

Integrasi Metode Pengalamatan Dinamis *Treecast* Dan Sinkronisasi Waktu Dengan *Reference Broadcast Synchronization* Pada Sensor Network

Maulita Intan Kripsita¹, Sabriansyah Rizqika Akbar², Mochammad Hannats Hanafi Ichsan³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹maulitaintankripsita@gmail.com, ²sabrian@ub.ac.id, ³hanas.hanafi@ub.ac.id

Abstrak

Teknologi saat ini semakin berkembang dengan pesat. Salah satunya *Wireless Sensor Network* (WSN) yang merupakan jaringan yang terhubung dengan perangkat seperti *sink node*, *router* dan *sensor node*. Namun perkembangannya menimbulkan permasalahan jika *node* sensor di WSN harus mengkonfigurasi manual. Setiap *node* di WSN memiliki frekuensi sama ketika berkomunikasi dan waktu lokal berbeda, memungkinkan *node* mengirim data pada waktu bersamaan yang mengakibatkan tabrakan data sehingga data tidak diterima dan tidak dapat disebarkan pada *node* lain. Untuk mengatasinya diterapkan metode *Treecast* dan *Reference Broadcast Synchronization* (RBS) yang diintegrasikan. *Treecast* memberikan alamat ke *sensor node* secara dinamis dan menjaga komunikasi agar bisa dilakukan walaupun ada kegagalan. Sedangkan *RBS* mampu memberikan penyetaraan waktu pada setiap *node*. Dengan satu *beacon* sebagai pengirim paket referensi, *beacon* memberikan perintah untuk melakukan sinkronisasi. Selanjutnya *receiver* melakukan sinkronisasi dan bertukar waktu dengan *receiver* lainnya hingga mendapatkan perbedaan waktu dan dijadikan acuan sinkronisasi waktu. Hal ini membuat *node* mampu memberikan jaminan dalam hal pengiriman-penerimaan data serta tidak ada *collision*. Hasil yang didapatkan adalah pengalamatan dinamis *Treecast* mampu diimplementasikan secara nyata dan memberikan alamat serta waktu untuk mendapatkan alamat pada tiap *node*. Kemudian sinkronisasi waktu yang menggunakan *RBS* memakai 4 *node receiver* dan 5 paket referensi yang mampu mencapai ketelitian sinkronisasi waktu sampai satuan waktu millisecond.

Kata Kunci : WSN, *Treecast*, pengalamatan dinamis, RBS, sinkronisasi waktu, tabrakan data.

Abstract

Technology is now growing rapidly. One of them is *Wireless Sensor Network* (WSN) which is a network connected to devices such as *sink node*, *router* and *sensor node*. However, the development is a problem if the *node* sensor in the *Wireless Sensor Network* to configure it manually. Each *node* in *Wireless Sensor Network* have the same frequency when communicating and local time is different, allowing *node* to send data at the same time that resulted in a collision data so that data is not received and can not be propagated on other nodes. To solve this problem, then applied the method *Treecast* and *Reference Broadcast Synchronization* (RBS) will be integrated. *Treecast* gives the address to the *sensor node* in a dynamic and keep the communication in order to remain able to do even if there are failures. While *Reference Broadcast Synchronization* is able to provide time equalization on each *node*. With a *node beacon* as the sender of the reference, *beacon* to give the order to synchronise. Next the *nodes receiver* synchronizes and share time with other receivers to get time difference and is used as time synchronization reference. This makes the *node* is able to provide guarantee in terms of sending and receiving data and no collision. The research results obtained are capable of dynamic addressing *Treecast* implemented in practice and is able to provide an address and a time to get the address of each *sensor nodes* in a *sensor network network*. Then for time synchronization using *Reference Broadcast Synchronization* *node receiver* uses 4 and 5 reference package that can achieve accuracy time synchronization to the unit millisecond time.

Keywords : WSN, *treecast*, dynamic addressing, RBS, time syhchronization, data collisons.

1. PENDAHULUAN

Wireless sensor network adalah jaringan *wireless* yang menggunakan sensor untuk memonitor fisik atau kondisi lingkungan seperti cuaca, suhu, gelombang elektromagnetik, tekanan, gerakan, getaran dan lain sebagainya (Setiyo budi, 2012). Untuk ukuran suatu *node* sensor pada WSN memiliki kisaran node sensor yang bisa mencapai besarnya dari sebuah kotak sepatu hingga seukuran debu. Aplikasi dan penggunaan dari WSN ada banyak dan bervariasi, tapi umumnya adalah untuk *monitoring*, *tracking* dan *controlling*. Dalam perkembangannya WSN pada masa kini banyak diterapkan di bidang militer, pengawasan dan control dibidang industri, kesehatan, dan bidang lainnya. Sensor-sensor pada WSN memiliki sifat dinamis, karena obyek yang dimonitoring terkadang bukan hanya obyek statis melainkan obyek yang sifatnya bergerak.

