

# RANCANG BANGUN SISTEM KOMUNIKASI MULTIHOP PADA JARINGAN SENSOR NIRKABEL

Deny Febriyanto<sup>1</sup>, Tri Budi Santoso<sup>1</sup>, Taufiqurrahman<sup>2</sup>.

<sup>1</sup>Laboratorium *Digital Signal Processing*, Politeknik Elektronika Negeri Surabaya

<sup>2</sup>Laboratorium *Industrial Electronic*, Politeknik Elektronika Negeri Surabaya

Kampus ITS, Surabaya 60111

e-mail : briantampan@student.eepis-its.edu

## ABSTRAK

Jaringan Sensor Nirkabel (JSN) merupakan jaringan tanpa kabel yang terdiri dari sekumpulan sensor (*node*), yang berfungsi untuk memonitor kondisi lingkungan tertentu dan kemudian mengirimkan data informasi hasil monitoring tersebut ke *node master* (*Administrator*).

JSN mempunyai cakupan area yang luas dan biasanya disebar pada kondisi lingkungan dimana seringkali terdapat *obstacle* (penghalang) yang dapat mengakibatkan *losses*. Maka dari itu, pada Tugas Akhir ini dirancang sebuah sistem komunikasi yang memungkinkan *node slave* mengirimkan data informasi hasil monitoring secara tidak langsung menuju *node master*, yaitu melalui (baca : melompati) *node-node* terdekat di sekitarnya. Inilah yang dinamakan sistem komunikasi Multihop.

Komunikasi paket data secara multihop pada proyek akhir ini dapat dikatakan berhasil, dimana tingkat keberhasilannya sebesar 98%, yaitu hampir semua paket data sukses dikirim dan diterima oleh tiap-tiap *node*, baik itu double hop ataupun hingga triple hop.

**Kata Kunci :** Jaringan Sensor Nirkabel, *node*, multihop, IEEE 802.15.4.

## I. PENDAHULUAN

Dewasa ini, perkembangan protokol komunikasi dan informasi sangatlah pesat. Ini diiringi dengan semakin cepatnya perkembangan perangkat elektronika. Sehingga banyak tercipta perangkat telekomunikasi generasi baru dengan teknologi yang semakin canggih. Salah satunya adalah Jaringan Sensor Nirkabel atau biasa disebut *Wireless Sensor Network*.

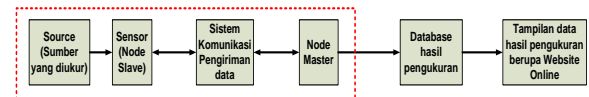
Keunggulan dari Jaringan Sensor Nirkabel ini adalah mempunyai daya jangkau yang lebih luas, akurat dan juga harga yang lebih murah. Ini dikarenakan kemajuan di bidang desain, konsep dan pemilihan material sehingga diperoleh sensor yang lebih murah, minimalis dan ringan. Tetapi memiliki kelemahan yaitu sering terjadi *losses* pada saat pengiriman data dikarenakan cakupan area yang luas dengan terdapat banyak *obstacle* (penghalang).

Untuk mengatasi kelemahan tersebut maka dibutuhkan sebuah sistem komunikasi yang dapat mengurangi *losses*. Sehingga munculah ide untuk membuat sistem komunikasi yang memungkinkan *node slave* mengirimkan data informasi hasil monitoring secara tidak langsung menuju *node master*, yaitu melalui *node-node* sensor terdekat di sekitarnya.

## II. PERENCANAAN SISTEM DAN TEORI PENUNJANG

### 2.1 PERANCANGAN SISTEM

Gambar 1 di bawah ini menunjukkan perencanaan sistem secara keseluruhan, dimana blok yang dibatasi dengan garis putus-putus menunjukkan lingkup pembahasan pada Proyek Akhir ini.



