

PENERAPAN PROTOKOL MQTT PADA TEKNOLOGI WAN (STUDI KASUS SISTEM PARKIR UNIVERISTAS BRAWIJAYA)

**Galih Yudha Saputra¹⁾, Ahimsa Denhas Afrizal²⁾, Fakhris Khusnu Reza Mahfud³⁾,
Farid Angga Pribadi⁴⁾, Firman Jati Pamungkas⁵⁾**

Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh November
Kampus Keputih Sukolilo, Surabaya, 60111

E-Mail : : gygalihyudha36@gmail.com ¹⁾, ahimsa.afrizal@gmail.com ²⁾, fakhris.cr7@gmail.com ³⁾,
anggatpl07@gmail.com ⁴⁾, firemanjp88@gmail.com ⁵⁾,

ABSTRAK

MQTT merupakan sebuah protokol yang diterapkan pada IOT. Protokol ini sangat mendukung untuk jaringan WAN, karena WAN mencakup area yang luas. Protokol MQTT mempunyai kelebihan yaitu dapat bekerja dengan energi dan media penyimpanan yang minimum. Jaringan pada sistem informasi parkir yang saat ini kebanyakan hanya bersifat LAN. Oleh karena itu pada penelitian ini sistem informasi parkir dikembangkan dengan menggunakan protokol MQTT yang diterapkan pada jaringan WAN. Berdasarkan hasil simulasi penelitian ini protokol MQTT dapat diterapkan pada teknologi WAN dengan average delay sebesar 0.028183014 second membuktikan bahwa protokol MQTT mempunyai kualitas yang baik berdasarkan parameter nilai delay. Selain itu packet loss dari publisher ke server pada protokol MQTT yaitu sebesar 0%, hal ini membuktikan bahwa akurasi pengiriman pada MQTT adalah 100%.

Kata Kunci : MQTT, WAN, IOT

1. PENDAHULUAN

Parkir telah menjadi salah satu hal yang krusial dalam lalu lintas di jalan, banyaknya jumlah alat transportasi yang membuat kebutuhan lahan sebagai sarana parkir alat transportasi tersebut yang semakin sempit. Kendaraan mobil dan motor sebagai alat transportasi utama yang membutuhkan tempat lebih luas sebagai area parkir. Tempat parkir tersebut dapat menjadi kendala bagi pengendara untuk mengetahui lahan parkir di bagian mana yang masih kosong. Sehingga, mengharuskan para pengendara mengelilingi lahan parkir untuk mencari tempat yang kosong.

Permasalahan di atas menimbulkan kesulitan pada penyediaan lahan yang luas terutama bagi mahasiswa Universitas Brawijaya. Dikarenakan hampir setiap mahasiswa memiliki kendaraan masing-masing, namun lahan parkir di Universitas Brawijaya tidak terlalu luas dan ditambah lagi peningkatan jumlah kendaraan di Universitas Brawijaya yang menampung mahasiswanya hingga 15000 tiap tahunnya, dan mengakibatkan lahan parkir yang semakin sempit dan ketidak tersediaan untuk mendapatkan lahan parkir.

Sistem parkir pada umumnya saat ini masih menggunakan teknologi LAN, oleh karena itu penelitian ini mengusulkan untuk diterapkannya MQTT. Penelitian ini diharapkan dapat memberikan solusi untuk kebutuhan pengguna parkir secara *real-time* tanpa harus kehilangan banyak waktu untuk mencari tempat parkir yang tersedia. Selain itu dengan kebutuhan energi yang rendah dan media penyimpanan terbatas dapat diatasi dengan menggunakan protokol mqtt yang diterapkan pada WAN.

Dengan memanfaatkan teknologi jaringan saat ini mengalami perkembangan yang pesat diantaranya perpaduan antara control technology dan Internet of Things (IOT). Internet of things (IOT) merupakan paradigma baru yang bertujuan dari konektivitas untuk segalanya yaitu dengan meminimalkan penyimpanan dan power. MQTT merupakan salah satu transmisi protokol yang dikembangkan oleh IBM terutama digunakan Internet Of Things, MQTT didesain untuk dapat melakukan publish / subscribe sebuah transmisi pesan yang ringan. MQTT dapat digunakan untuk optimasi jaringan dengan bandwidth rendah dan perlengkapan computing energi minimum, MQTT digunakan untuk bermacam-macam skenario aplikasi dari Internet of Things (IOT).

