

**IMPLEMENTASI PROTOKOL *ROUTING DIRECTED DIFFUSION*
PADA *WIRELESS SENSOR NETWORK* MENGGUNAKAN
MEDIA KOMUNIKASI RF**

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Elsandio Bramudya Putra Fathoni

NIM: 135150301111081



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

PENGESAHAN

IMPLEMENTASI PROTOKOL ROUTING DIRECTED DIFFUSION PADA WIRELESS
SENSOR NETWORK MENGGUNAKAN MEDIA KOMUNIKASI RF

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Elsandio Bramudya P.F
NIM: 135150301111081

Skripsi ini telah diuji dan dinyatakan lulus pada
1 Agustus 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Sabriansyah Rizqika Akbar, S.T., M.Eng.
NIP: 198208092012121004

Dosen Pembimbing II



Mochammad Hannats Hanafi Ichsan, S.ST., M.T.
NIK: 2014058812291001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T., M.T., Ph.D.
NIP: 197105182003121001

A

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, 9 Agustus 2018



Elsandio Bramudya P.F.

NIM: 135150301111081

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan penulisan skripsi yang berjudul “Implementasi Protokol *Routing Directed Diffusion* Pada *Wireless Sensor Network* Menggunakan Media Komunikasi RF”.

Pada penulisan dan penyusunan skripsi ini tidak terlepas dari bantuan, bimbingan, dukungan, serta doa dari berbagai pihak. Dengan kerendahan hati penulis ingin mengucapkan banyak terima kasih kepada:

1. Kedua orangtua penulis, yaitu Ayahanda Khurul Fathoni dan Ibunda Eni Kusmayana serta seluruh keluarga besar atas segala bimbingan, nasihat, kasih sayang serta dukungan kepada penulis selama menempuh pendidikan di Fakultas Ilmu Komputer Universitas Brawijaya, serta senantiasa tiada hentinya memberikan doa demi terselesaikannya skripsi ini.
2. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng. dan Mochammad Hannats Hanafi Ichsan, S.ST, M.T. selaku dosen pembimbing skripsi yang telah membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini dengan baik.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D., selaku Ketua Jurusan Teknik Informatika.
4. Segenap dosen Fakultas Ilmu Komputer Universitas Brawijaya pada umumnya dan dosen Teknik Komputer Universitas Brawijaya pada khususnya, atas ilmu dan nasehat yang telah diberikan selama penulis menempuh pendidikan di Universitas Brawijaya.
5. Seluruh Civitas Akademik yang telah memberikan bantuan dan dukungan selama penulis menempuh studi di Teknik Informatika keminatan Teknik Komputer Universitas Brawijaya.
6. Seluruh keluarga besar mahasiswa Teknik Informatika keminatan Teknik Komputer Universitas Brawijaya atas kerjasama, bantuan dan semangat yang diberikan.
7. Teman-teman Himawibu : Tatit Kisyaprakasa, Reza Ridlo N., Rizki Septiansyah, Ade Ronialta, Noor Ilmi, Adiet Ilmawan, Tegar A.N., dan Alrynto yang telah memberikan dukungan dan kerjasama.
8. Teman-teman kos Panjaitan 180 : Master Prayogi, Abdul Aziz, Rizky, Irfan Zain, Ade Kurniawan, Sam Bayu, Faisal Alfahd dan M. Faqih Anshori yang telah memberikan doa dan dukungan.

Penulis menyadari bahwa pada penulisan dan penyusunan skripsi ini masih jauh dari kesempurnaan, sehingga saran dan kritik yang bersifat membangun sangat diharapkan demi kesempurnaan dalam penulisan selanjutnya. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 9 Agustus 2018

Penulis

bramudya.elsandio@gmail.com



ABSTRAK

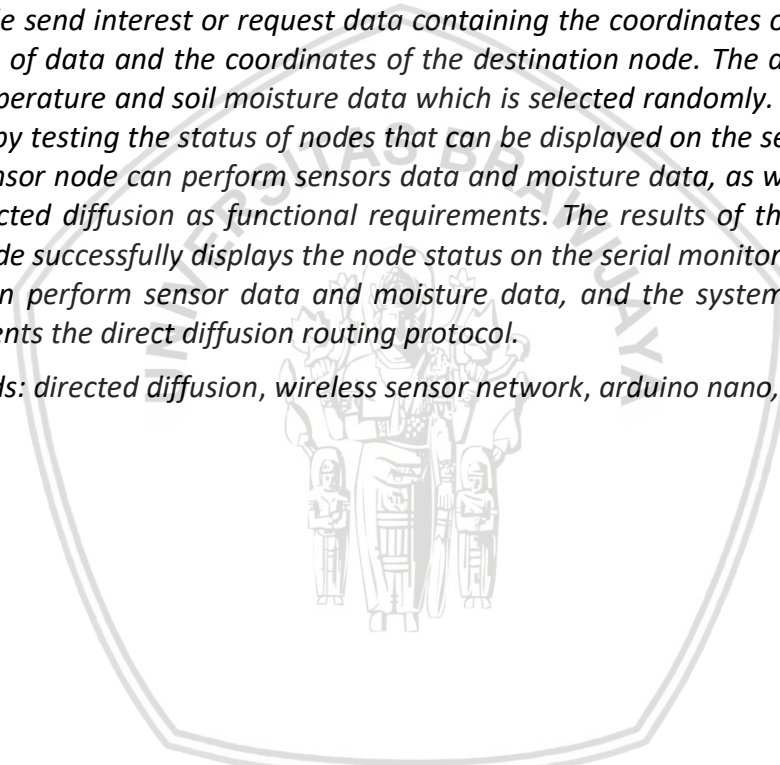
Implementasi protokol *routing* pada *wireless sensor network* telah dilakukan khususnya pada arsitektur topologi jaringan *flat*. Dimana pada topologi jaringan *flat* terdapat beberapa protokol *routing* yang dapat diimplementasikan salah satunya adalah protokol *routing directed diffusion*. Sayangnya, implementasi tersebut dilakukan menggunakan aplikasi simulasi jaringan. Oleh karena itu, pada penelitian ini penulis mengimplementasikan protokol *routing directed diffusion* pada *wireless sensor network* pada sebuah *real device* yaitu menggunakan media komunikasi RF. Perangkat keras untuk membangun sebuah *node* menggunakan mikrokontroler arduino nano 3.0 sebagai pemroses data sensor dan modul NRF24L01 sebagai media transmisi data. Pada implementasi *directed diffusion*, *sink node* mengirimkan *interest* atau permintaan data yang berisi koordinat pengirim, jenis data yang diminta dan koordinat *node* tujuan. Data yang digunakan ialah data suhu dan kelembaban tanah yang ditentukan secara acak karena *node* sensor tidak menggunakan modul sensor. Pengujian dilakukan dengan menguji status *node* dapat ditampilkan pada serial monitor, setiap *node* sensor dapat melakukan pemrosesan data sensor suhu dan kelembaban tanah, serta pengujian implementasi *directed diffusion* yang berupa pengujian fungsional. Hasil dari pengujian, setiap *node* berhasil menampilkan status *node* pada serial monitor, setiap *node* sensor dapat melakukan pemrosesan data suhu dan kelembaban tanah, serta sistem berhasil mengimplementasikan protokol *routing directed diffusion*.

Kata kunci: *directed diffusion, wireless sensor network, arduino nano, nrf24l01*.

ABSTRACT

Implementation of wireless sensor network routing protocols has been carried out specifically on flat topology architectures. Where in the flat topology there are several routing protocols that has been implemented, one of which is directed diffusion routing protocol. Unfortunately, the implementation is done using a network simulation application. Therefore, in this research, the authors implemented routing protocols directed diffusion on a wireless sensor network on a real device that uses RF communication media. Hardwares to build a node using arduino nano 3.0 microcontroller as sensor data processor and NRF24L01 module as data transmission medium. At the time of application of directed diffusion, the sink node send interest or request data containing the coordinates of the sender, the type of data and the coordinates of the destination node. The data used are the temperature and soil moisture data which is selected randomly. The research is done by testing the status of nodes that can be displayed on the serial monitor, each sensor node can perform sensors data and moisture data, as well as testing the directed diffusion as functional requirements. The results of the research is each node successfully displays the node status on the serial monitor, each sensor node can perform sensor data and moisture data, and the system successfully implements the direct diffusion routing protocol.

Keywords: directed diffusion, wireless sensor network, arduino nano, nrf24l01.



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
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
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Tinjauan Pustaka	6
2.2 Dasar Teori.....	6
2.2.1 <i>Wireless Sensor Network (WSN)</i>	7
2.2.2 <i>Directed Diffusion</i>	8
2.2.3 <i>Arduino Nano v3.0</i>	8
2.2.4 <i>Modul NRF24L01</i>	9
BAB 3 METODOLOGI	11
3.1 Studi Literatur	11
3.2 Rekayasa Kebutuhan.....	12
3.2.1 Rekayasa Kebutuhan Perangkat Keras.....	12
3.2.2 Rekayasa Kebutuhan Perangkat Lunak	12
3.3 Perancangan	13
3.3.1 Perancangan Sistem.....	13
3.3.2 Perancangan Topologi <i>Node</i>	14

3.4 Implementasi	14
3.5 Pengujian Sistem.....	14
3.6 Analisis Pengujian	14
3.7 Penarikan Kesimpulan Dan Saran	15
BAB 4 REKAYASA KEBUTUHAN.....	16
4.1 Pendahuluan	16
4.1.1 Tujuan	16
4.1.2 Kegunaan.....	16
4.1.3 Karakteristik Pengguna	16
4.2 Kebutuhan Sistem.....	16
4.2.1 Kebutuhan Fungsional	16
4.2.2 Kebutuhan Non-Fungsional	17
4.2.3 Kebutuhan Perangkat Keras.....	18
4.2.4 Kebutuhan Perangkat Lunak.....	18
BAB 5 PERANCANGAN DAN IMPLEMENTASI	19
5.1 Perancangan Sistem.....	19
5.1.1 Perancangan Perangkat Keras	19
5.1.2 Perancangan Perangkat Lunak.....	20
5.1.2.1 Perancangan Format Perintah.....	20
5.1.2.2 Perancangan Format Data.....	21
5.1.2.3 Diagram Alir <i>Sink Node</i>	22
5.1.2.4. Diagram Alir <i>Node</i> Sensor Tetangga Dari <i>Sink Node</i>	24
5.1.2.5. Diagram Alir <i>Node</i> Sensor Yang Bukan Tetangga Dari <i>Sink Node</i> ..	25
5.1.3 Perancangan Topologi.....	26
5.2 Implementasi	26
5.2.1 Batasan Implementasi.....	27
5.2.2 Spesifikasi Sistem	27
5.2.3 Implementasi Perangkat Keras	27
5.2.4 Implementasi Perangkat Lunak.....	28
5.2.4.1 Implementasi Kode Program <i>Sink Node</i>	28
5.2.4.1 Implementasi Kode Program <i>Node</i> Tetangga <i>Sink Node</i>	30
5.2.4.1 Implementasi Kode Program <i>Node</i> Bukan Tetangga <i>Sink Node</i>	32

BAB 6 PENGUJIAN DAN ANALISIS.....	34
6.1 Pengujian Status <i>Node</i> Pada Serial Monitor.....	34
6.1.1 Tujuan Pengujian.....	34
6.1.2 Prosedur Pengujian	34
6.1.3 Hasil Dan Analisis Pengujian	34
6.2 Pengujian Sensor <i>Node</i> Menampilkan Data Sensor	36
6.2.1 Tujuan Pengujian.....	36
6.2.2 Prosedur Pengujian	36
6.2.3 Hasil Dan Analisis Pengujian	36
6.3 Pengujian Fungsional	38
6.3.1 Tujuan Pengujian.....	38
6.3.2 Prosedur Pengujian.....	39
6.3.3 Hasil Dan Analisis Pengujian	39
6.3.3.1 Hasil Dan Analisis Pengujian Pengiriman Data Sensor Suhu	40
6.3.3.2 Hasil Dan Analisis Pengujian Pengiriman Data Sensor Kelembaban Tanah	59
BAB 7 PENUTUP	80
7.1 Kesimpulan	80
7.2 Saran	81
DAFTAR PUSTAKA.....	82
LAMPIRAN	83

DAFTAR TABEL

Tabel 2.1 Tinjauan Pustaka Untuk Penelitian	6
Tabel 2.2 Spesifikasi Arduino Nano 3.0.....	9
Tabel 2.3 Spesifikasi Modul nRF24L01	10
Tabel 4.1 Tabel Kebutuhan Perangkat Keras	18
Tabel 4.2 Tabel Kebutuhan Perangkat Lunak.....	18
Tabel 5.1 Tabel Skema <i>Node</i>	19
Tabel 5.2 Tabel Fungsi Setup <i>Sink Node</i>	28
Tabel 5.3 Tabel Fungsi Pengiriman <i>Interest</i>	28
Tabel 5.4 Tabel Fungsi Penerimaan Data Sensor	29
Tabel 5.5 Tabel Fungsi Pengiriman Data Sensor	30
Tabel 5.6 Tabel Fungsi Pengiriman Ulang <i>Interest</i>	31
Tabel 5.7 Tabel Fungsi <i>Directed Diffusion</i>	32
Tabel 6.1 Tabel Pengujian Status <i>Node</i> Pada Serial Monitor.....	36
Tabel 6.2 Tabel Pengujian Data Sensor Serial Monitor.....	38
Tabel 6.3 Tabel Pengiriman Data Sensor Suhu Oleh <i>Node</i> (1,3).....	40
Tabel 6.4 Tabel Pengiriman Data Sensor Suhu Oleh <i>Node</i> (1,7).....	43
Tabel 6.5 Tabel Pengiriman Data Sensor Suhu Oleh <i>Node</i> (4,5).....	46
Tabel 6.6 Tabel Pengiriman Data Sensor Suhu Oleh <i>Node</i> (7,3).....	49
Tabel 6.7 Tabel Pengiriman Data Sensor Suhu Oleh <i>Node</i> (7,7).....	52
Tabel 6.8 Tabel Pengiriman Data Sensor Suhu Oleh <i>Node</i> (4,1).....	55
Tabel 6.9 Tabel Hasil Pengiriman Dan Penerimaan <i>Interest</i>	59
Tabel 6.10 Tabel Hasil Pengiriman Dan Penerimaan Data Sensor Suhu.....	59
Tabel 6.11 Pengiriman Data Sensor Kelembaban Tanah Oleh <i>Node</i> (1,3)	60
Tabel 6.12 Pengiriman Data Sensor Kelembaban Tanah Oleh <i>Node</i> (1,7)	62
Tabel 6.13 Pengiriman Data Sensor Kelembaban Tanah Oleh <i>Node</i> (4,5)	65
Tabel 6.14 Pengiriman Data Sensor Kelembaban Tanah Oleh <i>Node</i> (7,3)	68
Tabel 6.15 Pengiriman Data Sensor Kelembaban Tanah Oleh <i>Node</i> (7,7)	71
Tabel 6.16 Pengiriman Data Sensor Kelembaban Tanah Oleh <i>Node</i> (4,1)	74
Tabel 6.17 Hasil Pengiriman Dan Penerimaan <i>Interest</i> Kelembaban Tanah	78
Tabel 6.18 Hasil Pengiriman Dan Penerimaan Data Sensor Kelembaban Tanah .	79

DAFTAR GAMBAR

Gambar 2.1 Arsitektur Topologi <i>Cluster</i>	7
Gambar 2.2 Arsitektur Topologi <i>Flat</i>	7
Gambar 2.3 Mekanisme Komunikasi <i>Directed Diffusion</i>	8
Gambar 2.4 Arduino Nano 3.0	9
Gambar 2.5 Modul nRF24L01	9
Gambar 3.1 Alur Penelitian	11
Gambar 3.2 Diagram Blok Sistem	13
Gambar 5.1 Skema Rancangan Rangkaian <i>Node</i>	19
Gambar 5.2 Format Perintah <i>Timeout</i> dan <i>Timecount</i>	20
Gambar 5.3 Format Perintah Permintaan Data Sensor	21
Gambar 5.4 Format Pengiriman <i>Interest</i>	21
Gambar 5.5 Format Pengiriman Data Sensor <i>Node</i> Tetangga <i>Sink</i>	22
Gambar 5.6 Format Pengiriman Data Sensor Bukan <i>Node</i> Tetangga <i>Sink</i>	22
Gambar 5.7 Diagram Alir <i>Sink Node</i> (1)	22
Gambar 5.8 Diagram Alir <i>Sink Node</i> (2)	23
Gambar 5.9 Diagram Alir <i>Node</i> Sensor tetangga <i>Sink Node</i>	24
Gambar 5.10 Diagram Alir <i>Node</i> Sensor Yang Bukan Tetangga <i>Sink Node</i>	25
Gambar 5.11 Perancangan Topologi.....	26
Gambar 5.12 Implementasi Perangkat Keras	27
Gambar 6.1 Status <i>Sink node</i> (4,9) dan Sensor <i>Node</i> (4,1)	34
Gambar 6.2 Status Sensor <i>Node</i> (1,3) dan Sensor <i>Node</i> (1,7)	35
Gambar 6.3 Status Sensor <i>Node</i> (4,5) dan Sensor <i>Node</i> (7,3)	35
Gambar 6.4 Status Sensor <i>Node</i> (7,7)	35
Gambar 6.5 Status Sensor <i>Node</i> (7,7) dan Sensor <i>Node</i> (4,1)	37
Gambar 6.6 Status Sensor <i>Node</i> (1,5) dan Sensor <i>Node</i> (1,7)	37
Gambar 6.7 Status Sensor <i>Node</i> (4,5) dan Sensor <i>Node</i> (7,3)	38

BAB 1 PENDAHULUAN

1.1 Latar belakang

Pada era ini berbagai teknologi di bidang *embedded* dan jaringan mengalami banyak perkembangan. Salah satunya ialah *Wireless Sensor Network* (WSN). WSN merupakan sebuah infrastruktur yang terdiri atas penginderaan, komputasi dan komunikasi yang memberikan administrator kemampuan untuk mengamati dan bereaksi terhadap peristiwa dan fenomena pada sebuah lingkungan yang spesifik (Sohraby, 2007). WSN dapat diimplementasikan di berbagai bidang diantaranya pertanian, peternakan, kesehatan, militer dan lain sebagainya.

Pada sebuah WSN terdapat 2 arsitektur topologi yaitu topologi *flat* dan topologi *cluster*. Topologi *flat* merupakan arsitektur WSN dimana semua *node* sensor yang ada di dalam jaringan memiliki kedudukan yang sama. Sedangkan pada topologi *cluster*, peran *node* sensor disusun secara hirarki. Pada sebuah topologi *flat* terdapat beberapa protokol *routing* pengiriman dan pengumpulan data diantaranya *Flooding*, *Sensor Protocol for Information via Negotiation* (SPIN), *Directed Diffusion* (DD), *Gradient Routing* dan *Rumour Routing*. *Routing* adalah proses pengiriman paket data dari *node* sumber ke *node* tujuan. Dalam WSN, *routing* secara umum bekerja pada beberapa *hop* untuk mencapai tujuan (Jain & Kan, 2014).

Directed Diffusion (DD) adalah sebuah mekanisme *routing* untuk pengumpulan data dimana konsumen (*sink*) mencari sumber data dengan mengirimkan paket ke tujuan dan menemukan rute terbaik untuk menerima data (Intanagonwiwat, 2003). DD merupakan pengembangan dari mekanisme pengiriman data *flooding* yang mana DD dapat mengurangi kemungkinan tabrakan data dan mengurangi jumlah transmisi data. DD dipilih sebagai objek penelitian karena merupakan protokol *routing* pengembangan dari *flooding* dan memiliki efisiensi waktu lebih baik daripada protokol *routing* SPIN dalam hal transmisi data.

Pengembangan dan analisis protokol *routing directed diffusion* sebelumnya telah banyak dilakukan. Berdasarkan penelitian yang telah dilakukan sebelumnya yang ditulis oleh Sahid Ridho pada tahun 2016, dimana protokol *routing directed diffusion* diimplementasikan pada WSN dengan berbasis *cluster*. Pada implementasinya menggunakan *J-sim* dan pemrograman java. Dimana hasil pada penelitian tersebut yaitu peningkatan *lifetime* sensor *node* pada seluruh jaringan *cluster*. Penelitian lainnya yang ditulis oleh Gourav Sharma pada tahun 2009 juga dilakukan dengan membandingkan dua protokol *routing* yaitu *flooding* dengan *directed diffusion*. Dimana pada implementasinya menggunakan simulator NS-2.

Kelemahan dari aplikasi simulator jaringan ialah selalu berjalan pada kondisi ideal. Berdasarkan masalah tersebut, penulis ingin mengimplementasikan protokol *routing directed diffusion* pada sebuah *wireless sensor network* pada sebuah *real device* yang memanfaatkan mikrokontroler arduino nano dan menggunakan modul komunikasi NRF24L01.

Arduino Nano dipilih karena merupakan mikrokontroler yang cukup umum digunakan dalam pengembangan sistem *embedded* dan juga arduino nano memiliki dimensi board yang cukup kecil. NRF24L01 dipilih sebagai media transmisi data karena pada penelitian yang telah dilakukan sebelumnya oleh Upik Jamil pada tahun 2018 dimana dilakukan perbandingan performa antara NRF24L01, Xbee dengan ESP8266. Hasilnya NRF24L01 memiliki keunggulan dalam penerimaan data dan memori yang terpakai untuk program modul NRF24L01 hanya 32% dari kapasitas memori mikrokontroler arduino uno. Ada pula keuntungan lainnya yaitu lebar *buffer* FIFO dari modul NRF24L01 yang sebesar 32 Bytes sudah cukup untuk menampung pengiriman dan penerimaan data.

Pada implementasinya, sistem nantinya akan diuji dengan menerapkan protokol *routing directed diffusion* dengan melakukan transmisi data suhu dan kelembaban tanah. Nilai data suhu dan kelembaban tanah dituliskan langsung pada program dan ditentukan secara acak tanpa menggunakan modul sensor.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dikemukakan maka rumusan masalah yang dapat ditarik antara lain :

1. Bagaimana merancang sistem yang mengimplementasikan protokol *routing directed diffusion* pada jaringan *wireless sensor network* yang menggunakan komunikasi RF ?
2. Bagaimana implementasi hasil perancangan dari sistem yang mengimplementasikan protokol *routing directed diffusion* pada jaringan *wireless sensor network* yang menggunakan komunikasi RF ?
3. Bagaimana hasil implementasi protokol *routing directed diffusion* pada jaringan *wireless sensor network* yang menggunakan komunikasi RF ?

1.3 Tujuan

Tujuan dari penelitian ini antara lain :

1. Merancang sistem yang mengimplementasikan protokol *routing directed diffusion* pada jaringan *wireless sensor network* yang menggunakan komunikasi RF.
2. Mengimplementasikan hasil perancangan sistem untuk protokol *routing directed diffusion* pada jaringan *wireless sensor network* yang menggunakan komunikasi RF.
3. Menganalisis hasil dari implementasikan protokol *routing directed diffusion* pada jaringan *wireless sensor network* yang menggunakan komunikasi RF.

1.4 Manfaat

Manfaat yang diperoleh dari penelitian ini adalah dapat memberikan gambaran bagaimana implementasi protokol *routing directed diffusion* pada sebuah *wireless sensor network* yang memanfaatkan komunikasi RF.

1.5 Batasan masalah

Agar pembahasan dalam penelitian ini dapat dilakukan secara terarah sehingga mendapatkan hasil sesuai tujuan penelitian, maka perlu diterapkannya batasan penelitian atau batasan masalah. Terdapat beberapa batasan masalah yang dapat dirumuskan, antara lain :

1. Implementasi sistem memanfaatkan komunikasi nirkabel memanfaatkan modul komunikasi NRF24L01.
2. *Node* sensor di dalam sistem menggunakan mikrokontroler Arduino Nano 3.0.
3. Data yang menjadi sumber data sensor ditentukan secara acak tanpa menggunakan modul sensor.
4. Jumlah *node* sensor sebanyak 7 *node* terdiri dari 1 *sink node* dan 6 *node* sensor.
5. Seluruh *node* mendapatkan sumber daya dari *USB Hub* sehingga jarak antar *node* sangat berdekatan.
6. Koordinat *node* hanya digunakan sebagai penamaan ID *node* bukan sebagai peletakan posisi *node*.

1.6 Sistematika pembahasan

Penjelasan singkat mengenai isi dari masing - masing bab adalah sebagai berikut :

BAB 1 PENDAHULUAN

Menjelaskan latar belakang dari penelitian sehingga memperoleh judul "*Implementasi Protokol Routing Directed Diffusion Pada Wireless Sensor Network Menggunakan Media Komunikasi RF*", rumusan masalah, tujuan dari penelitian, manfaat yang dapat diperoleh dari penelitian, batasan masalah pada penelitian dan sistematika pembahasan dari penelitian.

BAB 2 LANDASAN KEPUSTAKAAN

Menjelaskan tentang tinjauan pustaka dan dasar teori yang dimanfaatkan sebagai rujukan dalam penelitian. Tinjauan pustakan dilakukan dengan merujuk pada penelitian terdahulu yang digunakan sebagai pembeda pada penelitian saat ini. Setelah sub bab tinjauan pustaka terdapat sub bab dasar teori yang berisi dasar-dasar teori yang menjadi landasan penelitian. Dasar teori yang digunakan antara lain tentang *wireless sensor network*, teori protokol *routing directed diffusion*, mikrokontroler arduino nano dan modul komunikasi NRF24L01.

BAB 3 METODOLOGI

Menjelaskan tentang langkah kerja atau metode yang digunakan dalam menyelesaikan masalah pada penelitian diantaranya studi

literatur, rekayasa kebutuhan sistem yang meliputi rekayasa kebutuhan perangkat keras dan lunak, perancangan sistem yang meliputi perancangan *node* dan perancangan topologi *node*, implementasi, pengujian sistem dimana dijelaskan secara umum poin-poin pengujian, analisis pengujian hingga penarikan kesimpulan yang berisi hasil dari penelitian dan saran yang berisi pertimbangan untuk penelitian selanjutnya.

BAB 4 REKAYASA KEBUTUHAN

Menjelaskan tentang kebutuhan – kebutuhan pada sistem. Terdapat beberapa sub bab diantaranya pendahuluan yang berisi tujuan dari sistem, kegunaan, dan karakteristik dari pengguna sistem. Kemudian terdapat sub bab kebutuhan yang berisi kebutuhan fungsional yaitu kebutuhan yang menjadi syarat sistem dapat dikatakan sesuai dengan tujuan implementasi sistem, kebutuhan non-fungsional yaitu proses tambahan yang tidak menjadi syarat utama sistem. Kemudian terdapat juga kebutuhan perangkat keras baik mikrokontroler dan modul komunikasi NRF24L01 yang digunakan untuk implementasi sistem, kebutuhan perangkat lunak yang berisi arduino IDE dan *library* RF24. Kemudian terdapat sub bab diagram blok yang berisi gambaran sistem dari perangkat dan modul yang saling terhubung serta alur kerja sistem dimana dijelaskan secara rinci proses kerja sistem mulai dari *input*, proses hingga *output* yang dihasilkan sistem.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

Berisi tentang perancangan sistem yang meliputi perancangan perangkat keras dimana rancangan skema *node* sensor dibuat, perancangan perangkat lunak dimana penulis akan membuat sebuah diagram alir sebagai acuan dalam pembuatan kode program arduino nano dan perancangan topologi dimana penulis akan menggambarkan topologi *node* yang memanfaatkan topologi *mesh*. Kemudian pada topologinya, letak *node* ditentukan berdasarkan posisi koordinatnya. Lalu ditambahkan dengan penjelasan mengenai implementasi perangkat keras yaitu memulai untuk membangun sistem berdasarkan kebutuhan perangkat keras, lalu implementasi perangkat lunak dimana penulis mulai untuk menuliskan kode program pada arduino IDE.

BAB 6 PENGUJIAN DAN ANALISIS

Berisi tentang metode pengujian dimana sistem akan diuji berdasarkan metode pengujian dan pengujian fungsional maupun non-fungsional. Kemudian dilakukan analisis berdasarkan hasil pengujian yang telah dilakukan.

BAB 7 PENUTUP

Berisi penarikan kesimpulan dan saran didapatkan dari hasil pengujian sistem yang telah dilakukan setelah melakukan proses perancangan dan implementasi. Isi dari kesimpulan diharapkan menjadi acuan dalam penelitian selanjutnya dan dapat dilakukan pengembangan dari sistem yang telah diimplementasikan. Selain itu pada akhir penulisan terdapat saran yang bertujuan untuk memberikan pertimbangan, perbaikan dan saran pengembangan pada penelitian selanjutnya.



BAB 2 LANDASAN KEPUSTAKAAN

Bab ini membahas tentang kajian pustaka yang digunakan sebagai acuan dalam penelitian ini. Beberapa teori yang dibahas menjadi pedoman dalam perancangan penelitian ini. Beberapa sub bab yang akan dibahas adalah tinjauan pustaka yang merupakan kajian dari penelitian sebelum nya milik orang lain dan dasar teori yang meliputi teori-teori tentang perangkat keras dan perangkat lunak yang akan digunakan.

2.1 Tinjauan Pustaka

Pada sub bab ini akan berisi tentang perbandingan penelitian yang pernah dilakukan sebelumnya. Berikut merupakan tinjauan pustaka yang digunakan yang akan dijelaskan pada Tabel 2.1.

Tabel 2.1 Tinjauan Pustaka Untuk Penelitian

No	Nama Penulis, Tahun, Judul	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1.	Gaurav Sharma, Suman Bala & Verman (2009). Comparison of Flooding and <i>Directed diffusion</i> for <i>Wireless Sensor Network</i>	Mengimplementasikan <i>directed diffusion</i> pada jaringan <i>wireless sensor network</i>	Implementasi <i>directed diffusion</i> dilakukan dengan simulator jaringan NS-2	Implementasi dilakukan dengan memanfaatkan mikrokontroler dan komunikasi RF
2.	Sahid Ridho, Indra Yasri (2016). Analisa Model <i>Directed diffusion</i> (DD) Berbasis Cluster Pada <i>Wireless sensor network</i> (WSN)	Mengimplementasikan <i>directed diffusion</i> pada jaringan <i>wireless sensor network</i>	Arsitektur jaringan yang digunakan adalah topologi <i>cluster</i>	Arsitektur jaringan yang digunakan adalah topologi <i>flat</i>

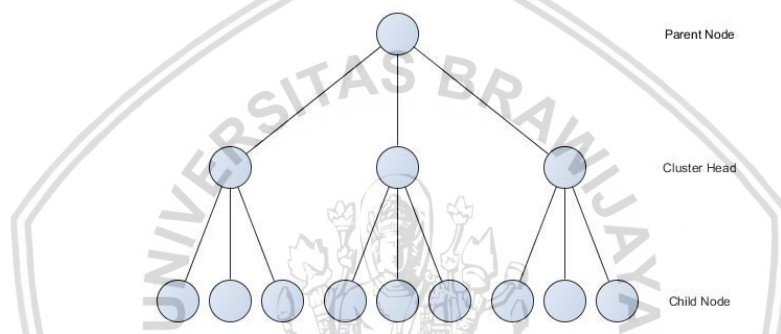
2.2 Dasar Teori

Berikut beberapa dasar teori yang digunakan untuk melandasi penelitian ini.

2.2.1 Wireless Sensor Network (WSN)

Wireless Sensor Network (WSN) merupakan sebuah infrastruktur yang terdiri atas penginderaan, komputasi dan komunikasi yang memberikan administrator kemampuan untuk mengamati dan bereaksi terhadap peristiwa dan fenomena pada sebuah lingkungan yang spesifik (Sohraby , 2007). WSN terdiri dari beberapa *node* dan *sensor node* yang saling bertukar informasi. *Sensor node* tidak hanya dapat berkomunikasi dengan *base station (BS)* saja, namun *sensor node* juga dapat memproses, menganalisis dan menyimpan data.

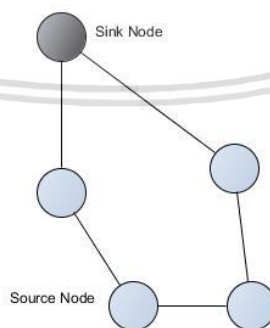
Mengenai *node sensor*, terdapat 2 macam arsitektur topologi yang memberikan peran tertentu pada sebuah *node* yaitu arsitektur topologi *cluster* dan topologi *flat*. Pada topologi *cluster*, *node-node* sensor tersusun dalam bentuk hirarki yang terdiri dari 3 macam *node* antara lain *child node*, *cluster head* dan *parent node*. Bentuk dari topologi *cluster* dapat dilihat pada Gambar 2.1.



