

## ANALISIS PERFORMANSI PROTOKOL COAP DAN MQTT-SN PADA SISTEM SMARTHOME DENGAN COOJA NETWORK SIMULATOR

### PERFORMANCE ANALYSIS OF COAP AND MQTT-SN PROTOCOL ON SMARTHOME SYSTEM USING COOJA NETWORK SIMULATOR

Satria Bagus Pratama, Rendy Munadi<sup>2</sup>, Akhmad Syauqi<sup>3</sup>

<sup>1,2</sup>Prodi S1 Teknik Telekomunikasi, Fakultas Teknik, Universitas Telkom

<sup>3</sup> PT. Telkom Akses

<sup>1</sup>virdelynov@students.telkomuniversity.ac.id, <sup>2</sup>rendymunadi@telkomuniversity.ac.id,

<sup>3</sup>syauqikarim@gmail.com

#### Abstrak

*Wireless Sensor Network* merupakan salah satu teknologi yang paling berkembang riset dan pengembangannya di dunia. Perkembangan tersebut didukung oleh berkembangnya juga pengaplikasian *Internet of Things* yang memungkinkan teknologi WSN digunakan dan dimanfaatkan lebih dalam. Merupakan sebuah kemajuan besar apabila WSN diterapkan untuk kebutuhan rumah tangga sehari-hari manusia, namun untuk mengimplementasikan hal tersebut terdapat tantangan tersendiri bagi WSN. Banyaknya *nodes* yang akan diimplementasikan dalam jaringan menjadi sebuah masalah terhadap beban daya sebuah jaringan WSN. Diperlukan sebuah jaringan wireless yang hemat dalam penggunaan daya demi memiliki *lifetime* yang panjang. Penggunaan protokol 802.15.4 pada WSN dianggap sebagai protokol yang cocok untuk kebutuhan tersebut, fitur yang dibutuhkan seperti daya rendah dan dapat berjalan pada besar *bandwidth* yang kecil mampu dipenuhi protokol tersebut. Pada penelitian kali ini hal tersebut akan dibuktikan dengan melakukan simulasi terhadap *prototype smarthome* milik PT. Telkom Akses dengan skema penambahan pada jumlah *nodes* mulai dari 5, 8 hingga 10 *nodes*. Simulasi dilakukan dengan *Cooja network simulator* dan akan dilakukan pengukuran parameter *delay*, *throughput*, *packet received* dan *power consumption* dari protokol CoAP dan MQTT-SN. Dari hasil penelitian yang didapat, protokol MQTT-SN memiliki potensi untuk dapat memenuhi kebutuhan penggunaan daya yang rendah dan dapat berjalan dalam *bandwidth* yang kecil dibandingkan dengan protokol CoAP. Dengan pengujian dengan skema penambahan jumlah sensor, dalam pengujianya, protokol MQTT-SN memiliki *average delay* secara keseluruhan sebesar 0.09386 detik, rata-rata *throughput* sebesar 8.45013 Kbps, besar rata-rata *packet received* sebesar 35.79073, dan rata-rata *power consumption* sebesar 7.66233 mW.

**Kata kunci :** 802.15.4, CoAP, MQTT-SN, Simulasi, *Wireless Sensor Network*

#### Abstract

Wireless Sensor Network is one of the most rapid developing technology in information and telecommunication technology industries. Internet of Things triggering wireless sensor technology development to be used in many sectors mainly for everyday use such a smart home system. However, there are many challenge for wireless sensor technology to be implemented for every day use.

The number of nodes that will be implemented in the network will be calculated to sum the power consumption of the network itself. Though, a power efficient network should be deployed then network lifetime is also guaranteed. The 802.15.4 communication standard now is used to develop a wireless sensor network, this standard can stand the regulation of a low power, low bandwidth and lossy communication being implemented. In this research, the superiority of 802.15.4 standards and its following protocol will be examined by simulating Telkom Akses smarthome prototype with CoAP and MQTT-SN as upper layer protocol and increasing number of nodes to compare, starts from 5,8 until 10 nodes in the simulation. The simulation is conducted using Cooja network simulator and there will be delay, throughput, packet received and power consumption analysis after the simulation run. The result shows MQTT-SN protocol is performing better than CoAP protocol to fulfill a better environment for the wireless sensor network. MQTT-SN could run in a low bandwidth environment and use less power than CoAP protocol, with a total average delay of 0.09386 second, total average throughput of 8.45013 Kbps, total average of packet received of 35.79073 Bytes, and the total average of power consumption of 7.66233 mW.

**Keywords :** 802.15.4, CoAP, MQTT-SN, Simulation, *Wireless Sensor Network*

## 1. Pendahuluan

### 1.1 Latar Belakang

Wireless Sensor Network (WSN) merupakan sebuah istilah jaringan nirkabel yang saling terkoneksi secara spasial dan perangkat yang terhubung dalam jaringan ini biasanya bekerja dan saling berkomunikasi secara otomatis berdasarkan perintah yang diberikan oleh pengguna ataupun mesin. Demi memenuhi jaringan *wireless sensor* yang dapat dibuat dengan biaya rendah dan mengeluarkan daya lebih kecil untuk kebutuhan masal, sebuah jaringan *wireless sensor* perlu didesain secara efektif dimulai dari perangkat hingga protokol yang digunakan untuk mengetahui sejauh mana jaringan *wireless sensor* yang didesain berjalan efektif.