Beberapa penelitian sebelumnya menyatakan bahwa MQTT dapat digunakan sebagai solusi dari sistem remote wireless pada pot cerdas [1]. selain itu Penerapan MQTT pada lingkungan IOT juga merupakan solusi dengan daya rendah yang digunakan untuk meningkatkan deteksi kegagalan pada perusahaan tenaga listrik [2]. MQTT dalam mengirim pesan mengalami lebih rendah penundaan dari COAP untuk packet loss yang lebih rendah, dan lebih tinggi penundaan dari COAP untuk packet loss lebih tinggi. Bahkan ketika pesan berukuran kecil dan loss rate lebih kecil samadengan 25%, CoAP menghasilkan sedikit extra traffic daripada MQTT untuk memastikan transmisi yang handal [3].

Pada penelitian ini akan merancang sebuah sistem yang dapat menginformasikan keadaan kuota tempat parkir mobil, dimana dalam sistem tersebut akan memberikan informasi kepada

pengguna tentang keadaan tempat parkir dalam keadaan penuh atau masih bisa ditempati dengan detail kuota yang dimiliki tempat parkir tersebut. Sistem keadaan tempat parkir ini juga diharapkan dapat mengurangi tingkat kemacetan, kehilangan kendaraan serta dapat dijadikan sebagai teknologi tempat parkir yang efisien dari sistem informasi ini yaitu pengguna tempat parkir dapat mengetahui keadaan tempat parkir lengkap dengan detail kuota yang terpakai dari jumlah kuota yang tersedia, sehingga pengguna dapat mengetahui dengan cepat informasi mengenai keadaan tempat parkir yang ada.

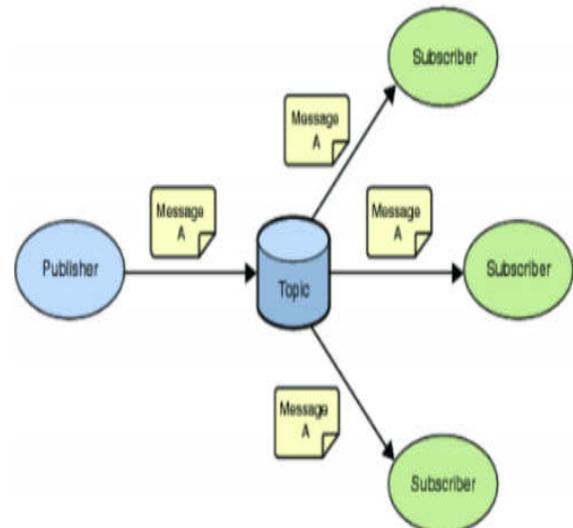
2. TINJAUAN PUSTAKA

A. IOT dan MQTT

Protokol MQTT (Message Queue Telemetry Transport) adalah protokol pesan ringan (lightweight) berbasis publish-subscribe digunakan di atas protokol TCP/IP. Protokol ini mempunyai ukuran paket data *low overhead* kecil (minimal 2 gigabyte) dengan konsumsi daya kecil. MQTT bersifat terbuka, simpel dan didesain agar mudah untuk diimplementasikan, yang mampu menangani ribuan *client* jarak jauh dengan hanya satu server. Karakteristik ini membuatnya ideal untuk digunakan dalam banyak situasi, termasuk lingkungan terbatas seperti dalam komunikasi *Machine to Machine* (M2M) dan konteks *Internet of Things* (IoT) dimana dibutuhkan kode footprint yang kecil dan/atau jaringan yang terbatas. Pola pesan publish-subscribe membutuhkan broker pesan. Broker bertanggung jawab untuk mendistribusikan pesan ke klien tertarik berdasarkan topik pesan[4].

Berikut merupakan fitur protokol MQTT[5]:

1. *Publish/subscribe message pattern* yang menyediakan distribusi *message* dari satu ke banyak dan *decoupling* aplikasi.
2. *Messaging transport* yang *agnostic* dengan isi dari payload.
3. Menggunakan TCP/IP sebagai konektivitas dasar jaringan.
4. Terdapat tiga level *Qualities of Service* (Qos) dalam penyampaian pesan :
 - 4.1 “*At most once*”, di mana pesan dikirim dengan upaya terbaik dari jaringan TCP/IP. Kehilangan pesan satau terjadi duplikasi dapat terjadi.
 - 4.2 “*At least once*”, dapat dipastikan pesan tersampaikan walaupun duplikasi dapat terjadi.
 - 4.3 “*Exactly once*”, dimana pesan dapat dipastikan tiba tepat satu kali.



