

## Konfigurasi *Autonomous Routing* untuk Jaringan Sensor Nirkabel Berbasis XBee

**Kholid Mukhtar\*<sup>1</sup>, Raden Sumiharto<sup>2</sup>**

<sup>1</sup>Program Studi Elektronika dan Instrumentasi

<sup>2</sup>Jurusan Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta

e-mail: \*[kholid.mukhtar@mail.ugm.ac.id](mailto:kholid.mukhtar@mail.ugm.ac.id), <sup>2</sup>[r\\_sumiharto@ugm.ac.id](mailto:r_sumiharto@ugm.ac.id)

### **Abstrak**

Telah berhasil dikembangkan teknik routing secara flooding sehingga dapat bekerja secara *autonomous* untuk melakukan proses konfigurasi routing untuk jaringan sensor nirkabel. Teknik routing secara flooding digunakan dalam proses pencarian jalur data pada saat inisiasi jaringan. Selanjutnya jalur data yang dipilih melalui sink node dikirimkan ke sensor node untuk konfigurasi pengalamatan modul komunikasi XBee.

Penelitian dilakukan dengan menerapkan *autonomous routing* hasil pengembangan teknik routing secara flooding pada sistem jaringan sensor nirkabel yang berbasis modul komunikasi XBee. Sistem jaringan sensor nirkabel tersebut berfungsi sebagai pemantau lingkungan dengan parameter suhu dan intensitas cahaya. Sensor yang digunakan dalam pemantauan lingkungan adalah LM35DZ untuk pengukuran suhu dan LDR untuk pengukuran intensitas cahaya, sedangkan sebagai pusat pemrosesan pada sensor node digunakan mikrokontroler ATmega8.

Hasil dari penelitian ini adalah sebuah algoritma routing yang diwujudkan dalam bentuk perangkat lunak yang ditanamkan pada sensor node. Sistem telah diuji kerja konfigurasi *autonomous*nya pada jaringan dengan menggunakan 6 buah sensor node dan satu sink node. Dari 10 macam kemungkinan topologi pada jaringan yang diuji menghasilkan waktu konfigurasi routing tercepat 14 detik pada jaringan *single-hop*. Waktu konfigurasi terlama dibutuhkan pada proses konfigurasi jaringan dengan 5 sensor node yaitu selama 60 detik untuk jaringan dengan tiga hop. Setiap penambahan satu hop dari sink node pada jaringan akan mengakibatkan penambahan waktu eksekusi sistem rata-rata 15 detik.

**Kata kunci**—*autonomous routing, jaringan sensor nirkabel, flooding, ATmega8, XBee*

### **Abstract**

A routing algorithm in wireless sensor network has been developed based on flooding technique which can work *autonomously*. Flooding technique used in initiation process of network to search the alternative data paths. Then data path selected by sink node sent to sensor nodes for address configuration process.

In this study, *autonomous routing* configuration system applied to wireless sensor network which based on XBee communication module. The wireless sensor network system is aimed for environmental monitoring with temperature and light intensity measuring. Sensor node integrated with LM35DZ for temperature sensor, LDR for light intensity sensor, and ATmega8 for processing unit.

The result of this study is a routing algorithm which realized in software program for sensor nodes. The *autonomous* system was tested on the wireless network consist of 6 sensor nodes and one sink node. There are 10 of network topologies are tested on system. Fastest time for network routing configuration is 14 seconds on *single-hop* network, while longest time required for routing configuration process is 60 seconds which tested on network consist of 5 sensor node with multi hop. Every addition in hop of sink node to sensor node will give an average 15 seconds execution time for each process on system.

**Keywords**—*autonomous routing, wireless sensor network, flooding, ATmega8, XBee*

## 1. PENDAHULUAN

Berbagai teknik dan algoritma dalam protokol *routing* telah diteliti guna mengatasi berbagai permasalahan yang terdapat pada sebuah sistem Jaringan Sensor Nirkabel (JSN). Algoritma dalam suatu protokol *routing* merupakan suatu metode yang spesifik untuk aplikasi tertentu, artinya ketika digunakan pada suatu aplikasi JSN yang lain belum tentu sesuai dan bekerja dengan baik [1]. Misalnya pada beberapa algoritma *flatrouting* seperti *dirrecreddiffusion* (DD) yang bersifat *querydrivenbase*. Pada algoritma ini diperlukan adanya pengiriman sinyal *query* setiap sebelum melakukan proses pengiriman data, hal ini mengakibatkan kurang efektif ketika digunakan pada suatu aplikasi pemantauan lingkungan yang kontinyu[2]. Begitu juga pada teknik *routing* secara *flooding* yang proses pengiriman datanya dilakukan secara *broadcast* untuk setiap *node* secara terus-menerus dan tidak terdapat proses penentuan jalur data yang tetap. Hal tersebut mengakibatkan penggunaan daya yang tidak efisien.

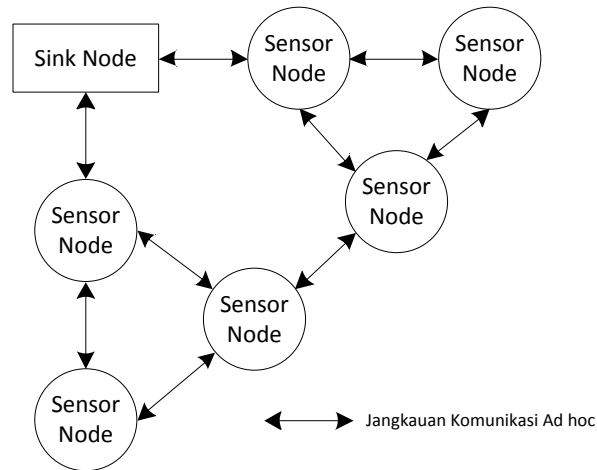
Pada penggunaan *flatrouting* pada JSN yang bersifat *datacentric*, metode pengalamatan *node* untuk penjaluran data dilakukan secara *dedicated*. Pada aplikasi JSN dengan jumlah sensor *node* yang sangat banyak, hal tersebut mengakibatkan kesulitan ketika harus dilakukan pengalamatan satu per satu untuk setiap *node*. Proses pengalamatan *node* juga harus diulangi ketika terjadi kerusakan pada jalur datanya.