Gambar 2.1. Arsitektur Topologi Cluster

Cluster head berfungsi sebagai koodinator dari *child node* dan berperan sebagai anggota di bawah *parent node*.

Sedangkan pada topologi *flat* hanya terdapat 2 macam *node* yaitu *sink node* dan *source node*.

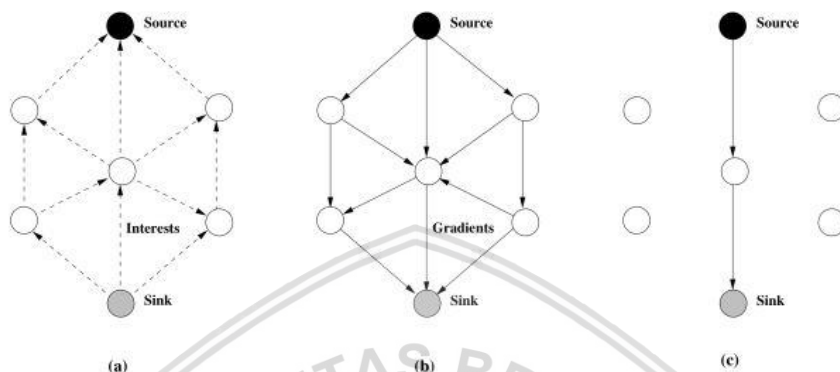


Gambar 2.2. Arsitektur Topologi Flat

Sink node berfungsi sebagai pengumpul data dari *source node* dan *source node* bekerja sebagai pengolah sumber data. Proses pertukaran data antar *node* melalui komunikasi nirkabel pada frekuensi yang telah ditentukan.

2.2.2 Directed Diffusion

Directed Diffusion (DD) adalah sebuah mekanisme *routing* untuk pengumpulan data dimana konsumen (*sink*) mencari sumber data dengan mengirimkan paket ke tujuan dan menemukan rute terbaik untuk menerima data (Intanagonwiwat dkk, 2003). Mekanisme komunikasi pada *routing* protokol DD dapat dilihat pada Gambar 2.1.



Gambar 2.3. Mekanisme Komunikasi *Directed Diffusion*

Pada Gambar 2.3. poin (a) digambarkan sebuah *sink node* menyebarluaskan tugas penginderaan dalam bentuk *interest* ke *node repeater* dan diteruskan hingga sampai ke *source node*. Pada poin (b) digambarkan setiap *node* membentuk sebuah *gradient* menuju ke *sink node*. Setelah itu pada poin (c) terbentuk jalur *reinforcements path* dari *source node* menuju ke *sink node* untuk mengirimkan data berdasarkan *gradient* yang telah terbentuk sebelumnya.

Terdapat beberapa elemen pada skema komunikasi *directed diffusion* antara lain :

1. *Interest* merupakan penamaan sebuah data. Data disebarluaskan oleh *sink node* kepada *node-node repeater* dalam tugas penginderaan.
2. *Sink node* merupakan *node* yang bertugas mengumpulkan data yang dikirimkan oleh *source node*.
3. *Gradient* adalah arah yang dibentuk oleh setiap *node* dalam setiap melakukan pengiriman dan penerimaan.
4. *Reinforcements path* adalah sebuah jalur yang dibentuk oleh *source node* untuk mengirimkan data.

2.2.3 Arduino Nano 3.0

Arduino Nano 3.0 merupakan bentuk pengembangan dari seri Nano sebelumnya. Arduino Nano memiliki dimensi yang cukup kecil dengan mikroprosesor Atmega328. Bentuk fisik Arduino Nano 3.0 dapat dilihat pada Gambar 2.4. Spesifikasi mikrokontroler Arduino Nano 3.0 dapat dilihat pada Tabel 2.2.



Gambar 2.4. Arduino Nano 3.0
(sumber : <https://www.arduino.cc/en/Main/ArduinoBoardNano>)

Tabel 2.2. Spesifikasi Arduino Nano 3.0

Chip mikrokontroler	Atmel Atmega328
Tegangan operasi	5 V
Tegangan <i>input</i>	7-12 V
Tegangan <i>input</i> (terbatas)	6 – 20 V
Pin I/O digital	14 (6 diantaranya menyediakan PWM)
Pin Analog (<i>input</i>)	8
Arus DC per pin I/O	40 mA
Memori <i>flash</i>	32 KB (2 KB telah digunakan sebagai bootloader)
SRAM	2 KB
EEPROM	1 KB
<i>Clock speed</i>	16 MHz
Dimensi	0.73" x 1.70"
Panjang	45 mm
Lebar	18 mm
Berat	5 g

2.2.4 Modul nRF24L01

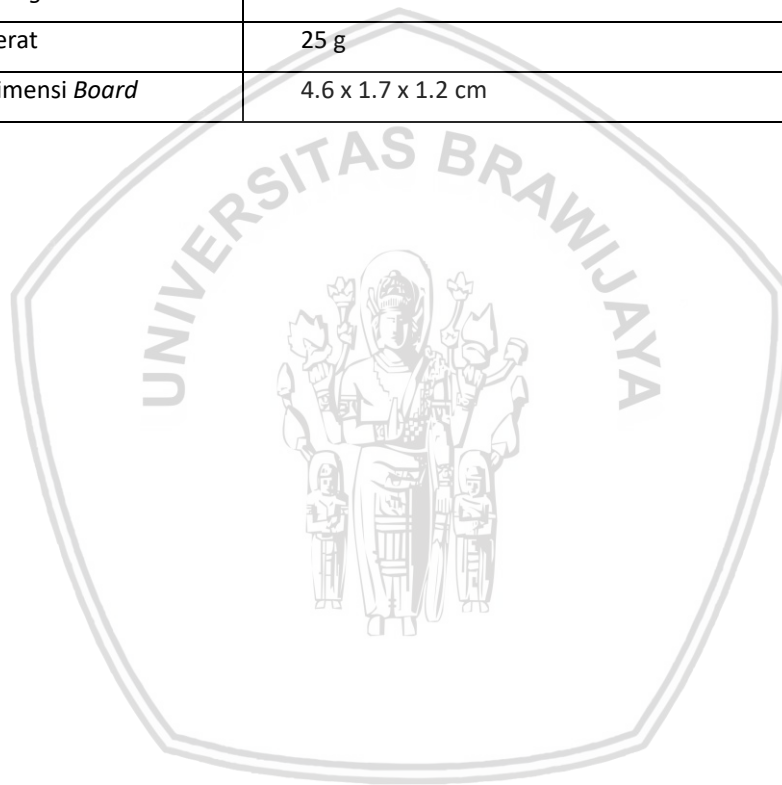
Modul *wireless* nRF24L01 adalah sebuah modul komunikasi radio jarak jauh yang memanfaatkan pita gelombang RF 2.4 GHz ISM (*Industrial, Scientific and Medical*). Modul ini menggunakan antarmuka SPI untuk berkomunikasi. Tegangan kerja dari modul ini adalah 3-3,6 Volt DC. Bentuk fisik dari modul nRF24L01 dapat dilihat pada Gambar 2.5. Spesifikasi mikrokontroler Arduino Nano 3.0 dapat dilihat pada Tabel 2.3.



Gambar 2.5. Modul NRF24L01
(Sumber : <http://www.robotshop.com/content/PDF/datasheet-wir020.pdf>)

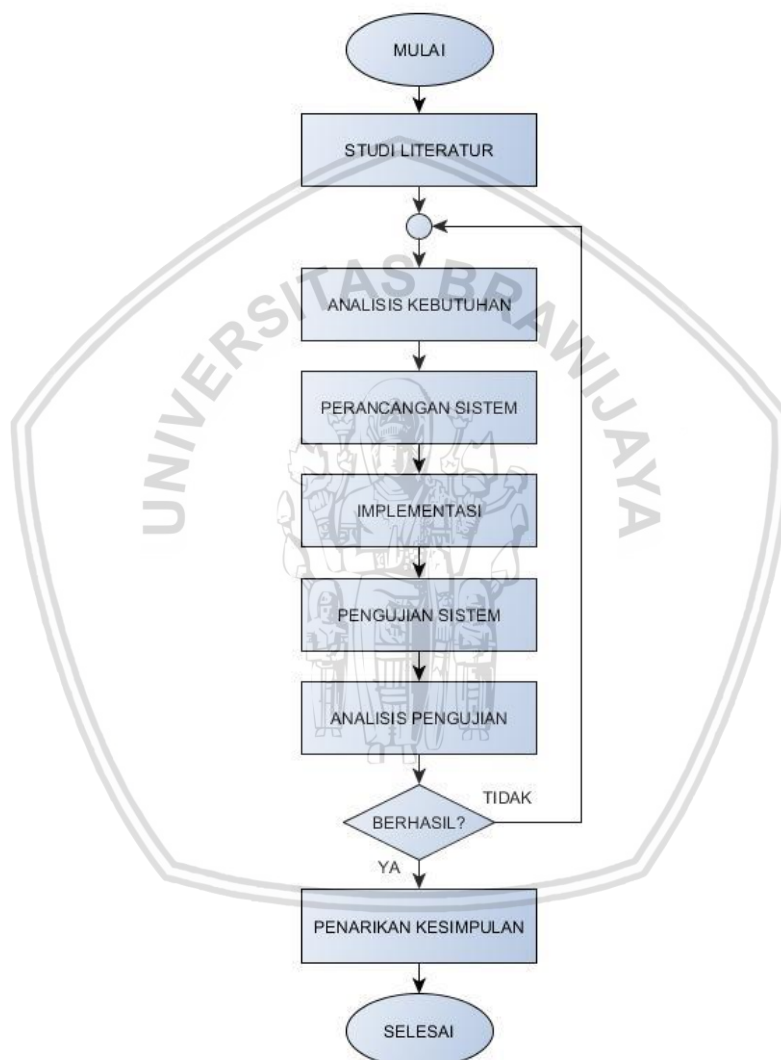
Tabel 2.3. Spesifikasi Modul nRF24L01

Tegangan	3-3.6 V
Maksimum <i>output power</i>	+20dBm
<i>Emission mode current (peak)</i>	115 mA
<i>Receive mode current (peak)</i>	45 mA
PA <i>gain</i>	20 Db
LNA <i>gain</i>	10 Db
Berat	25 g
Dimensi <i>Board</i>	4.6 x 1.7 x 1.2 cm



BAB 3 METODOLOGI

Penelitian ini dimulai dengan melakukan studi literatur yang terkait dengan landasan kepustakaan pada bab 2. Penelitian yang dilakukan penulis bersifat implementatif pembangunan karena mengimplementasikan *routing* protokol *directed diffusion* dengan cara membangun sistem jaringan wireless sensor *node*. Proses atau alur penelitian yang dilakukan untuk pembangunan sistem ini dapat dilihat pada diagram alir pada Gambar 3.1.



Gambar 3.1. Alur Penelitian

3.1 Studi Literatur

Studi literatur adalah tahap dimana penulis mencari, mengumpulkan dan melakukan penyusunan teori yang akan digunakan sebagai rujukan, landasan dan pendukung penelitian. Berikut adalah dasar teori yang digunakan penulis sebagai landasan dan pendukung penelitian.

1. Pengenalan *Wireless Sensor Network*

Studi literatur dilakukan pada jurnal, buku dan berbagai sumber yang kandungannya terkait dengan pembahasan *wireless sensor network* khususnya pada implementasinya.

2. Pengenalan Protokol *Routing Directed Diffusion*

Studi literatur dilakukan pada jurnal, buku dan berbagai sumber yang kandungannya terkait dengan pembahasan protokol *routing directed diffusion* baik dari mekanisme kerja dan implementasinya.

3. Arduino Nano 3.0

Studi literatur mengenai mikrokontroler arduino nano dibutuhkan sebagai refensi dalam penggunaannya baik dari spesifikasi, arduino IDE dan pemrograman arduino.

4. Modul NRF24L01

Studi literatur mengenai mikrokontroler modul NRF24L01 dibutuhkan sebagai refensi dalam penggunaannya baik dari spesifikasi dan *library* RF24 untuk menunjang penulisan program pada arduino IDE.

3.2 Rekayasa Kebutuhan

Analisis kebutuhan dilakukan dengan tujuan untuk menganalisis kebutuhan yang diperlukan oleh sistem agar sistem dapat berjalan sesuai tujuan. Dalam analisis kebutuhan terdapat beberapa sub bab analisis, diantaranya sebagai berikut.

3.2.1 Rekayasa Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras yaitu analisis kebutuhan perangkat keras yang terkait dengan kebutuhan dalam implemantasi sistem dalam penelitian ini. Adapun perangkat keras yang digunakan sebagai penunjang penelitian ini antara lain :

- a. Arduino Nano 3.0
- b. Modul *wireless* NRF24L01
- c. *Trainer board*
- d. Kabel *jumper*
- e. *USB Hub*

3.2.2 Rekayasa Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak yaitu analisis kebutuhan perangkat lunak yang terkait dengan kebutuhan dalam implemantasi sistem dalam penelitian ini. Adapun perangkat lunak yang digunakan sebagai penunjang penelitian ini antara lain :

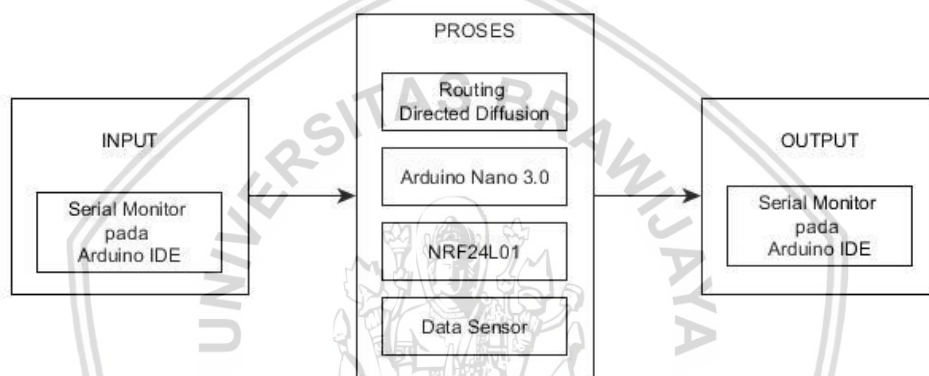
- a. Arduino IDE
- b. *Library* RF24

3.3 Perancangan

Perancangan merupakan tahap dimana dilakukan perancangan awal dimana desain sistem dibuat. Tahap-tahap perancangan akan dijelaskan pada sub bab di bawah ini.

3.3.1 Perancangan Sistem

Perancangan sistem merupakan tahap dimana peneliti merancang gambaran umum sistem. Gambaran umum sistem akan digambarkan pada diagram blok di bawah ini.



Gambar 3.2. Diagram Blok Sistem

Berikut di bawah ini adalah penjelasan dari Gambar 3.2.

- a. Pada blok *input* terdapat serial monitor dimana serial monitor digunakan oleh user untuk memasukkan perintah permintaan data pada *sink node*. User akan memasukkan perintah yang berisi permintaan jenis data dan koordinat *node* tujuan.
- b. Pada blok proses terdapat proses utama dari sistem yaitu *routing directed diffusion* dimana prosesnya yaitu setelah *sink node* mengirim pesan *interest* secara *broadcast*, maka *node* tetangga dari *sink node* yang menerima *interest* akan langsung melakukan *broadcast interest* ke *node* tetangga lainnya. Jika *node* tujuan menerima *interest* maka *node* sensor akan langsung membalas dengan mengirimkan data sensor yang diminta oleh *sink node*. Jika *node* sensor tujuan adalah tetangga dari *sink node*, maka data sensor akan langsung dikirim ke *sink node*. Namun jika *node* sensor tujuan bukan tetangga dari *sink node*, maka data sensor akan dikirim melalui jalur *node* sensor berdasarkan jumlah *hop* terpendek. Arduino nano sebagai pengolah data hasil penerimaan dan pengiriman data. Sedangkan NRF24L01 sebagai media transmisi data.

- c. Pada blok *output* yaitu proses ditampilkannya data sensor yang telah diterima oleh *sink node*.

3.3.2 Perancangan Topologi Node

Pada perancangan topologi, identitas *node* akan ditentukan berdasarkan titik koordinat garis (x,y). Titik koordinat hanya digunakan sebagai penamaan identitas *node* bukan sebagai posisi peletakan *node* saat diimplementasikan. Sehingga permintaan dan pengiriman data berdasarkan alamat dari titik koordinat masing-masing *node*.

3.4 Implementasi

Implementasi merupakan tahap penelitian yang dilakukan setelah proses perancangan. Alur implementasi dimulai dari pembangunan rangkaian *node* hingga implementasi topologi *node* akan dijelaskan pada poin-poin di bawah ini.

- a. Pembangunan rangkaian *node* dilakukan sesuai dengan rancangan.
- b. Pembuatan kode program dilakukan pada masing-masing *node* sehingga *node* dapat melakukan beberapa tugas yaitu mendapatkan data sensor, melakukan pengiriman data dan penerimaan data.
- c. Implementasi topologi *node* yaitu dengan memberikan setiap *node* sebuah identitas atau ID berdasarkan titik koordinat.

3.5 Pengujian Sistem

Pengujian sistem dilakukan untuk memenuhi kebutuhan sistem. Berikut adalah langkah-langkah pengujian sistem.

1. Pengujian setiap *node* dapat menampilkan status *node* pada serial monitor.
2. Setiap *node* sensor dapat menampilkan data suhu dan kelembaban tanah.
3. Pengujian sistem berdasarkan kebutuhan fungsional yaitu mengimplementasikan protokol *routing directed diffusion*.

3.6 Analisis Pengujian

Analisis pengujian didasarkan pada kebutuhan sistem. Sehingga analisis dilakukan dengan cara sebagai berikut.

1. Analisis bahwa setiap *node* dapat menampilkan status *node* pada serial monitor.
2. Analisis bahwa setiap *node* sensor dapat menampilkan data suhu dan kelembaban tanah.
3. Analisis bahwa sistem dapat memenuhi kebutuhan fungsional yaitu mengimplementasikan protokol *routing directed diffusion*.

3.7 Penarikan Kesimpulan dan Saran

Penarikan kesimpulan dan saran didapatkan dari hasil pengujian yang dilakukan setelah melakukan proses perancangan dan implementasi. Isi dari kesimpulan diharapkan menjadi acuan dalam penelitian selanjutnya dan dapat dilakukan pengembangan dari sistem yang telah diimplementasikan. Selain itu pada akhir penulisan terdapat saran yang bertujuan untuk memberikan pertimbangan, perbaikan dan saran pengembangan pada penelitian selanjutnya.



BAB 4 REKAYASA KEBUTUHAN

4.1 Pendahuluan

Pada bab rekayasa kebutuhan dijelaskan secara rinci terkait kebutuhan-kebutuhan sistem dalam mengimplementasikan protokol *routing directed diffusion* menggunakan komunikasi RF. Terdapat beberapa sub bab pada bab ini diantaranya pendahuluan yang meliputi tujuan, kegunaan dan karakteristik pengguna. Kemudian kebutuhan sistem yang meliputi kebutuhan fungsional, kebutuhan non-fungsional, kebutuhan antarmuka perangkat keras dan kebutuhan antarmuka perangkat lunak.

4.1.1 Tujuan

Tujuan utama dari rekayasa kebutuhan sistem ini yaitu agar dapat dibangun sebuah sistem yang mengimplementasikan protokol *routing directed diffusion* pada jaringan *wireless sensor network* yang menggunakan komunikasi RF yang memanfaatkan mikrokontroler arduino nano dan modul NRF24L01.

4.1.2 Kegunaan

Kegunaan rekayasa kebutuhan sistem ini adalah sebagai acuan peneliti dalam merancang dan mengimplementasikan hasil perancangan. Sehingga hasil yang telah dianalisis dapat sesuai dengan kebutuhan sistem yang telah dirumuskan.

4.1.3 Karakteristik Pengguna

Pengguna pada sistem berperan melakukan *input* permintaan pada *sink node* melalui serial monitor dan juga berperan melakukan pengawasan lalu lintas data yang terjadi melalui serial monitor.

4.2 Kebutuhan Sistem

Kebutuhan sistem menjelaskan tentang kebutuhan fungsional dan non-fungsional serta kebutuhan antarmuka perangkat keras dan lunak yang digunakan dalam perancangan sistem.

4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang wajib terpenuhi sehingga sistem dapat dikatakan berjalan maupun bekerja sesuai tujuan dimana sistem dapat mengimplementasikan protokol *routing directed diffusion* pada *wireless sensor network* yang memanfaatkan mikrokontroler arduino nano dan modul komunikasi NRF24L01. Berikut beberapa kebutuhan fungsional sistem yang wajib terpenuhi :

1. ***Sink node* dapat mengirimkan pesan *interest***

Pada proses ini *sink node* mengirimkan pesan *interest* yang berisi alamat *node* tujuan dan jenis data yang *diinput* oleh user melalui serial monitor dengan cara *broadcast* ke seluruh *node* tetangganya.

2. Node sensor tetangga sink node dapat menerima pesan *interest*

Pada proses ini *node* sensor tetangga sink node dapat menerima pesan *interest* dari *sink node* atau dari *node* sensor yang lain.

3. Node sensor tetangga sink node dapat membalas pesan *interest*

Pada proses ini *node* sensor akan langsung membalas pesan *interest* dari *sink node* secara langsung apabila *node* sensor tersebut merupakan tetangga dari *sink node*. Pesan balasan berupa data sensor.

4. Node sensor tetangga sink node dapat meneruskan pesan *interest*

Pada proses ini *node* sensor tetangga *sink node* meneruskan pesan *interest* dari *sink node* atau dari *node* sensor lain dengan cara *broadcast* kepada seluruh *node* tetangganya apabila dia bukan node tujuan yang diminta *sink node*.

5. Node sensor yang bukan tetangga dari sink node dapat menerima *interest*

Pada proses ini *node* sensor yang bukan tetangga dari *sink node* dapat menerima pesan *interest* dari *node* sensor yang lain.

6. Node sensor bukan tetangga dari sink node dapat membalas *interest*

Pada proses ini *node* sensor yang bukan tetangga dari *sink node* dapat membalas *interest* yang diterima dari *node* sensor lain. Pesan dibalas setelah *node* sensor penerima menerima maksimal 5 pesan atau dalam waktu sepuluh detik setelah pesan pertama diterima. Pada saat yang sama, *node* sensor akan memilih jalur balasan terbaik berdasarkan jumlah *hop* terkecil. Pesan balasan berisi data sensor.

7. Node sensor tetangga sink node dapat menerima pesan balasan

Pada proses ini *node* sensor tetangga *sink node* dapat menerima pesan balasan yang berisi data sensor.

8. Node sensor tetangga sink node dapat meneruskan pesan balasan

Pada proses ini *node* sensor tetangga *sink node* dapat meneruskan pesan balasan yang berisi data sensor ke *node* selanjutnya berdasarkan urutan pengiriman *interest* sebelumnya.

9. Sink node dapat menerima data sensor

Pada proses ini *sink node* dapat menerima pesan balasan yang berisi data sensor yang dibutuhkan.

4.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional dari sistem ini antara lain :

1. Pengguna dapat mengatur waktu *timeout* dan jumlah pesan *interest* yang akan dikirim ulang jika waktu *timeout* habis.

2. Lalu lintas data ditampilkan pada serial monitor.

4.2.3 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras merupakan perangkat keras yang digunakan sebagai komponen untuk membangun sistem. Kebutuhan antarmuka perangkat keras dari sistem ini akan dijelaskan pada tabel 4.1. di bawah ini.

Tabel 4.1. Tabel Kebutuhan Perangkat Keras

No.	Perangkat	Keterangan
1.	Arduino Nano 3.0	Merupakan komponen utama dari sebuah <i>node</i> yang berfungsi sebagai pemroses data.
2.	Modul NRF24L01	Merupakan komponen yang berfungsi sebagai media komunikasi antar <i>node</i> .
3.	Kabel Jumper	Merupakan komponen yang berfungsi untuk menghubungkan sensor, arduino nano dan modul NRF24L01.
4.	Trainer Board	Merupakan komponen yang berfungsi untuk media menghubungkan sensor, arduino nano dan modul NRF24L01.
5.	Kabel USB	Merupakan komponen yang berfungsi untuk menghubungkan arduino nano dengan <i>notebook</i> .

4.2.4 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak merupakan perangkat lunak yang digunakan sebagai komponen untuk menjalankan sistem. Kebutuhan antarmuka perangkat lunak dari sistem ini akan dijelaskan pada tabel 4.2. di bawah ini.

Tabel 4.2. Tabel Kebutuhan Perangkat Lunak

No.	Perangkat Lunak	Keterangan
1.	Arduino IDE	Merupakan perangkat lunak yang berfungsi untuk menulis, <i>compiling</i> , <i>debugging</i> dan <i>upload</i> program ke arduino nano.
2.	<i>Library</i> RF24	Merupakan <i>library</i> tambahan pada arduino IDE yang berfungsi untuk mempermudah dalam mengoperasikan fungsi-fungsi pada modul NRF24L01.

BAB 5 PERANCANGAN DAN IMPLEMENTASI

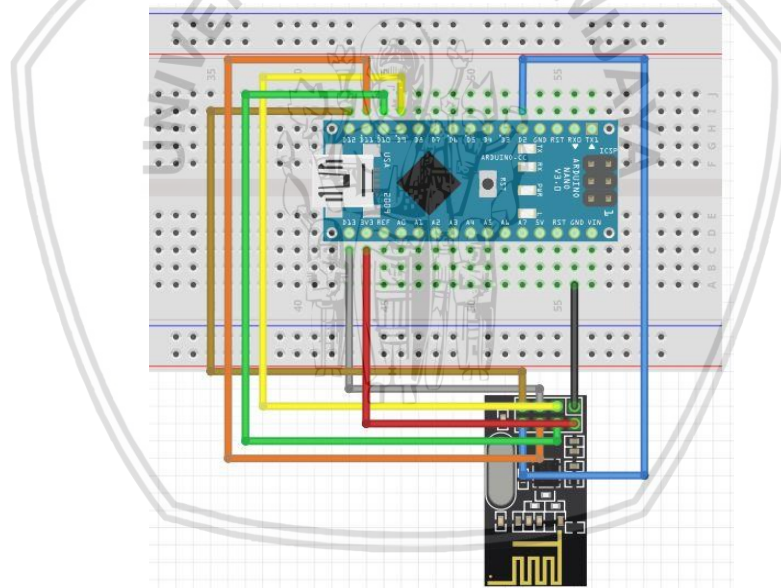
Pada tahap perancangan dan implementasi, pertama dilakukan perancangan sistem yang meliputi perancangan perangkat keras dan juga perangkat lunak. Kemudian akan dilakukan implementasi perangkat keras dan implementasi perangkat lunak.

5.1 Perancangan Sistem

Pada tahap perancangan berisikan mengenai gambaran rangkian instrumen yang digunakan dalam pembangunan sistem. Terdapat beberapa poin perancangan diantaranya sebagai berikut :

5.1.1 Perancangan Perangkat Keras

Perancangan perangkat keras *node* nantinya terdiri atas arduino nano 3.0 sebagai pemroses data dan modul NRF24L01 sebagai media komunikasi antar *node*. Skematik rangkaian *node* sensor digambarkan pada Gambar 5.1. di bawah ini.



Gambar 5.1. Skema Rancangan Rangkaian *Node*

Tabel 5.1 di bawah ini akan menjelaskan skema rangkaian *node* yang menggunakan arduino nano dengan modul NRF24L01 pada Gambar 5.1. di atas.

Tabel 5.1. Tabel Skema *Node*

Pin Arduino Nano	Warna Kabel	Pin NRF24L01
3v3	Merah	VCC
GND	Hitam	GND
D2	Biru	IRQ

D9	Kuning	CE
D10	Hijau	CSN
D11	Oranye	MOSI
D12	Cokelat	MISO
D13	Abu-abu	SCK

5.1.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak terdiri atas perancangan format perintah, format data, diagram alir *sink node*, diagram alir *node* sensor tetangga dari *sink node*, diagram alir *node* sensor yang bukan tetangga dari *sink node* dan diagram alir *directed diffusion*.

5.1.2.1 Format Perintah

Format perintah adalah proses merancang format perintah permintaan data oleh *sink node* yang dilakukan pada serial monitor. Terdapat 2 rancangan perintah yaitu perintah untuk menentukan *timeout* dan *timecount* kemudian perintah untuk mengirimkan permintaan data sensor. Perintah dilakukan dengan memasukkan kode-kode serial, daftar kode-kode serial tersebut antara lain :

1. STO = *Set Time Out*, yaitu kode batas waktu tunggu dalam sekali pengiriman permintaan data sensor.
2. STC = *Set Time Count*, yaitu kode jumlah perintah permintaan akan diulangi.
3. GDS = *Get Data Sensor*, yaitu kode perintah untuk melakukan permintaan data sensor.
4. NS = *Node Sensor*, yaitu kode ID *node* tujuan.

Rancangan perintah akan digambarkan pada Gambar 5.2. dan Gambar 5.3. di bawah ini.

- a. Format perintah *timeout* dan *timecount*

#	Kode Serial	#	Value	#
---	-------------	---	-------	---

Gambar 5.2. Format Perintah *Timeout* dan *Timecount*

Pada Gambar 5.2. di atas dapat dilihat bahwa setiap kode dipisahkan dengan karakter “#” untuk mempermudah pembacaan perintah. Kode Serial diisi dengan kode STO/STC sedangkan pada Value diisi dengan nilai rentang waktu *timeout* yang memiliki nilai maksimal 20000 milisekon atau diisi dengan jumlah perintah permintaan data sensor akan diulang dengan maksimal 10 kali.

b. Format perintah permintaan data sensor

#	Kode Serial	#	Tipe Request	#	Node Tujuan	#	ID Node	#
---	-------------	---	--------------	---	-------------	---	---------	---

Gambar 5.3. Format Perintah Permintaan Data Sensor

Pada Gambar 5.3. di atas dapat dilihat bahwa setiap kode juga dipisahkan dengan karakter “#” untuk mempermudah pembacaan perintah. Pada blok Kode Serial diisi dengan kode GDS, pada blok Tipe Request diisi dengan tipe request dimana tipe request dibagi menjadi 2 antara lain :

- Kode angka 1 sebagai permintaan data kelembaban tanah
- Kode angka 2 sebagai permintaan data suhu udara

Kemudian pada blok *Node Tujuan* diisi dengan kode NS dan pada blok *ID Node* diisi dengan ID *node* sensor tujuan.

5.1.2.2 Format Data

Format data adalah proses merancang format data sensor yang akan dikirim oleh *sink node* dan *node* tetangga dari *sink node*, kemudian data sensor yang dikirim oleh sensor *node*. Pada format data juga terdapat beberapa kode antara lain :

- S = *Sender*, yaitu pengirim permintaan.
- D = *Destination*, yaitu *node* tujuan.
- R = *Request*, yaitu tipe data sensor yang diminta.
- H = *Hop*, yaitu jumlah *hop*.
- C = *Content*, yaitu nilai data sensor.

Terdapat 3 jenis format data yang digunakan antara lain.

- Format pengiriman *interest*.

#	S	#	Susunan ID Pengirim	#	D	#	Node Tujuan	#	R	#	Tipe Request	#	H	#	Jumlah Hop	#
---	---	---	---------------------	---	---	---	-------------	---	---	---	--------------	---	---	---	------------	---

Gambar 5.4. Format Pengiriman *Interest*

Pada Gambar 5.4. di atas dapat dilihat bahwa kode “S” sebagai *header* dari susunan ID pengirim, “D” sebagai *header node* tujuan, “R” sebagai *header* dari tipe permintaan jenis data sensor dan “H” sebagai *header* untuk jumlah *hop*.

- Format pengiriman data sensor *node* oleh sensor *node* tetangga *sink node*.

#	S	#	Susunan ID Pengirim	#	C	#	Data Sensor	#
---	---	---	---------------------	---	---	---	-------------	---

Gambar 5.5. Format Pengiriman Data Sensor *Node* Tetangga *Sink*

Pada Gambar 5.5.. di atas dapat dilihat bahwa kode "S" sebagai *header* dari susunan ID pengirim dan "C" sebagai *header* konten yang berisi data sensor.

