

**IMPLEMENTASI METODE REFERENCE BROADCAST TIME
SYNCHRONIZATION DAN TIME DIVISION MULTIPLE
ACCESS PADA WIRELESS SENSOR NETWORK**

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Aulia Arif Darmawan

NIM: 125150301111018



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

PENGESAHAN

IMPLEMENTASI METODE REFERENCE BROADCAST TIME SYNCHRONIZATION DAN
TIME DIVISION MULTIPLE ACCESS PADA WIRELESS SENSOR NETWORK

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Aulia Arif Darmawan

NIM: 125150301111018

Skripsi ini telah diuji dan dinyatakan lulus pada
7 April 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Sabriansyah Rizqika Akbar, S.T, M.Eng.

NIP: 19820809 201212 1 004

Mochammad Hannats Hanafi I., S.ST, M.T.

NIK: 201405 881229 1 001

Mengetahui

Ketua Program Studi Teknik Informatika

Issa Arwani, S.Kom., M.Sc.

NIP: 19830922 201212 1 003

PERNYATAAN ORISINALITAS

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

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 7 April 2016

Aulia Arif Darmawan

NIM: 125150301111018



KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas limpahan rahmat dan petunjuknya hingga penulis dapat menyelesaikan skripsi yang berjudul "*Implementasi Metode Reference Broadcast Time Synchronization dan Time Division Multiple Access pada Wireless Sensor Network*".

Atas bantuan moral dan materiil yang diberikan pada penyusunan skripsi ini, maka penulis mengucapkan banyak terima kasih kepada:

1. Bapak Sabriansyah Rizqika Akbar, S.T, M.Eng dan Bapak Mochammad Hannats Hanafi I., S.ST, M.T selaku dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini dengan baik.
2. Bapak Drs. Marji, M.T. selaku ketua Program Studi Informatika Program Teknologi Informasi dan Ilmu Komputer.
3. Ayahanda dan Ibunda dan seluruh keluarga besar atas segala nasehat, kasih sayang, perhatian dan kesabarannya memberikan semangat kepada penulis, serta senantiasa tiada hentinya memberikan doa dan semangat demi terselesaikannya skripsi ini.
4. Seluruh civitas akademika Informatika Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
5. Rekan-rekan Sistem Komputer yang selalu memberikan motivasi dan bantuan dalam pengerjaan skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 7 April 2016

Penulis

arifauliad7997@gmail.com

ABSTRAK

Berbagai macam teknologi telah di kembangkan untuk membantu manusia dalam memperoleh informasi. Salah satunya adalah *Wireless Sensor Network* (WSN). WSN merupakan jaringan nirkabel yang terdiri dari beberapa node yang terhubung. Node pada WSN merupakan perangkat yang berukuran kecil memiliki sumber daya berupa baterai dan memiliki kemampuan untuk memantau, mendeteksi, memberi informasi serta berkomunikasi antar node lain. Teknologi ini banyak diterapkan pada bidang yang membutuhkan pengawasan dan pengumpulan data. Namun masalah utama dalam teknologi ini adalah pengumpulan data. Pada WSN setiap node berkomunikasi menggunakan frekuensi yang setara/sama dan memiliki waktu lokal yang berbeda, hal ini memungkinkan adanya node yang mengirimkan data pada waktu yang bersamaan. Akibatnya adalah adanya tabrakan data yang membuat data tidak dapat diterima oleh node lain.

Cara untuk mengatasi hal ini yaitu dengan menerapkan metode *Time Division Multiple Access* (TDMA). Metode TDMA mampu memberikan jadwal pengiriman data pada tiap-tiap node. Sehingga nantinya tidak ada node yang mengirimkan data pada waktu yang bersamaan. Namun untuk dapat menerapkan metode tersebut tiap node harus memiliki waktu yang setara/sama. Salah satu metode yang mampu memberikan penyetaraan waktu adalah metode *Reference Broadcast Synchronization* (RBS). Metode ini mensinkronisasikan beberapa node yang memiliki waktu lokal yang berbeda (*receiver*) dengan bantuan node yang memberikan tanda sinkronisasi (*beacon*).

Hasilnya, setiap node dalam WSN mampu mengirimkan data sesuai dengan metode TDMA yang telah diimplementasikan. Selain itu sinkronisasi waktu yang dilakukan dengan menggunakan metode RBS memberikan waktu yang setara dengan ketelitian hingga satuan *microsecond*. Hal ini tentunya menjadikan node dalam WSN mampu memberikan informasi yang akurat dengan jaminan tidak adanya data yang *error* karena tabrakan data.

Kata kunci: WSN, tabrakan data, penjadwalan, TDMA, sinkronisasi waktu, RBS, *error*.

ABSTRACT

A many variety of technologies have been developed to help people to obtain information. One of them is Wireless Sensor Network (WSN). WSN is a wireless network that consist by many connected nodes. WSN nodes is a device that have small size, power resources like batteries and have the ability to monitor, detect, provide information and communicate with other nodes. This technology is widely applied in the fields that require surveillance and data collection. But the main problem in this technology is data collection. In WSN all nodes communicate using the same frequency and have different local time, it allows the nodes that transmit data at the same time. The result is a collision of data that makes the data can not be received by the other nodes.

To handle this problem is by applying the Time Division Multiple Access (TDMA). TDMA is method that able to make schedule transmitting data of each node. So there are no nodes that transmit data at the same time. But to implementing to the system each node should have equal time / same. One method that is able to provide the time synchronization is Reference Broadcast Synchronization (RBS). This method synchronizes multiple nodes that have different local time (receiver) with the assist of node that provides synchronization alert (beacons).

As a result, each node in a WSN is able to transmit data accordance TDMA method that had been implemented. Besides time synchronization performed using RBS give synchronization time with microsecond accuracy. It certainly makes the WSN node capable of provide accurate information to guarantee the absence of errors due to data collisions.

Keywords: WSN, data collisions, schedule, TDMA, time synchronization, RBS, error.

DAFTAR ISI

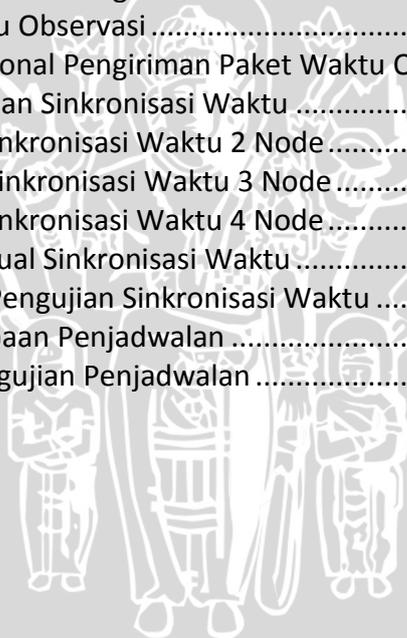
| | |
|---|-----------|
| PENGESAHAN | ii |
| PERNYATAAN ORISINALITAS | iii |
| KATA PENGANTAR..... | iv |
| ABSTRAK..... | v |
| ABSTRACT | vi |
| DAFTAR ISI | vii |
| DAFTAR TABEL..... | ix |
| DAFTAR GAMBAR..... | x |
| DAFTAR LAMPIRAN | xi |
| BAB 1 PENDAHULUAN..... | 1 |
| 1.1 Latar belakang..... | 1 |
| 1.2 Rumusan Masalah..... | 2 |
| 1.3 Tujuan | 3 |
| 1.4 Manfaat..... | 3 |
| 1.5 Batasan masalah | 3 |
| 1.6 Sistematika pembahasan | 4 |
| BAB 2 LANDASAN KEPUSTAKAAN | 5 |
| 2.1 Tinjauan Pustaka | 5 |
| 2.2 Dasar Teori | 7 |
| 2.2.1 <i>Wireless Sensor Network</i> | 7 |
| 2.2.2 Sinkronisasi Waktu <i>Reference Broadcast Synchronization</i> | 8 |
| 2.2.3 Penjadwalan <i>Time Division Multiple Access</i> | 9 |
| 2.2.4 Mikrokontroler Arduino Pro Mini | 10 |
| 2.2.5 Arduino IDE | 10 |
| 2.2.6 Modul nRf24L01 | 11 |
| BAB 3 METODE PENELITIAN | 12 |
| 3.1 Studi Literatur | 13 |
| 3.2 Analisis Kebutuhan..... | 13 |
| 3.2.1 Kebutuhan Sistem | 13 |
| 3.2.2 Kebutuhan Perangkat Keras..... | 14 |
| 3.2.3 Kebutuhan Perangkat Lunak | 14 |
| 3.3 Perancangan Sistem..... | 14 |
| 3.4 Implementasi Sistem | 16 |
| 3.5 Pengujian dan Analisis | 16 |
| 3.6 Kesimpulan..... | 17 |
| BAB 4 Rekayasa kebutuhan..... | 18 |
| 4.1 Gambaran Umum Sistem..... | 18 |
| 4.1.1 Tujuan..... | 18 |
| 4.1.2 Kegunaan..... | 18 |
| 4.1.3 Karakteristik Pengguna | 19 |
| 4.1.4 Lingkungan Operasi..... | 19 |
| 4.2 Kebutuhan Sistem | 19 |

| | | |
|--------------|---|-----------|
| 4.2.1 | Kebutuhan Fungsional..... | 19 |
| 4.2.2 | Kebutuhan Non-Fungsional | 20 |
| 4.2.3 | Kebutuhan Perangkat Keras..... | 20 |
| 4.2.4 | Kebutuhan Perangkat Lunak | 21 |
| 4.2.5 | Kebutuhan Komunikasi | 21 |
| BAB 5 | Perancangan dan implementasi | 22 |
| 5.1 | Perancangan Sistem..... | 22 |
| 5.1.1 | Perancangan Alur Kerja Sistem | 22 |
| 5.1.2 | Perancangan Perangkat Keras | 26 |
| 5.1.3 | Perancangan Perangkat Lunak..... | 27 |
| 5.1.4 | Perancangan Paket Data | 27 |
| 5.1.5 | Perancangan Algoritma..... | 29 |
| 5.1.6 | Perancangan Konfigurasi Frekuensi dan Channel..... | 33 |
| 5.2 | Implementasi Sistem | 33 |
| 5.2.1 | Implementasi Secara Keseluruhan..... | 33 |
| 5.2.2 | Spesifikasi Perangkat Keras..... | 34 |
| 5.2.3 | Spesifikasi Perangkat Lunak | 35 |
| 5.2.4 | Implementasi Paket Data..... | 36 |
| 5.2.5 | Implementasi Algoritma..... | 37 |
| 5.2.6 | Implementasi Konfigurasi Frekuensi dan Channel..... | 39 |
| BAB 6 | PENGUJIAN | 40 |
| 6.1 | Pengujian Fungsional | 40 |
| 6.1.1 | Pengiriman Paket Referensi..... | 40 |
| 6.1.2 | Pengiriman Waktu Observasi..... | 42 |
| 6.2 | Pengujian Sinkronisasi Waktu..... | 44 |
| 6.2.1 | Hasil Pengujian | 46 |
| 6.2.2 | Pengujian Matematis | 48 |
| 6.2.3 | Analisa Pengujian | 50 |
| 6.3 | Pengujian Penjadwalan..... | 51 |
| 6.3.1 | Hasil Pengujian..... | 52 |
| 6.3.2 | Analisa Pengujian | 53 |
| BAB 7 | PENUTUP | 54 |
| 7.1 | Kesimpulan..... | 54 |
| 7.2 | Saran | 54 |
| | DAFTAR PUSTAKA..... | 56 |
| | LAMPIRAN A PENGUJIAN PENJADWALAN | 58 |



DAFTAR TABEL

| | |
|---|----|
| Tabel 2.1 Perbandingan Sinkronisasi RBS dengan NTP | 6 |
| Tabel 2.2 Perbandingan Keunggulan Dan Kekurangan Protokol | 6 |
| Tabel 5.1 Contoh Pertukaran Waktu Observasi | 24 |
| Tabel 5.2 Contoh Perhitungan Phase Offset | 24 |
| Tabel 5.3 Contoh Perhitungan Waktu Setara | 25 |
| Tabel 5.4 Konfigurasi Pin Arduino Pro Mini dan nRF24L01 | 26 |
| Tabel 5.5 Paket Data Boadcast Node <i>Beacon</i> | 27 |
| Tabel 5.6 Paket Data Waktu Observasi Node <i>Receiver</i> | 28 |
| Tabel 5.7 Paket Data Informasi Node <i>Receiver</i> | 28 |
| Tabel 5.8 Spesifikasi Mikrokontroller Arduino Pro Mini | 34 |
| Tabel 5.9 Spesifikasi Modul nRF24L01 | 34 |
| Tabel 5.10 Spesifikasi Personal Computer | 35 |
| Tabel 5.11 Spesifikasi Arduino IDE | 35 |
| Tabel 6.1 Prosedur pengujian Pengiriman Paket referensi | 41 |
| Tabel 6.2 Percobaan Fungsional Pengiriman Paket referensi | 42 |
| Tabel 6.3 Pengiriman Waktu Observasi | 43 |
| Tabel 6.4 Percobaan Fungsional Pengiriman Paket Waktu Observasi | 44 |
| Tabel 6.5 Prosedur Percobaan Sinkronisasi Waktu | 45 |
| Tabel 6.6 Hasil Pengujian Sinkronisasi Waktu 2 Node | 46 |
| Tabel 6.7 Hasil Pengujian Sinkronisasi Waktu 3 Node | 47 |
| Tabel 6.8 Hasil Pengujian Sinkronisasi Waktu 4 Node | 47 |
| Tabel 6.9 Perhitungan Manual Sinkronisasi Waktu | 49 |
| Tabel 6.10 Nilai Rata-Rata Pengujian Sinkronisasi Waktu | 50 |
| Tabel 6.11 Prosedur Percobaan Penjadwalan | 52 |
| Tabel 6.12 Rangkuman Pengujian Penjadwalan | 52 |



DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2.1 Perbedaan Sinkronisasi Metode Tradisional dan RBS | 5 |
| Gambar 2.2 Ilustrasi WSN Sumber : Yinbiao, et al. (2014) | 8 |
| Gambar 2.3 Komponen Utama Sensor Node Sumber : Yinbiao, et al. (2014)..... | 8 |
| Gambar 2.4 Ilustrasi Slot Waktu TDMA | 10 |
| Gambar 2.5 Arduino Pro mini Sumber : Arduino (2016) | 10 |
| Gambar 2.6 Tampilan Arduino IDE..... | 10 |
| Gambar 2.7 Modul <i>Transceiver</i> nRF24L01 Sumber : Chantrell (2013) | 11 |
| Gambar 3.1 Diagram Alir Penelitian..... | 12 |
| Gambar 3.2 Diagram blok perangkat sensor node | 14 |
| Gambar 3.3 Topologi Jaringan | 15 |
| Gambar 3.4 Diagram Alur Kerja Sistem | 16 |
| Gambar 5.1 Susunan Node <i>Beacon</i> dan <i>Receiver</i> | 22 |
| Gambar 5.2 Ilustrasi Pengiriman Paket Referensi dari Node <i>Beacon</i> | 23 |
| Gambar 5.3 Ilustrasi Pengiriman Paket Waktu Observasi | 23 |
| Gambar 5.4 Pengiriman Paket Dari <i>Receiver</i> ke <i>Beacon</i> | 25 |
| Gambar 5.5 Diagram Blok Komponen Node..... | 26 |
| Gambar 5.6 Skematik Konfigurasi Pin | 27 |
| Gambar 5.7 Ilustrasi paket data..... | 27 |
| Gambar 5.8 Diagram Alir Node <i>Beacon</i> | 29 |
| Gambar 5.9 Diagram Alir Node <i>Receiver</i> | 30 |
| Gambar 5.10 Diagram Alir Sinkronisasi RBS | 31 |
| Gambar 5.11 Diagram Alir Penjadwalan TDMA..... | 32 |
| Gambar 5.12 Komponen Utama Node | 33 |
| Gambar 5.13 Implementasi Paket Data Broadcast Node <i>Beacon</i> | 36 |
| Gambar 5.14 Implementasi Paket Data Waktu Observasi Node <i>Receiver</i> | 36 |
| Gambar 5.15 Implementasi Paket Data Informasi Node <i>Receiver</i> | 37 |
| Gambar 5.16 Implementasi Algoritma pada Node <i>Beacon</i> | 37 |
| Gambar 5.17 Implementasi Penerimaan Paket referensi pada Node <i>Receiver</i> ... | 38 |
| Gambar 5.18 Implementasi Metode RBS pada Node <i>Receiver</i> | 38 |
| Gambar 5.19 Implementasi TDMA pada Node <i>Receiver</i> | 39 |
| Gambar 5.20 Implementasi Konfigurasi Channel | 39 |
| Gambar 6.1 Skenario Pengujian Pengiriman Paket Referensi | 40 |
| Gambar 6.2 Hasil Pengujian Pengiriman Paket referensi | 41 |
| Gambar 6.3 Skenario Pengujian Waktu Observasi | 43 |
| Gambar 6.4 Pengiriman Waktu Observasi | 43 |
| Gambar 6.5 Skenario Tahap Satu Pengujian Sinkronisasi Waktu | 44 |
| Gambar 6.6 Skenario Tahap Dua Pengujian Sinkronisasi Waktu..... | 45 |
| Gambar 6.7 Hasil Pengujian Sinkronisasi Waktu | 46 |
| Gambar 6.8 Rincian Pengujian Sinkronisasi | 48 |
| Gambar 6.9 Grafik Nilai Rata-Rata Pengujian Sinkronisasi Waktu | 50 |
| Gambar 6.10 Skenario Pengujian Penjadwlan | 51 |
| Gambar 6.11 Ilustrasi Penjadwalan | 51 |

DAFTAR LAMPIRAN

| | |
|---|----|
| LAMPIRAN A PENGUJIAN PENJADWALAN | 58 |
| A.1 Penjadwalan Pengiriman 2 Node dengan Time Slot 2 detik | 58 |
| A.2 Penjadwalan Pengiriman 2 Node dengan Time Slot 3 detik | 59 |
| A.3 Penjadwalan Pengiriman 2 Node dengan Time Slot 4 detik | 60 |
| A.4 Penjadwalan Pengiriman 2 Node dengan Time Slot 5 detik | 61 |
| A.5 Penjadwalan Pengiriman 3 Node dengan Time Slot 2 detik | 62 |
| A.6 Penjadwalan Pengiriman 3 Node dengan Time Slot 3 detik | 63 |
| A.7 Penjadwalan Pengiriman 3 Node dengan Time Slot 4 detik | 64 |
| A.8 Penjadwalan Pengiriman 3 Node dengan Time Slot 5 detik | 65 |
| A.9 Penjadwalan Pengiriman 4 Node dengan Time Slot 2 detik | 66 |
| A.10 Penjadwalan Pengiriman 4 Node dengan Time Slot 3 detik | 67 |
| A.11 Penjadwalan Pengiriman 4 Node dengan Time Slot 4 detik | 68 |
| A.12 Penjadwalan Pengiriman 4 Node dengan Time Slot 5 detik | 69 |



BAB 1 PENDAHULUAN

1.1 Latar belakang

Banyak teknologi dikembangkan untuk membantu manusia dalam memperoleh informasi. Salah satu teknologi yang sekarang berkembang adalah *Wireless Sensor Network* (WSN). WSN adalah jaringan nirkabel yang terdiri dari node yang membentuk suatu jaringan. Node pada jaringan ini merupakan perangkat yang berukuran kecil, murah, bertenaga baterai sehingga memiliki sumber daya yang terbatas dan dilengkapi dengan sensor dan modul komunikasi. Node ditempatkan secara strategis di wilayah yang diinginkan dan bertugas untuk melakukan pemantauan, pendeteksian, memberikan informasi dan berkomunikasi. Setiap node memiliki tujuan pengumpulan data yang disebut sink (node pusat yang menampung data sensor dari node lain) (AlSkaif, et al., 2015).

Sejak pertama kali teknologi ini muncul, WSN sudah banyak diterapkan. Salah satu fitur paling penting adalah pengumpulan data, dimana data sensor dikumpulkan dari beberapa node dikirimkan melalui komunikasi nirkabel ke node sink. Namun masalahnya adalah bagaimana data ini dapat dikirimkan ke *node sink* tanpa kehilangan informasi (Wang & Liu, 2011). WSN biasanya bertukar data pada frekuensi yang sama. Dapat dibayangkan apabila pada suatu keadaan node mengirim data secara bersamaan, tentunya hal ini membuat terjadinya tabrakan data yang berakibat data tidak dapat diterima dan informasi tidak dapat disampaikan. Karena itu, pada WSN biasanya memiliki metode dan aturan tersendiri untuk menangani masalah ini. Salah satu metode adalah dengan memberikan jadwal pada setiap node untuk mengirim dan menerima data. Sehingga nantinya tidak ada node yang mengirimkan data bersamaan yang berakibat adanya tabrakan data.

Untuk menunjang pemberian jadwal pada tiap node dalam suatu jaringan, maka setiap node harus memiliki waktu yang setara. Yaitu waktu lokal antara tiap node tidak memiliki perbedaan atau selisih waktu yang besar. Sedangkan dalam WSN, node sering ditemukan tidak memiliki waktu yang setara dikarenakan tidak memungkinkan dapat mengaktifkan setiap sensor node dalam waktu yang bersamaan. Untuk menangani hal ini adalah dengan memberikan suatu metode yang dapat menyetarakan waktu pada setiap node atau dapat disebut *time synchronization* (sinkronisasi waktu). Dengan adanya waktu yang setara dan jadwal pengiriman-penerimaan data pada setiap node, menjadikan setiap node pada WSN memiliki alur pengiriman data yang baik. Sehingga informasi dapat di sebar dan diterima dengan baik juga.

Salah satu metode untuk penyetaraan waktu adalah metode *Reference Broadcast Synchronization* (RBS). Seperti yang ditulis pada penelitian berjudul "*Fine-Grained Network Time Synchronization using Reference Broadcasts*". Penulis mengaktakan bahwa RBS dapat lebih presisi dalam sinkronisasi, fleksibel, dan efisien sumber daya dibandingkan dengan sinkronisasi metode tradisional. Metode tradisional mensinkronisasikan antara *sender to receiver*. Dimana

receiver harus menerima paket waktu yang dikirimkan sender dan menjadikannya waktu lokalnya. Sehingga saat ada delay pengiriman yang terlalu lama menjadikan selisih waktu antara sender dan receiver semakin besar. Berbeda dengan metode tradisional, RBS mensinkronisasikan satu *receiver* dengan *receiver* lainnya. Metode RBS bekerja dengan adanya node yang memberikan tanda sinkronisasi (*beacon*), node ini secara berkala mengirim paket referensi untuk node *receiver* secara broadcast. *Receiver* menggunakan waktu kedatangan paket sebagai titik acuan untuk membandingkan waktu lokal mereka dengan waktu lokal node *receiver* lain. Pesan yang dikirimkan tidak berupa *timestamp* dari *beacon*. Sehingga tidak menjadi masalah apabila terjadi *delay* pengiriman paket (Elson, et al., 2002).

Penelitian lain berjudul "*Time Synchronization in Wireless Sensor Networks: A Survey*" membandingkan beberapa metode sinkronisasi dan mengatakan bahwa RBS memiliki akurasi yang sangat baik. Bahkan akurasi yang dapat dijangkau bisa sampai mikro detik. Berbeda dengan sinkronisasi lainnya, RBS memiliki kemampuan untuk mensinkronisasikan lebih dari satu node hanya dengan satu paket referensi. Dilanjutkan dengan sinkronisasi antar node *receiver* (Lasassmeh & Conrad, 2010).

Pada penelitian yang lain dengan judul "*A Survey on Scheduling Algorithms for Wireless Sensor Networks*", menjelaskan bahwa pada WSN terdapat suatu permasalahan yang menantang seperti daya dan bandwidth yang terbatas. Mereka mengatakan *Time Division Multiple Access* (TDMA) dapat memecahkan masalah tersebut. Mereka mengusulkan berbagai metode penjadwalan TDMA dengan tujuan yang berbeda. TDMA sendiri merupakan metode yang digunakan untuk memberikan penjadwalan pada slot waktu. Sehingga pada satu slot waktu hanya terjadi satu kegiatan. Hal ini membuat adanya jaminan dalam tidak adanya tabrakan data pada WSN (Kumar & Chauhan, 2011).

