

BAB 4 ANALISIS KEBUTUHAN DAN PERANCANGAN

Pada bab ini akan dibahas kebutuhan dari sistem diantaranya kebutuhan perangkat keras dan perangkat lunak, kebutuhan sistem akan dijelaskan melalui sebuah table ataupun gambaran umum. Selanjutnya akan dijelaskan proses perancangan yang meliputi perancangan alur dari sistem, perancangan middleware, perancangan interface, perancangan paket sensor serta perancangan untuk proses pengujian.

4.1 Deskripsi Umum Sistem

Tujuan utama dari dikembangkannya sistem ini adalah agar middleware dapat memberikan gateway bagi sensor untuk mengirimkan data melalui Jaringan Non-IP dan dirubah ke jaringan IP. Selanjutnya middleware akan menyediakan gateway untuk aplikasi agar dapat mengirimkan datanya ke Redis. Sistem ini sangat berguna nantinya agar pengguna tidak melakukan penyetingan ulang secara terus menerus apabila ingin menggunakan sistem peneriman ataupun protokol yang berbeda.

Sistem ini akan menggunakan proses pengiriman menggunakan Bluetooth dan menggunakan protokol MQTT, middleware bekerja sebagai broker serta redis yang akan menjadi media penyimpan sementara dan menampilkan data secara realtime.

4.2 Analisis Kebutuhan

Analisis kebutuhan dibuat bertujuan agar dapat mengetahui kebutuhan apa saja yang diperlukan dalam proses membangun sistem dan mengimplementasikan sistem. Kebutuhan sistem ini terbagi menjadi 2 yaitu kebutuhan Fungsional dan Kebutuhan Sistem.

4.2.1 Kebutuhan Fungsional

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

Tabel 4. 1 Kebutuhan Fungsional

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

4.2.2 Kebutuhan Sistem

Kebutuhan sistem ini terbagi menjadi 2 yaitu kebutuhan perangkat keras dan perangkat lunak.

4.2.2.1 Kebutuhan Perangkat Keras (Hardware)

Kebutuhan perangkat keras yang digunakan dalam sistem adalah sebagai berikut:

Tabel 4. 2 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.
NodeMCU (ESP32)	ESP32 adalah merupakan microcontroller Bluetooth 4.0 serta WiFi yang akan dipasangkan dengan Arduino untuk dapat mengirimkan data
ASUS X45U AMD E3	Perangkat ini digunakan untuk memonitor data dari redis

4.2.2.2 Kebutuhan Perangkat Lunak (Software)

Kebutuhan perangkat lunak yang dibutuhkan untuk mengembangkan sistem adalah sebagai berikut:

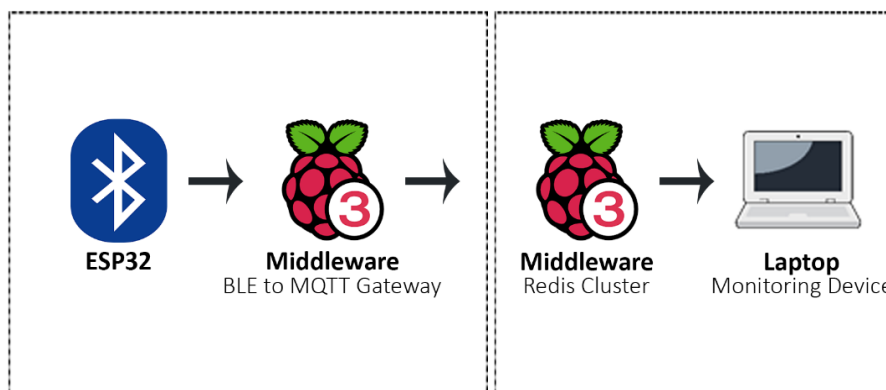
Tabel 4. 3 Kebutuhan Perangkat Lunak

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
Espruino Hub	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

4.3 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.

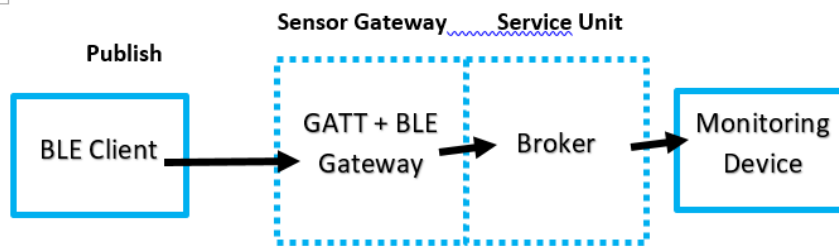


Gambar 4. 1 Alur Komunikasi BLE

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.

