

Sistem Monitoring Lingkungan dengan Jaringan Sensor Nirkabel

Lucky Karunia Setyawan Pratama¹, Taufiqurrahman², Wahjoe Tjatur Sesulihatien³, Bambang Sumantri⁴

¹Penulis, Mahasiswa Jurusan Teknik Elektronika PENS - ITS

²Dosen Pembimbing, Staf Pengajar di Jurusan Teknik Elektronika PENS - ITS
Politeknik Elektronika Negeri Surabaya

Electronics Engineering Polytechnic Institute of Surabaya

Institut Teknologi Sepuluh Nopember Surabaya

Kampus ITS Sukolilo, Surabaya 60111, INDONESIA

Tel: +62 (31) 594 7280; Fax: +62 (31) 594 6114

email : ustonese@student.eepis-its.edu

taufiq@eepis-its.edu

wahjoe@eepis-its.edu

bambang@eepis-its.edu

Abstrak— *Monitoring* lingkungan saat ini sudah bergerak ke era *wireless*. Sistem *wireless* akan bermasalah jika tidak dilakukan manajemen komunikasi yang benar. Sehingga jika tidak ada data yang masuk, maka data yang direkam akhirnya didapat dengan nilai rata-rata. Jadi, data yang direkam menjadi tidak valid atau data tersebut hanyalah data buatan. Oleh karenanya dalam proyek akhir kali ini, dilakukan manajemen komunikasi dengan mengadopsi sistem DSR (*Dynamic Source Routing*). Dimana, dalam sistem ini komunikasi dapat dilakukan dengan baik, secara singlehop maupun multihop. Dan dalam sistem ini juga dapat melakukan berbagai rute atau *rule* sehingga komunikasi tetap terjalin antar *node sensor* maupun *sink* dengan *node sensor*. Dalam komunikasi ini, informasi yang dikirim adalah data informasi suhu, kelembaban dan gas CO₂. Dimana suhu dan kelembaban diambil dari sensor SHT11 sedangkan untuk gas CO₂ diambil dari sensor TGS4160. Data – data tersebut ditampilkan pada HMI (*Human Machine Interface*) sehingga *user* mengetahui suhu, kelembaban dan kadar CO₂ pada tiap *node sensor*. Komunikasi yang dilakukan secara *wireless* ini digunakan perangkat XBEE. Dengan adanya sistem komunikasi multihop ditambah dengan mengadopsi sistem DSR, maka tingkat keberhasilan dalam komunikasi adalah 100% serta data yang ditampilkan pada HMI berhasil ditampilkan dan disimpan ke dalam *database* dengan tingkat keberhasilan 100%. Dimana tingkat keberhasilan komunikasi sebesar 100% tersebut pada sisi server, *sink*, maupun pada sisi *node sensor*. Keberhasilan komunikasi tersebut juga didukung dengan adanya pengecekan eror, yaitu dengan pengecekan nilai *FCS* sebagai bit-bit pembanding benar tidaknya suatu data yang dikirim dengan yang diterima. Dalam proyek akhir ini, semua data yang diterima dan yang dikirim selalu sama sehingga bit-bit *FCS* tidak ada yang eror. Oleh karena itu, komunikasi selalu berjalan dengan baik dengan batasan jarak kurang dari 100 meter.

Kata kunci - Jaringan sensor, jaringan sensor nirkabel, *monitoring* lingkungan, xbee, multihop.

1. PENDAHULUAN

Pada Jaringan Sensor tersebut terdiri dari beberapa *node* yang saling terintegrasi dimana tiap – tiap *node* terdiri dari beberapa sensor. Sensor – sensor tersebut telah ditetapkan sebelumnya sesuai dengan penggunaan.

Jaringan Sensor Nirkabel adalah jaringan perangkat spasial terdistribusi kecil yang dapat berkomunikasi satu sama lain melalui udara. "Ini adalah bagian terpenting kedua setelah jaringan internet komputer" (Dikutip dari forum Industri Jaringan Sensor Internasional). Kinerja yang baik seperti konsumsi daya rendah dan jaringan yang handal mengorganisir diri telah memungkinkan untuk digunakan sebagai modus utama untuk pemantauan dan pengumpulan data di lingkungan sebenarnya.

