

## Pengembangan Interface Bluetooth Low Energy (BLE) Pada IoT Middleware Untuk Mendukung Network Interoperability

Rasidy Cakra Pratama<sup>1</sup>, Eko Sakti Pramukantoro<sup>2</sup>, Achmad Basuki<sup>3</sup>

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya  
Email: <sup>1</sup>rasidy.pratama@student.ub.ac.id, <sup>2</sup>ekosakti@ub.ac.id, <sup>3</sup>abazh@ub.ac.id

### Abstrak

Pada penelitian terdahulu dikembangkan sebuah *middleware* dengan pendekatan *event-driven* yang mampu mendukung *interoperabilitas* berbagai macam perangkat atau sensor. Dari hasil pengujian *middleware* yang diusulkan mampu mengatasi masalah *syntatic interoperability* dengan menyediakan *gateway* untuk berkomunikasi dengan perangkat sensor IoT menggunakan protokol MQTT dan CoAP, serta mampu berkomunikasi dengan aplikasi lain menggunakan protokol Websocket. Berkaca dari solusi yang telah ditawarkan sebelumnya untuk mengatasi masalah interoperabilitas, diusulkan sebuah jalur transmisi lain yaitu *Bluetooth Low Energy (BLE)* yang akan melakukan publish data ke *Middleware*. Untuk mengetahui kinerja dari transmisi jaringan BLE dilakukan pengujian dengan beberapa parameter yaitu delay, konsumsi *bandwidth*, serta CPU dan *memory usage*. Hasil pengujian menunjukkan rata – rata penggunaan CPU adalah 31% dan penggunaan memori adalah 5%. Delay yang diterima saat menggunakan transmisi BLE mencapai 0,04523 detik dan saat menggunakan transmisi BLE bersamaan dengan transmisi WiFi mencapai 0,032407248 detik. Dari segi konsumsi *Bandwidth* saat menggunakan transmisi BLE mencapai 7786,6bits/detik dan saat menggunakan Transmisi BLE bersama dengan Transmisi WiFi menjadi dua kali lipatnya. Dari hasil penelitian menunjukkan bahwa *Bluetooth Low Energy* dapat digunakan untuk mengatasi masalah *Interoperabilitas*.

**Kata kunci:** BLE, Interface, MQTT, IoT, Middleware, Bandwidth, Interoperabilitas

### Abstract

*In an earlier research developed an event-driven middleware with the ability to support the interoperability of various devices or sensors. From the results of the proposed middleware testing is able to overcome the problem of syntatic interoperability by providing gateway to communicate with IoT sensor device using MQTT and CoAP protocol, and able to communicate with other application using Websocket protocol. Reflecting on the previously offered solution to solve the interoperability problem, it is proposed another transmission line that is Bluetooth Low Energy (BLE) which will publish data to Middleware. To know the performance of BLE network transmission is tested with several parameters ie delay, bandwidth consumption, and CPU and memory usage. The test results show average CPU usage is 31% and memory usage is 5%. Delay received when using BLE transmission reaches 0.04523 seconds and when using BLE transmission along with WiFi transmission reach 0.032407248 seconds. In terms of consumption Bandwidth when using BLE transmissions reached 7786,6bits / sec and when using BLE Transmission along with WiFi Transmission doubled. From the research results show that Bluetooth Low Energy can be used to solve Interoperability problems.*

**Keywords:** BLE, Interface, MQTT, IoT, Middleware, Bandwidth, Interoperability

## 1. PENDAHULUAN

*Internet of Things (IoT)* telah berkembang pesat dan memiliki dampak yang cukup besar dalam kehidupan sehari – hari. IoT memungkinkan penggunaanya dalam mengakses serta mengelola peralatan elektronik

secara wireless melalui internet. *Internet of Things (IoT)* telah banyak diterapkan menggunakan perangkat sensor yang berbeda – beda. Contohnya pada penelitian terdahulu (Anwari, 2016) dikembangkan sebuah *middleware* dengan pendekatan *event-driven* yang mampu mendukung *interoperabilitas*

