

**ANALISA METODE *PUBLISH/SUBSCRIBE* UNTUK
KOMUNIKASI DATA ANTAR PERANGKAT DALAM
LINGKUNGAN *SMART HOME***

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

M Fikri Nusantara

NIM: 105090600111044

UNIVERSITAS BRAWIJAYA



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

JUDUL SKRIPSI

ANALISA METODE *PUBLISH/SUBSCRIBE* UNTUK KOMUNIKASI DATA ANTAR PERANGKAT DALAM LINGKUNGAN *SMART HOME*

Diajukan untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer

Disusun Oleh :
M Fikri Nusantara
NIM: 105090600111044

Skripsi ini telah diuji dan dinyatakan lulus pada
4 Agustus 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Sabriansyah Rizqika Akbar, S.T, M.Eng
NIP: 19820809 201212 1 004

Aditya Rachmadi, S.ST., M.TI
NIK: 201201 860421 1 001

Mengetahui
Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 17 Agustus 2016



M Fikri Nusantara

NIM: 105090600111044

KATA PENGANTAR

Puji Syukur penulis panjatkan kehadirat Allah SWT karena berkat Rahmat dan Karunia-Nya penulis dapat menyelesaikan penyusunan skripsi ini. Shalawat beserta salam semoga senantiasa tercurahkan kepada Nabi Muhammad SAW , kepada keluarganya, para sahabatnya hingga kepada umatnya hingga akhir zaman.

Penulisan skripsi ini diajukan untuk memenuhi salah satu syarat untuk memperoleh gelar sarjana pada Program Studi Informatika Fakultas Ilmu Komputer Universitas Brawijaya. Judul yang penulis ajukan adalah “ Analisa Metode *Publish/Subscriber* untuk Komunikasi antar Perangkat Dalam Lingkungan *Smart Home*”. Dalam penyusunan dan penulisan skripsi ini tidak terlepas dari bantuan, bimbingan serta dukungan dari berbagai pihak. Oleh karena itu dalam kesempatan ini penulis ingin mengucapkan rasa terima kasih kepada pihak yaitu:

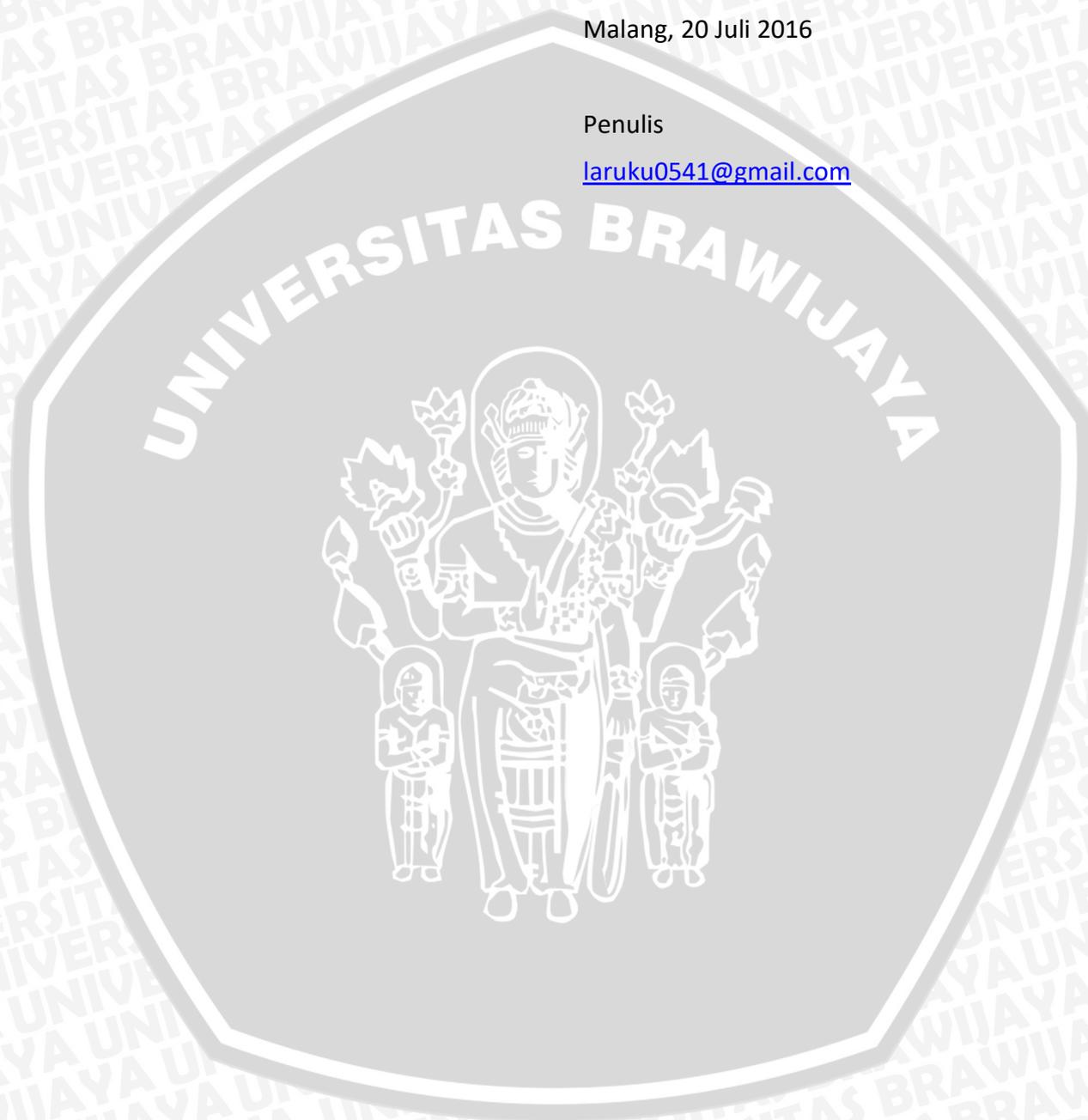
1. Yang terhormat bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku dekan Fakultas Ilmu Komputer Universitas Brawijaya
2. Yang terhormat bapak Agus Wahyu Widodo, S.T, M.Cs selaku Kepala Program Studi Informatika yang telah menyetujui secara resmi judul skripsi yang diajukan oleh penulis
3. Yang terhormat bapak Sabriansyah Rizqika Akbar, S.T, M.Eng selaku pembimbing I yang selalu dengan senang hati memberikan bimbingan dan nasehat kepada penulis agar lancar dalam melakukan penelitian skripsi
4. Yang terhormat bapak Aditya Rachmadi, S.ST., M.TI selaku pembimbing II yang selalu membantu penulis dengan memberikan penilaian terhadap tulisan yang ditulis oleh penulis agar sesuai dengan tata cara yang di berlakukan
5. Kepada kedua orang tua penulis atas jasa-jasanya , kesabaran, doa, dan tidak pernah lelah dan mendidik dan mendukung penulis sejak kecil
6. Kepada Andrias Yudianto Putra, Wahyu Sasongko Jati, Maskur Purwiadi, Panser Brigade Muhammad, Muchlas Mughniy yang sudah menjadi teman seperjuangan mahasiswa angkatan akhir dan berkulat dengan skripsi
7. Kepada Dita Oktaria yang dimana saya bersyukur sudah bisa mengenalnya dan bisa sempat dekat dengannya
8. Teman-teman dan saudara seperjuangan lainnya yang telah memberi dukungan moril dan semangat kepada penulis agar bisa menyelesaikan tugas akhir skripsi ini
9. Semua pihak yang telah membantu penulis dalam menyelesaikan tugas akhir skripsi ini

Semoga Allah SWT memberikan balasan yang berlipat ganda kepada semuanya. Demi perbaikan selanjutnya saran dan kritik yang membangun akan diterima dengan senang hati oleh penulis. Akhirnya, hanya kepada Allah SWT penulis serahkan segalanya mudah-mudahan skripsi ini dapat bermanfaat bagi penulis dan bagi orang lain yang membaca skripsi ini, terima kasih.

Malang, 20 Juli 2016

Penulis

laruku0541@gmail.com



ABSTRAK

Komunikasi data sudah banyak di gunakan di berbagai bidang bahkan ada beberapa metode yang bisa digunakan untuk melakukan komunikasi data antar perangkat. Salah satunya dari metode komunikasi data adalah metode *Publish/Subscribe* yang dimana metode tersebut termasuk tipe MOM (*Message Oriented Middleware*). Pada penelitian ini membahas kinerja metode *Publish/Subscriber* untuk komunikasi data antar perangkat di lingkungan *Smart Home*. Tujuan penelitian ini adalah mengetahui kinerja pengiriman data menggunakan metode *Publish/Subscribe* dari sisi *publisher* ke *broker*, *broker* ke *subscriber* dan *publisher* ke *subscriber*. Penelitian ini menggunakan metode penelitian dengan cara membangun simulasi komunikasi data menggunakan metode *Publish/Subscribe* dan pengujian akan dilakukan dengan menggunakan dua protokol yang digunakan dalam metode ini yaitu protokol MQTT (*Message Queue Telemetry Transport*) dan protokol AMQP (*Advanced Message Queue Protocol*). Pengumpulan data dilakukan dengan cara merekam hasil waktu pengiriman data dari setiap pengujian yang dilakukan. Hasil penelitian utama yang didapat adalah waktu pengiriman data dari sisi *publisher* ke *broker*, *broker* ke *subscriber* dan *publisher* ke *subscriber* pada kedua protokol yang sudah disebutkan sebelumnya. Dari hasil penelitian dapat disimpulkan waktu pengiriman data menggunakan protokol AMQP memiliki waktu pengiriman hampir sama untuk setiap tahap pengiriman dan waktu pengiriman data menggunakan protokol MQTT akan mengalami penurunan seiring dengan banyak data yang dikirimkan.

Kata kunci: *Publish/Subscribe*, MQTT, AMQP, *Smart Home*, *Publisher*, *Subscriber*

ABSTRACT

Data communication has been widely used in many areas, there are even some methods that can be used to perform data communication between devices. One of these methods is Publish/Subscribe method which the method is categorized as MOM (Message Oriented Middleware). In this research, the researcher discusses the performance of Publish/Subscribe method for data communication between devices in the Smart Home. The purpose of this research is to determine the performance of the data transmission using the Publish/Subscribe method, from the publisher to the broker, the broker to the subscriber and the publisher to the subscriber. The research method is done by building the data communication simulation using Publish/Subscribe method and the test will be done by using two protocols that is used in this method namely MQTT protocol (Message Queue Telemetry Transport) and AMQP (Advanced Message Queue Protocol). The data collection is done by recording the data transmission time of each test that was performed. The research results obtained is the data transmission time from the publisher to the broker, the broker to the subscriber and the publisher to subscriber at the two protocols that have been mentioned previously. From the research result it can be concluded that the data transmission time using AMQP protocol almos got same transmisson time in every data transfer phase and the data transmission time using MQTT protocol will decline along with the number of the data.

Keywords: *Publish / Subscribe, MQTT, AMQP, Smart Home, Publisher, Subscriber*

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan masalah	2
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	4
2.1 Kajian Pustaka	4
2.2 Dasar Teori.....	4
2.2.1 Smart Home	4
2.2.2 <i>Publish/Subscribe</i>	5
2.2.3 <i>Komponen Sistem Publish/Subscribe</i>	6
2.2.4 <i>Model Subscription</i>	11
2.2.5 <i>AMQP (Advanced Message Queueing Protocol)</i>	12
2.2.6 <i>MQTT (Message Queueing Telemetry Transport)</i>	13
2.2.7 <i>Virtual Private Server (VPS)</i>	14
BAB 3 METODOLOGI	17
3.1 Metode Penelitian	17

3.1.1 Studi Literatur	17
3.1.2 Perancangan Simulasi	18
3.1.3 Implementasi Metode ke Simulasi Komunikasi Data	21
3.1.4 Pengujian Simulasi Komunikasi Data	21
3.1.5 Pengambilan Kesimpulan	23
BAB 4 HASIL dan pembahasan	24
4.1 Pengujian dan Analisa	24
4.2 Sistematika Kinerja <i>Publish/Subscribe</i> dengan protokol AMQP dan MQTT	24
4.2.1 Kinerja <i>Publish/Subscribe</i> dengan protokol MQTT	24
4.2.2 Kinerja <i>Publish/Subscribe</i> dengan protokol AMQP	26
4.3 Skenario Pengujian	28
4.3.1 Hasil Pengujian Menggunakan RabbitMQ (AMQP)	29
4.3.2 Hasil Pengujian Menggunakan Mosquitto (MQTT).....	32
BAB 5 Penutup	35
5.1 Kesimpulan.....	35
5.2 Saran	36
LAMPIRAN A Lampiran SKRIPSI.....	39



DAFTAR TABEL

Tabel 2.1 Ringkasan dari <i>decoupling properties</i>	10
Tabel 3.1 Struktur Data Meteran Listrik.....	19
Tabel 3.2 Struktur Data Alarm Kebakaran	20
Tabel 3.3 Struktur Data Meteran Air.....	20
Tabel 3.4 Struktur Data Alarm Pencuri	20
Tabel 4.1 Tabel waktu pengiriman data dari sisi <i>publisher</i> ke <i>broker</i>	29
Tabel 4.2 Tabel waktu pengiriman data dari <i>Broker</i> ke <i>Subscriber</i>	30
Tabel 4.3 Tabel waktu pengiriman data dari <i>Publisher</i> ke <i>Subscriber</i>	31
Tabel 4.4 Tabel waktu pengiriman data dari sisi <i>Publisher</i> ke <i>Broker</i>	32
Tabel 4.5 Tabel waktu pengiriman data dari <i>Broker</i> ke <i>Subscriber</i>	33
Tabel 4.6 Tabel waktu pengiriman data dari <i>Publisher</i> ke <i>Subscriber</i>	34



DAFTAR GAMBAR

Gambar 2.1 Contoh Skema <i>Smart Home</i>	5
Gambar 2.2 Model Sistem <i>Publish/Subscribe</i>	6
Gambar 2.3 Contoh <i>Publish/Subscribe Overlay Network</i>	8
Gambar 2.4 <i>Decoupling</i> sifat pada sistem <i>pub/sub</i> (a) <i>Space Decoupling</i> (b) <i>Time Decoupling</i> (c) <i>Synchronization Decoupling</i>	10
Gambar 2.5 Arsitektur AMQP	13
Gambar 2.6 Arsitektur MQTT	14
Gambar 2.7 contoh arsitektur komputer tradisional	15
Gambar 2.8 contoh arsitektur komputer yang sudah di virtualisasi	16
Gambar 3.1 Skema Tahap-Tahap Metodologi Penelitian	17
Gambar 3.2 Skema Komunikasi Data di Lingkungan <i>Smart Home</i>	18
Gambar 3.3 Skema Detail Komunikasi Data Menggunakan Metode <i>Publish Subscribe</i>	19
Gambar 3.4 Skema Simulasi Komunikasi Data	21
Gambar 3.5 Diagram Alir Pengiriman Data untuk Setiap <i>Publisher</i>	22
Gambar 4.1 <i>Capture Traffic</i> data komunikasi data <i>publish/subscribe</i>	24
Gambar 4.2 Proses koneksi <i>subscriber</i> ke <i>broker</i>	25
Gambar 4.3 Proses <i>publish</i> data	25
Gambar 4.4 Proses pengiriman data dari <i>publisher</i> ke <i>broker</i>	26
Gambar 4.5 Proses pengiriman data dari <i>broker</i> ke <i>subscriber</i>	27
Gambar 4.6 Grafik waktu pengiriman data dari <i>Publisher</i> ke <i>Broker</i>	29
Gambar 4.7 Grafik waktu pengiriman data dari <i>broker</i> ke <i>subscriber</i>	30
Gambar 4.8 Grafik waktu pengiriman data dari <i>Publisher</i> ke <i>Subscriber</i>	31
Gambar 4.9 Grafik waktu pengiriman data dari sisi <i>Publisher</i> ke <i>Broker</i>	32
Gambar 4.10 Grafik waktu pengiriman data dari sisi <i>broker</i> ke <i>subscriber</i>	33
Gambar 4.11 Grafik waktu pengiriman data dari <i>Publisher</i> ke <i>Subscriber</i>	34

DAFTAR LAMPIRAN

LAMPIRAN A LAMPIRAN SKRIPSI.....	38
A.1 Capture paket data protokol MQTT dengan Wireshark	38
A.2 <i>Capture</i> paket data protokol AMQP dengan Wireshark.....	38
A.3 Rekaman trafik di broker RabbitMQ Proses <i>publish</i>	39
A.4 Rekaman trafik di broker RabbitMQ Proses <i>subscriber</i>	39
A.5 Data final protokol MQTT.....	40
A.6 Data final protokol AMQP	44
A.7 Contoh data yang dikirim.....	49
A.8 Contoh data yang diterima.....	49



BAB 1 PENDAHULUAN

1.1 Latar belakang

Komunikasi data adalah pertukaran informasi antar perangkat melalui media transmisi tertentu seperti contohnya melalui kabel ataupun melalui sinyal radio (Forouzan, 2007). Dewasa ini teknologi komunikasi data sudah semakin berkembang semakin banyak metode yang digunakan dalam komunikasi data. Semua metode tersebut mempunyai ruang lingkup dalam penggunaannya dan keunggulan masing-masing. Salah satu contoh dari metode komunikasi data adalah metode *Publish Subscribe*.

Publish/Subscribe adalah sebuah interaksi pendistribusian informasi atau data antara *publisher (sender)* dan *subscribers (receiver)*. Dalam proses komunikasi data *publish/subscribe*, pihak *subscribers* akan menyatakan ketersediaannya dalam menerima pesan dan pihak *publisher* akan langsung mengirim pesan tersebut kepada *subscribers* yang memintanya. Dalam pengiriman pesan di *publish/subscribe*, pesan tersebut akan bersifat anonim, maksudnya *publisher* akan tetap mengirim pesan tanpa perlu mengetahui terlebih dahulu identitas *subscribarnya* ataupun ada atau tidak keberadaannya. (Tarkoma, 2012)

Metode *Publish Subscribe* ini kebanyakan digunakan dalam sistem *Smart Home* yaitu sebuah sistem dimana komunikasi data terjadi pada setiap perangkat yang termasuk dalam HAN (*Home Area Network*). Perangkat tersebut bisa berupa perangkat sensor seperti sensor listrik, sensor suhu, sensor gerak dan sensor lainnya yang dibutuhkan dalam sistem *smart home* ini. Selain perangkat sensor, perangkat lainnya yang melakukan komunikasi data dalam HAN adalah perangkat elektronik lainnya seperti PC (*Personal Computer*), *SmartPhone*, Printer, dan Telepon.

