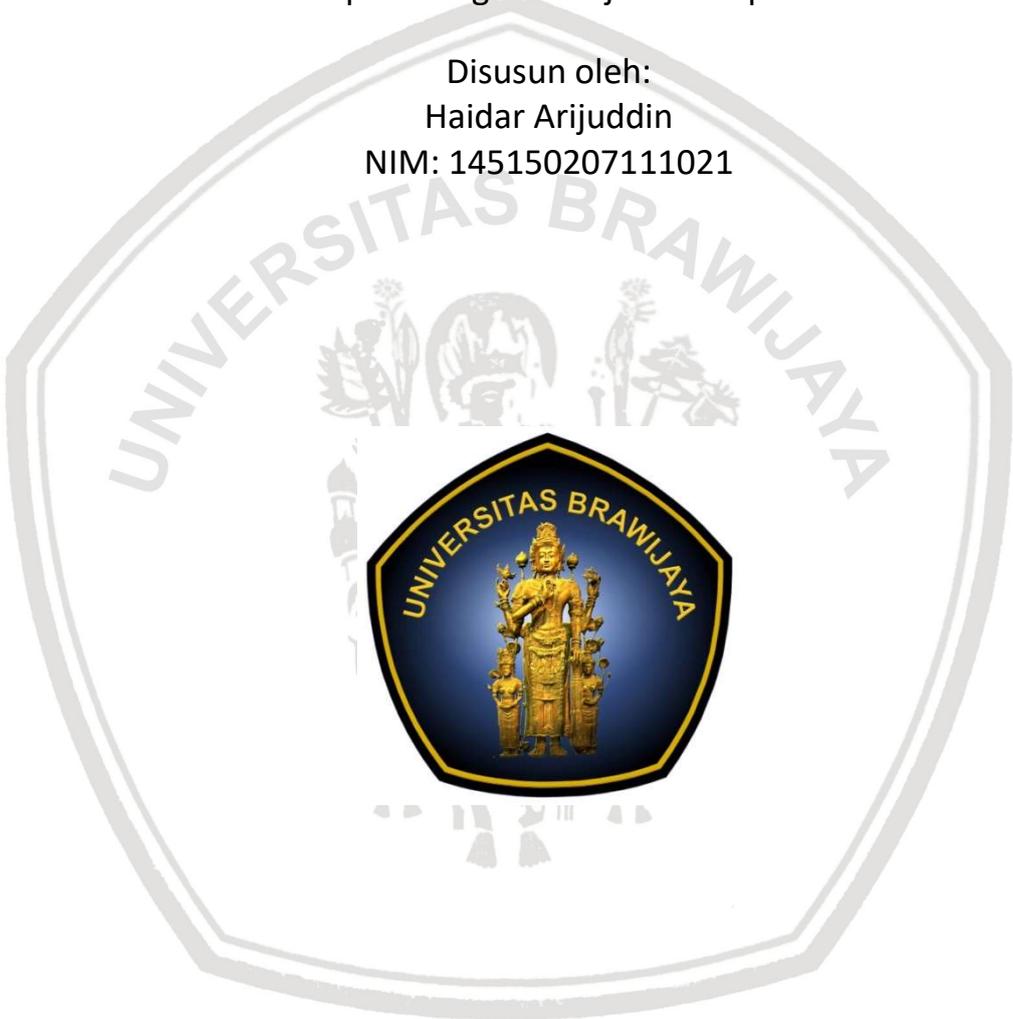


**PENGEMBANGAN SISTEM PERANTARA PENGIRIMAN DATA  
MENGUNAKAN MODUL KOMUNIKASI LORA dan  
PROTOKOL MQTT PADA *WIRELESS SENSOR NETWORK***

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Haidar Arijuddin  
NIM: 145150207111021



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018





## PENGESAHAN

PENGEMBANGAN SISTEM PERANTARA PENGIRIMAN DATA MENGGUNAKAN  
MODUL KOMUNIKASI LORA dan PROTOKOL MQTT PADA *WIRELESS SENSOR  
NETWORK*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh :  
Haidar Arijuddin  
NIM: 145150207111021

Skripsi ini telah diuji dan dinyatakan lulus pada  
17 Desember 2018  
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Adhitya Bhawiyuga, S.Kom., M.Sc.  
NIP: 19890720 201803 1 002

Dosen Pembimbing II

Kasyful Amron, S.T., M.Sc.  
NIP: 19750803 200312 1 003

Mengetahui  
Ketua Jurusan Teknik Informatika



Tri Astoto Kurnjawan, 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 di dalam 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, 19 Desember 2018



Haidar Arijuddin

NIM: 145150207111021

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa, karena atas berkat, rahmat dan hidayah-Nya penulis dapat menyelesaikan penulisan skripsi berjudul **“Pengembangan Sistem Perantara Pengiriman Data Sensor Menggunakan Modul Komunikasi LoRa Dan protokol MQTT Pada *Wireless Sensor Network*”** ini dapat terselesaikan.

Dalam penulisan skripsi ini, peneliti menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari berbagai pihak, oleh karena itu penulis ingin menyampaikan rasa hormat beserta terima kasih kepada:

1. Bapak Adhitya Bhawiyuga, S.Kom., M.S dan Bapak Kasyful Amron, S.T., M.Sc selaku dosen pembimbing skripsi yang telah meluangkan waktu pikiran dan tenaga sehingga skripsi ini dapat terselesaikan.
2. Kedua Orang Tua yang telah mendidik dan membesarkan peneliti serta yang tiada hentinya memberi dukungan, do’a, dan semangat hingga terselesainya skripsi ini.
3. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D selaku Ketua Jurusan Teknik Informatika.
4. Warga kontrkan Kalpataru yang senantiasa membantu pengujian sistem dan memberikan saran hingga penelitian ini dapat terselesaikan.
5. Kepada Johanna Rindawati Sani Putranti yang telah memberi semangat dan dukungan sehingga dapat terselesaikan penelitian ini.
6. Seluruh dosen dan civitas Program Studi Teknik Informatika Universitas Brawijaya atas kesediaan membagi ilmunya kepada penulis.

Penulis juga menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun senantiasa penulis harapkan demi perbaikan skripsi in. Semoga skripsi ini dapat memberikan manfaat bagi penulis, pihak yang terlibat dan semua pihak membacanya.

Malang, 19 Desember 2018

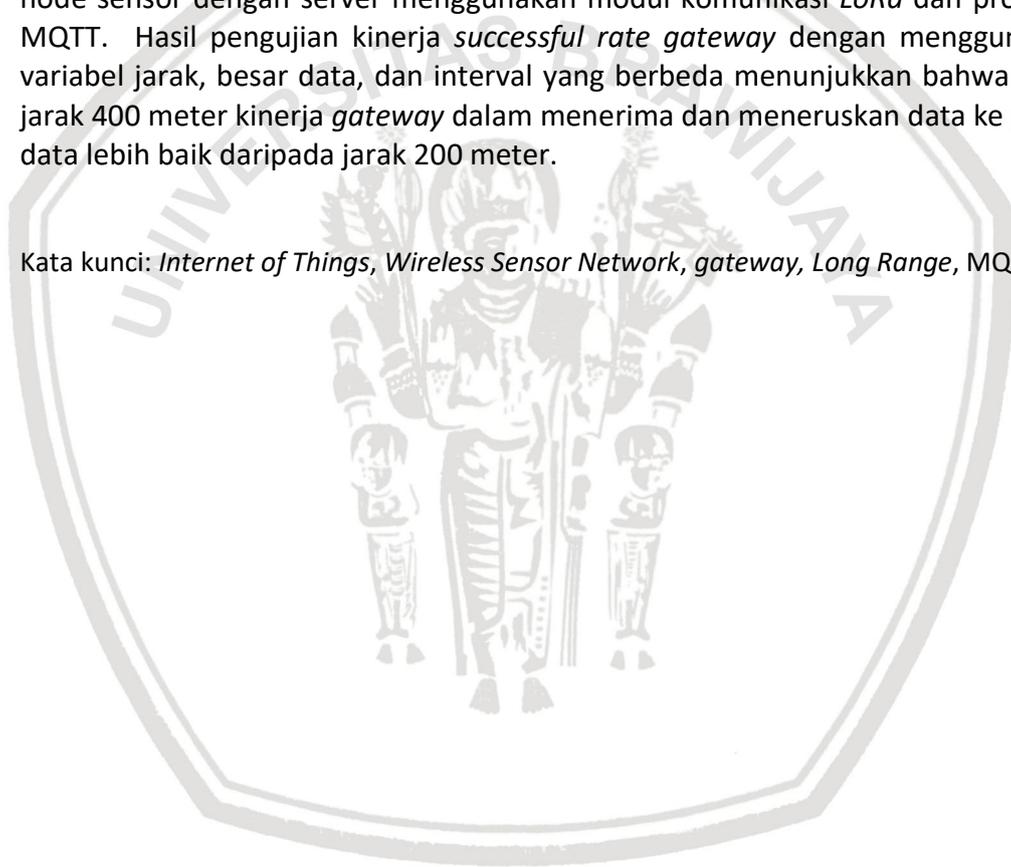
Haidar Arijuddin

[ari\\_haidar@yahoo.com](mailto:ari_haidar@yahoo.com)

## ABSTRAK

Pengembangan *Internet of Things* selalu didukung dengan konsep WSN (*Wireless Sensor Network*). Sebagian besar konsep WSN menggunakan konsumsi daya baterai, sehingga terdapat kebutuhan konstan untuk mengurangi kebutuhan energi. Teknologi LoRa (*Long Range*) memiliki penggunaan konsumsi daya rendah dan memiliki jangkauan komunikasi luas lebih dari 2 km, Namun tidak dapat melakukan pengiriman data langsung ke server. Sehingga diperlukan sistem pengiriman data untuk menghubungkan antar perangkat di node sensor dengan server yang disebut gateway. Komunikasi antara gateway dan pusat data menggunakan protokol MQTT karena mendukung karakteristik IoT seperti penggunaan *bandwidth* yang kecil. Pada penelitian ini akan dilakukan pengembangan sistem gateway agar dapat menghubungkan komunikasi antara node sensor dengan server menggunakan modul komunikasi LoRa dan protokol MQTT. Hasil pengujian kinerja *successful rate gateway* dengan menggunakan variabel jarak, besar data, dan interval yang berbeda menunjukkan bahwa pada jarak 400 meter kinerja gateway dalam menerima dan meneruskan data ke pusat data lebih baik daripada jarak 200 meter.

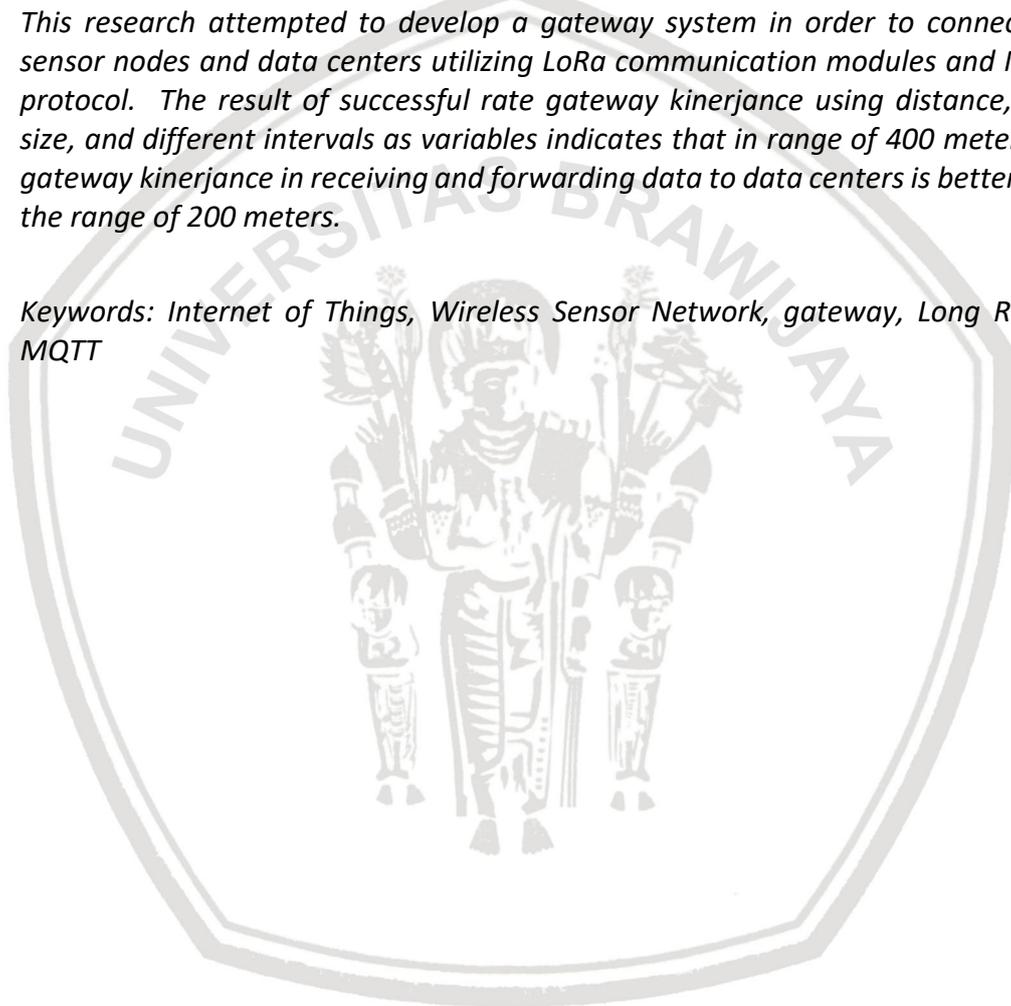
Kata kunci: *Internet of Things, Wireless Sensor Network, gateway, Long Range, MQTT*



## ABSTRACT

*The development of Internet of Things is always supported by the WSN (Wireless Sensor Network) concept but the battery power consumption of WSN raises a constant need to lower the energy requirement. LoRa (Long Range) technology acquires low power consumption and broad communication range of more than 2 km, however cannot send data directly to the server. So that a data delivery system is needed to connect between devices at the sensor node to a server called the gateway. The communication between gateway and data centers employs MQTT protocol since it supports the characteristics of IoT such as small bandwidth usage. This research attempted to develop a gateway system in order to connect the sensor nodes and data centers utilizing LoRa communication modules and MQTT protocol. The result of successful rate gateway kinerjance using distance, data size, and different intervals as variables indicates that in range of 400 meters the gateway kinerjance in receiving and forwarding data to data centers is better than the range of 200 meters.*

**Keywords:** *Internet of Things, Wireless Sensor Network, gateway, Long Range, MQTT*

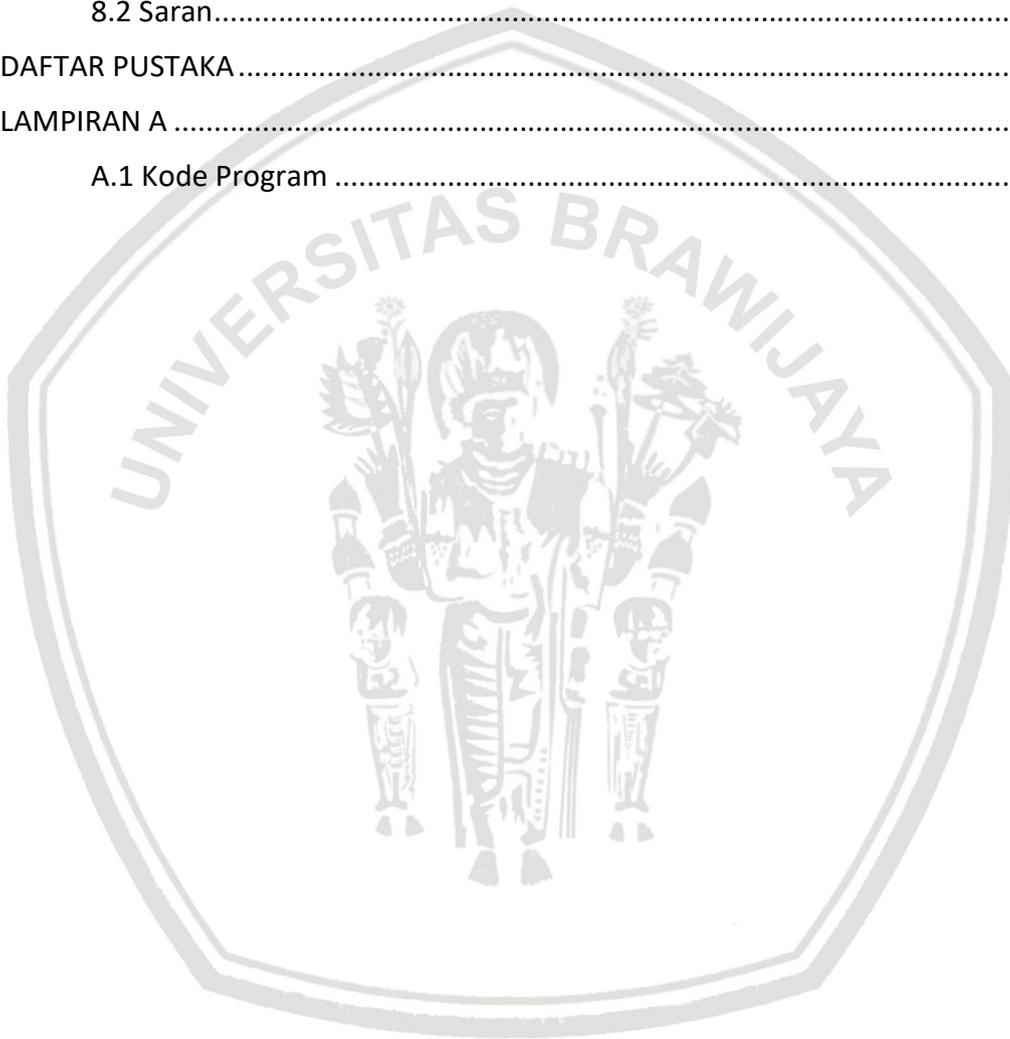


## DAFTAR ISI

PENGESAHAN .....	i
PERNYATAAN ORISINALITAS.....	ii
KATA PENGANTAR.....	iii
ABSTRAK .....	iv
ABSTRACT .....	v
DAFTAR ISI .....	vi
DAFTAR TABEL.....	ix
DAFTAR GAMBAR .....	x
<b>BAB 1 PENDAHULUAN .....</b>	<b>11</b>
1.1 Latar Belakang.....	11
1.2 Rumusan Masalah.....	12
1.3 Tujuan.....	12
1.4 Manfaat.....	12
1.5 Batasan Masalah.....	13
1.6 Sistematika Pembahasan .....	13
<b>BAB 2 LANDASAN KEPUSTAKAAN.....</b>	<b>15</b>
2.1 Tinjauan Pustaka .....	15
2.2 Kajian Teori.....	16
2.2.1 Wireless Sensor Network (WSN) .....	16
2.2.2 Internet of Things .....	17
2.2.3 Gateway.....	17
2.2.4 Raspberry Pi.....	17
2.2.5 LoRa ( <i>Long Range</i> ) .....	20
2.2.6 <i>Message Queuing Telemetry Transport</i> (MQTT).....	21
2.2.7 <i>Publish-Subscribe (Pub-Sub)</i> .....	21
2.2.8 <i>Python</i> .....	22
2.2.9 Parameter pengujian.....	22
<b>BAB 3 METODOLOGI.....</b>	<b>23</b>
3.1 Metodologi Penelitian.....	23
3.2 Studi Literatur.....	24

3.3	Rekayasa Kebutuhan .....	25
3.3.1	Analisis kebutuhan .....	25
3.3.2	Analisis Kebutuhan Sistem .....	25
3.3.3	Analisis Kebutuhan Perangkat Keras .....	26
3.3.4	Analisis Kebutuhan Perangkat Lunak .....	26
3.4	Perancangan Sistem .....	27
3.4.1	Perancangan alur kerja sistem .....	28
3.5	Implementasi Sistem .....	30
3.6	Pengujian Sistem .....	31
3.7	Penarikan Kesimpulan dan Saran .....	31
<b>BAB 4</b>	<b>REKAYASA KEBUTUHAN .....</b>	<b>32</b>
4.1	Deskripsi Umum .....	32
4.1.1	Sistematika .....	32
4.1.2	Perspektif Sistem .....	32
4.1.3	Istilah .....	33
4.1.4	Kegunaan Sistem .....	33
4.1.5	Batasan Perancangan dan Implementasi .....	34
4.2	Kebutuhan Fungsional .....	34
4.3	Kebutuhan Non-Fungsional .....	35
<b>BAB 5</b>	<b>PERANCANGAN SISTEM .....</b>	<b>36</b>
5.1	Perancangan .....	36
5.1.1	Gambaran Umum Sistem .....	36
5.1.2	Perancangan Alur Kerja Sistem .....	37
5.1.3	Perancangan Node sensor .....	37
5.1.4	Perancangan <i>Gateway</i> .....	41
5.1.5	Perancangan Pengujian .....	44
<b>BAB 6</b>	<b>IMPLEMENTASI SISTEM .....</b>	<b>46</b>
6.1	Implementasi Node Sensor .....	46
6.2	Implementasi <i>Gateway</i> .....	47
<b>BAB 7</b>	<b>PENGUJIAN SISTEM .....</b>	<b>51</b>
7.1	Pengujian Kebutuhan Fungsional .....	51
7.1.1	Pengujian Kode PL-01 .....	51

7.1.2 Pengujian Kode PL-02 .....	52
7.1.3 Pengujian Kode PL-03 .....	53
7.1.4 Pengujian Kode PL-04 .....	53
7.1.5 Pengujian Kode PL-05 .....	54
7.2 Pengujian Kinerja .....	55
BAB 8 PENUTUP .....	60
8.1 Kesimpulan .....	60
8.2 Saran.....	61
DAFTAR PUSTAKA .....	62
LAMPIRAN A .....	63
A.1 Kode Program .....	63