**Gambar 1** Blok diagram sistem keseluruhan

Dari blok diagram sistem keseluruhan dapat dijelaskan mengenai mekanisme dari sistem secara keseluruhan. *Node slave* pada blok di atas terdiri dari sensor suhu dan kelembaban (SHT11), mikrokontroler ATmega8535 yang berfungsi untuk memproses data hasil pengukuran sensor dan mengirimkannya melalui modul *Transceiver* Xbee Pro. Xbee Pro ini merupakan perangkat

komunikasi yang berbasis komunikasi 802.15.4. Dimana pada proyek Akhir ini, XBee Pro dirancang agar dapat mengirimkan paket data menuju *node destination* yang berubah-ubah sesuai dengan rute yang diberikan. Rute yang dimaksud adalah rute multihop atau komunikasi multihop.

## 2.1 BAHAN DAN ALAT

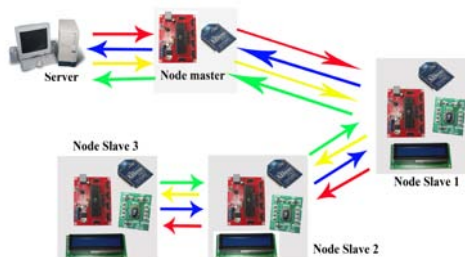
### 2.1.1 Wireless Sensor Network

Jaringan Sensor Nirkabel ini terdiri dari dua komponen utama yaitu sensor (*node slave*) dan *node master* (*Administrator*). Berbeda dengan jaringan Wi-fi yang menggunakan sistem pengiriman *Carrier Sense Multiple Access with Collision Detection* (CSMA/CD), pada Jaringan Sensor Nirkabel ini menggunakan sistem pengiriman *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) dimana data baru akan dikirim ketika kanal transmisi telah kosong atau tidak terpakai sehingga data yang dikirimkan jarang terjadi losses. Seperti pada tabel 1 berikut :

**Tabel 1** Perbandingan sistem pengiriman data Jaringan Sensor Nirkabel dengan Jaringan Wi-Fi

Jaringan Wi-Fi (CSMA/CD)	Jaringan Sensor Nirkabel (CSMA/CA)
Pengiriman data terus menerus dan tidak menghiraukan ketersediaan kanal transmisi	Pengiriman data baru dilakukan ketika kanal transmisi telah kosong atau tidak terpakai
Data yang dikirim sering terjadi losses	Data yang dikirim jarang/tidak pernah terjadi losses

## 2.2 CARA KERJA

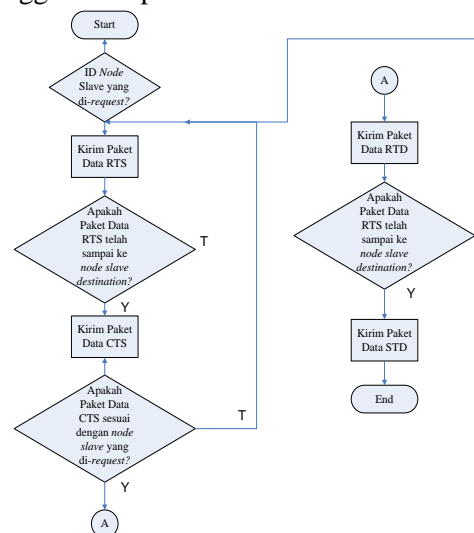


**Gambar 2** Cara kerja sistem

Keterangan :

- : Request to Send (RTS)
- ← : Clear to Send (CTS)
- : Request the data (RTD)
- ← : Send the data (STD)

Untuk me-request data dari *node slave* yang diinginkan atau *node slave* tujuan maka *node master* akan mengirimkan paket data yang menyatakan rute dan *node slave* mana yang akan direquest, ini yang dimaksud dengan *Request to Send*. Jika paket data tersebut telah diterima dengan utuh oleh *node slave* tujuan, maka *node slave* tujuan tersebut akan membalas paket data request dari *node master* dengan sebuah paket data yang berisi *Clear to Send* yang menyatakan bahwa paket data *Request* dari *node master* telah diterima oleh *node slave* tujuan yang dituju. Rute dari paket data *Clear to Send* ini adalah kebalikan dari paket data *Request*. Kemudian *node master* akan merequest data hasil pengukuran dari *node slave* tujuan dengan mengirimkan paket data request the data, maka *node slave* tujuan tersebut akan mengirimkan data hasil pengukuran suhu dan kelembabannya menggunakan paket data *send the data*.