Kelebihan metode ini adalah metode ini bisa menangani distribusi data dalam skala besar, data yang akan distribusikan berdasarkan topik yang diminati oleh *subscriber* selain itu pesan yang dikirimkan bersifat anonim yang maksudnya pesan tersebut tidak perlu mengetahui identitas *end-point*-nya atau mengetahui bahwa *end-point*-nya itu ada atau tidak. Ada beberapa protokol dalam komunikasi data yang bisa menggunakan metode *publish/subscribe* ini salah satunya adalah protokol AMQP dan MQTT

AMQP adalah sebuah protokol pengiriman pesan yang kebanyakan digunakan dalam sebuah perusahaan yang bergerak dibidang komunikasi. Protokol ini di desain untuk *support* hampir disemua pengiriman pesan berbasis aplikasi distribusi dan bisnis. (Luzuriaga, et al., 2015)

MQTT adalah sebuah protokol pengiriman pesan yang ringan yang terjadi antara *machine-to-machine* dan protokol ini juga memiliki ruang lingkup *mobile network*. Prosedur pertukaran informasi yang terjadi pada protokol ini adalah berbasis sumber daya yang efisien dan juga tidak memerlukan format data yang spesifik untuk mengirimkan pesannya. (Luzuriaga, et al., 2015)

Dalam hal ini peneliti ingin melakukan analisis efisiensi dan kualitas metode *publish/subscribe* pada komunikasi data antar perangkat yang terjadi di dalam lingkungan *Smart Home*. Analisis yang akan dilakukan nanti adalah dengan cara melakukan pengiriman data dengan menggunakan dua protokol dari beberapa protokol yang bisa digunakan dalam metode *Publish/Subscriber*. Dua protokol tersebut adalah protokol MQTT dan AMQP.

1.2 Rumusan masalah

1. Bagaimana merancang simulasi untuk komunikasi data antar perangkat di lingkungan smart home?
2. Bagaimana kinerja pengiriman data dari *publisher* ke *broker*, dari *broker* ke *subscriber* dan dari *publisher* ke *subscriber* menggunakan protokol AMQP dan MQTT?

1.3 Tujuan

Tujuan penelitian ini adalah untuk mengetahui bagaimana caranya membangun sebuah simulasi komunikasi data dengan metode *publish/subscribe*. Selain itu tujuan lainnya juga untuk mengetahui kinerja metode tersebut dari sisi *publisher* ke *broker*, *broker* ke *subscriber*, dan dari sisi *publisher* ke *subscriber* dengan menggunakan dua protokol yang berbeda yaitu protokol AMQP dan protokol MQTT.

1.4 Manfaat

1. Untuk Peneliti agar bisa mengetahui sistem kerja komunikasi data menggunakan metode *Publish/Subscribe* dalam lingkungan *Smart Home*
2. Untuk Kedepannya agar simulasi ini bisa dikembangkan menjadi infrastruktur *Smart Home* yang sebenarnya

1.5 Batasan masalah

Batasan masalah dari penelitian ini adalah :

1. Pembangunan simulasi metode komunikasi data *publish/subscribe* ini akan dibatasi dengan 4 *publisher*, 1 *broker*, dan 4 *subscriber*
2. *Broker* akan diletakan di vps yang terletak di jaringan yang berbeda dengan *publisher* dan *subscriber* agar data yang didapat nanti bisa lebih bernilai
3. Analisis ini untuk menguji kualitas dan performa dari metode *publish/subscribe* ketika diterapkan kedalam sistem komunikasi data
4. Pada analisis ini tidak akan membahas perhitungan data , data yang digunakan nanti adalah data jadi yang digenerate secara random sesuai dengan tipe data perangkat yang akan diujikan nanti

5. Untuk *broker* yang digunakan adalah RabbitMQ dengan protokol AMQP dan Mosquito dengan protokol MQTT untuk pembandingan kinerja metode pubsub menggunakan *broker* RabbitMQ dan *broker* Mosquito
6. Untuk menerapkan metodenya akan menggunakan bahasa pemrograman *Java*, *python* dan system operasi yang digunakan adalah system operasi Ubuntu 14.04

1.6 Sistematika pembahasan

Penelitian ini disusun dengan sistematika sebagai berikut :

BAB 1 PENDAHULUAN

Membahas mengenai latar belakang penelitian, rumusan masalah yang akan dibahas, batasan masalah dalam penelitian, tujuan yang ingin dicapai, serta manfaat penelitian.

BAB 2 DASAR TEORI

Membahas mengenai kajian pustaka dari penelitian telah ada serta dasar teori yang dijadikan landasan dalam perancangan sistem pakar.

BAB 3 METODOLOGI PENELITIAN

Memaparkan metode yang digunakan dalam penelitian yang diantaranya studi literatur, perancangan simulasi, implementasi metode yang digunakan ke simulasi, hasil dan pembahasan dari pengujian dan pengambilan kesimpulan.

BAB 4 HASIL DAN PEMBAHASAN

Penjelasan menggunakan hasil dan pembahasan dari pengujian menggunakan metode yang digunakan dalam penelitian ini

BAB 5 PENUTUP

Berisi kesimpulan dari hasil penelitian disertai dengan saran untuk pengembangan lebih lanjut.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Menurut Urs Hunkeler dan Hong Linh Truong dalam jurnalnya yang berjudul “MQTT-S – A Publish/Subscribe Protokol for Wireless Sensor Networks” disimpulkan bahwa metode distribusi data yang cocok digunakan WSN (*Wireless Sensor Network*) pada *smart home* adalah metode *publish/subscribe*. Karena dengan metode *publish/subscribe* data yang dikomunikasikan antara perangkat sensor yang ada pada lingkungan *smart home* dapat dilakukan berdasarkan dengan ketertarikan *subscriber* terhadap topik tertentu sehingga proses pengiriman data lebih cepat daripada menggunakan alamat dari perangkat yang melakukan komunikasi data seperti pada metode *point-to-point* lakukan. (Hunkeler, Truong, & Stanford-Clark, 2008).

2.2 Dasar Teori

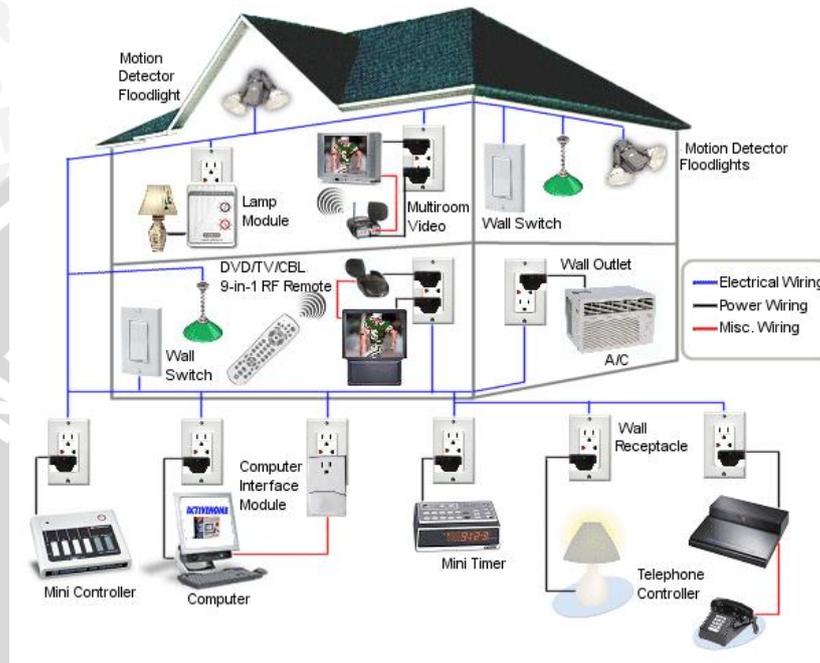
2.2.1 Smart Home

Teknologi masa kini sudah berkembang sangat pesat segala hal sesuatu sudah mulai di jalankan oleh komputer dan segala sesuatu juga sudah mulai terhubung dengan internet. Mulai dari bidang ekonomi, kesehatan, pendidikan, transportasi, dan bidang yang lainnya sudah mulai terhubung dengan internet. Begitu juga dengan rumah tinggal ataupun bangunan sudah mulai mempunyai infrastruktur jaringan salah satunya adalah *Smart Home*. *Smart Home* adalah rumah atau bangunan yang dilengkapi oleh pencahayaan, pemanas, dan perangkat elektronik yang dapat diatur secara *remotely* melalui ponsel ataupun melalui komputer. Perangkat-perangkat ini dapat saling berkomunikasi satu sama lain dikarenakan adanya suatu infrastruktur jaringan komunikasi yang sudah di bangun di dalam *Smart Home* yang dapat memungkinkan perangkat-perangkat tersebut dapat diatur secara *remote*. (Lalanda, Bourcier, J.Bardin, & S.Chollet, 2010)

Sejarah *Smart Home* dimulai dari masuknya perangkat rumah tangga yang menggunakan tenaga listrik antara tahun 1915-1920. Pada tahun tersebut listrik sudah mulai ditemukan dan sudah mulai diterapkan di dalam rumah tangga. Namun hal ini masih termasuk hal yang mewah bagi masyarakat pada saat itu jadi hanya beberapa saja yang dapat menggunakan listrik di rumah mereka. Pada tahun 1940 peminat penggunaan listrik pada perangkat rumah tangga semakin meningkat namun listrik tersebut hanya digunakan sebatas untuk pencahayaan saja namun ada beberapa yang menggunakan untuk hal yang lain. Hal ini disebabkan dikarenakan mahalnya harga listrik pada saat itu sehingga hanya orang tertentu saja yang bisa memanfaatkan sepenuhnya sebagian besar yang lainnya hanya memanfaatkan listrik untuk pencahayaan. Akhirnya semakin bertambah tahun penggunaan listrik semakin pesat dan semakin banyak perangkat rumah tangga yang menggunakan listrik. Pada tahun 1960 seorang warga Amerika

membangun sebuah rumah berkabel dan hampir keseluruhannya menggunakan listrik. (Gerhart, 1999)

Pada tahun 1966 seorang insinyur bernama Jim Sutherland membuat *Home Automation System* bernama ECHO IV namun project ini bersifat pribadi dan tidak pernah di publikasikan. Akhirnya pada tahun 1984 istilah *Smart Home* mulai dipublikasikan oleh *American Association of Housebuilders*. (Harper, 2003)



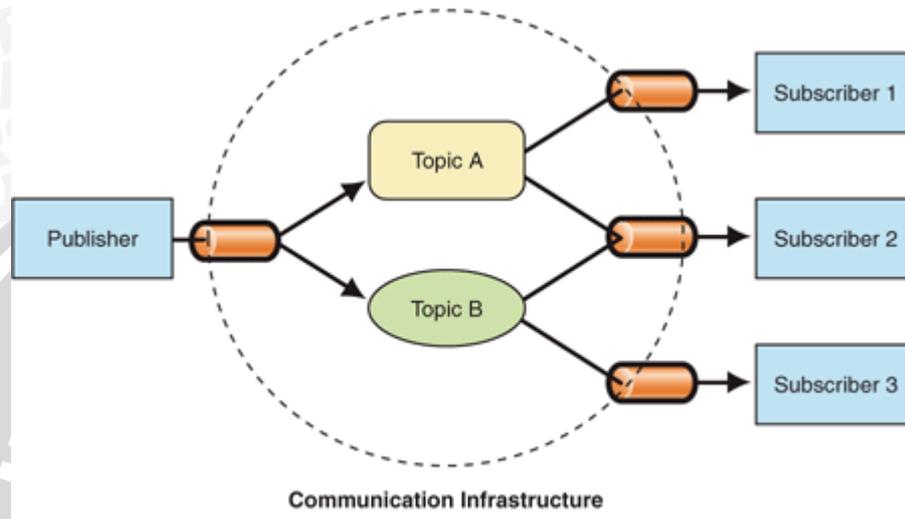
Gambar 2.1 Contoh Skema Smart Home

2.2.2 Publish/Subscribe

Publish/Subscribe adalah sebuah interaksi pendistribusian informasi atau data antara *publisher* (*sender*) dan *subscribers* (*receiver*). Dalam proses komunikasi data *publish/subscribe*, pihak *subscribers* akan menyatakan ketersediaannya dalam menerima pesan dan pihak *publisher* akan langsung mengirim pesan tersebut kepada *subscribers* yang memintanya. Dalam pengiriman pesan di *publish/subscribe*, pesan tersebut akan bersifat anonim, maksudnya *publisher* akan tetap mengirim pesan tanpa perlu mengetahui terlebih dahulu identitas *subscribers*nya ataupun ada atau tidak keberadaannya.

Sebuah sistem *Publish/Subscribe* terdiri dari *publisher*, *subscriber*, dan *publish/subscribe message broker* atau yang biasa disebut sebagai *message router*. *Publisher* dan *Subscribers* adalah peran yang dipegang oleh aplikasi atau sistem yang sedang dibangun. Dalam sebuah sistem *publish/subscribes* sebuah klien dapat memiliki peran sebagai *publisher* maupun peran sebagai *subscribers* dalam waktu yang bersamaan. *Subscribers* akan menyatakan keminatannya terhadap sebuah topik dengan cara berlangganan pada sebuah topik yang ada pada sistem *publish/subscribes*. Untuk *Publisher* dia akan melaporkan setiap event yang terjadi kepada sistem *publish/subscribe*. (Liu & Ozu, 2009)

Metode ini mendukung gaya komunikasi *many-to-many* dimana sumber data disebut *publish* dan ujung dimana data berakhir disebut *subscribe*. Perbedaan metode *publish subscribe* terletak pada cara *subscriber* untuk meminta atau berlangganan pada *publisher*-nya, perbedaannya terletak juga pada struktur dan format dari pesannya dan juga terletak pada arsitektur sistemnya. *Publish Subscribe* kebanyakan digunakan dalam berbagai bidang salah satunya dalam proses monitoring sistem jaringan. (Liu & Ozu, 2009)



Gambar 2.2 Model Sistem *Publish/Subscribe*

Sumber : (Liu & Ozu, 2009)

2.2.3 Komponen Sistem *Publish/Subscribe*

Dalam sistem *Publish/Subscribe* ada beberapa komponen, istilah sentral, dan struktur yang membangun sistem *Publish/Subscribe*.

- **Basic System**

Bentuk utama dari sistem *Publish/Subscribe* adalah pihak *publisher* (penyedia konten) dan *subscribers* (penerima konten). Dalam sistem ini sebuah *publisher* akan mendeteksi adanya event atau permintaan penyediaan konten, setelah event berhasil terdeteksi selanjutnya *publisher* akan mengumumkan event tersebut dalam bentuk pemberitahuan (*notification*). Sebuah pemberitahuan akan merangkung semua informasi yang berkaitan untuk mengamati sebuah event yang terjadi dalam sistem.

Ada banyak istilah untuk bentuk dari *pub/sub* ataupun event dari sistem, sebagai contohnya seperti *subscribers*, *consumer*, dan ujung dari sebuah event yang identik. Istilah tadi adalah macam-macam istilah dari pihak *subscribe*. Sama halnya dengan pihak *subscribe*, pihak *publish* juga terkenal dengan banyak istilah seperti *publisher*, *producer*, dan Sumber dari event yang identik. Seperti yang

disebutkan diatas tadi, sebuah notifikasi atau event dari pesan tersebut berfungsi untuk menandakan sebuah pengamatan event yang sedang terjadi dalam sebuah sistem.

Sebuah event dapat mewakili dari segala keadaan yang berbeda dari sebuah transisi yang telah terjadi dan yang telah di kirimkan dari satu entitas ke beberapa entitas yang lainnya. Event dalam sistem dapat dikategorikan berdasarkan atributnya, seperti sifat fisik yang mana yang dapat dihubungkan. Misalnya seperti event yang spasial dan event yang temporer akan dicatat aktivitas fisiknya. Selain itu sebuah event dapat merupakan kombinasi dari kedua hal diatas seperti contohnya, sebuah event dapat memiliki informasi yang bersifat spasial dan temporer dalam satu waktu. (Tarkoma, 2012)

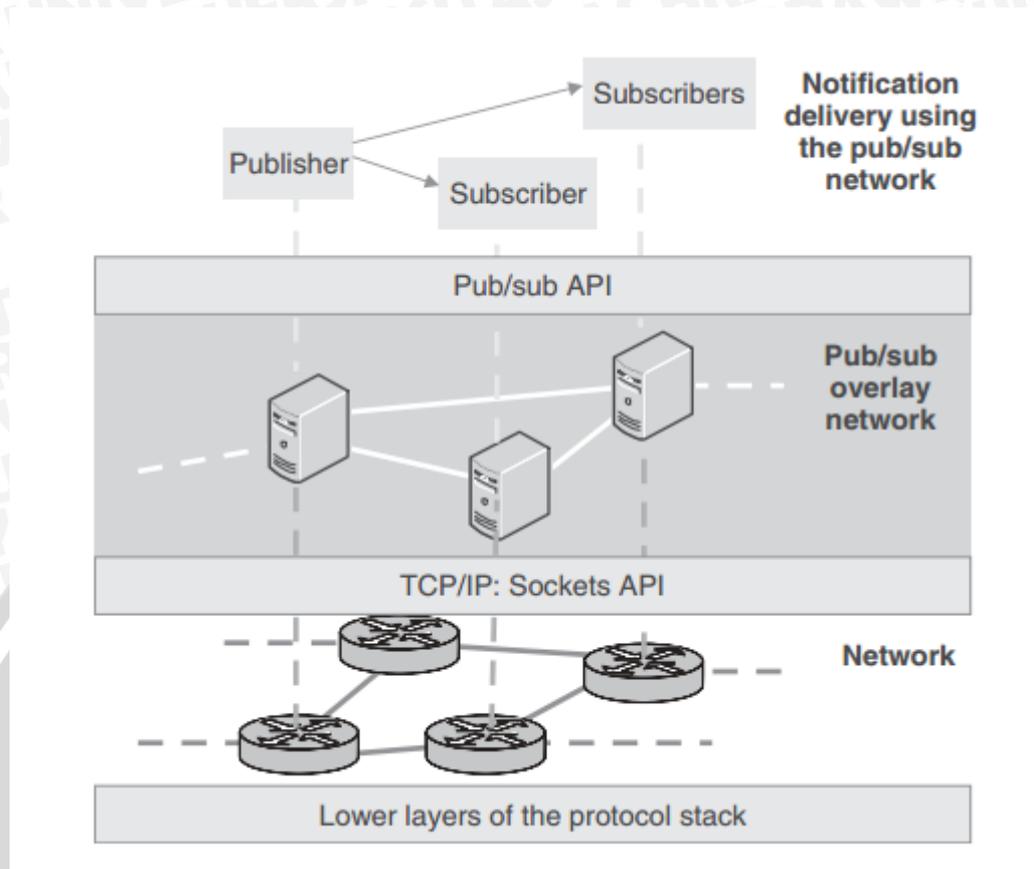
- **Distribusi dan *Overlay Network***

Sistem *Publish/Subscribe* kemungkinan dalam aslinya dapat bersifat central ataupun terdistribusi. Proses notifikasi dan kewajiban pengiriman dapat disediakan oleh entitas yang berbeda. Entitas tersebut adalah, *Publisher*, *Centralized Broker*, dan sebuah set dari *broker* dalam konfigurasi routing yang biasa disebut sebagai *overlay network*.

Sebuah event dan proses notifikasi dapat dengan mudah di implementasikan pada *publisher* dan *centralized broker*. Seperti yang dijelaskan diatas tadi pendekatan ini tidak diukur berdasarkan banyaknya entitas dan event yang ada pada sebuah sistem. Skalabilitas dapat di tingkatkan dengan cara mengimplementasikan sistem *pub/sub* dengan sebuah set dari *broker* sebagai konstruksi lapisan atas diatas lapisan network.

Sebuah layer aplikasi *overlay network* akan di implementasikan diatas layer network dan layer aplikasi tersebut akan menyediakan beberapa layanan seperti *resource lookup*, *overlay multicast*, dan penyimpanan terdistribusi. Sebuah *overlay network* biasanya menyediakan beberapa fitur yang berguna seperti contohnya penyebaran fungsi terdistribusi secara mudah, fleksibilitas terhadap kegagalan network, dan toleransi kegagalan (Lagutin, Visala, Zahmesky, Burbridge, & Marias, 2010).