Gambar 1. Arsitektur MQTT

B. Implementasi MQTT

Berdasarkan berbagai keunggulan khusus MQTT tersebut, banyak peneliti berusaha mengimplementasikan protokol tersebut di berbagai bidang. Zambrano et al (2016) mengimplementasikan MQTT dan smartphone sebagai alat untuk peringatan dini gempa bumi [6]. MQTT digunakan karena lebih efisien dalam mengkonsumsi *bandwith*, serta menggunakan daya baterai yang tidak besar, sehingga sangat cocok dalam situasi darurat seperti gempa bumi.

MQTT juga diterapkan untuk memonitor detak jantung pasien secara jarak jauh [7]. Chooruang dan Mangkalakeeree (2016) mengkombinasikan teknologi wifi modul ESP8266 pada Arduino microcontroller dan protokol MQTT untuk mentransmisikan data. MQTT dipilih sebagai protokol karena sangat ringan digunakan untuk berkomunikasi.

Sistem yang dikembangkan memiliki fungsi untuk memonitor detak jantung pasien secara real time. Sistem ini dirancang menggunakan infrared dan photo detector untuk mendeteksi dan mengambil datasyinal detak jantung, lalu mengirimkan datanya kepada broker MQTT yang berjalan pada Raspberry pi, sebuah komputer dengan harga murah seukuran kartu kredit. Hasilnya, penerapan teknologi ini sangat membantu petugas kesehatan untuk memonitor detak jantung pasien.

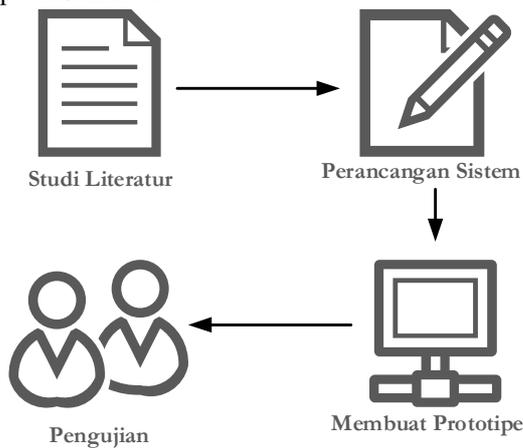
Beberapa tahun belakangan ini, implementasi *Internet of Things* (IoT) pada peralatan rumah tangga semakin marak. Tren ini juga didukung oleh murahnya peralatan smartphone yang digunakan sebagai alat input dan kontroler bagi IoT. MQTT sebagai protokol komunikasi dengan *bandwith* rendah turut mendukung implementasi IoT. Hartalkar., et al (2015) dalam penelitiannya melakukan otomasi peralatan rumah tangga menggunakan teknologi *Global System for Mobile Communication* (GSM) [8]. *Serial Communication*

dan *AT Commands* digunakan dalam implementasi GSM pada peralatan rumah tangga. Fokus pada penelitian ini adalah pada fungsionalitas Apps Inventor yang melakukan konversi dari suara percakapan ke teks. Dengan konversi dan penggunaan MQTT sebagai protokol komunikasi pada penelitian tersebut, memudahkan penyandang disabilitas maupun orang lanjut usia dalam memberikan perintah pada *IoT* di dalam saluran komunikasi yang aman.

Berbagai penelitian yang sudah ada tentang penggunaan protokol MQTT untuk menghubungkan mesin dengan mesin maupun pada *IoT*, dapat disimpulkan bahwa protokol MQTT dapat diimplementasikan dan handal digunakan untuk berbagai bidang. Pada penelitian ini, protokol MQTT digunakan sebagai protokol komunikasi untuk sistem parkir Mall real time di Kota Surabaya

C. Metodologi

Tahapan penelitian diawali dengan melakukan, studi literature, perancangan sistem dan menentukan komponen yang akan digunakan, membuat prototip software, dan melakukan pengujian. Diagram alir tahapan penelitian seperti pada Gambar 2.