**Gambar 3** Flowchart Sistem

Pada Gambar 3, saat *node master* me-request data, maka *ID Node Slave Destination* akan diikuti dalam paket data RTS (*Request to Send*). Jika, penerima paket RTS tersebut bukan *node slave* yang dituju, maka *node slave* tersebut akan meneruskan paket RTS tersebut menuju ke *node slave* yang dituju. Tetapi jika penerima paket RTS tersebut adalah *node slave* yang dituju, maka *node slave* tersebut akan membalas dengan paket CTS (*Clear to Send*) menuju *node master*. Apabila *ID node slave destination* pada paket CTS sesuai, maka *node master* akan mengirimkan paket RTD (*Request the Data*) kepada *node slave destination*. Kemudian setelah paket RTD tersebut sampai kepada *node slave destination*, maka data hasil

pengukuran suhu dan kelembaban pada *node slave destination* tersebut akan dikirimkan dengan paket STD (*Send the Data*).

### BAB III PEMBUATAN SISTEM, PENGUJIAN DAN ANALISA

Pada bab ini akan dibahas tentang langkah-langkah dalam pembuatan sistem yang merupakan realisasi dari perencanaan sistem yang telah didesain. Dimana Proyek Akhir ini secara umum terbagi menjadi 2 bagian, yaitu :

1. Perancangan dan pembuatan perangkat keras
2. Perancangan dan pembuatan sistem komunikasi dan protokol.

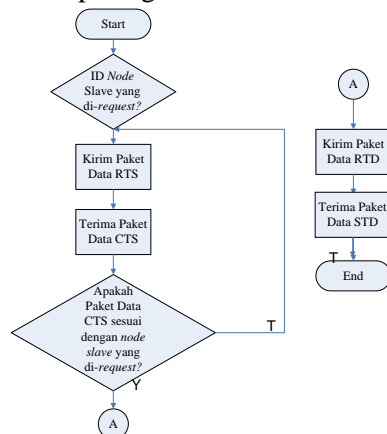
## 3.1 PEMBUATAN SISTEM

### 3.1.1 Pembuatan Hardware

#### 3.1.1.1 Node Master

*Node Master* pada Proyek Akhir ini terdiri dari mikrokontroler ATmega162 dan *driver board* XBee Pro. Proses yang terjadi pada *node master* ini antara lain *me-routing*-kan paket data yang akan dikirim menuju *node slave destination*, memproses data yang diterima dari *node slave* apakah paket data sesuai dengan yang *di-request* atau tidak dan sebagai jembatan komunikasi antara *node slave* dengan *Server* yaitu PC.

*Flowchart* sistem dari *node master* dapat dilihat pada gambar 4 berikut.



**Gambar 4** Flowchart node master

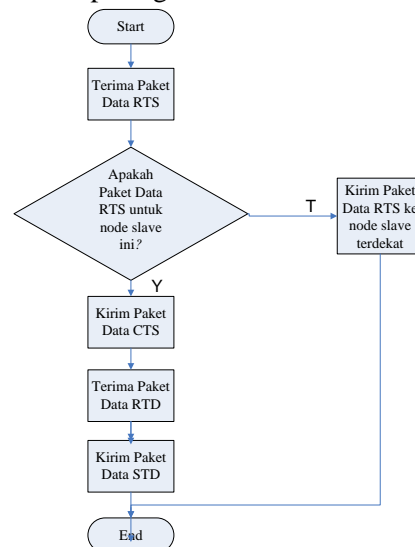
Pada *flowchart* Gambar 4, *node master* akan *me-routing*-kan paket data RTS, dimana ID *node destination* pada paket data diinputkan melalui PC. Kemudian jika paket data RTS telah sampai ke *node slave* yang dituju, maka *node master* akan memperoleh paket data CTS yang merupakan paket balasan yang berisi

status apakah ID *node* yang membalas sudah sesuai dengan ID *node* yang dituju. Jika benar, maka *node master* akan *me-request* data hasil pengukuran sensor pada *node destination* tersebut dengan menggunakan paket data RTD melalui rute yang sama seperti rute paket data RTS tadi. Lalu *node master* akan menerima paket data STD yang berisi data hasil pengukuran sensor pada *node destination* yang *di-request*.