Dalam aplikasi JSN untuk sistem pemantauan pengiriman barang dan pemantauan perilaku hewan muncul masalah lain ketika *node* yang ditanam pada barang atau hewan tersebut posisinya berubah-ubah terhadap *node-node* yang lain. Pada aplikasi tersebut juga terdapat kendala ketika terdapat salah satu *node* yang keluar dan masuk dalam jaringan. Hal tersebut mengakibatkan penggunaan *routing* bersifat tetap atau *dedicated* akan mengalami kegagalan dalam pengiriman data dari *sensornode* ke *sinknode*.

Saat ini jaringan nirkabel berbasis modul komunikasi XBee telah banyak digunakan untuk JSN terutama pada aplikasi pemantauan lingkungan. Pada kasus JSN pemantauan lingkungan dengan cakupan area yang luas dan terdapat kemungkinan terjadinya kerusakan pada salah satu *nodenya* dibutuhkan adanya *routing* yang dapat bekerja secara *autonomous*. Jaringan sensor nirkabel yang bersifat multi hop membutuhkan adanya suatu *self-organizing* ad hoc dalam jaringannya, tanpa dengan bantuan manusia[3]. Dengan menggunakan algoritma *routing* yang bekerja secara *autonomous* diharapkan dapat mengatasi kesulitan dalam proses inisiasi penjaluran data dan adanya *node* yang keluar atau masuk jaringan. Teknik *routing* secara *flooding* dijadikan sebagai bahan pengembangan untuk membangun *routing* yang dapat bekerja secara *autonomous*. Teknik *routing* secara *flooding* merupakan teknik dasar dalam proses *routing* JSN yang mempunyai cara kerja sederhana sehingga lebih mudah untuk dikembangkan.

## 2. METODE PENELITIAN

Sistem yang dibangun adalah sistem jaringan sensor nirkabel yang memiliki kemampuan dalam mengatur *routing* data antar *sensornode* secara *autonomous*, sehingga data dari semua *sensornode* dapat diterima oleh *sinknode*. Kemampuan tersebut ditujukan untuk membangun sebuah algoritma *routing* yang adaptif terhadap kondisi jaringan yang dinamis seperti bertambahnya *node* baru ke dalam jaringan. Blok diagram sistem jaringan sensor nirkabel yang akan dibangun dapat dilihat pada Gambar 1.



Gambar 1 Blok Diagram Sistem

Sistem yang akan dibangun terdiri dari enam buah *sensornode* dan satu buah *sinknode*. *Sensornode* berfungsi untuk memperoleh data dari lingkungan untuk selanjutnya dikirimkan ke *sinknode*. Pada *node* tersebut, terdapat beberapa parameter yang diambil datanya oleh sensor yang tertanam. Selain itu, *sensornode* juga berfungsi sebagai relay dalam jaringan, yang bertugas untuk meneruskan data yang diperoleh dari *sensornode* sebelumnya ke *sensornode* berikutnya dalam suatu jalur data. Masing-masing *sensornode* memiliki kemampuan dan karakteristik yang sama dalam jaringan. *Sinknode* berfungsi sebagai pusat agregasi data dari semua sensor node, dan menjadi *node* yang menyediakan antarmuka dalam mengatur jaringan. Koneksi nirkabel antar *node* menggunakan komunikasi *radiofrequency* (RF) dengan modul XBee.

*Sensornode* dapat ditempatkan dalam posisi acak selama masih masuk dalam jangkauan *sensornode* lain, minimal satu buah. Hal tersebut ditujukan agar jaringan dapat membentuk suatu topologi. Agar jaringan lebih stabil, maka suatu *sensornode* dimasukkan dalam dua atau lebih jangkauan dari *sensornode* yang lain agar ketika satu jalur komunikasi rusak, dapat diganti dengan jalur komunikasi yang lain.

Meskipun mempunyai *routing* bersifat *autonomous*, jaringan sensor nirkabel pada sistem menggunakan konfigurasi jaringan secara DHCP (*Dynamic Host Configuration Protocol*). Pada konfigurasi DHCP, *server* akan mengatur jaringan secara umum seperti konfigurasi jalur data dan proses inisiasi jaringan. Dalam sistem jaringan sensor nirkabel ini, yang bertugas sebagai *server* adalah *sinknode* yang berhubungan secara langsung dengan pengguna sehingga dapat mengatur dan memonitor kerja jaringan.

Untuk dapat bekerja sistem membutuhkan dua langkah dalam melakukan pengaturan *routing* jaringan. Proses yang pertama dilakukan sistem adalah inisiasi jaringan ketika *sinknode* mengirimkan sinyal khusus ke *sensornode* yang ada didekatnya. *Sensornode* yang menerima sinyal tersebut akan merespon dengan mengirimkan alternatif jalur data dari semua *sensornode* secara *flooding*.

