

Implementasi Perangkat *Mobile Publisher Subscriber* Sebagai Perantara Pengiriman Data Sensor Dari Lapangan Ke Pusat Data

Sukma Alamsyah Budianto¹, Adhitya Bhawiyuga², Dany Primanita Kartikasari³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹sukmaalamsyah@gmail.com, ²bhawiyuga@ub.ac.id, ³dany.jalin@ub.ac.id

Abstrak

Perangkat sensor yang berada di lapangan terdapat masalah ketika mengirim data pada kondisi jaringan yang tidak stabil menuju ke pusat data. Untuk solusi dari permasalahan ini dikembangkan sebuah aplikasi android yang mengimplementasikan komunikasi *publish/subscribe* sebagai perantara pengiriman data sensor di lapangan menuju pusat data dengan menggunakan protokol MQTT. Dilakukan tiga buah pengujian aplikasi yaitu pengujian fungsional, pengujian skenario dan pengujian performance. Pada pengujian fungsional, lima kebutuhan fungsional yang dirancang dapat dipenuhi oleh aplikasi sebagai perantara pengiriman data sensor ke pusat data. Pada pengujian skenario, dari tiga pengujian skenario yang dilakukan aplikasi bisa dijalankan pada ketiga kondisi skenario yang telah dirancang tanpa ada kesalahan pada aplikasi. Pada pengujian performance dilakukan perhitungan performance aplikasi dan didapatkan hasil *Subscriber Throughput* sama dengan 161.55 message/s, *Publisher Throughput* sama dengan 2.13 message/s, rata-rata *message latency Subscriber apps* sama dengan 1488.28 ms dan rata-rata *message latency Publisher apps* sama dengan 5208.86 ms.

Kata kunci: IoT, MQTT, *publish/subscribe*, android, perantara

Abstract

Sensor devices implemented in the field facing a problem when sending data on unstable network conditions to the data center. To solve these problems, we develop an application based on the Android operating system implementing publish/subscribe communication as an intermediary for sending sensor data in the field to the data center using the MQTT protocol. To prove the application can operate as an intermediary for sending sensor data in the field to the data center, three tests are carried out. There are functional testing, scenario testing and performance testing. In functional tests, five functional requirements designed can be fulfilled by the application as an intermediary for sending sensor data to the data center. In scenario tests, from the three test scenarios performed the results of the application can be carried out without any application errors. In performance testing by calculation application performances and get the results of Subscriber Throughput is equal to 161.55 message/s, Publisher Throughput is equal to 2.13 message/s, the latency message average Subscriber apps are 1488.28 ms and the publisher message latency average app is 5208.86 ms.

Keywords: IoT, MQTT, *publish/subscribe*, android, intermediary

1. PENDAHULUAN

Internet of Things (IoT), merupakan konsep jaringan pintar yang menghubungkan berbagai perangkat ke Internet dengan tujuan untuk bertukar informasi dan berkomunikasi dengan perangkat *sensing* dengan menggunakan protokol yang telah disetujui (Chen, dkk., 2014). Terdapat tiga komponen pada pengembangan teknologi berbasis IoT yaitu komponen *hardware* yang terdiri dari sensor, *akutuator*,

dan *hardware* untuk berkomunikasi, komponen *middleware* yang terdiri dari *on demand storage* dan perlengkapan komputasi untuk menganalisis data dan komponen *presentation* yang berperan sebagai visualisasi dan interpretasi yang bisa diakses oleh berbagai *platform* dan bisa dirancang untuk berbagai penerapan (Gubbi, dkk., 2013).

Perangkat sensor dengan teknologi berbasis IoT, memiliki beberapa permasalahan. Permasalahannya yaitu pada *hardware* yang

dipakai oleh *node* sensor memiliki kapasitas penyimpanan yang terbatas untuk menyimpan data yang dihasilkan dari pembacaan lingkungan dan terbatasnya kemampuan untuk pemrosesan data yang kompleks (Emekaroha, dkk., 2015). Dengan munculnya permasalahan tersebut, dibutuhkan sebuah solusi untuk mengurangi beban kapasitas penyimpanan dan juga pemrosesan pada perangkat IoT.

