

**IMPLEMENTASI *NETWORK BRIDGE SYSTEM* UNTUK
KOMUNIKASI PERANGKAT IP DAN PERANGKAT NON IP
BLUETOOTH**

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Lilis Suryani

NIM: 125150300111005



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016**

PENGESAHAN

IMPLEMENTASI *NETWORK BRIDGE SYSTEM* UNTUK KOMUNIKASI PERANGKAT IP
DAN PERANGKAT NON IP BLUETOOTH

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Lilis Suryani

NIM: 125150300111005

Skripsi ini telah diuji dan dinyatakan lulus pada
01 Juli 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Sabriansyah Rizqika Akbar, S.T, M.Eng
NIP. 19820809 201212 1 004

Aswin Suharsono, S.T, M.T
NIP. 201102 840919 1 001

Mengetahui
Pjs Ketua Program Studi Informatika

Issa Arwani, S.Kom, M.Sc
NIP. 19830922 201212 1 003

PERNYATAAN ORISINALITAS

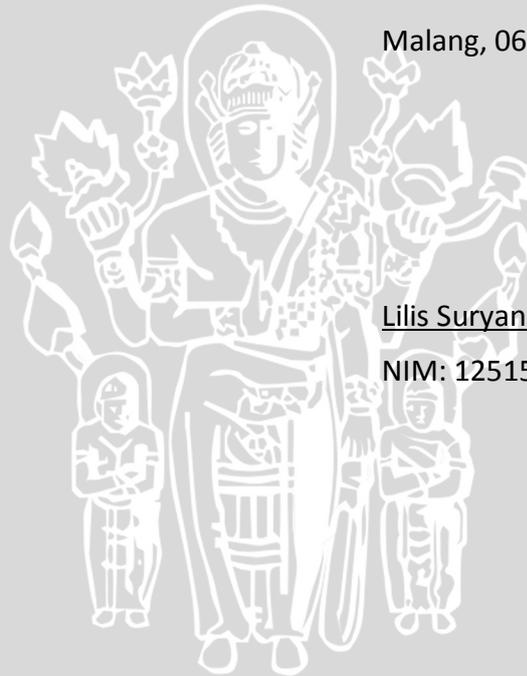
Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam 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, 06 Juni 2016

Lilis Suryani

NIM: 125150300111005



KATA PENGANTAR

Puji dan rasa syukur mendalam penulis panjatkan kehadirat Allah SWT, karena berkat limpahan rahmat, hidayah, dan inayah-Nya maka skripsi ini dapat diselesaikan dengan baik. Salam dan shalawat semoga selalu tercurah pada baginda Rasulullah Muhammad SAW.

Skripsi yang berjudul "*Implementasi Network bridge system untuk Komunikasi Perangkat IP dan Perangkat Non IP Bluetooth*" ini saya susun untuk memenuhi persyaratan kurikulum sarjana strata-1 (S-1) pada Jurusan Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya.

Penulis mengucapkan rasa terima kasih yang sebesar-besarnya atas semua bantuan yang telah diberikan, baik secara langsung maupun tidak langsung selama penyusunan tugas akhir ini hingga selesai. Secara khusus rasa terimakasih tersebut saya sampaikan kepada:

1. Bapak **Sabriansyah Rizqika Akbar, S.T, M.T** selaku Dosen Pembimbing I yang telah banyak membantu dalam penyusunan skripsi.
2. Bapak **Aswin Suharsono, S.T, M.T** selaku Dosen Pembimbing II yang telah banyak membantu dalam penyusunan skripsi.
3. Bapak **Issa Arwani, S.Kom, M.Sc** selaku Ketua Prodi Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
4. Bapak **Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D** selaku Dekan Fakultas Ilmu Komputer, Universitas Brawijaya
5. Seluruh dosen dan karyawan Jurusan Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya, atas ilmu, bimbingan dan bantuannya hingga penulis selesai menyusun tugas akhir ini.
6. Ayahanda Parluji dan Ibu Lusiana Yesi Susdiani, selaku orang tua penulis yang telah membesarkan dan mendidik, serta memberikan dukungan dan doa kepada penulis.
7. Kakak Imam Hari Kristanto dan adik Shela Mita Tri Wahyuni, selaku saudara kandung yang telah memberikan dukungan dan doa kepada penulis.
8. Sahabat hidup Rizki Habibie yang selalu mengingatkan saya untuk menyelesaikan skripsi dan selalu memberikan dukungan kepada penulis.
9. Sahabat baik Armita Miranjani, Irsyad Nur Hamid, Ike Mia Wulandari, Singgih Saputra, Ardy Novian, Ulfa Kurniawati, Kiki M. Rizki dan Muh Rifqi Maulana yang selalu memberikan dukungan untuk menyelesaikan skripsi ini.
10. Sedulur – sedulur saya satu daerah kebumen Areska W, Fita, Hazun, Sukur, Arman, olin, putri, tasi, dopi, azis dan seluruh keluarga himalaya atas dukukungan dan doanya.

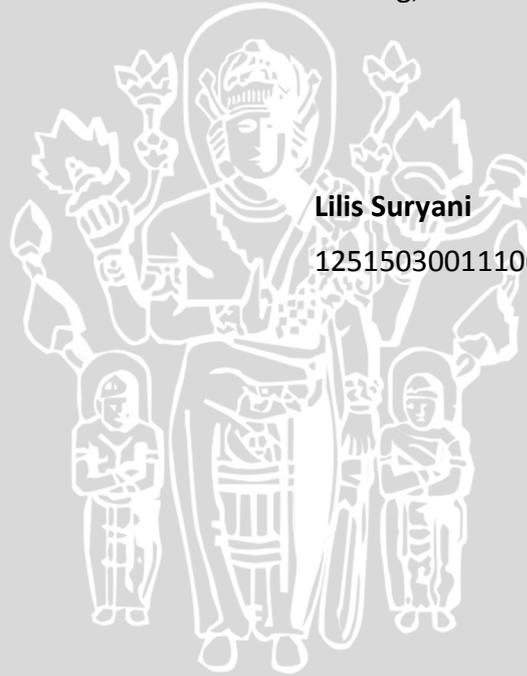
11. Teman – teman Kos Sofi Hanifah, yaya, Jehan, Endah, Elva yang selalu mengingatkan dan memotivasi saya.
12. Teman - teman di Jurusan Sistem Komputer Tahun 2012, Fakultas Ilmu Komputer, Universitas Brawijaya yang juga telah banyak membantu penulis.
13. Dan semua teman – teman yang tidak bisa saya sebutkan satu persatu terimakasih atas dukungan dan doanya.

Penulis menyadari bahwa tugas akhir ini belum sempurna, baik dari segi materi maupun penyajiannya. Untuk itu saran dan kritik yang membangun sangat diharapkan dalam penyempurnaan tugas akhir ini.

Malang, 06 Juni 2016

Lilis Suryani

125150300111005@mail.ub.ac.id



ABSTRAK

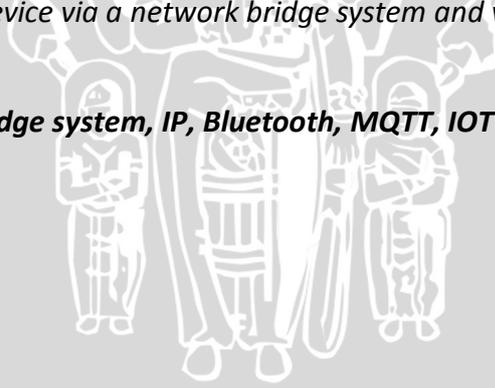
Kemajuan teknologi informasi dari berbagai sisi mendorong sebuah inovasi revolusi teknologi untuk kedepannya. Teknologi informasi telah berkembang sampai *cloud computing*, *Internet of Things* dan *Smart City*. Salah satu perangkat teknologi informasi adalah perangkat bluetooth. Bluetooth merupakan salah satu perangkat pendukung *Internet of Things*. Bluetooth adalah teknologi komunikasi *wireless* yang digunakan sebagai alternatif nirkabel untuk bertukar data menggunakan radio transmisi. Namun teknologi bluetooth mempunyai beberapa kekurangan dalam hal jangkauan pengiriman. Bluetooth hanya dapat mengirimkan data dalam radius jangkauan yang dekat dan hanya dapat mengirimkan data dalam lingkup personal area. Tidak seperti perangkat bluetooth, perangkat berbasis IP mempunyai kelebihan dalam pengiriman data dan jangkauan komunikasi yang lebih luas. Tujuan penelitian ini adalah menghubungkan perangkat bluetooth dan perangkat IP agar saling terhubung. Harapannya jika perangkat bluetooth dan perangkat IP dapat saling terhubung maka penyebaran data dari perangkat bluetooth lebih luas jangkauannya. Salah satu solusi yang ditawarkan agar perangkat bluetooth dan perangkat IP dapat saling terhubung adalah menggunakan *Network bridge system*. *Network bridge system* dalam sistem adalah sistem jaringan yang menjembatani komunikasi antar perangkat yang mempunyai protokol komunikasi yang berbeda. *Network bridge system* diimplementasikan menggunakan minikomputer *Raspberry pi* dengan tambahan fitur *wifi* dan bluetooth. *Network bridge system* diprogram untuk menghubungkan komunikasi antara perangkat bluetooth dan perangkat IP. Hasil pengujian yang didapat pada penelitian ini adalah perangkat bluetooth dapat berkomunikasi dengan perangkat IP melalui *network bridge system* dan sebaliknya.

Kata kunci: *Network bridge system*, IP, Bluetooth, MQTT, IoT

ABSTRACT

Advances in information technology from various sides to encourage a technological revolution for future innovation. Information technology has evolved to cloud computing, Internet of Things and Smart City. One of the tools of information technology is a bluetooth device. Bluetooth is one device supporting the Internet of Things. Bluetooth is a wireless communication technology that is used as a wireless alternative to exchange data using radio transmission. However, Bluetooth technology has some drawbacks in terms of range of delivery. Bluetooth can only transmit data within a radius close range and can only send personal data within the scope of the area. Unlike bluetooth devices, IP-based devices has advantages in the delivery of data and reach a broader communications. The purpose of this study was to connect a bluetooth device and IP devices to be connected. The hope is if a bluetooth device and IP device can be connected to each other, the dissemination of data from the device bluetooth wider scope. One of the solutions offered for the device and a Bluetooth device can connect to each IP is using the Network bridge system. Network bridge system in the system is a network system that bridges communication between devices with different communication protocols. Network bridge system is implemented using Raspberry pi minicomputer with additional features wifi and bluetooth. Network bridge system is programmed to connect the communication between Bluetooth devices and IP devices. The test results obtained in this study is a bluetooth device can communicate with IP device via a network bridge system and vice versa.

Keywords: Network bridge system, IP, Bluetooth, MQTT, IOT



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS.....	iii
KATA PENGANTAR	iv
ABSTRAK.....	vi
<i>ABSTRACT</i>	vii
DAFTAR ISI	viii
DAFTAR TABEL	xii
DAFTAR GAMBAR	xiii
DAFTAR LAMPIRAN.....	xv
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah	2
1.3 Tujuan.....	2
1.4 Manfaat	2
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
1.7 Jadwal penelitian	4
BAB 2 LANDASAN KEPUSTAKAAN.....	5
2.1 Tinjauan pustaka	5
2.2 Dasar teori	6
2.2.1 <i>Network bridge system</i>	6
2.2.2 Bluetooth.....	7
2.2.3 Mikrokomputer <i>Raspberry pi 2</i>	8
2.2.4 IPv6.....	9
2.2.5 MQTT <i>Publisher Subscriber</i>	10
BAB 3 METODOLOGI	11
3.1 Studi Literatur	12
3.2 Rekayasa Kebutuhan	13
3.2.1 Kebutuhan Pengguna	13
3.2.2 Kebutuhan Sistem	13

3.2.2 Kebutuhan Hardware	13
3.2.2 Kebutuhan software	14
3.3 Perancangan sistem	15
3.4 Implementasi	16
3.5 Pengujian dan analisa	17
3.6 Kesimpulan	17
BAB 4 Rekayasa Kebutuhan.....	18
4.1 Pendahuluan	18
4.1.1 Tujuan	18
4.1.2 Ruang Lingkup	18
4.1.3 Istilah	18
4.1.4 Referensi	19
4.1.5 Sistematika	19
4.2 Deskripsi Umum	20
4.2.1 Perspektif Produk/Sistem	20
4.2.2 Kegunaan.....	20
4.2.3 Karakteristik Pengguna	20
4.2.4 Lingkungan Operasi	20
4.2.5 Batasan Perancangan dan Implementasi.....	20
4.2.6 Asumsi dan Ketergantungan	21
4.3 Kebutuhan Antarmuka Eksternal.....	21
4.3.1 Kebutuhan Antarmuka Pengguna	21
4.3.2 Antarmuka Perangkat Keras.....	21
4.3.3 Antarmuka Perangkat Lunak	21
4.3.4 Antarmuka Komunikasi	21
4.4 Kebutuhan Fungsional.....	22
4.4.1 Fungsi Komunikasi Perangkat IP.....	22
4.4.2 Fungsi Komunikasi Perangkat Non IP	22
4.4.3 Fungsi <i>Network bridge system</i>	23
4.5 Kebutuhan Non Fungsional	23
4.5.1 Kebutuhan Performansi	23
4.5.2 Kebutuhan Keselamatan	23



4.5.3 Kebutuhan Keamanan.....	23
BAB 5 Perancangan dan implementasi.....	24
5.1 Perancangan Sistem.....	24
5.1.1 Gambaran umum arsitektur sistem.....	24
5.1.2 Alur Kerja Sistem.....	24
5.1.3 Perancangan Perangkat Keras.....	26
5.1.4 Perancangan Perangkat Lunak.....	26
5.1.5 Perancangan sistem Bluetooth.....	26
5.1.6 Perancangan <i>Network Bridge</i> Sistem.....	29
5.1.7 Perancangan sistem MQTT <i>Publisher Subscriber</i>	38
5.2 Implementasi Sistem.....	48
5.2.1 Spesifikasi Perangkat Keras.....	48
5.2.2 Spesifikasi Perangkat Lunak.....	51
5.2.3 Batasan Implementasi.....	51
5.2.4 Implementasi Sistem Bluetooth.....	51
5.2.5 Implementasi <i>Network Bridge</i> Sistem.....	53
5.2.6 Implementasi Sistem MQTT.....	55
BAB 6 Pengujian.....	58
6.1 Pengujian Sistem Bluetooth.....	58
6.1.1 Tujuan Pengujian.....	58
6.1.2 Pengujian.....	58
6.1.3 Hasil Pengujian.....	60
6.2 Pengujian Sistem <i>Network bridge system</i>	61
6.2.1 Tujuan Pengujian.....	61
6.2.2 Pengujian.....	61
6.2.3 Hasil Pengujian.....	62
6.3 Pengujian Sistem Perangkat IP.....	63
6.3.1 Tujuan Pengujian.....	63
6.3.2 Pengujian.....	63
6.3.3 Hasil Pengujian.....	64
6.3 Pengujian Performa.....	64
6.3.4 Hasil Pengujian.....	65

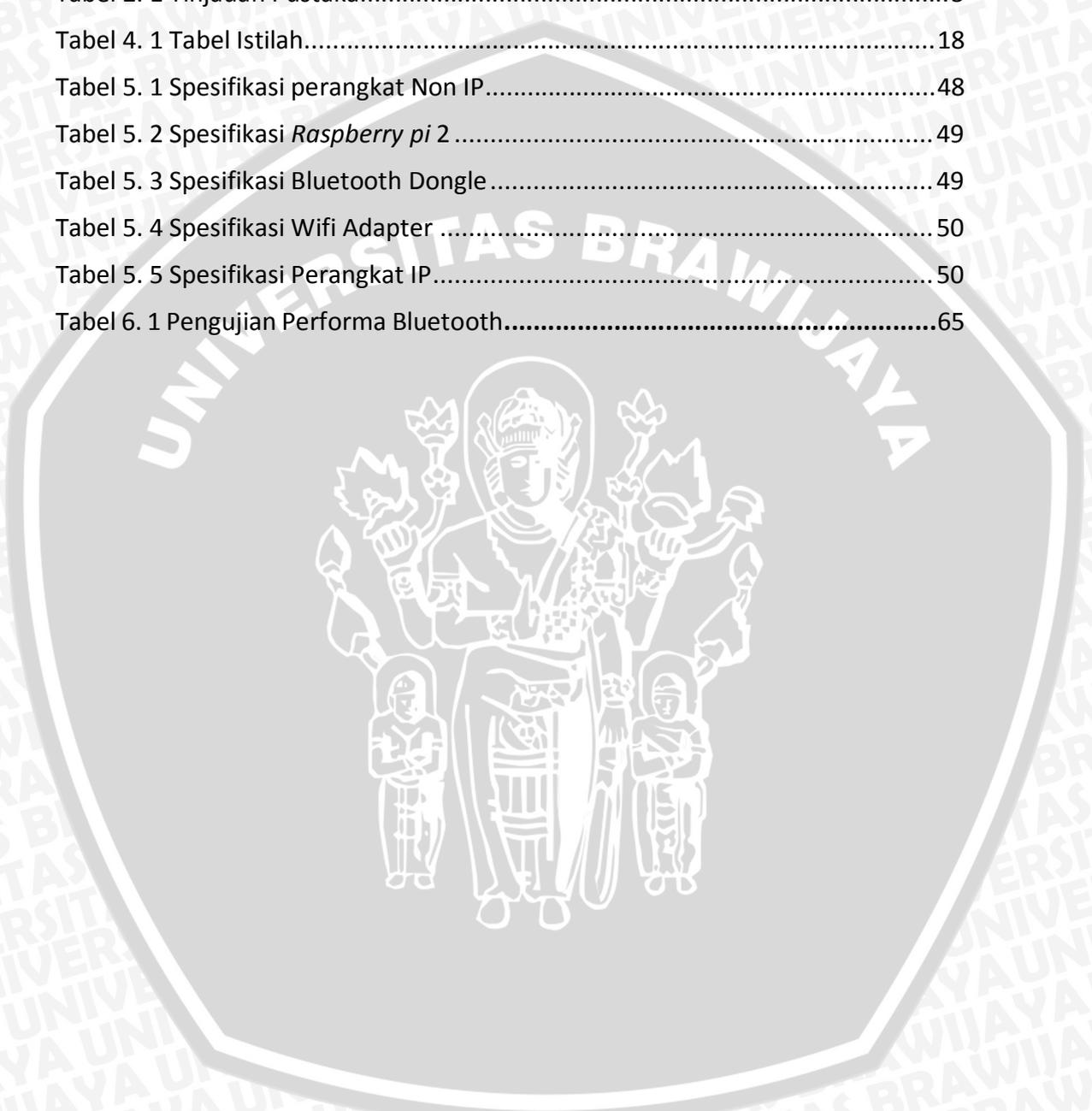


BAB 7 Penutup.....	67
7.1 Kesimpulan	67
7.2 Saran.....	67
DAFTAR PUSTAKA	68
LAMPIRAN	69



DAFTAR TABEL

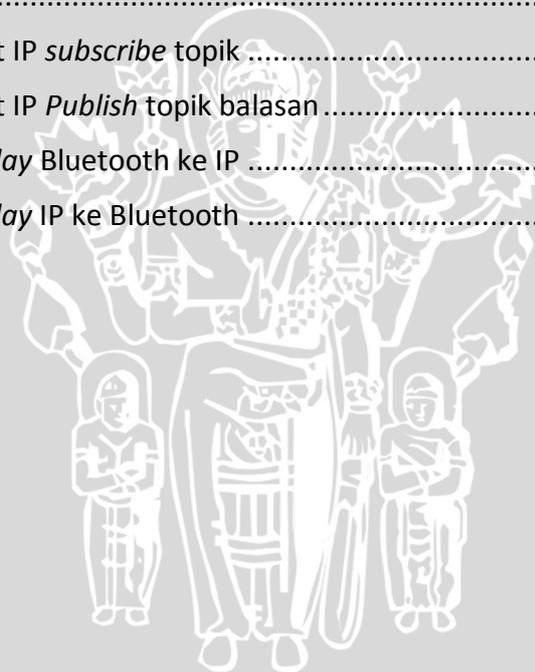
Tabel 1. 1 Jadwal Pelaksanaan Penelitian.....	4
Tabel 2. 1 Tinjauan Pustaka.....	5
Tabel 4. 1 Tabel Istilah.....	18
Tabel 5. 1 Spesifikasi perangkat Non IP.....	48
Tabel 5. 2 Spesifikasi <i>Raspberry pi 2</i>	49
Tabel 5. 3 Spesifikasi Bluetooth Dongle	49
Tabel 5. 4 Spesifikasi Wifi Adapter	50
Tabel 5. 5 Spesifikasi Perangkat IP.....	50
Tabel 6. 1 Pengujian Performa Bluetooth.....	65



DAFTAR GAMBAR

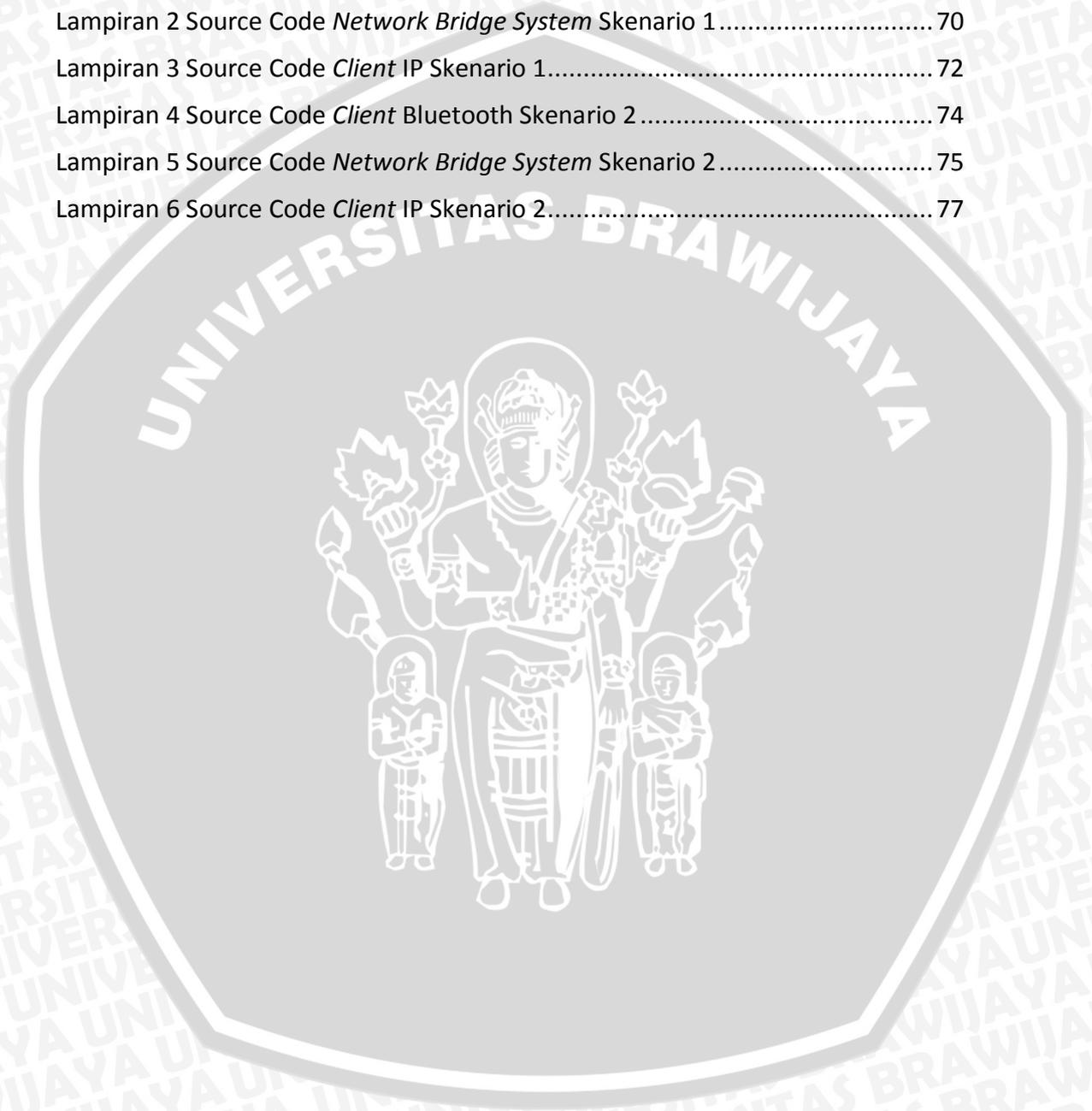
Gambar 2. 1 Contoh <i>Network Bridge</i>	6
Gambar 2. 2 Perbandingan L2CAP dan RFCOMM	8
Gambar 2. 3 <i>Raspberry pi 2 Model B</i>	8
Gambar 2. 4 Format Ipv6	9
Gambar 2. 5 Komunikasi MQTT	10
Gambar 3. 1 Diagram Alir Metode Penelitian.....	11
Gambar 3. 2 Kebutuhan <i>Hardware</i>	14
Gambar 3. 3 Kebutuhan <i>Software</i>	14
Gambar 3. 4 Perancangan Sistem.....	15
Gambar 3. 5 Implementasi Sistem.....	16
Gambar 5. 1 Arsitektur Sistem.....	24
Gambar 5. 2 Diagram Alir <i>System</i>	25
Gambar 5. 3 <i>Flowchart</i> Perancangan Sistem Bluetooth Skenario 1.....	27
Gambar 5. 4 <i>Flowchart</i> Perancangan Sistem Bluetooth Skenario 1.....	28
Gambar 5. 5 <i>Flowchart Network bridge system</i> skenario 1.....	30
Gambar 5. 6 <i>Flowchart</i> Fungsi <i>Signal Handler</i> pada <i>Network bridge system</i>	31
Gambar 5. 7 <i>Flowchart Class Main Thread</i> pada <i>Network bridge system</i>	32
Gambar 5. 8 <i>Flowchart Fungsi on_message</i> pada <i>Network bridge system</i>	34
Gambar 5. 9 <i>Flowchart Network bridge system</i> skenario 2.....	35
Gambar 5. 10 <i>Flowchart Network bridge system main class</i> skenario 2.....	37
Gambar 5. 11 <i>Flowchart Network bridge system on_message</i> skenario 2	38
Gambar 5. 12 <i>Flowchart</i> Sistem MQTT pada perangkat IP Skenario 1.....	40
Gambar 5. 13 <i>Flowchart</i> Fungsi <i>Signal Handler</i> pada MQTT	41
Gambar 5. 14 <i>Flowchart</i> Fungsi <i>on_subscribe</i> pada MQTT.....	42
Gambar 5. 15 <i>Flowchart</i> Fungsi <i>on_message</i> pada MQTT	43
Gambar 5. 16 <i>Flowchart</i> Fungsi <i>on_connect</i> pada MQTT.....	44
Gambar 5. 17 <i>Flowchart</i> Fungsi <i>on_publish</i> pada MQTT	45
Gambar 5. 18 <i>Flowchart</i> Sistem MQTT pada perangkat IP Skenario 2.....	46
Gambar 5. 19 <i>Flowchart</i> Sistem MQTT <i>on_message</i> IP Skenario 2	47
Gambar 5. 20 Implementasi Perangkat Keras Sistem Bluetooth	52