## DAFTAR TABEL

Tabel 2.1 Penelitian Terkait.....	15
Tabel 2.2 Komponen GPIO pada Raspberry Pi .....	18
Tabel 2.3 Spesifikasi pada Raspberry Pi 2 Model B.....	18
Tabel 2.4 Spesifikasi pada Raspberry Pi 3 Model B.....	19
Tabel 3.2 Kebutuhan Perangkat Keras .....	26
Tabel 3.3 Kebutuhan Perangkat Lunak.....	26
Tabel 4.1 Tabel Istilah.....	33
Tabel 4.2 Kebutuhan Fungsional Sistem .....	34
Tabel 5.1 <i>Pinout</i> Modul LoRa HopeRF-RFM9x dengan Raspberry Pi .....	38
Tabel 5.2 <i>Pinout Moisture Sensor</i> dengan Raspberry Pi.....	39
Tabel 5.4 <i>Pinout</i> Modul LoRa HopeRF-RFM9x dengan Raspberry Pi .....	41
Tabel 5.5 Perancangan Skenario Pengujian Fungsional .....	44
Tabel 6.1 <i>Library</i> Pada Node Sensor.....	46
Tabel 6.2 <i>Pseudocode</i> Node Sensor .....	47
Tabel 6.3 <i>Library</i> Pada Node Sensor.....	48
Tabel 6.4 <i>Pseudocode Gateway</i> .....	48
Tabel 6.5 <i>Library</i> Pada Server.....	49
Tabel 6.6 <i>Pseudocode</i> Server.....	50
Tabel 7.1 Pengujian Kode PL-01 .....	51
Tabel 7.2 Pengujian Kode PL-02 .....	52
Tabel 7.3 Pengujian Kode PL-03 .....	53
Tabel 6.4 Pengujian Kode PL-04 .....	54
Tabel 7.5 Pengujian Kode PL-05 .....	54
Tabel 7.11 Pengujian Kinerja Sistem .....	55

## DAFTAR GAMBAR

Gambar 2.1 Arsitektur <i>Wireless Sensor Network</i> .....	16
Gambar 2.2 Raspberry Pi 2 Model B .....	19
Gambar 2.3 Raspberry Pi 3 Model B .....	20
Gambar 2.4 Perbandingan Protokol Komunikasi <i>Wireless</i> .....	20
Gambar 3.1 Alur Penelitian .....	23
Gambar 3.2 Rancangan Arsitektur Sistem.....	27
Gambar 3.3 <i>Flowchart</i> Alur Kerja Sistem Secara Umum.....	29
Gambar 5.1 Gambaran Umum Sistem .....	36
Gambar 5.2 Alur Kerja Sistem Secara Umum.....	37
Gambar 5.3 Arsitektur Node Sensor .....	38
Gambar 5.4 Alur Kerja Sistem Node Sensor .....	40
Gambar 5.5 Arsitektur <i>Gateway</i> .....	41
Gambar 5.5 Alur Kerja Sistem <i>Gateway</i> .....	43
Gambar 6.1 Implementasi Perangkat Keras Node Sensor .....	46
Gambar 6.2 Implementasi Perangkat Keras <i>Gateway</i> .....	48
Gambar 7.1 Hasil Dari Pengujian PL-01.....	52
Gambar 7.2 Hasil Dari Pengujian PL-02.....	52
Gambar 7.3 Hasil Dari Pengujian PL-03.....	53
Gambar 7.4 Hasil Dari Pengujian PL-04.....	54
Gambar 7.5 Hasil Dari Pengujian PL-05.....	55
Gambar 7.11 Konfigurasi SSH pada MobaXterm .....	56
Gambar 7.12 Grafik Hasil Pengujian <i>Successful Rate</i> Dengan Interval Waktu 0.5 Detik .....	57
Gambar 7.13 Grafik Hasil Pengujian <i>Successfull Rate</i> Dengan Interval Waktu 1 Detik .....	58
Gambar 7.14 Grafik Hasil Pengujian <i>Successfull Rate</i> Dengan Interval Waktu 5 Detik .....	59

## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

Saat ini banyak upaya dalam pengembangan teknologi Informasi terkait dengan tingginya kepadatan sensor yang digunakan dalam konsep IoT (*Internet of Things*). IoT sendiri merupakan konsep dimana suatu node dapat saling berkomunikasi satu sama lain dengan berbagai protokol yang berbeda (Guoqiang, et al., 2013). Pengembangan *Internet of Things* selalu didukung dengan konsep WSN (*Wireless Sensor Network*). *Wireless Sensor Network* merupakan konsep yang terdiri dari node sensor yang cerdas dan dihubungkan menggunakan sistem jaringan nirkabel (I.F. Akyildiz, et al., 2001). Sebagian besar konsep WSN menggunakan konsumsi daya baterai, dengan demikian terdapat kebutuhan konstan untuk mengurangi kebutuhan energi. Terlebih lagi semakin banyak perangkat IoT yang terhubung diperkirakan mencapai 50 miliar perangkat pada tahun 2020 (Chase, 2013). Oleh karena itu, dibutuhkan teknologi yang memiliki penggunaan baterai rendah seperti yang terdapat pada teknologi LoRa (*Long Range*). Tak hanya memiliki daya yang rendah, teknologi LoRa memiliki ketahanan terhadap noise atau gangguan dari sinyal-sinyal yang tidak diinginkan yang selalu ada dalam suatu sistem transmisi yang nantinya dapat mengganggu dalam proses penerimaan dan pengiriman data. Teknologi *LoRa* juga memiliki jangkauan komunikasi lebih dari 2 km dengan konfigurasi, lingkungan yang sesuai, dan tipe jenis modul yang digunakan (Wixted, et al., 2016).

Dari kelebihan teknologi *LoRa* yang sudah dijelaskan, teknologi *LoRa* juga memiliki keterbatasan tidak dapat melakukan pengiriman data langsung ke server. Komunikasi yang digunakan teknologi LoRa adalah menggunakan gelombang radio, yang berarti dibutuhkan sebuah perangkat yang dapat menjembatani pengiriman data agar sampai ke server agar dapat dimonitoring. Maka dari itu diperlukan sistem perantara untuk mengintegrasikan komunikasi menggunakan teknologi LoRa ke pusat data yang dikenal dengan gateway.

Pada penelitian sebelumnya oleh *Chang-le Zhong* (2015) menyebutkan dalam penelitiannya yang berjudul "*Study on the IOT Architecture and Gateway Technology*" bahwa secara umum *gateway* dapat menghubungkan berbeda protokol komunikasi. *Gateway* juga berfungsi sebagai *protocol converter* dan menjadi penghubung berbagai macam protokol komunikasi. Agar *gateway* dan pusat data dapat saling berkomunikasi, maka dibutuhkan suatu protokol komunikasi untuk dapat saling bertukar informasi. Salah satu protokol yang dapat digunakan untuk perpesanan yaitu dengan protokol *MQTT*. Protokol *MQTT* digunakan karena keandalan dalam pengiriman data dan karakteristik mendukung kemampuan yang dimiliki IoT seperti dapat bekerja pada *low power* dan penggunaan *bandwidth* yang kecil. *MQTT* berjalan diatas TCP dan menggunakan model *Publish-Subscribe*. Model *Publish-Subscribe* didesain agar mudah untuk diimplementasikan, dan bersifat open (dhall & Solanki, 2017).

Pada penelitian ini, peneliti melakukan pengembangan *gateway* agar dapat menjadi perantara pengiriman data dari node sensor ke server. Data sensor akan mengirimkan data ke *gateway* menggunakan modul komunikasi *LoRa*. Kemudian *gateway* akan menerima data dari node sensor dan meneruskan ke server menggunakan protokol MQTT dengan menggunakan metode *publish-subscribe*. *Gateway* dan node sensor akan di hubungkan pada mikrokomputer Raspberry Pi karena memiliki komputasi yang lebih baik daripada mikrokontroler lain. Parameter pengujian yang akan dilakukan peneliti berupa *successfull rate gateway* dalam menerima dan meneruskan data sampai ke pusat data dengan variabel jarak, besar paket, dan interval.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka dapat dirumuskan beberapa permasalahan sebagai berikut:

1. Bagaimana penerapan *gateway* dapat menjadi perantara pengiriman data menggunakan modul komunikasi *LoRa* dan protokol *MQTT* pada WSN?
2. Bagaimana pengelolaan pertukaran data antara node sensor, gateway dan server?
3. Bagaimana kinerja *gateway* sebagai perantara pengiriman data dari node sensor ke server dengan menggunakan modul komunikasi *LoRa* dan protokol *MQTT*?

## 1.3 Tujuan

Dari hal yang telah dirumuskan pada rumusan masalah, dapat ditentukan tujuan skripsi ini sebagai berikut:

1. Agar *gateway* dapat berperan sebagai perantara komunikasi antara node sensor dengan server yang diterapkan pada WSN, sehingga data dari node sensor dapat dimonitoring melalui server.
2. Mengetahui pengolahan pertukaran data antara node sensor dan server melalui *gateway*.
3. Mengetahui kinerja dari *gateway* yang dibuat sebagai perantara pengiriman data dari node sensor ke server.

## 1.4 Manfaat

Dari fungsi yang akan diteliti dan diimplementasikan, dapat di ketahui manfaat sebagai berikut:

1. *Gateway* dapat membantu menjadi penghubung antara pengiriman data node sensor dan server.
2. Dapat membantu para peneliti dalam mengembangkan teknologi *LoRa* serta implementasinya ke dalam sistem yang dibuat.

3. Bagi penulis, penelitian ini dapat menjadi penerapan ilmu yang didapatkan selama kuliah. Serta menambah wawasan peneliti dalam mengeksplorasi teknologi yang sedang berkembang saat ini.

### 1.5 Batasan Masalah

Agar penyusunan skripsi penelitian ini sesuai dengan latar belakang dan fokus dengan rumusan masalah, maka batasan masalah pada penelitian ini sebagai berikut:

1. Faktor keamanan pada sistem yang dibuat, tidak menjadi fokus utama penelitian ini.
2. Perangkat LoRa *module* yang digunakan tipe HopeRF-RFM9x.
3. Frekuensi yang digunakan oleh modul LoRa HopeRF-RFM9x adalah 433 MHz.
4. Node sensor dan *gateway* menggunakan mikrokomputer Raspberry Pi.
5. Server tidak menyimpan data yang di dapat dari node sensor, melainkan hanya ditampilkan.
6. Penggunaan *qos* dalam metode *publish-subscribe* tidak menjadi fokus utama penelitian ini.
7. Data yang dipertukarkan node sensor dengan *gateway* berbentuk *struct*.
8. Data yang dipertukarkan *gateway* dengan server berbentuk *json*.

### 1.6 Sistematika Pembahasan

Sistematika pembahasan dari penyusunan penelitian yang direncanakan adalah sebagai berikut :

#### BAB 1 PENDAHULUAN

Pendahuluan terdiri dari latar belakang pembuatan tugas akhir, identifikasi, pembatasan masalah, rumusan masalah, tujuan dan manfaat serta sistematika penulisan dari tugas akhir.

#### BAB 2 LANDASAN KEPUSTAKAAN

Bab landasan kepastakaan menguraikan tentang teori – teori yang dipakai dalam penelitian, temuan dan bahan penelitian yang diperoleh dari beberapa referensi yang menunjang penelitian dalam penulisan tugas akhir.

#### BAB 3 METODOLOGI

Bab Metodologi ini menguraikan dan membahas langkah kerja dalam penulisan tugas akhir seperti studi literatur, perancangan dan implementasi, analisis kebutuhan sistem, pengujian serta analisis hasil yang di bahas secara umum.

#### **BAB 4 REKAYASA KEBUTUHAN**

Bab rekayasa kebutuhan menjelaskan tentang kebutuhan apa saja yang dibutuhkan sistem seperti kebutuhan perangkat lunak dan perangkat keras yang digunakan.

#### **BAB 5 PERANCANGAN SISTEM**

Bab perancangan sistem membahas tentang perancangan sistem yang berdasarkan bab sebelumnya. Pada bab ini menjabarkan perancangan sistem secara umum, alur kerja sistem, dan perancangan komponen utama sistem yang dibuat. Perancangan pengujian juga dijelaskan pada bab ini dan akan diterapkan pada bab selanjutnya.

#### **BAB 6 IMPLEMENTASI SISTEM**

Bab implementasi merupakan bab yang menjelaskan tentang bagaimana sistem akan diterapkan berdasarkan sistematika yang telah dibuat sebelumnya secara rinci beserta langkah-langkah pengerjaan yang dilakukan serta menampilkan gambar-gambar implementasi yang dilakukan. Setelah sistem berhasil diimplementasikan.

#### **BAB 6 PENGUJIAN SISTEM**

Bab Pengujian menampilkan skenario pengujian yang ada pada sistem ini. Bab ini juga menyajikan data hasil pengujian beserta hasil analisis dari implementasi sistem yang telah dibangun. Hasil Pengujian didapatkan dari pengujian yang telah dilakukan sesuai dengan skenario pengujian yang telah ditentukan. Dari hasil tersebut dapat dilakukan analisis terhadap kinerja sistem.

#### **BAB 7 PENUTUP**

Pada bab penutup dipaparkan hasil kesimpulan dari pelaksanaan penelitian. Kesimpulan ini dibuat dengan hasil pengujian dan analisis terhadap perancangan dan implementasi arsitektur sistem sebagai dasarnya. Saran-saran juga diberikan agar hasil dari penelitian ini dapat diperbaiki dan disempurnakan apabila penelitian ini dikembangkan kemudian.

## BAB 2 LANDASAN KEPUSTAKAAN

Pada Bab Landasan keputustakaan ini terdapat tinjauan pustaka dari penelitian yang telah ada dan yang diusulkan. Landasan teori dari berbagai sumber pustaka yang terkait dengan teori dan metode yang digunakan dalam penelitian.

### 1.1 Tinjauan Pustaka

Terdapat beberapa penelitian terdahulu yang memiliki kemiripan dengan topik penulis sebagai berikut:

**Tabel 2.1 Penelitian Terkait**

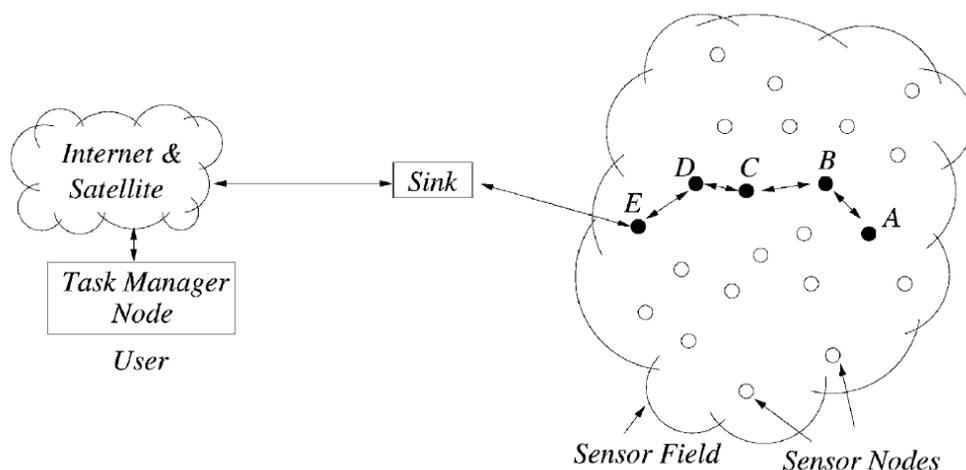
No	Judul	Sitasi	Hasil Penelitian	Penelitian Penulis
1	Study on the IOT Architecture and Gateway Technology	(Chang-le Zhong, 2015)	IoT <i>gateway</i> memiliki peran sebagai penghubung <i>devices</i> yang bermacam – macam dan memungkinkan komunikasi berbeda protokol.	Membuat <i>Gateway</i> agar dapat menjadi penghubung komunikasi antara node sensor dengan pusat data.
2	Evaluation of LoRa and LoRaWAN for Wireless sensor Networks	(Wixted, et al., 2016)	Mengevaluasi kinerja teknologi LoRa pada <i>Wireless Sensor Network</i>	Menggunakan protokol komunikasi LoRa sebagai pengiriman data dari node sensor ke gateway
3	Rancang Bangun lot Cloud Platform Berbasis Protokol Komunikasi MQTT	(Habibi, et al., 2017)	MQTT diterapkan untuk menyelesaikan masalah heterogenitas dan penggunaan cloud karena keterbatas penyimpanan pada perangkat IoT	Menggunakan protokol MQTT sebagai komunikasi antara <i>gateway</i> dengan server

## 1.2 Kajian Teori

Dalam melakukan penelitian ini adapun dasar teori yang mendasari penelitian “Pengembangan Sistem Perantara Pengiriman Data Sensor Menggunakan Modul Komunikasi LoRa Dan protokol MQTT Pada *Wireless Sensor Network*” ini yaitu *Internet of Things*, *Wireless Sensor Network (WSN)*, *Gateway*, *Rasspberry Pi*, *Arduino Nano*, *LoRa (Long Range)*, *Message Queuing Telemetry Transport (MQTT)*, *Publish-Subscribe*, dan *Python*.

### 1.2.1 Wireless Sensor Network (WSN)

Wireless Sensor Network merupakan sebuah infrastruktur jaringan yang terdiri dari komputasi, penginderaan, dan unsur-unsur lain yang dapat mengamati dan bereaksi terhadap peristiwa dan fenomena di lingkungan tertentu. Sebelumnya sensor langsung terhubung dengan controller dan diproses secara langsung. Namun ketika terjadi peningkatan pada jumlah sensor maka pengolahan data dilakukan secara terpusat. Karena berkembangnya teknologi penggunaan aplikasi jaringan tidak hanya membutuhkan beberapa atau puluhan sensor, tetapi dapat ratusan atau bahkan ribuan node sensor. Berikut merupakan konsep sederhana yang dapat menjelaskan tentang arsitektur WSN dapat dilihat pada gambar 2.1.



**Gambar 2.1 Arsitektur *Wireless Sensor Network***

Sumber: (I.F. Akyildiz, et al., 2001)

Pada gambar 2.1 terdapat beberapa komponen dasar penyusun *Wireless Sensor Networks* seperti: (1) sebuah node sensor yang berfungsi membaca lingkungan atau objek yang akan dipantau. Dalam hal ini untuk keperluan pembacaan dan penginderaan diperlukan beberapa perangkat sensor, (2) selanjutnya node yang berfungsi sebagai pengumpul penginderaan dari node sensor yang bisa disebut dengan *Sink*, (3) jaringan interkoneksi (biasanya berbasis nirkabel), (4) perangkat komputer (user) yang menangani dalam pengolahan data yang didapat dari node sensor seperti korelasi, *event trending*, *status query*, dan *data mining*.

### 1.2.2 Internet of Things

*Internet of Things* merupakan konsep dimana suatu objek mati dapat saling berkomunikasi satu sama lain dengan berbagai protokol yang berbeda (Guoqiang, et al., 2013). IoT memungkinkan suatu objek dapat “melihat”, “mendengar”, dan “berfikir” dengan mengharuskan untuk saling “berkomunikasi” dan dengan objek lain. Dalam hal ini, IoT dapat merubah benda mati seakan-akan menjadi hidup dengan memanfaatkan teknologi pook yang terdapat pada objek tersebut. IoT sendiri juga memiliki komponen utama untuk dapat menjalankan fungsionalitasnya yang terdiri dari *constrained device* (aktuator dan sensor), *gateway* dan *cloud*.

*Constrained device* adalah komponen yang memiliki tugas tertentu serta memiliki fungsi komputasi terbatas dan sering dikaitkan dengan hal fisik seperti sensor yang terhubung ke mikrokomputer. Selanjutnya yaitu *gateway*, komponen ini berfungsi sebagai titik koordinasi aktuator terhubung ke jaringan luar. Komponen terakhir yaitu *cloud*, komponen ini memiliki peran sebagai penyimpan data dalam jumlah besar. Cara kerja *Internet of Things* juga cukup mudah. Setiap benda harus memiliki identitas yaitu *IP Address* yang membuat benda tersebut dapat di perintah dari benda lain dalam jaringan yang sama. Dan benda tersebut dapat dihubungkan ke jaringan internet.

### 1.2.3 Gateway

*Gateway* merupakan suatu perangkat yang menghubungkan lebih dari satu jaringan komputer dengan media komunikasi yang berbeda (Yacchirema & Palau, 2016). Gateway sendiri merujuk pada hardware atau software yang menjembatani kedua aplikasi agar dapat saling berkomunikasi. Salah satu fungsi *gateway* adalah melakukan *protocol converting*, yang dapat menghubungkan dua protokol jaringan yang berbeda agar dapat saling berkomunikasi seperti penggunaan *gateway* adalah pada email, sehingga pertukaran email dapat dilakukan pada sistem yang berbeda karena salah satu fungsi *gateway* adalah melakukan konversi protokol, agar dua arsitektur atau protokol yang berbeda dapat saling berkomunikasi satu sama lain.

### 1.2.4 Raspberry Pi

*Raspberry Pi* adalah sebuah SBC (Single Board Computer) memiliki ukuran sebesar kartu kredit dan memiliki input output digita port seperti pada sebuah board mikrocontroller yang dikembangkan oleh *Raspberry Pi Foundation* di Inggris. Raspi bermula di design untuk sitem operasi linux, namun seiring berkembangnya zaman tak hanya sistem operasi linux yang mengembangkan tetapi windows juga telah mengembangkan sistem operasinya untuk *Raspberry Pi* dalam bentuk IoT. *Raspberry Pi* menggunakan system on a chip (SoC) dari Broadcom BCM2853, juga sudah termasuk prosesor ARMv10, kecepatan prosesor 700MHz – 1GHz dan 4 GPU, *Raspberry Pi* Model B ini memiliki RAM sebesar 512 MB, dan untuk menyimpan data layaknya komputer, laptop menggunakan Hardisk sebagai media penyimpanannya, tetapi *Raspberry Pi* ini hanya menggunakan kartu memori yang biasa digunakan untuk penyimpanan data di handphone baik berbasis symbian, android. Untuk dapat mnghubungkan antara *Raspberry Pi* dengan hardware lain digunakanlah

GPIO. Terdapat banyak pin dalam GPIO dengan tiap- tiap pin memiliki fungsi masing – masing, hal ini dapat dilihat lebih lengkap pada tabel 2.2.

