IMPLEMENTASI KOMUNIKASI MULTI-HOP MENGGUNAKAN METODE *CONTROLLED* *FLOODING* PADA *WIRELESS SENSOR NETWORK* BERBASIS LORA

SKRIPSI

Untuk memenuhi sebagian persyaratan   
memperoleh gelar Sarjana Komputer

Disusun oleh:

Zeddin Arief

NIM: 155150200111234



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS BRAWIJAYA

MALANG

2020

PENGESAHAN

IMPLEMENTASI KOMUNIKASI *MULTI-HOP* MENGGUNAKAN METODE *CONTROLLED FLOODING* PADA *WIRELESS SENSOR NETWORK* BERBASIS LORA

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan

memperoleh gelar Sarjana Komputer

Disusun Oleh :

Zeddin Arief

NIM: 155150200111234

Skripsi ini telah diuji dan dinyatakan lulus pada

2 Januari 2015

Telah diperiksa dan disetujui oleh:

|  |  |
| --- | --- |
| Dosen Pembimbing I  Nama Dosen Pembimbing 1  NIK: 123456789 | Dosen Pembimbing 2  Nama Dosen Pembimbing 2  NIK: - |

Mengetahui

Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D

NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

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

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, 1 Januari 2015



­

Zeddin Arief

NIM: 155150200111234

PRAKATA

Bagian ini memuat pernyataan resmi untuk menyampaikan rasa terima kasih penulis kepada berbagai pihak yang telah membantu penyelesaian skripsi ini. Nama-nama penerima ucapan terima kasih sebaiknya dituliskan lengkap, termasuk gelar akademik, dan pihak-pihak yang tidak terkait dihindari untuk dituliskan. Bahasa yang digunakan seharusnya mengikuti kaidah bahasa Indonesia yang baku. Prakata boleh diakhiri dengan paragraf yang menyatakan bahwa penulis menerima kritik dan saran untuk pengembangan penelitian selanjutnya. Terakhir, prakata ditutup dengan mencantumkan kota dan tanggal penulisan prakata, lalu diikuti dengan kata “Penulis”.

Malang, 1 Januari 2015

Zeddin Arief

email@domain.com

ABSTRAK

Zeddin Arief, Implementasi Komunikasi *Multi-Hop* Menggunakan Metode *Controlled Flooding* pada *Wireless Sensor Network* Berbasis *LoRa*

Pembimbing: Nama Pembimbing 1 dan Nama Pembimbing 2

Bagian ini diisi dengan abstrak dalam Bahasa Indonesia. Abstrak adalah uraian singkat (umumnya 200-300 kata) yang merupakan intisari dari sebuah skripsi. Abstrak membantu pembaca untuk mendapatkan gambaran secara cepat dan akurat tentang isi dari sebuah skripsi. Melalui abstrak, pembaca juga dapat menentukan apakah akan membaca skripsi lebih lanjut. Oleh karena itu, abstrak sebaiknya memberikan gambaran yang padat tetapi tetap jelas dan akurat tentang (1) apa dan mengapa penelitian dikerjakan: sedikit latar belakang, pertanyaan atau masalah penelitan, dan/atau tujuan penelitian; (2) bagaimana penelitian dikerjakan: rancangan penelitian dan metodologi/metode dasar yang digunakan dalam penelitian; (3) hasil penting yang diperoleh: temuan utama, karakteristik artefak, atau hasil evaluasi artefak yang dibangun; (4) hasil pembahasan dan kesimpulan: hasil dari analisis dan pembahasan temuan atau evaluasi artefak yang dibangun, yang dikaitkan dengan pertanyaan/tujuan penelitian.

Yang harus dihindari dalam sebuah abstrak diantaranya (1) penjelasan latar belakang yang terlalu panjang; (2) sitasi ke pustaka lainnya; (3) kalimat yang tidak lengkap; (3) singkatan, jargon, atau istilah yang membingungkan pembaca, kecuali telah dijelaskan dengan baik; (4) gambar atau tabel; (5) angka-angka yang terlalu banyak.

Di akhir abstrak ditampilkan beberapa kata kunci (normalnya 5-7) untuk membantu pembaca memposisikan isi skripsi dengan area studi dan masalah penelitian. Kata kunci, beserta judul, nama penulis, dan abstrak biasanya dimasukkan dalam basis data perpustakaan. Kata kunci juga dapat diindeks dalam basis data sehingga dapat digunakan untuk proses pencarian tulisan ilmiah yang relevan. Oleh karena itu pemilihan kata kunci yang sesuai dengan area penelitian dan masalah penelitian cukup penting. Pemilihan kata kunci juga bisa didapatkan dari referensi yang dirujuk.

Kata kunci: abstrak, skripsi, intisari, kata kunci, artefak

ABSTRACT

Zeddin Arief, *Implementation of Multi-Hop Communication Using the Controlled Flooding Method on LoRa-Based Wireless Sensor Networks*

Supervisors: First Supervisor’s Name *and* Second Supervisor’s Name

The absract of your skripsi in English is written here.

DAFTAR ISI

[PENGESAHAN ii](#_Toc30856340)

[PERNYATAAN ORISINALITAS iii](#_Toc30856341)

[PRAKATA iv](#_Toc30856342)

[ABSTRAK v](#_Toc30856343)

[ABSTRACT vi](#_Toc30856344)

[DAFTAR ISI vii](#_Toc30856345)

[DAFTAR TABEL ix](#_Toc30856346)

[DAFTAR GAMBAR x](#_Toc30856347)

[DAFTAR LAMPIRAN xi](#_Toc30856348)

[BAB 1 PENDAHULUAN 1](#_Toc30856349)

[1.1 Latar Belakang 1](#_Toc30856350)

[1.2 Rumusan Masalah 3](#_Toc30856351)

[1.3 Tujuan 3](#_Toc30856352)

[1.4 Manfaat 3](#_Toc30856353)

[1.5 Batasan Masalah 4](#_Toc30856354)

[1.6 Sistematika Pembahasan 4](#_Toc30856355)

[BAB 2 LANDASAN KEPUSTAKAAN 6](#_Toc30856356)

[2.1 Kajian Pustaka 6](#_Toc30856357)

[2.2 Dasar Teori 7](#_Toc30856358)

[2.2.1 *Wireless Sensor Network* (WSN) 7](#_Toc30856359)

[2.2.2 *Long Range* (LoRa) 8](#_Toc30856360)

[2.2.3 ModulLoRaRa-02 SX1278 11](#_Toc30856361)

[2.2.4 Metode *Flooding* 11](#_Toc30856362)

[2.2.5 Arduino Nano 12](#_Toc30856363)

[BAB 3 METODOLOGI 13](#_Toc30856364)

[3.1 Kerangka Penelitian 13](#_Toc30856365)

[3.2 Perancangan Sistem 13](#_Toc30856366)

[3.2.1 Perancangan *Controlled Flooding* 15](#_Toc30856367)

[3.2.2 Perancangan *Sensor Node* 16](#_Toc30856368)

[3.2.3 Perancangan *Relay Node* 17](#_Toc30856369)

[3.2.4 Perancangan *Gateway Node* 18](#_Toc30856370)

[3.3 Metode Evaluasi 20](#_Toc30856371)

[3.3.1 Pengujian Fungsional Sistem 20](#_Toc30856372)

[3.3.2 Pengujian Kinerja Sistem 20](#_Toc30856373)

[BAB 4 IMPLEMENTASI 22](#_Toc30856374)

[4.1 A 22](#_Toc30856375)

[4.2 B 22](#_Toc30856376)

[BAB 5 PENGUJIAN DAN PEMBAHASAN 23](#_Toc30856377)

[5.1 Pengujian 23](#_Toc30856378)

[5.2 Pembahasan 23](#_Toc30856379)

[BAB 6 Penutup 24](#_Toc30856380)

[6.1 Kesimpulan 24](#_Toc30856381)

[6.2 Saran 24](#_Toc30856382)

[DAFTAR REFERENSI 25](#_Toc30856383)

DAFTAR TABEL

[Tabel 2.1 Tabel Kajian Pustaka 6](#_Toc20849163)

[Tabel 2.2 Spesifikasi Arduino Nano versi 3.x 11](#_Toc20849164)

[Tabel 3.1 Tabel Pengujian Fungsional Sistem 20](#_Toc20849164)

DAFTAR GAMBAR

[Gambar 2.1 Arsitektur Sederhana *Wireless Sensor Network* 8](#_Toc20849165)

[Gambar 2.2 Perbandingan Lora , Wi-Fi, BLE, Cellular 9](#_Toc20849166)

[Gambar 2.3 Struktur *physical frame LoRa* 10](#_Toc20849167)

[Gambar 2.4 LoRa Ra-02 11](#_Toc20849168)

[Gambar 2.5 Arduino Nano 12](#_Toc20849168)

[Gambar 3.1 Perancangan Sistem 14](#_Toc20849170)

[Gambar 3.2 *Sequence* Diagram Perancangan Sistem 15](#_Toc20849170)

[Gambar 3.3 Diagram Alir Mekanisme *Controlled Flooding* 16](#_Toc20849170)

[Gambar 3.4 Diagram Alir Perancangan *Sensor Node* 17](#_Toc20849170)

[Gambar 3.5 Diagram Alir Perancangan *Relay Node* 18](#_Toc20849170)

[Gambar 3.6 Diagram Alir Perancangan *Gateway Node* 19](#_Toc20849170)

[Gambar 3.1 Perancangan Sistem 14](#_Toc20849170)

DAFTAR LAMPIRAN

[LAMPIRAN A PERSYARATAN FISIK DAN TATA LETAK 29](#_Toc496081034)

[A.1 Kertas 29](#_Toc496081035)

[A.2 Margin 29](#_Toc496081036)

[A.3 Jenis dan Ukuran Huruf 29](#_Toc496081037)

[A.4 Spasi 29](#_Toc496081038)

[A.5 Kepala Bab dan Subbab 29](#_Toc496081039)

[A.6 Nomor Halaman 30](#_Toc496081040)

[LAMPIRAN B PENGGUNAAN BAHASA 31](#_Toc496081041)

# PENDAHULUAN

## Latar Belakang

*Wireless sensor network* (WSN) merupakan suatu jaringan nirkabel yang terdiri dari kumpulan node sensor yang tersebar di suatu area tertentu (*sensor field*). Tiap node sensor memiliki kemampuan untuk mengumpulkan data dan berkomunikasi dengan node sensor lainnya (Prana, et al., 2019). WSN secara umum digunakan untuk memantau suatu lingkungan dengan dilakukannya *sensing* oleh *node – node* *sensor*. Selain itu, WSN juga dapat digunakan untuk mengontrol suatu lingkungan dengan mengirimkan perintah kepada *node – node* yang tersebar. Contohnya, pada bidang pertanian untuk menjaga kualitas hasil panen dan *home automation* untuk otomatisasi perangkat di rumah (Sohraby, 2007).

Penerapan teknologi WSN membutuhkan minimal terdapat dua *node* yang saling terhubung untuk berkomunikasi atau bertukar data antar *node*. *Node-node* tersebut dapat terdiri dari perangkat yang dapat melakukan *sensing*, lalu meneruskan paket ke *gateway node* yang berfungsi untuk menerima data dari *sensor*, kemudian mengirimkan data ke *network server* tergantung arsitektur WSN yang akan diterapkan. Setiap *node sensor* dalam WSN biasanya dilengkapi dengan *sensor*, mikrokontroler (menyediakan konversi analog ke digital dan kemampuan komputasi dan penyimpanan), perangkat radio *transceiver* (menyediakan kemampuan komunikasi nirkabel) dan sumber energi atau penyimpanan energi (biasanya dalam bentuk baterai elektrokimia) (Flammini & Sisinni, 2014). Sedangkan untuk *gateway node*, umumnya terdri dari mikrokomputer, radio *transceiver*, dan modul komunikasi yang terhubung ke internet untuk mengirim data ke server jaringan. Seluruh *node* pada WSN dapat saling terhubung menggunakan modul komunikasi. Salah satu modul komunikasi yang memiliki konsumsi daya yang rendah dan jarak jangkau yang cukup jauh adalah *Long Range* (LoRa).

*Long Range* (LoRa) adalah sistem telekomunikasi nirkabel yang didesain untuk jarak jauh, berdaya rendah, bitrate rendah, dan tidak berlisensi (Wixted, 2016). Salah satu kelebihan yang dimiliki LoRa, yaitu dapat mengakomodasi jarak mencapai 1 km lebih dengan konfigurasi yang tepat dan juga lingkungan yang mendukung seperti di pedesaan yang tidak padat penduduk dan sedikitnya gangguan sehingga LoRa sangat cocok untuk monitoring daerah pertanian di Indonesia yang dikenal sebagai negara agraris karena luasnya lahan pertanian yang dimiliki. Selain itu, LoRa juga menghasilkan sinyal yang tahan terhadap interferensi (Susanto, 2018). Modul LoRa terdiri dari LoRa *transceiver* dan LoRa *Gateway*. LoRa *transceiver* digunakan pada *sensor node* yang akan mengirim data ke LoRa *gateway*. Sedangkan LoRa *gateway* berfungsi untuk menerima data yang dikirim dari *sensor node* dan juga mengirimkan kan data ke server jaringan.

Meskipun memiliki banyak kelebihan, terdapat beberapa potensi kelemahan pada LoRa. Salah satunya, yaitu keterbatasan LoRa yang penerapan komunikasinya bersifat *single-hop* sepertiyang dijelaskan oleh Augustine, et al. (2016) bahwa tipikal jaringan LoRa menggunakan topologi *star* yang mana perangkat akhir berkomunikasi dengan *gateway* menggunakan LoRa kemudian oleh *gateway* data dari perangkat akhir diteruskan ke server jaringan menggunakan koneksi berbasis IP, karakteristik *single-hop* ini menimbulkan masalah ketika diterapkan pada daerah yang padat penduduk seperti di daerah perkotaan, dengan *gateway node* yang berada di pusat kota namun tetap harus menjangkau perangkat terjauh yang mungkin berada di daerah pinggiran kota. Selain itu, permasalahan juga timbul ketika *end-device* berada di *dark spot* atau daerah yang sulit dijangkau oleh jaringan. Sehingga memungkinkan *gateway node* tidak menjangkau perangkat *end node* terjauh. Selain itu, terdapat tantangan lain yaitu LoRa tidak memiliki lapisan jaringan sehingga LoRa tidak dapat membedakan *node* tujuan dengan *node – node* yang lain.