4.4 Perancangan Middleware

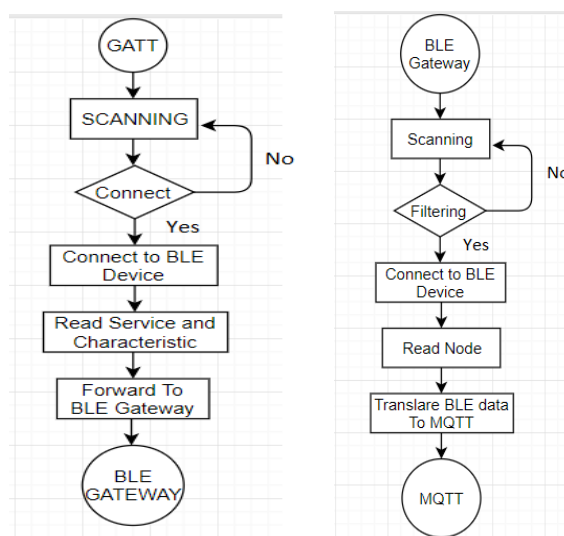
Pada proses perancangan transmisi Data BLE to MQTT terdapat *Middleware* dalam penelitian ini yang dikembangkan serta memiliki arsitektur seperti gambar 4.2 berikut:



Gambar 4. 2 Perancangan Middleware

Seperti pada gambar 4.2, middleware akan memiliki 2 komponen sensor gateway, service unit. Sensor gateway menyediakan interface untuk Middleware sehingga dapat membaca data yang ditransmisikan oleh BLE client dan juga berfungsi sebagai *Bridge* untuk mentranslasikan data yang ditranmisikan menggunakan jaringan BLE ke dalam protokol MQTT sehingga yang awalnya berupa Non-Ip menjadi IP. Kemudian komponen service unit disini berfungsi sebagai database sementara, *data store*, dan *message broker*, sehingga apabila nanti ingin mengimplementasikan data center atau websocket dapat melakukan subscribe melalui broker tersebut.

Perancangan BLE gateway berfungsi untuk menyediakan interface agar transmisi BLE dari sensor dapat diterima oleh middleware dan ditranslasikan menjadi transmisi MQTT. Gateway yang akan ditanamkan pada Middleware ini adalah EspruinoHub. Pertama sensor akan mengirimkan data menuju middleware dengan cara berinteraksi dengan GATT *middleware*, yang kemudian diteruskan ke EspruinoHub untuk ditranslasikan menjadi MQTT dikarenakan jaringan BLE merupakan jaringan *non-ip* sehingga tidak dapat langsung mengirimkan data menggunakan protokol MQTT yang berupa jaringan yang memiliki IP. Diagram alir perancangan GATT dan BLE Gateway dapat dilihat pada gambar 4.3:



Gambar 4. 3 Diagram alir program GATT dan BLE Gateway

4.4.1.1 Perancangan Generic Attribute Profile (GATT)

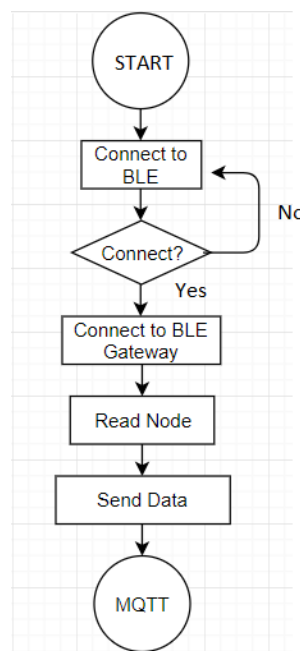
Perancangan GATT dilakukan agar Middleware dapat terkoneksi dengan sensor melalui transmisi BLE, GATT dibutuhkan karena jaringan BLE tidak dapat langsung terkoneksi dengan perangkat lain dengan mudah, GATT dibutuhkan untuk mendefinisikan Servis dan karakteristik dari perangkat sensor yang akan terkoneksi. Koneksi yang menggunakan GATT adalah koneksi eksklusif dimana hanya satu komunikasi BLE antara sensor dengan middleware yang dapat terjadi dalam satu waktu.

4.4.1.2 Perancangan Filtering BLE Device

Perancangan Filtering MAC BLE Device berfungsi untuk memfilter perangkat pada saat *scanning* yang dilakukan oleh BLE gateway. Proses filtering dilakukan karena pada saat scanning, BLE gateway akan menangkap semua Bluetooth device yang sedang menyala, semua perangkat BLE yang aktif datanya akan di ambil oleh BLE Gateway.