- c. Format pengiriman data sensor oleh sensor *node* yang bukan tetangga *sink node*.

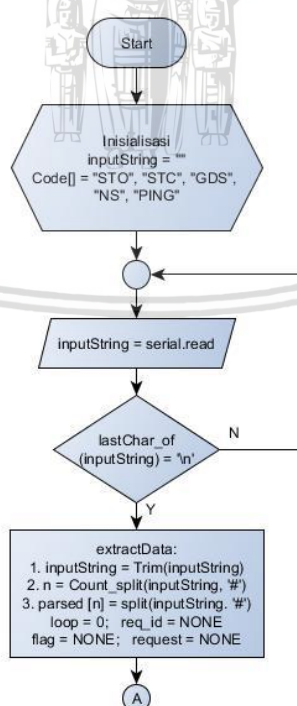
#	S	#	Susunan ID Pengirim	#	D	#	Node Tujuan	#	H	#	Jumlah Hop	#	C	#	Data Sensor	#
---	---	---	---------------------	---	---	---	-------------	---	---	---	------------	---	---	---	-------------	---

Gambar 5.6. Format Pengiriman Data Sensor Bukan *Node* Tetangga *Sink*

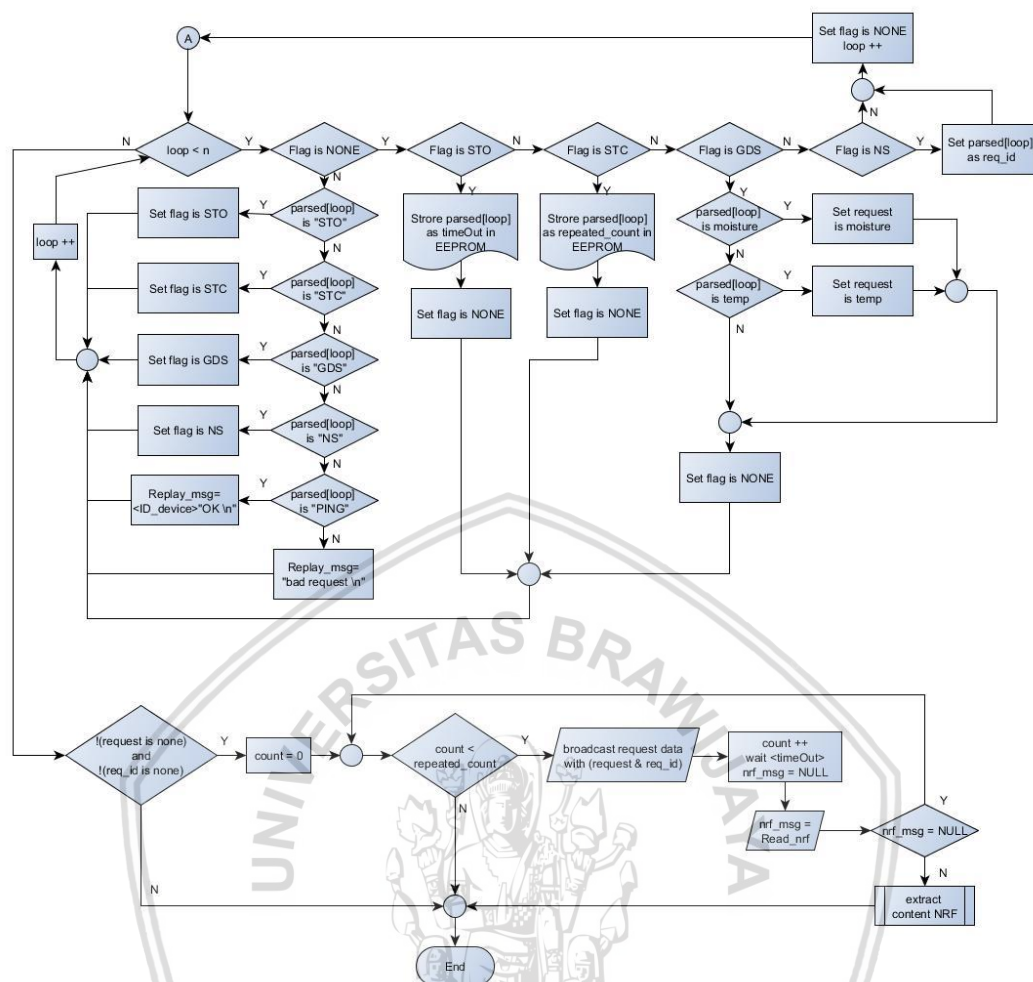
Pada Gambar 5.6. di atas dapat dilihat bahwa kode "S" sebagai *header* dari susunan ID pengirim, "D" sebagai *header node* tujuan, "H" sebagai *header* untuk jumlah hop dan "C" sebagai *header* konten yang berisi data sensor.

5.1.2.3 Diagram Alir *Sink node*

Diagram alir *sink node* akan digambarkan pada Gambar 5.7. dan Gambar 5.8. di bawah ini.



Gambar 5.7. Diagram Alir *Sink Node* (1)



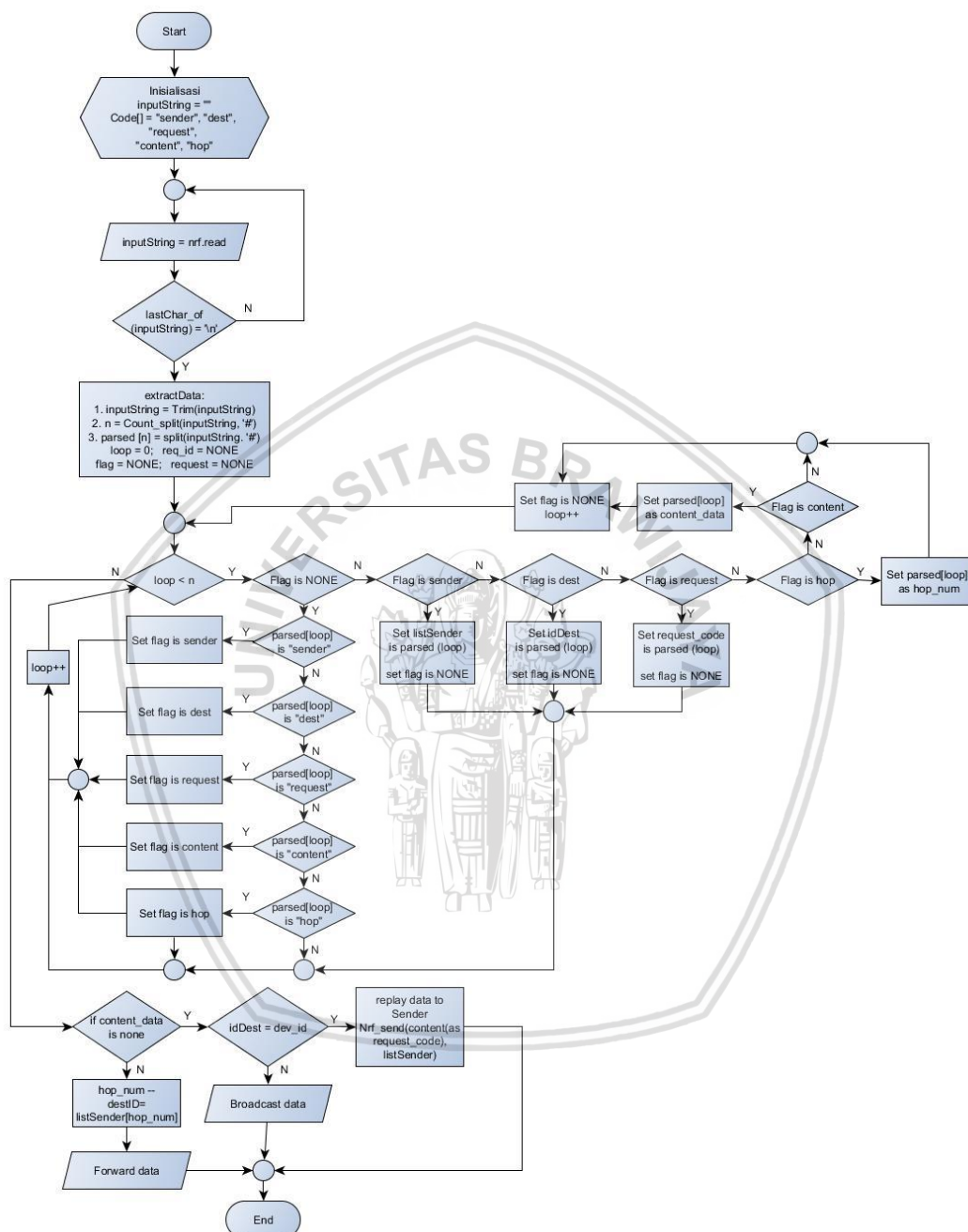
Gambar 5.8. Diagram Alir Sink Node (2)

Berikut penjelasan mengenai diagram alir *sink node*.

1. Proses awal yaitu inialisasi *input* yaitu dengan memasukkan beberapa perintah pada serial monitor yang berisikan format perintah seperti pada perancangan format perintah.
2. Setelah perintah diterima maka akan dilakukan proses pemilahan data perintah untuk memisahkan tiap-tiap kode perintah agar dapat dikenali dan dijalankan. Jika perintah tidak dikenali maka sistem tidak akan mengeksekusi perintah.
3. Setelah melakukan proses pemilahan perintah berhasil, maka sistem akan melakukan proses pengiriman permintaan data sensor dengan cara *broadcast* ke semua *node* tetangganya.
4. Jika permintaan data telah dibalas, maka akan terjadi interupsi sehingga sistem akan memprioritaskan data sensor masuk untuk dilakukan proses ekstrak.

5.1.2.4 Diagram Alir *Node* Sensor Tetangga Dari *Sink node*

Diagram alir *node* sensor tetangga dari *sink node* akan digambarkan pada Gambar 5.9. di bawah ini.



Gambar 5.9. Diagram Alir *Node* Sensor tetangga *Sink Node*

Berikut penjelasan mengenai diagram alir *node* tetangga *sink node*.

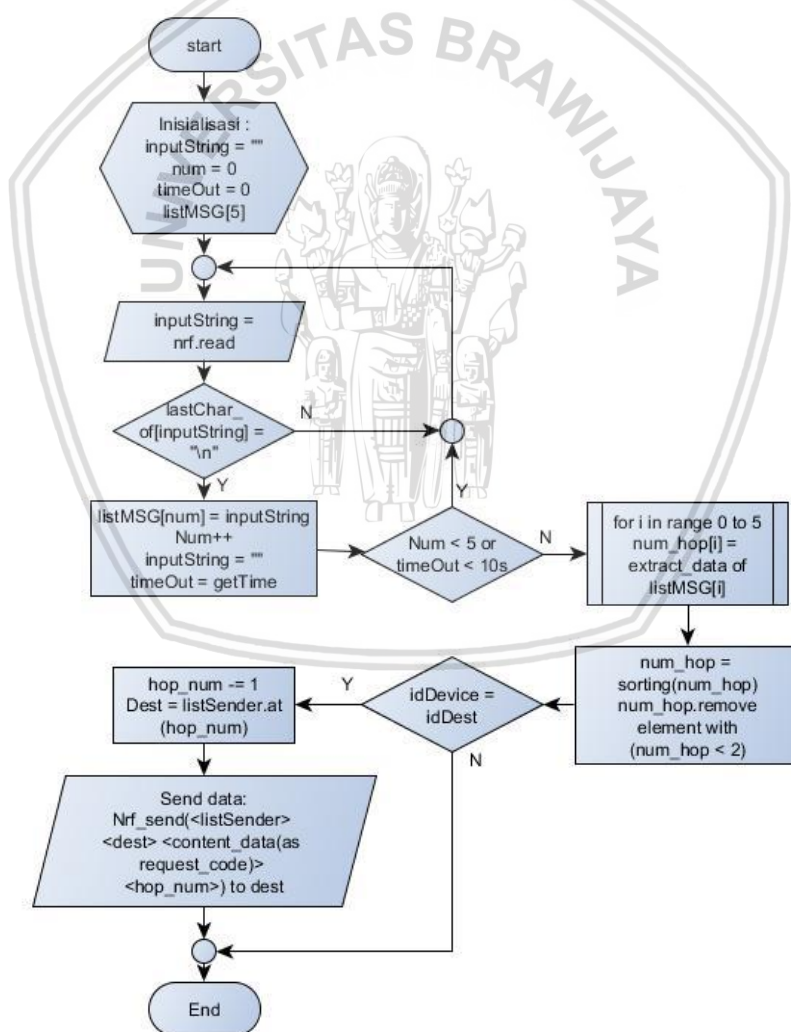
1. Proses awal yaitu inisialisasi *input* yaitu dengan melakukan cek data masuk dari modul NRF24L01.
2. Setelah perintah diterima maka akan dilakukan proses pemilahan data untuk memisahkan tiap-tiap kode perintah agar dapat dikenali dan

dijalankan. Jika perintah tidak dikenali maka sistem tidak akan mengeksekusi perintah.

3. Setelah melakukan proses pemilahan perintah berhasil, maka sistem akan melakukan cek pada data apakah terdapat konten atau tidak. Jika terdapat konten, maka sistem akan meneruskan data sensor ke *node* selanjutnya.
4. Jika tidak terdapat konten maka sistem akan melakukan cek lagi apakah ID tujuan sama dengan ID miliknya sendiri. Jika sama maka *node* sensor akan langsung membalas permintaan *sink node*. Lalu jika berbeda maka *node* sensor akan melakukan *broadcast* pesan kepada *node* tetangganya.

5.1.2.5 Diagram Alir *Node* Sensor Yang Bukan Tetangga *Sink node*

Diagram alir *node* sensor yang bukan tetangga dari *sink node* akan digambarkan pada Gambar 5.10. di bawah ini.



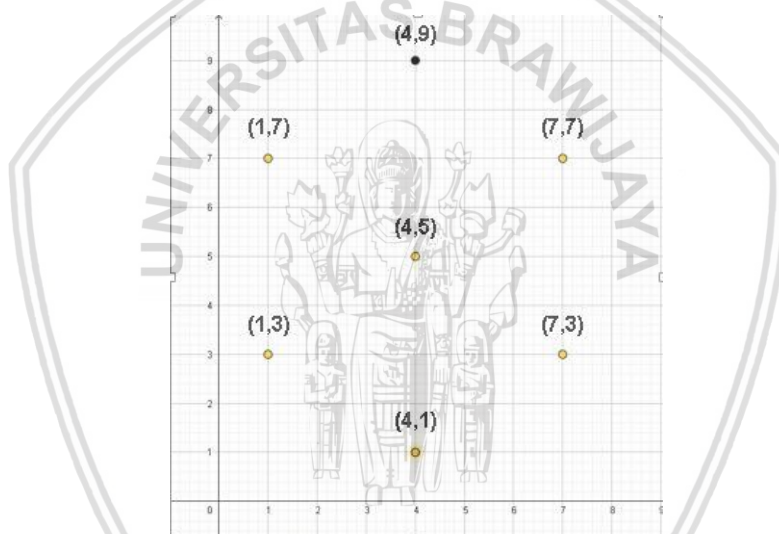
Gambar 5.10. Diagram Alir *Node* Sensor Yang Bukan Tetangga *Sink Node*

Berikut penjelasan mengenai diagram alir *node* yang bukan tetangga dari *sink node*.

1. Proses awal yaitu inisialisasi *input* yaitu dengan melakukan cek data masuk dari modul NRF24L01.
2. Setelah pesan diterima maka *node* akan menumpuk pesan hingga berjumlah 5 pesan atau kurang dari waktu *timeout*.
3. Kemudian *node* akan melakukan ekstrak data dan melakukan *sorting* jumlah *hop* untuk tiap pesan berdasarkan jumlah *hop* terkecil.
4. Jika ID tujuan sama dengan ID miliknya sendiri, maka *node* akan mengirim data sensor kepada *node* yang berada pada susunan ID pengirim terakhir pada pesan yang telah dipilih.

5.1.3 Perancangan Topologi

Berikut gambaran perancangan topologi jaringan pada penelitian ini.



Gambar 5.11. Perancangan Topologi

Pada Gambar 5.11. dapat dilihat bahwa ID *node* nantinya akan diisi berdasarkan letak koordinat pada topologi jaringan. Titik koordinat ditentukan secara acak. Sehingga pada implementasinya, titik koordinat digunakan sebagai penamaan ID *node* pada kode program bukan sebagai tata letak *node*.

5.2 Implementasi

Pada sub bab ini akan dijelaskan implementasi sistem berdasarkan perancangan yang telah dilakukan. Implementasi akan menjelaskan secara rinci mengenai batasan implementasi, spesifikasi sistem, implementasi perangkat keras dan perangkat lunak.

5.2.1 Batasan Implementasi

Dalam implementasi sistem terdapat beberapa batasan implementasi yang akan dijabarkan sebagai berikut.

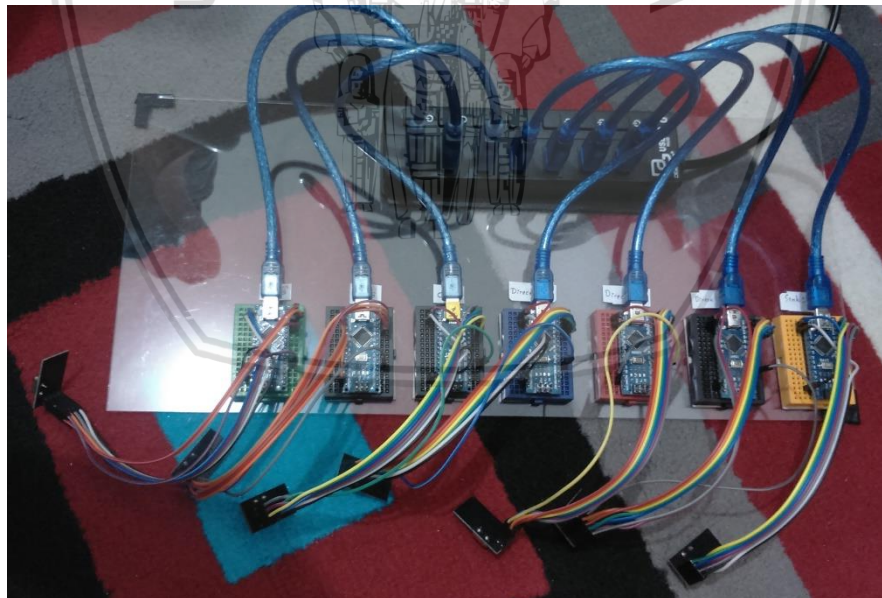
1. Mikrokontroler menggunakan Arduino Nano 3.0 dan media transmisi data menggunakan modul NRF24L01.
2. Data sensor diperoleh melalui perhitungan secara acak langsung dari kode program.
3. *Node* berjumlah 7 *node* yang terdiri dari 1 *sink node* dan 6 *sensor node*, dimana terdapat 5 *sensor node* sebagai tetangga *sink node* dan 1 *node* sensor sebagai *node* yang bukan tetangga dari *sink node*.
4. Koordinat *node* digunakan sebagai penamaan ID, bukan sebagai posisi peletakan *node*.

5.2.2 Spesifikasi Sistem

Sistem ini bertujuan mengimplementasikan protokol *routing directed diffusion*. Sistem dirancang menggunakan 7 arduino nano sebagai pemroses data, 7 modul NRF24L01 sebagai media transmisi, dan 7 *project board*.

5.2.3 Implementasi Perangkat Keras

Implementasi perangkat keras ditunjukkan oleh gambar 5.12. di bawah ini.



Gambar 5.12. Implementasi Perangkat Keras

Pada gambar 5.12 menunjukkan implementasi perangkat keras yang terdapat 7 *node* dimana 1 buah *project board* menampung 1 buah *node*. Semua komponen pada masing-masing *node* dihubungkan oleh kabel *jumper*. Masing-masing *node* nantinya akan mendapatkan daya dari USB *hub* yang terhubung ke PC.

5.2.4 Implementasi Perangkat Lunak

Implementasi perangkat lunak pada penelitian ini yaitu merupakan penulisan kode program yang dikerjakan pada perangkat lunak Arduino IDE yang memanfaatkan *library* RF24.

5.2.4.1 Implementasi Kode Program *Sink node*

Implementasi kode program *sink node* dilakukan dengan menuliskan kode program pada mikrokontroler arduino nano yang berperan sebagai *sink node*. *Sink node* bertugas meminta data sensor dan menerima data sensor. Berikut adalah beberapa fungsi penting pada kode program sebuah *sink node*.

1. Fungsi setup *sink node*

Tabel 5.2. Tabel Fungsi *Setup Sink Node*

Fungsi Setup Sink Node	
	<pre> #include "split_string.h" //import libabry split_string.h #include "string_querries.h" //import libabry string_querries.h #include "nrf_function.h" //import libabry nrf_fucntion.h //-----setup device-----// void setup() { Serial.begin(9600); serial_in.reserve(200); Serial.println(F(" ")); EEPROM_readAnything(maxCount_addr,maxCount); //membaca nilai maxCount pada EEPROM EEPROM_readAnything(timeout_addr,timeOut); //membaca nilai timeOut pada EEPROM nrfSetup(String(ID_sink).toInt()); //setup nrf24l01 dengan mengubah ID sink menjadi Integer sprintf(textBuff, "##### \n#Device ID \t: %s\n#Device Type \t: Sink Node\nTime out : %d ms \t Max round : %d\n#####", ID_sink, timeOut, maxCount); Serial.println(textBuff); //menampilkan status node } </pre>

Pada Tabel 5.2. dijelaskan fungsi *setup sink node* dimulai dengan melakukan *import library*. Kemudian pada fungsi *void setup* dilakukan dengan memulai komunikasi serial pada *baudrate* 9600 dengan menerima 200 *input* karakter dari serial monitor. Pada saat yang sama juga memanggil fungsi *nrfsetup* untuk mengaktifkan modul NRF24L01.

2. Fungsi pengiriman *interest*

Tabel 5.3. Tabel Fungsi Pengiriman *Interest*

Fungsi Pengiriman Interest	
	<pre> void broadcastCommand(String msg, String selfID) { </pre>

<pre> for (uint8_t i = 0; i<=9; i++) // x-axes { for(uint8_t k = 0; k<=9; k++) // y-axes { String tempDest=String(i)+String(k); //variabel tempDest menampung ID tujuan if(selfID.equalsIgnoreCase(tempDest)) continue; //saat broadcast melewati ID sendiri maka dilewati nrfSendMsg(tempDest.toInt(), msg); //menjalakan fungsi pengiriman dimana tempDest sebagai alamat tujuan dan msg sebagai isi pesan delay(10); } } exit_: Serial.println(F("broadcast is finished!")); </pre>

Pada Tabel 5.3. dijelaskan fungsi pengiriman *interest* diberi nama *broadcastCommand* dengan tipe *void* yang memiliki parameter pesan yang akan dikirim dan ID miliknya sendiri. Pengiriman dilakukan dengan menyusun ID tujuan yang dimasukkan pada variabel *tempDest*. Kemudian memanggil fungsi pengiriman yaitu *nrfSendMsg* dengan parameter isi pesan dan ID tujuan.

3. Fungsi penerimaan data sensor

Tabel 5.4. Tabel Fungsi Penerimaan Data Sensor

Fungsi Penerimaan Data Sensor
<pre> String nrfGetMsg() { String tmpS=""; //variabel tmpS dikosongkan terlebih dahulu boolean nrf_msgStatus=false; int len=0; len = radio.getDynamicPayloadSize(); //variabel len sebesar buffer nrf char text[len]=""; radio.read(&text, len); //membaca pesan masuk sepanjang len tmpS+=text; //tmpS menerima tambahan isi dari text for (int i = 0; i < len; i++) //cek status pesan { if((text[i]=='\n') (text[i]=='\r')) //mencari penanda akhir pesan { nrf_msgStatus=true; //mengubah status nrf_msgStatus break; } }; if(nrf_msgStatus) //kondisi saat pesan masuk { statusInbox=true; //mengubah status statusInbox, digunakan oleh fungsi broadcastCommand queString+=tmpS; //isi queString ditambah oleh tmpS queString.trim(); //menggabungkan pesan (menghilangkan spasi) Serial.println(F(" Inbox :")); Serial.println(" \t "+queString); //menampilkan pesan yang diterima </pre>

<pre> tmpS=queString; //memberikan tmpS isi pesan untuk di return queString=""; //mengosongkan kembali queString } return (tmpS); } </pre>
--

Pada Tabel 5.4. dijelaskan fungsi penerimaan data sensor yang diberi nama *nrfGetMsg* dengan tipe *string* dimana nilai *return* akan berupa *string*. Pesan masuk ditampung pada variabel *text* dan kemudian diwariskan kepada variabel *tmpS*. Kemudian variabel *queString* akan menampilkan pesan masuk pada serial monitor.

5.2.4.2 Implementasi Kode Program *Node* Tetangga *Sink node*

Implementasi kode program *node* tetangga *sink node* dilakukan dengan menuliskan kode program pada mikrokontroler arduino nano yang berperan sebagai *node* tetangga dari *sink node*. *Node* tetangga dari *sink node* bertugas untuk mengambil data sensor, mengirim data sensor dan mengirim ulang pesan *interest*. Berikut adalah beberapa fungsi penting pada kode program sebuah *node* tetangga dari *sink node*.

1. Fungsi pengiriman data sensor

Tabel 5.5. Tabel Fungsi Pengiriman Data Sensor

Fungsi Pengiriman Data Sensor
<pre> switch(dataDirect.sens) //cek kode perintah data sensor { case moisture_sens : //kondisi saat terdapat permintaan data sensor kelembaban tanah for (int j=0; j<dataDirect.IDList.length(); j++) { listOf_ID[j]=dataDirect.IDList.charAt(j); //listOf_ID index ke j berisi karakter dari dataDirect.IDList index ke j } sprintf(textBuf,"#S#s#s#C#mois:%d#\n",listOf_ID , deviceID, moist); Serial.println(textBuf); //menampilkan pesan yang akan dikirim delay(850); sendStat = nrfSendMsg(getIDFromTable(dataDirect.IDList,0).toInt(), String(textBuf)); //sendstat berisi hasil return variabel done pada fungsi nrfSendMsg() di tab nrf_function.h while((sendStat==false)&&(send_count<2)) { if (freeRam ()<668) software_Reset(); //kondisi saat RAM penuh delay(250); sendStat = nrfSendMsg(getIDFromTable(dataDirect.IDList,0).toInt(), String(textBuf)); //ulangi mengirim send_count++; }; break; </pre>

<pre> case temp_sens ://kondisi saat terdapat permintaan data sensor suhu for (int j=0; j<dataDirect.IDList.length(); j++) { listOf_ID[j]=dataDirect.IDList.charAt(j); //listOf_ID index ke j berisi karakter dari dataDirect.IDList index ke j } sprintf(textBuf, "#S#%s%s#C#temp=%d#\n",listOf_ID , deviceID, int(temp)); Serial.println(textBuf); //menampilkan pesan yang akan dikirim delay(850); sendStat = nrfSendMsg(getIDFromTable(dataDirect.IDList,0).toInt(), String(textBuf)); //sendstat berisi hasil return variabel done pada fungsi nrfSendMsg() di tab nrf_function.h while((sendStat==false)&&(send_count<2)) { if (freeRam ()<668) software_Reset(); //kondisi saat RAM penuh delay(250); sendStat = nrfSendMsg(getIDFromTable(dataDirect.IDList,0).toInt(), String(textBuf)); //ulangi mengirim send_count++; }; break; } </pre>
--

Pada Tabel 5.5. dijelaskan fungsi pengiriman data sensor bekerja dengan melakukan cek data sensor apa yang diminta. *Case moisture_sens* berfungsi untuk mengirimkan data sensor kelembaban tanah. Sedangkan *case temp_sens* berfungsi untuk mengirimkan data sensor suhu. Pengiriman dilakukan dengan memanggil fungsi *nrfSendMsg* yang diberikan ID tujuan dan isi pesan yang berisi data sensor.

2. Fungsi pengiriman ulang *interest*

Tabel 5.6. Tabel Fungsi Pengiriman Ulang *Interest*

Fungsi Pengiriman Ulang <i>Interest</i>
<pre> if((dataDirect.content=="") && (dataDirect.hop<4)) //kondisi jika jumlah hop masih kurang dari 4 { dataDirect.IDList+=String(deviceID); //menambahkan IDnya sendiri pada IDList char list_id[dataDirect.IDList.length()+1]; //variabel list_id dengan panjang dari dataDirect.IDList + 1 char dest_id[]="00"; int lenD = (dataDirect.idIN.length()>2)?2:dataDirect.idIN.length(); for (int j=0; j<lenD; j++) //mencari id tujuan { dest_id[2-1- j]=dataDirect.idIN.charAt(dataDirect.idIN.length()-1-j); } dataDirect.IDList.toCharArray(list_id, dataDirect.IDList.length()+1); </pre>

```

sprintf(textBuf, "S#%sD#%sR#%dH#%d\n", list_id, dest_id, dataDirect.sens, dataDirect.hop+1); //menyusun pesan interest
Serial.println(textBuf); //menampilkan pesan yang dikirim
broadcastCommand(String(textBuf),
String(dataDirect.IDList)); //broadcast pesan interest
}

```

Pada Tabel 5.6. dijelaskan fungsi pengiriman ulang *interest* dilakukan saat kondisi tidak terdapat data sensor dan jumlah *hop* masih kurang dari 4. Pesan *interest* disusun dengan menambahkan ID miliknya sendiri pada pesan *interest* yang telah diterima sebelumnya. Kemudian memanggil fungsi *broadcastCommand* untuk melakukan pengiriman *interest*.

5.2.4.3 Implementasi Kode Program Node Yang Bukan Tetangga Sink node

Implementasi kode program *node* yang bukan tetangga *sink node* dilakukan dengan menuliskan kode program pada mikrokontroler arduino nano yang berperan sebagai *node* yang bukan tetangga dari *sink node*. *Node* yang bukan tetangga dari *sink node* bertugas untuk mengambil data sensor dan mengirim data sensor berdasarkan jumlah *hop* terkecil. Berikut adalah beberapa fungsi penting pada kode program sebuah *node* yang bukan tetangga dari *sink node*.

1. Fungsi *directed diffusion*

Tabel 5.7. Tabel Fungsi *Directed Diffusion*

Fungsi Directed Diffusion	
	<pre> int directDiffusion (String msgQuery[], int numMsg) //parameter yaitu isi pesan yang masuk dan jumlah pesan { int numI= 0; int idx = 0; int tempHopArray [numMsg]; int tempIDxArray [numMsg]; for(int id=0; id< numMsg; id++) //pengulangan mencari inisialisasi hop array { msgQuery[id].trim(); //menghapus spasi pada isi dari msgQuery pada index ke id Serial.println("pesan ke-"+String(id)+" "+msgQuery[id]); // menampilkan isi dari msgQuery index ke id pada serial monitor query_nrf dataExtracted = queries_nrf(msgQuery[id]); //dataExtracted berisi data return dari fungsi queries_nrf() pada tab queries_exec.h Serial.println(" hop = "+String(dataExtracted.hop)); //menampilkan jumlah hop dari dataExtracted.hop Serial.println(" ID destination = "+dataExtracted.idIN + \t ID device: "+String(deviceID)); //menampilkan ID tujuan dan ID miliknya sendiri // jika dataExtracted.knownData bernilai true dan jumlah hop lebih dari 1 serta node tujuan sama dengan ID miliknya sendiri </pre>

```

        if( ( dataExtracted.knownData ) && (dataExtracted.hop>1)
        && (dataExtracted.idIN.equalsIgnoreCase(String(deviceID))) )
        {
            tempHopArray[idx] = dataExtracted.hop; //tempHopArray
            menyimpan data dari dataEctracted.hop pada index ke idx
            (berisi daftar sususan jumlah hop)
            tempIDxArray[idx] = id; //tempIDxArray menyimpan data
            dari id pada index ke idx
            idx++; // data idx di increment
            Serial.println(idx);
        }
    }
    int hopArray[idx]; // hopArray memiliki lebar array sebesar
    idx
    int IDxArray[idx]; // IDxArray memiliki lebar array sebesar
    idx

    for ( int id = 0; id<idx; id++) // menyamakan isi array
    {
        hopArray[id] = tempHopArray[id];
        IDxArray[id] = tempIDxArray[id];
    }
    bubbleSort(hopArray, IDxArray); //Urutkan bilangan yang
    lebih kecil daripada bilangan yang pertama didalam hop array,
    dan juga id array mulai dari yang terkecil

    Serial.println(" num of array = "+String(idx));
    int numData = idx;
    idx=0;
    for ( int id=0; id<numData; id++) // cari data hop array
    yang terkecil
    {
        Serial.println("hop array = "+String(hopArray[id]));
        if( hopArray [id]>1) // jika hop array lebih besar dari 1
        (pewarisan data dai hop terkecil)
        {
            idx = IDxArray[id];
            break;
        }
    }
    numI = idx;
    return numI;
}

```

Pada Tabel 5.7. dijelaskan fungsi *directed diffusion* merupakan fungsi dengan tipe *integer* dimana akan *return* data *integer*. Fungsi *directed diffusion* bekerja dengan membuat variabel *tempHopArray* untuk menampung pesan permintaan yang masuk ketika pesan yang masuk lebih dari satu. Kemudian isi dari *tempHopArray* diwariskan pada variabel *hopArray* untuk dilakukan *sorting* pesan dengan memanggil fungsi *bubbleSort*. Fungsi *bubbleSort* berfungsi untuk melakukan *sorting* pesan berdasarkan jumlah *hop* terkecil. Nilai *return* yang dikembalikan yaitu nilai indeks pesan dengan jumlah *hop* terpendek.