Dengan perkembangan WSN tersebut, tentunya menimbulkan permasalahan itu sendiri jika *node-node* sensor yang ada pada WSN harus dikonfigurasi secara manual. Tentunya mengurangi performa dan skalabilitas pada WSN itu sendiri dan menghabiskan sumber daya. Maka untuk mengatasi hal tersebut perlu menggunakan suatu metode yang bisa mengkonfigurasi *node-node* sensor tersebut secara dinamis tanpa harus ada campur tangan manusia. Selain itu dalam suatu pengiriman data jika terjadi pengiriman secara bersamaan tentunya terjadi tabrakan data yang membuat data tersebut tidak diterima. Tentunya hal ini mengakibatkan informasi tidak dapat disebarkan. Maka dari itu untuk menangani masalah tersebut harus memiliki metode dan aturan. Metode yang digunakan yaitu metode yang bisa memberikan waktu pada *node* saat pengiriman dan penerimaan data, agar tidak terjadi tabrakan sewaktu mengirimkan data secara bersamaan.

Dengan adanya masalah diatas banyak peneliti yang mengembangkan suatu metode untuk pengalamatan secara dinamis dengan *Treecast* salah satunya ialah penelitian yang berjudul “ *TreeCast : A Stateless Addressing and Routing Architecture for Sensor Networks* “. Peneliti ini menggunakan metode *Treecast* pada suatu jaringan *Wireless Sensor Network*. Penelitian yang membahas mengenai penerapan alokasi pengalamatan secara dinamis pada WSN sudah dilakukan oleh Santashill Pal Chaudhuri

dkk. Menerapkan metode *Treecast* ini dalam pemberian alamat secara dinamis. Sistem ini bertujuan untuk memberikan alamat ke *node* sensor secara dinamis, selain itu menjaga komunikasi sensor agar tetap dilakukan meski ada kegagalan dalam salah satu node pada WSN (Santashill Pal Chaudhuri dkk,2004).

Sedangkan penelitian yang lainnya ini mengembangkan suatu metode untuk sinkronisasi waktu salah satunya adalah penelitian yang berjudul “ *Fine-Grained Network Time Synchronization using Reference Broadcast* “. Dimana peneliti menggunakan metode *Reference Broadcast Synchronization* atau RBS pada *Wireless Sensor Network*. Dengan menggunakan satu node beacon, beacon merupakan node yang mengirimkan paket referensi. *Beacon* diberikan perintah untuk melakukan sinkronisasi. Lalu setelah itu para receiver melakukan sinkronisasi. Kemudian *receiver* saling bertukar waktu dengan *receiver* lainnya. Jadi selanjutnya didapatkan perbedaan antara waktu dan *receiver* yang dijadikan acuan untuk mensinkronisasikan waktu masing-masing.

Dengan adanya masalah yang ada penulis memiliki usulan dengan menggabungkan kedua sistem tersebut. Sistem yang digabungkan adalah metode *Treecast* untuk pengalamatan secara dinamis dan Sinkronisasi waktu dengan RBS atau *Reference Broadcast Synchronization*. Harapan bagi penulis kedua sistem tersebut dapat berhasil dilakukan guna memberikan jaminan dalam hal pengiriman-penerimaan data atau informasi.

2. DASAR TEORI

2.1 *Wireless Sensor Network*

WSN (*Wireless Sensor Network*) merupakan suatu jaringan *wireless* yang terdiri dari kumpulan *node* yang bersifat *dynamic* dan WSN itu sendiri dapat langsung melakukan komunikasi antara satu *node* ke *node* yang lainnya karena *node* memiliki kemampuan yaitu sebagai *router*. *Wireless Sensor Network* memiliki dua kategori dasar. Kategori dasar yang dimaksud adalah kategori satu dan kategori dua. Pada *Wireless Sensor Network* kategori satu memiliki sistem yang hampir semuanya berbasis topologi *mesh* dengan konektivitas radio *multihop* yang berada di dalam atau di antara *wireless nodes*. Kemudian untuk *routing* menggunakan *routing* dinamis baik di jaringan dengan kabel atau tanpa kabel (*nirkabel*).

Contoh dari *Wireless Sensor Network* kategori satu adalah sistem *military theater*. Sedangkan untuk *Wireless Sensor Network* kategori dua adalah sistem *point-to-point* dengan konektivitas radio *singlehop* atau menggunakan *multipoint-to-point* (topologi star). *Routing* yang digunakan adalah *routing* statis di dalam komunikasi *wireless nodes*. Sistem yang menggunakan kategori dua ini adalah sistem kontrol perumahan (Kazem Sohraby, 2007).

2.2 Treecast

Treecast merupakan pengalaman global dan arsitektur *routing stateless* untuk *sensor network*. Metode *treecast* pada pengalaman di *node* sensor secara dinamis ini memiliki tujuan yang memberikan alamat pada *node-node* sensor dengan bentuk topologi jaringan yang layaknya sebuah pohon atau *tree*. Bentuk *tree* ini mulai dibentuk dari *sink node* dari sebuah jaringan WSN. Ketika *sink node* menjadi *root* dari struktur *tree* yang selanjutnya dibentuk. Yang kemudian *node-node* sensor terhubung secara langsung dengan *sink node* memilih sebuah *n-bit* alamat yang digunakan oleh dirinya sendiri (Fathoni, Fadhil Arief, 2015).

2.3 Reference Broadcast Synchronization

Sinkronisasi waktu yang bisa disebut *Time Synchronization* ini muncul dikarenakan adanya waktu lokal pada beberapa komputer yang berbeda. Hal itu disebabkan ada beberapa hal salah satunya yaitu perbedaan pada *clock rate* dari tiap-tiap *hardware*-nya. Karena faktor inilah banyak peneliti mengusulkan metode untuk mensinkronisasikan waktu tersebut dari *node* pada *Wireless Sensor Network*. Metode yang diusulkan yaitu metode *Reference Broadcast Synchronization*.