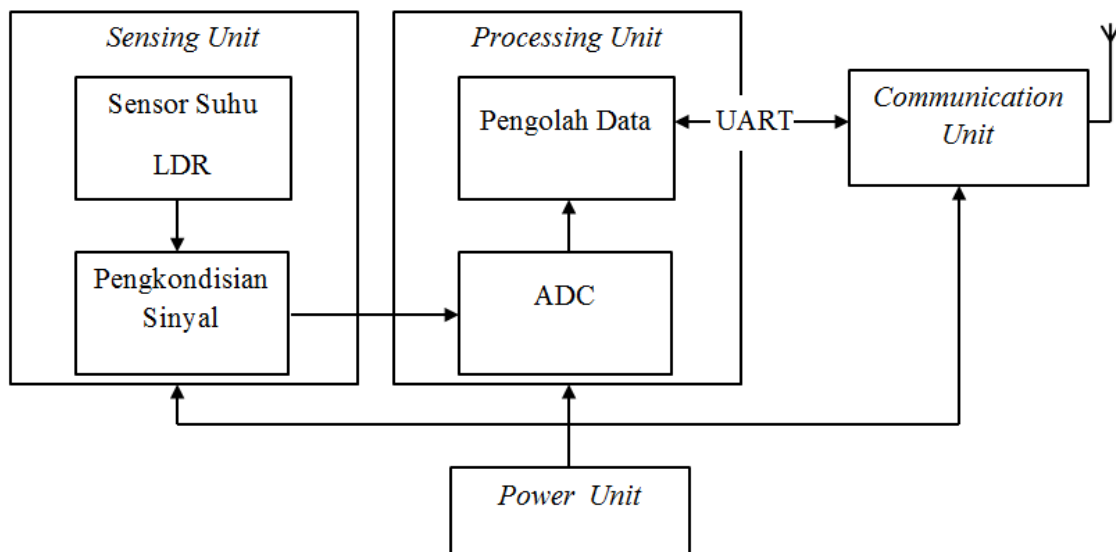
Setelah *sinknode* menerima semua kemungkinan jalur data dalam jaringan, maka langkah berikutnya adalah pemilihan jalur data oleh operator untuk dikirimkan ke semua *sensornode*. Perintah tersebut dikirimkan dengan paket data khusus yang dikenali semua *sensornode*. Kemudian *sensornode* yang menerima paket data perintah akan meneruskan ke *sensornode* berikutnya sesuai jalur data sebelum mengatur pengalamatan untuk *sensornode* tujuan transmisi data sensor. Proses tersebut berulang sampai *sensornode* paling ujung dari suatu jalur data dalam jaringan.

Ketika semua *sensornode* sudah melakukan konfigurasi pengalamatan jalur data, maka *sensornode* yang paling ujung akan memulai mengambil data dari sensor dan mengirimkan ke *node* berikutnya. *Sensornode* yang menerima paket data tersebut akan menambah paket data dengan data sensor yang dimiliki dan mengirimkan kembali ke *node*. Proses tersebut berulang

hingga paket data diterima oleh *sinknode*. Pada *sinknode* paket data yang diterima akan diolah dan ditampilkan melalui antarmuka. Ketika terdapat *node* baru atau *node* yang hilang, operator melalui *sinknode* akan mereset jaringan dan mengulangi proses konfigurasi *routing* dari awal.

### 2.1. Perangkat Keras

Perangkat keras sistem terdiri dari sebuah *sinknode* sebagai *basestation* serta enam buah *sensornode* yang bekerjasama membentuk jaringan sensor nirkabel. *Sinknode* terdiri dari sub sistem *communicationunit* sebagai media komunikasi nirkabel, dan komputer sebagai pusat pemrosesan dan penyedia antar muka. Sedangkan perangkat keras *sensornode* seperti ditunjukkan Gambar 2 terdiri dari sub sistem *powerunit*, *sensingunit*, *processingunit* dan *communicationunit*. Rancangan tersebut identik untuk seluruh *sensornode*.



Gambar 2 Blok diagram *sensornode*

*Sensingunit* terdiri dari sensor suhu yang bertugas untuk mengukur suhu dan LDR yang berfungsi mengukur intensitas cahaya dari lingkungan. Pada *sensingunit* terdapat pula sub sistem pengkondisian sinyal hasil dari pengukuran sensor. Sinyal analog dari *sensingunit* akan diproses oleh *processingunit*. Proses-proses yang terjadi disini berupa konversi data analog ke digital dan pengiriman data ke RF. Aktivitas komunikasi nirkabel antar *node* dilakukan oleh *communicationunit*. Untuk beroperasi, ketiga bagian tersebut mendapat suplai tegangan dari *powerunit*.

### 2.2. Perangkat Lunak

Perangkat lunak pada *sinknode* dan *sensornode* mempunyai tugas dan fungsi berbeda. Begitu pula dengan perangkat lunak yang digunakan untuk membuat bahasa pemrograman, karena rancangan perangkat keras yang berbeda antara *sinknode* dan *sensornode*. *Sinknode* yang menggunakan komputer sebagai pemroses dapat menggunakan bahasa pemrograman yang lebih bervariasi dan mempunyai antarmuka sendiri. Pada *sensornode*, pemrograman dilakukan dengan perangkat lunak BASCOM untuk mengatur kerja mikrokontroler agar bekerja sesuai rancangan protokol.