Bruno mengatakan dalam penelitiannya bahwa kemampuan jarak jauh LoRa mungkin tidak memadai pada semua kasus. Jangkauan ini mungkin juga tidak mencukupi di daerah dengan koneksi yang buruk sehingga mengganggu koneksi pada jaringan LoRa. Oleh karena itu, untuk mengatasi permasalahan tersebut Bruno menguji kelayakan jika memasukkan *forwarder node* di antara *gateway node* dan *end-device* untuk meningkatkan jangkauan dan kualitas dari komunikasi LoRa (Velde, 2017). Penelitian lain oleh Daniel Lundell, disebutkan bahwa untuk mengatasi keterbatasan LoRa yang bersifat *single-hop* dibutuhkan protokol *routing* untuk melakukan komunikasi *multi-hop* agar *end node* yang tidak dapat menjangkau *gateway node* secara langsung dapat melalui *node* perantara atau *relay node* sehingga jangkauan dari jaringan LoRa tersebut dapat lebih luas. Daniel menyajikan protokol baru untuk menyediakan komunikasi dan perutean antara *gateway* LoRa untuk menyediakan cakupan di daerah terpencil. Protokol yang disajikan didasarkan pada *Hybrid Wireless Mesh Protocol* (HWMP) dan *Ad-hoc On-Demand Distance Vector Routing* (AODV) untuk menyediakan transmisi *multi*-*hop* pada jaringan LoRa (Lundell, et al., 2018).

Dari permasalahan yang dijelaskan di paragraf sebelumnya dan juga penelitian yang telah dibahas pada paragraf sebelumnya, penelitian ini akan menerapkan *multi-hop* pada jaringan LoRa menggunakan metode *controlled flooding*. Pemilihan metode flooding yaitu karena sangat sederhana yaitu dimana setiap *node* hanya perlu menginformasikan ke *node* tetangganya dan tidak perlu menyimpan tabel perutean dari topologi sebelumnya. Dalam hal ini persisten data melalui jaringan dijamin. Paket-paket akan mencapai tujuannya terlepas dari rute. Namun, beberapa peringatan perlu diambil untuk mencegah duplikasi paket atau perulangan tak terbatas dan juga harus menghindari tabrakan (Munoz, et al., 2015). Oleh karena itu, metode *controlled flooding* atau *flooding* terkontrol diusulkan dalam penelitian ini. Dengan harapan metode *controlled flooding* ini dapat mengatasi duplikasi paket atau pengiriman paket berulang tanpa batas dan menghindari adanya tabrakan ketika transmisi data.

Dalam penelitian ini, modul LoRa akan dihubungkan dengan mikrokontroler yaitu Arduino Nano yang diletakkan pada setiap *node*. Mikrokontroler Arduino Nano digunakan untuk menyimpan konfigurasi LoRa dan kode program pengiriman paket pada setiap *node*, untuk menyematkan konfigurasi dan kode program pada Arduino Nano menggunakan Arduino IDE. Pada sisi *relay node* atau *node* perantara akan disematkan kode program menggunakan Arduino IDE untuk meneruskan paket yang dikirim dari *node* sumber menuju *node* tujuan. Pada *gateway node* akan disematkan kode program pengiriman paket permintaan yang di dalamnya terdapat informasi *node* sumber dan *node* tujuan. Sedangkan pada *sensor node* akan disematkan kode program pengiriman paket balasan menuju *gateway node* yang meminta data *sensor*. Pengiriman paket pada penelitian ini menggunakan metode *controlled* *flooding* yang mengirimkan paket ke semua *node* tetangga hingga paket sampai pada *node* tujuan.

## Identifikasi Masalah

## Rumusan Masalah

Berdasarkan uraian latar belakang yang telah dipaparkan, maka rumusan masalah yang dapat diperoleh adalah sebagai berikut:

1. Bagaimana cara untuk menerapkan komunikasi *multi-hop* LoRa pada *wireless sensor network* (WSN)?
2. Bagaimana cara konfigurasi *relay node* sehingga dapat meneruskan paket dari *node* sumber menuju *node* tujuan?
3. Bagaimana cara untuk menerapkan metode *flooding* untuk pengiriman paket pada perangkat berbasis LoRa?
4. Bagaimana cara mencegah duplikasi paket yang diterima dan pengiriman paket berulang oleh setiap *node*?

## Tujuan

Keterbatasan LoRa yang memiliki karakter *single-hop* dalam penerapan komunikasinya menimbulkan permasalahan ketika *end-device* berada di *dark spot* atau daerah yang sulit dijangkau oleh jaringan. Mengacu pada permasalahan tersebut, maka penelitian ini bertujuan untuk menerapkan komunikasi *multi-hop* pada jaringan LoRa untuk memperluas jangkauan dengan metode *controlled flooding*.

## Manfaat

Manfaat yang diharapkan dari skripsi ini adalah sebagai berikut:

Penelitian ini diharapkan dapat menjadi solusi untuk pengaplikasian sistem pemantauan dan kontrol jarak jauh dengan biaya dan energi yang rendah.

Penelitian ini juga dapat bermanfaat untuk pengembangan LoRa maupun segala penelitian yang berkaitan dengan penelitian ini di masa mendatang.

Penelitian ini dapat menjadi wadah untuk mengaplikasikan atau menerapkan ilmu yang telah didapatkan dan juga menambah wawasan tentang berbagai hal yang terdapat di penelitian ini.

## Batasan Masalah

Dalam penyusunan skripsi yang baik diberikan beberapa batasan masalah yang akan menjadi batasan dalam melakukan penelitian dan juga dapat memudahkan penyusunan laporan yang sistematis sehingga mudah dipahami. Batasan batasan yang digunakan dalam skripsi ini antara lain :

Modul komunikasi yang digunakan adalah modul LoRa bertipe SX1278.

Frekuensi yang digunakan pada modul LoRa adalah 433MHz.

Mikrokontroler yang digunakan pada *node* adalah Arduino Nano.

Faktor keamanan pada sistem yang diimplementasikan tidak menjadi fokus pada penelitian ini.

Fokus penlitian ini yaitu komunikasi *multihop* pada LoRa.

Pengiriman paket menggunakan metode *controlled* *flooding*.

Data yang dikirim oleh *sensor node* merupakan data yang diperoleh dari *sensor* DHT11.

## Sistematika Pembahasan

Sistematika pembahasan berfungsi sebagai gambaran dan uraian pada penelitian ini, secara garis besar diantaranya meliputi beberapa bab, sebagai berikut:

**BAB 1 : PENDAHULUAN**

Pada bab ini memberikan penjelasan mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, serta sistematika pembahasan penelitian yang berfungsi sebagai acuan untuk melakukan pembahasan terhadap penelitian.

**BAB 2 : LANDASAN KEPUSTAKAAN**

Bab ini memuat dan menguraikan kajian dan teori yang mendasari penelitian dan perancangan penelitian ini. Hal ini diperlukan untuk mempermudah dalam melakukan penelitian dan juga sebagai pedoman penelitian yang resmi karena sudah di terbitkan.

**BAB 3 : METODOLOGI PENELITIAN**

Bab ini akan membahas tentang sistematika dalam melakukan penelitian. Langkah-langkah seperti perancangan, implementasi, pengujian dibahas secara umum.

**BAB 4 : IMPLEMENTASI**

Bab ini membahas tentang implementasi dari sistem berdasarkan kebutuhan dan perancangan yang telah di definisikan. Implementasi akan sesuai dengan perancangan yang telah dibuat sebelumnya secara rinci beserta langkah-langkah pengerjaan penelitian. Selain itu juga memberikan pseudocode utama yang digunakan dalam penelitian ini.

**BAB 5 : PENGUJIAN DAN PEMBAHASAN**

Bab ini menjelaskan tentang bagaimana saat sistem telah diimplementasikan dan dijalankan pada lingkungan uji yang telah ditentukan. Hasil-hasil yang diperoleh akan dijabarkan pada bab ini. Selanjutnya akan dilakukan pengujian dan hasil pengujian didapatkan dari pengujian yang telah dilakukan sesuai dengan skenario pengujian yang telah ditentukan.

**BAB 6 : PENUTUP**

Bab ini memuat kesimpulan dari seluruh proses penelitian, mulai dari rumusan masalah hingga tahap pengujian. Selain itu, juga memberikan saran untuk pengembangan lebih lanjut dengan penelitian yang terkait.

# LANDASAN KEPUSTAKAAN

## Kajian Pustaka

Kajian pustaka ini menunjukkan beberapa penelitian yang telah dilakukan sebelumnya yang memiliki keterkaitan dengan penelitian ini. Berikut ditunjukkan pada Tabel 2.1 .

Tabel 2.1 Tabel Kajian Pustaka

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No | Judul | Tahun | Penulis | Penelitian | Penelitian penulis |
| 1 | *Multi-hop LoraWAN: including a forwarding node* | 2017 | Bruno Van de Velde | Menganalisis dampak dari penerapan *forwarder node* di antara perangkat akhir dan *gateway*. | Mengoptimalkan komunikasi *multi-hop* pda LoRa dengan menggunakan metode *controlled flooding* . |
| 2 | *A Routing Protocol for LoRa Mesh Networks* | 2018 | Daniel Lundell, Anders Hedberg, Christian Nyberg, dan Emma Fitzgerald | Menerapkan komunikasi *multi-hop* dengan menggunakan protokol *routing* yang didasarkan protokol *routing* AODV dan HWMP untuk komunikasi antar perangkat *node*. | Menerapkan komunikasi *multi-hop* pada jaringan LoRa menggunakan metode *controlled* *flooding.* |

Pada penelitian terdahulu yang berjudul *Multi-hop LoraWAN: including a forwarding node* menganalisis tentang dampak dari penerapan *forwarder node* di antara perangkat akhir atau *end-device* dengan *gateway*. Dalam penelitiannya, Bruno menerapkan tiga perangkat masing – masing sebagai *gateway*, *forwarder*, dan *end-device* atau perangkat akhir. Penempatan perangkat juga dilakukan pada daerah dengan banyak hambatan untuk menguji jaringan dengan koneksi yang buruk. Dari penelitian tersebut diperoleh hasil bahwa penambahan *forwarder node* pada jaringan LoRa dapat dilakukan dan memberikan beberapa keuntungan. Keuntungannya yaitu rentang komunikasi dapat ditingkatkan dan komunikasi dapat dimungkinkan di mana perangkat akhir dan *gateway* tidak dapat saling menjangkau. Selain itu, menerapkan *forwarder* juga menyebabkan konsumsi energi tiga kali lebih rendah dibandingkan tanpa menggunakan *forwarder*. Namun terjadi peningkatan *­packet loss* ketika penerapannya menggunakan *adaptive ­data rates*.

Penelitian lain yang berjudul *A Routing Protocol for LoRa Mesh Networks* menerapkan komunikasi *mesh* pada jaringan LoRa untuk mengatasi keterbatasan jaringan LoRa yang memiliki karakteristik *single-hop* karena menurut Daniel dkk, karakteristik *single-hop* ini menimbulkan masalah ketika diterapkan pada daerah yang padat penduduk seperti di daerah perkotaan, dengan *gateway node* yang berada di pusat kota namun tetap harus menjangkau perangkat terjauh yang mungkin berada di daerah pinggiran kota. Dalam implementasinya, Daniel Lundel dkk menyajikan protokol baru yang berdasarkan pada *Hybrid Wireless Mesh Protocol* (HWMP) dan *Ad-hoc On-Demand Distance Vector Routing* (AODV) untuk menyediakan transmisi *multi*-*hop* pada jaringan LoRa. Pengujian yang dilakukan terhadap protokol tersebut yaitu menggunakan topologi linear dengan jumlah hop yang bervariasi. Penelitian yang dilakukan memberikan hasil bahwa protokol yang disajikan pada penelitian tersebut dapat menyediakan transmisi *multi-hop* pada jaringan LoRa dan dapat menunjukkan efektifitas *routing multi-hop* untuk jaringan LoRa. Namun, protokol yang disajikan masih belum memiliki fitur yang lengkap. Pada penelitian tersebut penanganan kegagalan rute seperti kesalahan rute masih belum disajikan.

Berdasarkan beberapa penelitian diatas yang berkaitan dengan rencana penelitian ini, penulis mengimplementasikan komunikasi *multi-hop* dengan menerapkan metode *controlled flooding* pada WSN yang berbasis LoRa dengan tujuan yang sama pada penelitian sebelumnya yaitu penerapan komunikasi *multi-hop* untuk mengatasi keterbatasan LoRa yang bersifat *single-hop*. Perbedaan penelitian ini dengan penelitian sebelumnya yaitu pada metode komunikasi yang digunakan. Rencana penelitian ini akan menggunakan metode *controlled flooding*.

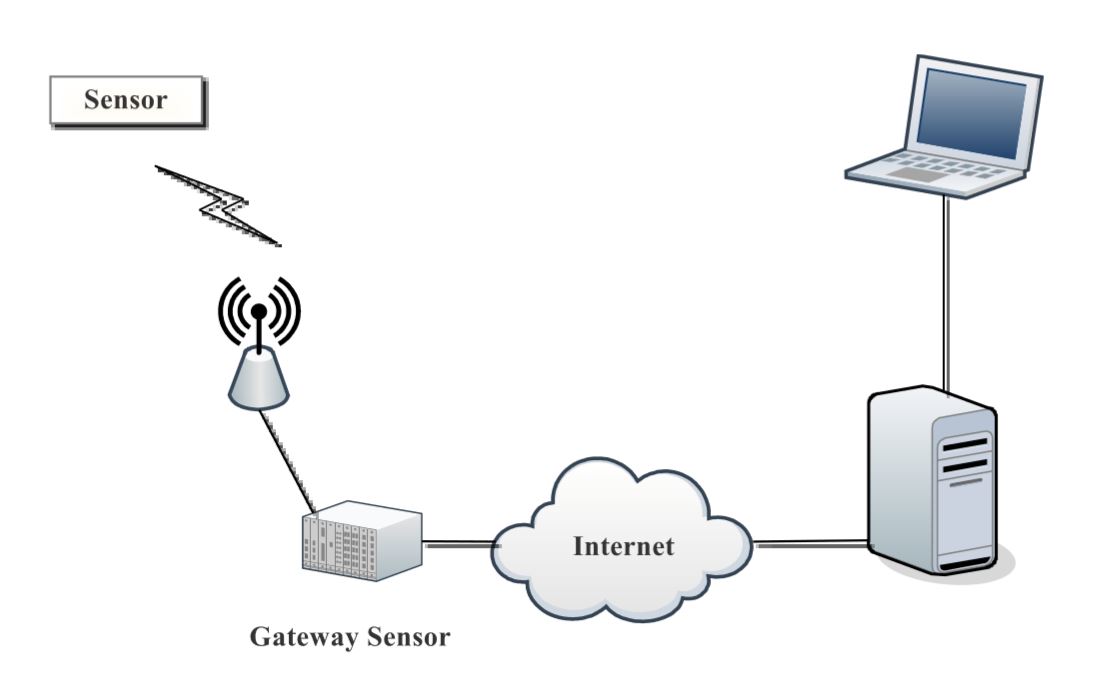
## Dasar Teori

Dalam melakukan penelitian ini, diperlukan beberapa referensi dan dasar teori yang mendukung penelitian. Hal ini guna mempermudah pada proses selanjutnya seperti perancangan dan implementasi. Berikut adalah beberapa teori yang menjadi dasar penelitian.