Dari masalah yang ada dan fakta dari penelitian sebelumnya penulis ingin mengusulkan sistem yang dapat mengatasi permasalahan pengiriman data pada WSN. Sistem ini menggabungkan dua metode yaitu sinkronisasi waktu dengan RBS dan metode penjadwalan TDMA. Diharapkan dengan penelitian ini menghasilkan suatu sistem yang dapat memberikan jaminan dalam hal pengiriman-penerimaan informasi pada WSN.

1.2 Rumusan Masalah

Adapun rumusan masalah yang didapatkan dari latar belakang sebagai berikut:

1. Bagaimana implementasi sensor node menggunakan mikrokontroler Arduino Pro mini dan modul transceiver nRF24L01?
2. Bagaimana implementasi metode RBS untuk sinkronisasi waktu pada wireless sensor node?

3. Bagaimana implementasikan metode TDMA menggunakan penggunaan slot waktu yang disediakan?
4. Bagaimana tingkat keberhasilan metode pengiriman data menggunakan sinkronisasi waktu RBS dan TDMA?

1.3 Tujuan

Beberapa tujuan dari penelitian ini antara lain:

1. Mendapatkan sensor node yang dapat bekerja dengan baik menggunakan mikrokontroler Arduino Pro Mini dan modul tranceiver nRF24L01.
2. Dapat mensinkronisasikan waktu pada sensor node dengan menggunakan metode RBS.
3. Dapat membagi slot waktu yang telah disediakan dengan menggunakan metode TDMA.
4. Mendapatkan sistem yang memiliki tingkat keberhasilan pengiriman data yang tinggi dengan metode pengiriman data menggunakan sinkronisasi waktu RBS dan TDMA.

1.4 Manfaat

Dengan penelitian ini, penulis berharap dapat memberikan inovasi dalam mengatasi masalah pengiriman data pada WSN. Hasil penelitian ini juga dapat menjadi bahan analisis kualitas dari penerapan metode RBS dan TDMA pada WSN. Penulis juga berharap dengan penelitian ini dapat digunakan sebagai acuan ilmiah dalam pengembangan bidang WSN. Sehingga dapat mengembangkan studi dalam bidang WSN dan menerapkannya pada lingkungan Universitas Brawijaya khususnya Fakultas Teknologi Informasi dan Ilmu Komputer.

1.5 Batasan masalah

Pada penelitian ini terdapat beberapa batasan masalah, antara lain:

1. Sistem menggunakan node menggunakan mikrokontroler Arduino Pro mini dan modul *transceiver* nRF2401L.
2. Terdapat satu node yang berfungsi sebagai *beacon* sekaligus base station dan empat node berfungsi sebagai *receiver*.
3. Menguji kualitas sinkronisasi waktu dan penjadwalan.
4. Penelitian dan pengujian dilakukan pada ruangan dan tidak ada penghalang diantara node.
5. Jarak dari tiap-tiap node yang berhubungan kurang dari 8 meter.
6. Setiap node dapat saling berkomunikasi tanda adanya perantara (single hop).

1.6 Sistematika pembahasan

Berikut struktur penulisan pada masing-masing bab yang digunakan pada penelitian ini:

BAB I Pendahuluan

Menjelaskan tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan dari “IMPLEMENTASI METODE REFERENCE BROADCAST TIME SYNCHRONIZATION DAN TIME DIVISION MULTIPLE ACCESS PADA WIRELESS SENSOR NETWORK”

BAB II Landasan Kepustakaan

Pada bab ini menjelaskan teori yang terkait dengan penelitian. Selain itu juga dijelaskan tentang penelitian serupa yang telah dilakukan sebelumnya.

BAB III Metode Penelitian

Membahas tentang langkah kerja yang dilakukan dalam penelitian yaitu studi literatur, analisis kebutuhan sistem, implementasi dan pengujian dari sistem.

BAB IV Rekayasa Kebutuhan

Membahas semua yang dibutuhkan oleh sistem. Mulai dari kebutuhan perangkat keras dan lunak hingga kebutuhan fungsional yang dibutuhkan.

BAB V Perancangan dan Implementasi

Menjelaskan tentang bagaimana sistem ini dirancang beserta bagaimana perangkat saling berhubungan. Kemudian implementasi dari perancangan yang telah dibuat.

BAB VI Pengujian

Pengujian dilakukan untuk melihat tingkat keberhasilan sistem. Sebelum dilakukan pengujian, skenario pengujian dibuat terlebih dahulu dan menentukan parameter yang diuji.

BAB VII Penutup

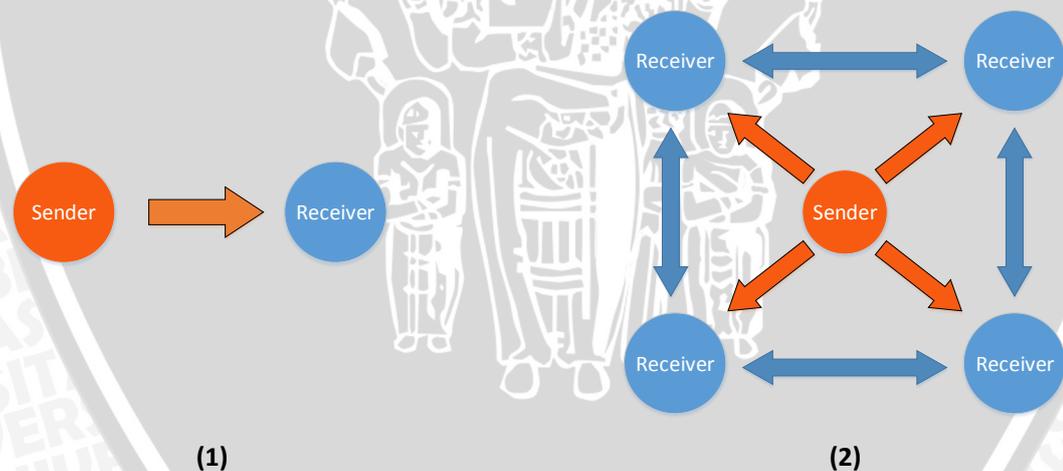
Pada bab penutup membahas kesimpulan akhir dari hasil penelitian yang telah dilakukan. Selain itu juga dituliskan beberapa saran penelitian yang dapat dilakukan agar penelitian ini dapat berlanjut.

BAB 2 LANDASAN KEPUSTAKAAN

Pada penelitian ini pustaka yang dijadikan referensi adalah penelitian berjudul “*Fine-Grained Network Time Synchronization using Reference Broadcasts*” yang ditulis oleh Jeremy Elson, Lewis Girod and Deborah Estrin. Dilanjutkan dengan penelitian “*Time Synchronization in Wireless Sensor Networks: A Survey*” oleh Sami Lasassmeh dan James Conrad. Yang terakhir adalah penelitian berjudul “*A Survey on Scheduling Algorithms for Wireless Sensor Networks*” yang dilakukan Sumit Kumar dan Siddhartha Chauhan. Ditambah dengan asar teori yang diperlukan berdasarkan latar belakang dan rumusan masalah adalah konsep dari *time synchronization*, *channel access method*, pemrograman arduino dan nrf2401l.

2.1 Tinjauan Pustaka

Pada penelitian berjudul “*Fine-Grained Network Time Synchronization using Reference Broadcasts*”. Penulis melihat peran metode RBS dalam sinkronisasi waktu pada WSN. Penelitian menggunakan perangkat *Berkeley motes*. Penulis Menjelaskan bahwa References Broadcast Synchronization (RBS) merupakan suatu metode yang berguna untuk mensinkronisasikan waktu dari node dalam WSN. Model sinkronisasi RBS berbeda dengan metode sinkronisasi waktu tradisional. Dimana sinkronisasi waktu tradisional menggunakan model *sender to receiver* (nomor 1) sedangkan RBS menggunakan model *receiver to receiver* (nomor 2). Ilustrasi terdapat pada Gambar 2.1 (Elson, et al., 2002).



Gambar 2.1 Perbedaan Sinkronisasi Metode Tradisional dan RBS

Pada sinkronisasi RBS memiliki satu node yang digunakan sebagai penanda sinkronisasi yang disebut node *beacon* dan node yang disinkronisasi bernama node *receiver*. Pertama node beacon mengirim paket referensi kepada node *receiver*. Node receiver menggunakan waktu kedatangan paket sebagai penanda dimulainya sinkronisasi. Waktu kedatangan paket itu juga digunakan untuk mencari *offset* relatif (perbedaan waktu) dari tiap node *receiver*. Dikarenakan yang digunakan adalah waktu kedatangan paket, metode ini dapat mengabaikan *error delay* waktu pengiriman. Salah satu kelebihan lain dari

metode ini adalah dapat meningkatkan kepresisian sinkronisasi waktu dengan meningkatkan jumlah paket referensi yang dikirimkan beacon (Lasassmeh & Conrad, 2010).

Terakhir adalah perbandingan kepresisian waktu metode RBS dengan metode tradisional yaitu *Network Time Protocol* (NTP). Dimana dapat dilihat bahwa metode 8 kali lebih baik dalam sinkronisasi dibandingkan dengan NTP. Dapat dilihat pada Tabel 2.1.

Tabel 2.1 Perbandingan Sinkronisasi RBS dengan NTP

| Network Load | Sync | Mean Error | Std Dev | 50% Bound | 95% Bound | 99% Bound |
|--------------|------------|------------|---------|-----------|-----------|-----------|
| Light | RBS | 6.29 | 6.45 | 4.22 | 20.53 | 29.61 |
| Light | NTP | 51.18 | 53.50 | 42.52 | 131.20 | 313.64 |
| Light | NTP-Offset | 204.17 | 599.44 | 48.15 | 827.42 | 4334.50 |
| Heavy | RBS | 8.44 | 9.37 | 5.86 | 28.53 | 48.61 |
| Heavy | NTP | 1542.27 | 1192.53 | 1271.38 | 3888.79 | 5577.82 |
| Heavy | NTP-Offset | 5139.08 | 6994.58 | 3163.11 | 15156.44 | 38897.36 |

Sumber: Elson, et al. (2002)

Menurut peneliti sinkronisasi RBS dapat 8 kali lebih presisi dari sinkronisasi menggunakan NTP pada *light network load*. Untuk *heavy network load*, NTP memiliki *error* yang sangat tinggi sedangkan RBS adalah hampir tidak terpengaruh (Elson, et al., 2002).

Pada penelitian kedua yang berjudul “A Survey on Scheduling Algorithms for *Wireless Sensor Networks*”. Pada penelitian ini, penulis membuat suatu perbandingan dengan menggunakan usulan baru dalam penggunaan TDMA. Penulis memberikan beberapa penggunaan TDMA sesuai kegunaan dalam suatu sistem. Usulan itu antara lain, *Energy-Efficient Wake-Up Scheduling MAC Protocol* (EEWS), *Minimum Delay Scheduling TDMA Protocol* (MDS), *Distributed Aggregation Scheduling* (DAS), *Traffic-Adaptive MAC Protocol* (TRAMA), *DMAC Protocol* Usulan tersebut memiliki keunggulan dan kekurangan masing masing. (Kumar & Chauhan, 2011). Keunggulan dan kelemahan di rangkum pada Tabel 2.2

Tabel 2.2 Perbandingan Keunggulan Dan Kekurangan Protokol

| Scheduling Protocol | Time Latency | Co-Channel Interference | Time Synchronization | Comm. Pattern support | Adaptivity to change |
|---------------------|--------------|-------------------------|---------------------------------|-----------------------|----------------------|
| EEWS | High | Yes | Reduced cost of Synchronization | All | Good |
| MDS | Low | Yes | Yes | All | Good |
| DAS | Low | No | Yes | All | Good |
| TRAMA | High | No | Yes | All | Good |
| DMAC | Low | No | Yes | Convergecast | Weak |

Sumber: Kumar dan Chauhan (2011)

Dari tabel di atas dapat disimpulkan bahwa protokol TDMA memiliki banyak sekali kegunaan yang dapat dikembangkan sesuai kebutuhan pada sistem. Selain itu menurut penulis, TDMA memiliki keunggulan yaitu bebas dari tabrakan data, namun dalam beberapa protokol ini menyebabkan gangguan dikarenakan pengulangan penggunaan slot. Sinkronisasi waktu diperlukan dalam protokol ini karena masalah perbedaan waktu pada node. Masalah lain pada sistem TDMA adalah masalah adaptasi terhadap perubahan topologi.

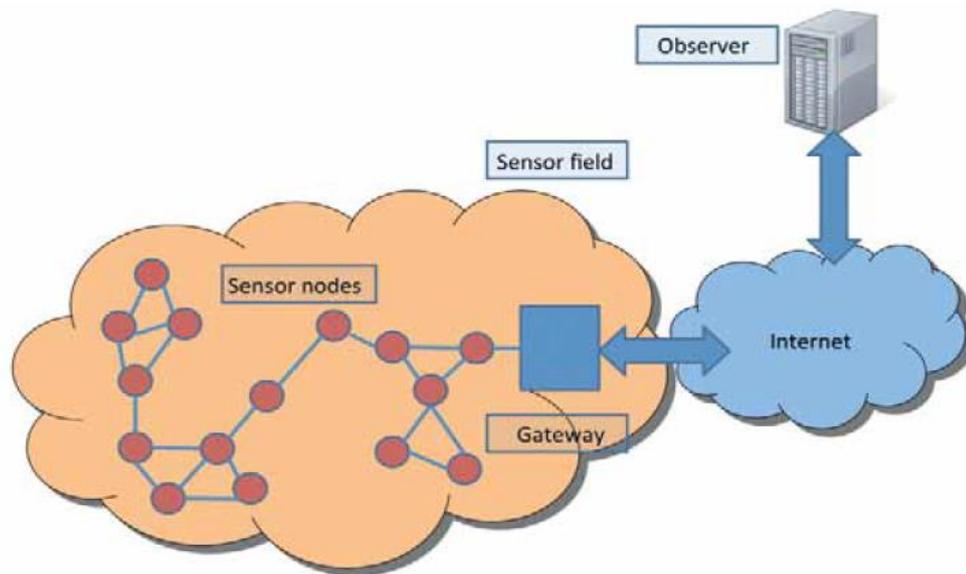
2.2 Dasar Teori

Pada bagian ini, menjelaskan tentang beberapa definisi tentang teori dan komponen yang digunakan pada penelitian ini. Beberapa hal seperti penjelasan *Wireless Sensor Network*, *Time Synchronization*, *Time Division Multiple Access*, Mikrokontroler dan module nRF2401I.

2.2.1 *Wireless Sensor Network*

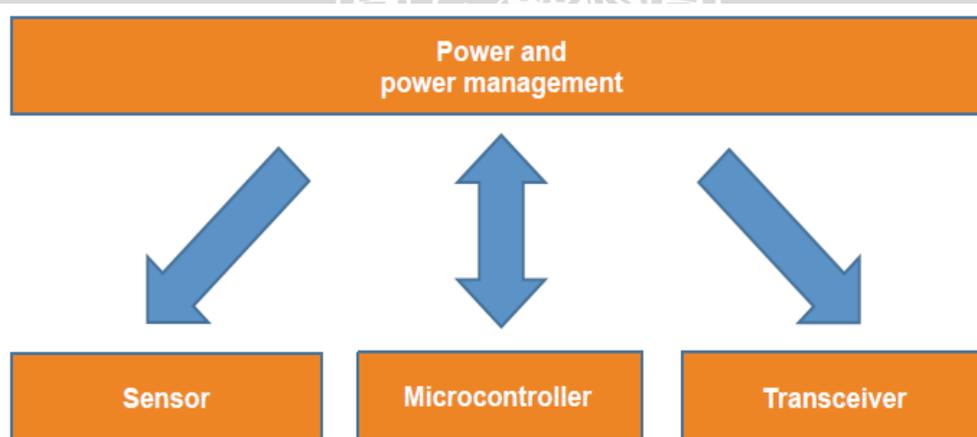
Wireless sensor network (WSN) secara umum dapat digambarkan sebagai jaringan node yang kooperatif yang dapat memantau daerah sekitar dan melakukan suatu aksi. Hal ini memungkinkan interaksi antara manusia, komputer dan lingkungan sekitarnya. Pada WSN biasanya terdapat sensor node, aktuator node, gateway dan klien. Untuk memantau suatu daerah node disebarkan ke beberapa titik di daerah pantauan dan node-node tersebut membentuk jaringan. Hasil pantauan dari tiap-tiap node dikirimkan ke gateway node. Yang terakhir gateway node mengirimkan data menuju ke node observer melalui internet atau satelit. Pada node observer pengguna dapat mengkonfigurasi dan mengelola WSN, memberikan perintah pemantauan dan pengumpulan data hasil pemantauan (Yinbiao, et al., 2014). Ilustrasi jaringan WSN dapat dilihat pada Gambar 2.2.





Gambar 2.2 Ilustrasi WSN
 Sumber : Yinbiao, et al. (2014)

Untuk sensor node umumnya meliputi empat bagian: sumber daya, sensor, mikrokontroler, dan sebuah *transceiver* nirkabel. Sumber daya harus memenuhi kebutuhan daya diperlukan sistem. Sensor bertugas mengumpulkan dan mengubah sinyal, seperti cahaya dan getaran, menjadi sinyal listrik dan kemudian diberikan ke mikrokontroler. Mikrokontroler menerima data dari sensor dan memproses data yang sesuai. Sementara wireless *transceiver* atau modul *radio frequency* (RF) digunakan untuk mengirimkan data (Yinbiao, et al., 2014).



Gambar 2.3 Komponen Utama Sensor Node
 Sumber : Yinbiao, et al. (2014)

2.2.2 Sinkronisasi Waktu *Reference Broadcast Synchronization*

Sinkronisasi waktu adalah suatu cara menciptakan kesetaraan skala waktu yang digunakan setiap perangkat dalam jaringan dan merupakan kebutuhan bagi banyak aplikasi (Lasassmeh & Conrad, 2010). Selain itu

sinkronisasi merupakan bagian penting dari infrastruktur untuk sistem terdistribusi. Jaringan sensor nirkabel sangat membutuhkan sinkronisasi waktu yaitu untuk mengintegrasikan pengukur kecepatan kendaraan, mengkoordinasikan penjadwalan di masa depan, untuk pemesanan login acara selama sistem debugging, dan sebagainya (Elson & Estrin, 2001).

Dalam metode RBS terdapat satu node yang berperan sebagai pengirim atau disebut node *beacon* dan beberapa node yang berperan sebagai *receiver*. Node *beacon* merupakan node yang memberikan paket referensi terhadap *receiver*. Paket referensi tidak berisi waktu dari *beacon*, tetapi paket hanya berisi perintah agar *receiver* melakukan sinkronisasi waktu. Tahap-tahap dari RBS yaitu:

1. Node *beacon broadcasts* paket referensi kepada *receiver*.
2. Setiap *receiver* menyimpan waktu diterimanya paket menurut waktu local tiap *receiver*.
3. *Receiver* saling bertukar hasil observasi paket referensi (waktu observasi)
4. *Receiver* menghitung *phase offset* (perbedaan waktu) dengan rumus:

$$PO : \frac{1}{m} \sum_{k=1}^m (T_{j,k} - T_{i,k}) \quad (2.1)$$

Dimana m adalah banyak paket referensi, T_j merupakan waktu observasi node tetangga dan T_i adalah Waktu observasi node. Setelah nilai *offset* didapatkan, maka sinkronisasi waktu dapat dilakukan dengan mengolah nilai *offset* tersebut dengan waktu lokal. Cara tersebut dilakukan sampai ke pada node terakhir ke node pertama selesai melakukan sinkronisasi waktu (Elson, et al., 2002).

2.2.3 Penjadwalan *Time Division Multiple Access*

Time Division Multiple Access (TDMA) adalah teknik akses media yang memungkinkan beberapa node mengirimkan paket pada channel yang sama, channel diakses secara bergantian sesuai periode waktu (slot waktu) yang diberikan kepada setiap perangkat, sehingga tidak akan terjadi pengiriman paket yang bersamaan yang berakibat adanya tabrakan data (Chu, et al., 2015). Pengiriman data berlangsung sesuai dengan slot waktu yang telah ditentukan. Node 1 dapat mengirimkan data pada slot waktu untuk node 1, node 2 dapat mengirimkan berupa data pada slot waktu untuk node 2, dan seterusnya. Disela-sela slot waktu biasanya diberikan *guard time* yang berguna untuk waktu transisi mode dari tiap node. Ilustrasi slot waktu dapat dilihat pada Gambar 2.4.



Gambar 2.4 Ilustrasi Slot Waktu TDMA

2.2.4 Mikrokontroler Arduino Pro Mini

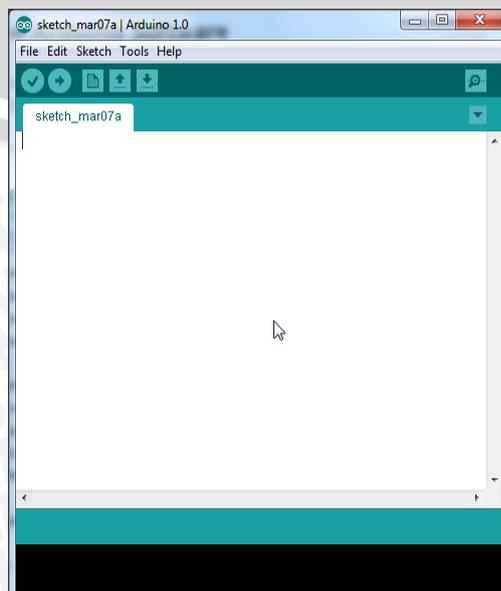
Mikrokontroler merupakan sebuah chip yang memiliki kemampuan untuk melakukan komputasi dan pengolahan data. Pada mikrokontroler terdapat unit pemrosesan, memori, dan pin *input output*. Salah satu mikrokontroler adalah Arduino pro mini. Mikrokontroler ini berbasis ATmega328. Memiliki 14 digital input/output pin (6 pin PWM), 8 pin analog, kristal osilator 16 MHz serta tombol reset. Untuk menanamkan program ke dalam Arduino pro mini dapat menggunakan software Arduino IDE. Bentuk dari arduino pro mini ditunjukkan dalam Gambar 2.5.



Gambar 2.5 Arduino Pro mini
Sumber : Arduino (2016)

2.2.5 Arduino IDE

Arduino *Integrated Development Environment* (IDE) adalah sebuah *software* yang dibuat oleh Arduino yang khusus digunakan perangkat Arduino. Arduino IDE berguna untuk menuliskan program, *compile* program dan melakukan *upload* ke dalam memory mikrokontroler. Selain itu juga terdapat serial monitor yang dapat digunakan untuk melihat jalannya program yang

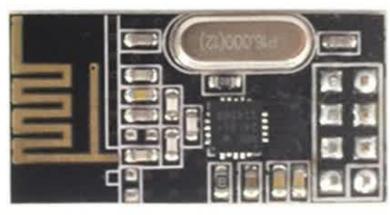


Gambar 2.6 Tampilan Arduino IDE

berjalan pada perangkat arduino. Berikut tampilan Arduino IDE.

2.2.6 Modul nRf24L01

Modul nRF24L01 adalah sebuah modul tanpa kabel yang memanfaatkan gelombang *radio frequency* (RF) 2.4GHz. Modul ini menggunakan komunikasi *Serial Peripheral Interface* (SPI). Selain itu modul ini menggunakan daya yang rendah. Dengan penggunaan daya yang rendah memungkinkan modul ini dapat digunakan dalam jangka waktu yang lama. Berikut adalah modul nRF24L01 yang ditunjukkan pada Gambar 2.7.