Yang dimaksud dengan Sistem Monitoring Lingkungan disini yaitu sistem dimana bekerja untuk mendapatkan data – data lingkungan dari sensor – sensor pada tiap – tiap *node* tersebut. Data – data yang diambil tersebut pun telah dikhususkan terhadap suatu masalah. Dalam hal ini adalah tentang peringatan dini kebakaran hutan. Sehingga sistem dapat mendeteksi akan adanya kebakaran.

Sensor – sensor yang digunakan antara lain :

1. Sensor suhu dan kelembaban
2. Sensor gas CO₂

Untuk Sensor suhu dan kelembaban menggunakan sensor SHT 11, Sensor Gas CO₂ menggunakan TGS 4160.

Komunikasi pada Jaringan Sensor ini menggunakan topologi *mesh* secara multihop. Dimana akan terjadi komunikasi antar *node* nantinya. Hal tersebut difungsikan sebagai proses *multihop*.

2. PERUMUSAN MASALAH

Berdasarkan latar belakang di atas, maka ditemukan beberapa rumusan masalah yaitu :

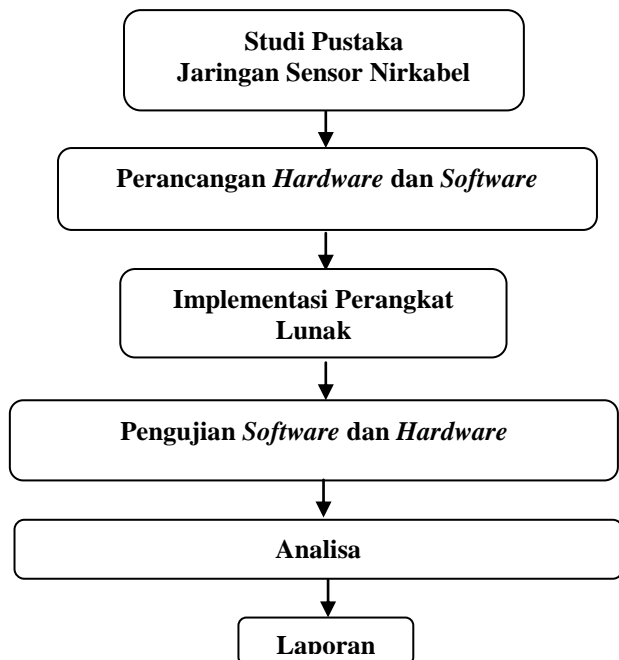
1. Bagaimana menentukan protokol komunikasi data.[8]
2. Bagaimana mendapatkan data.[9]
3. Bagaimana menentukan rule komunikasi multihop.[7]
4. Bagaimana melakukan pengolahan data.[14]
5. Bagaimana menampilkan data dan menyimpan data.[12]
6. Bagaimana menentukan waktu delay pengiriman data.[7]

Adapun beberapa hal yang membatasi pembuatan aplikasi dalam proyek akhir ini, antara lain :

1. Jumlah *sink* hanya 1.
2. Jumlah *node* sebanyak 3 *node*.
3. Jarak antar node idealnya kurang dari atau sama dengan 100 meter.[6]
4. Tidak ada router sebagai pemisah blok jaringan sensor karena hanya menggunakan 1 blok jaringan sensor.
5. *Server* hanya satu.
6. Metode DSR yang digunakan hanya terbatas pada penggunaan *cache rule*.[9]

3. METODOLOGI

Target akhir dari proyek akhir ini adalah mendapatkan data – data dari sensor dengan hasil yang sesuai dan dapat ditampilkan pada GUI sebagai tindakan monitoring. Langkah-langkah pembuatan proyek akhir ini dapat didefinisikan sebagai berikut :



Gambar 1. Blok Diagram tahapan metode penelitian

3.1. Tinjauan Pustaka

3.1.1. Environmental Sensor

a. Sensor suhu dan kelembapan[1]

Sensor yang digunakan untuk mendeteksi suhu dan kelembapan lingkungan adalah SHT 11. Dimana SHT 11 dapat mendeteksi suhu dan kelembapan dengan kondisi maksimum bekerja pada 120 derajat celcius serta bekerja normal pada suhu -20°C – 100°C . SHT 11 ini memiliki tingkat kebenaran yang cukup bagus dengan persentase error :

1. Untuk pembacaan data kelembapan \rightarrow $\pm 3\%$ rH (relative humidity).
2. Untuk pembacaan data suhu \rightarrow $\pm 0.4^{\circ}\text{C}$.