Proses simulasi secara virtual terhadap pembuatan jaringan akan mengurangi biaya *test bed* secara signifikan, untuk memprediksi perilaku jaringan sensor dan mengetahui besar parameter yang akan dihasilkan oleh jaringan. Perilaku jaringan yang berpengaruh terhadap pertambahan jumlah sensor ataupun penggunaan protokol pada implementasi *wireless sensor* akan secara mudah diprediksi melalui proses simulasi jaringan, tanpa harus terlebih dulu mengimplementasikan jaringan menggunakan *hardware*.

Penelitian yang berjudul “Analisis Performansi Sistem Smarthome Indihome Berbasis Standar 802.15.4” ini merupakan penelitian yang dilakukan untuk memprediksi perilaku *prototype* jaringan sistem smarthome milik PT Telkom Akses. Penelitian dilakukan dengan membandingkan dua protokol OSI bagian atas yang digunakan dalam sebuah sistem IoT yaitu protokol CoAP dan MQTT-SN. Hasil pengujian dengan cara simulasi virtual dari penelitian ini digunakan untuk menentukan jaringan dengan protokol mana yang lebih efisien untuk diimplementasikan ke dalam sebuah sistem nyata.

### 1.2 Tujuan penelitian

Tujuan dari penelitian yang dilakukan adalah :

1. Merancang sistem yang sesuai dengan *prototype* jaringan smarthome milik PT Telkom Akses.
2. Mengetahui dan membandingkan performa dari sistem dengan protokol CoAP dan MQTT-SN
3. Menganalisis performansi sistem dengan parameter ukur *delay*, *throughput*, *packet received* dan *power consumption*.

### 1.3 Manfaat penelitian

Manfaat dari penelitian ini adalah untuk menentukan jaringan dengan protokol IoT mana yang lebih efisien

### 1.4 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan sebelumnya maka dapat dirumuskan beberapa masalah berikut:

1. Bagaimana cara menguji performansi model jaringan *wireless sensor* dengan cara simulasi?
2. Mengetahui perbandingan performansi dari protokol CoAP dan MQTT-SN dengan mengukur parameter ukur *delay*, *packet received*, *throughput* dan *power consumption* dengan simulasi.
3. Menentukan protokol mana yang lebih baik jika ditinjau dari prasyarat *low cost*, dan *low energy* dari *wireless sensor network*.

### 1.5 Batasan Masalah

Demi menghindari perluasan pembahasan pada penelitian ini, berikut adalah batasan masalah untuk penelitian ini:

1. Simulasi dilakukan dengan platform simulator jaringan Cooja network simulator.
2. Mengabaikan segala elemen jaringan yang tidak dibutuhkan oleh sensor.
3. Simulasi dijalankan menggunakan protokol 6LowPAN berbasis standar 802.15.4, dengan protokol RPL, protokol UDP, dan protokol CoAP atau MQTT-SN pada tiap model jaringan.
4. Perangkat yang digunakan adalah Zolertia atau Z1 dengan sensor suhu TMP102, dan switch sensor.
5. Analisis dilakukan berdasarkan model jaringan dengan jumlah sensor yang ditentukan.
6. Simulasi dijalankan diatas sebuah platform virtualisasi.

## 2. Landasan Teori

### 2.1 Standar 802.15.4

Standar 802.15.4 merupakan salah satu pengembangan dari standar 802 milik IEEE. Tujuan dibuatnya standar 802.15.4 adalah untuk merealisasikan komunikasi antar device sederhana dan dapat memenuhi ekspektasi *lifetime* yang panjang pada device tersebut. Salah satu fitur kunci dari standar ini untuk memenuhi tujuan diatas adalah kemampuan device dengan standar 802.15.4 untuk masuk kedalam sebuah kondisi *sleeping* dimana dalam kondisi ini device akan mengkonsumsi daya yang sangat kecil, dan untuk meningkatkan efisiensi setiap device

802.15.4 memiliki sebuah metode deteksi energi untuk mengurangi kemungkinan scanning terhadap seluruh keadaan network sehingga dapat menghemat daya, hal ini diimplementasikan dengan adanya sebuah Link Quality Indication (LQI) yang memberikan pengukuran metrik terhadap kuat sinyal antar device. Seperti pada standar 802 lainnya, standar 802.15.4 tidak menspesifikasikan standar tertentu pada layer bagian atas dari layer OSI.

Dalam standar 802.15.4 terdapat 2 jenis device fisik yang terbagi atas dua fungsi yaitu: Full Function Device (FFD) dan Reduced Function Device (RFD). FFD merupakan perangkat yang dapat berkomunikasi dengan RFD, dimana RFD merupakan perangkat yang hanya dapat berkomunikasi dengan satu FFD namun tidak bias berkomunikasi dengan RFD lainnya. Salah satu contoh dari sebuah RFD adalah switch lampu, dan sensor.