### *Wireless Sensor Network* (WSN)

*Wireless sensor network* (jaringan sensor nirkabel) adalah suatu jaringan nirkabel yang terdiri dari kumpulan *node* sensor yang tersebar di suatu area tertentu (*sensor field*). Tiap node sensor memiliki kemampuan untuk mengumpulkan data dan berkomunikasi dengan node sensor lainnya (Prana, 2018). Fungsi WSN sendiri untuk memonitoring lingkungan misalnya suhu dan kelembaban dan juga dapat digunakan untuk mengontrol suatu mesin. WSN memiliki keunggulan lebih praktis dan lebih murah dibandingkan menggunakan teknologi kabel (*wired).* WSN secara umum digunakan untuk memantau dan juga mengontrol suatu lingkungan dengan dilakukannya *sensing* oleh *node – node* *sensor*. Contohnya, Pada bidang pertanian untuk menjaga kualitas hasil panen dan Home automation untuk otomatisasi perangkat di rumah (Sohraby, et al., 2007).

Pada implementasi WSN sederhana terdapat *sensor node* yang telah disematkan *sensor* untuk mengambil data yang diperlukan dan mengirimkan data tersebut ke suatu *node* yang merupakan *gateway node* kemudian *gateway node* tersebut mengirimkan ke *server* jaringan melalui internet seperti pada Gambar 2.1. Pada *sensor node* terdapat mikrokontroler dan modul komunikasi nirkabel yang berfungsi untuk mengirimkan data dari *sensor* menuju *gateway node*. Kemudian terdapat juga *relay node* yang berfungsi untuk meneruskan paket data dari *sensor node* menuju *gateway node*. Penggunaan *relay node* terdapat pada WSN dengan topologi tertentu seperti topologi *tree* dan topologi *mesh*.



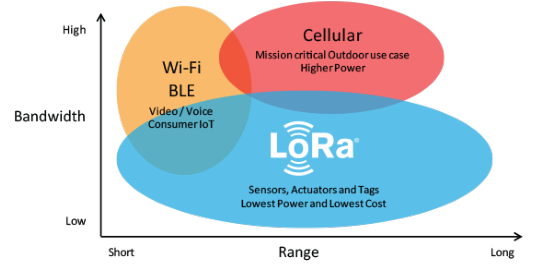
Gambar 2.1 Arsitektur Sederhana *Wireless Sensor Network*

Sumber: (Wazir, et al., 2016)

Pada penelitian ini, peneliti menerapkan komunikasi *multihop* pada WSN menggunakan lima *node* dengan satu *sender*, satu *receiver* dan tiga *relay*. Fokus penelitian ini yaitu bagian komunikasi antara *sensor node* (*sender*) dan *gateway node* (*receiver*) yang komunikasinya menggunakan salah satu modul komunikasi *Low Power Wide Area Network* (LPWAN) yaitu LoRa.

### *Long Range* (LoRa)

LoRa adalah sistem telekomunikasi nirkabel yang didesain untuk komunikasi jarak jauh dan berdaya rendah. Teknologi lora yang merupakan merek dagang dari semtech, adalah salah satu dari sejumlah pita ISM baru dengan teknologi berdaya rendah. (Wixted, et al., 2016). Pita frekuensi pada LoRa bermacam-macam sesuai daerahnya, jika di Asia frekuensi yang digunakan yaitu 433 MHz, di Eropa nilai frekuensi yang digunakan yaitu 868 MHz, sedangkan di Amerika Utara frekuensi yang digunakan yaitu 915 MHz. LoRa bekerja pada physical layer, dimana LoRa menggunakan teknik modulasi yang merupakan teknologi hak milik Semtech yang tidak sepenuhnya terbuka. LoRa menggunakan modulasi radio Chirp Spread Spectrum (CSS) yang menggunakan chirp (*Compressed High Intensity Radar Pulse*) dengan variasi frekuensi linear dari waktu ke waktu untuk menyandikan informasi (Augustin, et al., 2016). Semtech juga menunjukkan perbandingan LoRa dengan Wi-Fi/BLE dan jaringan seluler berdasarkan kapasitas *bandwidth* dan jarak jangkaunya pada Gambar 2.2 berikut.



Gambar 2.2 Perbandingan Lora , Wi-Fi, BLE, Cellular

Sumber: (Susanto, et al., 2019)

Gambar 2.3 diatas merupakan perbandingan dari teknologi komunikasi *WiFi*, *Bluetooth Low Energy* (BLE), *Cellular*, dan *Long range* (LoRa) berdasarkan ukuran *bandwith* dan jarak jangkauan. Gambar tersebut menunjukkan bahwa LoRa dapat menjangkau jarak yang jauuh melebihi *cellular* walaupun *bandwith* nya tidak sebesar *cellular*.

LoRa adalah lapisan fisik atau modulasi nirkabel yang digunakan untuk membuat komunikasi jarak jauh. Sistem nirkabel lama banyak yang menggunakan modulasi Frequency Shift Keying (FSK) sebagai lapisan fisik karena merupakan modulasi yang sangat efisien untuk menghasilkan daya yang rendah. LoRa didasarkan pada modulasi Chirp Spread Spectrum (CSS), yang mempertahankan karakteristik daya rendah yang sama dengan modulasi FSK tetapi secara signifikan meningkatkan jangkauan komunikasi (Ruano, 2016). Modulasi LoRa menggunakan bentuk modulasi *spread spectrum* dikombinasikan dengan *cyclic error correction coding*. Kombinasi keduanya berpengaruh pada peningkatan *link budget* dan kekebalan terhadap interferensi. Modulasi LoRa dapat di optimasi dengan tiga parameter sebagai berikut:

*Spreading Factor* (SF)

*Spreading factor* merupakan durasi dari chirp yang merupakan meningkatnya sinyal frekuensi *up-chirp* atau *down-chirp* terhadap waktu. *Spreading factor* juga mendefinisikan banyaknya bit yang dapat dikodekan pada simbol. *Spreading* factor pada LoRa beroperasi di rentang 7 sampai 12. *Spreading factor* 7 (SF7) menunjukkan waktu yang diperlukan untuk mengirim satu chirp paling kecil atau waktu tersingkat di udara sedangkan *spreading factor* 12 (SF12) menunjukkan waktu yang diperlukan untuk mengirim satu chirp paling besar atau waktu terlama di udara, sehingga *Spreading factor* juga mempengaruhi *bitrate* dari pengiriman paket.

*Coding Rate* (CR)

*Coding rate* digunakan untuk menangani *Packet Error Rate* (PER) akibat adanya interferensi. *Coding rate* diformulasikan sebagai CR = 4/(4+n) yang mana n adalah {1,2,3,4}. Semakin kecil angka *coding rate* maka paket yang dikirim akan lebih toleran terhadap interferensi tetapi rasio *overhead* semakin tinggi.

*Bandwidth* (BW)

*Bandwidth* merupakan parameter yang sangat penting pada penentuan *chip rate*. *Chip* merupakan satuan elemen yang berbentuk *pulse* dalam konteks *chirp spread spectrum* (CSS). Peningkatan sinyal *bandwidth*  memungkinkan penggunaan kecepatan data efektif yang lebih tinggi sehingga mengurangi waktu transmisi dengan mengorbankan sensitivitas menjadi lebih rendah.

Pengaturan ketiga parameter diatas memberikan pengaruh terhadap bitrate dan jarak jangkauan dari LoRa. Peningkatan *bandwidth* akan menurunkan sensitivitas penerima tapi menghasilkan bitrate yang tinggi. Peningkatan *spreading factor* (SF) akan meningkatkan sensitivitas penerima tetapi akan bitrate akan menurun. Penurunan angka *coding rate* akan memberikan toleransi terhadap pengiriman paket yang tinggi namun meningkatkan rasio *overhead*. Hal ini menunjukkan bahwa perangkat LoRa harus di konfigurasi secara tepat untuk memberikan hasil yang sesuai.

Struktur *physical frame* pada LoRa ditentukan oleh Semtech yang diimplementasikan pada *transmitter* dan *receiver* milik Semtech. Struktur *physical frame* pada LoRa ditunjukkan pada Gambar 2.3.



Gambar 2.3 Struktur *physical frame LoRa*

Sumber: (Augustin, et al., 2016)

Seperti yang ditunjukkan pada Gambar 2.3, *physical frame* LoRa dimulai dengan *preamble* yang digunakan untuk menyinkronkan *receiver* dengan aliran data yang masuk dan membedakan jaringan LoRa yang menggunakan frekuensi yang sama. Kemudian terdapat *header* yang mana bersifat opsional. *Header* iniditransmisikan dengan *code rate* 4/8. Kegunaan *header* ini yaitu untuk menunjukkan informasi ukuran *payload* (dalam byte), *code rate* yang digunakan untuk akhir transmisi, dan keberadaan adanya CRC 16 bit opsional untuk *payload*.

### ModulLoRaRa-02 SX1278

Dalam menerapkan komunikasi menggunakan teknologi LoRa, tentunya menggunakan modul LoRa yang banyak dijual dipasaran. Modul LoRa sendiri mempunyai banyak tipe yang dapat digunakan yaitu SX1276/77/78/79 berdasarkan. LoRa memilki dua tipe yaitu LoRa *tranceiver* yang difungsikan pada bagian *node sensor* dan LoRa *gateway* yang difungsikan sebagai *gateway* atau perantara *sensor* dengan server jaringan.

Penelitian ini menggunakan LoRa Ra-02 SX1278 dengan frekuensi 433MHz. LoRa SX1278 merupakan LoRa *transceiver* yang dapat dihubungkan dengan mikrokontroler ataupun mikrokomputer.Modul LoRa ini akan dihubungkan pada arduino nano untuk menerapkan konfigurasi LoRa untuk memproses paket pada setiap *node* agar antar setiap *node* dapat saling berkirim paket. Berikut Gambar 2.4 adalah gambar dari modul LoRa Ra-02 SX1278.



Gambar 2.4 LoRa Ra-02

Sumber: [www.modtronix.com](http://www.modtronix.com), diakses 04-02-2019

Modul LoRa Ra-02 merupakan modul LoRa dengan frekuensi 433 MHz menggunakan *chip* dari SX1278 yang didesain oleh AI-THINKER. LoRa SX1278 memiliki keunggulan yaitu sensitivitas yang tinggi mencapai -148 dBm dengan *power output* +20 dBm, transmisi jarak jauh dan keandalan yang tinggi.

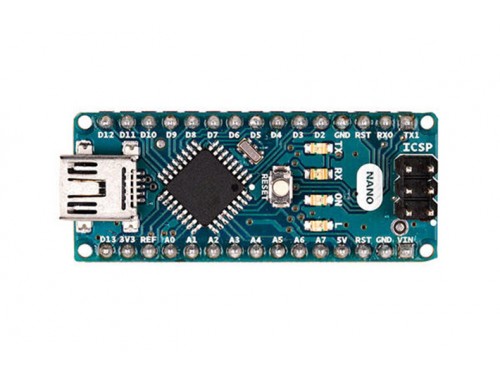
### Metode *Flooding*

Algoritme *flooding* merupakan salah satu algoritma komunikasi paling dasar untuk dengan cepat menyebarkan informasi di semua node dalam WSN. Pada algoritme ini, setiap *node* mencoba meneruskan setiap pesan ke semua tetangganya kecuali *node* sumbernya dan pesan hanya dibagikan sekali oleh setiap *node*. Penggunaan metode *flooding* tidak memerlukan pengetahuan mengenai struktur jaringan, sehingga menghilangkan tabel perutean (Bondu, et al., 2017). Keuntungan menggunakan algoritme *flooding* yaitu memliki kesederhanaan dalam kode perutean yang tidak memerlukan tabel perutean, pengiriman paket melewati jalur terpendek, terhindar dari *Single Point of Failure* yang akan melewati jalur lain ketika terdapat jalur yang bermasalah.

Dalam penelitian ini algoritme *flooding* akan diterapkan sebagai metode untuk pengiriman paket oleh modul LoRa. Paket yang dikirimkan merupakan paket *request* yang dikirimkan oleh *gateway node* kepada *sensor node* dan juga paket *reply* yang dikirimkan oleh *sensor node* kepada *gateway node*. Namun, metode *flooding* memiliki kelemahan yaitu adanya duplikasi data pada *node* tujuan sehingga akan dilakukan mekanisme untuk mencegah duplikasi data dengan men*drop* paket yang di terima dengan id yang sama.

### Arduino Nano

Arduino Nano adalah salah satu mikrokontroler yang mirip dengan Aduino Uno namun memiliki ukuran yang lebih kecil dan tetap mendukung penggunaan *breadboard*. Arduino Nano menggunakan mikrokontroler berbasis ATmega328P untuk Arduino Nano versi 3.x dan ATmega168P untuk Arduino Nano versi 2.x. Perbedaannya yaitu Nano tidak memiliki colokan listrik DC, dan bekerja menggunakan kabel USB mini-B. Arduino menggunakan Arduino IDE dalam melakukan pemrograman sistem yang akan dibuat (Arduino, 2018).



Gambar 2.5 Arduino Nano

Sumber : (Arduino, 2018)

Penelitian ini akan menggunakan mikrokontroler Arduino Nano versi 3.x yang ditunjukkan pada Gambar 2.5. Arduino Nano versi 3.x memiliki spesifikasi yang dijelaskan pada Tabel 2.2.

Tabel 2.2 Spesifikasi Arduino Nano versi 3.x

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *Processor* | *Voltage* | *CPU Speed* | *Analog Input / Out* | *Digital IO / PWM* | *EEPROM [kB]* | *SRAM [kB]* | *Flash [kB]* | *USB* |
| ATmega328P | 5 V / 7-9 V | 16MHz | 8/0 | 14/6 | 1 | 2 | 32 | Mini |

Pada Arduino Nano ini akan diterapkan sistem penerusan paket pada *relay node* agar dapat meneruskan paket dari *source node*  menuju *destination node* menggunakan komunikasi LoRa. Pengguna nantinya hanya perlu mengunggah kode program yang berfungsi meneruskan paket dengan menggunakan Arduino IDE pada perangkat Arduino Nano yang berperan sebagai *relay node*. Selain itu, mikrokontroler ini juga digunakan sebagai *sensor node* dan *gateway node*.