Gambar 5. 21 Implementasi Program pada Sistem Bluetooth	53
Gambar 5. 22 Implementasi Perangkat Keras <i>Network bridge system</i>	54
Gambar 5. 23 Implementasi Program pada <i>Network bridge system</i>	55
Gambar 5. 24 Implementasi Perangkat Keras komunikasi MQTT	56
Gambar 5. 25 Implementasi Program pada Sistem MQTT	57
Gambar 6. 1 Tes Koneksi.....	59
Gambar 6. 2 Pengujian Sistem Bluetooth Skenario 1.....	60
Gambar 6. 3 Pengujian Sistem Bluetooth Skenario 2.....	60
Gambar 6. 4 <i>Network bridge system</i> Mengontrol data dari Perangkat Bluetooth Skenario 1.....	62
Gambar 6. 5 <i>Network bridge system</i> Mengontrol data dari Perangkat IP Skenario 2.....	62
Gambar 6. 6 Perangkat IP <i>subscribe</i> topik	63
Gambar 6. 7 Perangkat IP <i>Publish</i> topik balasan	64
Gambar 6. 8 Grafik <i>Delay</i> Bluetooth ke IP	66
Gambar 6. 9 Grafik <i>Delay</i> IP ke Bluetooth	66



DAFTAR LAMPIRAN

Lampiran 1 Source Code <i>Client</i> Bluetooth Skenario 1	69
Lampiran 2 Source Code <i>Network Bridge System</i> Skenario 1	70
Lampiran 3 Source Code <i>Client</i> IP Skenario 1	72
Lampiran 4 Source Code <i>Client</i> Bluetooth Skenario 2	74
Lampiran 5 Source Code <i>Network Bridge System</i> Skenario 2	75
Lampiran 6 Source Code <i>Client</i> IP Skenario 2	77



BAB 1 PENDAHULUAN

1.1 Latar belakang

Kemajuan teknologi informasi dari berbagai sisi mendorong sebuah inovasi dan revolusi teknologi untuk kedepannya. Teknologi informasi berkembang mulai dari sisi perangkat keras, perangkat lunak dan sistem arsitektur perangkatnya. Teknologi informasi adalah hasil rekayasa manusia terhadap proses penyampaian informasi dari bagian pengiriman ke penerima sehingga pengiriman informasi tersebut akan lebih cepat, lebih luas penyebarannya, dan lebih lama penyimpanannya (Christine Wibhowo, 2011). Perkembangan teknologi Informasi saat ini telah berkembang seiring adanya jaringan komputer seperti *Cloud Computing*, *Internet of Things* dan *Smart City* (I Putu Agus E.P., 2014). Proses perkembangan inilah muncul berbagai macam jenis teknologi informasi dari perangkat keras dan teknologi informasi dari perangkat lunak.

Salah satu perangkat teknologi informasi adalah bluetooth. Bluetooth merupakan salah satu teknologi pendukung *Internet of Things*. Bluetooth adalah sebuah standar teknologi khususnya sebuah protokol yang menyatakan cara dua perangkat berkomunikasi menggunakan gelombang pendek radio untuk mengirimkan data. Perangkat bluetooth berisi cip kecil sehingga dapat berkomunikasi dengan perangkat bluetooth lainnya. Teknologi bluetooth yang ada pada perangkat terus dikembangkan sampai saat ini. Salah satu contoh perangkat bluetooth adalah komputer *desktop*, PC Laptop, telepon genggam, PDA, *smartphone*, *headset*, mikrofon kamera digital dan lain sebagainya (Shelly Cashman Vermaat, 2007). Namun teknologi bluetooth mempunyai beberapa kelemahan yaitu jarak komunikasi masih dekat dan komunikasi antar perangkat yang masih personal area. Kelemahan ini yang menyebabkan penyebaran informasi bluetooth hanya antar perangkat yang terhubung dalam jarak komunikasi yang dekat. Tidak seperti perangkat bluetooth, perangkat IP mempunyai jangkauan komunikasi yang luas dan dapat berkomunikasi dengan berbagai perangkat melalui jaringan internet. Untuk mengatasi kelemahan dari perangkat bluetooth salah satu solusi yang ditawarkan adalah menghubungkan perangkat bluetooth dengan perangkat IP. Karena perangkat bluetooth dan perangkat IP mempunyai protokol komunikasi yang berbeda dibutuhkan *network bridge system* untuk menghubungkannya. *Network bridge system* dalam penelitian ini adalah sistem jaringan yang menjembatani komunikasi antar perangkat yang mempunyai protokol komunikasi yang berbeda.

Salah satu penelitian yang terkait pembangunan jaringan antara dua sistem yang berbeda adalah *Network bridge system* yang terdiri dari bagian jaringan *Zigbee* dan bagian jaringan UpnP yang menjelaskan desain *Network bridge system* yang dapat mengatur interoperasi dari *Zigbee* dan jaringan UpnP (Seong Joong Kim, et al 2011). Namun penelitian ini digunakan untuk perangkat *zigbee* namun belum menghubungkan perangkat non IP yang lain. Oleh karena itu dalam penelitian ini dibutuhkan *network bridge system* yang menjembatani komunikasi antara

perangkat bluetooth dan perangkat IP. Dengan adanya *Network bridge system* akan membantu permasalahan komunikasi berbagai perangkat teknologi informasi dengan fitur yang berbeda – beda dapat saling berhubungan, berkomunikasi dan berbagi data. Konsep dari *Network bridge system* dalam penelitian ini menghubungkan perangkat Bluetooth dan perangkat IP menggunakan protokol komunikasi bluetooth RFCOMM untuk perangkat bluetooth yang dihubungkan dengan protokol komunikasi MQTT *publisher Subscriber* untuk komunikasi perangkat IP. *Network bridge system* yang digunakan adalah mikrokomputer *Raspberry pi*. *Raspberry pi* digunakan untuk memprogram proses pertukaran data antar perangkat. Yang nantinya jika sistem ini dapat diimplementasikan maka akan mendukung perkembangan teknologi informasi dan jaringan komputer seperti *Internet of Things*.

1.2 Rumusan masalah

Berdasarkan latar belakang dapat dirumuskan masalah penelitian sebagai berikut.

1. Bagaimana Implementasi *Network bridge system* untuk menghubungkan perangkat IP dan perangkat bluetooth?
2. Bagaimana mekanisme pertukaran data antar perangkat IP dan perangkat bluetooth melalui *network bridge system*?
3. Bagaimana performa *Network bridge system* dalam menghubungkan perangkat IP dan perangkat bluetooth?

1.3 Tujuan

Tujuan Umum:

Mengimplementasikan *network bridge sytem* untuk menghubungkan perangkat berbasis IP dan perangkat non IP seperti Bluetooth.

Tujuan khusus:

1. Mengimplementasikan *network bridge sytem* untuk menghubungkan perangkat berbasis IP dan perangkat bluetooth.
2. Mengetahui mekanisme pertukaran data antar perangkat berbasis IP dan perangkat Bluetooth
3. Mengetahui performa *network bridge system* dalam komunikasi perangkat IP dan perangkat bluetooth.

1.4 Manfaat

1. Bagi Masyarakat

Manfaat dari penelitian ini bagi masyarakat adalah memanfaatkan perangkat dengan fitur yang berbeda – beda agar saling komunikasi atau perangkat teknologi informasi agar dapat berkomunikasi dan

bertukar data. Karena semakin berkembangnya teknologi sebuah inovasi yang cerdas dibutuhkan agar memaksimalkan fungsi dari perangkat pengolah informasi dapat berfungsi dengan sebaik-baiknya.

2. Bagi Penulis

Manfaat penelitian ini bagi penulis adalah menambah wawasan tentang berbagai jenis media jaringan beserta protokol komunikasi perangkat IP dan perangkat bluetooth. Dan mengembangkan minat dan ketrampilan dalam bidang komputer terutama tentang jaringan komputer.

1.5 Batasan masalah

Batasan masalah diperlukan agar ruang lingkup permasalahan dan pembahasan masalah tidak terlalu meluas. Batasan masalah yang dihadapi pada implementasi *Network bridge system* untuk menghubungkan perangkat IP dan Non IP adalah :

1. Perangkat IP yang digunakan adalah Komputer dengan fitur *wifi*
2. Perangkat Non IP yang digunakan adalah Komputer dengan fitur Bluetooth
3. *Bridge* penghubung yang digunakan adalah *Raspberry pi 2* model B
4. *Client* bluetooth yang dapat berkomunikasi adalah bluetooth *client* yang sudah terdaftar pada *Network bridge system*
5. IPv6 yang digunakan adalah *IPv6 unicast*
6. Protokol komunikasi yang digunakan pada perangkat IP adalah MQTT
7. Bahasa pemrograman yang digunakan pada sistem adalah python

1.6 Sistematika pembahasan

Sistematika pembahasan skripsi yang disusun ini akan dibahas pada bab-bab yang akan diuraikan di bawah ini :

BAB I PENDAHULUAN

Menguraikan tentang latar belakang masalah, merumuskan inti permasalahan dan menentukan tujuan untuk kegunaan penelitian yang kemudian diikuti dengan pembatasan masalah, asumsi metodologi penelitian serta sistematika penulisan.

BAB II KAJIAN PUSTAKA

Bab ini berisi tinjauan pustaka dari penelitian sebelumnya dan dasar-dasar teori untuk digunakan selanjutnya pada bagian pembahasan.

BAB III METODOLOGI PENELITIAN

Dalam bab ini akan membahas studi literatur, analisa kebutuhan, perancangan sistem dan implementasi *Network bridge system* untuk komunikasi perangkat berbasis IP dan Perangkat non IP Bluetooth.

BAB IV REKAYASA PERSYARATAN

Membahas tentang semua kebutuhan atau *requirements systems*, serta kebutuhan fungsional dan non fungsional sistem.

BAB V PERANCANGAN DAN IMPLEMENTASI

Bab ini berisi tentang perancangan dan implementasi *Network bridge system* untuk komunikasi perangkat berbasis IP dan Perangkat non IP Bluetooth

BAB VI PENGUJIAN DAN ANALISIS

Bab ini berisi tentang pengujian dan analisa dari bab sebelumnya dengan menganalisa fungsionalitas sistem *Network bridge system* dalam komunikasi data dan analisa seberapa jauh komunikasi bluetooth dalam sistem ini.

BAB VII KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari implementasi *Network bridge system* dengan perangkat berbasis IP dan Perangkat non IP Bluetooth. Untuk lebih meningkatkan hasil akhir yang lebih baik maka diberikan juga saran-saran untuk perbaikan serta penyempurnaan proyek akhir ini.

1.7 Jadwal penelitian

Tabel 1. 1 Jadwal Pelaksanaan Penelitian

Kegiatan	Bulan						
	Desember 2015	Januari 2016	Februari 2016	Maret 2016	April 2016	Mei 2016	Juni 2016
Studi Literatur							
Analisa Kebutuhan Sistem							
Perancangan Sistem							
Pembelian Kebutuhan Sistem							
Pembuatan Program Komunikasi MQTT							
Pembuatan Program Komunikasi Bluetooth							
Pembuatan Program <i>Network bridge system</i>							
Implementasi Sistem							

BAB 2 LANDASAN KEPUSTAKAAN

Pada penelitian ini, pustaka yang penulis jadikan referensi adalah beberapa jurnal referensi tentang konsep *Network bridge system* dan penerapan *Network bridge system* yaitu *UPNP Network Bridge for supporting Interoperability through Non-IP Channels* peneliti Young Min Baek et al , *Network bridge system for Interoperation of ZigBee-UpnP* peneliti Seong-Joong Kim,et al. Untuk referensi tentang dasar teori sebagai pengetahuan tentang beberapa komponen pendukung pembuatan sistem dan teknologi yang digunakan seperti mikrokomputer *Raspberry pi* dan perangkat bluetooth. Serta beberapa referensi lain seperti protokol komunikasi MQTT, Ipv6 dan pengetahuan umum tentang *Bridge Network*.

2.1 Tinjauan pustaka

Pada penelitian ini tinjauan pustaka diambil dari beberapa penelitian yang pernah ada sebelumnya. Tinjauan pustaka diambil dari beberapa jurnal resmi yang dijadikan pedoman dalam penelitian ini. Berikut tabel perbandingan beberapa jurnal terkait.

Tabel 2. 1 Tinjauan Pustaka

No.	Nama Penulis, Tahun dan Judul	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1.	Seong-Joong; Hae-Moon Seo; Woo-Chool Park; Seong-Dong, 2011, <i>Network bridge system for Interoperation of Zigbee-UPNP Network</i>	Melakukan penerapan <i>network bridge system</i> untuk komunikasi antar perangkat	Menjelaskan implementasi <i>network bridge system</i> untuk interoperasi perangkat zigbee pada jaringan Upnp	Mengimplementasikan <i>network bridge system</i> untuk komunikasi perangkat bluetooth dan perangkat IP
2.	Young Min Baek; Sang Chul Ahn; Young-Moo Kwon, 2010, <i>UpnP Network Bridge for Supporting Interperability through Non-IP Channels</i>	Melakukan penerapan <i>network bridge</i> untuk menghubungkan 2 sistem	Menjelaskan implementasi <i>network bridge</i> untuk mendukung operasi antar jaringan Upnp melalui jaringan non IP	Mengimplementasikan <i>network bridge system</i> untuk mendukung penyebaran informasi perangkat bluetooth yang lebih luas

2.2 Dasar teori

Pada bagian dasar teori menjelaskan tentang beberapa definisi tentang komponen dan aplikasi pendukung penelitian ini. Beberapa dasar teori yang dijelaskan adalah *Bridge Network*, Bluetooth, *Raspberry pi 2*, Ipv6 dan MQTT *Publisher Subscriber*.

2.2.1 Network bridge system

Arsitektur perangkat keras mempunyai metode komunikasi yang berbeda – beda dengan jumlah yang sangat banyak. *Network bridge system* merupakan salah satu cara untuk mengatasi permasalahan tersebut. *Network bridge system* merupakan sistem atau alat perantara untuk menghubungkan atau mengontrol beberapa perangkat yang mempunyai arsitektur yang sama ataupun berbeda. *Network bridge system* ini dapat diprogram dengan berbagai teknologi komunikasi sesuai kebutuhan. Dalam mengimplementasikan *network bridge system* dapat menggunakan bermacam – macam mikrokontroler dan mikrokomputer. Berikut contoh penerapan *network bridge system* seperti pada gambar 2.1

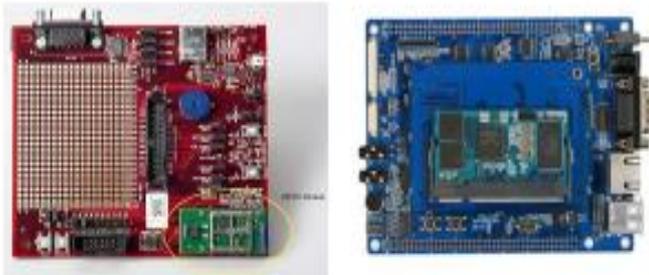


Fig. 3. (a) Zigbee Experiment board (b) network bridge board

Gambar 2. 1 Contoh Network Bridge

Sumber : (Seong-Joong et al, 2011)

Penjelasan gambar 2.1 adalah bagian dari arsitektur *network bridge system* terdiri dari beberapa bagian sesuai dengan penerapan dan kegunaannya. Bagian pertama terdapat modul RX/TX sebagai *input output* data yang akan diproses. Dalam bagian ini *input output* modul RX/TX dapat berupa transmisi sinyal data yang dikirim dari perangkat tujuan. Bagian ini bertanggung jawab untuk mengendalikan pengiriman dan penerimaan data dari perangkat asal ke perangkat tujuan. Bagian kedua terdapat proses translator. Dalam proses translator ini data yang dikirim akan dipetakan sesuai dengan format data yang dikirim. Data yang dikirim dari dua perangkat yang mempunyai perbedaan arsitektur komunikasi akan dipetakan berdasarkan formatnya. Agar data yang dikirim dapat terbaca oleh perangkat penerima. Bagian ketiga adalah *management*. Pada bagian ini setelah data diterima akan diproses sesuai dengan kebutuhan. Data diolah untuk selanjutnya dikirim ke perangkat tujuan.

2.2.2 Bluetooth

Teknologi bluetooth adalah standar global jaringan nirkabel yang diterapkan pada *Internet of Things* (IoT). Bluetooth diciptakan pada tahun 1994. Bluetooth merupakan teknologi komunikasi wireless yang digunakan sebagai alternatif nirkabel untuk bertukar data menggunakan radio transmisi. Salah satu aplikasi paling populer yang memanfaatkan teknologi Bluetooth adalah headset audio dan streaming musik dari ponsel atau tablet. Aplikasi ini menggunakan teknologi Bluetooth yang disebut BR/EDR (*Bit Rate/Enhanced Data Rate*) yang dioptimalkan untuk aliran data berkualitas tinggi (music) dengan cara menghemat daya. Bluetooth adalah sebuah teknologi wireless yang menyediakan komunikasi data dan suara dengan jarak jangkauan yang terbatas. Beroperasi dalam pita frekuensi 2,4 GHz yang mampu menyediakan layanan komunikasi data dan suara secara real-time dengan menggunakan frekuensi hopping tranceiver.

Dalam perkembangan bluetooth saat ini, perusahaan bluetooth telah mengeluarkan berbagai macam teknologi dan aplikasinya. Pengembangan teknologi bluetooth diawasi oleh Bluetooth Special Interest Group (SIG) dan secara teratur diperbaharui dan ditingkatkan untuk memenuhi perkembangan teknologi dan kebutuhan pasar. Bluetooth mempunyai 2 jenis teknologi aplikasi pada perangkat teknologi. Jenis aplikasi teknologi bluetooth yang umum digunakan adalah bluetooth Basic Rate/ Enhanced Data Rate yang diadopsi sebagai versi 2.0/2.1 dan bluetooth dengan low energy (LE) yang diadopsi sebagai versi 4.0/4.1/4.3. Perbedaan bluetooth BR/EDR dan LE adalah bluetooth BR/EDR menetapkan komunikasi jarak pendek, koneksi nirkabel yang berkelanjutan dan bluetooth LE memungkinkan komunikasi jarak jauh sehingga ideal untuk penerapan teknologi *Internet of Things* (IOT).

Dalam pemrograman komunikasi pada perangkat bluetooth terdapat beberapa protokol komunikasi yang digunakan seperti L2CAP dan RFCOMM. L2CAP atau kepanjangan dari *Link Layer Common Acces Protokol* merupakan protokol komunikasi bluetooth yang mirip dengan UDP. L2CAP menyediakan koneksi berorientasi 3 protokol yang mengirmkan datagram dengan panjang maksimum yang tetap. Ukuran paket maksimum *default* adalah 672 byte. L2CAP dapat dikonfigurasi untuk berbagai tingkat kehandalan. L2CAP mempunyai 3 peraturan dalam komunikasi yaitu tidak akan mentransmit data ulang saat proses pengiriman, transmisi ulang dilakukan saat paket selesai dikirim atau seluruh paket gagal dikirim, drop paket dan peralihan data ke antrian dilakukan setelah batas waktu tertentu yaitu 0 sampai 1279 milidetik. Sedangkan RFCOMM adalah protokol komunikasi bluetooth yang mirip dengan TCP. Dalam protokol RFCOMM jika sebagian dari data tidak terkirim dalam waktu waktu tertentu maka sambungan akan diakhiri dan kesalahan akan dilaporkan. RFCOMM memungkinkan pemberian nomor *port* sampai 30.

Requirement	Internet	Bluetooth
Reliable, streams-based	TCP	RFCOMM
Reliable, datagram	TCP	RFCOMM or L2CAP with infinite retransmit
Best-effort, datagram	UDP	L2CAP (0-1279 ms retransmit)

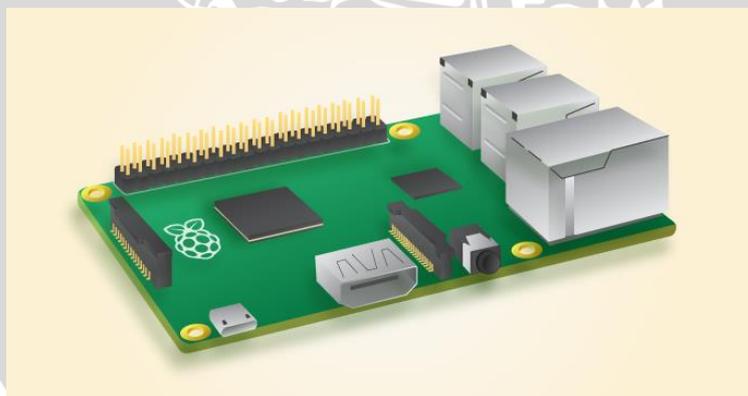
Gambar 2. 2 Perbandingan L2CAP dan RFCOMM

Sumber : (Albert Huang et al, 2005)

Pada gambar 2.2 terlihat perbandingan protokol L2CAP dan RFCOMM pada komunikasi bluetooth dengan membandingkan dengan protokol yang ada di internet. Terlihat perbedaan kebutuhan dalam komunikasi. Dalam kebutuhan sistem yang *reliable* berdasarkan *streaming* pada internet menggunakan protokol TCP dan bluetooth RFCOMM. Untuk best-effort pada datagram pada internet menggunakan protokol UDP dan L2CAP pada bluetooth.

2.2.3 Mikrokomputer *Raspberry pi 2*

Mikrokomputer *Raspberry pi* adalah salah satu jenis mikrokomputer dengan harga murah yang bisa digunakan seperti komputer *Raspberry pi* dapat dihubungkan ke kabel monitor, TV dan mouse atau keyboard. *Raspberry pi* merupakan komputer kecil dengan berat 45 gram. *Raspberry pi* telah mengeluarkan berbagai jenis produk untuk jenis *Raspberry pi*. Salah satu jenis raspberry adalah *Raspberry pi 2* model B yang digunakan dalam penelitian ini.



Gambar 2. 3 *Raspberry pi 2* Model B

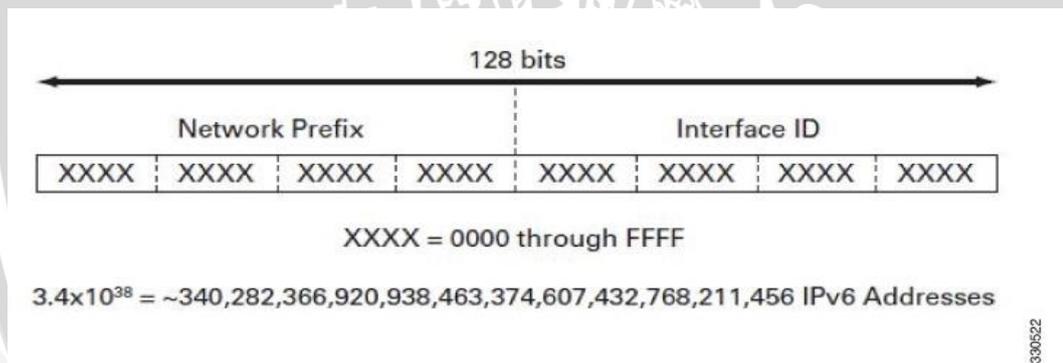
Sumber: (Raspberrypi, 2016)

Seperti pada gambar 2.3 yaitu contoh desain dari *Raspberry pi 2* Model B. *Raspberry pi 2* Model B merupakan generasi kedua dari *Raspberry pi* menggantikan *Raspberry pi 1* Model B+ pada february 2015. *Raspberry pi 2* mempunyai arsitektur sistem dengan dukungan processor quad core ARM Cortex-A7 dengan kecepatan 900 MHz. *Raspberry pi 2* model B dibekali memory 1GB LPDDR2 SDRAM.

2.2.4 IPv6

IPv6 atau kepanjangan dari Internet Protocol Version 6 adalah generasi selanjutnya dari IPv4 untuk dapat terhubung ke internet. IPv6 didesain untuk menyediakan beberapa kelebihan daripada IPv4. IPv4 dan IPv6 mendefinisikan protokol layer *network*, contohnya bagaimana data dikirim dari satu komputer ke komputer lain melalui paket-switched *network* seperti internet. IPv6 adalah jenis pengalamatan di dalam sebuah jaringan yang digunakan di dalam protokol jaringan TCP/IP. IPv6 merupakan protokol internet generasi terbaru yang dirancang untuk menggantikan internet protokol saat ini yaitu IPv4. Salah satu manfaat utama dari protokol internet IPv6 dibandingkan IPv4 adalah memiliki ruang alamat lebih besar yang tidak akan habis hingga beberapa masa kedepan.