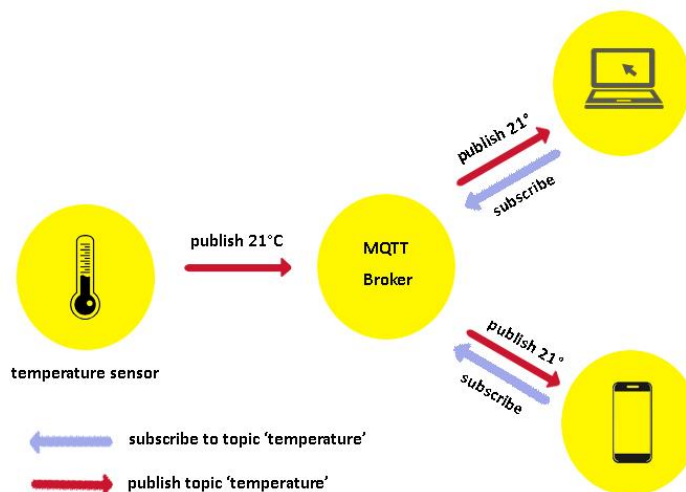
Terdapat 3 jenis perangkat dalam jaringan standar 802.15.4 ketiganya adalah : PAN coordinator, router dan end device. Sebuah PAN coordinator adalah sebuah FFD yang bertanggung jawab atas alokasi alamat dan forwarding dalam jaringan tersebut. Sedangkan router merupakan sebuah FFD yang melakukan tugas routing terhadap paket data dalam jaringan tersebut. End Device merupakan perangkat yang hanya bertanggung jawab atas kerja perangkat itu sendiri dan komunikasi data dengan perangkat yang terhubung dengan end device tersebut.[1]

## 2.2 Protokol MQTT

Merupakan singkatan dari Message Queueing Telemetry Transport, MQTT adalah sebuah protokol komunikasi yang didesain pada tahun 1999 untuk komunikasi satelit untuk komunikasi data yang dapat memenuhi kondisi jaringan dengan delay besar dengan penggunaan bandwidth yang sangat kecil. Dikemudian hari protokol tersebut digunakan untuk pengaplikasian Machine to Machine Communication atau biasa disebut M2M, ataupun pada perangkat IoT. Protokol MQTT bekerja pada layer 5 – 7 pada layer OSI, berikut adalah ilustrasi dari stack layer OSI.[2]

Ada beberapa konsep yang perlu kita mengerti untuk mengerti bagaimana protokol MQTT bekerja.

1. Publish/subscribe : Dalam protokol MQTT, publish merupakan kegiatan dari nodes yang biasanya berupa sensor dalam memberikan informasi yang dibutuhkan oleh subscriber. Subscriber men-subscribe terhadap topik yang subscriber butuhkan dan akan menerima tiap informasi yang di publish oleh terhadap topik tersebut. Disisi lain setiap client dapat mem-publish –tiap topik yang ia ingin publish agar semua subscriber dapat mengakses informasi terhadap topik tersebut.
2. Topics dan subscriptions : Dalam MQTT, publisher melakukan publish terhadap topik yang dianggap dapat menjadi sebuah informasi, sebagai contoh sebuah subscriber ingin mendapatkan informasi dari sensor temperatur yang ada. Dalam skenario ini sensor merupakan publisher yang akan mengumpulkan data dari hardware sensor dan memberikan informasi dari sensor temperatur tersebut kepada subscriber. Subscriber dapat memilih dan menentukan topik apa saja yang ia ingin subscribe.
3. Quality of Service : untuk menjaga reliability protokol MQTT menggunakan protokol transport TCP. MQTT juga memiliki tiga tingkat QoS yang akan dijabarkan sebagai berikut:
  - a) QoS0 : Pada tingkat QoS ini, pesan dalam jaringan MQTT akan dikirim hanya sekali dan tidak menjamin terkirimnya pesan tersebut
  - b) QoS1 : Pada tingkat QoS ini, pesan dalam jaringan MQTT akan dikirim minimal satu kali sehingga memungkinkan untuk dilakukan pengiriman ulang terhadap paket.
  - c) QoS2 : Pada tingkat QoS ini, pesan dalam jaringan MQTT akan dikirim satu kali namun dalam QoS2 dilakukan metode 4-Way Handshaking dalam komunikasinya.



### Gambar 2.1 Ilustrasi Publish/Subscribe pada protokol MQTT

Secara umum arsitektur MQTT dibagi menjadi 2 komponen yang dapat kita lihat pada Gambar 2.5. Berikut adalah penjelasan dari gambar tersebut :

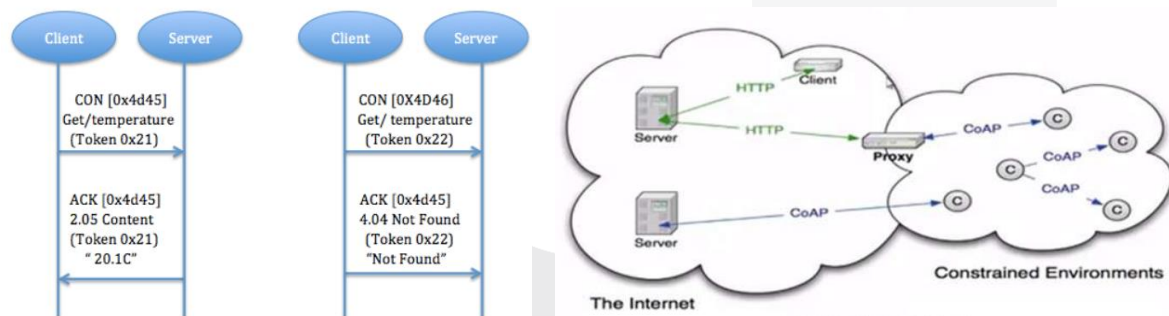
1. Client : Client dapat berperan sebagai publisher atau subscriber dalam jaringan, dan selalu membuat hubungan dalam jaringan ke Server (Broker). Client dapat melakukan hal sebagai berikut :
  - Melakukan publish pesan terhadap subscriber yang membutuhkan
  - Men-subscribe terhadap subjek topik yang ingin diterima
  - Melakukan unsubscribe
  - Melepaskan hubungan dengan broker
2. Broker : Broker melakukan kontrol dan distribusi dari informasi jaringan dan bertanggung jawab terhadap diterimanya pesan yang di publish oleh publisher, melakukan fungsi logik terhadap metode subscribe/publish itu sendiri. Broker dapat melakukan hal sebagai berikut :
  - Menerima request dari client
  - Menerima pesan yang di publish oleh user
  - Memproses request seperti subscribe dan unsubscribe dari user.
  - Melakukan penerusan pesan dari publisher terhadap user yang membutuhkan