**Tabel 2.2 Komponen GPIO pada Raspberry Pi**

Komponen	Fungsi
Power	Pin power ini berguna untuk memberikan aliran listrik pada perangkat yang tersambung pada <i>Raspberry Pi</i> . Terdapat 2 macam pin power yaitu 5V dan 3V
Ground (GND)	Pin Ground digunakan untuk menerima ground dari perangkat yang tersambung. Ground digunakan untuk melindungi perangkat dari tegangan berbahaya ketika insulasi elektrik mengalami kegagalan
SPI	Pin SPI memungkinkan untuk menghubungkan modul hardware dengan hanya dua pin kontrol
GPIO	Pin GPIO digunakan untuk mengirim dan menerima data. Pin ini dibuat agar <i>Raspberry Pi</i> bisa diakses oleh perangkat dengan beberapa Bahasa seperti C, Ruby, dan Python
Do Not Connect(DNC)	Pin DNC disediakan untuk koneksi dengan hardware BCM2836, Selain dari itu, apabila dihubungkan menggunakan pin ini, maka <i>Raspberry Pi</i> akan mengalami kerusakan

**Tabel 2.3 Spesifikasi pada Raspberry Pi 2 Model B**

SoC	Broadcom BCM2836 (CPU, GPU, DSP, SDRAM)
CPU	900 MHz quad-core ARM Cortex A7 (ARMv7 instruction set)
GPU	Broadcom VideoCore IV @ 250 MHz
Memory	1 GB (shared with GPU)
USB Port	4
Video Input	15-pin MIPI camera interface (CSI) connector
Video Output	15-pin MIPI camera interface (CSI) connector
Audio Input	I <sup>2</sup> S
Audio Output	Analog via 3.5 mm jack; digital via HDMI and I <sup>2</sup> S
Storage	MicroSD
Network	10/100Mbps Ethernet
Peripherals	17 GPIO plus specific functions, and HAT ID bus
Power Rating	800 mA (4.0 W)
Power Source	5V Via MicroUSB or GPIO header

Size	85.60 mm x 56.5 mm
Weight	45g (1.6 oz)
GPIO	40



**Gambar 2.2 Raspberry Pi 2 Model B**

Sumber: [www.raspberrypi.org](http://www.raspberrypi.org), diakses 12-07-2018

**Tabel 2.4 Spesifikasi pada Raspberry Pi 3 Model B**

SoC	BCM2837
CPU	Quad Cortex A53 @1.2GHz
Instruction Set	ARMv8-A
GPU	400 Mhz VideoCore IV
RAM	1GB SDRAM
Storage	microSD
Ethernet	10/100Mbps
Wireless	802.11n / Bluetooth 4.0
Video Output	HDMI / Composite
Audio Output	HDMI / Headphone
Size	85.60 mm x 56.5 mm
Weight	45g (1.6 oz)
GPIO	40

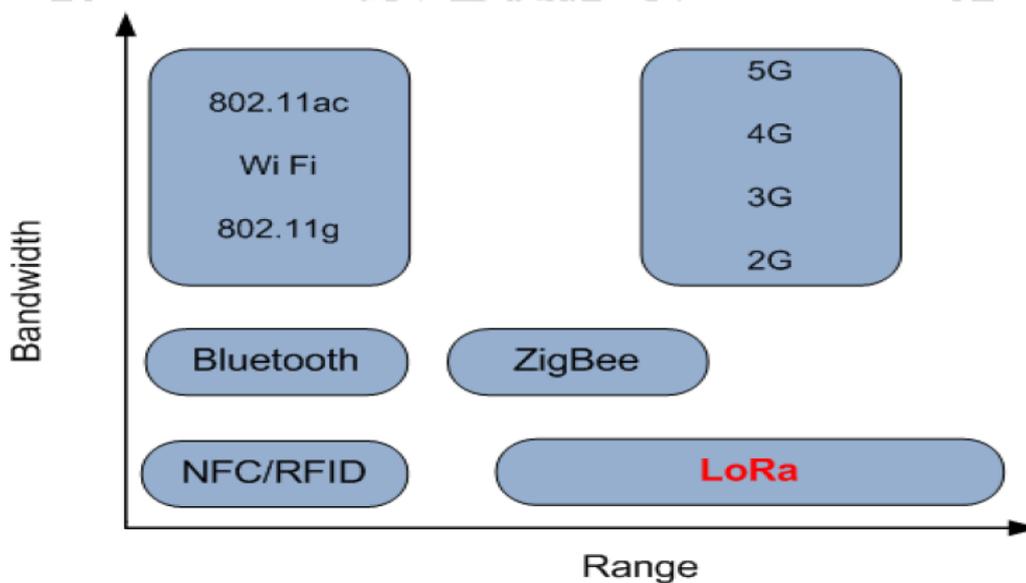


Gambar 2.3 Raspberry Pi 3 Model B

Sumber: www.raspberrypi.org, diakses 12-07-2018

### 1.2.5 LoRa (Long Range)

LoRa merupakan teknologi nirkabel yang dikembangkan oleh Semtech untuk memungkinkan komunikasi dengan *data rate* rendah yang dibuat untuk komunikasi dengan jarak yang jauh (Wixted, et al., 2016). Jarak yang dapat ditempuh dengan menggunakan teknologi LoRa dapat mencapai lebih dari 2 km tergantung jenis modul LoRa dan antena yang digunakan. Pemilihan lingkungan juga berpengaruh dalam hal jarak, terlebih lagi di lingkungan perkotaan. Semakin padat penduduk serta banyaknya rumah, maka besar pula RSSI (*Received Signal Strength Indicator*) yang di dapatkan. Namun lain halnya jika di daerah di pedesaan, tidak padatnya populasi penduduk dan sedikit rumah yang ada dapat mendukung kinerja LoRa dalam hal jarak lebih baik. Arsitektur yang digunakan LoRa disebut LPWAN yang dapat menghubungkan lebih dari ratusan bahkan sampai jutaan node. LoRa memiliki kelebihan dibandingkan jenis komunikasi lainnya, seperti dapat dilihat pada Gambar 2.4



Gambar 2.4 Perbandingan Protokol Komunikasi Wireless

Sumber: (Lavric & Popa, 2017)

Pada gambar 2.4 dibandingkan dengan *Zigbee*, *Wifi*, dan perangkat lainnya, *LoRa* memiliki masa baterai paling lama lebih dari 10 tahun karena memiliki konsumsi daya yang rendah sehingga penggunaannya sangat cocok untuk dioperasikan pada area yang cukup luas. Namun *LoRa* memiliki keterbatasan dalam kecepatan pengiriman data yaitu pada kisaran 0.3-50 kbps (Lavric & Popa, 2017). Walaupun demikian hal ini tidak menjadi masalah jika data yang dikirimkan sensor terbilang kecil. Hal ini sangat cocok untuk transmisi sensor suhu dan kelembapan, kelembapan tanah, curah hujan, dan lain lain.

*LoRa* juga memiliki elemen yang berfungsi sangat penting dalam kinerjanya seperti elemen *RF interface / physical layer* yang mengatur aspek sinyal RF yang ditransmisikan antara node atau titik akhir, yaitu node sensor dan *gateway* tempat sinyal diterima. Elemen ini mengatur aspek sinyal termasuk frekuensi, format modulasi yang menggunakan *Chirp Speed Spectrum* (CSS) yang dikembangkan oleh Semtech, tingkat daya yang digunakan, dan sinyal antara elemen pemancar dan penerima.

### 1.2.6 *Message Queuing Telemetry Transport* (MQTT)

*MQTT* adalah protokol yang pertama kali dikenalkan oleh Dr Andy Stanford-Clark dan Arlen Nipper pada tahun 1999. *MQTT* merupakan M2M (*Machine to Machine*) protokol. *MQTT* didesain simple, ringan, dan mudah diimplementasikan untuk *messaging*. Karakteristik ini membuat *MQTT* menjadi sangat ideal untuk digunakan pada berbagai situasi termasuk situasi dimana lingkungan dengan sumber daya yang terbatas seperti komunikasi antara *Machine to Machine* dan *Internets of Things* dimana hanya dibutuhkan sedikit baris kode dan *bandwith* jaringan yang bagus adalah suatu kemewahan (Oasis, 2015). *MQTT* dirancang untuk berjalan di atas protokol TCP/IP. Karena menggunakan metode komunikasi *Publish-Subscribe*, maka *MQTT* menggunakan *topic*, dimana *topic* menentukan kemana pesan dari *publisher* akan diteruskan. *Topic* adalah data yang bertipe *string*. *MQTT* dirasa tepat untuk menjadi protokol *IoT* karena *MQTT* bersifat *light weighted message* dan di desain untuk perangkat yang memiliki sumber daya terbatas (Kim, et al., 2015).

### 1.2.7 *Publish-Subscribe* (Pub-Sub)

*Publish-Subscribe* adalah sebuah metode pertukaran pesan secara tidak langsung (*indirect communications*). Pada komunikasi seperti ini, *publisher* membuat *event*, kemudian *event* tersebut dikirim ke *Broker*. *Broker* berfungsi untuk mengirimkan pesan yang sudah dibuat oleh *publisher* ke *subscriber*, *Broker* hanya akan mengirimkan konten kepada *subscriber* yang sudah melakukan *sub* ke konten-konten tertentu. Semisal *Publisher* melakukan *pub* konten A, maka *Broker* hanya akan memberikan konten A ke *subscriber* yang melakukan *sub* ke konten A. Karena pada metode komunikasi ini tidak ada interaksi secara *end to end*, maka hal ini menyebabkan *Pub-Sub* termasuk ke dalam metode *indirect communications*. Dengan adanya *Publish-Subscribe*, seseorang tidak diharuskan menerima semua informasi yang ada, tetapi informasi yang sesuai dengan kebutuhannya tanpa

mengganggu kerja utama sistem yang ada. Model *Publish-Subscribe* didesain agar mudah untuk diimplementasikan dan bersifat open (Dhall & Solanki, 2017).

### 1.2.8 Python

Menurut Dr. Andrew N Harrington (2009), bahasa python adalah bahasa pemrograman yang memiliki banyak fungsi, interaktif, berorientasi objek dan merupakan bahasa pemrograman tingkat tinggi. Bahasa python merupakan bahasa pemrograman formal dengan aturan-aturan dan format spesifiknya sendiri.

*Python* adalah bahasa tingkat tinggi dengan *syntax* yang sangat jelas dan tersusun rapi. *Python* juga memiliki banyak *library* dan *system call*. Selain itu juga dapat digunakan sebagai *extension language* untuk aplikasi yang memerlukan *interface* yang dapat diprogram (Python, 2017). Tidak seperti bahasa pemrograman lain yang susah dibaca dan dipahami, *python* lebih menekankan pada keterbacaan kode agar lebih mudah memahami sintaks. Bahasa pemrograman ini mendukung hampir semua sistem operasi, bahkan untuk sistem operasi linux hampir semua distronya sudah menyertakan *python* di dalamnya.

### 1.2.9 Parameter pengujian

Parameter yang diuji pada sistem yang dibuat adalah *successfull rate gateway*. *Successfull rate gateway* merupakan tingkat keberhasilan *gateway* dalam meneruskan paket baik dari node sensor ke *gateway* ataupun *gateway* ke node sensor. Parameter pengujian ini akan dihitung ketika *gateway* menerima data dari node sensor kemudian *gateway* meneruskan ke server. Berikut persamaan dari *successfull rate gateway*:

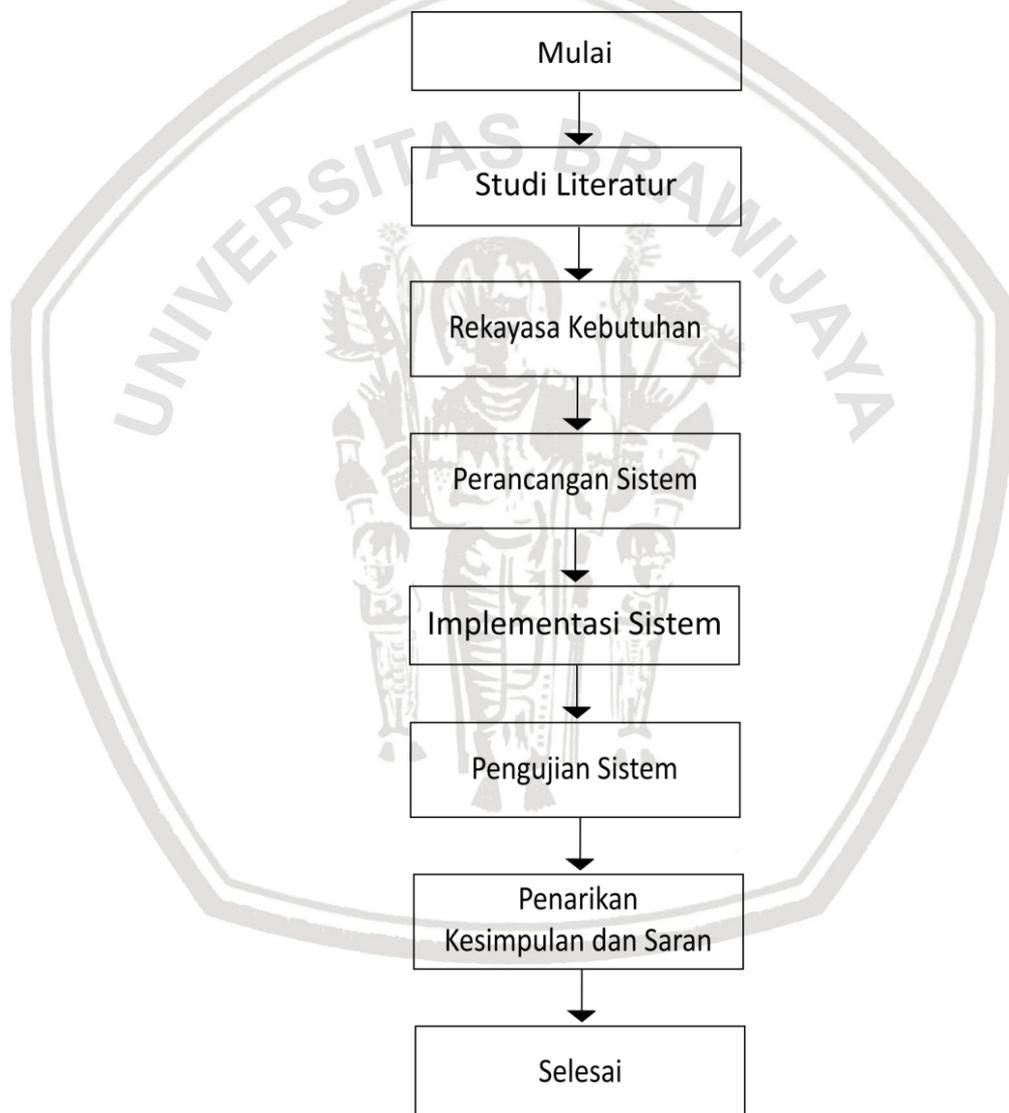
$$\text{successfull rate} = \left( \frac{\text{paket yang dikirim} - \text{paket yang gagal}}{\text{paket yang dikirim}} \times 100\% \right)$$

## BAB 3 METODOLOGI

Bab ini menguraikan setiap langkah yang akan ditempuh dalam proses penelitian yang akan dilakukan diantaranya adalah studi literatur, rekayasa kebutuhan, perancangan, implementasi, pengujian dan analisis, dan yang terakhir adalah proses pengambilan kesimpulan.

### 1.1 Metodologi Penelitian

Metodologi penelitian pada tugas akhir ini membahas alur pelaksanaan penelitian sehingga proses penelitian dapat terarah dengan baik dan sesuai dengan tujuan. Pada gambar 3.1 merupakan tahap penelitian yang ditunjukkan dalam bentuk *flowchart*.



**Gambar 3.1 Alur Penelitian**

Pada Gambar 3.1 menjelaskan alur penelitian yang akan dilakukan dimulai dari studi literatur untuk memperdalam pemahaman sistem yang akan dibuat. Studi literatur berisi penelitian terdahulu serta dasar teori yang akan digunakan sebagai

pemahaman dari sistem yang akan dibuat. Selanjutnya rekayasa kebutuhan yang digunakan untuk menganalisis kebutuhan apa saja untuk membangun sistem yang dilakukan pada bab tiga.

Setelah melakukan analisis kebutuhan yang dibutuhkan sistem, tahap selanjutnya adalah melakukan perancangan dari apa yang sudah dianalisis pada bab tiga. Perancangan sistem menjabarkan bagaimana sistem bekerja dari perancangan sistem secara umum, alur kerja sistem, dan perancangan komponen utama sistem. Perancangan sistem merupakan tahap dasar untuk mengimplementasikan sistem yang dibuat. Pada tahap implementasi diharapkan agar sistem dapat berjalan dengan semestinya sesuai dari apa yang dirancang dan di analisis pada bab sebelumnya.

Setelah melakukan implementasi sistem yang dibuat, selanjutnya terdapat tahap pengujian dan analisis. Tahap ini dilakukan ketika sistem berhasil diimplementasikan. Tahap pengujian dibuat melalui skenario yang telah disepakati untuk menguji sistem secara fungsional dan non fungsional. Hasil dari pengujian ini akan dianalisis guna memenuhi tahap selanjutnya yaitu penarikan kesimpulan dan saran. Tahap penarikan kesimpulan dilakukan berdasarkan dari penelitian yang telah dilakukan dan selanjutnya diberikan saran untuk referensi pengembangan sistem selanjutnya.

## 1.2 Studi Literatur

Sebagai bahan penunjang pengerjaan penelitian ini dilakukan studi literatur untuk memberikan pengetahuan yang cukup dalam mengembangkan sistem yang penulis buat. Studi literatur dilaksanakan dengan cara mengumpulkan teori dan pustaka yang berkaitan dengan penelitian ini yang meliputi:

1. *Wireless Sensor Networks (WSN)*

Pada bagian ini, dilakukan studi mengenai arsitektur *Wireless Sensor Networks* serta menjelaskan kelebihan dan komponen apa saja yang terdapat pada WSN.

2. *Internet of Things*

Pada bagian ini, dilakukan studi mengenai konsep *Internet of Things* dan kelebihan menggunakan konsep ini. Serta menjelaskan komponen apa saja yang terdapat pada *Internet of Things*.

3. *Gateway*

Pada bagian ini, dilakukan studi mengenai konsep *gateway* dan bagaimana cara *gateway* diterapkan. Salah satu alasannya adalah *gateway* dapat menjadi perantara untuk komunikasi dua protokol.

4. *Raspberry Pi*

Pada bagian ini, membahas terkait spesifikasi Raspberry Pi dan kelebihannya dibanding mikrokomputer lain yang sangat memungkinkan dijadikan sebagai *gateway* dari beberapa node sensor.

### 5. LoRa (*Long Range*)

Pada bagian ini, dilakukan studi mengenai konsep *Long Range* dan kelebihan menggunakan konsep ini. Alasan memilih LoRa yaitu untuk mengatasi masalah jarak sebagaimana dijelaskan di latar belakang.

### 6. *Message Queuing Telemetry Transport (MQTT)*

Pada bagian ini, dilakukan studi mengenai konsep komunikasi pada protokol *Message Queuing Telemetry Transport (MQTT)*. Tentang bagaimana protokol ini berkerja serta kelebihan menggunakan protokol ini.

### 7. *Publish-Subscribe*

Pada bagian ini, dilakukan studi mengenai metode *Publish-Subscribe* dan bagaimana cara kerja serta kelebihan menggunakan konsep ini.

### 8. Python

Pada bagian ini, dilakukan studi tentang apa itu python. Bagaimana cara kerja python dan kelebihan menggunakan python ini.

## 1.3 Rekayasa Kebutuhan

Rekayasa kebutuhan diperlukan untuk menganalisis kebutuhan yang diperlukan pada penelitian ini. Rekayasa kebutuhan pada penelitian ini dijelaskan sebagai berikut:

### 1.3.1 Analisis kebutuhan

Analisis kebutuhan dibutuhkan untuk mencari tahu kebutuhan apa saja yang dibutuhkan agar sistem dapat berjalan sebagaimana mestinya. Analisis kebutuhan juga berfungsi mengetahui hal apa saja yang dibutuhkan dalam membuat *gateway* sebagai perantara komunikasi dari node sensor ke server. Dalam membuat sistem yang kompleks, mendefinisikan kebutuhan sistem terlebih dahulu dapat mempermudah untuk melakukan tahap perancangan, implementasi dan pengujian sistem.

### 1.3.2 Analisis Kebutuhan Sistem

Kebutuhan sistem merupakan suatu kebutuhan yang harus terpenuhi agar sistem dapat berjalan sesuai tujuan awal. Adapun kebutuhan sistem yang peneliti buat sebagai berikut:

1. Node sensor dapat mengambil data dari sensor yang terhubung dengan node sensor,
2. Node sensor dapat mengirimkan data sensor menuju *gateway* dengan menggunakan modul komunikasi *LoRa*,
3. *Gateway* dapat menerima data dari node sensor kemudian diteruskan ke server dengan protokol MQTT,
4. Server dapat menerima data dari *gateway* kemudian data yang diterima akan ditampilkan di monitor server.

