Implementasi Time Synchronization Pada WSN Untuk Metode TDMA Menggunakan Algoritma TPSN

Ardy Novian Erwanda, Sabriansyah Rizqika Akbar, S.T., M.Eng., Aswin Suharsono, S.T., MT.
Jurusan Sistem Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya (UB)
Jl. Veteran No 8, Malang 65145, Indonesia
e-mail: ardy.erwanda@gmail.com, sabrian.akbar@gmail.com,

Abstrack - WSN telah menjadi peralatan pengiriman data yang populer saat ini, dikarenakan sifatnya yang lebih mudah untuk diimplementasikan dan praktis serta tidak banyak memakan biaya jika dibandingkan jaringan kabel biasa. Namun, WSN juga memiliki kekurangan, yaitu permasalahan interferensi pada pengiriman data. Salah satu metode untuk menyelesaikannya adalah menggunakan metode TDMA. TDMA bekerja dengan membuat banyak node untuk dapat mengirimkan data secara bergantian, demi menghindari adanya interferensi pada pengiriman data. Untuk mengaplikasikan metode TDMA, dibutuhkan metode sinkronisasi waktu untuk menyamakan waktu lokal setiap node dalam WSN. Dalam jaringan berkabel, proses sinkronisasi waktu tidak menemui kendala yang signifikan, namun dalam WSN, akan mendapati beberapa kendala seperti delay propagasi yang cukup besar. Dari masalah yang ada, maka diterapkan salah satu metode sinkronisasi waktu dalam WSN yaitu TPSN, supaya metode TDMA bisa dijalankan dengan baik. Perancangan sistem akan menggunakan Arduino Nano, NRF24L01, dan laptop. Skenario pengujian berdasarkan pada waktu yang dibutuhkan pada masing-masing tahapan implementasi. Terdapat 3 tahapan yang diamati pada pengujian, yaitu discovery, sinkronisasi, dan TDMA. Dari ketiga tahapan tersebut, didapatkan hasil yang menunjukkan bahwa pembentukan hierarki sampai level 2 membutuhkan waktu kurang dari 10 detik. Dengan rata-rata level 1 kurang dari 2 detik, dan level 2 kurang dari 9 detik. Algoritma TPSN yang diimplementasikan sudah berhasil membuat node WSN tersinkronisasi dan pengiriman data berjalan sesuai *time slot* yang telah dibuat. Apabila terjadi perubahan struktur hierarki tidak akan mempengaruhi keberhasilan algoritma TPSN dan pengiriman data pada pengujian TDMA, selama persediaan time slot masih ada. Keakuratan waktu penerimaan data dengan jadwal TDMA memiliki selisih maksimal 90 milidetik.

Kata Kunci— *Time Synchronization*, TPSN, TDMA, *node* WSN, interferensi, *time slot*

I. PENDAHULUAN

Wireless sensor network (WSN) telah menjadi perhatian hampir di seluruh dunia dalam beberapa tahun terakhir ini. Dimana banyak sekali penelitian yang dilakukan terkait dengan perkembangan WSN ini dari berbagai aspek, dikarenakan banyaknya keterbatasan pada jaringan nirkabel jika dibandingkan dengan jaringan kabel. Salah satu bentuk pengembangan teknologi ini adalah dari sisi protokol akses, misalnya *Time Division Multiple Access* atau TDMA yang membutuhkan metode sinkronisasi waktu atau disebut juga dengan *Time Synchronization*.

Dengan protokol TDMA, memungkinkan bagi modul perangkat WSN untuk meningkatkan jumlah data yang bisa ditransmisikan secara nirkabel dalam spektrum frekuensi yang sama (Sohraby & Minoli & Znati, 2007). TDMA akan membuat banyak node untuk dapat mengrimkan data secara bergantian, demi menghindari adanya tubrukan atau interferensi data. Untuk bisa menggunakan TDMA, terlebih dahulu perlu digunakan metode *Time Synchronization* guna menyamakan *clock* waktu dari semua perangkat WSN yang diletakkan secara terpisah satu sama lain.

Time Synchronization adalah kunci dari berbagai aplikasi dan sistem operasi yang digunakan pada sistem komputasi terdistribusi. Beberapa protokol telah dikembangkan dan digunakan untuk

Time Synchronization jaringan kabel maupun nirkabel. Dengan banyak keterbatasan, jaringan nirkabel –atau dalam hal ini WSN- membutuhkan lebih banyak perhatian dan pengembangan daripada jaringan kabel tradisional. Berdasarkan berbagai aplikasi yang diterapkan, presisi waktu WSN yang ditingkatkan hanya beberapa milidetik, dapat meningkatkan performa aplikasi tersebut secara signifikan (Simon et al, 2004)

Namun, beberapa modul perangkat WSN seperti NRF24L01, tidak memiliki fitur otomatis dalam hal *Time Synchronization* tersebut. Untuk itu, perlu dilakukan pembuatan secara manual kode program yang bisa mengatur mekanisme *Time Synchronization* dalam TDMA WSN seperti dijelaskan di atas. Beberapa algoritma berhasil ditemukan dan diuji coba pada peralatan WSN.

