

Rancang Bangun M2M (*Machine-to-Machine*) Communication Berbasis 6LoWPAN

Doni Pradana ^{*1}, Bakhtiar Alldino Ardi Sumbodo ²

¹ Prodi Elins. Departemen Ilmu Komputer dan Elektronika FMIPA UGM

² Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta
e-mail: ^{*1} donipradana29@gmail.com, ² b.alldino.as@mail.ugm.ac.id

Abstrak

Pada masa sekarang ini perkembangan teknologi menyediakan komunikasi tanpa menggunakan kabel yang disebut dengan teknologi nirkabel. Salah satu teknologi komunikasi nirkabel tersebut yaitu komunikasi M2M (*Machine to Machine*). Pada penelitian ini menggunakan protokol 6LoWPAN sebagai basis komunikasi M2M, dikarenakan pada protokol Zigbee ditemui kelemahan seperti nilai *end-to-end delay* dan *packet loss* yang lebih besar.

Pengerjaan penelitian ini adalah merancang sistem M2M communication berbasis 6LoWPAN dan membandingkan data hasil performansi dari protokol 6LoWPAN dengan protokol Zigbee dari segi *end-to-end delay* dan *packet loss* yang disertai dengan analisis terhadap sistem. Variasi pengujian adalah dengan mengatur baud rate Xbee 1200 bps, 4800 bps, 9600 bps, 19200 bps, 38400 bps, 57600 bps, 115200 bps. Selain baud rate, variasi juga dilakukan dengan mengatur jarak antar node 10 meter sampai dengan 60 dengan interval 10 meter. Waktu rata-rata *end-to-end delay* sebesar 1869 milidetik pada 6LoWPAN, sedangkan protokol Zigbee 422 milidetik. Dari segi *packet loss*, protokol 6LoWPAN tidak menyediakan *packet loss* dikarenakan penggunaan UDP memang tidak memiliki *acknowledgement* dan juga *sequence number* untuk melacak *packet loss*, sedangkan protokol Zigbee menghasilkan rata-rata *packet loss* sebesar 26%.

Kata kunci—6LoWPAN, Zigbee, Xbee, delay, packet loss

Abstract

At the present time of technological development provides a communication without using cable which are called wireless technology. One of wireless communication technology is M2M (*Machine to Machine*) communication. In this research, using a protocol 6LoWPAN as the basis of M2M communications, because Zigbee protocol has weakness such as the value of *end-to-end delay* and *packet loss* is greater than 6LoWPAN protocol.

Processing on this research is design M2M communication system based on 6LoWPAN protocol, and then compare 6LoWPAN protocol with Zigbee protocol in data performance aspect, terms of *end-to-end delay* and *packet loss*. Variations of testing is to set the baud rate Xbee 1200 bps, 4800 bps, 9600 bps, 19200 bps, 38400 bps, 57600 bps, 115200 bps. Beside of baud rate, variation also adjusting the distance between nodes from 10 meters up to 60 meters by 10 meter intervals. Average of *end-to-end delay* time is 1899 milliseconds on 6LoWPAN, while protocol Zigbee is 422 milliseconds. In 6LoWPAN protocol has not provided *packet loss*, because UDP not have an *acknowledgement* and *sequence number* to track *packet loss*, while average of *packet loss* Zigbee protocol is 26%.