### 1.3.3 Analisis Kebutuhan Perangkat Keras

Pada bagian ini menjelaskan terkait dengan kebutuhan perangkat keras pada sistem ini. Perangkat keras ini meliputi dari sekumpulan komponen elektronika yang akan membentuk suatu sistem perangkat keras baru yang memiliki fungsi tertentu. Berikut adalah kebutuhan perangkat keras yang digunakan pada proses implementasi sistem, yakni sebagai berikut :

**Tabel 3.2 Kebutuhan Perangkat Keras**

Nama Perangkat	Keterangan
Raspberry Pi	Mikrokomputer yang bertugas sebagai otak untuk menjalankan kode program system yang dibangun.
LoRa <i>module</i> HopeRF-RFM9x	Perangkat ini berfungsi sebagai <i>transceiver</i> antara node sensor dan <i>gateway</i> .
Sensor	Sensor yang berfungsi mengukur kelembapan tanah dipasangkan di node sensor.
Server	Perangkat yang berfungsi sebagai <i>end point</i> dari sistem yang dibuat.

Tabel 3.2 merupakan tabel kebutuhan perangkat keras yang termasuk dalam semua komponen pada node sensor, *gateway*, dan server. Perancangan perangkat keras diatas merupakan komponen sistem yang akan peneliti bangun sesuai tujuan dari penelitian ini. Seperti mikrokomputer Raspberry Pi yang akan di gunakan pada sisi node sensor dan *gateway* untuk menjalankan kode program. Modul LoRa HopeRF-RFM9x sebagai media komunikasi node sensor ke *gateway* dengan menggunakan frekuensi 433MHz. Server yang digunakan untuk manampikan data yang diperoleh.

### 1.3.4 Analisis Kebutuhan Perangkat Lunak

Pada bagian ini menjelaskan terkait dengan kebutuhan perangkat lunak yang digunakan pada sistem ini. Berikut adalah kebutuhan perangkat lunak yang digunakan pada proses implementasi sistem, yakni sebagai berikut :

**Tabel 3.3 Kebutuhan Perangkat Lunak**

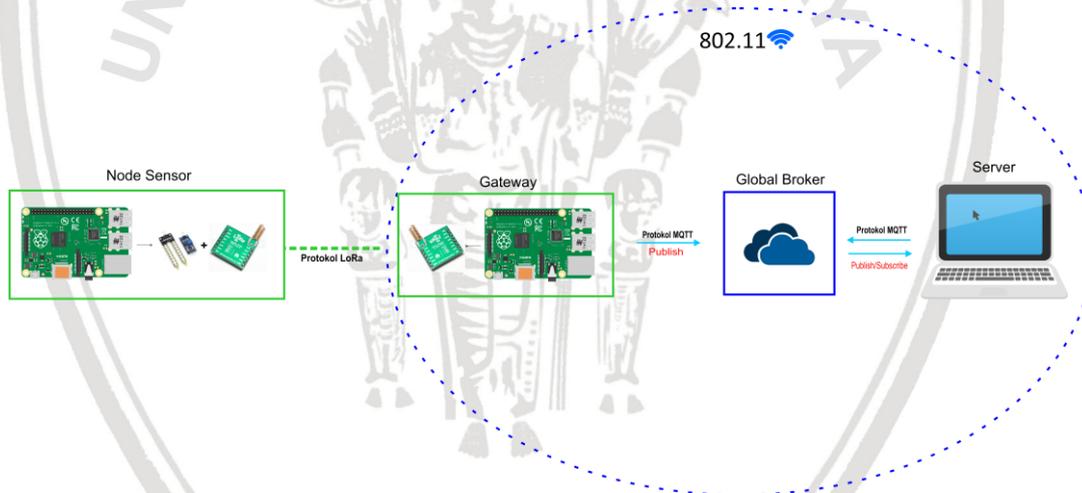
Nama Perangkat	Keterangan
Sistem Operasi	Perangkat Lunak yang digunakan untuk membuat kode program dan menjalankan kode program.
Library rf95	Library yang ditambahkan ke Bahasa Python untuk memudahkan pemrograman pada LoRa.
Library paho-mqtt	Library yang ditambahkan ke Bahasa Python untuk mendefinisikan fungsi-fungsi agar mempermudah dalam menggunakan proses <i>Publish-Subscribe</i> .

<i>Text Editor</i>	Teks editor digunakan untuk menulis kode program Python.
--------------------	--

Pada Tabel 3.3 dipaparkan macam-macam perangkat lunak yang digunakan untuk menjalankan sistem pada penelitian ini. Pada table diatas termasuk dalam semua komponen perangkat lunak yang dibutuhkan pada node sensor, gateway, dan server. Sistem operasi merupakan perangkat lunak yang berfungsi sebagai sistem dasar dari sebuah perangkat. Dalam hal ini tanpa adanya sistem operasi maka sistem yang peneliti buat tidak akan dapat berjalan sebagaimana mestinya. Seperti text editor yang digunakan peneliti untuk menulis kode program yang kemudian dijalankan pada mikrokomputer. Serta tidak lupa dengan peran *library* sebagai alat bantu untuk memudahkan peneliti agar sistem dapat berjalan sesuai tujuan awal dibuatnya.

### 1.4 Perancangan Sistem

Setelah dilakukan analisis kebutuhan yang diperlukan sistem. Pada tahap perancangan sistem akan dilakukan perancangan yang diperlukan agar sistem berjalan sebagaimana mestinya. Perancangan sistem dilakukan meliputi perancangan node sensor dan perancangan gateway. Dari hasil analisis maka dibuatlah perancangan sistem berdasarkan analisis kebutuhan yang telah dibuat sebelumnya. Berikut akan dijabarkan dalam gambar 3.2.



**Gambar 3.2 Rancangan Arsitektur Sistem**

Pada gambar 3.2 ditunjukkan rancangan arsitektur sistem secara umum yang akan di implementasikan. Terdapat beberapa komponen utama yaitu node sensor, gateway dan server. Pada komponen node sensor terdapat satu mikrokomputer Raspberry Pi dan sensor pendukung seperti *moisture sensor* untuk mengukur kelembapan pada tanah dan modul *LoRa HoperRF-RFM9x* berperan sebagai *transceiver* antara node sensor dengan gateway. Ketiga perangkat ini dihubungkan melalui pin-pin yang ada dengan menggunakan kabel *jumper*.

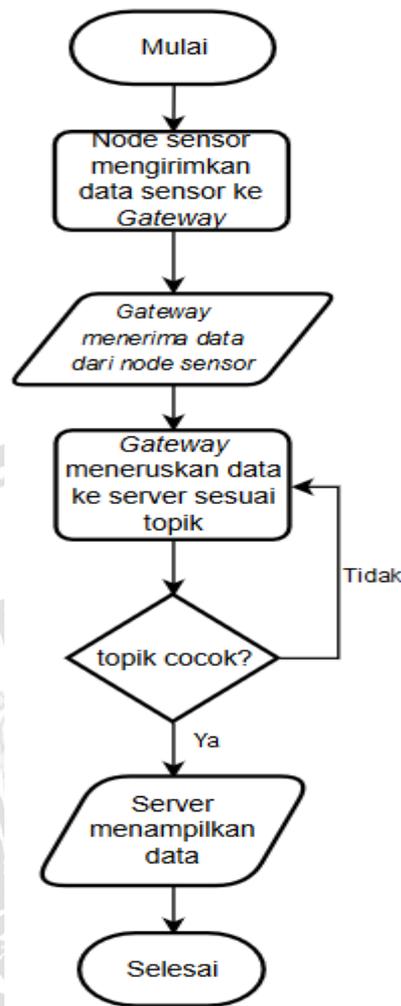
Komponen gateway pada sistem ini terdiri dari dua perangkat keras yaitu satu mikrokomputer Raspberry Pi untuk menjalankan kode program yang dibuat

dan modul LoRa HopeRF-RFM9x sebagai *transceiver* penghubung dengan node sensor. komponen ini sama dengan komponen pertama, yang menggunakan kode program yang di buat dengan bahasa Python dan selanjutnya di jalankan di Raspberry Pi. Tidak hanya menerima paket dari node sensor, perangkat gateway ini juga berperan sebagai perantara komunikasi dengan server. Dimana data yang di diterima dari node sensor akan diteruskan ke server. Kedua perangkat ini dihubungkan menggunakan kabel *jumper* melalui pin-pin yang telah tersedia di kedua perangkat.

Server yang terdapat pada sistem ini terdiri dari sebuah laptop yang mana data dari node sensor akan di kirim ke server melalui *gateway*. *gateway* mengirim data ke server menggunakan fungsi *publish* yang dikirim ke *broker*, kemudian *broker* meneruskan data ke server yang telah melakukan *subscribe* dengan menerima paket yang sesuai permintaan. agar *gateway* dan server dapat saling berkomunikasi satu sama lain, peneliti membuat jaringan internet lokal dengan menggunakan protokol 802.11. Untuk memperjelas alur kerja sistem secara detail serta bagaimana data diterima dan dikirim setiap komponen dan rancangan komponen sistem, dapat dilihat pada sub bab berikutnya.

#### 1.4.1 Perancangan alur kerja sistem

Pada bagian ini akan dijabarkan *flowchart* gambaran dari alur kerja sistem keseluruhan yang peneliti buat. Alur kerja sistem terdiri dari node sensor, *gateway*, dan server. Namun sebelum membahas satu persatu alur kerja sistem akan diberikan gambaran umum dan cara kerja sistem. Untuk mempermudah penjelasan mengenai alur kerja sistem dapat dilihat pada gambar 3.3.



**Gambar 3.3 Flowchart Alur Kerja Sistem Secara Umum**

Pada gambar 3.3 menunjukkan *flowchart* alur kerja sistem secara umum. Pertama kali yang harus dilakukan sistem adalah node sensor mengirimkan data berupa data pengambilan sensor yang di *pack* dengan menggunakan *struct* ke *gateway*. Kemudian *gateway* menerima data dari node sensor dan meneruskan ke server menggunakan protokol MQTT sesuai dengan topik yang telah ditentukan. Paket akan dikirimkan ke *broker* dan *broker* akan meneruskan ke server. Jika topik yang dikirim tidak sesuai dengan topik yang di *subscribe* server maka data akan di *drop* oleh server, jika topik sesuai data akan ditampilkan di server. Setelah dijelaskan alur kerja sistem secara umum, perancangan akan di bagi menjadi 3 bagian yang meliputi perancangan node sensor, perancangan *gateway*, dan perancangan pengujian.

Node sensor akan berfungsi melakukan pengambilan data sensor lalu mengirimkan ke *gateway* dengan menggunakan modul LoRa HoperF-RFM9x. Maka dari itu perancangan node sensor di perlukan agar mempermudah peneliti dalam melakukan implementasi sistem. Dalam pengembangan sistem ini digunakan Raspberry Pi sebagai mikrokomputer untuk menjalankan sensor yang telah dihubungkan dengan kabel *jumper*. Kode program ditulis dengan menggunakan *text*

*editor* yang nanti akan di jalankan pada mikrokontoller tersebut. Kode program tersebut akan mengambil data dari sensor yang kemudian akan di kirim ke *gateway*.

Perancangan selanjutnya yaitu perancangan *gateway* yang akan menjadi sebuah perantara komunikasi antara node sensor dan server. Agar *gateway* dapat menjalankan fungsinya, maka dibuat kode program menggunakan *text editor* dengan bahasa pemrograman Python yang kemudian di tanamkan ke perangkat keras berupa Raspberry Pi yang di hubungkan modul LoRa HopeRF-RFM9x sebagai *transceiver* antara *gateway* dengan node sensor. Kemudian, untuk mengirimkan data yang di dapat ke server akan dikirim menggunakan protokol MQTT. Yang mana data akan di *pack* ke json yang kemudian akan dikirim sesuai topik yang di *subscribe* oleh server.

Perancangan bagian akhir akan menjelaskan skema pengujian yang akan berguna untuk mempermudah apa yang akan diujikan nantinya. Pada perancangan pengujian akan jelaskan menjadi dua bagian meliputi pengujian fungsional dan pengujian non-fungsional berupa pengujian kinerja sistem yang dibuat.

## 1.5 Implementasi Sistem

Implementasi yang merupakan tindak lanjut dari perancangan yang sudah disusun sebelumnya. Implementasi di buat sesuai rancangan yang sudah ada dan pada bab ini terdapat dua komponen yang harus di implementasikan yaitu implementasi node sensor dan implementasi *gateway*.

### 1. Implementasi Node Sensor

Untuk mengimplementasikan node sensor pertama dimulai dengan membangun perangkat keras yang dibutuhkan. Komponen perangkat keras tersebut terdiri dari tiga komponen utama seperti mikrokomputer Raspberry Pi, sensor, dan modul LoRa HopeRF-RFM9x. Ketiga perangkat keras dihubungkan dengan kabel *jumper*. Kemudian untuk *pinout* perangkat tersebut akan dibahas lebih detail pada bab perancangan.

Tahap selanjutnya yaitu mengimplementasikan perangkat lunak yang dibutuhkan berupa kode program untuk node sensor. Kode program yang akan ditampilkan tidak keseluruhan melainkan fungsi utama dari node sensor saja, untuk lebih detail mengenai pembahasan kode program akan di bahas pada bab implementasi sistem.

### 2. Implementasi Gateway

Implementasi *gateway* merupakan tahap setelah node sensor berhasil di implemenstasikan. Seperti sebelumnya, implementasi *gateway* dimulai dari membangun perangkat keras yang dibutuhkan. Komponen perangkat keras tersebut terdiri dari dua komponen yaitu Raspberry Pi sebagai mikrokomputer dan modul LoRa HopeRF-RFM9x sebagai *transceiver* antara node dan *gateway*.

Tahap selanjutnya yaitu mengimplementasikan perangkat lunak yang dibutuhkan berupa kode program untuk menjalankan *gateway* agar

dapat menjadi perantara data yang dikirim dari node sensor ke server. Server bertugas menerima dan mengirim data ke *gateway* lalu menampilkannya. Karena fokus utama pada implementasi yaitu kode program, maka di bab ini akan dijelaskan mengenai fungsi dari kode program yang digunakan.

## 1.6 Pengujian Sistem

Pada tahap ini, sistem berhasil diimplementasikan dan peneliti melakukan pengujian terhadap sistem yang telah di buat dengan menggunakan paramater uji yang telah di siapkan. Proses pengujian dibagi menjadi pengujian secara fungsional dan non-fungsional. Pengujian fungsional dilakukan dengan beberapa skenario yang akan di bahas pada bab perancangan. Tahap pengujian ini akan menguji fungsional *gateway* yang diharapkan dapat menjadi perantara komunikasi antara node sensor dan server.

Proses pengujian secara non-fungsional akan dilakukan pada tempat yang mendukung modul komunikasi *LoRa* dan sistem yang dibuat. Tempat yang digunakan sebagai pengujian sistem berada di lahan pertanian seluas kurang lebih 1000 meter tanpa hambatan dan tidak padat lingkungan penduduk. Pengujian kinerja dilakukan berdasarkan parameter tingkat keberhasilan *gateway* dapat menerima paket dari node sensor yang kemudian diteruskan ke server yang ditentukan dengan pengaruh jarak, besar paket yang dikirim, dan interval waktu yang digunakan saat mengirim paket.

## 1.7 Penarikan Kesimpulan dan Saran

Penarikan keputusan dilakukan setelah melihat analisis pada hasil pengujian yang dilakukan. Dari hasil analisis maka akan didapatkan jawaban untuk disimpulkan dan menjadi jawaban atas rumusan masalah yang didefinisikan diawal penelitian. Dan diakhir penulisan terdapat saran yang juga di butuhkan untuk mengembangkan sistem ini kedepannya.

## BAB 4 REKAYASA KEBUTUHAN

Bab rekayasa kebutuhan ini menjelaskan tujuan dan fitur dari sistem. Analisis kebutuhan dibutuhkan untuk mencari tahu kebutuhan apa saja yang dibutuhkan agar sistem dapat berjalan sebagaimana mestinya. Analisis kebutuhan juga berfungsi mengetahui hal apa saja yang dibutuhkan dalam membuat *gateway* sebagai perantara komunikasi dari node sensor ke server. Dalam membuat sistem yang kompleks, mendefinisikan kebutuhan sistem terlebih dahulu dapat mempermudah untuk melakukan tahap perancangan, implementasi dan pengujian sistem. Berikut merupakan analisis kebutuhan pada penelitian ini sebagai berikut:

### 4.1 Deskripsi Umum

#### 4.1.1 Sistematika

Dalam mempermudah meninjau pembahasan, maka diperlukan sistematika yang digunakan sebagai pedoman penulisan *Software Requirement Specification* (SRS) ini. Bagian pertama dalam SRS terdapat deskripsi umum, yang akan menjelaskan tentang pandangan sistem dan untuk apa sistem tersebut dibuat. Pada deskripsi umum juga terdapat beberapa istilah yang akan sering digunakan dalam penyusunan penelitian ini yang nantinya akan membantu pembaca dalam memahami sistem yang dibuat. Lalu terdapat penjelasan kegunaan dan karakteristik agar pembaca memahami kegunaan sistem dan karakteristik sistem yang penulis buat. Batasan sistem juga merupakan hal yang akan dijelaskan pada deskripsi umum agar pembaca memahami batasan-batasan yang terdapat pada sistem.

Bagian terakhir terdapat spesifikasi kebutuhan sistem untuk mendeskripsikan kebutuhan fungsional dan non-fungsional sistem. Kebutuhan fungsional merupakan kebutuhan yang harus terpenuhi agar sistem dapat berjalan sesuai dengan tujuan awal sistem dibuat. Sedangkan, kebutuhan non-fungsional merupakan kebutuhan tambahan untuk mengukur kinerja sistem agar dapat berjalan secara optimal.

#### 4.1.2 Perspektif Sistem

Komponen utama penyusun sistem ini ada tiga yaitu node sensor, *gateway*, dan server. node sensor akan mengambil data dari sensor kemudian mengirimkan data tersebut ke *gateway*. Komunikasi antara node sensor dan *gateway* di lakukan oleh modul *transceiver LoRa HopeRF-RFM9x* dengan menggunakan frekuensi 433MHz. *Gateway* akan menerima data dari node sensor kemudian data tersebut dikirimkan ke server dengan mekanisme komunikasi *publish-subscribe*. Sebelum data sampai ke server, data akan dilewatkan ke *broker* dan *broker* akan meneruskan ke server sesuai dengan topik yang di *subscribe*. Sistem yang dibangun berfokus pada *gateway* yang akan menjadi perantara komunikasi antara node sensor dengan server.

### 4.1.3 Istilah

Tabel 4.1 Tabel Istilah

Istilah	Definisi
<i>Gateway</i>	Merupakan komponen yang akan menerima data dari node sensor dan server.
Node sensor	Merupakan sebuah node dalam jaringan sensor yang mampu melakukan beberapa pemrosesan dan mampu berkomunikasi dengan node lain yang terhubung dalam jaringan.
<i>LoRa (Long Range)</i>	<i>Long Range</i> merupakan teknologi nirkabel yang dikembangkan <i>Semtech</i> yang memungkinkan komunikasi jarak yang jauh dengan <i>data rate</i> rendah
<i>Wireless Sensor Networks (WSN)</i>	<i>Wireless Sensor Networks (WSN)</i> merupakan jaringan nirkabel yang terdiri dari perangkat otonom yang didistribusikan secara spasial dan menggunakan sensor untuk memantau lingkungan sekitar.
<i>Publisher</i>	Merupakan sebuah metode komunikasi <i>Publish-Subscribe</i> yang memiliki tugas untuk mengirim pesan ke <i>Broker</i> .
<i>Broker</i>	Merupakan sebuah bagian dari metode komunikasi <i>Publish-Subscribe</i> yang memiliki tugas menerima pesan dari <i>Publisher</i> dan mengirikan ke <i>Subscriber</i> .
<i>Subscriber</i>	Merupakan sebuah bagian dari metode komunikasi <i>Publish-Subscriber</i> yang memiliki tugas menerima pesan dari <i>Broker</i> .
<i>Software Requirements Specification (SRS)</i>	Sebuah dokumen yang menggambarkan sebuah fungsi dari sistem yang diusulkan dan bagaimana kebutuhan agar sistem dapat beroperasi.

### 4.1.4 Kegunaan Sistem

Dengan adanya sistem ini diharapkan pemakaian *gateway* dan teknologi *LoRa* maupun perangkatnya seperti modul *LoRa HopeRF-RFM9x* dengan frekuensi 433 MHz dapat lebih efektif. Dengan fitur *gateway* mampu menjadi jembatan komunikasi antara node sensor dan server dalam sebuah arsitektur *Wireless Sensor Network (WSN)* hal ini tentunya akan menambah efektifitas kerja sistem. Serta teknologi *LoRa* yang memiliki ketahanan dari segi daya dan dapat mengirimkan data dengan jarak tempuh yang jauh dengan lingkungan dan perangkat yang sesuai, hal ini tentunya akan menambah keuntungan sendiri bagi pengguna. Para pengguna dengan dengan pengetahuan yang minim diharapkan dapat menggunakan modul *LoRa HopeRF-RFM9x* dengan sistem ini dengan lebih mudah.

Pengguna dalam sistem ini bertugas menjalankan node sensor, *gateway*, dan juga server dengan aliran daya untuk menjalankan ketiga hal tersebut. Pada node sensor dan *gateway* kode program akan diunggah ke mikrokomputer Raspberry Pi dengan menggunakan bahasa *Python*. Server menjalankan kode program dengan bahasa *Python* dan bertugas menerima data dari node sensor kemudian menampilkan ke monitor. Karena itu pengguna hanya bertindak menjalankan program dan menunggu tampilan data dari node sensor pada sisi server melalui monitor.