RBS ini merupakan *node* untuk mensinkronisasikan waktu dengan resolusi yang diperlukan misalnya aplikasi jaringan sensor nirkabel. RBS ini memungkinkan *node* yang menerima paket sinkronisasi menggunakan waktu paket ini sebagai titik acuan sinkronisasi waktu. Sebab waktu propagasi non-deterministik ini sebagian besar terlibat di dalam transmisi paket melewati saluran nirkabel yang terletak diantara pembangunan paket dan *transceiver* pengirim seperti *delay* pada antrian pengirim, dan RBS menghilangkan sebagian keterlambatan yang terlibat dalam protokol sinkronisasi waktu. Algoritma RBS menggunakan jaringan *single-hop* yang sederhana (Darmawan, Aulia Arif, 2016).

2.4 Mikrokontroler Arduino Nano

Mikrokontroler adalah sebuah *chip* yang mempunyai fungsi sebagai pengontrol rangkaian elektronik dan umumnya dapat menyimpan program didalamnya. Mikrokontroler yang digunakan pada penelitian ini adalah Arduino Nano. Arduino Nano merupakan *board* yang mikrokontrollernya Atmega 328, 14 digital I/O, 6 *input* analog, 16 MHz osilator Kristal, koneksi USB dan tombol *reset*. Arduino Nano menggunakan koneksi USB sebagai penghubung ke komputer tegangan yang didapatkan dari komputer. (Arduino, 2015)



Gambar 2.1 Arduino Nano tampak dari depan
(Sumber : Arduino, 2015)

2.5 Modul NRF24L01

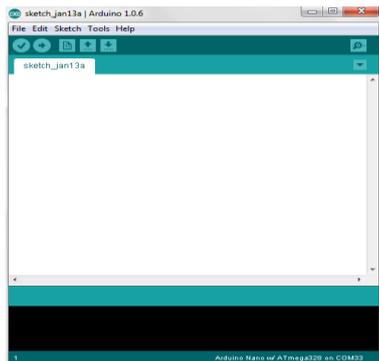
Modul nRF24L01 merupakan sebuah modul komunikasi jarak jauh yang memanfaatkan pita gelombang RF 2.4GHz ISM. Selain itu modul ini menggunakan antarmuka SPI untuk berkomunikasi. Tegangan kerja modul 5V DC. nRF24L01 juga memiliki *true ULP solution*, yang memungkinkan daya tahan baterai berbulan-bulan sampai bertahun-tahun.



Gambar 2.2 Modul nRF24L01
(Sumber : Chantrell, 2013)

2.6 Arduino IDE

Arduino IDE merupakan sebuah software yang dibuat khusus untuk perangkat Arduino itu sendiri. Fungsi dari arduino IDE itu sendiri yaitu untuk menulis program, compile program dan upload program ke memory mikrokontroler. Terdapat serial monitor yang berfungsi sebagai monitoring output serta jalannya suatu program pada perangkat arduino. Berikut tampilan dari Arduino IDE 1.0.6 pada Gambar 2.5. (Arduino, 2015):



Gambar 2.3 Tampilan Arduino IDE 1.0.6 (Sumber : Arduino, 2015)

2.7 Library Mirf

Pada penelitian ini menggunakan *library Mirf*, karena *library mirf* merupakan modul yang digunakan sebagai kebutuhan suatu pemrograman NRF24L01 pada Arduino dan *library mirf* dapat mengakses radio frekuensi NRF24L01. Dan *library Mirf* menyediakan beberapa fungsi *controlling* umum NRF yang dapat langsung dipakai meliputi konfigurasi dan operasional, yang dapat bekerja yang pada ATMEGA168 dan ATMEGA328 (Arduino, 2013).

3. PERANCANGAN DAN IMPLEMENTASI

3.1. Perancangan Sistem

Perancangan sistem ini menjelaskan perangkat yang digunakan serta spesifikasinya dan alur komunikasi sistem hingga sistem dapat berjalan. Dalam perangkat keras tentunya menggunakan beberapa komponen agar perangkat WSN dapat menjalankan sistem ini. Berikut ini diagram blok sensor *node*:



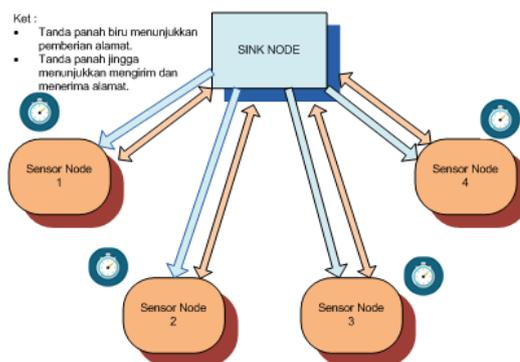
Gambar 3.1 Diagram Blok Perangkat Pada Sensor Node

Pada Gambar 3.1 terdapat modul *transceiver* NRF24L01 merupakan perangkat komunikasi nirkabel fungsinya sebagai komunikasi pada tiap-tiap node dalam sistem ini. Selanjutnya ada mikrokontroler Arduino Nano yang fungsinya memberi perintah serta pengolahan data hasil komunikasi. Terakhir ada sistem monitoring atau personal *computer* fungsinya untuk menampilkan hasil dari

pengolahan data agar memudahkan untuk memantau data.