Berbagai penelitian tentang *Time Synchronization* WSN telah banyak dipublikasikan. Diantaranya adalah penelitian berjudul *Post Facto Synchronization* yang dilakukan oleh Elson dan Estrin, pada pendekatan tersebut, setiap *clock node* secara alami tidak tersinkronkan, sebuah *node beacon* secara periodik melakukan *broadcast* pesan kepada *node-node* sensor yang berada dalam jangkauannya. Ketika sebuah even terdeteksi, tiap *node* melakukan *record* terhadap waktu even tersebut, membandingkannya dengan waktu lokal masing-masing. Sebuah algoritma bernama *Reference Broadcast Synchronization* atau RBS, menjadi hasil bagi penelitian ini (Elson, 2002). Algoritma tersebut dijalankan dengan model topologi jaringan *star*, dimana *node beacon* (kemudian dinamakan *node root*) adalah pusat dari topologi ini. Dalam satuan waktu tertentu, *node beacon* akan melakukan pengiriman waktu even secara periodik. Kemudian setelah semua *node* menerima, untuk mengurangi jumlah *delay* –yang bisa mengurangi tingkat presisinya– , semua *node* penerima akan saling mencocokkan waktu even satu sama lain. Algoritma akan ini memiliki kelemahan signifikan, jika dilakukan pada jaringan WSN skala besar, karena *node beacon* tidak akan bisa menjangkau semua *node*. Beberapa penelitian terkait yang dilakukan oleh Yoon et al (2010) menggunakan algoritmanya bernama Tiny-Sync, mencoba memperbaiki sistem perhitungan waktu RBS lebih akurat. Namun, tidak membahas mengenai pengembangan dalam skala yang lebih besar.

Berdasarkan pemaparan di atas, penulis berminat untuk melanjutkan penelitian dengan menggunakan algoritma TPSN yang menggunakan topologi *tree*, sehingga bisa diterapkan dalam skala yang lebih besar. Penelitian dilakukan dengan modul WSN NRF24L01 yang akan diprogram untuk implementasi algoritma tersebut.

II. LANDASAN KEPUSTAKAAN

A. WSN

WSN singkatan dari *Wireless sensor network* (jaringan sensor nirkabel) adalah suatu jaringan nirkabel yang terdiri dari kumpulan *node* sensor yang tersebar di suatu area tertentu (sensor field); serta sebuah *base station*, yang merupakan komponen penerima dan pengumpul semua informasi dari semua *node* untuk kemudian diolah menjadi informasi dan keperluan lainnya. Tiap *node* sensor memiliki kemampuan untuk mengumpulkan data lingkungan, mengolahnya menjadi data digital, dan berkomunikasi dengan *node* sensor lainnya menggunakan protokol tertentu.

Komponen suatu *node* pada WSN ini meliputi sensor, modul *wireless* atau *transceiver*, sumber daya, mikrokontroler, serta memori. Seluruh komponen yang dibutuhkan akan membentuk suatu jaringan yang dimana membentuk suatu fungsi sistem monitoring yang mampu bekerjasama mengumpulkan data yang didapat dari lapangan berupa karakteristik dari sensor yang digunakan dengan menggunakan media *wireless*. Dalam hal ini, karena WSN dapat digunakan untuk berbagai aplikasi maka penggunaan sensor dapat dipilih sesuai kebutuhan sistem.

Sistem WSN ini lebih jauh efisien dibandingkan dengan penggunaan kabel. Sistem ini memiliki fungsi untuk berbagai jenis aplikasi yang dimana WSN mampu memenuhi kebutuhan teknologi dalam berbagai bidang ilmu, seperti halnya pada bidang biologi, pertanian, perikanan dan lain sebagainya. (Kazem S,2007).

B. TDMA

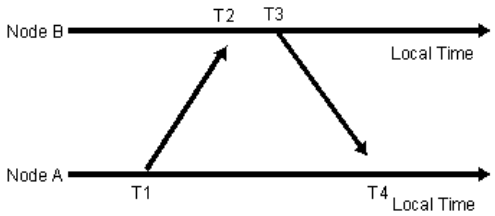
TDMA atau *Time Division Multiple Access* adalah teknologi untuk peralatan

pengiriman data digital menggunakan teknik multipleksi pembagian waktu. TDMA bekerja dengan membagi frekuensi radio ke dalam slot waktu yang dapat mendukung pemakaian aliran kanal data secara bersama-sama. Transmisi dalam bentuk urutan *frame*, dimana tiap frame dibagi menjadi beberapa slot waktu, dan tiap slot waktu bersifat *dedicated* untuk sebuah transmitter tertentu. (Stallings, 2005). Setiap pengirim akan memiliki jadwal pengiriman yang berbeda dengan jadwal pengirim yang lain.

C. TPSN

TPSN (*Time-Sync Protocol for Sensor Networks*) adalah sebuah algoritma sinkronisasi waktu pada WSN, yang menggunakan pemodelan topologi *tree*. Dimana terdapat sebuah *node* sebagai *root* yang berada di pangkal *tree*, sinkronisasi waktu dimulai dari *root* tersebut. Dengan menggunakan aliran data yang disebut *timestamp* yang dikirim dari *root* ke hierarki dibawahnya, waktu disamakan melalui *timestamp* tersebut. TPSN memiliki keunggulan dibandingkan beberapa algoritma lain seperti skala yang lebih besar daripada algoritma RBS (Alson et al, 2001).