### 2.3 MQTT-SN

MQTT-SN merupakan varian protokol MQTT yang didesain untuk Sensor Nodes (SN). Protokol MQTT-SN difokuskan untuk hardware dengan spesifikasi terbatas. Tidak seperti MQTT, MQTT-SN menggunakan UDP sebagai protokol komunikasi datanya. Kelebihan dari penggunaan UDP ini akan menghilangkan proses handshake yang ada pada TCP sehingga komunikasi bersifat connectionless dan lebih ringan.[3]

### 2.4 CoAP

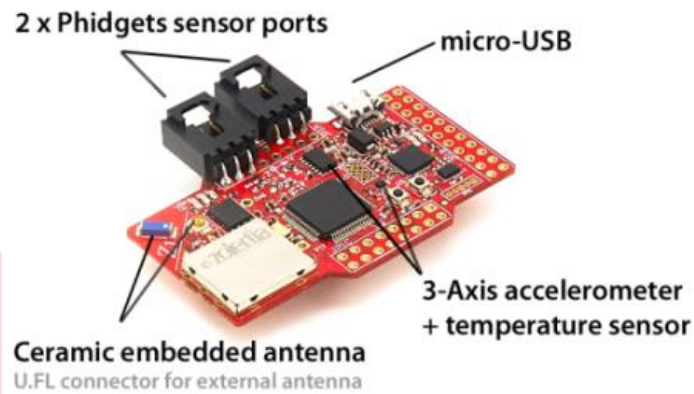
CoAP merupakan protokol pada IoT yang merupakan singkatan dari Constrained Application Protocol. Protokol ini memungkinkan pengaplikasian cara kerja mirip dengan protokol HTTP untuk melakukan fungsi kontrol terhadap perangkat IoT yang memiliki keterbatasan resource hardware pada perangkatnya. Tidak seperti protokol yang berbasis HTTP, CoAP beroperasi menggunakan UDP. Sama dengan MQTT, CoAP juga bekerja pada layer bagian atas pada layer OSI yaitu layer 5 hingga layer 7.[3]



Gambar 2.2 Message layer dan model jaringan CoAP

### 2.5 Zolertia

Zolertia atau disingkat dengan Z1 merupakan sebuah development platform untuk wireless sensor network. Platform ini dilengkapi dengan MSP430F2617 low power microcontroller, dengan 8KB RAM, 92KB Flash Memory, dan transceiver CC2420, dan mendukung standar wireless 802.15.4. Zolertia juga dilengkapi dengan sensor digital bawaan seperti sensor accelerometer digital (ADXL345) dan sensor temperature digital (TMP102) yang sudah tertanam pada mainboardnya.[4]



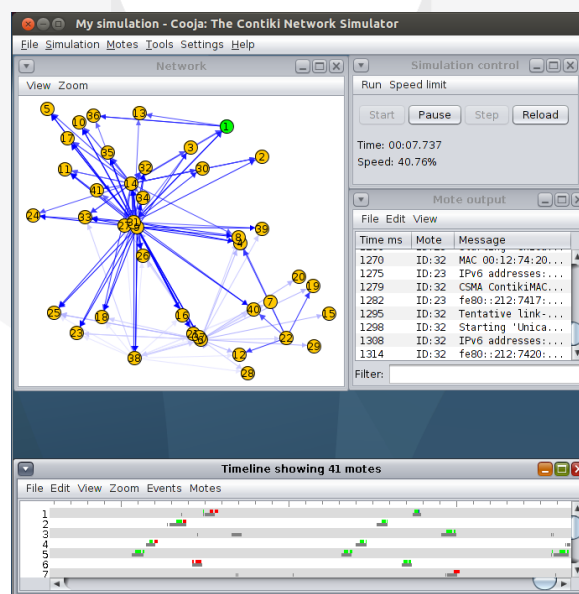
Gambar 2.3 Mainboard dari Zolertia

## 2.6 Cooja

Cooja merupakan network simulator dari sistem operasi embedded device Contiki, dimana network simulator ini memiliki kemampuan meniru kondisi asli dari hardware perangkat dalam jaringan dan mensimulasikannya dengan berbagai jenis protokol. Cooja mendukung beberapa simulasi dari berbagai standar komunikasi diantaranya 1100, TI CC2420, Contiki-RPL, dan IEEE 802.15.4.