Solusi yang bisa digunakan untuk mengurangi beban penyimpanan dan juga pemrosesan pada perangkat IoT dengan memindahkannya ke sebuah sistem *cloud computing*. *Cloud computing* adalah sebuah model yang menyediakan kebutuhan jaringan yang mudah diakses yang terdiri atas resource komputasi, database, server, dan storage yang dapat dikonfigurasi (Kamboj & Ghuman, 2016). Pada penelitian yang dilakukan oleh Khattab (2016) memanfaatkan *cloud* sebagai tempat menyimpan data, mengolah dan menganalisis data pertanian yang telah didapat menggunakan perangkat IoT yang dikirimkan melalui Internet. Ketika perangkat IoT sudah mendapatkan datanya, akan langsung mengirimkan data tersebut menuju *cloud* untuk disimpan ataupun diolah. Sehingga ketersediaan jaringan komunikasi antara *node* sensor dengan *cloud* memiliki peran yang sangat penting untuk perangkat IoT untuk bisa mengirimkan data menuju layanan *cloud* yang menjadi pusat data.

Untuk mengirimkan data pada sistem IoT dari *node* sensor menuju ke pusat data bisa menggunakan berbagai macam protokol, seperti MQTT, CoAP, AMQP dan HTTP. Keempat protokol ini memiliki kelemahan yang sama yaitu belum bisa mengatasi permasalahan pengiriman data pada kondisi jaringan yang memiliki konektivitas jaringan yang intermiten karena pada protokol-protokol ini harus membangun jaringan end-to-end path sebelum mengirimkan data (Naik, 2017). Kondisi seperti ini bisa ditemui pada *node* sensor yang ditempatkan pada daerah pegunungan atau pedesaan dan pusat data berada pada daerah perkotaan.

Penelitian yang dilakukan oleh Laksana (2018) memanfaatkan aplikasi berbasis sistem operasi android sebagai perantara pengiriman data dari *node* sensor menuju pusat data dengan menggunakan protokol DTN dengan memanfaatkan teknik store and forward. Akan tetapi pada pengimplementasiannya di lapangan

tidak sepenuhnya menggunakan protokol DTN saja. Untuk mengambil data sensor membutuhkan perangkat mobile menggunakan protokol MQTT dan untuk mengirimkan data menuju pusat data juga membutuhkan sebuah perangkat lagi dengan kemampuan upload menggunakan protokol HTTP. Untuk mengimplementasikan sistem ini membutuhkan tiga buah perangkat yang mengimplementasikan tiga protokol yang berbeda.

Dengan mengangkat permasalahan pengiriman data sensor yang berada di lapangan dengan pusat data yang memiliki konektivitas jaringan intermiten. Penelitian ini dilakukan dengan pendekatan praktis yaitu mengembangkan sebuah aplikasi berbasis sistem operasi android yang mengimplementasikan komunikasi publish/subscribe sebagai perantara pengiriman data sensor di lapangan menuju pusat data dengan menggunakan protokol MQTT. Aplikasi dioperasikan pada sebuah perangkat mobile berbasis android. Aplikasi ini bisa langsung mengambil data sensor yang berada di lapangan dengan melakukan perintah subscribe. Setelah data sensor berhasil didapatkan, dengan memanfaatkan teknik store untuk menyimpan data yang telah diambil untuk disimpan pada aplikasi sampai aplikasi bisa terhubung dengan pusat data menggunakan jaringan cellular atau WiFi. Dan jika aplikasi sudah bisa terhubung dengan pusat data, aplikasi akan mengirimkan seluruh data sensor yang telah tersimpan dengan melakukan perintah publish.

2. PENELITIAN TERKAIT

Penelitian yang dilakukan oleh Laksana (2018) yaitu membangun sebuah aplikasi berbasis protokol DTN sebagai perantara pengiriman data sensor dari lapangan dengan pusat data yang memiliki konektivitas jaringan yang intermiten. Sistem yang digunakan ini membutuhkan sebuah perangkat mobile yang mengimplementasikan protokol MQTT untuk mengambil data sensor di lapangan. Setelah data sensor berhasil diambil, perangkat ini akan mengirimkan datanya menuju perangkat mobile berbasis DTN. Apabila perangkat DTN memiliki kemampuan upload, maka perangkat akan mengirimkan data yang telah tersimpan menuju pusat data menggunakan protokol HTTP.