SHT 11 ini bekerja pada tegangan *typical* 3.3 v namun dapat pula bekerja pada tegangan minimal 2.4 volt dan maksimal 5.5 volt.



Gambar 2. Sensor SHT11[1]

b. Sensor gas CO2[2]

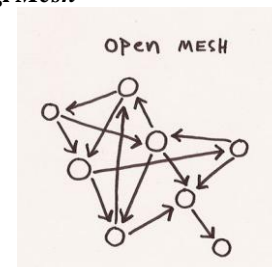
Sensor gas CO2 yang digunakan adalah TGS 4160. Dimana TGS 4160 adalah sebuah sensor yang dibentuk dari element karbon dioksida yang sensitif dan sebuah thermistor. Range pendeteksiannya lebar yaitu 350 – 50.000ppm karbon dioksida. Sensor TGS 4160 ini dapat dikatakan dapat bekerja normal dengan perubahan nilai kelembapan. Yang perlu diingat juga bahwa TGS 4160 ini butuh pemanasan terlebih dahulu, dengan maksud mempertahankan elemen penginderaan di suhu yang optimal untuk penginderaan.



Gambar 3. Sensor Figaro TGS 4160[2]

3.1.2. Jaringan Sensor[3]

a. Topologi Mesh

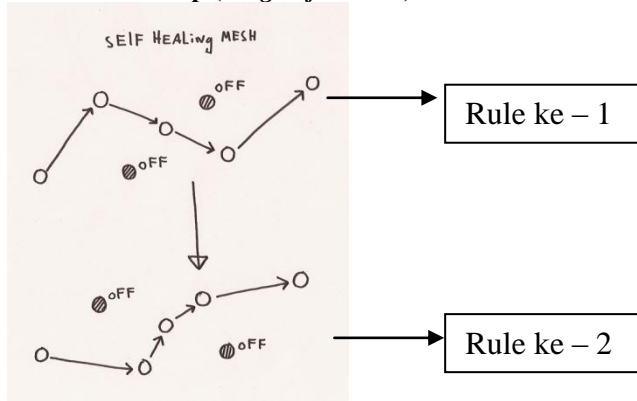


Gambar 4. Topologi Mesh[3]

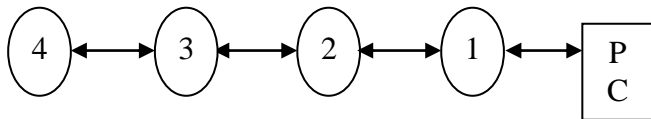
Gambar di atas merupakan Gambaran umum dari topologi *mesh*. Dimana dengan topologi *mesh* tersebut, komunikasi

antar *node* dapat tercapai namun tetap sesuai dengan *rule* yang telah dibuat. Komunikasi dengan topologi *mesh* tersebut memang membutuhkan asupan daya yang lebih. Oleh karena itu, jarak antar *node* tidak boleh terlalu jauh. Kali ini jarak maksimal antar *node* ditentukan kurang dari 100 meter (kondisi standar).

b. Multihop (dengan fix route)



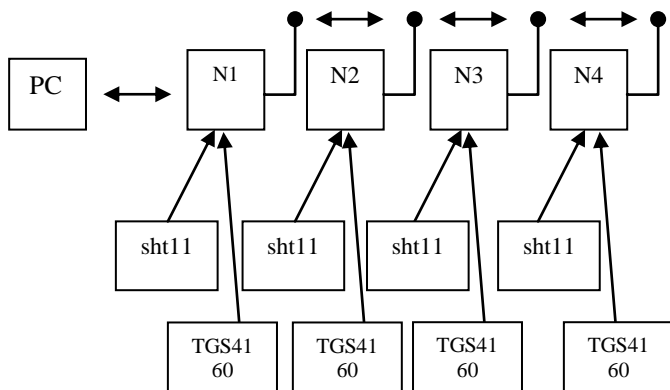
Gambar 5. Salah satu contoh multihop[3]



Gambar 6. Jalur komunikasi multihop

Seperti Gambar di atas bahwa komunikasi secara multihop ini dilakukan secara dinamis, maksudnya adalah bila *rule* ke - 1 gagal maka terdapat *rule* ke - 2 dan seterusnya. Sehingga server dapat menerima data dengan berbagai jalan / *rule*.

3.2. Perencanaan Sistem



Gambar 7. Salah satu rule multihop.

Dari disain di atas maka terjadi proses multihop.