Pada dasarnya IPv6 dikembangkan untuk mengantisipasi kelangkaan IP address yang disediakan oleh IPv4. Karena IPv6 ini tidak lagi menggunakan 32 bit biner tetapi 128 bit biner, sehingga alamat yang mampu disediakan yaitu 2^{128} atau sebesar 3×10^{38} alamat. Selain itu juga dilakukan perubahan dalam penulisannya yaitu 128 bit alamat dipisahkan menjadi masing-masing 16 bit yang tiap bagian dipisahkan dengan “:” dan dituliskan dengan bilangan hexadesimal. Pengalamatan pada ipv6 dibagi menjadi *network prefix* dan *Interface ID*. Pengalamatan ipv6 dijelaskan pada gambar 2.4



Gambar 2. 4 Format Ipv6

Sumber : (Cisco, 2016)

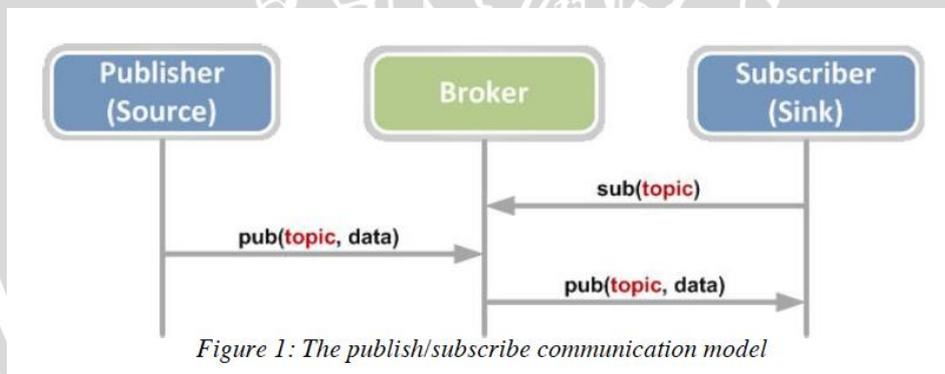
Dalam pengalamatannya alamat IPV6 diklasifikasikan menjadi tiga kategori yaitu :

1. Alamat unicast yang bertindak sebagai pengidentifikasi untuk satu antarmuka. Paket IPV6 dikirim ke alamat unicast ke interface yang dikenali oleh alamat tersebut.
2. Alamat multicast yang bertindak sebagai pengidentifikasi untuk kelompok atau lebih dari satu antarmuka yang dimiliki oleh node yang berbeda. Paket IPV6 akan dikirim ke beberapa interface.
3. Alamat Anycast yang bertindak sebagai pengidentifikasi untuk satu set antarmuka yang dimiliki oleh node yang berbeda. Paket IPV6 akan dikirim khusus untuk alamat anycast ke salah satu antarmuka yang dikenali oleh alamat.

2.2.5 MQTT *Publisher Subscriber*

MQTT singkatan dari *Message Queuing Telemetry Transport*. MQTT adalah *publish* atau *subscribe* sistem yang sederhana dan mempunyai protokol pesan yang mudah. Dirancang untuk perangkat yang dibatasi oleh bandwidth yang rendah, latency yang tinggi dan jaringan yang tidak stabil. Prinsip desain MQTT adalah meminimalkan bandwidth jaringan, kebutuhan daya dari perangkat dan memastikan jaminan pengiriman pesan. Dari prinsip tersebut MQTT merupakan protokol ideal untuk komunikasi “machine to machine (M2M)” atau “*Internet of Things*” karena perangkat di dunia dapat saling terhubung dimana bandwidth dan daya perangkat pada level premium.

MQTT pada dasarnya adalah protokol *publish/subscribe*. Hal ini memungkinkan *client* yang terhubung sebagai *publisher*, *Subscriber* atau keduanya. *Client* harus terhubung ke *broker* untuk mendapatkan topik atau pesan yang diinginkan. *Broker* adalah sistem yang menangani semua pesan yang lewat. Data yang dikirim dan diterima dalam berkomunikasi berupa topik. Topik akan dikirim dari sisi *publisher* ke *broker*. *Subscriber* akan mengambil topik ke *broker* sesuai kebutuhan. Dalam protokol MQTT topik yang dikirim dipetakan menjadi path file. Hal tersebut dilakukan untuk memastikan bahwa pesan akan dikirim ke penerima yang benar. Pada gambar 2.5 dijelaskan bagaimana komunikasi menggunakan protokol MQTT.



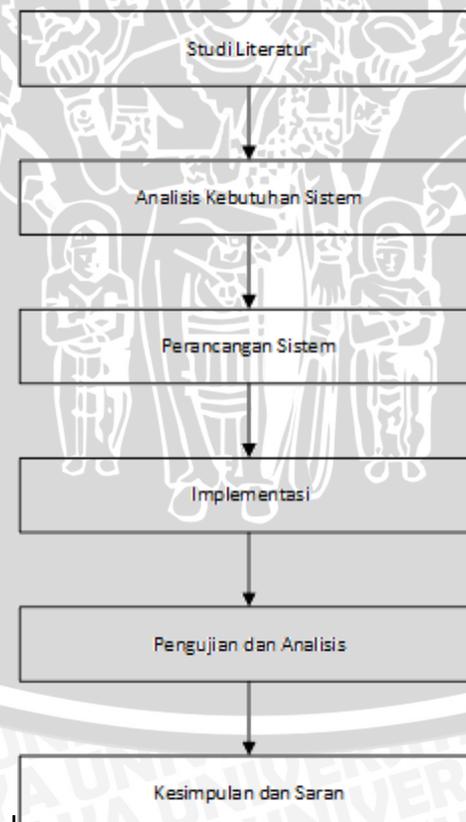
Gambar 2. 5 Komunikasi MQTT

Sumber : (Tiziano Madotti, 2012)

BAB 3 METODOLOGI

Penelitian ini dimulai dengan studi literatur yang terkait dengan pembuatan skripsi, studi literatur berasal dari jurnal, buku dan website resmi digunakan untuk mendukung tinjauan pustaka dan dasar teori. Penelitian ini bersifat implementatif karena penelitian ini bertujuan untuk mengimplementasikan sistem *bridging network* untuk menghubungkan perangkat IP dan Bluetooth dengan menggunakan konsep komunikasi protokol MQTT. Dalam tahap pembuatan awal penelitian ini dibutuhkan analisa kebutuhan baik fungsional maupun non fungsional dan blok diagram tahap alur kerja sistem. Blok diagram digunakan untuk menjelaskan secara umum alur kerja pada sebuah sistem secara keseluruhan. Lalu tahap perancangan sistem yang di desain sesuai dengan kebutuhan yang sudah dibuat. Tahap implementasi dilakukan setelah tahap perancangan telah selesai. Implementasi dilakukan dengan beberapa tahap pembuatan komunikasi antar perangkat sampai semua perangkat dapat terhubung yang selanjutnya dilakukan pengujian fungsionalitas sistem berdasarkan analisa kebutuhan sistem.

Pada bagian ini dijelaskan mengenai metodologi yang dilakukan dalam penelitian. Diagram alir metodologi penelitian ditunjukkan pada Gambar 3.1.



Gambar 3. 1 Diagram Alir Metode Penelitian

3.1 Studi Literatur

Studi literatur mempelajari mengenai penjelasan literatur pustaka dan dasar teori yang digunakan untuk menunjang penulisan penelitian. Teori-teori pendukung tersebut diperoleh dari buku, artikel, jurnal, *e-book*, dan dokumentasi project. Teori-teori pendukung tersebut meliputi:

1. *Internet of Things (IoT)*

Studi literatur terkait dengan IoT dibutuhkan untuk mempelajari konsep tentang IoT secara umum. Dalam mengkaji konsep IoT akan didapatkan konsep bagaimana IoT diimplementasikan pada suatu jaringan. Serta konsep dasar *things* dapat bekerja di dalam jaringan sehingga dapat dikatakan teknologi dengan penerapan konsep IoT. Studi literatur ini akan mempelajari bagaimana melakukan komunikasi pertukaran pesan atau informasi dalam IoT.

2. *Network Bridge*

Studi literatur terkait dengan *Network Bridge* dilakukan dengan mencari beberapa jurnal ilmiah yang terkait bagaimana *network Bridge* di implementasikan dalam suatu jaringan. Pada tahapan ini, dipelajari bagaimana pengembangan dan penerapan dari *network Bridge* dalam teknologi informasi.

3. Bluetooth

Studi literatur terkait dengan Bluetooth dilakukan dengan mencari beberapa jurnal ilmiah yang terkait dengan penerapan bluetooth dalam suatu sistem. Pada tahapan ini, dipelajari bagaimana arsitektur komunikasi bluetooth dan bagaimana memprogram perangkat bluetooth agar dapat melakukan pertukaran informasi sesuai dengan yang diinginkan.

4. Mikrokomputer *Raspberry pi*

Studi literatur terkait dengan *Raspberry pi* dilakukan dengan mengakses beberapa website terpercaya terkait penggunaan *Raspberry pi*. Pada tahapan ini, dipelajari bagaimana menggunakan *Raspberry pi*, komponen – komponen yang dapat dipakai dan memprogram *Raspberry pi* agar dapat bekerja sesuai dengan yang diinginkan.

5. Ipv6

Studi literatur terkait dengan Ipv6 dilakukan dengan mencari beberapa jurnal terkait dengan penerapan ipv6. Kajian ini dibutuhkan agar ipv6 dapat diterapkan pada sistem. Jika nantinya arsitektur teknologi dalam perkembangan teknologi informasi sudah menerapkan ipv6. Sistem ini tetap dapat digunakan.

6. MQTT

Studi literatur terkait dengan konsep protokol komunikasi MQTT dilakukan dengan mencari beberapa jurnal ilmiah dan website terpercaya.

Pada tahapan ini, dipelajari bagaimana konsep komunikasi dan pertukaran pesan atau informasi dengan menggunakan protokol MQTT. Serta mempelajari konsep dasar 3 bagian protokol MQTT yaitu *publisher*, *Subscriber*, dan *broker* dalam jaringan.

3.2 Rekayasa Kebutuhan

Analisa Kebutuhan bertujuan untuk melakukan analisis pada beberapa kebutuhan yang diperlukan sistem pada penelitian. Analisa kebutuhan pada penelitian ini akan dijelaskan sebagai berikut.

3.2.1 Kebutuhan Pengguna

Kebutuhan pengguna dalam penelitian ini adalah sistem dapat melakukan pengiriman informasi dari *client* IP menuju *client* Non-IP dan sebaliknya melalui *Bridge network System*.

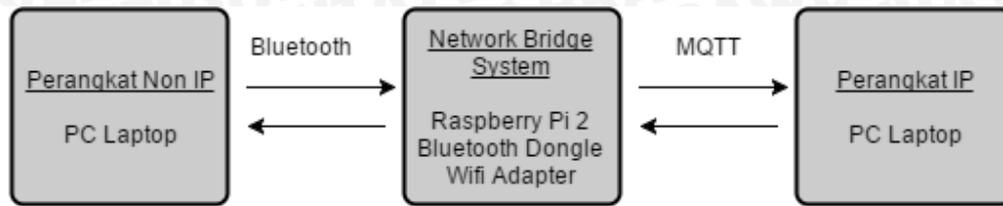
3.2.2 Kebutuhan Sistem

Kebutuhan sistem dalam penelitian ini berdasarkan perancangan sistem adalah:

- Perangkat bluetooth akan melakukan proses pairing terlebih dahulu ke *Bridge System* untuk mendaftarkan ID Bluetooth
- Bluetooth *Bridge Raspberry pi* akan menerima data dari perangkat bluetooth
- Bluetooth *Bridge Raspberry pi* akan menforwarding data yang dikirim dari perangkat bluetooth ke perangkat IP
- Perangkat IP akan melakukan koneksi ke *Bridge raspberry pi* dan menerima data dari *Bridge Bridge System*
- Data yang diterima di akan dibalas oleh *client* IP dengan pesan bahwa data yang dikirim sudah terima, data dikirim ke *Bridge* sistem
- Bridge System* akan menerima data dari perangkat IP
- Bridge Raspberry pi* akan menforwarding data dari perangkat IP ke perangkat bluetooth
- Perangkat bluetooth akan menerima pesan balasan dari *Bridge System*.

3.2.2 Kebutuhan Hardware

Dalam perancangan dan implementasi pada sistem dibutuhkan hardware untuk mewujudkan sistem secara keseluruhan. Beberapa hardware yang dibutuhkan ditunjukkan pada gambar 3.2



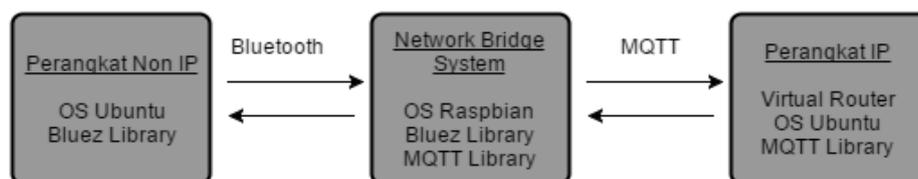
Gambar 3. 2 Kebutuhan Hardware

Penjelasan terkait Gambar 3.2 tentang kebutuhan hardware dijelaskan sebagai berikut.

1. Perangkat Non-IP
 - a. PC Laptop/ Desktop dengan fitur Bluetooth berfungsi sebagai *client* dari perangkat Non IP yang didalamnya terdapat fitur bluetooth. *Client* akan diprogram agar dapat mengirimkan dan menerima data melalui komunikasi bluetooth
2. Network Bridge Sistem
 - a. *Raspberry pi* berfungsi sebagai alat implementasi dari *Network bridge system* yang diprogram agar dapat mengontrol pesan untuk diteruskan ke perangkat IP atau Non IP
 - b. Bluetooth Dongle berfungsi sebagai penerima dan pengirim data yang ada pada *Bridge System* untuk komunikasi Bluetooth
 - c. Wifi Adapter berfungsi sebagai penerima dan pengirim pesan yang ada pada *Bridge System* untuk komunikasi MQTT dengan Ipv6
3. Perangkat IP
 - a. PC Laptop/Desktop dengan fitur wifi berfungsi sebagai *client* dari perangkat IP yang dapat berkomunikasi dalam jaringan IP yang sama dengan *Bridge System*. *Client* akan diprogram agar dapat mengirimkan dan menerima data melalui komunikasi MQTT dengan ipv6

3.2.2 Kebutuhan software

Dalam perancangan dan implementasi pada sistem dibutuhkan hardware untuk mewujudkan sistem secara keseluruhan. Beberapa software yang dibutuhkan ditunjukkan pada gambar 3.3



Gambar 3. 3 Kebutuhan Software

Penjelasan terkait Gambar 3.3 tentang kebutuhan hardware dijelaskan sebagai berikut.

1. Perangkat Non-IP

- a. Sistem operasi Linux Ubuntu dibutuhkan untuk konfigurasi bluetooth dan pemrograman bluetooth untuk bertukar pesan antar perangkat bluetooth.
- b. Bluez library dibutuhkan untuk menjalankan fungsi bluetooth secara keseluruhan saat menulis kode program.

2. Network Bridge Sistem

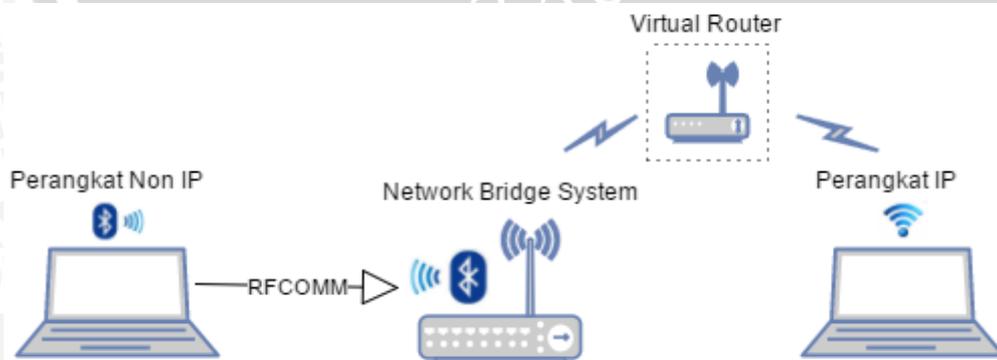
- a. Sistem Operasi Raspbian Wheezy dibutuhkan untuk mengoperasikan Raspberry sesuai kebutuhan agar dapat diprogram sesuai dengan keinginan.
- b. Bluez library dibutuhkan untuk menjalankan menjalankan fungsi bluetooth secara keseluruhan saat menulis kode program.
- c. MQTT library dibutuhkan untuk menjalankan menjalankan fungsi MQTT secara keseluruhan saat menulis kode program.

3. Perangkat IP

- a. Sistem operasi Linux Ubuntu dibutuhkan untuk konfigurasi protokol MQTT dan pemrograman MQTT untuk bertukar pesan antar perangkat IP.
- b. MQTT library digunakan untuk menjalankan menjalankan fungsi MQTT secara keseluruhan saat menulis kode program.

3.3 Perancangan sistem

Perancangan sistem dilakukan agar pembuatan sistem pada penelitian ini dapat terencana dan terstruktur dengan baik. Perancangan dilakukan apabila seluruh kebutuhan sistem telah terpenuhi. Perancangan sistem pada penelitian ini dijelaskan pada gambar 3.4



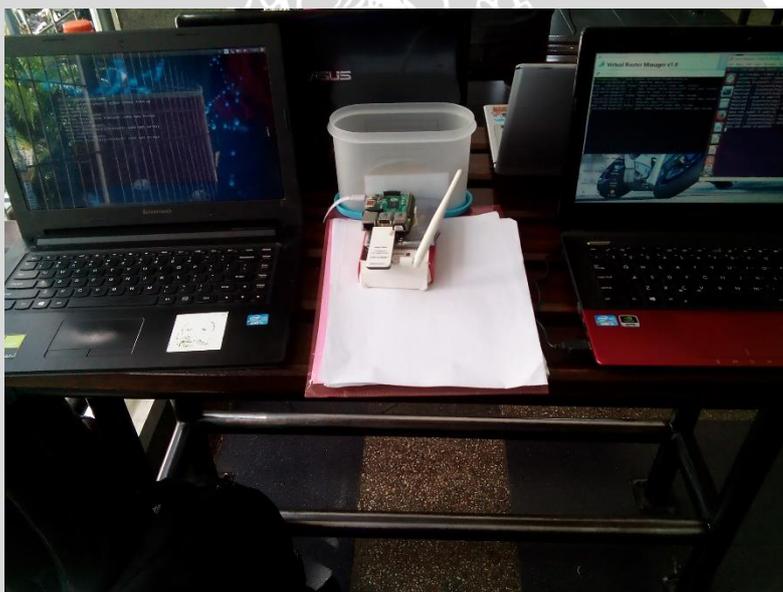
Gambar 3. 4 Perancangan Sistem

Penjelasan perancangan sistem pada gambar 3.4 sebagai berikut.

1. Perangkat non IP dengan fitur bluetooth bertindak mengirimkan dan menerima pesan dari perangkat IP melalui *Bridge System* menggunakan komunikasi bluetooth dengan protokol komunikasi RFCOMM
2. *Bridge System* yang bertindak mengontrol pengiriman pesan dari perangkat IP dan Non IP
3. Virtual Router bertindak sebagai pembagi sinyal jaringan IP agar dapat saling terhubung dalam jaringan yang sama
4. Perangkat IP bertindak mengirimkan dan menerima pesan dari perangkat Non IP melalui *Bridge System* menggunakan komunikasi MQTT

3.4 Implementasi

Implementasi *System* dilakukan dengan memperhatikan pada perancangan *System* yang telah dibuat sebelumnya. Implementasi sistem dijelaskan pada Gambar 3.5 berikut.



Gambar 3. 5 Implementasi Sistem

Beberapa tahapan implementasi sebagai berikut.

1. *Network bridge system*

Implementasi *Network bridge system* yang digunakan adalah *Raspberry pi 2* dengan tambahan bluetooth dongle dan wifi adapter. *Bridge System* berfungsi untuk menghubungkan perangkat IP dan Non IP agar dapat berkomunikasi. Lalu membuat program pada *Raspberry pi* untuk melakukan *forwading* data untuk komunikasi perangkat IP dan perangkat Non IP.

2. Perangkat IP

Implementasi perangkat IP yang digunakan adalah PC Laptop. PC Laptop dengan sistem operasi Linux Ubuntu dapat terhubung dengan

Network *Bridge System* dengan jaringan yang sama. Lalu membuat suatu program agar dapat saling bertukar pesan menggunakan komunikasi MQTT. Pengalamatan IP yang digunakan adalah Ipv6 *unicast*.

3. Perangkat Non IP

Implementasi perangkat Non IP yang digunakan adalah PC Laptop yang mempunyai fitur bluetooth. Perangkat Non IP menggunakan sistem operasi Linux Ubuntu dapat terhubung dengan *Bridge System* dengan komunikasi Bluetooth. Tahapan selanjutnya membuat suatu program agar dapat saling bertukar pesan menggunakan komunikasi bluetooth.

3.5 Pengujian dan analisa

Pengujian *System* pada penelitian ini dilakukan untuk menunjukkan bahwa pengembangan dari *System* ini telah mampu bekerja sesuai dengan spesifikasi dari keutuhan yang melandasinya. Beberapa skenario pengujian yang dilakukan sebagai berikut.

- a. Pengujian *System client* Bluetooth dapat mengirimkan data
- b. Pengujian *System network bridge system* dapat menerima data dan meneruskan data ke *client IP*
- c. Pengujian *System client IP* dapat menerima dan mengembalikan data ke *Bridge System*
- d. Pengujian *System network bridge system* dapat menerima data dan meneruskan data ke *client bluetooth*
- e. Pengujian sistem bluetooth *client* dapat menerima data
- f. Pengujian pengukuran jarak komunikasi bluetooth

3.6 Kesimpulan

Kesimpulan pada penelitian ini didapatkan apabila seluruh pengujian sistem telah selesai dilakukan. Sehingga didapatkan kesimpulan secara keseluruhan.

BAB 4 REKAYASA KEBUTUHAN

4.1 Pendahuluan

4.1.1 Tujuan

Bab ini bertujuan untuk menyajikan penjelasan rinci tentang sistem *network bridge system* untuk komunikasi perangkat non IP Bluetooth dan perangkat IP berdasarkan format Software Requirement Specification (SRS) IEEE 830. Bab ini menjelaskan tujuan dan fitur dari sistem, antarmuka sistem, apa yang akan dilakukan sistem, batasan bagaimana sistem akan beroperasi serta bagaimana sistem akan bereaksi dengan rangsangan eksternal. Bab ini ditujukan untuk stakeholder atau pemangku kepentingan dan para pengembang sistem, untuk selanjutnya diusulkan sebagai persetujuan skripsi.

4.1.2 Ruang Lingkup

Tujuan dari sistem ini untuk menghubungkan perangkat berbasis IP dan perangkat Non IP melalui *Network bridge system*. *Network bridge system* ini mampu mengontrol pesan keluar masuk antar perangkat. *Network bridge system* dapat menangani komunikasi bluetooth dengan protokol komunikasi RFCOMM antar perangkat bluetooth, menangani komunikasi perangkat IP dengan protokol MQTT sebagai salah satu protokol dalam konsep *Internet of Things (IOT)* dan menghubungkan 2 protokol komunikasi tersebut agar dapat saling bertukar pesan atau data. Pembuatan sistem ini ditujukan kepada penguji, developer dan pengguna sehingga kedepannya sistem ini dapat dikembangkan dan dapat digunakan oleh masyarakat luas dalam era perkembangan teknologi saat ini. Sistem ini dapat diterapkan dan dikembangkan dalam smart home, industry dan embedded technology karena penggunaan protokol MQTT dengan ipv6 yang mendukung konsep *Internet of Things (IOT)*.

4.1.3 Istilah

Tabel 4. 1 Tabel Istilah

Term	Definition
<i>Internet of Things (IoT)</i>	Paradigma baru yang bertujuan menciptakan suatu jaringan komunikasi yang menghubungkan “segalanya” dengan daya penyimpanan minimal dan kemampuan komputasi, perangkat yang saling terhubung dapat berkomunikasi kapanpun, dimanapun dan dalam bentuk apapun.
Perangkat IP	Perangkat yang menggunakan Internet Protokol dalam penggunaan dan penyebaran teknologi informasi.
Perangkat Non IP	Perangkat yang menggunakan sinyal radio transmisi

MQTT	Protokol komunikasi berbasis <i>broker</i> , <i>publisher</i> dan <i>Subscriber</i> yang didesain untuk perangkat dengan daya proses dan penyimpanan yang lemah .
<i>Publisher</i>	Pengirim pesan berbasis topik menuju <i>broker</i> dalam komunikasi MQTT
<i>Subscriber</i>	Penerima atau peminta informasi berbasis topik menuju <i>broker</i> dalam komunikasi MQTT
<i>Broker</i>	MQTT server yang mengkoleksi informasi dari <i>publisher</i> dan meneruskan informasi yang diminta pada <i>Subscriber</i>
<i>Client</i>	Pengguna layanan dalam suatu sistem
Bluetooth	Teknologi komunikasi wireless yang digunakan sebagai alternatif nirkabel untuk bertukar data menggunakan radio transmisi pada frekuensi 2,4 Ghz
Ipv6	Internet protokol versi 6 yang mempunyai pengalamatan sebesar 128 bit
RFCOMM	Salah satu protokol komunikasi bluetooth yang menjamin penyampaian data seperti protokol TCP

4.1.4 Referensi

IEEE.IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998

4.1.5 Sistematika

Untuk mempermudah melihat dan mengetahui pembahasan yang ada, maka diperlukan sistematika sebagai kerangka dan pedoman dalam penulisan SRS ini. Dokumen SRS ini dibagi menjadi 3 bagian sebagai berikut

1. Pendahuluan

Pada bagian ini akan dijelaskan tujuan, ruang lingkup, istilah, referensi dan sistematika penulisan.

