Daftar Tabel

Tabel 2.1 Perbedaan fitur MQTT dan DDS 6
Tabel 2.2 Jenis Pesan pada MQTT 8
Tabel 2.2.1.1 MQTT Control Packet 9
Tabel 2.2.1.2 Fixed Header dari MQTT Control Packet: 9
Tabel 2.2.1.3 Format Packet Identifier Packet byte 9
Tabel 2.2.1.4 Control Packet dan Packet Identifier Field 10
Tabel 2.2.1.5 MQTT Control Packet Type 11
Tabel 3.3.1 Skenario uji jaringan performansi tinggi 22
Tabel 3.3.2.1 Skenario uji jaringan high latency layanan MQTT 24
Tabel 3.3.2.2 Skenario uji jaringan high latency layanan DDS 24
Tabel 4.1.1 Hasil pengujian delay jaringan Cloud ke Cloud layanan MQTT 25
Tabel 4.1.2 Hasil pengujian delay jaringan Cloud ke Cloud layanan DDS 26
Tabel 4.1.3 Hasil perbandingan pengujian delay jaringan Cloud ke Cloud antara layanan MQTT dan DDS 26
Tabel 4.1.4 Hasil pengujian drop packet jaringan Cloud ke Cloud layanan MQTT 27
Tabel 4.1.5 Hasil pengujian drop packet jaringan Cloud ke Cloud layanan DDS 27
Tabel 4.2.1 Hasil pengujian delay jaringan LAN layanan MQTT dengan traffic background 1000000 packet 28
Tabel 4.2.2 Hasil pengujian delay jaringan LAN layanan MQTT dengan traffic background 10000 packet 29
Tabel 4.2.3 Hasil pengujian delay jaringan LAN layanan MQTT dengan traffic background 100 packet 29
Tabel 4.2.2.1 Hasil pengujian delay jaringan LAN layanan DDS dengan traffic background 1000000 packet 30
Tabel 4.2.2.2 Hasil pengujian delay jaringan LAN layanan DDS dengan traffic background 10000 packet 30
Tabel 4.2.2.3 Hasil pengujian delay jaringan LAN layanan DDS dengan traffic background 100 packet. 30
Tabel 4.2.3.1 Hasil perbandingan pengujian delay jaringan LAN antara layanan MQTT dan DDS dengan traffic background 1000000 packet 31
Tabel 4.2.3.2 Hasil perbandingan pengujian delay jaringan LAN antara layanan MQTT dan DDS dengan traffic background 10000 packet 32
Tabel 4.2.3.3 Hasil perbandingan pengujian delay jaringan LAN antara layanan MQTT dan DDS dengan traffic background 100 packet 32

Tabel 4.2.4.1 Hasil pengujian drop packet jaringan LAN layanan MQTT dengan traffic background 1000000 packet 33
Tabel 4.2.4.2 Hasil pengujian drop packet jaringan LAN layanan MQTT dengan traffic background 10000 packet 33
Tabel 4.2.4.3 Hasil pengujian drop packet jaringan LAN layanan MQTT dengan traffic background 100 packet 34
Tabel 4.2.5.1 Hasil pengujian drop packet jaringan LAN layanan DDS dengan traffic background 1000000 packet 34
Tabel 4.2.5.2 Hasil pengujian drop packet jaringan LAN layanan DDS dengan traffic background 10000 packet 35
Tabel 4.2.5.3 Hasil pengujian drop packet jaringan LAN layanan DDS dengan traffic background 100 packet 35

1 Pendahuluan

1.1 Latar Belakang

Protokol *publish/subscribe* merupakan salah satu protokol yang popular digunakan untuk komunikasi pada *device-device* yang digunakan pada *Internet of Things* [1]. Protokol populer lainnya adalah *request/reply*. Salah satu kelebihan dari protokol *publish/subscribe* dibandingkan protokol *request/reply* adalah kemampuannya dalam pengiriman *message* secara berkelanjutan, sehingga sangat cocok untuk data yang bersifat non-diskrit [2]. Data non-diskrit misalnya adalah data tentang perubahan suhu, data tentang denyut jantung, atau data lainnya yang semacam itu. Contoh protokol *publish/subscribe* yang menggunakan broker adalah *Message Queue Telemetry Transport* (MQTT) [3]. Sedangkan contoh protokol *publish/subscribe* yang tidak menggunakan broker adalah *Data Distribution Service* (DDS)[4].