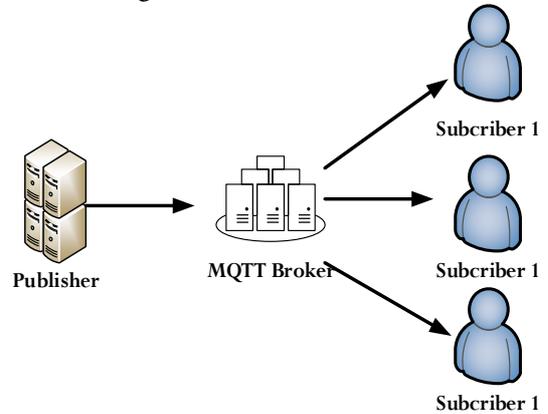
Gambar 2. Alur Tahapan Penelitian

3. HASIL DAN PEMBAHASAN

A. Desain Sistem

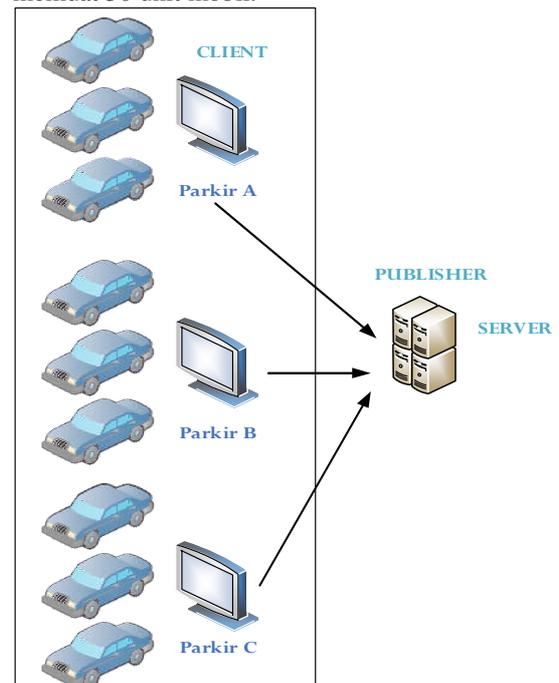
Sistem yang dibuat merupakan gabungan dari 2 metode, dimana pada sistem ini metode yang diterapkan yaitu menggunakan MQTT (*Message Queuing Telemetry Transport*) dan RPC (*Remote Procedure Call*). Pada MQTT terdapat 3 komponen penting didalamnya, dimana terdapat *Publisher*, *Broker*, dan *Subscriber*. Dalam MQTT ini terdapat 3 topik yang akan di ikuti oleh *subscriber*. Topik tersebut yaitu keadaan tempat parkir mobil yang dalam implementasinya diwakili oleh *publisher*. *Publisher* tersebut akan memberikan informasi kepada *subscriber* melalui *broker* sesuai dengan topik yang diikuti oleh *subscriber*. Dalam sistem ini, terdapat beberapa *subscriber* yang semuanya adalah pengguna tempat parkir yang dalam hal ini diwakili oleh mahasiswa, dosen ataupun staff. Setiap *subscriber* berhak memilih topik sesuai dengan

yang diharapkan yaitu tempat parkir yang akan dituju. Ketika suatu *subscriber* memilih topik, maka *subscriber* tersebut akan mendapatkan informasi keadaan tempat parkir yang ada di kampus. Dimana informasi tersebut akan memberitahu tempat parkir dalam keadaan penuh atau masih bisa di tempati. Rancangan desain MQTT yang dibangun dapat dilihat dalam gambar 3 dibawah ini.