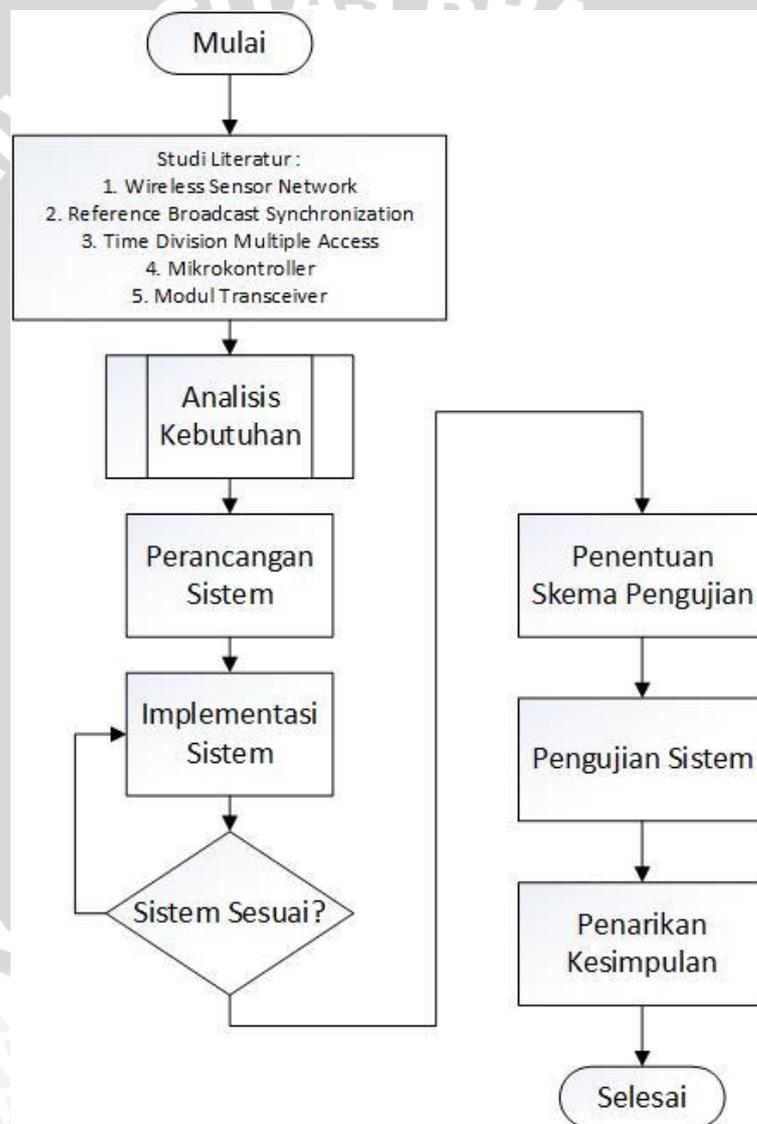


Gambar 2.7 Modul *Transceiver* nRF24L01
Sumber : Chantrell (2013)



BAB 3 METODE PENELITIAN

Pengembangan sistem dalam penelitian diawali dengan studi literatur terkait dengan tinjauan pustaka dari penelitian-penelitian sebelumnya dan dasar teori yang terkait dengan WSN dan sinkronisasi waktu. Penelitian yang dilakukan bersifat implementatif, pada tahap awal dilakukan analisa kebutuhan fungsional sistem serta menggambarkan sistem secara garis besar dalam bentuk blok diagram. Blok diagram akan menunjukkan garis besar sistem secara keseluruhan, dan dibentuk melalui beberapa sub sistem yang memiliki fungsi masing-masing. Proses perancangan akan dilakukan berdasarkan desain blok diagram sistem dan diimplementasikan dan pengujian sistem tersebut berdasarkan analisa kebutuhan sistem yang dilakukan. Diagram alir penelitian ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

3.1 Studi Literatur

Studi literatur dilaksanakan dengan melaksanakan pemahaman terhadap dasar teori yang terkait dengan:

1. *Wireless Sensor Network* (WSN)

Studi literatur terkait dengan WSN dilakukan dengan mencari beberapa paper survey yang menjelaskan secara umum bagaimana teori dan implementasi WSN dilakukan. Pada tahapan ini dipelajari bagaimana struktur umum dari topologi WSN, kebutuhan minimal dan jenis motes atau sensor node yang digunakan pada sebuah kondisi tertentu.

2. Metode *Reference Broadcast Synchronization* (RBS)

Pada bagian ini dilakukan kajian terkait dengan penelitian terdahulu bidang WSN yang telah menerapkan metode RBS sebagai metode mensinkronisasikan waktu pada jaringan WSN. Disini ditemukan dan dipelajari tahap dan penggunaan metode RBS dalam WSN.

3. Metode *Time Division Multiple Access* (TDMA)

Pada bagian ini juga dilakukan kajian terkait dengan penelitian terdahulu bidang WSN yang telah menerapkan metode TDMA sebagai metode pembagian waktu pengiriman-penerimaan pada tiap node dalam jaringan WSN. Disini ditemukan dan dipelajari tata cara penggunaan metode TDMA dalam WSN.

4. Mikrokontroler

Literatur terkait dengan mikrokontroler yang yang berguna untuk mengetahui karakteristik dan cara mengakses mikrokontroler.

5. Modul *Transceiver*

Literatur terkait dengan modul *transceiver* yang digunakan agar modul ini mampu diakses melalui mikrokontroler.

3.2 Analisis Kebutuhan

Analisa kebutuhan ditujukan untuk melakukan analisis pada beberapa kebutuhan yang diperlukan sistem pada penelitian. Analisa kebutuhan pada penelitian dibagi menjadi 3, yaitu kebutuhan sistem, kebutuhan perangkat keras dan perangkat lunak.

3.2.1 Kebutuhan Sistem

Beberapa kebutuhan sistem yang diperlukan pada perancangan sistem antara lain:

1. Node *beacon* harus mengirimkan paket referensi kepada setiap node.
2. Setiap node *receiver* harus dapat mensinkronisasikan waktunya dengan metode RBS.

3. Setelah memiliki waktu yang sinkron, setiap node harus dapat saling mengirim dan menerima data sesuai jadwal yang telah diberikan.

3.2.2 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras pada sistem ini antara lain:

1. 1 Unit Komputer.
2. 5 unit mikrokontroller Arduino pro mini
3. 5 unit modul tranceiver nRF24L01.

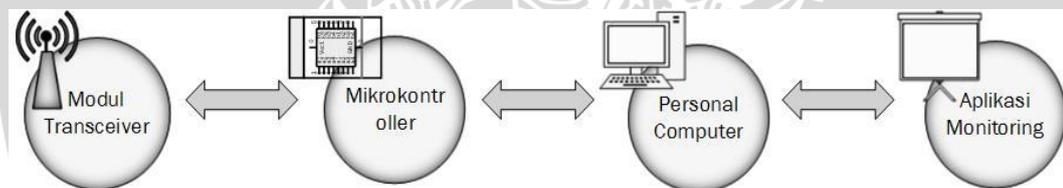
3.2.3 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak pada sistem ini antara lain:

1. Arduino IDE
2. Bahasa pemrograman C (Arduino)

3.3 Perancangan Sistem

Pada perancangan sistem ini menjelaskan tentang perangkat yang digunakan beserta spesifikasinya serta alur komunikasi sistem hingga sistem dapat berjalan. Perangkat keras yang digunakan terdiri dari komponen yang pembentuk sebuah node (perangkat WSN) yang menjadi aktor yang menjalankan sistem ini. Berikut diagram blok sensor node pada sistem ini :

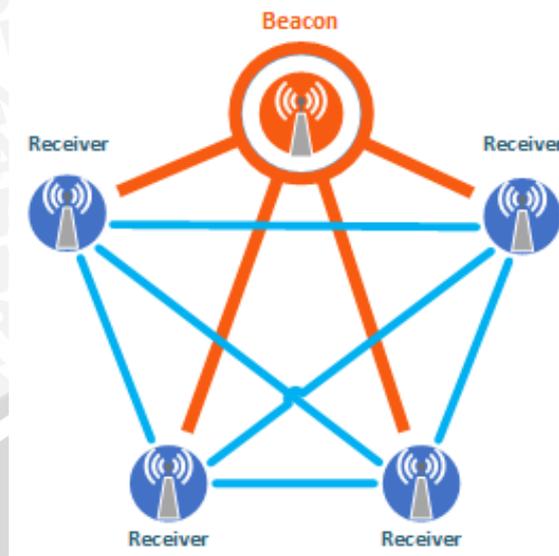


Gambar 3.2 Diagram blok perangkat sensor node

Pada Gambar 3.2 menjelaskan bahwa:

1. Modul tranceiver berguna sebagai perangkat komunikasi nirkabel yang berfungsi untuk mengkomunikasikan tiap-tiap node dalam sistem ini.
2. Mikrokontroler berfungsi sebagai pemberi perintah dan pengolah data hasil komunikasi.
3. Personal Comuputer befungsi untuk melihat data hasil pengolahan mikrokontroler.
4. Untuk melihat hasil pengolahan data personal komputer menggunakan aplikasi monitoring untuk memudahkan pengguna untuk memantau data.

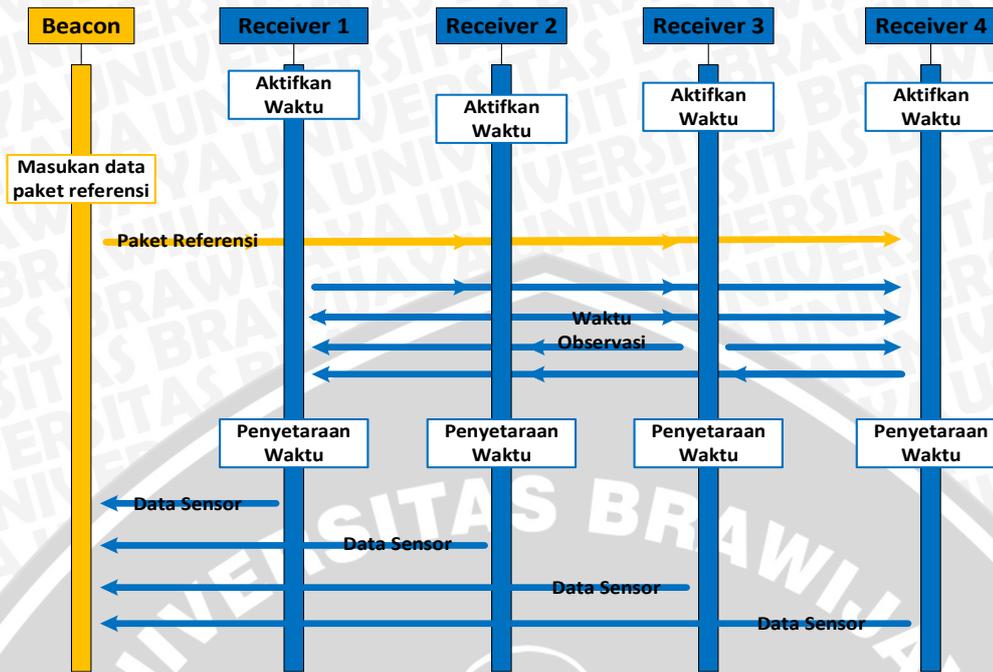
Pada sistem ini, node dibedakan menjadi dua yaitu node *beacon* dan node *receiver*. Node *beacon* berfungsi sebagai pemberi tanda sinkronisasi waktu dan node tujuan pengiriman data setelah terjadi penjadwalan. Sementara node *receiver* merupakan node subyek dari sinkronisasi dan pengirim data. Setiap node membentuk suatu jaringan dan dapat berkomunikasi langsung dengan node lain (single hop). Berikut gambaran topologi jaringan yang digunakan:



Gambar 3.3 Topologi Jaringan

Alur kerja sistem ini dimulai dari mengaktifkan setiap node yang digunakan. Khusus untuk node *receiver* diberikan waktu lokal masing masing. Yang pertama kali dilakukan oleh node beacon adalah mengirimkan paket referensi atau tanda paket tanda sinkronisasi kepada setiap node receiver. Kemudian node receiver menyimpan waktu kedatangan paket (waktu observasi). Selanjutnya node receiver saling bertukar waktu observasi dan menghitung selisih waktu. Setelah mengetahui selisih waktu, tiap node melakukan penyetaraan waktu dengan mengolah waktu lokal tiap node dengan selisih waktu. Yang terakhir node receiver bergantian mengirimkan data sesuai dengan jadwal yang telah diberikan.





Gambar 3.4 Diagram Alur Kerja Sistem

3.4 Implementasi Sistem

Implementasi merupakan perancangan sistem sampai pada hasil akhir. Tahapan dari implementasi sistem ini yaitu :

1. Implementasi perangkat arduino pro mini dengan module nRF2401l yang dapat dikontrol dan dimonitor oleh komputer.
2. Implementasi metode RBS pada sistem sebagai metode sinkronisasi waktu pada node *receiver*.
3. Implementasi metode TDMA dalam pengiriman data sesuai slot waktu sesuai waktu yang didapat dari metode RBS.

Dari implementasi sistem ini, diharapkan mendapatkan hasil sebagai berikut:

1. Perangkat arduino pro mini dengan module nRF2401l yang dapat dikontrol dan dimonitor oleh komputer.
2. Terjadinya sinkronisasi waktu antar node pada sistem dengan menggunakan metode RBS.
3. Terhindarnya tabrakan data karena adanya pengiriman data yang diatur oleh metode TDMA sesuai waktu yang didapatkan dari metode RBS.

3.5 Pengujian dan Analisis

Pengujian dari penelitian ini yaitu dengan menguji coba kinerja metode RBS dan TDMA. Pengujian antara lain yaitu:

1. Menguji kemampuan fungsional dari perangkat sensor node yang menggunakan mikrokontroler Arduino Pro Mini dan modul nRF24L01.
2. Menguji kualitas metode sinkronisasi waktu RBS node dengan indikasi keberhasilan ketepatan waktu yang tersinkronisasi pada setiap node receiver.
3. Menguji peran metode TDMA dalam memberikan jadwal pengiriman data sesuai slot waktu yang ditentukan dengan indikasi keberhasilan semakin sedikitnya terjadi tabrakan data.

Dari pengujian tersebut, analisis dibutuhkan sebagai landasan penyesuaian metode pada sistem. Sehingga metode ini dapat digunakan secara maksimal pada sistem. Hal ini juga berdampak pada terciptanya sistem WSN yang menjamin terjadinya pertukaran data yang akurat.

3.6 Kesimpulan

Kesimpulan merupakan gambaran dari hasil implementasi metode Reference Broadcast Synchronization dan Time Division Multiple Access. Kesimpulan juga merupakan jawaban dari rumusan masalah. Terakhir adalah penulisan saran yang dilakukan untuk memperbaiki kekurangan yang ada pada penelitian dan mempertimbangkan penggunaan metode Reference Broadcast Synchronization dan Time division Multiple Access.



BAB 4 REKAYASA KEBUTUHAN

Pada rekayasa kebutuhan menjelaskan tentang segala sesuatu yang dibutuhkan sistem. Dimulai dari gambaran umum sistem, kebutuhan sistem dan yang terakhir adalah alur kerja secara umum sistem.

4.1 Gambaran Umum Sistem

Sistem ini secara garis besar adalah untuk memecahkan masalah dalam pertukaran data pada *Wireless Sensor Network* (WSN) yaitu masalah tabrakan data dikarenakan setiap perangkat pada jaringan WSN mengirimkan datanya secara broadcast. Salah satu metode adalah menggunakan metode *Time Division Multiple Access* (TDMA). Untuk menunjang bekerjanya metode ini, setiap node dalam jaringan harus memiliki waktu yang setara. Hal ini dapat diatasi dengan metode *Reference Broadcast Synchronization* (RBS).

4.1.1 Tujuan

Sistem ini bertujuan untuk menghindari adanya tabrakan data pada suatu infrastruktur *Wireless Sensor Network* (WSN). Sistem ini menggunakan beberapa node yang menjadi aktor dalam berlangsungnya sistem ini. Sementara pengguna dapat memberikan input yang dibutuhkan dalam sistem ini. Selain itu pengguna juga dapat memonitoring pengiriman data melalui aplikasi monitoring.

Pada sistem ini menggunakan metode *Time Multiple Division Access* (TDMA) sebagai metode penjadwalan pengiriman data. Nantinya terdapat slot-slot waktu yang telah disediakan dan pada slot waktu itu setiap node memiliki wewenang untuk mengirim atau menerima data. Sehingga tidak ada tabrakan data. Tetapi metode ini membutuhkan waktu yang sinkron (setara) pada setiap node. Karena itu, sistem ini menggunakan metode *Reference Broadcast Synchronization* (RBS) untuk metode sinkronisasi waktu. Metode RBS membutuhkan paket referensi yang dikirim node *beacon* kepada node *receiver*. Node *receiver* menerima dan mencatat sebagai waktu observasi. Setelah itu waktu observasi tersebut saling dikirimkan untuk mendapatkan fase offset (selisih waktu) relatif. Nantinya fase offset digunakan untuk menyamakan waktu pada setiap node dengan cara mengurangi waktu node dengan fase offset yang telah didapatkan.

4.1.2 Kegunaan

Sistem ini berguna untuk mengatasi pengiriman dan penerimaan data pada WSN. Menggunakan metode RBS sebagai sinkronisasi waktu antar node dan menggunakan metode TDMA untuk memberikan jadwal penerimaan dan pengiriman data. Dengan bekerjanya sistem ini memberikan jaminan pertukaran data yang bebas dari *data collision*. Sehingga informasi dapat diterima dengan baik.

4.1.3 Karakteristik Pengguna

Pengguna sistem bertindak sebagai pemberi input banyaknya paket referensi dari node *beacon*. Paket referensi digunakan sebagai trigger dimulainya sinkronisasi waktu antar node. Pengguna juga dapat melihat data yang diproses dari tiap node setelah tiap node memiliki jadwal pengiriman data.

4.1.4 Lingkungan Operasi

Pada persyaratan kebutuhan lingkungan yang mendukung kebutuhan sistem yaitu:

1. Setiap node memiliki frekuensi dan channel yang sama.
2. Setiap node berada pada satu wilayah dimana setiap node dapat menjangkau node lainnya (radius 1-8 meter).

4.2 Kebutuhan Sistem

Kebutuhan sistem bertujuan untuk menggambarkan kebutuhan-kebutuhan dari sistem. Kebutuhan fungsional dan non-fungsional dianalisis sesuai dengan kebutuhan sistem sehingga mempermudah dalam mendesain dan mengimplementasikan sistem.

4.2.1 Kebutuhan Fungsional

Kebutuhan Fungsional yang harus dipenuhi oleh sistem ini antara lain:

1. Node *Beacon* mengirimkan paket referensi secara broadcast kepada setiap *receiver*
Node *beacon* yang merupakan node yang mengirimkan paket referensi harus mampu mengirimkan paketnya kepada setiap node *receiver* yang beradapa pada setiap node pada jaringan. Pengiriman paket ini merupakan tanda dimulainya sinkronisasi waktu yang dilakukan oleh node *receiver*.
2. Node *Receiver* menerima data paket referensi dari node *beacon*.
Data paket referensi yang diterima oleh *receiver* sesuai dengan data yang dikirimkan oleh node *beacon*. Setelah itu node *receiver* juga menyimpan waktu lokal diterimanya paket referensi. Selain itu node *receiver* dapat menghitung rata-rata waktu apabila node *beacon* mengirim lebih dari satu paket referensi.
3. Node *receiver* bertukar data waktu observasi dan mensinkronisasikan waktu.

Paket waktu observasi saling ditukarkan dan setelah setiap node menerima paket waktu observasi dari tiap node. Maka tiap node membandingkan dengan waktu observasi node itu sendiri. Setelah itu maka didapatkan phase offset (pengurangan dari waktu observasi *receiver* lain dengan waktu observasi node *receiver* sendiri). Variabel phase offset inilah yang digunakan sebagai penyetara waktu sehingga tiap node memiliki waktu yang sinkron (relatif sama).

4. Sistem memberikan penjadwalan pengiriman dan penerimaan data dengan metode TDMA.

Node *receiver* memiliki waktu yang sinkron dan dari waktu yang sinkron tiap node mendapatkan slot waktu sesuai IDnya. Pada satu slot waktu setiap node *receiver* tidak boleh memiliki aksi pengiriman data.

4.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional meliputi beberapa kebutuhan yang harus dipenuhi agar sistem berjalan dengan baik. Pada sistem ini kebutuhan fungsionalnya antara lain:

1. Sistem dapat melakukan sinkronisasi waktu menggunakan metode RBS dengan range *error* antar node kurang dari 50 milisecond

Dengan menggunakan metode RBS, maka didapatkan sinkronisasi waktu antar node dengan range *error* yang kecil. Dengan range *error* yang kecil maka dapat dikatakan setiap node memiliki waktu yang relatif dan waktu yang relatif ini dapat meningkatkan akurasi dari penjadwalan yang menggunakan metode TDMA.

2. Setiap node memberikan penjadwalan menggunakan metode TDMA. Pada tiap jadwal pengiriman data terdapat waktu toleransi (*idle*)

Setelah setiap node memiliki jadwal pengiriman. Maka tiap node mengirimkan paket datanya sesuai slot-slot waktu yang didapatkan. Dan pada setiap slot waktu diselipak waktu *idle* yang digunakan untuk waktu toleransi *error* dari sinkronisasi waktu. Selain itu pemberian waktu *idle* juga untuk menstabilkan pengiriman data.

4.2.3 Kebutuhan Perangkat Keras

Setiap node pada penelitian ini memiliki kebutuhan perangkat keras yaitu:

1. Mikrokontroler Arduino Pro mini
Mikrokontroler merupakan perangkat yang ditanami program sehingga bekerja sesuai yang diinginkan. Selain itu juga merupakan pengolah data yang dikirimkan atau diterima.
2. Modul *Transceiver* nRF24L01
Pada setiap node terdapat modul *transceiver*. Dimana modul ini digunakan untuk saling berhubungan dengan node lain. Modul *transceiver* menggunakan frekuensi yang sama untuk saling berhubungan.
3. Komputer
Komputer digunakan untuk memonitoring pengolahan data pada mikrokontroler. Selain itu juga untuk memberkan input yang dibutuhkan saat program berjalan.

4.2.4 Kebutuhan Perangkat Lunak

Setiap node pada sistem ini memiliki kebutuhan antarmuka perangkat lunak yaitu:

1. Arduino RF24 Library
Library ini digunakan untuk mengakses modul transceiver melalui mikrokontroler. Sehingga modul transceiver dapat melakukan perintah yang diberikan oleh mikrokontroler.
2. Arduino IDE
Perangkat lunak ini digunakan untuk menuliskan program, compile dan upload program ke mikrokontroler. Selain itu pada perangkat lunak ini terdapat fitur yang dapat digunakan untuk membantu komputer untuk melakukan monitoring jalannya program, yaitu berupa *serial monitor*.

4.2.5 Kebutuhan Komunikasi

Komunikasi yang terjadi pada sistem memanfaatkan modul *transceiver* nRF24L01. Dimana modul ini bekerja pada frekuensi 2.4GHz dan banyak kanal 126 dengan range antara 0-125. Data yang dikirimkan berbentuk sebuah paket yang berukuran 8 byte yang dibagi menjadi empat bagian yang berisi informasi dari tiap node.



