan masalah sebagai solusi yang dapat diterapkan untuk menyelesaikan permasalahan, artinya dari topik penelitian yang diangkat akan didapat hasil sebagai solusi dari permasalahan yang ditemukan.

2. Pengumpulan Data

Pengumpulan data bertujuan untuk mengumpulkan informasi atau data yang relevan dan berkualitas untuk menjawab pertanyaan penelitian. Data dikumpulkan melalui berbagai sumber dan teknik, pada penelitian ini pengumpulan data dilakukan dengan membaca dan mengkaji studi literatur yang relevan untuk memperkuat topik penelitian dan melakukan wawancara sebagai tahap awal pengumpulan data dengan objek dan tempat penelitian. Data ini kemudian dianalisis untuk memperoleh hasil yang valid dan dapat dipercaya dalam penelitian

3. Perancangan Aplikasi

Pada tahap ini, dilakukan analisis dan juga perancangan aplikasi. Peneliti menggunakan metode analisis dan design berorientasi objek dengan tahapan analisis kebutuhan dilakukan untuk memenuhi tujuan dan kebutuhan yang ditentukan Seperti analisis data, analisis sistem yang berjalan, analisis kebutuhan fungsional hingga analisis kebutuhan non fungsional. Perancangan dilakukan sebagai upaya untuk melibatkan memahami dan memecahkan masalah yang akan ditemui oleh aplikasi, serta memastikan bahwa aplikasi tersebut mudah digunakan, skalabel, dan sesuai dengan kebutuhan pengguna. Perancangan sendiri terbagi menjadi perancangan arsitektur, perancangan jaringan semantik, dan perancangan antarmuka.

4. Pembangunan Aplikasi

Dalam proses ini, dilakukan tahap-tahap seperti implementasi perancangan, pemrograman, dan pengujian. Setelah melalui tahap perancangan, maka dilanjutkan dengan tahap pemrograman, dimana peneliti akan melakukan implementasi dari hasil perancangan sebelumnya sesuai dengan bahasa pemrograman yang telah ditentukan. Pada tahap pengujian, peneliti menggunakan metode *QoS*, dimana peneliti akan melakukan pengujian aplikasi secara fungsional. Tujuan akhir dari pembangunan aplikasi dalam penelitian adalah untuk memperoleh hasil yang berkaitan dengan penelitian yang sedang dilakukan.

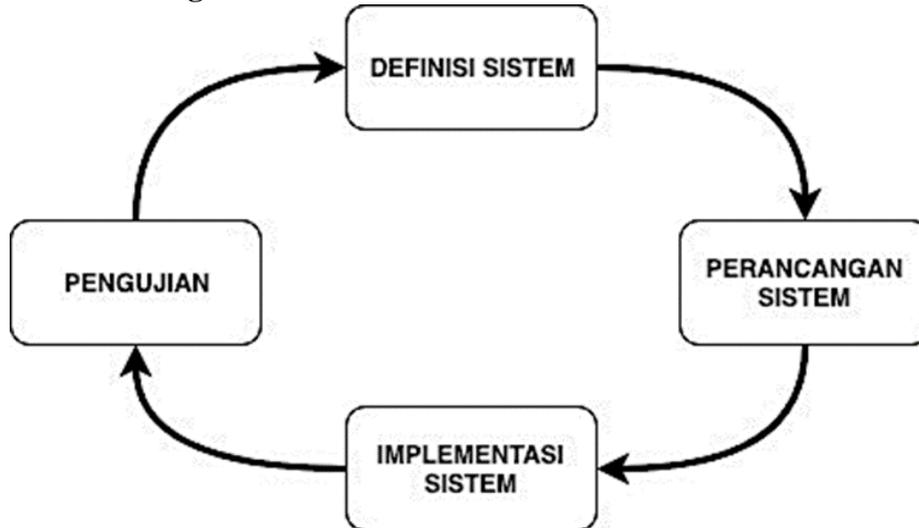
5. Hasil dan Kesimpulan

Tahap ini merupakan tahap terakhir dari penelitian, dimana peneliti dapat memberikan kesimpulan dan hasil dari penelitian. Peneliti dapat mengungkapkan hasil akhir dari penelitian yang mengacu pada temuan atau keterangan yang diperoleh dari penelitian. Peneliti dapat memberikan kesimpulan sesuai dengan rumusan tujuan yang diharapkan. Kesimpulan dan hasil dalam penelitian sangat penting karena membantu memahami hasil penelitian dan menentukan arah untuk penelitian selanjutnya.