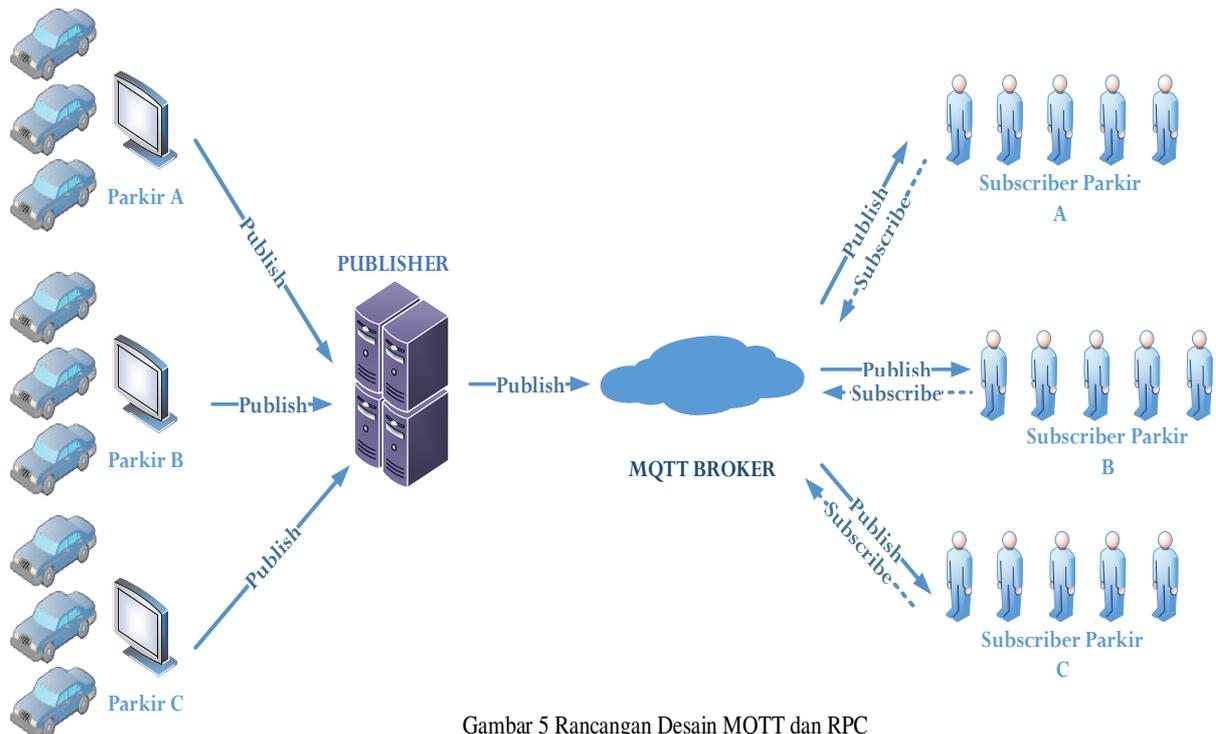


Gambar 3 Desain Topologi pada Metode MQTT

Penggunaan metode kedua yaitu RPC dimana terdapat 2 komponen yaitu *client* dan *server*. Pada desain sistem, server RPC diwakili oleh *Publisher* yang akan terhubung dengan *Client* yang diwakili oleh tempat parkir sesuai topik yang ada. Dalam topologi terdapat 3 tempat parkir, yaitu parkir A, parkir B, dan parkir C yang ketiganya harus memberikan informasi tentang keadaan tempat parkir. Keadaan yang dimaksud yaitu penuh tidaknya tempat parkir dengan detail jumlah kendaraan yang masuk pada tempat parkir yang ada. Untuk tempat parkir mobil diindikasikan dapat memuat 50 unit mobil.



Gambar 4 Desain Topologi pada Metode RPC



Gambar 5 Rancangan Desain MQTT dan RPC

Komunikasi antara tempat parkir yang dalam hal ini sebagai *client* dengan *publisher* sebagai server akan dilakukan menggunakan metode RPC. Jadi, masing-masing tempat parkir dapat mengakses sebuah prosedur yang berada pada *publisher* untuk memberikan informasi terbaru mengenai keadaan tempat parkir. Rancangan desain RPC yang dibangun dapat dilihat pada gambar 4.

Dari kedua metode tersebut akan digabungkan menjadi suatu sistem penuh yang akan berjalan sesuai dengan kebutuhan yang akan diinginkan. Dengan gabungan 2 metode tersebut, informasi keadaan tempat parkir akan diinformasikan dari setiap tempat parkir yang ada kemudian akan dikirimkan ke *publisher*. Setelah terkumpul pada *publisher*, informasi akan disampaikan kepada *subscriber* sesuai dengan topik yang diinginkan melalui *broker*. Rancangan desain keseluruhan menggunakan metode MQTT dan RPC dapat dilihat pada gambar 5 dibawah ini.

B. Node Publisher dan Subscriber

Node publisher dan subscriber di dalam arsitektur sistem merupakan komponen penting yang akan digunakan oleh pengguna baik untuk melakukan input data kendaraan yang parkir ataupun menerima data yang dikirimkan oleh server. Dalam implementasi sistem penelitian ini, proses input data masih menggunakan cara manual dengan menggunakan form pengisian parkir yang akan diisi oleh petugas parkir. Data yang telah diisi akan disimpan ke didalam server untuk selanjutnya akan di salurkan oleh broker kepada pengguna parkir.