#### 3.1.1.2 Node Slave

Pada Proyek Akhir ini, semua *node slave* terdiri dari sebuah mikrokontroler ATmega8535, 1 buah XBee Pro beserta rangkaian *drivernya*, sensor SHT11, dan LCD 16x2.

*Flowchart* sistem dari *node master* dapat dilihat pada gambar 5 berikut.



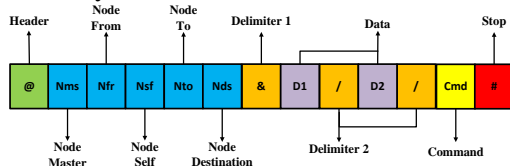
**Gambar 5** Flowchart sistem node slave

Pada *flowchart* Gambar 5, *node slave* akan menerima paket data RTS dari *node master*, jika *node slave* tersebut bukan *node destination*, maka *node slave* tersebut akan meneruskan paket data RTS kepada *node slave* yang dituju. Tetapi jika paket data tersebut memang ditujukan untuk *node slave* tersebut (*node slave* tersebut adalah *node destination*) maka *node slave* tersebut akan mengirimkan balasan berupa paket data CTS menuju *node master*. Kemudian *node destination* akan menerima paket data RTD yang berasal dari *node master*. Setelah itu *node destination* akan merespon dengan membalas paket tersebut dengan paket data STD yang berisi data hasil pengukuran dari *node destination* tersebut.

### 3.1.2 Pembuatan Protokol

Dalam sebuah sistem komunikasi, dibutuhkan adanya sebuah protokol untuk mengatur format antar perangkat yang akan dikomunikasikan, ini bertujuan agar paket data yang diterima pada *Administrator* tidak terjadi kerancuan. Protokol pada tugas akhir ini dirancang menggunakan kode ASCII.

Berikut ini adalah rancangan format protokolnya :



**Gambar 6** Format protokol

Keterangan :

- @ = Header atau awalan protokol.
- Nms = Node Master. Yaitu ID node sumber yang me-request (*nodemaster*).
- Nfr = Node From. Merupakan ID node yang me-request langsung.
- Nsf = Node Self. Merupakan ID node itu sendiri.
- Nto = Node To. Yaitu ID node yang akan dilewati/dilompati menuju *node destination*.
- Nds = Node Destination. Merupakan ID node tujuan yang *direquest*.
- & = Delimiter 1. Batasan antara rute dengan data pengukuran.
- / = Delimiter 2. Batasan antar data pengukuran dan dengan *Command*.
- D1 = Data hasil pengukuran suhu udara.
- D2 = Data hasil pengukuran kelembaban udara.
- Cmd = Command. Yaitu perintah, apakah paket data tersebut berisi *request*, balasan, atau isi data hasil pengukuran. Q = Request; B = Balasan; D = Isi Data
- # = Stop. Adalah akhiran atau penutup protokol.

### 3.1.3 Pembuatan Algoritma Multihop

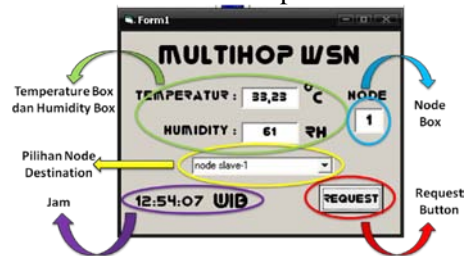
Untuk awal, XBee Pro di-*setting* sesuai alamat ID *node* urutan selanjutnya. Berikut ini tampilan *setting* XBee Pro menggunakan *software X-CTU*.

Ketika sistem melakukan komunikasi multihop, maka tiap-tiap modul XBee Pro pada tiap-tiap *node* harus mampu mengganti alamat ID *node* tujuan sesuai rute yang di-*routing*-kan

oleh *node master*. Agar XBee Pro dapat mengganti alamat secara otomatis, maka mikrokontroler pada *node* tersebut yang akan men-*setting* ATDL (alamat ID *node* tujuan).