# METODOLOGI

Pada bab ini dijelaskan langkah-langkah atau metode yang dilakukan dalam perancangan, implementasi dan pengujian dari implementasi komunikasi *multi-hop* menggunakan metode *controlled* *flooding* pada *wireless sensor network* berbasis LoRa. Setelah pengujian akan ditarik kesimpulan sebagai catatan atas proses implementasi sistem tersebut.

## Kerangka Penelitian

Kerangka penelitian ini merupakan langkah – langkah yang dilakukan dalam penelitian ini bertujuan sebagai acuan dalam melakukan penelitian ini. Kerangka yang digunakan yaitu yang berkaitan dengan penelitian yang akan dilakukan. Kerangka penelitian yang diperlukan untuk menunjang penelitian implementasi komunikasi *multi-hop* menggunakan metode *controlled flooding* pada *wireless sensor network* berbasis LoRa yaitu sebagai berikut:

Studi literatur

Studi literatur ini digunakan sebagai referensi dan juga sebagai acuan dalam melakukan penelitian ini. Literatur yang digunakan yaitu yang berkaitan dengan penelitian yang akan dilakukan seperti penjelasan tentang *wireless sensor network*, LoRa, perangkat yang digunakan untuk menunjang penelitian, dll.

Analisis kebutuhan lingkungan uji

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan untuk mendukung implementasi komunikasi *multi-hop* menggunakan metode *controlled flooding* pada *wireless sensor network* berbasis LoRa. Kebutuhan dalam hal ini mencakup kebutuhan fungsional sistem, kebutuhan perangkat lunak dan kebutuhan perangkat keras.

Perancangan lingkungan uji

Perancangan lingkungan uji merupakan perancangan sistem yang bertujuan untuk memenuhi kebutuhan fungsional yang sudah di definisikan pada tahap analisis kebutuhan. Dilakukan perancangan alur kerja sistem, perancangan *node*, dan perancangan skenario pengujian. Perancangan ini guna mempermudah dalam pelaksanaan implementasi.

Implementasi sistem

Implementasi merupakan tahapan yang dilakukan untuk membangun suatu sistem. Implementasi dilakukan berdasarkan perancangan sistem yang telah dibuat pada tahap perancangan. Implementasi yang dilakukan meliputi implementasi lingkungan uji dan implementasi metode *controlled flooding*.

Metode evaluasi

Metode evaluasi pada penelitian ini menjelaskan tentang parameter dan scenario uji yang digunakan sebagai dasar untuk melakukan pengujian dan pembahasan. Pengujian yang dilakukan pada penelitian ini yaitu pengujian komunikasi *multi-hop* pada LoRa, pengujian pencegahan infinite loop, dan pengujian kinerja sistem.

Penarikan kesimpulan

Tahap ini merupakan tahap penarikan kesimpulan untuk mengetahui apakah penelitian ini berhasil atau tidak berdasarkan analisis dan pembahasan mengenai pengujian fungsional dan pengujian kinerja dari sistem komunikasi *multi-hop* menggunakan metode *controlled flooding* pada jaringan WSN berbasis LoRa terhadap tujuan penelitian. Selain mengambil kesimpulan dari hasil pengujian, peneliti juga memberikan saran untuk penelitian lanjutan.

## Kebutuhan Lingkungan Uji

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan untuk mendukung implementasi komunikasi *multi-hop* menggunakan metode *controlled flooding* pada *wireless sensor network* berbasis LoRa. Kebutuhan dalam hal ini mencakup kebutuhan fungsional sistem, kebutuhan perangkat lunak dan kebutuhan perangkat keras.

Kebutuhan perangkat keras dijelaskan sebagai berikut:

Modul LoRa Ra-02 SX1278 yang berfungsi sebagai modul komunikasi pada penelitian ini.

Arduino Nano yang merupakan mikrokontroler untuk menjalankan program dan integrasi dengan modul komunikasi LoRa.

Laptop untuk melakukan pengujian pengiriman data dan menampilkan hasil data *sensor.*

Kemudian kebutuhan perangkat lunak akan dijelaskan sebagai berikut:

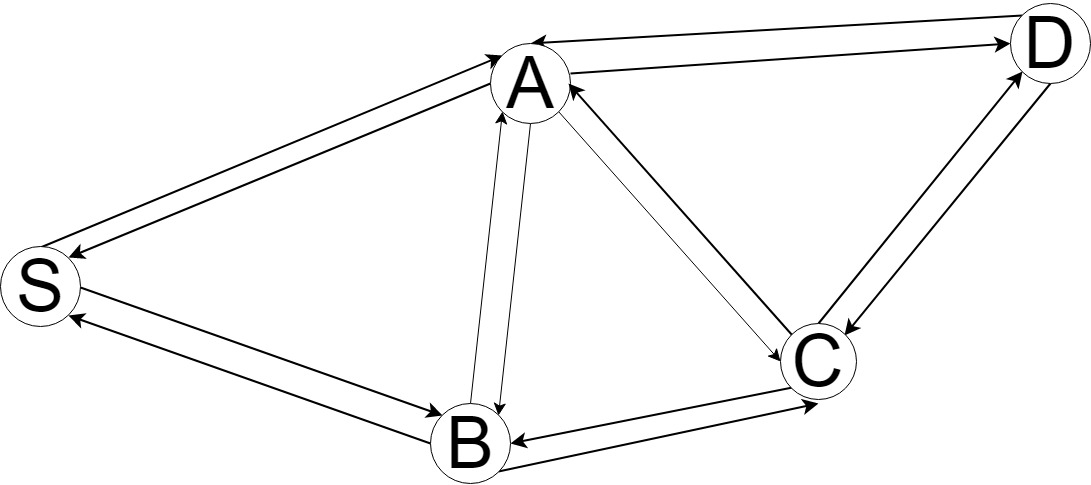
* + - 1. Arduino IDE yang digunakan untuk melakukan pemrograman pada perangkat Arduino, mengupload kode program ke perangkat Arduino dan juga menampilkan hasil cetak dari program yang dijalankan.
      2. Library Arduino LoRa pada Arduino IDE yang digunakan pada kode program bertujuan untuk mempermudah dalam melakukan pemrograman yang berhubungan dengan perangkat LoRa.

Kemudian yang terakhir adalah kebutuhan fungsional pada penelitian ini yang akan dijelasakan sebagai berikut:

1. Sistem dapat melakukan komunikasi *multi-hop* menggunakan modul komunikasi LoRa.
2. Sistem dapat mencegah terjadinya *infinite loop transmission* atau pengiriman data berulang – ulang.
3. Sistem dapat melakukan pengiriman ulang paket yang hilang atau tidak mendapat *reply* dalam selang waktu yang ditentukan.
4. Sistem mampu mengurangi terjadinya *collision* pada saat transmisi data.

## Perancangan Lingkungan Uji

Dalam tahap perancangan dilakukan perancangan sistem yang mampu memenuhi semua kebutuhan yang sudah di definisikan pada tahap analisis kebutuhan. Dilakukan perancangan alur kerja sistem, perancangan *node*, dan perancangan skenario pengujian. Perancangan ini guna mempermudah dalam pelaksanaan implementasi. Perancangan sistem yang akan dibangun dtunjukkan oleh gambar pada Gambar 3.1.

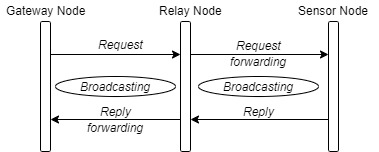


Gambar 3.1 Perancangan Lingkungan Uji

Berdasarkan Gambar 3.1, sistem terdiri dari tiga jenis komponen yaitu *sensor node* / *end node*, *node* perantara / *relay node*, dan *gateway node*. *Gateway node* yang pada gambar ditunjukkan oleh *node* S akan mengirim paket data menuju *sensor node* yang ditunjukkan oleh *node* D melalui *node* perantara yang ditunjukkan oleh *node* A, B dan C dikarenakan *sensor node* dan *gateway node* yang tidak dapat saling menjangkau secara langsung. *Node* perantara inilah yang merupakan kunci dari terjadi komunikasi *multi-hop* pada LoRa karena *node*  perantara bekerja dengan meneruskan paket data dari *end node* menuju *gateway node*.

Dalam penerapan sistem tersebut komunikasi pada jaringan LoRa tersebut menggunakan metode sederhana yaitu *flooding*. Namun, penerapan *flooding* biasa masih akan menyebabkan beberapa permasalahan seperti duplikasi pengiriman paket atau pengiriman paket yang berulang tanpa akhir dan juga menyebabkan terjadinya *collision* ketika terjadi transmisi data dalam waktu yang sama. Oleh sebab itu, pada penelitian ini akan menggunakan metode *controlled flooding* atau *flooding* terkontrol yang dibuat dengan memodifikasi algoritme *flooding* agar dapat menghindari beberapa permasalahan yang telah dijelaskan sebelumnya.

Pada perancangan sistem ini menjelaskan proses kerja sistem secara keseluruhan dari awal pengiriman paket *request* oleh *gateway node* sampai menerima paket balasan yang berisi data *sensor* yang dikirim oleh *sensor node* kemudian hasilnya ditampilkan pada *serial monitor*. Perancangan sistem ini bertujuan untuk mempermudah dalam pelaksanaan implementasi dan pengujian agar sesuai dengan tujuan penelitian yang dilakukan. Diagram alir perancangan sistem untuk penyelesaian permasalahan dalam sistem komunikasi *Multi-Hop* pada *Wireless Sensor Network* berbasis modul komunikasi *LoRa* menggunakan metode *Controlled Flooding* ditunjukkan pada Gambar 3.2.

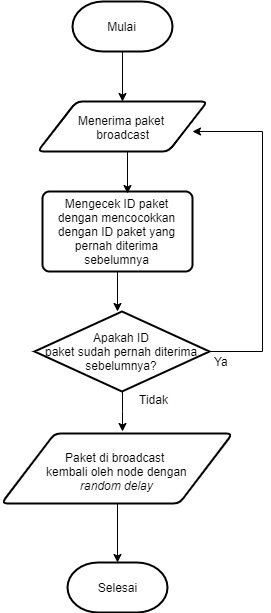


**Gambar 3.2 *Sequence* Diagram Perancangan Sistem**

Pada Gambar 3.2 dijelaskan bagaimana keseluruhan alur sistem yang diawali dengan *gateway* mengirim *broadcast* paket *request*. Selanjutnya, *relay node* menerima paket *broadcast* kemudian mengecek ID paket untuk mencegah duplikasi. Kemudian, Node relay mengirim broadcast paket dari gateway menuju node sensor dengan mencocokkan ID tujuan dengan ID node sensor. Node sensor akan mengirimkan paket balasan dengan data sensor kepada gateway secara broadcast melewati node relay. Paket broadcast akan diterima node relay kemudian mengecek ID paket untuk mencegah duplikasi. *Node relay* mengirim broadcast paket dari node sensor. Node *gateway* menerima paket broadcast dan mencocokkan ID tujuan dengan ID node *gateway* sehingga node *gateway* akan menampilkan data dari sensor.

### Metode *Controlled Flooding*

Metode *controlled flooding* atau *flooding* terkontrol akan dibuat dengan memodifikasi algoritme *flooding* sederhana.Penggunaan metode *controlled flooding* bertujuan untuk menghindari beberapa permasalahan yang umum terjadi pada jaringan yang menggunakan metode *flooding*. Permasalahan yang akan diatasi pada metode *controlled flooding* ini yaitu adanya duplikasi paket yang diterima oleh setiap node, pengiriman paket berulang akibat adanya penerimaan duplikasi paket, dan terjadinya *collision* ketika transmisi data. Mekanisme *controlled flooding* dijelaskanpada Gambar 3.3.

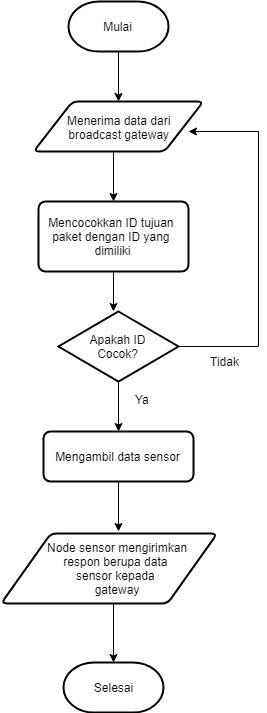


Gambar 3.3 Diagram Alir Mekanisme *Controlled Flooding*

Seperti yang dijelaskan pada Gambar 3.3, algoritme *controlled flooding* memiliki mekanisme pengecekan ID paket yang diterima apakah paket dengan ID tersebut sudah pernah diterima sebelumnya sehingga mencegah terjadi transimisi data berulang kali. Setiap *node* pada jaringan akan memiliki tabel *record* untuk menyimpan informasi ID paket, ID sumber, dan ID tujuan dari paket yang diterima. Tabel *record* tersebut tidak menyimpan informasi selamanya tetapi akan menghapus data informasi yang paling tua setiap sepuluh detik. Pengecekan paket dilakukan dengan mencocokkan ID paket dengan informasi yang ada pada tabel *record*. Setelah dilakukan pengecekan, jika paket tersebut belum pernah diterima sebelumnya maka paket tersebut akan di-*forward* kembali dengan *random delay* antara0 – 0,5 detik oleh *relay node* atau mengirim paket *reply* oleh *sensor node*. Mekanisme pengecekan paket dan mem-*forward* paket dengan random delay tersebut bertujuan untuk mengatasi adanya duplikasi penerimaan paket dan juga mengurangi terjadinya *collision*.

### Perancangan *Sensor Node*

*Sensor node* pada penelitian ini berperan untuk *sensing data* ketika mendapat permintaan dari *gateway node*. *Sensor node* terdiri dari mikrokontroler Arduino yang digunakan untuk menyimpan kode program, modul LoRa yang digunakan untuk komunikasi data, dan *sensor* DHT11 untuk mengambil data suhu. Pada mikrokontroler akan ditanamkan kode program yang telah dibuat dengan meng-*upload* program tersebut menggunakan Arduino IDE. Pemrograman pada *sensor node* membutuhkan *library* untuk menggunakan fungsi yang terdapat pada perangkat *sensor* DHT11 dan modul komunikasi LoRa. *Library* yang digunakan untuk komunikasi data pada perangkat LoRa yaitu *library* Arduino-Lora, kemudian untuk sensor DHT11 menggunakan *library* DHT *sensor* untuk dapat menggunakan fungsi untuk mengambil data dari perangkat keras *sensor*. Kemudian pada sisi perangkat keras, yaitu menggunakan *Arduino Nano* yang merupakan mikrokontroler, LoRa Ra-02 sebagai modul komunikasi, dan modul *sensor* DHT11. Ketiga komponen yaitu mikrokontroler, LoRa, dan *sensor* DHT11 dihubungkan menggunakan papan dengan rangkaian yang sudah dibuat. Diagram alir perancangan *sensor node* untuk penyelesaian permasalahan dalam sistem komunikasi *Multi-Hop* pada *Wireless Sensor Network* berbasis modul komunikasi *LoRa* menggunakan metode *Controlled* *Flooding* ditunjukkan pada Gambar 3.4.