Gambar 2.10 adalah ilustrasi dari sebuah *pub/sub overlay network*. Dua komponen penting dari *distributed pub/sub network* adalah *topology broker* dan bagaimana keadaan routing yang di munculkan dan di rawat oleh *broker*. Dengan menyebarkan status dari routing berarti seberapa tertariknya *subscribers* terhadap sebuah informasi yang dimiliki oleh *publisher*. Pada dasarnya status routing yang disimpan oleh *broker* harus mengaktifkan status routing-nya untuk meneruskan pesan dari event kepada *broker* yang lain ataupun *subscriber* yang lain yang sebelumnya sudah berlangganan terhadap notifikasi. (Tarkoma, 2012)



Gambar 2.3 Contoh *Publish/Subscribe Overlay Network*

Sumber : (Tarkoma, 2012)

- **Aggrements**

Sistem *Pub/Sub* digunakan untuk memfasilitasi pengiriman dari sebuah pesan namun, arti sebenarnya dari event yang ada pada penjelasan di atas sebelumnya adalah aplikasi dan domain yang spesifik. Dalam rangka perancangan sistem *pub/sub* dengan entitas yang banyak harus mengikuti beberapa perjanjian yang harus diperhatikan. Perjanjian-perjanjian tersebut adalah:

- Perjanjian format dan *syntax* dari pesan notifikasi. Perjanjian ini juga kemungkinan mengandung rincian tambahan seperti hal yang berkaitan dengan tanda waktu ataupun keamanan konten.
- Perjanjian protokol yang digunakan untuk transfer sebuah *event* antar dua entitas. Hal ini juga termasuk beberapa parameter yang terkait seperti contohnya adalah keamanan, reliabilitas, dan lain-lain.
- Perjanjian penyaringan notifikasi *semantics*. Hal ini menspesifikasikan element apa saja yang akan diteruskan berdasarkan dari *publisher*, waktu pengamatan, dan tipe dari *event* tersebut.

- Perjanjian pada kejelasan dari event yang diterbitkan. Hal itu mungkin penting untuk membatasi pengiriman dan proses dari sebuah *event* yang ada di lingkungan operasi atau lingkungan sistem
- Perjanjian aplikasi dan *domain* yang spesifik dari *event*. Perjanjian ini berada diluar dari cakupan *pub/sub system*.

Itulah beberapa perjanjian yang tersirat atau jelas yang dibutuhkan untuk mendesain ataupun membangun sebuah sistem *pub/sub* pada lingkungan yang memiliki banyak entitas. (Tarkoma, 2012)

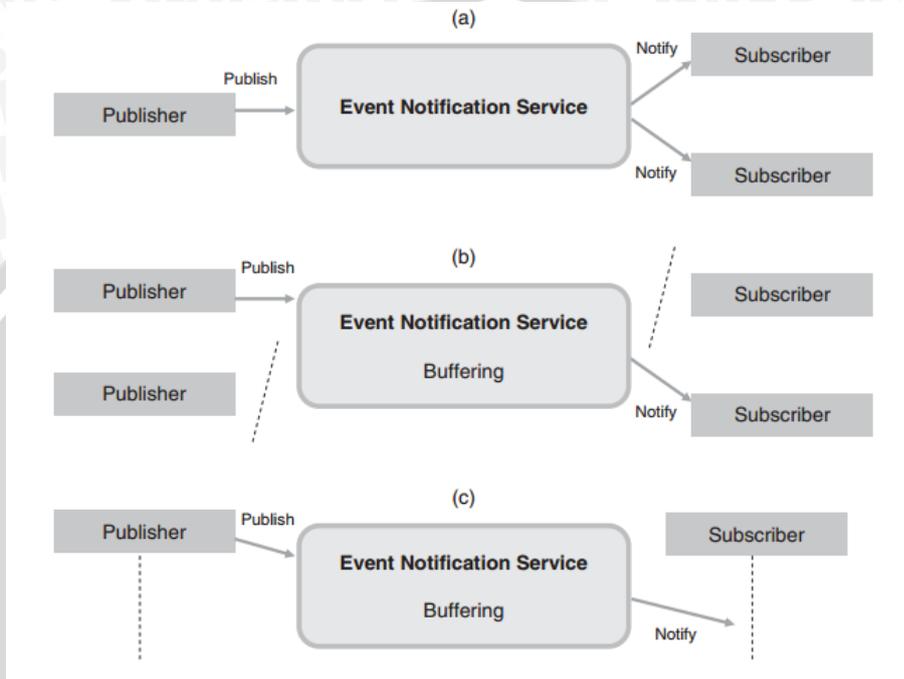
- **Sifat Dasar (*Basic Properties*)**

Teknologi *pub/sub* sudah berkembang sejak bermula pada akhir tahun 1980 untuk menjanjikan teknologi untuk menghubungkan komponen dari perangkat lunak secara lintas waktu (*time*), ruang (*space*), dan sinkronisasi (*synchronization*) (Kurana, 2005). Tiga sifat diatas meringkas fitur yang menonjol dari teknologi. Pada gambar 2.11 akan dijelaskan skema *decoupling* dari tiga sifat yang ada pada penjelasan sebelumnya.

Space decoupling dijelaskan pada sub gambar A, yang dimana sebuah layanan notifikasi *event* memisahkan *publisher* dan *subscriber*. Sebuah pesan *event* berasal dari *publisher* di transfer ke *event notification service* lalu setelah itu pesan tersebut akan di transferkan kepada *subscriber*. Pada sub gambar B menjelaskan tentang *time decoupling*. Prosesnya hampir sama dengan proses *space decoupling* yang membedakannya adalah adanya proses *buffering* ada sisi layanannya. *Time decoupling* dapat dicapai dengan cara menyimpan pesan pada *message buffer* yang ada pada *event notification service* yang pada akhirnya akan dikirimkan kepada *subscriber*. Sub gambar C mengilustrasikan tentang proses *synchronization decoupling*, yang dimana proses ini menekankan pada aspek temporer atau sementara. Fase *publish* dan fase pemberitahuan pengiriman *event* dipisahkan dan fase-fase tersebut tidak membutuhkan proses sinkornisasi. Pesan pertama dikirim ke *event notification service* lalu akan dikirim ke *subscriber*.

Gambar 2.12 adalah ringkasan dari proses *decoupling properties* atau yang bisa dikenal dengan teknik atau cara komunikasi. Dapat dilihat sebelumnya, sebuah teknik komunikasi tidak bersifat *orthogonal* melainkan digabungkan dalam rangka mengimplementasikan sistem yang yang lebih mutakhir atau terbaru. Metode komunikasi *Remote Procedure Calls (RPC)*, *Remote Method Invocation (RMI)* dan *asynchronous RPC/RMI* tidak menawarkan proses *decoupling* ruang dan waktu. Tapi mereka menawarkan proses *decoupling synchronization*. Metode Komunikasi *tuple space* menawarkan proses *decoupling* pada ruang dan waktu melalui ruang yang di bagi. Namun pembacaan *tuple space* di blok dan dengan demikian *tuple space* tidak menawarkan proses *decoupling synchronization* (Visala, Lagutin, & Tarkoma, 2009). Lain halnya dengan metode komunikasi *message queuing*. *Message queuing* menewarkan ketiga *properties* yaitu *space*, *time*, dan *synchronization* dan hal itu ditujukan untuk teknologi yang lebih mutakhir.

Dari penjelasan diatas dapat diketahui bahwa sistem *pub/sub* berdasarkan pada metode komunikasi *message queuing* dan *message-oriented middleware*. *Message queuing* adalah metode komunikasi yang mempekerjakan *message passing* antara pengirim dan penerima dengan pertolongan dari antrian pesan dari pihak pengirim. Setelah pengiriman telah terjadi, pesan dapat di hilangkan dari antrian. Meskipun telah dihapus dari antrian pesan tetap bisa dikirimkan jika terjadi kesalahan dalam penerimaan pesan. (Tarkoma, 2012)



Gambar 2.4 *Decoupling* sifat pada sistem *pub/sub* (a) *Space Decoupling* (b) *Time Decoupling* (c) *Synchronization Decoupling*

Sumber : (Tarkoma, 2012)

Tabel 2.1 Ringkasan dari *decoupling properties*

Abstraction	Space/time decoupling	Synchronization decoupling
Message passing	No	Varies
RPC/RMI	No	Invoker is blocked
Async RPC/RMI	No	Yes
Tuple spaces	Yes	Reader is blocked
Message queuing	Yes	Varies
Pub/sub	Yes	Yes

Sumber : (Tarkoma, 2012)

2.2.4 Model *Subscription*

Banyak cara untuk mengidentifikasi ketertarikan terhadap suatu topik pada pihak *subscribers*, hal ini menyebabkan munculnya beberapa type model *subscribers* berdasarkan ketertarikannya terhadap pemberitahuan yang diberikan oleh pihak *publisher*. Model tersebut ada tiga yaitu *topik-based model*, *content-based model*, dan *type-based model*. (Virgillito, 2003)

- **Topik-based Model**

Pada model ini, setiap pemberitahuan (*notification*) akan di kelompokkan sesuai dengan topiknya atau subjeknya. Sebagai contohnya misalkan sebuah *subscriber* akan mendeklarasikan ketertarikannya terhadap topik tertentu dan *subscribers* tersebut kelak akan mendapatkan semua pemberitahuan (*notification*) sesuai dengan topik yang diminati oleh *subscribers* tersebut. Pada model ini, setiap topik yang berkaitan terhadap *logical event channel*-nya akan secara ideal melakukan koneksi terhadap setiap kemungkinan *publisher* yang dimana *subscriber* tersebut tertarik pada topik yang di berikan oleh *publisher* tersebut.

Oleh karena itu, munculah asosiasi yang bersifat statis antara sebuah channel dan setiap *subscribers* yang ada di channel tersebut. Setelah itu ketika *publisher* akan memberikan notifikasinya, sistem tersebut tidak perlu melakukan perhitungan terhadap semua penerimaan topik tersebut. Model ini telah menjadi solusi yang diadaptasi oleh setiap inkarnasi dari pub/sub sistem. (Virgillito, 2003)

- **Content-based Model**

Pada model ini, sebuah *subscribers* akan mengekspresikan ketertarikannya terhadap kondisi yang spesifikasi pada isi dari notifikasi yang ingin diterima oleh *subscribers*. Dengan kata lain, filter yang ada pada langganan adalah permintaan yang disusun oleh konjugasi dari sebuah kendala atas nilai dari atribut pada notifikasi. (Virgillito, 2003)

- **Type-based Model**

Pada model ketiga dari model *subscription* adalah model berdasarkan tipenya. Model ini hampir sama dengan konsep *object-oriented* pada pemrograman. Pemberitahuan pesan dalam model ini di deklarasikan sebagai sebuah objek yang dimiliki oleh tipe yang spesifik yang dengan demikian dapat merangkum atribut serta *method* yang digunakan sama seperti pada *oop* (*object oriented programming*). (Virgillito, 2003)

Di model ini, pendeklarasian tipe yang diinginkan adalah dari atribut pembeda yang utama, maksudnya sehubungan dengan model tersebut, *type-based pub/sub* akan menempatkan dirinya bagaimanapun juga dibagian tengah dari sistem. Caranya dengan memberikan struktur yang berbentuk butiran kasar pada pemberitahuan (seperti pada *topik-based*) yang dimana batasan yang halus dapat diekspresikan diatas atributnya (seperti pada *content-based*) atau diatas metodenya (sebagai konsekuensinya dari pendekatan secara *object-oriented*).

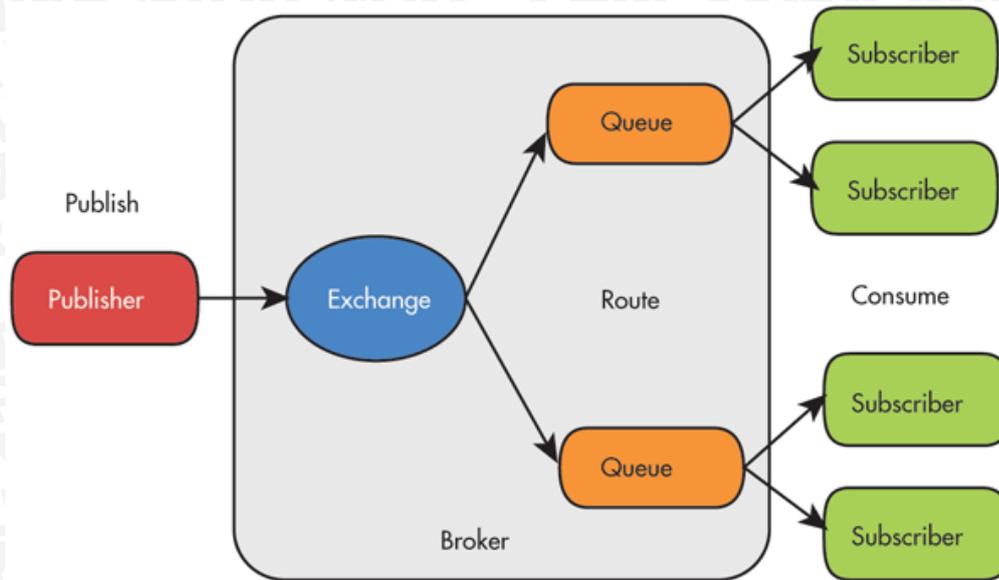
Dalam hal ini, *type-based pub/sub* lebih menyerupai dengan *topik model* yang telah disaring.

2.2.5 AMQP (Advanced Message Queueing Protocol)

AMQP adalah sebuah protokol pengiriman pesan yang kebanyakan digunakan dalam sebuah perusahaan yang bergerak dibidang komunikasi. Protokol ini di desain untuk *support* hampir disemua pengiriman pesan berbasis aplikasi distribusi dan bisnis. Protokol ini bekerja seperti pesan instan atau seperti email yang membedakan adalah protokol AMQP ini terdiri dari kedua protokol jaringan yaitu *producer/consumer* dan *broker* yang saling bekerja sama satu sama lain dan sebuah model protokol yang dimana menjelaskan secara spesifik perwakilan dari sebuah pesan dan memberikan perintah kepada setiap pihak untuk saling bekerja sama. Selain itu, pesan yang ada di protokol AMQP bersifat mandiri dan data yang berada pada besan tersebut bersifat kekal selain itu pesan dari AMPQ juga tidak memiliki batasan ukuran meskipun pesan tersebut berukuran sebesar 4 *Gbyte* atau berukuran 4*Kbyte* AMQP masih dapat mendukungnya dan tidak berpengaruh kepada keamanan, reliabilitas, dan keefisienan ketiga hal tersebut masih dapat bekerja dengan baik berapapun ukuran pesan yang di kirimkan.

Selain itu protokol ini juga menyediakan alur yang dapat dikendalikan, sebuah komunikasi berorientasi pesan dengan jaminan pengiriman pesan seperti contohnya pesan tersebut akan dikirim sekali atau tidak sama sekali dan jaminan pesan tersebut akan dikirim sampai ke tempat tujuan dengan konfirmasi dari pihak end point setidaknya satu kali. Dalam sisi keamanan, protokol ini memiliki keamanan yang berbasis autentifikasi atau enkripsi pada SASL (*Simple Authentication Security Layer*) dan TLS (*Transport Layer Security*). (Luzuriaga, et al., 2015)





Gambar 2.5 Arsitektur AMQP

sumber (Electronic Design, 2016)

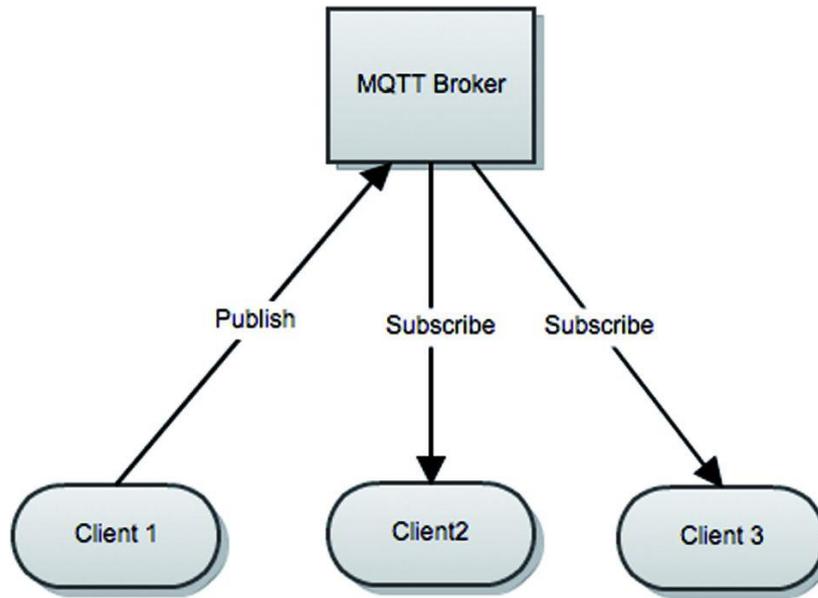
Di protokol AMQP ada 3 komponen utama yang mana digunakan untuk menyambungkan ke rantai pemrosesan yang ada di server untuk membuat fungsi yang diinginkan. Ketiga komponen itu adalah *exchange*, *message queue*, dan *binding*. Komponen *exchange* bertugas menerima pesan dari aplikasi *publisher* dan mengalihkan pesan tersebut ke “*message queue*” berdasarkan kriteria tertentu biasanya kriteria itu bisa dalam bentuk isi pesan tersebut ataupun sifat dari pesan tersebut. Untuk komponen *message queue* berfungsi untuk menyimpan pesan hingga pesa tersebut akan aman di proses oleh klien atau banyak klien. Untuk komponen *binding* bertugas untuk menyatakan relasi antara “*message queue*” dan sebuah “*exchange*” dan komponen ini akan menyediakan kriteria untuk routing pesan. (Aiyagari, Arrot, Atwell, & Broner, 2008)

2.2.6 MQTT (Message Queueing Telemetry Transport)

MQTT adalah sebuah protokol pengiriman pesan yang ringan yang terjadi antara *machine-to-machine* dan protokol ini juga memiliki ruang lingkup *mobile network*. Prosedur pertukaran informasi yang terjadi pada protokol ini adalah berbasis sumber daya yang efisien dan juga tidak memerlukan format data yang spesifik untuk mengirimkan pesannya. Selain itu protokol ini juga menawarkan keamanan dan kepastian semua pesan akan terkirim meskipun koneksi terputus sementara. Hal ini menjadikan protokol tersebut sebagai solusi terhadap kondisi jaringan yang tidak stabil seperti pada jaringan yang digunakan pada perangkat *mobile*.