### 3.1.4 Pembuatan Graphical User Interface

Untuk mempermudah Administrator dalam melakukan *request* atau dalam menggunakan sistem komunikasi multihop ini, maka dibuat sebuah GUI yang berbasis *Visual Basic (VB)*. Dimana tampilan ini terdapat *combo box* untuk memilih *node* mana yang akan di-*request* dan *button Request* yang berfungsi untuk memulai *request*. Kemudian data hasil pengukuran berupa nilai suhu dan nilai kelembaban ditampilkan pada dua buah *box*. Serta *node box* yang mempresentasikan *node slave* berapa yang menjadi *node destination*. Berikut ini tampilan dari GUI :



**Gambar 7** Tampilan GUI

## 3.2 PENGUJIAN SISTEM, HASIL DAN PEMBAHASAN

### 3.2.1 Pengujian Baudrate Komunikasi

**Tabel 2** Hasil pengujian sistem menggunakan *baudrate* 9600

No.	Jarak	Paket Data yang diterima <i>Node master</i>	
		Paket CTS	Paket STD
1.	1 meter	@11000&dddd/dd B#	@11000&31.55/72/D#
2.	5 meter	@11000&dddd/dd B#	@11000&31.42/73/D#
3.	10 meter	@11000&dddd/dd B#	@11000&31.48/773/D#

**Tabel 3** Hasil pengujian sistem menggunakan *baudrate* 4800

No.	Jarak	Paket Data yang diterima <i>Node master</i>	
		Paket CTS	Paket STD
1.	1 meter	-	-
2.	5 meter	-	-
3.	10 meter	-	-

**Tabel 4** Hasil pengujian sistem menggunakan *baudrate* 19200

No.	Jarak	Paket Data yang diterima <i>Node master</i>	
		Paket CTS	Paket STD
1.	1 meter	-	-
2.	5 meter	-	-
3.	10 meter	-	-

### Analisa

Dari hasil pengujian sistem dengan menggunakan *baudrate* yang berbeda-beda dapat dianalisa bahwa hanya *baudrate* 9600 yang dapat digunakan untuk komunikasi antara *node master* dengan *node slave* dengan tingkat keberhasilan 100%. Sedangkan untuk *baudrate* 4800 dan *baudrate* 19200, tidak ada satupun paket data CTS maupun STD yang diterima oleh *node master*. Hal ini dapat dikarenakan pengaruh perbandingan panjang bit paket data yang digunakan dengan *baudrate* yang digunakan. Jadi, untuk paket data dengan panjang 18 karakter (8 bit), penggunaan *baudrate* yang optimal untuk komunikasi antara *node master* dengan *node slave* adalah sebesar 9600.

### 3.2.2 Pengujian Jangkauan Maksimum Modul Transceiver

**Tabel 5** Hasil pengujian jangkauan maksimum modul *transceiver* pada kondisi LOS

Jarak (meter)	Paket Data yang diterima <i>Node master</i>		Delay (detik)
	Paket CTS	Paket STD	
5 m	@11000&dddd dd B#	@11000&31.78 73 D#	9
10 m	@11000&dddd dd B#	@11000&31.65 73 D#	9
20 m	@11000&dddd dd B#	@11000&31.66 72 D#	9
30 m	@11000&dddd dd B#	@11000&31.57 73 D#	9
40 m	@11000&dddd dd B#	@11000&31.78 72 D#	9
50 m	@11000&dddd dd B#	@11000&31.53 72 D#	9
60 m	@11000&dddd dd B#	@11000&31.69 72 D#	10

**Tabel 6** Hasil pengujian jangkauan maksimum modul *transceiver* pada kondisi NLOS

Posisi <i>Node Slave</i>	Paket Data yang diterima <i>Node master</i>		Delay (detik)
	Paket CTS	Paket STD	
R.Chamber	@11000&dddd dd B#	@11000&31.78 73 D#	9
Dapur	@11000&dddd dd B#	@11000&31.65 73 D#	10
Depan Kelas B-301	-	-	-

### Analisa

Dari hasil pengujian sistem pada kondisi LOS diperoleh jangkauan maksimum dengan jarak antara *node master* dengan *node slave* sejauh 15 meter. Sedangkan hasil pengujian sistem pada kondisi NLOS diperoleh jangkauan maksimum sejauh 10 meter. Hal ini dapat dipengaruhi oleh jenis material *obstacle*, ketebalan dan ketinggian *obstacle*.