Terdapat dua fase pada algoritma TPSN, fase pertama adalah *level discovery*, fase ini digunakan saat pertama kali seluruh *node* memilih sebuah *node* untuk dijadikan sebagai *node root*. Berlanjut ke node level 1, level 2, dan seterusnya. Seluruh node akan mengingat posisinya masing-masing. Sistem ini dilakukan dengan pertama kali aliran data "*level_discovery*" dari *node root* diteruskan sampai ke level paling ujung, sehingga semua node masuk ke daftar hierarki *tree*. Fase kedua adalah fase sinkronisasi, di fase ini, dimulailah sinkronisasi waktu. Semua dimulai dari *node root* terlebih dahulu. *Node level* 1 akan mengirim *synchronization_pulse* pada waktu T1 ke *node root* (level 0), jika root menerima pada waktu T2, akan membalas dengan *acknowledgment_packet* pada waktu T3, yang berisi nomor hierarki, waktu milik *Node Root*, T1, T2, dan T3. *Node level* 1 akan melakukan perhitungan ketika menerima balasan *acknowledgment_packet* pada T4 dari aliran skenario ini dan melakukan pencocokan waktu dan menghitung delay propagasi seperti diilustrasikan pada Gambar 2.1. Perhitungan delay seperti ditunjukkan pada Persamaan 2.1. Perhitungan delay akan ditambahkan pada waktu lokal. *Timestamp* T1,T2,T3, dan T4Hal serupa akan dilakukan oleh *node level* 2 kepada *node level* 1, begitu seterusnya sampai *node level* i yang paling ujung melakukan hal yang sama kepada *node level* i-1.

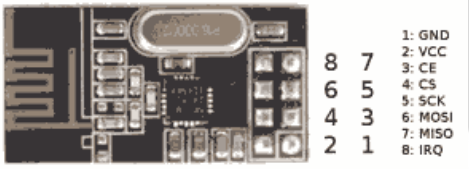


Gambar 2. 1 Mekanisme TPSN

$$d = \frac{(T2 - T1) + (T4 - T3)}{2} \quad \textbf{(2.1)}$$

D. NRF24L01

NRF24L01 adalah sebuah modul komunikasi jarak jauh yang memanfaatkan pita gelombang rf 2.4GHz ISM (*Industrial Scientific and Medical*). Modul ini menggunakan antarmuka SPI untuk berkomunikasi. Tegangan dari modul ini adalah 3,3 Volt DC.



Gambar 2.2 Modul *wireless* NRF24L01

E. Arduino Nano

Arduino Nano merupakan board mikrokontroler milik Arduino untuk digunakan dalam pemrograman pada penelitian ini. Arduino Nano cocok untuk digunakan sebagai *node* pada modul WSN, karena memiliki beberapa keunggulan, Nano didesain untuk pengguna yang membutuhkan fleksibilitas, harga terjangkau, dan ukuran yang kecil (Arduino, 2015). Dalam ukurannya yang kecil, tersedia pin 5 Volt dan 3,3 Volt sehingga bisa digunakan untuk menghidupkan peralatan-peralatan yang membutuhkan tegangan pada kedua nilai tersebut. Tampilan Arduino Nano seperti ditunjukkan gambar 2.2

Gambar 2.3 Arduino Nano

F. Arduino IDE

Arduino IDE adalah Software resmi Arduino yang digunakan untuk pemrograman semua Mikrokontroler Arduino. Dalam penelitian ini, penulis menggunakan Arduino IDE untuk membuat program dalam bahasa C dan *debugging*-nya, meng-*upload* program ke board Arduino, menambahkan *library*, dan mengamati aktivitas Arduino Nano menggunakan Serial Monitor.

III. METODOLOGI

A. Analisis Kebutuhan

Dari studi pustaka yang telah dilakukan, maka penulis kemudian akan melakukan analisis kebutuhan mengenai fungsional dari sistem ini.

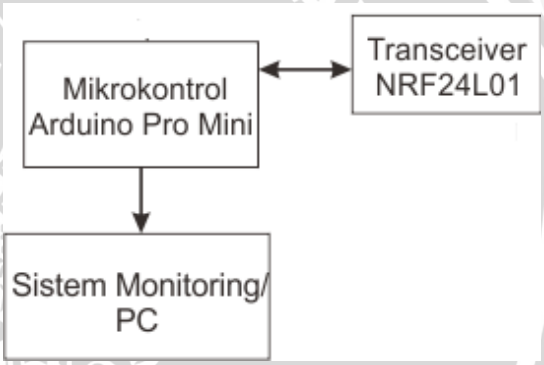
- Kebutuhan fungsional
  a. Sistem memiliki mekanisme waktu yang terstruktur secara keseluruhan
  b. Sistem dapat melakukan pengiriman data secara terjadwal sesuai mekanisme TDMA yang diterapkan
- Kebutuhan non-fungsional

  Untuk mengetahui spesifikasi kebutuhan non-fungsional sistem, dilakukan analisis yang melibatkan perangkat keras yang digunakan.
  a. 4 buah mikrokontroller Arduino Nano sebagai unit pemroses data
  b. 4 buah modul NRF24L01 sebagai perangkat *wireless*
  c. 1 buah komputer sebagai monitoring keberhasilan pengiriman data