Selain digunakan dalam perangkat mobile, ternyata protokol ini juga digunakan dalam perangkat sensor, *embedded system* pada kendaraan, atau laptop dan komputer skala besar. Protokol ini mendukung pola pengiriman pesan

Publish-Subscribe dan protokol ini sangatlah simple sehingga protokol ini sangat ringan untuk digunakan dalam komunikasi data. (Luzuriaga, et al., 2015)



Gambar 2.6 Arsitektur MQTT

Sumber (Entwickler.de, 2016)

Di Protokol MQTT ini terdapat QoS (*Quality of Services*) dalam pengiriman pesannya. QoS di MQTT ada 3 tingkatan yaitu tingkat 0 (*at most once*), tingkat 1 (*at least once*), dan tingkat 2 (*exactly once*). QoS tingkat 0 adalah QoS paling minimal di MQTT dan QoS ini akan menjamin usaha yang paling baik dalam pengiriman pesannya. Sebuah pesan yang di kirim dengan QoS ini tidak akan di *Acknowledge* oleh penerima pesan dan pesan ini juga tidak akan disimpan ataupun dikirim kembali oleh pengirim pesan QoS tingkat 0 ini juga bisa disebut dengan *fire and forget*. Untuk QoS tingkat 1 sebuah pesan akan dikirim oleh pengirimnya setidaknya satu kali pengiriman pesan kepada penerima. Tetapi sebuah pesan yang dikirimkan dengan QoS tingkat 1 ini juga bisa dikirimkan ke penerima lebih dari satu kali dan pengirim akan menyimpan pesan dan akan mengirimkannya setelah mendapatkan *acknowledge* dari penerima.

Untuk QoS tingkat 2 adalah tingkat yang paling tinggi. QoS ini akan menjamin pesan tersebut pasti sampai ke penerima. QoS ini adalah yang paling aman dan yang paling lambat. Jaminan tersebut disediakan oleh dua aliran bolak balik dari pengirim dan penerima. (MQTT Essentials Part 6: Quality of Service 0, 1 & 2, 2016)

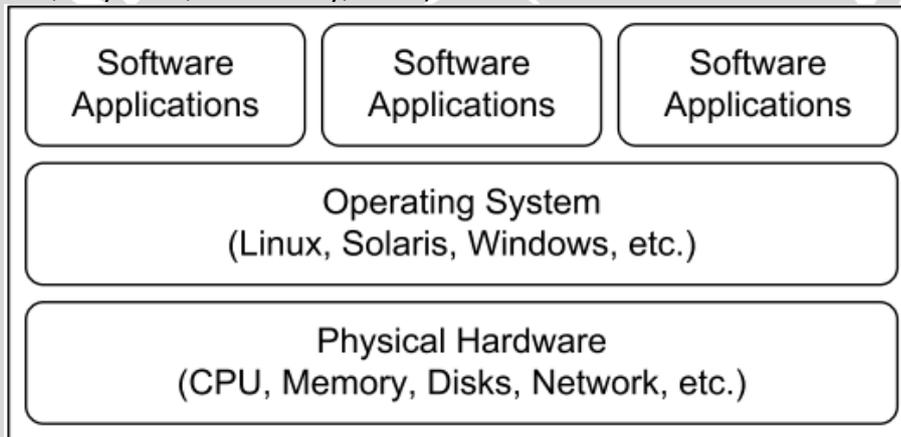
2.2.7 Virtual Private Server (VPS)

Virtual Private Server termasuk salah satu teknologi dalam bidang server yang disebut dengan *virtualization*. Teknologi virtualisasi adalah salah satu cara mengubah sebuah komputer fisik seolah-olah ada dua komputer atau lebih dalam satu fisik komputer. Masing-masing komputer non-fisik atau “*virtualized*”

dilengkapi dengan arsitektur basis yang sama dengan komputer fisik pada umumnya. Sehingga meskipun memiliki satu komputer fisik namun bisa melakukan kinerja seperti beberapa komputer melakukan satu proses.

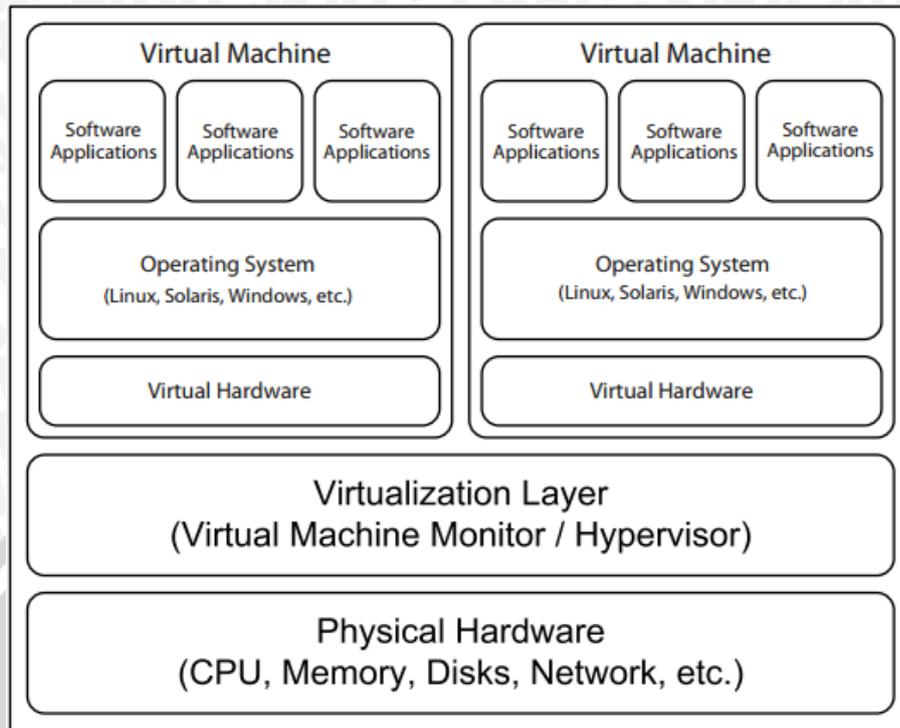
Virtualisasi adalah sebuah konsep dimana sebuah sumber daya dari satu komputer dapat dibagi ke beberapa lingkungan secara bersamaan. Lingkungan ini dapat beroperasi dan akan saling berkomunikasi dan sadar antar lingkungan satu dengan lainnya. Lingkungan ini umumnya disebut dengan mesin virtual atau VM (*Virtual Machine*). Hampir semua VM dapat di install sebuah sistem operasi (contohnya seperti windows, linux, dll). Sistem operasi yang diinstall di sebuah VM disebut dengan *guest operating sistem*.

Biasanya semua perintah yang diberikan oleh sebuah VM akan langsung tertuju ke perangkat fisik sebuah komputer yang telah di virtualisasi. Hal ini dapat membuat proses yang dilakukan oleh sebuah komputer fisik akan lebih cepat dan lebih efisien dibandingkan dengan metode emulasi. Untuk lebih memahami tentang virtualisasi, akan dijelaskan di gambar 2.9 dan 2.10 perbedaan antara arsitektur komputer tradisional dan arsitektur komputer yang sudah di virtualisasi. (Marshall, Reynolds, & McCrory, 2006)



Gambar 2.7 contoh arsitektur komputer tradisional

Sumber : (Marshall, Reynolds, & McCrory, 2006)



Gambar 2.8 contoh arsitektur komputer yang sudah di virtualisasi

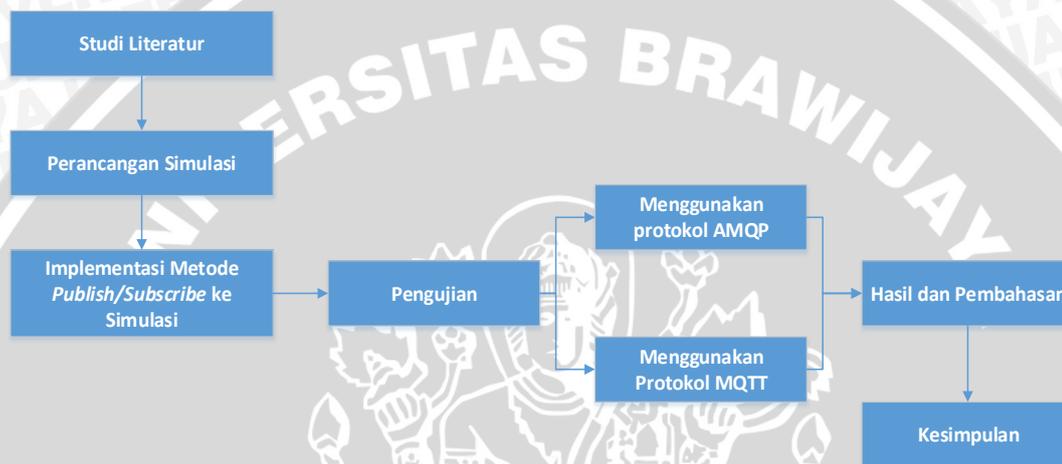
Sumber: (Marshall, Reynolds, & McCrory, 2006)



BAB 3 METODOLOGI

3.1 Metode Penelitian

Pada bab ini akan dijelaskan langkah-langkah perancangan, implementasi, pengujian, serta analisa yang akan diterapkan untuk penelitian yang berjudul Analisa Metode *Publish/Subscribe* untuk Komunikasi Data Antar Perangkat di Lingkungan Smart Home. Selain itu akan di ambil kesimpulan dari hasil analisa dan pengujian untuk di gunakan kedepannya sebagai informasi dalam membangun infrastruktur komunikasi *Smart Home*. Tahap-tahap dari keseluruhan proses penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Skema Tahap-Tahap Metodologi Penelitian

3.1.1 Studi Literatur

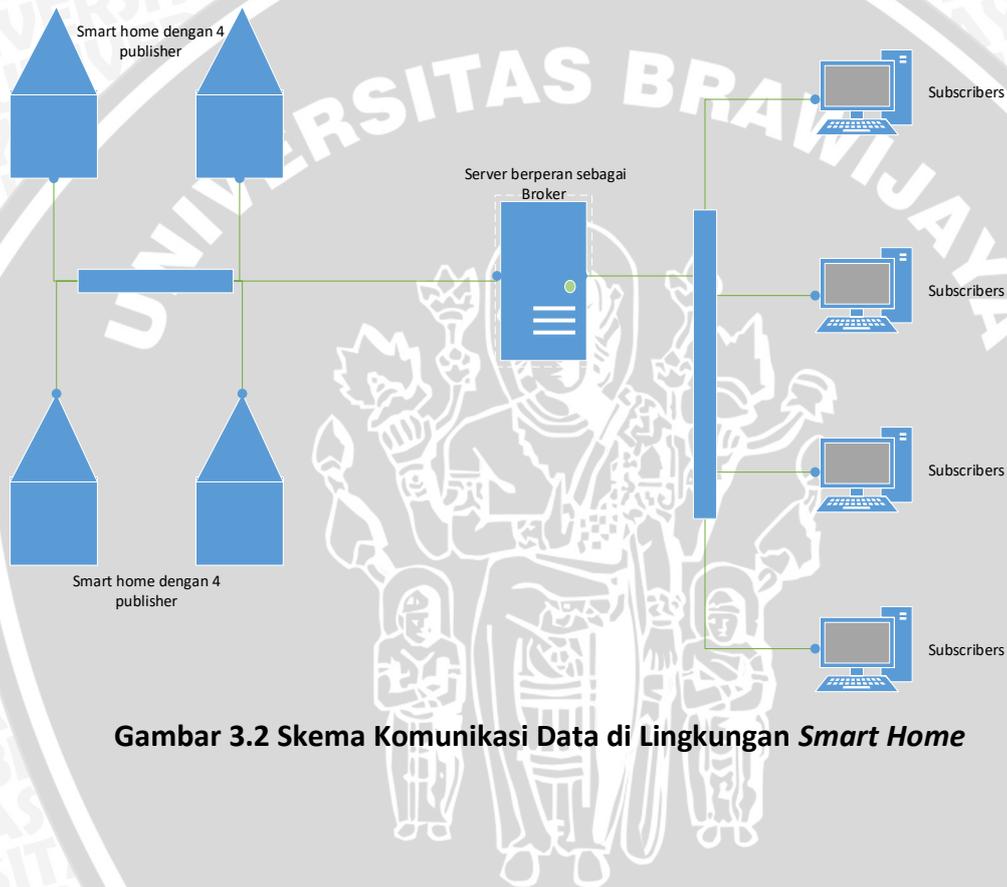
Pada penelitian yang bersifat non-implementatif (Analitik) ini membutuhkan studi literatur untuk mempelajari dan membangun simulasi komunikasi data antar perangkat di lingkungan *smart home* menggunakan metode *publish/subscribe*. Teori – teori pendukung yang akan digunakan dalam penelitian ini antara lain:

1. Teknologi Smart Home
2. Metode *Publish/Subscribe*
3. Protokol MQTT dan AMQP
4. Teknologi VPS (*Virtual Private Server*)
5. Bahasa Pemrograman *Java*

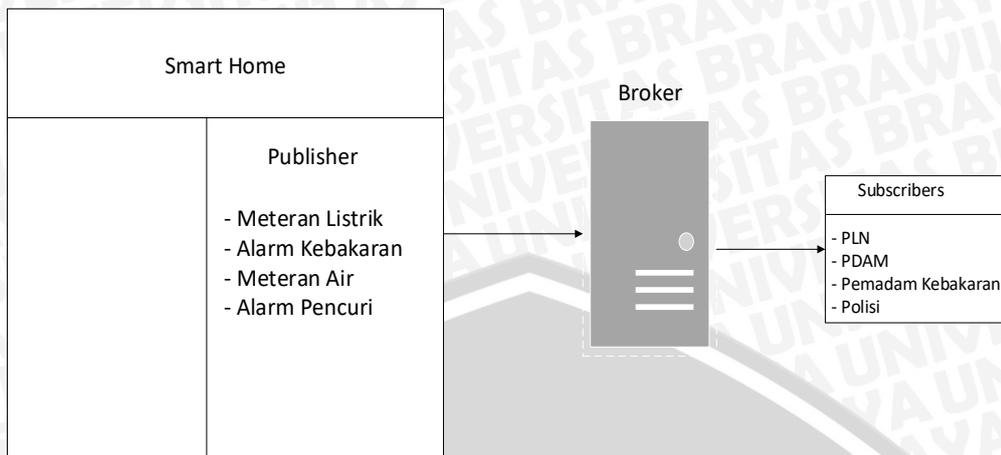
Referensi yang digunakan dalam pembangunan simulasi metode *publish/subscribe* didapatkan dari literatur yang relevan dan *valid* yang diantaranya adalah buku, *paper* (jurnal ilmiah), dokumen dari internet serta *thesis* peneliti lain.

3.1.2 Perancangan Simulasi

Konsep penelitian ini adalah membangun sebuah simulasi komunikasi data di lingkungan *smart home* dengan menggunakan metode komunikasi data *publish/subscribe* dan dengan objek uji 4 *publisher*, 1 *broker* yang terletak diluar *smart home* dan 4 *subscribers* sebagai penerima data. Untuk *broker* akan menggunakan vps (virtual private server) atau server ditaruh diperangkat berbeda dan berdiri sendiri agar data yang didapatkan bisa digunakan untuk analisis dan lebih bernilai untuk di analisis. Pada Gambar 3.2 menunjukan skema komunikasi data dalam suatu sistem komunikasi data di lingkungan *smart home* secara umumnya.



Gambar 3.2 Skema Komunikasi Data di Lingkungan *Smart Home*



Gambar 3.3 Skema Detail Komunikasi Data Menggunakan Metode *Publish Subscribe*

Pada Gambar 3.3 dijelaskan setiap *publisher* yang ada di setiap rumah di lingkungan *smart home* akan mengirimkan data yang dihasilkannya kepada *broker* yang terletak pada server yang sudah disediakan di lingkungan smart home. Setelah data yang ada pada setiap *publisher* dikirimkan ke *broker*, kemudian data tersebut akan diambil oleh setiap *subscriber* sesuai setiap kebutuhan *subscriber* terhadap data yang ada di *broker* yang telah dikirim oleh *publishernya*. Dalam penelitian ini *publisher* yang digunakan ada 4 *publisher*. Masing-masing *publisher* mempunyai struktur data yang berbeda. Struktur data dari setiap *publisher* akan dijelaskan dibawah ini

- a. Meteran Listrik

Tabel 3.1 Struktur Data Meteran Listrik

Tipe Data	Panjang Data	Satuan	Contoh
Integer	256 byte	KwH(Kilowatt/Hour)	1500 KwH

Contoh data yang dikirim “R 001 3547 KwH” Panjang data tersebut memiliki panjang data sebesar 14 byte terhitung dari huruf “R” sampai dengan angka “7” karakter spasi dimasukan dalam hitungan. Untuk satuan “KwH” tidak ikut dikirimkan melainkan hanya dicetak secara local

b. Alarm Kebakaran

Tabel 3.2 Struktur Data Alarm Kebakaran

Tipe Data	Panjang Data	Satuan	Contoh
Boolean	1 byte	-	1 / 0

Contoh data yang dikirim "R 001 1" Panjang data tersebut memiliki panjang data sebesar 7 byte terhitung dari huruf "R" sampai dengan angka "1" pada posisi akhir karakter spasi dimasukkan dalam hitungan.

c. Meteran Air

Tabel 3.3 Struktur Data Meteran Air

Tipe Data	Panjang Data	Satuan	Contoh
Integer	256 byte	M ³ (Meter Kubik)	3000 M ³

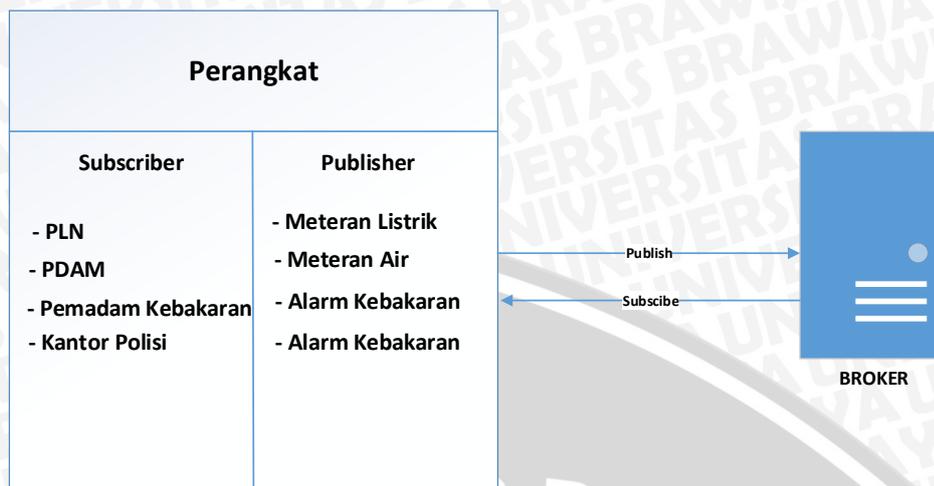
Contoh data yang dikirim "R 001 3547 Meter Kubik" Panjang data tersebut memiliki panjang data sebesar 14 byte terhitung dari huruf "R" sampai dengan angka "7" karakter spasi dimasukkan dalam hitungan. Untuk satuan "Meter Kubik" tidak ikut dikirimkan melainkan hanya dicetak secara local

d. Alarm Pencuri

Tabel 3.4 Struktur Data Alarm Pencuri

Tipe Data	Panjang Data	Satuan	Contoh
boolean	1 byte	-	1/0

Contoh data yang dikirim "R 001 1" Panjang data tersebut memiliki panjang data sebesar 7 byte terhitung dari huruf "R" sampai dengan angka "1" pada posisi akhir karakter spasi dimasukkan dalam hitungan.