3. Mekanisme kerja sistem

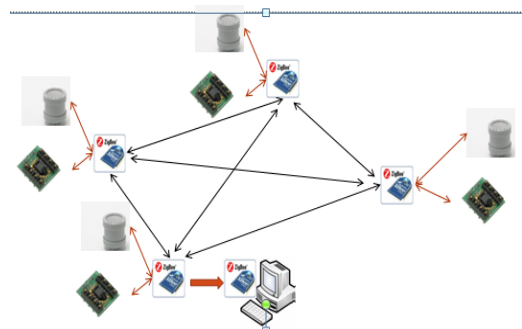
Pengiriman data dilakukan bila ada permintaan dari server. Dimana pada *node* - *node* tertentu (yang tidak terlihat / jauh dari server), pengiriman datanya dilakukan secara estafet

(multi-point). Server menerima data - data tersebut yang kemudian akan diolah (di-parsing) sehingga server dapat menterjemahkan data tersebut dari *node* ke-berapa dan data apa yang dikirimkan tersebut. Hasil pemrosesan data tersebut ditampilkan pada GUI sehingga dapat terlihat secara keseluruhan data yang diterima oleh server (monitoring). Dengan data yang diterima server (PC) tersebut, server dapat menentukan tindakan bahwa akan ada kebakaran atau tidak.

3.2. Perangkat Lunak (Software)

Software yang digunakan untuk *user interface* adalah QT. Sistem monitoringnya yaitu *by request* oleh *mikroserver* (*hardware* yang berfungsi sebagai server atau *sink*) untuk menghindari *data collision* (tabrakan data). Oleh karenanya dalam sistem ini tidak digunakan *system autosend*. Karena pada *system autosend* tidak Ada kontrol pada setiap *node sensor*.

Dari *mikroserver* tersebut berlanjut ke *Server* (PC). Dimana pada *server* data akan dimonitoring atau ditampilkan dan disimpan dalam *database*.



Gambar 8. Design sistem keseluruhan

3.3. Pembuatan Protokol

mikroserver ke server meliputi pengecekan kondisi *node* (aktif / non aktif) dan akuisisi data yang akan ditransmisikan. Pada proses ini komunikasi data menggunakan protokol yang disesuaikan dengan kebutuhan yang diinginkan (apakah untuk pengecekan kondisi *node* atau transmisi hasil akuisisi data sensor).

3.3.1. Protokol Sensor / Node – Mikroserver / SINK

1. Protokol pengecekan *node*

Protokol ini digunakan untuk mengetahui kondisi sensor apakah aktif / tidak untuk menentukan proses berikutnya.

□ *Format request* Protokol

Tabel 1. Protokol *request* data

Simbol	Keterangan	Isi	Banyak (karakter)
Del	<i>Delimiter(header)</i>	#	1

MY	Alamat asal	MY dari asal 1-4	1
DL	Alamat tujuan	MY dari tujuan 1-4	1
NH1	Hope 1	MY dari Hope 1-4	1
NH2	Hope 2	MY dari Hope 1-4	1
NH3	Hope 3	MY dari Hope 1-4	1
NH4	Hope 4	MY dari Hope 1-4	1
Hn	Banyak Hope	Total Hope 1-4	2
ID	Identitas node	MY dari pengirim 1-4	2
R	Rule	Rule 1-6	2
D	D untuk request data.	D	1
F	Nilai FCS dari paket data	2 digit FCS	2
T	<i>Terminator(tail)</i>	!	1

Tabel di atas merupakan table untuk *request* data *node*. Protokol tersebut dipakai untuk seluruh *node* tanpa terkecuali. Jadi, protokol tersebut menjadi fleksible untuk dipakai oleh *node*.