BAB 5 PERANCANGAN DAN IMPLEMENTASI

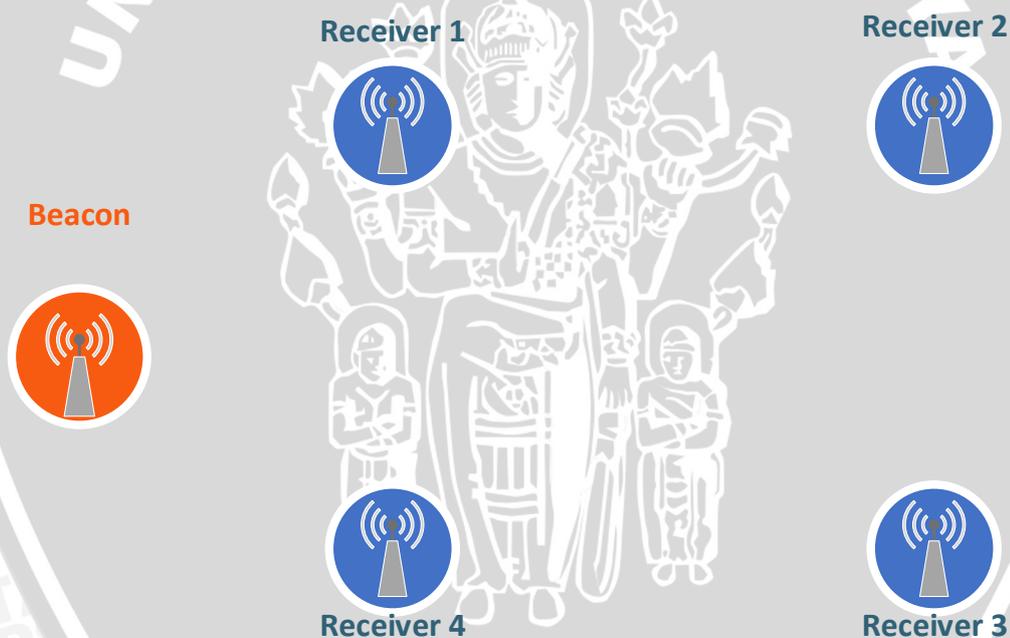
Pada bab ini menjelaskan tentang perancangan sistem dari penelitian, serta implementasi sistem berupa implementasi perangkat keras, perangkat lunak, dan implementasi pendukung sistem lainnya.

5.1 Perancangan Sistem

Secara umum, perancangan terdiri dari perancangan perangkat keras, perancangan perangkat lunak dan perangkat keras serta perancangan paket data dan komunikasi yang terjadi pada jaringan ini.

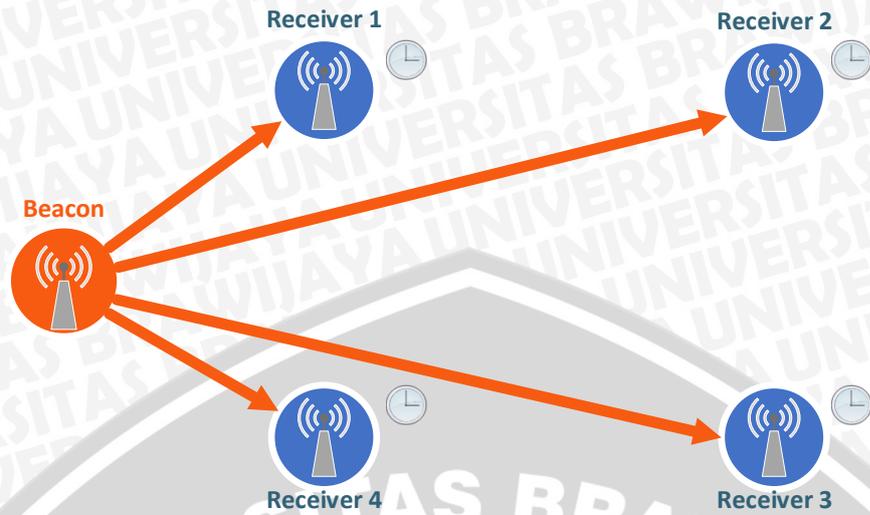
5.1.1 Perancangan Alur Kerja Sistem

Pada penelitian ini terdapat satu node *beacon* dan dua-empat node *receiver*. Setiap node berkomunikasi secara nirkabel hingga membentuk suatu jaringan. Selain itu juga terdapat beberapa tahap yang harus dilakukan oleh tiap node agar sistem berjalan sesuai dengan yang diharapkan. Untuk susunan peletakan node *beacon* dan *receiver* dapat dilihat pada Gambar 5.1.



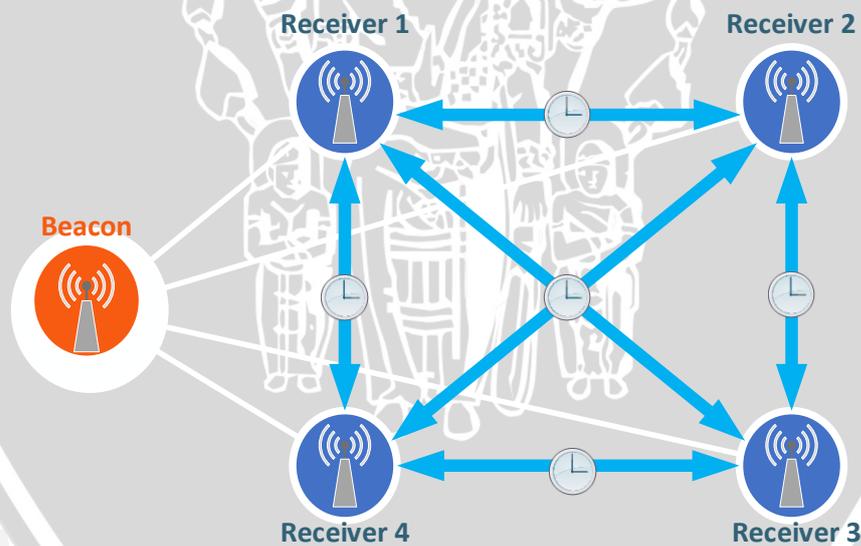
Gambar 5.1 Susunan Node *Beacon* dan *Receiver*

Pada sistem ini hanya node *receiver* yang melakukan sinkronisasi sementara node *beacon* hanya menjadi pemberi perintah untuk melakukan sinkronisasi. Pada awal sistem ini bekerja setiap node *receiver* memiliki waktu lokalnya sendiri yang berbeda dari node *receiver* lain. Kemudian setelah itu node *beacon* memberikan perintah untuk melakukan sinkronisasi dengan mengirimkan paket referensi. Sementara node *receiver* yang menerima paket referensi mencatat waktu kedatangan paket sesuai waktu lokal tiap node *receiver* yang disebut waktu observasi. Lihat ilustrasi pengiriman paket referensi pada Gambar 5.2.



Gambar 5.2 Ilustrasi Pengiriman Paket Referensi dari Node Beacon

Tahap Selanjutnya adalah sinkronisasi yang dilakukan oleh tiap node *receiver*. Dimana tiap node *receiver* bertukar waktu observasinya dengan node lain. Dikarenakan ini merupakan sinkronisasi antar node *receiver*, node *beacon* tidak ikut serta dalam pertukaran waktu observasi ini. Sementara pertukaran data dalam tahap ini dilakukan secara estafet dimulai dari node *receiver* 1 hingga node *receiver* 4. Berikut ilustrasi pengiriman paket waktu observasi dari tiap node:



Gambar 5.3 Ilustrasi Pengiriman Paket Waktu Observasi

Pertukaran waktu observasi ini dilakukan agar tiap node mengetahui waktu diterimanya paket referensi dari node *receiver* lain. Sehingga dapat mengetahui selisih waktu atau *phase offset* dari node *receiver* lain. Sebagai contoh lihat Tabel 5.1 yang menunjukkan pertukaran waktu observasi dimana *receiver* 1 memiliki waktu observasi 10, *receiver* 2 adalah 20, *receiver* 3 adalah 30 dan *receiver* 4 adalah 40. Waktu yang dikirimkan dalam satuan detik.

Tabel 5.1 Contoh Pertukaran Waktu Observasi

| | | Waktu Observasi dari Node - | | | |
|----------|------------|-----------------------------|------------|------------|------------|
| | | Receiver 1 | Receiver 2 | Receiver 3 | Receiver 4 |
| Penerima | Receiver 1 | | 20 | 30 | 40 |
| | Receiver 2 | 10 | | 30 | 40 |
| | Receiver 3 | 10 | 20 | | 40 |
| | Receiver 4 | 10 | 20 | 30 | |

Kemudian tiap waktu observasi yang diterima dari node *receiver* lain dibandingkan dengan waktu observasi milik node *receiver* itu sendiri sehingga didapat *phase offset*. Untuk melakukan sinkronisasi waktu, *phase offset* yang digunakan adalah yang memiliki nilai ternegatif, sementara apabila terdapat *receiver* yang mendapat nilai *phase offset* yang positif maka nilai *phase offset* dianggap 0. Lihat Tabel 5.2 yang merupakan contoh proses didaptkannya nilai *phase offset*.

Tabel 5.2 Contoh Perhitungan Phase Offset

| | Waktu Observasi yang diterima | Waktu Observasi Sendiri | Selisih Waktu (<i>Phase Offset</i>) | <i>Phase Offset</i> Yang digunakan |
|------------|-------------------------------|-------------------------|---------------------------------------|------------------------------------|
| Receiver 1 | 20 | 10 | 10 | 0 |
| | 30 | | 20 | |
| | 40 | | 30 | |
| Receiver 2 | 10 | 20 | -10 | -10 |
| | 30 | | 10 | |
| | 40 | | 20 | |
| Receiver 3 | 10 | 30 | -20 | -20 |
| | 20 | | -10 | |
| | 40 | | 10 | |
| Receiver 4 | 10 | 40 | -30 | -30 |
| | 20 | | -20 | |
| | 30 | | -10 | |

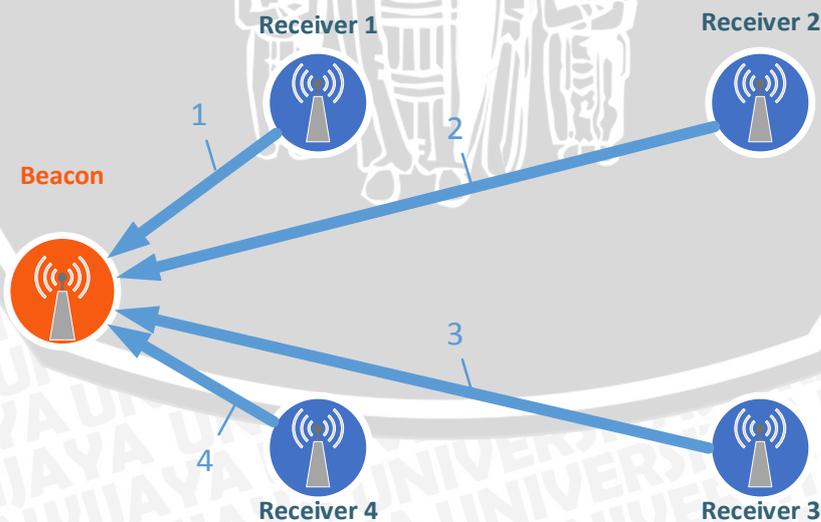
Tahap berikutnya adalah melakukan penyetaraan waktu dengan cara menambahkan waktu lokal tiap receiver dengan *phase offset* yang telah didapatkan. Waktu lokal yang digunakan adalah waktu lokal yang terbaru atau waktu lokal setelah terjadi proses pertukan waktu observasi dan perhitungan *phase offset* (contoh: jika *receiver* 1 waktu lokal saat menerima paket referensi

dari node beacon atau waktu observasi adalah detik ke-10, kemudian waktu proses pertukan waktu observasi dan perhitungan *phase offset* adalah 1 detik maka waktu lokal receiver 1 adalah detik ke-11). Kemudian waktu lokal ditambahkan dengan *phase offset* sehingga didapatkan nilai waktu setara pada tiap node *receiver*. Hal terakhir yang dilakukan oleh *receiver* saat sinkronisasi waktu adalah menjadikan nilai waktu setara menjadi waktu lokal sehingga didapatkan waktu lokal yang setara dari tiap node *receiver*. Perhitungan keseluruhan dapat dilihat pada Tabel 5.3.

Tabel 5.3 Contoh Perhitungan Waktu Setara

| | Waktu observasi | Waktu Proses | Waktu Lokal Terkini | Phase Offset | Waktu Setara |
|------------|-----------------|--------------|---------------------|--------------|--------------|
| Receiver 1 | 10 | 1 | 11 | 0 | 11 |
| Receiver 2 | 20 | | 21 | -10 | 11 |
| Receiver 3 | 30 | | 31 | -20 | 11 |
| Receiver 4 | 40 | | 41 | -30 | 11 |

Setelah waktu tiap node *receiver* setara, maka tiap node *receiver* melakukan penjadwalan menggunakan metode TDMA. Penjadwalan ini dilakukan menggunakan sesuai id node *receiver* dan slot waktu. Nilai slot waktu dikirimkan oleh node *beacon* didalam paket referensi. Node *receiver* dengan id 1 mengisi slot waktu pertama dan diikuti node id selanjutnya pada slot waktu berikutnya. Setiap node *receiver* mengirimkan data secara bergantian sesuai penjadwalan. Sehingga didapatkan pengiriman data tanpa adanya tabrakan data. Tugas terakhir node *beacon* adalah harus mampu menerima setiap data yang telah dikirimkan dan menampilkannya pada aplikasi monitoring.



Gambar 5.4 Pengiriman Paket Dari Receiver ke Beacon



5.1.2 Perancangan Perangkat Keras

Terdapat dua jenis perangkat yang digunakan pada sistem ini. Perangkat tersebut adalah node *beacon* dan node *receiver*. Masing masing node memiliki komponen yang sama yaitu modul *transceiver* nRF24L01, mikrokontroler Arduino Pro Mini yang digunakan untuk menghubungkan mikrokontroler dengan *Personal Computer* (PC), PC sendiri digunakan untuk melihat hasil jalannya program pada mikrokontroler dan aplikasi monitoring menggunakan serial monitor dari arduino sendiri. Untuk lebih jelasnya bagaimana komponen tersebut terhubung, dapat dilihat pada diagram blok Gambar 5.5.



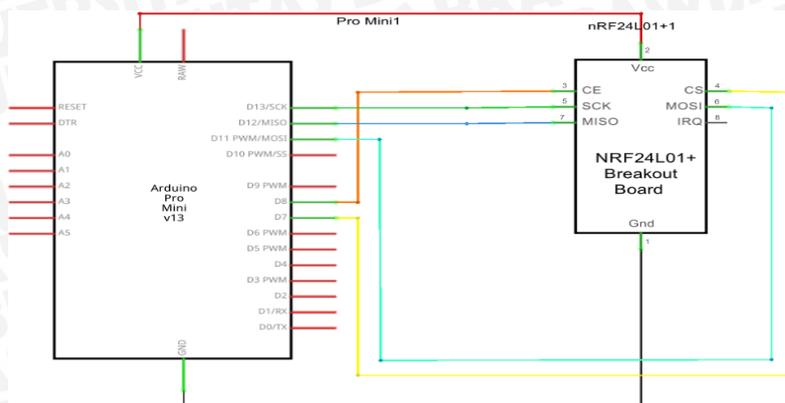
Gambar 5.5 Diagram Blok Komponen Node

Modul transceiver nRF24L01 dan mikrokontroler Arduino Pro mini terhubung melalui pin yang terdapat pada kedua perangkat tersebut. Pada Tabel 5.4 dijelaskan konfigurasi pin yang harus dihubungkan agar kedua perangkat tersebut dapat digunakan sebagai sebuah node.

Tabel 5.4 Konfigurasi Pin Arduino Pro Mini dan nRF24L01

| nRF24L01 | Pin nRF24L01 | Pin Pro Mini |
|------------|--------------|--------------|
| GND | 1 | GND |
| Vcc / 3.3V | 2 | VCC |
| CE | 3 | 9 |
| CSN | 4 | 10 |
| SCK | 5 | 13 |
| MOSI | 6 | 11 |
| MISO | 7 | 12 |

Dari Tabel 5.1, dapat digambarkan sebuah rancangan skematik. Rancangan ini mewakili perangkat yang digunakan dan gambaran dari konfigurasi pin yang dihubungkan menggunakan garis. Rancangan ini dapat dilihat pada Gambar 5.6.



Gambar 5.6 Skematik Konfigurasi Pin

5.1.3 Perancangan Perangkat Lunak

Perancangan perangkat lunak pada perangkat sistem mencakup perangkat lunak untuk menulis dan *upload source code program* pada mikrokontroler. Bahasa pemrograman yang digunakan adalah bahasa C Arduino. Pembuatan dan *upload program* dilakukan pada *software* Arduino IDE. Untuk membantu pengaksesan modul transceiver nRF24L01 dibantu dengan library Arduino RF24, nRF24L01 dan SPI.

5.1.4 Perancangan Paket Data

Paket data merupakan data yang dikirim oleh tiap perangkat sensor node untuk saling berkomunikasi. Pada penelitian ini paket yang dikirimkan sebesar 8 byte dan dikirimkan secara broadcast.



Gambar 5.7 Ilustrasi paket data

Sesuai ilustrasi pada Gambar 5.5, data dibagi menjadi 4 bagian dan setiap bagian memiliki data yang berbeda pada setiap paket. Sementara untuk paket data dibedakan menjadi tiga macam.

5.1.4.1 Paket Data Broadcast Node Beacon

Paket ini merupakan paket yang dikirimkan dari node *beacon* kepada setiap node *receiver* yang ada. Paket ini digunakan untuk mengirimkan paket referensi yang berfungsi sebagai tanda dimulainya sinkronisasi waktu. Isi pada paket ini dapat dilihat pada Tabel 5.5.

Tabel 5.5 Paket Data Broadcast Node Beacon

| No | Data | Ukuran |
|----|-------------|--------|
| I | Header / ID | 2 byte |

| | | |
|-----|------------------------|--------|
| II | Jumlah node | 2 byte |
| III | Jumlah Paket referensi | 2 byte |
| IV | Time Slot | 2 byte |

Data header menunjukkan identitas dari pengirim, data ini bertujuan agar node yang menerima paket data mampu mengidentifikasi node pengirim paket dan fungsi dari paket yang dikirimkan. Data jumlah node digunakan untuk menginformasikan berapa banyak node *receiver* yang harus bersinkronisasi sehingga setiap node mampu menyiapkan berapa data yang harus dikirim dan diterima sesuai dengan banyaknya node *receiver* yang harus saling bersinkronisasi. Data paket referensi berfungsi untuk menginformasikan berapa banyak data yang dikirimkan oleh node *beacon* secara beruntun sehingga node *receiver* dapat mengetahui berapa banyak paket data yang harus diterima dan diproses. Sementara data *time slot* merupakan informasi banyaknya waktu yang diberikan untuk mengirimkan data saat penjadwalan. Selain itu lama waktu satu kali putaran dari tiap node juga tergantung dari data *time slot* ini.

5.1.4.2 Paket Data Waktu Observasi Node Receiver

Paket waktu merupakan paket yang saling dikirim oleh *receiver* saat terjadi sinkronisasi waktu. Isi pada paket ini dapat dilihat pada Tabel 5.6.

Tabel 5.6 Paket Data Waktu Observasi Node Receiver

| No | Data | Ukuran |
|-----|-----------------------------|--------|
| I | Header / ID | 2 byte |
| II | Waktu Observasi Detik | 2 byte |
| III | Waktu Observasi Milisecond | 2 byte |
| IV | Waktu Observasi Microsecond | 2 byte |

Pada paket data ini data header juga menunjukkan identitas dari pengirim, dan bertujuan agar node yang menerima paket data mampu mengidentifikasi node pengirim paket dan fungsi dari paket yang dikirimkan. Sementara untuk paket bagian dua hingga empat merupakan data waktu observasi yang dibagi dalam detik, *milisecond* dan *microsecond*.

5.1.4.3 Paket Data Informasi Node Receiver

Paket data merupakan paket yang dikirim *receiver* secara broadcast setelah terjadi sinkronisasi waktu dan penjadwalan. Isi pada paket ini dapat dilihat pada Tabel 5.7.

Tabel 5.7 Paket Data Informasi Node Receiver

| No | Data | Ukuran |
|----|----------------|--------|
| I | Header / ID | 2 byte |
| II | Data Informasi | 2 byte |

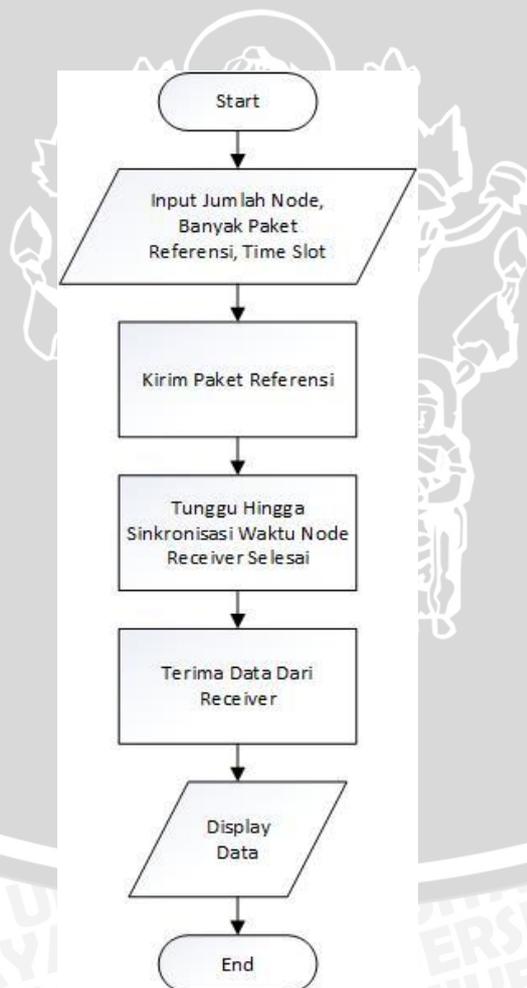
Pada paket data ini data header memiliki peran yang sama yaitu menunjukkan identitas dari pengirim, dan bertujuan agar node yang menerima paket data mampu mengidentifikasi node pengirim paket dan fungsi dari paket pada paket bagian dua yang merupakan data informasi merupakan data dari sensor yang digunakan. Namun pada penelitian ini data yang digunakan merupakan angka acak.

5.1.5 Perancangan Algoritma

Perancangan algoritma menjelaskan secara detail bagaimana sistem bekerja sesuai alur yang telah dijelaskan sebelumnya. Perancangan dibagi menjadi dua bagian yaitu pada node *beacon* dan node *receiver*.

5.1.5.1 Perancangan Algoritma Node *Beacon*

Node *beacon* berkerja sebagai pemberi paket referensi yang digunakan *receiver* untuk sinkronisasi waktu dan sebagai penerima paket dari *receiver* setelah terjadi penjadwalan. Berikut diagram alir node *beacon*:



Gambar 5.8 Diagram Alir Node *Beacon*

Seperti diagram alir pada Gambar 5.8. Node *beacon* bekerja dan mengirim paket sesuai keinginan dari user. Setelah itu node *beacon*

mengirimkannya sesuai jumlah yang diinputkan. Kemudian node *beacon* menunggu hingga sinkronisasi selesai. Diteruskan dengan berubah ke mode penerimaan untuk menerima semua paket dari node *receiver*.