berbagai macam perangkat atau sensor. Dari hasil pengujian *middleware* yang diusulkan mampu mengatasi masalah *syntactic interoperability* dengan menyediakan *gateway* untuk berkomunikasi dengan perangkat sensor IoT menggunakan protokol MQTT dan CoAP, serta mampu berkomunikasi dengan aplikasi lain menggunakan protokol WebSocket. Pada implementasinya antara *middleware* dengan sensor menggunakan media transmisi wireless dengan protokol komunikasi CoAP dan MQTT. Serta pada penelitian berjudul *Topic based IoT data storage framework for heterogenous sensor data* (Pramukantoro, 2017), Mengimplementasikan proses pengiriman data didalam IoT menggunakan protokol MQTT dan QOAP dan disimpan menggunakan database yang berbeda – beda penelitian tersebut membuktikan bahwa penggunaan database yang berbeda – beda dapat diterapkan dalam IoT. Pada ekosistem IoT dibutuhkan kemampuan *Interoperabilitas* untuk menciptakan kinerja yang baik dari setiap perangkat IoT agar dapat menjalankan potensinya secara maksimal untuk terhubung satu sama lain. IoT dalam prosesnya harus dapat di komunikasikan, dijalankan, dan diprogram dengan menggunakan perangkat apa saja terlepas dari versi, model manufaktur, atau industry manapun, intinya konektivitas antar perangkat tidak perlu memerlukan usaha tambahan dari aplikasi atau pengembang. IoT dalam kenyataannya interoperabilitas merupakan masalah yang paling sering dihadapi dalam membangun IoT dikarenakan perangkat yang satu tidak dapat saling berkomunikasi dengan perangkat yang lain, terdapat tiga macam *interoperability* yaitu *syntactic interoperability* (Razzaque, 2016). Dari pembahasan sebelumnya, maka pada penelitian ini bertujuan untuk menambahkan komunikasi IoT menggunakan transmisi BLE pada *middleware* untuk menambah skalabilitas komunikasi agar dapat berkomunikasi dengan perangkat selain wifi dan Ethernet sehingga dapat menjawab permasalahan dalam kemampuan *Network Interoperability*. Digunakannya BLE pada penelitian ini karena BLE mudah dalam proses pairing, konsumsi daya yang dikeluarkan sedikit, serta transmisi BLE mendukung untuk penelitian skala kecil. Penelitian ini akan dimulai dengan membangun node sensor dan perangkat gateway dengan interface Bluetooth Low Energy, menambahkan fungsi komunikasi BLE pada IoT *middleware*, dan menguji performansi komunikasi node sensor dengan *middleware*

menggunakan transmisi BLE.

## 2. LANDASAN KEPUSTAKAAN

### 2.1. Kajian pustaka

Pada penelitian sebelumnya A BLE-based data collection system for IoT, penelitian tersebut membahas menggunakan BLE sebagai data collector untuk mengambil data dari sistem.(Nouali, 2015). Persamaan penelitian diatas dan adalah penggunaan jaringan BLE sebagai media komunikasi pengiriman data. pada penelitian tersebut, penulis memanfaatkan penggunaan BLE sebagai jaringan untuk mengambil data dari beberapa sensor

Pada penelitian yang kedua, dengan judul Bluetooth Low Energy (BLE) based Wireless sensor (Mackensen, 2013). Pada penelitian ini dilakukan penelitian menggunakan BLE dengan basis wireless atau tanpa kabel, penelitian tersebut dijadikan dasar bahwa jaringan BLE dapat digunakan sebagai media pengiriman data tanpa kabel.

Penelitian yang terakhir dengan judul *Topic based IoT data storage framework for heterogenous sensor data* (Pramukantoro, 2017). Pada penelitian tersebut mengimplementasikan proses pengiriman data didalam IoT menggunakan protokol MQTT dan QOAP dan disimpan menggunakan database yang berbeda – beda.

## 3. PERANCANGAN SISTEM

Tujuan utama dari *middleware* yang dikembangkan adalah menyediakan *BLE Gateway* untuk node mengirimkan data melalui protokol MQTT untuk mendukung network Interoperability.