Gambar 3.4 Skema Simulasi Komunikasi Data

Gambar 3.4 menjelaskan tentang skema simulasi yang akan dipakai dalam pengujian nanti. Dapat dilihat dari skema diatas, posisi broker nanti akan terletak diluar jaringan lokal yang berbeda dengan perangkat yang berisi 4 *publisher* dan 4 *subscriber* dalam satu perangkat.

3.1.3 Implementasi Metode ke Simulasi Komunikasi Data

Implementasi akan dilakukan sesuai dengan perancangan yang telah ditentukan sebelumnya. Simulasi komunikasi data menggunakan metode *publish/subscribe* ini akan dibangun dengan menggunakan bahasa pemrograman *Java* dan untuk *brokernya* akan diletakan disebuah VPS (*Virtual Private Server*) atau di letakan di perangkat berbeda yang berdiri sendiri maksudnya diluar dari perangkat yang digunakan sebagai *publisher* dan menggunakan OS Ubuntu Server 14.04. Sedangkan dalam sisi *subscriber*-nya untuk sementara penampilan datanya akan berupa console dari terminal.

3.1.4 Pengujian Simulasi Komunikasi Data

Pengujian dan analisis simulasi komunikasi data dengan metode *publish/subscribe* bertujuan untuk mengetahui kualitas dan performa dari metode *publish/subscribe* dalam komunikasi data di lingkungan smart home. Lokasi pengujian simulasi ini nanti akan diuji di jaringan lokal dengan posisi *broker* ada diluar jaringan lokal (*broker* menggunakan cloud VPS) . Skenario pengujian yang akan dilakukan nanti adalah dengan cara mengirimkan data dari *publisher* yang ada pada setiap rumah ke *broker*. Pengiriman data akan terjadi dalam 5 tahap pengiriman data yaitu 100 data, 200 data, 300 data, 400 data, dan 500 data.

Masing-masing tahap tersebut nanti akan dilakukan 10 kali percobaan untuk mendapatkan data yang lebih akurat. Data yang akan di dihasilkan dan dikirim dari setiap *publisher* akan di *generate* secara random dan setiap data akan diberi *header* untuk untuk penomoran data agar data yang terkirim sesuai jumlah

dengan yang dikirim atau tidak. Gambar 3.4 adalah flowchart pengiriman data yang dilakukan setiap *publisher* pada tahap pertama pengiriman data. Untuk tahap selanjutnya diagram alir yang digunakan sama hanya bedanya ada di jumlah data yang akan di *generate*.



Gambar 3.5 Diagram Alir Pengiriman Data untuk Setiap *Publisher*

Setelah dari *broker* data tersebut akan diambil oleh *subscriber* yang membutuhkannya. Pengujian ini akan dilakukan dengan memakai dua *broker* yang berbeda yaitu *broker RabbitMQ* dan *broker Mosquitto* yang masing-masing mempunyai protokol sendiri-sendiri. Untuk *broker RabbitMQ* dengan protokol AMQP (*Ashynronus Messgae Queueing Protokol*) dan *broker Mosquitto* dengan protokol MQTT (*Message Query Telemetry Transport*).

Akan ada 2 skenario pengujian. Skenario pertama adalah pengiriman 5 tahap data (100 data, 200 data, 300 data, 400 data, 500 data) dan merekam data hasil pengujian dari sisi *publisher* ke *broker*, dari sisi *broker* ke *subscriber*, dan dari sisi *publisher* ke *subscriber* dan dengan menggunakan protokol AMQP. Untuk Skenario kedua adalah pengiriman 5 tahap data (100 data, 200 data, 300 data, 400 data, 500 data) dan merekam data hasil pengujian dari sisi *publisher* ke *broker*, dari sisi *broker* ke *subscriber*, dan dari sisi *publisher* ke *subscriber* dan menggunakan protokol MQTT.

3.1.5 Pengambilan Kesimpulan

Kesimpulan ditarik berdasarkan tahapan yang telah dilalui, yaitu tahap perancangan hingga tahap analisa dan pengujian simulasi tersebut. Saran berguna untuk memberikan gambaran serta ruang dalam pengembangan penelitian ini kepada peneliti berikutnya.



BAB 4 HASIL DAN PEMBAHASAN

Pada bab ini akan diuraikan tentang skenario pengujian simulasi komunikasi data menggunakan metode *publish/subscribe* antar perangkat pada lingkungan smart home. Dalam pengujian ini akan dilakukan beberapa kali percobaan untuk mendapatkan data yang kelak akan digunakan untuk melakukan analisa. Setelah pengujian selesai tahap selanjutnya adalah melakukan analisa dari data yang didapat sebelumnya dan mengambil kesimpulan dari hasil analisa tersebut.

4.1 Pengujian dan Analisa

Pengujian dan Analisa ini bertujuan untuk mengetahui kinerja dari sistem komunikasi data menggunakan metode *publish subscribe* serta mengetahui komponen-komponen apa saja yang membangun sistem komunikasi data ini.

4.2 Sistematika Kinerja *Publish/Subscribe* dengan protokol AMQP dan MQTT

Dalam Subbab ini akan dijelaskan kinerja metode *publish/subscriber* dengan dua protokol yang berbeda yaitu protokol AMQP dan protokol MQTT

4.2.1 Kinerja *Publish/Subscribe* dengan protokol MQTT

Pada bagian ini akan dijelaskan sistematika kerja metode *publish/subscribe* dengan menggunakan protokol MQTT. Penjelasan akan dilakukan dengan menggunakan *traffic* data yang telah di *capture* menggunakan *wireshark*.

54	23:50:28.422	192.168.1.49	54.175.185.219	MQTT	105 Connect Command
58	23:50:28.718	54.175.185.219	192.168.1.49	MQTT	70 Connect Ack
60	23:50:28.718	192.168.1.49	54.175.185.219	MQTT	78 Subscribe Request
66	23:50:28.907	192.168.1.49	54.175.185.219	MQTT	105 Connect Command
67	23:50:29.007	54.175.185.219	192.168.1.49	MQTT	71 Subscribe Ack
71	23:50:29.179	54.175.185.219	192.168.1.49	MQTT	70 Connect Ack
73	23:50:29.179	192.168.1.49	54.175.185.219	MQTT	76 Subscribe Request
76	23:50:29.396	192.168.1.49	54.175.185.219	MQTT	105 Connect Command
77	23:50:29.459	54.175.185.219	192.168.1.49	MQTT	71 Subscribe Ack
83	23:50:29.680	54.175.185.219	192.168.1.49	MQTT	70 Connect Ack
85	23:50:29.680	192.168.1.49	54.175.185.219	MQTT	80 Subscribe Request
90	23:50:29.952	192.168.1.49	54.175.185.219	MQTT	105 Connect Command
91	23:50:29.964	54.175.185.219	192.168.1.49	MQTT	71 Subscribe Ack
97	23:50:30.233	54.175.185.219	192.168.1.49	MQTT	70 Connect Ack
99	23:50:30.234	192.168.1.49	54.175.185.219	MQTT	76 Subscribe Request
100	23:50:30.522	54.175.185.219	192.168.1.49	MQTT	71 Subscribe Ack
161	23:50:39.309	192.168.1.49	54.175.185.219	MQTT	96 Connect Command
168	23:50:39.596	54.175.185.219	192.168.1.49	MQTT	70 Connect Ack

Gambar 4.1 *Capture Traffic* data komunikasi data *publish/subscribe*

Gambar 4.1 adalah hasil rekam data komunikasi data menggunakan *publish/subscriber*. Bisa dilihat diatas terdapat beberapa paket data yang berhasil direkam. Paket diatas tersebut menjelaskan proses koneksi *subscriber* ke *broker*.

No.	Time	Source	Destination	Protocol	Info	Length
54	23:50:28.422	192.168.1.49	54.175.185.219	MQTT	Connect Command	105
58	23:50:28.718	54.175.185.219	192.168.1.49	MQTT	Connect Ack	70
60	23:50:28.718	192.168.1.49	54.175.185.219	MQTT	Subscribe Request	78
66	23:50:28.907	192.168.1.49	54.175.185.219	MQTT	Connect Command	105
67	23:50:29.007	54.175.185.219	192.168.1.49	MQTT	Subscribe Ack	71

Gambar 4.2 Proses koneksi subscriber ke broker

Pada gambar 4.2 dijelaskan proses koneksi *subscriber* ke *broker*. Sebuah *publisher* untuk melakukan koneksi ke *broker* hal yang dilakukan sebelumnya adalah mengirimkan permintaan untuk melakukan koneksi, hal itu terjadi pada paket data no 54. Setelah melakukan permintaan koneksi, *broker* akan membalas dengan mengirimkan *ack* ke *ip* yang melakukan permintaan untuk koneksi hal itu terjadi pada paket data no 58. Setelah koneksi terbentuk, *subscriber* akan meminta ijin untuk melakukan *subscriber* ke topic yang sudah disediakan oleh broker setelah ijin disetujui, *broker* akan mengirimkan *ack* kembali ke *subscriber* untuk menyatakan bahwa *subscribe* telah terbentuk. Proses itu terjadi pada paket data nomor 60 dan 67. Untuk paket data nomor 66 untuk koneksi *subscriber* lainnya.

161	23:50:39.309	192.168.1.49	54.175.185.219	MQTT	Connect Command	96
168	23:50:39.596	54.175.185.219	192.168.1.49	MQTT	Connect Ack	70
171	23:50:39.657	192.168.1.49	54.175.185.219	MQTT	Publish Message	84
175	23:50:39.945	54.175.185.219	192.168.1.49	MQTT	Publish Ack	70
176	23:50:39.945	54.175.185.219	192.168.1.49	MQTT	Publish Message	82

Gambar 4.3 Proses publish data

Gambar 4.3 menjelaskan proses *publish* data pada protokol MQTT. Sebelum melakukan pengiriman data, *publisher* akan melakukan koneksi dulu ke *broker*. Proses untuk melakukan koneksi sama dengan proses koneksi dari *subscriber* ke *broker*. Setelah koneksi terbentuk, *publisher* akan langsung mengirimkan datanya ke *broker* setelah data terkirim, *broker* akan memberikan *ack* ke *publisher* untuk memberitahu bahwa data telah sampai ke broker, proses tersebut terjadi pada paket data nomor 171 dan 175. Untuk paket data nomor 176 adalah proses pengiriman data dari *broker* ke *subscriber*.

4.2.2 Kinerja *Publish/Subscribe* dengan protokol AMQP

Pada bagian ini akan dijelaskan sistematika kerja metode *publish/subscribe* dengan menggunakan protokol AMQP. Penjelasan akan dilakukan dengan menggunakan *traffic data* yang telah di *capture* menggunakan *wireshark*.

No.	Time	Source	Destination	Protocol	Info	Length
51	18:08:32.646	192.168.1.48	54.175.185.219	AMQP	Protocol-Header 0-9-1	74
54	18:08:32.922	54.175.185.219	192.168.1.48	AMQP	Connection.Start	576
56	18:08:32.986	192.168.1.48	54.175.185.219	AMQP	Connection.Start-Ok	453
58	18:08:33.262	54.175.185.219	192.168.1.48	AMQP	Connection.Tune	86
60	18:08:33.270	192.168.1.48	54.175.185.219	AMQP	Connection.Tune-Ok	86
61	18:08:33.271	192.168.1.48	54.175.185.219	AMQP	Connection.Open vhost=/ /	82
63	18:08:33.545	54.175.185.219	192.168.1.48	AMQP	Connection.Open-Ok	79
64	18:08:33.558	192.168.1.48	54.175.185.219	AMQP	Channel.Open	79
68	18:08:33.838	54.175.185.219	192.168.1.48	AMQP	Channel.Open-Ok	82
69	18:08:33.841	192.168.1.48	54.175.185.219	AMQP	Queue.Declare q=listrik	93
72	18:08:34.115	54.175.185.219	192.168.1.48	AMQP	Queue.Declare-Ok q=listrik	94
73	18:08:34.123	192.168.1.48	54.175.185.219	AMQP	Basic.Publish x= rk=listrik ...	134
74	18:08:34.124	192.168.1.48	54.175.185.219	AMQP	Basic.Publish x= rk=listrik ...	134
75	18:08:34.124	192.168.1.48	54.175.185.219	AMQP	Basic.Publish x= rk=listrik ...	134
76	18:08:34.125	192.168.1.48	54.175.185.219	AMQP	Basic.Publish x= rk=listrik ...	202
77	18:08:34.125	192.168.1.48	54.175.185.219	AMQP	Channel.Close reply=OK	87
81	18:08:34.400	54.175.185.219	192.168.1.48	AMQP	Channel.Close-Ok	78
82	18:08:34.406	192.168.1.48	54.175.185.219	AMQP	Connection.Close reply=OK	87
86	18:08:34.680	54.175.185.219	192.168.1.48	AMQP	Connection.Close-Ok	78

Gambar 4.4 Proses pengiriman data dari *publisher* ke *broker*

Gambar 4.4 adalah hasil rekam paket pengiriman data menggunakan metode *publish/subscribe* dengan menggunakan protokol AMQP. Dari gambar diatas dapat dijelaskan bahwa sebelum melakukan pengiriman data, *publisher* akan mengirim header ke broker. Setelah header dikirim, *broker* akan memberikan respon yang menyatakan untuk memulai koneksi ke *publisher*. Setelah *publisher* mendapatkan respon dari *broker*, *publisher* akan langsung mengirim respon balik ke *broker*. Setelah koneksi antar *publisher* ke *broker* terbangun, *publisher* akan meminta *broker* untuk membuka koneksi ke vhost yang ada di *broker*. Setelah koneksi terbentuk dan vhost dengan nama “/” terbuka, tahap selanjutnya *publisher* akan meminta membuka saluran (*channel*) yang ada di vhost “/”.

Setelah *channel* terbuka, *publisher* akan mendeklarasikan *queue* yang akan disimpan oleh *broker* yang nantinya akan di *subscribe* oleh *subscriber*. Setelah *queue* terdeklarasikan, *publisher* langsung melakukan pengiriman data ke *broker* proses tersebut terdapat pada paket data nomor 73 sampai 76. Setelah data terkirim, *publisher* langsung mengirimkan permintaan untuk menutup *channel* dan koneksi ke broker.

No.	Time	Source	Destination	Protocol	Info	Length
290	18:08:47.209	192.168.1.48	54.175.185.219	AMQP	Protocol-Header 0-9-1	74
295	18:08:47.460	54.175.185.219	192.168.1.48	AMQP	Connection.Open-Ok	79
296	18:08:47.477	192.168.1.48	54.175.185.219	AMQP	Channel.Open	79
298	18:08:47.507	54.175.185.219	192.168.1.48	AMQP	Connection.Start	576
301	18:08:47.582	192.168.1.48	54.175.185.219	AMQP	Connection.Start-Ok	453
304	18:08:47.649	192.168.1.48	54.175.185.219	AMQP	Protocol-Header 0-9-1	74
307	18:08:47.756	54.175.185.219	192.168.1.48	AMQP	Channel.Open-Ok	82
308	18:08:47.757	192.168.1.48	54.175.185.219	AMQP	Queue.Declare q=alarm_kebakaran	101
311	18:08:47.877	54.175.185.219	192.168.1.48	AMQP	Connection.Tune	86
313	18:08:47.889	192.168.1.48	54.175.185.219	AMQP	Connection.Tune-Ok	86
314	18:08:47.890	192.168.1.48	54.175.185.219	AMQP	Connection.Open vhost=/	82
316	18:08:47.923	54.175.185.219	192.168.1.48	AMQP	Connection.Start	576
318	18:08:48.001	192.168.1.48	54.175.185.219	AMQP	Connection.Start-Ok	453
319	18:08:48.035	54.175.185.219	192.168.1.48	AMQP	Queue.Declare-Ok q=alarm_kebakara...	102
320	18:08:48.039	192.168.1.48	54.175.185.219	AMQP	Basic.Consume q=alarm_kebakaran	102
323	18:08:48.094	192.168.1.48	54.175.185.219	AMQP	Protocol-Header 0-9-1	74
325	18:08:48.184	54.175.185.219	192.168.1.48	AMQP	Connection.Open-Ok	79
326	18:08:48.201	192.168.1.48	54.175.185.219	AMQP	Channel.Open	79
328	18:08:48.276	54.175.185.219	192.168.1.48	AMQP	Connection.Tune	86
330	18:08:48.286	192.168.1.48	54.175.185.219	AMQP	Connection.Tune-Ok	86
331	18:08:48.287	192.168.1.48	54.175.185.219	AMQP	Connection.Open vhost=/	82
332	18:08:48.318	54.175.185.219	192.168.1.48	AMQP	Basic.Consume-Ok Basic.Deliver x=...	650
335	18:08:48.367	54.175.185.219	192.168.1.48	AMQP	Connection.Start	576

Gambar 4.5 Proses pengiriman data dari *broker* ke *subscriber*

Gambar 4.5 menjelaskan proses *subscribing* yang terjadi antara *subscriber* dan *broker*. Untuk paket data nomor 290, 295, 296, 298, 301, adalah proses awal untuk melakukan koneksi dari *subscriber* ke *broker*. Setelah permintaan pembukaan *channel* dan koneksi yang dilakukan oleh *subscriber* disetujui oleh *broker*, tahap selanjutnya adalah *subscriber* akan mendeklarasikan *queue* yang akan ditujunya, maksudnya *subscriber* akan melakukan *subscribe* ke topik yang sudah deklarasikan oleh *publisher* sebelumnya.

Setelah melakukan deklarasi, *subscriber* akan meminta kepada *broker* untuk dibuka *vhost* dengan nama "/" setelah *vhost* terbuka, *broker* akan membuka koneksi ke *subscriber* dan *subscriber* akan memberikan respon kalau koneksi sudah terhubung. Untuk paket data nomor 319 adalah respon dari *broker* ke *subscriber* bahwa *queue* yang di deklarasikan sebelumnya bisa digunakan dan diambil datanya. Proses selanjutnya adalah setelah mendapatkan respon dari *broker*, *subscriber* akan langsung mengambil data (*consume*) dari *broker* setelah itu akan terulang proses menjalin koneksi antara *broker* dan *subscriber*, setelah proses ini *broker* akan memberikan respon bahwa data bisa diambil dari *queue*.

4.3 Skenario Pengujian

Dalam pengujian ini akan ada 2 skenario pengujian. Skenario pertama adalah pengiriman 5 tahap data (100 data, 200 data, 300 data, 400 data, 500 data) dan merekam data hasil pengujian dari sisi *publisher* ke *broker*, dari sisi *broker* ke *subscriber*, dan dari sisi *publisher* ke *subscriber* dan dengan menggunakan protokol AMQP. Untuk Skenario kedua adalah pengiriman 5 tahap data (100 data, 200 data, 300 data, 400 data, 500 data) dan merekam data hasil pengujian dari sisi *publisher* ke *broker*, dari sisi *broker* ke *subscriber*, dan dari sisi *publisher* ke *subscriber* dan menggunakan protokol MQTT. Masing-masing tahap akan dilakukan pengujian sebanyak 10 kali dan data dari 10 kali pengujian tersebut nantinya akan diambil rata-ratanya untuk dijadikan data *final* dan akan ditarik kesimpulan dari data *final* tersebut.

Data yang diambil dalam setiap pengujian adalah waktu lama pengiriman data dari posisi awal sampai ke posisi akhirnya, paket data nantinya akan di *capture* menggunakan wireshark dan TShark. Posisi untuk *tapping* paket data akan dilakukan di dua sisi yaitu pada sisi *publisher* dan *subscriber* (laptop yang digunakan penguji untuk melakukan percobaan) dan pada sisi *broker* (*cloud vps*).