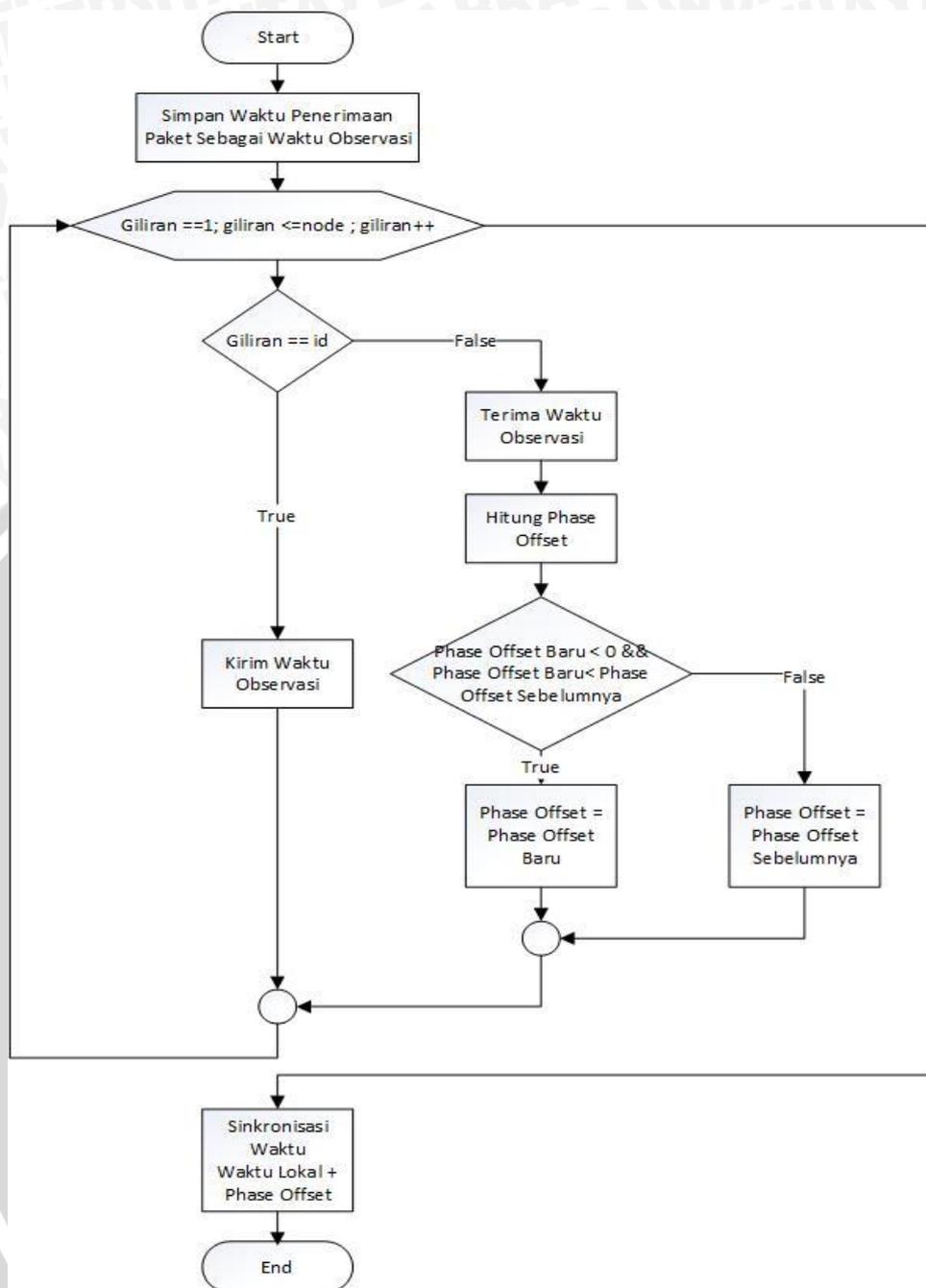
5.1.5.2 Perancangan Algoritma Node Receiver

Pada sinkronisasi menggunakan metode RBS, node *receiver* memiliki peran sebagai obyek sinkronisasi. Dimana *receiver* mensinkronisasikan waktunya dengan *receiver* lain. Sementara node *beacon* tidak disinkronisasikan. Berikut diagram alir algoritma sinkronisasi yang dilakukan tiap node *receiver*:



Gambar 5.9 Diagram Alir Node Receiver

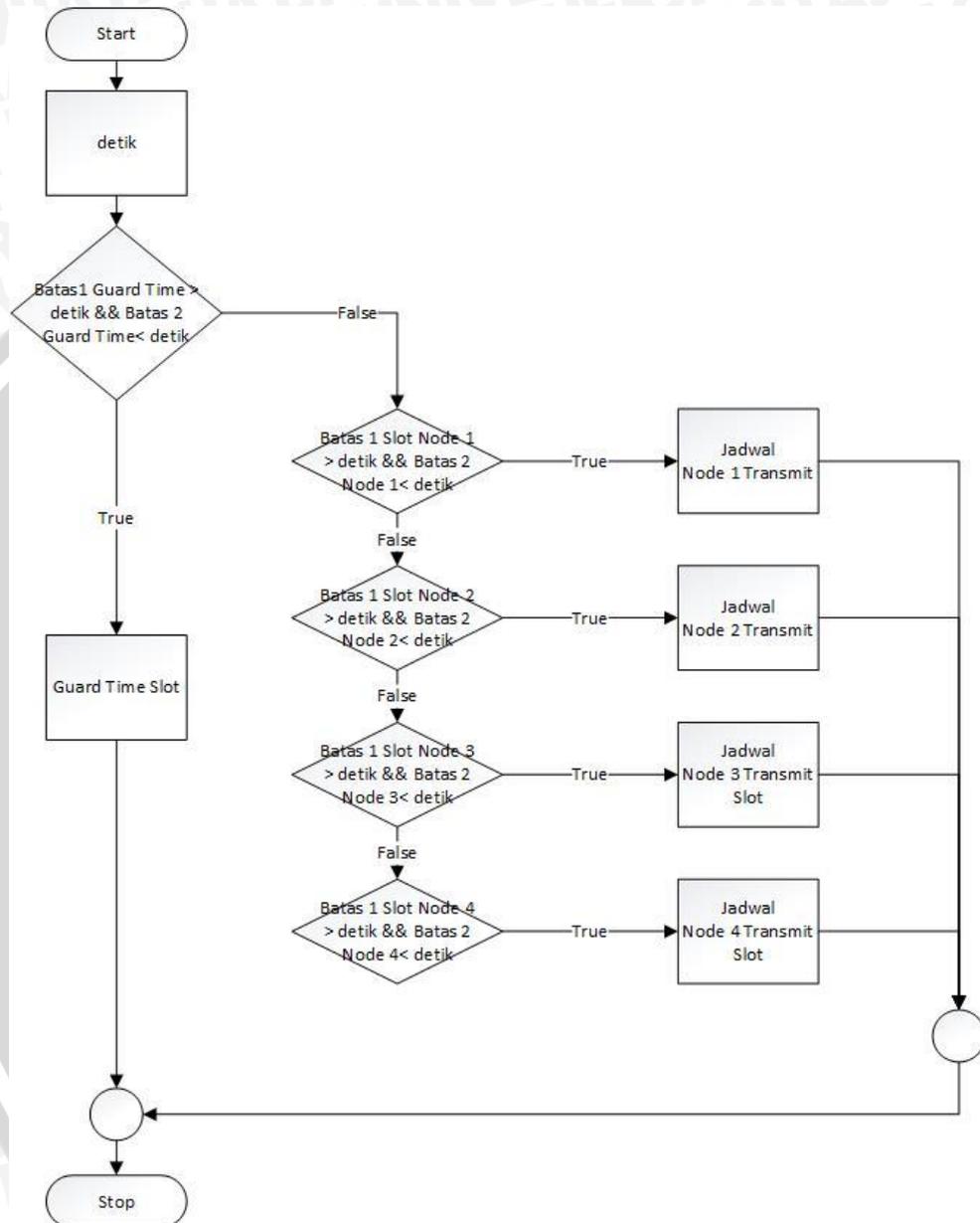
Seperti pada diagram alir Gambar 5.9. Alur kerja node *receiver* dimulai setelah mendapatkan paket referensi dari node *beacon*. Node menyimpan waktu penerimaan paket sebagai waktu observasi yang digunakan pada sinkronisasi waktu RBS. Setelah sinkronisasi selesai terjadi penjadwalan menggunakan metode TDMA. Kemudian setiap node *receiver* melakukan pengiriman data sesuai jadwal yang telah ditetapkan. Untuk tahap sinkronisasi waktu RBS dijelaskan digram alir pada Gambar 5.10. Sementara diagram alir penjadwalan TDMA dapat dilihat lebih detail pada Gambar 5.11.



Gambar 5.10 Diagram Alir Sinkronisasi RBS

Penjelasan diagram alir Gambar 5.10 yaitu sinkronisasi bekerja setelah node *receiver* menerima paket referensi, node *receiver* menyimpan waktu penerimaan paket dan dijadikan sebagai waktu observasi. Kemudian dilanjutkan pengiriman paket waktu observasi antar node *receiver*. Node dengan id satu mendapatkan giliran mengirimkan paket terlebih dahulu sementara node *receiver* lain dalam mode penerimaan. Dalam mode penerimaan node menghitung phase offsetnya dengan mengurangi waktu observasi node lain dengan waktu observasinya sendiri, dan apabila didapatkan phase offset kurang dari 0 dan kurang dari phase offset sebelumnya maka yang dijadikan phase offset

adalah phase offset yang baru. Proses ini dilakukan hingga semua id selesai mengirimkan paket waktu observasinya. Kemudian setiap node melakukan sinkronisasi waktu dengan mengurangi waktu lokalnya dengan *phase offset*.



Gambar 5.11 Diagram Alir Penjadwalan TDMA

Diagram alir 5.11 menjelaskan tahap penjadwalan TDMA. Penjadwalan ini dapat dikatakan sebagai pencocokan waktu satuan detik dari waktu yang telah disinkronisasikan dengan batas batas yang telah ditetapkan dari paket referensi berupa time slot. Jika detik waktu node memenuhi kondisi batas *guard time* maka pada saat itu node menjadi mode guard (idle). Sementara pada saat detik memenuhi kondisi batas pengiriman node 1 maka saat itu diharuskan node 1 mengirimkan paketnya ke node *beacon* dan seterusnya.

5.1.6 Perancangan Konfigurasi Frekuensi dan Channel

Pada perancangan frekuensi dan channel yang digunakan pada penelitian ini di atur secara otomatis. Modul nRF24L01 dapat bekerja pada range channel antara 0-125. Pada penelitian ini terdapat setting channel yang menggunakan library RF24. Setiap satu channel memiliki frekuensi sebesar 1Mhz. Penggunaan channel juga mempengaruhi frekuensi yang digunakan oleh sensor node. Penggunaan frekuensi pada nRF24L01 dimulai dari 2.400GHz sampai dengan 2.525GHz. Pada data sheet nRF24L01 terdapat rumus untuk menentukan frekuensi yang digunakan yaitu.

$$F0 = 2400 + RF \text{ Channel [Mhz]} \quad (5.1)$$

Jika pada penelitian ini menggunakan channel 123 maka frekuensi yang digunakan adalah:

$$F0 = 2400 + 123 \quad (5.2)$$

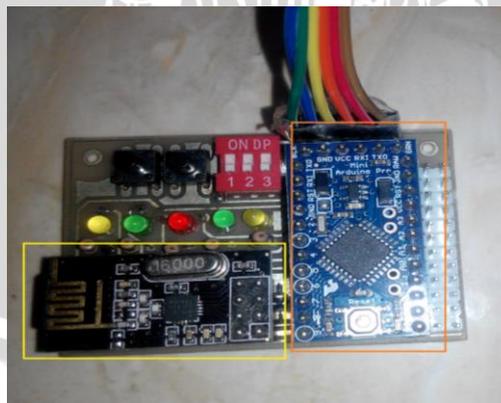
$$F0 = 2523 \text{ Mhz} = 2.523 \text{ Ghz} \quad (5.3)$$

5.2 Implementasi Sistem

Implementasi sistem merupakan penerapan dari perancangan yang telah dibuat pada subbab sebelumnya. Pembahasan dari implementasi adalah menjelaskan tentang perangkat keras dan perangkat lunak yang digunakan, penjelasan tentang paket data serta algoritma yang digunakan pada sistem ini.

5.2.1 Implementasi Secara Keseluruhan

Implementasi komponen utama node yang terdiri dari mikrokontroller Arduino pro mini dan modul tranceiver nRF24L01 serta personal komputer yang berguna untuk melihat jalan program dibantu dengan software Arduino berupa serial monitor. Implementasi perangkat node dapat dilihat pada Gambar 5.12.



Gambar 5.12 Komponen Utama Node

Pada Gambar 5.10 komponen yang diberikan tanda warna kuning adalah nRF24L01 dan komponen bertanda jingga adalah mikrokontroller Arduino Pro mini.

5.2.2 Spesifikasi Perangkat Keras

Berikut perangkat keras beserta spesifikasinya yang digunakan pada penelitian ini:

1. Mikrokontroler Arduino Pro Mini

Mikrokontroler berfungsi sebagai pengontrol sensor node yang didalamnya telah diberikan algoritma untuk melakukan suatu aksi. Spesifikasi dapat dilihat pada Tabel 5.8.

Tabel 5.8 Spesifikasi Mikrokontroler Arduino Pro Mini

| Spesifikasi | Keterangan |
|-------------------------------|---|
| Microcontroller | Atmega328 |
| Operating Voltage | 5V (<i>depending on model</i>) |
| Input Voltage | 5 - 12 V (5V model) |
| Digital I/O Pins | 14 (<i>of which 6 provide PWM Output</i>) |
| Analog Input Pins | 8 |
| DC Current per I/O Pin | 40 mA |
| Flash Memory | 16 KB (<i>of which 2 KB used by bootloader</i>) |
| SRAM | 1 KB |
| EEPROM | 512 bytes |
| Clock Speed | 16 MHz (5V model) |
| Microcontroller | Atmega328 |
| Operating Voltage | 5V (<i>depending on model</i>) |
| Input Voltage | 5 - 12 V (5V model) |

2. Modul nRF24L01

Merupakan modul tranceiver yang digunakan untuk berkomunikasi dengan perangkat lain secara nirkabel. Berikut spesifikasinya:

Tabel 5.9 Spesifikasi Modul nRF24L01

| Spesifikasi | Keterangan |
|--------------------------------|---|
| Gelombang | 2.4 Ghz ISM band |
| Communication Interface | SPI (<i>Serial Pheripheral Interface</i>) |
| Operating Voltage | 1.9 – 3.6V DC |

3. Personal Computer

Personal Computer pada sistem ini memiliki peran untuk memonitoring berjalannya program dan memantau hasil/data yang dikirimkan dan diterima oleh node. Spesifikasi Personal Computer yang digunakan dapat dilihat pada Tabel 5.10.

Tabel 5.10 Spesifikasi Personal Computer

| Spesifikasi | Keterangan |
|-------------------------|--|
| <i>Jenis Komputer</i> | ASUS |
| <i>Processor</i> | Inter Core i5-2430M CPU @2.50GHz (4 CPUs) ~ 2.5Ghz |
| <i>RAM</i> | 4096 MB RAM |
| <i>System Type</i> | 64-Bit <i>Operating system</i> |
| <i>Operating System</i> | Windows 10 Pro |

5.2.3 Spesifikasi Perangkat Lunak

1. Arduino IDE

Arduino IDE digunakan untuk menuliskan program, megupload dan memiliki aplikasi monitoring. Kelebihan lain adalah, Arduino memiliki banyak library yang dapat digunakan untuk mengakses nRF24L01. Berikut spesifikasi arduino IDE:

Tabel 5.11 Spesifikasi Arduino IDE

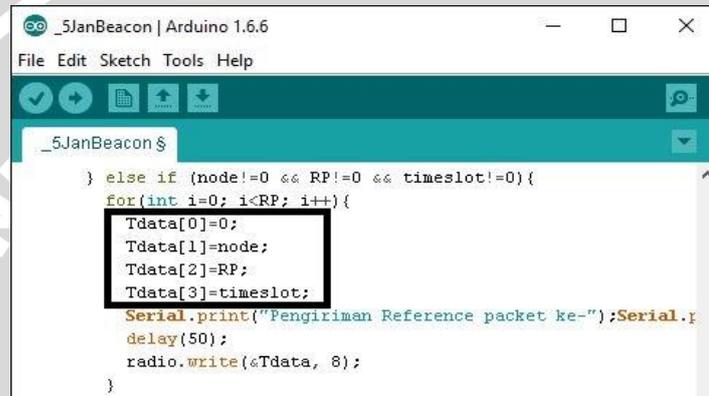
| Spesifikasi | Keterangan |
|-------------------------------|---|
| <i>Sistem Operasi</i> | Windows 7 64Bit |
| <i>Bahasa Pemrograman</i> | C (Arduino) |
| <i>Tools Pemrograman</i> | Arduino IDE versi 1.06 |
| <i>Library yang Digunakan</i> | RF24 Arduino Library nRF24L01 SPI |

5.2.4 Implementasi Paket Data

Perancangan paket data digunakan untuk membedakan paket data yang dikirimkan pada tiap-tiap proses pengiriman paket pada sistem ini. Berikut beberapa paket yang digunakan pada sistem ini. Implementasi dilakukan dengan cara menuliskan nilai pada source code.

5.2.4.1 Paket Data Broadcast Node *Beacon*

Paket dari node *beacon* terdiri dari header, banyaknya node yang disinkronisasi, banyaknya referance packet dan time slot yang digunakan. Setiap nilainya dimasukan pada array dengan format sesuai dengan perancangan. Untuk lebih jelasnya dapat dilihat pada Gambar 5.13.



```

} else if (node!=0 && RP!=0 && timeslot!=0){
  for(int i=0; i<RP; i++){
    Tdata[0]=0;
    Tdata[1]=node;
    Tdata[2]=RP;
    Tdata[3]=timeslot;
    Serial.print("Pengiriman Reference packet ke-");Serial.p
    delay(50);
    radio.write(&Tdata, 8);
  }
}

```

Gambar 5.13 Implementasi Paket Data Broadcast Node *Beacon*

5.2.4.2 Paket Data Waktu Observasi Node *Receiver*

Paket yang dikirimkan dari *receiver* ke *receiver* lain. Terdiri dari header, dan waktu observasi yang dibagi menjadi 3 nilai yaitu detik, *milisecond* dan *mikrosecond*. Setiap nilainya dimasukan pada array dengan format sesuai dengan perancangan. Untuk lebih jelasnya dapat dilihat pada Gambar 5.14.



```

} else if (state==1){
  if (id==idcount){
    setupkirim();
    delay(200);
    Tdata[0]=id;
    Tdata[1]=obsdetik;
    Tdata[2]=obsmilis;
    Tdata[3]=obsmikros;
    Serial.println("Mengirimkan Waktu Observasi!!!");
  }
}

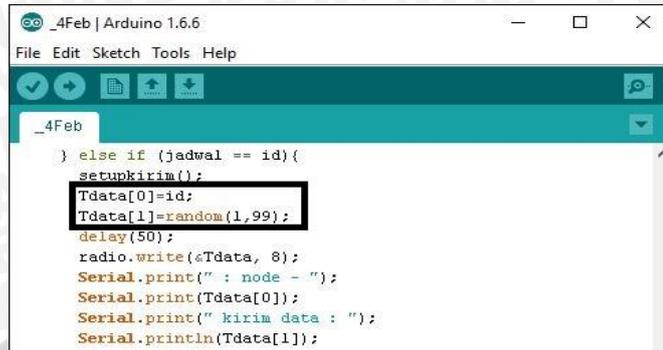
```

Gambar 5.14 Implementasi Paket Data Waktu Observasi Node *Receiver*

5.2.4.3 Paket Data Informasi Node *Receiver*

Paket yang dikirimkan dari *receiver* ke node *beacon*. Terdiri dari header, dan nilai informasi node yang dalam penelitian ini berupa nilai random. Setiap

nilainya dimasukan pada array dengan format sesuai dengan perancangan. Untuk lebih jelasnya dapat dilihat pada Gambar 5.15.



```

} else if (jadwal == id){
  setupkirim();
  Tdata[0]=id;
  Tdata[1]=random(1,99);
  delay(50);
  radio.write(Tdata, 8);
  Serial.print(" : node - ");
  Serial.print(Tdata[0]);
  Serial.print(" kirim data : ");
  Serial.println(Tdata[1]);
}

```

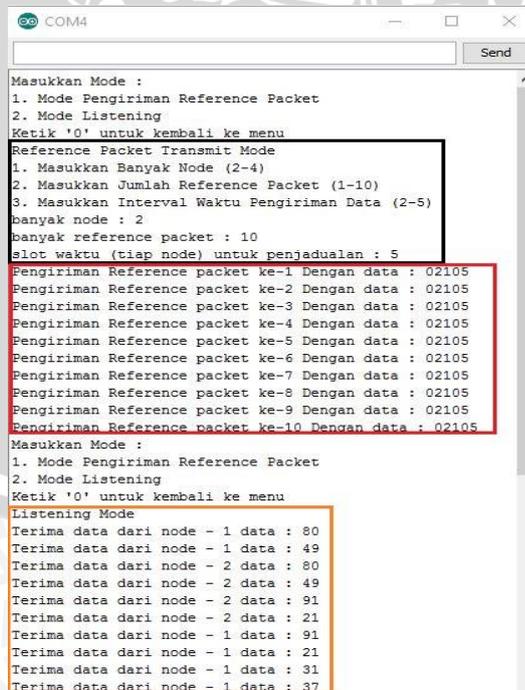
Gambar 5.15 Implementasi Paket Data Informasi Node Receiver

5.2.5 Implementasi Algoritma

Implementasi algoritma merupakan penjelasan dari hasil perancangan algoritma yang telah dibuat. Berikut hasil implementasi dari perancangan algoritma:

5.2.5.1 Node Beacon

Hasil implementasi terdapat pada Gambar 5.16. Dimana bagian yang ditandai kotak warna hitam menandakan proses input yang dilakukan user, pada kotak warna merah menandakan proses pengiriman paket referensi semenara untuk bagian yang diberikan kotak warna jingga menandakan penerimaan paket dari receiver yang telah melakukan sinkronisasi dan memiliki jadwal pengiriman.



```

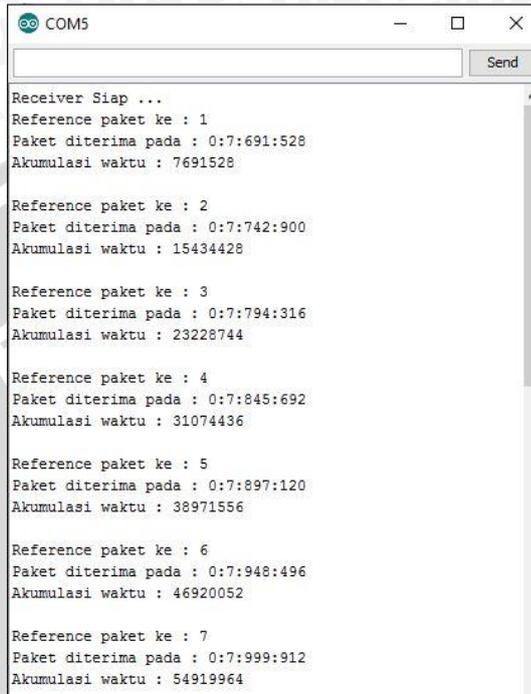
Masukkan Mode :
1. Mode Pengiriman Reference Packet
2. Mode Listening
Ketik '0' untuk kembali ke menu
Reference Packet Transmit Mode
1. Masukkan Banyak Node (2-4)
2. Masukkan Jumlah Reference Packet (1-10)
3. Masukkan Interval Waktu Pengiriman Data (2-5)
banyak node : 2
banyak reference packet : 10
slot waktu (tiap node) untuk penjadwalan : 5
Pengiriman Reference packet ke-1 Dengan data : 02105
Pengiriman Reference packet ke-2 Dengan data : 02105
Pengiriman Reference packet ke-3 Dengan data : 02105
Pengiriman Reference packet ke-4 Dengan data : 02105
Pengiriman Reference packet ke-5 Dengan data : 02105
Pengiriman Reference packet ke-6 Dengan data : 02105
Pengiriman Reference packet ke-7 Dengan data : 02105
Pengiriman Reference packet ke-8 Dengan data : 02105
Pengiriman Reference packet ke-9 Dengan data : 02105
Pengiriman Reference packet ke-10 Dengan data : 02105
Masukkan Mode :
1. Mode Pengiriman Reference Packet
2. Mode Listening
Ketik '0' untuk kembali ke menu
Listening Mode
Terima data dari node - 1 data : 80
Terima data dari node - 1 data : 49
Terima data dari node - 2 data : 80
Terima data dari node - 2 data : 49
Terima data dari node - 2 data : 91
Terima data dari node - 2 data : 21
Terima data dari node - 1 data : 91
Terima data dari node - 1 data : 21
Terima data dari node - 1 data : 31
Terima data dari node - 1 data : 37

```

Gambar 5.16 Implementasi Algoritma pada Node Beacon

5.2.5.2 Node Receiver

Implementasi pertama yang dilakukan pada node *receiver* adalah implementasi penerimaan paket referensi dari node *beacon*. Saat diterima paket node *receiver* menyimpan waktu kedatangan paket sebagai waktu observasi. Hasil implementasi terdapat pada Gambar 5.17.



```
COM5
Receiver Siap ...
Reference paket ke : 1
Paket diterima pada : 0:7:691:528
Akumulasi waktu : 7691528

Reference paket ke : 2
Paket diterima pada : 0:7:742:900
Akumulasi waktu : 15434428

Reference paket ke : 3
Paket diterima pada : 0:7:794:316
Akumulasi waktu : 23228744

Reference paket ke : 4
Paket diterima pada : 0:7:845:692
Akumulasi waktu : 31074436

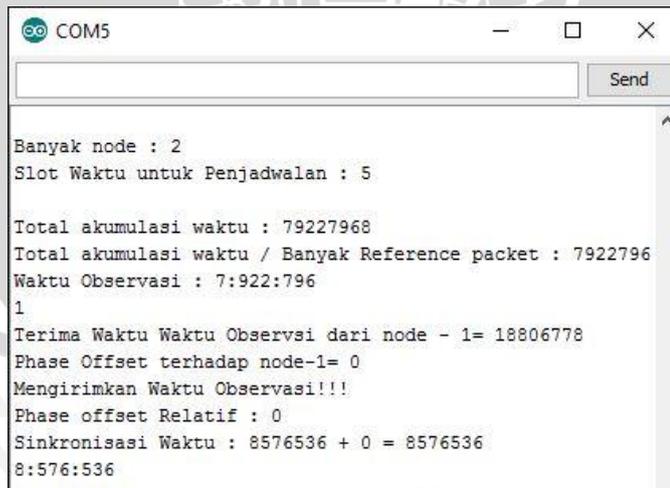
Reference paket ke : 5
Paket diterima pada : 0:7:897:120
Akumulasi waktu : 38971556

Reference paket ke : 6
Paket diterima pada : 0:7:948:496
Akumulasi waktu : 46920052

Reference paket ke : 7
Paket diterima pada : 0:7:999:912
Akumulasi waktu : 54919964
```

Gambar 5.17 Implementasi Penerimaan Paket referensi pada Node Receiver

Setelah itu implementasi metode RBS. Pada Gambar 5.18 paket waktu observasi yang telah diterima saling ditukarkan dan menghitung *phase offset* antara *receiver*. Kemudian yang terakhir adalah tahap sinkronisasi waktu.