### 3.1 Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang wajib terpenuhi. Berikut adalah kebutuhan fungsional pada penelitian ini dan akan dijelaskan dalam tabel berikut :

Tabel 1. Kebutuhan Fungsional

No	Kebutuhan Fungsional
1.	Sensor dapat Terkoneksi dengan <i>Generic Attribute Profile</i> <i>middleware</i> .
2.	<i>Generic Attribute Profile</i> <i>middleware</i> dapat mendeteksi koneksi dari BLE sensor.
3.	BLE <i>gateway</i> dapat terkoneksi dengan sensor

- node
- 4. BLE Gateway dapat menerima data yang di *publish* oleh sensor node melalui jaringan BLE
- 5. BLE gateway dapat mentranslasikan data dari sensor node kedalam protokol MQTT.
- 6. BLE gateway dapat mempublish data kedalam Redis Middleware melalui protocol MQTT.
- 7. Redis Middleware dapat Merima data yang dikirimkan BLE gateway melalui protocol MQTT.

### 3.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional ini terbagi menjadi 2 yaitu kebutuhan perangkat keras dan perangkat lunak.

Tabel 1. Kebutuhan Perangkat Keras

Perangkat	Keterangan
Raspberry Pi 3	Raspberry pi 3 digunakan sebagai perangkat yang akan menjadi penghubung antara sensor melalui Bluetooth dan menjalankan middleware.
ESP32	ESP32 adalah merupakan microcontroller Bluetooth 4.0 serta wifi yang akan dipasangkan dengan Arduino untuk dapat mengirimkan data.
ASUS X45U	Perangkat ini digunakan untuk memonitor data dari redis

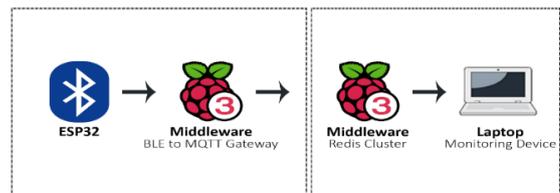
Tabel .2 Kebutuhan Perangkat Lunak

Perangkat	Keterangan
Raspbian Jessie	OS yang digunakan pada Raspberry Pi 3
Node.js	Framework yang digunakan untuk proses mengembangkan middleware dengan berbasis event-driven.
Redis	Media penyimpanan data yang bertindak sebagai broker, proses penyimpanan data terjadi pada memori
Arduino IDE	Merupakan IDE untuk sketch program yang akan diimplementasikan pada Arduino UNO.
ESP-IDF	ESP-IDF merupakan framework yang digunakan untuk ESP32.
EspruinoHub	EspruinoHub merupakan BLE gateway sekaligus Bridge untuk yang berfungsi untuk mentranslasikan transmisi BLE menjadi MQTT
Bluez GATT	Bluez digunakan sebagai GATT yang ditanamkan di middleware agar middleware dapat

berkomunikasi dengan sensor node melalui jaringan BLE

### 3.3 Perancangan Alur Komunikasi

Terdapat 3 komponen yang berinteraksi pada sistem ini yaitu ESP32 sebagai *Publisher*, Raspberry Pi 3 sebagai *Middleware* dan Laptop sebagai media *Subscriber* untuk meminta data dengan cara menjalankan redis.



Gambar 1. Alur Komunikasi BLE

Pada sistem ini data akan dikirimkan ke *middleware* melalui koneksi Bluetooth low energy. Untuk melakukan koneksi dari non-IP ke IP ditambahkan Bluetooth gateway pada *Middleware* yang kemudian akan melakukan translasi BLE untuk di *publish* menggunakan MQTT sehingga dapat menerima data dari sensor. Untuk dapat mengambil data dari sensor, laptop terlebih dahulu harus terhubung ke jaringan access point milik *middleware*. Setelah itu laptop dapat melakukan monitoring ke *middleware* dengan membuka *pm2 logs* untuk mendapatkan data.

## 4. IMPLEMENTASI

Bagian ini menjelaskan bagaimana sistem dikembangkan dari dasar. Dimulai dari Implementasi *Middleware*, Implementasi GATT, implementasi BLE Gateway, hingga Implementasi Sensor NodeMCU.