BAB 6 PENGUJIAN DAN ANALISIS

Pada tahap pengujian dan analisis berisikan mengenai pengujian sistem dimana sistem akan diuji berdasarkan daftar pengujian sistem yang telah dituliskan di bab 3 pada sub bab pengujian sistem.

6.1 Pengujian Status *Node* Pada Serial Monitor

Pengujian status *node* pada serial monitor merupakan pengujian awal dari keseluruhan proses pengujian sistem.

6.1.1 Tujuan Pengujian

Tujuan dari pengujian status *node* pada serial monitor adalah untuk mengetahui atau memastikan bahwa *node* telah terhubung dengan serial monitor dan siap untuk beroperasi.

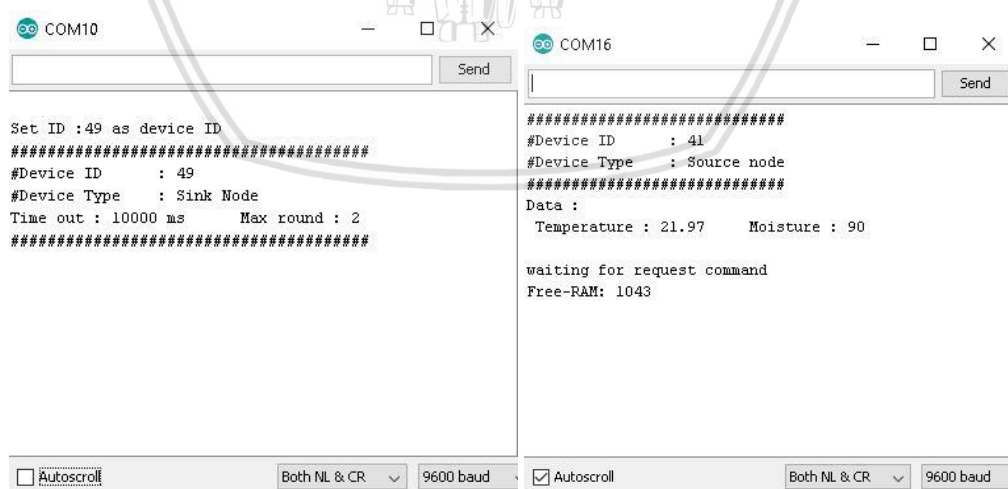
6.1.2 Prosedur Pengujian

Berikut adalah prosedur dari pengujian status *node* pada serial monitor yang dilakukan dengan beberapa langkah yaitu :

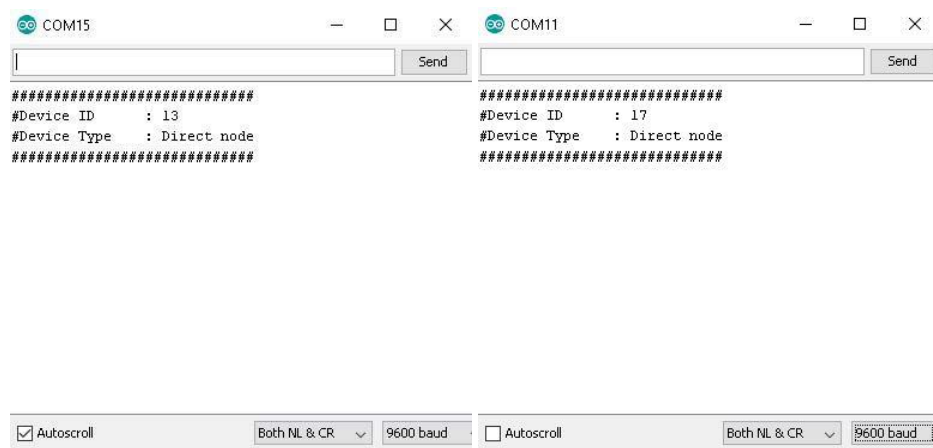
1. Menghubungkan setiap *node* dengan *notebook* melalui USB Hub.
2. Membuka Arduino IDE.
3. Membuka serial monitor pada masing-masing *node* yang terhubung.

6.1.3 Hasil Dan Analisis Pengujian

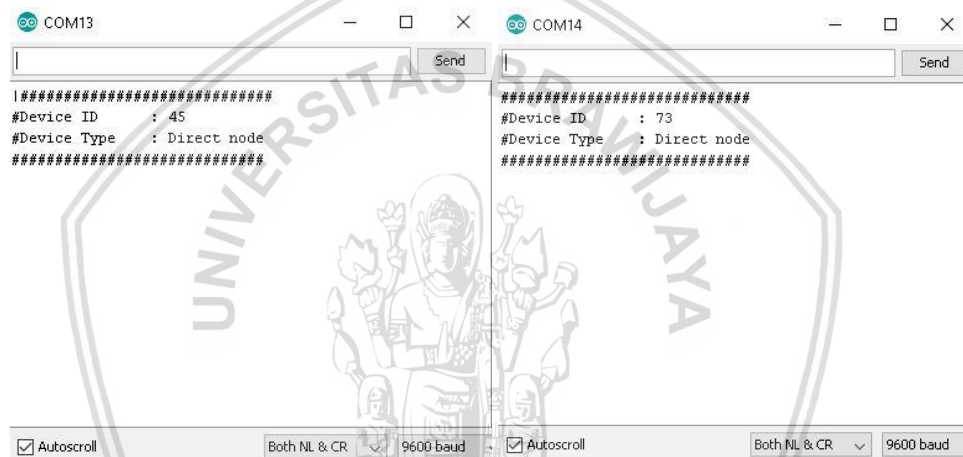
Pengujian status *node* pada serial monitor dilakukan sesuai dengan prosedur pengujian. Berikut hasil dari pengujian status *node* pada serial monitor.



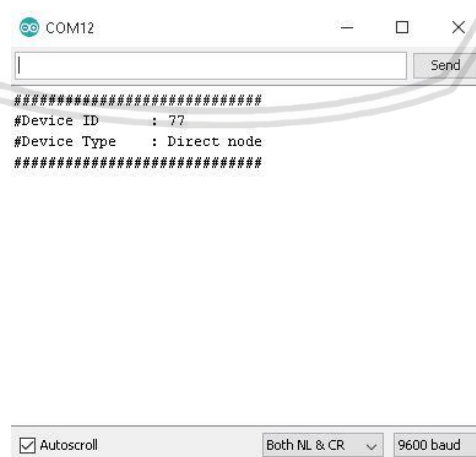
Gambar 6.1. Status *Sink node* (4,9) dan *Sensor Node* (4,1)



Gambar 6.2. Status Sensor *Node* (1,3) dan Sensor *Node* (1,7)



Gambar 6.3. Status Sensor *Node* (4,5) dan Sensor *Node* (7,3)



Gambar 6.4. Status Sensor *Node* (7,7)

Berikut adalah tabel dari hasil pengujian status *node* pada serial monitor berdasarkan Gambar 6.1 sampai Gambar 6.4.

Tabel 6.1. Tabel Pengujian Status *Node* Pada Serial Monitor

ID <i>Node</i>	Tipe <i>Node</i>	Status <i>Node</i>
49	<i>Sink node</i>	Berhasil Ditampilkan
41	<i>Sensor Node</i>	Berhasil Ditampilkan
15	<i>Sensor Node</i>	Berhasil Ditampilkan
17	<i>Sensor Node</i>	Berhasil Ditampilkan
45	<i>Sensor Node</i>	Berhasil Ditampilkan
73	<i>Sensor Node</i>	Berhasil Ditampilkan
77	<i>Sensor Node</i>	Berhasil Ditampilkan

Berdasarkan data yang diperoleh dari Tabel 6.1. setiap *node* berhasil menampilkan status *node* pada serial monitor. Status *node* yang ditampilkan yaitu ID *node* dan tipe *node*.

6.2 Pengujian Sensor *Node* Menampilkan Data Sensor

Pengujian sensor *node* dapat menampilkan data sensor merupakan pengujian kedua dari keseluruhan proses pengujian sistem.

6.2.1 Tujuan Pengujian

Tujuan dari pengujian sensor *node* dapat menampilkan data sensor adalah untuk mengetahui atau memastikan bahwa *node* dapat memproses data sensor suhu dan kelembaban tanah secara acak melalui program dan menampilkannya pada serial monitor.

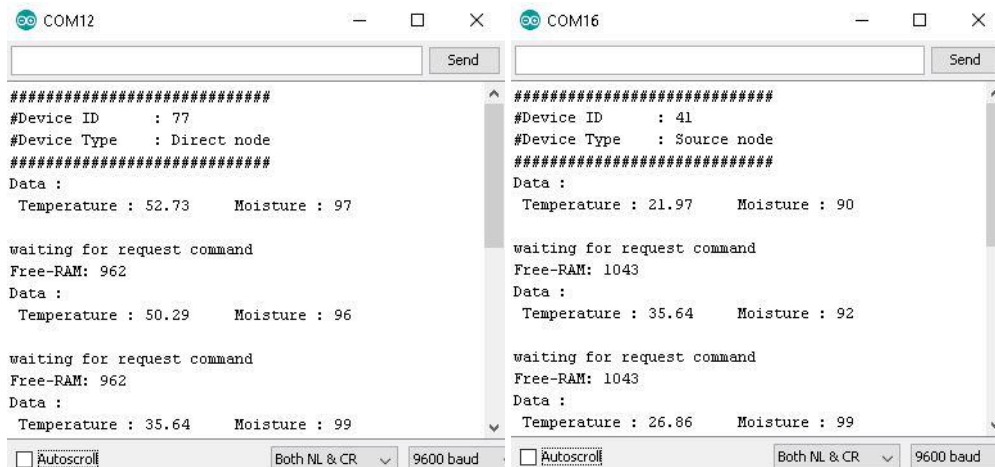
6.2.2 Prosedur Pengujian

Berikut adalah prosedur dari pengujian sensor *node* dapat menampilkan data sensor yang dilakukan dengan beberapa langkah yaitu :

1. Menghubungkan setiap *node* dengan *notebook* melalui USB Hub.
2. Membuka Arduino IDE
3. Membuka serial monitor pada masing-masing *node* yang terhubung.

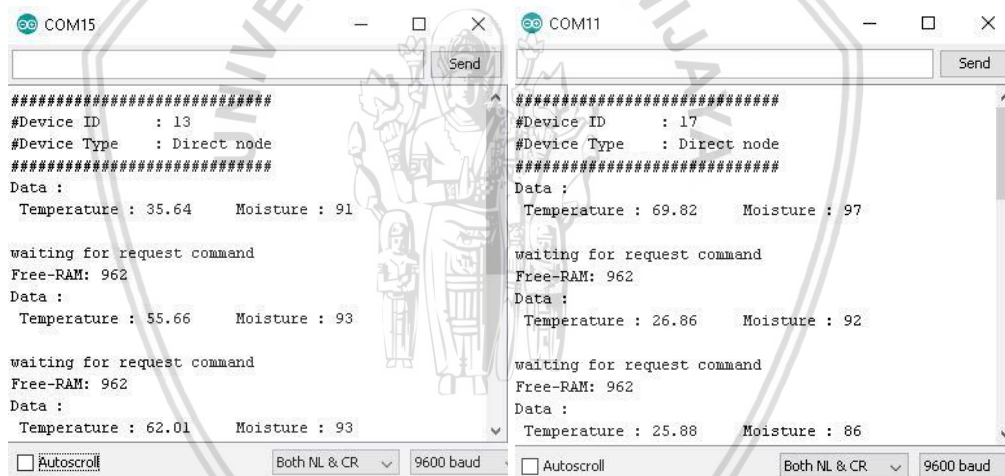
6.2.3 Hasil Dan Analisis Pengujian

Pengujian sensor *node* dapat menampilkan data sensor dilakukan sesuai dengan prosedur pengujian. Berikut hasil dari pengujian sensor *node* dapat menampilkan data sensor.



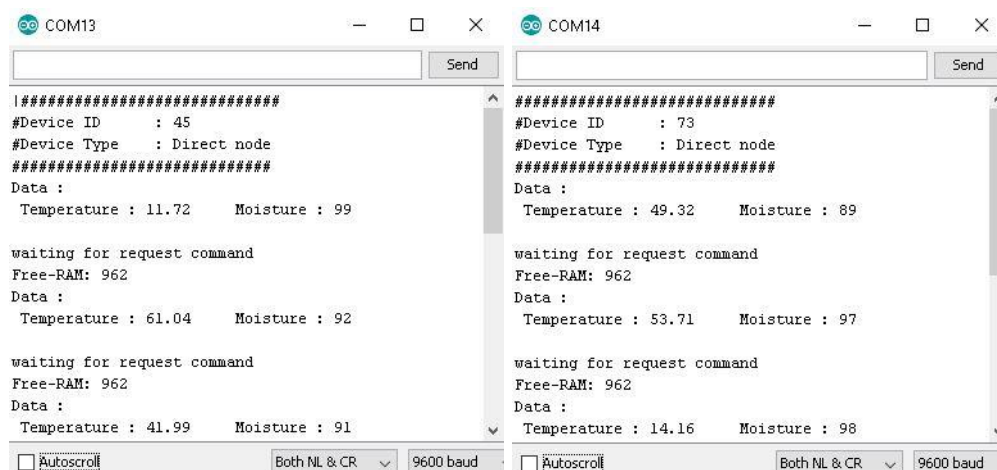
Gambar 6.5. Status Sensor *Node* (7,7) dan Sensor *Node* (4,1)

Gambar 6.5. menunjukkan sensor *node* koordinat (7,7) pada COM12 dan (4,1) pada COM16 dapat menampilkan data sensor suhu dan kelembaban tanh pada serial monitor.



Gambar 6.6. Status Sensor *Node* (1,3) dan Sensor *Node* (1,7)

Gambar 6.6. menunjukkan sensor *node* koordinat (1,3) pada COM15 dan (1,7) pada COM11 dapat menampilkan data sensor suhu dan kelembaban tanh pada serial monitor.



Gambar 6.7. Status Sensor *Node* (4,5) dan Sensor *Node* (7,3)

Gambar 6.7. menunjukkan sensor *node* koordinat (4,5) pada COM13 dan (7,3) pada COM14 dapat menampilkan data sensor suhu dan kelembaban tanah pada serial monitor.

Berikut adalah tabel dari hasil pengujian status *node* pada serial monitor berdasarkan Gambar 6.5 sampai Gambar 6.7.

Tabel 6.2. Tabel Pengujian Data Sensor Serial Monitor

ID <i>Node</i>	Tipe <i>Node</i>	Data Sensor
41	Sensor <i>Node</i>	Berhasil Ditampilkan
15	Sensor <i>Node</i>	Berhasil Ditampilkan
17	Sensor <i>Node</i>	Berhasil Ditampilkan
45	Sensor <i>Node</i>	Berhasil Ditampilkan
73	Sensor <i>Node</i>	Berhasil Ditampilkan
77	Sensor <i>Node</i>	Berhasil Ditampilkan

Berdasarkan data yang diperoleh dari Tabel 6.2. setiap *node* berhasil menampilkan data sensor pada serial monitor. Nilai dari sensor yang ditampilkan adalah nilai sensor suhu dan kelembaban tanah.

6.3 Pengujian Fungsional

Pengujian fungsional merupakan pengujian ketiga dari keseluruhan proses pengujian sistem.

6.3.1 Tujuan Pengujian

Tujuan dari pengujian fungsional adalah untuk mengetahui apakah sistem telah beroperasi atau berjalan sesuai protokol *routing directed diffusion*.

6.3.2 Prosedur Pengujian

Sistem akan diuji dengan melakukan pengiriman data sensor suhu, kemudian data sensor kelembaban tanah. Berikut adalah prosedur dari pengujian fungsional yang dilakukan dengan beberapa langkah yaitu :

1. Menghubungkan setiap *node* dengan *notebook* melalui USB Hub.
2. Membuka Arduino IDE
3. Membuka serial monitor pada masing-masing *node* yang terhubung.
4. Memasukkan perintah permintaan data sensor ke *sink node* untuk meminta data ke *node* tetangga *sink node* melalui serial monitor sesuai dengan format perintah permintaan data sensor.
5. Cek status penerimaan permintaan data setiap tetangga *sink node* pada serial monitor.
6. Cek status *broadcast* data setiap tetangga *sink node* pada serial monitor.
7. Cek status *node* sensor membalas permintaan dengan mengirim data sensor langsung ke *sink node* di serial monitor.
8. Cek penerimaan data sensor pada *sink node* yang ditampilkan pada serial monitor.
9. Prosedur nomor 4 – 8 diulang sebanyak jumlah *node* tetangga dari *sink node*.
10. Memasukkan perintah permintaan data sensor ke *sink node* untuk meminta data ke *node* yang bukan tetangga *sink node* melalui serial monitor sesuai dengan format perintah permintaan data sensor.
11. Cek status penerimaan permintaan data setiap tetangga *sink node* pada serial monitor.
12. Cek status *broadcast* data setiap tetangga *sink node* pada serial monitor.
13. Cek status *node* sensor tujuan menerima permintaan data sensor.
14. Cek status *node* sensor membalas permintaan dengan mengirim data sensor melalui jumlah hop terkecil pada serial monitor.
15. Cek status penerimaan data sensor oleh tetangga *sink node* untuk diteruskan ke *sink node*.
16. Cek penerimaan data sensor pada *sink node* yang ditampilkan pada serial monitor.

6.3.3 Hasil Dan Analisis Pengujian

Berikut adalah hasil dan analisis pengujian data sensor suhu dan data sensor kelembaban tanah yang akan diuraikan pada sub bab di bawah ini.

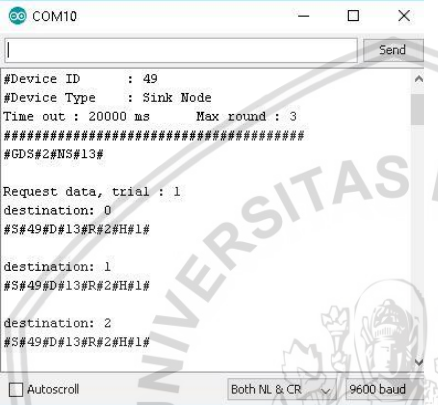
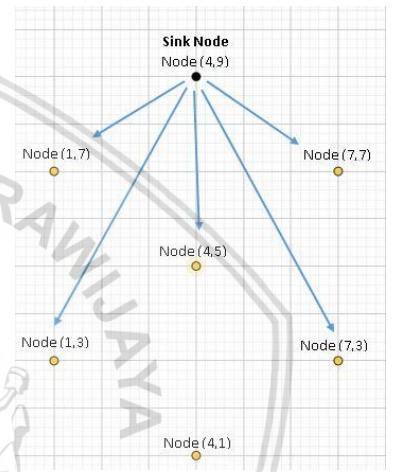
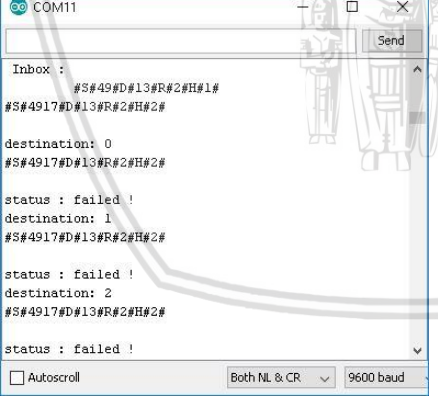
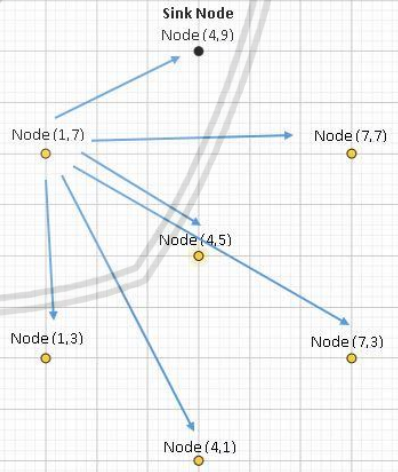
6.3.3.1 Hasil Dan Analisis Pengujian Pengiriman Data Sensor Suhu

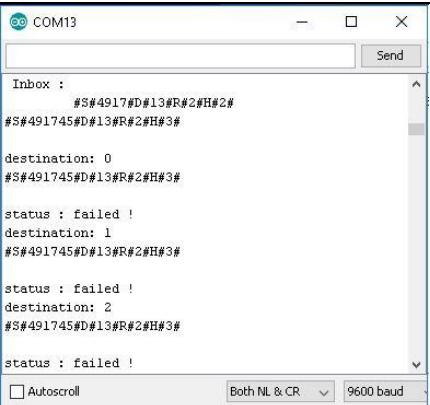
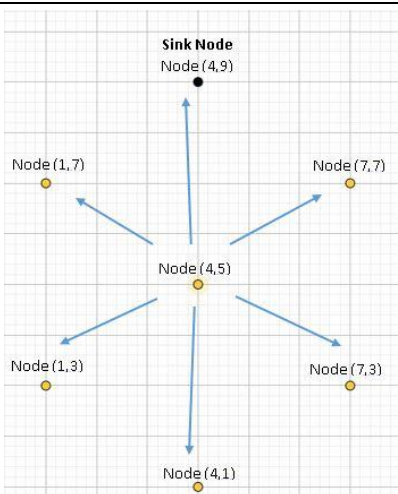
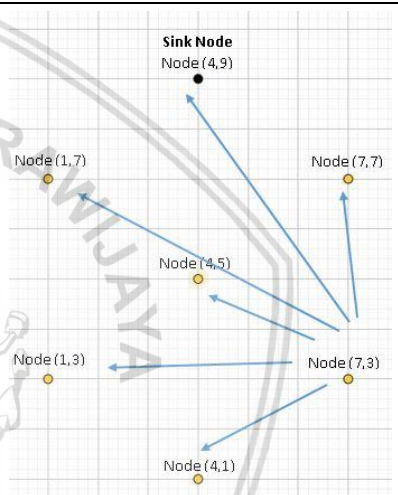
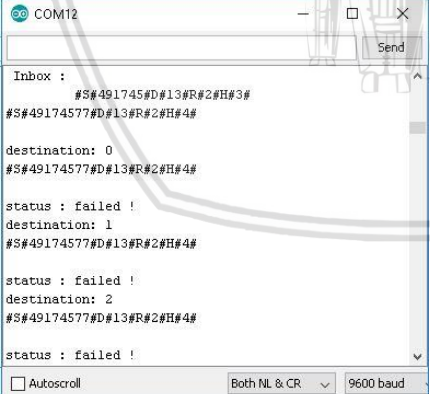
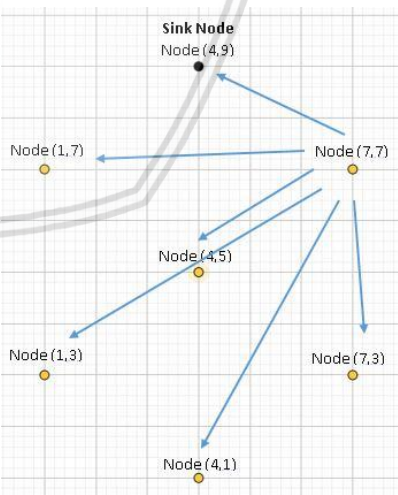
Pengujian fungsional pengiriman data sensor suhu dilakukan dengan menguji semua *node* sensor untuk mengirimkan data sensor suhu. Berikut hasil dari pengujian fungsional pengiriman data sensor suhu.

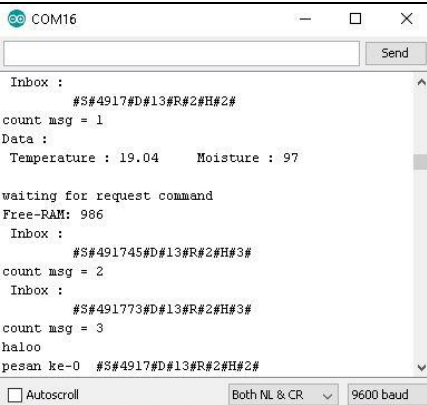
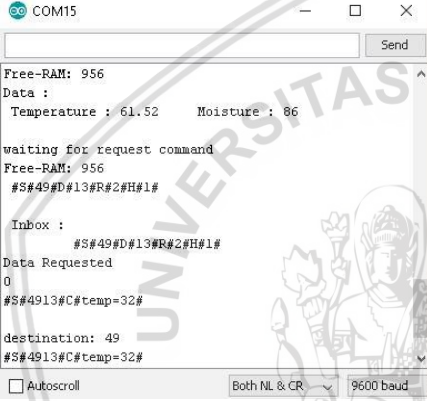
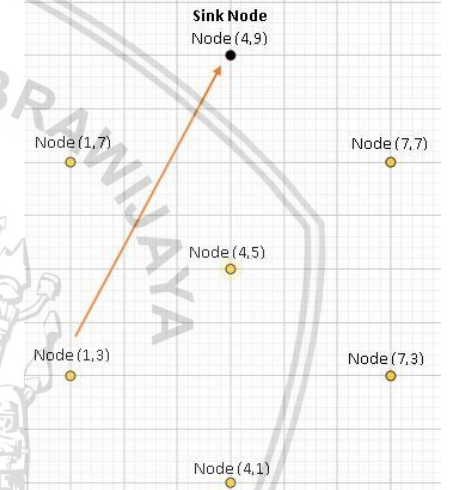
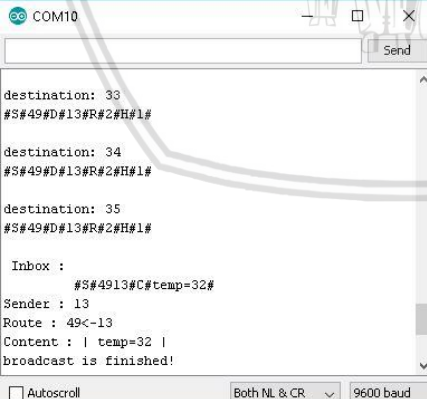
Berikut di bawah hasil pengujian fungsional pengiriman data sensor suhu diantaranya.

1. Pengiriman Data Sensor Suhu Oleh *Node* (1,3)

Tabel 6.3. Tabel Pengiriman Data Sensor Suhu Oleh *Node* (1,3)

Proses	Hasil	Skema Proses
1.	 Proses saat <i>sink node</i> broadcast interest ke seluruh <i>node</i> tetangganya.	
2.	 Proses saat <i>node</i> (1,7) menerima <i>interest</i> lalu broadcast ke seluruh <i>node</i> tetangganya.	

3.	 Proses saat <i>node</i> (4,5) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
4.	 Proses saat <i>node</i> (7,3) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
5.	 Proses saat <i>node</i> (7,7) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	

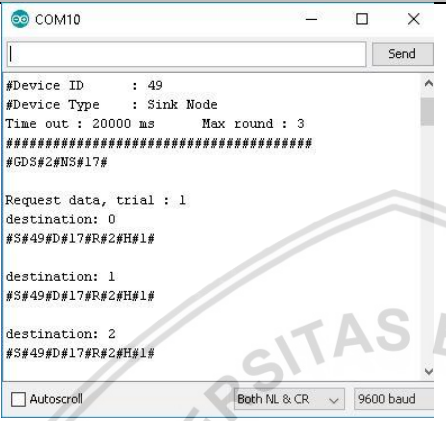
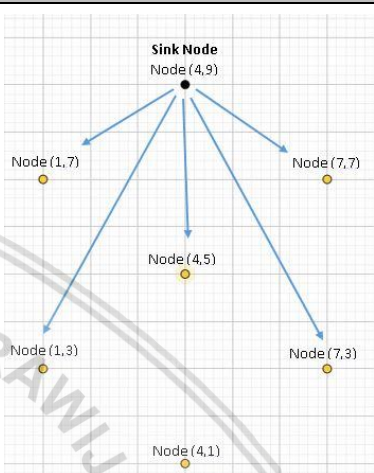
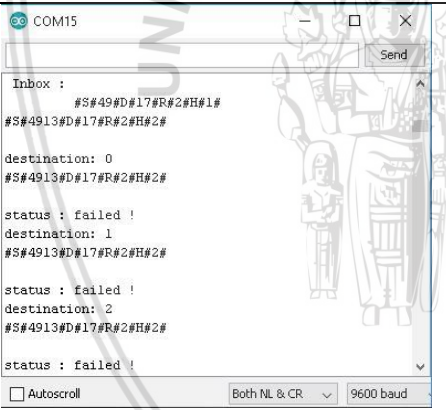
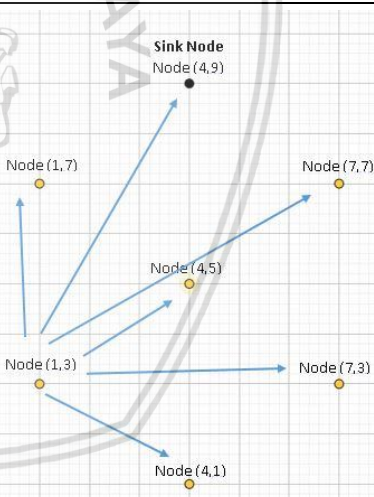
6.	 Proses saat <i>node</i> (4,1) menerima beberapa <i>interest</i>, namun tidak dieksekusi karena <i>node</i> tujuan buka dirinya sendiri.	
7.	 Proses saat <i>node</i> (1,3) menerima <i>interest</i> dan langsung membalas dengan mengirim data sensor ke <i>sink node</i>.	
8.	 Proses saat <i>sink node</i> menerima data sensor dari <i>node</i> (1,3)	

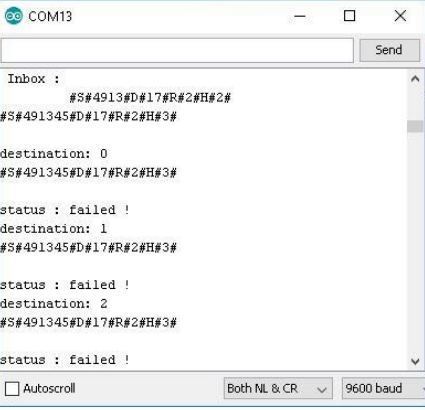
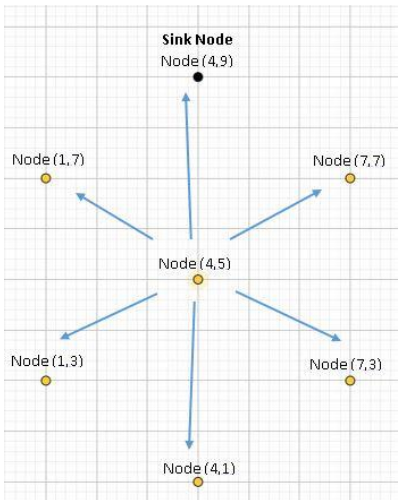
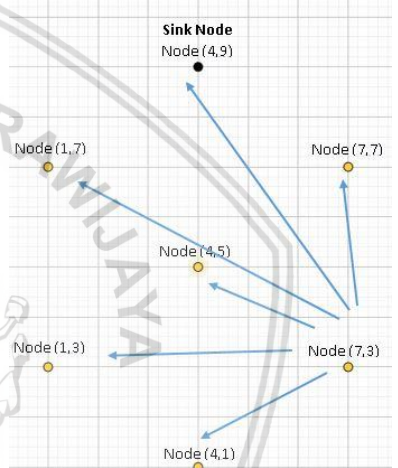
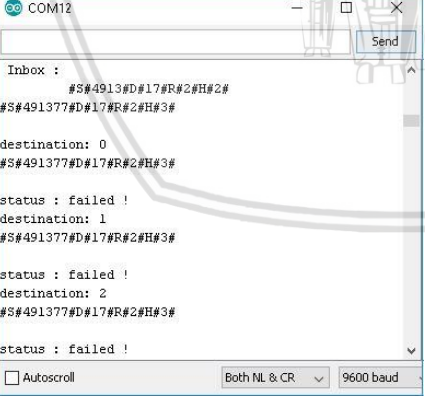
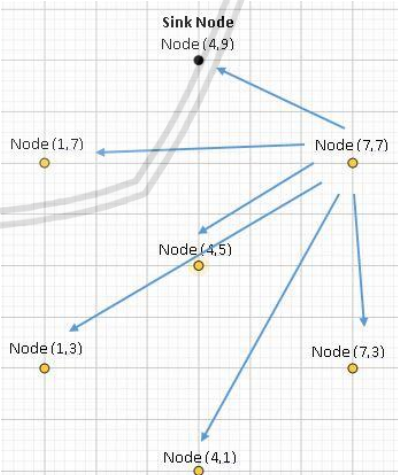
Berdasarkan hasil pengujian pengiriman data sensor suhu oleh *node* (1,3) yang dijabarkan pada Tabel 6.3., dapat dianalisis bahwa *sink node* dapat mengirimkan *interest* ke *node* tetangganya. Kemudian *node* tetangga yang lain yang berhasil menerima *interest* meneruskan pesan dengan melakukan *broadcast* ke *node*

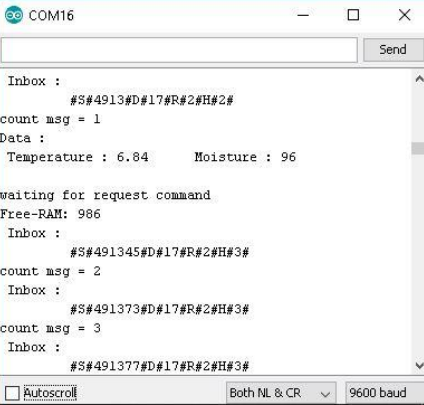
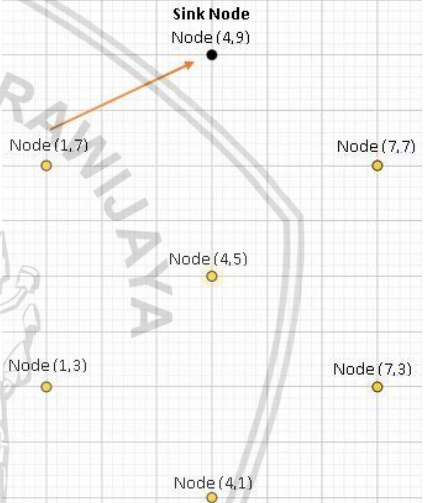
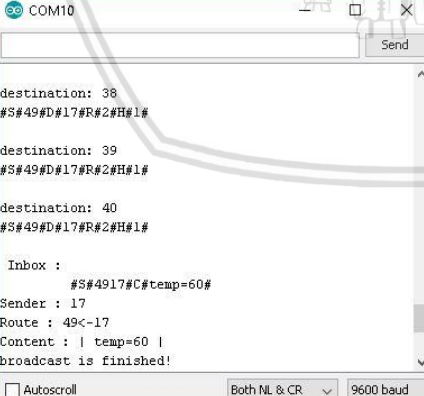
tetangga lainnya. Lalu *node* (1,3) yang menerima *interest* dan merupakan tetangga *sink node* langsung mengirimkan balasan yang berisi data sensor suhu kepada *sink node*. *Sink node* berhasil menerima data sensor.