### 3.2.3 Pengujian Komunikasi Multihop dengan Posisi *Node Slave* Urut



**Gambar 8** Posisi *node master* dan *node slave* dengan *node slave* 1 sebagai *node destination*

**Tabel 7** Hasil pengujian sistem multihop dengan *node slave* 1 sebagai *node destination*

No.	Paket Data yang diterima <i>Node master</i>		Delay (detik)
	Paket CTS	Paket STD	
1.	@11000&dddd dd B#	@11000&31.55 77 D#	10
2.	@11000&dddd dd B#	@11000&31.42 78 D#	9
3.	@11000&dddd dd B#	@11000&31.44 77 D#	9
4.	@11000&dddd dd B#	@11000&31.55 77 D#	9
5.	@11000&dddd dd B#	@11000&31.42 78 D#	9
6.	@11000&dddd dd B#	@11000&31.48 77 D#	9
7.	@11000&dddd dd B#	@11000&31.55 77 D#	9
8.	@11000&dddd dd B#	@11000&31.42 78 D#	10
9.	@11000&dddd dd B#	@11000&31.48 77 D#	9
10.	@11000&dddd dd B#	@11000&31.48 77 D#	9



**Gambar 9** Posisi *node master* dan *node slave* dengan *node slave* 2 sebagai *node destination*

**Tabel 8** Hasil pengujian sistem multihop dengan *node slave* 2 sebagai *node destination*

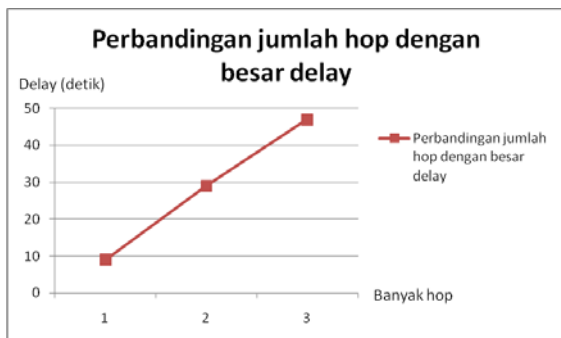
No.	Paket Data yang diterima <i>Node master</i>		Delay (detik)
	Paket CTS	Paket STD	
1.	@21000&dddd dd B#	@21000&31.55 77 D#	30
2.	@21000&dddd dd B#	@21000&31.42 78 D#	29
3.	@21000&dddd dd B#	@21000&31.47 7 D#	29
4.	@21000&dddd dd B#	@21000&31.55 77 D#	29
5.	@21000&dddd dd B#	@21000&31.42 78 D#	29
6.	@21000&dddd dd B#	@21000&31.48 77 D#	29
7.	@21000&dddd dd B#	@21000&31.55 77 D#	29
8.	@21000&dddd dd B#	@21000&31.42 78 D#	29
9.	@21000&dddd dd B#	@21000&31.48 77 D#	29
10.	@21000&dddd dd B#	@21000&31.48 77 D#	29



**Gambar 10** Posisi *node master* dan *node slave* dengan *node slave* 3 sebagai *node destination*

**Tabel 9** Hasil pengujian sistem Hasil pengujian sistem multihop dengan *node slave* 3 sebagai *node destination*

No.	Paket Data yang diterima <i>Node master</i>		Delay (detik)
	Paket CTS	Paket STD	
1.	@31000&dddd dd B#	@31000&31.55 77 D#	44
2.	@31000&dddd dd B#	@31000&31.42 78 D#	48
3.	@31000&dddd dd B#	@31000&31.48 77 D#	48
4.	@31000&dddd dd B#	@31000&31.55 77 D#	48
5.	@31000&dddd dd B#	@31000&31.42 78 D#	48
6.	@31000&dddd dd B#	@31000&31.48 77 D#	47
7.	@31000&dddd dd B#	@31000&31.55 77 D#	47
8.	@31000&dddd dd B#	@31000&31.42 7 D#	47
9.	@31000&dddd dd B#	@31000&31.46 77 D#	47
10.	@31000&dddd dd B#	@31000&31.48 77 D#	48