### 4.1 Implementasi Middleware

Pada Konfigurasi *Middleware* akan ditanamkan OS yang akan mendukung implementais sistem yaitu Raspbian Jessie serta dilakukan Instalasi redis sebagai database dan broker.

### 4.2 Implementasi GATT

Pada tahap ini akan dilakukan proses konfigurasi GATT *middleware* . GATT yang akan digunakan adalah Bluez Gatt yang dapat berjalan pada Raspberry Pi yang memiliki OS Raspbian Jessie.

### 4.3 Implementasi BLE Gateway

Sesuai dengan kebutuhan sistem yang telah dibahas pada bab sebelumnya, bahwa akan dikembangkan BLE gateway yang akan ditanamkan pada middleware. BLE gateway yang akan digunakan pada penelitian ini adalah EspruinoHub.

### 4.4 Implementasi NodeMCU

Node masih belum bisa berjalan karena tidak terdapat logic yang menggerakannya pada sensor node ESP32. Untuk membuat node bekerja, hal yang harus dilakukan pertama kali adalah menginstal Arduino IDE dan memasukan Library untuk node serta modul yang digunakan.

## 5. PENGUJIAN DAN ANALISIS

Pengujian merupakan teknik yang dilakukan peneliti untuk mendapatkan sebuah hasil berupa nilai berdasarkan fokus penelitian, dalam penelitian ini peneliti fokus pada pengukuran delay dan pengaruh penerapan jaringan BLE pada node sensor. pengukuran dilakukan secara langsung dengan mengamati data *capture*. Sehingga diketahui seberapa besar delay yang diterima oleh tiap proses yang dilakukan oleh node. Proses yang diukur dalam hal ini adalah proses membaca delay dari jarak yang berbeda – beda.

Pengujian dilakukan dengan beberapa skenario pengujian. Skenario pengujian bermanfaat untuk mendapatkan data dari hasil pengukuran dengan beberapa skenario, Karena dengan menggunakan skenario dapat dijadikan sebagai representasi keadaan ketika sistem diterapkan sebagaimana fungsinya. Pada tahapan ini dilakukan proses perancangan skenario pengukuran yang terdiri dari beberapa skenario. Untuk tiap skenario, peneliti menggunakan parameter ukur yang berbeda - beda yaitu *Delay*, *Bandwidth Consumption*, dan CPU dan *Memory Usage*.

### 5.1 Pengujian Delay (BLE+WiFi)

Pengujian ini dilakukan untuk mengetahui delay dari transmisi BLE pada saat pengiriman data, serta pada saat transmisi BLE berjalan bersama dengan transmisi WiFi untuk mengirimkan data. Terdapat dua Skenario pengujian delay dengan menggunakan cara yang berbeda, untuk skenario pertama dilakukan pengujian terhadap delay pada saat pengiriman data menggunakan transmisi BLE, dan yang kedua dilakukan pengujian delay pada saat

pengiriman data menggunakan transmisi BLE bersamaan dengan Wifi juga melakukan pengiriman data ke *Middleware*.

### Skenario 1: Delay transmisi BLE

Dalam pengujian ini, penguji melakukan pengiriman data RSSI dari Node ke *Middleware* menggunakan transmisi BLE selama 10 menit.

No.	Time	Source	Destinatio	Protocol	Length	Delta Time
52	2.863877	controller	host	HCI_EVT	7	0.000465
53	2.889952	controller	host	HCI_EVT	46	0.083551
54	2.923607	controller	host	HCI_EVT	46	0.020021
55	2.93433	controller	host	HCI_EVT	46	0.078765
56	2.944206	controller	host	HCI_EVT	46	0.104257
57	3.0297	controller	host	HCI_EVT	46	0.030729

Gambar 2. Sample Pengujian Delay BLE

Pada gambar 2 menunjukkan potongan data yang dicapture menggunakan aplikasi wireshark untuk mendapatkan data delta time yang berupa delay dari proses pengiriman data menggunakan transmisi BLE yang dilakukan selama 10 menit.