2. Pengiriman Data Sensor Suhu Oleh *Node* (1,7)

Tabel 6.4. Tabel Pengiriman Data Sensor Suhu Oleh *Node* (1,7)

Proses	Hasil	Skema Proses
1.	<pre>#Device ID : 49 #Device Type : Sink Node Time out : 20000 ms Max round : 3 ##### #GDS#2#NS#17# Request data, trial : 1 destination: 0 #S#49#D#17#R#2#H#1# destination: 1 #S#49#D#17#R#2#H#1# destination: 2 #S#49#D#17#R#2#H#1# Autoscroll Both NL & CR 9600 baud</pre> Proses saat <i>sink node</i> broadcast <i>interest</i> ke seluruh <i>node</i> tetangganya.	
2.	<pre>Inbox : #S#49#D#17#R#2#H#1# #S#49#D#17#R#2#H#2# destination: 0 #S#49#D#17#R#2#H#2# status : failed ! destination: 1 #S#49#D#17#R#2#H#2# status : failed ! destination: 2 #S#49#D#17#R#2#H#2# status : failed ! Autoscroll Both NL & CR 9600 baud</pre> Proses saat <i>node</i> (1,3) menerima <i>interest</i> lalu broadcast ke seluruh <i>node</i> tetangganya.	

3.	 Proses saat <i>node</i> (4,5) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
4.	 Proses saat <i>node</i> (7,3) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
5.	 Proses saat <i>node</i> (7,7) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	

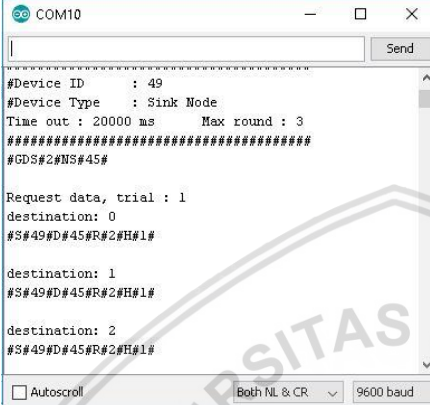
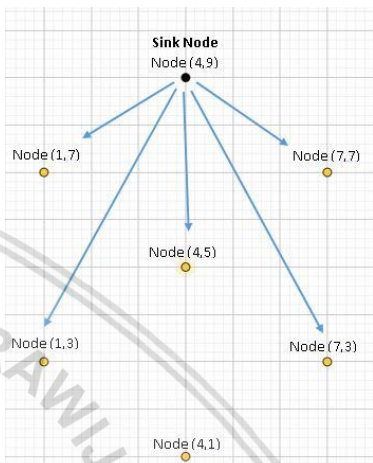
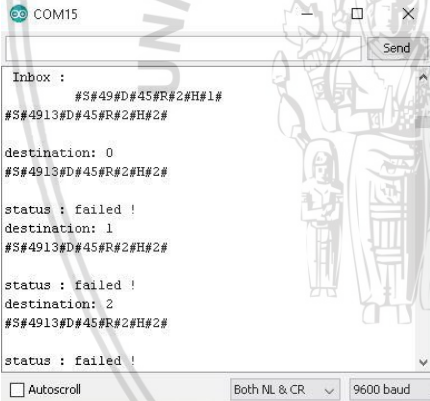
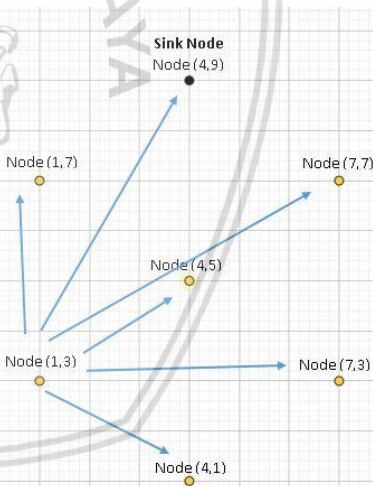
6.	 Proses saat <i>node</i> (4,1) menerima beberapa <i>interest</i>, namun tidak dieksekusi karena <i>node</i> tujuan buka dirinya sendiri.	
7.	 Proses saat <i>node</i> (1,7) menerima <i>interest</i> dan langsung membalas dengan mengirim data sensor ke <i>sink node</i>.	
8.	 Proses saat <i>sink node</i> menerima data sensor dari <i>node</i> (1,7)	

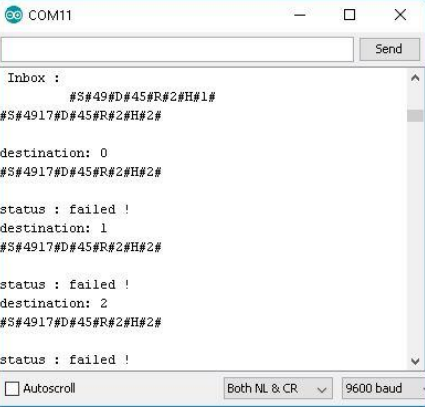
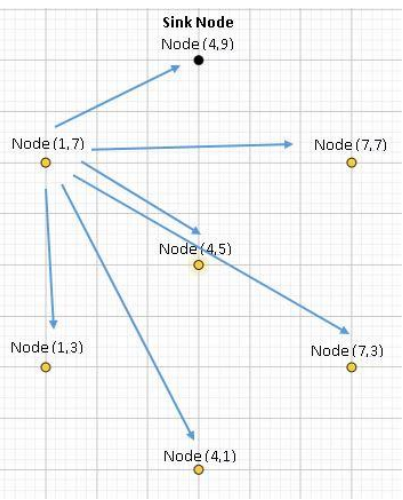
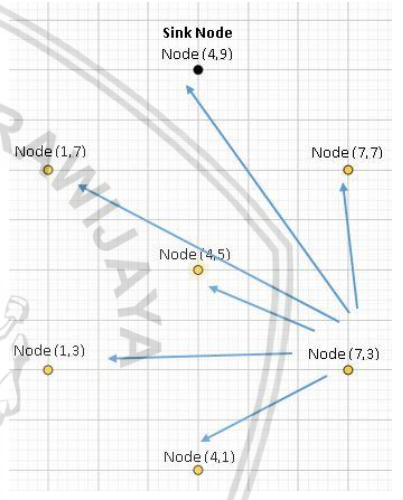
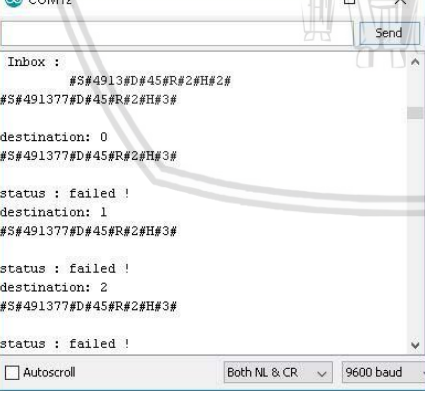
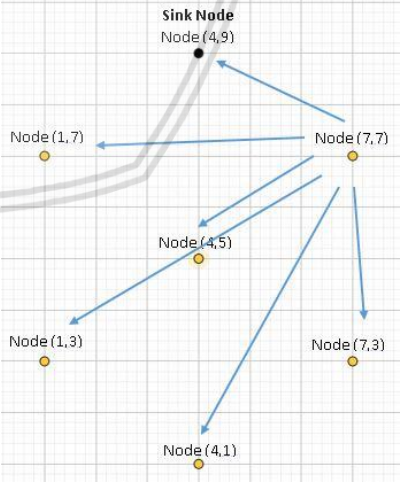
Berdasarkan hasil pengujian pengiriman data sensor suhu oleh *node* (1,7) yang dijabarkan pada Tabel 6.4., dapat dianalisis bahwa *sink node* dapat mengirimkan *interest* ke *node* tetangganya. Kemudian *node* tetangga yang lain yang berhasil menerima *interest* meneruskan pesan dengan melakukan *broadcast* ke *node*

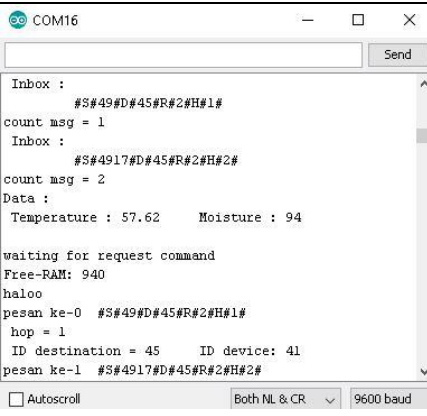
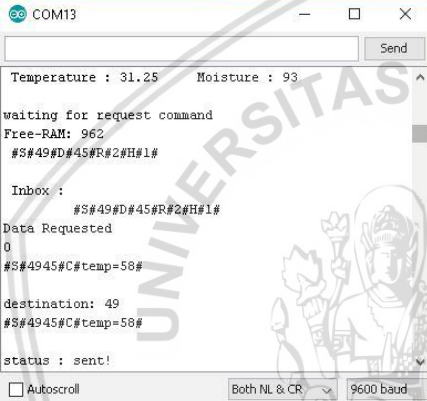
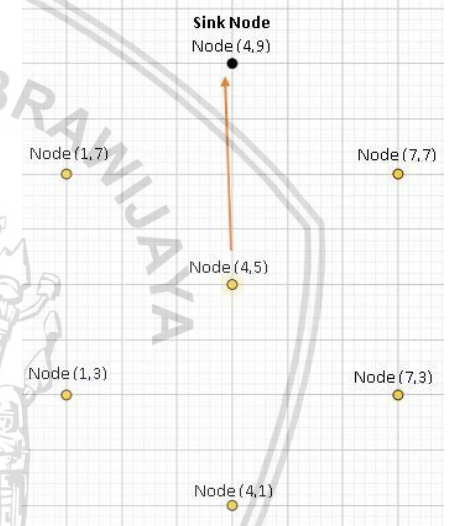
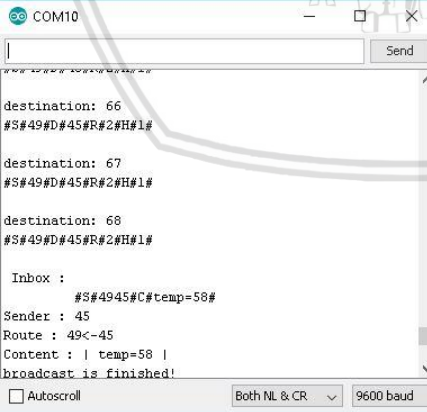
tetangga lainnya. Lalu *node* (1,7) yang menerima *interest* dan merupakan tetangga *sink node* langsung mengirimkan balasan yang berisi data sensor suhu kepada *sink node*. *Sink node* berhasil menerima data sensor.

3. Pengiriman Data Sensor Suhu Oleh *Node* (4,5)

Tabel 6.5. Tabel Pengiriman Data Sensor Suhu Oleh *Node* (4,5)

Proses	Hasil	Skema Proses
1.	<pre>COM10 #Device ID : 49 #Device Type : Sink Node Time out : 20000 ms Max round : 3 ##### #GDS#2#NS#45# Request data, trial : 1 destination: 0 #S#49#D#45#R#2#H#1# destination: 1 #S#49#D#45#R#2#H#1# destination: 2 #S#49#D#45#R#2#H#1# Autoscroll Both NL & CR 9600 baud</pre> Proses saat <i>sink node</i> broadcast <i>interest</i> ke seluruh <i>node</i> tetangganya.	
2.	<pre>COM15 Inbox : #S#49#D#45#R#2#H#1# #S#4913#D#45#R#2#H#2# destination: 0 #S#4913#D#45#R#2#H#2# status : failed ! destination: 1 #S#4913#D#45#R#2#H#2# status : failed ! destination: 2 #S#4913#D#45#R#2#H#2# status : failed ! Autoscroll Both NL & CR 9600 baud</pre> Proses saat <i>node</i> (1,3) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	

3.	 Proses saat <i>node</i> (1,7) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
4.	 Proses saat <i>node</i> (7,3) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
5.	 Proses saat <i>node</i> (7,7) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	

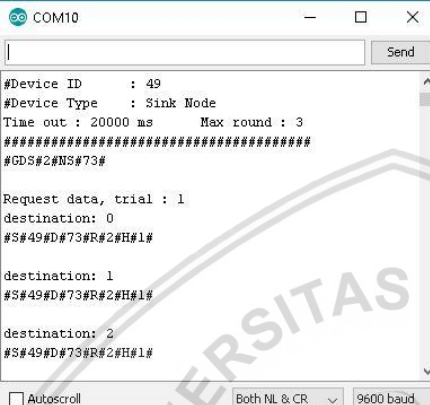
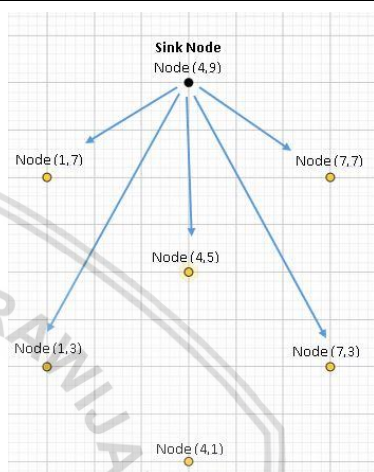
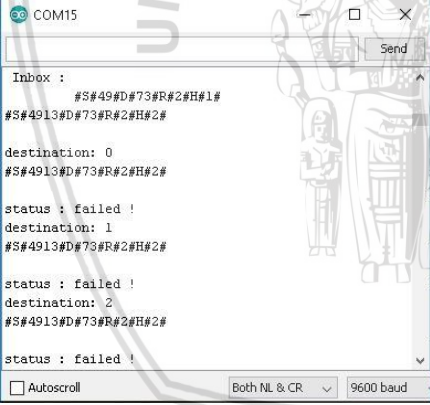
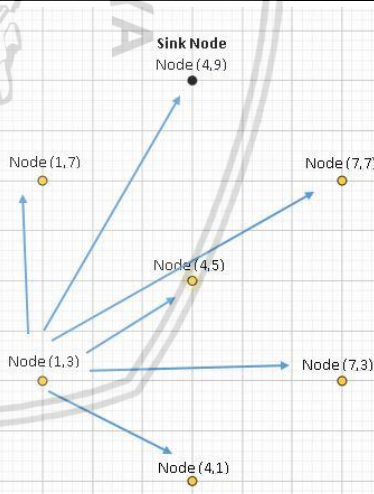
6.	 Proses saat <i>node</i> (4,1) menerima beberapa <i>interest</i>, namun tidak dieksekusi karena <i>node</i> tujuan buka dirinya sendiri.	
7.	 Proses saat <i>node</i> (4,5) menerima <i>interest</i> dan langsung membalas dengan mengirim data sensor ke <i>sink node</i>.	
8.	 Proses saat <i>sink node</i> menerima data sensor dari <i>node</i> (4,5)	

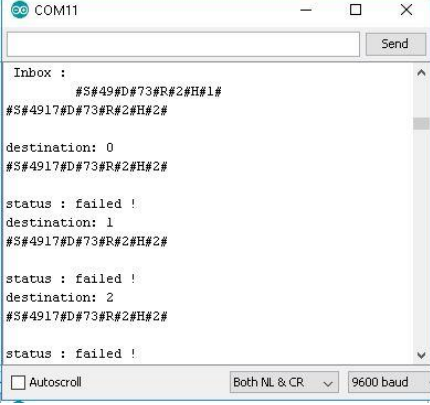
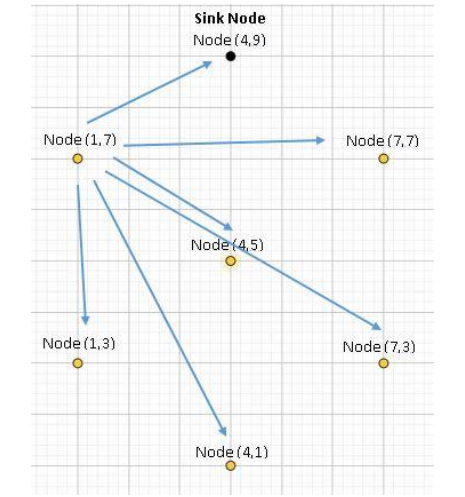
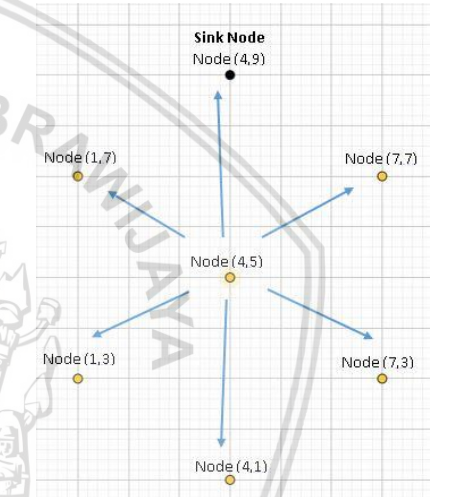
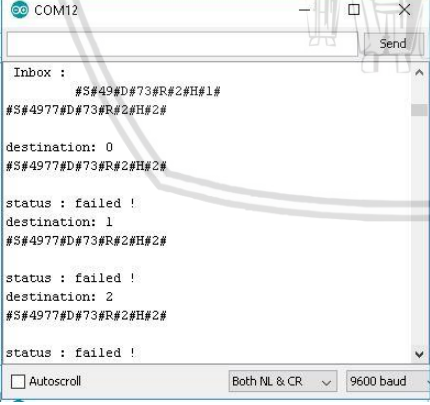
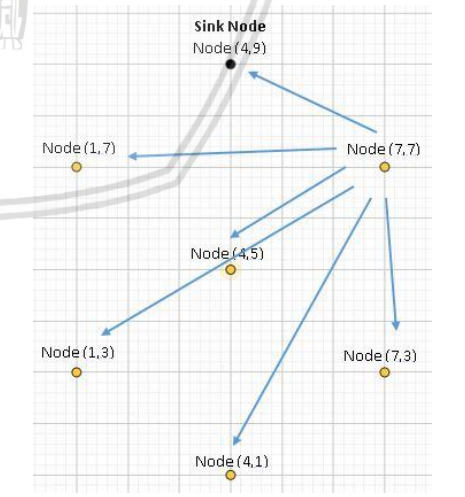
Berdasarkan hasil pengujian pengiriman data sensor suhu oleh *node* (4,5) yang dijabarkan pada Tabel 6.5., dapat dianalisis bahwa *sink node* dapat mengirimkan *interest* ke *node* tetangganya. Kemudian *node* tetangga yang lain yang berhasil

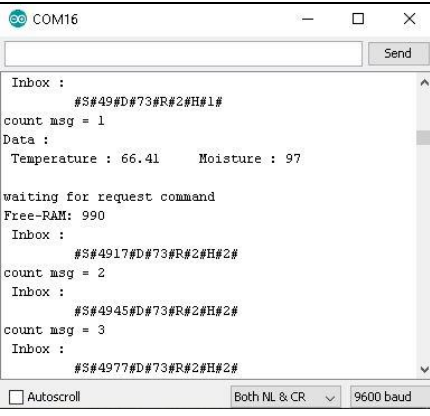
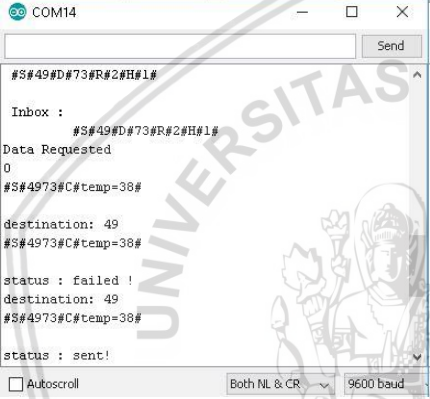
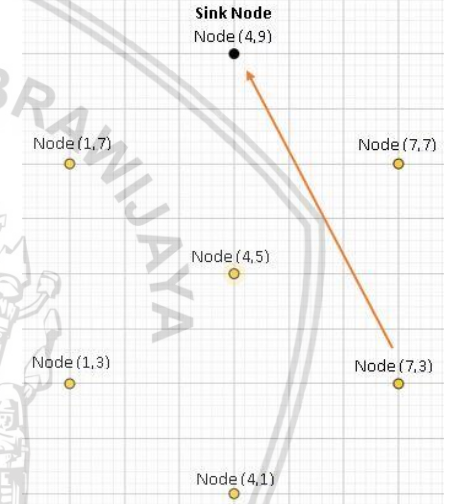
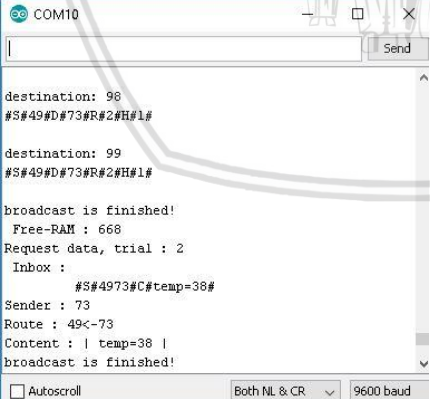
menerima *interest* meneruskan pesan dengan melakukan *broadcast* ke *node* tetangga lainnya. Lalu *node* (4,5) yang menerima *interest* dan merupakan tetangga *sink node* langsung mengirimkan balasan yang berisi data sensor suhu kepada *sink node*. *Sink node* berhasil menerima data sensor.

4. Pengiriman Data Sensor Suhu Oleh *Node* (7,3)

Tabel 6.6. Tabel Pengiriman Data Sensor Suhu Oleh *Node* (7,3)

Proses	Hasil	Skema Proses
1.	 Proses saat <i>sink node</i> broadcast <i>interest</i> ke seluruh <i>node</i> tetangganya.	
2.	 Proses saat <i>node</i> (1,3) menerima <i>interest</i> lalu broadcast ke seluruh <i>node</i> tetangganya.	

3.	 Proses saat <i>node</i> (1,7) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
4.	 Proses saat <i>node</i> (4,5) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
5.	 Proses saat <i>node</i> (7,7) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	

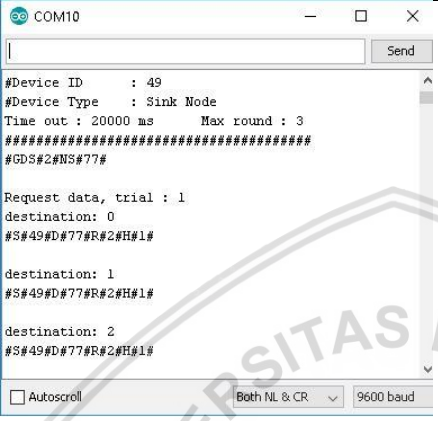
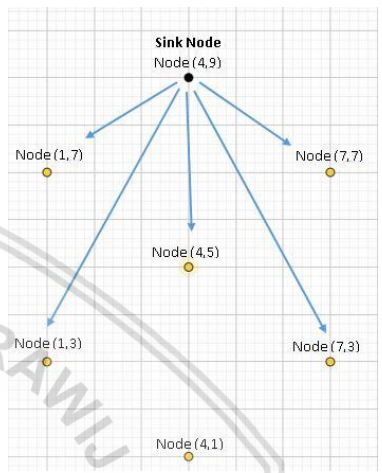
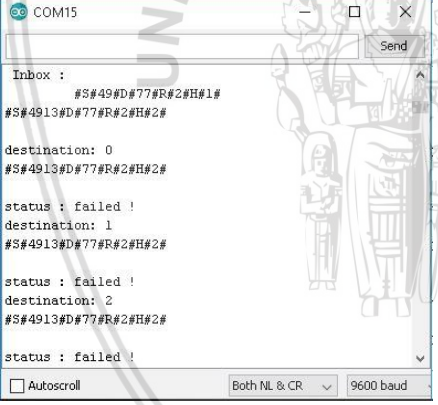
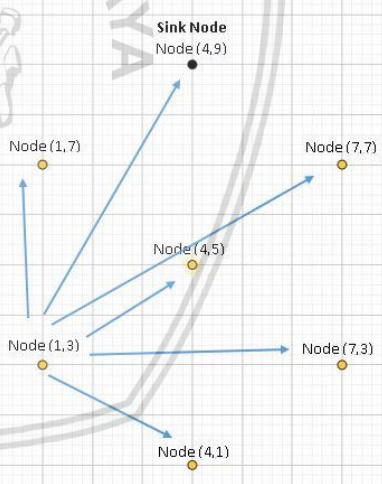
6.	 Proses saat <i>node</i> (4,1) menerima beberapa <i>interest</i>, namun tidak dieksekusi karena <i>node</i> tujuan buka dirinya sendiri.	
7.	 Proses saat <i>node</i> (7,3) menerima <i>interest</i> dan langsung membalas dengan mengirim data sensor ke <i>sink node</i>.	
8.	 Proses saat <i>sink node</i> menerima data sensor dari <i>node</i> (7,3)	

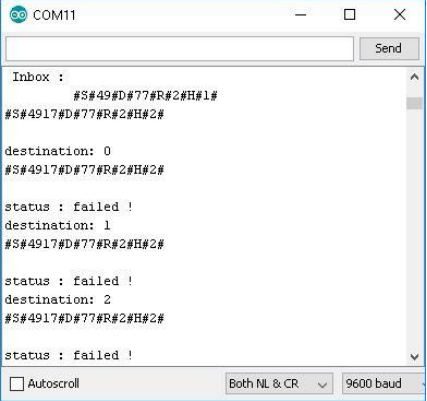
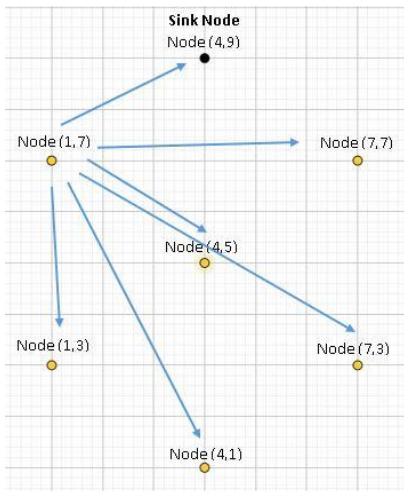
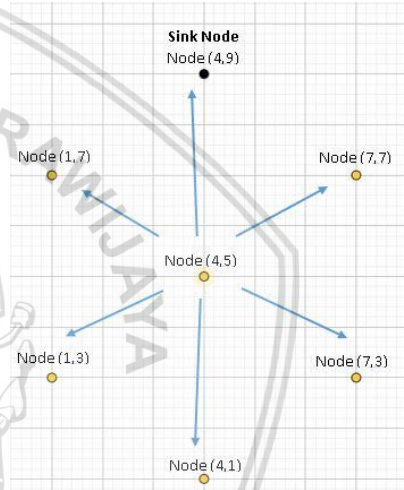
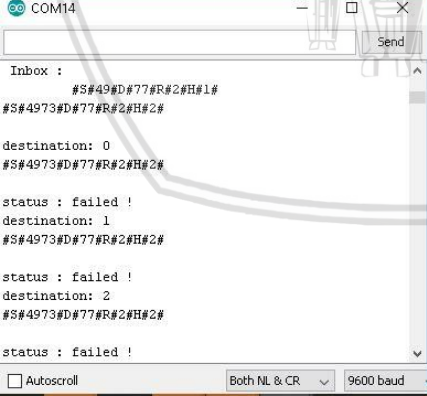
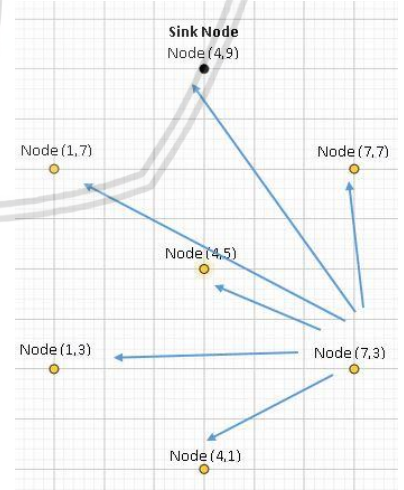
Berdasarkan hasil pengujian pengiriman data sensor suhu oleh *node* (7,3) yang dijabarkan pada Tabel 6.6., dapat dianalisis bahwa *sink node* dapat mengirimkan *interest* ke *node* tetangganya. Kemudian *node* tetangga yang lain yang berhasil menerima *interest* meneruskan pesan dengan melakukan *broadcast* ke *node*

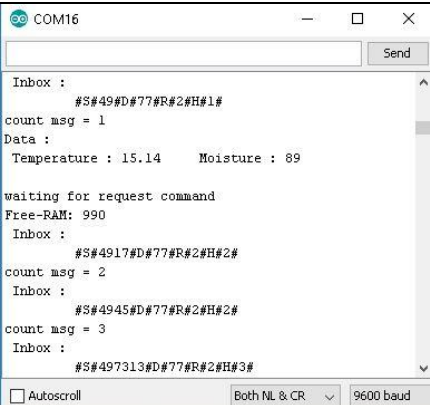
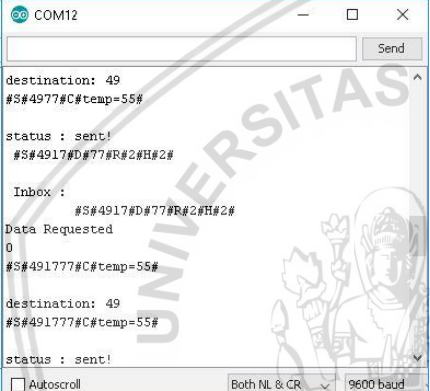
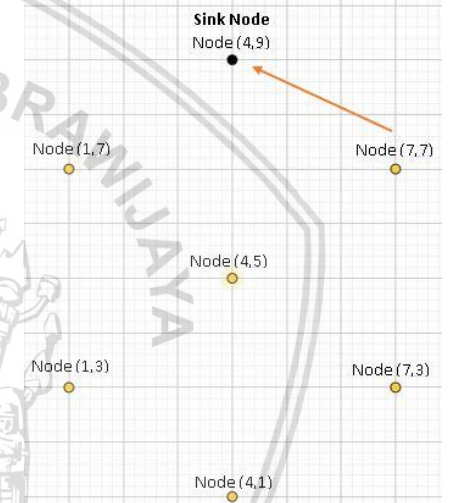
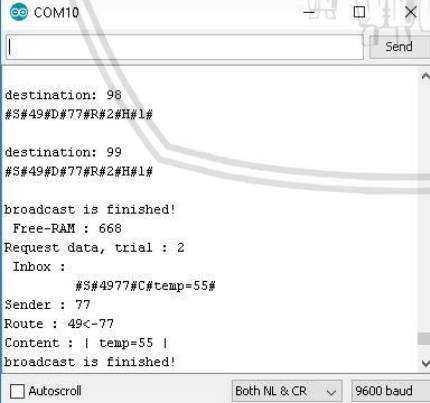
tetangga lainnya. Lalu *node* (7,3) yang menerima *interest* dan merupakan tetangga *sink node* langsung mengirimkan balasan yang berisi data sensor suhu kepada *sink node*. *Sink node* berhasil menerima data sensor.