Dalam simulator cooja, ada empat jenis model propagasi ketika akan menjalankan simulasi. Salah satu model dari keempat model propagasi dalam cooja adalah model propagasi constant loss unit Disk Graph Medium (UDGM). Penggunaan model propagasi UDGM akan membuat nodes hanya dapat berkomunikasi pada jarak komunikasi ideal nodes tersebut, jika ada nodes yang berada diluar jarak komunikasi dengan nodes lainnya, maka nodes tersebut dapat tidak akan menerima paket data sama sekali didalam simulasi.

Cooja memiliki empat user interface utama secara default. Network window menampilkan susunan dari network yang kita buat. Untuk membuat sebuah topologi, kita dapat secara praktis menggunakan network window ini untuk memindahkan posisi nodes yang kita akan buat. Pada network window kita juga dapat menampilkan radius transmisi dari nodes, ip address, radio traffic dan memberi perbedaan jenis nodes pada network window. Sebagai contoh fitur network window kita membedakan tipe nodes berdasarkan warna. Nodes dengan warna hijau ada sink sedangkan nodes berwarna kuning adalah nodes pengirim. User interface selanjutnya adalah simulation control. Simulation control berfungsi untuk mengontrol jalannya simulasi, memulai, meberhentikan, reload, dan mempercepat simulasi dapat dilakukan pada window simulation control. User interface selanjutnya adalah timeline. Pada window timeline kita diberikan sajian visual trafik jaringan, dan adanya konsumsi daya yang terjadi dalam simulasi. Mengikuti penomoran masing masing nodes pada window timeline, jika nodes memperlihatkan garis linear abu abu atau putih mengartikan bahwa nodes tersebut idle, biru merupakan transmisi data, hijau penerimaan paket data, dan merah menandakan adanya interferensi atau gagalnya paket data selesai ditransmisikan.[5]



Gambar 2.4 Interface dari Cooja Network Simulator

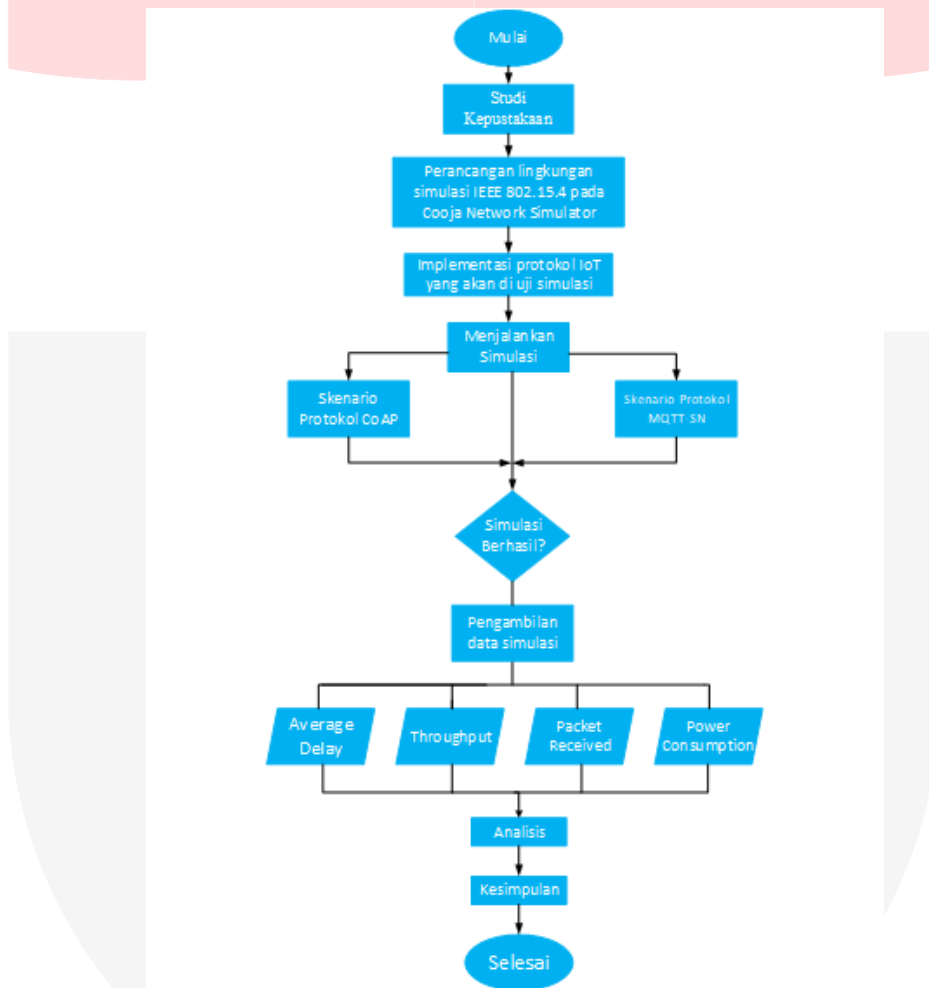
### 3. Perancangan Sistem

#### 3.1 Skenario Perancangan dan Pengujian Sistem

Uji performa pada protokol CoAP dan MQTT-SN dilakukan dengan Cooja network simulator dan model jaringan akan mengadopsi model jaringan smart home dari Indihome milik PT Telkom Akses. Sebagai representasi jaringan dalam penelitian kali ini, jumlah sensor yang akan digunakan dalam simulasi akan dikondisikan seperti layaknya pihak Indihome akan implementasikan pada program smarthome-nya yaitu 5 hingga 10 buah sensor, dengan 1 buah border-router yaitu router yang menjadi Gateway sensor-sensor dengan internet, dan semua perangkat dalam proses simulasi berkomunikasi dibawah standar 802.15.4 dan menggunakan sistem pengalamatan IPv6.