### Skenario 2: Delay Transmisi BLE + Wifi

Dalam pengujian ini, penulis melakukan pengirim data RSSI dari Node ke *Middleware* menggunakan transmisi BLE selama 10 menit bersamaan dengan adanya proses pengiriman data yang dikerjakan oleh Node lain menggunakan transmisi WiFi ke *Middleware*.

No.	Time	Source	Destinatio	Protocol	Length	Delta Time
52	2.863877	controller	host	HCI_EVT	7	0.000439
53	2.889952	controller	host	HCI_EVT	46	0.026075
54	2.923607	controller	host	HCI_EVT	46	0.033655
55	2.93433	controller	host	HCI_EVT	46	0.010723
56	2.944206	controller	host	HCI_EVT	46	0.009876

Gambar 3. Sample Pengujian Delay BLE + WiFi

Pada gambar 6.21 menunjukkan sample data yang dicapture menggunakan aplikasi wireshark untuk mendapatkan data delta time yang berupa delay dari proses pengiriman data menggunakan transmisi BLE bersamaan dengan pengiriman data yang dilakukan oleh transmisi WiFi selama 10 menit.

### 5.2 Pengujian CPU dan Memory Usage (BLE+WiFi)

Pengujian ini dilakukan untuk mendapatkan data dari CPU dan *Memory Usage* pada saat pengiriman data. Terdapat dua skenario berbeda dalam pengujian ini, skenario pertama menggunakan transmisi BLE dan juga pada saat pengiriman data menggunakan transmisi BLE bersamaan dengan pengiriman data yang dikerjakan transmisi WiFi.

**Skenario 1: CPU dan Memory Usage Transmisi BLE**

Dalam pengujian ini, penulis mengukur CPU dan *Memory Usage* pada saat melakukan pengiriman data ke *Middleware* menggunakan transmisi BLE melalui protokol MQTT.

```

null { memory: 61464576,
memoryInfo: { rss: 61464576, vsize: 656005922816 },
cpu: 4.407724109596076,
cpuInfo:
{ pcpu: 4.407724109596076,
pcpuUser: 0.5110032611765915,
pcpuSystem: 3.8967208484194846,
cpuTime: undefined } }
2018-01-15T14:41:01.648Z
null { memory: 62521344,
memoryInfo: { rss: 62521344, vsize: 663002021888 },
cpu: 31.045969353764203,
cpuInfo:
{ pcpu: 31.045969353764203,
pcpuUser: 4.197201865423056,
pcpuSystem: 26.84876748834113,
cpuTime: 4.6600000000000004 } }
2018-01-15T14:41:16.615Z
null { memory: 52178944,
memoryInfo: { rss: 52178944, vsize: 665149505536 },
cpu: 38.149134487350224,
cpuInfo:
{ pcpu: 38.149134487350224,

```

Gambar 4. Log CPU dan Memory Transmisi BLE

Pada gambar menunjukkan hasil log dari penggunaan CPU dan *Memory usage* pada *Middleware* selama pengujian pada Skenario 1 berlangsung. Log tersebut menunjukkan *output* dari program untuk monitoring CPU dan *memory usage* yang telah dijelaskan pada bab sebelumnya. Tujuan dari pengujian adalah untuk mengukur pemakaian CPU dan Memory didalam *middleware*.

**Skenario 2: CPU dan Memory Usage Transmisi BLE + WiFi**

Dalam pengujian ini, penulis mengukur CPU dan *Memory Usage* pada saat melakukan pengiriman data ke *Middleware* menggunakan transmisi BLE melalui protokol MQTT bersamaan dengan pengiriman data yang dikerjakan oleh transmisi WiFi melalui protokol MQTT.

```

null { memory: 57315328,
memoryInfo: { rss: 57315328, vsize: 673303232512 },
cpu: 0.6727424388374114,
cpuInfo:
{ pcpu: 0.6727424388374114,
pcpuUser: 0.10226067013249555,
pcpuSystem: 0.5704817687049157,
cpuTime: undefined } }
2018-01-16T02:41:08.453Z
null { memory: 58417152,
memoryInfo: { rss: 58417152, vsize: 677598199808 },
cpu: 51.165889407055055,
cpuInfo:
{ pcpu: 51.165889407055055,
pcpuUser: 9.127248500998125,
pcpuSystem: 42.03864090605693,
cpuTime: 7.6800000000000007 } }
2018-01-16T02:41:23.393Z
null { memory: 58040320,
memoryInfo: { rss: 58040320, vsize: 676004364288 },
cpu: 53.23117921385027,
cpuInfo:
{ pcpu: 53.23117921385027,