Penelitian yang dilakukan oleh Bauer & Achenbruck (2017) menawarkan sebuah

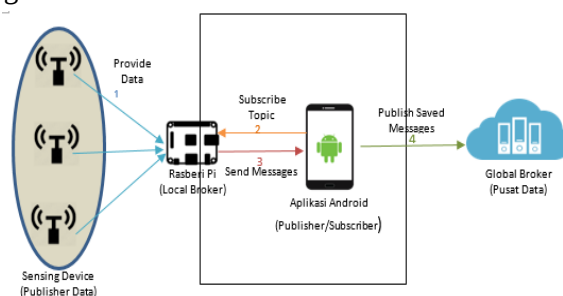
rancangan pertanian pintar dengan menggunakan protokol MQTT untuk mengetahui kapasitas tangki penyimpanan pertanian gandum apakah sudah penuh agar bisa dilakukan proses pengambilan dan pengiriman gandum dengan tepat. Di latar belakang dengan waktu yang dibutuhkan untuk mengambil gandum di tangki dan proses pengirimannya hanya terdapat 1 jalan saja serta lahan pertaniannya yang sangat luas akan menghemat waktu dengan peralatan IoT. Dengan memanfaatkan perangkat *mobile publisher* dan *subscriber* untuk mengetahui apakah jumlah tangki sudah penuh. Sebuah MQTT broker ditempatkan diantara jalan dan tempat penyimpanan gandum untuk komunikasi antara perlengkapan *mobile* dengan tangki penyimpanan gandum yang sudah dilengkapi dengan perangkat sensor. Petani akan meminta kondisi tangki dengan melakukan *subscribe* di jalan besar pertanian, dan tangki akan mengirimkan kondisi kapasitas penyimpanannya dengan melakukan *publish*. Dengan perantara MQTT broker yang telah dipasang pada lahan pertanian, petani akan mendapatkan kondisi tangki kapanpun yang dia inginkan. Ketika tangki sudah penuh petani akan langsung mengambil hasil gandum dan langsung mengirimkannya.

Pada penelitian oleh Khattab, dkk., (2016). Khattab menawarkan sebuah design dan implementasi sistem pertanian yang akurat dengan menggunakan *cloud-based IoT*. Perangkat IoT ini bisa digunakan untuk memonitoring dan melakukan irigasi lahan pertanian, dengan menggunakan *microkontroller* yang bertugas mengumpulkan data yang telah didapat oleh sensor. Sensor yang digunakan yaitu sensor tekanan udara, kelembapan udara, kelembapan tanah, tingkat kebasahan daun, arah dan kecepatan angin dan curah hujan. Data yang dihasilkan oleh masing-masing sensor akan dikumpulkan di *microkontroller*. Data yang telah dikumpulkan tadi akan dikirimkan ke *cloud* menggunakan *gateway* yang tersambung ke Internet menggunakan sinyal WiFi. Dan pada *cloud* data yang diperoleh akan disimpan serta dilakukan analisa untuk mendapatkan visualisasi data yang ada pada pertanian. Dengan menggunakan sistem *cloud* ini para petani dapat mudah melihat kondisi tanaman yang ada pada lahan pertanian dari mana saja melalui perangkat *smartphone* ataupun *desktop* melalui website.

Penelitian oleh Yi, dkk., (2016), Yi menawarkan sebuah design dan implementasi sistem monitoring kesehatan dengan menggunakan protokol MQTT pada perangkat Android *client*. Dengan menggunakan perangkat *embedded* yang telah dilengkapi dengan sensor suhu digital, sensor detak jantung, dan sensor ECG yang disatukan pada perangkat *embedded* untuk mendapatkan data kesehatan. Dengan menggunakan protokol MQTT data tersebut dikirimkan ke *cloud server* untuk disimpan. Perangkat *embedded* akan mengirimkan data yang telah didapatkan menuju *cloud server* untuk diolah dan mempresentasikan data tersebut pada sebuah web yang sudah dirancang. Pada perangkat Android *client* terdapat aplikasi menggunakan protokol MQTT yang dapat mengambil data kesehatan seseorang yang dengan melakukan *subscribe* ke *cloud server*. Dengan adanya aplikasi Android ini akan mempermudah melakukan pengecekan data kondisi kesehatan pasien jika diimplementasikan pada rumah sakit dimana perawat bisa mengecek kesehatan pasien melalui *smartphone* mereka.

3. PERANCANGAN SISTEM

Pada penelitian ini dirancang sebuah aplikasi Android dengan bahasa pemrograman *java* dimana aplikasi ini akan digunakan untuk mengambil pesan yang berada pada *node* sensor yang menghasilkan data dengan perantara *Local Broker* pada rasberi pi dan akan meneruskannya ke *cloud Global Broker*. Berikut ini adalah gambaran desain sistem yang dirancang pada gambar 3.1.



Gambar 3.1 Rancangan Desain Sistem

Berikut ini adalah komponen-komponen yang digunakan untuk membangun sistem:

1. Sensing Device

Node sensor yang digantikan dengan *script python* yang dapat menghasilkan data dummy berupa angka *integer* atau data *string* yang akan dikirimkan menuju *Local*

Broker yang berperan sebagai *publisher* data.