2. Deskripsi Umum

Pada bagian ini akan dijelaskan perspektif produktif, fungsi produk, karakteristik pengguna, batasan – batasan, asumsi dan ketergantungan.

3. Spesifikasi Kebutuhan

Pada bagian ini akan dijelaskan kebutuhan fungsional, kebutuhan antarmuka eksternal, dan kebutuhan non fungsional

4.2 Deskripsi Umum

4.2.1 Perspektif Produk/Sistem

Sistem ini dapat bekerja dan beroperasi apabila keseluruhan dari setiap sesi *System* telah dijalankan, mulai dari perangkat Non IP sebagai pengirim pesan awal berupa pesan string yang terhubung dengan *Bridge System*. *Network bridge system* yang akan menerima pesan dari perangkat Non IP lalu meneruskan pesan ke perangkat IP. Perangkat IP akan menerima pesan dari *Bridge System* dan memberikan balasan pesan bahwa pesan sudah diterima. Sistem ini dapat melakukan pertukaran pesan dari perangkat Non IP ke perangkat IP melalui *Bridge System*.

4.2.2 Kegunaan

Dengan adanya sistem ini pertukaran komunikasi perangkat IP dan perangkat Non IP melalui *Bridge System* dapat dijalankan sebagai penerapan komunikasi berbasis *Internet of Things (IoT)*. Selain itu dengan memanfaatkan protokol MQTT memudahkan *Bridge System* mengontrol dan meneruskan pesan dalam kondisi jaringan low-bandwidth, penggunaan alokasi memori yang minim, serta daya proses dan penyimpana yang kecil.

4.2.3 Karakteristik Pengguna

Pengguna dalam implementasi sistem ini bertindak sebagai pengirim dari perangkat bluetooth, penerima dari perangkat IP dan sebaliknya.

4.2.4 Lingkungan Operasi

Kebutuhan lingkungan operasi yang mendukung kebutuhan sistem dijelaskan sebagai berikut.

1. Kondisi lingkungan tidak terlalu basah karena dapat merusak komponen sistem
2. Kondisi suhu lingkungan operasi tidak boleh lebih dari 50 Celcius

4.2.5 Batasan Perancangan dan Implementasi

Beberapa batasan dalam perancangan dan implementasi sistem dijelaskan sebagai berikut.

1. Komunikasi Bluetooth akan mudah beroperasi apabila bluetooth telah terdaftar di *Bridge System* dengan proses pairing
2. Sistem bekerja apabila terdapat supply daya untuk menghidupkan *Bridge System*
3. Sistem akan bekerja dengan baik ketika seluruh sesi telah terkoneksi dan terhubung satu sama lain

4.2.6 Asumsi dan Ketergantungan

Beberapa asumsi dan ketergantungan dalam persyaratan sistem dijelaskan sebagai berikut.

1. Komunikasi perangkat bluetooth dapat beroperasi dengan baik apabila jarak komunikasi kurang dari 30 meter
2. Perangkat IP dalam sistem dapat mengirimkan informasi dengan protokol MQTT dengan salah satu jenis QoS yaitu QoS0, QoS1 dan QoS2

4.3 Kebutuhan Antarmuka Eksternal

4.3.1 Kebutuhan Antarmuka Pengguna

Sistem dapat bekerja apabila semua perangkat sudah saling terhubung sesuai perancangan. Lalu program dijalankan dan pengguna dapat memasukkan pesan yang diinginkan melalui antarmuka program *Command Line Interface (CLI)* pada perangkat IP dan Perangkat Non IP.

4.3.2 Antarmuka Perangkat Keras

1. Perangkat IP
 - a. PC Laptop
2. *Network bridge system*
 - a. *Raspberry pi 2*
 - b. Bluetooth Dongle
 - c. Wifi Adapter
3. Perangkat Non IP
 - a. PC Laptop

4.3.3 Antarmuka Perangkat Lunak

1. Perangkat Non-IP
 - a. Sistem operasi Linux Ubuntu
 - b. Bluez library.
2. *Network Bridge Sistem*
 - a. Sistem Operasi Raspbian Wheezy
 - b. Bluez library
 - c. MQTT library
 - d. Putty
3. Perangkat IP
 - a. Sistem operasi Linux Ubuntu
 - b. MQTT library

4.3.4 Antarmuka Komunikasi

Komunikasi yang terjadi pada sistem menggunakan 2 jenis arsitektur komunikasi sehingga perangkat dapat saling bertukar yaitu komunikasi bluetooth antara perangkat Non IP dan *Bridge System* lalu komunikasi MQTT antara perangkat IP dan *Bridge System*.

4.4 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang harus dipenuhi agar suatu sistem dapat berjalan, beberapa kebutuhan fungsional yang harus ada pada sistem ini dijelaskan sebagai berikut.

4.4.1 Fungsi Komunikasi Perangkat IP

1. Penjelasan

Fitur ini mengharuskan sistem dapat berkomunikasi antara perangkat berbasis IP dan *Bridge System*. Komunikasi dalam hal ini perangkat IP dapat mengirim ataupun menerima data dari perangkat Non IP melalui *Bridge System*. Jenis *Priority High Priority*.

2. Respon Sistem

Perangkat IP dapat terhubung ke *Bridge System* melalui jaringan Internet. *Client* IP dapat mengirimkan *request* koneksi dan mendapatkan *reply* dari *Bridge* jaringan menggunakan protokol komunikasi MQTT.

3. Kebutuhan Fungsional

Perangkat IP dapat mengirimkan data atau menerima data dari *Bridge System*. Data tersebut harus sama seperti yang dikirimkan oleh pengirim perangkat Non IP bluetooth.

4.4.2 Fungsi Komunikasi Perangkat Non IP

1. Penjelasan

Fitur ini mengharuskan sistem dapat berkomunikasi antara perangkat Non IP Bluetooth dan *Bridge System*. Komunikasi dalam hal ini Perangkat Non IP Bluetooth dapat mengirim ataupun menerima data dari perangkat IP melalui *Bridge System*. Jenis *Priority High Priority*.

2. Respon Sistem

Perangkat Non IP Bluetooth dapat terhubung ke *Bridge System* melalui transmisi sinyal dari bluetooth dongle yang terpasang pada *Bridge System*. Perangkat Non IP Bluetooth dapat mengirimkan *request* koneksi dan mendapatkan *reply* dari *Bridge System*.

3. Kebutuhan Fungsional

Perangkat Non IP Bluetooth dapat mengirimkan data atau menerima data dari *Bridge System*. Data tersebut harus sama seperti yang dikirimkan oleh pengirim perangkat Non IP bluetooth. Dalam proses terjalannya komunikasi *client* Non IP Bluetooth harus melewati proses *scanning port* dan *pairing device* ketika baru pertama kali terhubung ke *Bridge System*.

4.4.3 Fungsi *Network bridge system*

1. Penjelasan

Fitur ini mengharuskan sistem dapat melayani *request* pertukaran data dari perangkat. Ketika adanya koneksi dari perangkat IP dan perangkat Non IP Bluetooth yang bersamaan dengan pengiriman data dari masing – masing perangkat. Ketika adanya data yang masuk ke *Bridge System* lalu data tersebut dapat diteruskan ke tujuan. Jenis *Priority High Priority*.

2. Respon Sistem

Respon dari *Bridge System* ketika adanya koneksi dari masing – masing perangkat IP dan perangkat Non IP Bluetooth dapat menangani koneksi masing – masing perangkat. Ketika adanya data yang masuk ke *Bridge System* lalu data tersebut dapat diteruskan ke tujuan

3. Kebutuhan Fungsional

Fungsi ini mengharuskan *Bridge System* dapat menangani koneksi yang masuk dalam jaringan. *Bridge System* dapat mengenali ID Bluetooth bluetooth yang terdaftar. Dan ketika adanya data yang masuk ke *Bridge* jaringan lalu data tersebut dapat diteruskan ke tujuan.

4.5 Kebutuhan Non Fungsional

4.5.1 Kebutuhan Performansi

Sistem akan bekerja dengan performa yang maksimum apabila beberapa faktor dapat terpenuhi, yaitu koneksi komunikasi pada perangkat *client* ke *Bridge* jaringan dapat terjalin dengan mudah. Lama pengiriman data antar *client* tidak lebih dari 2 detik. *System Bridge* mampu meneruskan data antar *client* dengan kondisi jaringan yang baik.

4.5.2 Kebutuhan Keselamatan

Kebutuhan keselamatan yang sebaiknya disarankan yaitu menggunakan supply daya yang sesuai pada *Bridge System* yaitu 5-12V karena kesalahan penggunaan daya dapat merusak komponen seperti *Raspberry pi*, Bluetooth Dongle dan Wifi Adapter.

4.5.3 Kebutuhan Keamanan

Kemanan data pada *Bridge System* harus terenkripsi pada lau lintas pengiriman datanya. Karena lingkup jaringan pada komunikasi perangkat IP adalah internet. Jika data tersebut terenkripsi dapat meningkatkan keamanan data agar sampai ke tujuan.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

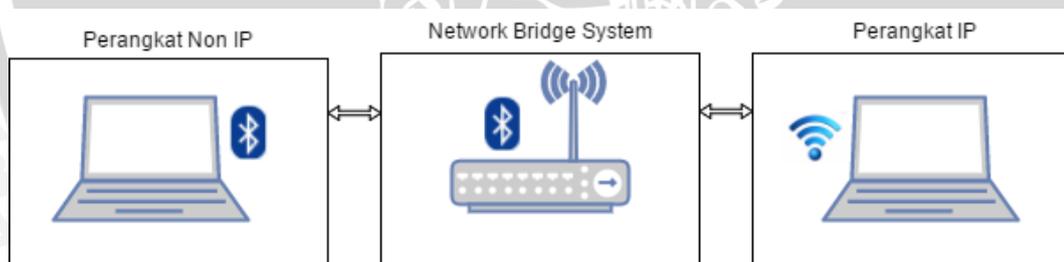
Bab ini menjelaskan tentang perancangan sistem pada penelitian ini, serta implementasi sistem meliputi perangkat keras, perangkat lunak dan pendukung sistem lainnya

5.1 Perancangan Sistem

Proses perancangan dibagi menjadi beberapa tahapan yang dimulai dengan mengetahui gambaran umum arsitektur sistem dan alur kerja sistem. Selanjutnya dilakukan perancangan perangkat keras, perangkat lunak serta perancangan pendukung sistem.

5.1.1 Gambaran umum arsitektur sistem

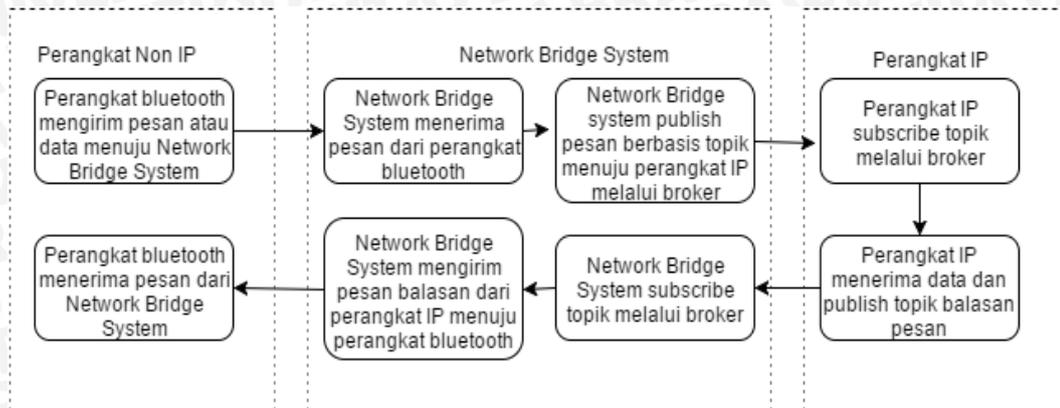
Konsep umum arsitektur sistem yang diterapkan adalah *Network bridge system* yang dapat menghubungkan perangkat IP dan perangkat Non IP bluetooth. Perangkat Non IP akan mengirimkan pesan ke *Network bridge system* menggunakan komunikasi Bluetooth dengan protokol RFCOMM. Pesan yang diterima dari perangkat IP ke *Network bridge system* akan diteruskan ke Perangkat IP menggunakan protokol komunikasi MQTT. Perangkat IP akan menerima pesan dari *Network bridge system*. Pesan yang sudah diterima akan dikembalikan sebagai *Acknowledge* pesan sudah diterima ke *Network bridge system*. Dan perangkat Non IP akan menerima pesan dari *Network bridge system* bahwa data yang dikirimkan sudah diterima. Pada gambar 5.1 berikut gambaran arsitektur sistem yang digunakan.



Gambar 5. 1 Arsitektur Sistem

5.1.2 Alur Kerja Sistem

Alur kerja sistem yang dilakukan pada penelitian ini adalah proses operasi pada sistem yang dimulai dengan pengiriman pesan dari perangkat bluetooth sampai proses perangkat bluetooth dapat menerima pesan balasan dari perangkat IP. Pada gambar 5.2 dijelaskan diagram alir sistem bagaimana sistem dapat bekerja.



Gambar 5. 2 Diagram Alir System

Pernjelasan dari Gambar 5.2 adalah proses komunikasi yang terjadi pada sistem antara perangkat non IP, *Network bridge system* dan Perangkat IP. Berikut proses yang alur komunikasi pada sistem.

1. Perangkat Bluetooth mengirim pesan kepada perangkat IP melalui *Network bridge system*. Perangkat bluetooth mengirim pesan berbasis text menuju *network bridge system* menggunakan komunikasi bluetooth dengan protokol RFCOMM.
2. Setelah perangkat bluetooth mengirim pesan *network bridge system* menerima pesan tersebut dan menampilkan pesan yang dikirim pada tampilan program.
3. Pesan yang diterima dari perangkat bluetooth dikirim ke perangkat IP dengan mengubah pesan menjadi topik. *Network bridge system publish* topik tersebut ke perangkat IP melalui *broker* menggunakan komunikasi MQTT.
4. Perangkat IP *subscribe* topik tersebut ke *broker* menggunakan komunikasi MQTT. Dan menampilkan hasil *subscribe* ke dalam tampilan program.
5. Setelah pesan diterima perangkat IP membalas pesan tersebut dengan balasan bahwa pesan tersebut telah diterima. Perangkat IP *publish* balasan berbasis topik ke *network bridge system* melalui *broker* menggunakan komunikasi MQTT.
6. *Network bridge system subscribe* topik balasan ke *broker* menggunakan komunikasi MQTT. Dan menampilkan hasil balasan ke dalam tampilan program.
7. Pesan balasan tersebut diteruskan ke perangkat bluetooth menggunakan komunikasi bluetooth dengan protokol RFCOMM
8. Perangkat bluetooth menerima pesan balasan dari perangkat IP melalui *network bridge system* dan menampilkan pesan balasan ke dalam tampilan program.

5.1.3 Perancangan Perangkat Keras

Perancangan perangkat keras pada sistem dilakukan dengan konfigurasi antar perangkat agar perangkat dapat terhubung sesuai dengan arsitektur sistem. Perangkat Non IP yang digunakan adalah PC Laptop dengan fitur bluetooth didalamnya yang berfungsi untuk mengirim pesan dan menerima pesan melalui *Network bridge system* menggunakan komunikasi bluetooth. Sedangkan perancangan *network bridge system* perangkat keras yang digunakan adalah mikrokomputer *Raspberry pi 2* model B dengan tambahan bluetooth dongle dan wifi adapter. *Network bridge system* berfungsi mengontrol dan meneruskan pesan dari perangkat IP dan perangkat Non IP. Perancangan perangkat keras yang terakhir untuk perangkat IP adalah PC Laptop dengan fitur wifi didalamnya. Perangkat IP berfungsi untuk menerima dan mengirim pesan melalui *Network bridge system* menggunakan komunikasi MQTT.

5.1.4 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada sistem adalah menanamkan perangkat lunak ke masing – masing komponen perangkat keras pada sistem. Perangkat lunak yang ditanamkan pada perangkat Non IP adalah IDE python dan bluez library yang digunakan untuk programming python bluetooth pada sistem. Perangkat lunak yang ditanamkan di *Network bridge system* adalah *System* operasi raspbian wheezy, IDE python untuk pemrograman python pada *System*, bluez library untuk pemrograman bluetooth pada *System* dan MQTT library untuk pemrograman MQTT untuk komunikasi ke perangkat IP. Sedangkan perangkat lunak pada perangkat IP yang digunakan adalah IDE python untuk pemrograman python dan MQTT library untuk pemrograman MQTT untuk komunikasi dengan *Bridge System*.

5.1.5 Perancangan sistem Bluetooth

Pada sistem bluetooth perancangan yang dilakukan dibagi menjadi 2 tahapan yaitu.

1. Perancangan perangkat keras

Perancangan perangkat keras yang dilakukan adalah konfigurasi perangkat keras yang digunakan. Perangkat keras pada sisi *client* perangkat non IP bluetooth yang digunakan adalah PC Laptop dan sisi *Bridge System* adalah mikrokomputer *Raspberry pi* dengan tambahan wifi dongle. Pesan atau data yang dikirimkan menggunakan media transmisi bluetooth dengan protokol komunikasi RFCOMM.

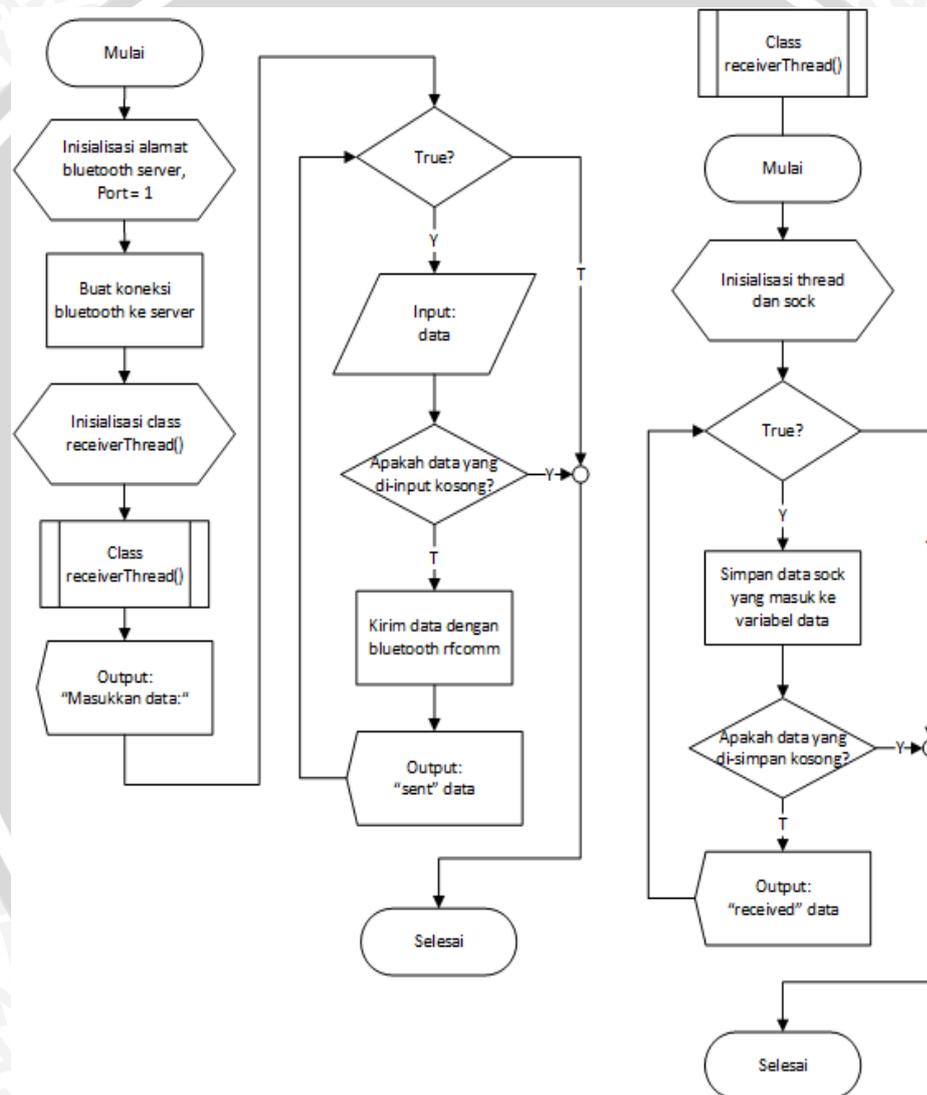
2. Perancangan perangkat lunak

Perancangan perangkat lunak yang dilakukan adalah menanamkan perangkat lunak yang dibutuhkan dalam sistem kedalam perangkat keras. Perangkat yang digunakan dalam sistem bluetooth adalah IDE python yang digunakan untuk pemrograman python buetooth agar dapat terhubung ke *Bridge System*. Dan menanamkan bluez library untuk menjalankan

beberapa *script* python dalam pemrograman bluetooth. Dalam perancangan dan implementasi sistem pada sistem bluetooth akan dibuat 2 skenario implementasi untuk pengujian sistem. Skenario satu yaitu sistem yang diawali dari sistem bluetooth dan sistem ip sebagai pemberi ack. Dan skenario dua yaitu sistem yang diawali dari sistem ip dan sistem bluetooth sebagai pemberi ack.

A. Skenario 1 Sistem Bluetooth

Berikut pada gambar 5.3 *flowchart* perancangan sistem bluetooth pada skenario 1.



Gambar 5. 3 Flowchart Perancangan Sistem Bluetooth Skenario 1

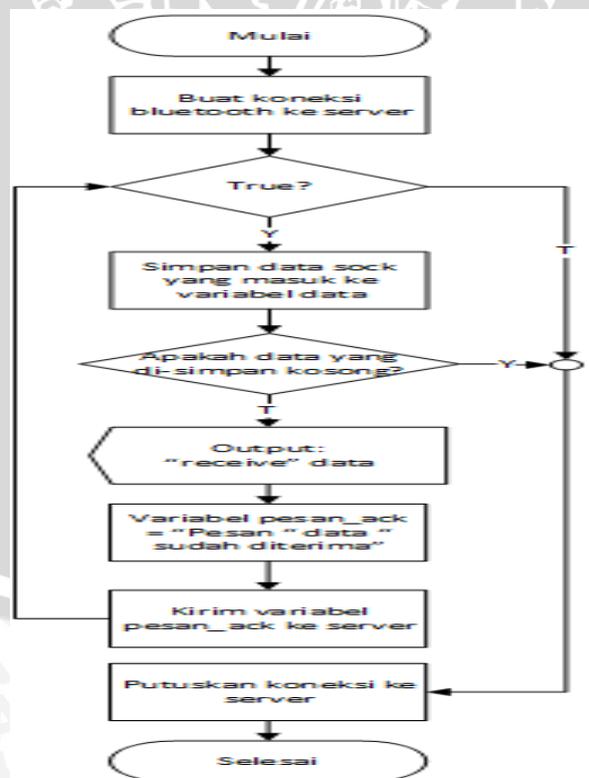
Berikut penjelasan Gambar 5.3 tentang *flowchart* perancangan sistem bluetooth skenario 1.

- a. Pada awal program bluetooth akan menginisialisasi *port* yang digunakan dalam komunikasi yaitu *port* 1

- b. Langkah selanjutnya bluetooth akan membuat koneksi ke server menggunakan protokol RFCOMM
- c. Proses selanjutnya akan menginisialisasi dan menjalankan fungsi *class thread* untuk menerima pesan
- d. Proses selanjutnya program akan menampilkan Masukkan data, saat program berjalan *client* bluetooth diharapkan untuk memasukkan pesan atau data yang akan dikirim
- e. Jika pesan sudah dimasukkan dan dikirim program ke *network bridge system* tetapi jika data belum dimasukkan program akan terus menunggu masukkan dari *client* sampai program dihentikan.
- f. Pada *class receiverThread()* proses awal yang dijalankan adalah inialisasi *thread* dan socket yang digunakan untuk menerima data. Jika ada data yang masuk, data kemudian disimpan dalam variabel data. Jika data yang dikirim bukan data kosong maka data akan ditampilkan kedalam *output* program yaitu "*received data(data yang diterima)*". Jika tidak ada data yang masuk program akan terus menunggu sampai ada data yang masuk dan program dihentikan.

B. Skenario 2 Sistem Bluetooth

Berikut perancangan sistem bluetooth pada skenario kedua dijelaskan pada gambar 5.4 *Flowchart* sistem bluetooth sknario 2.



Gambar 5. 4 Flowchart Perancangan Sistem Bluetooth Skenario 1



Berikut penjelasan Gambar 5.4 tentang *flowchart* perancangan sistem bluetooth skenario 2.

- a. Pada awal program bluetooth akan menginisialisasi *port* yang digunakan dalam komunikasi yaitu *port 1*
- b. Langkah selanjutnya bluetooth akan membuat koneksi ke server menggunakan protokol RFCOMM
- c. Proses selanjutnya adalah menunggu data masuk dari perangkat IP. Jika pesan sudah dikirim program ke *network bridge system* tetapi jika data belum dimasukkan program akan terus menunggu masukkan dari *client* sampai program dihentikan.
- d. Jika ada data yang masuk maka data tersebut akan disimpan pada variabel data
- e. Lalu akan ditampilkan pada program data yang masuk
- f. Proses selanjutnya akan membalas data tersebut dengan pesan ACK bahwa pesan tersebut sudah diterima.

5.1.6 Perancangan *Network Bridge System*

Pada sistem *network bridge system* perancangan yang dilakukan dibagi menjadi 2 tahapan yaitu.