```

Gambar 5. Log CPU dan Memory Transmisi BLE + WiFi

Pada gambar menunjukkan hasil log dari penggunaan CPU dan *Memory usage* pada *Middleware* selama pengujian pada Skenario 2

berlangsung. Log tersebut menunjukkan *output* dari program untuk monitoring CPU dan *memory usage* yang telah dijelaskan pada bab sebelumnya. Tujuan dari pengujian adalah untuk mengukur pemakaian CPU dan Memory didalam *middleware* pada pengiriman data menggunakan transmisi BLE dan Wifi secara bersamaan.

**5.3 Pengujian Bandwidth Consumption (BLE+WiFi)**

Pengujian *Bandwidth* dilakukan untuk mengetahui *Bandwidth* dari proses pengiriman data ke *Middleware* menggunakan transmisi jaringan yang berbeda yaitu BLE dan WiFi melalui protokol MQTT. Terdapat dua skenario berbeda dimana pada skenario pertama akan dilakukan pengujian terhadap bandwidth dengan menggunakan transmisi jaringan BLE melalui protokol MQTT dan yang kedua dilakukan pengujian terhadap throughput dengan pengiriman data menggunakan transmisi jaringan BLE melalui protokol MQTT bersamaan dengan pengiriman data yang dilakukan transmisi WiFi melalui protokol MQTT.

**Skenario 1: Bandwidth transmisi BLE**

Dalam pegujian ini, penulis mengukur *Bandwidth* pengiriman data menggunakan transmisi jaringan BLE melalui protokol MQTT, pengujian dilakukan dengan menjalankan aplikasi sebanyak 5 kali selama 5 menit.

Statistics		
Measurement	Captured	Displayed
Packets	6420	6420 (100.0%)
Time span, s	298.256	298.256
Average pps	21.5	21.5
Average packet size, B	45.5	45.5
Bytes	292569	292569 (100.0%)
Average bytes/s	980	980
Average bits/s	7847	7847

Gambar 6. Pengujian Bandwith BLE

Gambar 6.24 menunjukkan sample pengujian bandwidth yang dilakukan pada *Middleware* dalam 5 menit transmisi data menggunakan BLE. Gambar tersebut merupakan statistik pengujian pertama untuk mengetahui rata – rata besar penggunaan bandwidth.

**Skenario 2: Bandwidth Transmisi BLE + WiFi**

Dalam pegujian ini, penulis mengukur *Bandwidth* pengiriman data menggunakan transmisi jaringan BLE melalui protokol MQTT dan bersamaan dengan pengiriman data yang

dilakukan transmisi WiFi melalui protokol MQTT, pengujian dilakukan dengan menjalankan aplikasi sebanyak 5 kali selama 5 menit.

Measurement	Captured	Displayed	Marked
Packets	14548	14548 (100.0%)	N/A
Time span, s	327.836	327.836	N/A
Average pps	44.4	44.4	N/A
Average packet size, B	39.5	39.5	N/A
Bytes	575104	575104 (100.0%)	0
Average bytes/s	1754	1754	N/A
Average bits/s	14 k	14 k	N/A

Gambar 7. Pengujian Bandwidth BLE + WiFi

Gambar 6.24 menunjukkan sample pengujian bandwidth yang dilakukan pada *Middleware* dalam 5 menit transmisi data menggunakan BLE dan Wifi secara bersamaan ke *Middleware*. Gambar tersebut merupakan statistik pengujian pertama untuk mengetahui rata – rata besar penggunaan bandwidth.