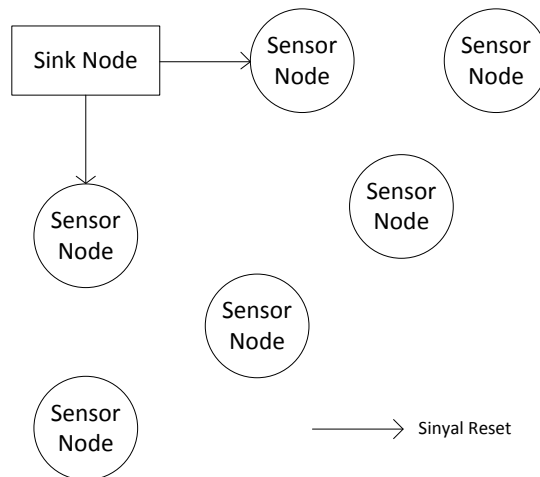
### 2.2. Rancangan Protokol

Jaringan sensor nirkabel pada sistem menggunakan konfigurasi jaringan dengan meniru konsep DHCP (*Dynamic Host Configuration Protocol*). Pada konfigurasi DHCP, server akan mengatur jaringan secara umum seperti konfigurasi jalur data dan proses inisiasi jaringan. Dalam sistem jaringan sensor nirkabel ini, yang bertugas sebagai server adalah *sinknode* yang

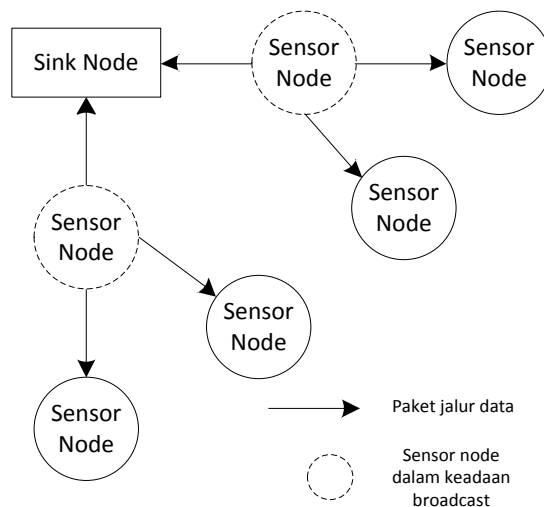
berhubungan secara langsung dengan pengguna sehingga dapat mengatur dan memonitor kerja jaringan. Dalam jaringan sensor nirkabel *sensornode* dapat berperan ganda, sebagai pemilik data dan *datarouter*[4]. Pada sistem ini *sensornode* selain bertugas dalam mengambil data dari sensor juga bertugas menerima data dari *sensornode*lain dan mentransmisikan ulang sesuai jalur datanya.

Untuk dapat bekerja sistem membutuhkan dua langkah dalam melakukan pengaturan *routing* jaringan. Proses yang pertama dilakukan sistem adalah inisiasi jaringan ketika *sinknode* mengirim sinyal khusus ke *sensornode* yang ada didekatnya. *Sensornode* yang menerima sinyal tersebut akan merespon dengan mengirimkan alternatif jalur data dari semua sensor node secara *flooding*.

Untuk lebih jelasnya proses inisiasi jaringan dalam mencari jalur data dengan *flooding* dapat dilihat pada Gambar 3 yang menunjukkan pengiriman sinyal reset oleh *sink node*. Pada Gambar 4 ditunjukkan perilaku *sensor node* setelah menerima sinyal reset yang akan mengubah komunikasinya menjadi *broadcast* sebelum mulai mengirimkan paket jalur data. Paket jalur data yang dikirimkan *sensornode* akan bekerja dengan teknik *routing* secara *flooding*. Pada algoritma ini setiap *sensornode* yang menerima paket data akan meneruskan transmisi paket data tersebut secara *broadcast*.

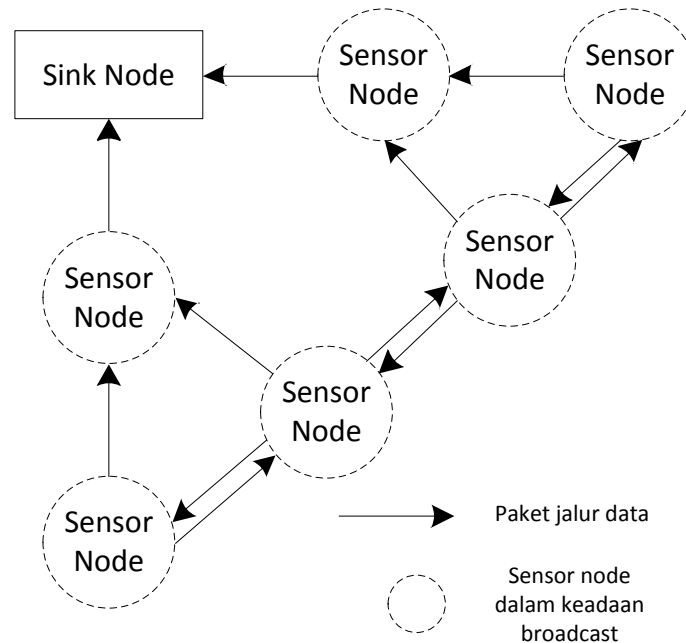


Gambar 3 Diagram blok proses pengiriman sinyal reset dari *sink node*



Gambar 4 Diagram blok proses inisiasi jalur data oleh *sensor node* secara *broadcast*

*Sensornode* yang ketika menerima paket jalur data belum menerima sinyal *reset/inisiasi* jaringan dari *sinknode* akan berperilaku sama ketika menerima paket jalur data. *Sensor node* tersebut akan mengubah komunikasi menjadi *broadcast* dan memulai pengiriman paket jalur data baru. Proses tersebut berlaku pada setiap *sensornode* dalam jaringan sehingga ketika proses pencarian jalur data melalui algoritma *flooding* selesai, *sinknode* akan menerima semua alternatif jalur data dari jaringan. Untuk lebih jelasnya, proses pengiriman paket jalur data jaringan secara *flooding* ini dapat diilustrasikan seperti pada Gambar 5.