1. Perancangan perangkat keras

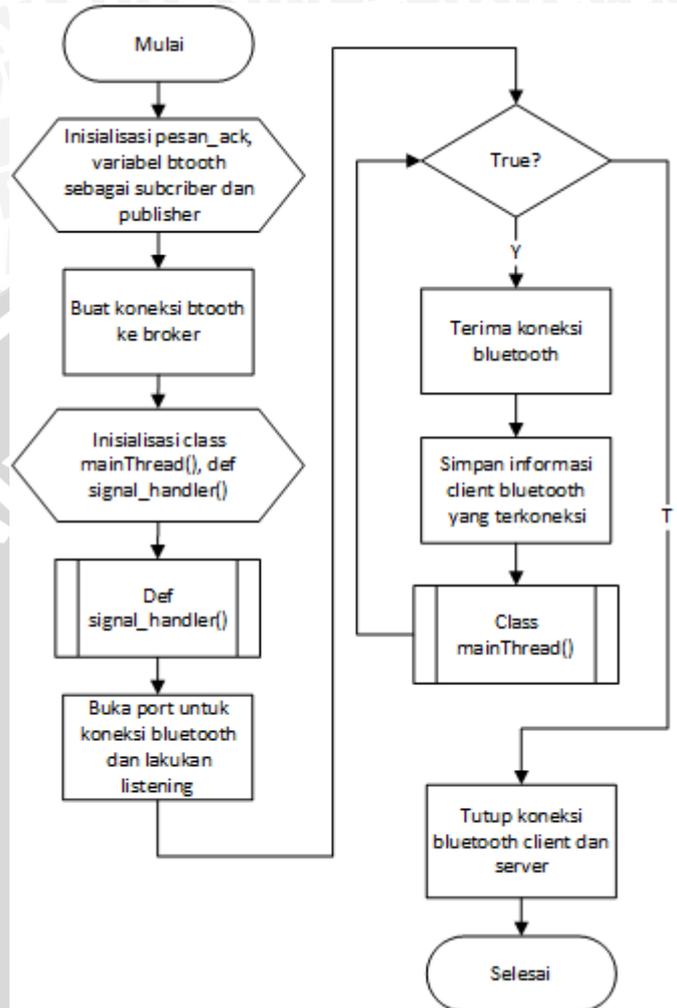
Perancangan perangkat keras pada sistem *Network bridge system* adalah konfigurasi perangkat keras yang digunakan adalah mikrokomputer *Raspberry pi 2* model B. Perangkat keras yang ditambahkan adalah bluetooth dongle dan wifi adapter.

2. Perancangan perangkat lunak

Perancangan perangkat lunak yang dilakukan adalah menanamkan perangkat lunak yang dibutuhkan dalam sistem kedalam perangkat keras. Perangkat lunak yang digunakan pada perangkat *network bridge system* adalah sistem operasi raspbian wheezy untuk mikrokomputer *Raspberry pi 2* model B, IDE python untuk pemrograman python yang digunakan untuk membuat program untuk mengontrol pesan atau data dan meneruskan pesan atau data. Tahap perancangan selanjutnya menanamkan bluez library untuk menjalankan *script* pemrograman bluetooth dan MQTT library untuk menjalankan *script* pemrograman MQTT. Dalam perancangan dan implementasi sistem pada sistem *network bridge system* akan dibuat 2 skenario implementasi untuk pengujian sistem. Skenario satu yaitu sistem yang diawali dari sistem bluetooth dan sistem ip sebagai pemberi ack. Dan skenario dua yaitu sistem yang diawali dari sistem ip dan sistem bluetooth sebagai pemberi ack.

A. Skenario 1 Network bridge system

Berikut pada gambar 5.5 *flowchart* perancangan sistem *network bridge system* pada skenario 1.

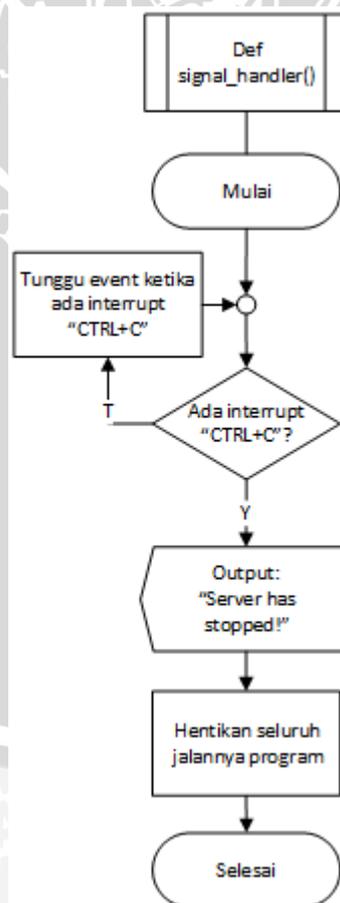


Gambar 5.5 Flowchart Network bridge system skenario 1

Berikut penjelasan Gambar 5.5 tentang *flowchart* perancangan *Network bridge system*.

- Pada awal program proses pertama yang dilakukan adalah inisialisasi variabel `pesan_ack` yang digunakan untuk menampung data yang diterima dari perangkat IP, variabel `btouth` sebagai *subscriber* dan *publisher*.
- Proses selanjutnya adalah membuat koneksi btouth ke alamat *broker*. Alamat *broker* yang digunakan pada alamat *loopback ipv6* yaitu “:::0” dengan *port* 1883.
- Proses selanjutnya adalah inisialisai *class* dari `mainThread` dan inisialisasi fungsi `signal_handler`.

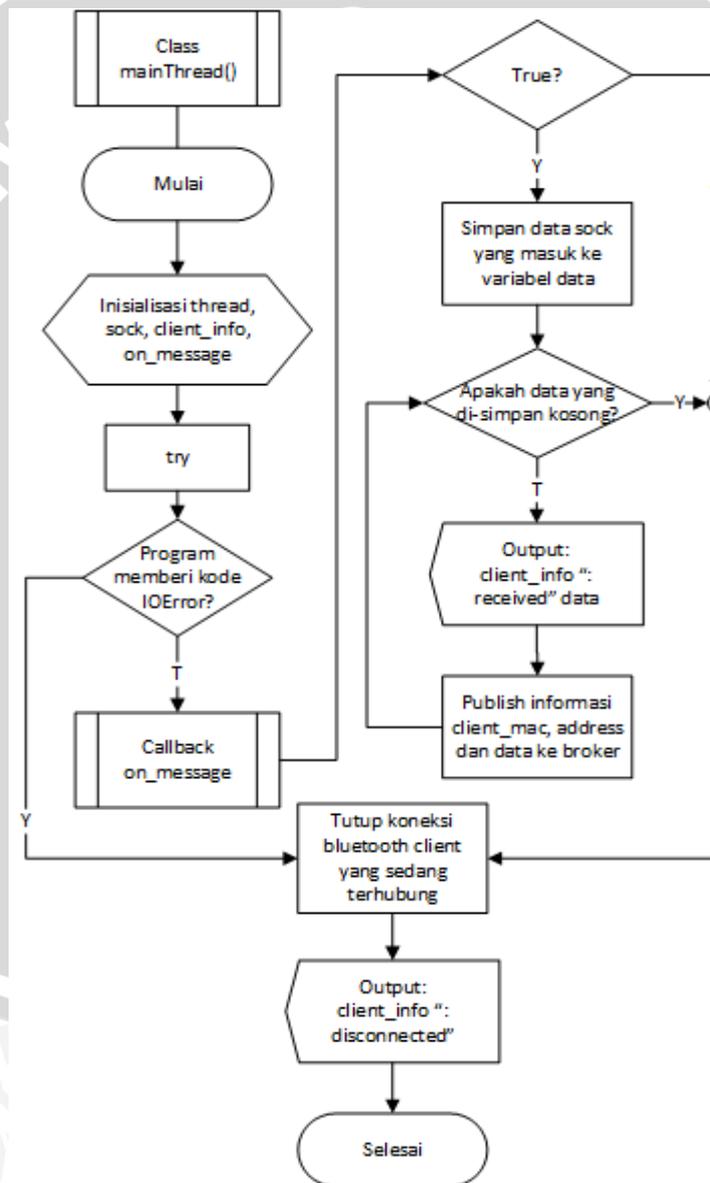
- d. Proses selanjutnya menjalankan fungsi *signal handler*, fungsi ini digunakan untuk menghandel program ketika program dipaksa untuk berhenti dan ketika ada even tertentu.
- e. Proses selanjutnya adalah proses membuka *port* untuk komunikasi perangkat bluetooth dan melakukan *listening* untuk perangkat bluetooth yang terhubung.
- f. Ketika ada perangkat bluetooth yang terhubung *network bridge system* maka koneksi akan diterima dan siap menyimpan pesan yang dikirim.
- g. Setelah pesan diterima dan disimpan siap menjalankan *class MainThread* untuk pemrosesan data selanjutnya yang akan dikirim ke perangkat IP
- h. Jika tidak ada perangkat bluetooth yang terhubung maka proses akan berhenti dan menunggu sampai adanya koneksi yang terhubung.
- i. Untuk fungsi *signal handler* akan dijelaskan pada *flowchart* berikut ini.



Gambar 5. 6 Flowchart Fungsi Signal Handler pada Network bridge system

Berikut penjelasan Gambar 5.6 tentang Fungsi *Signal Handler* pada *Network bridge system*.

- Proses awal pada fungsi *signal handler* adalah *forever looping* sampai adanya *event interrupt* dari user.
- Jika ada *interrupt* dari *user* yang berupa penekanan *keyboard* "CTRL+C" maka akan menampilkan "Server has stopped" pada tampilan program dan seluruh proses akan dihentikan.
- Jika tidak ada *interrupt* dari *user* program akan terus berjalan dan berulang sampai adanya *interrupt* dari *user*.

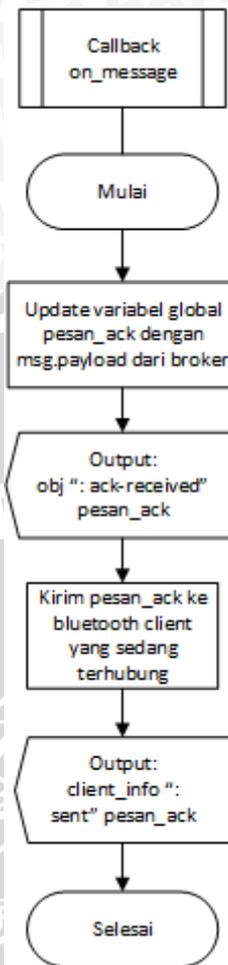


Gambar 5. 7 Flowchart Class Main Thread pada Network bridge system

Berikut penjelasan Gambar 5.7 tentang *Class Main Thread* pada *Network bridge system*.

- a. Proses awal yang dilakukan pada *class mainThread* adalah menginisialisasi socket yang digunakan, *client* info dan fungsi *on_message*.
- b. Proses selanjutnya adalah adalah kondisi ketika terjadinya socket error. Jika tidak ada socket error maka panggil fungsi *on_message*.
- c. Jika didalam fungsi terdapat data yang diterima dari perangkat IP dan data tidak kosong maka data disimpan dan dimasukkan kedalam variabel data.
- d. Proses selanjutnya jika data yang diterima tidak kosong dan sudah disimpan tampilkan pada program “received (data yang diterima)”
- e. Data yang disimpan tadi *publish* berupa informasi *client* mac, alamatnya dan data yang dikirimkan
- f. Proses akan berulang kembali
- g. Jika didalam fungsi *on_message* data yang diterima kosong dan perangkat bluetooth menutup koneksi maka program langsung menuju proses menutup koneksi tersebut dan menampilkan bahwa perangkat bluetooth tersebut *disconnected*.





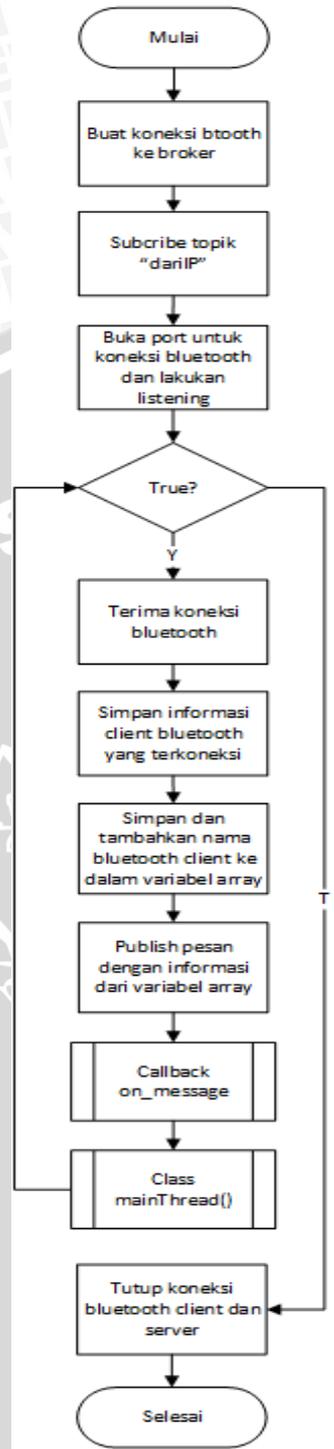
Gambar 5. 8 Flowchart Fungsi on_message pada Network bridge system

Berikut penjelasan Gambar 5.8 tentang Fungsi *on_message* pada *Network bridge system*

- Proses awal yang dijalankan pada fungsi *on_message* adalah *update variable* global *pesan_ack* dengan *msg payload* atau data dalam komunikasi MQTT yang diambil dari *broker*
- Lalu kirimkan pesan balasan dari perangkat IP ke perangkat bluetooth dan tampilkan pada program berupa pesan balasan “ack-received(pesan balasan)”
- Proses selanjutnya tampilkan pada program bahwa telah dikirim pesan ack balasan dari perangkat IP ke perangkat bluetooth.

B. Skenario 2 Network bridge system

Berikut perancangan sistem *Network bridge system* pada skenario 2 dijelaskan pada gambar 5.9 *Flowchart sistem Network bridge system* skenario 2.



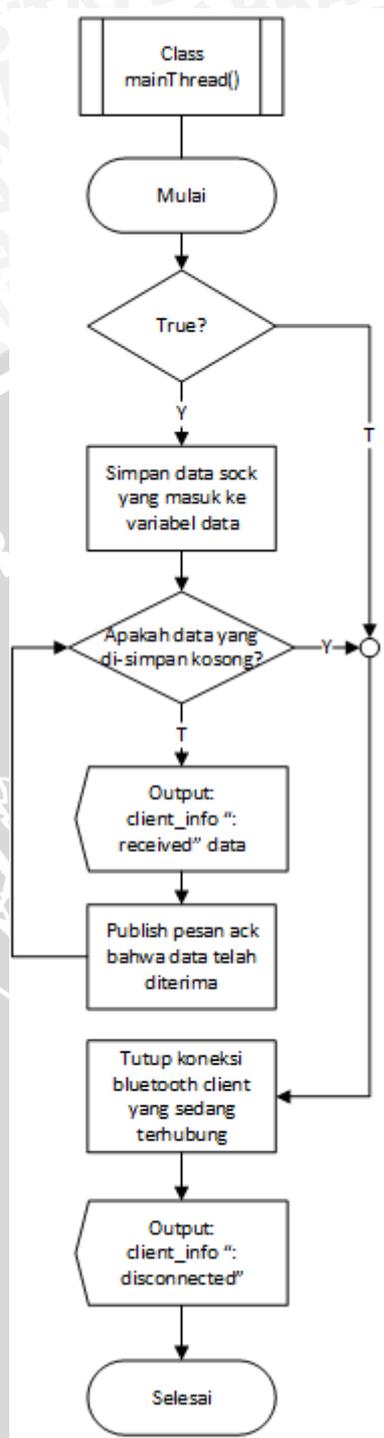
Gambar 5. 9 Flowchart Network bridge system skenario 2

Berikut penjelasan Gambar 5.9 tentang *flowchart* perancangan *Network bridge system*.

- a. Pada awal program proses pertama yang dilakukan adalah melakukan koneksi dari variabel btooth ke *broker*

- b. Proses selanjutnya adalah *subscribe* topik yang dikirimkan dari perangkat IP
- c. Proses selanjutnya adalah membuka *port* koneksi ke perangkat bluetooth dan lakukan *listening client* bluetooth
- d. Jika ada perangkat bluetooth yang ingin terhubung terima koneksi dari perangkat bluetooth.
- e. Simpan informasi *client* bluetooth yang sudah terhubung. informasi yang disimpan berupa *MAC Address* dan nama ID Bluetooth
- f. Simpan dan tambahkan nama bluetooth *client* ke dalam variabel array
- g. *Publish* pesan informasi tentang perangkat bluetooth yang terhubung dari variabel array
- h. Lalu memanggil ulang fungsi *on_message*
- i. Lalu jalankan fungsi *mainThread* untuk pengolahan pesannya
- j. Jika tidak ada koneksi dari perangkat bluetooth atau perangkat bluetooth mengakhiri sei maka program akan berakhir



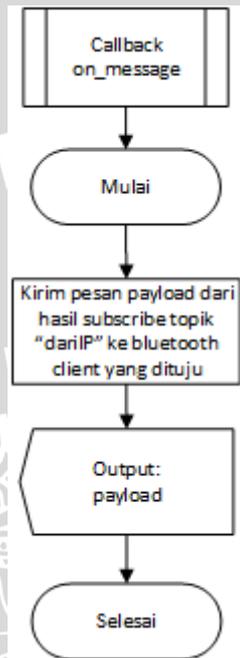


Gambar 5. 10 Flowchart Network bridge system main class skenario 2

Berikut penjelasan Gambar 5.10 tentang *flowchart* perancangan *Network bridge system* pada fungsi *mainThread*.

- a. Pada awal program proses pertama yang dilakukan adalah lakukan perulangan jika ada data yang masuk dari perangkat IP simpan data kedalam variabel data

- b. Jika data tidak kosong maka tampilkan pada program bahwa telah menerima data yang dikirimkan dari perangkat IP
- c. Pesan selanjutnya adalah mengirimkan data balasan berupa pesan ack bahwa data sudah diterima dari perangkat bluetooth menuju perangkat IP
- d. Jika data tidak ada yang masuk *network bridge system* maka sistem akan terus menunggu sampai ada data yang masuk atau sistem diberhentikan



Gambar 5. 11 Flowchart Network bridge system on_message skenario 2

Berikut penjelasan Gambar 5.11 tentang *flowchart* perancangan *Network bridge system*.

- a. Pada awal program proses pertama yang dilakukan adalah *subscribe* topik data dari perangkat IP
- b. Lalu kirimkan data tersebut ke perangkat bluetooth yang dituju
- c. Lalu tampilkan hasil data yang diterima pada program

5.1.7 Perancangan sistem MQTT *Publisher Subscriber*

Pada sistem MQTT perancangan yang dilakukan dibagi menjadi 2 tahapan yaitu.

1. Perancangan perangkat keras

Perancangan perangkat keras yang dilakukan adalah konfigurasi perangkat keras yang digunakan. Perangkat keras pada sisi *client* perangkat IP yang digunakan adalah PC Laptop dengan wifi didalamnya dan sisi *Bridge System* adalah mikrokomputer *Raspberry pi* dengan tambahan wifi

adapter. Pesan atau data yang dikirimkan menggunakan media transmisi MQTT dengan protokol komunikasi MQTT.

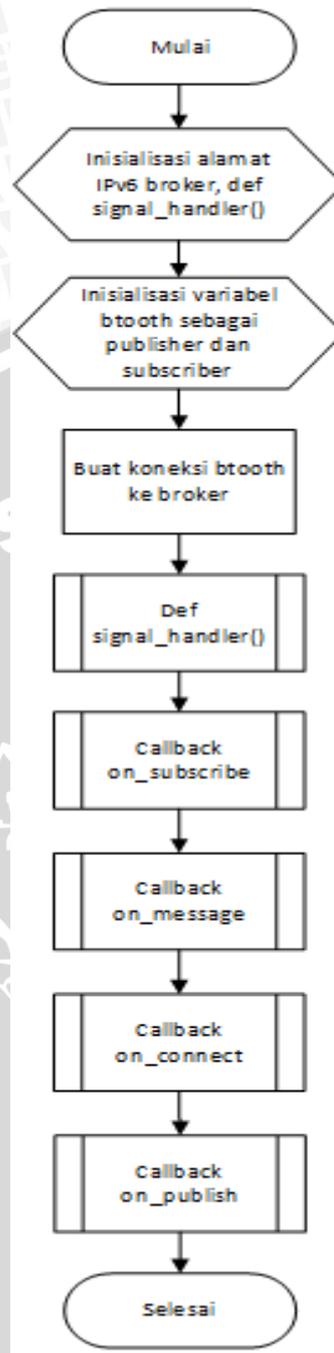
2. Perancangan perangkat lunak

Perancangan perangkat lunak yang dilakukan adalah menanamkan perangkat lunak yang dibutuhkan dalam sistem kedalam perangkat keras. Perangkat yang digunakan dalam sistem MQTT adalah IDE python yang digunakan untuk pemrograman python MQTT agar dapat terhubung ke *Bridge System*. Dan menanamkan MQTT library untuk menjalankan beberapa *script* python dalam pemrograman MQTT. Berikut perancangan sistem MQTT *publisher subscriber* pada perangkat IP. Dalam perancangan dan implementasi sistem pada sistem MQTT akan dibuat 2 skenario implementasi untuk pengujian sistem. Skenario satu yaitu sistem yang diawali dari sistem bluetooth dan sistem ip sebagai pemberi ack. Dan skenario dua yaitu sistem yang diawali dari sistem ip dan sistem bluetooth sebagai pemberi ack.

A. Skenario 1 Sistem IP

Berikut pada gambar 5.3 *flowchart* perancangan sistem IP MQTT pada skenario 1.



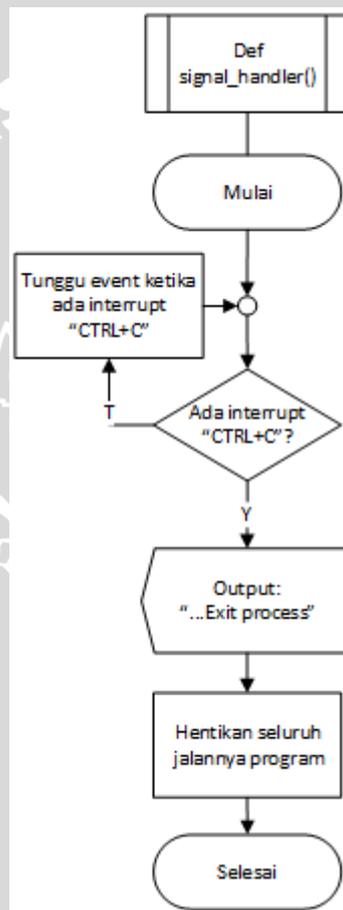


Gambar 5. 12 Flowchart Sistem MQTT pada perangkat IP Skenario 1

Berikut penjelasan Gambar 5.12 tentang sistem MQTT pada perangkat IP.

- Proses awal yang dilakukan pada sistem MQTT pada perangkat IP adalah inisialisasi alamat Ipv6 broker yaitu "fe80::fa1a:67ff:fe0a:744c" dan port komunikasi yaitu 1883
- Proses selanjutnya adalah inisialisasi fungsi signal handler untuk mengontrol even tertentu saat program dijalankan.

- c. Proses selanjutnya menginisialisasi variabel *bt* yang digunakan untuk membuat fungsi *publisher* dan *subscriber*
- d. Lalu buat koneksi *bt* ke alamat *broker*
- e. Proses selanjutnya menjalankan fungsi *signal handler*
- f. Proses selanjutnya saat program berjalan adalah memanggil kembali fungsi *on_subscribe*, *on_message*, *on_connect* dan *on_publish* ketika fungsi tersebut dibutuhkan.
- g. Program akan berulang sampai adanya *event* tertentu dari *user* untuk menghentikan jalannya program.

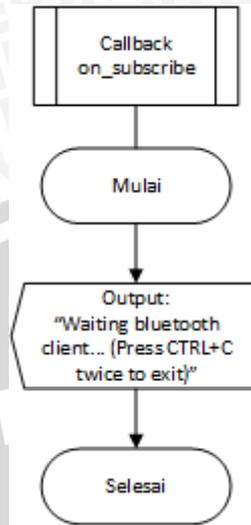


Gambar 5. 13 Flowchart Fungsi Signal Handler pada MQTT

Berikut penjelasan Gambar 5.13 tentang fungsi *signal handler* pada MQTT pada perangkat IP.

- a. Proses awal pada fungsi *signal handler* adalah *forever looping* sampai adanya *event interrupt* dari user.
- b. Jika ada *interrupt* dari *user* yang berupa penekanan *keyboard* "CTRL+C" maka akan menampilkan "Server has stopped" pada tampilan program dan seluruh proses akan dihentikan.

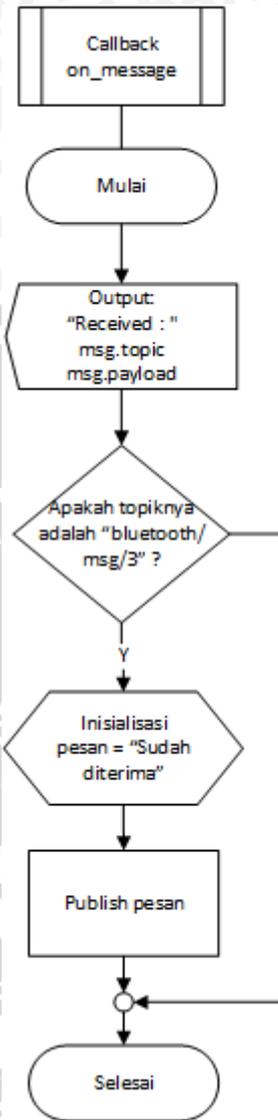
- c. Jika tidak ada *interrupt* dari *user* program akan terus berjalan dan berulang sampai adanya *interrupt* dari *user*.



Gambar 5. 14 Flowchart Fungsi *on_subscribe* pada MQTT

Berikut penjelasan Gambar 5.14 tentang fungsi *on_subscribe* pada MQTT pada perangkat IP.