Wireshark akan digunakan untuk *tapping* pada sisi *pub* dan *sub* sedangkan TShark untuk sisi *broker*. TShark adalah wireshark berbasis *command prompt* disini penguji menggunakan TShark untuk *capture* data pada sisi *broker* dikarenakan *broker* yang dipakai menggunakan OS *console base* bukan *desktop base*. Untuk *command script* Tshark yang dijalankan di sisi *broker* adalah :

```
" sudo tshark -i eth 0 -w /var/tmp capture.pcap"
```

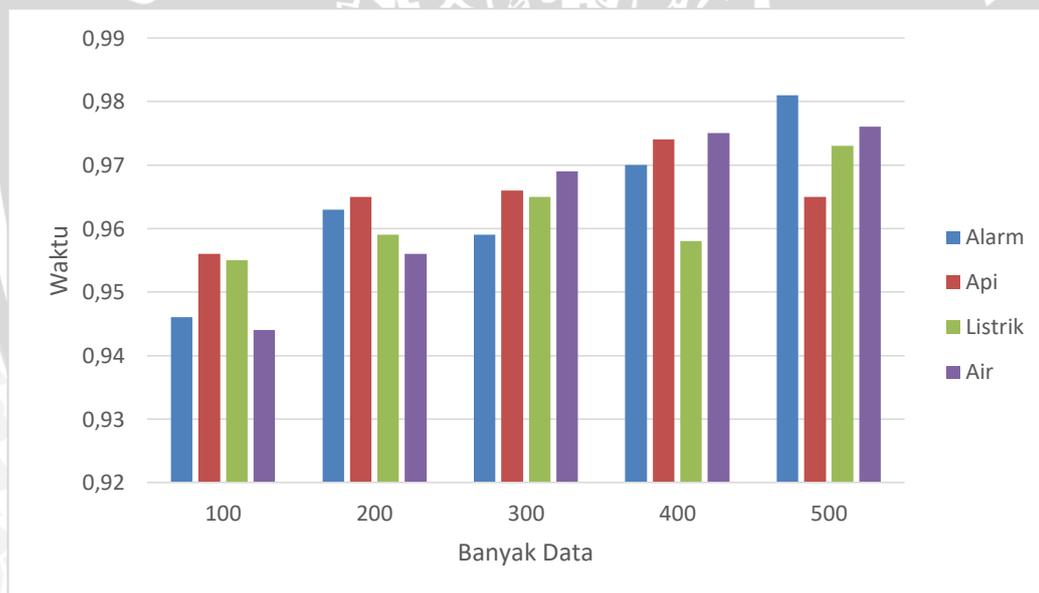


4.3.1 Hasil Pengujian Menggunakan RabbitMQ (AMQP)

Setelah dilakukan pengujian sebanyak 5 tahap dan masing-masing tahap dilakukan percobaan sebanyak 10 kali, hasil data yang di dapat di masing-masing tahap dengan 10 kali percobaan akan diambil rata-ratanya setelah di ambil rata-ratanya, akan di ambil kesimpulan dari data *final* tersebut. Berikut ini adalah data *final* yang telah diambil rata-rata dari semua tahap pengujian diatas

Tabel 4.1 Tabel waktu pengiriman data dari sisi *publisher* ke *broker*

Jumlah paket	Topik(second)			
	Alarm	Api	Listrik	Air
100	0,946	0,956	0,955	0,944
200	0,963	0,965	0,959	0,956
300	0,959	0,966	0,965	0,969
400	0,970	0,974	0,958	0,975
500	0,981	0,965	0,973	0,976

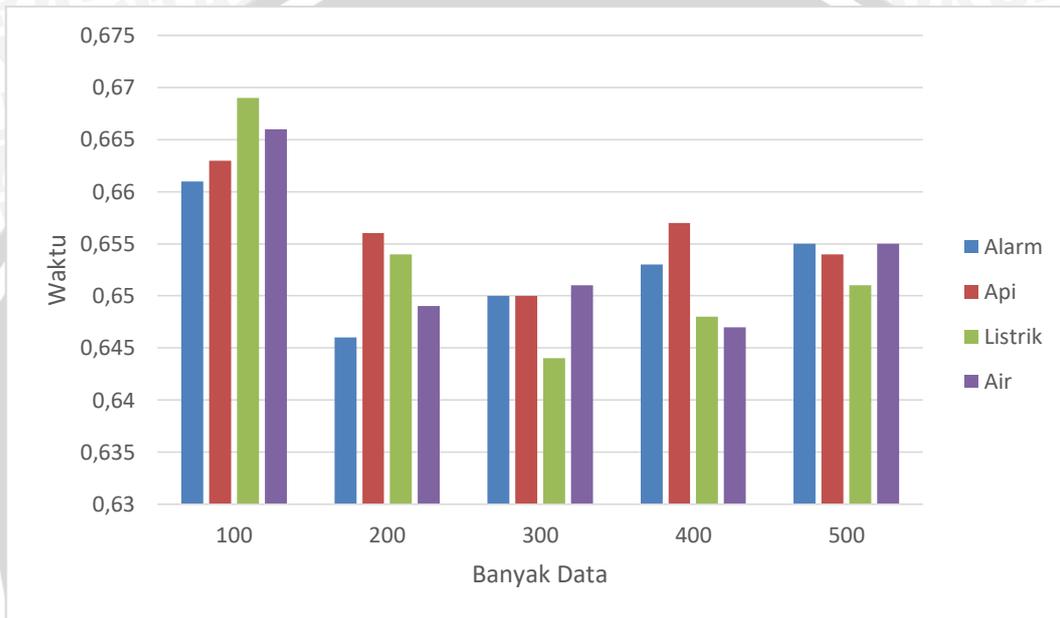


Gambar 4.6 Grafik waktu pengiriman data dari *Publisher* ke *Broker*

Gambar diatas adalah grafik rata-rata dari 10 kali percobaan waktu pengiriman data dari sisi *publisher* ke *broker* Dari grafik diatas dapat dilihat bahwa waktu pengiriman data dari sisi *publisher* ke *broker* terjadi naik turun antar topik namun terjadi peningkatan untuk setiap tahap data yang dikirimkan. Kecuali pada topic listrik pada tahap 400 data mengalami penurunan namun tidak terlalu jauh.

Tabel 4.2 Tabel waktu pengiriman data dari *Broker* ke *Subscriber*

Jumlah paket	Topik(second)			
	Alarm	Api	Listrik	Air
100	0,661	0,663	0,669	0,666
200	0,646	0,656	0,654	0,649
300	0,650	0,650	0,644	0,651
400	0,653	0,657	0,648	0,647
500	0,655	0,654	0,651	0,655

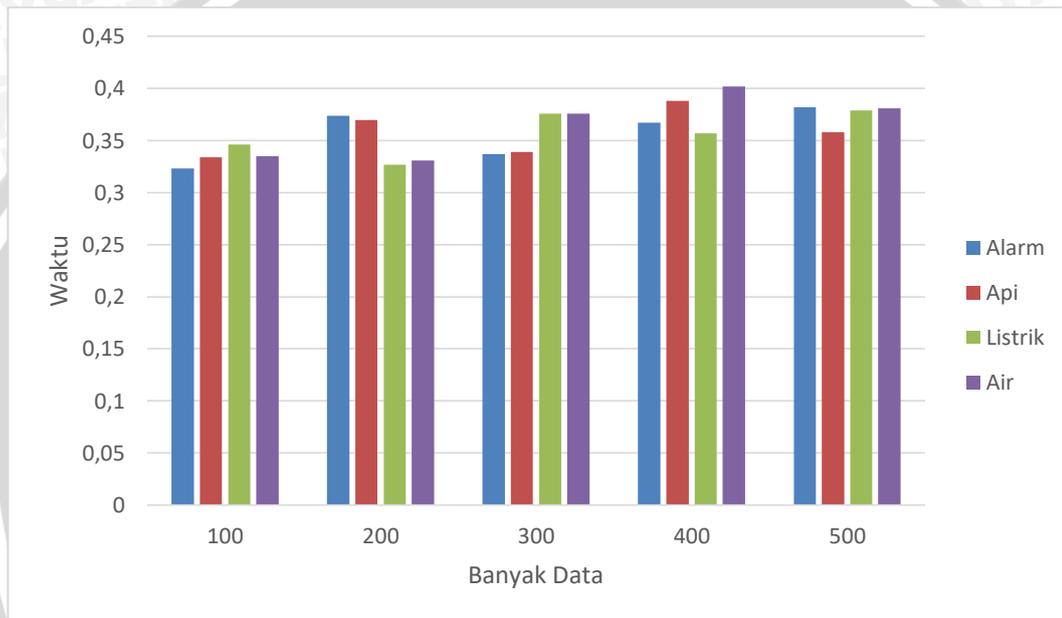


Gambar 4.7 Grafik waktu pengiriman data dari *broker* ke *subscriber*

Untuk gambar diatas adalah gambar tentang grafik rata-rata waktu pengiriman data dari 10 kali percobaan dari sisi *broker* ke *subscriber*. Dari grafik diatas dapat diliha bahwa waktu pengiriman data pada tahap pertama yaitu 100 data memiliki waktu lebih lama daripada tahap lainnya. Untuk tahap 500 data adalah tahap dimana waktu pengiriman hampir stabil untuk setiap topiknya.

Tabel 4.3 Tabel waktu pengiriman data dari *Publisher* ke *Subscriber*

Jumlah paket	Topik(second)			
	Alarm	Api	Listrik	Air
100	0,323	0,334	0,346	0,335
200	0,374	0,370	0,327	0,331
300	0,337	0,339	0,376	0,376
400	0,367	0,388	0,357	0,402
500	0,382	0,358	0,379	0,381



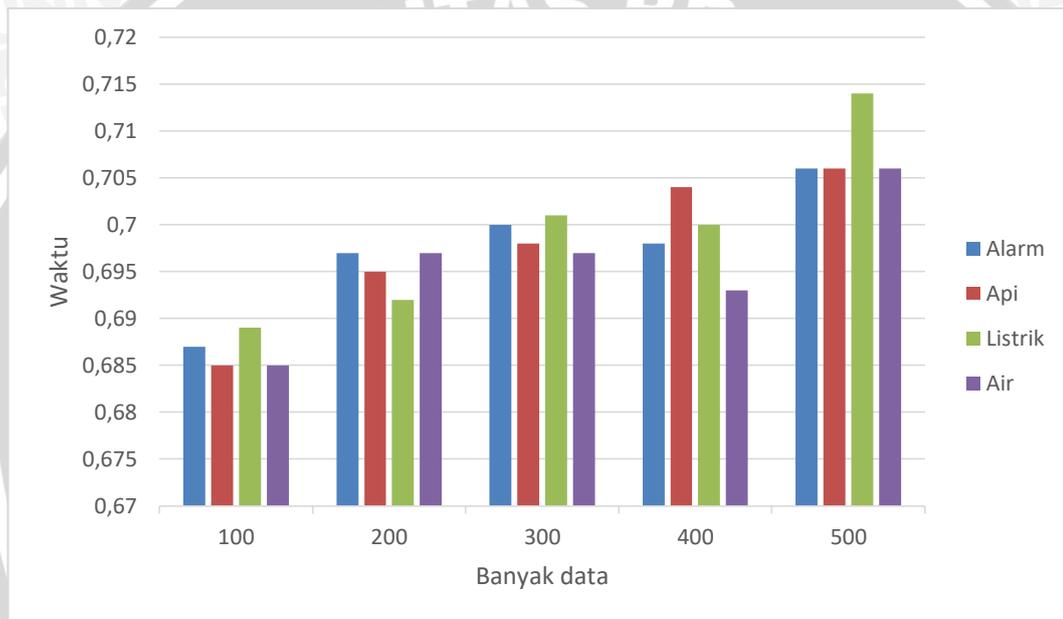
Gambar 4.8 Grafik waktu pengiriman data dari *Publisher* ke *Subscriber*

Grafik diatas menunjukkan tentang rata-rata waktu pengiriman data dari 10 kali percobaan dari sisi *publisher* ke *subscriber*. Dari grafik diatas dapat dilihat bahwa waktu pengiriman data antar topik untuk setiap tahap bisa dikatakan hampir stabil. Tidak ada perbedaan yang mencolok dalam waktu pengiriman data antar topiknya.

4.3.2 Hasil Pengujian Menggunakan Mosquitto (MQTT)

Tabel 4.4 Tabel waktu pengiriman data dari sisi *Publisher* ke *Broker*

Jumlah paket	Topik(second)			
	Alarm	Api	Listrik	Air
100	0,687	0,685	0,689	0,685
200	0,697	0,695	0,692	0,697
300	0,700	0,698	0,701	0,697
400	0,698	0,704	0,700	0,693
500	0,706	0,706	0,714	0,706

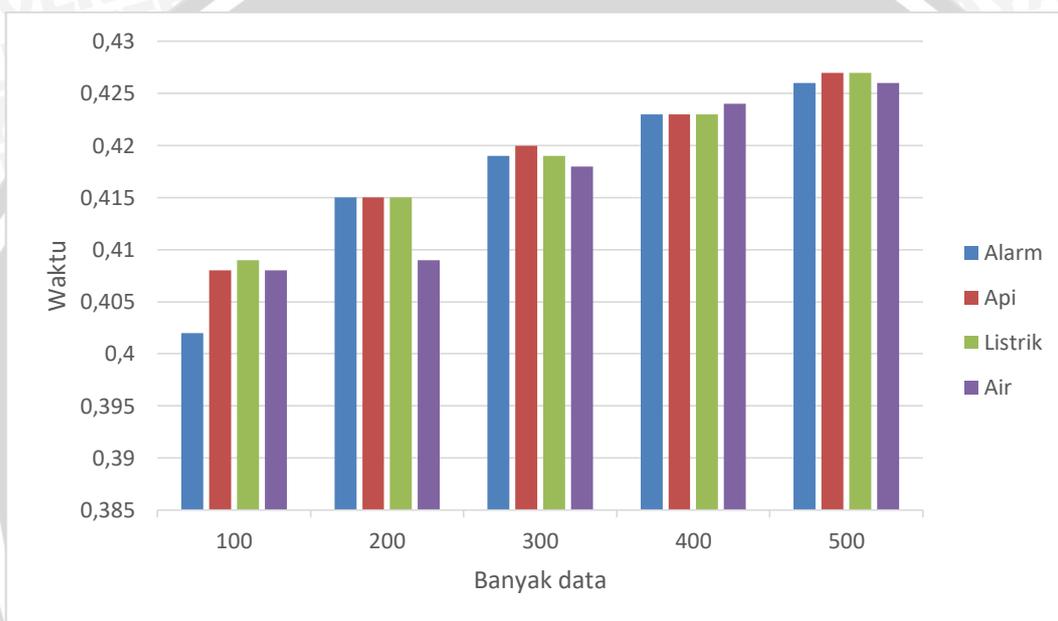


Gambar 4.9 Grafik waktu pengiriman data dari sisi *Publisher* ke *Broker*

Grafik diatas menunjukkan tentang waktu pengiriman data dari sisi *publisher* ke *broker*. Dari grafik diatas dapat diketahui bahwa waktu pengiriman data meningkat seiring dengan banyak data yang dikirimkan. Peningkatan waktu terjadi pada ketiga topic dan untuk topic air mengalami penurunan pada tahap 400 data namun penurunannya tidak terlalu besar.

Tabel 4.5 Tabel waktu pengiriman data dari *Broker* ke *Subscriber*

Jumlah paket	Topik(second)			
	Alarm	Api	Listrik	Air
100	0,402	0,408	0,409	0,408
200	0,415	0,415	0,415	0,409
300	0,419	0,420	0,419	0,418
400	0,423	0,423	0,423	0,424
500	0,426	0,427	0,427	0,426

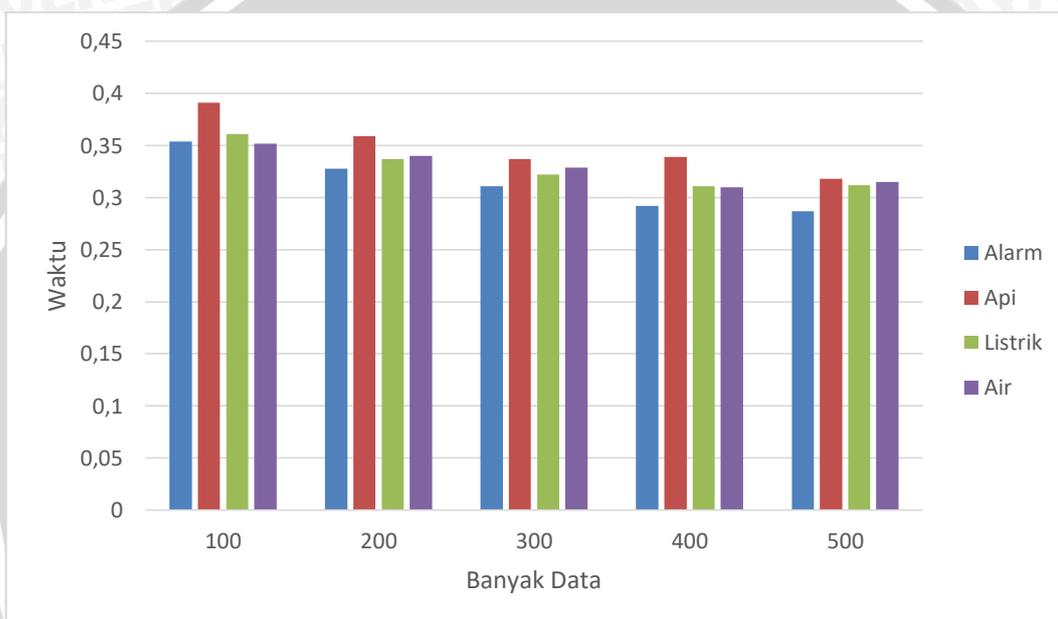


Gambar 4.10 Grafik waktu pengiriman data dari sisi *broker* ke *subscriber*

Grafik diatas adalah grafik tentang waktu pengiriman data dari 10 kali percobaan pengiriman dari dari sisi *broker* ke *subscriber*. Dari grafik diatas dapat dikatakan bahwa waktu pengiirman data meningkat seiring dengan banyaknya data yang dikirimkan begitu juga untuk setiap topiknya mengalami peningkatan seiring banyaknya data yang dikirimkan

Tabel 4.6 Tabel waktu pengiriman data dari *Publisher* ke *Subscriber*

Jumlah paket	Topik(second)			
	Alarm	Api	Listrik	Air
100	0,354	0,391	0,361	0,352
200	0,328	0,359	0,337	0,340
300	0,311	0,337	0,322	0,329
400	0,292	0,339	0,311	0,310
500	0,287	0,318	0,312	0,315



Gambar 4.11 Grafik waktu pengiriman data dari *Publisher* ke *Subscriber*

Grafik diatas menunjukkan tentang waktu rata-rata dari 10 kali percobaan pengiriman data dari sisi *publisher* ke *broker*. Dari grafik diatas dapat dilihat bahwa waktu pengiriman data dari sisi *publisher* ke *broker* menggunakan protokol MQTT mengalami penurunan dapat dikatakan semakin banyaknya data waktu yang dibutuhkan untuk mengirim semakin sedikit dan dari grafik juga dapat dilihat penurunan waktunya bisa dikatakan stabil.