B. Perancangan

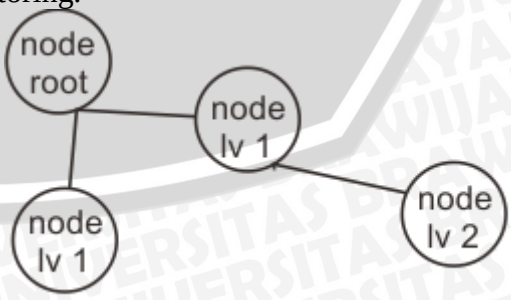
Perancangan sistem adalah tahap mulai merancang suatu sistem yang mampu memenuhi semua kebutuhan fungsional dalam penelitian ini. Perancangan ini terdiri dari perancangan perangkat keras, yaitu tentang pembuatan skematik dan pemasangan secara *embedded node* WSN, meliputi mikrokontroler Arduino dan modul NRF24L01. Kemudian pemasangan perangkat lunak Arduino beserta kebutuhan *driver*. Dan peracangan *Flowchart* pemrograman TPSN sistem.

Perancangan perangkat keras untuk tiap *node* seperti ditunjukkan pada gambar 3.1, terdapat mikrokontroller yang akan bertindak sebagai pemroses data, pada mikrokontroller ini pula akan diisikan program untuk algoritma TPSN dan TDMA. Mikrokontroler ini menggunakan sumber tegangan melalui USB yang tersambung dengan laptop, yang sekaligus menjadi media pengiriman data serial Arduino, kemudian tersambung ke *transceiver* NRF24L01 guna aliran data *wireless*. Disambungkan pula *output led* sebagai indikator.



Gambar 3. 1 Diagram blok *node* WSN

Pada gambar 3.2, ditunjukkan sebuah topologi *tree* yang menjadi dasar dari mekanisme algoritma TPSN, terdapat sebuah *node root* di pangkalnya, yang kemudian tersambung ke *node-node* lain membentuk sebuah topologi, yang di salah satu ujungnya akan disambungkan ke sistem monitoring.



Gambar 3. 2 Topologi *tree* dalam TPSN

TPSN bekerja melalui dua fase. Fase pertama adalah fase *discovery*, yaitu fase untuk *node root* menentukan struktur topologi *tree* yang

akan dibentuk, dan menentukan *node-node* yang mengisi tiap *level* topologi. Sedangkan fase kedua adalah fase *synchronization*, yaitu fase untuk mencocokkan waktu semua *node* dengan *node root*.

Fase pertama diawali ketika *root* (sebagai *level* 0) mengirim paket *discovery* kepada *node* terdekatnya. *Node* yang bisa menerima akan menjadi anggota *tree* pada level 1, kemudian *node level* 1 akan mengirim paket yang serupa untuk mencari anggota *level* berikutnya, sampai semua *node* tergabung dalam topologi.

Fase kedua, dimulailah sinkronisasi waktu. Sama seperti fase pertama, sinkronisasi juga dimulai dari *node root*. Sebuah *node* akan mengirim paket *synchronization* dengan waktu T1, diterima oleh *node* sebelumnya. *Node* sebelumnya tersebut akan membalas dengan rincian T2, waktu penerimaan paket; dan T3, waktu pembalasan. *Node* pengirim awal akan menerima balasan pada T4, kemudian akan melakukan perhitungan waktu, dan mencocokkan waktu

IV. IMPLEMENTASI

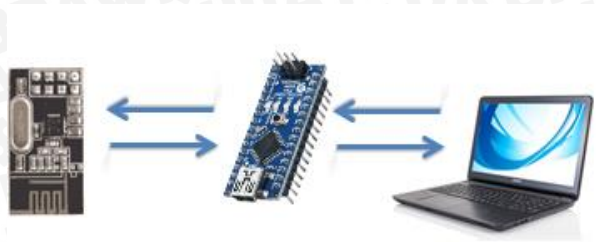
A. Perancangan Sistem

Perancangan system ini meliputi perancangan perangkat keras dan perangkat lunak. Perancangan perangkat keras system akan membahas perangkat yang digunakan dan hubungannya. Perancangan perangkat lunak akan membahas program bahasa C++ yang dijalankan dalam IDE yang kemudian diupload pada perangkat keras Arduino.

B. Perancangan Perangkat Keras

Perancangan perangkat keras ini akan menjelaskan tentang perangkat yang digunakan dalam system. Perangkat keras yang digunakan adalah Arduino Nano, NRF24L01, Laptop. Arduino Nano sebagai mikrokontroler akan tersambung dengan NRF24L01.