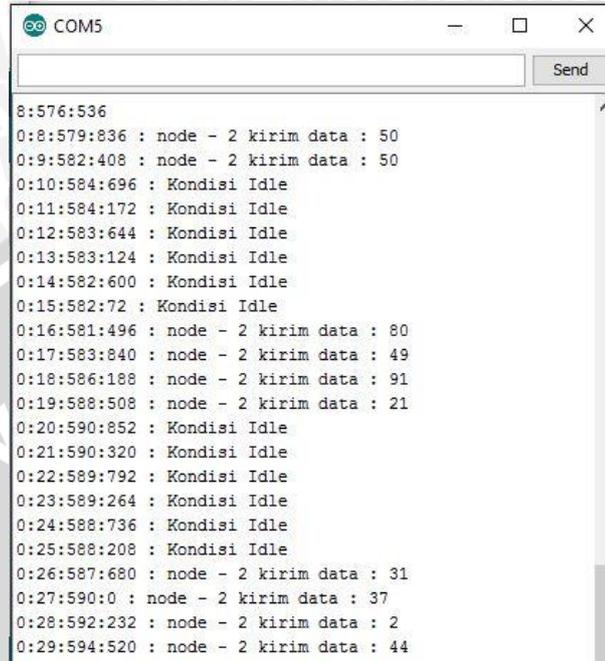


```
COM5
Banyak node : 2
Slot Waktu untuk Penjadwalan : 5

Total akumulasi waktu : 79227968
Total akumulasi waktu / Banyak Reference packet : 7922796
Waktu Observasi : 7:922:796
1
Terima Waktu Waktu Observsi dari node - 1= 18806778
Phase Offset terhadap node-1= 0
Mengirimkan Waktu Observasi!!!
Phase offset Relatif : 0
Sinkronisasi Waktu : 8576536 + 0 = 8576536
8:576:536
```

Gambar 5.18 Implementasi Metode RBS pada Node Receiver

Yang terakhir adalah implementasi penjadwalan menggunakan TDMA. Seperti yang dijelaskan sebelumnya, node *receiver* hanya mengirimkan data pada saat mendapat giliran sementara saat bukan giliran node *receiver* beradapa pada kondisi *guard/idle*. Berikut hasil implementasi penggunaan metode TDMA.

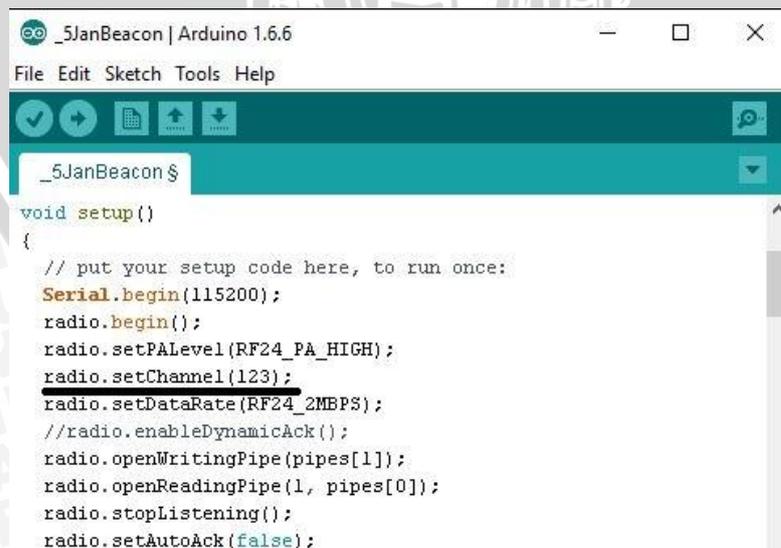


```
COM5
8:576:536
0:8:579:836 : node - 2 kirim data : 50
0:9:582:408 : node - 2 kirim data : 50
0:10:584:696 : Kondisi Idle
0:11:584:172 : Kondisi Idle
0:12:583:644 : Kondisi Idle
0:13:583:124 : Kondisi Idle
0:14:582:600 : Kondisi Idle
0:15:582:72 : Kondisi Idle
0:16:581:496 : node - 2 kirim data : 80
0:17:583:840 : node - 2 kirim data : 49
0:18:586:188 : node - 2 kirim data : 91
0:19:588:508 : node - 2 kirim data : 21
0:20:590:852 : Kondisi Idle
0:21:590:320 : Kondisi Idle
0:22:589:792 : Kondisi Idle
0:23:589:264 : Kondisi Idle
0:24:588:736 : Kondisi Idle
0:25:588:208 : Kondisi Idle
0:26:587:680 : node - 2 kirim data : 31
0:27:590:0 : node - 2 kirim data : 37
0:28:592:232 : node - 2 kirim data : 2
0:29:594:520 : node - 2 kirim data : 44
```

Gambar 5.19 Implementasi TDMA pada Node Receiver

5.2.6 Implementasi Konfigurasi Frekuensi dan Channel

Sementara pada konfigurasi frekuensi dan channel pada node. Penulis memudahkan tanpa perlu mengkonfigurasi frekuensi. Hanya dengan menuliskan *channel* yang digunakan. Lihat pada Gambar 5.20, bagian yang terdapat garis bawah merupakan source code untuk mengkonfigurasi channel pada perangkat node.



```
_5JanBeacon | Arduino 1.6.6
File Edit Sketch Tools Help
void setup()
{
  // put your setup code here, to run once:
  Serial.begin(115200);
  radio.begin();
  radio.setPALevel(RF24_PA_HIGH);
  radio.setChannel(123);
  radio.setDataRate(RF24_2MBPS);
  //radio.enableDynamicAck();
  radio.openWritingPipe(pipes[1]);
  radio.openReadingPipe(1, pipes[0]);
  radio.stopListening();
  radio.setAutoAck(false);
}
```

Gambar 5.20 Implementasi Konfigurasi Channel

BAB 6 PENGUJIAN

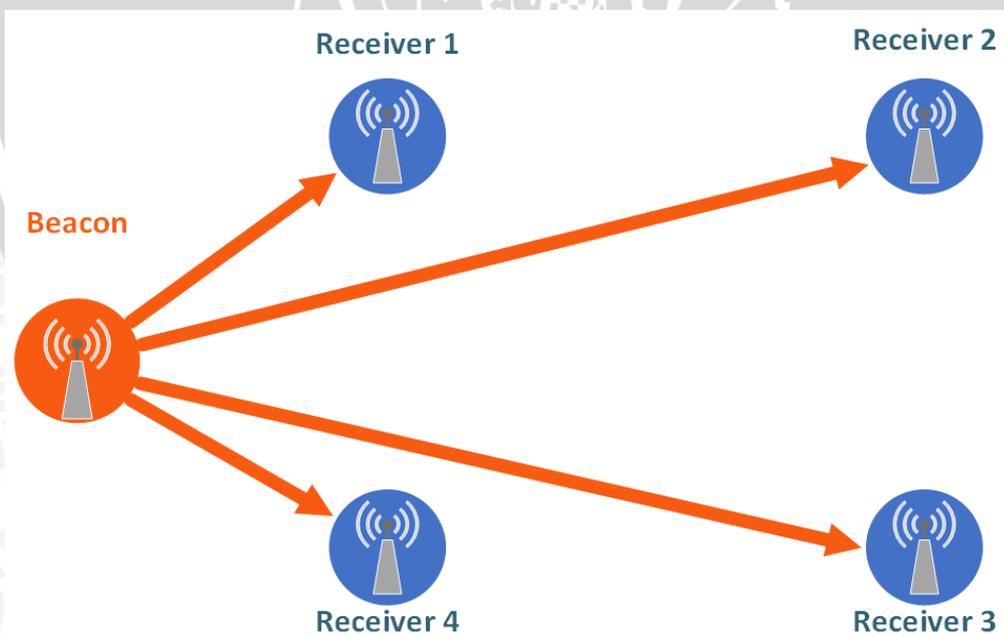
Pada bab ini membahas mengenai pengujian dan pembahasan hasil dari implementasi sistem. Pengujian di fokuskan pada kualitas dari metode *Reference Broadcast Synchronizaton* dan Time Divison Multiple Access yang diterapkan pada Arduino pro mini dan modul nRF24L01. Proses pembahasan pengujian dan analisis bertujuan untuk menarik kesimpulan dari hasil pengujian hasil implementasi metode *Broadcast Synchronization* dan Time Divison Multiple Access pada Arduino pro mini dan modul nRF24L01. Proses pembahasan mengacu pada dasar teori yang sesuai dengan hasil pengujian yang didapatkan. Pembahasan dan analisis dilakukan terhadap hasil pengujian yang akan dilakukan.

6.1 Pengujian Fungsional

Pengujian fungsional sistem bertujuan untuk mengetahui tingkat keberhasilan pengiriman dan penerimaan data dari tiap node. Pada pengujian fungsional ini terdapat satu node sebagai node *Beacon* dan terdapat 4 node sebagai node *receiver*.

6.1.1 Pengiriman Paket Referensi

Pengujian ini bertujuan untuk melihat kesiapan node *beacon* dalam mengirimkan beberapa paket referensi secara berkala dan juga untuk melihat kesiapan node *receiver* dalam menerima paket paket referensi. Berikut gambaran skenario pengujiannya:



Gambar 6.1 Skenario Pengujian Pengiriman Paket Referensi

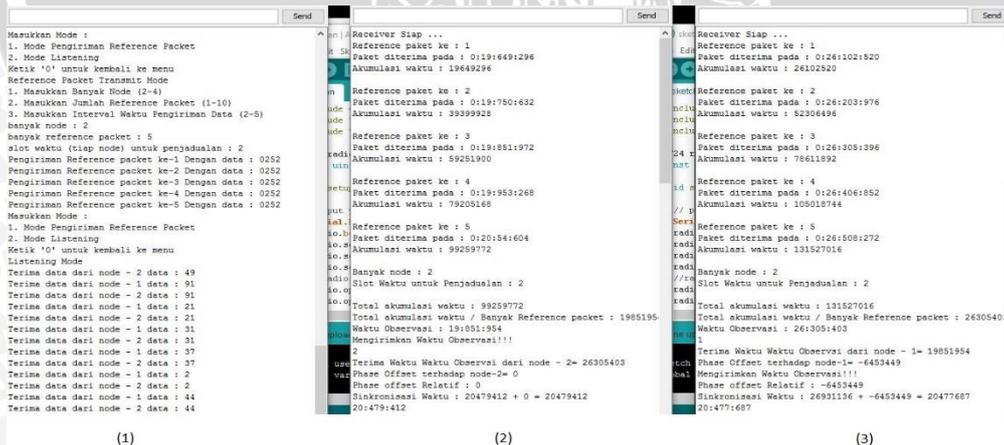
Pengujian ini memiliki beberapa prosedur yang harus diikuti. Berikut prosedur pengujian pengiriman paket referensi.

Tabel 6.1 Prosedur pengujian Pengiriman Paket referensi

| | |
|-------------------------|--|
| Kasus Pengujian | Pengiriman Paket referensi Oleh Node <i>Beacon</i> kepada Node <i>Receiver</i> |
| Objek Pengujian | Kesesuaian jumlah paket referensi yang diterima oleh node <i>receiver</i> dengan paket referensi yang dikirim oleh node <i>beacon</i> . |
| Tujuan Pengujian | Mendapatkan data penerimaan paket referensi node <i>receiver</i> dengan jumlah paket referensi yang dikirimkan oleh node <i>beacon</i> |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Penguji mengunggah program kepada setiap node. 2. Penguji memasang node dengan seperti Gambar 6.1 3. Penguji memberikan input banyak paket referensi yang harus dikirimkan oleh node <i>beacon</i>. 4. Penguji melihat serial monitor dan menganalisa banyak paket referensi yang dikirimkan node <i>beacon</i> dan paket referensi yang diterima node <i>receiver</i>. 5. Penguji mencatat hasil pengujian yang dilakukan. 6. Pengujian diulang selama 10 kali dengan menggunakan 1-10 paket referensi. |

6.1.1.1 Hasil Pengujian

Hasil dari pengujian ini dapat dilihat pada Gambar 6.2. Gambar ini merupakan sample hasil pengujian pengiriman paket kepada 2 node *receiver* dengan 5 paket referensi. Dimana nomor 1 merupakan hasil monitoring node *beacon* yang mengirimkan paket referensi dan nomor 2 dan 3 merupakan node



Gambar 6.2 Hasil Pengujian Pengiriman Paket referensi

receiver yang menerima paket referensi.

Hasil dari keseluruhan pengujian dapat dilihat pada Tabel 6.2. Pada tabel ini dituliskan 10 kali pengujian dengan banyak paket referensi sesuai dengan yang dituliskan pada prosedur pengujian. Data dituliskan pada tiap kolom node

dengan format data yang diterima / data yang dikirim (contoh: data 5/5 berarti dari 5 paket diterima dari 5 paket yang dikirimkan).

Tabel 6.2 Percobaan Fungsional Pengiriman Paket referensi

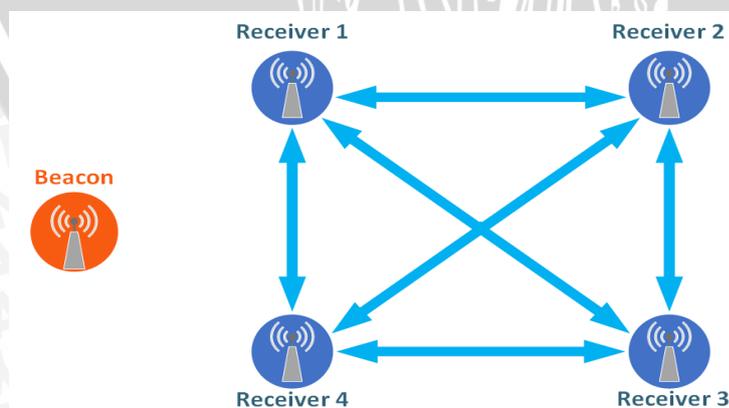
| Pengujian | Banyaknya Reference | Node ke | | | |
|-----------|---------------------|---------|-------|-------|-------|
| | | 1 | 2 | 3 | 4 |
| Uji 1 | 1 | 1/1 | 1/1 | 1/1 | 1/1 |
| Uji 2 | 2 | 2/2 | 2/2 | 2/2 | 2/2 |
| Uji 3 | 3 | 3/3 | 3/3 | 3/3 | 3/3 |
| Uji 4 | 4 | 4/4 | 4/4 | 4/4 | 4/4 |
| Uji 5 | 5 | 5/5 | 5/5 | 5/5 | 5/5 |
| Uji 6 | 6 | 6/6 | 6/6 | 6/6 | 6/6 |
| Uji 7 | 7 | 7/7 | 7/7 | 7/7 | 7/7 |
| Uji 8 | 8 | 8/8 | 8/8 | 8/8 | 8/8 |
| Uji 9 | 9 | 9/9 | 9/9 | 9/9 | 8/9 |
| Uji 10 | 10 | 10/10 | 10/10 | 10/10 | 10/10 |

6.1.1.2 Analisa Pengujian

Setelah melihat hasil pada Tabel 6.2. Penulis menganalisa node *beacon* telah mampu mengirimkan paket referensi sesuai dengan yang diinputkan oleh penulis. Sementara pada node *receiver* juga mampu menerima paket referensi yang telah dikirimkan meskipun pada pengujian ke 9 node 4 paket referensi tidak dapat diterima secara utuh. Hal ini mengindikasikan adanya pengaruh dari banyaknya paket referensi yang dikirim secara berkala terhadap kemampuan penerimaan data dari node *receiver* meskipun pengaruh itu sangat kecil.

6.1.2 Pengiriman Waktu Observasi

Pengujian ini bertujuan untuk melihat kesiapan antar tiap node *receiver* dalam mengirim dan menerima paket waktu observasi secara bergantian. Dimulai dari pengiriman paket node pertama hingga node terakhir. Saat satu node mengirim data maka node lain harus dalam kondisi menerima paket.



Berikut gambaran skenario pengujian waktu observasi :

Gambar 6.3 Skenario Pengujian Watu Observasi

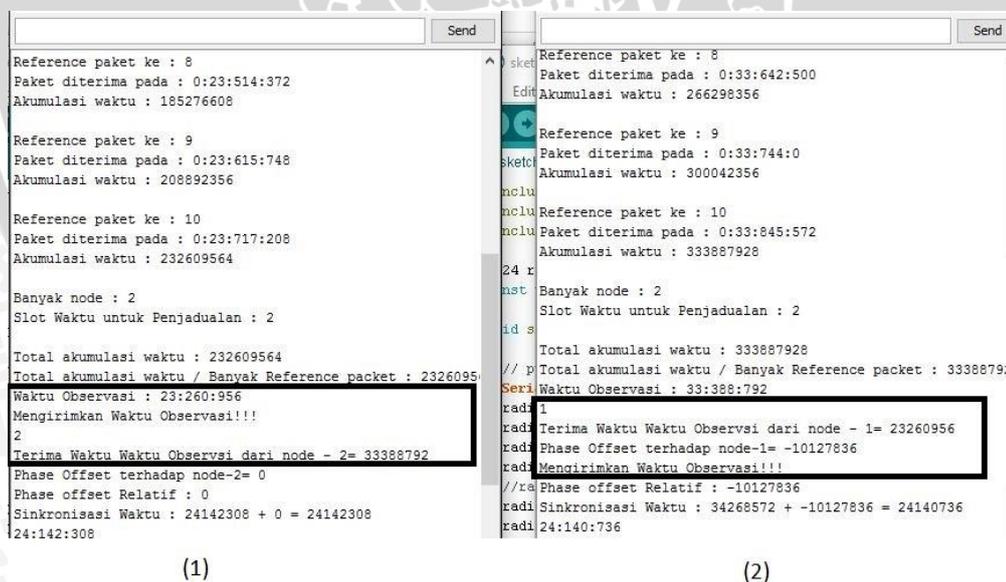
Pengujian ini memiliki beberapa prosedur yang harus diikuti. Prosedurnya dapat dilihat pada tabel 6.3:

Tabel 6.3 Pengiriman Waktu Observasi

| | |
|-------------------------|---|
| Kasus Pengujian | Pengiriman Paket Waktu Observasi Oleh Node <i>Receiver</i> kepada Node <i>Receiver</i> lain |
| Objek Pengujian | Kemampuan pengiriman dan penerimaan paket waktu observasi dari tiap node <i>receiver</i> |
| Tujuan Pengujian | Mendapatkan data hasil pengiriman dan penerimaan paket waktu observasi dari tiap tiap node |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Penguji mengunggah program kepada setiap node 2. Penguji memasang node dengan seperti Gambar 6.3 3. Penguji melihat serial monitor dan menganalisa keberhasilan pengiriman dan penerimaan paket dari tiap node <i>receiver</i>. 4. Penguji mencatat hasil pengujian yang dilakukan. 5. Pengujian diulang selama 10 kali. |

6.1.2.1 Hasil Pengujian

Hasil dari pengujian ini dapat dilihat pada Gambar 6.4 dimana gambar yang bertanda nomor satu adalah receiver 1 dan bertanda nomor 2 adalah receiver 2. Sementara hasil dari pengiriman waktu observasi terdapat pada gambar



Gambar 6.4 Pengiriman Waktu Observasi

yang diberikan tanda hitam.

Untuk hasil dari keseluruhan pengujian dapat dilihat pada Tabel 6.4. Pada tabel ini dituliskan 10 kali pengujian dengan banyak paket referensi sesuai

dengan yang dituliskan pada prosedur pengujian. Pada tabel terdapat nomer node pengirim dan penerima. Paket dikirimkan secara *broadcast* sehingga diharapkan paket diterima secara bersamaan oleh tiap node *receiver* penerima. Setiap keberhasilan pengiriman paket diberikan tanda 'v' sementara kegagalan pengiriman diberikan tanda 'x'.

Tabel 6.4 Percobaan Fungsional Pengiriman Paket Waktu Observasi

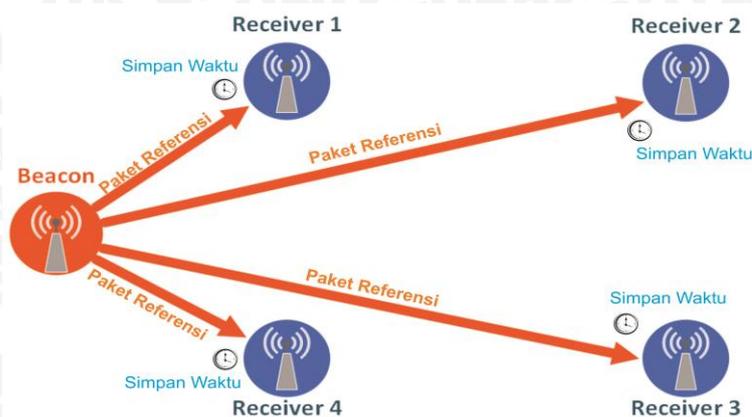
| Pengujian | Pengirim | 1 | | | 2 | | | 3 | | | 4 | | |
|-----------|----------|---|---|---|---|---|---|---|---|---|---|---|---|
| | Penerima | 2 | 3 | 4 | 1 | 3 | 4 | 1 | 2 | 4 | 1 | 2 | 3 |
| Uji 1 | | v | v | v | v | v | v | v | v | v | v | v | v |
| Uji 2 | | v | v | v | v | v | v | v | v | v | v | v | v |
| Uji 3 | | v | v | v | v | v | v | v | v | v | v | v | v |
| Uji 4 | | v | v | v | v | v | v | v | v | v | v | v | v |
| Uji 5 | | v | v | v | v | v | v | v | v | v | v | v | v |
| Uji 6 | | v | v | v | v | v | v | v | v | v | v | v | v |
| Uji 7 | | v | v | v | x | x | x | x | x | x | x | x | x |
| Uji 8 | | v | v | v | v | v | v | v | v | v | v | v | v |
| Uji 9 | | v | v | v | v | v | v | v | v | v | v | v | v |
| Uji 10 | | v | v | v | v | v | v | v | v | v | v | v | v |

6.1.2.2 Analisa Pengujian

Setelah melihat hasil pada Tabel 6.4. Penulis menganalisa tiap node *receiver* telah mampu untuk saling mengirim dan menerima paket waktu observasi. Hanya saja pada pengujian ke 7 terdapat kegagalan pengiriman data dikarenakan terjadinya packet loss yang mengakibatkan node ke 2 tidak dapat menerima paket. Karena node 2 tidak dapat menerima paket node 2 tidak mengetahui waktu gilirannya untuk mengirimkan paket. Sehingga hal ini berimbas pada node 3 dan 4 yang tidak dapat mengirimkan paket. Tetapi secara keseluruhan tingkat keberhasilan pengiriman dan penerimaan paket observasi waktu dapat dikatakan baik sehingga diharapkan tidak mengganggu kinerja sistem secara keseluruhan.