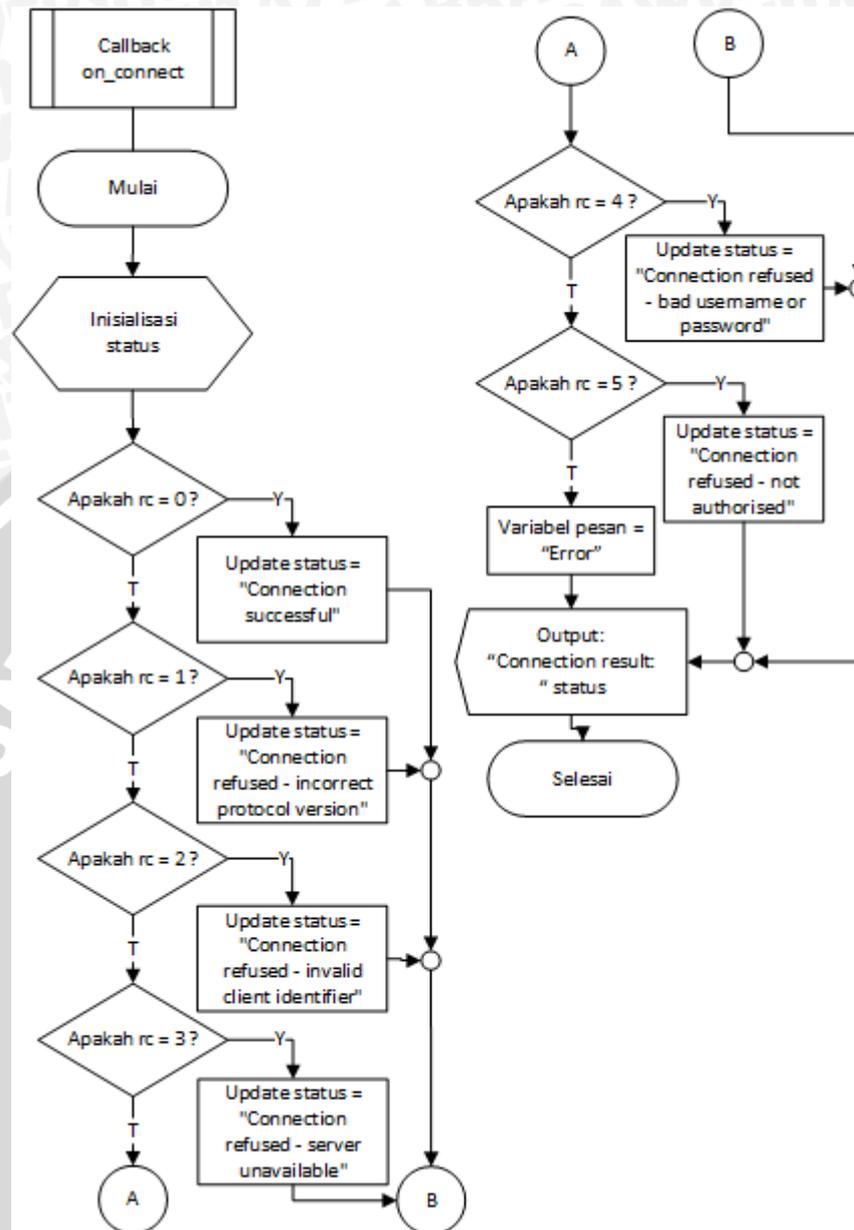
- a. Proses ketika dijalankannya fungsi *on_subscribe* pada komunikasi MQTT adalah menampilkan output “waiting bluetooth *client*.. (Press CTRL+C twice to exit” dan menunggu pesan dari perangkat bluetooth.



Gambar 5. 15 Flowchart Fungsi on_message pada MQTT

Berikut penjelasan Gambar 5.15 tentang fungsi *on_message* pada MQTT pada perangkat IP.

- Proses awal yang dijalankan pada fungsi *on_message* adalah menampilkan *msg topic msg.payload* adalah data yang telah *unsubscribe* dan disimpan dalam bentuk *msg.payload*
- Jika topik pesan yang diterima berupa *path file* "bluetooth/msg/3" maka inialisasi variabel pesan yang berisi pesan yang dikirimkan sudah diterima.
- Lalu *publish* pesan tersebut menggunakan topik dengan *path file* "bluetooth/msg/3"
- Jika topik [esan yang diterima bukan berupa *path file* "bluetooth/msg/3" maka fungsi akan berulang dan berhenti sampai ada even tertentu

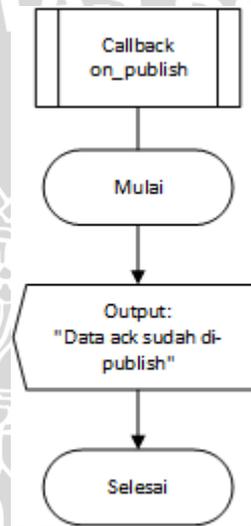


Gambar 5. 16 Flowchart Fungsi on_connect pada MQTT

Berikut penjelasan Gambar 5.16 tentang fungsi *on_connect* pada MQTT pada perangkat IP.

- Proses awal ketika fungsi *on_connect* dijalankan adalah menginisialisasi status yang digunakan untuk mengetahui status koneksi yang terjadi dalam komunikasi MQTT
- Jika *rc* atau status koneksi sama dengan 0 maka *update* variabel status berisi *connection succesful*
- Jika *rc* atau status koneksi sama dengan 1 maka *update* variabel status berisi *connection refused – incorrect protocol version*

- d. Jika rc atau status koneksi sama dengan 2 maka *update* variabel status berisi *connection refused – invalid client identifier*
- e. Jika rc atau status koneksi sama dengan 3 maka *update* variabel status berisi *connection refused – server unavailable*
- f. Jika rc atau status koneksi sama dengan 4 maka *update* variabel status berisi *connection refused – bad username of password*
- g. Jika rc atau status koneksi sama dengan 5 maka *update* variabel status berisi *connection refused – not authorised*
- h. Lalu proses akan berulang sampai menemukan kondisi status koneksi dan tampilkan connection result berupa status koneksi yang diperoleh.

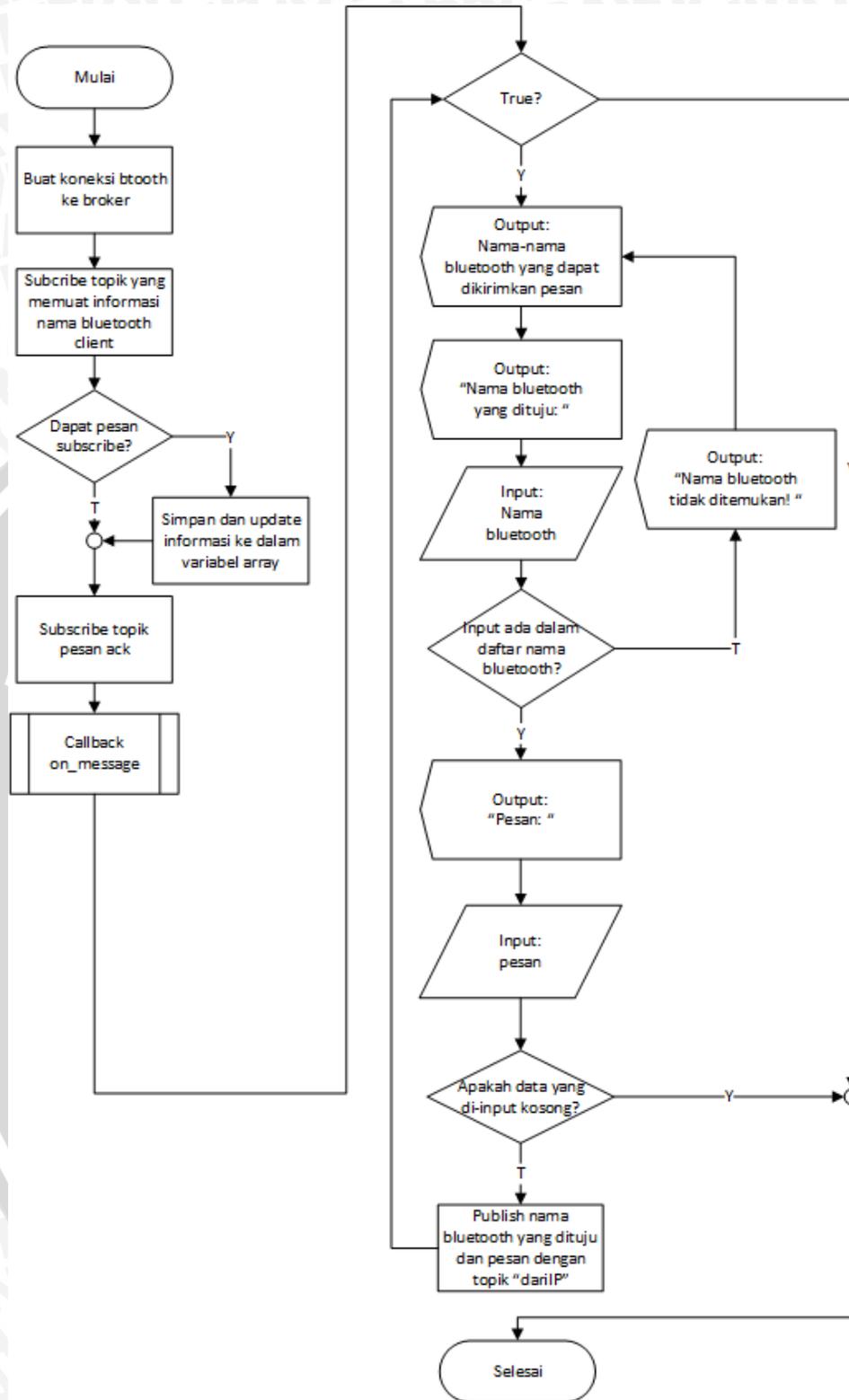


Gambar 5. 17 Flowchart Fungsi *on_publish* pada MQTT

Berikut penjelasan Gambar 5.17 tentang fungsi *on_publish* pada MQTT pada perangkat IP. Proses yang dijalankan pada fungsi *on_publish* pada MQTT perangkat IP adalah menampilkan bahwa pesan ack balasan sudah *dipublish*

B. Skenario 2 Sistem IP

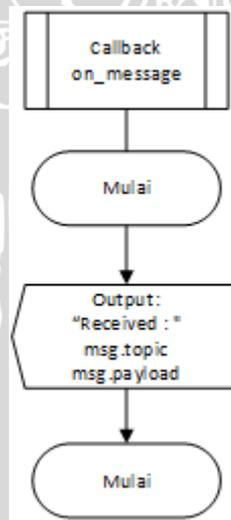
Berikut pada gambar 5.18 *flowchart* perancangan sistem IP MQTT pada skenario 2.



Gambar 5. 18 Flowchart Sistem MQTT pada perangkat IP Skenario 2

Berikut penjelasan Gambar 5.18 tentang flowchart perancangan Network bridge system.

- a. Pada awal program proses pertama yang dilakukan adalah membuat koneksi variabel bttooth ke *broker*
- b. Proses selanjutnya adalah *subscribe* topik dari *network bridge system* yang memuat informasi nama bluetooth *client*
- c. Jika mendapatkan pesan *subscribe* simpan dan update informasi kedalam variabel array jika tidak lakukan proses *subscribe* sampai mendapat data
- d. Lalu *subscribe* topik pesan ack balasan dari perangkat IP
- e. Panggil kembali fungsi *on_message*
- f. Jika ada data pada *on_message* tampilkan nama – nama bluetooth yang dapat menerima pesan
- g. Lalu tampilkan informasi nama bluetooth yang dituju
- h. Lalu masukkan nama bluetooth yang akan dikirimkan data
- i. Lalu masukkan pesan yang akan dikirim dari perangkat IP
- j. Jika data yang dimasukkan tidak kosong maka *publish* nama bluetooth yang dituju dan pesan dengan topik dari IP
- k. Jika data kosong program akan berulang sampai ada data yang masuk



Gambar 5. 19 Flowchart Sistem MQTT *on_message* IP Skenario 2

Berikut penjelasan Gambar 5.19 tentang *flowchart* perancangan *Network bridge system*.

- a. Pada awal program proses pertama yang dilakukan adalah menampilkan hasil *subscribe* berupa topik yang sudah dipublish oleh *network bridge system*

5.2 Implementasi Sistem

Implementasi sistem dilakukan ketika tahap perancangan telah selesai dilakukan. Tahapan implementasi yang dijelaskan dalam penelitian ini meliputi spesifikasi perangkat keras, spesifikasi perangkat lunak, batasan implementasi, implementasi sistem bluetooth, implementasi *network bridge system* dan implementasi sistem MQTT.

5.2.1 Spesifikasi Perangkat Keras

Pada tahapan implementasi sistem perangkat keras dibutuhkan spesifikasi perangkat keras yang digunakan agar sistem dapat beroperasi sesuai dengan kemampuan perangkat kerasnya. Dalam tahapan spesifikasi perangkat keras terdapat 3 bagian yaitu spesifikasi perangkat non IP, spesifikasi *network bridge system* dan spesifikasi perangkat IP. Spesifikasi perangkat keras sistem dijelaskan sebagai berikut.

1. Perangkat non IP

Perangkat non IP digunakan untuk mengirim dan menerima pesan melalui *network bridge system* agar dapat berkomunikasi dengan perangkat IP. Adapun spesifikasi perangkat keras yang digunakan adalah.

Tabel 5. 1 Spesifikasi perangkat Non IP

Perangkat Non IP	
Product Name	Laptop Lenovo G400
OS	Ubuntu 12.04
Processor	Intel Core i3-3110M
Memory (RAM)	4 GB
Hard Disk	500 GB
Bluetooth Model	Bluetooth 2.0
MAC Address Bluetooth	24:FD:52:5A:AC:59

2. *Network bridge system*

Network bridge system digunakan untuk mengontrol pesan atau data dan mengirimkan pesan ke tujuan. Dalam *Network bridge system* terdapat 3 perangkat keras yang bergabung dalam satu kesatuan sistem. Adapun spesifikasi 3 perangkat keras tersebut adalah.

a. Mikrokomputer *Raspberry pi 2*

Mikrokomputer *Raspberry pi 2* berfungsi sebagai board *Bridge System* menggantikan komputer yang diprogram untuk dapat menontrol pesan atau data dari perangkat Non IP dan perangkat IP

Tabel 5. 2 Spesifikasi *Raspberry pi 2*

<i>Raspberry pi 2</i>	
Product Name	<i>Raspberry pi 2 Model B</i>
OS	Raspbian
Chip	Broadcom BCM2836 SoC
Core Architecture	Quad-core ARM Cortex-A7
CPU	900 MHz
Memory	1 GB LPDDR2
GPU	Dual Core VideoCore IV® Multimedia Co-Processor Provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decode Capable of 1 Gpixel/s, 1.5 Gtexel/s or 24 GFLOPs with texture filtering and DMA infrastruktur
Dimensions	85 x 56 x 17 mm
Power	Micro USB socket 5v, 2A
Operating System	Boots from Micro SD card, running a version of the Linux operating System
On Board Storage	SD, MMC, SDIO card slot
HDMI output	Yes

b. Bluetooth Dongle

Bluetooth dongle digunakan dalam *Network bridge system* untuk komunikasi Perangkat Non IP bluetooth.

Tabel 5. 3 Spesifikasi Bluetooth Dongle

Bluetooth Dongle	
Product Name	Bluetooth USB Dongle
Interface	USB 2.0
Modulation	FHSS / GFSK 1 Mbps 1600hop / s

Version	2.0
Frekuensi	2,4 GHz
MAC Address	00:15:83:3D:0A:57

c. Wifi Adapter

Wifi adapter digunakan dalam *Network bridge system* untuk komunikasi perangkat IP.

Tabel 5. 4 Spesifikasi Wifi Adapter

Wifi Adapter	
Product Name	TP-Link USB Adapter
Sistem Model	TL-WN821N
Interface	USB 2.0
Wireless Standards	IEEE 802.11n, IEEE 802.11g, IEEE 802.11b
Signal Rate	11n: Up to 300Mbps(dynamic) 11g: Up to 54Mbps(dynamic) 11b: Up to 11Mbps(dynamic)
Wireless Modes	Ad-Hoc / Infrastructure mode
Wireless Security	WEP, WPA/WPA2, WPA-PSK/WPA2-PSK

3. Perangkat IP

Perangkat IP digunakan untuk mengirim dan menerima pesan melalui *network bridge system* agar dapat berkomunikasi dengan perangkat Non IP. Adapun spesifikasi perangkat keras yang digunakan adalah.

Tabel 5. 5 Spesifikasi Perangkat IP

Perangkat IP	
Product Name	ASUS A45VD
Processor	Intel core i3
Memory	6 GB
Hard Disk	500 GB
Wireless Standards	IEEE 802.11n, IEEE 802.11g, IEEE 802.11b

5.2.2 Spesifikasi Perangkat Lunak

Pada tahapan implementasi sistem perangkat lunak dibutuhkan spesifikasi perangkat lunak yang digunakan agar sistem dapat beroperasi sesuai dengan kemampuannya. Dalam tahapan spesifikasi perangkat lunak terdapat 3 bagian yaitu spesifikasi perangkat non IP, spesifikasi *network bridge system* dan spesifikasi perangkat IP. Spesifikasi perangkat lunak sistem dijelaskan sebagai berikut.

1. Perangkat Non IP
 - a. Ubuntu OS 12.04
 - b. IDE Python
 - c. Bluez library
2. *Network bridge system*
 - a. Raspbian Wheezy OS
 - b. IDE Python
 - c. Bluez Library
 - d. MQTT library
3. Perangkat IP
 - a. Ubuntu OS 14.04
 - b. IDE Python
 - c. MQTT library

5.2.3 Batasan Implementasi

Beberapa batasan yang diterapkan pada implementasi sistem dijelaskan sebagai berikut.

1. Pesan yang dikirim maupun diterima oleh perangkat IP dan perangkat Non IP adalah pesan yang berbentuk string.
2. Perangkat IP dan perangkat Non IP yang digunakan adalah PC Laptop
3. Implementasi sistem dapat bekerja diawali dari menjalankan *Network bridge system*
4. Interface yang digunakan user untuk menjalankan program dalam bentuk command line interface
5. Konsumsi daya listrik pada implementasi sistem saat ini didapat melalui kabel power arus AC.

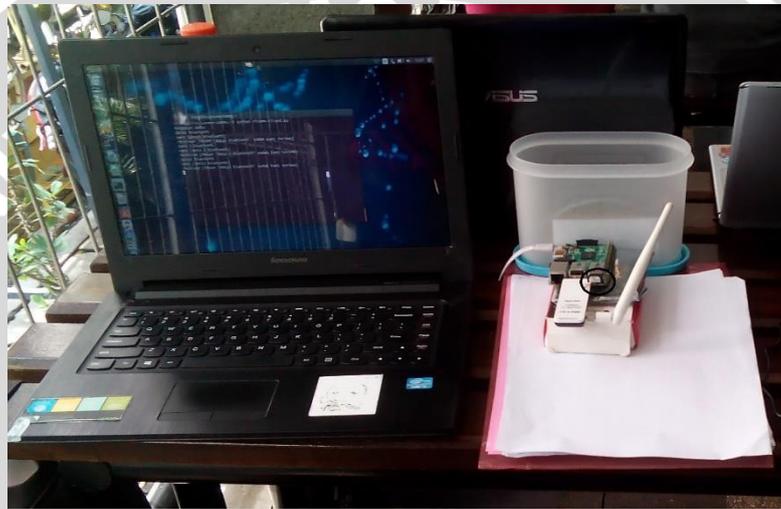
5.2.4 Implementasi Sistem Bluetooth

Pada implementasi sistem bluetooth akan dibagi menjadi 2 tahapan implementasi sesuai dengan perancangan sistem yang telah dibuat. Tahap

implementasi pertama dilakukan dengan implementasi perangkat keras dan tahap implementasi kedua adalah implementasi perangkat lunak. Berikut penjelasan tahapan implementasi sistem bluetooth.

1. Implementasi perangkat keras

Tahapan implementasi perangkat keras pada sistem bluetooth adalah mengaktifkan fungsi bluetooth yang ada pada perangkat keras PC Laptop. Langkah awal untuk konfigurasi bluetooth pada masing – masing perangkat. Langkah kedua pairing antara perangkat Non IP Bluetooth dan bluetooth dongle yang ada pada *Network bridge system*. Langkah terakhir tes koneksi dengan Ping MAC Address tujuan. Pada gambar 5.20 contoh implementasi perangkat keras sistem bluetooth.



Gambar 5. 20 Implementasi Perangkat Keras Sistem Bluetooth

2. Implementasi perangkat lunak

Tahapan implementasi perangkat lunak pada sistem bluetooth adalah menanamkan perangkat lunak yang digunakan pada *System* komunikasi bluetooth dan memprogram agar antar perangkat bluetooth dapat bertukar pesan. Langkah awal untuk implementasi perangkat lunak adalah menanamkan IDE Python untuk membuat program pada masing – masing perangkat bluetooth. Langkah kedua tanamkan library untuk menjalankan code bluetooth pada program pada masing – masing perangkat. Langkah ketiga membuat code program yang digunakan sistem bluetooth untuk dapat mengirim dan menerima pesan menggunakan komunikasi bluetooth. Untuk inisialisasi bluetooth *address* yaitu *MAC Address* tujuan yang terdiri dari 48 bit alamat hexa. *MAC Address network bridge system* yaitu “00:15:83:3D:0A:57”. Sedangkan *MAC Address* untuk perangkat bluetooth adalah “24:FD:52:5A:AC:59”. Langkah terakhir tes komunikasi antar perangkat bluetooth dan jalankan program untuk mengirim dan menerima pesan atau data. Berikut implementasi program sistem bluetooth pada sistem operasi ubuntu dengan menggunakan *tools* nano. Pada gambar 5.21 Implementasi program pada sistem bluetooth.

```

lilis@lilis-Lenovo-G400s: ~
GNU nano 2.2.6 File: rfcomm-client.py

from bluetooth import *
import threading

bd_addr = "00:15:83:3D:0A:57"

port = 1

# Create the client socket
sock=BluetoothSocket( RFCOMM )
sock.connect((bd_addr, port))

class receiverThread(threading.Thread):
    def __init__(self,sock):
        threading.Thread.__init__(self)
        self.sock = sock
    def run(self):
        while True:
            data = self.sock.recv(1024)
            if len(data) == 0: break

^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell

```

Gambar 5. 21 Implementasi Program pada Sistem Bluetooth

5.2.5 Implementasi *Network Bridge* Sistem

Pada implementasi sistem *Network bridge system* akan dibagi menjadi 2 tahapan implementasi sesuai dengan perancangan sistem yang telah dibuat. Tahap implementasi pertama dilakukan dengan implementasi perangkat keras dan tahap implementasi kedua adalah implementasi perangkat lunak. Berikut penjelasan tahapan implementasi sistem *Network bridge system*.

1. Implementasi perangkat keras

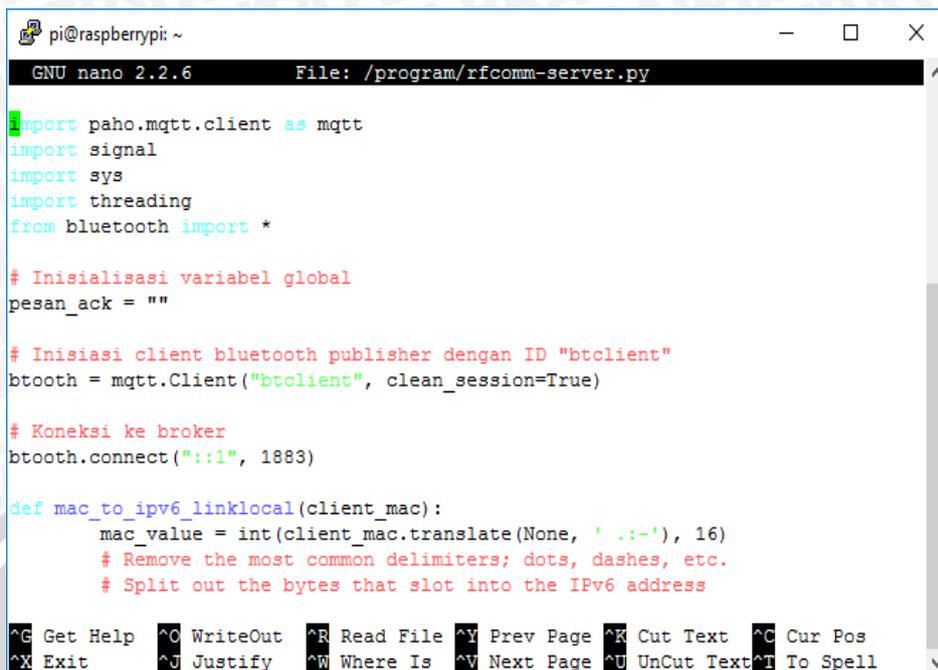
Tahapan implementasi perangkat keras pada sistem *Network bridge system* adalah mengaktifkan fungsi *Raspberry pi* yang digunakan sebagai alat pengontrol pesan atau datanya. Langkah awal untuk konfigurasi *Raspberry pi* yaitu instalasi sistem operasi raspbian wheezy. Langkah kedua adalah memasang bluetooth dongle untuk komunikasi ke perangkat Non IP bluetooth. Langkah ketiga memasang wifi adapter untuk komunikasi ke perangkat IP dan mengkonfigurasi wifi agar dapat berkomunikasi pada jaringan yang sama. Pada gambar 5.22 contoh implementasi perangkat keras *network bridge system*.