2. Local Broker

Sebuah Raspberi Pi sebagai *access point* dan *broker* yang menjadi *gateway* dan server yang digunakan oleh *node* sensor dan aplikasi Android dalam proses pertukaran pesan. *Broker* yang digunakan menggunakan *library paho MQTT*.

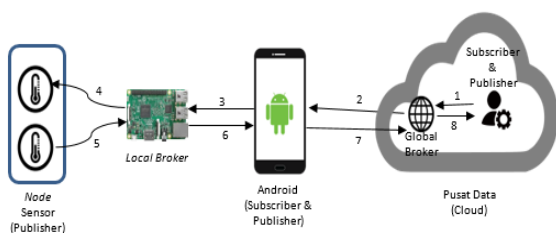
3. Aplikasi Android

Digunakan sebagai perantara pengiriman data dari *node* sensor menuju pusat data. Data sensor yang telah dikirimkan menuju *Local Broker* akan diambil oleh aplikasi Android menggunakan jaringan lokal. Dan pada saat mengirimkan data menuju pusat Aplikasi android mengirimkannya menggunakan jaringan *cellular* ketika terhubung dengan *Global Broker* pada pusat data.

4. Pusat Data

Pusat data terdiri dari tiga komponen yaitu *Global Broker*, *client subscriber* dan *client publisher*. *Global Broker* yang merupakan layanan *broker* yang tersedia secara *online* yang terdapat pada Internet. Layanan *broker* yang digunakan yaitu *broker online* milik *HiveMQ*. *Client Subscriber* digunakan pusat data sebagai peminta data yang telah dikirimkan oleh aplikasi Android. *Client publisher* digunakan oleh pusat data untuk mengirimkan pesan menuju *node* sensor melalui perantara aplikasi Android.

Perancangan alur komunikasi pada sistem ini ada empat bagian utama yang menjalankan tugasnya masing masing, yaitu *node* sensor menjadi *publisher* data, Rasperry pi 3 model B menjadi *Local Broker*, Perangkat Android sebagai *client subscriber* sekaligus *publisher*, dan pusat data berbasis *cloud* sebagai tujuan akhir data. Perancangan alur komunikasi sistem dapat dilihat pada gambar 3.2.



Gambar 3.2 Alur Komunikasi Sistem

Berikut keterangan angka yang berada pada gambar:

1. Terdapat dua komponen *client* pada cloud yaitu *Subscriber* dan *Publisher*. *Subscriber* pada *Global Broker* melakukan *subscribe* topic sesuai dengan topic pada sensor. *Publisher* mengirimkan pesan perintah yang ingin disampaikan kepada perangkat sensor melalui *Global Broker*.
2. Aplikasi Android menerima pesan perintah dari pusat data melalui *Global Broker* dengan melakukan *subscribe*.
3. Aplikasi Android terhubung dengan *Local Broker*, kemudian aplikasi akan melakukan *subscribe* topic sesuai dengan sensor untuk bisa menerima data yang dikirimkan oleh sensor dan juga akan melakukan *publish* untuk mengirimkan pesan perintah dari pusat data.
4. *Node Sensor subscribe* pesan perintah dari pusat data melalui *Local Broker*.
5. *Node Sensor* melakukan *publish* data sensor menuju *Local Broker*.
6. Ketika topic sesuai data akan diterima dan disimpan pada perangkat Android dengan melakukan *subscribe*.
7. Mengecek jika bisa terhubung dengan *Global Broker*, data yang disimpan akan di *publish*.
8. Data yang dikirimkan oleh aplikasi Android diterima oleh *subscriber* pada pusat data

4. IMPLEMENTASI SISTEM

Dari desain sistem yang telah dirancang diimplementasikan sebuah aplikasi android yang digunakan sebagai perantara pengiriman data sensor dilapangan menuju pusat data. Berikut ini adalah beberapa fungsional aplikasi yang telah diimplementasikan.

1. Aplikasi mencari dan terkoneksi dengan Local Broker

Pada fungsional ini aplikasi bisa mencari perangkat *local broker* yang digunakan sebagai *gateway* oleh *node* sensor dan dapat terkoneksi dengan jaringan *local broker*. Berikut ini adalah *pseudocode* yang diimplementasikan pada tabel 4.1.