Gambar 5 *Sensor node* akan meneruskan paket jalur data secara *broadcast* dengan algoritma *flooding*

Algoritma *flooding* hanya digunakan pada jaringan dalam mendapatkan semua jalur data yang mungkin dalam jaringan. Meskipun demikian, teknik *routing* secara *flooding* yang digunakan pada protokol sistem harus diatur agar tidak terjadi *trafficimplosion* dan *overlapping* data. Pengaturan tersebut dilakukan dengan memberikan variabel berupa cacah untuk meneruskan data didalam paket jalur data sehingga paket data tidak terus menerus berputar dalam jaringan. Setiap *sensornode* melakukan transmit paket jalur data, cacah akan berkurang sampai nantinya bernilai 0 dan paket jalur data sampai pada *sinknode*. Besarnya nilai variabel cacah disesuaikan dengan besarnya diameter jaringan, jika digunakan penempatan *node* yang membentuk topologi sistem seperti pada Gambar 1, maka cacah diberi 3 sampai 4. Namun apabila dikehendaki penggunaan topologi dengan rantai hop terpanjang, maka besar nilai cacah yang diberikan bernilai sama dengan jumlah sensor node dalam jaringan. Untuk menghindari terjadinya *collision*, diperlukan pewaktuan yang tepat dengan cara memberikan jeda waktu pada *sensornode* saat proses berlangsung dengan lama yang berbeda-beda.

Setelah semua *sensornode* mengirimkan paket jalur data secara *flooding*, maka *sinknode* memperoleh semua kemungkinan jalur data yang dapat dipakai dalam jaringan. Selanjutnya operator melalui *sinknode* yang mengambil keputusan untuk memilihkan jalur data yang akan dipakai dalam jaringan. Pemilihan jalur data didasarkan pada deret *node* yang paling panjang agar dapat mengurangi proses transmisi data sehingga konsumsi daya pada *sensornode* lebih efisien. Selanjutnya sink node akan mengirimkan paket data secara *broadcast* yang berisi

alamat-alamat *node* untuk konfigurasi *routing*. Paket konfigurasi *routing* tersebut akan diterima *sensornode* yang berada dalam jangkauan *sinknode*, lalu diteruskan secara *unicast* ke *node* selanjutnya sesuai dengan jalur data yang terdapat pada paket konfigurasi *routing*. *Sensornode* yang sudah menerima konfigurasi akan mengubah pengaturan *unicast* untuk alamat *node* tujuan sesuai dengan jalur data sehingga siap dalam menerima paket data sensor. *Sensornode* lain yang menerima paket konfigurasi *routing* akan melakukan proses yang sama hingga semua *node* yang terdapat pada paket tersebut mengatur pengalamatan *node* tujuan. *Node* yang paling ujung dalam suatu paket data, nantinya akan bekerja sebagai awal jalur data dan bertugas menginisiasi pengiriman paket data sensor.

Setelah ujung dari setiap jalur data yang dipilih mulai mengirimkan paket data sensor, *sensornode* yang menerima akan menambah paket data dengan data sensor yang dimiliki, lalu mengirimkan kembali ke *sensornode* tujuan. Proses tersebut berulang sehingga paket data sensor sampai pada *sinknode*.

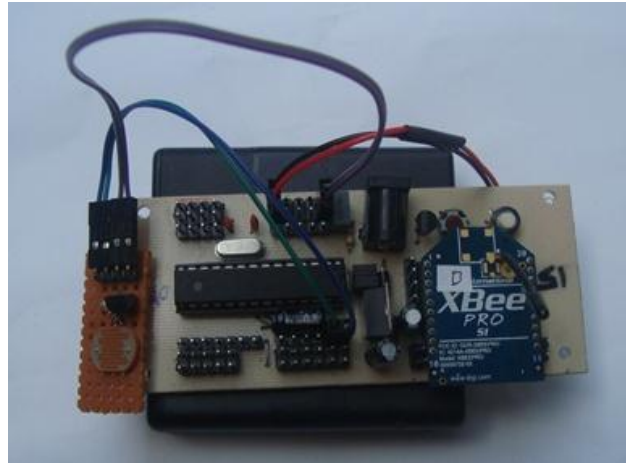
*Sink node* akan menerima semua kemungkinan jalur data dalam jaringan dan operator akan memilih jalur data yang dianggap paling baik untuk dikirimkan ke semua *sensornode*. Perintah tersebut dikirimkan dengan paket data khusus yang dikenali semua *sensor node*. Kemudian *sensor node* yang menerima paket data perintah akan meneruskan ke *sensor node* berikutnya sesuai jalur data sebelum mengatur pengalamatan untuk *sensor node* tujuan transmisi data sensor. Proses tersebut berulang sampai *sensor node* paling ujung dari suatu jalur data dalam jaringan.

Ketika semua *sensor node* sudah melakukan konfigurasi pengalamatan jalur data, maka *sensor node* yang paling ujung akan memulai mengambil data dari sensor dan mengirimkan ke *node* berikutnya. *Sensor node* yang menerima paket data tersebut akan menambah paket data dengan data sensor yang dimiliki dan mengirimkan kembali ke *node*. Proses tersebut berulang hingga paket data diterima oleh *sink node*. Pada *sink node* paket data yang diterima akan diolah dan ditampilkan melalui antarmuka. Ketika terdapat *node* baru atau *node* yang hilang, operator melalui *sink node* mereset jaringan dan mengulangi proses konfigurasi *routing* dari awal.