## 6. HASIL DAN PEMBAHASAN

Hasil berfungsi untuk memberikan hasil pelaksanaan terkait teknik penelitian dan menyuguhkan data yang mendukung hasil penelitian. Pada bab ini menyuguhkan hasil perancangan dan pengukuran yang dilakukan berdasarkan skenario yang telah dibuat. Dari hasil capture data yang dilakukan pada program wireshark, diambil data dengan menjalankan proses pengiriman data selama 10 menit. Pada Skenario yang pertama ini didapatkan rata – rata delay pada pengirim data sebanyak 487 data selama 10 menit adalah 0,04523s. Pada Skenario yang kedua ini didapatkan rata – rata delay pada pengirim data sebanyak 487 data selama 10 menit adalah 0,032407248s. Untuk pengujian CPU dan *Memory Usage* didapatkan rata – rata pemakaian CPU sebesar 31% dan memori 5% untuk skenario satu sedangkan untuk skenario dua rata – rata pemakaian CPU sebesar 46% dan memori 7%.

## 7. KESIMPULAN

Berdasarkan Hasil Perancangan, Implementasi, dan pengujian yang telah dilakukan didapatkan kesimpulan sebagai berikut:

1. Dari hasil pengujian fungsional, semua fungsi yang diuji berhasil berjalan sesuai tujuan sistem. Dan juga hasil pengujian delay terhadap jarak dapat disimpulkan bahwa kinerja dari Node dan *Gateway* menggunakan Transmisi *Bluetooth Low*

*Energy* menunjukkan hasil delay yang tidak terlalu besar yaitu 0.777434 detik untuk pengiriman data menggunakan Jaringan BLE dengan posisi Node berjarak satu meter dari *Middleware*, dan 2.10614 detik pada saat node berjarak 5 meter, dapat diambil kesimpulan bahwa kinerja dari transmisi BLE masih dapat dipengaruhi oleh jarak.

2. Pengujian kinerja interoperabilitas apabila Transmisi BLE berjalan bersama dengan transmisi WiFi juga memberikan hasil bahwa Pengiriman data menggunakan transmisi BLE dapat dilakukan secara bersamaan dengan Transmisi WiFi, akan tetapi *Bandwidth Consumption* dari pengiriman data menggunakan transmisi BLE yang rata – rata sebesar 7786,6bits/detik akan menjadi dua kali lipat apa bila berjalan bersama dengan pengiriman data menggunakan transmisi WiFi, serta penggunaan CPU dan memori memiliki perbedaan antara 12% hingga 17% tergantung data pengiriman.

## 8. SARAN

1. Untuk mendukung network interoperability tidak hanya dapat dilakukan dengan menggunakan Bluetooth, Wifi, ataupun Ethernet. Masih banyak jaringan transmisi lain yang dapat digunakan dan mungkin akan memberikan solusi yang lebih baik.
2. Penelitian selanjutnya dapat mengimplementasikan banyak sensor lainnya, sehingga meripakan data yang lebih actual.

## DAFTAR PUSTAKA

- Anwari, H. (2017). Pengembangan IoT middleware berbasis eventbased dengan protokol komunikasi COAP, MQTT. J-PTIHK
- Boulouache-Nouali, A.O. (2015) “A BLE-Based data collection system for IoT”. Algiers, Algeria.
- Durfee.W, “Arduino Microcontroller Guide”
- Gupta.N, (2016) “Inside Bluetooth Low Energy” Artech House 2<sup>nd</sup> Edition
- Inigo, P. (2008). “ Bluetooth “. Furtwangen im Schwarzwald, Germany. Hochschule furtwangen university. 1, 6 – 12.

- Mackensen-Mathias, E.L. (2013). “ Bluetooth Low Energy Based Wireless sensors” Offenburg, Germany. University of Applied Sciences Offenburg badstrabe. 1, 1 – 3.
- Pramukantoro E.S, W. Yahya, G. Arganata, A. Bhawiyuga and A. Basuki, "Topic based IoT data storage framework for heterogeneous sensor data," *2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA)*, Lombok, Indonesia, 2017, pp. 1-4.  
doi: 10.1109/TSSA.2017.8272895