Tabel 4.1 *Pseudocode* aplikasi mencari dan terkoneksi dengan *Local Broker*

```
Run Search
if LocalBroker is Visible
```

```
Get Wifi Configuration
Connect
```

```
if Connect Success
Stay Connected
else if Connect Fail
Loop Reconnect
```

```
else
Loop Search
```

2. Aplikasi menjalankan service 1 (Publish/Subscribe dari node sensor menuju Pusat data).

Pada fungsional ini aplikasi dapat menjalankan service 1, dimana service ini digunakan aplikasi menjalankan fungsi subscribe untuk mengambil data sensor yang telah dikirimkan oleh node sensor ketika terhubung dengan jaringan local broker. Dan aplikasi akan melakukan publish untuk mengirimkan data yang telah tersimpan ketika aplikasi terhubung dengan global broker yang digunakan oleh pusat data dengan menggunakan jaringan cellular 4G atau WiFi. Berikut ini adalah pseudocode yang diimplementasikan pada tabel 4.2.

Tabel 4.2 Pseudocode aplikasi menjalankan service 1

```
Run Service 1
if settings equals True
GetPreferenceSetting

if LocalBroker state equals
Connected
Run Subscriber Client
Subscribe Get All Sensor Data
Save data to database
else
Idle Until Connected

if GlobalBroker state equals
Connected
Run Publisher Client
Get All Unsent Data from
database
Publish All Unsent Data
Change Sent Status
else
Idle Until Connected

else
check your configuration
```

3. Aplikasi melihat data yang berhasil di dapat pada database.

Pada fungsional ini aplikasi dapat melihat data yang berhasil didapat oleh aplikasi.

Berikut ini adalah pseudocode yang diimplementasikan pada tabel 4.3.

Tabel 4.3 Pseudocode aplikasi melihat data yang berhasil di ambil pada database

```
MenuExtra run
if Choose Data From
Display DataService 1
Get All Data from Database
else
Display DataService 2
Get All Data From Database
```

Pada implementasi node sensor yang terdiri atas script publisher dan subscriber menggunakan bahasa pemrograman python. Pada publisher node sensor bisa mengirimkan data yang dihasilkan dari fungsi random berupa angka dengan besaran antara 10 sampai 100. Berikut ini pseudocode dari node sensor yang digunakan pada tabel 4.4.

Tabel 4.3 Pseudocode subscriber node sensor

```
DEFINE function on_connect()
DO connect to broker
SET broker_address :192.168.10.1
SET port :1883
SET client :Sensor1
DO client.on_connect

DO client.loop_start()

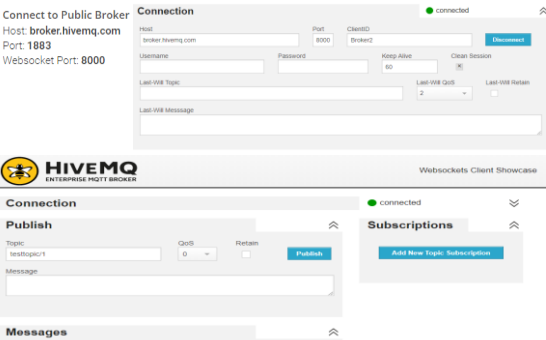
IF connect != true
DO sleep(0.1)
SET topic : raw_input
SET delay : raw_input
SET total message : raw_input
IF true
DO total message
SET message = randomint(10,100)
DO client.publish

(topic,message,qos=2,retain=False
)

DO print message
DO print times sent
DO print size
Do sleep(delay)
```

```
ELSE keyboardinterrupt
DO client.disconnect()
DO client.loop_stop()
END
END
```

Pusat data yang digunakan pada sistem terdiri atas *broker* yang berbasis cloud dan client subscriber untuk meminta data yang dikirimkan oleh aplikasi Android. Layanan *broker* yang digunakan adalah HiveMQ, *broker* online yang bisa dipakai bebas secara gratis. Pada HiveMQ sudah tersedia broker dengan berbasis protokol MQTT atau websocket dan juga bisa membuat client subscriber sebanyak apapun pada web HiveMQ yang dapat diakses melalui aplikasi browser yang dapat dilihat pada gambar 4.1



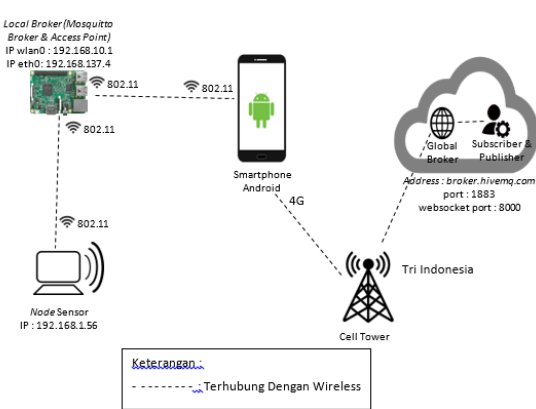
Gambar 4.1 Implementasi Pusat Data

5. PENGUJIAN

Topologi pengujian dibutuhkan untuk mengetahui setiap komponen sistem yang berinteraksi pada saat melakukan pengujian. Alur dimulai dengan *node* sensor yang dibuat dengan *script* mengirimkan data menuju *local broker*.