Kemudian Arduino Nano akan disambungkan dengan Laptop menggunakan kabel USB untuk upload program, sumber daya tegangan, dan komunikasi data untuk monitoring. Secara keseluruhan hubungan perangkat keras ditunjukkan seperti gambar 5.1



Gambar 5. 1 Hubungan perangkat keras

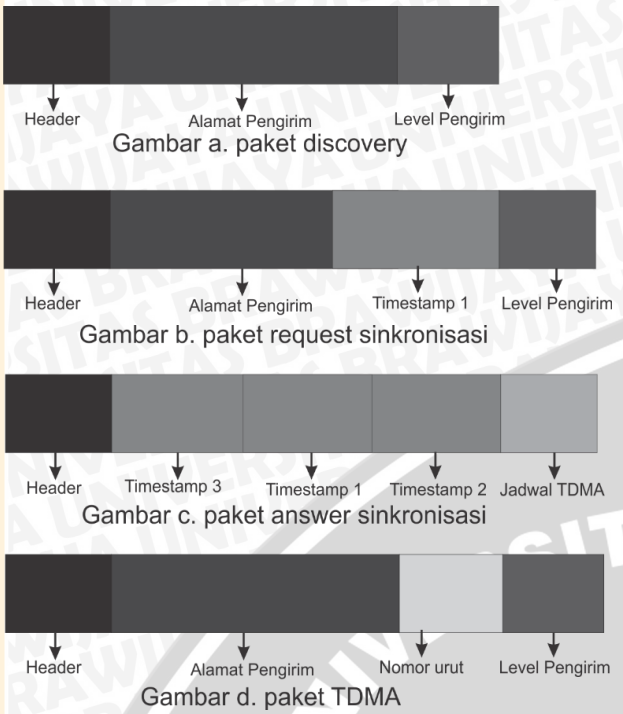
Komunikasi data nirkabel yang dilewatkan antar NRF24L01 akan diproses dan diteruskan oleh Arduino Nano untuk kemudian diteruskan ke PC sebagai sistem monitoring.

C. Perancangan Perangkat Lunak

Perancangan perangkat lunak pada system berisi kode program yang di-*compile* dan diupload menggunakan Arduino IDE dari laptop ke Arduino. Untuk menambahkan fitur NRF24L01, maka pada program akan ditambahkan library Mirf yang akan mengatur komunikasi data seperti pengiriman dan penerimaan data, pengalamatan, ukuran *payload*, dan kanal frekuensi yang digunakan.

Program ini terdiri dari dua jenis, program root dan program node. Program root akan membuat node yang menjalankannya, melakukan tahapan discovery dan sinkronisasi. Sedangkan program node akan dikenali oleh root atau *parent*-nya dan tersinkronisasi waktunya, baru kemudian akan menjalankan tahapan discovery dan sinkronisasi lagi terhadap *child*-nya.

Untuk komunikasi antar node pada masing-masing tahapan atau level, digunakan beberapa paket berbeda, tiap-tiap paket dikenali melalui “header” yang merupakan data array urutan pertama dari paket tersebut. Paket Discovery berawalan karakter “B”; paket Sinkronisasi berawalan karakter “R” untuk request dan “S” untuk balasan; dan paket TDMA berawalan “T”. Sehingga sebuah node penerima akan melakukan pegolahan berdasarkan paket yang sesuai. Secara keseluruhan, keempat paket ditunjukkan oleh gambar 5.2 a – d.



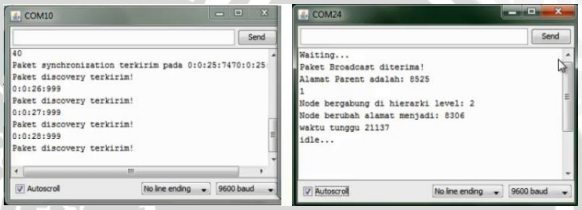
Gambar 5.2 Macam-macam Paket

V. PENGUJIAN SISTEM

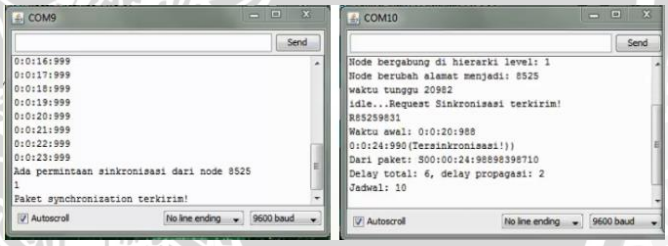
A. Pengujian Fungsional Sistem

Pengujian fungsional digunakan untuk mengamati waktu selama proses berlangsung dari ketiga tahapan: *discovery*, sinkronisasi, dan TDMA. Untuk melakukan pengujian fungsional, maka diperlukan skenario sebagai berikut:

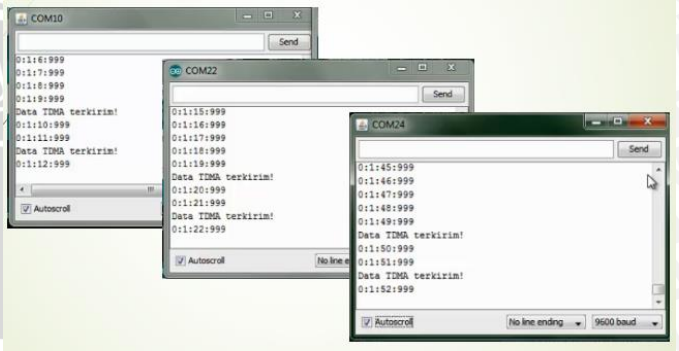
1. *Node Root* (level 0) diaktifkan, kemudian langsung disusul *node* calon penghuni *level* 1. *Node Root* akan melakukan *broadcast* Paket Discovery selama 6 kali pada 6 detik awal.
2. Calon *node* level 1 menerima Paket *Discovery* dan menjadi *child* dari *Node Root* dan menjadi node level 1.
3. Setelah *Node Root* berhenti melakukan *broadcast* Paket *Discovery*, maka node calon node level 2 baru diaktifkan supaya tidak masuk ke level 1.
4. *Node* level 1 memasuki kondisi *idle*, menunggu selama waktu acak yang dihasilkan sendiri.
5. *Node* level 1, setelah terjadi *timeout* pada waktu *idle*-nya, akan langsung melakukan *request* sinkronisasi ke Root. Root membalas, kemudian *Node level* 1 akan mencocokkan waktu, dan mencatat jadwal TDMA yang juga disertakan bersama balasan dari *Node Root*.
6. Setelah tersinkronisasi dengan *Root*, maka *node level* 1 melakukan *broadcast* paket *Discovery* untuk menemukan calon node level 2.
7. Calon *node* level 2 menerima Paket *Discovery*, menjadi *node* level 2, menjadi *child* dari salah satu *node* level 1, kemudian melakukan hal yang sama seperti *parent*-nya.
8. Setelah waktu berjalan 1 menit, semua node akan mengirimkan Paket TDMA sesuai jadwal/*time slot* masing-masing.
9. Waktu tahapan *Discovery*, Sinkronisasi, dan TDMA diukur.



Gambar 6.4 Tampilan Tahapan Discovery



Gambar 6.5 Tampilan Tahapan Sinkronisasi

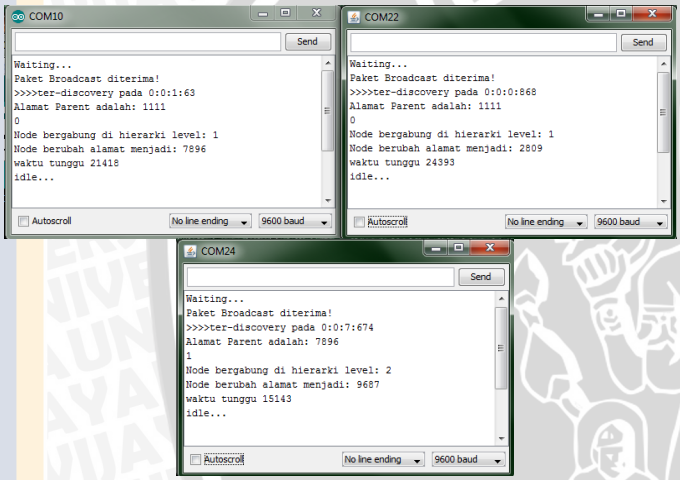


Gambar 6.6 Tampilan Tahapan TDMA

B. Pengujian Waktu untuk Proses Discovery

Proses discovery berguna untuk memberitahukan kepada setiap node, alamat dari *parent*-nya yang nanti akan dimintai paket

sinkronisasi. Pengujian ini untuk mencatat waktu yang dibutuhkan bagi sebuah *node* untuk masuk ke hierarki, mengenali alamat *parent*-nya, dan mencatat levelnya sendiri. Waktu dihitung dari pertama *node* aktif sampai menerima paket *discovery* dari *parent*-nya. Node *level* 1 langsung menerima paket discovery dari *root* (*level* 0), dan *node* level 2 menerima paket discovery dari node level 1. Pada level 1, terdapat dua *Node*, sedangkan pada *level* 2 terdapat satu *Node*. Gambar 6.7 menunjukkan hasil percobaan 1, setelah ketiga node menerima paket discovery. COM10 dan COM22 menjadi level 1, dan COM24 menjadi level 2 yang merupakan *child* dari COM10. Percobaan diulang sampai tiga kali, dan hasilnya dapat dilihat pada table 6.1.



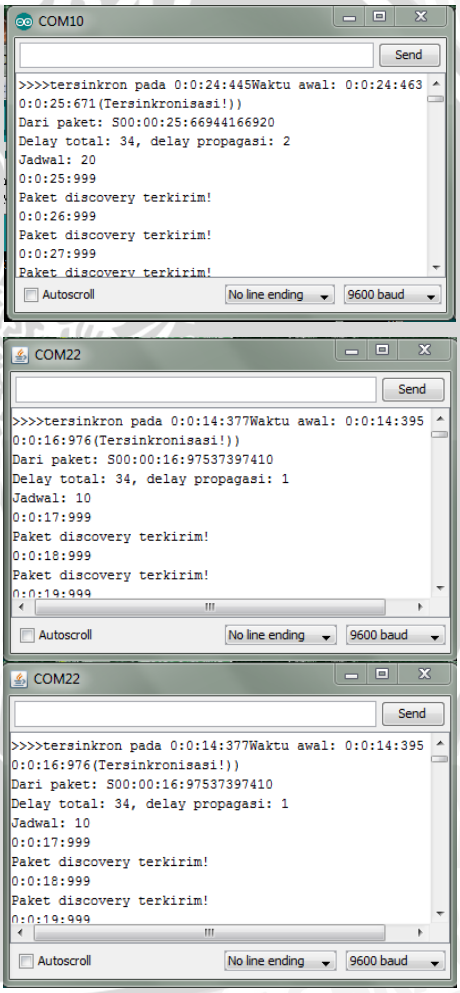
Gambar 6.7 Node-node setelah menerima paket *discovery*