#### 4.1.5 Batasan Perancangan dan Implementasi

Dalam menyusun sistem terdapat batasan-batasan yang akan diterapkan guna dalam melakukan perancangan dan implementasi sistem peneliti dapat berfokus pada sistem yang dibuat. Pembuatan sistem pada node sensor dan *gateway* menggunakan perangkat keras berupa Raspberry Pi yang dihubungkan dengan laptop dengan kabel *micro usb* sebagai daya agar perangkat dapat berjalan sesuai tujuan awal sistem dibuat. Komunikasi node sensor dan *gateway* menggunakan modul *transceiver LoRa HopeRF-RFM9x* dengan frekuensi 433 MHz.

Pengiriman data pada node sensor dengan *gateway* akan menggunakan *struct*. Data yang didapat dari sensor akan di *pack* ke dalam *struct* kemudian data akan dikirim ke *gateway* dan selanjutnya *gateway* akan melakukan *unpack* data yang diterima dari node sensor dan menampilkannya. Data yang berhasil diterima oleh *gateway* akan dilakukan *pack* ke format data *json* untuk pertukaran antara *gateway* dengan server. Data dari *gateway* dalam bentuk *json* akan di *publish* ke server dengan perantara *broker*. *Broker* kemudian akan meneruskan data tersebut ke server yang telah melakukan *subscribe*. Data yang berhasil diterima server akan ditampilkan pada monitor server.

Penggunaan jenis data yang dipertukarkan merupakan bagian dari ketergantungan sistem agar sistem dapat berjalan seperti tujuan awal dibuatnya. Serta penggunaan modul *LoRa HopeRF-RFM9x* juga memiliki ketergantungan seperti faktor lingkungan yang digunakan sebagai tempat pengujian, kepadatan lingkungan dan bangunan, jenis antena yang digunakan, dan tipe modul *LoRa* yang digunakan.

#### 4.2 Kebutuhan Fungsional

Kebutuhan fungsional sistem merupakan suatu kebutuhan yang harus terpenuhi agar sistem dapat berjalan sesuai tujuan awal. Adapun kebutuhan fungsional sistem yang peneliti buat sebagaimana Tabel 4.2.

**Tabel 4.2 Kebutuhan Fungsional Sistem**

No	Kebutuhan
1	Node sensor dapat mengambil data sensor.
2	Node sensor dapat mengirimkan data ke <i>gateway</i> .

3	<i>Gateway</i> dapat menerima data dari node sensor.
4	<i>Gateway</i> dapat meneruskan data dari node sensor ke server.
5	Server dapat menerima data dari <i>gateway</i> .

### 4.3 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan kebutuhan tambahan yang bertujuan untuk mengukur sistem dapat berjalan secara optimal. Pada penelitian ini kebutuhan non-fungsional digunakan untuk menguji kinerja dari *gateway* yang berdasarkan parameter tingkat keberhasilan *gateway* dalam menjadi perantara komunikasi antara node sensor dan server maupun sebaliknya, sehingga diharapkan pada pengujian non-fungsional ini dapat mendapatkan kinerja sistem yang optimal.

Pengujian kinerja *gateway* akan dilakukan pada area yang mendukung modul *LoRa*. Pengujian akan dilakukan di area persawahan dengan keadaan minim penduduk dan tidak terdapat banyak rumah penduduk. Perangkat akan diletakkan pada ketinggian kurang lebih 1,5 meter di atas permukaan tanah agar dapat lebih baik dalam hal penerimaan dan pengiriman data. Variabel pengiriman data yang akan digunakan untuk pengujian juga sudah ditentukan seperti interval, jarak antara node sensor dengan *gateway*, dan juga variasi besar paket yang akan dikirimkan.

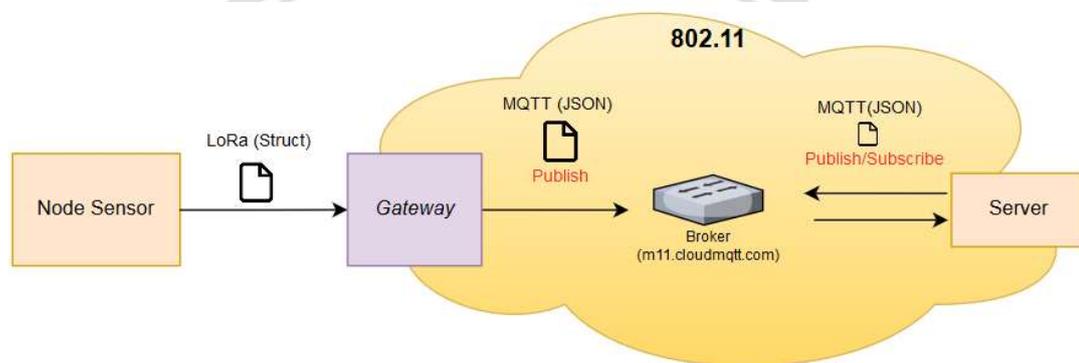
## BAB 5 PERANCANGAN SISTEM

Bab ini bertujuan menjelaskan perancangan sistem dan perancangan pengujian yang akan di lakukan. Rancangan yang dijabarkan seperti perancangan arsitektur jaringan, alur kerja sistem, serta perancangan pengujian.

### 5.1 Perancangan

#### 5.1.1 Gambaran Umum Sistem

Pada tahap perancangan, sistem harus memenuhi tujuan penelitian yaitu *gateway* dapat menerima data dari node sensor dan kemudian meneruskannya ke server. Mekanisme dalam pertukaran data ini akan digambarkan dalam perancangan sistem sebagaimana dapat dilihat pada gambar 5.1.



**Gambar 5.1 Gambaran Umum Sistem**

Pada gambar 3.2 di bab metodologi telah dijelaskan gambaran sistem secara umum yang akan dirancangan. Sistem tersebut merepresentasi gambaran umum sistem melalui perangkat keras apa saja yang dibutuhkan pada komponen utama sistem seperti node sensor yang menggunakan beberapa sensor dan sebuah mikrokontroler Raspberry Pi serta modul *LoRa* sebagai *transceiver*. Disisi *gateway* terdapat mikrokomputer Raspberry Pi yang terhubung modul *LoRa* sebagai *transceiver* dan terdapat server yang akan menampilkan data yang diterima. Pada Gambar 5.1 ini akan dijelaskan lebih detail mengenai pertukaran data dan bagaimana sistem berkerja.

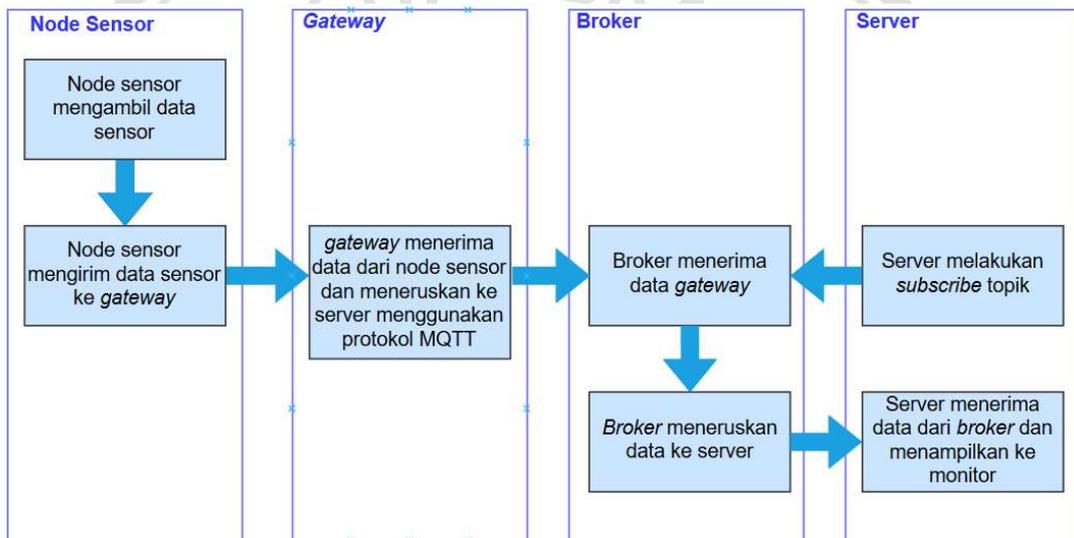
Kompenen pertama yaitu node sensor yang akan mengambil data sensor dan mengirimkan data tersebut ke *gateway*. Data sensor yang dikirim di *pack* menggunakan *struct* dengan tipe data *Integer* dan *String*. Untuk melakukan *packing* ke *struct* dalam kode program harus menambahkan *library struct* agar kode program dapat berjalan sebagaimana mestinya. Setelah data di *packing*, data dikirim ke *gateway* menggunakan modul *LoRa HopeRF-RFM9x* dengan frekuensi 433 MHz yang telah terhubung pada node sensor.

Komponen kedua yaitu *gateway* yang bertugas menjadi perantara pengiriman data baik dari node sensor dan server. Pada paragraf sebelumnya node sensor mengirim data sensor yang di *pack* menggunakan *struct* dan dikirm ke

*gateway* menggunakan *transceiver* modul *LoRa*. Kemudian *gateway* menerima paket tersebut dan melakukan *unpack struct* agar data dapat ditampilkan dalam aslinya. Setelah data di *unpack*, data akan di *pack* kembali ke bentuk *json* untuk *publish* ke server menggunakan protokol MQTT. Komunikasi antara *gateway* dengan server menggunakan protokol 802.11. Data yang dikirim *gateway* akan dilewatkan melalui *broker* kemudian *broker* meneruskan ke server sesuai dengan topik yang di *subscribe* oleh server. Server menerima data dari node sensor dan menampilkan melalui monitor yang tersedia.

### 5.1.2 Perancangan Alur Kerja Sistem

Pada bagian ini akan dijabarkan *flowchart* gambaran dari alur kerja sistem keseluruhan yang peneliti buat. Alur kerja sistem terdiri dari node sensor, gateway, dan server. Namun sebelum membahas satu persatu alur kerja sistem akan diberikan gambaran umum dan cara kerja sistem secara umum. Untuk mempermudah penjelasan mengenai alur kerja sistem dapat dilihat pada gambar 5.2.



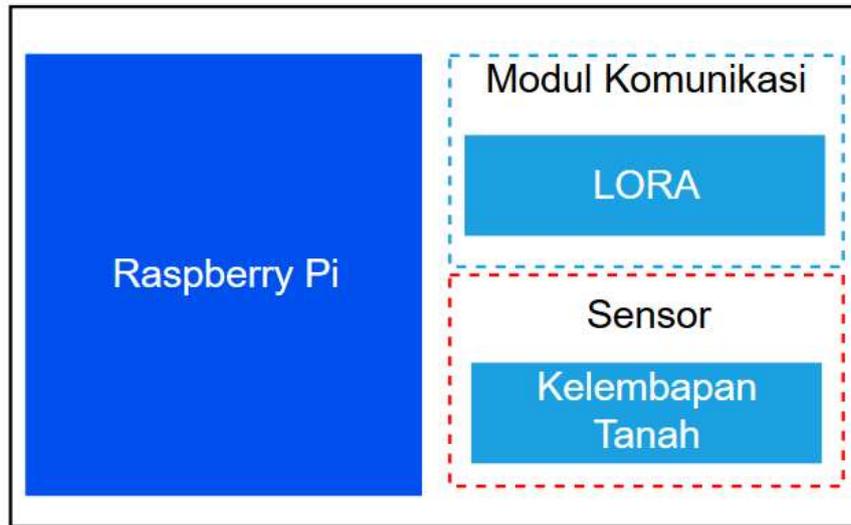
Gambar 5.2 Alur Kerja Sistem Secara Umum

Pada gambar 5.2 menunjukkan alur kerja sistem secara umum. Pertama kali yang harus dilakukan sistem adalah node sensor mengirimkan data berupa pengambilan data sensor yang di *pack* dengan menggunakan *struct* kemudian dikirim ke *gateway*. *Gateway* menerima data dari node sensor dengan format *struct*, yang kemudian data *struct* tersebut di *unpack* dan melakukan *pack* kembali ke dalam bentuk *json* agar dapat dikirim ke server menggunakan protokol MQTT. *Broker* menerima data dari *gateway* dan meneruskan data tersebut ke server sesuai dengan topik yang telah di *subscribe* oleh server dan data yang diterima di server akan ditampilkan di monitor.

### 5.1.3 Perancangan Node sensor

Pada tahap perancangan ini node sensor akan dihubungkan dengan beberapa sensor dan modul *LoRa* HopeRF-RFM9x sebagai *transceiver* dengan

gateway. Peran perancangan perangkat keras dalam membangun sistem ini sangatlah penting karena sistem berbentuk nyata bukan *virtual*. Perancangan tersebut dapat dilihat pada Gambar 5.3.



**Gambar 5.3 Arsitektur Node Sensor**

Pada gambar 5.3 merupakan perancangan arsitektur pada sisi node sensor, salah satunya menggunakan mikrokomputer Raspberry Pi sebagai komponen yang digunakan untuk menjalankan kode program dengan bahasa pemrograman *Python*. Raspberry Pi terhubung dengan beberapa sensor dan modul komunikasi *LoRa* sebagai *transceiver* yang dihubungkan dengan beberapa kabel *jumper female-female*. Kabel *jumper* digunakan untuk menyambungkan perangkat sensor, modul *LoRa*, dan juga Raspberry Pi agar kode program yang dibuat dapat berjalan sebagaimana mestinya. Komposisi sambungan antara perangkat tersebut disebut dengan *pinout*. *Pinout* dari beberapa perangkat tersebut dapat dilihat pada Tabel berikut:

**Tabel 5.1 Pinout Modul LoRa HopeRF-RFM9x dengan Raspberry Pi**

Modul LoRa HopeRF-RFM9x	Raspberry Pi
GND	GND [5]
DIO0	GPIO 25 [22]
3.3V	3.3V [1]
MISO	GPIO 9 [21]
MOSI	GPIO 10 [19]
SCK	GPIO11 [23]
NSS	GPIO 7 [26]
RESET	GPIO 22 [15]

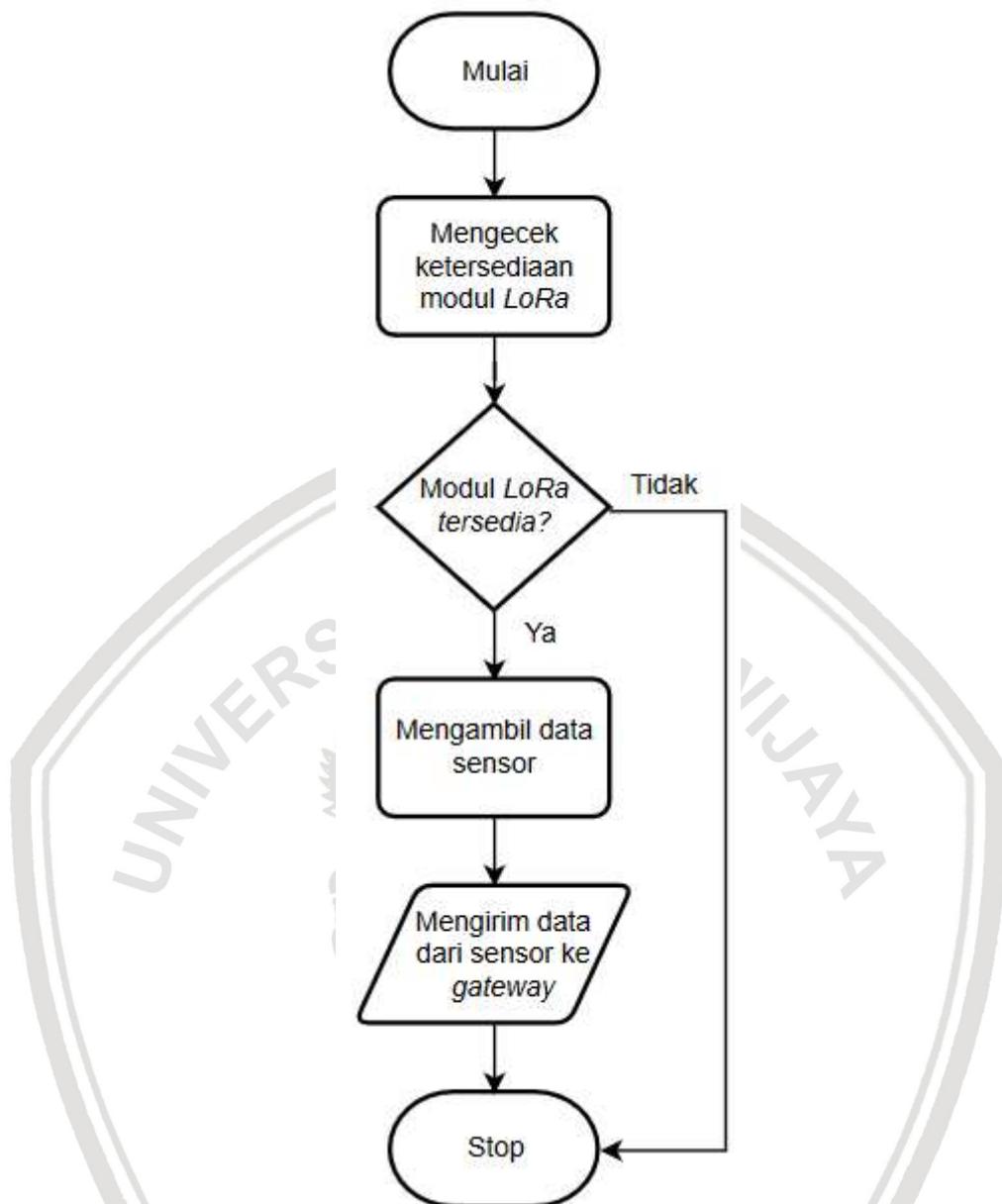
Tabel 5.1 merupakan konfigurasi dari pin modul LoRa HopeRF-RFM9x dan Raspberry Pi, terdapat 8 pin yang saling terhubung dan harus diperhatikan agar pin terhubung sesuai tabel diatas agar sistem dapat bekerja sebagaimana mestinya. Kedua perangkat ini dihubungkan dengan kabel *jumper female-female*. Pada Tabel 5.1 diatas modul lora menggunakan daya sebesar 3.3V yang di hubungkan dengan Raspberry Pi di pin ke 1. *Interrupt* (DIO0) modul *LoRa* dihubungkan pada GPIO 25 pin 22. *MISO* (*Master In Slave Out*) yang terhubung pada GPIO 9 pin 21 dan *MOSI* (*Master Out Slave In*) yang terhubung pada GPIO 10 pin 19 digunakan untuk pengiriman data pada modul *LoRa*.

**Tabel 5.2 Pinout Moisture Sensor dengan Raspberry Pi**

Moisure	Raspberry Pi
VCC	5V [2]
DATA (DIGITAL)	GPIO 21 [40]
GND	GND [39]

Tabel 5.2 atas merupakan konfigurasi *pinout moisture sensor* dengan Raspberry Pi, *output* yang dihasilkan oleh *moisure sensor* adalah data digital. Konfigurasi pin antara *moisure sensor* dengan Raspberry Pi yaitu menggunakan pin untuk data pada *moisure sensor* dengan pin GPIO 4 untuk Raspberry Pi. Untuk tegangan (VCC) yang digunakan oleh DHT11 adalah 5V.

Setelah semua perangkat keras terpasang dengan rancangan *pinout* yang sesuai, maka node sensor akan ditanamkan kode program yang sudah dibuat menggunakan *text editor* dengan bahasa *Python* ke dalam Raspberry Pi. Pada pemrograman yang akan dibuat membutuhkan beberapa *library* yang harus ditambahkan untuk memudahkan menjalankan fungsi sistem. *Library* pertama yang harus ditambahkan yaitu *rf95*. *Library* ini merupakan *library* yang akan digunakan untuk memudahkan menjalankan modul *LoRa* HopeRF-RFM9x yang terpasang pada Raspberry Pi sebagai *transceiver*. Kedua yaitu *library struct* yang akan digunakan sebagai pertukaran data untuk mengirim ke *gateway* dengan format yang telah disepakati. Setelah *library* berhasil ditambahkan ke kode program, maka persiapan untuk membuat kode program pada sisi node sensor dapat dilakukan. Perancangan alur kerja sensor node juga harus dilakukan agar diketahui cara kerja dari sudut node sensor. Perancangan alur kerja node sensor dapat dilihat pada diagram *flowchart* yang ditunjukkan pada Gambar 5.4.

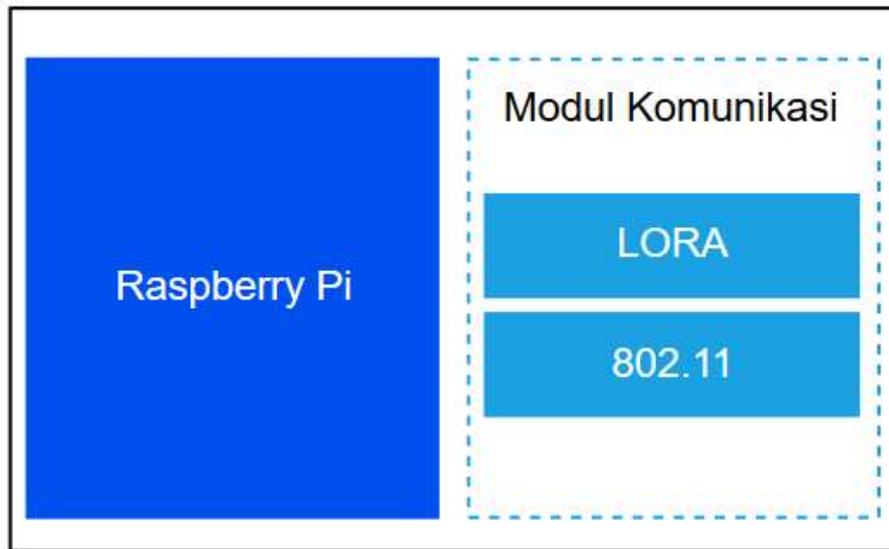


**Gambar 5.4 Alur Kerja Sistem Node Sensor**

Pada gambar 5.4 node sensor akan mengecek ketersediaan modul *LoRa* terlebih dahulu karena untuk mengirim data ke *gateway* menggunakan modul *transceiver LoRa* dengan gelombang radio, maka harus terdapat pengecekan terhadap modul tersebut tersedia atau tidak. Apabila proses pengecekan *LoRa* tersedia node sensor akan mengambil data dari sensor yang telah dihubungkan dengan kabel *jumper* dengan format pengiriman yang telah disepakati yaitu menggunakan *struct* pada data yang dikirim. Data dari node sensor yang di *pack* dalam bentuk *struct* akan dikirim ke *gateway* menggunakan modul *LoRa HopeRF-RFM9x* dengan frekuensi 433 MHz. Kemudian alur kerja dari node sensor berakhir dengan dikirimkannya data dari node sensor ke *gateway*.