Lalu aplikasi pada *smartphone* Android akan mengambil data tersebut dari *broker*. Setelah pesan selesai diambil *smartphone* akan memutus koneksi dengan *local broker*. Sambil mencari sinyal cellular yang dipancarkan *cell tower* data yang telah diambil akan disimpan pada aplikasi.

Apabila aplikasi mendapatkan sinyal dan dapat terhubung dengan *Global Broker* pada pusat data, aplikasi akan mengirimkan data yang tersimpan tadi menuju *Global Broker* dan *subscriber* akan mengambil data yang telah dikirim pada pusat data. Berikut ini gambar topologi pengujian dapat dilihat pada gambar



5.1.

Gambar 5.1 Topologi Pengujian

Dengan menggunakan teknik *blackbox testing* dilakukan pengujian fungsional terhadap aplikasi yang telah diimplementasikan dan didapatkan hasil pengujian yang bisa dilihat pada tabel 4.1.

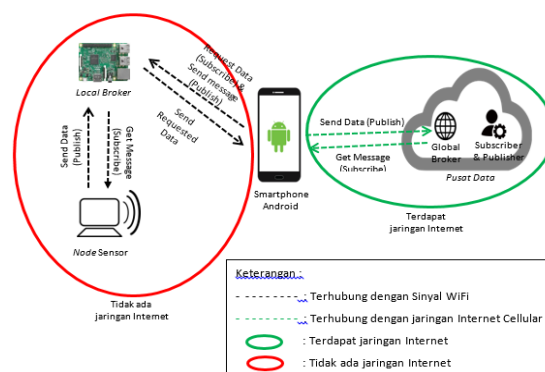
Tabel 5.1 Hasil Pengujian Fungsional Aplikasi

No	Kode	Kebutuhan Fungsional	Hasil	Status
1	F_01	Aplikasi Android dapat mencari perangkat <i>Local Broker</i> dan dapat terhubung.	Aplikasi Android berhasil mencari perangkat <i>Local Broker</i> dan bisa terhubung dengan menggunakan jaringan WiFi.	VALID
2	F_02	Aplikasi Android dapat menjalankan 2 service <i>Publish/Subscribe</i> .	Aplikasi Android berhasil menjalankan 2 service <i>Publish/Subscribe</i> secara bersamaan.	VALID
3	F_03	Service 1 : <i>Publish/Subscribe</i> dari <i>node</i> sensor menuju pusat data a. Melakukan <i>subscribe</i> dari <i>node</i> sensor. b. Melakukan <i>publish</i>	a. Aplikasi Android berhasil melakukan <i>subscribe</i> dari <i>node</i> sensor untuk mengambil data. b. Aplikasi Android berhasil melakuka	VALID

		menuju pusat data.	n publish menuju pusat data untuk mengirim data node sensor yang telah disimpan	
4	F_04	Service 2 : Publish/Subscribe dari pusat data menuju node sensor:	a. Aplikasi Android berhasil melakukan subscribe dari pusat data untuk meminta pesan dari pusat data.	VALID
		a. Melakukan subscribe dari pusat data.	b. Aplikasi Android berhasil melakukan n publish menuju node sensor untuk mengirim pesan dari pusat data yang telah disimpan	VALID
5	F_05	Aplikasi Android dapat menyimpan dan melihat data yang telah didapat pada database aplikasi.	Aplikasi berhasil menyimpan data dari node sensor dan juga dari pusat data serta bisa melihat isi dari masing masing data pada database aplikasi.	VALID

Pengujian skenario disini dilakukan dengan melakukan pengujian berdasarkan skenario yang mungkin terjadi pada saat menjalankan aplikasi pada saat berkomunikasi dengan node sensor dan pusat data. Berikut ini skema pengujian yang

akan dilakukan pada gambar 5.2.