### 2.3. Implementasi Perangkat Keras Sistem

Implementasi sistem dari *Power Unit*, *Processing Unit* dan *Communication Unit* dibangun dalam satu *board*. Hal ini ditujukan untuk menciptakan suatu *node* yang memiliki sistem tersentralisasi dalam satu *development system* sehingga dimensi dari sistem dapat dimaksimalkan. Agar menjadi satu kesatuan sistem *node*, maka berbagai unit perangkat keras yang dibangun dalam dua buah *board* perlu diinterkoneksi. *Board development system* sebagai sentral sistem disalurkan dengan catu tegangan dari *power supply* baterai sebagai *power unit* yang disalurkan pada unit-unit lain. Untuk interkoneksi antara *board development system* dengan *board sensing unit* digunakan kabel penghubung.

Integrasi sistem perangkat keras *sensor node* identik untuk seluruh *node*. Implementasi perangkat *sensor node* secara keseluruhan ditunjukkan pada Gambar 6. Setelah implementasi *sensor node*, maka dilakukan ekspansi dengan mengimplementasikan beberapa *node* sekaligus untuk membangun sebuah jaringan sensor nirkabel. Dalam jaringan ini, masing-masing *sensornode* mempunyai peran dan fungsi yang sama. Perangkat keras *sink node* berupa komputer yang terhubung melalui komunikasi serial.



Gambar 6 Integrasi *sensor node*

#### 2.4. Implementasi Perangkat Lunak Sistem

Perangkat lunak sistem dibuat untuk memprogram dua jenis perangkat keras dalam sistem yaitu *sensor node* dan *sink node*. Perangkat lunak untuk *sensor node* merupakan pemrograman untuk mikrokontroler. Pembuatan program dilakukan menggunakan BASCOM AVR dengan bahasa pemrograman BASIC.

Implementasi perangkat lunak pada *sensor node* disesuaikan agar dapat bekerja untuk menjalankan rancangan protokol komunikasi dalam proses *routing*. Dari rancangan tersebut seluruh *sensor node* dipasang dengan perangkat lunak yang identik. Fungsi utama yang akan dijalankan oleh *sensor node* adalah bekerja secara *autonomous* untuk mencari dan mengkonfigurasi jalur data dalam jaringan. Fungsi lain yang dijalankan *sensor node* adalah mengambil data dari kedua sensor yang terpasang yaitu data parameter suhu dan intensitas cahaya.

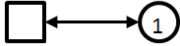
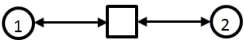

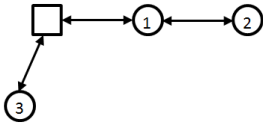
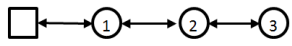

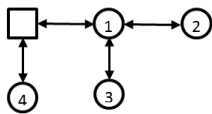
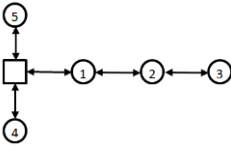
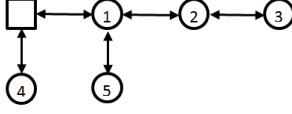
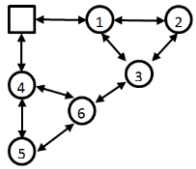
Pemrograman perangkat lunak untuk *sink node* sekaligus dibuat untuk antar muka dengan komputer. Dengan menggunakan komputer sebagai pemroses utama pada *sink node* dapat digunakan bahasa pemrograman pascal. DELPHI 7 digunakan sebagai *tool* pemrograman *sink node* dan pembuatan antar mukanya. Sebagai program antar muka, maka fungsi-fungsi yang dijalankan pada *sink node* sebagian besar dipicu melalui tombol pada antar muka. Fungsi-fungsi yang terdapat pada implementasi perangkat lunak ini antara lain: fungsi mengirimkan perintah reset/inisiasi jaringan, fungsi untuk menerima berbagai jalur data yang mungkin dari jaringan, fungsi untuk mengirimkan konfigurasi *routing* sesuai pilihan jalur data, dan fungsi untuk mengurai paket data sensor menjadi data parameter yang sesuai serta fungsi mengenali *sensor node* pengirim.



### 3. HASIL DAN PEMBAHASAN

Pengujian dilakukan untuk mengetahui unjuk kerja sistem secara keseluruhan. Pengujian dilakukan di luar ruangan dengan lokasi yang digunakan adalah kompleks perumahan Nusa Indah, Sekip Blok N. Pengujian mula-mula dilakukan dengan menggunakan satu buah *sensor node* dalam jaringan, kemudian secara bertahap ditambah jumlah *sensor node*-nya. Setiap jumlah *sensor node* tertentu dilakukan variasi topologi penempatan *sensor node*-nya sehingga untuk setiap kemungkinan topologi jaringan dapat diketahui keluaran dari sistem. Hal tersebut ditujukan pula untuk mengetahui kestabilan dan performa sistem dalam berbagai kondisi jaringan. Hasil dari pengujian ini dapat dilihat pada Tabel 1.