### 5.1.4 Perancangan Gateway

Pada penelitian ini *gateway* berperan sangat penting karena bertugas menjadi perantara pengiriman data dari node sensor hingga data sampai ke *gateway*, maka dari itu *gateway* perlu dirancang dengan baik sebagaimana pada perancangan node sensor agar dapat berjalan sesuai dengan tujuan sistem. Perancangan tersebut dapat dilihat pada Gambar 5.4.



Gambar 5.5 Arsitektur Gateway

Pada Gambar 5.4 peran perancangan arsitektur sistem sangatlah penting agar sistem dapat berjalan semestinya sesuai tujuan awal dibuatnya. Dalam arsitektur sistem pada Gambar 5.5 dapat dilihat terdapat mikrokomputer Raspberry Pi yang digunakan untuk menjalankan kode program dengan bahasa *Python*. Mikrokomputer ini terhubung dengan modul *LoRa* HopeRF-RFM9x sebagai *transceiver* komunikasi dengan node sensor. Modul komunikasi pada *gateway* menggunakan jaringan komunikasi 802.11 dan modul *LoRa* HopeRF-RFM9x dengan frekuensi 433 MHz yang dihubungkan dengan Raspberry Pi melalui kabel *jumper* yang bisa dilihat pada Tabel 5.4.

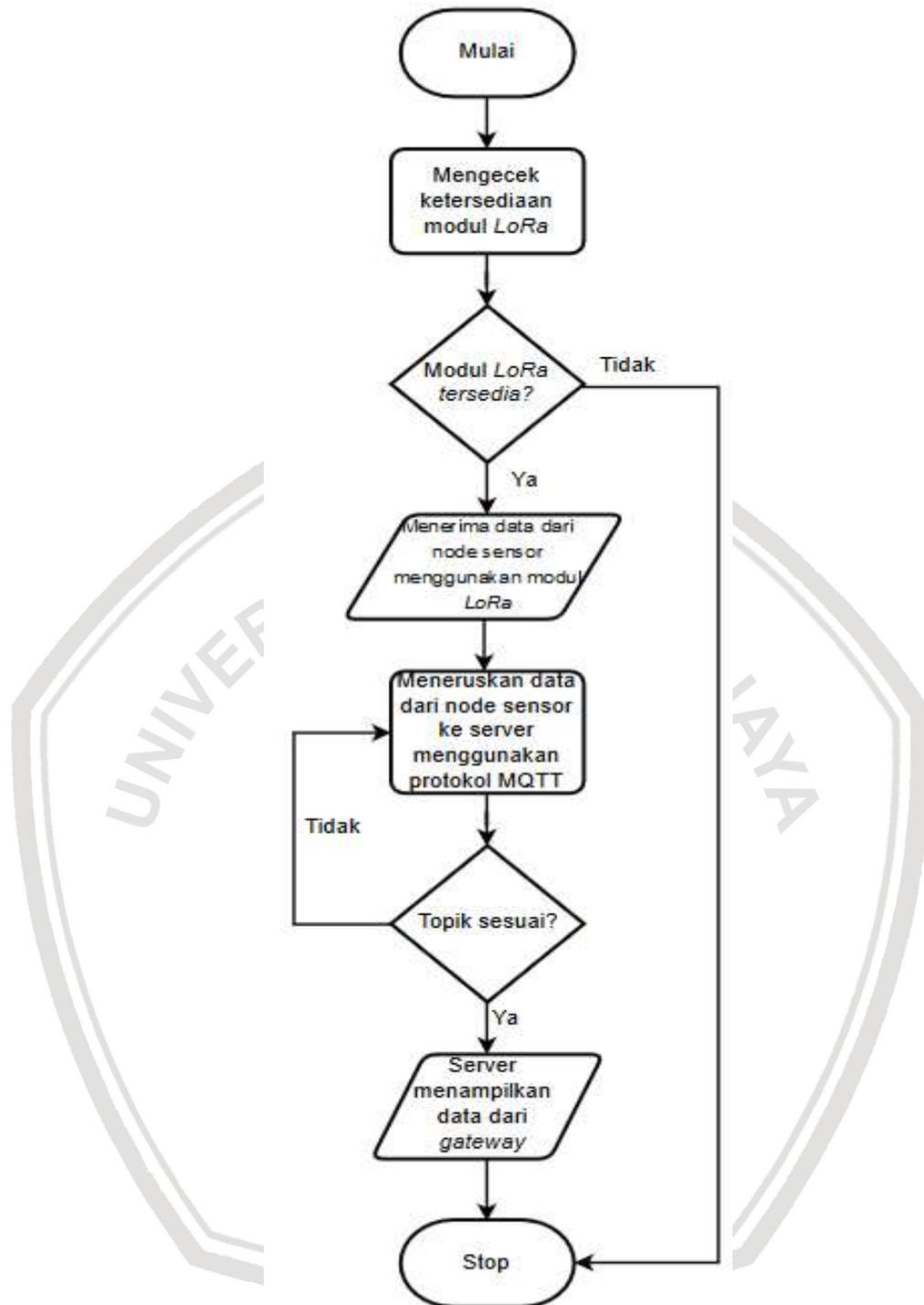
Tabel 5.4 Pinout Modul LoRa HopeRF-RFM9x dengan Raspberry Pi

Modul LoRa HopeRF-RFM9x	Raspberry Pi
GND	GND [5]
DIO0	GPIO 25 [22]
3.3V	3.3V [1]
MISO	GPIO 9 [21]
MOSI	GPIO 10 [19]
SCK	GPIO11 [23]

NSS	GPIO 7 [26]
RESET	GPIO 22 [15]

Tabel 5.4 diatas merupakan *pinout* modul LoRa HopeRF-RFM9x dan Raspberry Pi, terdapat 8 pin yang saling terhubung adan harus diperhatikan, agar *pinout* terhubung sesuai tabel diatas agar sistem dapat bekerja sebagaimana mestinya. Kedua perangkat ini dihubungkan dengan kabel *jumper female-female*. Pada Tabel 5.4 diatas modul *LoRa* menggunakan daya sebesar 3.3V yang di hubungkan dengan Raspberry Pi di pin ke 1. *Interrupt* (DIO0) modul *LoRa* dihubungkan pada GPIO 25 pin 22. *MISO* (*Master In Slave Out*) yang terhubung pada GPIO 9 pin 21 dan *MOSI* (*Master Out Slave In*) yang terhubung pada GPIO 10 pin 19 digunakan untuk pengiriman data pada modul *LoRa*.

Setelah perangkat keras terpasang pada gateway, maka *gateway* akan ditanamkan kode program dengan bahasa *Python* ke dalam mikrokomputer Raspberry Pi. Pada kode program yang akan tanamkan membutuhkan beberapa *library* yang harus ditambahkan untuk memudahkan menjalankan fungsi sistem. *Library* pertama yang harus ditambahkan yaitu *rf95*. *Library* ini merupakan *library* yang akan digunakan menjalankan modul *tranceiver LoRa HopeRF-RFM9x* yang tersang pada Raspberry pi sebagai komunikasi dengan node sensor. Kedua yaitu *library struct* dan *library json* yang akan digunakan sebagai format pertukaran data dengan node sensor dan server. *Library* selanjutnya yang harus ditambahkan yaitu *library paho-mqtt* dan *json* yang akan digunakan untuk pertukaran data antara *gateway* dengan server. Karena perbedaan protokol komunikasi yang digunakan antara node sensor dengan *gateway* dan *gateway* dengan server, maka peran *gateway* sangatlah penting untuk data sampai ketujuan. Untuk dapat memahami alur kerja gateway ini maka dibuatlah diagram *flowchart* yang ditunjukkan pada Gambar 4.4.



**Gambar 5.5 Alur Kerja Sistem Gateway**

Pada Gambar 5.5 tahap awal alur *flowchart* yang ditunjukkan *gateway* akan mengecek ketersediaan modul *LoRa* HoperRF-RFM9x pada Raspberry Pi. Apabila modul *LoRa* tidak tersedia, program akan langsung berhenti dan jika modul *LoRa* tersedia, maka *gateway* akan menerima data dari node sensor yang kemudian akan meneruskan data tersebut ke server dengan topik yang sesuai. Topik yang tidak sesuai akan didrop dan program akan mengirimkan pesan selanjutnya. Setelah data

diterima server, data dari node sensor yang dikirim melalui *gateway* akan ditampilkan pada sisi server menggunakan monitor yang tersedia.

### 5.1.5 Perancangan Pengujian

Perancangan pengujian perlu dilakukan untuk menentukan parameter keberhasilan penelitian sesuai skenario yang telah di tentukan. Skenario tersebut harus dilakukan untuk mengetahui hasil dari penelitian ini.

#### 5.1.5.1 Pengujian Fungsional

Pengujian fungsional sistem difokuskan agar sistem dapat berjalan sesuai tujuan awal dibuatnya. Pengujian ini dilakukan berdasarkan paparan dalam subbab analisis kebutuhan. Sesuai dengan subbab yang telah didefinisikan, pengujian akan dilakukan dengan interaksi antara tiga komponen yaitu node sensor, *gateway*, dan server. pengujian ini dilakukan di tempat yang telah ditentukan oleh peneliti, agar modul *LoRa* dapat berjalan lebih baik tanpa banyak gangguan pada perangkat. Penentuan tempat yang dimaksudkan yaitu tempat yang tidak memiliki banyak penduduk dan bangunan seperti pada lahan pertanian di lingkungan pedesaan. Jika interaksi yang terjadi sesuai perancangan pengujian, maka pengujian fungsional dianggap berhasil. Berikut adalah tabel perancangan skenario pengujian fungsional yang dapat dilihat pada Tabel 5.5.

**Tabel 5.5 Perancangan Skenario Pengujian Fungsional**

No	Kode	Fungsi	Skenario Pengujian
1	PL-01	Node sensor mampu mengambil data sensor.	<ol style="list-style-type: none"> <li>1. Node sensor dalam keadaan aktif.</li> <li>2. Terdapat sensor yang terhubung pada node sensor.</li> <li>3. Mengambil data sensor yang terpasang pada node sensor.</li> <li>4. Menampilkan data keluaran dari sensor</li> </ol>
2	PL-02	Node sensor dapat mengirimkan data ke gateway.	<ol style="list-style-type: none"> <li>1. Node sensor dalam keadaan aktif.</li> <li>2. Modul <i>LoRa</i> HopeRF-RFM9x dalam keadaan aktif</li> <li>3. Node sensor telah mengambil data dari sensor.</li> <li>4. Node sensor mengirim data ke gateway.</li> </ol>
3	PL-03	Gateway dapat menerima data dari node sensor.	<ol style="list-style-type: none"> <li>1. Gateway dalam keadaan aktif.</li> <li>2. Modul <i>LoRa</i> HopeRF-RFM9x dalam keadaan aktif.</li> <li>3. <i>Gateway</i> menampilkan data yang diterima dari node sensor.</li> </ol>
4	PL-04	Gateway dapat meneruskan data	<ol style="list-style-type: none"> <li>1. Gateway dalam keadaan aktif.</li> <li>2. Modul <i>LoRa</i> HopeRF-RFM9x dalam keadaan aktif.</li> </ol>

		dari node sensor ke server.	<ol style="list-style-type: none"> <li>3. <i>Gateway</i> telah menerima data dari node sensor.</li> <li>4. <i>Gateway</i> menulis topik untuk <i>publish</i> pesan.</li> <li>5. <i>Gateway</i> meneruskan data dari node sensor ke server.</li> </ol>
5	PL-05	Server dapat menerima data dari <i>gateway</i> .	<ol style="list-style-type: none"> <li>1. Server dalam keadaan aktif</li> <li>2. Server melakukan <i>subscribe</i> topik yang telah ditentukan.</li> <li>3. Server menerima data dari <i>gateway</i> sesuai topik.</li> </ol>

### 5.1.5.2 Pengujian Kinerja

Pengujian kinerja dalam sistem ini berfokus pada parameter *sucessfull rate* pada *gateway*. Paramater tersebut diuji dengan variasi jarak 200 meter dan 400 meter. Pemilihan jarak untuk pengujian yang dilakukan karena kemungkinan perubahan suhu kelembapan tanah yang signifikan di jarak 50 meter sampai 100 meter sangatlah jarang, maka dari itu peneliti memilih menggunakan jarak 200 meter dan 400 meter. Pengiriman data dengan menggunakan modul *LoRa* memiliki batas maksimum pengiriman data sebesar 255 *bytes*, maka dari itu variasi besar paket yang ditentukan untuk pengujian 44 *Byte*, 56 *Byte*, 80 *Byte*, 128 *Byte*. Pemilihan besar paket didasari dari fungsionalitas sistem, yang kurang lebih untuk mengirimkan data sensor kelembapan tanah membutuhkan 12 *Byte*, namun untuk pengujian kinerja harus ditambahkan 32 *Byte* sebagai *payload* untuk kebutuhan mengukur kinerja sistem.

Pengujian ini juga menggunakan interval pengiriman data yang berbeda yaitu 0,5 detik, 1 detik, dan 5 detik. Pemilihan interval waktu yang digunakan untuk mengetahui kinerja sistem jika data dikirim dengan interval waktu yang berbeda. Pengujian sistem akan diukur dengan 100 paket yang dikirim dengan *generate data* yang telah ditentukan pada masing-masing variasi jarak, besar data, dan interval. Pada penjelasan tersebut, pegujian kinerja akan dilakukan dengan menggunakan parameter *successful rate*

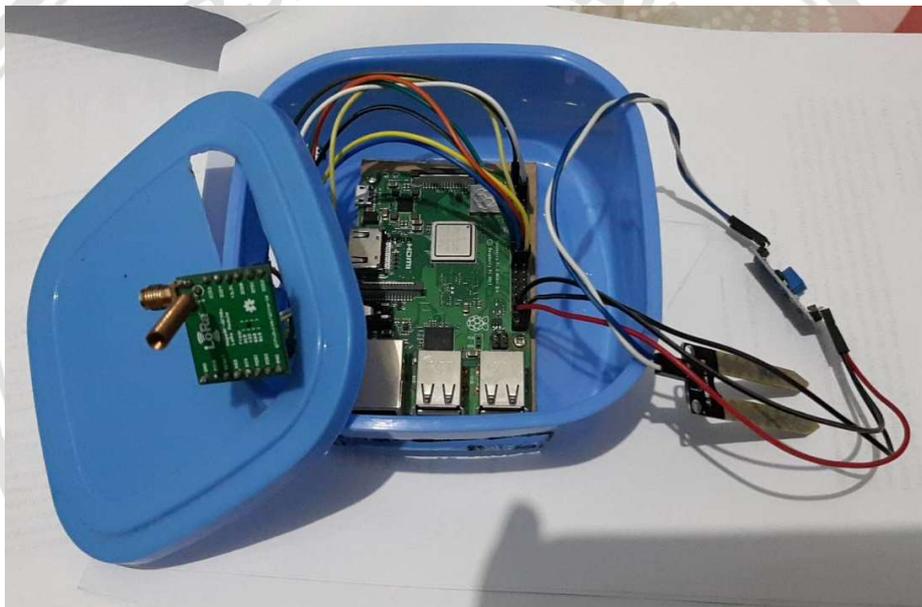
Pengujian *Successful Rate* akan menguji tingkat keberhasilan *gateway* dalam menerima data dari node sensor yang kemudian diteruskan ke server. Dalam pengujian sistem ini *successful rate* dapat mencakup beberapa faktor seperti kondisi lingkungan pengujian, kondisi perangkat yang digunakan, dan juga kekuatan *signal* yang terdapat pada modul *transceiver LoRa* dan jaringan lokal yang digunakan.

## BAB 6 IMPLEMENTASI SISTEM

Pada bab ini akan dijelaskan implementasi sistem untuk membentuk sistem perbagian secara lengkap. Proses implementasi pada bab ini dilakukan berdasarkan rancangan sistem pada subbab sebelumnya yaitu perancangan sistem. Implementasi yang dilakukan mencakup implementasi pada sisi node sensor dan *gateway*.

### 6.1 Implementasi Node Sensor

Implementasi node sensor dalam sistem ini melibatkan tiga buah perangkat keras yaitu mikrokomputer Raspberry Pi, sensor dan modul *LoRa* HopeRF-RFM9x sebagai *transceiver* dengan *gateway*. Perangkat keras sensor, dan modul *LoRa* yang telah disiapkan, di hubungkan dengan kabel *jumper* ke mikrokomputer Raspberry Pi sesuai perancangan yang telah disepakati. Setelah perangkat keras saling terhubung selanjutnya memberi daya pada Raspberry Pi agar dapat diaktifkan. Raspberry Pi dapat disambungkan ke Laptop untuk mendapatkan sumber daya listrik dengan kabel *micro usb* seperti yang dapat dilihat pada Gambar 6.1.



**Gambar 6.1 Implementasi Perangkat Keras Node Sensor**

Setelah perangkat penyusun node sensor berhasil diimplementasikan. Selanjutnya adalah menambahkan *library* yang dibutuhkan untuk memudahkan menjalankan kode program. Seperti yang dijelaskan pada bab perancangan, terdapat dua *library* yang akan digunakan dalam implementasi node sensor yang dapat dilihat pada Tabel 6.1

**Tabel 6.1 Library Pada Node Sensor**

1	<code>import struct</code>
2	<code>import rf95</code>

*Library* `rf95` digunakan untuk memudahkan peneliti menjalankan fungsi perangkat modul *LoRa* sebagai *transceiver* seperti frekuensi yang digunakan

perangkat *LoRa* dan pengecekan ketersediaan modul *LoRa*. Selanjutnya *library struct* digunakan untuk memudahkan peneliti untuk melakukan *pack* data yang akan dikirim menggunakan modul *LoRa* dalam bentuk *struct* data. Setelah menyiapkan *library* yang dibutuhkan, pembuatan utama kode program dapat dilakukan. Penerapan kode program node sensor dapat dilihat pada Tabel 6.2.

**Tabel 6.2 Pseudocode Node Sensor**

Import <i>library</i> Inisialisasi <i>library</i> rf95 dan inisialisasi <i>pinout</i> Mengecek ketersediaan modul <i>LoRa</i> Melakukan inisialisasi frekuensi modul <i>LoRa</i> Inisialisasi GPIO untuk sensor Membuat fungsi <i>sensor_node</i> Inisialisasi GPIO sensor Jika GPIO input <i>low</i> , maka Menampilkan hasil data Selain dari kondisi yang disebutkan, Menampilkan hasil data Jika panjang data sama dengan 6, maka Melakukan <i>pack</i> data Jika panjang data sama dengan 12, maka Melakukan <i>pack</i> data Mengirimkan data menggunakan modul <i>LoRa</i> Menampilkan data yang dikirim Menjalankan fungsi <i>sensor_node</i>
--

Pada Tabel 6.2 berisi kode program node sensor dalam *pseudocode* untuk menjalankan fungsi sistem pada node sensor. Kode program diawali dengan melakukan import *library* yang dibutuhkan dalam menulis kode program, lalu melakukan inisialisasi *library* rf95 dan inisialisasi *pinout* pada modul *LoRa*. Selanjutnya, kode program tersebut mengecek ketersediaan modul *LoRa*, apabila modul *LoRa* tersedia maka program akan berjalan, namun apabila modul *LoRa* tidak tersedia program akan langsung dihentikan. Selanjutnya, melakukan inisialisasi GPIO pada sensor yang terhubung dengan mikrokomputer Raspberry Pi agar sensor yang terhubung dengan Raspberry Pi dapat berfungsi sebagaimana dengan yang ditulis pada kode program. Selanjutnya dibuat fungsi *sensor\_node* sebagai fungsi utama pada node sensor yang digunakan untuk melakukan pengambilan data dari sensor berupa input digital dari sensor yang kemudian akan dikirim menggunakan modul *LoRa* dengan format data *struct* yang telah disepakati sebelumnya.

## 6.2 Implementasi Gateway

Implementasi pada *gateway* melibatkan dua buah perangkat keras yaitu mikrokomputer Raspberry Pi, dan modul *LoRa* HoperF-RFM9x sebagai *transceiver* dengan node sensor. Modul *LoRa* yang telah disiapkan di hubungkan dengan kabel *jumper* ke mikrokomputer Raspberry Pi sesuai perancangan yang telah disepakati. Setelah perangkat keras saling terhubung selanjutnya memberi daya pada

Raspberry Pi agar dapat diaktifkan. Raspberry Pi dapat disambungkan ke Laptop untuk mendapatkan sumber daya listrik dengan kabel *micro* usb seperti yang dapat dilihat pada Gambar 6.2.