Broker definisi umumnya adalah sebagai perantara. Fungsi utamanya adalah berlaku sebagai penampung sementara atau menjalankan mekanisme *queue*. Sehingga model arsitektur jaringan yang menggunakan broker ini sangat cocok digunakan pada jaringan dengan karakter *high latency* [3]. Sedangkan protokol *publish/subscribe* yang tidak menggunakan broker pada model arsitektur jaringannya, bertujuan untuk meminimalisir kompleksitas komunikasi datanya, dengan harapan agar dapat memenuhi karakternya, yaitu skalabilitas dengan menghilangkan kompleksitas dalam komunikasi datanya [3].

Berdasarkan referensi yang berkaitan, pada tugas akhir ini dilakukan analisis pengujian terhadap kedua protokol MQTT dan DDS, sehingga didapatkan hasil analisis yang mengacu pada pengukuran terhadap parameter *delay* dan *drop packet.*

1.2 Perumusan Masalah

Adapun perumusan masalah untuk tugas akhir ini adalah sebagai berikut:

a. Bagaimana pengaruh broker pada metode *publish/subscribe* terkait model jaringan *low latency* dan model jaringan *high latency,* dengan skalabilitas yang besar?
b. Bagaimana performansi *delay* dan *drop packet* pada protokol MQTT bila berada pada jaringan *low latency* atau pada model jaringan *high latency,* dengan skalabilitas yang besar?
c. Bagaimana performansi *delay* dan *drop packet* pada protokol DDS bila berada pada jaringan *low latency* atau pada model jaringan *high latency,* dengan skalabilitas yang besar?

1.3 Tujuan

Tujuan yang ingin dicapai dalam tugas akhir ini adalah:

a. Mendapatkan analisis pengaruh adanya atau tidak adanya broker pada metode *publish/subscribe* terkait model jaringan *low latency* dan model jaringan *high latency*.
b. Mendapatkan analisis performansi *delay* dan *drop packet* pada protokol MQTT pada jaringan *low latency* atau jaringan *high latency*.
c. Mendapatkan analisis performansi *delay* dan *drop packet* pada protokol DDS pada jaringan *low latency* atau jaringan *high latency*.

1.4 Hipotesa

Broker pada protokol *publish/subscribe* memiliki mekanisme *queue*, sehingga untuk komunikasi data yang tidak cepat, seperti pada jaringan yang memiliki trafik yang besar, dapat membantu agar paket-paket yang dikirim pada komunikasi data tidak serta-merta di-*drop*. Sehingga protokol *publish/subscribe* dengan arsitektur yang menggunakan broker seperti MQTT ini akan memiliki performansi yang lebih baik daripada yang tanpa broker seperti DDS bila dilihat dari parameter *drop packet* terutama pada jaringan dengan *high latency*.

Namun penggunaan broker ini menambah kompleksitas dalam pengiriman paket ke *receiver*. Sehingga bila dikaitkan sengan skalabilitas penggunaan aplikasi *(subscriber)* yang banyak, maka DDS akan memberikan performansi yang lebih baik daripada MQTT, terutama dilihat dari parameter *delay,* lantaran pengiriman paket pada DDS dilakukan tanpa pengecekan *error* terhadap paket.

1.5 Batasan Masalah

Batasan masalah pada tugas akhir ini adalah sebagai berikut:

a. Platform broker yang digunakan pada protokol MQTT adalah *Mosquitto,* dengan jumlah broker hanya satu.
b. Platform koneksi yang digunakan pada protokol DDS adalah *RTI Connext DDS*.
c. Hanya sensor suhu yang digunakan pada sistem untuk mewakili skenario pengujian protokol *publish/subscribe*.
d. Sensor suhu yang digunakan tidak bersifat fisik, namun diemulasi dengan mengirimkan data sebanyak 1000 data suhu secara kontinu dari sensor tersebut.
e. *Receiver* yang digunakan dalam bentuk aplikasi *mobile* berbasis *Android*.
f. *Receiver* tidak dalam bentuk fisik, melainkan diemulasikan agar bisa mengemulasikan jumlah aplikasi yang banyak (maksimal 100 aplikasi).
g. Jaringan *high latency* diciptakan dengan membuat *traffic background* yang besar dengan perangkat lunak *traffic generator*, yaitu Ostinato.
h. Parameter performansi yang dianalisa adalah *delay* dan *drop packet*.
i. Berfokus pada pengiriman pesan *one-to-many messaging.*