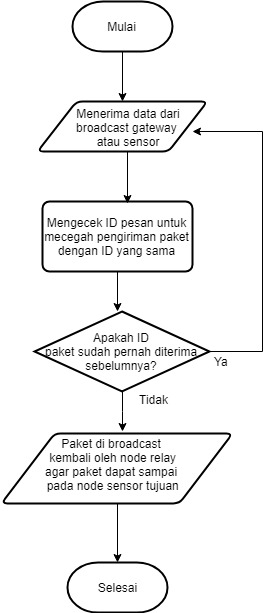


**Gambar 3.4 Diagram Alir Perancangan *Sensor Node***

Pada gambar 3.4 menjelaskan alur kerja sistem pada *sensor node* yang diawali dengan menerima pesan permintaan data dari *broadcast gateway*. Selanjutnya, mencocokkan ID tujuan paket dengan ID *node* yang dimiliki, lalu dilakukan *sensing* data sensor oleh modul DHT11. Kemudian, node sensor mengirimkan respon berupa data sensor kepada *gateway* secara *flooding*.

### Perancangan *Relay Node*

Pada penelitian ini *relay node* memiliki peran untuk meneruskan paket yang diterima dari *node* lain yang mana paket tersebut belum pernah diterima oleh *relay node* sebelumnya. *Relay node* terdiri dari 2 komponen perangkat keras utama yaitu mikrokontroler *Arduino Nano* dan modul LoRa untuk komunikasi data. Pada *relay node* juga ditanamkan kode program yang mengatur bagaimana perangkat *node* bekerja. Kode program pada *relay node* juga membutuhkan *library* yang hampir sama dengan *sensor node* yaitu *library Arduino-Lora* yang digunakan untuk komunikasi data dengan *node* yang lain. Kemudian, kode program akan ditanamkan dengan meng-*upload* kode programke mikrokontroler arduino. Diagram alir perancangan *relay node* untuk penyelesaian permasalahan dalam sistem komunikasi *Multi-Hop* pada *Wireless Sensor Network* berbasis modul komunikasi *LoRa* menggunakan metode *Controlled* *Flooding* ditunjukkan pada Gambar 3.5.

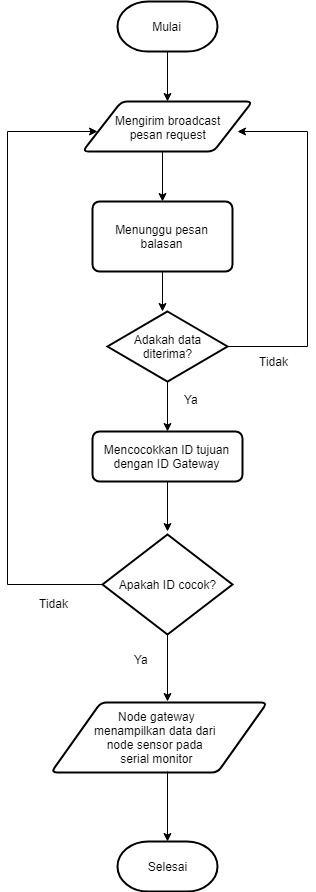


**Gambar 3.5 Diagram Alir Perancangan *Relay Node***

Pada gambar 3.5 menjelaskan proses kerja sistem pada *relay node* yang diawali dengan menerima data dari *broadcast gateway* atau sensor. Selanjutnya, paket yang diterima akan dicek ID paketnya untuk mencegah duplikasi data atau pengiriman paket dengan ID yang sama. Kemudian, paket di kirim kembali secara *flooding* oleh *relay node* agar paket dapat sampai pada *node* yang dituju.

### Perancangan *Gateway Node*

Pada penelitian ini *gateway node* memiliki peran untuk mengirim paket *request* menuju dan menerima paket balasan dari *sensor node* lalu menampilkan data *sensor* pada *serial monitor*. Pada *gateway node* terdiri dari perangkat keras mikrokontroler *Arduino Nano* yang digunakan untuk menyimpan kode program dan modul LoRa yang digunakan untuk komunikasi data dengan *node* yang lain. Kode program pada *gateway node* juga membutuhkan *library Arduino-Lora* untuk bertukar data dengan *node* yang lain. Kemudian, kode program akan ditanamkan pada mikrokontroler *Arduino Nano* dengan melakukan *uploading* kode program ke mikrokontroler tersebut. Diagram alir perancangan *gateway node* untuk penyelesaian permasalahan dalam sistem komunikasi *Multi-Hop* pada *Wireless Sensor Network* berbasis modul komunikasi *LoRa* menggunakan metode *Controlled Flooding* ditunjukkan pada Gambar 3.6.



**Gambar 3.6 Diagram Alir Perancangan *Gateway Node***

Pada Gambar 3.6 dijelaskan proses kerja sistem pada *gateway node* yang diawali dengan *gateway* mengirim *broadcast* pesan *request* yang berisi ID *node sensor* yang dituju. Selanjutnya, *gateway node* menunggu pesan balasan yang dikirim oleh *sensor node*. Kemudian, setelah mendapat paket balasan maka ID tujuan paket dicocokkan dengan ID *node* milik *gateway*, Jika ID cocok maka data akan ditampilkan pada serial monitor.

## Implementasi

Implementasi merupakan tahapan yang dilakukan untuk membangun suatu sistem. Implementasi dilakukan berdasarkan perancangan sistem yang telah dibuat pada tahap perancangan. Implementasi yang dilakukan meliputi implementasi lingkungan uji dan implementasi metode *controlled flooding*.

### Lingkungan Uji

Implementasi lingkungan uji merupakan tahapan yang dilakukan untuk menyusun lingkungan sistem yang telah dijelaskan pada subbab kebutuhan dan perancangan lingkungan uji. Implementasi yang dilakukan yaitu menyusun beberapa perangkat keras yang akan digunakan sebagai *node* sehingga terbentuk suatu jaringan yang terdiri berikut ini:

*Sensor node*

Perangkat sensor node terdiri dari tiga buah perangkat keras yang terhubung dengan papan PCB yang harus dihubungkan melalui pin-pin yang terdapat pada ketiga perangkat tersebut. Perangkat keras pada *sensor node* meliputi mikrokontroller Arduino Nano, modul LoRa Ra-02 SX1278 dan *sensor* DHT11 untuk mengambil data suhu. *Pinout* yang terhubung antara Arduino Nano dan LoRa Ra-02 SX1278 dijelaskan pada Tabel 3.1.

Tabel 3.1 *Pinout* PCB untuk Arduino Nano dan LoRa Ra02 SX1278

|  |  |
| --- | --- |
| **Arduino Nano** | **LoRa Ra-02 SX1278** |
| GND | GND |
| 3V3 | 3.3V |
| D2 | DIO0 |
| RST | RST |
| D10 | NSS |
| D11 | MOSI |
| D12 | MISO |
| D13 | SCK |

Tabel 3.1 menunjukkan pin mana saja yang digunakan pada masing – masing perangkat untuk menghubungkan kedua perangkat. Jika kedua perangkat sudah terhubung dengan benar maka Arduino nano dapat menggunakan LoRa untuk transmisi data. Selanjutnya *pinout* yang menghubungkan antara Arduino Nano dan *sensor* DHT11 dapat dilihat pada Tabel 3.2.

Tabel 3.2 *Pinout* PCB untuk Arduino Nano dan *sensor* DHT11

|  |  |
| --- | --- |
| **Arduino Nano** | ***Sensor* DHT11** |
| GND | GND |
| +5V | VCC (+) |
| D3 | OUT |

Tabel 3.2 menunjukkan pin mana saja yang digunakan pada masing – masing perangkat untuk menghubungkan kedua perangkat. Jika kedua perangkat sudah terhubung dengan benar maka Arduino nano dapat menggunakan *sensor* DHT11 untuk mengambil data suhu. Setelah semua perangkat terhubung dengan benar, akan dilakukan pengunggahan kode program pada mikrokontroler Arduino Nano untuk mengatur kerja dari perangkat.

*Relay node*

Implementasi pada *relay node* yaitu merakit perangkat keras yang menjadi penyusun *relay node*. Perangkat keras penyusun *relay node* terdiri dari dua komponen yaitu mikrokontroller Arduino Nano dan modul LoRa Ra-02 SX1278. Kedua komponen tersebut dihubungkan dengan papan PCB yang terhubung dengan pinout pada masing – masing komponen. *Pinout* yang terhubung antara Arduino Nano dan LoRa Ra-02 SX1278 dijelaskan pada Tabel 3.3.

Tabel 3.3 *Pinout* PCB untuk Arduino Nano dan LoRa Ra02 SX1278

|  |  |
| --- | --- |
| **Arduino Nano** | **LoRa Ra-02 SX1278** |
| GND | GND |
| 3V3 | 3.3V |
| D2 | DIO0 |
| RST | RST |
| D10 | NSS |
| D11 | MOSI |
| D12 | MISO |
| D13 | SCK |

Tabel 3.3 menunjukkan pin mana saja yang digunakan pada masing – masing perangkat untuk menghubungkan kedua perangkat. Jika kedua perangkat sudah terhubung dengan benar maka Arduino nano dapat menggunakan LoRa untuk transmisi data. Setelah semua perangkat terhubung dengan benar, akan dilakukan pengunggahan kode program pada mikrokontroler Arduino Nano untuk mengatur kerja dari perangkat.

*Gateway node*

Implementasi pada *gateway node* sama seperti *relay node* yaitu menggunakan dua perangkat keras yang terdiri dari mikrokontroler Arduino Nano dan modul LoRa Ra-02 SX1278 dihubungkan dengan papan PCB yang terhubung dengan pinout pada masing – masing komponen. *Pinout* yang terhubung antara Arduino Nano dan LoRa Ra-02 SX1278 dijelaskan pada Tabel 3.4.

Tabel 3.4 *Pinout Pinout* PCB untuk Arduino Nano dan LoRa Ra02 SX1278

|  |  |
| --- | --- |
| **Arduino Nano** | **LoRa Ra-02 SX1278** |
| GND | GND |
| 3V3 | 3.3V |
| D2 | DIO0 |
| RST | RST |
| D10 | NSS |
| D11 | MOSI |
| D12 | MISO |
| D13 | SCK |

Tabel 3.4 menunjukkan pin mana saja yang digunakan pada masing – masing perangkat untuk menghubungkan kedua perangkat. Jika kedua perangkat sudah terhubung dengan benar maka Arduino nano dapat menggunakan LoRa untuk transmisi data. Kemudian akan dilakukan pengunggahan kode program pada mikrokontroler Arduino Nano setelah kedua perangkat penyusun *gateway node* terhubung dengan benar untuk memberikan logika kerja pada perangkat.

### Metode *Controlled Flooding*

Implementasi metode *controlled flooding* atau *flooding* terkontrol akan dibuat dengan memodifikasi algoritme *flooding* sederhana.Modifikasi yang dimaksud adalah dengan menambahkan beberapa mekanisme yang bertujuan untuk mengatasi beberapa kelemahan pada algoritme *flooding*. Mekanisme pada metode *controlled flooding* akan dijabarkan sebagai berikut:

Mekanisme *forwarding*

Mekanisme *forwarding* pada penelitian dibutuhkan untuk pengiriman paket pada LoRa karena pada dasarnya hanya sebatas *single-hop*. Sehingga mekanisme ini berfungsi untuk menerapkan sistem komunikasi *multi-hop* pada LoRa.

Mekanisme pencegahan *infinite loop*

Salah satu kelemahan pada metode *flooding* yaitudapat terjadi *infinite loop* atau perulangan transmisi yang tak terbatas. Perulangan tersebut dapat terjadi akibat suatu *node* dapat menerima paket dengan id yang sama dan diteruskan kembali ke *node* tetangga. Oleh karena itu, mekanisme ini digunakan untuk mengatasi hal seperti yang sudah dijelaskan.

Mekanisme *random delay*

Kelemahan lain yang mungkin dapat terjadi ketika menerapkan metode *flooding* adalah adanya *collision* yang menyebabkan *packet loss*. *Collision* dapat terjadi akibat adanya *node* yang bertetangga lalu bersamaan mengirim paket sehingga sangat memungkinkan terjadinya tabrakan atau *collision*, sedangkan LoRa tidak memiliki mekanisme *collision avoidance* ataupun mekanisme *collision detection*. Salah satu solusi yang digunakan pada penelitian ini untuk mengurangi kemungkinan terjadinya *collision* yaitu dengan menambahkan mekanisme *random delay transmission* pada setiap perangkat *node* ketika hendak mengirimkan atau meneruskan paket data.

Mekanisme *retransmission packet*

Penelitian menggunakan metode *flooding* sebagai metode komunikasi memungkinkan terjadinya *packet loss*, untuk mengatasi kondisi tersebut pada penelitian ini menerapkan mekanisme *retransmission packet* agar paket dikirim kembali.

Dengan mekanisme – mekasnime diatas, penelitian ini dibuat untuk membangun suatu sistem komunikasi *multi-hop* pada WSN berbasis LoRa.

## Metode Evaluasi

Metode evaluasi pada penelitian ini menjelaskan tentang parameter dan scenario uji yang digunakan sebagai dasar untuk melakukan pengujian dan pembahasan. Pengujian yang dilakukan pada penelitian ini yaitu pengujian komunikasi *multi-hop* pada LoRa, pengujian pencegahan infinite loop, dan pengujian kinerja sistem. Berdasarkan penjelasan sebelumnya maka metode evaluasi dibagi sebagai berikut:

Pengujian komunikasi *multi-hop* pada LoRa

Pengujian komunikasi *multi-hop* pada LoRa dilakukan untuk menguji fungsionalitas dari sistem komunikasi *multi-hop* pada WSN berbasis modul komunikasi LoRa. Pengujian ini dilakukan untuk mengetahui apakah komunikasi *multi-hop* pada LoRa dapat terjadi dengan memanfaatkan *node* perantara atau *relay node* untuk meneruskan paket menuju tujuan. Pengujian ini dilakukan dengan melakukan percobaan pengiriman paket dari *gateway node* menuju *sensor node* yang mana melalui *relay node* karena kedua *node* tersebut tidak saling menjangkau. Pada kode program akan ditambahkan variabel untuk menyimpan informasi jalur yang dilewati oleh paket.