Tabel 2. Protokol kirim data

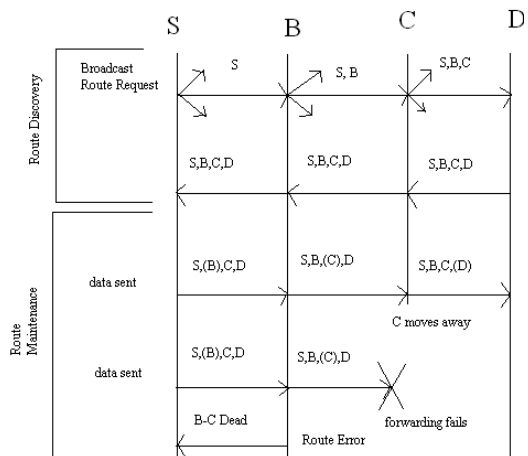
Simbol	Keterangan	Isi	Banyak (karakter)
Del	<i>Delimiter(header)</i>	#	1
MY	Alamat asal	MY dari asal 1-4	1
DL	Alamat tujuan	MY dari tujuan 1-4	1
NH1	Hope 1	MY dari Hope 1-4	1
NH2	Hope 2	MY dari Hope 1-4	1
NH3	Hope 3	MY dari Hope 1-4	1
NH4	Hope 4	MY dari Hope 1-4	1
Hn	Banyak Hope	Total Hope 1-4	2

ID	Identitas node	MY dari pengirim 1-4	2
R	Rule	Rule 1-6	2
D	D untuk request data.	D	1
S	Data suhu	3 digit data suhu	3
H	Data kelembaban	3 digit data kelembaban	3
G	Data gas CO ₂	4 digit data gas CO ₂	4
F	Nilai FCS dari paket data	2 digit FCS	2
T	<i>Terminator(tail)</i>	!	1

Tabel di atas merupakan protokol yang digunakan untuk menirinkan data *node*. Seperti halnya protokol *request* data *node*, protokol untuk mengirim data juga menjadi fleksible untuk digunakan oleh *node* untuk mengirim data.

Dengan pembuatan protokol seperti di atas membuat komunikasi lebih fleksibel.

3.3.2. Sistem Komunikasi



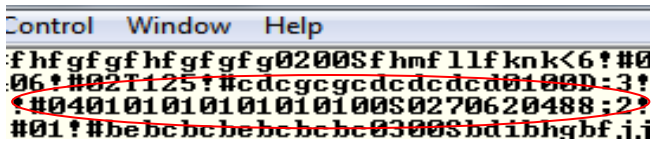
Gambar 9. Sistem komunikasi dengan DSR (*Dynamic Source Routing*)

Gambar di atas merupakan jalannya system komunikasi dengan menggunakan DSR. Namun pada proyek kali ini, yang digunakan adalah fitur *route cache*. Maksudnya adalah pada *sink* sudah tertanam 7 rule untuk mencapai 3 *node* yang tersedia.

4. PENGUJIAN DAN ANALISA

4.1. Sensor

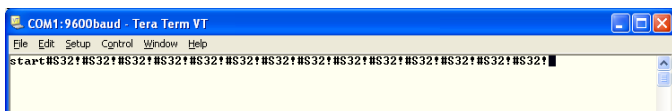
Pengujian yang dilakukan adalah pembacaan data sensor suhu dan kelembaban, CO₂. Dimana pembacaan sht11 dengan menggunakan *2wire serial interface*. Sedangkan untuk sensor gas CO₂ pembacaannya dengan ADC.



Gambar 10. Hasil balasan dari request data sensor

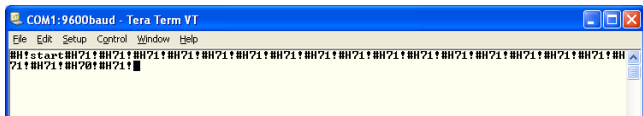
4.1.1. Pembacaan Suhu dan Kelembaban

Walaupun pembacaan sensor ini dengan menggunakan *2 wire*, pembacaan sensor ini tidak bisa menggunakan sistem I2C. Pada Gambar di bawah ini merupakan pengambilan data suhu (sudah terpakatkan dalam komunikasi wireless).



Gambar 10. Pengambilan data suhu

Pengambilan data suhu dengan cara mengirimkan data serial *#S!* dari PC atau GUI. Maka pada sisi mikro (*node* atau *sink*) akan memberi balasan ,contoh: *#S32!* , dimana *32* merupakan data suhu saat ini yang di-*sensing* oleh SHT11.

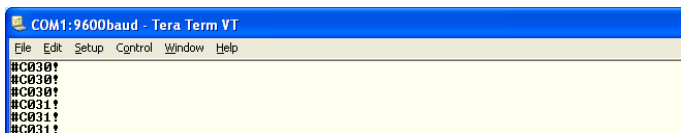


Gambar 11. Pengambilan data kelembaban

Pada Gambar di atas ini merupakan Gambar pengambilan data kelembaban. Pengambilan data kelembaban dilakukan dengan cara mengirimkan data serial *#H!* dari PC atau GUI. Maka pada sisi mikro (*node* atau *sink*) akan memberi balasan, contoh *#H71!*, dimana *71* merupakan data kelembaban.