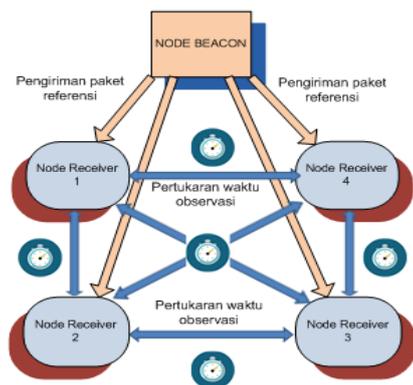
3.1.1 Perancangan Alur Sistem

Sistem berawal dari *sink node* atau *parent* yang merupakan titik pusat pada sensor *network*. Sebuah node dikatakan *parent* (orang tua) jika memiliki alamat jaringan yang sudah dikonfigurasi. Lalu *parent* mengirimkan pesan secara menyeluruh atau *broadcast* kepada *node-node* yang belum memiliki alamat serta waktu yang didapat setelah memiliki alamat yang telah disetujui. Berikut ilustrasi komunikasi sistem *treecast* saat pengiriman alamat dan penerimaan alamat dari *sink node* ke sensor *node* bahkan sebaliknya dari sensor *node* ke *sink node* pada Gambar 3.2



Gambar 3.2 Ilustrasi Komunikasi Sistem Treecast

Jika sistem *Treecast* telah selesai menjalankan prosesnya maka tahap selanjutnya adalah melakukan sinkronisasi waktu ada satu *node beacon* dan ada 4 *node* sebagai *receiver*. *Node beacon* berperan sebagai pemberi perintah untuk melakukan sinkronisasi waktu. Awal bekerjanya sistem ini ialah setiap *node receiver* mempunyai waktu lokal sendiri dan waktu lokal yang dimiliki *node receiver* satu dengan yang lainnya berbeda. Lalu *node beacon* memberi perintah untuk melakukan sinkronisasi dengan mengirim paket referensi, sementara *node receiver* menerima paket referensi yang mencatat waktu datangnya pakat yang sesuai dengan waktu lokal tiap-tiap *node receiver* kondisi ini disebut kondisi waktu observasi saat itu waktu observasi sedang bertukar waktu dari *receiver* satu ke *receiver* lainnya. Maksudnya untuk mengetahui waktu yang diterima paket referensi oleh *node receiver* lainnya, serta dapat mengetahui selisih waktu dan *phase offset* oleh *node receiver* lainnya. Berikut ini adalah ilustrasi komunikasi sistem RBS



Gambar 3.3 Ilustrasi Komunikasi Sistem RBS

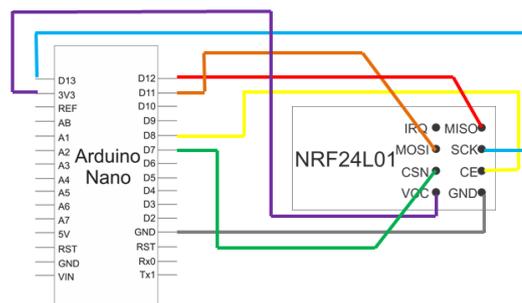
3.2. Perancangan Perangkat Keras

Pada perancangan perangkat keras ini menjelaskan perangkat apa saja yang digunakan pada sistem. Perangkatnya yaitu *sink node* (*node beacon*) dan *sensor node* (*node receiver*). Pada tiap-tiap *node* mempunyai komponen yang sama yaitu berupa mikrokontroller Arduino Nano , modul NRF24L01, dan *Personal Computer* (PC). Dan PC fungsinya sebagai penghubung mikrokontroller dengan NRF24L01, selain itu PC juga sebagai aplikasi monitoring yang menggunakan serial monitor dari Arduino. Berikut Tabel 3.1 yang menjelaskan konfigurasi modul NRF24L01 dan mikrokontroller Arduino Nano yang terhubung melalui pin terdapat pada kedua perangkat tersebut.

Tabel 3.1 Konfigurasi Pin Arduino Nano dan NRF2L01

No.	Pin NRF24L01	Pin Arduino Nano
1	GND	1
2	Vcc / 3.3V	2
3	CSN	7
4	CE	8
5	MOSI	11
6	MISO	12
7	SCK	13

Selanjutnya Arduino Nano disambungkan dengan PC menggunakan kabel USB untuk *upload* program, sumber daya tegangan serta komunikasi data sebagai monitoring. Hubungan perangkat keras ditunjukkan pada Gambar 3.4