BAB 5 PENUTUP

5.1 Kesimpulan

Setelah melakukan perancangan simulasi, implementasi metode, pengujian dan analisa hasil langkah selanjutnya adalah pengambilan kesimpulan dari semua langkah yang sudah dilakukan tadi. Kesimpulan yang dapat diambil yaitu:

1. Simulasi data yang dibangun untuk komunikasi data antar perangkat di lingkungan *smart home* mempunyai beberapa komponen pembentuknya yaitu *publisher*, *subscriber*, dan *broker*. Rancangannya adalah dengan menggunakan 4 *publisher* yaitu alarm kebakaran, meteran listrik, meteran air, dan alarm keamanan, 1 *broker*, dan 4 *subscriber* yang masing-masing berlangganan ke setiap topik yang disediakan oleh *broker*. Untuk 4 *publisher* di atas masing-masing mempunyai struktur data sendiri dan format data yang dikirimkan. Untuk *broker* diletakkan di cloud vps dan untuk *publisher* dan *subscriber* berada didalam jaringan local dan dalam satu perangkat. Jadi dalam satu perangkat yang dipakai untuk pengujian nanti terdapat 4 *publisher* sebagai *sender* dan 4 *subscriber* sebagai *receiver*.
2. *Broker* RabbitMQ yang menggunakan protokol AMQP (*Advanced Message Queue Protokol*), hasil pengujian yang penguji lakukan dilihat dari sisi *publisher* ke *broker* dapat disimpulkan bahwa waktu pengiriman data akan mengalami peningkatan sesuai dengan semakin banyaknya data yang dikirim. Untuk sisi *broker* ke *subscriber* waktu pengiriman data mengalami penurunan pada tahap 200, 300, 400 dan mulai stabil pada tahap 500. Untuk sisi *publisher* ke *subscriber* memiliki waktu pengiriman data hampir sama untuk di setiap tahapnya
3. Untuk *Broker* Mosquitto yang menggunakan protokol MQTT (*Message Queue Telemetry Transport*) dapat disimpulkan waktu pengiriman data dari *publisher* ke *subscriber* meningkat seiring dengan banyaknya data yang dikirimkan begitu pula untuk sisi *broker* ke *subscriber*. Untuk sisi *publisher* ke *subscriber* waktu pengiriman datanya mengalami penurunan seiring dengan banyaknya data yang dikirimkan.

5.2 Saran

Saran yang diberikan untuk penelitian ini untuk pengembangan selanjutnya adalah :

1. Saran pertama untuk peletakan posisi *publisher* dan *subscriber* sebaiknya pada perangkat yang berbeda agar bisa terlihat jelas perbedaan untuk proses *publish* pesan dan proses *subscriber* pesan
2. Penambahan parameter pengujian seperti contohnya pengaruh level QoS terhadap lama pengiriman, delay pengiriman data setiap paket yang dikirimkan kemampuan maksimal melakukan pengiriman pesan, dan parameter-parameter lainnya.
3. Penambahan scenario pengujian dimana ada lebih dari satu *subscriber* berlangganan ke topik yang sama.



DAFTAR PUSTAKA

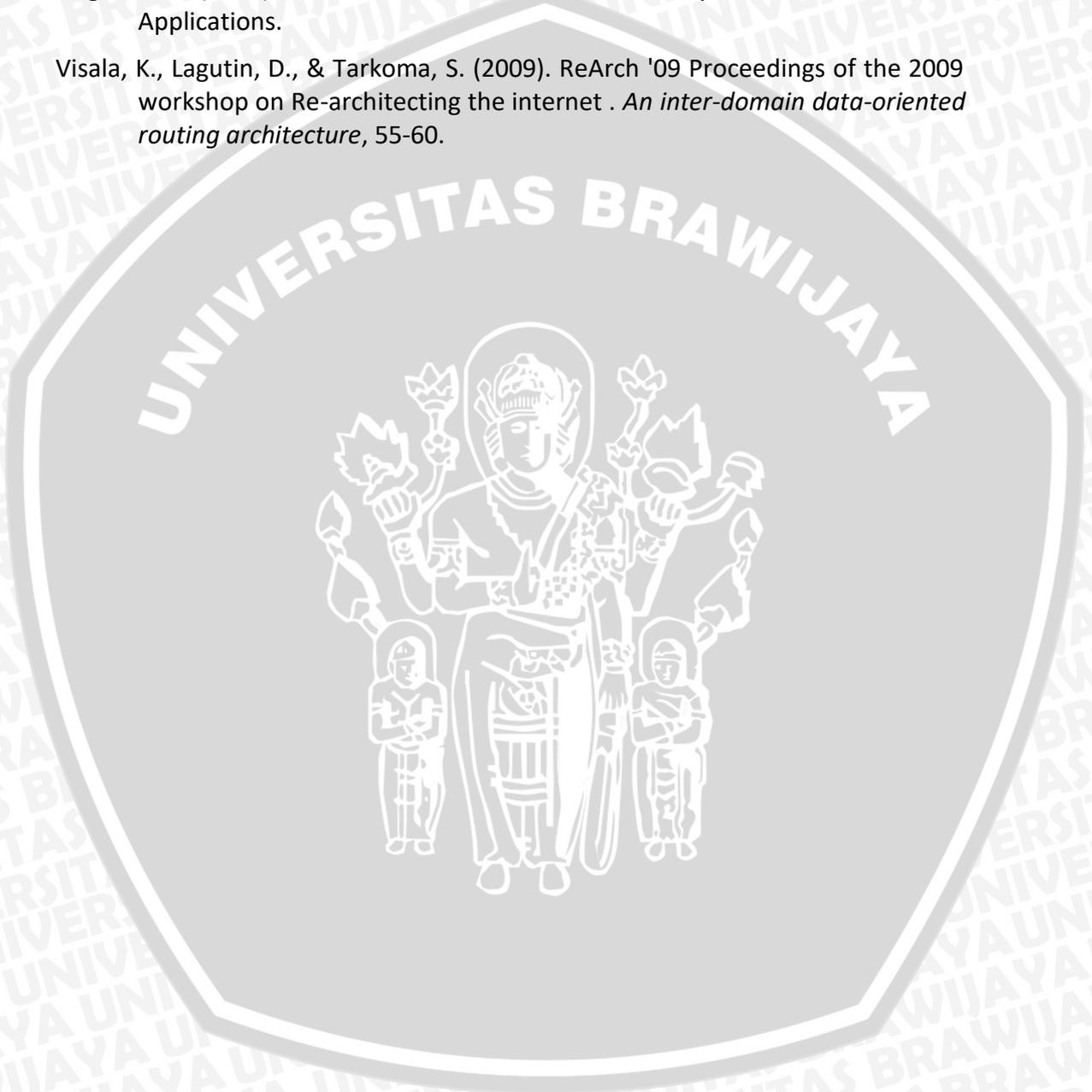
- Aiyagari, S., Arrot, M., Atwell, M., & Broner, J. (2008, November 13). Protocol Specification. *AMQP Advance Message Queuing Protocol Version 0-9-1*, 1-39.
- Electronic Design*. (2016, Mei 07). Retrieved from Electronic Design: <http://electronicdesign.com/site-files/electronicdesign.com/files/uploads/2013/04/0502WTDrtiFig1.gif>
- Entwickler.de*. (2016, Mei 07). Retrieved from *entwickler.de*: https://entwickler.de/wp-content/uploads/2014/07/goetz_mqtt_1.jpg
- Forouzan, B. A. (2007). *Data Communication and Networking*. New York, America: McGraw-Hill.
- Gerhart, J. (1999). *Home Automation and Wiring*. (1st, Ed.) New York, United States of America: Mc Graw-Hill Profesional.
- Harper, R. (2003). *Inside the Smart Home*. London: Springer.
- Hunkeler, U., Truong, H. L., & Stanford-Clark, A. (2008, January). Communication Systems Software and Middleware and Workshops. *MQTT-S – A Publish/Subscribe Protocol For Wireless Sensor Network*, 791-798.
- Kurana, H. (2005). SAC '05 Proceedings of the 2005 ACM Symposium on Applied Computing. *Scalable Security and Accounting Services for Content-Based Publish/Subscribe Systems*, 801-807.
- Lagutin, D., Visala, K., Zahmesky, A., Burbridge, T., & Marias, G. F. (2010, June). Computer and Communication (ISCC). *Roles and Security in a Publish/Subscribe Network Arcechure*, 68-74.
- Lalanda, P., Bourcier, J., J.Bardin, & S.Chollet. (2010, Februari). Smart Home Systems. *Smat Home Systems, I*, 001.
- Liu, L., & Ozu, M. (2009). *Encyclopedia of Database System*. New York, United States of America: Springer.
- Luzuriaga, J. E., Perez, M., Boronat, P., Cano, J. C., Calafate, C., & Manzoni, P. (2015). A comparative evaluation of AMQP and MQTT. *2015 IEEE 12th Consumer Communications and Networking Conference (CCNC): CCNC 2015 Workshops - VENITS*, 931-936.
- Marshall, D., Reynolds, W. A., & McCrory, D. (2006). *Advanced Server Virtualization*. Florida, United States of America: Auerbach Publications.

MQTT Essentials Part 6: Quality of Service 0, 1 & 2. (2016, April 04). Retrieved from HiveMQ Enterprise MQTT Broker: <http://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels>

Tarkoma, S. (2012). *Publish/Subscribe System Design and Principles*. West Sussex, United Kingdom: John Wiley and Sons Ltd.

Virgillito, A. (2003). *Publish/Subscribe Communication System: from Models to Applications*.

Visala, K., Lagutin, D., & Tarkoma, S. (2009). ReArch '09 Proceedings of the 2009 workshop on Re-architecting the internet . *An inter-domain data-oriented routing architecture*, 55-60.



LAMPIRAN A LAMPIRAN SKRIPSI

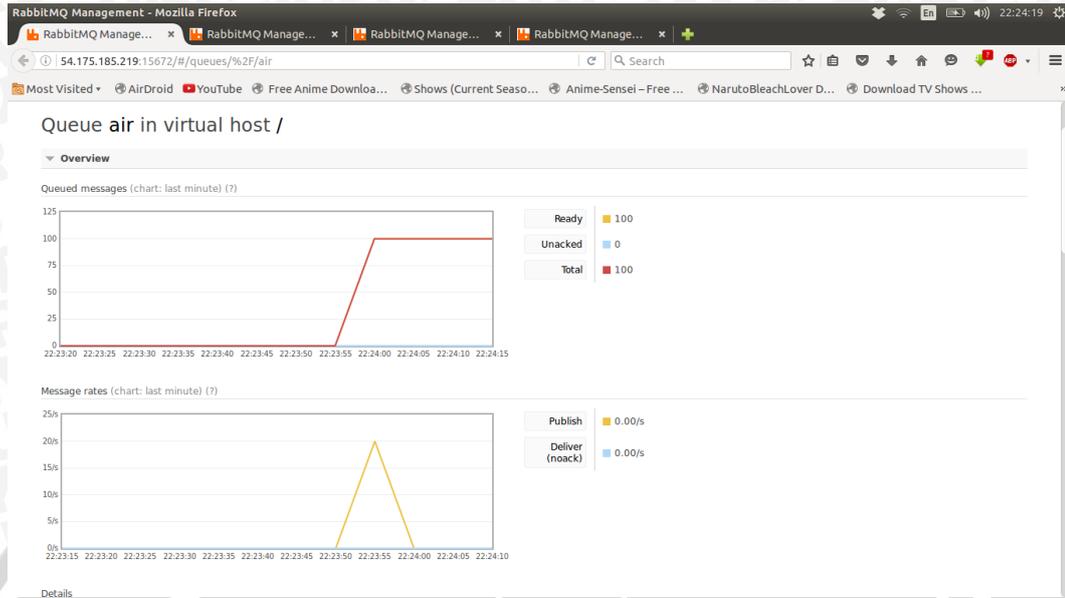
A.1 Capture paket data protokol MQTT dengan Wireshark

No.	Time	Source	Destination	Protocol	Info	Length
54	23:50:28.422	192.168.1.49	54.175.185.219	MQTT	Connect Command	105
58	23:50:28.718	54.175.185.219	192.168.1.49	MQTT	Connect Ack	70
60	23:50:28.718	192.168.1.49	54.175.185.219	MQTT	Subscribe Request	78
66	23:50:28.907	192.168.1.49	54.175.185.219	MQTT	Connect Command	105
67	23:50:29.007	54.175.185.219	192.168.1.49	MQTT	Subscribe Ack	71
71	23:50:29.179	54.175.185.219	192.168.1.49	MQTT	Connect Ack	70
73	23:50:29.179	192.168.1.49	54.175.185.219	MQTT	Subscribe Request	76
76	23:50:29.396	192.168.1.49	54.175.185.219	MQTT	Connect Command	105
77	23:50:29.459	54.175.185.219	192.168.1.49	MQTT	Subscribe Ack	71
83	23:50:29.680	54.175.185.219	192.168.1.49	MQTT	Connect Ack	70
85	23:50:29.680	192.168.1.49	54.175.185.219	MQTT	Subscribe Request	80
90	23:50:29.952	192.168.1.49	54.175.185.219	MQTT	Connect Command	105
91	23:50:29.964	54.175.185.219	192.168.1.49	MQTT	Subscribe Ack	71
97	23:50:30.233	54.175.185.219	192.168.1.49	MQTT	Connect Ack	70
99	23:50:30.234	192.168.1.49	54.175.185.219	MQTT	Subscribe Request	76
100	23:50:30.522	54.175.185.219	192.168.1.49	MQTT	Subscribe Ack	71
161	23:50:39.309	192.168.1.49	54.175.185.219	MQTT	Connect Command	96
168	23:50:39.596	54.175.185.219	192.168.1.49	MQTT	Connect Ack	70
171	23:50:39.657	192.168.1.49	54.175.185.219	MQTT	Publish Message	84
175	23:50:39.945	54.175.185.219	192.168.1.49	MQTT	Publish Ack	70
176	23:50:39.945	54.175.185.219	192.168.1.49	MQTT	Publish Message	82
179	23:50:39.978	192.168.1.49	54.175.185.219	MQTT	Publish Message	84
182	23:50:40.264	54.175.185.219	192.168.1.49	MQTT	Publish Message	82
184	23:50:40.264	54.175.185.219	192.168.1.49	MQTT	Publish Ack	70
186	23:50:40.321	192.168.1.49	54.175.185.219	MQTT	Publish Message	84
190	23:50:40.607	54.175.185.219	192.168.1.49	MQTT	Publish Message	82
192	23:50:40.608	54.175.185.219	192.168.1.49	MQTT	Publish Ack	70
194	23:50:40.642	192.168.1.49	54.175.185.219	MQTT	Publish Message	84

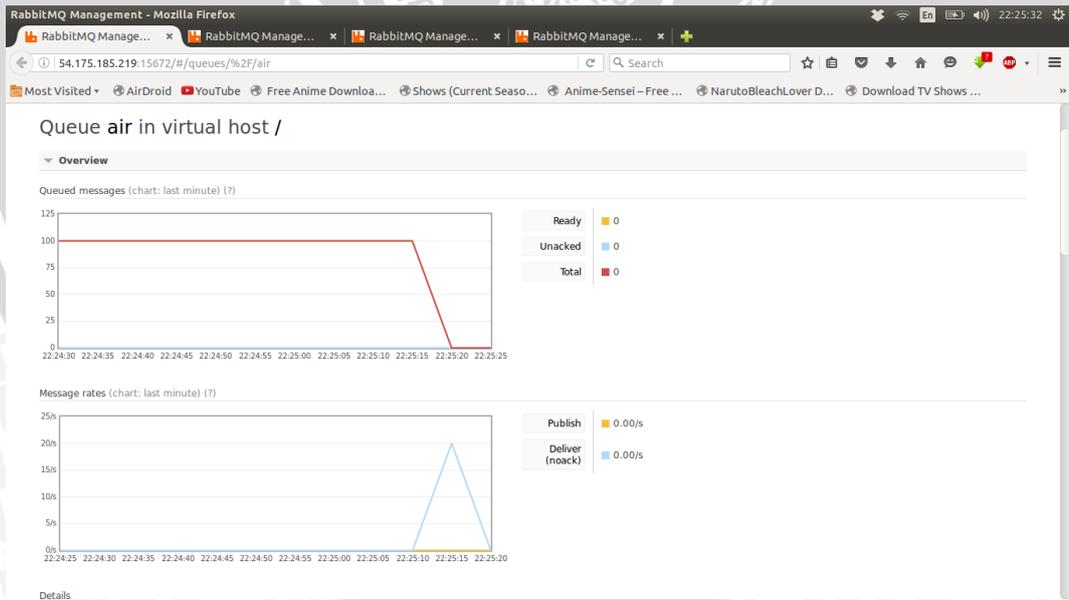
A.2 Capture paket data protokol AMQP dengan Wireshark

No.	Time	Source	Destination	Protocol	Info	Length
51	18:08:32.646	192.168.1.48	54.175.185.219	AMQP	Protocol-Header 0-9-1	74
54	18:08:32.922	54.175.185.219	192.168.1.48	AMQP	Connection.Start	576
56	18:08:32.986	192.168.1.48	54.175.185.219	AMQP	Connection.Start-Ok	453
58	18:08:33.262	54.175.185.219	192.168.1.48	AMQP	Connection.Tune	86
60	18:08:33.270	192.168.1.48	54.175.185.219	AMQP	Connection.Tune-Ok	86
61	18:08:33.271	192.168.1.48	54.175.185.219	AMQP	Connection.Open vhost=/	82
63	18:08:33.545	54.175.185.219	192.168.1.48	AMQP	Connection.Open-Ok	79
64	18:08:33.558	192.168.1.48	54.175.185.219	AMQP	Channel.Open	79
68	18:08:33.838	54.175.185.219	192.168.1.48	AMQP	Channel.Open-Ok	82
69	18:08:33.841	192.168.1.48	54.175.185.219	AMQP	Queue.Declare q=listrik	93
72	18:08:34.115	54.175.185.219	192.168.1.48	AMQP	Queue.Declare-Ok q=list...	94
73	18:08:34.123	192.168.1.48	54.175.185.219	AMQP	Basic.Publish x= rk=lis...	134
74	18:08:34.124	192.168.1.48	54.175.185.219	AMQP	Basic.Publish x= rk=lis...	134
75	18:08:34.124	192.168.1.48	54.175.185.219	AMQP	Basic.Publish x= rk=lis...	134
76	18:08:34.125	192.168.1.48	54.175.185.219	AMQP	Basic.Publish x= rk=lis...	202
77	18:08:34.125	192.168.1.48	54.175.185.219	AMQP	Channel.Close reply=OK	87
81	18:08:34.400	54.175.185.219	192.168.1.48	AMQP	Channel.Close-Ok	78
82	18:08:34.406	192.168.1.48	54.175.185.219	AMQP	Connection.Close reply=...	87
86	18:08:34.680	54.175.185.219	192.168.1.48	AMQP	Connection.Close-Ok	78
94	18:08:35.243	192.168.1.48	54.175.185.219	AMQP	Protocol-Header 0-9-1	74
98	18:08:35.526	54.175.185.219	192.168.1.48	AMQP	Connection.Start	576
100	18:08:35.529	192.168.1.48	54.175.185.219	AMQP	Connection.Start-Ok	453
104	18:08:35.811	54.175.185.219	192.168.1.48	AMQP	Connection.Tune	86
105	18:08:35.812	192.168.1.48	54.175.185.219	AMQP	Connection.Tune-Ok	86
106	18:08:35.813	192.168.1.48	54.175.185.219	AMQP	Connection.Open vhost=/	82
110	18:08:36.094	54.175.185.219	192.168.1.48	AMQP	Connection.Open-Ok	79
111	18:08:36.094	192.168.1.48	54.175.185.219	AMQP	Channel.Open	79
114	18:08:36.377	54.175.185.219	192.168.1.48	AMQP	Channel.Open-Ok	82