Gambar 5.2 Skema Pengujian Skenario

Dari skema pengujian diatas akan dilakukan tiga kondisi pengujian yang dilakukan dan didapatkan hasil sebagai berikut:

1. **Skenario satu: Node Sensor mengirimkan data ketika perangkat android sedang offline.**

Dilakukan percobaan semua data yang telah dikirimkan sensor sebelumnya berhasil diterima seluruhnya oleh aplikasi android. Dengan hasil ini menunjukkan bahwa aplikasi berhasil dijalankan pada skenario satu.

2. **Skenario dua: Koneksi perangkat android terputus pada saat ditengah-tengah melakukan subscribe data sensor dari Local Broker.**

Data yang dikirimkan oleh node sensor sebanyak 100 data, data yang diterima oleh aplikasi pada saat sebelum koneksi terputus sebanyak 45 data, dan data yang berhasil diterima oleh aplikasi setelah terhubung kembali sebanyak 55 data. Jumlah pesan yang diterima tetap sama yaitu 100 data dan tidak ada data yang dibuang ketika koneksi perangkat android terputus pada saat melakukan subscribe. Dengan hasil ini menunjukkan bahwa aplikasi berhasil dijalankan pada skenario dua.

3. **Skenario tiga: Koneksi perangkat android terputus dan tersambung berulang kali pada saat ditengah-tengah melakukan publish data sensor menuju pusat data.**

Seluruh data yang berjumlah 100 data berhasil terkirim walaupun pada saat melakukan proses publish sering terjadi gangguan koneksi cellular ataupun terputus. Jumlah pesan yang diterima tetap sama yaitu 100 data dan tidak ada data yang dibuang

atau redudansi pada pusat data ketika koneksi perangkat android terputus pada saat melakukan *publish*. Dengan hasil ini menunjukkan bahwa aplikasi berhasil dijalankan pada skenario tiga.

Pengujian performance disini dilakukan untuk mengetahui performance dari aplikasi *publish/subscribe* yang telah dirancang pada sistem operasi android. Berikut ini adalah parameter yang digunakan untuk menguji performance dari aplikasi android:

1. Subscriber Throughput

Mengetahui kemampuan dari *client subscriber* pada aplikasi android dalam menerima data dari *node* sensor setiap detiknya yang dapat dihitung dengan menggunakan persamaan 5.1.

$$Ts = \frac{N}{t} \quad (5.1)$$

Ts = Subscriber Throughput

N = Jumlah total pesan diterima

t = Waktu pengiriman pesan

2. Publisher Throughput

Mengetahui kemampuan dari *client publisher* pada aplikasi android dalam mengirimkan data menuju pusat data setiap detiknya yang dapat dihitung dengan menggunakan persamaan 5.2.

$$Tp = \frac{N}{t} \quad (5.2)$$

Tp = Publisher Throughput

N = Jumlah total pesan terkirim

t = Waktu pengiriman pesan

3. Message Latency

Mengetahui waktu yang dibutuhkan *client subscriber* dan *publisher* pada aplikasi android untuk menerima data dari *node* sensor dan mengirimkannya ke pusat data yang dapat dihitung menggunakan persamaan 5.3.

$$\Delta t = tS - t \quad (5.3)$$

Δt = Message Latency

tS = Waktu Subscriber menerima pesan

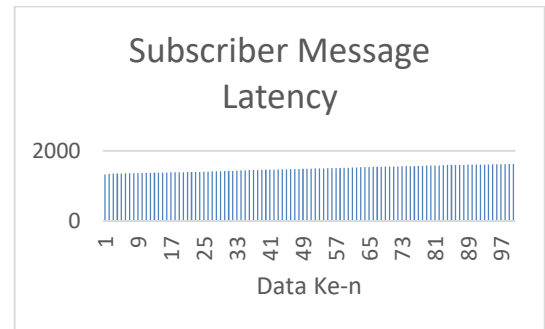
tP = Waktu Publishser mengirimkan pesan

Dari hasil perhitungan pengujian performance didapatkan hasil akhir performance aplikasi *publish/subscribe* pada perangkat android yaitu:

1. Subscriber Throughput = 161.55 message/s

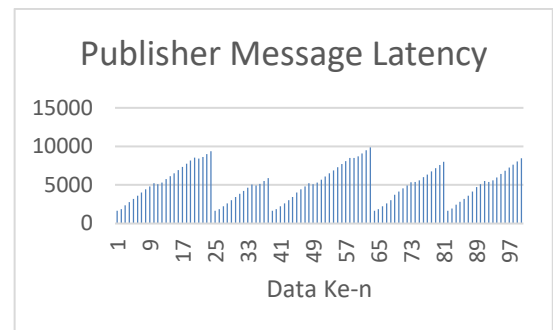
2. Publisher Throughput = 2.13 message/s