Perangkat yang disebutkan sebelumnya nantinya akan disimulasikan dengan Cooja network simulator dan cuplikan pengiriman paket dalam jaringan akan diolah dengan wireshark. Setelah dilakukan pengolahan dengan wireshark kedua protokol akan dibandingkan nilai QoS-nya. Parameter QoS yang dianalisa antara lain adalah average delay, received packet, throughput dan power consumption.

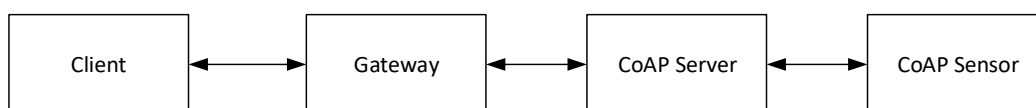
#### 3.2 Diagram Alir Sistem



Gambar 3.1 Diagram Alir Sistem

#### 3.3 Desain dan Langkah Pengujian Jaringan Protokol CoAP

Model dari simulasi jaringan protokol CoAP dapat dilihat pada gambar 3.2. Client merupakan sebuah entitas komputer virtual dengan sistem operasi Ubuntu yang juga terhubung dengan jaringan virtual yang berisi sensor dan juga server CoAP. Dalam simulasi ini client akan memberi perintah request informasi sensor yang diinginkan CoAP server kemudian akan ada feedback dari sensor berupa pesan teks pesan.

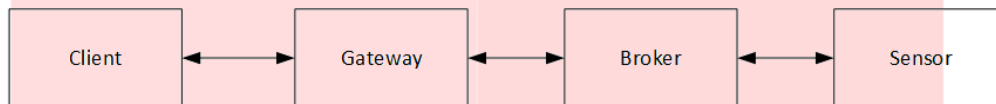


Gambar 3.2 Model jaringan pengujian protokol CoAP

Setiap paket data yang dikirimkan dari sensor menuju sensor atau CoAP server yang ada akan direkam selama 300 detik yang nantinya akan menjadi data untuk diolah menggunakan wireshark dan dan Microsoft Excel. Model jaringan dalam simulasi juga akan disesuaikan berdasarkan jarak yang sesuai dengan implementasi smarthome, sehingga jarak antar perangkat dalam simulasi kali ini tidak akan lebih dari 10 meter.

### 3.4 Desain dan Langkah Pengujian Jaringan Protokol MQTT-SN

Pada simulasi jaringan MQTT-SN, client merupakan subscriber dan sensor merupakan publisher dari sistem MQTT-SN. Sama seperti pada sistem CoAP client merupakan entitas komputer virtual yang berjalan dengan sistem operasi Ubuntu, dan melakukan fungsi subscribe melalui terminal ubuntu kepada broker dan sensor. Opsi QoS yang digunakan pada pengujian MQTT-SN adalah tipe QoS 0.



Gambar 3.3 Model jaringan penujian protokol MQTT-SN

Sama dengan proses simulasi pada CoAP, pada simulasi MQTT-SN Setiap paket data yang dikirimkan dari sensor menuju sensor atau broker yang ada akan direkam selama 300 detik yang nantinya akan menjadi data untuk diolah menggunakan wireshark dan Microsoft Excel. Model jaringan dalam simulasi juga akan disesuaikan berdasarkan jarak yang sesuai dengan implementasi smarthome, sehingga jarak antar perangkat dalam simulasi kali ini tidak akan lebih dari 10 meter.

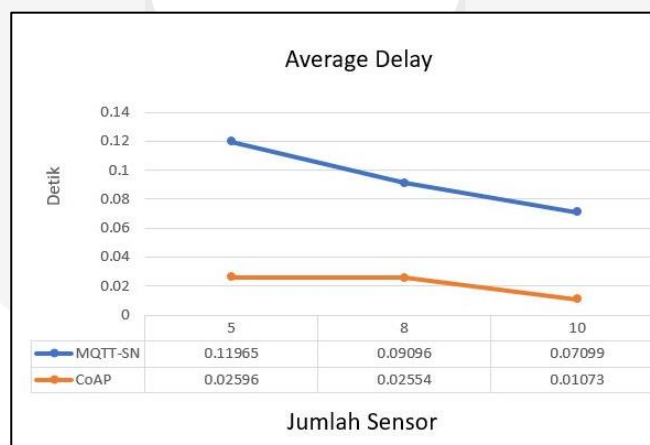
Tabel 3.1 Skenario pengujian keseluruhan

Skenario	Protokol	Jumlah Sensor	Durasi
1	CoAP	5	300 detik
2	CoAP	8	300 detik
3	CoAP	10	300 detik
4	MQTT	5	300 detik
5	MQTT	8	300 detik
6	MQTT	10	300 detik

## 4. Analisis Hasil Simulasi

### 4.1 Pengujian Average Delay

Pengujian average delay dalam penelitian kali ini adalah melakukan perhitungan terhadap rata-rata jeda antara pengiriman satu paket dengan paket selanjutnya yang dilakukan oleh sensor dalam proses transmisi pada percobaan simulasi.