Node publisher disimulasikan akan digunakan petugas parkir menggunakan *Personal Computer* (PC) dengan koneksi jaringan internet. Program pada node publisher diimplementasikan menggunakan bahasa python. Ada 2 fungsional utama pada node publisher, fungsional pertama digunakan langsung oleh petugas parkir untuk melakukan inialisasi data kendaraan parkir, proses ini akan mengalami *looping* ketika proses pengisian selesai untuk dapat melakukan proses pengisian data selanjutnya. Fungsional kedua pada node publisher untuk melakukan pengiriman data ke server broker dan mengatur topik yang akan dipublikasi oleh server broker MQTT. *Node subscriber* disimulasikan akan digunakan oleh pengguna layanan parkir. Fungsional utama pada node subscriber adalah menampilkan data jumlah kendaraan yang ada di lokasi parkir.

C. MQTT Broker

Server/Broker MQTT menggunakan perangkat lunak *Mosquitto* yang dapat berjalan pada sistem operasi Windows, ataupun Linux. Eclipse mosquitto™ merupakan broker pesan open source (EPL / EDL berlisensi) yang mengimplementasikan MQTT protokol versi 3.1 dan 3.1.1. MQTT menyediakan metode yang ringan untuk dapat mempublikasikan / berlangganan pesan sesuai dengan topik pesan apa yang diinginkan. Hal ini membuat eclipse mosquitto cocok untuk "Internet of Things" dengan sensor daya rendah atau perangkat mobile seperti ponsel, sistem tertanam atau mikrokontroler seperti Arduino. Server broker MQTT secara default menggunakan port 1833 walaupun di dalam implementasinya server MQTT dapat menggunakan beberapa port lain dengan

fungsi yang berbeda. Beberapa port yang digunakan di dalam server broker MQTT Mosquitto seperti:

- 1883 : MQTT, unencrypted
- 8883 : MQTT, encrypted
- 8884 : MQTT, encrypted, client certificate required
- 8080 : MQTT over WebSockets, unencrypted
- 8081 : MQTT over WebSockets, encrypted

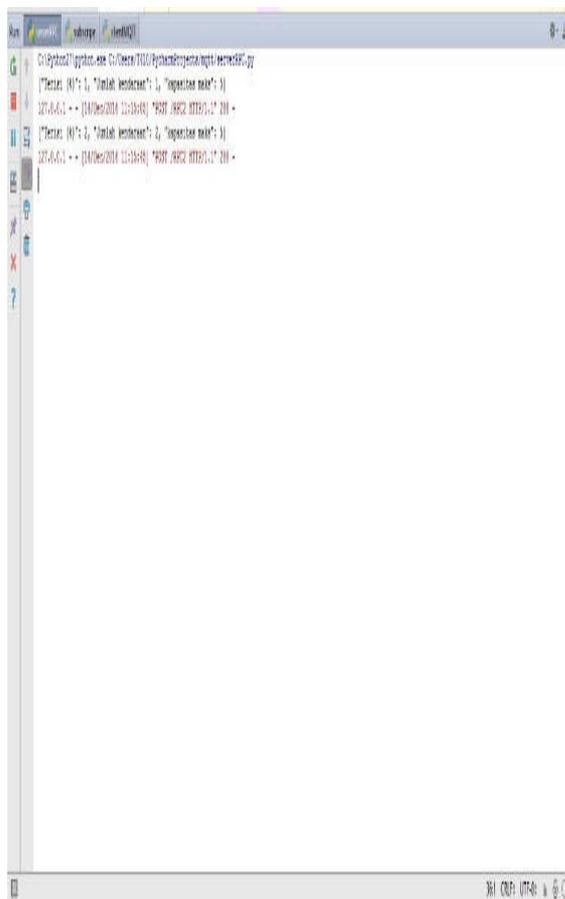
Jarak lokasi parkir Universitas Brawijaya yang saling berjauhan satu sama lain dan juga adanya pengembangan lokasi kampus yang berbeda membuat server broker yang akan digunakan tidak memungkinkan diimplementasikan menggunakan jaringan LAN. Sehingga dibutuhkan topologi jaringan yang dapat menjangkau area yang lebih luas. Di dalam penelitian ini, server broker akan ditanam disebuah server cloud VPS sehingga proses input data dan pengiriman data dapat dikirimkan dari jarak yang berjauhan menggunakan internet.