**Gambar 11** Grafik perbandingan jumlah hop dengan besar *delay* pengiriman



**Gambar 12** Grafik perbandingan jumlah hop dengan banyaknya paket data yang *loss*

#### Analisa

Dari hasil pengujian sistem dengan posisi *node slave* acak diperoleh hasil yang kurang optimal dengan tingkat keberhasilan 96,5%. Ini dikarenakan terjadinya kesalahan transmisi paket data ketika proses komunikasi antar node. Karena memang seharusnya dalam *men-deploy node-node* posisi dan letak harus diperhatikan untuk memaksimalkan kinerja sistem.

#### 4.1 KESIMPULAN

Setelah melakukan tahap perencanaan dan pembuatan sistem yang kemudian dilanjutkan dengan tahap pengujian dan analisa maka dapat diambil kesimpulan sebagai berikut :

- *Baudrate* yang paling maksimal untuk sistem komunikasi multihop adalah 9600, dimana tingkat keberhasilannya 100% atau sangat optimal.
- Topografi area yang dimonitoring sangat berpengaruh terhadap kinerja sistem, dimana pada kondisi area yang terdapat banyak *obstacle* (NLOS) paket data tidak bisa atau sulit untuk dikirimkan. Topografi

area juga sangat mempengaruhi jangkauan maksimum dari modul *transceiver*.

- Untuk komunikasi paket data secara multihop pada proyek akhir ini dapat dikatakan berhasil, dimana tingkat keberhasilannya sebesar 98%, dimana hampir semua paket data sukses dikirim dan diterima oleh tiap-tiap node, baik itu double hop ataupun hingga triple hop.
- Semakin banyak node atau lompatan, maka delay waktu pengiriman paket data juga akan semakin besar dan kemungkinan paket data yang *loss* juga semakin besar.
- Posisi dan letak dari tiap-tiap *node* sangat berpengaruh terhadap kinerja sistem.

#### DAFTAR PUSTAKA

- [1] Feng Zhao, Leonidas Guibas, "Wireless Sensor Network", Elsevier, San Fransisco, 2004.
- [2] Heri Andrianto, "Pemrograman Mikrokontroler AVR ATMega16 Menggunakan Bahasa C (CodeVision AVR)", Informatika, Bandung, 2008.
- [3] Gilang Kharisma, "Perancangan dan Pembuatan Perangkat Keras untuk Pemantau Kondisi Hutan dengan Sistem Komunikasi 802.15.4 (ZigBee)", Proyek Akhir, Teknik Telekomunikasi Politeknik Elektronika Negeri Surabaya – Institut Teknologi Sepuluh Nopember, Surabaya, 2009.
- [4] Hariski priyo Sangadi, "Algoritma Multihop dan Pencarian Rute otomatis untuk Sistem Komunikasi pada Monitoring Daya Listrik Secara Wireless", Proyek Akhir, Teknik Elektronika Politeknik Elektronika Negeri Surabaya – Institut Teknologi Sepuluh Nopember, Surabaya, 2008.
- [5] "Datasheet ATMega162", diakses 28 Februari 2010, dari Atmel [www.atmel.com/atmel/acrobat/doc2513.pdf](http://www.atmel.com/atmel/acrobat/doc2513.pdf)
- [6] "Datasheet ATMega162", diakses 28 Februari 2010, dari Atmel [www.atmel.com/atmel/acrobat/doc2502.pdf](http://www.atmel.com/atmel/acrobat/doc2502.pdf)
- [7] "Datasheet SHT11", diakses pada 10 Maret 2010, dari Sensirion [www.sensirion.com/de/pdf/product.../D atasheet-humidity-sensor-SHT1x.pdf](http://www.sensirion.com/de/pdf/product.../D atasheet-humidity-sensor-SHT1x.pdf)