Tabel 1 Hasil pengujian sistem konfigurasi *autonomous routing* pada jaringan

No	Jumlah sensor node	Topologi*	Alternatif jalur data**	Waktu Eksekusi (detik)	
				Pencarian jalur data	Konfigurasi jalur data
1	1		N1	13 14 13 <b>13,3</b>	14 14 14 <b>14</b>
2	2		N1 N2	14 13 14 <b>13,7</b>	17 19 19 <b>18,3</b>
3	2		N1 N1N2	29 28 28 <b>28,3</b>	28 29 29 <b>28,6</b>
4	3		N1 N3 N1N2	28 28 28 <b>28</b>	32 32 34 <b>33</b>
5	3		N1 N1N2 N1N2N3	41 40 41 <b>40,6</b>	42 43 43 <b>42,6</b>
6	4		N1 N3 N1N2 N3N4	29 28 30 <b>29</b>	34 34 37 <b>35,3</b>
7	4		N1 N4 N1N2 N1N3	28 28 29 <b>28,3</b>	55 60 57 <b>57,3</b>
8	5		N1 N4 N5 N1N2 N1N2N3	44 45 45 <b>44,6</b>	47 48 47 <b>47,3</b>
9	5		N1 N4 N1N2 N1N5 N1N2N3	44 46 46 <b>45,3</b>	59 62 60 <b>60,3</b>
10	6		N1 N4 N1N2 N1N3 N4N6 N1N2N3 N4N5N6	45 45 46 <b>45,3</b>	42 43 43 <b>42,6</b>

\*  : Sink Node       : Sensor Node

\*\* Jalur data yang dicetak tebal merupakan jalur data yang digunakan

Pengujian dilakukan tidak secara berurutan untuk mensimulasikan adanya penambahan *sensornode* baru atau hilangnya suatu *sensornode* pada jaringan. Begitu pula dengan topologi jaringan yang diujikan merupakan topologi yang mewakili beberapa kemungkinan sehingga dapat menunjukkan performa sistem dalam setiap kondisi jaringan. Penggunaan topologi yang sesuai juga dapat memperluas cakupan sistem dalam aplikasi pengamatan lingkungan yang diterapkan.

Dari hasil perhitungan secara teori diperkirakan untuk eksekusi satu siklus pencarian alternatif jalur data dibutuhkan waktu 13 detik untuk setiap *hop* yang tidak berhubungan dengan *sinknode*. Waktu tersebut merupakan penjumlahan dari waktu untuk melakukan perubahan konfigurasi XBee melalui komunikasi serial dalam mode perintah *at.Guardtime* sebelum dan sesudah pengiriman karakter “+++” membutuhkan waktu masing-masing 1,1 detik dan waktu untuk mengubah pengaturan XBee agar dapat melakukan pengiriman data dengan konfigurasi yang baru adalah selama 10,1 detik. Sehingga total waktu untuk mengubah konfigurasi XBee sampai siap melakukan pengiriman data adalah 12,3 detik. Penambahan waktu juga dilakukan program untuk waktu jeda masing-masing *sensornode* yang besarnya antara 0,1 sampai 1 detik. Sehingga total waktu rata-rata yang dibutuhkan secara teori adalah 13 detik untuk setiap siklus satu *hop* proses.

Hasil perhitungan secara teori waktu untuk eksekusi satu siklus pencarian alternatif jalur data yang dibutuhkan adalah 13 detik untuk setiap *hop* yang tidak berhubungan dengan *sink node*. Waktu tersebut merupakan penjumlahan dari waktu untuk melakukan perubahan konfigurasi XBee melalui komunikasi serial dalam mode perintah *at.Guard time* sebelum dan sesudah pengiriman karakter “+++” membutuhkan waktu masing-masing 1,1 detik dan waktu untuk mengubah pengaturan XBee agar dapat melakukan pengiriman data dengan konfigurasi yang baru adalah selama 10,1 detik. Sehingga total waktu untuk mengubah konfigurasi XBee sampai siap melakukan pengiriman data adalah 12,3 detik. Penambahan waktu juga dilakukan program untuk waktu jeda masing-masing sensor node yang besarnya antara 0,1 sampai 1 detik. Sehingga total waktu rata-rata yang dibutuhkan secara teori adalah 13 detik untuk setiap siklus satu *hop* proses.

Hasil pengujian di lapangan dibutuhkan rata-rata 15 detik untuk setiap siklus pencarian alternatif jalur data. Hal tersebut disebabkan diberikannya waktu jeda dalam proses pengiriman data untuk masing-masing node yang berbeda dengan tujuan menghindari adanya *collision* data. *Datarate* modul XBee-PRO sendiri menurut *datasheet* adalah sebesar 250.000 *bit per second*, sehingga untuk mengirim data sebesar 1 bit diperlukan waktu 0,000004 detik [5]. Oleh karena itu waktu yang dibutuhkan dalam pengiriman suatu paket data tidak terlalu signifikan pada total waktu yang dibutuhkan setiap siklus. Berdasarkan hasil perhitungan dan pengujian dapat disimpulkan penambahan satu *node* menambah waktu eksekusi untuk satu siklus kurang lebih sebesar 14 detik. Waktu eksekusi tersebut akan bertambah setiap penambahan *hop* pada posisi paling luar dalam topologi jaringan. Tingkat ketelitian pengukuran waktu dalam pengujian juga mempengaruhi hasil percobaan. Pada pengujian ini hanya digunakan pewaktu dari antarmuka delphi dengan tingkat ketelitian 1 detik.

Waktu yang dibutuhkan dalam pengiriman jalur konfigurasi data pada pengujian relatif lebih besar dibanding dengan waktu eksekusi proses pencarian jalur data. Hal tersebut disebabkan pada topologi tertentu seperti pada pengujian topologi 2, 4, 6, 7, 8, 9, dan 10 dibutuhkan lebih dari satu kali pengiriman paket data konfigurasi. Sehingga topologi dengan jumlah *sensornode* yang semakin banyak akan mengakibatkan kemungkinan dibutuhkannya pengiriman perintah konfigurasi *routing* yang semakin banyak pula.