Gambar 5. 22 Implementasi Perangkat Keras *Network bridge system*

2. Implementasi perangkat lunak

Tahap implementasi perangkat lunak pada sistem *network bridge system* menanamkan perangkat lunak yang diperlukan pada sistem dan membuat program untuk mengontrol dan menjembatani pertukaran data antara perangkat Non IP bluetooth dan perangkat IP. Langkah awal yang dilakukan adalah menanamkan IDE python untuk membuat program untuk menjembatani komunikasi perangkat bluetooth dan perangkat IP. Langkah kedua adalah menanamkan library yang digunakan untuk menjalankan code program yang digunakan. Langkah terakhir adalah pembuatan program untuk mengontrol komunikasi perangkat Non IP bluetooth dan perangkat IP. Dalam program *network bridge system* inialisasi MAC Address *network bridge system* yaitu "00:15:83:3D:0A:57". Sedangkan MAC Address untuk perangkat bluetooth adalah "24:FD:52:5A:AC:59". Berikut adalah implementasi program pada *network bridge system* yang dibuat pada sistem *remote putty* dengan tool nano. Pada gambar 5.23 Implementasi program pada *network bridge system*.



```

pi@raspberrypi: ~
GNU nano 2.2.6 File: /program/rfcomm-server.py

import paho.mqtt.client as mqtt
import signal
import sys
import threading
from bluetooth import *

# Inisialisasi variabel global
pesan_ack = ""

# Inisiasi client bluetooth publisher dengan ID "btclient"
bttooth = mqtt.Client("btclient", clean_session=True)

# Koneksi ke broker
bttooth.connect("::1", 1883)

def mac_to_ipv6_linklocal(client_mac):
    mac_value = int(client_mac.translate(None, ':-'), 16)
    # Remove the most common delimiters; dots, dashes, etc.
    # Split out the bytes that slot into the IPv6 address

```

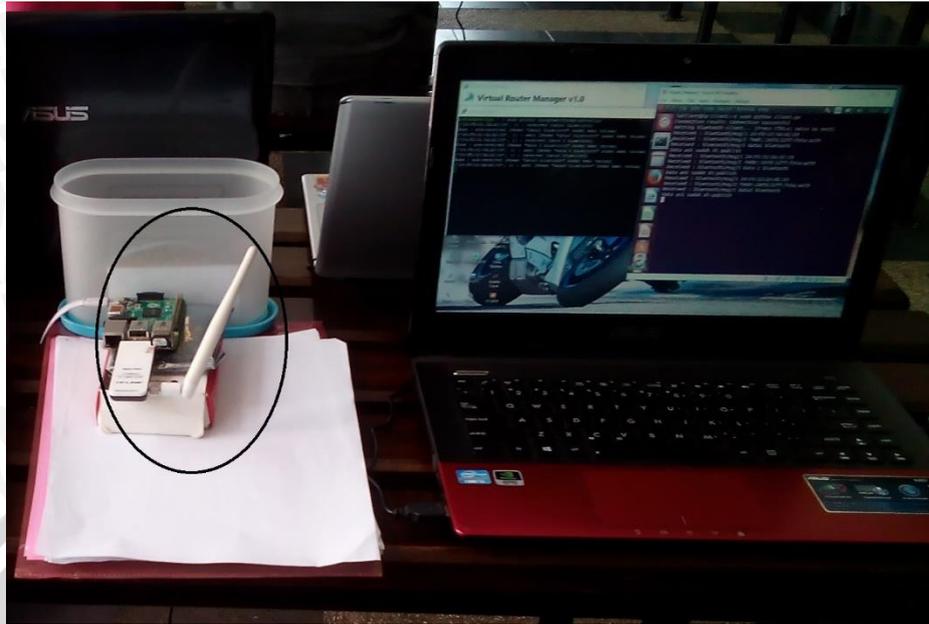
Gambar 5. 23 Implementasi Program pada *Network bridge system*

5.2.6 Implementasi Sistem MQTT

Pada implementasi sistem MQTT akan dibagi menjadi 2 tahapan implementasi sesuai dengan perancangan sistem yang telah dibuat. Tahap implementasi pertama dilakukan dengan implementasi perangkat keras dan tahap implementasi kedua adalah implementasi perangkat lunak. Berikut penjelasan tahapan implementasi sistem MQTT.

1. Implementasi perangkat keras

Tahapan implementasi perangkat keras pada sistem MQTT adalah mengaktifkan fungsi wifi yang ada pada perangkat keras PC Laptop agar perangkat IP dapat berkomunikasi dengan *Bridge System*. Langkah awal untuk konfigurasi perangkat IP adalah menghubungkan wifi ke dalam jaringan yang sama dengan *Bridge System*. Untuk inisialisasi ip address yang digunakan adalah ipv6. Ipv6 mendukung perkembangan teknologi informasi seperti *Internet of Things*. Berikut pengalamatan Ipv6 menggunakan alamat *unicast*, alamat ipv6 dari perangkat IP adalah "fe80::" dan alamat ipv6 dari *network bridge system* adalah "fe80::fa1a:67ff:fe0a:744c". pada Langkah terakhir adaah tes koneksi dengan Ping IP adress Ipv6 tujuan. Berikut implementasi perangkat keras yang digunakan. Pada gambar 5.24 Implementasi perangkat keras pada sistem IP.



Gambar 5. 24 Implementasi Perangkat Keras komunikasi MQTT

2. Implementasi perangkat lunak

Tahapan implementasi perangkat lunak pada sistem komunikasi MQTT adalah menanamkan perangkat lunak yang digunakan pada *System* komunikasi MQTT dan memprogram agar antar perangkat IP dapat bertukar pesan. Langkah awal untuk implementasi perangkat lunak adalah menanamkan IDE Python untuk membuat program pada masing – masing perangkat IP. Langkah kedua tanamkan library untuk menjalankan code MQTT pada program pada masing – masing perangkat. Langkah ketiga membuat code program yang digunakan sistem IP untuk dapat mengirim dan menerima pesan menggunakan komunikasi MQTT. Berikut pengalamanan ipv6 menggunakan alamat *unicast*, alamat ipv6 dari perangkat IP adalah “fe80::” dan alamat ipv6 dari *network bridge system* adalah “fe80::fa1a:67ff:fe0a:744c”. Langkah terakhir tes komunikasi antar perangkat bluetooth dan jalankan program untuk mengirim dan menerima pesan atau data. Pada gambar 5.25 Implementasi program pada sistem perangkat IP.

```

GNU nano 2.2.6      File: client.py

import paho.mqtt.client as mqtt
import signal
import sys

brokerIP = "fe80::fa1a:67ff:fe0a:744c%eth0"

# Untuk handle interrupt Keyboard Ctrl+C
def signal_handler(signal, frame):
    print("...Exit process")
    sys.exit(1)

# Callback fungsi ketika subscribe
def on_subscribe(bttooth, obj, mid, granted_qos):
    print("Waiting bluetooth client... (Press CTRL+C twice$

# Callback fungsi ketika connect
def on_connect(bttooth, obj, flags, rc):
    status = ""

[ Read 57 lines ]
^G Get Hel^O WriteOu^R Read Fi^Y Prev Pa^K Cut Tex^C Cur Pos
^X Exit    ^J Justify^W Where I^V Next Pa^U UnCut T^T To Spell
    
```

Gambar 5. 25 Implementasi Program pada Sistem MQTT



BAB 6 PENGUJIAN

Bab pengujian menguraikan proses pengujian dan analisa terhadap sistem setelah proses perancangan dan implementasi telah dilakukan. Proses pengujian dilakukan untuk mengetahui kebutuhan sistem yang meliputi kebutuhan fungsional dan kebutuhan non fungsional sistem telah terpenuhi. Proses pengujian yang dilakukan akan dibagi menjadi beberapa proses sesuai kebutuhan. Hasil pengujian yang didapatkan akan dijelaskan dalam proses analisis untuk mendapatkan data hasil pengujian

6.1 Pengujian Sistem Bluetooth

6.1.1 Tujuan Pengujian

Pengujian perangkat bluetooth dapat mengirimkan data atau pesan dilakukan untuk mendapatkan informasi pesan atau data yang akan dikirim diawal. Tujuan yang harus dicapai dalam pengujian ini adalah perangkat Non IP bluetooth dapat mengirimkan pesan atau data ke *network bridge system* menggunakan komunikasi sinyal bluetooth dengan protokol RFCOMM.

6.1.2 Pengujian

Dalam pengujian fungsionalitas sistem perangkat bluetooth dapat mengirimkan data atau pesan dan membalas pesan ack dilakukan 2 prosedur sebagai berikut.

1. Prosedur tes koneksi antara Perangkat non IP Bluetooth dengan *network bridge system*

Pada tahap prosedur pengujian ini akan diuji apakah perangkat non IP bluetooth sudah terhubung dengan bluetooth dongle yang ada di *network bridge system*. Pengujian dilakukan dengan melakukan tes koneksi ping MAC Address tujuan dari masing - masing. Jika terdapat *Reply* balasan dari tes koneksi ping tersebut maka sudah dipastikan bahwa antar perangkat sudah terhubung.

2. Prosedur pengujian sistem bluetooth

Pada tahap prosedur pengujian ini akan dibuat dua skenario pengujian pada sistem bluetooth yaitu skenario 1 ketika data yang dikirimkan dari perangkat bluetooth dan skenario 2 ketika data yang dikirimkan dari perangkat IP, perangkat bluetooth harus dapat membalas berupa pesan ack yang berisi bahwa data yang dikirmkan sudah diterima.

Pelaksanaan pengujian dilakukan dengan pengujian terhadap sistem yang sudah diimplementasikan. Pelaksanaan ujian dilakukan dengan melakukan tes koneksi dan pengujian sistem bluetooth dapat meangani berbagai macam komunikasi. Berikut pelaksanaan pengujian dijelaskan pada gambar 6.1, gambar 6.2 dan gambar 6.3

Network Bridge System

```

pi@raspberrypi ~$ sudo l2ping 24:FD:52:5A:AC:59
Ping: 24:FD:52:5A:AC:59 from 00:15:83:3D:0A:57 (data size 44) ...
44 bytes from 24:FD:52:5A:AC:59 id 0 time 28.74ms
44 bytes from 24:FD:52:5A:AC:59 id 1 time 11.90ms
44 bytes from 24:FD:52:5A:AC:59 id 2 time 25.04ms
44 bytes from 24:FD:52:5A:AC:59 id 3 time 13.78ms
Recv failed: Connection reset by peer
pi@raspberrypi ~$ sudo l2ping 24:FD:52:5A:AC:59
Ping: 24:FD:52:5A:AC:59 from 00:15:83:3D:0A:57 (data size 44) ...
44 bytes from 24:FD:52:5A:AC:59 id 0 time 29.60ms
44 bytes from 24:FD:52:5A:AC:59 id 1 time 8.65ms
44 bytes from 24:FD:52:5A:AC:59 id 2 time 22.51ms
44 bytes from 24:FD:52:5A:AC:59 id 3 time 12.51ms
44 bytes from 24:FD:52:5A:AC:59 id 4 time 12.61ms
44 bytes from 24:FD:52:5A:AC:59 id 5 time 25.12ms
44 bytes from 24:FD:52:5A:AC:59 id 6 time 27.64ms
44 bytes from 24:FD:52:5A:AC:59 id 7 time 16.38ms
44 bytes from 24:FD:52:5A:AC:59 id 8 time 7.65ms
44 bytes from 24:FD:52:5A:AC:59 id 9 time 10.17ms
44 bytes from 24:FD:52:5A:AC:59 id 10 time 10.07ms
44 bytes from 24:FD:52:5A:AC:59 id 11 time 23.88ms
44 bytes from 24:FD:52:5A:AC:59 id 12 time 28.91ms
44 bytes from 24:FD:52:5A:AC:59 id 13 time 17.60ms
    
```

Perangkat Bluetooth

```

lilis@lilis-Lenovo-G400s: ~$ sudo l2ping 00:15:83:3D:0A:57
Ping: 00:15:83:3D:0A:57 from 24:FD:52:5A:AC:59 (data size 44) ...
44 bytes from 00:15:83:3D:0A:57 id 0 time 29.45ms
44 bytes from 00:15:83:3D:0A:57 id 1 time 29.81ms
44 bytes from 00:15:83:3D:0A:57 id 2 time 29.86ms
44 bytes from 00:15:83:3D:0A:57 id 3 time 29.76ms
44 bytes from 00:15:83:3D:0A:57 id 4 time 29.76ms
44 bytes from 00:15:83:3D:0A:57 id 5 time 30.00ms
44 bytes from 00:15:83:3D:0A:57 id 6 time 29.78ms
44 bytes from 00:15:83:3D:0A:57 id 7 time 29.98ms
44 bytes from 00:15:83:3D:0A:57 id 8 time 29.85ms
44 bytes from 00:15:83:3D:0A:57 id 9 time 29.83ms
44 bytes from 00:15:83:3D:0A:57 id 10 time 28.52ms
44 bytes from 00:15:83:3D:0A:57 id 11 time 29.90ms
44 bytes from 00:15:83:3D:0A:57 id 12 time 29.71ms
44 bytes from 00:15:83:3D:0A:57 id 13 time 29.91ms
44 bytes from 00:15:83:3D:0A:57 id 14 time 29.95ms
44 bytes from 00:15:83:3D:0A:57 id 15 time 29.84ms
44 bytes from 00:15:83:3D:0A:57 id 16 time 29.84ms
44 bytes from 00:15:83:3D:0A:57 id 17 time 29.65ms
44 bytes from 00:15:83:3D:0A:57 id 18 time 29.87ms
44 bytes from 00:15:83:3D:0A:57 id 19 time 29.85ms
44 bytes from 00:15:83:3D:0A:57 id 20 time 29.82ms
44 bytes from 00:15:83:3D:0A:57 id 21 time 29.87ms
    
```

Gambar 6. 1 Tes Koneksi

Berikut penjelasan gambar 6.1 pengujian pada tes koneksi pada perangkat bluetooth dilakukan untuk membangun koneksi antar perangkat bluetooth. Tes koneksi dilakukan dengan melakukan ping MAC Address tujuan. MAC Address dari perangkat bluetooth adalah “24:FD:52:5A:AC:59” dan MAC Address dari bluetooth dongle dari *network bridge system* adalah “00:15:83:3D:0A:57”. Perintah yang dilakukan untuk tes koneksi antar perangkat adalah “\$sudo l2ping (MAC Address tujuan)”. Data yang dikirim dalam tes ping koneksi adalah 44 bytes.

A. Skenario 1

Pengujian pada skenario 1 dilakukan dengan mengirimkan data awal dari perangkat bluetooth. Perangkat bluetooth harus dapat mengirimkan data ke *network bridge system*, menampilkan data yang sudah dikirim dan menerima data balasan dari perangkat IP. Berikut hasil pengujian sistem bluetooth pada skenario 1 dilihat dari gambar 6.2



```
lilis@lilis-Lenovo-G400s: ~  
lilis@lilis-Lenovo-G400s:~$ sudo python rfcomm-client.py  
Masukkan data:  
data1  
sent [data1]  
received [Pesan "data1" sudah kami terima]
```

Gambar 6. 2 Pengujian Sistem Bluetooth Skenario 1

B. Skenario 2

Pengujian pada skenario 2 dilakukan dengan mengirimkan data awal dari perangkat IP. Perangkat bluetooth harus dapat menerima data ke *network bridge system*, menampilkan data yang diterima dan mengirimkan data balasan ke perangkat IP. Berikut hasil pengujian sistem bluetooth pada skenario 2 dilihat dari gambar 6.3

```
lilis@lilis-Lenovo-G400s: ~  
lilis@lilis-Lenovo-G400s:~$ sudo python rfcomm-client-2.py  
received [data 1]  
sent [Pesan [data 1] sudah diterima]
```

Gambar 6. 3 Pengujian Sistem Bluetooth Skenario 2

6.1.3 Hasil Pengujian

Berdasarkan ada gambar 6.1 telah dilakukan tes koneksi antara perangkat non IP bluetooth dengan *network bridge system* dengan hasil bahwa masing – masing perangkat mendapatkan *reply* balasan dari perangkat yang dituju. Pada gambar 6.2 dan gambar 6.3 telah dilakukan pengiriman pesan atau data dari perangkat IP bluetooth dan perangkat bluetooth dapat membalas data balasan

berupa pesan ack dengan hasil bahwa pesan atau data dapat dikirimkan dan diterima dapat terlaksana.

Setelah dilakukan pengujian fungsionalitas *System* perangkat non IP bluetooth didapatkan hasil bahwa perangkat Non IP bluetooth dapat mengirimkan pesan atau data ke *Network bridge system* dan menerima data dari perangkat IP dengan membalas data balasan berupa pesan ACK. Berdasarkan hasil pengujian yang dilakukan untuk menguji fungsionalitas sistem agar perangkat Non IP bluetooth dapat mengirim pesan atau data ke *Network bridge system* dan menerima data dari perangkat IP dengan membalas data balasan berupa pesan ACK didapatkan bahwa kebutuhan fungsionalitas sistem dapat terpenuhi.

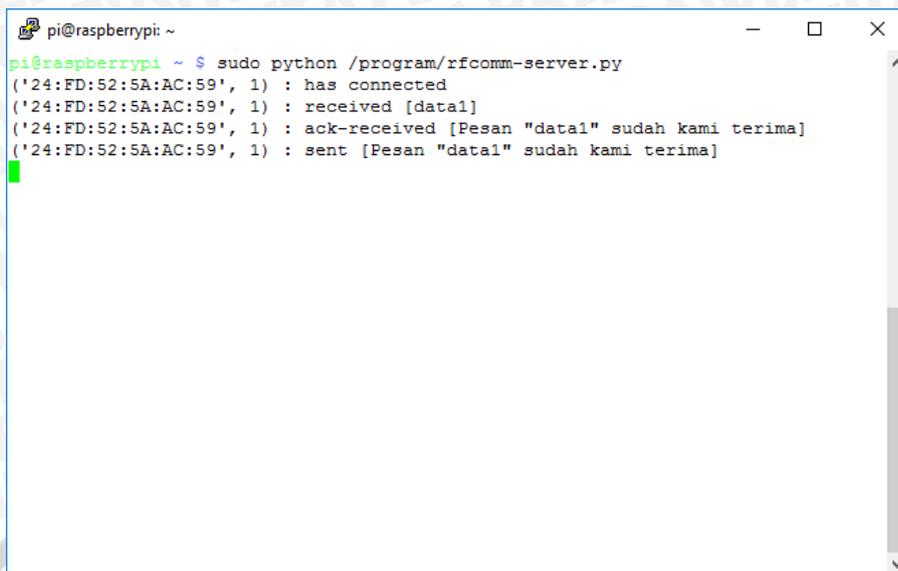
6.2 Pengujian Sistem *Network bridge system*

6.2.1 Tujuan Pengujian

Pengujian sistem *Network bridge system* dapat menerima data dan meneruskan data ke perangkat IP dilakukan untuk menguji fungsionalitas sistem *Network bridge system* dapat mengontrol pesan dari perangkat bluetooth. Tujuan yang harus dicapai dalam pengujian ini adalah *network bridge system* dapat menerima pesan atau data dari perangkat non IP bluetooth dan dapat meneruskan pesan atau data tersebut ke perangkat IP berupa topik pesan menggunakan komunikasi MQTT.

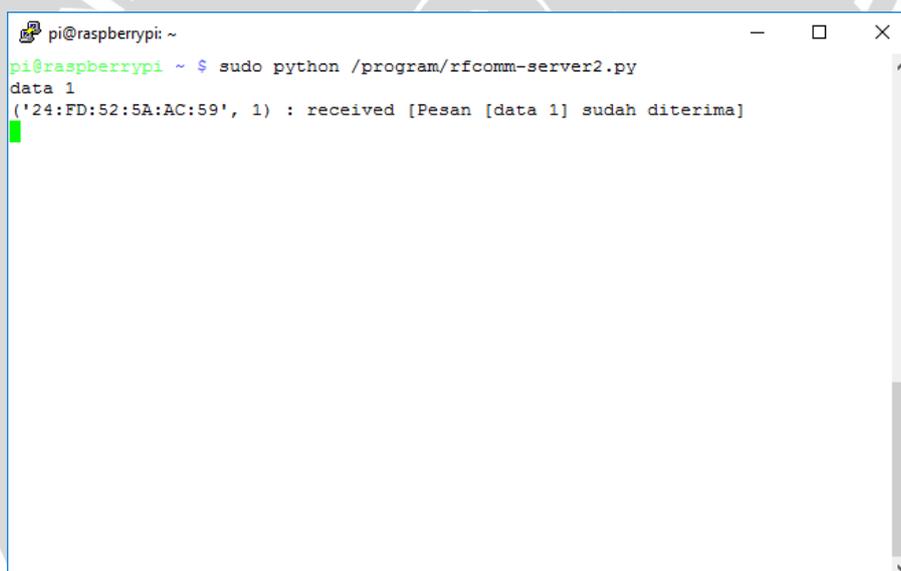
6.2.2 Pengujian

Prosedur Pengujian fungsionalitas sistem *Network bridge system* dilakukan dengan 2 skenario. Pengujian skenario 1 dan skenario 2 dilakukan dengan menguji *network bridge system* dapat menerima dan meneruskan data ke perangkat tujuan dilakukan dengan cara melihat interface Command Line Interface dari *network bridge system*. Ketika *network bridge system* dijalankan tampilan running program dan menunggu data diterima. Berikut pelaksanaan pengujian dijelaskan pada gambar 6.4 dan gambar 6.5.



```
pi@raspberrypi: ~  
pi@raspberrypi ~ $ sudo python /program/rfcomm-server.py  
(('24:FD:52:5A:AC:59', 1) : has connected  
(('24:FD:52:5A:AC:59', 1) : received [data1]  
(('24:FD:52:5A:AC:59', 1) : ack-received [Pesan "data1" sudah kami terima]  
(('24:FD:52:5A:AC:59', 1) : sent [Pesan "data1" sudah kami terima]
```

Gambar 6. 4 *Network bridge system* Mengontrol data dari Perangkat Bluetooth Skenario 1



```
pi@raspberrypi: ~  
pi@raspberrypi ~ $ sudo python /program/rfcomm-server2.py  
data 1  
(('24:FD:52:5A:AC:59', 1) : received [Pesan [data 1] sudah diterima]
```

Gambar 6. 5 *Network bridge system* Mengontrol data dari Perangkat IP Skenario 2

6.2.3 Hasil Pengujian

Berdasarkan ada gambar 6.4 *network bridge system* dapat menerima pesan dari perangkat bluetooth dengan menampilkan data yang dikirim dari perangkat bluetooth dan dapat meneruskan pesan tersebut ke perangkat IP. Dan pada gambar 6.5 *network bridge system* dapat menerima pesan dari perangkat IP dengan menampilkan data yang dikirim dari perangkat bluetooth dan dapat meneruskan pesan tersebut ke perangkat bluetooth. Sehingga untuk pengujian *network bridge system* dapat menerima pesan dan meneruskan pesan ke perangkat tujuan dikatakan berhasil dan telah memenuhi kebutuhan fungsionalitas sistem.

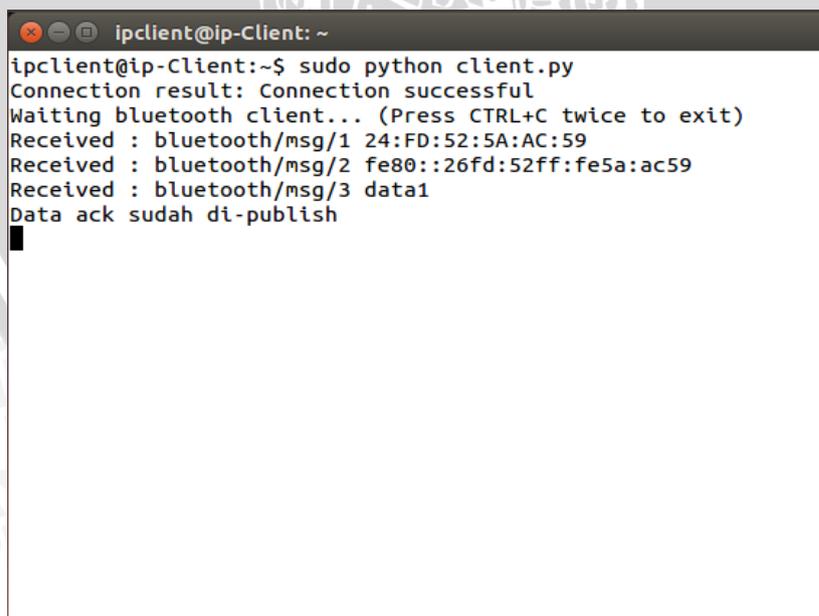
6.3 Pengujian Sistem Perangkat IP

6.3.1 Tujuan Pengujian

Pengujian sistem Perangkat IP dilakukan dengan pengujian 2 skenario. Skenario 1 pengujian dilakukan dengan menguji perangkat IP dapat menerima pesan atau data dari *network bridge system* yang berupa topik pesan berupa isi pesan dari perangkat non IP bluetooth dan mengembalikan pesan atau data tersebut dengan balasan bahwa pesan tersebut sudah diterima. Skenario 2 pengujian dilakukan dengan menguji perangkat IP dapat mengirim pesan atau data dari *network bridge system* yang berupa topik pesan menuju perangkat non IP bluetooth dan menerima pesan atau data dengan balasan bahwa pesan tersebut sudah diterima. Tujuan yang harus dicapai dalam pengujian ini adalah Perangkat IP dapat menerima pesan dan mengirim pesan menuju perangkat bluetooth menggunakan komunikasi MQTT melalui *network bridge system*.