4.1.2. Pembacaan Gas CO₂

Pembacaan sensor CO₂ dengan membaca nilai tegangan output pada modul cdm4160-L00.

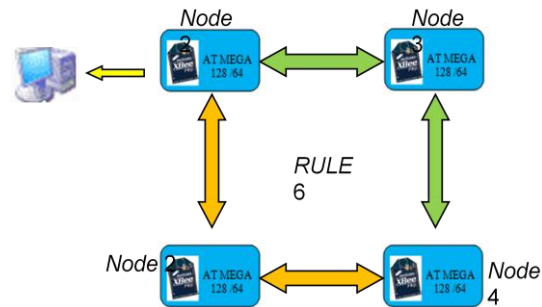


Gambar 12. Pengambilan dataCO₂

Gambar di atas ini merupakan Gambar pengambilan data ADC dari sensor CO₂. Pengambilan data ADCnya dengan mengirimkan data serial ,contoh *#C!*. Maka sisi mikro akan membalas dengan ,contoh: *#C031!* , dimana *31* merupakan nilai ADC.

Protokol yang dipakai sesuai dengan apa yang direncanakan. Dan hasilnya adalah seperti pada Gambar di atas. Pada pengiriman data kali ini menggunakan konfigurasi Xbee secara *broadcast*, dengan kata lain semua xbee menerima data dan data tersebut diolah oleh mikrokontroler dengan cara *mem-parsing*.

4.1.3. Uji coba komunikasi

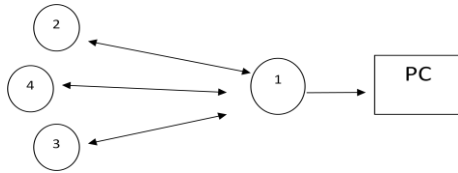


Gambar 13. Uji coba komunikasi

Pada gambar di atas yang dpkai adalah *rule 6* karena jarak *pada* sink mengabsen seluruh *node*, *node 4* tidak hadir. Sebenarnya *rule* yang dipakai juga dapat menggunakan *rule 5*. Namun dalam sistem ini telah dibotkan pada *rule 6* jika *rule 5* dan *rule6* memungkinkan untuk dipakai.

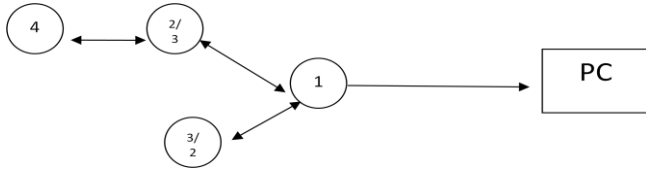
Kemudian jika terjadi eror atau tidak dapat menerima data maka *rule* akan berubah sesuai dengan Tabel di bawah ini.

Tabel 3. Rule recovery



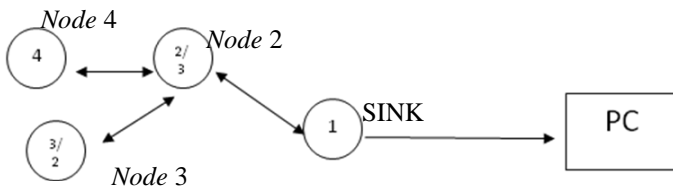
Gambar 14. Rule 0

Huruf X pada tabel atas adalah dianggap segala kondisi. Jadi, segala node yang aktif kecuali yang *error* dalam hal ini.



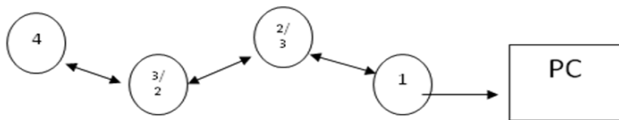
Gambar 15. Rule 5 / 6

Pada gambar 14 dikatakan *rule 5* jika id *node 2* = 2 dan id *node 3* = 3. Sedangkan dikatakan *rule 5* jika id *node 2* = 3 dan id *node 3* = 2.



Gambar 16. Rule 1 / 2.

Pada gambar 15 dikatakan *rule 1* jika id *node 2* = 2 dan id *node 3* = 3. Sedangkan dikatakan *rule 2* jika id *node 2* = 3 dan id *node 3* = 2.