Tabel 6. 1 Waktu untuk *discovery*

Percobaan ke-	Kode Node	Level Node	Waktu Discovery
1	A	1	1,630 detik
	B	1	0,868 detik
	C	2	7,674 detik
2	A	1	0,994 detik
	B	1	0,681 detik
	C	2	9,207 detik
3	A	1	0,710 detik
	B	1	0,293 detik
	C	2	9,260 detik

C. Pengujian Waktu untuk Proses Sinkronisasi

Proses sinkronisasi dilakukan setelah proses *discovery* selesai. Setiap *node* akan menunggu selama kurun waktu yang acak supaya memperkecil kemungkinan interferensi dengan node lain saat melakukan *request* sinkronisasi. Pada proses sinkronisasi, setiap node akan meminta *timestamp* ke *parent*-nya, untuk kemudian dicocokkan dengan waktu lokalnya sendiri. Percobaan pertama pada gambar 6.8 menunjukkan proses penerimaan paket balasan sinkronisasi dan pencocokan waktu dari ketiga node. Percobaan untuk proses sinkronisasi tersebut dilakukan sebanyak tiga kali ditunjukkan pada tabel 6.2.



Gambar 6.8 *Node-node* setelah mendapatkan paket balasan sinkronisasi

Tabel 6. 2 Waktu untuk sinkronisasi

Percobaan ke-	Kode Node	Level Node	Waktu dari awal node aktif	Waktu dari awal percobaan
1	A	1	24,445 detik	25,671 detik
	B	1	14,377 detik	16,976 detik
	C	2	21,849 detik	30,585 detik
2	A	1	23,462 detik	25,631 detik
	B	1	15,606 detik	19,263 detik
	C	2	22,626 detik	31,707 detik
3	A	1	14,936 detik	16,387 detik
	B	1	20,024 detik	23,002 detik
	C	2	20,907 detik	28, 893 detik

D. Pengujian Fleksibilitas Topologi Jaringan

Proses TDMA dilakukan setelah waktu percobaan di atas 1 menit untuk memastikan semua node dalam hierarki sudah tersinkronisasi semua. Semua *node* mengirimkan paket TDMA kepada *Node Root* ketika saatnya tiba setiap menit, dan mengirim ulang dua detik kemudian, namun pengambilan *sampling* hanya pada pengiriman pertama. Waktu diamati pada *Node Root* ketika paket TDMA telah sampai, kemudian dibandingkan dengan jadwal TDMA yang telah ditentukan untuk tiap-tiap node tersebut.

Pengujian fleksibilitas topologi menggunakan tiga skenario, yang masing-masing berbeda struktur topologinya. Ketiga skenario tersebut sebagai berikut.

1. 4 *node*, dengan susunan: 1 *Node Root*, 2 node level 1, 1 node level 2.
2. 4 *node*, dengan susunan: 1 *Node Root*, 3 node level 1.
3. 4 *node*, dengan susunan: *1 Node Root*, 1 node level 1, 2 node level 2.

Kemudian hasil pengujian ketiga skenario ditunjukkan pada tabel 6.3. dari tabel 6.3, dapat diketahui bahwa ketiga skenario pengujian berhasil membuktikan bahwa sistem yang dibuat fleksibel terhadap berbagai topologi yang diberikan.

Tabel 6. 3 Fleksibilitas Topologi

Skenario ke-	Kode Node	Level Node	Jadwal TDMA / Slot Waktu	Status Penerimaan TDMA pada Node Root
1	A	1	Detik ke-10	Diterima
	B	1	Detik ke-30	Diterima
	C	2	Detik ke-40	Diterima
2	A	1	Detik ke-10	Diterima
	B	1	Detik ke-20	Diterima
	C	1	Detik ke-30	Diterima
3	A	1	Detik ke-10	Diterima
	B	2	Detik ke-40	Diterima
	C	2	Detik ke-50	Diterima

E. Pengujian Akurasi Waktu

Pengujian akurasi waktu digunakan untuk mengetahui tingkat akurasi sinkronisasi waktu pada pengiriman TDMA. Waktu akan diukur ketika paket TDMA diterima pada sisi penerima dan akan dibandingkan dengan *time slot*, dengan asumsi delay propagasi tidak lebih dari 5 milidetik. Pengujian dilakukan menggunakan jarak antar *time slot* yang berbeda-beda pada tiap skenario. Pengujian akurasi waktu menggunakan 3 skenario sebagai berikut.

1. Skenario TDMA dengan jarak *time slot* 5 detik
2. Skenario TDMA dengan jarak *time slot* 3 detik
3. Skenario TDMA dengan jarak *time slot* 1 detik

Dari tabel 6.4, dapat diambil kesimpulan bahwa semua node berhasil mengirim pada *time slot* yang telah dijadwalkan. Meskipun setelah dikurangi delay propagasi, dan hasilnya sedikit terlambat dari jadwal beberapa puluh milidetik, namun pengiriman TDMA berhasil dilakukan tanpa adanya interferensi.