1.6 Metode Pengumpulan Data

Metodologi yang akan digunakan penulis pada penelitian ini adalah metode kuantitatif. Metode kuantitatif adalah pendekatan penelitian yang berfokus pada pengumpulan dan analisis data kuantitatif (angka, statistik, dan ukuran numerik) untuk memahami fenomena, membuat generalisasi, dan menarik kesimpulan. Pendekatan ini melibatkan penggunaan instrumen-instrumen yang dirancang untuk mengumpulkan data yang dapat diukur secara numerik, seperti survei, uji coba eksperimental, analisis statistik, dan pendekatan-pendekatan lain yang berfokus pada kuantitas dan pengukuran. Metode kuantitatif sering kali digunakan untuk menjawab pertanyaan-pertanyaan penelitian yang terukur, menetapkan hubungan sebab-akibat, dan mengidentifikasi pola atau tren dalam data yang dikumpulkan.

1.7 Metode Pembangunan Sistem



Gambar 1. 2 Metode Pembangunan Sistem

Pada penelitian ini digunakan sebuah metode prototype yang dimana dimulai dengan tahap Definisi Sistem, Perancangan Sistem, Implementasi, dan Pengujian[9].

1. Definisi Sistem

Pada tahap ini didefinisikan sistem yang akan dibangun yaitu Analisis perbandingan mqtt broker menggunakan analisis Quality of Service (QoS) menggunakan VPS untuk pengiriman data IoT

2. Perancangan Sistem

Perancangan Sistem mencakup konfigurasi parameter QoS apa saja yang akan di monitor.

3. Implementasi Sistem

Pada bagian ini di implementasikan sebuah website yang bisa terintegrasi dengan VPS, yang dapat memonitor traffic yang terjadi pada saat pengiriman data IoT.

4. Pengujian

Pada tahap ini system akan diuji hingga mendapat hasil yang sesuai

1.8 Sistematika Penulisan

Sebagai acuan penulis agar penulisan skripsi dapat tersusun secara sistematis dan sesuai dengan harapan, serta memberikan gambaran umum terkait penulisan penelitian maka berikut sistematika penulisan dalam penelitian ini:

BAB 1 PENDAHULUAN

Pada bab ini, berisikan uraian secara umum terkait latar belakang masalah, identifikasi masalah, maksud dan tujuan, batasan masalah, metodologi penelitian, serta sistematika penulisan penelitian yang dilakukan.

BAB 2 TINJAUAN PUSTAKA

Pada bab ini, berisikan tinjauan umum mengenai berbagai teori pendukung serta konsep dasar mengenai aplikasi yang dibangun.

BAB 3 ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini, berisikan pemaparan terkait dengan analisis masalah, analisis data, analisis kebutuhan *nonfungsional*, dan analisis kebutuhan *fungsional*. Hasil dari analisis akan digunakan sebagai acuan perancangan perangkat lunak yang terdiri dari perencanaan struktur menu, perancangan jaringan semantik, dan sebagainya.

BAB 4 IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini, berisikan hasil implementasi analisis yang didapat dari BAB 3 dan pembangunan aplikasi yang telah dilakukan, serta hasil pengujian aplikasi untuk mengetahui apakah sistem keamanan yang dibangun sudah sesuai dan memenuhi kebutuhan.

BAB 5 KESIMPULAN DAN SARAN

Pada bab ini, berisikan kesimpulan yang diperoleh dari hasil penelitian dan saran untuk pengembangan aplikasi di masa mendatang.