5. Pengiriman Data Sensor Suhu Oleh *Node* (7,7)

Tabel 6.7. Tabel Pengiriman Data Sensor Suhu Oleh *Node* (7,7)

Proses	Hasil	Skema Proses
1.	<pre>#Device ID : 49 #Device Type : Sink Node Time out : 20000 ms Max round : 3 ##### #GDS#2#NS#77# Request data, trial : 1 destination: 0 #S#49#D#77#R#2#H#1# destination: 1 #S#49#D#77#R#2#H#1# destination: 2 #S#49#D#77#R#2#H#1#</pre> Proses saat <i>sink node</i> broadcast <i>interest</i> ke seluruh <i>node</i> tetangganya.	
2.	<pre>Inbox : #S#49#D#77#R#2#H#1# #S#49#D#77#R#2#H#2# destination: 0 #S#49#D#77#R#2#H#2# status : failed ! destination: 1 #S#49#D#77#R#2#H#2# status : failed ! destination: 2 #S#49#D#77#R#2#H#2# status : failed !</pre> Proses saat <i>node</i> (1,3) menerima <i>interest</i> lalu broadcast ke seluruh <i>node</i> tetangganya.	

3.	 Proses saat <i>node</i> (1,7) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
4.	 Proses saat <i>node</i> (4,5) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
5.	 Proses saat <i>node</i> (7,3) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	

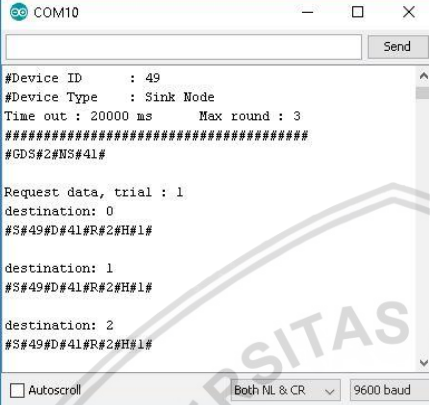
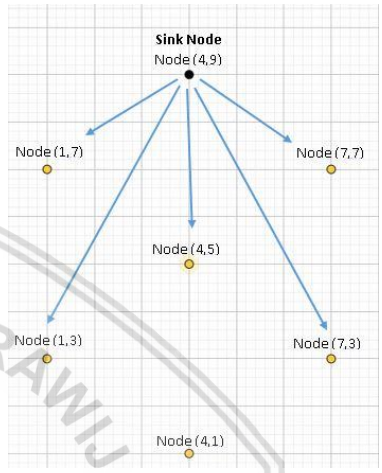
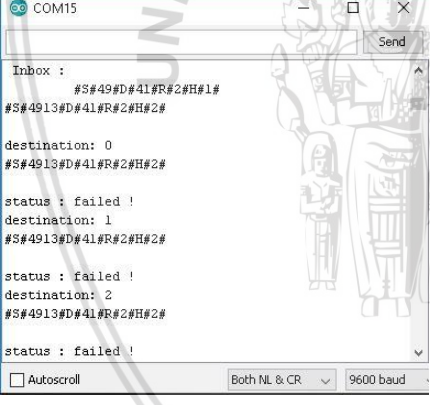
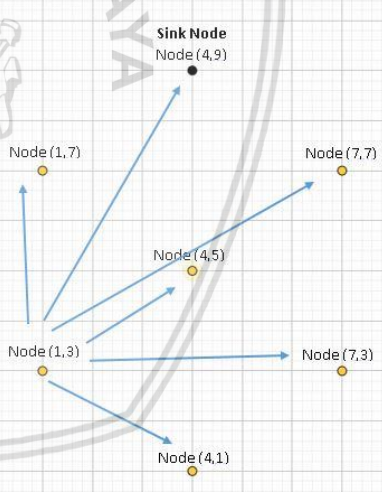
6.	 Proses saat <i>node</i> (4,1) menerima beberapa <i>interest</i>, namun tidak dieksekusi karena <i>node</i> tujuan buka dirinya sendiri.	
7.	 Proses saat <i>node</i> (7,7) menerima <i>interest</i> dan langsung membalas dengan mengirim data sensor ke <i>sink node</i>.	
8.	 Proses saat <i>sink node</i> menerima data sensor dari <i>node</i> (7,7)	

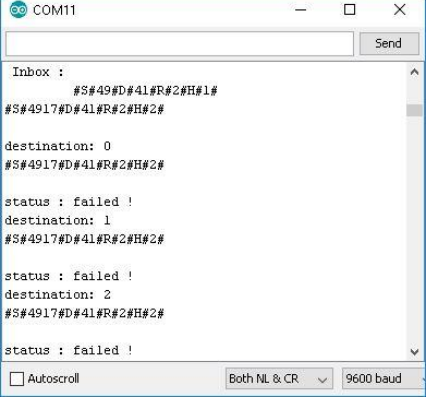
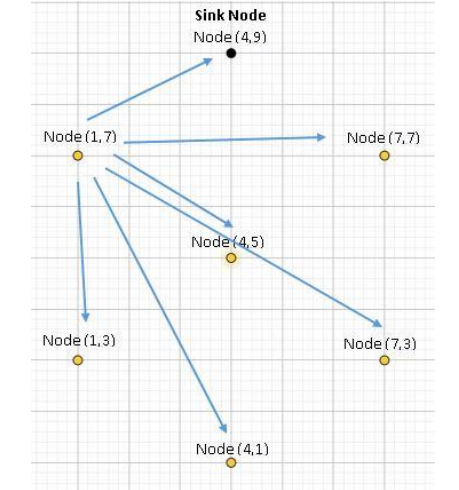
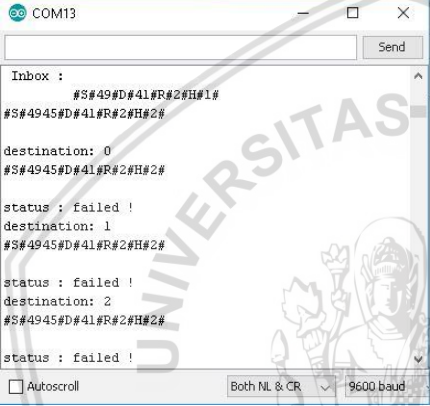
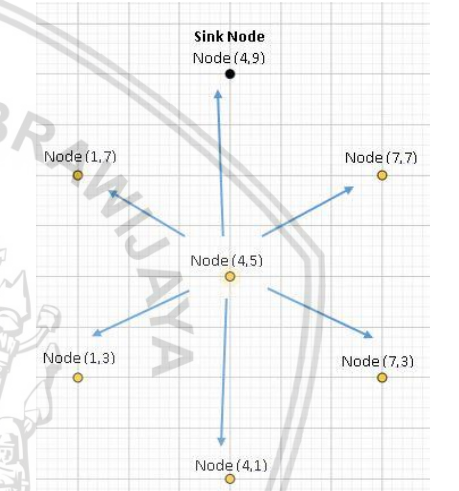
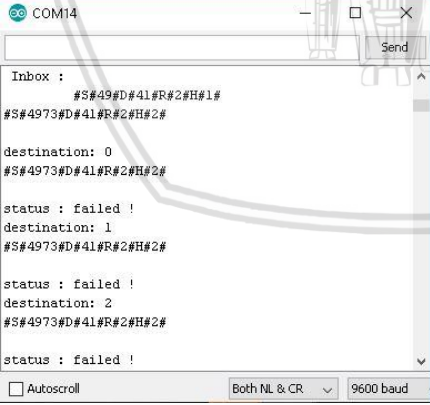
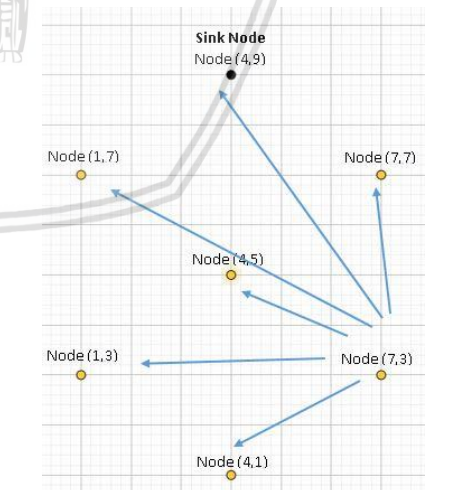
Berdasarkan hasil pengujian pengiriman data sensor suhu oleh *node* (7,7) yang dijabarkan pada Tabel 6.7., dapat dianalisis bahwa *sink node* dapat mengirimkan *interest* ke *node* tetangganya. Kemudian *node* tetangga yang lain yang berhasil menerima *interest* meneruskan pesan dengan melakukan *broadcast* ke *node*

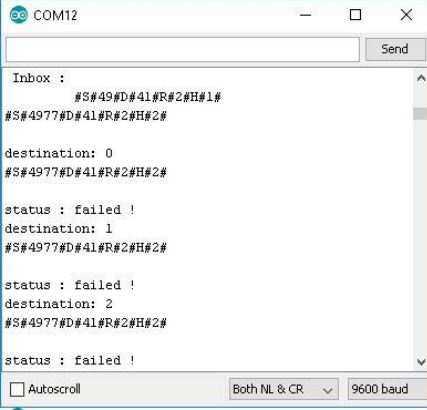
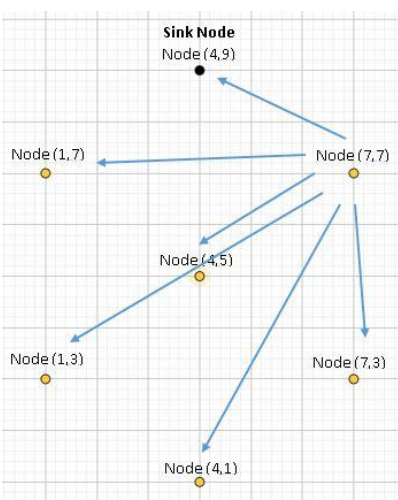
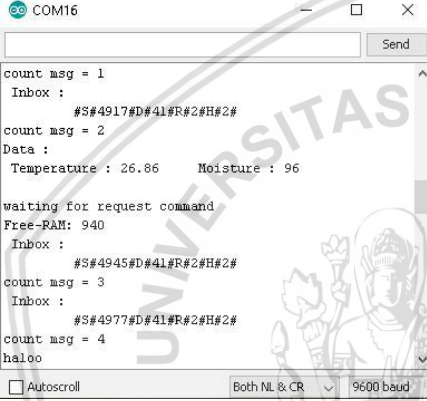
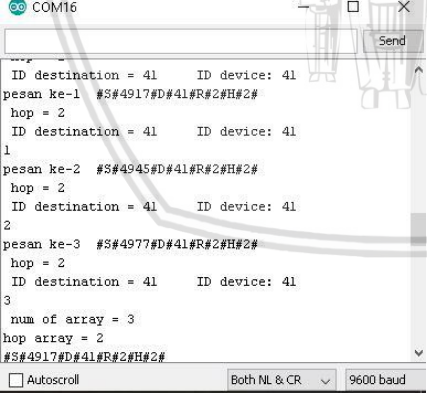
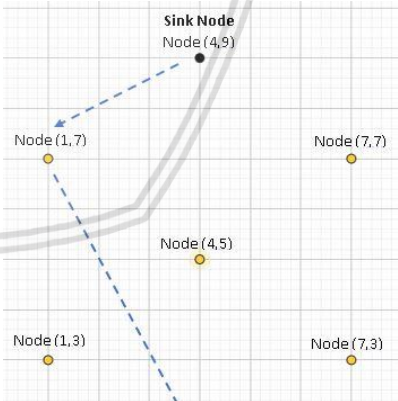
tetangga lainnya. Lalu *node* (7,7) yang menerima *interest* dan merupakan tetangga *sink node* langsung mengirimkan balasan yang berisi data sensor suhu kepada *sink node*. *Sink node* berhasil menerima data sensor.

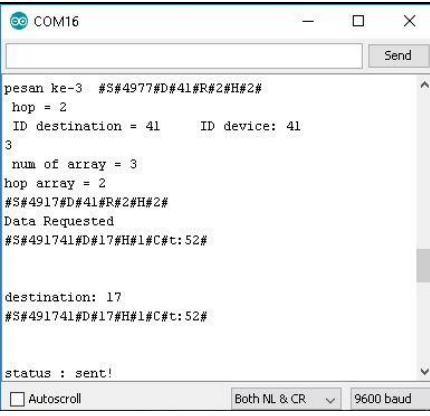
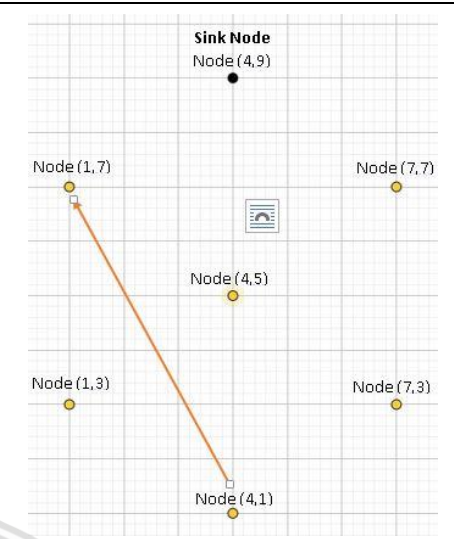
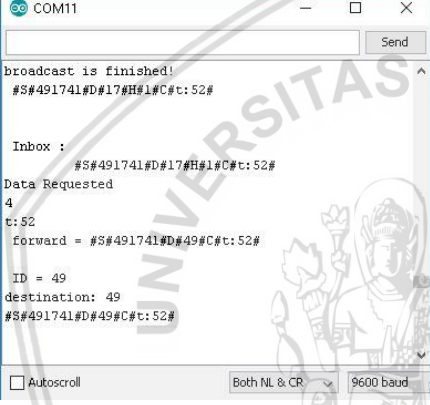
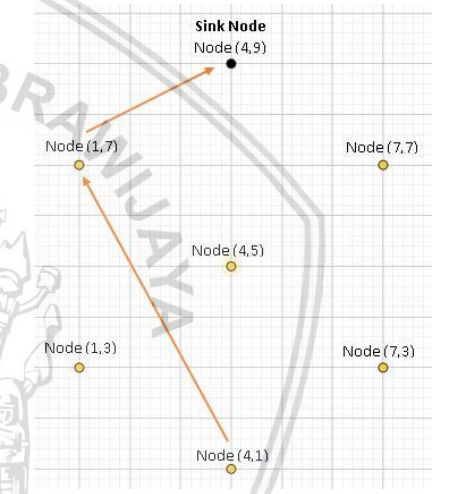
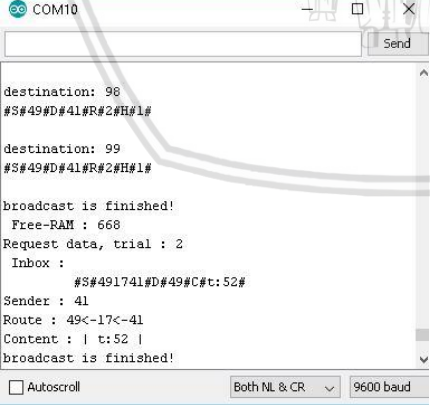
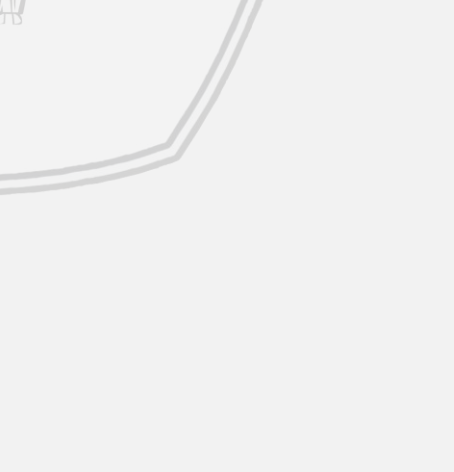
6. Pengiriman Data Sensor Suhu Oleh *Node* (4,1)

Tabel 6.8. Tabel Pengiriman Data Sensor Suhu Oleh *Node* (4,1)

Proses	Hasil	Skema Proses
1.	 <pre> COM10 #Device ID : 49 #Device Type : Sink Node Time out : 20000 ms Max round : 3 ##### #GDS#2#NS#41# Request data, trial : 1 destination: 0 #S#49#D#41#R#2#H#1# destination: 1 #S#49#D#41#R#2#H#1# destination: 2 #S#49#D#41#R#2#H#1# Autoscroll Both NL & CR 9600 baud </pre> Proses saat <i>sink node</i> broadcast <i>interest</i> ke seluruh <i>node</i> tetangganya.	 <pre> graph TD SinkNode["Sink Node Node (4,9)"] Node17["Node (1,7)"] Node77["Node (7,7)"] Node45["Node (4,5)"] Node13["Node (1,3)"] Node73["Node (7,3)"] Node41["Node (4,1)"] SinkNode --> Node17 SinkNode --> Node77 SinkNode --> Node45 SinkNode --> Node13 SinkNode --> Node73 SinkNode --> Node41 </pre>
2.	 <pre> COM15 Inbox : #S#49#D#41#R#2#H#1# #S#4913#D#41#R#2#H#2# destination: 0 #S#4913#D#41#R#2#H#2# status : failed ! destination: 1 #S#4913#D#41#R#2#H#2# status : failed ! destination: 2 #S#4913#D#41#R#2#H#2# status : failed ! Autoscroll Both NL & CR 9600 baud </pre> Proses saat <i>node</i> (1,3) menerima <i>interest</i> lalu broadcast ke seluruh <i>node</i> tetangganya.	 <pre> graph TD SinkNode["Sink Node Node (4,9)"] Node17["Node (1,7)"] Node77["Node (7,7)"] Node45["Node (4,5)"] Node13["Node (1,3)"] Node73["Node (7,3)"] Node41["Node (4,1)"] Node13 --> SinkNode Node13 --> Node77 Node13 --> Node45 Node13 --> Node73 Node13 --> Node41 </pre>

3.	 Proses saat <i>node</i> (1,7) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
4.	 Proses saat <i>node</i> (4,5) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
5.	 Proses saat <i>node</i> (7,3) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	

6.	 Proses saat <i>node</i> (7,7) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
7.	 Proses saat <i>node</i> (4,1) menerima <i>interest</i>.	
8.	 Proses saat <i>node</i> (4,1) melakukan sorting <i>interest</i> dengan jumlah hop terkecil.	

9.	 <pre> pesan ke-3 #S#4977#D#41#R#2#H#2# hop = 2 ID destination = 41 ID device: 41 3 num of array = 3 hop array = 2 #S#4917#D#41#R#2#H#2# Data Requested #S#491741#D#17#H#1#C#t:52# destination: 17 #S#491741#D#17#H#1#C#t:52# status : sent! Autoscroll Both NL & CR 9600 baud </pre> Proses saat <i>node</i> (4,1) mengirim data sensor ke <i>node</i> (1,7) karena <i>node</i> tersebut merupakan <i>node</i> pengirim <i>interest</i> sebelumnya.	 The diagram shows a network of nodes on a grid. Nodes are labeled: Sink Node (4,9), Node (1,7), Node (7,7), Node (4,5), Node (1,3), Node (7,3), and Node (4,1). An orange arrow points from Node (4,1) to Node (1,7), indicating the direction of data transmission.
10.	 <pre> broadcast is finished! #S#491741#D#17#H#1#C#t:52# Inbox : #S#491741#D#17#H#1#C#t:52# Data Requested 4 t:52 forward = #S#491741#D#49#C#t:52# ID = 49 destination: 49 #S#491741#D#49#C#t:52# Autoscroll Both NL & CR 9600 baud </pre> Proses saat <i>node</i> (1,7) menerima data sensor dan meneruskannya kepada <i>sink node</i>.	 The diagram shows the same network of nodes. In addition to the path from Node (4,1) to Node (1,7), a second orange arrow points from Node (1,7) to Sink Node (4,9), indicating the forwarding of data.
11.	 <pre> destination: 98 #S#49#D#41#R#2#H#1# destination: 99 #S#49#D#41#R#2#H#1# broadcast is finished! Free-RAM : 668 Request data, trial : 2 Inbox : #S#491741#D#49#C#t:52# Sender : 41 Route : 49<-17<-41 Content : t:52 broadcast is finished! Autoscroll Both NL & CR 9600 baud </pre> Proses saat <i>sink node</i> menerima data sensor dari <i>node</i> (1,7)	 The diagram shows the network with all nodes. The path of data transmission is complete, from Node (4,1) to Node (1,7) and finally to Sink Node (4,9).

Berdasarkan hasil pengujian pengiriman data sensor suhu oleh *node* (4,1) yang dijabarkan pada Tabel 6.8., dapat dianalisis bahwa *sink node* dapat mengirimkan *interest* ke *node* tetangganya. Kemudian *node* tetangga yang lain yang berhasil menerima *interest* meneruskan pesan dengan melakukan *broadcast* ke *node*

tetangga lainnya. Lalu *node* (4,1) yang menerima *interest* dan bukan merupakan tetangga *sink node* melakukan sorting *interest* untuk memilih jalur pengiriman data sensor suhu melalui hop terkecil. *Node* (4,1) berhasil mengirim data sensor suhu ke pengirim *interest* sebelumnya yaitu *node* (1,7). Saat *node* (1,7) berhasil menerima data sensor suhu, *node* (1,7) berhasil meneruskan data sensor ke *sink node* sebagai pengirim *interest* pertama kali.

Analisi hasil dari seluruh pengujian pengiriman data sensor suhu maka dapat disusun tabel sebagai berikut.

Tabel 6.9. Tabel Hasil Pengiriman Dan Penerimaan *Interest*

Node Pengirim <i>Interest</i>	Node Penerima <i>Interest</i>	Status Penerimaan <i>Interest</i>
(4,9)	(1,5)	Interest Diterima
	(1,7)	Interest Diterima
	(4,1)	Interest Diterima
	(4,5)	Interest Diterima
	(7,3)	Interest Diterima
	(7,7)	Interest Diterima

Tabel 6.10. Tabel Hasil Pengiriman Dan Penerimaan Data Sensor Suhu

Node Pengirim Data Sensor	Status Pengiriman Data Sensor	Status Penerimaan Data Sensor Oleh <i>Sink node</i>
(1,5)	Dikirim	Data Sensor Diterima
(1,7)	Dikirim	Data Sensor Diterima
(4,1)	Dikirim	Data Sensor Diterima
(4,5)	Dikirim	Data Sensor Diterima
(7,3)	Dikirim	Data Sensor Diterima
(7,7)	Dikirim	Data Sensor Diterima

Berdasarkan data yang diperoleh dari Tabel 6.9. dan Tabel 6.10. setiap *node* dapat merima *interest* dari *sink node* dapat membalas *interest* dan membalas *interest* dengan mengirimkan data sensor suhu kepada *sink node*.

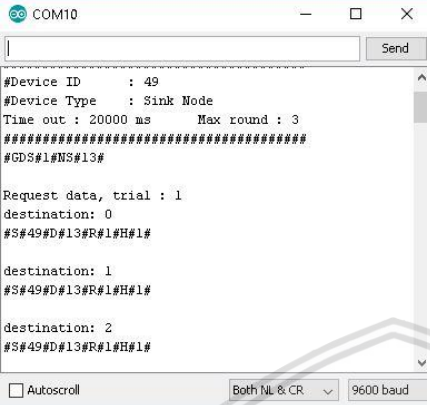
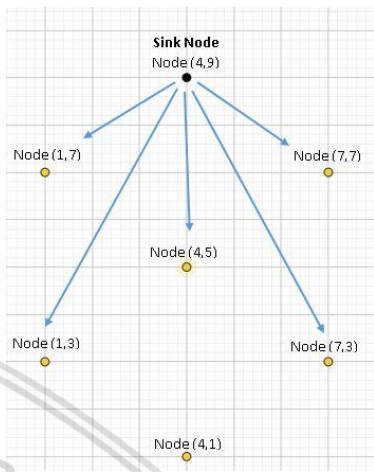
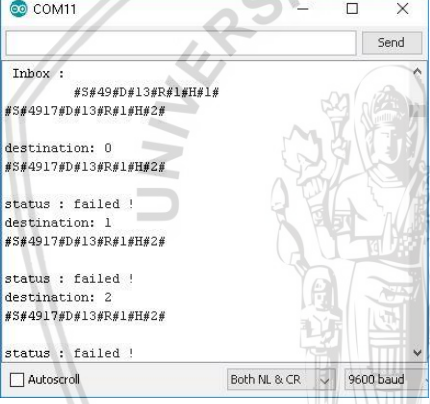
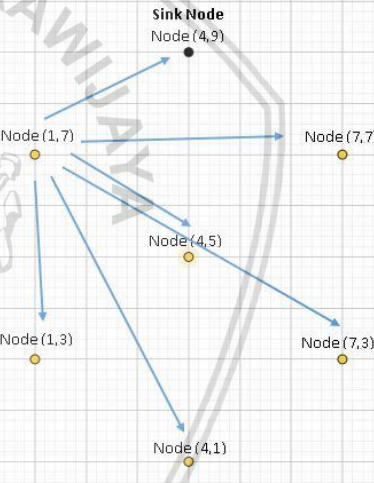
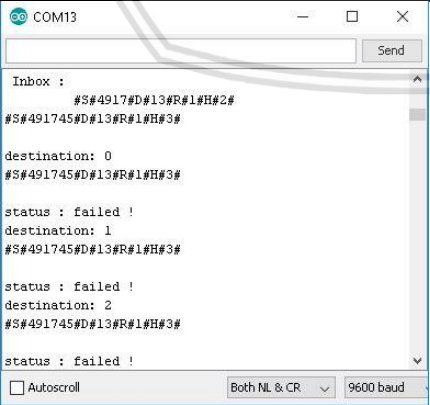
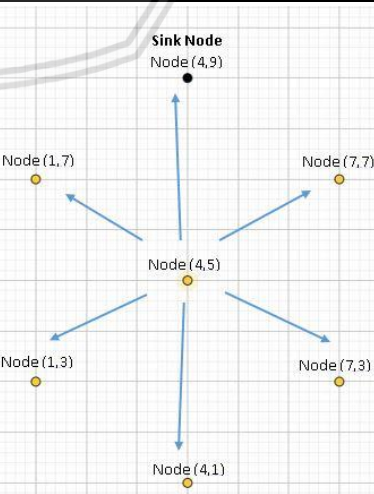
6.3.3.2 Hasil Dan Analisis Pengujian Pengiriman Data Sensor Kelembaban Tanah

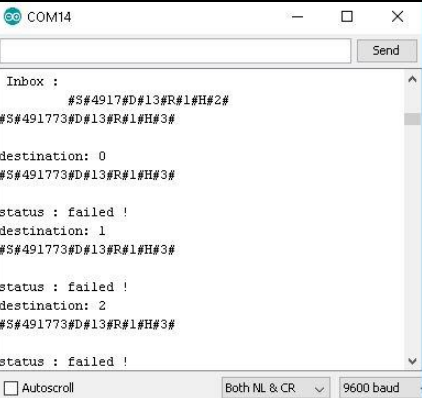
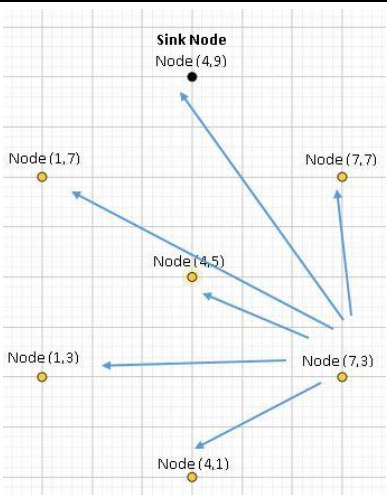
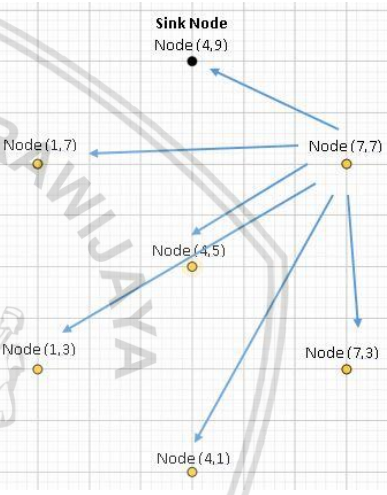
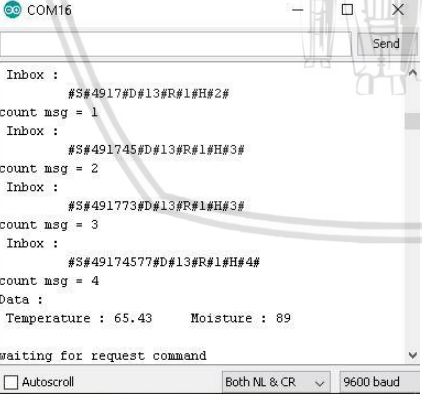
Pengujian fungsional pengiriman data sensor kelembaban tanah dilakukan dengan menguji semua *node* sensor untuk mengirimkan data sensor kelembaban tanah. Berikut hasil dari pengujian fungsional pengiriman data sensor kelembaban tanah.

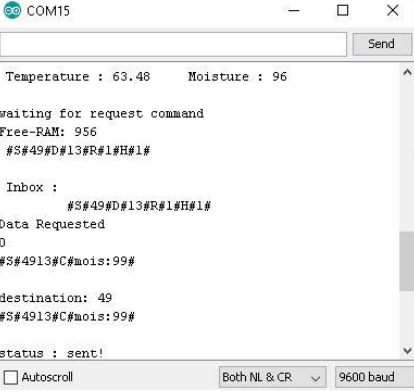
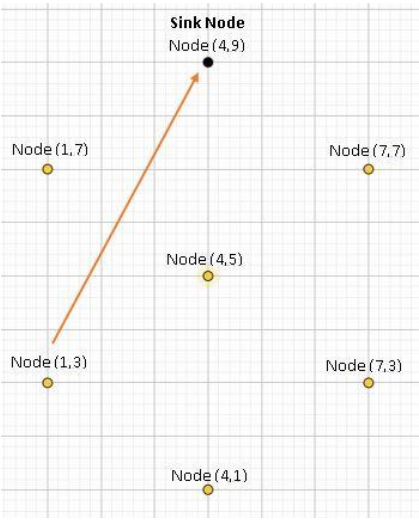
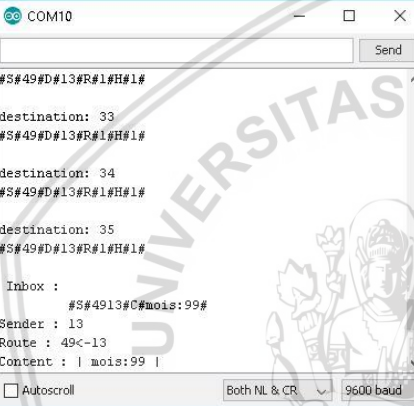
Berikut di bawah hasil pengujian fungsional pengiriman data sensor kelembaban tanah diantaranya.