4.5 Perancangan Sensor Node

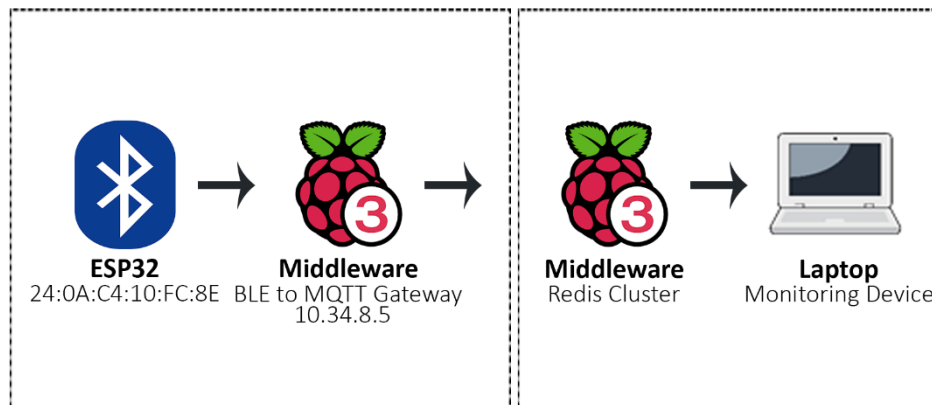
Sensor node akan dibangun menggunakan perangkat keras berupa mikrokontroler NODEMCU (ESP32) yang merupakan transmitter jaringan BLE. Pada mikrokontroler NODEMCU (ESP32) tersebut akan ditanamkan code berbasis C sehingga dapat melakukan proses pengiriman data sederhana melalui jaringan BLE. Langkah – langkah yang harus dilakukan agar mikrokontroler ini dapat membaca kode C adalah, harus menginstall module firmware ESP32 dan diletakan pada Arduino IDE sehingga code dapat ditanamkan pada ESP32.



Gambar 4. 4 Diagram Alir Program pada NodeMCU

4.6 Perancangan Topologi Jaringan

Supaya sistem dapat beroperasi sesuai keinginan, perlu dirancangan sebuah topologi jaringan antar tiap perangkat didalam sistem. Topologi jaringan ini akan memperlihatkan bagaimana sensor dapat terhubung dan mentransmisikan datanya melalui jaringan BLE ke middleware dan akan ditranslasi menjadi transmisi MQTT.



Gambar 4. 5 Topologi Jaringan

Pada gambar 4.6 tersebut sensor Node yang memiliki MAC Address 24:0A:C4:10:FC:8E akan mentransmisikan datanya melalui jaringan BLE yang akan ditangkap oleh GATT middleware kemudian diteruskan menuju BLE gateway yang akan mentranslasikan transmisi BLE tersebut menjadi transmisi MQTT yang memiliki IP 10.34.8.5 dan akan dikirimkan ke redis, dan yang terakhir untuk memonitor data yang masuk ke redis dapat dilihat menggunakan laptop.

4.7 Perancangan Pengujian

Perancangan pengujian dilakukan agar dapat menentukan batasan – batasan yang harus didapatkan serta skenario pengujian pada middleware yang dikembangkan.

4.7.1 Pengujian Fungsional

Pengujian fungsional adalah pengujian yang dirancang untuk mengetahui apakah system sudah berjalan sesuai kebutuhan fungsional yang telah ditentukan sebelumnya. Skenario pengujian fungsional dapat dilihat pada tabel 4.4:

Tabel 4. 4 Skenario Pengujian Fungsional

KODE	Fungsi	Skenario
EP_001	Sensor dapat Terkoneksi dengan GATT middleware	<ol style="list-style-type: none"> 1. Middleware telah berjalan 2. Sensor telah diaktifkan 3. Pengguna menjalankan GATT middleware