D. Hasil Input dan Output pada Server

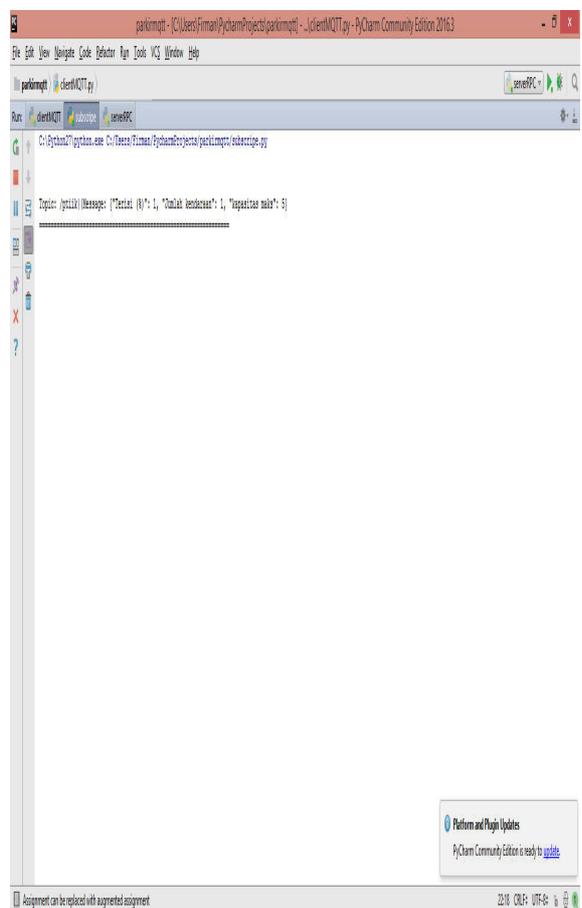
Hasil input dan output data pada server node publisher dan subscriber diperlihatkan masing masing pada Gambar 10 dan Gambar 11. Tampilan input kendaraan parkir disimulasikan pada node subscriber, output jumlah kendaraan parkir disimulasikan pada node publisher.



Gambar 7 Output Client MQTT



Gambar 6 Input Node Publisher



Gambar 8 Output Publisher MQTT

Pengujian sistem dilakukan dengan mengirimkan data dari *node publisher* ke komputer broker server yang terinstall Mosquitto MQTT sebelum akan diteruskan kepada *node subscribe* yang akan digunakan oleh pengguna layanan parkir. Pengiriman dilakukan setelah proses input data di *node publisher* dilakukan, proses ini akan diloopng jika ada data kendaraan parkir yang dimasukkan ke dalam sistem. Kemudian dilakukan pengukuran kinerja oleh paket data yang dikirimkan dari *node publisher* menuju server broker dan *node subscribe* dengan parameter *delay*, dan *packet loss*. Pengujian yang dilakukan menggunakan perangkat lunak Wireshark untuk dapat menangkap lalu lintas data yang ada di dalam sistem.



Gambar 9 Server Broker MQTT Mosquitto

E. Pengujian Packet Loss

Packet Loss adalah banyaknya paket yang hilang pada suatu jaringan paket yang disebabkan oleh tabrakan (*collision*), penuhnya kapasitas jaringan, dan penurunan paket yang disebabkan oleh habisnya TTL (*Time To Live*) dari paket. Parameter packet loss menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang, dapat terjadi karena *collision* dan *congestion* pada jaringan. Nilai packet loss sesuai dengan versi TIPHON (*Telecommunications and Internet Protocol Harmonization Over Networks*) (Joesman 2008) sebagai berikut :

Tabel 1. Parameter Nilai Packet Loss

Kategori Degradasi	Packet Loss	Indeks
Sangat Bagus	0 %	4
Bagus	3 %	3
Sedang	15 &	2
Jelek	25 %	1

Cara menghitung *packet loss* dapat menggunakan persamaan:

$$Packet\ loss = \frac{paket\ total\ tercapture - paket\ terkirim}{paket\ total\ tercapture} \times 100\%$$

Pengujian *Packet loss* menghasilkan 0%. Hal ini dibuktikan dengan pantauan dan analisa menggunakan perangkat bantuan wireshark. Terlihat bagaimana jumlah bit frame data yang dikirimkan dari satu node sama dengan jumlah bit data yang diterima oleh node yang lain.

F. Pengujian Delay

Adalah waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. Delay dapat dipengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama. Delay merupakan akumulasi berbagai waktu tunda dari ujung ke ujung pada jaringan Internet. *Delay* mempengaruhi kualitas layanan (*QoS*) karena waktu tunda menyebabkan suatu paket lebih lama mencapai tujuan. ITU-T G.114 merekomendasikan *Delay* tidak lebih besar dari 150 ms untuk berbagai aplikasi, dengan batas 400 ms untuk komunikasi multimedia yang masih dapat diterima. Sementara itu untuk aplikasi *Voice* seperti VoIP, batasan 3 *delay* maksimal adalah 300 ms.