Pengujian mekanisme pencegahan *infinite loop*

Pencegahan *infinite loop* pada penelitian ini merupakan mekanisme yang bertujuan untuk mencegah terjadinya pengiriman paket berulang kali yang menyebabkan masalah pada jaringan WSN. Pengujian ini dilakukan untuk mengetahui apakah mekanisme pencegahan *infinite loop* di setiap *node* pada WSN berbasis LoRa dapat terjadi dengan menerapkan mekanisme tersebut. Langkah pengujian mekanisme ini yaitu dengan menambahkan kode program pada setiap *node* untuk mencetak informasi ketika menerima paket dengan ID yang sama. Kemudian akan dilakukan percobaan pengiriman paket untuk mendapatkan hasil pengujian.

Pengujian mekanisme *retransmission packet*

*Retransmission packet* merupakan mekanisme yang bertujuan untuk mengatasi jika terjadi *collision* yang menyebabkan terjadinya *packet loss* dengan memberikan waktu *timeout* untuk menerima paket *reply*. Langkah pengujian mekanisme ini yaitu dengan melakukan pengiriman paket dan mematikan *sensor node* sehingga paket *reply* tidak akan diterima dan dinyatakan sebagai *packet loss*. Hasil pengujian akan dilihat pada perangkat *gateway node* melalui *serial monitor* pada perangkat lunak Arduino IDE.

Pengujian kinerja sistem

Pengujian ini dilakukan agar diketahui bagaimana kemampuan sistem dengan penerapan metode *controlled flooding* pada WSN menggunakan LoRa sebagai modul komunikasi. Pengujian yang dilakukan pada penelitian ini yaitu menghitung persentase paket *loss, round trip time* dan *troughput*. Menghitung persentase paket *loss* bertujuan untuk mengetahui kualitas transmisi data dengan melihat seberapa banyak persentase paket yang hilang dan tidak mendapat balasan. Menghitung *round trip time* bertujuan untuk mengukur efisiensi waktu dari penggunaan metode *controlled flooding*. Penghitungan *troughput* sendiri bertujuan untuk mengetahui rata – rata kecepatan transmisi data pada jaringan.

## Penarikan Kesimpulan

Tahap ini merupakan tahap penarikan kesimpulan berdasarkan analisis dan pembahasan mengenai pengujian fungsional dan pengujian kinerja dari sistem komunikasi *multi-hop* menggunakan metode *controlled flooding* pada jaringan WSN berbasis LoRa. Pada tahap ini bisa diketahui apakah penelitian ini berhasil atau tidak berdasarkan rumusan masalah dan tujuan. Selain mengambil kesimpulan dari hasil pengujian, peneliti juga memberikan saran untuk penelitian lanjutan.

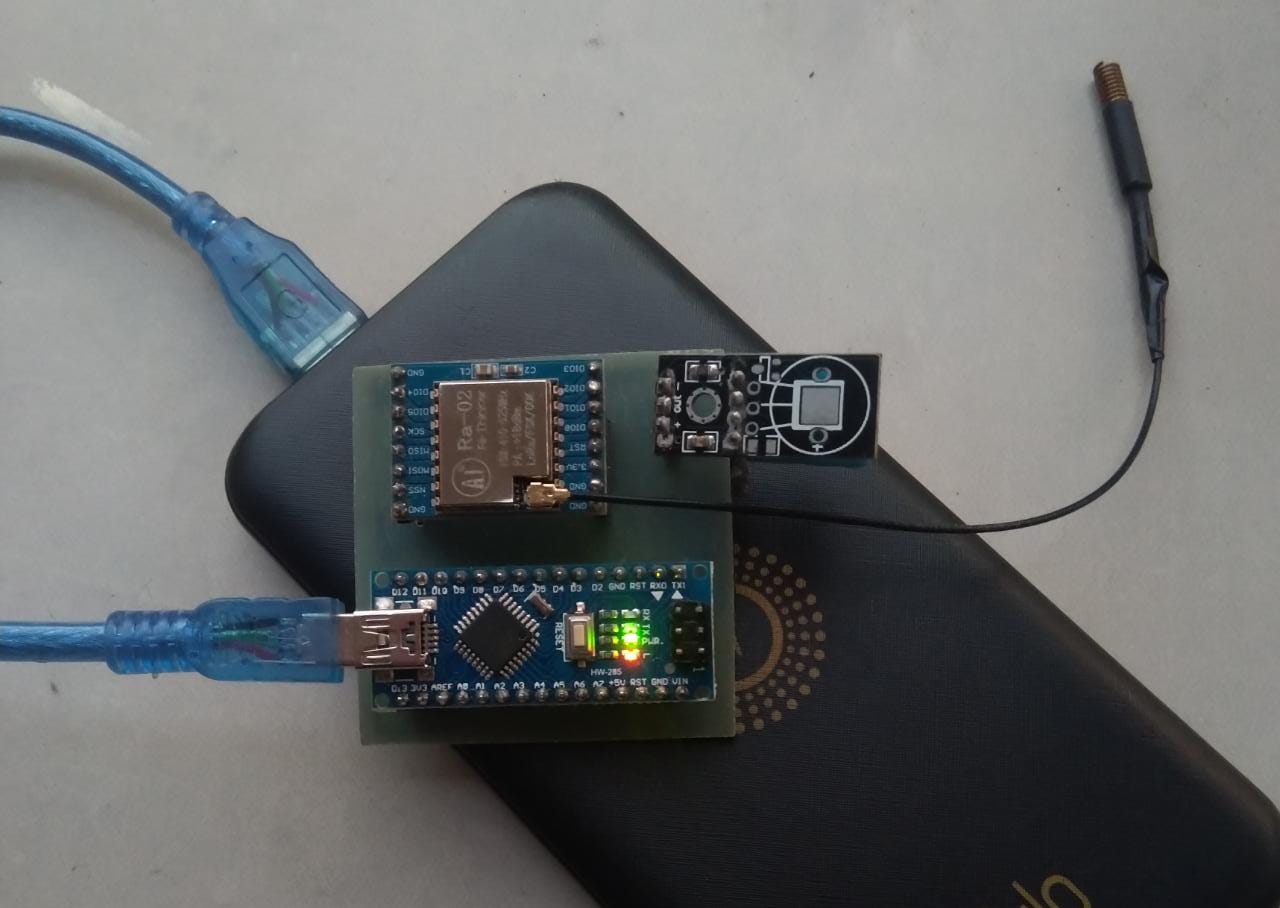
# IMPLEMENTASI

Bab ini membahas tentang mekanisme implementasi dari sistem komunikasi *Multi-Hop* pada *Wireless Sensor Network* berbasis modul komunikasi *LoRa* menggunakan metode *Controlled* *Flooding* yang dibangun dalam bentuk program. Implementasi mencakup mekanisme *forwarding*, pencegahan *infinite loop* ataupengiriman paket dengan ID yang sama, mekanisme mencegah dan mengatasi adanya *collision*.

## Lingkungan Uji

### *Sensor Node*

Perangkat sensor node terdiri dari tiga buah perangkat keras yang terhubung dengan papan PCB yang harus dihubungkan melalui pin-pin yang terdapat pada ketiga perangkat tersebut. Perangkat keras pada *sensor node* meliputi mikrokontroller Arduino Nano, modul LoRa Ra-02 SX1278 dan sensor DHT11 untuk mengambil data suhu. Setelah ketiga perangkat tersebut terhubung dengan benar, daya pada *sensor node* diberikan melalui mikrokontroller Arduino Nano. Arduino Nano disambungkan pada laptop ataupun *power bank* dengan kabel usb mikro untuk mendapatkan sumber daya listrik seperti pada Gambar 4.1.

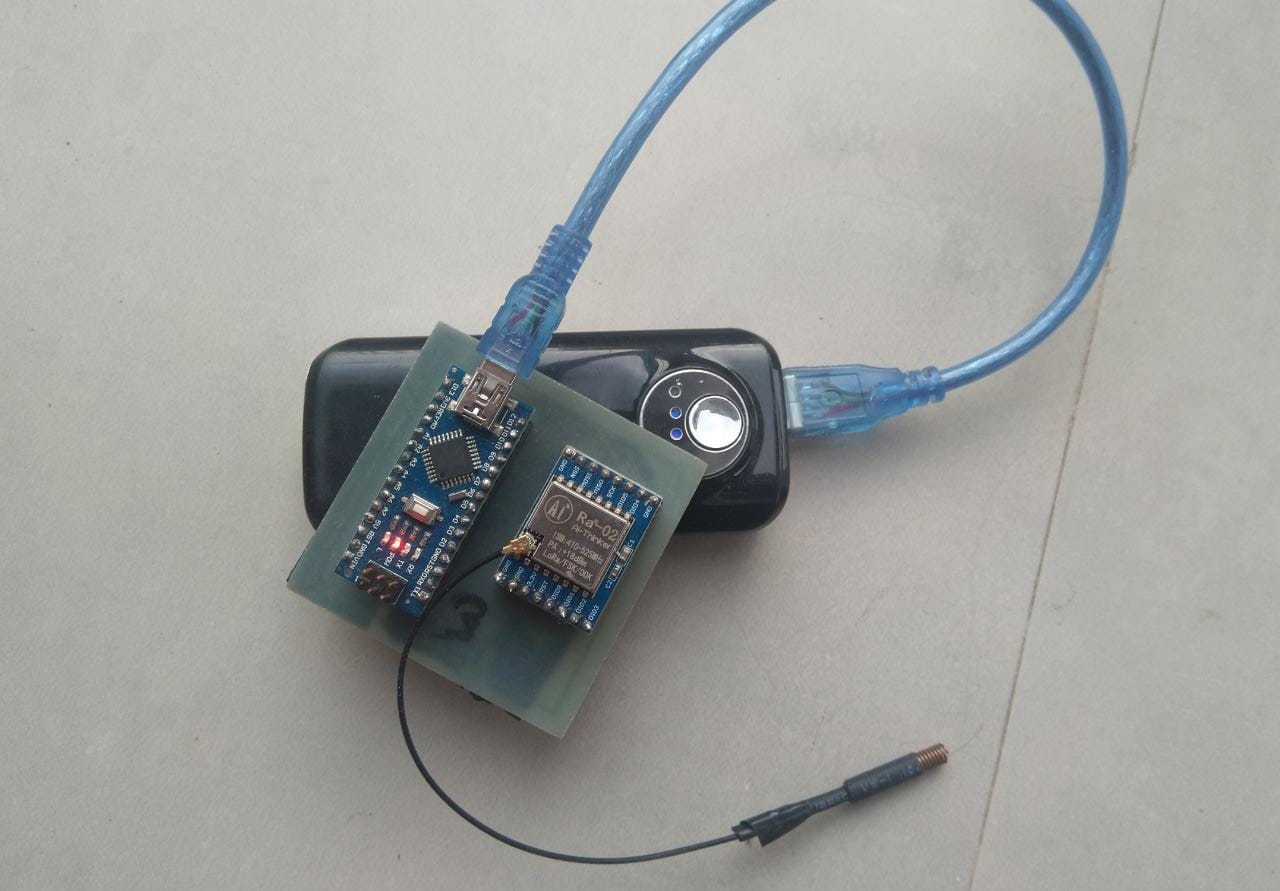


Gambar 4.1 *Sensor Node*

Langkah selanjutnya setelah semua perangkat untuk menyusun *sensor node* sudah terpasang dengan benar yaitu dengan mengunggah kode program pada mikrokontroler arduino nano. Sebelum pembuatan kode program, akan dilakukan instalasi *library* pendukung untuk menulis kode program. *Library* pendukung tersebut yaitu *library* *Arduino-lora* dan DHT11. *Library* tersebut akan memudahkan dalam pembuatan program dengan menggunakan fungsi yang sudah ada pada *library* tersebut. Kode program yang dibuat untuk *sensor node* memiliki fungsional yaitu mengambil data suhu ketika menerima paket *request* lalu mengirimkannya dalam paket *reply*. Selain itu pada *sensor node* juga memiliki fungsi pencegahan *infinite loop transmission*.

### *Relay Node*

Implementasi pada *relay node* yaitu merakit perangkat keras yang menjadi penyusun *relay node*. Perangkat keras penyusun *relay node* terdiri dari dua komponen yaitu mikrokontroller Arduino Nano dan modul LoRa Ra-02 SX1278. Kedua komponen tersebut dihubungkan dengan papan PCB yang terhubung dengan pinout pada masing – masing komponen. Setelah kedua perangkat tersebut dihubungkan dengan benar, daya yang digunakan pada *relay node* sama seperti *sensor node* yaitu *power bank* yang dihubungkan dengan mikrokontroler Arduino Nano menggunakan kabel usb seperti yang ditunjukkan pada Gambar 4.2.



Gambar 4.2 *Relay Node*

Setelah semua perangkat untuk menyusun *relay node* sudah terpasang dengan benar, selanjutnya mengunggah kode program pada mikrokontroler arduino nano. Sebelum pembuatan kode program, akan dilakukan instalasi *library* pendukung untuk menulis kode program. *Library* pendukung tersebut yaitu *library* *Arduino-lora*. *Library* tersebut akan memudahkan dalam pembuatan program dengan menggunakan fungsi yang sudah ada pada *library* tersebut. Kode program yang dibuat untuk *relay node* memiliki fungsional yaitu mencegahan *infinite loop transmission* dan mengurangi adanya *collision* dengan *random delay*.

### *Gateway Node*

Implementasi pada *gateway node* sama seperti *relay node* yaitu menggunakan dua perangkat keras yang terdiri dari mikrokontroler Arduino Nano dan modul LoRa Ra-02 SX1278 dihubungkan dengan papan PCB yang terhubung dengan pinout pada masing – masing komponen. Setelah kedua perangkat tersebut dihubungkan dengan benar, *gateway node* menggunakan laptopsebagai pemberi daya perangkat yang dihubungkan dengan mikrokontroler Arduino Nano menggunakan kabel usb seperti yang ditunjukkan pada Gambar 4.3. *Gateway node* menggunakan laptop sebagai pemberi daya sekaligus untuk menguji apakah transmisi data berhasil dilakukan dengan melihat hasil pada *serial monitor* di perangkat lunak Arduino IDE.



Gambar 4.3 *Gateway Node*

Kemudian setelah semua perangkat penyusun *gateway node* sudah terpasang dengan benar, kode program akan diunggah pada mikrokontroler arduino nano. Sebelum itu, kode program akan dibuat dengan *library* pendukung yang sama dengan *node* yang lain yaitu *library* *Arduino-lora*. *Library* tersebut akan memudahkan dalam pembuatan program dengan menggunakan fungsi yang sudah ada pada *library* tersebut. Kode program yang dibuat untuk *gateway node* memiliki fungsional yaitu mencegahan *infinite loop transmission* dan mengatasi adanya *collision* atau *packet loss* dengan mekanisme *retransmission packet*.