EP_002	GATT middleware dapat mendeteksi koneksi dari BLE sensor	<ol style="list-style-type: none"> 1. Middleware telah berjalan 2. Sensor telah diaktifkan 3. Pengguna menjalankan GATT middleware
EP_003	BLE gateway dapat terkoneksi dengan Sensor Node melalui jaringan BLE	<ol style="list-style-type: none"> 1. Middleware telah berjalan 2. Pengguna menjalankan BLE
EP_004	BLE gateway dapat menerima data yang di publish oleh sensor node melalui jaringan BLE	<ol style="list-style-type: none"> 1. Middleware telah berjalan 2. Pengguna menjalankan BLE gateway pada middleware 3. Pengguna memonitor log pada middleware
EP_005	BLE gateway dapat mentranslasikan data dari sensor node kedalam protokol MQTT	<ol style="list-style-type: none"> 1. Middleware telah berjalan 2. Pengguna menjalankan BLE Gateway pada middleware 3. Pengguna memonitor log pada middleware
EP_006	BLE gateway dapat mempublish data kedalam Redis Middleware melalui protocol MQTT	<ol style="list-style-type: none"> 1. Middleware telah berjalan 2. Pengguna menjalankan BLE Gateway 3. Pengguna memonitor log pada middleware 4. Pengguna memonitor data pada redis
EP_007	Redis Middleware dapat subscribe data yang dikirimkan BLE gateway melalui protocol MQTT	<ol style="list-style-type: none"> 1. Middleware telah berjalan 2. Pengguna menjalankan BLE gateway 3. Pengguna memonitor log pada middleware 4. Pengguna memonitor data pada redis

4.7.2 Pengujian Perubahan Jarak

Pengujian perubahan jarak dilakukan untuk dapat mengetahui performansi dari *Bluetooth Low Energy (BLE)*, parameter yang akan diukur pada pengujian ini adalah *Delay* pada proses pengirimannya. Langkah – langkah skenario pada pengujian perubahan jarak dapat dilihat pada tabel 4.5:

Tabel 4. 5 Skenario Pengujian Perubahan Jarak

Kode	Fungsi	Prosedur
------	--------	----------

Skenario_1	Hitung Delay Pengiriman Data dengan Jarak 1 Meter	<ol style="list-style-type: none"> 1. Middleware Telah berjalan 2. Pengguna Mengaktifkan NodeMCU dengan jarak 1 meter dari Middleware 3. Pengguna Menjalankan BLE Gateway 4. Pengguna Mengcapture data yang masuk menggunakan Wireshark dengan jumlah 20 packet data 5. Pengguna menghitung rata – rata delay pengiriman
Skenario_2	Hitung Delay Pengiriman Data dengan Jarak 2 Meter	<ol style="list-style-type: none"> 1. Middleware Telah berjalan 2. Pengguna Mengaktifkan NodeMCU dengan jarak 2 meter dari Middleware 3. Pengguna Menjalankan BLE Gateway 4. Pengguna Mengcapture data yang masuk menggunakan Wireshark dengan jumlah 20 packet data 5. Pengguna menghitung rata – rata delay pengiriman
Skenario_3	Hitung Delay Pengiriman Data dengan Jarak 3 Meter	<ol style="list-style-type: none"> 1. Middleware Telah berjalan 2. Pengguna Mengaktifkan NodeMCU dengan jarak 3 meter dari Middleware 3. Pengguna Menjalankan BLE Gateway 4. Pengguna Mengcapture data yang masuk menggunakan Wireshark dengan jumlah 20 packet data 5. Pengguna menghitung rata – rata delay pengiriman
Skenario_4	Hitung Delay Pengiriman Data dengan Jarak 4 Meter	<ol style="list-style-type: none"> 1. Middleware Telah berjalan 2. Pengguna Mengaktifkan NodeMCU dengan jarak 4 meter dari Middleware 3. Pengguna Menjalankan BLE Gateway 4. Pengguna Mengcapture data yang masuk menggunakan Wireshark dengan jumlah 20 packet data 5. Pengguna menghitung rata – rata delay pengiriman

Skenario_5	Hitung Delay Pengiriman Data dengan Jarak 5 Meter	<ol style="list-style-type: none"> 1. Middleware Telah berjalan 2. Pengguna Mengaktifkan NodeMCU dengan jarak 5 meter dari Middleware 3. Pengguna Menjalankan BLE Gateway 4. Pengguna Mengcapture data yang masuk menggunakan Wireshark dengan jumlah 20 packet data 5. Pengguna menghitung rata – rata delay pengiriman
------------	---	---

4.7.3 Pengujian Interoperability

Pengujian ini dilakukan untuk menguji performansi dari *Bluetooth Low Energy* (BLE) saat proses mengirimkan data menggunakan transmisi BLE, dan juga saat mengirimkan data bersamaan dengan Transmisi WiFi.