Gambar 4.1 Hasil pengujian average delay

Pada gambar 4.1 dapat dilihat bahwa protokol CoAP lebih unggul dibandingkan dengan protokol MQTT-SN jika ditinjau dari hasil average delay terhadap penambahan jumlah sensor. Pada protokol CoAP nilai average delay tertinggi didapat pada saat sensor berjumlah 5 dengan nilai average delay 0.2596 detik dan nilai average delay terendah berada pada jumlah sensor berjumlah 10 dengan nilai average delay 0.01073 detik, dengan average delay keseluruhan sebesar 0.02074 detik.

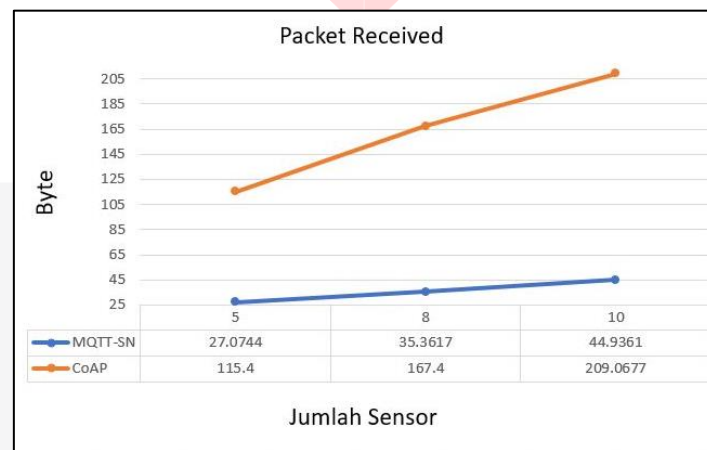
Pada gambar 4.1 juga dapat dilihat bahwa nilai average delay tertinggi pada protokol MQTT-SN didapat pada saat sensor berjumlah 5 dengan nilai average delay sebesar 0.11965 detik, dan nilai average delay terendah didapat pada sensor berjumlah 10 dengan average delay sebesar 0.07099 detik, dengan average delay keseluruhan sebesar 0.09386 detik.

Dari gambar 4.1 dapat dilihat bahwa jika jumlah sensor semakin banyak maka nilai average delay akan semakin rendah. Hal tersebut disebabkan oleh mekanisme routing DODAG yang memungkinkan data ditransmisikan dari penerima ke penerima melalui berbagai jalur dalam jaringan yang idle sehingga data tidak menumpuk pada satu jalur dalam jaringan.

Jika ditinjau dari cara kerja protokol, protokol CoAP juga unggul dibandingkan dengan protokol MQTT-SN dikarenakan proses registrasi yang dilakukan oleh publisher terhadap broker yang dilakukan dalam protokol MQTT-SN sehingga menghasilkan waktu yang lebih lama untuk melakukan transmisi data.

#### 4.2 Pengujian Received Packet

Pengujian received packet dilakukan untuk mengetahui berapa besar paket data yang diterima dalam proses transmisi dari semua data yang ditransmisikan dari sensor ke broker ataupun CoAP server dalam proses simulasi. Perhitungan dilakukan dengan membagi jumlah paket yang diterima dengan besar paket rata-rata paket data yang dikirimkan. Besaran received packet dalam penelitian kali ini dinyatakan dalam byte.



**Gambar 4.2 Hasil pengujian total packet received**

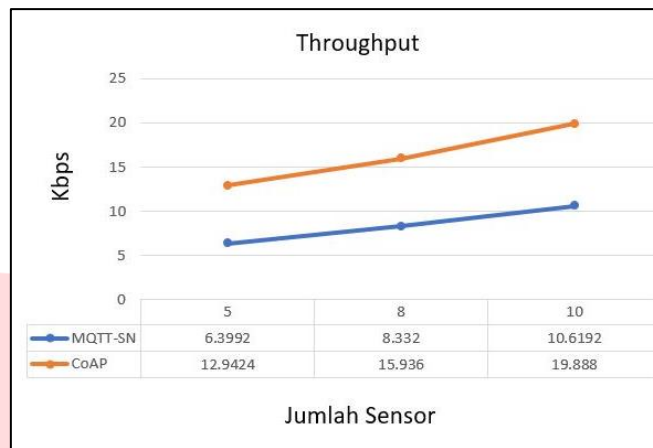
Pada gambar 4.2 dapat dilihat bahwa protokol CoAP mengirim total paket data paling banyak saat sensor berjumlah 10 yaitu sebesar 209.0677 byte dan paling sedikit pada saat sensor berjumlah 5 yaitu sebesar 115.4 byte, dan protokol CoAP menghasilkan besar packet received secara keseluruhan sebesar 163.9559 byte.

Sedangkan protokol MQTT-SN berhasil mengirim total besar paket lebih sedikit dibandingkan dengan protokol CoAP dengan hasil paket dikirimkan paling banyak pada saat sensor berjumlah 10 yaitu sebesar 44.9361 Byte dan paling sedikit saat sensor berjumlah 5 buah yaitu sebesar 27.0744 byte, dan protokol MQTT-SN menghasilkan besar packet received secara keseluruhan sebesar 35.79073 byte.