1.6 Metodologi Penyelesaian Masalah

Metodologi yang digunakan dalam menyelesaikan masalah untuk tugas akhir ini antara lain:

1. Studi Literatur

   Studi literatur dilakukan dengan mempelajari konsep dan teori pendukung yang berkaitan dengan tugas akhir ini. Pengumpulan data dan memahami berbagai referensi yang berhubungan dengan MQTT, DDS, dan materi pendukung lainnya.

2. Konsultasi dengan Pembimbing

   Konsultasi ini dilakukan untuk mengetahui apa saja yang perlu diperhatikan dan bagaimana konsep yang harus dipahami sebelum melakukan implementasi lebih lanjut guna mendapat hasil yang maksimal.

3. Pembangunan model

   Pada tahap ini, dilakukan perancangan sistem dan berbagai material yang diperlukan berdasarkan referensi yang sudah didapatkan sebelumnya dari studi literature maupun konsultasi dengan pembimbing.

4. Implementasi

   Dalam tahap ini, dilakukan percobaan protokol dengan membuat *prototype* sistem, lalu menggunakan MQTT dan DDS dalam metode pengiriman datanya.

5. Pengujian Sistem

   Pada tahap ini, dilakukan pengujian terhadap implementasi dengan parameter dan lingkungan pengujian yang telah ditentukan.

6. Analisa hasil dan penulisan laporan

   Pada tahap ini, melakukan analisa atas hasil pengujian sistem, menentukan faktor yang dapat memperngaruhi akurasi keberhasilan sistem dan dapat menentukan performansi protokol yang diimplementasikan, MQTT dan DDS. Pembuatan laporan dilakukan dengan pengambilan kesimpulan berdasarkan hasil uji coba pada sistem dan mendokumentasikan setiap tahapan yang telah dilakukan dalam penyelesaian tugas akhir ini.

2 Landasan Teori

Terdapat beberapa literatur yang berkaitan dengan penelitian tugas akhir ini, diantaranya sebagai berikut:

Pada [5] David Barnett melakukan penelitian tentang "*MQTT and DDS Comparison*" tahun 2013, tujuan dari penelitian tersebut adalah untuk mengetahui perbedaan kemampuan dari layanan MQTT dan DDS secara umum. Hasil yang didapat bahwa arsitektur jaringan menggunakan protokol MQTT (*Message Queuing Telemetry Transport)* bersifat terpusat (*centralized*), membuat semua proses pengiriman data harus melalui broker terlebih dahulu. Jadi pengiriman data harus melalui *queue.*

Dipaparkan bahwa arsitektur jaringan menggunakan protokol DDS (*Data Distribution Service*) bersifat tersebar (*decentralized*), tidak menggunakan broker pada arsitekturnya. Pengiriman data tidak melalui *gateway* protokol, memungkinkan pengiriman data lebih cepat.

Pada [3] Andrew Foster melakukan penelitian tentang "*Messaging Technologies for the Industrial Internet and the Internet of Things*" tahun 2013, tujuan dari penelitian tersebut adalah untuk membandingkan atara beberapa protokol *IoT* sehingga didapatkan protokhol yang mendukung skenario yang dibutuhkan seperti misalnya komunikasi *Inter* dan *Intra Device,* komunikasi *Device to Cloud*, dan komunikasi *Inter Data Center*, khususnya untuk performansi dan toleransi kesalahan pada setiap pengiriman data.

Dipaparkan juga bahwa arsitektur jaringan protokol MQTT cocok digunakan untuk model jaringan *high latency* atau pengiriman data yang sedikit. Untuk aplikasi yang membutuhkan performansi tinggi, *real-time, many-to-many*, DDS memiliki keunggulan dibanding protokol yang lain. Terutama pada *cloud-based Data Center*.