6.3.2 Pengujian

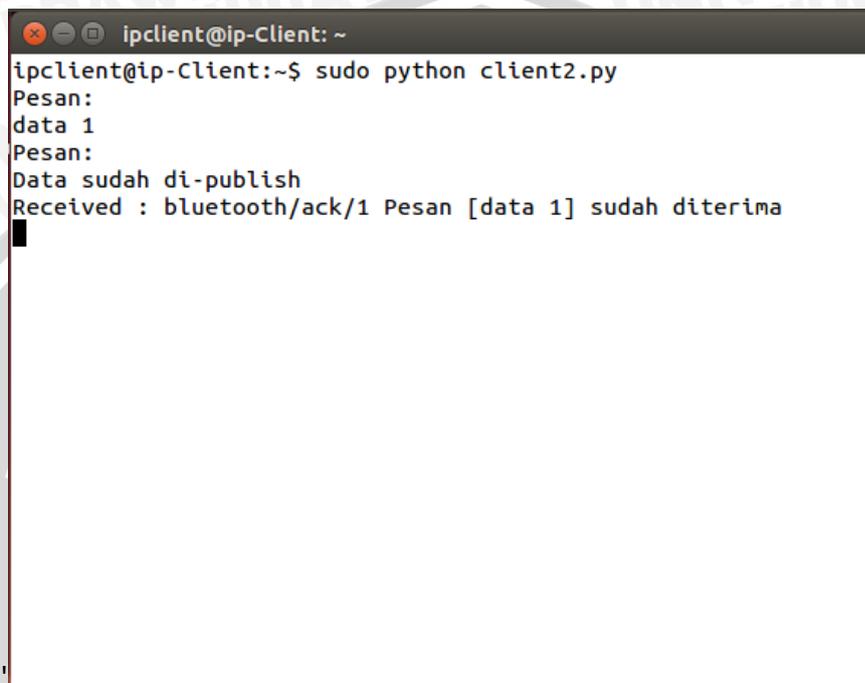
Pengujian pada skenario 1 dilakukan setelah perangkat bluetooth mengirim pesan ke *network bridge system* dan meneruskan pesan tersebut. Langkah selanjutnya *network bridge system publish* pesan berbasis topik melalui *broker*. Perangkat IP *subscribe* topik yang telah di *publish* ke *broker*. Topik yang berisi pesan tersebut akan ditampilkan ke dalam tampilan program perangkat IP. Setelah pesan diterima perangkat IP harus dapat memberikan balasan bahwa pesan sudah diterima. Perangkat IP *publish* pesan balasan berbasis topik menuju perangkat *network bridge system* melalui *broker* dan menampilkan pesan balasan sudah di *publish* pada tampilan program. Berikut tampilan pengujian sistem dijelaskan pada Gambar 6.6



```
ipclient@ip-Client: ~  
ipclient@ip-Client:~$ sudo python client.py  
Connection result: Connection successful  
Waiting bluetooth client... (Press CTRL+C to exit)  
Received : bluetooth/msg/1 24:FD:52:5A:AC:59  
Received : bluetooth/msg/2 fe80::26fd:52ff:fe5a:ac59  
Received : bluetooth/msg/3 data1  
Data ack sudah di-publish  
█
```

Gambar 6. 6 Perangkat IP *subscribe* topik

Pengujian pada skenario 2 dilakukan dengan perangkat IP mengirim pesan ke *network bridge system* dan meneruskan pesan tersebut ke perangkat bluetooth. Langkah selanjutnya *network bridge system* mengirim pesan tersebut menuju perangkat bluetooth tujuan. Perangkat IP *subscribe* topik yang telah di *publish* ke *broker*. Topik yang berisi pesan tersebut akan ditampilkan ke dalam tampilan program perangkat IP. Berikut tampilan pengujian sistem dijelaskan pada Gambar 6.7



```
ipclient@ip-Client: ~
ipclient@ip-Client:~$ sudo python client2.py
Pesan:
data 1
Pesan:
Data sudah di-publish
Received : bluetooth/ack/1 Pesan [data 1] sudah diterima
```

Gambar 6. 7 Perangkat IP *Publish* topik balasan

6.3.3 Hasil Pengujian

Berdasarkan pengujian yang telah dilakukan dan dijelaskan pada gambar 6.4 bahwa perangkat IP dapat menerima pesan yang dikirimkan oleh perangkat bluetooth melalui *network bridge system* menunjukkan bahwa pengujian fungsionalitas sistem berhasil. Pada Gambar 6.7 dijelaskan bahwa perangkat IP dapat mengirim pesan menuju perangkat bluetooth melalui *network bridge system* dan menerima pesan balasan berupa ack pesan telah diterima menunjukkan bahwa pengujian fungsionalitas sistem berhasil.

6.3 Pengujian Performa

Pengujian dilakukan dengan mengukur jarak komunikasi dengan parameter satuan ukur meter dengan alat ukur meteran dan menguji performa *delay* dari sistem. Pengukuran jarak komunikasi diimplementasikan di Gedung H Fakultas Ilmu Komputer Universitas Brawijaya. Parameter yang digunakan dalam pengujian *delay* adalah jarak komunikasi dilakukan dengan jarak 5 meter sampai 50 meter, panjang data yang dikirimkan adalah 12 bytes dengan pesan “data skripsi”. Pengujian diukur menggunakan *software wireshark*. Berikut hasil

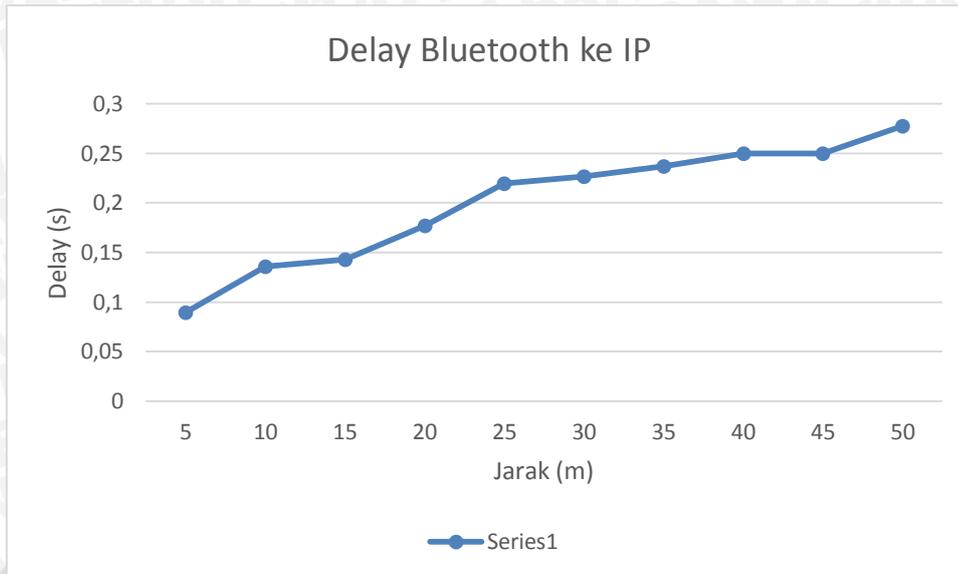
pengujian pengukuran jarak komunikasi bluetooth dan *delay* dijelaskan pada tabel 6.1

Tabel 6. 1 Pengujian Performa Bluetooth

NO.	Jarak (m)	Hasil Pengujian		
		Pengiriman	Delay (s)	
			Bluetooth ke IP	IP ke Bluetooth
1.	5	Berhasil	0,089	0,040
2.	10	Berhasil	0,136	0,172
3.	15	Berhasil	0,142	0,2102
4.	20	Berhasil	0,177	0,213
5.	25	Berhasil	0,219	0,258
6.	30	Berhasil	0,227	0,2708
7.	35	Berhasil	0,237	0,280
8.	40	Berhasil	0,2501	0,302
9.	45	Berhasil	0,2501	0,317
10.	50	Berhasil	0,278	1,649

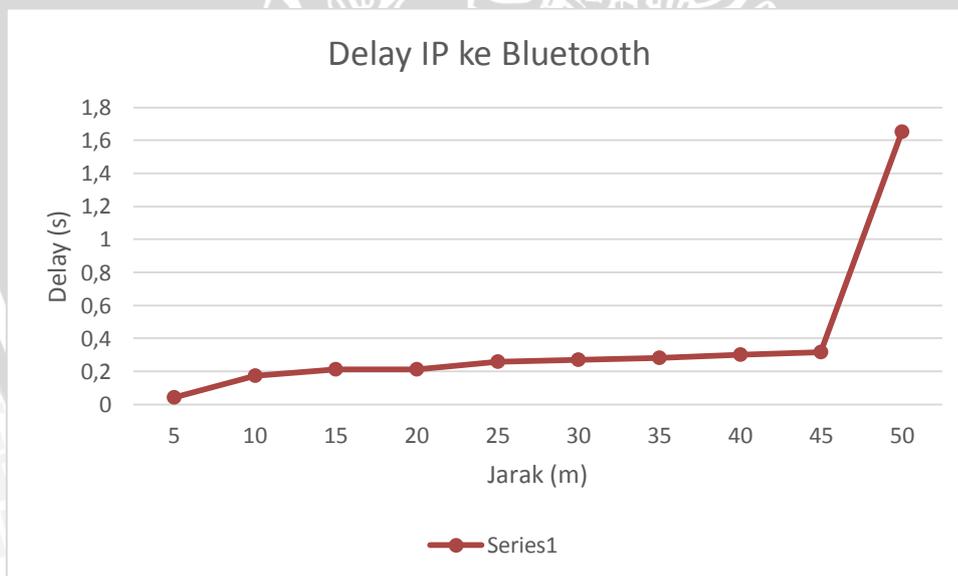
6.3.4 Hasil Pengujian

Berdasarkan hasil analisa pengujian yang dilakukan dengan menguji jarak komunikasi perangkat Bluetooth dengan *network bridge system* didapatkan hasil sesuai dengan tabel 6.1. Hasil menunjukkan performa komunikasi bluetooth dapat mengirimkan pesan dan bertukar pesan hingga jarak 50 meter. Pengiriman data pada percobaan pertama gagal pada jarak komunikasi 45 dan 50 meter karena perangkat bluetooth kehilangan sinyal bluetooth dan tidak terhubung ke *network bridge system* dan beberapa faktor terhalangnya sinyal bluetooth untuk terhubung antara perangkat bluetooth dan *network bridge system*. Pada percobaan kedua pengiriman dan penerimaan pesan dari perangkat bluetooth dan *network bridge system* berhasil sampai jarak 50 meter karena sinyal frekuensi bluetooth terhubung dengan baik tanpa ada halangan atau interferensi. Berikut Hasil grafik analisa hasil *delay* selama pengujian pada gambar 6.8 Dan gambar 6.9



Gambar 6. 8 Grafik Delay Bluetooth ke IP

Pada gambar 6. terlihat grafik naik yang menunjukkan bahwa ketika jarak semakin jauh maka *delay* semakin tinggi. Hal itu dikarenakan sinyal gelombang frekuensi dari perangkat bluetooth semakin menjauh dan gangguan interferensi semakin tinggi.



Gambar 6. 9 Grafik Delay IP ke Bluetooth

Pada gambar 6. terlihat grafik naik yang menunjukkan bahwa ketika jarak semakin jauh maka *delay* semakin tinggi. Hal itu dikarenakan sinyal gelombang frekuensi dari perangkat bluetooth semakin menjauh dan gangguan interferensi semakin tinggi. Kenaikan signifikan terjadi ketika jarak 50 meter *delay* dalam pengiriman data dari perangkat IP menuju perangkat bluetooth diatas 1 detik. Sinyal frekuensi bluetooth hanya dapat menjangkau kurang lebih 50 meter.

BAB 7 PENUTUP

Bab penutup akan dijelaskan uraian kesimpulan berdasarkan hasil perancangan, implementasi dan pengujian yang telah dilakukan.

7.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi dan pengujian yang telah dilakukan didapatkan kesimpulan sebagai berikut.

1. Perangkat bluetooth memungkinkan berkomunikasi dan bertukar data dengan perangkat IP melalui *network bridge system*
2. *Network bridge system* mampu menangani pertukaran pesan dan komunikasi antara perangkat IP dan perangkat bluetooth.
3. Perangkat bluetooth mampu berkomunikasi dengan *network bridge system* dengan protokol komunikasi bluetooth dan menampilkan hasil data yang dikirim dan diterima.
4. Perangkat IP mampu berkomunikasi dengan *network bridge system* dengan protokol komunikasi MQTT dan menampilkan hasil data yang dikirim dan diterima.
5. Performa sistem dalam hal jarak komunikasi antara perangkat bluetooth didapatkan hasil perangkat bluetooth mampu berkomunikasi dengan *network bridge system* hingga jarak 50 meter
6. Performa sistem dalam hal reliabilitas ketika jarak komunikasi perangkat semakin jauh maka *delay* yang dihasilkan akan semakin tinggi.

7.2 Saran

Saran yang diharapkan dalam pengembangan sistem ini adalah sistem dapat dikembangkan lebih lanjut pada implementasi sistem berbasis aplikasi dan antarmuka berbasis graphical. Sistem ini juga diharapkan dapat dikembangkan untuk komunikasi yang lebih luas dan dapat menghubungkan banyak perangkat bluetooth dan perangkat IP. Sehingga nantinya sistem dapat diimplementasikan pada masyarakat luas dan dapat dirasakan manfaatnya.

DAFTAR PUSTAKA

- Baek, Y. M., Ahn, S. C. & Kwon, Y.-M., 2010. UPnP Network Bridge for Supporting Interoperability. *IEEE Transactions on Consumer Electronics*, 56(4), pp. 2226-2232.
- Communities, U., 2016. *Device to Cloud: MQTT and the power of topic notation*. [Online] Available at: <http://www.embedded.com/> [Diakses Monday June 2016].
- Eka Pratama, A. & I, P., 2014. *Handbook Jaringan Komputer*. Bandung: Informatika.
- Element 14, 2012. *Quick Start Guide The Raspberry Pi – Single Board Computer*, s.l.: s.n.
- Foundation, R. P., 2016. *Raspberry Pi*. [Online] Available at: <https://www.raspberrypi.org> [Diakses Monday June 2016].
- Huang, A. & Rudolph, L., 2005. *Bluetooth for Programmers*. 1 penyunt. Cambridge: TODO.
- Inc, C. S., 2011. *Routing and Bridging Guide, Cisco*. 1 penyunt. San Jose: Americas Headquarters.
- Inc, I., 2008. *The Source for IPv6 Information, Training, Consulting & Hardware*. [Online] Available at: <http://ipv6.com/> [Diakses Mondy June 2016].
- Inc, P., 2016. *What is MQTT*. [Online] Available at: <https://www.pubnub.com> [Diakses Monday June 2016].
- Kurawar, A., Koul, A. & Tukaram Patill, P. V., 2014. Survey of Bluetooth appication. *International Journal of Advanced Research in Computer Engineering and Technology*, 3(8), pp. 2832-2837.
- Mqtt.org, 2016. *MQTT*. [Online] Available at: <http://mqtt.org/> [Diakses Monday June 2016].
- Pohl, K., 2010. The Requirements Engineering Framework . Dalam: *Requirement Engineering Fundamentals, Principles, and Techniques* . s.l.:Springer, pp. 41-57.
- R. Carpeno, C. & Woodward, A., 2014. A survey of IPV6 address usage in the public. *Security Research Institute Conferences*, 1(1), pp. 91-99.
- Seong-Joong Kim, e. a., 2011. *Network bridge system for Interoperation of ZigBee-UPnP Network*. *Fourth International Conference on Intelligent Networks*.
- Vermaat & Cashman, S., 2007. *DISCOVERING COMPUTERS=MENJELAJAH DUNIA KOMPUTER FUNDAMENTAL ED 3*. 3 penyunt. Jakarta: Salemba Infotek.
- Wibhowo, C. & Sanjaya, R., 2011. *Stimulasi Kecerdasan Anak Menggunakan Teknologi Informasi*. Jakarta: PT Elex Media Komputindo.

LAMPIRAN

Lampiran 1 Source Code *Client* Bluetooth Skenario 1

```
1 from bluetooth import *
2 import threading
3
4 bd_addr = "00:15:83:3D:0A:57"
5
6 port = 1
7
8 # Create the client socket
9 sock=BluetoothSocket( RFCOMM )
10 sock.connect((bd_addr, port))
11
12 class receiverThread(threading.Thread):
13     def __init__(self,sock):
14         threading.Thread.__init__(self)
15         self.sock = sock
16     def run(self):
17         while True:
18             data = self.sock.recv(1024)
19             if len(data) == 0: break
20             print "received [%s]" % data
21
22 receiver = receiverThread(sock)
23 receiver.setDaemon(True)
24 receiver.start()
25 print "Masukkan data:"
26 while True:
27     data = raw_input()
28     if len(data) == 0: break
29     sock.send(data)
30     print "sent [%s]" % data
31 sock.close()
```

Lampiran 2 Source Code *Network Bridge System* Skenario 1

```

1 import paho.mqtt.client as mqtt
2 import signal
3 import sys
4 import threading
5 from bluetooth import *
6
7 # Inisialisasi variabel global
8 pesan_ack = ""
9
10 # Inisiasi client bluetooth publisher dengan ID "btclient"
11 bttooth = mqtt.Client("btclient", clean_session=True)
12
13 # Koneksi ke broker
14 bttooth.connect("::1", 1883)
15
16 # Subscribe
17 bttooth.subscribe("bluetooth/ack/", qos=1)
18
19 def mac_to_ipv6_linklocal(client_mac):
20     mac_value = int(client_mac.translate(None, ' .:-'), 16)
21     # Remove the most common delimiters; dots, dashes, etc.
22     # Split out the bytes that slot into the IPv6 address
23     # XOR the most significant byte with 0x02, inverting
24     # Universal / Local bit
25     high2 = mac_value >> 32 & 0xffff^0x0200
26     high1 = mac_value >> 24 & 0xff
27     low1 = mac_value >> 16 & 0xff
28     low2 = mac_value & 0xffff
29     return
30     'fe80::{:04x}::{:02x}ff:fe{:02x}::{:04x}'.format(high2, high1,
31     low1, low2)
32
33 def pub_ack(socketid, socketinfo, pesanack):
34     socketid.send(pesanack)
35     print socketinfo, ": sent [%s]" % pesanack
36
37 class mainThread(threading.Thread):
38     def __init__(self, sock, client_info):

```

```
38         threading.Thread.__init__(self)
39         self.sock = sock
40         self.client_info = client_info
41         def on_message(bttooth, obj, msg):
42             pesan_ack = str(msg.payload)
43             print obj, ": ack-received [%s]" %
pesan_ack
44             pub_ack(self.sock, self.client_info,
pesan_ack)
45             self.on_message = on_message
46         def run(self):
47             try:
48                 # Daftarkan fungsi callback-nya
49                 bttooth.on_message = self.on_message
50                 bttooth.loop_start()
51                 while True:
52                     data = self.sock.recv(1024)
53                     if len(data) == 0: break
54                     print self.client_info, ":
received [%s]" % data
55                     bttooth.publish("bluetooth/msg/1", payload=str(client_mac),
qos=1)
56                                     address =
57                     mac_to_ipv6_linklocal(client_mac)
58                     bttooth.publish("bluetooth/msg/2", payload=str(address), qos=1)
59                     bttooth.publish("bluetooth/msg/3", payload=str(data), qos=1)
60                 except IOError:
61                     pass
62                 self.sock.close()
63                 print self.client_info, ": disconnected"
64
65         # Fungsi untuk handle interrupt CTRL+C
66         def signal_handler(signal, frame):
67             print('\nServer has stopped!')
68             sys.exit(0)
69
70         signal.signal(signal.SIGINT, signal_handler)
```

```

71 server_sock = BluetoothSocket( RFCOMM )
72 server_sock.bind(("", PORT_ANY))
73 server_sock.listen(1)
74 port = server_sock.getsockname()[1]
75 while True:
76     client_sock, client_info = server_sock.accept()
77     client_mac = client_info[0]
78     run = mainThread(client_sock, client_info)
79     run.setDaemon(True)
80     run.start()
81
82 client_sock.close()
83 server_sock.close()

```

Lampiran 3 Source Code *Client* IP Skenario 1

```

1 import paho.mqtt.client as mqtt
2 import signal
3 import sys
4
5 brokerIP = "fe80::fala:67ff:fe0a:744c%eth0"
6
7 # Untuk handle interrupt Keyboard Ctrl+C
8 def signal_handler(signal, frame):
9     print("...Exit process")
10    sys.exit(1)
11
12 # Callback fungsi ketika subscribe
13 def on_subscribe(bttooth, obj, mid, granted_qos):
14    print("Waiting bluetooth client... (Press CTRL+C twice
15    to exit)")
16
17 # Callback fungsi ketika connect
18 def on_connect(bttooth, obj, flags, rc):
19    status = ""
20    if (rc == 0): status = "Connection successful"
21    elif (rc == 1): status = "Connection refused - incorrect
22    protocol version"
23    elif (rc == 2): status = "Connection refused - invalid
24    client identifier"

```

```
22     elif (rc == 3): status = "Connection refused - server
unavailable"
23     elif (rc == 4): status = "Connection refused - bad username
or password"
24     elif (rc == 5): status = "Connection refused - not
authorised"
25     else : status = "Error"
26     print("Connection result: "+status)
27
28 # Callback fungsi ketika menerima message
29 def on_message(bttooth, obj, msg):
30     print("Received : "+msg.topic+" "+str(msg.payload))
31     if(msg.topic == "bluetooth/msg/3"):
32 #         pesan = raw_input("Pesan: ")
33         pesan = "Sudah diterima"
34         bttooth.publish("bluetooth/ack/1",
payload=str(pesan), qos=1)
35
36 def on_publish(bttooth, obj, mid):
37     print("Data ack sudah di-publish")
38
39 # Inisiasi client dengan ID "client"
40 bttooth = mqtt.Client("ipclient", clean_session=True)
41
42 # Interrupt CTRL+C
43 signal.signal(signal.SIGINT, signal_handler)
44
45 # Registrasi fungsi callback yang dipanggil setiap ada event
tertentu
46 bttooth.on_subscribe = on_subscribe
47 bttooth.on_message = on_message
48 bttooth.on_connect = on_connect
49 bttooth.on_publish = on_publish
```

Lampiran 4 Source Code *Client* Bluetooth Skenario 2

```
1 from bluetooth import *
2 import threading
3
4 bd_addr = "00:15:83:3D:0A:57"
5
6 port = 1
7
8 # Create the client socket
9 sock=BluetoothSocket( RFCOMM )
10 sock.connect((bd_addr, port))
11
12 #class senderThread(threading.Thread):
13 #     def __init__(self,sock):
14 #         threading.Thread.__init__(self)
15 #         self.sock = sock
16 #     def run(self):
17 #         while True:
18 #             if len(data) == 0: break
19 #             sock.send(data)
20 #             print "sent [%s]" % data
21
22 #sender = senderThread(sock)
23 #sender.setDaemon(True)
24 #sender.start()
25 while True:
26     data = sock.recv(1024)
27     if len(data) == 0: break
28     print "received [%s]" % data
29     pesan_ack = "Pesan [%s] sudah diterima" % data
30     sock.send(str(pesan_ack))
31     print "sent [%s]" % pesan_ack
32
33 sock.close()
```

Lampiran 5 Source Code *Network Bridge System* Skenario 2

```
1 import paho.mqtt.client as mqtt
2 import signal
3 import sys
4 import threading
5 from bluetooth import *
6
7 # Inisialisasi variabel global
8 pesan_ack = ""
9
10 # Inisiasi client bluetooth publisher dengan ID "btclient"
11 btooth = mqtt.Client("btclient", clean_session=True)
12
13 # Koneksi ke broker
14 btooth.connect("::1", 1883)
15
16 # Subscribe
17 btooth.subscribe("bluetooth/dariip/+", qos=1)
18
19 class mainThread(threading.Thread):
20     def __init__(self, sock, client_info):
21         threading.Thread.__init__(self)
22         self.sock = sock
23         self.client_info = client_info
24     def run(self):
25         try:
26             while True:
27                 data = self.sock.recv(1024)
28                 if len(data) == 0: break
29                 print self.client_info, ":
30 received [%s]" % data
31                 btooth.publish("bluetooth/ack/1",
payload=str(data), qos=1)
32             except IOError:
33                 pass
34                 self.sock.close()
35                 print self.client_info, ": disconnected"
36
37 # Fungsi untuk menghandle interrupt CTRL+C
```

```
38 def signal_handler(signal, frame):
39     print('\nServer has stopped!')
40     sys.exit(0)
41
42 signal.signal(signal.SIGINT, signal_handler)
43 server_sock = BluetoothSocket( RFCOMM )
44 server_sock.bind(("", PORT_ANY))
45 server_sock.listen(1)
46 port = server_sock.getsockname()[1]
47 while True:
48     bttooth.loop_start()
49     client_sock, client_info = server_sock.accept()
50     run = mainThread(client_sock, client_info)
51     run.setDaemon(True)
52     run.start()
53     def on_message(bttooth, obj, msg):
54         client_sock.send(str(msg.payload))
55         print(str(msg.payload))
56         bttooth.on_message = on_message
57
58 client_sock.close()
59 server_sock.close()
```

Lampiran 6 Source Code *Client IP* Skenario 2

```
1 import paho.mqtt.client as mqtt
2 import signal
3 import sys
4
5 brokerIP = "fe80::fala:67ff:fe0a:744c%eth0"
6
7 # Untuk handle interrupt Keyboard Ctrl+C
8 def signal_handler(signal, frame):
9     print("Program has stopped!")
10    sys.exit(1)
11
12 # Callback fungsi ketika menerima message
13 def on_message(bttooth, obj, msg):
14    print("Received : "+msg.topic+" "+str(msg.payload))
15
16 def on_publish(bttooth, obj, mid):
17    print("Data sudah di-publish")
18
19 # Inisiasi client dengan ID "client"
20 bttooth = mqtt.Client("ipclient", clean_session=True)
21
22 # Interrupt CTRL+C
23 signal.signal(signal.SIGINT, signal_handler)
24
25 # Registrasi fungsi callback yang dipanggil setiap ada event
    tertentu
26 bttooth.on_message = on_message
27 bttooth.on_publish = on_publish
28
29 # Koneksi ke broker
30 bttooth.connect(brokerIP, 1883)
31
32 #Subscribe bluetooth
33 bttooth.subscribe("bluetooth/ack+", qos=1)
34
35 while True:
36    # Start looping forever
```

```
37     btooth.loop_start()  
38     print("Pesan:")  
39     data = raw_input()  
40     if len(data) == 0: break  
41     btooth.publish("bluetooth/dariip/1", payload=str(data),  
                    qos=1)
```