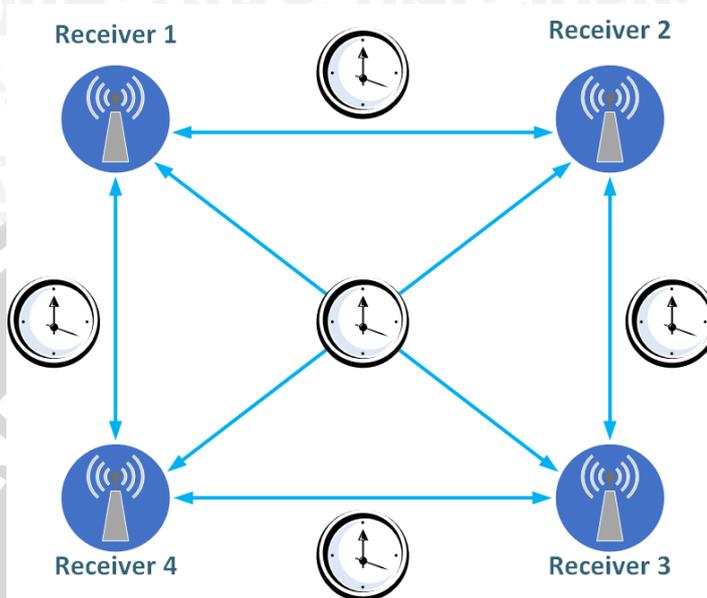
6.2 Pengujian Sinkronisasi Waktu

Pada pengujian sinkronisasi terdapat dua tahap yang harus dilakukan. Pertama adalah node beacon mengirimkan paket referensi sementara receiver harus mampu menerima paket tersebut dan menyimpan waktu kedatangan paket sebagai waktu observasi. Lihat pada Gambar 6.5.



Gambar 6.5 Skenario Tahap Satu Pengujian Sinkronisasi Waktu

Tahap kedua yaitu pengiriman waktu observasi antar node. Dari waktu observasi tersebut didapatkan phase offset. Phase offset tersebut yang digunakan untuk menyetarakan waktu antara tiap node. Ilustrasi skenario pengiriman waktu observasi dapat dilihat pada Gambar 6.6.



Gambar 6.6 Skenario Tahap Dua Pengujian Sinkronisasi Waktu

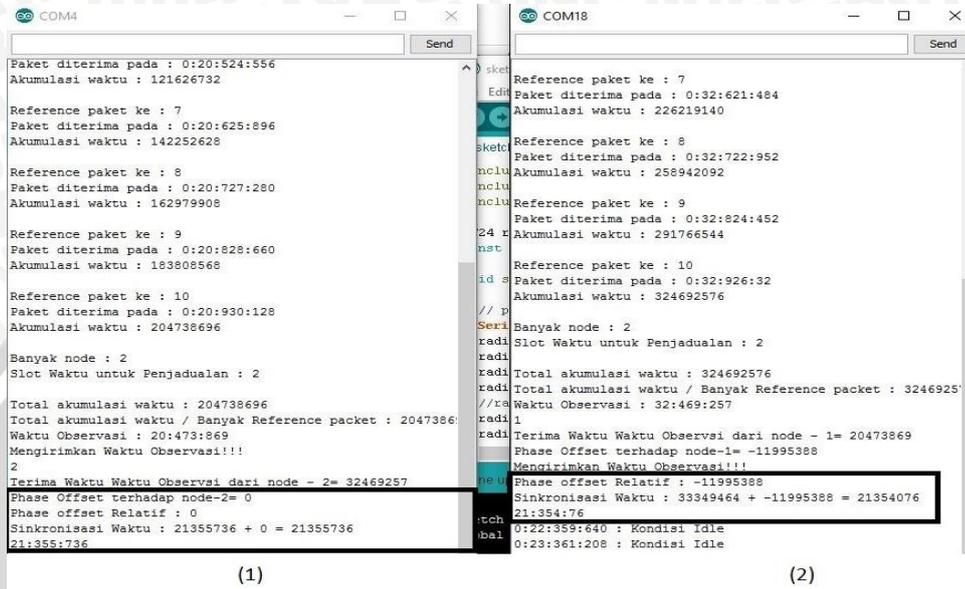
Sementara prosedur yang digunakan pada pengujian ini antara lain :

Tabel 6.5 Prosedur Percobaan Sinkronisasi Waktu

| | |
|-------------------------|---|
| Kasus Pengujian | Sinkronisasi antar node |
| Objek Pengujian | Tingkat <i>error</i> waktu dari sinkronisasi waktu menggunakan metode <i>reference broadcast synchronization</i> . |
| Tujuan Pengujian | Melihat pengaruh dari banyaknya node dan banyaknya paket referensi yang digunakan dalam sinkronisasi waktu |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Penguji mengunggah program kepada setiap node 2. Penguji menginputkan banyak node yang akan disinkronisasi dan banyak paket referensi yang dibutuhkan pada node <i>beacon</i>. 3. Penguji melihat serial monitor dan menganalisa <i>error</i> waktu dari sinkronisasi waktu. 4. Penguji mencatat hasil pengujian. 5. Pengujian dilakukan dengan menggunakan node sebanyak 2, 3 dan 4. Dalam tiap pengujian node tersebut juga dilakukan pengujian dengan mengirimkan paket referensi sebanyak 1, 5 dan 10. |

6.2.1 Hasil Pengujian

Dari hasil pengujian ini didapatkan waktu sinkronisasi dari node receiver. Dari waktu tersebut diambil waktu terendah dan waktu tertinggi yang digunakan untuk mengetahui nilai *error* waktu sinkronisasi waktu. Proses pengujian dapat dilihat pada Gambar 6.7.



Gambar 6.7 Hasil Pengujian Sinkronisasi Waktu

Sementara untuk hasil secara keseluruhan di bagi menjadi 3 pengujian itu sinkronisasi waktu 2, 3 dan 4 node. Dalam pengujian tersebut terdapat pengujian pengiriman sinkronisasi waktu dengan jumlah 1, 5 dan 10 yang semua hasilnya dituliskan dalam tabel. Sementara untuk *error* waktunya dituliskan dengan satuan *microsecond*.

6.2.1.1 Hasil Pengujian Sinkronisasi Waktu 2 Node

Berikut hasil pengujian sinkronisasi waktu 2 node yang dirangkum pada Tabel 6.6:

Tabel 6.6 Hasil Pengujian Sinkronisasi Waktu 2 Node

| Pengujian | Paket referensi | | | |
|---------------------|-----------------|--------|--------|------|
| | 1 | 5 | 10 | |
| Error (microsecond) | Uji 1 | 2372 | 1900 | 2243 |
| | Uji 2 | 2044 | 2061 | 1660 |
| | Uji 3 | 2864 | 1606 | 1561 |
| | Uji 4 | 2252 | 2290 | 1598 |
| | Uji 5 | 2672 | 1725 | 1572 |
| | Uji 6 | 2532 | 2227 | 1668 |
| | Uji 7 | 2336 | 2247 | 1673 |
| | Uji 8 | 2848 | 1863 | 2219 |
| | Uji 9 | 2144 | 1853 | 1568 |
| | Uji 10 | 2340 | 1822 | 1830 |
| Rata-rata | 2440.4 | 1959.4 | 1759.2 | |

6.2.1.2 Hasil Pengujian Sinkronisasi Waktu 3 Node

Berikut hasil pengujian sinkronisasi waktu 3 node yang dirangkum pada Tabel 6.7:

Tabel 6.7 Hasil Pengujian Sinkronisasi Waktu 3 Node

| Pengujian | | Paket referensi | | |
|---------------------|--------|-----------------|--------|--------|
| | | 1 | 5 | 10 |
| Error (microsecond) | Uji 1 | 2420 | 2287 | 2134 |
| | Uji 2 | 2840 | 2230 | 1996 |
| | Uji 3 | 2394 | 2044 | 2003 |
| | Uji 4 | 2816 | 2211 | 1947 |
| | Uji 5 | 2352 | 2192 | 1947 |
| | Uji 6 | 2466 | 2129 | 2018 |
| | Uji 7 | 2260 | 2182 | 1785 |
| | Uji 8 | 2568 | 1854 | 2254 |
| | Uji 9 | 2452 | 1886 | 1936 |
| | Uji 10 | 2435 | 2231 | 1975 |
| Rata-rata | | 2500.3 | 2124.6 | 1999.5 |

6.2.1.3 Hasil Pengujian Sinkronisasi Waktu 4 Node

Berikut hasil pengujian sinkronisasi waktu 4 node. Pengujian dilakukan 10 kali dengan banyak paket referensi yang berbeda. Hasilnya didapatkan selisih waktu dari sinkronisasi menggunakan metode RBS. Pada akhir tabel ditambahkan rata-rata dari 10 hasil pengujian tersebut. Data keseluruhan dapat dilihat pada Tabel 6.8.

Tabel 6.8 Hasil Pengujian Sinkronisasi Waktu 4 Node

| Pengujian | | Paket referensi | | |
|---------------------|--------|-----------------|--------|------|
| | | 1 | 5 | 10 |
| Error (microsecond) | Uji 1 | 3204 | 4014 | 2054 |
| | Uji 2 | 3212 | 3258 | 2122 |
| | Uji 3 | 2536 | 1847 | 2286 |
| | Uji 4 | 2876 | 2309 | 1742 |
| | Uji 5 | 2876 | 3310 | 2064 |
| | Uji 6 | 2872 | 3101 | 1982 |
| | Uji 7 | 2688 | 2744 | 1954 |
| | Uji 8 | 3532 | 2754 | 2096 |
| | Uji 9 | 3256 | 2752 | 2206 |
| | Uji 10 | 2882 | 2752 | 1684 |
| Rata-rata | | 2993.4 | 2884.1 | 2019 |

6.2.2 Pengujian Matematis

Pengujian matematis digunakan untuk memastikan kesesuaian sinkronisasi waktu menggunakan metode RBS dengan yang telah digunakan dalam pengujian. Pada pengujian ini menggunakan sample dari Tabel 6.6 pada pengujian ke 5. Berikut rincian proses sinkronisasi waktu yang dapat dilihat pada Gambar 6.4.

```
Receiver Siap ...
Reference paket ke : 1
Paket diterima pada : 0:22:804:792
Akumulasi waktu : 22804792

Reference paket ke : 2
Paket diterima pada : 0:22:906:168
Akumulasi waktu : 45710960

Reference paket ke : 3
Paket diterima pada : 0:23:7:508
Akumulasi waktu : 68718468

Reference paket ke : 4
Paket diterima pada : 0:23:108:908
Akumulasi waktu : 91827376

Reference paket ke : 5
Paket diterima pada : 0:23:210:244
Akumulasi waktu : 115037620

Reference paket ke : 6
Paket diterima pada : 0:23:311:620
Akumulasi waktu : 138349240

Reference paket ke : 7
Paket diterima pada : 0:23:412:996
Akumulasi waktu : 161762236

Reference paket ke : 8
Paket diterima pada : 0:23:514:372
Akumulasi waktu : 185276608

Reference paket ke : 9
Paket diterima pada : 0:23:615:748
Akumulasi waktu : 208892356

Reference paket ke : 10
Paket diterima pada : 0:23:717:208
Akumulasi waktu : 232609564

Banyak node : 2
Slot Waktu untuk Penjadualan : 2

Total akumulasi waktu : 232609564
Total akumulasi waktu / Banyak Reference packet : 23260956
Waktu Observasi : 23:260:956
Mengirimkan Waktu Observasi!!!
2
Terima Waktu Waktu Observsi dari node - 2= 33388792
Phase Offset terhadap node-2= 0
Phase offset Relatif : 0
Sinkronisasi Waktu : 24142308 + 0 = 24142308
24:142:308
```

```
Receiver Siap ...
Reference paket ke : 1
Paket diterima pada : 0:32:932:108
Akumulasi waktu : 32932108

Reference paket ke : 2
Paket diterima pada : 0:33:33:596
Akumulasi waktu : 65965704

Reference paket ke : 3
Paket diterima pada : 0:33:135:48
Akumulasi waktu : 99100752

Reference paket ke : 4
Paket diterima pada : 0:33:236:536
Akumulasi waktu : 132337288

Reference paket ke : 5
Paket diterima pada : 0:33:338:40
Akumulasi waktu : 165675328

Reference paket ke : 6
Paket diterima pada : 0:33:439:532
Akumulasi waktu : 199114860

Reference paket ke : 7
Paket diterima pada : 0:33:540:996
Akumulasi waktu : 232655856

Reference paket ke : 8
Paket diterima pada : 0:33:642:500
Akumulasi waktu : 266298356

Reference paket ke : 9
Paket diterima pada : 0:33:744:0
Akumulasi waktu : 300042356

Reference paket ke : 10
Paket diterima pada : 0:33:845:572
Akumulasi waktu : 333887928

Banyak node : 2
Slot Waktu untuk Penjadualan : 2

Total akumulasi waktu : 333887928
// p Total akumulasi waktu / Banyak Reference packet : 333887928
Seri Waktu Observasi : 33:388:792
rad 1
rad Terima Waktu Waktu Observsi dari node - 1= 23260956
rad Phase Offset terhadap node-1= -10127836
rad Mengirimkan Waktu Observasi!!!
//ra Phase offset Relatif : -10127836
radi Sinkronisasi Waktu : 34268572 + -10127836 = 24140736
radi 24:140:736
```

(1)

(2)

Gambar 6.8 Rincian Pengujian Sinkronisasi

Perhitungan dilakukan sesuai rumus sinkronisasi RBS yang telah dijelaskan pada bab sebelumnya. Sementara untuk perhitungan waktu observasi di tuliskan pada Tabel 6.10.

Tabel 6.9 Perhitungan Manual Sinkronisasi Waktu

| No Referece Packet | Waktu Penerimaan data Node 1 | Waktu Penerimaan data Node 2 |
|--|------------------------------|------------------------------|
| 1 | 22804792 | 32932108 |
| 2 | 22906168 | 33033596 |
| 3 | 23007508 | 33135048 |
| 4 | 23108908 | 33236536 |
| 5 | 23210244 | 33338040 |
| 6 | 23311620 | 33439532 |
| 7 | 23412996 | 33540996 |
| 8 | 23514372 | 33642500 |
| 9 | 23615748 | 33744000 |
| 10 | 23717208 | 33845572 |
| Total waktu ($\sum_{k=1}^m$) | 232609564 | 333887928 |
| Waktu Observasi ($\frac{1}{m} \sum_{k=1}^m$) | 23260956 | 33388792 |

Setelah diketahui waktu observasi, maka setiap node saling mengirimkan waktu observasi masing masing dan melakukan perhitungan seperti pada rumus.

$$Phase\ Offset = \frac{1}{m} \sum_{k=1}^m (T_{jk} - T_{ik}) = \left(\frac{1}{m} \sum_{k=1}^m T_{jk} \right) - \left(\frac{1}{m} \sum_{k=1}^m T_{ik} \right) \quad (6.1)$$

Perhitungan Phase Offset

$$Node\ 1 = 33388792 - 23260956 = 10127836$$

$$Node\ 2 = 23260956 - 33388792 = -10127836$$

Pada proses sinkronisasi, nilai phase offset yang digunakan adalah yang negatif. Apabila node memiliki nilai phase offset positif maka phase offsetnya menjadi 0. Sebelumnya dicatat waktu lokal yang terbaru untuk digunakan dalam sinkronisasi.

$$Sinkronisasi\ waktu\ Node\ 1 = 24142308 + 0 = 24142308$$

$$Sinkronisasi\ waktu\ Node\ 2 = 34268572 + (-10127836) = 24140736$$

$$Error = \text{Selisih sinkronisasi waktu node 1 dan sinkronisasi waktu node 1}$$

$$Error = 24142308 - 24140736 = 1572$$

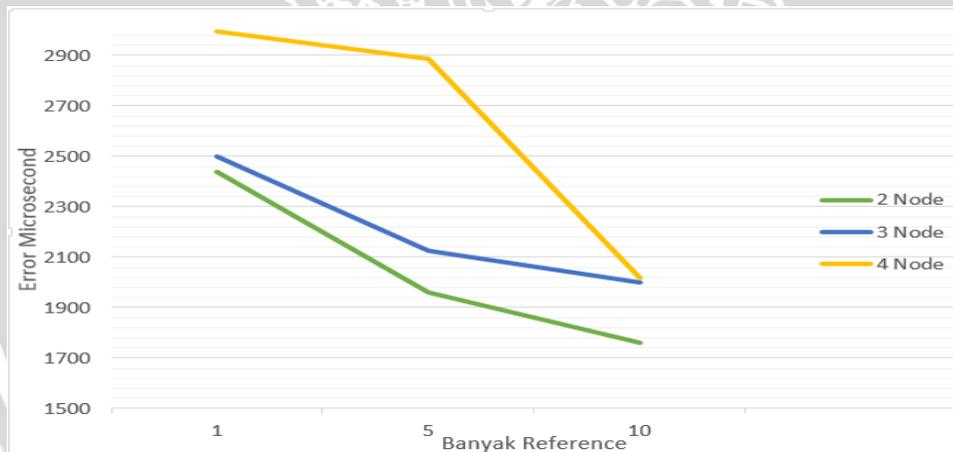
6.2.3 Analisa Pengujian

Analisa hasil pengujian pertama-tama dilakukan dengan membuat penyederhanaan dari hasil nilai rata-rata pengujian sinkronisasi waktu. Nilai tersebut dituliskan pada Tabel 6.9.

Tabel 6.10 Nilai Rata-Rata Pengujian Sinkronisasi Waktu

| | | 2 Node | 3 Node | 4 Node |
|-----------------|----|----------|--------|--------|
| Paket referensi | 1 | 2440.4 | 2500.3 | 2993.4 |
| | 5 | 1959.4 | 2124.6 | 2884.1 |
| | 10 | 1759.2 | 1999.5 | 2019 |
| Rata-rata | | 2297.766 | | |

Dari tabel diatas dapat digambarkan sebuah grafik yang mewakili hasil nilai rata-rata sinkronisasi waktu. Grafik digambarkan dengan sumbu horizontal mewakili banyak paket referensi yang dikirim sementara sumbu vertikal mewakili nilai *error*. Sementara itu nilai rata-rata diwakilkan oleh titik koordinat dan disambungkan dengan garis. Seperti pada grafik 6.4.



Gambar 6.9 Grafik Nilai Rata-Rata Pengujian Sinkronisasi Waktu

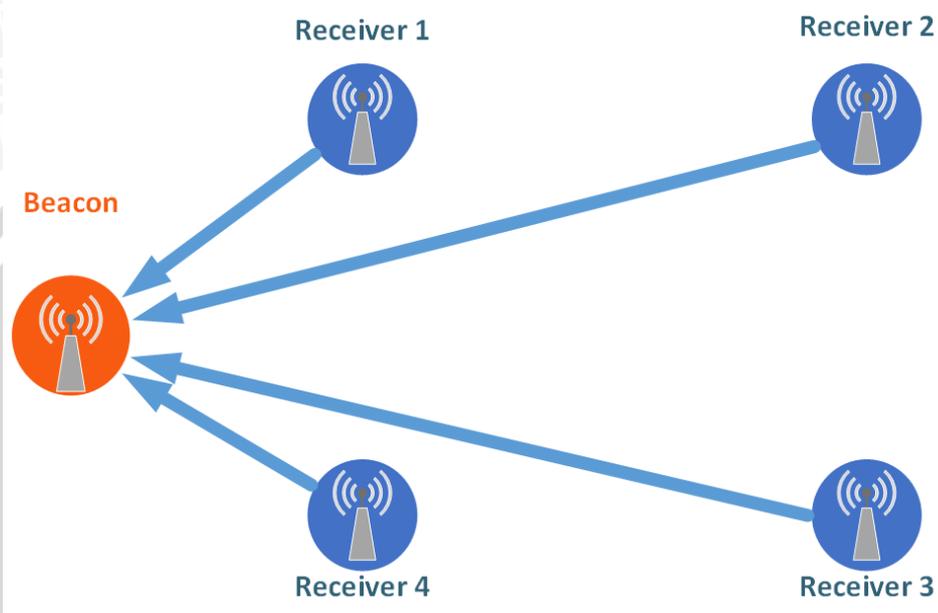
Dari grafik diatas dapat dianalisa bahwa dengan bertambahnya node yang di sinkronisasi, menambah *error* waktu sinkronisasi. Hal ini dikarenakan beban sinkronisasi bertambah yaitu setiap node harus membandingkan lebih banyak waktu observasi untuk melakukan sinkronisasi waktu.

Namun sebaliknya dengan bertambahnya paket referensi yang dikirimkan mengurangi tingkat *error* waktu sinkronisasi. Hal ini dikarenakan banyak paket referensi membuat node dapat melakukan pengambilan sample waktu penerimaan yang lebih banyak sehingga rata-rata waktu lebih baik daripada yang menggunakan paket referensi yang lebih sedikit.

Keberanan perhitungan sinkronisasi waktu dibuktikan dengan percobaan matematis. Dari pengujian matematis, didapatkan sinkronisasi waktu pada penelitian ini sama dengan perhitungan secara matematis (manual).

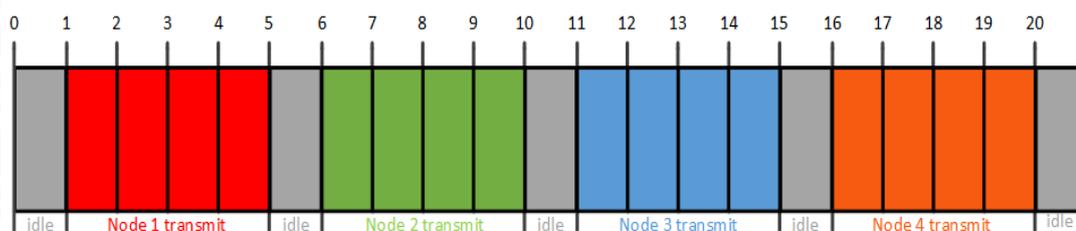
6.3 Pengujian Penjadwalan

Pada pengujian penjadwalan mempresentasikan pengiriman data dari tiap node *receiver* kepada node *beacon* yang terjadwal sesuai slot waktu yang telah ditentukan. Lihat pada Gambar 6.10.



Gambar 6.10 Skenario Pengujian Penjadwalan

Pada pengujian ini terdapat 4 slot waktu yaitu 2, 3, 4 dan 5 (dalam detik). Pada setiap slot waktu memiliki waktu toleransi selama 1 detik yang berguna untuk transisi mode pengiriman ke mode penerimaan. Pengujian dilakukan dengan melihat pengiriman data dari tiap node selama 30 detik. Tiap satu siklus pengiriman terdapat slot waktu sesuai banyaknya node dan node mengirimkan data 1 kali dalam tiap detik sesuai slot waktu yang ada. Lihat pada gambar 6.11 yang merupakan ilustrasi menggunakan slot waktu 5 detik menggunakan 4 node. Terdapat 1 detik waktu toleransi dan 4 detik sisanya digunakan untuk mengirim data sebanyak 4 kali.



Gambar 6.11 Ilustrasi Penjadwalan

Hasil dari pengujian dituliskan dalam bentuk prosentase Pengujian ini bertujuan untuk melihat kemampuan metode TDMA dalam memberikan penjadwalan untuk tiap node *receiver* dalam mengirimkan data kepada node *beacon*.