Dari paparan di atas, secara ringkas dapat dijelaskan pada tabel sebagai berikut:

Tabel 2.1 Perbedaan fitur MQTT dan DDS

Protokol / **Perbedaan**	**MQTT**	**DDS**
Messagging Use	Menggunakan broker	Tanpa broker
Architecture	Terpusat: semua pesan dialirkan ke broker.	Desentralisasi.
Quality of Service (QoS)	Berfokus pada pengiriman pesan *one-to-many messaging.*	Berfokus pada pengiriman pesan *one-to-many messaging* atau *many-to-many messaging*.
Karakteristik	Jaminan terhadap pengiriman data, terdapat mekanisme *queue*.	Pengiriman data tanpa pengecekan *error*.

2.1 Sistem *Publish/Subscribe*

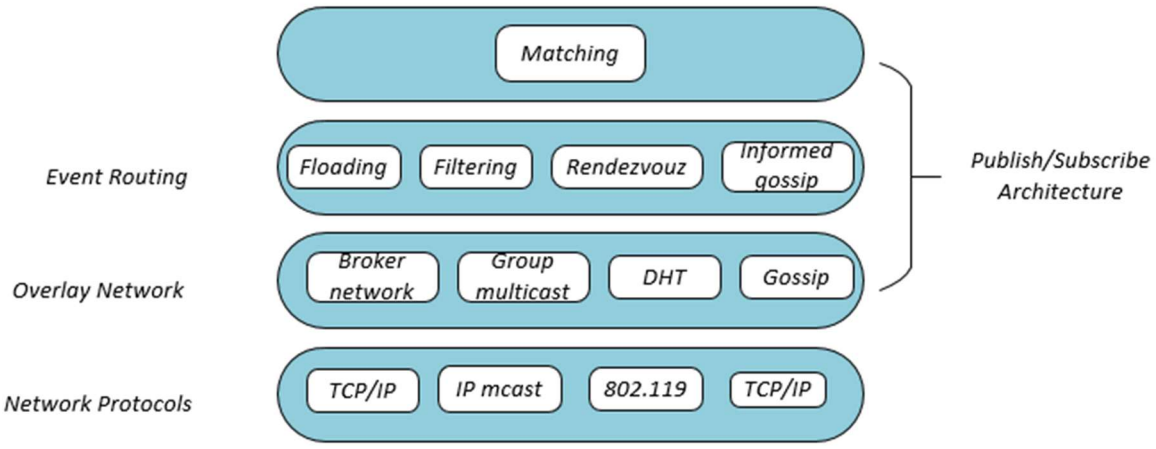
Publish/subscribe merupakan salah satu model sistem terdistribusi yang terdiri dari *publisher* dan *subscribe*. *Publisher* adalah pengirim data, sedangkan *subscriber* adalah penerima data berdasarkan langganan *(subscription)* [6]. Terdapat jenis yang membedakan di *publish/subscribe*, yaitu yang menggunakan broker dan tidak menggunakan broker.

Broker berfungsi sebagai perantara *publisher* ke *subscriber* dan mengelola *subscription* dari *subscriber* [5]. *Publisher* mengirim data kepada broker tertentu bersama dengan beberapa metadata, dan dapat memanipulasi segala data yang dikirim tersebut, seperti *filtering* data*, compressing* data*, aggregating* data (menggabungkan data). Sebelum mengirim data oleh *subscriber*, broker melakukan *filtering* data dengan cara mencocokan nilai-nilai metadata dan *subscriber* yang menyatakan kecocokan metadata tersebut [6].

Tidak seperti MQTT yang menggunakan broker pada arsitekturnya, DDS tidak menggunakannya. Selain itu, DDS menyediakan mekanisme layanan *real-time* untuk

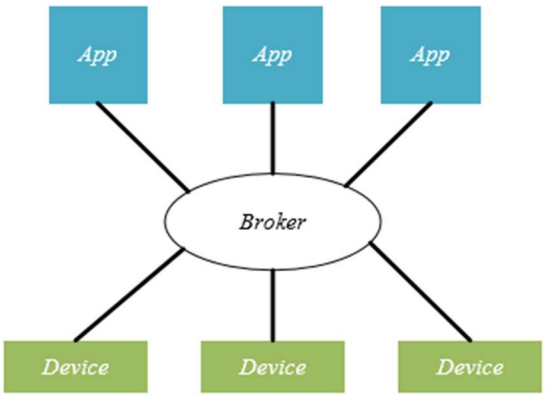
bus dengan performansi tinggi. Sehingga DDS cocok digunakan sebagai protokol pengiriman pesan pada *Data Cente*r berbasis *Cloud* [7].