4.7.3.1 Pengujian CPU dan Memory Usage (WiFi + BLE)

Pada pengujian CPU dan *Memory usage* akan menggunakan dua jalur transmisi yang berbeda yaitu BLE yang dikembangkan pada penelitian ini dan WiFi yang telah dikembangkan oleh peneliti sebelumnya, akan terdapat tiga skenario pengujian delay seperti pada tabel berikut:

Tabel 4. 6 Skenario Pengujian CPU dan Memori

Kode	Fungsi	Prosedur
Skenario_1	Hitung CPU dan Memory usage di <i>Middleware</i> pada saat pengiriman data RSSI menggunakan transmisi BLE ke <i>Middleware</i> selama 1 jam 30 menit.	<ol style="list-style-type: none"> 1. <i>Middleware</i> sudah berjalan 2. Pengguna menjalankan NodeMCU (ESP32) 3. Pengguna menjalankan BLE gateway 4. Pengguna mengcapture data CPU dan Memory usage menggunakan software yang telah dibuat. 5. Pengguna menghitung rata – rata data CPU dan <i>Memory usage</i>.
Skenario_2	Hitung CPU dan <i>Memory Usage</i> pada <i>Middleware</i> saat mengirimkan data RSSI milik transmisi BLE bersamaan dengan pengiriman data yang dilakukan transmisi WiFi selama 1 jam 30 menit.	<ol style="list-style-type: none"> 1. <i>Middleware</i> sudah berjalan 2. Pengguna menjalankan NodeMCU (ESP32) 3. Pengguna menjalankan NodeMCU (ESP8266) 4. Pengguna menjalankan BLE gateway 5. Pengguna mengcapture data CPU dan <i>Memory usage</i> menggunakan software yang telah dibuat 6. Pengguna menghitung rata – rata data CPU dan <i>Memory usage</i>.

4.7.3.2 Pengujian Delay (WiFi + BLE)

Pada pengujian delay akan menggunakan dua jalur transmisi berbeda yaitu WiFi yang telah dikembangkan oleh peneliti sebelumnya dan BLE yang dikembangkan pada penelitian ini, akan terdapat dua skenario pengujian delay seperti pada tabel berikut:

Tabel 4. 7 Skenario Pengujian Delay

Kode	Fungsi	Prosedur
Skenario_1	Hitung Delay pengiriman data RSSI ke <i>Middleware</i> melalui transmisi BLE selama 10 menit.	<ol style="list-style-type: none"> 1. Middleware sudah berjalan 2. Pengguna menjalankan NodeMCU (ESP32) 3. Pengguna menjalankan BLE gateway 4. Pengguna mengcapture data dari wireshark 5. Pengguna menghitung rata – rata delay
Skenario_2	Hitung Delay pengiriman data RSSI ke <i>Middleware</i> melalui transmisi BLE selama 10 menit bersamaan dengan pengiriman data yang dilakukan transmisi WiFi.	<ol style="list-style-type: none"> 1. Middleware sudah berjalan 2. Pengguna menjalankan NodeMCU (ESP32) 3. Pengguna menjalankan NodeMCU (ESP8266) 4. Pengguna mengcapture data dari wireshark 5. Pengguna menghitung rata – rata delay

4.7.3.3 Pengujian Bandwidth (WiFi + BLE)

Pada pengujian bandwidth akan terdapat 2 jalur komunikasi berbeda yaitu WiFi yang telah dikembangkan pada penelitian sebelumnya dan BLE yang dikembangkan pada penelitian ini, pada pengujian terdapat dua skenario pengujian bandwidth seperti pada table berikut:

Tabel 4. 8 Skenario Pengujian Bandwidth

Kode	Fungsi	Prosedur
Skenario_1	Hitung <i>bandwith</i> pengiriman data RSSI ke <i>Middleware</i> melalui transmisi BLE sebanyak 5 kali menjalankan program pengiriman dengan waktu pengiriman selama 5 menit	<ol style="list-style-type: none"> 1. Middleware sudah berjalan 2. Pengguna menjalankan NodeMCU (ESP32) 3. Pengguna menjalankan BLE gateway 4. Pengguna mengcapture data dari wireshark 5. Pengguna menghitung rata – rata <i>throughput</i>

Skenario_2	<p>Hitung <i>bandwidth</i> pengiriman data RSSI ke <i>Middleware</i> melalui transmisi BLE dan suhu melalui transmisi WiFi sebanyak 5 kali menjalankan program pengiriman dengan waktu pengiriman selama 5 menit.</p>	<ol style="list-style-type: none"> 1. Middleware sudah berjalan 2. Pengguna menjalankan NodeMCU (ESP32) 3. Pengguna menjalankan NodeMCU (ESP8266) 4. Pengguna mengcapture data dari wireshark 5. Pengguna menghitung rata – rata <i>Throughput</i>
------------	---	---