Gambar 17. Rule 3 / 4.

Pada gambar 16 dikatakan *rule 3* jika id *node 2* = 2 dan id *node 3* = 3. Sedangkan dikatakan *rule 4* jika id *node 2* = 3 dan id *node 3* = 2.

5. KESIMPULAN

Setelah melakukan tahap perancangan dan pembuatan sistem yang kemudian dilanjutkan dengan tahap pengujian dan analisa. Maka dapat diambil kesimpulan sebagai berikut :

1. Dengan menggunakan komunikasi multihop, *hidden node* dapat diketahui dan dapat dicari. Jadi, seluruh data *node* dapat diraih.
2. Dengan menggunakan reuest bergantian menghasilkan tidak adanya data yang *crash*.
3. Enkripsi data dengan menggunakan metode Caesar berhasil. Semua data dapat dienkripsi dan didekripsi dengan baik.
4. Menggunakan fitur *cache rule* pada metode DSR(*Dynamic Source Rouing*) ditambah dengan adanya

Current Rule	Node (n)error	Node acive	Next rule
0	2	X	6
	3	X	5
	2,4	X	6
	3,4	X	5
1	2	X	6
	4	2,3	3
	3	2,4	5
2	4,3	2	6
	3	X	5
	4	3,2	4
3	2	3,4	6
	4,2	3	5
	2	X	6
4	3	2	5
	4	2,3	5
	3	X	5
5	2	3	6
	4	3,2	6
	2	X	6
6	4	2,3	6
	4,3	2	3
	3	2,4	1
6	3	X	5
	4	3,2	5
	4,2	3	4
	2	3,4	2

acknowledge sangat berguna karena tidak aka nada data yang *crash*.

5. Dengan mengadopsi metode DSR, seluruh *routing* terstruktur dengan baik.

6. SARAN

1. Untuk tugas akhir selanjutnya, lebih baik menggunakan mode API dari XBEE karena itu akan membantu membentuk *routing* dengan mudah.
2. Untuk tugas akhir selanjutnya, pikirkan konsumsi daya yang digunakan.
3. *Full DSR* harus digunakan pada tugas akhir selanjutnya karena dengan menggunakan *full DSR*, komunikasi *sink* ke *node*, *node* ke *node*, dan *node* ke *sink* lebih fleksible.
4. Untuk tugas akhir selanjutnya, akan lebih baik untuk membagi wilayah *node - node* tersebut.

7. REFERENSI

1. www.sensirion.com/en/pdf/.../Datasheet-humidity-sensor-SHT1x.pdf
2. www.aesensors.nl/pdf/figaro/CO2GasSensorModuleCDM4161.pdf

3. <http://www.johnhenryshammer.com/WOW2/pagesHowTo/networkOverview.php#index>
4. www.atmel.com/atmel/acrobat/doc2467.pdf
5. http://www.atmel.com/dyn/resources/prod_documents/2490S.pdf
6. http://site.gridconnect.com/docs/MaxStream/XBee_Manual_GC.pdf
7. http://undergraduate.csse.uwa.edu.au/courses/CITS7211/materials/Barre_netxea-WSNcommunication101.pdf
8. http://lanoswww.epfl.ch/studinfo/courses/Dynamical_Networks/Miniprojects/Jerome_Rousselot/Routing_sensor_networks_pres.pdf
9. [http://www.bth.se/fou/cuppsats.nsf/all/cf795b6489e5bc9ec12576a500363a0e/\\$file/Thesis%20documentation.pdf](http://www.bth.se/fou/cuppsats.nsf/all/cf795b6489e5bc9ec12576a500363a0e/$file/Thesis%20documentation.pdf)
10. Hendhi Hermawan. "PENERAPAN TEKNOLOGI WIRELESS RF DAN SMS GATEWAY PADA SISTEM MONITORING PEMAKAIAN AIR PDAM SKALA RUMAH TANGGA YANG TERINTEGRASI DATABASE VIA INTERNET", Tugas akhir: T.Elektronika PENS-ITS:2009
11. http://www.comnets.uni-bremen.de/typo3site/uploads/media/Multi-Path-Performance_01.ppt
12. <http://www.johnhenryshammer.com/WOW2/pagesHowTo/xbeeSeries1.php>
13. CodeVisionAVR, help.
14. QT,help.
15. <http://www.qtforum.org>