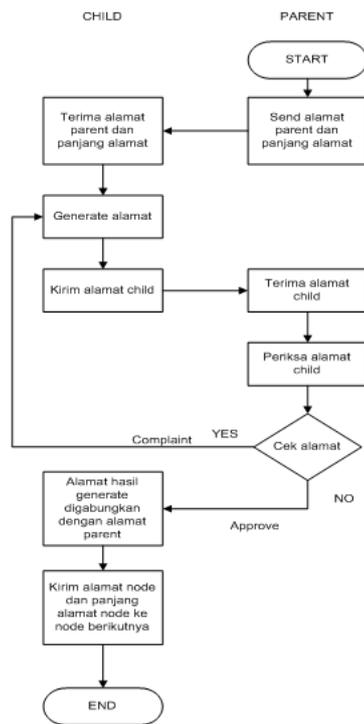


Gambar 3.4 Skematik rangkaian Arduino Nano dan nRF24L01

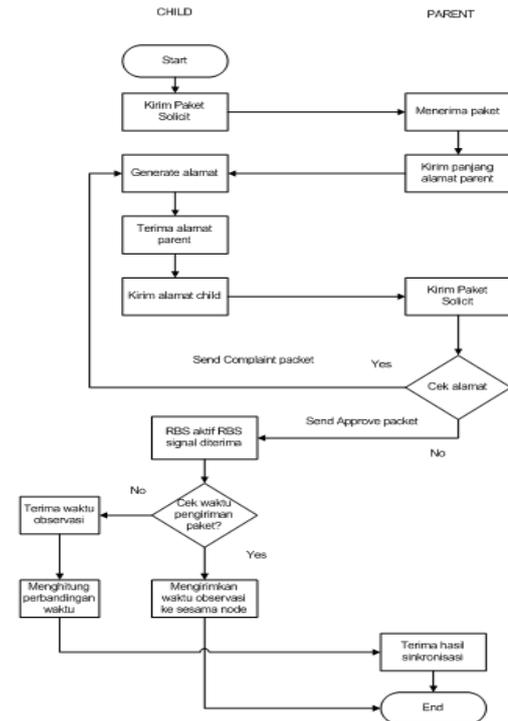
3.3 Perancangan Algoritma

3.3.1 Perancangan Algoritma Sink Node

Sink node bisa menjadi *parent* jika memiliki alamat jaringan yang sudah dikonfigurasi. Kemudian *parent* mengirim pesan secara *broadcast* (menyeluruh) kepada sensor-sensor *node* yang belum mempunyai alamat. Lalu *parent* mengirim pesan yang berisi alamat beserta alamat panjang yang digunakan. *Child* menerima pesan dari *parent*, lalu membaca alamat yang digunakan oleh *parent* dan panjang alamat. Selanjutnya *child* *generate* sebuah alamat secara acak dengan panjang yang sama dengan alamat dari *parent*. Hasil dari alamat *generate* akan dikirimkan ke *parent* untuk diperiksa apakah alamat digunakan apa tidak. Jika alamat sudah digunakan maka *parent* mengirim pesan berupa *complaint*. Apabila alamat belum digunakan maka *parent* mengirim pesan berupa *approve* kepada *child* (*sensor node*). Setelah itu, *child* menggabungkan alamat dari *parent* dengan alamat yang disetujui dari *parent*, maka alamat tersebut digunakan oleh *child*. Berikut diagram alir dari pemberian alamat *sink node* kepada *sensor node*



Gambar 3.5 Diagram Alir Pemberian Alamat Sink Node kepada Sensor Node



Gambar 3.6 Diagram Alir Algoritma Sensor Node

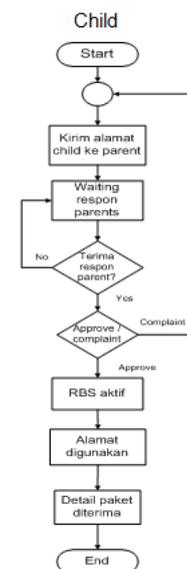
3.3.2 Perancangan Algoritma Sensor Node

Setelah pemberian alamat, tahap selanjutnya yaitu pemberian alamat kepada node baru. Awalnya *child* mengirim paket *solicit* kepada *parent*, lalu *parent* menerima paket tersebut. Diagram alir algoritma sensor node ditunjukkan pada Gambar 3.6.

Pada diagram alir Gambar 3.6, setelah *parent* menerima paket kemudian *parent* mengirim panjang alamat *parent* kepada *child* sehingga menghasilkan *generate* alamat, kemudian menerima alamat *parent* dan mengirimkan alamat *child*. *Parent* menerima alamat *child*, lalu *parent* memeriksa alamat *child* tersebut. Apakah alamat tersebut sudah digunakan atau belum, jika sudah digunakan maka paket mengirimkan *complaint* dan kembali ke *generate* alamat, apabila alamat belum digunakan maka paket mengirimkan *approve* kepada *child*. Selanjutnya RBS aktif dan RBS memberi signal bahwa signal sudah diterima, kemudian memeriksa waktu pengiriman paket jika iya maka hanya mengirimkan waktu observasi ke sesama *node*, apabila tidak maka menerima waktu observasi dan menghitung perbandingan waktu yang akhirnya menerima hasil sinkronisasi.