A.3 Rekaman trafik di broker RabbitMQ Proses *publish*



A.4 Rekaman trafik di broker RabbitMQ proses *subscribe*



A.5 Data final protokol MQTT

100 Data

No	Pub-Brok(second) 100			
	Alarm	Api	Listrik	Air
1	0,688	0,687	0,698	0,689
2	0,683	0,676	0,691	0,684
3	0,689	0,695	0,694	0,678
4	0,684	0,690	0,688	0,690
5	0,687	0,689	0,691	0,680
6	0,693	0,685	0,681	0,685
7	0,692	0,679	0,692	0,688
8	0,685	0,683	0,684	0,694
9	0,676	0,678	0,688	0,677
10	0,693	0,689	0,680	0,682
Rata-rata	0,687	0,685	0,689	0,685

200 Data

No	Pub-Brok(second) 100			
	Alarm	Api	Listrik	Air
1	0,692	0,701	0,692	0,691
2	0,703	0,686	0,697	0,692
3	0,682	0,695	0,697	0,701
4	0,688	0,694	0,691	0,703
5	0,698	0,701	0,693	0,684
6	0,702	0,693	0,689	0,699
7	0,700	0,688	0,693	0,705
8	0,699	0,694	0,690	0,702
9	0,706	0,703	0,686	0,688
10	0,697	0,696	0,688	0,703
Rata-rata	0,697	0,695	0,692	0,697

No	Brok-Sub(second) 100			
	Alarm	Api	Listrik	Air
1	0,334	0,406	0,406	0,406
2	0,407	0,407	0,408	0,408
3	0,408	0,405	0,409	0,409
4	0,407	0,408	0,409	0,408
5	0,410	0,406	0,407	0,407
6	0,410	0,407	0,410	0,408
7	0,410	0,409	0,409	0,408
8	0,408	0,408	0,410	0,411
9	0,411	0,410	0,410	0,409
10	0,411	0,409	0,411	0,410
Rata-rata	0,402	0,408	0,409	0,408

No	Brok-Sub(second) 100			
	Alarm	Api	Listrik	Air
1	0,414	0,414	0,412	0,412
2	0,412	0,412	0,413	0,412
3	0,412	0,414	0,413	0,412
4	0,413	0,413	0,415	0,413
5	0,415	0,413	0,415	0,413
6	0,415	0,415	0,415	0,356
7	0,416	0,414	0,413	0,416
8	0,416	0,417	0,418	0,419
9	0,416	0,417	0,419	0,417
10	0,417	0,417	0,417	0,418
Rata-rata	0,415	0,415	0,415	0,409



No	Pub-Sub(second) 100			
	Alarm	Api	Listrik	Air
1	0,354	0,395	0,371	0,361
2	0,350	0,376	0,362	0,347
3	0,354	0,397	0,369	0,338
4	0,354	0,395	0,359	0,357
5	0,361	0,397	0,367	0,354
6	0,359	0,396	0,352	0,359
7	0,353	0,378	0,361	0,356
8	0,356	0,391	0,356	0,356
9	0,342	0,391	0,361	0,344
10	0,357	0,401	0,348	0,344
Rata-rata	0,354	0,391	0,361	0,352

No	Pub-Sub(second) 100			
	Alarm	Api	Listrik	Air
1	0,332	0,364	0,341	0,340
2	0,327	0,347	0,346	0,335
3	0,337	0,358	0,343	0,344
4	0,316	0,368	0,333	0,350
5	0,326	0,370	0,336	0,333
6	0,331	0,356	0,330	0,343
7	0,331	0,352	0,343	0,344
8	0,322	0,357	0,330	0,338
9	0,330	0,362	0,330	0,329
10	0,331	0,356	0,335	0,348
Rata-rata	0,328	0,359	0,337	0,340

300 Data

No	Pub-Brok(second) 300			
	Alarm	Api	Listrik	Air
1	0,702	0,689	0,706	0,702
2	0,693	0,692	0,697	0,689
3	0,686	0,694	0,703	0,704
4	0,694	0,717	0,687	0,697
5	0,704	0,695	0,699	0,696
6	0,699	0,689	0,709	0,693
7	0,707	0,695	0,704	0,692
8	0,705	0,705	0,703	0,704
9	0,698	0,705	0,700	0,689
10	0,714	0,702	0,705	0,703
Rata-rata	0,700	0,698	0,701	0,697

400 Data

No	Pub-Brok(second) 400			
	Alarm	Api	Listrik	Air
1	0,698	0,714	0,695	0,703
2	0,691	0,705	0,696	0,690
3	0,702	0,702	0,707	0,703
4	0,698	0,703	0,692	0,715
5	0,702	0,716	0,696	0,711
6	0,701	0,707	0,706	0,652
7	0,701	0,698	0,708	0,657
8	0,701	0,696	0,698	0,697
9	0,698	0,704	0,704	0,704
10	0,694	0,693	0,698	0,699
Rata-rata	0,698	0,704	0,700	0,693



No	Brok-Sub(second) 300			
	Alarm	Api	Listrik	Air
1	0,416	0,418	0,416	0,419
2	0,417	0,420	0,418	0,406
3	0,417	0,420	0,420	0,418
4	0,421	0,419	0,420	0,418
5	0,419	0,420	0,420	0,419
6	0,420	0,418	0,420	0,420
7	0,419	0,419	0,419	0,419
8	0,420	0,420	0,420	0,419
9	0,421	0,420	0,420	0,419
10	0,420	0,422	0,421	0,422
Rata-rata	0,419	0,420	0,419	0,418

No	Brok-Sub(second) 400			
	Alarm	Api	Listrik	Air
1	0,421	0,421	0,420	0,421
2	0,420	0,423	0,423	0,421
3	0,423	0,423	0,423	0,427
4	0,424	0,422	0,423	0,423
5	0,424	0,424	0,422	0,424
6	0,423	0,424	0,424	0,424
7	0,423	0,423	0,423	0,425
8	0,423	0,423	0,422	0,422
9	0,425	0,423	0,424	0,424
10	0,424	0,425	0,424	0,426
Rata-rata	0,423	0,423	0,423	0,424

No	Pub-Sub(second) 300			
	Alarm	Api	Listrik	Air
1	0,323	0,327	0,329	0,332
2	0,305	0,333	0,319	0,331
3	0,300	0,332	0,322	0,334
4	0,299	0,367	0,310	0,327
5	0,312	0,330	0,318	0,326
6	0,312	0,330	0,331	0,322
7	0,317	0,331	0,326	0,323
8	0,309	0,343	0,323	0,338
9	0,307	0,343	0,319	0,320
10	0,322	0,336	0,321	0,332
Rata-rata	0,311	0,337	0,322	0,329

No	Pub-Sub(second) 400			
	Alarm	Api	Listrik	Air
1	0,289	0,340	0,309	0,320
2	0,285	0,326	0,306	0,309
3	0,300	0,322	0,318	0,323
4	0,292	0,325	0,305	0,332
5	0,291	0,469	0,308	0,330
6	0,295	0,327	0,316	0,272
7	0,298	0,321	0,319	0,269
8	0,299	0,321	0,311	0,314
9	0,287	0,325	0,313	0,319
10	0,283	0,309	0,310	0,314
Rata-rata	0,292	0,339	0,311	0,310



500 Data

No	Pub-Brok(second)			
	Alarm	Api	Listrik	Air
1	0,705	0,698	0,735	0,728
2	0,698	0,706	0,705	0,709
3	0,714	0,707	0,713	0,696
4	0,699	0,690	0,706	0,697
5	0,703	0,715	0,714	0,695
6	0,712	0,709	0,720	0,711
7	0,698	0,698	0,707	0,720
8	0,714	0,711	0,718	0,700
9	0,707	0,712	0,714	0,702
10	0,713	0,717	0,712	0,704
Rata-rata	0,706	0,706	0,714	0,706

No	Brok-Sub(second)			
	Alarm	Api	Listrik	Air
1	0,423	0,426	0,426	0,426
2	0,424	0,425	0,425	0,426
3	0,425	0,425	0,425	0,425
4	0,424	0,435	0,435	0,425
5	0,426	0,427	0,427	0,428
6	0,426	0,428	0,428	0,428
7	0,428	0,427	0,427	0,424
8	0,427	0,428	0,428	0,427
9	0,428	0,427	0,427	0,428
10	0,429	0,427	0,427	0,428
Rata-rata	0,426	0,427	0,427	0,426



No	Pub-Sub(second)			
	Alarm	Api	Listrik	Air
1	0,293	0,309	0,310	0,342
2	0,282	0,317	0,305	0,316
3	0,296	0,320	0,314	0,305
4	0,280	0,311	0,309	0,307
5	0,284	0,325	0,316	0,301
6	0,293	0,316	0,319	0,319
7	0,276	0,307	0,307	0,331
8	0,294	0,322	0,317	0,306
9	0,287	0,324	0,310	0,308
10	0,289	0,328	0,316	0,311
Rata-rata	0,287	0,318	0,312	0,315

A.6 Data final protokol AMQP

100 Data				
no	Pub-Brok (second) 100			
	Alarm	Api	Listrik	Air
1	0,926	0,950	0,953	0,935
2	0,926	0,941	0,931	0,928
3	0,946	0,940	0,950	0,943
4	0,950	0,981	0,959	0,940
5	0,945	0,945	0,952	0,909
6	0,950	0,947	0,957	0,958
7	0,954	0,962	0,952	0,945
8	0,962	0,974	0,967	0,955
9	0,949	0,967	0,965	0,956
10	0,951	0,949	0,965	0,965
Rata-Rata	0,946	0,956	0,955	0,944

200 Data				
no	Pub-Brok (second) 200			
	Alarm	Api	Listrik	Air
1	0,975	0,965	0,947	0,955
2	0,957	0,947	0,954	0,940
3	0,957	0,957	0,966	0,949
4	0,963	0,964	0,962	0,947
5	0,953	0,961	0,954	0,955
6	0,948	0,967	0,953	0,958
7	0,965	0,979	0,953	0,985
8	0,975	0,965	0,969	0,959
9	0,966	0,958	0,971	0,966
10	0,974	0,987	0,959	0,950
Rata-Rata	0,963	0,965	0,959	0,956



no	Brok-Sub (second)100			
	Alarm	Api	Listrik	Air
1	0,658	0,657	0,657	0,658
2	0,621	0,657	0,634	0,657
3	0,672	0,672	0,673	0,673
4	0,674	0,674	0,673	0,648
5	0,673	0,673	0,675	0,676
6	0,654	0,640	0,674	0,674
7	0,674	0,675	0,675	0,647
8	0,675	0,657	0,675	0,675
9	0,653	0,675	0,677	0,677
10	0,653	0,652	0,675	0,676
Rata-Rata	0,661	0,663	0,669	0,666

no	Brok-Sub (second) 200			
	Alarm	Api	Listrik	Air
1	0,637	0,650	0,651	0,643
2	0,679	0,645	0,650	0,635
3	0,636	0,635	0,658	0,634
4	0,655	0,666	0,662	0,638
5	0,673	0,655	0,659	0,662
6	0,635	0,655	0,639	0,664
7	0,550	0,680	0,643	0,666
8	0,655	0,659	0,652	0,647
9	0,657	0,634	0,683	0,657
10	0,683	0,682	0,641	0,645
Rata-Rata	0,646	0,656	0,654	0,649

no	Pub-Sub (second) 100			
	Alarm	Api	Listrik	Air
1	0,268	0,341	0,352	0,337
2	0,307	0,337	0,312	0,330
3	0,340	0,319	0,328	0,343
4	0,349	0,363	0,364	0,296
5	0,331	0,324	0,337	0,338
6	0,310	0,309	0,343	0,346
7	0,347	0,350	0,345	0,301
8	0,356	0,338	0,362	0,347
9	0,320	0,355	0,364	0,361
10	0,299	0,303	0,353	0,351
Rata-Rata	0,323	0,334	0,346	0,335

no	Pub-Sub (second) 200			
	Alarm	Api	Listrik	Air
1	0,473	0,393	0,313	0,319
2	0,335	0,313	0,328	0,311
3	0,315	0,324	0,330	0,314
4	0,408	0,392	0,321	0,304
5	0,328	0,397	0,333	0,318
6	0,313	0,402	0,321	0,322
7	0,526	0,378	0,321	0,391
8	0,347	0,396	0,327	0,313
9	0,347	0,330	0,347	0,403
10	0,347	0,373	0,329	0,310
Rata-Rata	0,374	0,370	0,327	0,331



300 Data

no	Pub-Brok (second) 300			
	Alarm	Api	Listrik	Air
1	0,946	0,958	0,953	0,961
2	0,970	0,961	0,966	0,961
3	0,969	0,992	0,962	0,971
4	0,961	0,973	0,976	0,976
5	0,952	0,952	0,969	0,968
6	0,975	0,963	0,965	0,973
7	0,949	0,954	0,970	0,967
8	0,942	0,981	0,960	0,965
9	0,959	0,971	0,963	0,969
10	0,968	0,956	0,969	0,980
Rata-Rata	0,959	0,966	0,965	0,969

400 Data

no	Pub-Brok (second) 400			
	Alarm	Api	Listrik	Air
1	0,977	0,967	0,963	0,972
2	0,972	0,968	0,969	0,980
3	0,964	0,985	0,962	0,975
4	0,972	0,973	0,961	0,982
5	0,958	0,974	0,978	0,976
6	0,963	0,975	0,964	0,977
7	0,977	0,969	0,967	0,973
8	0,969	0,971	0,955	0,976
9	0,968	0,971	0,965	0,964
10	0,979	0,987	0,900	0,972
Rata-Rata	0,970	0,974	0,958	0,975

no	Brok-Sub (second)300			
	Alarm	Api	Listrik	Air
1	0,639	0,649	0,634	0,664
2	0,650	0,644	0,638	0,666
3	0,643	0,639	0,650	0,658
4	0,650	0,651	0,635	0,644
5	0,647	0,645	0,635	0,643
6	0,637	0,644	0,645	0,640
7	0,656	0,658	0,634	0,663
8	0,667	0,664	0,652	0,653
9	0,653	0,659	0,668	0,651
10	0,660	0,650	0,655	0,632
Rata-Rata	0,650	0,650	0,644	0,651

no	Brok-Sub (second) 400			
	Alarm	Api	Listrik	Air
1	0,637	0,669	0,652	0,636
2	0,642	0,646	0,644	0,650
3	0,649	0,671	0,672	0,639
4	0,657	0,649	0,643	0,648
5	0,653	0,659	0,646	0,646
6	0,655	0,664	0,595	0,646
7	0,677	0,652	0,652	0,659
8	0,645	0,647	0,649	0,638
9	0,661	0,657	0,652	0,650
10	0,659	0,652	0,677	0,655
Rata-Rata	0,653	0,657	0,648	0,647



no	Pub-Sub (second) 300			
	Alarm	Api	Listrik	Air
1	0,310	0,381	0,459	0,373
2	0,380	0,309	0,412	0,379
3	0,392	0,382	0,379	0,398
4	0,336	0,370	0,412	0,315
5	0,284	0,300	0,424	0,326
6	0,407	0,311	0,326	0,434
7	0,321	0,358	0,427	0,371
8	0,296	0,394	0,278	0,401
9	0,361	0,308	0,309	0,320
10	0,279	0,273	0,337	0,447
Rata-Rata	0,337	0,339	0,376	0,376

no	Pub-Sub (second) 400			
	Alarm	Api	Listrik	Air
1	0,394	0,395	0,342	0,421
2	0,335	0,363	0,372	0,399
3	0,312	0,403	0,347	0,411
4	0,442	0,412	0,312	0,416
5	0,302	0,403	0,402	0,397
6	0,319	0,397	0,438	0,373
7	0,435	0,341	0,319	0,390
8	0,331	0,377	0,379	0,460
9	0,375	0,365	0,321	0,390
10	0,424	0,428	0,337	0,365
Rata-Rata	0,367	0,388	0,357	0,402

500 Data

no	Pub-Brok (second) Final			
	Alarm	Api	Listrik	Air
1	0,969	0,973	0,959	0,968
2	0,975	0,975	0,985	0,978
3	1,000	0,877	1,016	0,974
4	0,980	0,974	0,985	0,977
5	0,995	0,978	0,968	0,978
6	0,966	0,978	0,958	0,964
7	0,987	0,972	0,966	0,985
8	0,981	0,970	0,921	0,982
9	0,977	0,973	0,990	0,974
10	0,975	0,982	0,985	0,983
Rata-Rata	0,981	0,965	0,973	0,976



no	Brok-Sub (second) Final			
	Alarm	Api	Listrik	Air
1	0,679	0,631	0,647	0,645
2	0,644	0,667	0,649	0,644
3	0,638	0,645	0,644	0,677
4	0,654	0,626	0,654	0,651
5	0,660	0,664	0,648	0,656
6	0,652	0,654	0,649	0,647
7	0,656	0,645	0,652	0,663
8	0,645	0,646	0,648	0,658
9	0,644	0,697	0,659	0,644
10	0,680	0,663	0,666	0,663
Rata-Rata	0,655	0,654	0,651	0,655

no	Pub-Sub (second) Final			
	Alarm	Api	Listrik	Air
1	0,368	0,355	0,337	0,377
2	0,289	0,305	0,319	0,423
3	0,476	0,339	0,375	0,408
4	0,436	0,459	0,439	0,361
5	0,446	0,356	0,376	0,403
6	0,311	0,342	0,400	0,312
7	0,404	0,373	0,347	0,419
8	0,328	0,330	0,331	0,345
9	0,377	0,308	0,451	0,345
10	0,379	0,411	0,418	0,417
Rata-Rata	0,382	0,358	0,379	0,381

A.7 Contoh data yang dikirim

R 001 0 2016-05-19 22:23:58.524
 R 002 0 2016-05-19 22:23:58.524
 R 003 1 2016-05-19 22:23:58.524
 R 004 1 2016-05-19 22:23:58.524
 R 005 1 2016-05-19 22:23:58.524
 R 006 1 2016-05-19 22:23:58.524
 R 007 1 2016-05-19 22:23:58.524
 R 008 1 2016-05-19 22:23:58.524
 R 009 0 2016-05-19 22:23:58.524
 R 010 1 2016-05-19 22:23:58.524

A.8 Contoh data yang diterima

Message Received b'R 001 1' 2016-05-21 23:50:28.628
 Message Received b'R 002 0' 2016-05-21 23:50:28.628
 Message Received b'R 003 0' 2016-05-21 23:50:28.628
 Message Received b'R 004 0' 2016-05-21 23:50:28.628
 Message Received b'R 005 0' 2016-05-21 23:50:28.628
 Message Received b'R 006 0' 2016-05-21 23:50:28.628
 Message Received b'R 007 1' 2016-05-21 23:50:28.628
 Message Received b'R 008 1' 2016-05-21 23:50:28.628
 Message Received b'R 009 0' 2016-05-21 23:50:28.628
 Message Received b'R 010 1' 2016-05-21 23:50:28.628