**Gambar 6.2 Implementasi Perangkat Keras Gateway**

Setelah perangkat keras penyusun *gateway* berhasil diimplemetasikan. Selanjutnya adalah menambahkan *library* yang dibutuhkan untuk memudahkan menjalankan kode program. Seperti yang dijelaskan pada bab perancangan, terdapat empat *library* utama yang akan digunakan dalam implementasi node sensor yang dapat dilihat pada Tabel 6.3

**Tabel 6.3 Library Pada Node Sensor**

1	<code>import rf95</code>
2	<code>import struct</code>
3	<code>import json</code>
4	<code>import paho.mqtt.client as mqtt</code>

*Library* *rf95* digunakan untuk memudahkan dalam membuat kode program dan menjalankan fungsi perangkat pada modul *LoRa* seperti frekuensi yang digunakan perangkat *LoRa* dan pengecekan ketersediaan modul *LoRa*. Selanjutnya *library* *struct* untuk memudahkan peneliti untuk melakukan *pack* data yang akan dikirim menggunakan modul *LoRa* dalam bentuk *struct* data. pada pengiriman *gateway* ke server dibutuhkan *import library json* sebagai format pertukaran data dan *library paho-mqtt* sebagai mekanisme untuk menggunakan metode *publish-subscribe*. Sesudah menyiapkan *library* yang dibutuhkan, pembuatan utama kode program dapat dilakukan. Penerapan kode program *gateway* dapat dilihat pada Tabel 5.4.

**Tabel 6.4 Pseudocode Gateway**

<p>Import <i>library</i>          Inisialisasi <i>library</i> <i>rf95</i> dan inisialisasi <i>pinout</i>          Mengecek ketersediaan modul <i>LoRa</i>          Melakukan inisialisasi frekuensi modul <i>LoRa</i></p>
---

Membuat fungsi *lora\_gateway* sebagai *main function*  
 Melakukan inisialisasi untuk *broker*  
 Menjalankan fungsi *LoRa receive*  
 Jika panjang data sama dengan 10, maka  
     Melakukan *unpack* data  
 Jika panjang data sama dengan 16, maka  
     Melakukan *unpack* data  
 Selain dari kondisi yang disebutkan,  
     Beri peringatan bahwa data tidak valid  
 Melakukan *pack* data ke *json*  
 Menampilkan data yang diterima  
 Meneruskan data yang diterima ke server

Pada Tabel 6.4 berisi kode program *gateway* dalam *pseudocode* untuk menjalankan fungsi sistem *gateway*. Kode program diawali dengan melakukan *import library* yang diperlukan dalam menulis kode program agar sistem dapat berjalan dengan semestinya. Selanjutnya, melakukan inisialisasi *library rf95*, apabila modul *LoRa* tersedia maka program akan berjalan, namun apabila modul *LoRa* tidak tersedia program akan langsung dihentikan. Inisialisasi *pinout* pada modul *LoRa* dibutuhkan agar sensor dan modul *LoRa HopeRF-RFM9x* dapat berjalan sesuai kode program yang diimplementasikan. Selanjutnya, melakukan inisialisasi frekuensi pada modul *LoRa* agar *gateway* dapat berkomunikasi satu sama lain dengan node sensor.

Selanjutnya yaitu membuat fungsi *lora\_gateway* yang merupakan *main function* yang akan menjalankan *library* yang telah didefinisikan sebelumnya. Pertama dalam fungsi ini akan melakukan inisialisasi *broker* dengan parameter *server, user, password, dan port* yang digunakan. *Broker* akan menjadi perantara pengiriman antara *gateway* dengan server. selanjutnya program akan menjalankan fungsi *LoRa receive* yang di panggil dari *library rf95* untuk menerima data dari node sensor. Data yang diterima dari node sensor dalam bentuk *struct* yang nanti akan di *unpack* dengan filter *length* data yang diterima. Data yang sudah dilakukan *unpacking* akan di *pack* kembali dalam bentuk *json* yang kemudian akan dikirim ke server dengan protokol MQTT.

Setelah implementasi *pseudocode gateway* tersedia, yang harus diimplementasikan selanjutnya yaitu pada server. Server bertugas menerima data dari *gateway* dan *user* memonitoring data yang didapat melalui monitor yang tersedia. Server juga dapat mengirim pesan ke node sensor untuk menjalankan perintah, maka dari itu dibutuhkannya kode program untuk menjalankan server. *library* yang digunakan hampir sama dengan *gateway* yang dapat dilihat pada Tabel 6.5.

**Tabel 6.5 Library Pada Server**

1	<code>import json</code>
2	<code>import paho.mqtt.client as mqtt</code>



Pada server tidak menggunakan modul *LoRa* karena komunikasi antara *gateway* dan server menggunakan protokol MQTT yang menggunakan mekanisme komunikasi *publish-subscribe*, maka digunakannya *library json* dan *library paho-mqtt* untuk memudahkan peneliti dalam menulis program agar sistem dapat berjalan seperti tujuan awal dibuatnya. Setelah menyiapkan *library* yang dibutuhkan, pembuatan kode program dapat dilakukan. Penerapan kode program *gateway* dapat dilihat pada Tabel 6.6.

**Tabel 6.6 Pseudocode Server**

Import <i>library</i> yang dibutuhkan Membuat fungsi <i>on_connect</i> untuk <i>subscribe</i> topik Membuat fungsi <i>on_messege</i> <ul style="list-style-type: none"> <li>    Menerima payload dari server sesuai topik</li> <li>    Melakukan unpack payload data</li> <li>    Menampilkan pesan dari <i>gateway</i></li> </ul> Membuat fungsi <i>subscribe</i> <ul style="list-style-type: none"> <li>    Melakukan inialisasi untuk <i>broker</i></li> <li>    Menjalankan fungsi <i>on_message</i> dan <i>on_connect</i></li> </ul>
---

Pada Tabel 6.6 berisi kode program server dalam *pseudocode* untuk menjalankan fungsi sistem. Kode program diawali dengan melakukan *import library* yang diperlukan dalam menulis kode program. Lalu dibuat fungsi *on\_connect* dan *on\_massege* untuk menjalankan mekanisme *publish-subscribe*. Fungsi *on\_connect* digunakan untuk *subscribe* topik yang akan dikirim oleh *gateway* dan fungsi *on\_messege* digunakan untuk menerima data dari *gateway* yang berupa format *json* dan kemudian ditampilkan dimonitor. Kemudian dibuat *main function* dengan nama *subscribe* untuk menjalankan fungsi-fungsi sebelumnya yang telah di definisikan. Pada fungsi *subscribe*, yang pertama melakukan inialisasi broker yang digunakan dengan paramater *server*, *user*, *password*, dan *port* sebagai perantara pengiriman antara *gateway* dengan server. Setelah melakukan inialisasi *broker* program akan menjalankan fungsi *on\_connect* dan *on\_message* yang telah didefinisikan sebelumnya.

## BAB 7 PENGUJIAN SISTEM

Bab ini akan menjelaskan tentang uji sistem berdasarkan implementasi dari sistem. Tahap ini dilakukan untuk mengetahui jalannya sistem apakah sudah sesuai dengan kebutuhan fungsional dan non-fungsional yang telah dirancang pada bab rekayasa kebutuhan

### 7.1 Pengujian Kebutuhan Fungsional

Pengujian fungsional termasuk pengujian yang dilakukan secara *black-box*. Pengujian *black-box* merupakan suatu pengujian yang dilakukan hanya untuk mengamati hasil akhir dari eksekusi program tersebut. Pengamatan hasil ini melalui hasil uji fungsional dari sistem itu sendiri. Pada pengujian ini, hasil akhir yang didapatkan diharapkan memenuhi fungsional sistem yang telah dijabarkan sebelumnya untuk menentukan keberhasilan pengujian. Pengujian fungsional ini telah diberikan beberapa skenario yang telah dijabarkan pada bab sebelumnya. Berikut skenario yang telah ditentukan.

#### 7.1.1 Pengujian Kode PL-01

Pengujian dengan kode PL-01 dengan kasus uji node sensor mampu mengambil data sensor yang dapat dilihat pada Tabel 7.1.

**Tabel 7.1 Pengujian Kode PL-01**

<b>Kode</b>	PL-01
<b>Fungsi</b>	Node sensor mampu mengambil data sensor
<b>Tujuan Pengujian</b>	Menguji proses pengambilan data sensor
<b>Skenario Pengujian</b>	<ol style="list-style-type: none"> <li>1. Node sensor dalam keadaan aktif.</li> <li>2. Terdapat sensor yang terhubung pada node sensor.</li> <li>3. Mengambil data sensor yang terpasang pada node sensor.</li> <li>4. Menampilkan data keluaran dari sensor</li> </ol>
<b>Hasil yang diharapkan</b>	Node sensor berhasil mengambil data sensor
<b>Hasil Pengujian</b>	Berhasil mengambil data sensor

```
pi@raspberrypi:~/FileGit/backup-skripsi $ python node_fungsional.py
RF95 LoRa mode ok, Let's Go!!
T R A N S M I T
-----
data ke | data | RSSI
0 Lembab -99
```

**Gambar 7.1 Hasil Dari Pengujian PL-01**

Pada Gambar 7.1 menunjukkan tampilan pada terminal Raspberry Pi yang diambil pada sisi node sensor. Sensor yang digunakan adalah kelembapan tanah, yang dipasangkan mikrokomputer Raspberry Pi. Hasil yang ditampilkan menunjukkan bahwa node sensor berhasil mengambil data sensor yang telah terpasang pada node sensor. Data sensor yang berhasil diambil nantinya akan di *pack* dengan menggunakan *struct* kemudian data tersebut dikirimkan ke *gateway* dengan menggunakan modul *LoRa* yang telah terhubung dengan Raspberry Pi.

### 7.1.2 Pengujian Kode PL-02

Pengujian dengan kode PL-02 dengan kasus uji node sensor dapat mengirimkan data ke *gateway* yang dapat dilihat pada Tabel 7.2.

**Tabel 7.2 Pengujian Kode PL-02**

<b>Kode</b>	PL-02
<b>Fungsi</b>	Node sensor dapat mengirimkan data ke <i>gateway</i> .
<b>Tujuan Pengujian</b>	Menguji proses pengiriman node sensor ke <i>gateway</i>
<b>Skenario Pengujian</b>	<ol style="list-style-type: none"> <li>1. Node sensor dalam keadaan aktif.</li> <li>2. Modul <i>LoRa</i> HopeRF-RFM9x dalam keadaan aktif</li> <li>3. Node sensor telah mengambil data dari sensor.</li> <li>4. Node sensor mengirim data ke <i>gateway</i>.</li> </ol>
<b>Hasil yang diharapkan</b>	Node sensor berhasil mengirim data ke <i>gateway</i>
<b>Hasil Pengujian</b>	Berhasil mengirim data ke <i>gateway</i>

```
pi@raspberrypi:~/FileGit/backup-skripsi $ python node_fungsional.py
RF95 LoRa mode ok, Let's Go!!
T R A N S M I T
-----
data ke | data | RSSI
0 Lembab -99
```

**Gambar 7.2 Hasil Dari Pengujian PL-02**

Pada Gambar 7.2 menunjukkan tampilan pada terminal Raspberry Pi yang diambil pada sisi node sensor. Hasil yang ditampilkan menunjukkan bahwa node sensor berhasil mengambil data sensor kelembapan tanah. Data tersebut di *pack* kedalam *struct* dengan tipe data *string* dan mengirim data tersebut ke *gateway*.



Data dikirimkan ke *gateway* menggunakan modul *LoRa* yang telah terhubung pada Raspberry Pi dengan frekuensi 433 MHz.

### 7.1.3 Pengujian Kode PL-03

Pengujian dengan kode PL-03 dengan kasus uji *gateway* dapat menerima data dari node sensor yang dapat dilihat pada tabel 7.3.

**Tabel 7.3 Pengujian Kode PL-03**

<b>Kode</b>	PL-03
<b>Fungsi</b>	Gateway dapat menerima data dari node sensor
<b>Tujuan Pengujian</b>	Menguji proses penerimaan data pada gateway
<b>Skenario Pengujian</b>	<ol style="list-style-type: none"> <li>1. Gateway dalam keadaan aktif.</li> <li>2. Modul <i>LoRa</i> HopeRF-RFM9x dalam keadaan aktif.</li> <li>3. <i>Gateway</i> menampilkan data yang diterima dari node sensor.</li> </ol>
<b>Hasil yang diharapkan</b>	Gateway berhasil menerima data dari node sensor
<b>Hasil Pengujian</b>	Berhasil menerima data dari node sensor

```
pi@raspberrypi:~/lora/backup-skripsi $ python gw_fungsional.py
RF95 LoRa mode ok, Let's Go!!
Subscribing topic ...
Connected with Code :0
G A T E W A Y
-----
data ke | data | RSSI
0 Lembab -23
-----Terkirim ke Server-----
```

**Gambar 7.3 Hasil Dari Pengujian PL-03**

Pada Gambar 7.3 menunjukkan tampilan pada terminal Raspberry Pi yang diambil pada sisi *gateway*. Hasil yang ditampilkan menunjukkan bahwa *gateway* berhasil menerima data sensor yang dikirimkan oleh node sensor. Data yang terdapat ada kotak merah merupakan dari node sensor yang diterima *gateway*. Data yang didapat dari node sensor berupa *struct*, yang mana harus dilakukan *unpacking* agar dapat ditampilkan seperti gambar diatas. Format data yang diterima yaitu “data ke” yang menunjukkan data ke berapa, “data” yang menunjukkan isi dari data yang dikirim , dan RSSI yang menunjukkan besar kekuatan sinyal modul *LoRa* saat itu.

### 7.1.4 Pengujian Kode PL-04

Pengujian dengan kode PL-04 dengan kasus uji *gateway* dapat meneruskan data dari node sensor ke server yang dapat dilihat pada tabel 7.4.

Tabel 6.4 Pengujian Kode PL-04

<b>Kode</b>	PL-04
<b>Fungsi</b>	Gateway dapat meneruskan data dari node sensor ke server
<b>Tujuan Pengujian</b>	Menguji proses meneruskan data pada gateway ke server
<b>Skenario Pengujian</b>	<ol style="list-style-type: none"> <li>1. Gateway dalam keadaan aktif.</li> <li>2. Modul <i>LoRa</i> HopeRF-RFM9x dalam keadaan aktif.</li> <li>3. <i>Gateway</i> telah menerima data dari node sensor.</li> <li>4. <i>Gateway</i> menulis topik untuk <i>publish</i> pesan</li> </ol>
<b>Hasil yang diharapkan</b>	Gateway berhasil meneruskan data dari node sensor ke server
<b>Hasil Pengujian</b>	Berhasil meneruskan data dari node sensor ke server

```

pi@raspberrypi:~/lora/backup-skripsi $ python gw_fungsional.py
RF95 LoRa mode ok, Let's Go!!
Subscribing topic ...
Connected with Code :0
G A T E W A Y
-----
data ke | data | RSSI
0 Lembab -23
-----Terkirim ke Server-----
    
```

Gambar 7.4 Hasil Dari Pengujian PL-04

Pada Gambar 7.4 menunjukkan tampilan pada terminal Raspberry Pi yang diambil pada sisi *gateway*. Data yang berhasil diterima dalam bentuk *struct* dan telah dilakukan *unpacking* akan di *pack* kembali dalam bentuk json untuk di kirim ke server dengan menggunakan protokol MQTT. Komunikasi protokol MQTT menggunakan metode *publish-subscribe* yang nanti *gateway* akan melakukan *publish* data ke server yang dikirim melalui *broker*. Hasil yang ditampilkan pada Gambar 7.4 menunjukkan bahwa *gateway* berhasil meneruskan data sensor ke server.

### 7.1.5 Pengujian Kode PL-05

Pengujian dengan kode PL-05 dengan kasus uji server dapat menerima data dari *gateway* yang dapat dilihat pada tabel 7.5.

Tabel 7.5 Pengujian Kode PL-05



<b>Kode</b>	PL-05
<b>Fungsi</b>	Server dapat menerima data dari <i>gateway</i>
<b>Tujuan Pengujian</b>	Menguji proses server dapat menerima data dari <i>gateway</i>
<b>Skenario Pengujian</b>	<ol style="list-style-type: none"> <li>1. Server dalam keadaan aktif</li> <li>2. Server melakukan <i>subscribe</i> topik yang telah ditentukan.</li> <li>3. Server menerima data dari <i>gateway</i> sesuai topik.</li> </ol>
<b>Hasil yang diharapkan</b>	Server berhasil menerima data dari <i>gateway</i>
<b>Hasil Pengujian</b>	Berhasil menerima data dari <i>gateway</i>

```
C:\Users\hades\Python\MQTT>python server_fungsional.py
Subscribing topic ...
Connected with Code :0
S U B S C R I B E D
-----
topic | data ke | data
nodes/0 0 Lembab
Delivered
```

**Gambar 7.5 Hasil Dari Pengujian PL-05**

Pada Gambar 7.5 menunjukkan tampilan pada terminal server pada sisi server. Data dari *gateway* dikirimkan dalam bentuk *json* yang dikirimkan ke server melalui *broker*. *Broker* kemudian meneruskan data tersebut ke server sesuai dengan topik yang telah di *subscribe* oleh server. Hasil yang ditampilkan menunjukkan bahwa server berhasil menerima data dari node sensor yang dikirim melalui *gateway* dengan topik *nodes/0*.

## 7.2 Pengujian Kinerja

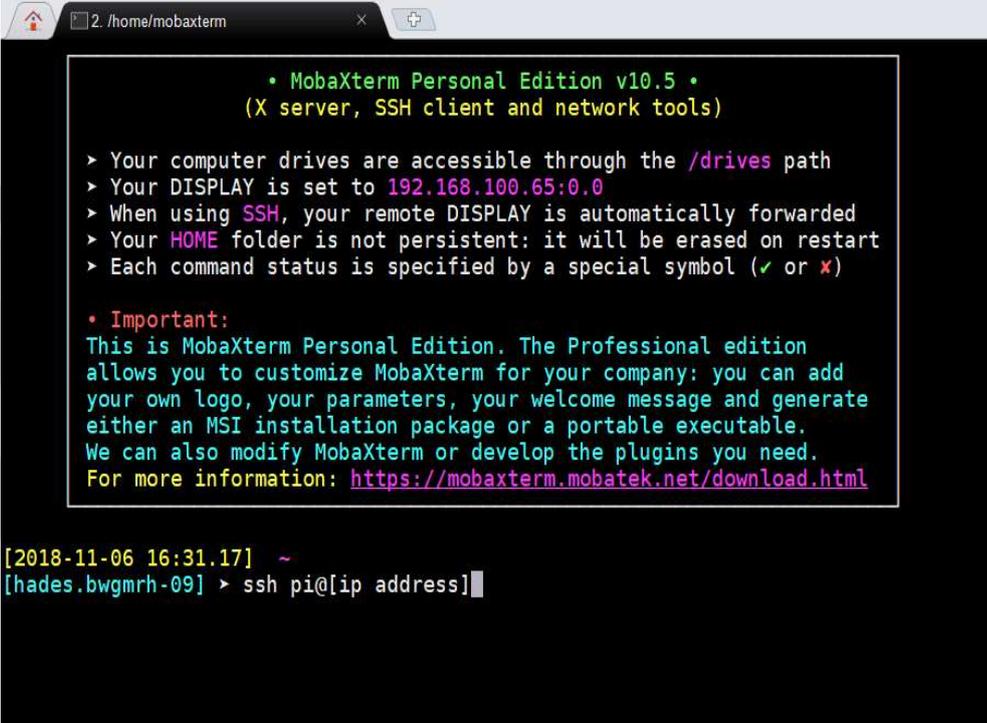
Pengujian kinerja dilakukan untuk menguji keandalan sistem yang telah dibuat. pengujian kinerja berfokus pada parameter *successful rate gateway* dalam menerima dan meneruskan paket sampai ke tujuan. Pegujian akan dilakukan dengan *tools* MobaXterm sebagai pengumpulan data yang akan diuji di lapangan. Skenario pengujian kinerja telah dijelaskan pada bab perancangan, sehingga disini akan langsung ditunjukkan hasil yang diperoleh pengujian kinerja yang sudah dilakukan. Kebutuhan, batasan serta hasil yang diharapkan dalam pengujian kinerja dapat dilihat pada Tabel 7.11.

**Tabel 7.11 Pengujian Kinerja Sistem**

Kode	Parameter	Keterangan
------	-----------	------------

PT_001	<i>Successful Rate</i>	Menggunakan <i>tools</i> MobaXterm untuk melakukan pengumpulan data dengan variasi jarak sebesar 200 meter, 400 meter dengan besar paket 44 <i>Byte</i> , 56 <i>Byte</i> , 80 <i>Byte</i> , 128 <i>Byte</i> dan interval waktu 0,5 detik, 1 detik, 5 detik.
--------	------------------------	---

Pengujian ini diawali dengan melakukan instalasi MobaXterm yang akan digunakan untuk pengambilan data. Tahap selanjutnya yaitu mempersiapkan perangkat yang akan digunakan seperti node sensor, *gateway*, dan server. Dibutuhkan beberapa laptop sebagai sumber daya untuk perangkat Raspberry Pi yang akan digunakan untuk pengujian. Pada setiap jarak, besar paket, dan interval yang jelaskan akan dilakukan proses pengiriman data sebesar 100 paket untuk mendapatkan hasil yang akurat. Selanjutnya, konfigurasi pada *tools* MobaXterm agar dapat melakukan pengambilan data dengan menggunakan SSH.