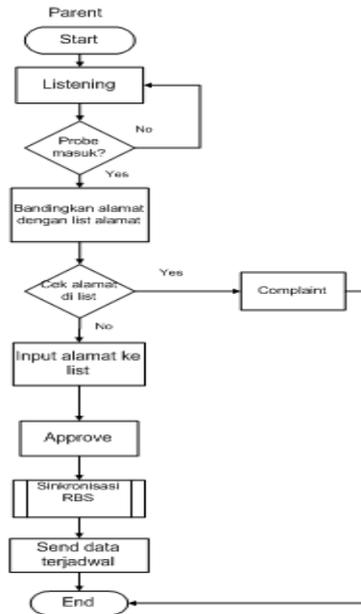
3.3.3 Perancangan Algoritma Duplicate Address Detection

Duplicate Address Detection berfungsi sebagai mendeteksi penggunaan alamat yang sama pada sensor *node*, guna sebagai menjamin jalannya komunikasi agar sampai pada sensor *node* (*child*) yang dituju. Berikut diagram alir pengecekan alamat pada sensor *node* (*child*)



Gambar 3.7 Diagram Alir Pemeriksaan Alamat pada Sensor Node

Sistem *Duplicate address detection* ini bekerja, ketika sebuah sensor *node* (*node child*) sudah melakukan *generate* alamat berdasarkan panjang alamat, *child* mengirim alamat tersebut kepada *parent* agar diperiksa apakah alamat tersebut digunakan atau tidak. Berikut diagram alir pengecekan alamat pada *sink node* (*parent*)

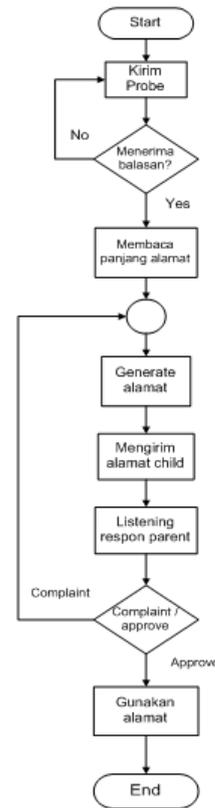


Gambar 3.8 Diagram Alir Pemeriksaan Alamat pada Sink Node

Kemudian *parent* memeriksa alamat yang telah diajukan *child*, *parent* selanjutnya melihat alamat tersebut pada *list*. Jika alamat tersebut sudah digunakan, *parent* mengirimkan paket berupa *complaint* bahwa alamat yang diajukan sudah digunakan oleh *node* lain. Tetapi, apabila alamat yang diajukan *child* tidak ada yang menggunakan, *parent* mengirimkan paket berupa *approve* sebagai pemberitahuan bahwa alamat yang diajukan *child* kepada *parent* telah disetujui, selanjutnya RBS mulai aktif. Lalu *child* bisa menggunakan alamat tersebut dan pengiriman data tersebut telah terjadwal.

3.3.4 Perancangan Algoritma Solicitation

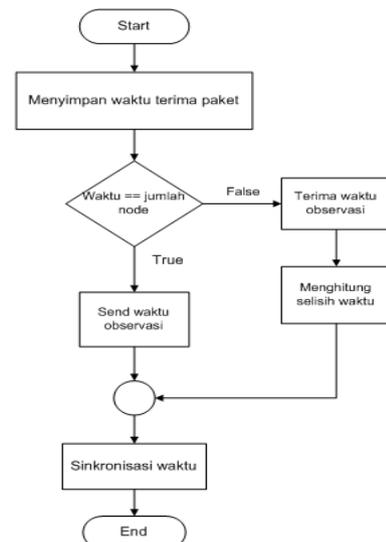
Penjelasan tentang *solicitation* yang ada di sensor *node* (*child*). Ketika proses sedang berjalan kemudian apabila pada waktu pengecekan *parent* dan *parent* tidak menjawab, itu terjadi karena *parent* mengalami masalah seperti kerusakan atau mati. Selanjutnya *child* mengirim paket *SOLICIT* pada tiap-tiap *node* lainnya, lalu *node-node* yang ada disekitarnya memberi balasan paket tersebut dengan mengirim alamat beserta panjang alamat yang digunakan. Berikut ini diagram alir *solicitation*



Gambar 3.9 Diagram Alir Solicitation

3.3.4 Perancangan Algoritma Reference Broadcast Synchronization

RBS (*Reference Broadcast Synchronization*) awal bekerjanya setelah sensor *node* (*receiver*) menerima paket referensi, lalu sensor *node* (*receiver*) menyimpan waktu yang diterima paket, dan dijadikan waktu observasi. Selanjutnya mengirim paket waktu observasi pada tiap sensor *node* (*receiver*). Berikut diagram alir sinkronisasi RBS



Gambar 3.10 Diagram Alir Sinkronisasi RBS

4. PENGUJIAN

4.1 Pengujian Pemberian Alamat pada Sensor Node

4.1.1 Hasil Pengujian

Hasilnya berupa pemberian alamat kepada sensor *node* dari *sink node*, dimana alamat tersebut sudah didapatkan oleh sensor *node* serta waktu untuk mendapatkan alamat tersebut. Hasil pengujian pemberian alamat pada sensor *node* dengan menggunakan 2 node dan mendapatkan alamat 1110 yang didapat dari alamat sink node dan alamat hasil generate dengan waktu 1019 *milisecond*. Berikut penjelasan dari hasil pengujian pemberian alamat tersebut ini dapat dilihat pada Tabel 4.1

Tabel 4.1 Hasil Pengujian Pemberian Alamat pada Sensor Node

No. Percobaan	Alamat yang didapat (Binary)	Waktu menjadi parent (ms)
1	1110	1019
2	1100	1020
3	1100	1018
4	1100	1030
5	1101	1018
6	1110	1018
7	1101	1018
8	1110	1018
9	1101	1018
10	1111	1030
Rata - rata		1020.7