Mekanisme *connectionless* yang diadopsi DDS, membuat pengiriman data **tanpa disertai pengecekan jika terjadi kesalahan**. Sebab pada DDS, target keberhasilan pengiriman pesan lebih ditujukan penggunaan jumlah aplikasi berskala besar dengan platform yang berbeda-beda [6].



Gambar 2.1 Arsitektur *Publish/Subscribe* [6].

2.2 Protokol MQTT *(Message Queueing Telemetry Transport)*



Gambar 2.2 Arsitektur MQTT [5]

MQTT dirancang sebagai protokol transport yang ringan, terbuka dan sederhana, agar mudah diimplementasikan [3]. Karakteristik ini membuat MQTT ideal digunakan dalam lingkungan jaringan yang terbatas, seperti, *bandwidth* yang rendah, dan memori

yang terbatas pada suatu sistem *embedded*. Protokol MQTT memiliki 14 jenis tipe pesan yang berbeda-beda, seperti pada tabel berikut ini [15]:

Tabel 2.2 Jenis pesan pada MQTT

Mnemonic	No.	*Description*
CONNECT	1	Klien *request to connect to Server*
CONNACK	2	*Connect Acknowledgment*
PUBLISH	3	*Publish message*
PUBACK	4	*Publish Acknowledgment*
PUBREC	5	*Publish Received-assured delivery part 1*
PUBREL	6	*Publish Release-assured delivery part 2*
PUBCOMP	7	*Publish Complete-assured delivery part 3*
SUBSCRIBE	8	Klien *Subscribe request*
SUBACK	9	*Subscribe Acknowledgment*
UNSUBSCRIBE	10	Klien *Unsubscribe request*
UNSUBACK	11	*Unsubscribe Acknowledgment*
PINGREQ	12	PING *Request*
PINGRESP	13	PING *Response*
DISCONNECT	14	Klien *is Disconnecting*

Protokol transport yang digunakan MQTT adalah TCP/IP. Sedangkan koneksi komunikasi data untuk layanan TCP/IP dapat digunakan protokol seperti TLS dan *WebSocket*.

2.2.1. *MQTT Control Packet*

Format *control packet* pada MQTT di bagi menjadi tiga bagian [15]:

Tabel 2.2.1.1 *MQTT Control Packet*

Fixed header, present in all MQTT Control Packets
Variable header, present in some MQTT Control Packets
Payload, present in some MQTT Control Packets

Setiap *MQTT control packet* memiliki *fixed header* dengan format berikut [15]:

Tabel 2.2.1.2 *Fixed Header* dari *MQTT Control Packet*:

Bit	7	6	5	4	3	2	1	0
byte 1	*MQTT Control Packet Type*				*Flags specific to each MQTT Control Packet type*			
byte 2	*Remaining Length*							

Pada beberapa tipe *MQTT control packet* terdapat *variable header. Variable header* ini berada di antara *fixed header* dan *payload.* Isi dari *variable header* bergantung pada jenis paket. *Packet identifier* dari *variable header* dapat dimiliki beberapa jenis paket, berikut format *packet identifier byte*.

Tabel 2.2.1.3 *Format Packet Identifier Packet byte*

Bit	7	6	5	4	3	2	1	0
byte 1	*Packet Identifier MSB*							
byte 2	*Packet Identifier LSB*							

Control Packet memerlukan *packet identifier* yang dijelaskan dengan tabel berikut [15]:

Tabel 2.2.1.4 *Control Packet* dan *Packet Identifier Field*

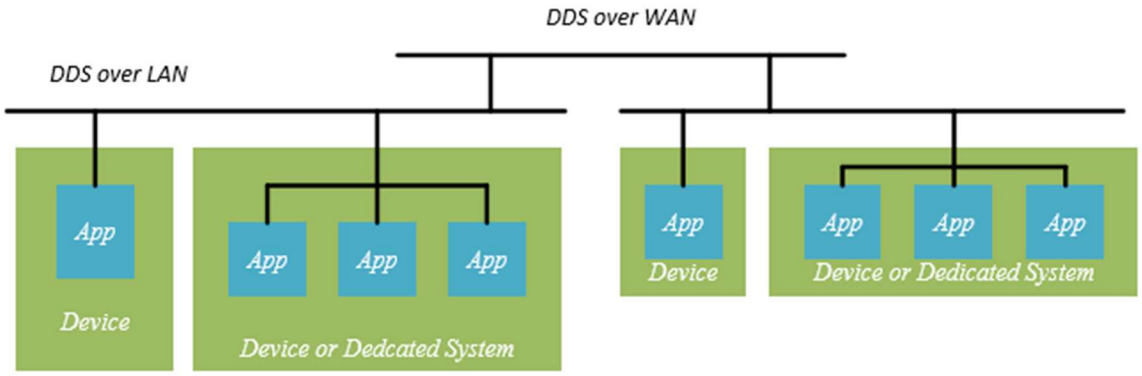
Control Packet	*Packet Identifier Field*
CONNECT	*NO*
CONNACK	*NO*
PUBLISH	*YES (If QoS > 0)*
PUBACK	*YES*
PUBREC	*YES*
PUBREL	*YES*
PUBCOMP	*YES*
SUBSCRIBE	*YES*
SUBACK	*YES*
UNSUBSCRIBE	*YES*
UNSUBACK	*YES*
PINGREQ	*NO*
PINGRESP	*NO*
DISCONNECT	*NO*

Untuk lebih jelasnya mengenai nilai dari *MQTT Control Packet Type*, dapat mengacu pada tabel berikut ini [15]:

Tabel 2.2.1.5 *MQTT Control Packet Type*

Name	*Value*	*Direction of Flow*	*Description*
Reserved	0	*Forbidden*	*Reserved*
CONNECT	1	*Client to Server*	*Client Request to connect to Server*
CONNACK	2	*Server to Client*	*Connect Acknowledgment*
PUBLISH	3	*Client to Server or Server to Client*	*Publish message*
PUBACK	4	*Client to Server*	*Publish Acknowledgment*
PUBREC	5	*Client to Server or Server to Client*	*Publish received (assured delivery part 1)*
PUBREL	6	*Client to Server or Server to Client*	*Publish release (assured delivery part 2)*
PUBCOMP	7	*Client to Server or Server to Client*	*Publish complete (assured delivery part 3)*
SUBSCRIBE	8	*Client to Server*	*Client subscribe request*
SUBACK	9	*Server to Client*	*Subscribe Acknowledgment*
UNSUBSCRIBE	10	*Client to Server*	*Unsubscribe request*
UNSUBACK	11	*Server to Client*	*Unsubscribe Acknowledgment*
PINGREQ	12	*Client to Server*	*PING request*
PINGRESP	13	*Server to Client*	*PING response*
DISCONNECT	14	*Client to Server*	*Client is disconnecting*
Reserved	15	*Forbidden*	*reserved*

2.3 *Protokol DDS (Data Distribution Services)*



Gambar 2.3 *DDS Architecture* [5]

Protokol DDS merupakan sistem *real-time* yang digunakan untuk *Object Management Group* (OMG) sebagai standar yang memiliki peran sebagai *middleware*. Arsitektur yang digunakan pada protokol DDS adalah berbeda dengan MQTT. Bila arsitektur MQTT adalah tersentralisasi, menggunakan layanan *API Gateway* berupa broker, maka pada arsitektur DDS adalah bersifat terdistribusi dan tidak menggunakan layanan *API Gateway*.

Secara keseluruhan, DDS adalah protokol yang paling serbaguna dibanding protokol yang lainnya. Hal ini dikarenakan DDS mampu mengelola *tiny device*, termasuk penggunaan level jaringan yang berbeda, seperti pada jaringan dengan performa tinggi [3].

Untuk sebagian besar aplikasi berbasiskan DDS memiliki sistem terdistribusi dengan setidaknya memiliki tiga karakteristik sebagai berikut [10] :

1. *Realibility*: lima menit atau kurang dari waktu yang ditentukan akan dianggap gagal mengirimkan suatu pesan.
2. *Performance or Scale*: sistem menunjukan waktu yang dibutuhkan pada milidetik atau mikrodetik, atau yang terhubung pada banyak aplikasi dengan banyak data.
3. *Longevity*: Siklus dari *code* setidaknya tiga tahun, dan membutuhkan pembangunan yang terkoordinasi.