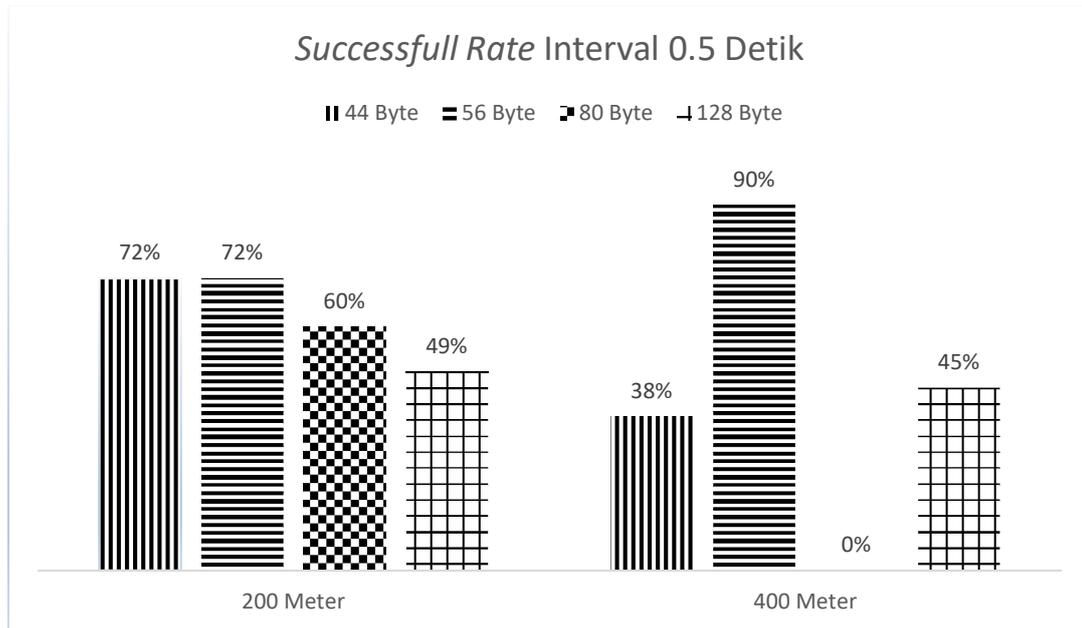


**Gambar 7.11 Konfigurasi SSH pada MobaXterm**

Pengujian yang dilakukan adalah *successfull rate gateway* pada interval 0.5 detik. Pengujian ini dilakukan dengan pengiriman paket sebanyak 100 kali. Pengujian ini dilakukan dengan variasi jarak yang telah ditentukan yaitu 200 meter dan 400 meter. Pada jarak tersebut terdapat variasi besar paket yang dikirim yaitu 44 *byte*, 56 *byte*, 80 *byte*, 128 *byte*. Pada percobaan yang dilakukan didapatkan hasil



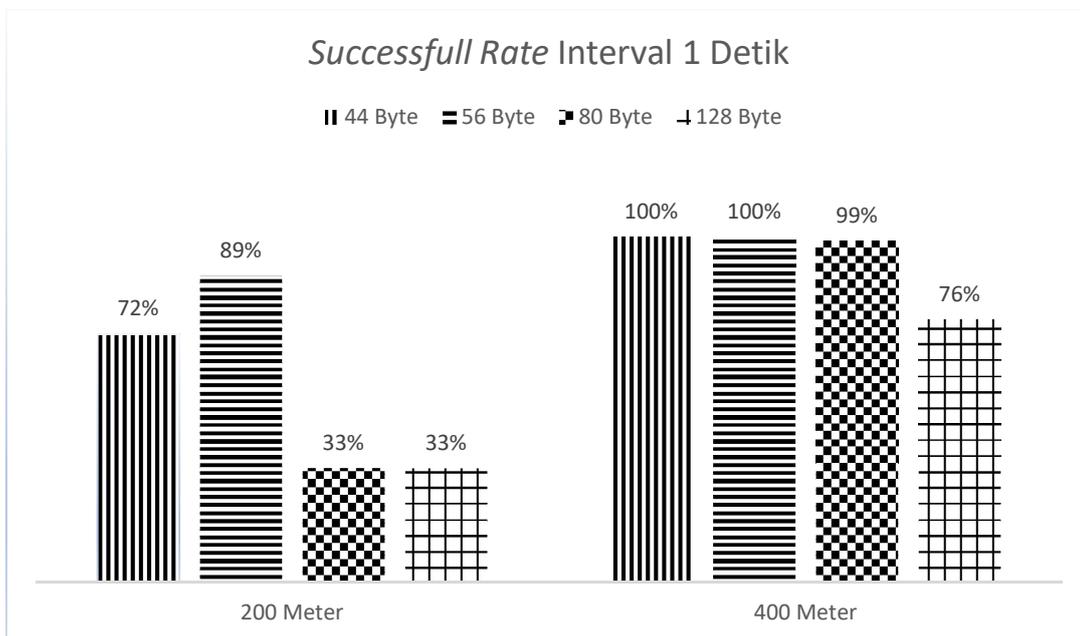
pengujian dengan tingkat keberhasilan besar paket pada setiap jarak yang dapat dilihat pada Gambar 7.12.



**Gambar 7.12 Grafik Hasil Pengujian *Successful Rate* Dengan Interval Waktu 0.5 Detik**

Pada Gambar 7.12 merupakan grafik yang menunjukkan hasil pengujian kinerja terhadap interval waktu 0.5 detik dengan parameter *successful rate*. Gambar grafik tersebut menunjukkan perbedaan variasi besar paket yang dikirim dan di paparkan dalam bentuk persen. Selisih tingkat keberhasilan *gateway* dalam menerima dan meneruskan paket ke server dalam interval 0,5 detik pada jarak 200 meter mendapatkan hasil 72% pada besar paket 44 *Byte* dan 56 *Byte*. Pada besar paket 80 *Byte* tingkat keberhasilan menurun sebesar 12% dan pada besar paket 128 *Byte* menjadi 49%. Pengujian jarak 400 meter juga mendapatkan hasil sebesar 38% pada besar paket 44 *Byte* dan naik menjadi 90% pada besar paket 56 *Byte*, namun pada besar paket 80 *Byte* tingkat keberhasilan menurun drastis menjadi 0% kemudian naik kembali menjadi 45% pada besar paket 128 *Byte*. Dari pengujian kinerja dengan interval 0,5 detik yang dilakukan, didapatkan hasil bahwa pada interval 0,5 detik pada jarak 200 meter kinerja *gateway* bisa dikatakan sangat baik, namun pada jarak 400 meter kinerja *gateway* kurang baik.

Pengujian kinerja berikutnya adalah pengujian *successful rate gateway* dengan interval waktu 1 detik. Pengujian ini dilakukan dengan mengirimkan paket sebanyak 100 kali. Pengujian ini dilakukan dengan variasi jarak yang sama seperti pengujian sebelumnya yaitu 200 meter dan 400 meter. Jarak tersebut terdapat variasi besar paket yang dikirim yaitu 44 *Byte*, 56 *Byte*, 80 *Byte*, 128 *Byte*. Pada percobaan yang dilakukan didapatkan hasil pengujian dengan tingkat keberhasilan besar paket pada setiap jarak yang dapat dilihat pada Gambar 7.13.



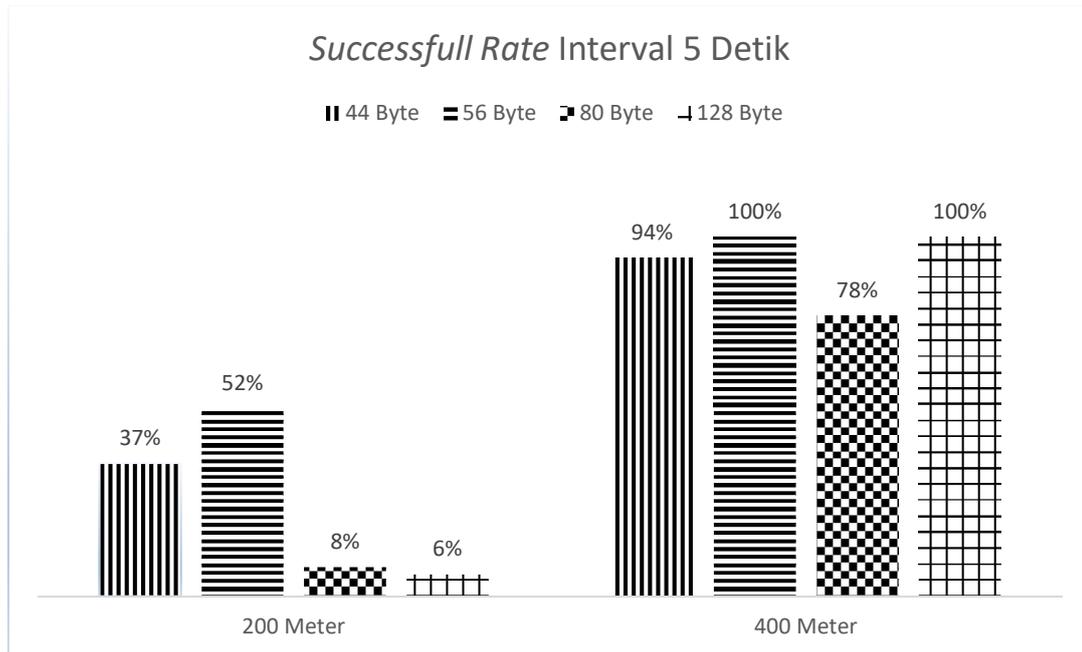
**Gambar 7.13 Grafik Hasil Pengujian *Successfull Rate* Dengan Interval Waktu 1 Detik**

Pada Gambar 7.13 merupakan grafik yang menunjukkan hasil pengujian kinerja terhadap interval waktu 1 detik dengan parameter *successfull rate*. Gambar grafik tersebut menunjukkan perbedaan variasi besar paket yang dikirim dan di paparkan dalam bentuk persen. Selisih tingkat keberhasilan *gateway* dalam menerima dan meneruskan paket ke server dalam interval 1 detik pada jarak 200 meter didapatkan 72% paket yang berhasil diterima dan diteruskan *gateway* dengan besar paket 44 *Byte*. Selanjutnya, paket mengalami kenaikan menjadi 89% di besar paket 56 *Byte* dan kemudian stabil menjadi 33% di besar data 80 *Byte* dan 128 *Byte*. Pada jarak 400 meter hasil yang di dapatkan cukup stabil di angka 100% sampai 99% di besar paket 44 *Byte*, 56 *Byte*, 80 *Byte*, namun pada besar paket 128 *Byte* mengalami penurunan sebesar 23%.

Selisih dalam grafik diatas dapat disebabkan oleh kondisi lingkungan saat pengujian ataupun *wiring* pada *hardware* yang dirancang, hal ini perlu dilakukan penelitian lebih lanjut untuk mengetahui penyebab terjadinya perbedaan yang begitu jauh. Dari pengujian kinerja yang dilakukan, didapatkan hasil bahwa pada interval 1 detik pada jarak 200 meter kinerja *gateway* bisa dikatakan baik dan pada jarak 400 meter kinerja *gateway* sangat baik.

Pengujian kinerja yang terakhir adalah pengujian *successfull rate gateway* dengan interval waktu 5 detik. Pengujian ini dilakukan dengan mengirimkan paket sebanyak 100 kali. Pengujian ini dilakukan dengan variasi jarak yang sama seperti pengujian sebelumnya yaitu 200 meter dan 400 meter. Jarak tersebut terdapat variasi besar paket yang dikirim yaitu 44 *Byte*, 56 *Byte*, 80 *Byte*, 128 *Byte*. Pada

percobaan yang dilakukan didapatkan hasil pengujian dengan tingkat keberhasilan besar paket pada setiap jarak yang dapat dilihat pada Gambar 7.14.



**Gambar 7.14 Grafik Hasil Pengujian *Successfull Rate* Dengan Interval Waktu 5 Detik**

Pada Gambar 7.14 menunjukkan grafik hasil pengujian kinerja terhadap interval waktu 5 detik dengan parameter *successfull rate*. Gambar grafik tersebut menunjukkan perbedaan variasi besar paket yang dikirim dan di paparkan dalam bentuk persen. Selisih tingkat keberhasilan gateway dalam menerima dan meneruskan paket ke server dalam interval 5 detik pada jarak 200 meter terdapat 37% paket dengan besar 44 Byte yang berhasil diterima dan diteruskan ke server. Pada paket 56 byte mengalami kenaikan sebesar 15%, lalu paket selanjutnya turun menjadi 8% di besar paket 80 Byte dan menjadi 6% di 128 Byte. Pada jarak 400 meter gateway berhasil menerima dan meneruskan paket sebesar 94% pada besar paket 44 Byte dan 78% di besar paket 80 Byte, namun pada besar paket 56 Byte dan 128 Byte *successfull rate gateway* naik menjadi 100%.

Selisih dalam grafik diatas dapat disebabkan oleh kondisi lingkungan saat pengujian ataupun wiring pada hardware yang dirancang, hal ini perlu dilakukan penelitian lebih lanjut untuk mengetahui penyebab terjadinya selisih yang begitu jauh. Dari pengujian kinerja yang dilakukan, didapatkan hasil bahwa pada interval 5 detik pada jarak 200 meter kinerja gateway bisa dikatakan kurang baik, namun pada jarak 400 meter kinerja gateway sangat baik.

## BAB 8 PENUTUP

Bab ini berisi tentang kesimpulan dari hasil penelitian yang telah dilakukan dan saran terhadap penelitian selanjutnya. Kesimpulan dan saran pada bab ini diperoleh dari hasil perancangan, implementasi dan pengujian.

### 8.1 Kesimpulan

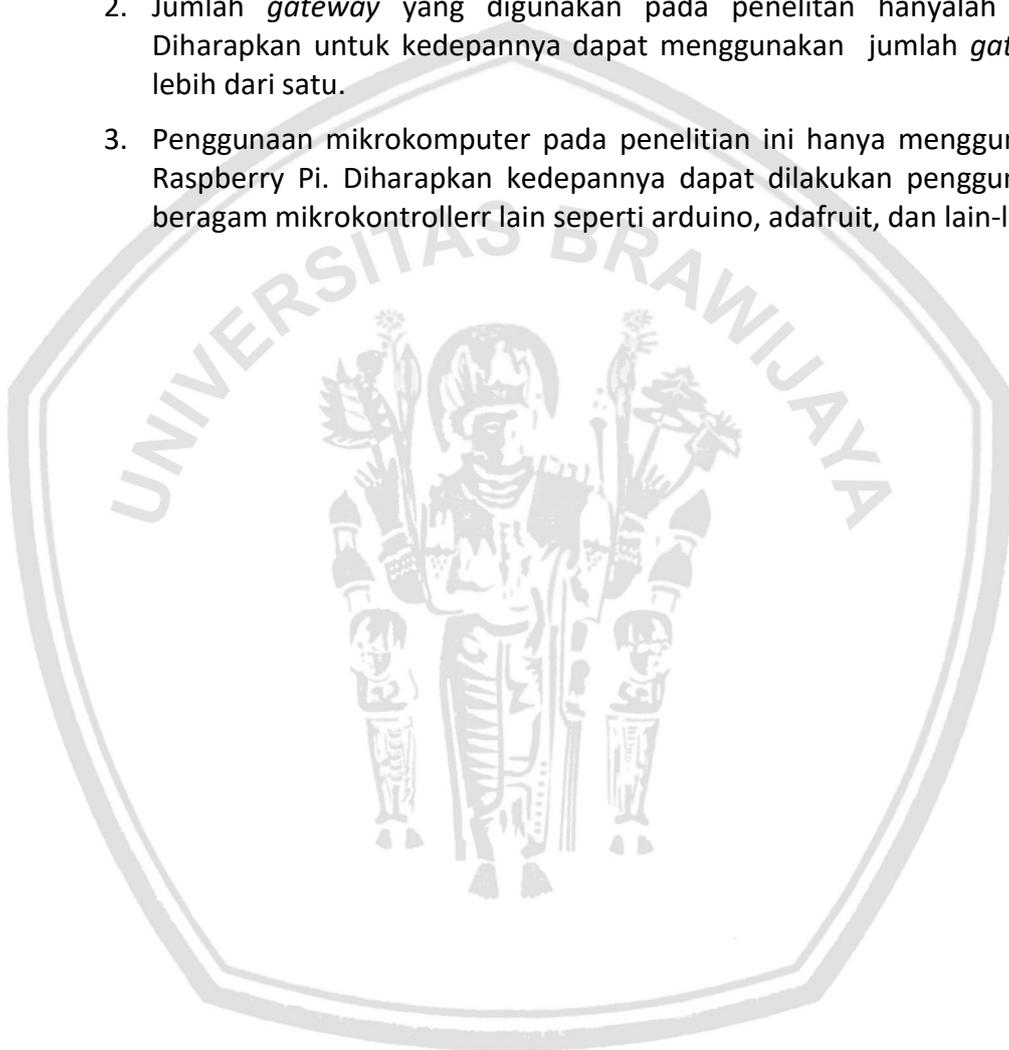
Kesimpulan yang diperoleh berdasarkan hasil pada proses bab perancangan, implementasi, dan pengujian yang telah dilakukan adalah sebagai berikut:

1. Sistem monitoring pada *Wireless Sensor Network (WSN)* menggunakan perangkat mikrokomputer Raspberry Pi, sensor kelembapan tanah, dan modul komunikasi *LoRa HopeRF-RFM9x* pada sisi node sensor. Pada sisi *gateway* menggunakan mikrokomputer Raspberry Pi dan modul komunikasi *LoRa HopeRF-RFM9x*. *LoRa HopeRF-RFM9x* menjembatani komunikasi nirkabel pada sistem *WSN* yang telah dibuat dengan dibekali antena helical yang merupakan kelengkapan bawaan dari modul komunikasi *LoRa HopeRF-RFM9x*. Sistem pada sisi node sensor diimplementasikan dengan node sensor mengambil data dari sensor node kemudian mengirimkan data tersebut ke *gateway* dengan menggunakan modul komunikasi *LoRa*. *Gateway* dapat menerima data dari node sensor dengan menggunakan modul *transceiver LoRa HopeRF-RFM9x*. Data dari *gateway* diteruskan ke server dengan protokol MQTT menggunakan jaringan 802.11.
2. Pertukaran data dalam komunikasi antara node sensor dan *gateway* diolah menggunakan *struct*. Data *struct* yang diterima oleh *gateway* akan diubah menjadi JSON untuk dikirim ke server. Data dikirim ke server dengan menggunakan metode komunikasi *publish-subscribe* dengan menggunakan protokol MQTT. Pertukaran data antara *gateway* dengan server menggunakan metode komunikasi *publish-subscribe*. *Gateway* melakukan *publish* data dengan topik yang telah ditentukan dan mengirim ke server melalui *broker*. *Broker* kemudian meneruskan data ke server sesuai topik yang di *subscribe* oleh server. Kemudian server menampilkan melalui monitor pada sisi server.
3. Kinerja dari sistem yang telah diuji terdapat kendala jika dilakukan pengujian terhadap jarak antara node sensor dan *gateway*. Menurut hasil dari pengujian kinerja yang telah dilakukan sebelumnya, dari parameter dan tiga variabel yang digunakan menghasilkan data grafik yang menurun signifikan pada interval waktu 1 detik dan 5 detik pada jarak 200 meter. Hal ini berdampak pada kinerja sistem perantara yang dibuat dengan menurunnya nilai dari parameter tersebut. Oleh karena itu, dapat disimpulkan bahwa pada jarak 400 meter *gateway* memiliki kinerja yang lebih baik daripada jarak 200 meter.

## 8.2 Saran

Penelitian yang telah dilakukan masih memiliki beberapa kekurangan. Oleh karena itu, diperlukan pengembangan dan perbaikan agar di penelitian selanjutnya dapat menjadi lebih baik. Berikut adalah beberapa saran yang disarankan peneliti untuk penelitian kedepannya:

1. Protokol yang digunakan pada penelitian ini untuk mengirimkan ke server hanya menggunakan protokol MQTT. Diharapkan kedepannya dapat dilakukan komparasi kinerja berbagai protokol yang beragam.
2. Jumlah *gateway* yang digunakan pada penelitian hanyalah satu. Diharapkan untuk kedepannya dapat menggunakan jumlah *gateway* lebih dari satu.
3. Penggunaan mikrokomputer pada penelitian ini hanya menggunakan Raspberry Pi. Diharapkan kedepannya dapat dilakukan penggunaan beragam mikrokontrollerr lain seperti arduino, adafruit, dan lain-lain.



## DAFTAR PUSTAKA

- Chang-le Zhong, Z. Z. R.-g. H., 2015. Study on the IOT Architecture and Gateway Technology. *IEEE LATIN AMERICA TRANSACTIONS*.
- Chase, J., 2013. Dalam: *The Evolution of the Internet of Things*. s.l.:Texas Instruments, p. 1.
- dhall, R. & Solanki, V., 2017. An IoT Based Predictive Connected Car Maintenance. *International Journal of Interactive Multimedia and Artificial Intelligence*, Volume IV.
- Guoqiang, S., Yanming, C., Chao, Z. & Yanxu, Z., 2013. Design and Implementation of a Smart IoT Gateway.
- Habibi, M. W., Bhawiyuga, A. & Basuki, A., 2017. Rancang Bangun IoT Cloud Platform Berbasis Protokol Komunikasi MQTT.
- I.F. Akyildiz, W. S., Sankarasubramaniam, Y. & Cayirci, E., 2001. Wireless sensor networks: a survey. p. 405.
- Kim, S.-M., Choi, H.-S. & Rhee, W. S., 2015. IoT Home Gateway for Auto-Configuration and Management of MQTT Devices.. *IEEE Conference on Wireless Sensors*.
- Lavric, A. & Popa, V., 2017. Internet of Things and LoRaTM Low-Power Wide-Area Networks: A Survey. *IEEE*.
- Oasis, 2015. *MQTT Version 3.1.1 Plus Errata 01*. [Online]  
Available at: [http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/errata01/os/mqtt-v3.1.1-errata01-os-complete.html#\\_Toc442180822](http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/errata01/os/mqtt-v3.1.1-errata01-os-complete.html#_Toc442180822)  
[Diakses 10 February 2018].
- Potsch, A. & Haslhofer, F., 2017. Practical Limitations for Deployment of LoRa Gateways. *IEEE*.
- Python, 2017. *What is Python*. [Online]  
Available at: <https://docs.python.org/3/faq/general.html#what-is-python>  
[Diakses 9 Februari 2018].
- Wixted, A. J. et al., 2016. Evaluation of LoRa and LoRaWAN for Wireless Sensor Network.
- Yacchirema, D. C. & Palau, C., 2016. Smart IoT Gateway For Heterogeneous Devices Interoperability. *IEEE LATIN AMERICA TRANSACTIONS*, Volume 14, pp. 3900-3906.