4.2 Pengujian Pemberian Alamat pada Node Baru

4.2.1 Hasil Pengujian

Hasilnya berupa pemberian alamat kepada sensor *node* baru dari *sink node*, dimana alamat tersebut sudah didapatkan oleh sensor *node* baru serta waktu untuk mendapatkan alamat tersebut. Dengan menggunakan 4 node hasil dari pengujian pemberian alamat kepada sensor *node* yang baru bergabung dalam sensor *network*, lalu sensor *node* baru juga memiliki alamat 1100 dengan waktu 1018 *milisecond*. Berikut hasil pengujian dapat dilihat pada Tabel 4.2

Tabel 4.2 Hasil Pengujian Alamat kepada Sensor Node Baru

No. Percobaan	Alamat yang didapat (Binary)	Waktu menjadi parent (ms)
1	1111	3032
2	1101	1018
3	1100	1018
4	1111	1018
5	1110	1077
6	1100	1041
7	1111	1018
8	1101	1077
9	1100	1054
10	1100	1054
Rata - rata		1240.7

4.3 Pengujian Pengiriman Paket Referensi

4.3.1 Hasil Pengujian

Pengujian	Banyaknya Reference	Node ke			
		I	II	III	IV
Pengujian 1	1	1/1	1/2	1/3	1/4
	2	2/1	2/2	2/3	2/4
	3	3/1	3/2	3/3	3/4
	4	4/1	4/2	4/3	4/4
	5	5/1	5/2	5/3	5/4
Pengujian 1	1	1/1	1/2	1/3	1/4
	2	2/1	2/2	2/3	2/4
	3	3/1	3/2	3/3	3/4
	4	4/1	4/2	4/3	4/4
	5	5/1	5/2	5/3	5/4
Pengujian 1	1	1/1	1/2	1/3	1/4
	2	2/1	2/2	2/3	2/4
	3	3/1	3/2	3/3	3/4
	4	4/1	4/2	4/3	4/4
	5	5/1	5/2	5/3	5/4

Hasil dari keseluruhan dalam pengujian dilakukan 5 kali dan paket referensi yang telah ditentukan sebelumnya berjumlah 5 paket referensi. Terdapat data yang tercatat setiap kolom *node* dengan format data yang dikirim maupun data yang diterima misalnya data 5/1, 5/2, 5/3, dan 5/4 artinya jumlah paket yang diterima berjumlah 5 dari paket yang dikirim dan dengan menggunakan 4 *node*, maka pada setiap *node* tersebut menerima paket referensi sejumlah 5 paket. Berikut Tabel 4.3 pengiriman paket referensi

Tabel 4.3 Hasil Percobaan Pengiriman Paket Referensi

Pengujian 4	1	1/1	1/2	1/3	1/4
	2	2/1	2/2	2/3	2/4
	3	3/1	3/2	3/3	3/4
	4	4/1	4/2	4/3	4/4
	5	5/1	5/2	5/3	5/4
Pengujian 5	1	1/1	1/2	1/3	1/4
	2	2/1	2/2	2/3	2/4
	3	3/1	3/2	3/3	3/4
	4	4/1	4/2	4/3	4/4
	5	5/1	5/2	5/3	5/4

4.4 Pengujian Waktu Observasi

4.4.1 Hasil Pengujian

Berikut ini hasil dari keseluruhan pengujian dapat dilihat pada Tabel 4.4 tertuliskan bahwa pengujian dilakukan dalam 10 kali pengujian dengan paket referensi yang telah ditentukan sebelumnya. Pengiriman paket dilakukan secara broadcast, sehingga diharapkan paket mampu menerima secara bersamaan pada tiap node receiver sebagai penerima. Berikut ini terdapat Tabel 4.4 yang memberikan hasil pengiriman paket dan paket yang bertanda ‘ • ‘ maka bertanda pengiriman berhasil jika terdapat tanda ‘ × ‘ maka pengiriman paket mengalami kegagalan.

Tabel 4.4 Hasil Pengujian Waktu Observasi

No.	Pengirim	1			2			3			4		
		2	3	4	1	3	4	1	2	4	1	2	3
1		•	•	•	•	•	•	•	•	•	•	•	•
2		•	•	•	•	•	•	•	•	•	•	•	•
3		•	•	•	•	•	•	•	•	•	•	•	•
4		•	•	•	•	•	•	•	•	•	•	•	•
5		•	•	•	•	•	•	•	•	•	•	•	•
6		•	•	•	•	•	•	•	•	•	•	•	•
7		•	•	•	•	•	•	•	•	•	•	•	•
8		•	•	•	•	•	•	•	•	•	•	•	•
9		•	•	•	•	•	•	•	•	•	•	•	•
10		•	•	•	•	•	•	•	•	•	•	•	•

4.5 Pengujian Sinkronisasi Waktu

4.5.1 Hasil Pengujian

Berikut ini merupakan hasil dari sinkronisasi waktu dengan 4 node dan 5 paket referensi. Hasil selisih waktu yang didapat dari sinkronisasi dengan menggunakan metode *Reference Broadcast Synchronization*. Data secara keseluruhan dirangkum pada Tabel 4.5