1. Pengiriman Data Sensor Kelembaban Tanah Oleh *Node* (1,3)

Tabel 6.11. Pengiriman Data Sensor Kelembaban Tanah Oleh *Node* (1,3)

Proses	Hasil	Skema Proses
1.	 Proses saat <i>sink node</i> broadcast interest ke seluruh <i>node</i> tetangganya.	
2.	 Proses saat <i>node</i> (1,7) menerima <i>interest</i> lalu broadcast ke seluruh <i>node</i> tetangganya.	
3.	 Proses saat <i>node</i> (4,5) menerima <i>interest</i> lalu broadcast ke seluruh <i>node</i> tetangganya.	

4.	 Proses saat <i>node</i> (7,3) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
5.	 Proses saat <i>node</i> (7,7) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
6.	 Proses saat <i>node</i> (4,1) menerima beberapa <i>interest</i>, namun tidak dieksekusi karena <i>node</i> tujuan buka dirinya sendiri.	

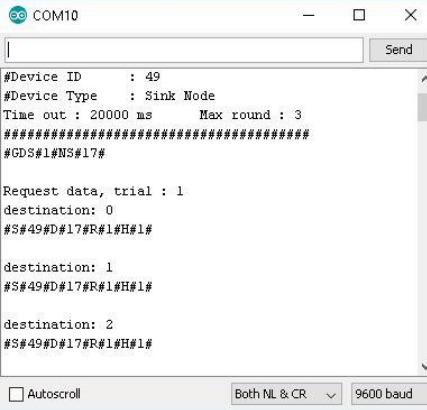
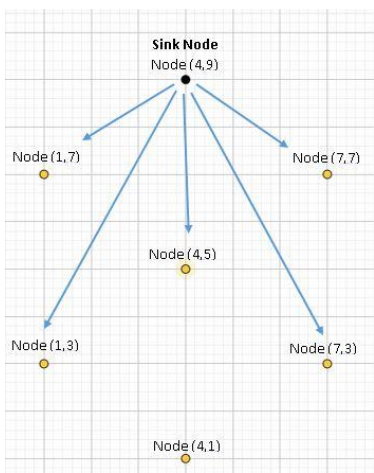
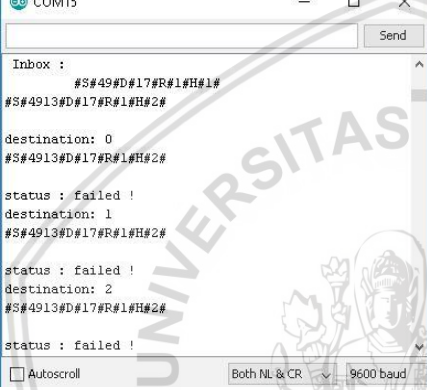
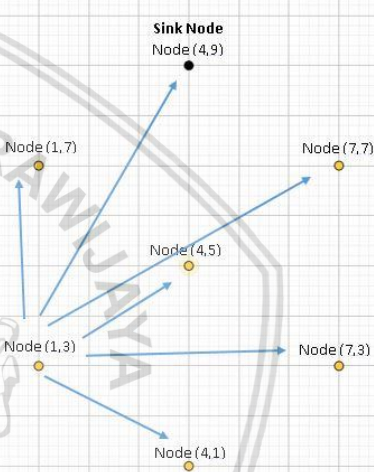
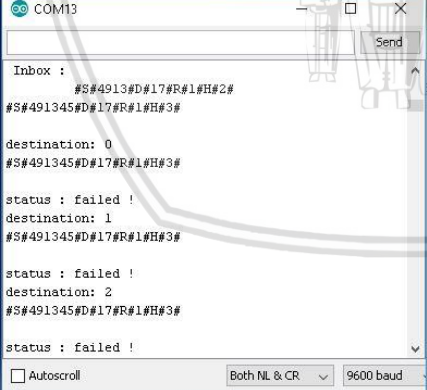
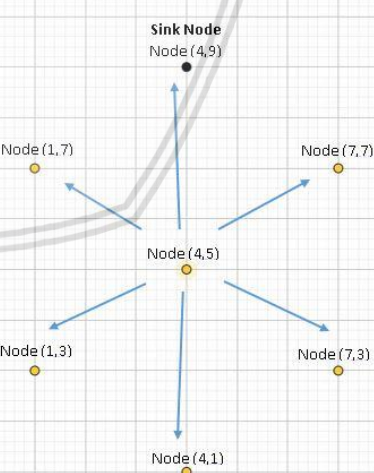
7.	 Proses saat <i>node</i> (1,3) menerima <i>interest</i> dan langsung membalas dengan mengirim data sensor ke <i>sink node</i>.	
8.	 Proses saat <i>sink node</i> menerima data sensor dari <i>node</i> (1,3)	

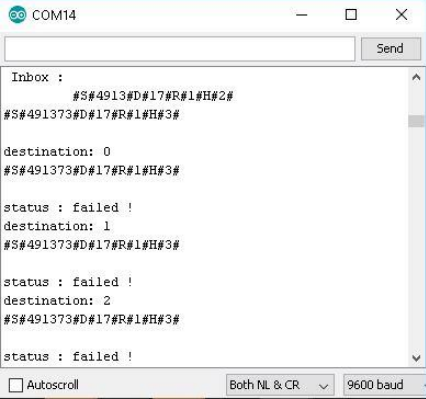
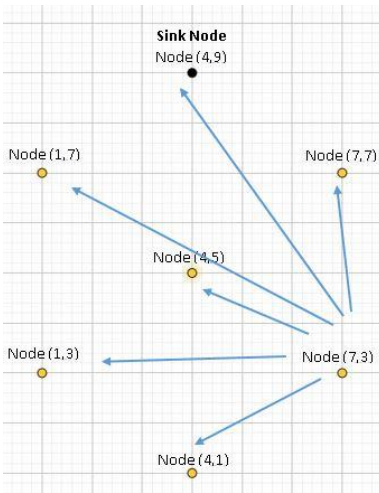
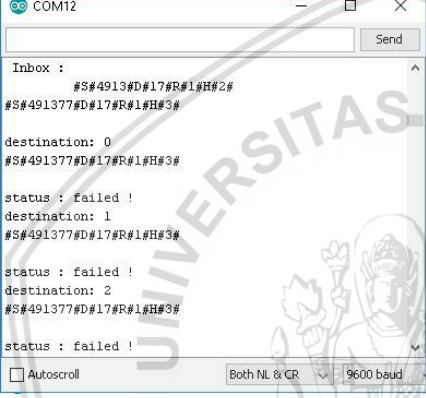
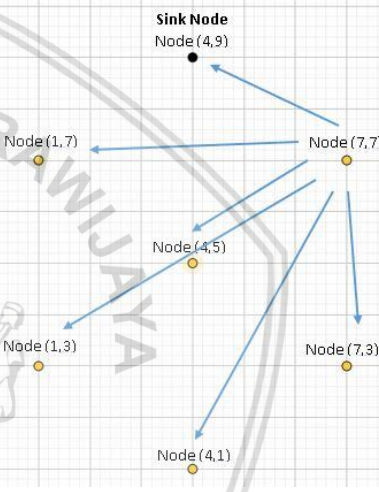
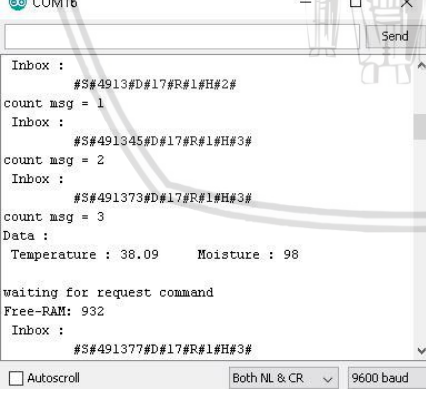
Berdasarkan hasil pengujian pengiriman data sensor kelembaban tanah oleh *node* (1,3) yang dijabarkan pada Tabel 6.11., dapat dianalisis bahwa *sink node* dapat mengirimkan *interest* ke *node* tetangganya. Kemudian *node* tetangga yang lain yang berhasil menerima *interest* meneruskan pesan dengan melakukan *broadcast* ke *node* tetangga lainnya. Lalu *node* (1,3) yang menerima *interest* dan merupakan tetangga *sink node* langsung mengirimkan balasan yang berisi data sensor kelembaban tanah kepada *sink node*. *Sink node* berhasil menerima data sensor.

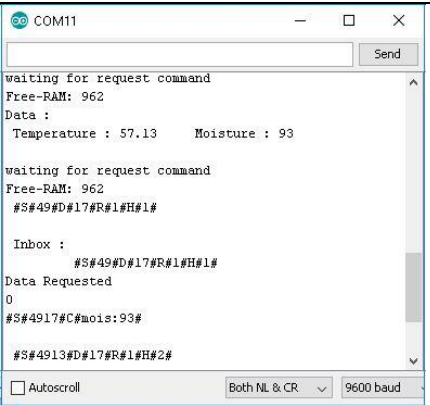
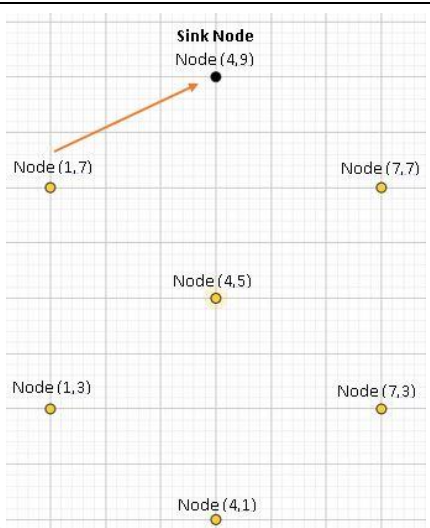
2. Pengiriman Data Sensor Kelembaban Tanah Oleh *Node* (1,7)

Tabel 6.12. Pengiriman Data Sensor Kelembaban Tanah *Node* (1,7)

Proses	Hasil	Skema Proses
--------	-------	--------------

1.	 COM10 <pre>#Device ID : 49 #Device Type : Sink Node Time out : 20000 ms Max round : 3 ##### #GDS#1#NS#17# Request data, trial : 1 destination: 0 #S#49#D#17#R#1#H#1# destination: 1 #S#49#D#17#R#1#H#1# destination: 2 #S#49#D#17#R#1#H#1#</pre> Proses saat <i>sink node</i> broadcast interest ke seluruh <i>node</i> tetangganya.	
2.	 COM15 <pre>Inbox : #S#49#D#17#R#1#H#1# #S#4913#D#17#R#1#H#2# destination: 0 #S#4913#D#17#R#1#H#2# status : failed ! destination: 1 #S#4913#D#17#R#1#H#2# status : failed ! destination: 2 #S#4913#D#17#R#1#H#2# status : failed !</pre> Proses saat <i>node</i> (1,3) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
3.	 COM13 <pre>Inbox : #S#4913#D#17#R#1#H#2# #S#491345#D#17#R#1#H#3# destination: 0 #S#491345#D#17#R#1#H#3# status : failed ! destination: 1 #S#491345#D#17#R#1#H#3# status : failed ! destination: 2 #S#491345#D#17#R#1#H#3# status : failed !</pre> Proses saat <i>node</i> (4,5) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	

4.	 Proses saat <i>node</i> (7,3) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
5.	 Proses saat <i>node</i> (7,7) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
6.	 Proses saat <i>node</i> (4,1) menerima beberapa <i>interest</i>, namun tidak dieksekusi karena <i>node</i> tujuan buka dirinya sendiri.	

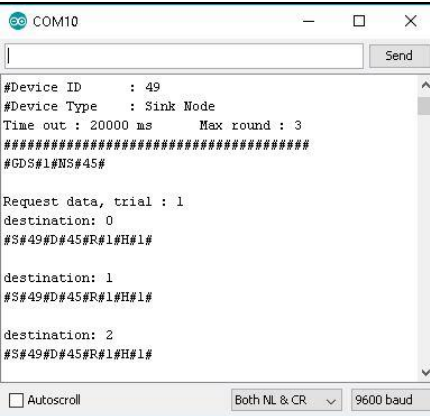
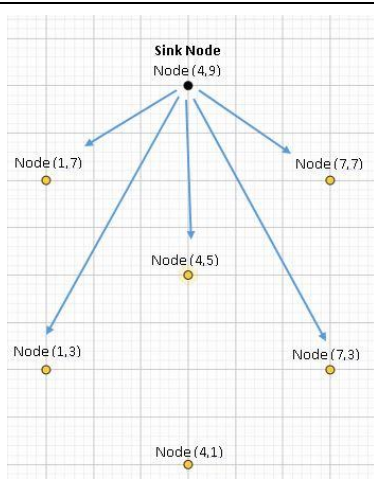
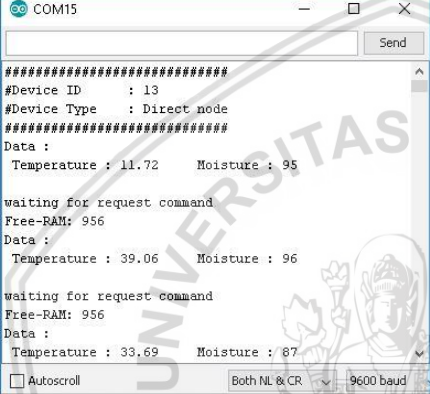
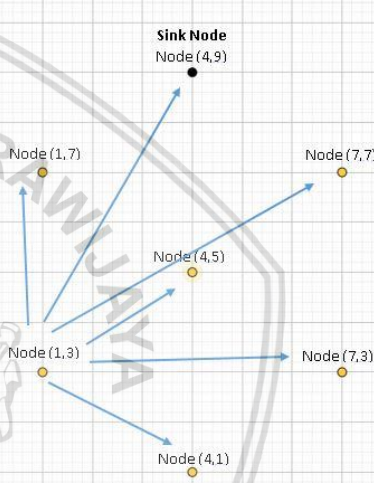
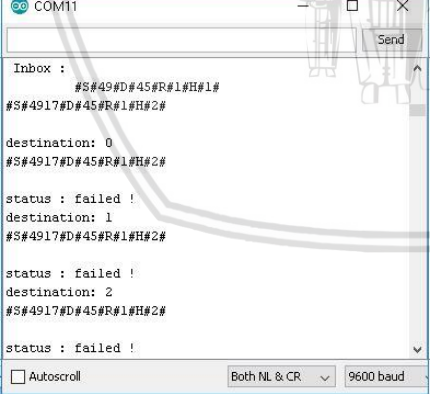
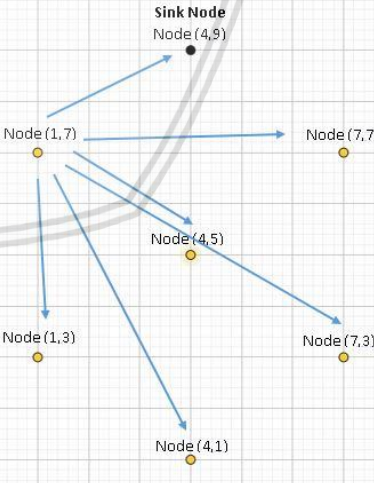
7.	 Proses saat <i>node</i> (1,7) menerima <i>interest</i> dan langsung membalas dengan mengirim data sensor ke <i>sink node</i>.	
8.	 Proses saat <i>sink node</i> menerima data sensor dari <i>node</i> (1,7)	

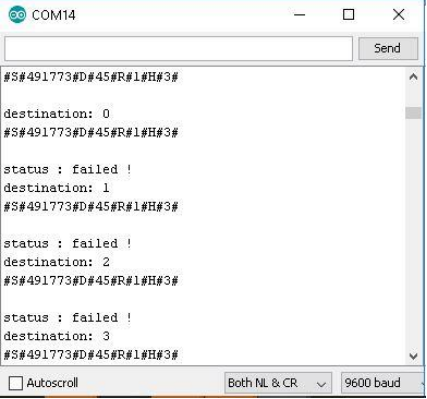
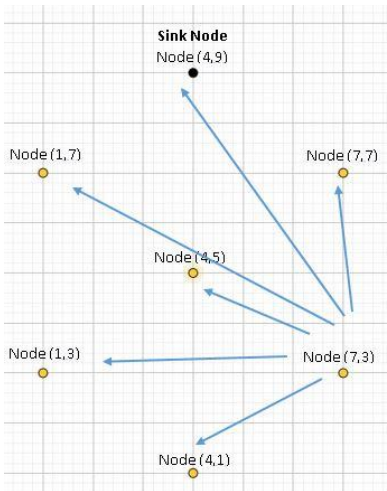
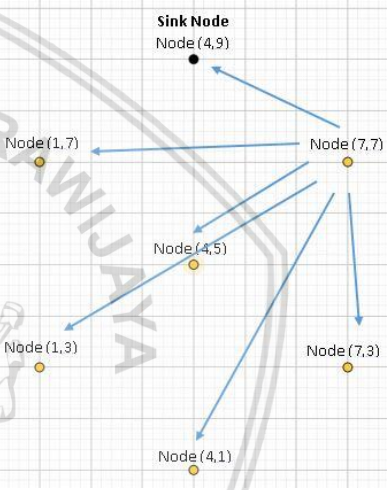
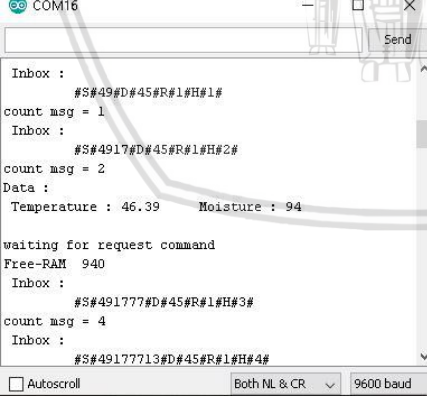
Berdasarkan hasil pengujian pengiriman data sensor kelembaban tanah oleh *node* (1,7) yang dijabarkan pada Tabel 6.12., dapat dianalisis bahwa *sink node* dapat mengirimkan *interest* ke *node* tetangganya. Kemudian *node* tetangga yang lain yang berhasil menerima *interest* meneruskan pesan dengan melakukan *broadcast* ke *node* tetangga lainnya. Lalu *node* (1,7) yang menerima *interest* dan merupakan tetangga *sink node* langsung mengirimkan balasan yang berisi data sensor kelembaban tanah kepada *sink node*. *Sink node* berhasil menerima data sensor.

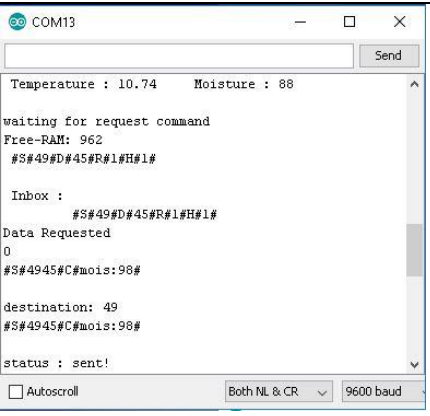
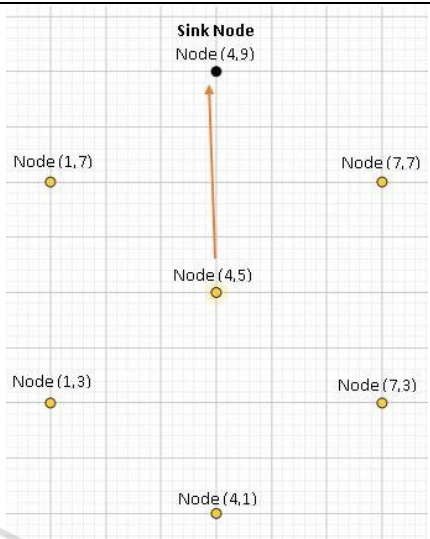
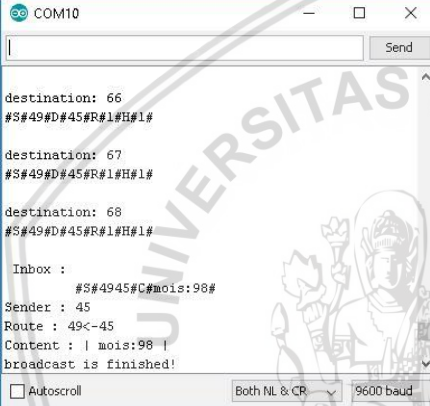
3. Pengiriman Data Sensor Kelembaban Tanah Oleh *Node* (4,5)

Tabel 6.13. Pengiriman Data Sensor Kelembaban Tanah Oleh *Node* (4,5)

Proses	Hasil	Skema Proses
--------	-------	--------------

1.	 <pre> COM10 #Device ID : 49 #Device Type : Sink Node Time out : 20000 ms Max round : 3 ##### #GDS#1#NS#45# Request data, trial : 1 destination: 0 #S#49#D#45#R#1#H#1# destination: 1 #S#49#D#45#R#1#H#1# destination: 2 #S#49#D#45#R#1#H#1# Autoscroll Both NL & CR 9600 baud </pre> Proses saat <i>sink node</i> broadcast interest ke seluruh <i>node</i> tetangganya.	
2.	 <pre> COM15 ##### #Device ID : 13 #Device Type : Direct node ##### Data : Temperature : 11.72 Moisture : 95 waiting for request command Free-RAM: 956 Data : Temperature : 39.06 Moisture : 96 waiting for request command Free-RAM: 956 Data : Temperature : 33.69 Moisture : 87 Autoscroll Both NL & CR 9600 baud </pre> Proses saat <i>node</i> (1,3) menerima interest lalu broadcast ke seluruh <i>node</i> tetangganya.	
3.	 <pre> COM11 Inbox : #S#49#D#45#R#1#H#1# #S#4917#D#45#R#1#H#2# destination: 0 #S#4917#D#45#R#1#H#2# status : failed ! destination: 1 #S#4917#D#45#R#1#H#2# status : failed ! destination: 2 #S#4917#D#45#R#1#H#2# status : failed ! Autoscroll Both NL & CR 9600 baud </pre> Proses saat <i>node</i> (1,7) menerima interest lalu broadcast ke seluruh <i>node</i> tetangganya.	

4.	 Proses saat <i>node</i> (7,3) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
5.	 Proses saat <i>node</i> (7,7) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
6.	 Proses saat <i>node</i> (4,1) menerima beberapa <i>interest</i>, namun tidak dieksekusi karena <i>node</i> tujuan buka dirinya sendiri.	

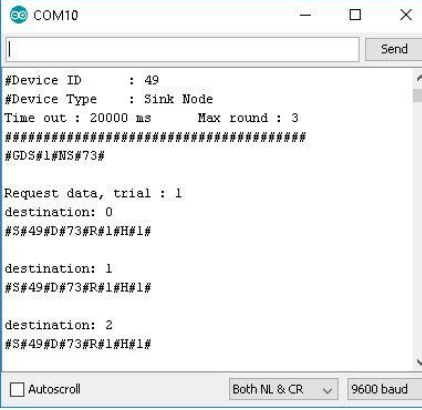
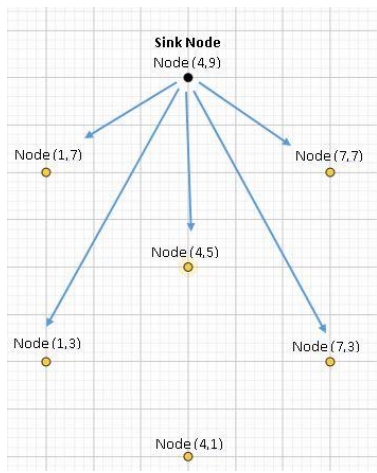
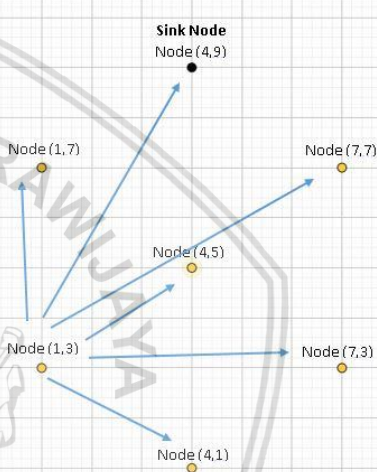
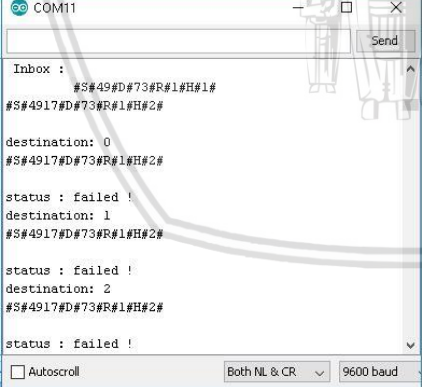
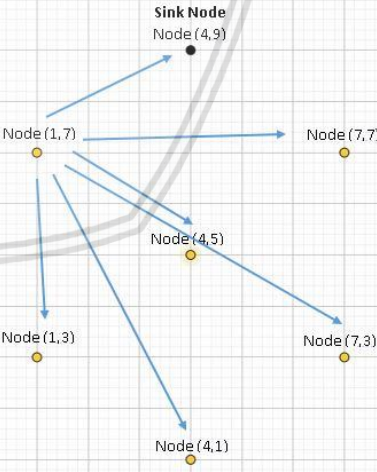
7.	 Proses saat <i>node</i> (4,5) menerima <i>interest</i> dan langsung membalas dengan mengirim data sensor ke <i>sink node</i>.	
8.	 Proses saat <i>sink node</i> menerima data sensor dari <i>node</i> (4,5)	

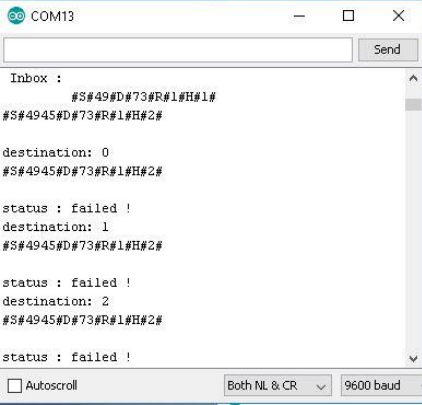
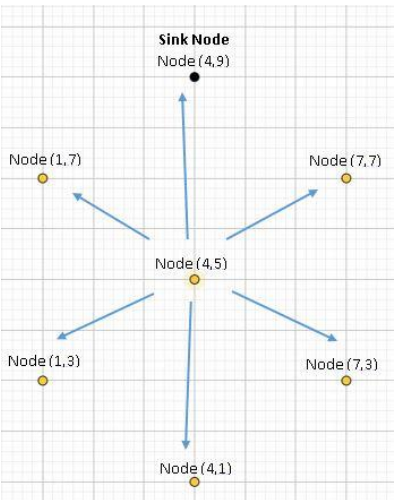
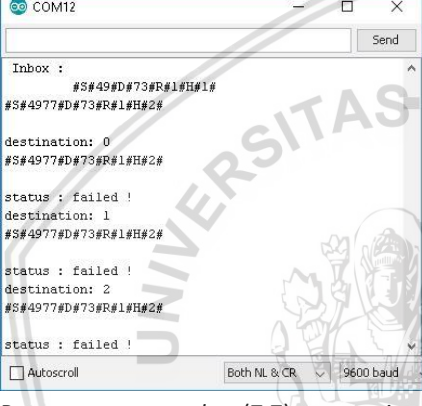
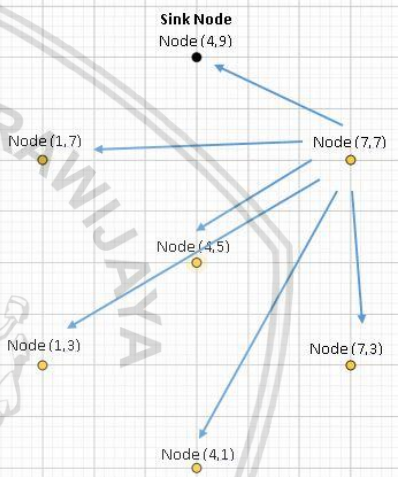
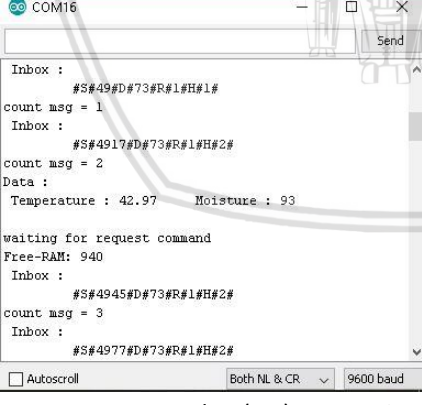
Berdasarkan hasil pengujian pengiriman data sensor kelembaban tanah oleh *node* (4,5) yang dijabarkan pada Tabel 6.13., dapat dianalisis bahwa *sink node* dapat mengirimkan *interest* ke *node* tetangganya. Kemudian *node* tetangga yang lain yang berhasil menerima *interest* meneruskan pesan dengan melakukan *broadcast* ke *node* tetangga lainnya. Lalu *node* (4,5) yang menerima *interest* dan merupakan tetangga *sink node* langsung mengirimkan balasan yang berisi data sensor kelembaban tanah kepada *sink node*. *Sink node* berhasil menerima data sensor.

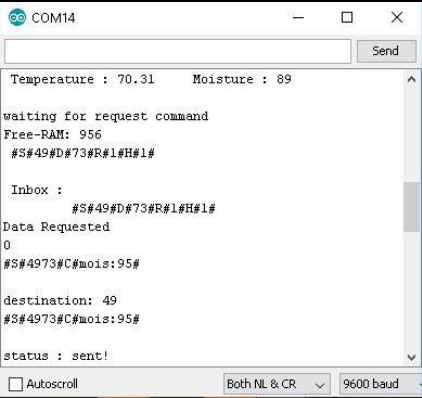
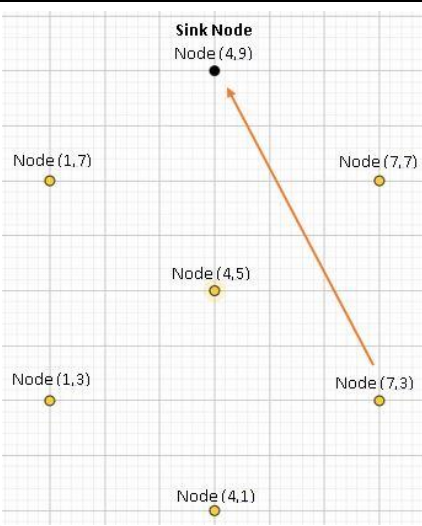
4. Pengiriman Data Sensor Kelembaban Tanah Oleh *Node* (7,3)

Tabel 6.14. Pengiriman Data Sensor Kelembaban Tanah Oleh *Node* (7,3)

Proses	Hasil	Skema Proses
--------	-------	--------------

1.	 Proses saat <i>sink node</i> broadcast interest ke seluruh <i>node</i> tetangganya.	
2.	 Proses saat <i>node</i> (1,3) menerima <i>interest</i> lalu broadcast ke seluruh <i>node</i> tetangganya.	
3.	 Proses saat <i>node</i> (1,7) menerima <i>interest</i> lalu broadcast ke seluruh <i>node</i> tetangganya.	

4.	 Proses saat <i>node</i> (4,5) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
5.	 Proses saat <i>node</i> (7,7) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
6.	 Proses saat <i>node</i> (4,1) menerima beberapa <i>interest</i>, namun tidak dieksekusi karena <i>node</i> tujuan buka dirinya sendiri.	