Tabel 6.4 Pengujian Akurasi Waktu

Skenario ke-	Jarak Slot Waktu	Kode Node	Level Node	Jadwal TDMA / Slot Waktu	Waktu Penerimaan (detik)
1	5	A	1	Detik ke-5	5,045
		B	1	Detik ke-10	10,090
		C	2	Detik ke-15	15,073
2	3	A	1	Detik ke-3	3,081
		B	1	Detik ke-6	6,042
		C	2	Detik ke-9	9,070
3	1	A	1	Detik ke-1	1,086
		B	1	Detik ke-2	2,040
		C	2	Detik ke-3	3,053

F. Analisis

Dari hasil pengujian, didapatkan hasil sebagai berikut, bahwa peralatan dapat bekerja dengan baik, dan sistem telah berjalan sesuai dengan keinginan dari proses awal sampai akhir. Proses Discovery dapat menghasilkan suatu hierarki *tree* berdasarkan model pengalamatan pada suatu node yang dicatat oleh *child*-nya, dan juga menggunakan penomoran level yang urut. Proses Discovery membutuhkan waktu rata-rata dari tiga pengujian sesuai perhitungan sebagai berikut.

1. Rata-rata waktu pembentukan level 1 (2 node):
   (1,630 + 0,868 + 0,994 + 0,681 + 0,710 + 0,293)/6 = 0,863 detik
2. Rata-rata waktu pembentukan level 2 (1 node):
   (7,674 + 9,207 + 9,260)/3 = 8,713 detik

Dari hasil pengujian Proses Sinkronisasi, didapatkan hasil pengujian rata-rata waktu yang dibutuhkan untuk sebuah level tersinkronisasi adalah sebagai berikut:

1. Rata-rata waktu sinkronisasi level 1 (2 node):
   - Dihitung dari awal node aktif:
     (24,445 + 14,377 + 23,462 + 15,606 + 14,936 + 20,024)/6 = 18, 809 detik
   - Dihitung dari awal percobaan:
     (25,671 + 16,976 + 25,631 + 19,263 + 16,387 + 23,002)/6 = 21,155 detik
2. Rata-rata waktu sinkronisasi level 2 (1 node):
   - Dihitung dari awal node aktif:
     (21,849 + 22,626 + 20,907)/3 = 21,794 detik
   - Dihitung dari awal percobaan:
     (30,585 + 31,707 + 28, 893) /3 = 30,395 detik

Hasil dari pengujian fleksibilitas program yang dibuat, terhadap perubahan struktur topologi membuktikan bahwa program TPSN tersebut fleksibel serta masih bisa berjalan untuk melakukan sinkronisasi waktu dan pengiriman TDMA sesuai dengan penjadwalan atau *time slot* yang diberikan. Program tetap bisa digunakan selama *time slot* masih ada atau belum digunakan oleh suatu node. Node level 1 selalu mendapatkan *time slot* dari angka 10 dan kelipatan 10. Node level 2 selalu mendapatkan *time slot* dimulai dari *time slot* yang dia miliki ditambah 30, dan berlaku kelipatan 10.

Untuk beberapa kasus seperti pengujian pada Gambar 6.9, node dengan port COM22 seharusnya mendapatkan *time slot* 20, namun justru mendapatkan nilai 30 dikarenakan node tersebut gagal dalam mendapatkan *timestamp* sinkronisasi pada *request* yang pertama, sehingga harus mengulangi *request* berikutnya. Kegagalan *request* sinkronisasi bisa disebabkan oleh waktu tunggu suatu node yang hamper sama dengan node lain, yang berakibat adanya tubrukan data sewaktu proses pengiriman.

Dari pengujian akurasi waktu yang telah dilakukan, dapat diambil analisis bahwa, pengiriman TDMA tetap bisa dilakukan meskipun menggunakan jadwal slot waktu yang sempit tanpa ada interferensi. Dan angka pengiriman sedikit terlambat beberapa puluh milidetik dari jadwal bisa disebabkan karena pengiriman waktu serial pada PC yang menimbulkan delay sewaktu akan dicetak

VI. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, pengujian dan analisis yang telah

dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Program algoritma TPSN yang dibuat cukup berjalan dengan baik dan memenuhi persyaratan yang sudah ditentukan sebelumnya. Ketepatan waktu yang dicapai mampu membuat pengiriman jadwal TDMA sesuai *time slot* yang diberikan.
2. Pembentukan hierarki sampai level 2 kurang dari 10 detik. Dengan rata-rata level 1 kurang dari 2 detik, dan level 2 kurang dari 9 detik.
3. Perubahan struktur hierarki tidak akan mempengaruhi keberhasilan algoritma TPSN dan pengiriman TDMA selama persediaan *time slot* masih ada.

DAFTAR PUSTAKA

Elson, J. and Estrin, D. 2001. *Time synchronization for wireless sensor networks*. San Francisco, CA.

Elson, J., Girod, L., and Estrin, D. 2002. *Fine-grained network time synchronization using reference broadcasts*. In UCLA Technical Report 020008.

Faludi, Robert. 2010. *Building Wireless Sensor Networks.*

Sohraby, dkk. 2007. *Wireless Sensor Networks: Technology, Protocols, and Applications.* Canada. Wiley.

Stallings, William. 2005. *Wireless Communication and Networks Second Edition*. Pearson Education, Inc. Sebastopol. O'Reilly.

Arduino, 2015. Product Arduino. [online] Tersedia di: <http://www.arduino.cc/en/Main/Products> [Diakses 20 April 2015]