Dalam menghitung *delay* dapat menggunakan persamaan:

$$Delay\ (sec) = \frac{Waktu\ paket\ yang\ dikirim}{Jumlah\ paket}$$

Tabel 2. Parameter Nilai Delay

Delay (ms)	Kualitas
0 – 150	Baik
150 – 400	Cukup, masih dapat diterima
> 400	Buruk

Dalam arsitektur sistem yang diimplementasikan kali ini, pengujian delay dilakukan menggunakan bantuan wireshark. Di dalam wireshark dilihat bagaimana time delta yang ditangkap setiap proses pengiriman data, baik itu pengiriman data dari *node publisher* ke server broker dan sebaliknya ataupun server broker ke *node subscriber*. Dari setiap time delta yang didapatkan maka diambil rata-rata dari keseluruhan data yang ditangkap. Hasil penghitungan rata-rata menunjukkan besaran delay yang didapatkan sebesar 0.028183014 second.

4. KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan pengujian dan analisis kinerja protokol MQTT pada *Smart Building* berbasis *platform* OpenMTC, maka diperoleh kesimpulan:

1. Protokol MQTT dapat diimplementasikan dengan menggunakan topologi WAN.
2. Pengujian implementasi protokol MQTT dengan parameter *delay* dihasilkan *average delay* sebesar 0.028183014 second. Dapat disimpulkan bahwa *delay* pengiriman data pada protokol MQTT memiliki kualitas yang baik berdasarkan parameter nilai *delay*.
3. Pengujian implementasi protokol MQTT dengan parameter *packet loss* dari *publisher* ke *server* yaitu 0% sehingga dapat disimpulkan bahwa akurasi pengiriman data pada pengujian ini adalah 100%.

B. Saran

Berdasarkan penelitian Tugas Akhir ini, penulis memiliki saran untuk pengembangan selanjutnya, yaitu:

1. Dalam penelitian ini hanya melakukan simulasi sistem saja, sehingga untuk penelitian kedepannya diharapkan dapat membuat GUI untuk memudahkan admin parkir dan user dalam mengakses sistem parkir.
2. Input data kendaraan masih bersifat manual, sehingga kedepannya diharapkan untuk dapat menggunakan sensor atau RFID pada kendaraan yang masuk, sehingga input data dapat dilakukan dengan otomatis.
3. Server broker hanya sementara, sehingga jika server broker mati maka data yang ada tidak dapat disimpan dan harus mengulang menginput data dari awal.

5. DAFTAR PUSTAKA

- [1]. Xing, J., 2016. Study on Remote Wireless Smart Pot System Based on ZIGBEE+MQTT. *Int. J. Future Gener. Commun. Netw.* 9, 1–8. doi:10.14257/ijfgcn.2016.9.5.01.
- [2]. Eslava, H., Rojas, L.A., Pereira, R., 2015. Implementation of Machine-to-Machine Solutions Using MQTT Protocol in Internet of Things (IoT) Environment to Improve Automation Process for Electrical Distribution Substations in Colombia. *J. Power Energy Eng.* 03, 92–96. doi:10.4236/jpee.2015.34014.
- [3]. Thangavel, D., Ma, X., Valera, A., Tan, H.-X., Tan, C.K.-Y., 2014. Performance evaluation of MQTT and CoAP via a common middleware, in: *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 2014 IEEE Ninth International Conference On. IEEE, pp. 1–6.
- [4]. Lampkin V, Leong W.T, Olivera L, Rawat S, Subrahmanyam N, Xiang R. Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry. *IBM Redbooks.* 2012:5-33.
- [5]. D. Locke, 2010. "MQ Telemetry Transport (MQTT) V3.1 Protocol,".
- [6]. Zambrano AM, Perez I, Palau C, Esteve M., 2016. Technologies of Internet of Things applied to an Earthquake Early Warning System.
- [7]. Chooruang K, Mangkalakeeree P., 2016. Wireless Heart Rate Monitoring System using MQTT, *International Electrical Engineering Congress*, 2-4 March 2016, Chiang Mai.
- [8]. Haltalkar T, Bhore S, Borawake K, Naik S, 2015, GSM Base Home Automation using MQTT, *International Journal of Engineering Technology, Management and Applied Science*, September 2015, Vol 3, Issue 9, ISSN 2349-4476