Pada topologi yang menggunakan jalur data yang memiliki percabangan seperti pada pengujian ke 7 dan ke 9, dibutuhkan waktu yang lebih lama untuk melakukan konfigurasi jalur data. Hal tersebut dikarenakan sewaktu *sensornode* pada percabangan sedang mengkonfigurasi jalur data untuk salah satu cabang, maka pengaturan untuk konfigurasi cabang yang lain harus menunggu sampai paket data konfigurasi yang diterima sebelumnya ditransmisikan terlebih dahulu pada *sensornode* yang sesuai. Pada sistem ini pemberian jeda waktu untuk mengirimkan

paket konfigurasi routing untuk sensor node yang bercabang masih dilakukan secara manual, dengan lama minimal 14 detik. Setelah jeda tersebut, sensor node yang merupakan percabangan dalam jalur data telah siap kembali untuk menerima paket konfigurasi *routing*. Meskipun jalur data yang digunakan mempunyai percabangan, namun *sensornode* pada percabangan hanya berfungsi untuk meneruskan paket data sensor, sehingga berbeda dengan seperti *parentnode* pada routing yang bersifat hierarki.

Pada pengujian juga didapati terjadinya kegagalan dalam mengirimkan perintah seperti dalam perintah reset jaringan maupun perintah konfigurasi *routing*. Kegagalan tersebut dimungkinkan terjadi karena penggunaan interupsi urxc yang terus di *loop* yang tidak menjamin interupsi dari serial selalu diteruskan. Akan tetapi kegagalan pengiriman tersebut dapat diatasi dengan mengirimkan ulang sinyal perintah dari *sinknode* yang mengakibatkan tertundanya konfigurasi dan penambahan waktu eksekusi.

Hasil pengujian ke 10 yang menggunakan 6 *sensornode* merupakan pengujian yang memiliki kesulitan terbesar. Selain karena topologi yang digunakan cukup rumit untuk dibangun, penggunaan *sensornode* yang semakin banyak juga mengakibatkan kemungkinan terjadinya *collision* data juga semakin besar. Pada saat proses pencarian jalur data, semua *sensornode* diharuskan mentransmit data pada saat yang hampir bersamaan sehingga kemungkinan terjadinya *collision* cukup besar. Adanya *collision* tersebut mengakibatkan beberapa jalur data tidak sampai pada *sink node*. Pada pengujian ke 6 beberapa jalur data jaringan yang dapat digunakan namun tidak dimunculkan oleh sistem diantaranya adalah N1N2N3N6 dan N4N5N6N3. Dari pengujian ini dapat disimpulkan bahwa semakin banyak jumlah *sensornode* yang digunakan dan banyaknya *sensor node* yang saling terhubung, akan mengakibatkan semakin besarnya kemungkinan jalur data yang tidak dapat ditampilkan oleh sistem.

Dari keseluruhan pengujian, secara umum algoritma *routing* yang digunakan sudah dapat bekerja secara *autonomous* dalam mencari alternatif jalur data dan melakukan konfigurasi jalur data untuk masing-masing *sensor node*. Meskipun pengujian hanya dilakukan dengan menggunakan jumlah paling banyak 6 sensor node, tetapi *routing* yang dibangun secara teori dapat dilakukan pada jaringan yang lebih besar. Namun dengan bertambah besarnya jaringan maka waktu dan kemungkinan terjadinya kegagalan dalam konfigurasi *routing* menjadi lebih besar. Hal tersebut dapat diminimalkan dengan menempatkan *sinknode* di tengah jaringan sehingga jumlah *hop* antara *sensornode* dengan *sinknode* dapat berkurang.

#### 4. KESIMPULAN

Modifikasi dan pengembangan teknik *routing* secara *flooding* agar dapat bekerja secara *autonomous* dalam jaringan sensor nirkabel telah berhasil dilakukan pada kasus jaringan nirkabel yang berbasis modul RF XBee. Kemampuan yang dimiliki sistem tersebut terdiri dari kemampuan konfigurasi routing melalui antarmuka di sink node, kemampuan adanya keluar masuknya suatu node di jaringan serta mampu memberikan berbagai alternatif jalur data pada setiap topologi penempatan sensor node dalam jaringan. Setiap penambahan satu *sensor node* pada bagian terluar jaringan mengakibatkan penambahan waktu eksekusi sistem rata-rata 15 detik. Selain itu dibutuhkan sistem pewaktuan yang handal dalam komunikasi secara *broadcast* pada jaringan untuk menghindari terjadinya *collision* data.

#### 5. SARAN

Terdapat peluang lebih lanjut tentang pengembangan algoritma protokol *routing* dalam jaringan sensor nirkabel yang lebih cerdas, cepat dan real time. Pengembangan lain dapat juga dilakukan dalam hal implementasi *autonomous routing* pada aplikasi jaringan sensor nirkabel lain yang lebih luas.

## DAFTAR PUSTAKA

- [1] Al-karaki, J. N. and Kamal, A. E., 2004, Routing techniques in Wireless Sensor Networks: A survey, *IEEE Wireless Communications*, 11:6–28.
- [2] Sohraby, K., Minoli, D., dan Znati T, 2007, *Wireless Sensor Networks: Technology, Protocols, and Applications*, Wiley & Sons, Inc., New Jersey.
- [3] Callaway, E. H., 2004, *Wireless Sensor Networks: Architectures and Protocols*, CRC Press.
- [4] Akyildiz, I. F. and Vuran, M. C., 2010, *Wireless Sensor Networks*, John Wiley & Sons Ltd, Chichester.
- [5] Digi International Inc., 2009, *XBee®/XBee-PRO® RF Modules Product Manual*, [www.digi.com](http://www.digi.com), diakses 24 April 2012.