Tabel 6.11 Prosedur Percobaan Penjadwalan

| | |
|-------------------------|--|
| Kasus Pengujian | Penjadwalan pengiriman data dari node <i>receiver</i> ke node <i>beacon</i> |
| Objek Pengujian | Kesesuaian pengiriman data node <i>receiver</i> dengan jadwal yang telah ditentukan dengan TDMA |
| Tujuan Pengujian | Melihat pengaruh dari penjadwalan dalam menghindari tabrakan data pada WSN |
| Prosedur Uji | <ol style="list-style-type: none"> 1. Penguji mengunggah program kepada setiap node 2. Penguji menginputkan slot waktu yang digunakan dalam penjadwalan. 3. Penguji melihat serial monitor dan menganalisa ketepatan pengiriman data dan memantau adanya data yang tidak dapat diterima node <i>beacon</i> (tabrakan data). 4. Penguji mencatat hasil pengujian. 5. Pengujian dilakukan menggunakan slot waktu 2, 3, 4 dan 5 (termasuk 1 detik waktu transisi). Dilakukan pada <i>receiver</i> yang berjumlah 2, 3 dan 4. |

6.3.1 Hasil Pengujian

Hasil pengujian digambarkan pada sebuah tabel yang merangkum semua pengujian yang telah dilakukan. Detail pengujian dapat dilihat pada Lampiran A1-A12. Rangkuman hasil pengujian penjadwalan dapat dilihat pada Tabel 6.12.

Tabel 6.12 Rangkuman Pengujian Penjadwalan

| Percobaan | | Jumlah Node | | |
|------------|---|------------------------------------|------------------------------------|------------------------------------|
| | | 2 | 3 | 4 |
| Slot Waktu | 2 | Berhasil Tidak Ada <i>Error</i> | Berhasil Tidak Ada <i>Error</i> | Berhasil Tidak Ada <i>Error</i> |
| | 3 | Berhasil Tidak Ada <i>Error</i> | Berhasil Tidak Ada <i>Error</i> | Berhasil Tidak Ada <i>Error</i> |
| | 4 | Berhasil Tidak Ada <i>Error</i> | Berhasil Tidak Ada <i>Error</i> | Berhasil Tidak Ada <i>Error</i> |
| | 5 | Berhasil Tidak Ada <i>Error</i> | Berhasil Tidak Ada <i>Error</i> | Berhasil Tidak Ada <i>Error</i> |

Tabel 6.12 menjelaskan bahwa pada pengujian penjadwalan tidak ada *error* pengiriman data. Hal ini dikarenakan setiap node *receiver* telah memiliki jadwal sendiri untuk mengirimkan data. Selama waktu tiap node sinkron dapat dipastikan penjadwalan pengiriman data bekerja dengan baik.

6.3.2 Analisa Pengujian

Dari hasil pengujian yang telah dilakukan. Tidak terjadi tabrakan data pada pengiriman paket dari *receiver* ke *beacon*. Hal ini terjadi karena setiap node mengirimkan data sesuai dengan jadwal yang telah ditentukan dengan metode TDMA. Selain itu waktu transisi juga berguna untuk mentoleransi *error* dari sinkronisasi waktu. Sehingga *error* yang terdapat pada sinkronisasi waktu RBS tidak mengganggu jadwal pengiriman.



BAB 7 PENUTUP

Pada bab ini penulis memberikan kesimpulan dari semua yang telah dituliskan pada bab sebelumnya. Selain itu penulis juga memberikan saran agar penelitian ini dapat dilanjutkan agar penelitian ini menjadi lebih baik dan berguna bagi pembaca.

7.1 Kesimpulan

Berdasarkan perancangan, implementasi, pengujian, dan analisis yang telah dilakukan, maka penulis menyimpulkan:

1. Sistem yang terdiri dari dua komponen utama yaitu node *beacon* dan *receiver* yang masing masing terdiri dari mikrokontroler Arduino Pro mini dan modul tranceiver nRF24L01, dapat melakukan tugas fungsionalnya dengan baik yaitu dalam hal pengiriman dan penerimaan data.
2. Dilihat dari hasil sinkronisasi RBS, banyaknya node yang harus disinkronisasi mengakibatkan bertambahnya *error* waktu dari tiap node. Namun banyaknya paket referensi yang diberikan saat sinkronisasi menjadikan *error* waktu semakin kecil.
3. Hasil dari implementasinya adalah setiap node memiliki jadwal pengiriman data berupa slot waktu, sehingga tiap node tidak dapat mengirimkan data pada waktu yang bersamaan. Sementara *error* waktu sinkronisasi dapat diatasi dengan memberikan waktu transisi tiap bergantinya jadwal pengiriman, hal berguna untuk menoleransi adanya *error* sinkronisasi waktu.
4. Sinkronisasi RBS menggunakan 2-4 *receiver* dan 2,5,10 reference packet mampu mencapai ketelitian sinkronisasi (*error* waktu sinkronisasi) hingga satuan waktu *milisecond*. Yaitu dengan *error* waktu rata-rata 2.298 *millisecond* (2297.766 *microsecond*). Sementara penjadwalan TDMA dengan memberikan jadwal pengiriman data dengan waktu pengiriman 2-5 detik (termasuk waktu transisi) berhasil memberikan penjadwalan pada tiap node sehingga didapatkan pengiriman data tanpa adanya *error* karena tabrakan data.

7.2 Saran

Berikut beberapa saran yang diharapkan dalam pengembangan sistem ini.

1. Disarankan untuk diterapkan dengan modul WSN yang berbeda seperti zigbee atau yang lainnya.
2. Mengembangkan sistem penjadwalan dan sinkronisasi waktu berbasis non-IP.
3. Metode TDMA dalam WSN juga dapat digunakan dalam penelitian manajemen daya.

4. Menggunakan metode yang berbeda dan melakukan perbandingan kualitas, sehingga didapatkan metode terbaik penjadwalan dan sinkronisasi waktu pada WSN.



DAFTAR PUSTAKA

- AlSkaif, T., Zapata, M. G. & Boris, B., 2015. Game theory for energy efficiency in Wireless Sensor Networks: Latest trends. *Network and Computer Applications*.
- Arduino, 2016. *Arduino Pro Mini*. [Online] Tersedia di: <https://www.arduino.cc/en/Main/arduinoBoardProMini> [Diakses pada 1 Maret 2016].
- Chantrell, N., 2013. *Experimenting with the nRF24L01+ 2.4GHz radios*. [Online] Tersedia di: <https://nathan.chantrell.net/20130810/experimenting-with-the-nrf24l01-2-4ghz-radios/> [Diakses pada 1 Maret 2016].
- Chu, Y. et al., 2015. Application of reinforcement learning to medium access control for wireless sensor networks. *Engineering Applications of Artificial Intelligence*, pp. 23-32.
- Elson, J. & Estrin, D., 2001. *Time Synchronization for Wireless Sensor Networks*. San Francisco, UCLA CS Technical Report 200028.
- Elson, J., Girod, L. & Estrin, D., 2002. *Fine-Grained Network Time Synchronization using Reference Broadcasts*. Boston, UCLA Computer Science Technical Report 020008.
- Kumar, S. & Chauhan, S., 2011. A Survey on Scheduling Algorithms for Wireless. *International Journal of Computer Applications*.
- Lasassmeh, S. M. & Conrad, J. M., 2010. Time synchronization in wireless sensor networks: A survey. *Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon)*, pp. 242--245.
- Wang, F. & Liu, J., 2011. Networked Wireless Sensor Data Collection. *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, Volume 13.
- Yinbiao, S. et al., 2014. *Internet of Things: Wireless Sensor Networks*. Geneva, Switzerland: International Electrotechnical Commission.





LAMPIRAN A PENGUJIAN PENJADWALAN

A.1 Penjadwalan Pengiriman 2 Node dengan Time Slot 2 detik

| Percobaan | | 1 | | 2 | | 3 | | 4 | | 5 | |
|-----------|----|------|---|---|---|---|---|---|---|---|---|
| | | Node | | | | | | | | | |
| | | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 |
| Time | 1 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 2 | | | | | | | | | | |
| | 3 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 4 | | | | | | | | | | |
| | 5 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 6 | | | | | | | | | | |
| | 7 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 8 | | | | | | | | | | |
| | 9 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 10 | | | | | | | | | | |
| | 11 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 12 | | | | | | | | | | |
| | 13 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 14 | | | | | | | | | | |
| | 15 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 16 | | | | | | | | | | |
| | 17 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 18 | | | | | | | | | | |
| | 19 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 20 | | | | | | | | | | |
| | 21 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 22 | | | | | | | | | | |
| | 23 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 24 | | | | | | | | | | |
| | 25 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 26 | | | | | | | | | | |
| | 27 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 28 | | | | | | | | | | |
| | 29 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 30 | | | | | | | | | | |

A.2 Penjadwalan Pengiriman 2 Node dengan Time Slot 3 detik

| Percobaan | | 1 | 2 | 3 | 4 | 5 | | | | | |
|-----------|----|------|---|---|---|---|---|---|---|---|---|
| | | Node | | | | | | | | | |
| | | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 |
| Time | 1 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 2 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 3 | | | | | | | | | | |
| | 4 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 5 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 6 | | | | | | | | | | |
| | 7 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 8 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 9 | | | | | | | | | | |
| | 10 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 11 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 12 | | | | | | | | | | |
| | 13 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 14 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 15 | | | | | | | | | | |
| | 16 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 17 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 18 | | | | | | | | | | |
| | 19 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 20 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 21 | | | | | | | | | | |
| | 22 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 23 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 24 | | | | | | | | | | |
| | 25 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 26 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 27 | | | | | | | | | | |
| | 28 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 29 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 30 | | | | | | | | | | |

A.3 Penjadwalan Pengiriman 2 Node dengan Time Slot 4 detik

| Percobaan | | 1 | | 2 | | 3 | | 4 | | 5 | |
|-----------|------|------|---|---|---|---|---|---|---|---|---|
| | Time | Node | | | | | | | | | |
| | | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 |
| 1 | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ | |
| 2 | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ | |
| 3 | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ | |
| 4 | | | | | | | | | | | |
| 5 | | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ |
| 6 | | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ |
| 7 | | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ |
| 8 | | | | | | | | | | | |
| 9 | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ | |
| 10 | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ | |
| 11 | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ | |
| 12 | | | | | | | | | | | |
| 13 | | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ |
| 14 | | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ |
| 15 | | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ |
| 16 | | | | | | | | | | | |
| 17 | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ | |
| 18 | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ | |
| 19 | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ | |
| 20 | | | | | | | | | | | |
| 21 | | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ |
| 22 | | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ |
| 23 | | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ |
| 24 | | | | | | | | | | | |
| 25 | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ | |
| 26 | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ | |
| 27 | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ | |
| 28 | | | | | | | | | | | |
| 29 | | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ |
| 30 | | ✓ | | | ✓ | | ✓ | | ✓ | | ✓ |

A.4 Penjadwalan Pengiriman 2 Node dengan Time Slot 5 detik

| Percobaan | | 1 | | 2 | | 3 | | 4 | | 5 | |
|-----------|----|------|---|---|---|---|---|---|---|---|---|
| | | Node | | | | | | | | | |
| | | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 |
| Time | 1 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 2 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 3 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 4 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 5 | | | | | | | | | | |
| | 6 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 7 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 8 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 9 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 10 | | | | | | | | | | |
| | 11 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 12 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 13 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 14 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 15 | | | | | | | | | | |
| | 16 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 17 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 18 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 19 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 20 | | | | | | | | | | |
| | 21 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 22 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 23 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 24 | ✓ | | ✓ | | ✓ | | ✓ | | ✓ | |
| | 25 | | | | | | | | | | |
| | 26 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 27 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 28 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 29 | | ✓ | | ✓ | | ✓ | | ✓ | | ✓ |
| | 30 | | | | | | | | | | |

A.5 Penjadwalan Pengiriman 3 Node dengan Time Slot 2 detik

| Percobaan | | 1 | | | 2 | | | 3 | | | 4 | | | 5 | | |
|-----------|------|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Time | Node | | | | | | | | | | | | | | |
| | | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| 1 | ✓ | | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| 2 | | | | | | | | | | | | | | | | |
| 3 | | ✓ | | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| 4 | | | | | | | | | | | | | | | | |
| 5 | | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| 6 | | | | | | | | | | | | | | | | |
| 7 | ✓ | | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| 8 | | | | | | | | | | | | | | | | |
| 9 | | ✓ | | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| 10 | | | | | | | | | | | | | | | | |
| 11 | | | ✓ | | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| 12 | | | | | | | | | | | | | | | | |
| 13 | ✓ | | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| 14 | | | | | | | | | | | | | | | | |
| 15 | | ✓ | | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| 16 | | | | | | | | | | | | | | | | |
| 17 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| 18 | | | | | | | | | | | | | | | | |
| 19 | ✓ | | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| 20 | | | | | | | | | | | | | | | | |
| 21 | | ✓ | | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| 22 | | | | | | | | | | | | | | | | |
| 23 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| 24 | | | | | | | | | | | | | | | | |
| 25 | ✓ | | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| 26 | | | | | | | | | | | | | | | | |
| 27 | | ✓ | | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| 28 | | | | | | | | | | | | | | | | |
| 29 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| 30 | | | | | | | | | | | | | | | | |

A.6 Penjadwalan Pengiriman 3 Node dengan Time Slot 3 detik

| Percobaan | 1 | | | 2 | | | 3 | | | 4 | | | 5 | | |
|-----------|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Node | | | | | | | | | | | | | | |
| | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| 1 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| 2 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| 3 | | | | | | | | | | | | | | | |
| 4 | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| 5 | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| 6 | | | | | | | | | | | | | | | |
| 7 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| 8 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| 9 | | | | | | | | | | | | | | | |
| 10 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| 11 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| 12 | | | | | | | | | | | | | | | |
| 13 | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| 14 | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| 15 | | | | | | | | | | | | | | | |
| 16 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| 17 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| 18 | | | | | | | | | | | | | | | |
| 19 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| 20 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| 21 | | | | | | | | | | | | | | | |
| 22 | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| 23 | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| 24 | | | | | | | | | | | | | | | |
| 25 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| 26 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| 27 | | | | | | | | | | | | | | | |
| 28 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| 29 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| 30 | | | | | | | | | | | | | | | |

A.7 Penjadwalan Pengiriman 3 Node dengan Time Slot 4 detik

| Percobaan | | 1 | | | 2 | | | 3 | | | 4 | | | 5 | | |
|-----------|----|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Node | | | | | | | | | | | | | | |
| | | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| Time | 1 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| | 2 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| | 3 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| | 4 | | | | | | | | | | | | | | | |
| | 5 | | ✓ | | | ✓ | | | | ✓ | | | ✓ | | | ✓ |
| | 6 | | ✓ | | | ✓ | | | | ✓ | | | ✓ | | | ✓ |
| | 7 | | ✓ | | | ✓ | | | | ✓ | | | ✓ | | | ✓ |
| | 8 | | | | | | | | | | | | | | | |
| | 9 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| | 10 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| | 11 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| | 12 | | | | | | | | | | | | | | | |
| | 13 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| | 14 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| | 15 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| | 16 | | | | | | | | | | | | | | | |
| | 17 | | ✓ | | | ✓ | | | | ✓ | | | ✓ | | | ✓ |
| | 18 | | ✓ | | | ✓ | | | | ✓ | | | ✓ | | | ✓ |
| | 19 | | ✓ | | | ✓ | | | | ✓ | | | ✓ | | | ✓ |
| | 20 | | | | | | | | | | | | | | | |
| | 21 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| | 22 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| | 23 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| | 24 | | | | | | | | | | | | | | | |
| | 25 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| | 26 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| | 27 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| | 28 | | | | | | | | | | | | | | | |
| | 29 | | ✓ | | | ✓ | | | | ✓ | | | ✓ | | | ✓ |
| | 30 | | ✓ | | | ✓ | | | | ✓ | | | ✓ | | | ✓ |

A.8 Penjadwalan Pengiriman 3 Node dengan Time Slot 5 detik

| Percobaan | | 1 | | | 2 | | | 3 | | | 4 | | | 5 | | |
|-----------|----|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Node | | | | | | | | | | | | | | |
| | | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |
| Time | 1 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| | 2 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| | 3 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| | 4 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| | 5 | | | | | | | | | | | | | | | |
| | 6 | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| | 7 | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| | 8 | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| | 9 | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| | 10 | | | | | | | | | | | | | | | |
| | 11 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| | 12 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| | 13 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| | 14 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| | 15 | | | | | | | | | | | | | | | |
| | 16 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| | 17 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| | 18 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| | 19 | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | |
| | 20 | | | | | | | | | | | | | | | |
| | 21 | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| | 22 | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| | 23 | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| | 24 | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ | |
| | 25 | | | | | | | | | | | | | | | |
| | 26 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| | 27 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| | 28 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| | 29 | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| | 30 | | | | | | | | | | | | | | | |

A.9 Penjadwalan Pengiriman 4 Node dengan Time Slot 2 detik

| Percobaan | | 1 | | | | 2 | | | | 3 | | | | 4 | | | | 5 | | | | | | | |
|-----------|------|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Time | Node | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | | | | |
| 1 | 1 | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | |
| 1 | 2 | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 3 | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | |
| 1 | 4 | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 5 | | | ✓ | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | |
| 1 | 6 | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 7 | | | | ✓ | | | | ✓ | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | |
| 1 | 8 | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 9 | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | |
| 1 | 10 | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 11 | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | |
| 1 | 12 | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 13 | | | ✓ | | | | ✓ | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | |
| 1 | 14 | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 15 | | | | ✓ | | | ✓ | | | ✓ | | | ✓ | | | | ✓ | | | | ✓ | | | ✓ |
| 1 | 16 | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 17 | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | |
| 1 | 18 | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 19 | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | |
| 1 | 20 | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 21 | | | ✓ | | | | ✓ | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | |
| 1 | 22 | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 23 | | | | ✓ | | | | ✓ | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | |
| 1 | 24 | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 25 | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | |
| 1 | 26 | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 27 | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | |
| 1 | 28 | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 29 | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | |
| 1 | 30 | | | | | | | | | | | | | | | | | | | | | | | | |

A.10 Penjadwalan Pengiriman 4 Node dengan Time Slot 3 detik

| Percobaan | | 1 | | | | 2 | | | | 3 | | | | 4 | | | | 5 | | | |
|-----------|----|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | n | Node | | | | | | | | | | | | | | | | | | | |
| | | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| Time | 1 | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | |
| | 2 | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | |
| | 3 | | | | | | | | | | | | | | | | | | | | |
| | 4 | | ✓ | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | |
| | 5 | | ✓ | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | |
| | 6 | | | | | | | | | | | | | | | | | | | | |
| | 7 | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | |
| | 8 | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | |
| | 9 | | | | | | | | | | | | | | | | | | | | |
| | 10 | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ |
| | 11 | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ |
| | 12 | | | | | | | | | | | | | | | | | | | | |
| | 13 | ✓ | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | | |
| | 14 | ✓ | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | | |
| | 15 | | | | | | | | | | | | | | | | | | | | |
| | 16 | | ✓ | | | | ✓ | | | | | ✓ | | | ✓ | | | | ✓ | | |
| | 17 | | ✓ | | | | ✓ | | | | | ✓ | | | ✓ | | | | ✓ | | |
| | 18 | | | | | | | | | | | | | | | | | | | | |
| | 19 | | | ✓ | | | | ✓ | | | | ✓ | | | ✓ | | | | ✓ | | |
| | 20 | | | ✓ | | | | ✓ | | | | ✓ | | | ✓ | | | | ✓ | | |
| | 21 | | | | | | | | | | | | | | | | | | | | |
| | 22 | | | | ✓ | | | ✓ | | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| | 23 | | | | ✓ | | | ✓ | | | | ✓ | | | ✓ | | | ✓ | | | ✓ |
| | 24 | | | | | | | | | | | | | | | | | | | | |
| | 25 | ✓ | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | | |
| | 26 | ✓ | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | | |
| | 27 | | | | | | | | | | | | | | | | | | | | |
| | 28 | | ✓ | | | | ✓ | | | | | ✓ | | | | ✓ | | | ✓ | | |
| | 29 | | ✓ | | | | ✓ | | | | | ✓ | | | | ✓ | | | ✓ | | |
| | 30 | | | | | | | | | | | | | | | | | | | | |



A.11 Penjadwalan Pengiriman 4 Node dengan Time Slot 4detik

| Percobaan n | 1 | | | | 2 | | | | 3 | | | | 4 | | | | 5 | | | | |
|----------------|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Node | | | | | | | | | | | | | | | | | | | | |
| | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | |
| Time | 1 | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | |
| | 2 | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | |
| | 3 | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | |
| | 4 | | | | | | | | | | | | | | | | | | | | |
| | 5 | | ✓ | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | |
| | 6 | | ✓ | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | |
| | 7 | | ✓ | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | |
| | 8 | | | | | | | | | | | | | | | | | | | | |
| | 9 | | | ✓ | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ |
| | 10 | | | ✓ | | | | ✓ | | | | | | ✓ | | | | ✓ | | | ✓ |
| | 11 | | | ✓ | | | | ✓ | | | | | | | ✓ | | | | ✓ | | ✓ |
| | 12 | | | | | | | | | | | | | | | | | | | | |
| | 13 | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ |
| | 14 | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ |
| | 15 | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ |
| | 16 | | | | | | | | | | | | | | | | | | | | |
| | 17 | ✓ | | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | |
| | 18 | ✓ | | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | |
| | 19 | ✓ | | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | |
| | 20 | | | | | | | | | | | | | | | | | | | | |
| | 21 | | ✓ | | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ |
| | 22 | | ✓ | | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ |
| | 23 | | ✓ | | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ |
| | 24 | | | | | | | | | | | | | | | | | | | | |
| | 25 | | | ✓ | | | | | ✓ | | | | | ✓ | | | | ✓ | | | ✓ |
| | 26 | | | ✓ | | | | | ✓ | | | | | ✓ | | | | ✓ | | | ✓ |
| | 27 | | | ✓ | | | | | ✓ | | | | | ✓ | | | | ✓ | | | ✓ |
| | 28 | | | | | | | | | | | | | | | | | | | | |
| | 29 | | | | ✓ | | | | | ✓ | | | | | ✓ | | | | | | ✓ |
| | 30 | | | | ✓ | | | | | ✓ | | | | | ✓ | | | | | | ✓ |

A.12 Penjadwalan Pengiriman 4 Node dengan Time Slot 5 detik

| Percobaan | | 1 | | | | 2 | | | | 3 | | | | 4 | | | | 5 | | | | | | | |
|-----------|----|------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--|
| | | Node | | | | | | | | | | | | | | | | | | | | | | | |
| | | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | | | | |
| Time | 1 | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | |
| | 2 | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | |
| | 3 | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | |
| | 4 | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | |
| | 5 | | | | | | | | | | | | | | | | | | | | | | | | |
| | 6 | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | |
| | 7 | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | |
| | 8 | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | |
| | 9 | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | |
| | 10 | | | | | | | | | | | | | | | | | | | | | | | | |
| | 11 | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | | | | | ✓ | | |
| | 12 | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | | | | | ✓ | | |
| | 13 | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | | | | | ✓ | | |
| | 14 | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | | | | | ✓ | | |
| | 15 | | | | | | | | | | | | | | | | | | | | | | | | |
| | 16 | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | | | | | ✓ | | |
| | 17 | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | | | | | ✓ | | |
| | 18 | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | | | | | ✓ | | |
| | 19 | | | | ✓ | | | | | ✓ | | | | ✓ | | | | ✓ | | | | | ✓ | | |
| | 20 | | | | | | | | | | | | | | | | | | | | | | | | |
| | 21 | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | |
| | 22 | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | |
| | 23 | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | |
| | 24 | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | |
| | 25 | | | | | | | | | | | | | | | | | | | | | | | | |
| | 26 | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | |
| | 27 | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | |
| | 28 | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | |
| | 29 | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | | | ✓ | | |
| | 30 | | | | | | | | | | | | | | | | | | | | | | | | |