## Mekanisme *Controlled Flooding*

Pada penelitian ini, implementasi komunikasi LoRa menggunakan metode *controlled flooding*. Pada *controlled flooding* terdapat beberapa mekanisme yang berfungsi untuk mengontrol *flooding* yaitu mekanisme *forwarding*, pencegahan *infinite loop* ataupengiriman paket dengan ID yang sama, mekanisme mencegah dan mengatasi adanya *collision*, dan mekanisme untuk mengatasi jika terjadi *collision* yang menyebabkan *packet loss*.

### Mekanisme *Forwarding*

Pengiriman paket menggunakan modul komunikasi LoRa pada dasarnya hanya sebatas *single-hop*.Oleh karena itu, implementasi *flooding* untuk pengiriman paket pada LoRa membutuhkan mekanisme *forwarding* agar paket dapat terkirim ke *node* paling ujung. Mekanisme *forwarding* ini diterapkan dengan menambahkan beberapa baris kode program. Kode program tersebut berupa fungsi yang dapat menyimpan kumpulan data yang diterima pada variabel untuk dilakukan pengiriman kembali pada *node – node* tetangganya. Berikut kode program mekanisme *forwarding*.

Tabel 4.1 Mekanisme *Forwarding*

|  |  |
| --- | --- |
| Mekanisme Forwarding | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33 | void forwardRequest(byte recipient, byte sender, byte msgId, byte msgType, byte sendTime[], String path) {  LoRa.beginPacket(); // start packet  LoRa.write(recipient); // add destination address  LoRa.write(sender); // add sender address  LoRa.write(msgId); // add message ID  LoRa.write(msgType); // add message type  // add message sendTIme  LoRa.write(sendTime[0]);  LoRa.write(sendTime[1]);  LoRa.write(sendTime[2]);  LoRa.write(sendTime[3]);  LoRa.write(path.length());  LoRa.print(path);  LoRa.endPacket(); // finish packet and send it  }  void forwardReply(byte recipient, byte sender, byte msgId, byte msgType, byte sendTime[], byte sensor, byte reqId, String path) {  LoRa.beginPacket(); // start packet  LoRa.write(recipient); // add destination address  LoRa.write(sender); // add sender address  LoRa.write(msgId); // add message ID  LoRa.write(msgType); // add message type  LoRa.write(reqId); // add id pkt request  LoRa.write(sensor); // add payload data sensor  // add message sendTIme  LoRa.write(sendTime[0]);  LoRa.write(sendTime[1]);  LoRa.write(sendTime[2]);  LoRa.write(sendTime[3]);  LoRa.write(path.length());  LoRa.print(path);  LoRa.endPacket(); // finish packet and send it  } |

Berdasarkan kode program pada Tabel 4.1 diatas, terdapat dua fungsi *forwarding* yaituuntuk meneruskan paket *request* dan paket *reply*. Fungsi dibedakan menjadi dua karena isi paket yang dikirimkan berbeda. Fungsi tersebut dipanggil ketika perangkat *router* menerima paket yang kemudian isi paket tersebut akan diteruskan pada *node* sekitarnya hingga mencapai tujuan. Penjelasan dari kode program tersebut akan dijelaskan pada keterangan dibawah ini:

* Baris 1, merupakan *method* untuk melayani *forwarding* paket *request* dengan parameter ID perangkat tujuan, ID perangkat pengirim, ID paket, jenis paket, waktu pengiriman, dan jalur yang telah dilewati.
* Baris 2, merupakan fungsi dari *library* LoRa sebagai pembuka dari isi paket.
* Baris 3 - 13, merupakan fungsi dari *library* LoRa untuk menambahkan ID perangkat tujuan, ID perangkat pengirim, ID paket, jenis paket, waktu pengiriman, dan jalur yang telah dilewati kedalam paket LoRa.
* Baris 14, merupakan fungsi dari *library* LoRa sebagai penutup dari paket LoRa sekaligus mengirim paket tersebut.
* Baris 17, merupakan *method* untuk melayani *forwarding* paket *reply* dengan parameter ID perangkat tujuan, ID perangkat pengirim, ID paket, jenis paket, waktu pengiriman, ID paket *request*, data sensor dan jalur yang telah dilewati.
* Baris 18, merupakan fungsi dari *library* LoRa sebagai pembuka dari isi paket.
* Baris 19 – 31, merupakan fungsi dari *library* LoRa untuk menambahkan ID perangkat tujuan, ID perangkat pengirim, ID paket, jenis paket, waktu pengiriman, ID paket *request*, data sensor dan jalur yang telah dilewati kedalam paket LoRa.

Baris 32, merupakan fungsi dari *library* LoRa sebagai penutup dari paket LoRa sekaligus mengirim paket tersebut.

### Pencegahan *Infinite Loop*

Metode *flooding* pada umumnya memiliki beberapa kelemahan, salah satunya yaitu dapat terjadi *infinite loop* atau perulangan transmisi yang tak terbatas. Perulangan tersebut dapat terjadi akibat suatu *node* dapat menerima paket dengan id yang sama dan diteruskan kembali ke *node* tetangga. Salah satu solusi yang akan dilakukan pada penelitian ini yaitu dengan menyimpan informasi paket yang telah lewat pada tabel histori transmisi. Berikut *pseudocode*  dari mekanisme pencegahan *infinite loop*.

Tabel 4.2 Pencegahan *Infinite Loop*

|  |  |
| --- | --- |
| Mekanisme Pencegahan Infinite Loop | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123 | INIT localAddress  INIT record[8] = (currentSender, currentRecipient, currentMsgId)  INIT lastPop time  Function loop:  IF millis() - lastPop > interval:  Call pop()  SET lastPop = GET time  ENDIF  Call onReceive(LoRa.parsePacket())  Endfunction  Function onReceive(packetSize):  IF packetSize < 10:  Return  ENDIF  // read packet header bytes:  INIT recipient = READ data stream  INIT sender = READ data stream  INIT incomingMsgId = READ data stream  INIT incomingMsgType = READ data stream  INIT reqId = 0  INIT incomingData = 0  IF incomingMsgType == 1:  SET reqId = READ data stream  SET incomingData = READ data stream  ENDIF  INIT waktu = READ data stream  INIT pathLength = READ data stream  INIT path = READ data stream  SET path += "-" + localAddress  // if the recipient isn't this device  IF recipient != localAddress:  IF Call search(sender, recipient, incomingMsgId):  PRINT "Message id : "  PRINT incomingMsgId  PRINT " has been received before"  Return  ENDIF    Call delay(random(50))  IF incomingMsgType == 0:  Call forwardRequest(recipient, sender, incomingMsgId, incomingMsgType, waktu, path)  ELSE IF incomingMsgType == 1:  Call forwardReply(recipient, sender, incomingMsgId, incomingMsgType, waktu, incomingData, reqId, path)  ENDIF    PRINT "forward packet id:"  PRINT incomingMsgId  PRINT "from: "  PRINT sender  PRINT "coba size paket: "  PRINT packetSize  PRINT path  Call push(sender, recipient, incomingMsgId)  ENDIF  Endfunction  Function pop()  INIT x = 0  FOR x < 8 :  IF x < 7:  IF record[x].currentSender == 0 && record[x].currentRecipient == 0:  Return  ENDIF  SET record[x].currentSender = record[x+1].currentSender  SET record[x].currentRecipient = record[x+1].currentRecipient  SET record[x].currentMsgId = record[x+1].currentMsgId  ELSE:  SET record[x].currentSender = 0  SET record[x].currentRecipient = 0  SET record[x].currentMsgId = 0  ENDIF  INCREMENT x  ENDFOR  Endfunction  Function push(byte senderId, byte recipientId, byte packetId):  INIT x = 0  FOR x < 8:  IF x < 7:  IF record[x].currentSender == 0 && record[x].currentRecipient == 0:  record[x] = (senderId,recipientId,packetId)  Return  ELSE:  IF record[x].currentSender == 0 && record[x].currentRecipient == 0:  SET record[x] = (senderId,recipientId,packetId)  ELSE:  Call pop()  SET record[x] = (senderId,recipientId,packetId)  ENDIF  ENDIF  INCREMENT x  ENDFOR  Endfunction  Function search(byte senderId, byte recipientId, byte packetId):  INIT x = 0  FOR x < 8:  IF record[x].currentSender == senderId && record[x].currentRecipient == recipientId && record[x].currentMsgId == packetId):  Return 1  ENDIF  INCREMENT x  ENDFOR  Return 0  Endfunction |

Berdasarkan kode program pada Tabel 4.2 diatas , terdapat mekanisme dari pencegahan *infinite loop* yang bertujuanuntuk mencegah terjadinya pengirimanpaket yang sama berulang kali sehingga paket yang sudah pernah lewat sebelumnya tidak akan diproses lagi. Mekanisme tersebut yaitu dengan menerapkan tabel untuk menyimpan informasi paket yang diterima berupa ID pengirim, ID tujuan dan ID paket dengan sistem *first in first out*(FIFO). Tabel yang digunakan tersebut merupakan sebuah array dengan panjang delapan indeks yang digunakan untuk menyimpan kumpulan informasi ID dalam variabel data *struct*. Penjelasan dari kode program tersebut akan dijelaskan pada keterangan dibawah ini:

* Baris 1-4, merupakan inisialisasi variabel yang diperlukan untuk melakukan pencegahan *infinite loop* seperti variabel localAddress untuk ID *node*, record[8] untuk menyimpan informasi paket yang diterima, dan lastPop untuk menyimpan nilai waktu terakhir melakukan hapus data record paling awal.
* Baris 6, merupakan fungsi loop pada arduino
* Baris 7 - 10, merupakan seleksi untuk melakukan pop atau penghapusan data paling tua pada tabel record ketika waktu sudah mencapai interval 10 detik.
* Baris 12, memanggil fungsi untuk menangani ketika terdapat paket LoRa yang diterima.
* Baris 13, merupakan penutup dari fungsi loop.
* Baris 15, merupakan fungsi untuk memproses paket LoRa yang diterima.
* Baris 16 – 18, merupakan seleksi ukuran paket jika kurang dari 10 byte maka tidak akan diproses lebih lanjut lalu kembali dalam mode mendengar.
* Baris 21 – 34, menyimpan aliran data yang diterima kedalam variabel yang telah ditentukan.
* Baris 36, merupakan penambahan ID *node* kedalam data *path*.
* Baris 39, merupakan seleksi jika ID tujuan bukan *node* ini maka akan diproses ke tahap selanjutnya.
* Baris 40 – 45, merupakan seleksi jika paket pernah lewat sebelumnya maka akan melakukan cetak pada serial monitor bahwa paket tersebut sudah pernah diterima sebelumnya dan kembali ke mode mendengar. Jika paket belum pernah diterima sebelumnya maka akan diproses ke tahap selanjutnya.
* Baris 47, memanggil fungsi *delay* untuk memberikan *delay* dengan nilai waktu secara acak antara 0 – 50 ms.
* Baris 48 – 54, merupakan seleksi jika paket yang terima merupakan paket *request* maka akan memanggil fungsi forwardRequest. Sebaliknya, jika paket yang terima merupakan paket *reply* maka akan memanggil fungsi forwardReply.
* Baris 56 – 62, mencetak informasi pada serial monitor bahwa paket telah diteruskan pada *node* tetangga.
* Baris 63, memanggil fungsi push untuk menambahkan informasi paket yang diterima kedalam tabel record.
* Baris 67 – 87, merupakan fungsi pop untuk menghapus data paling tua pada tabel record.
* Baris 89 – 109, merupakan fungsi push untuk menambahkan data kedalam tabel record.
* Baris 111 – 123, merupakan fungsi mencari apakah data yang dicari sudah ada pada tabel record.

### Mekanisme *Random Delay*

Kelemahan lain yang mungkin dapat terjadi ketika menerapkan metode *flooding* adalah adanya *collision* yang menyebabkan *packet loss*. *Collision* dapat terjadi akibat adanya *node* yang bertetangga lalu bersamaan mengirim paket sehingga sangat memungkinkan terjadinya tabrakan atau *collision*, sedangkan LoRa tidak memiliki mekanisme *collision avoidance* ataupun mekanisme *collision detection*. Salah satu solusi yang digunakan pada penelitian ini untuk mengurangi kemungkinan terjadinya *collision* yaitu dengan menambahkan mekanisme *random delay* pada kode program. *Random delay* bertujuan untuk mencegah terjadinya mengirim paket dalam waktu yang sama dengan memberikan sejumlah *delay* waktu secara acak. Berikut kode program *random delay* yang ditambahkan.

Tabel 4.3 Mekanisme *Random Delay*

|  |  |
| --- | --- |
| Mekanisme Random Delay | |
| 1  2  3  4  5  6 | delay(random(50));  if (incomingMsgType == 0) {  forwardRequest(recipient, sender, incomingMsgId, incomingMsgType, waktu, path);  } else if (incomingMsgType == 1) {  forwardReply(recipient, sender, incomingMsgId, incomingMsgType, waktu, incomingData, reqId, path);  } |

Berdasarkan kode program pada Tabel 4.3 diatas , terdapat fungsi dari *random delay* yaituuntuk mencegah terjadinya *forwarding* paket secara bersamaan. *Delay* yang di berikan merupakan nilai acak antara 0 – 50 ms sehingga *node* dapat segera kembali dalam mode *receiver*. Penjelasan dari kode program tersebut akan dijelaskan pada keterangan dibawah ini:

* Baris 1, merupakan fungsi untuk memberikan *delay* pada program dengan nilai waktu *random* antara 0 – 50 ms.
* Baris 2, merupakan seleksi jenis paket yang masuk apakah paket yang diterima adalah paket *request* atau tidak.
* Baris 3, merupakan fungsi untuk *forwarding* paket *request*dengan parameter ID tujuan, ID pengirim, ID paket, jenis paket yang masuk, waktu pengiriman, dan jalur yang dilewati.
* Baris 4, merupakan seleksi jenis paket yang masuk apakah paket yang diterima adalah paket *reply* atau tidak.
* Baris 5, merupakan fungsi untuk *forwarding* paket *reply* dengan parameter ID tujuan, ID pengirim, ID paket, jenis paket yang masuk, waktu pengiriman, data *sensor*, ID paket *request* dan jalur yang dilewati.

### Mekanisme *Retransmission Packet*

Penggunaan *flooding* sebagai metode komunikasi memungkinkan terjadinya *packet loss*, untuk mengatasi kondisi tersebut pada penelitian ini menerapkan mekanisme *retransmission packet* agar paket dikirim kembali. Retransmisi ini dilakukan oleh *gateway node* ketika tidak menerima paket balasan dengan waktu *timeout* selama tiga detik. Berikut ini kode program mekanisme retransmisi pada *gateway node*.