#### 4.3 Pengujian Throughput

Pengujian throughput merupakan pengujian untuk mengetahui seberapa besar bandwidth yang dibutuhkan oleh jaringan. Besar dari throughput dapat dijadikan acuan dalam merancang kebutuhan bandwidth pada jaringan agar tidak terjadi bottleneck dalam jaringan.





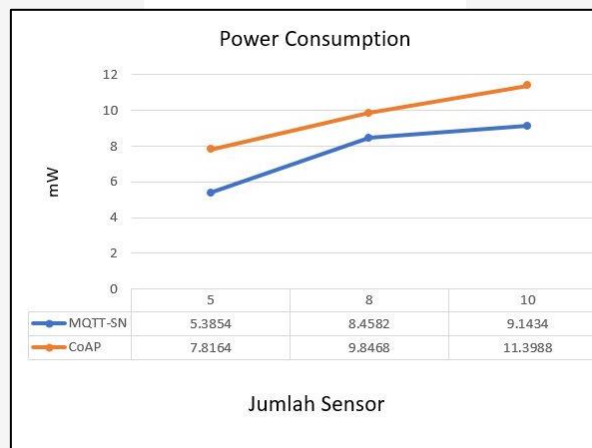
**Gambar 4.3 Hasil pengujian throughput**

Pada protokol CoAP nilai throughput tertinggi didapat pada saat sensor berjumlah 10 dengan besar throughput 19.888 Kbps dan besar throughput terendah didapat pada skenario sensor berjumlah 5 dengan besar throughput 12.9424 Kbps, dan protokol CoAP menghasilkan throughput secara keseluruhan sebesar 16.25546 Kbps.

Pada gambar 4.3 juga dapat dilihat bahwa nilai throughput tertinggi pada protokol MQTT-SN didapat pada saat sensor berjumlah 10 dengan besar throughput 10.6192 Kbps dan throughput terendah didapat pada sensor berjumlah 5 dengan besar throughput 6.3992 Kbps, dan protokol MQTT-SN menghasilkan throughput secara keseluruhan sebesar 8.45013 Kbps.

**4.4 Pengujian Power Consumption**

Pengujian power consumption dilakukan untuk mengukur seberapa besar daya yang digunakan oleh kedua protokol dan sensor dalam proses simulasi kemudian membandingkannya.



**Gambar 4.4 Hasil pengujian power consumption**

Pada gambar 4.4 dapat dilihat bahwa protokol CoAP menghasilkan keluaran daya paling tinggi pada saat sensor berjumlah 10 dengan besar daya 11.3988 mW dan besar daya terendah didapat pada skenario sensor berjumlah 5 dengan besar daya sebesar 7.8164 mW. Hasil power consumption secara keseluruhan protokol CoAP sebesar 9.68733 mW.

Protokol MQTT-SN menghasilkan keluaran daya lebih rendah dibandingkan dengan protokol CoAP dengan keluaran besar daya terendah didapat pada saat sensor berjumlah 5 dengan besar daya yang gunakan 5.3854 mW dan besar keluaran daya terbesar didapat pada sensor berjumlah 10 dengan besar daya yang digunakan 9.1434 mW. Hasil power consumption secara keseluruhan protokol MQTT-SN sebesar 7.6623 mW.

## 5. Kesimpulan

Berdasarkan hasil pengujian dan analisis data dari penelitian ini, maka dapat ditarik kesimpulan sebagai berikut:

1. Melakukan simulasi secara virtual terhadap suatu sistem jaringan, akan membantu seorang perancang jaringan untuk menentukan bagaimana jaringan akan diimplementasikan, dan memperkecil cost dari test bed untuk implementasi jaringan itu sendiri.
2. Walaupun unggul dalam mengatasi besarnya delay. Parameter throughput, packet received, dan power consumption yang terukur pada protokol CoAP menghasilkan angka yang lebih besar jika dibandingkan dengan protokol MQTT-SN. Jika dinilai secara menyeluruh, protokol MQTT-SN lebih baik untuk diimplementasikan dalam jaringan wireless sensor.
3. Mengacu pada poin 1, protokol MQTT-SN akan cocok diterapkan pada perangkat yang berspesifikasi lebih rendah dibandingkan Zolertia yang digunakan secara virtual pada simulasi kali ini, dengan kata lain MQTT-SN dapat beroperasi diperangkat yang lebih murah dan mengurangi cost dari sistem.

## 6. Daftar Pustaka

- [1] J. T. Adams, "An Introduction to IEEE STD 802.15.4," *2006 IEEE Aerosp. Conf.*, pp. 1–8, 2006.
- [2] D. Soni and A. Makwana, "A Survey on MQTT: A Protocol of Internet of Things(IoT)," *Int. Conf. Telecommun. Power Anal. Comput. Tech. (Ictpac - 2017)*, no. April, pp. 0–5, 2017.
- [3] M. H. Amaran, N. A. M. Noh, M. S. Rohmad, and H. Hashim, "A Comparison of Lightweight Communication Protocols in Robotic Applications," *Procedia Comput. Sci.*, vol. 76, no. Iris, pp. 400–405, 2015.
- [4] Zolertia, "Z1 Datasheet," pp. 1–20, 2010.
- [5] E. R. Afroditi Argiropoulou, "CoAP-enabled Sensors for the Internet-of-Things," *CoAP-enabled Sensors for the Internet-of-Things*, 2014.