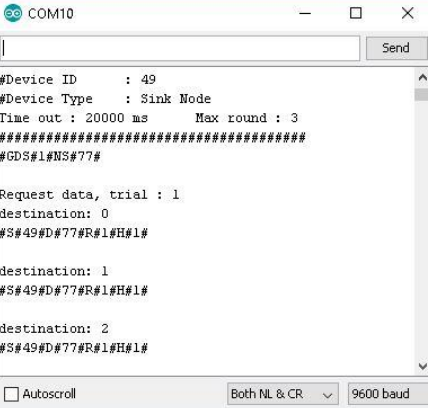
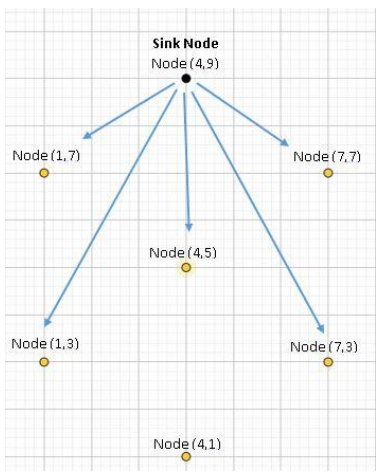
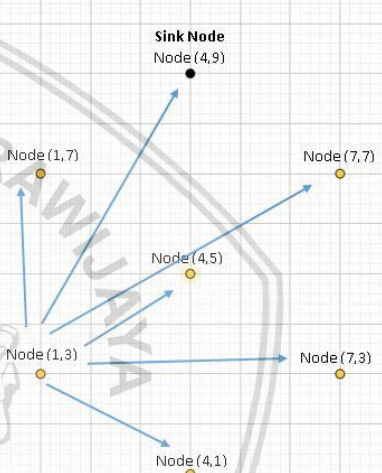
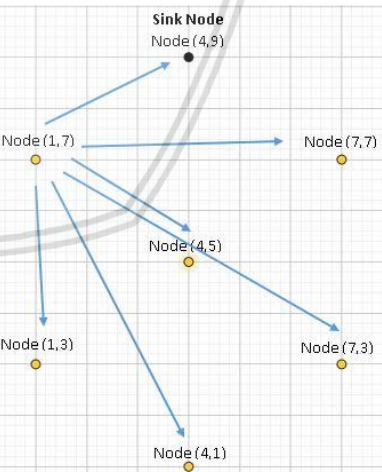
7.	 Proses saat <i>node</i> (7,3) menerima <i>interest</i> dan langsung membalas dengan mengirim data sensor ke <i>sink node</i>.	
8.	 Proses saat <i>sink node</i> menerima data sensor dari <i>node</i> (7,3)	

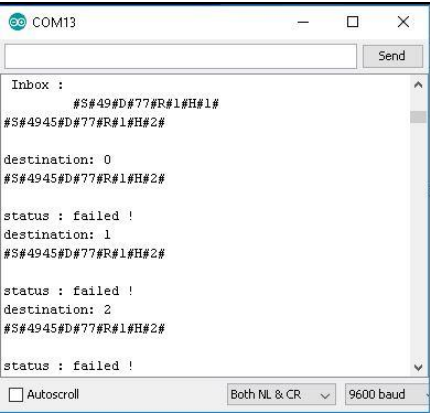
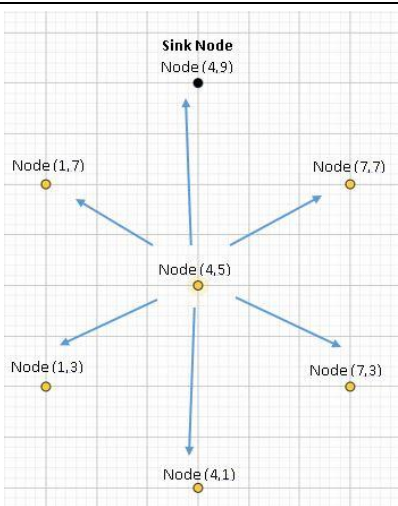
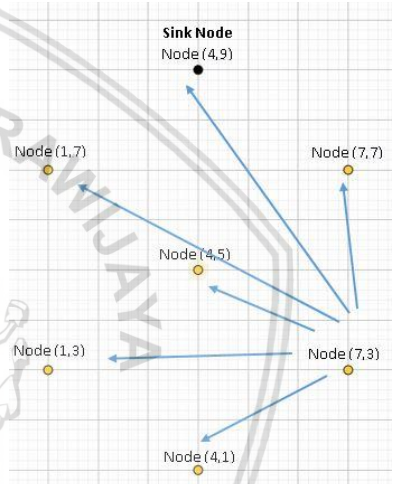
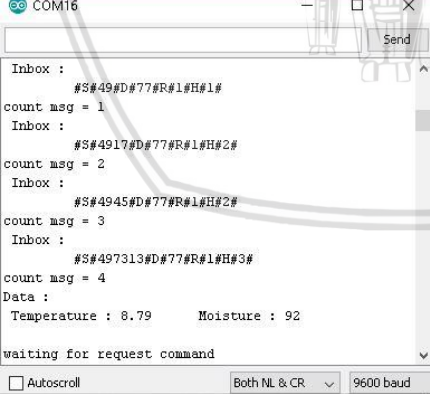
Berdasarkan hasil pengujian pengiriman data sensor kelembaban tanah oleh *node* (7,3) yang dijabarkan pada Tabel 6.14., dapat dianalisis bahwa *sink node* dapat mengirimkan *interest* ke *node* tetangganya. Kemudian *node* tetangga yang lain yang berhasil menerima *interest* meneruskan pesan dengan melakukan *broadcast* ke *node* tetangga lainnya. Lalu *node* (7,3) yang menerima *interest* dan merupakan tetangga *sink node* langsung mengirimkan balasan yang berisi data sensor kelembaban tanah kepada *sink node*. *Sink node* berhasil menerima data sensor.

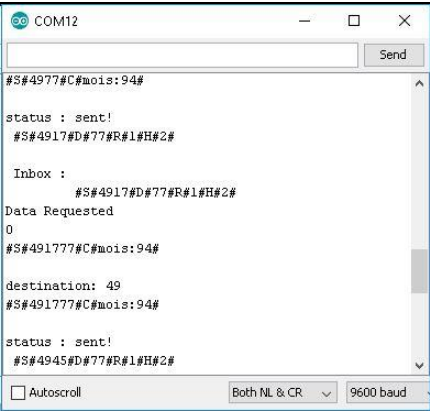
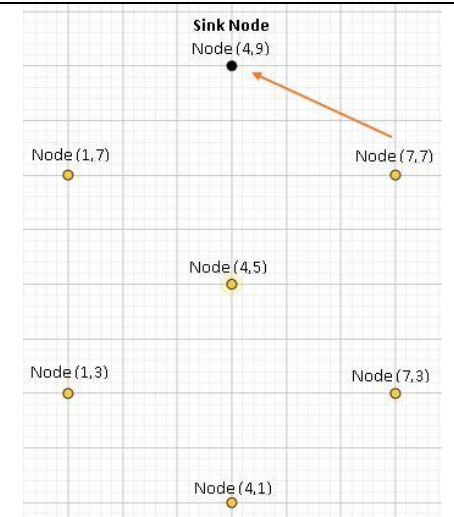
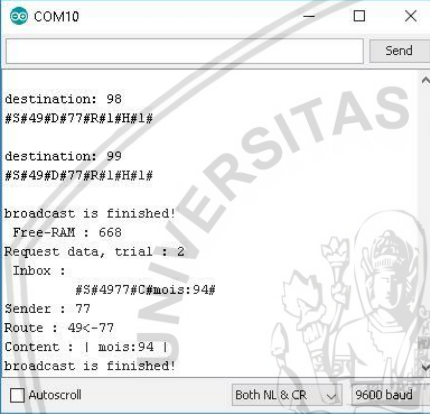
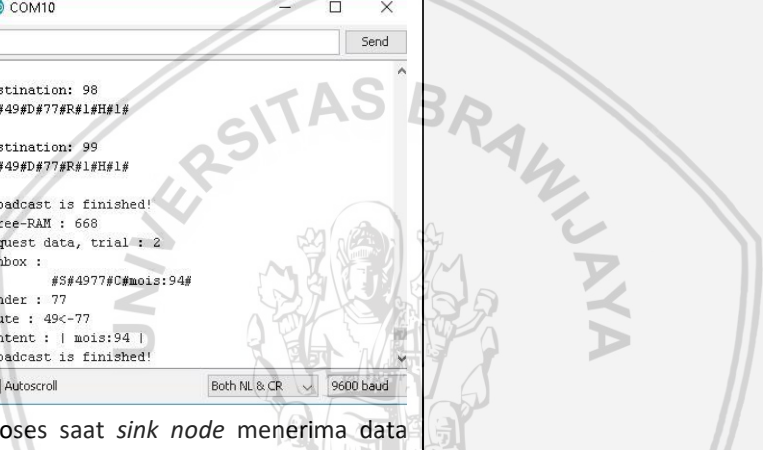
5. Pengiriman Data Sensor Kelembaban Tanah Oleh *Node* (7,7)

Tabel 6.15. Pengiriman Data Sensor Kelembaban Tanah Oleh *Node* (7,7)

Proses	Hasil	Skema Proses
--------	-------	--------------

1.	 Proses saat <i>sink node</i> broadcast interest ke seluruh <i>node</i> tetangganya.	
2.	 Proses saat <i>node</i> (1,3) menerima interest lalu broadcast ke seluruh <i>node</i> tetangganya.	
3.	 Proses saat <i>node</i> (1,7) menerima interest lalu broadcast ke seluruh <i>node</i> tetangganya.	

4.	 Proses saat <i>node</i> (4,5) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
5.	 Proses saat <i>node</i> (7,3) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
6.	 Proses saat <i>node</i> (4,1) menerima beberapa <i>interest</i>, namun tidak dieksekusi karena <i>node</i> tujuan buka dirinya sendiri.	

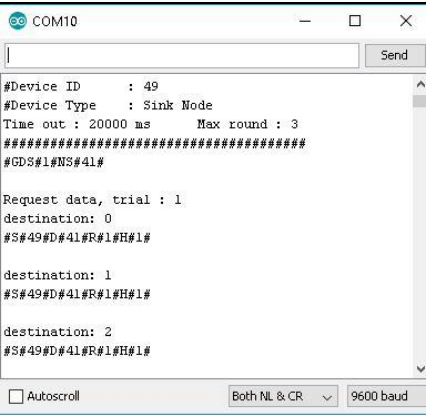
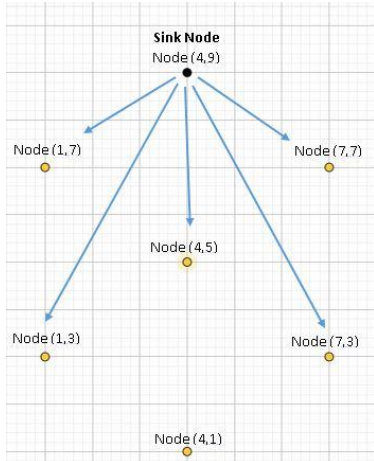
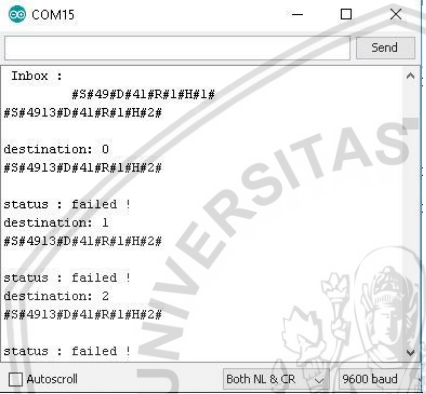
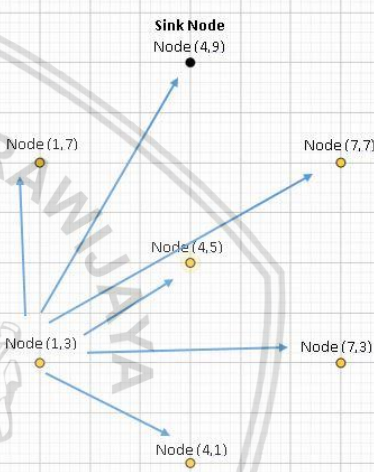
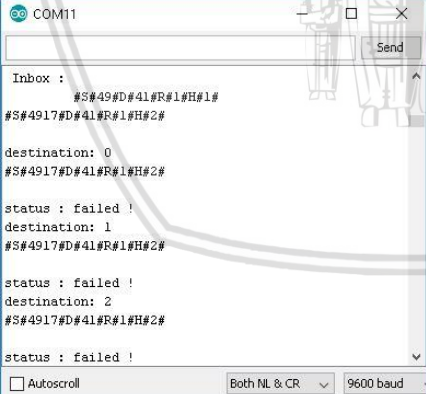
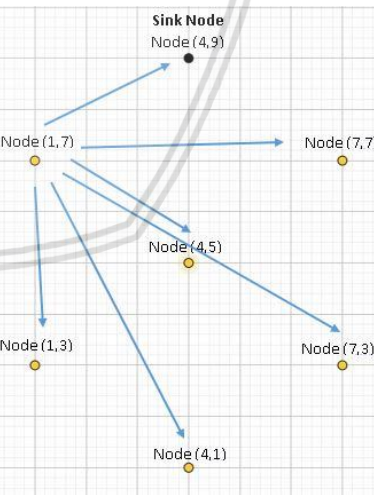
7.	 Proses saat <i>node</i> (7,7) menerima <i>interest</i> dan langsung membalas dengan mengirim data sensor ke <i>sink node</i>.	
8.	 Proses saat <i>sink node</i> menerima data sensor dari <i>node</i> (7,7)	

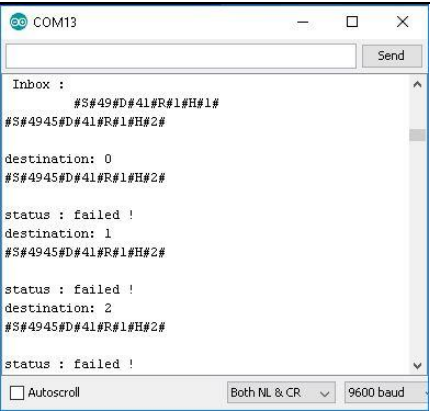
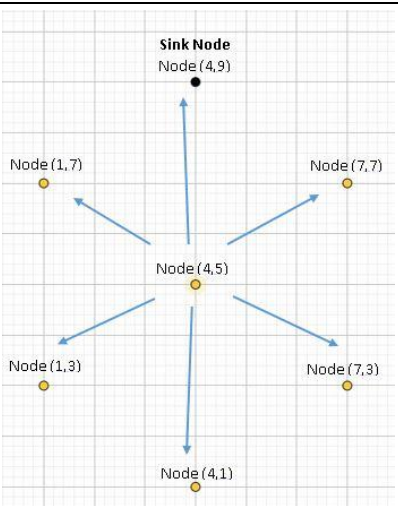
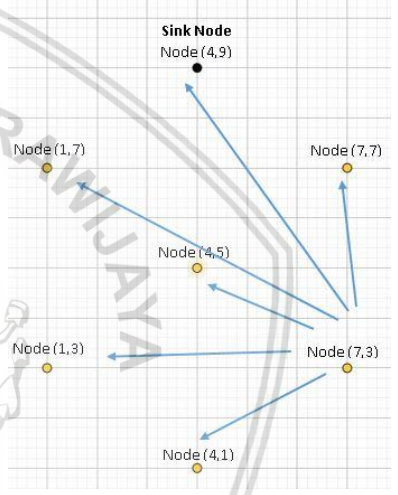
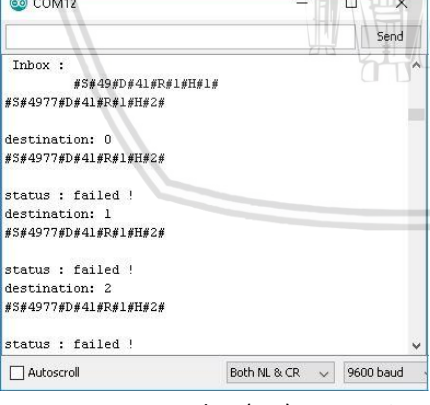
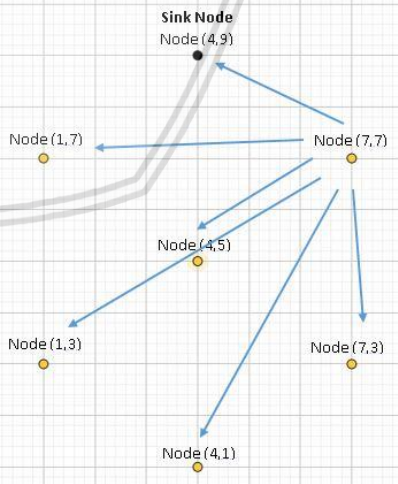
Berdasarkan hasil pengujian pengiriman data sensor kelembaban tanah oleh *node* (7,7) yang dijabarkan pada Tabel 6.15., dapat dianalisis bahwa *sink node* dapat mengirimkan *interest* ke *node* tetangganya. Kemudian *node* tetangga yang lain yang berhasil menerima *interest* meneruskan pesan dengan melakukan *broadcast* ke *node* tetangga lainnya. Lalu *node* (7,7) yang menerima *interest* dan merupakan tetangga *sink node* langsung mengirimkan balasan yang berisi data sensor kelembaban tanah kepada *sink node*. *Sink node* berhasil menerima data sensor.

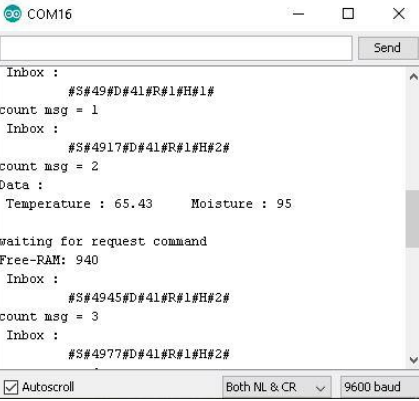
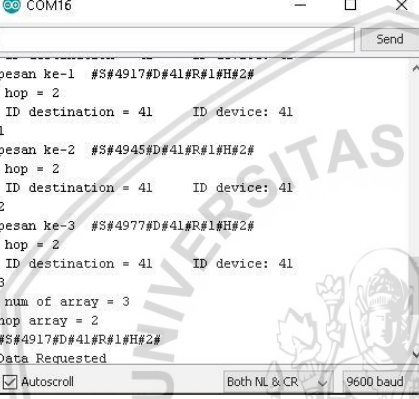
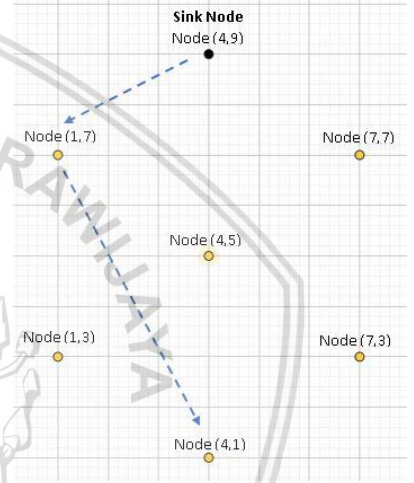
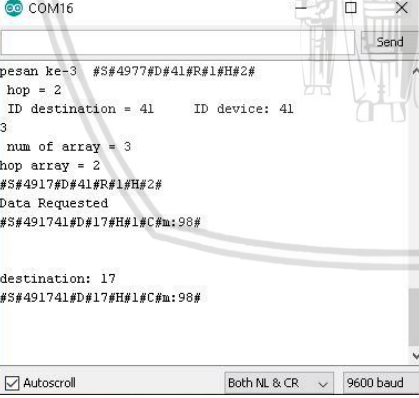
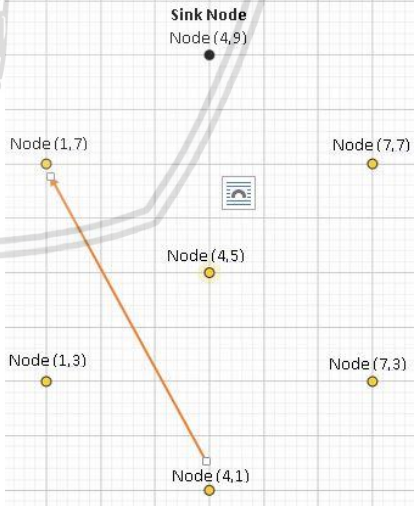
6. Pengiriman Data Sensor Kelembaban Tanah Oleh *Node* (4,1)

Tabel 6.16. Pengiriman Data Sensor Kelembaban Tanah Oleh *Node* (4,1)

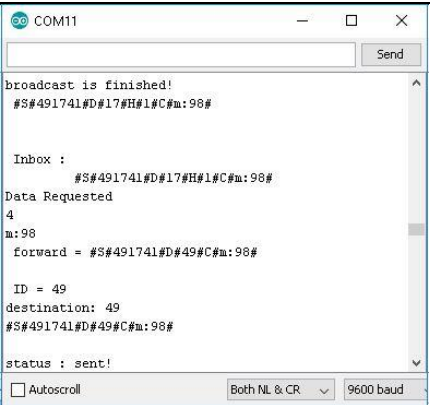
Proses	Hasil	Skema Proses
--------	-------	--------------

1.	 Proses saat <i>sink node</i> broadcast interest ke seluruh <i>node</i> tetangganya.	
2.	 Proses saat <i>node</i> (1,3) menerima <i>interest</i> lalu broadcast ke seluruh <i>node</i> tetangganya.	
3.	 Proses saat <i>node</i> (1,7) menerima <i>interest</i> lalu broadcast ke seluruh <i>node</i> tetangganya.	

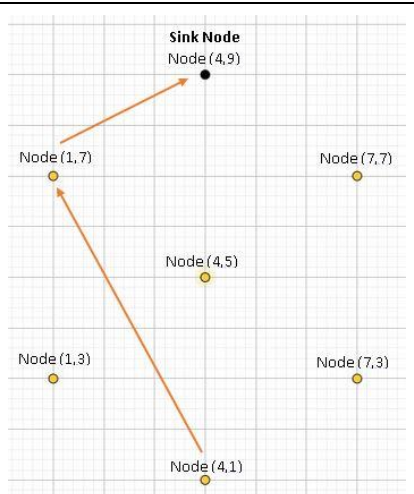
4.	 Proses saat <i>node</i> (4,5) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
5.	 Proses saat <i>node</i> (7,3) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	
6.	 Proses saat <i>node</i> (7,7) menerima <i>interest</i> lalu <i>broadcast</i> ke seluruh <i>node</i> tetangganya.	

7.	 Proses saat <i>node</i> (4,1) menerima <i>interest</i>.	
8.	 Proses saat <i>node</i> (4,1) melakukan sorting <i>interest</i> dengan jumlah hop terkecil.	
9.	 Proses saat <i>node</i> (4,1) mengirim data sensor ke <i>node</i> (1,7) karena <i>node</i> tersebut merupakan <i>node</i> pengirim <i>interest</i> sebelumnya.	

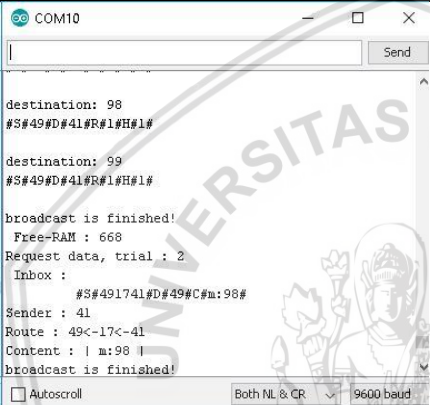
10.



Proses saat *node* (1,7) menerima data sensor dan meneruskannya kepada *sink node*.



11.



Proses saat *sink node* menerima data sensor dari *node* (1,7)



Berdasarkan hasil pengujian pengiriman data sensor kelembaban tanah oleh *node* (4,1) yang dijabarkan pada Tabel 6.16., dapat dianalisis bahwa *sink node* dapat mengirimkan *interest* ke *node* tetangganya. Kemudian *node* tetangga yang lain yang berhasil menerima *interest* meneruskan pesan dengan melakukan *broadcast* ke *node* tetangga lainnya. Lalu *node* (4,1) yang menerima *interest* dan bukan merupakan tetangga *sink node* melakukan sorting *interest* untuk memilih jalur pengiriman data sensor kelembaban tanah melalui hop terkecil. *Node* (4,1) berhasil mengirim data sensor kelembaban tanah ke pengirim *interest* sebelumnya yaitu *node* (1,7). Saat *node* (1,7) berhasil menerima data sensor kelembaban tanah, *node* (1,7) berhasil meneruskan data sensor ke *sink node* sebagai pengirim *interest* pertama kali.

Analisi hasil dari seluruh pengujian pengiriman data sensor kelembaban tanah maka dapat disusun tabel sebagai berikut.

Tabel 6.17. Hasil Pengiriman Dan Penerimaan Interest Kelembaban Tanah

Node Pengirim Interest	Node Penerima Interest	Status Penerimaan Interest
(4,9)	(1,5)	Interest Diterima

	(1,7)	Interest Diterima
	(4,1)	Interest Diterima
	(4,5)	Interest Diterima
	(7,3)	Interest Diterima
	(7,7)	Interest Diterima

Tabel 6.18. Hasil Pengiriman Dan Penerimaan Data Sensor Kelembaban Tanah

Node Pengirim Data Sensor	Status Pengiriman Data Sensor	Status Penerimaan Data Sensor Oleh Sink node
(1,5)	Dikirim	Data Sensor Diterima
(1,7)	Dikirim	Data Sensor Diterima
(4,1)	Dikirim	Data Sensor Diterima
(4,5)	Dikirim	Data Sensor Diterima
(7,3)	Dikirim	Data Sensor Diterima
(7,7)	Dikirim	Data Sensor Diterima

Berdasarkan data yang diperoleh dari Tabel 6.17. dan Tabel 6.18 setiap *node* dapat merima *interest* dari *sink node* dapat membalas *interest* dan membalas *interest* dengan mengirimkan data sensor kelembaban tanah kepada *sink node*.

BAB 7 PENUTUP

Pada bab ini akan dipaparkan kesimpulan dan saran dari penelitian yang telah dilakukan. Pengambilan kesimpulan didasarkan sebagai jawaban atas rumusan masalah yang telah dirumuskan.

7.1 Kesimpulan

Berdasarkan hasil dari penelitian dari implementasi protokol *routing directed diffusion* pada *wireless sensor network* menggunakan media komunikasi RF sehingga dapat diambil kesimpulan sebagai berikut.

1. Sistem dirancang dengan melakukan perancangan perangkat keras, perancangan perangkat lunak dan perancangan topologi. Pada perancangan perangkat keras dilakukan dengan merancang skema node dimana arduino nano dan modul NRF24L01 dari terhubung. Pada perancangan perangkat lunak dilakukan dengan merancang format perintah dan format data. Kemudian membuat diagram alir *sink node*, diagram alir *node* sensor tetangga *sink node* dan diagram alir *node* sensor yang bukan tetangga *sink node*. Lalu dilakukan perancangan topologi dimana setiap node diberikan ID berdasarkan titik koordinat.
2. Sistem diimplementasikan berdasarkan hasil perancangan perangkat keras, perangkat lunak dan perancangan topologi. Pada implementasi perangkat keras, *node* yang dibangun berjumlah 7 *node* diantaranya 1 buah *sink node*, 5 buah *node* sensor tetangga *sink node* dan 1 buah *node* sensor yang bukan tetangga *sink node*. Seluruh *node* terhubung ke satu sumber daya melalui *USB Hub*. Pada implementasi perangkat lunak, kode program yang dibuat melalui arduino IDE memanfaatkan *library* RF24 yaitu *sink_node.ino* untuk kode program *sink node*, *direct_node.ino* untuk kode program *node* sensor tetangga *sink node* dan *source_node.ino* untuk kode program *node* sensor yang bukan tetangga *sink node*. Setiap *node* diberikan ID berdasarkan perancangan topologi melalui kode program masing-masing *node*.
3. Pada analisis hasil pengujian statu *node* pada serial monitor, setiap *node* dapat menampilkan status *node* yang berisi ID *node* dan tipe *node*. Pada analisis hasil pengujian sensor *node* menampilkan data sensor, setiap *node* sensor dapat melakukan pemrosesan data sensor dan ditampilkan pada serial monitor. Data sensor yang ditampilkan ialah data sensor suhu dan kelembaban tanah. Kemudian pada hasil analisis pengujian fungsional dimana sistem diuji dengan 2 skenario yaitu pengiriman data sensor suhu kemudian pengiriman data sensor kelembaban tanah. Berdasarkan hasil pengujian pengiriman data sensor suhu, *sink node* berhasil mengirim *interest* ke seluruh *node* tetangganya dan seluruh *node* tetangga berhasil menerima *interest*. Kemudian setiap *node* sensor yang menjadi tujuan permintaan data, berhasil mengirim data sensor hingga diterima oleh *sink*

node. Berdasarkan hasil pengujian pengiriman data sensor kelembaban tanah, *sink node* berhasil mengirim *interest* ke seluruh *node* tetangganya dan seluruh *node* tetangga berhasil menerima *interest*. Kemudian setiap *node* sensor yang menjadi tujuan permintaan data, berhasil mengirim data sensor hingga diterima oleh *sink node*.

7.2 Saran

Berdasarkan hasil dari penelitian implementasi protokol *routing directed diffusion* pada *wireless sensor network* menggunakan media komunikasi RF sehingga dapat memberikan saran sebagai berikut.

1. Dalam penelitian ini, *node* sensor menggunakan data sensor yang berasal dari perhitungan program secara acak dan tidak menggunakan modul sensor sebenarnya. Sehingga diharapkan untuk penelitian selanjutnya setiap *node* sensor dapat memiliki modul sensor.
2. Pada penelitian ini juga, *sink node* sebagai *node* yang meminta data hanya berjumlah satu. Sehingga diharapkan pada penelitian selanjutnya agar setiap *node* sensor yang melakukan permintaan data sensor.
3. Sensor *node* yang mengirimkan data sensor, hanya dapat mengirimkan satu jenis data sensor. Sehingga diharapkan untuk dilakukan pengembangan agar setiap *node* sensor dapat mengirimkan lebih dari satu jenis data sensor.

DAFTAR PUSTAKA

- Akkaya, K., Younis, M., 2005. *A Survey On Routing Protocol For Wireless Sensor Networks*. Elsevier Ad Hoc Network Journal, 3, 325-249.
- Arduino., "Product Arduino". 2017. [online] Tersedia di : <http://www.arduino.cc/en/Main/Product>.
- Dargie Waltenegus., Poellabauer C., 2010. *Fundamentals of Wireless sensor networks : Theory and Practice*. USA : WILEY.
- Devika R., 2013. *Survey on Routing Protocol in Wireless sensor network*. International Journal of Engineering and Technology (IJET).
- Hac Anna., 2003. *Wireless sensor network Designs*. John Wiley & Sons Ltd.
- Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J., & Silva, F. 2003. Directed Diffusion For Wireless Sensor Networking. *Journal IEEE/ACM Transactions on Networking (TON)*, 11, 2-16.
- Jain, V., & Khan, N., A. 2014. Simulation Analysis of Directed diffusion and SPIN Routing Protocol in Wireless sensor network. *Conference on IT in Business, Industry and Government (CSIBIG)*, 1, 1-6.
- Jamil, U. 2018. Analisis Kinerja Pengiriman Data Modul Transceiver NRF24L01, Xbee dan Wifi ESP8266 Pada Wireless sensor network. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 2, 1510-1517.
- Nordic Semiconductor., "Single Chip 2.4 GHz Transceiver NRF24L01", 2017. [online] Tersedia di : <http://www.nordicsemi/eng/nRF24L01>.
- Perillo, M. A., Heinzelman, W. B., 2005. *Wireless sensor network Protocol*. Department of Electrical and Computer Engineering, University of Rochester.
- Ridho, S., & Yasri, I. 2016. Analisa Model Directed diffusion (DD) Berbasis Cluster Pasa Wireless sensor network (WSN). *Jurnal Online Mahasiswa Fakultas Teknik : Universitas Riau*, 3, 1-14.
- Robomania., "Arduino Nano v3", 2017. [online] Tersedia di : <https://www.robomania.ro/arduino-nano-v3>.
- Samaras S., 2016. *On Direct Diffusion Routing for Wireless Sensor Networks*. IEEEExplore.
- Sharma, G., Bala, S., & Verma, A., K. 2009. Comparison of Flooding and Directed Diffusion for Wireless Sensor Network. *Annual IEEE India Conference*, 1, 1-4.
- Sohraby, K., Minoli, D., & Znati, T., F. 2007. *Wireless sensor networks: technology, protocols, and applications*. John Wiley & Sons. New York.