Tabel 4.4 Mekanisme *Retransmission Packet*

|  |  |
| --- | --- |
| Mekanisme Retransmission Packet | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25 | INIT lastRetransTime  INIT timeout  Function waitReply():  INIT lora = 0  INIT totTimeout = 0  WHILE lora == 0:  SET lora = LoRa.parsePacket()  IF lora > 0:  IF not onReceive(lora):  SET lora = 0  ENDIF  ENDIF  IF totTimeout < 3:  IF millis() - lastRetransTime > timeout:  PRINT "Retransmisi..."  Call sendRequest()  SET lastRetransTime = millis()  INCREMENT totTimeout  ENDIF  ELSE:  Return  ENDIF  ENDWHILE  Endfunction |

Berdasarkan kode program pada Tabel 4.4 diatas , terdapat fungsi dari *retransmisson packet* yaituuntuk mengatasi jika terjadi *packet loss* maka akan dilakukan pengiriman kembali paket yang hilang. Retransmisi dilakukan ketika paket *reply* belum diterima dengan waktu timeout selama 3 detik. Penjelasan dari kode program tersebut akan dijelaskan pada keterangan dibawah ini:

* Baris 1 – 2, merupakan inisiasi variabel lastRetransTime untuk menyimpan nilai waktu retransmisi terakhir dan variabel timeout untuk menentukan waktu timeout dari transmisi paket.
* Baris 4, merupakan fungsi untuk menunggu paket balasan.
* Baris 5 – 6, merupakan inisiasi variabel lora untuk menyimpan besar paket yang diterima dan variabel totTimeout untuk menyimpan jumlah terjadinya timeout.
* Baris 7, merupakan perulangan untuk menerima paket lora.
* Baris 8, menyimpan besar lora pada variabel lora.
* Baris 9 – 13, merupakan pengecekan apakah paket yang diterima merupakan balasan yang benar.
* Baris 14, merupakan pengecekan jika total timeout masih kurang dari tiga maka dilanjutkan ke tahap selanjutnya.
* Baris 15 – 20, jika selama 3 detik belum menerima balasan maka memanggil fungsi untuk mengirim paket *request* baru.

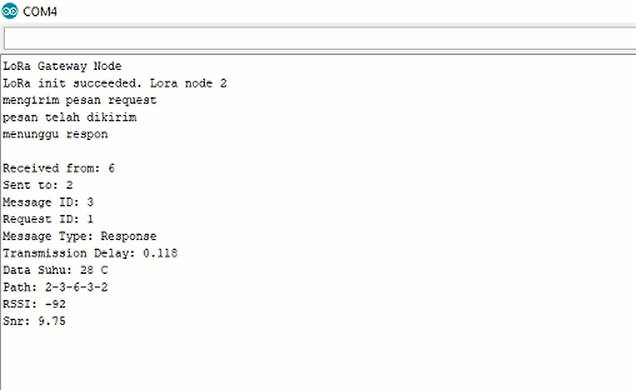
# PENGUJIAN DAN PEMBAHASAN

Bab ini membahas tentang pengujian dan pembahasan yang diperoleh dari sistem komunikasi *Multi-Hop* pada *Wireless Sensor Network* berbasis modul komunikasi *LoRa* menggunakan metode *Controlled* *Flooding* yang dibangun dalam bentuk program.

## Pengujian Metode *Controlled Flooding*

### Pengujian Komunikasi *Multi-hop* pada LoRa

Pengujian komunikasi *multi-hop*  pada LoRa dilakukan untuk menguji fungsionalitas dari sistem komunikasi *multi-hop* pada WSN berbasis modul komunikasi LoRa. Pengujian ini dilakukan untuk mengetahui apakah komunikasi *multi-hop* pada LoRa dapat terjadi dengan memanfaatkan *node* perantara untuk meneruskan paket menuju tujuan. Mekanisme pengujian yang dilakukan untuk mengetahui apakah komunikasi *multi-hop* pada LoRa dapat bekerja sesuai yang diharapkan yaitu dengan melakukan percobaan pengiriman paket dari *gateway node* menuju *sensor node* yang mana melalui *node* perantara karena kedua *node* tersebut tidak saling menjangkau. Sebelum dilakukan percobaan, di setiap *node* akan ditambahkan baris kode untuk menambahkan informasi id *node* yang dilewati sehingga setiap *node* yang dilewati oleh paket akan memodifikasi informasi jalur pada paket tersebut dengan menambahkan informasi ID *node* tersebut.

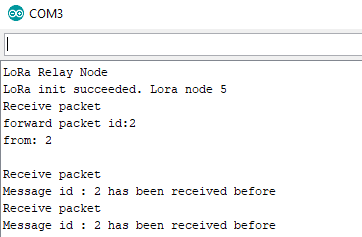


Gambar 5.1 Hasil Pengiriman Paket Menggunakan Komunikasi *Multi-Hop*

Tampilan hasil pengiriman paket menggunakan komunikasi *multi-hop* ditunjukkan pada Gambar 5.1 diatas, yang menunjukkan isi paket balasan berupa ID pesan, data *sensor* suhu, jalur yang dilewati, dll. Komunikasi *multi-hop* LoRa ditunjukkan dengan informasi jalur yang dilewati yaitu pengirim yang merupakan *node* 2 mengirimkan paket kepada penerima yang merupakan *node* 6 melewati *node* 3 yang terletak diantara dua *node* tersebut. Berdasarkan hasil yang ditunjukkan pada Gambar 5.1, menunjukkan bahwa komunikasi *multi-hop* dapat dilakukan menggunakan modul komunikasi LoRa dan memungkinkan untuk diterapkan pada WSN.

### Pengujian Mekanisme Pencegahan *Infinite Loop*

Pencegahan *infinite loop* pada penelitian ini merupakan mekanisme yang bertujuan untuk mencegah terjadinya pengiriman paket berulang kali yang menyebab masalah pada jaringan WSN. Pengujian ini dilakukan untuk mengetahui apakah mekanisme pencegahan *infinite loop* di setiap *node* pada WSN berbasis LoRa dapat terjadi dengan menerapkan mekanisme tersebut. Langkah pengujian mekanisme ini yaitu dengan menambahkan kode program pada setiap *node* untuk mencetak informasi ketika menerima paket dengan ID yang sama dengan yang ada pada tabel *record*. Kemudian akan dilakukan percobaan pengiriman paket untuk mendapatkan hasil pengujian. Hasil pengujian ini akan dilihat pada *relay node* melalui *serial monitor* Arduino IDE seperti yang ditunjukkan pada Gambar 5.2.

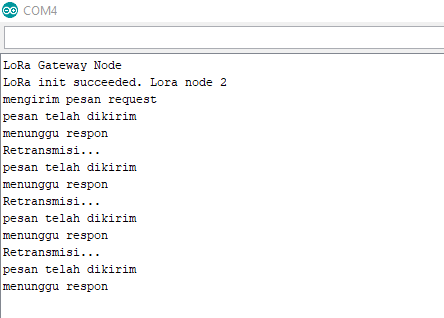


Gambar 5.2 Hasil Pengujian Mekanisme Pencegahan *Infinite Loop*

Hasil uji dari mekanisme pencegahan *infinite loop transmission* dapat dilihat pada Gambar 5.2, yang menunjukkan informasi terdapat tiga pesan dengan ID 2 yang pertama di-*forward* dan yang lain mencetak informasi bahwa paket sudah pernah diterima sebelumnya. Hasil tersebut menunjukkan bahwa pesan dengan ID 2 yang diterima pertama kali akan di-*forward* ke *node* tetangganya dan informasi ID pesan disimpan di tabel *record* sedangkan paket yang selanjutnya ID pesan yang sama akan dilakukan *drop packet* oleh *node* tersebut karena informasi paket tersebut sudah ada pada tabel *record*. Berdasarkan hasil tersebut dapat diketahui bahwa mekanisme ini dapat mencegah terjadinya *infinite loop transmission*.

### Pengujian Mekanisme *Retransmission Packet*

Implementasi sistem jaringan nirkabel menggunakan metode *flooding* sebagai metode komunikasi memungkinkan terjadinya *collision* yang dapat menyebabkan *packet loss*, untuk mengatasi kondisi tersebut pada penelitian ini menerapkan mekanisme *retransmission packet* agar paket dikirim kembali. Pengujian mekanisme *retransmission packet* ini bertujuan untuk mengetahui apakah mekanisme ini berhasil untuk mengatasi jika terjadi *packet loss* dengan mengirim paket kembali untuk mengganti paket yang tanpa balasan sebelumnya. Mekanisme pengujian ini yaitu dengan menyalakan perangkat *gateway node* dengan laptop untuk melihat hasil pengujian melalui *serial monitor* pada perangkat lunak Arduino Nano. Kemudian mematikan *sensor node* untuk menyebabkan terjadinya *packet loss* lalu dilakukan pengiriman paket oleh *gateway node* untuk mengetahui hasil dari pengujian seperti ini yang ditunjukkan pada Gambar 5.3.



Gambar 5.3 Hasil Pengujian Mekanisme *Retransmission Packet*

Tampilan hasil pengujian pada Gambar 5.3 terdapat informasi yang muncul empat kali yaitu yang pertama informasi bahwa user telah mengirim pesan *request* lalu muncul informasi bahwa dilakukan retransmisi sebanyak tiga kali yang mana muncul setelah tidak detik sekali. Hasil pengujian tersebut menunjukkan bahwa setelah dilakukan pengiriman paket, tidak ada paket *reply* yangditerima selama tiga detik sehingga dilakukan retransmisi sampai sebanyak tiga kali. Berdasarkan hasil tersebut dapat diketahui bahwa mekanisme *retransmission packet* berhasil dilakukan ketika paket *request* yang dkirimkan tidak mendapatkan respon dalam waktu tiga detik atau terjadi *packet loss* dengan retransmisi maksimal sebanyak tiga kali.

## Pengujian Kinerja

Subbab ini akan mengulas bagaimana kinerja dari sistem komunikasi *multi-hop* LoRa dengan menggunakan metode *controlled flooding*. Pengujian kinerja sistem tersebut terdiri dari tiga parameter yaitu *network latency*, *packet loss*, dan *troughput*. Percobaan yang dilakukan untuk menguji ketiga parameter tersebut yaitu sebanyak 50 kali percobaan transmisi dengan sistem komunikasi *multi-hop* LoRa menggunakan metode *controlled flooding*. Parameter uji dan hasil uji akan dijelaskan pada subbab berikut.

### *Round Trip Time*

*Network latency* dalam sebuah jaringan, mengukur waktu yang diperlukan beberapa data untuk mencapai tujuannya di seluruh jaringan. Biasanya diukur sebagai waktu *delay* yang dibutuhkan untuk informasi untuk sampai ke tujuannya dan kembali lagi. Latensi biasanya diukur dalam milidetik (milidetik).

### *Packet Loss*

### *Troughput*

## *Ds*

# Penutup

Bab ini membahas tentang kesimpulan dan saran yang didapatkan dari hasil penelitian sistem komunikasi *Multi-Hop* pada *Wireless Sensor Network* berbasis modul komunikasi *LoRa* menggunakan metode *Controlled* *Flooding*. Kesimpulan dan saran dapat digunakan sebagai referensi untuk penelitian berikutnya.

## Kesimpulan

Kesimpulan merupakan pernyataan-pernyataan yang singkat, jelas, dan tepat tentang hasil penelitian yang diperoleh berdasarkan tujuannya. Bagian ini merupakan penegasan dari yang telah dijelaskan pada bagian Pembahasan dan tidak memuat informasi yang baru. Bagian ini juga mencerminkan jawaban dari rumusan masalah (pertanyaan penelitian).

## Saran

Saran berisi pernyataan-pernyataan yang ringkas dan jelas tentang masalah-masalah atau hal-hal yang dapat dilakukan untuk mengembangkan penelitian ini lebih lanjut. Saran itu dapat diarahkan pada aspek metode, instrumen, populasi/sampel, dan sebagainya.

DAFTAR REFERENSI

Arduino, 2018. *Arduino.* [Online]   
Available at: http://www.arduino.cc  
[Diakses 3 April 2019].

Augustin, A., Yi, J., Clausen, T. & Townsley, W. M., 2016. A Study of LoRa: Long Range & Low Power Networks for the Internet of Things. *Sensor,* 16(9).

Bondu, J., Mishra, A., Laxmi, V. & Gaur, M. S., 2017. *Flooding in Secure Wireless Sensor Network.* Jaipur, IN, India, Association for Computing Machinery.

Dias, J. & Grilo, A., 2018. *LoRaWAN multi-hop uplink extension.* Lisboa, Portugal, Instituto Superior Técnico Universidade de Lisboa.

Flammini, A. & Sisinni, E., 2014. *Wireless Sensor Networking in the Internet of Things and Cloud.* Brescia, Elsevier Ltd.

Lundell, D., Hedberg, A., Nyberg, C. & Fitzgerald, E., 2018. *A Routing Protocol for LoRa Mesh Networks.* Lund, 2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks".

Munoz, J. et al., 2015. *A flooding routing algorithm for a wireless sensor network for seismic events.* Xalapa, Mexico, Institute of Electrical and Electronics Engineers.

Prana, T. A., Ichsan, M. H. H. & Setyawan, G. E., 2019. Monitoring Daya Menggunakan Algoritma Shortest Job First Pada Wireless Sensor Network. *J-PTIIK UB,* 3(1), pp. 593-602.

Ruano, E., 2016. *LoRa protocol. Evaluation, limitation, and practical test,* Barcelona: Universitat Politecnica De Catalunya BarcelonaTech.

Sohraby, K., Minoli, D. & Znati, T., 2007. *Wireless Sensor Network Technology, Protocols, and Application.* Canada: John Wiley & Sons, Inc.

Susanto, A. R., Bhawiyuga, A. & Amron, K., 2019. Implementasi Sistem Gateway Discovery pada Wireless Sensor Network (WSN) Berbasis Modul Komunikasi LoRa. *J-PTIIK UB,* 3(2), pp. 2138-2145.

Velde, B. V. d., 2017. Multi-hop LoraWAN: including a forwarding node.

Wazir, J., Barukab, O., Almagrabi, A. O. & Khan, S. A., 2016. MINIMIZING DENIAL OF SERVICE ATTACK FOR MULTIPLE BASE STATIONS IN WIRELESS SENSOR NETWORK. *VFAST Transactions on Software Engineering,* 11(2), pp. pp. 01-14.

Wixted, A. J. et al., 2016. Evaluation of LoRa and LoRaWAN for Wireless Sensor Networks. *Orlando: IEEE SENSOR,* pp. 1-3.