Tabel 4.5 Hasil Pengujian Sinkronisasi Waktu

Pengujian		Paket Referensi
		5
Error (microsecond)	Pengujian 1	1791
	Pengujian 2	1298
	Pengujian 3	310
	Pengujian 4	1383
	Pengujian 5	2193
	Pengujian 6	1752
	Pengujian 7	392
	Pengujian 8	1650
	Pengujian 9	1542
	Pengujian 10	270
Rata - rata		1258.1

Dalam pengujian sinkronisasi waktu diperlukan perhitungan waktu sinkronisasi pada tiap *node receiver*. Berikut ini adalah Tabel 4.6 perhitungan sinkronisasi waktu pada *node receiver*

Tabel 4.6 Perhitungan Sinkronisasi Waktu pada Node Receiver

No Reference Packet	Waktu Penerimaan Data Node 1	Waktu Penerimaan Data Node 2
1	4035228	34240160
2	4040412	34245428
3	4050612	34255712
4	4060984	34266168
5	4071268	34276536
Total waktu ($\sum_{k=1}^m$)	20258504	171284004
Waktu Observasi ($\frac{1}{m} \sum_{k=1}^m$)	4051700	34256800

Berikut ini adalah Tabel 4.7 hasil rata-rata pengujian sinkronisasi waktu

Tabel 4.7 Rata-Rata Pengujian Sinkronisasi Waktu

Paket Referensi	4 Node
5	1258.1
Rata - rata	251.62

Dari hasil rata-rata yang diperoleh saat pengujian peneliti menganalisa jika menggunakan 4 node tidak menambah terjadinya *error* waktu sinkronisasi. Hal ini disebabkan karena ketika menggunakan 4 node dan menambahkan jumlah paket referensi lebih banyak maka tingkat *error* waktu sinkronisasi

sangat kecil.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

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

1. Sistem *Treecast* dan RBS menggunakan 2 komponen utama yaitu *sink node* dan sensor *node* yang terdapat pada sistem *Treecast* sedangkan pada sistem RBS terdapat *node beacon* dan *node receiver*. Masing-masing dari komponen tersebut terdiri dari Mikrokontroler Arduino Nano dan modul transceiver nRF24L01, cukup mampu untuk melakukan proses dengan baik dalam hal pengiriman serta penerimaan suatu data berupa alamat.
2. Dari hasil pengalamatan dinamis *Treecast*, metode ini mampu diimplementasikan secara nyata dan mampu memberikan alamat serta waktu untuk mendapatkan alamat pada tiap sensor *node* dalam sebuah jaringan sensor *network*. Sinkronisasi waktu yang menggunakan *Reference Broadcast Synchronization* ini memakai 4 node receiver dan 5 paket referensi yang mampu mencapai ketelitian sinkronisasi waktu sampai satuan waktu *millisecond*. Sistem *Treecast* dan RBS menggunakan topologi *star* meskipun menggunakan topologi *star* dan menggunakan level 1, ternyata kedua sistem tersebut mampu menjalankan tugasnya masing-masing dengan baik.

5.2 Saran

Berikut ini beberapa saran yang diharapkan dalam pengembangan sistem ini antara lain:

1. Disarankan untuk menggunakan Arduino yang spesifikasinya yang lebih tinggi.
2. Untuk penelitian selanjutnya metode dapat dikembangkan dengan menggunakan metode TDMA.
3. Penelitian selanjutnya diharapkan untuk pengguna *node* bisa lebih banyak.
4. Dan penelitian selanjutnya diharapkan menggunakan paket referensi yang lebih banyak.

DAFTAR PUSTAKA

- Arduino, 2015. Arduino Nano. [Online] Tersedia di : <<http://arduino.cc/en/Main/ArduinoBoardUno>> [Diakses 27 Mei 2017]
- Arduino, 2015. Product Arduino. [online] Tersedia di: <<http://www.arduino.cc/en/Main/Products>> [Diakses 27 Mei 2017]
- 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 26 Mei 2017].
- Darmawan, Aulia Arif, 2016. "Implementasi Metode *REFERENCE BROADCAST TIME SYNCHRONIZATION* dan *TIME DIVISION MULTIPLE ACCESS* pada *WIRELESS SENSOR NETWORK*".
- Fathoni, Fadhil Arief, 2015. "Implementasi Alokasi Pengalamatan Dinamis pada *WIRELESS SENSOR NETWORK*".
- Kazem Sohraby, D. M. T. Z., 2007. *WIRELESS SENSOR NETWORKS: Technology, Protocols, and Applications*. Hoboken, New Jersey: John Wiley & Sons, Inc.
- "Preliminary Product Specification nRF24L01," Nordic Semiconductor, Tonstad, 2006.
- S. PalChaudhuri, S. Du, A. K. Saha and D. B. Johnson, "TreeCast A Stateless Addressing and Routing Architecture for Sensor Network," *International Parallel and Distributed Processing Symposium*, pp. 1-8, 2004.
- Sivrikaya, Fikret, and Bulent Yener. "Time synchronization in sensor networks: a survey." *Network*, IEEE 18.4 (2004): 45-50. Tersedia di : <https://www.cs.rpi.edu/~yener/PAPERS/WINET/timesync04.pdf>
- Wiyanto Eddy. "Analisis Teknik Pengalamatan pada *WIRELESS SENSOR NETWORKS*", 2012.