3. Message latency Subscriber apps



Rata-rata = 1488.28 ms / 1.488 s

4. Message latency Publisher apps



Rata-rata = 5208.86 ms / 5.208 s

6. PENUTUP

Berdasarkan implementasi dan analisis hasil pengujian implemtasi perangkat *mobile publisher subscriber* sebagai perantara pengiriman data sensor dari lapangan ke pusat data, didapatkan beberapa kesimpulan sebagai berikut:

Untuk mengambil data sensor yang berada dilapangan dengan perangkat *mobile*, terdapat aplikasi yang mengimplementasikan *subscriber client* menggunakan protokol MQTT sebagai penerima data pada perangkat *mobile*. Dengan melakukan *subscribe* data sensor dapat diterima oleh aplikasi android dan disimpan pada perangkat *mobile*.

Untuk mengirimkan data sensor yang tersimpan sebelumnya menuju pusat data dengan perangkat *mobile*, terdapat aplikasi yang mengimplementasikan *publisher client* menggunakan protokol MQTT sebagai pengirim data yang telah disimpan sebelumnya pada perangkat *mobile*. Dengan melakukan *publish* data sensor yang telah tersimpan pada perangkat *mobile* dapat dikirimkan menuju ke pusat data.

Dari hasil pengujian aplikasi didapatkan beberapa hasil berikut:

Berdasarkan hasil pengujian fungsional yang dilakukan, lima kebutuhan fungsional yang dirumuskan semuanya berhasil dipenuhi oleh aplikasi dan dapat diimplementasikan sebagai perantara pengiriman data sensor dari lapangan ke pusat data.

Pada pengujian skenario, aplikasi mampu dijalankan pada tiga skenario yang telah dilakukan tanpa ada kesalahan pada aplikasi dan redundansi data pada aplikasi serta pusat data.

Dari hasil perhitungan pengujian performance aplikasi didapatkan hasil:

1. *Subscriber Throughput* = **161.55 message/s**
2. *Publisher Throughput* = **2.13 message/s**
3. *Rata-rata message latency Subscriber apps* = **1488.28 ms / 1.488 s**
4. *Rata-rata message latency Publisher apps* = **5208.86 ms / 5.208 s**

Saran pada penelitian ini untuk melakukan perbaikan atau pengembangan lebih lanjut adalah sebagai berikut:

Perbaikan pada *client publisher* aplikasi ketika menggunakan QoS level 2 untuk mengirimkan data menuju pusat data, *latency* yang dihasilkan tidak terus bertambah dari pengiriman data sebelumnya.

Implementasi aplikasi *mobile* sebagai perantara pengiriman data sensor dilapangan ke pusat data dengan menggunakan protokol-protokol lain untuk bisa dijadikan sebagai perbandingan.

7. DAFTAR PUSTAKA

- Bauer, J. & Aschenbruck, N., 2017. Measuring and Adapting MQTT in Cellular Networks for Collaborative Smart Farming. *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*, pp. 294 - 302.
- Chen, S. et al., 2014. A Vision of IoT: Applications, Challenges, and Opportunities With China Perspective. *IEEE Internet of Things Journal*, 1(4), pp. 349 - 359.
- Emeakaroha, V. C., Cafferkey, N., Healy, P. & Morrison, J. P., 2015. A Cloud-Based IoT Data Gathering and Processing Platform. *2015 3rd International Conference on Future Internet of Things and Cloud*, pp. 50-57.
- Gubbi, J., Buyya, R., Marusic, S. & Palaniswami, M., 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), pp. 1645-1660.
- Kamboj, S. & Ghumman, N. S., 2016. A Survey on Cloud Computing and Its Type. *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 2971 - 2974.
- Khattab, A., Abdelgawad, A. & Yelmarthi, K., 2016. Design and implementation of a cloud-based IoT scheme for precision agriculture. *2016 28th International Conference on Microelectronics (ICM)*, pp. 201-204.
- Laksana, H. D., 2018. RANCANG BANGUN PERANGKAT MOBILE BERBASIS DELAY TOLERANT NETWORK SEBAGAI PERANTARA PENGIRIMAN DATA SENSOR DARI LAPANGAN KE PUSAT DATA, Malang: Universitas Brawijaya.
- Naik, N., 2017. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. *2017 IEEE International Systems Engineering Symposium (ISSE)*, pp. 1-7.
- Yi, D., Binwen, F., Xiaoming, K. & Qianqian, M., 2016. Design and Implementation of Mobile Health Monitoring System based on MQTT Protocol. *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pp. 1679 - 1682.