Keywords—6LoWPAN, Zigbee, Xbee, delay, packet loss

1. PENDAHULUAN

Pada masa sekarang ini perkembangan teknologi memungkinkan layanan komunikasi tanpa menggunakan kabel yang disebut dengan teknologi nirkabel. Salah satu teknologi komunikasi nirkabel tersebut yaitu M2M (Machine to Machine) Communication. Komunikasi M2M itu sendiri merupakan komunikasi antar dua perangkat atau lebih yang memiliki kemampuan sama lalu terhubung satu sama lain [1]. Komunikasi M2M ini juga banyak diaplikasikan untuk pemantauan suhu, kelembapan, kesehatan, hingga pengendalian jarak jauh. Komunikasi pada perangkat M2M pada saat ini banyak menggunakan media komunikasi, salah satu media yang digunakan yaitu menggunakan standar komunikasi 802.15.4. Penggunaan protokol dalam komunikasi M2M ini juga sangat diperhatikan, mengingat pada pengoperasiannya dibutuhkan kinerja yang cepat dan handal. Protokol 6LoWPAN memberikan rata-rata *end-to-end delay* yang lebih kecil dari pada protokol Zigbee, dengan *end-to-end delay* 167 milidetik pada protokol 6LoWPAN dan 392,3 milidetik pada protokol Zigbee [2].

Maka dari itu, pada penelitian ini digunakanlah protokol 6LoWPAN (*IPv6 Low Power Personal Area Network*) yang dikatakan lebih baik kinerjanya dari protokol Zigbee. Kinerja yang dimaksud adalah *end-to-end delay* dan *packet loss*. [3] *End-to-end delay* adalah waktu yang diperlukan oleh suatu paket data yang berasal dari *source node* hingga mencapai *destination node*. [4] Sedangkan *packet loss* merupakan parameter yang menunjukkan jumlah total paket yang hilang. Mengingat protokol Zigbee dan protokol 6LoWPAN sama-sama menggunakan standar komunikasi 802.15.4 pada PHY dan MAC layer. 6LoWPAN merupakan suatu protokol yang digunakan sebagai standar integrasi IP pada jaringan IoT (*Internet of Things*) yang berdaya rendah [5]. IoT adalah interkoneksi dari perangkat yang teridentifikasi secara unik dalam keberadaan infrastruktur internet [6]. Melihat hal tersebut, penelitian ini akan ditunjukkan untuk mengimplementasikan protokol 6LoWPAN pada komunikasi M2M (*Machine to Machine*) dalam meningkatkan kinerja yaitu dalam segi *end-to-end delay* dan juga *packet loss*.

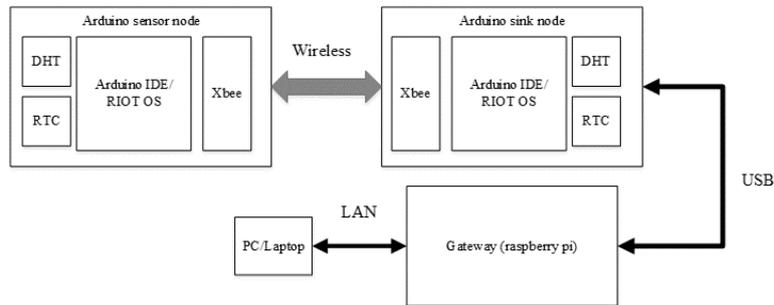
2. METODE PENELITIAN

2.1 Analisis Sistem

Sistem yang akan dibuat terdiri dari 2 bagian yaitu rancang bangun sistem perangkat keras dan sistem perangkat lunak. Rancangan perangkat keras dari *sensor node* dan juga *sink node* menggunakan Arduino Mega 2560 sebagai *microcontroller*, modul sensor DHT11 untuk membaca parameter fisik yang berupa suhu dan kelembapan, dan juga modul RTC (*Real Time Clock*) untuk menampilkan . Dimana data dari modul sensor DHT11 dan juga RTC digunakan untuk pembuatan *frame data*. Sebagai komponen utama dalam melakukan komunikasi yang menyediakan standar 802.15.4 sebagai media komunikasi dalam hal ini menggunakan Xbee Series 1 untuk penggunaan protokol 6LoWPAN dan Xbee Series 2 untuk penggunaan protokol Zigbee. Karakteristik dari 6LoWPAN adalah paket yang dikirim berukuran kecil sebesar 127 bytes, mendukung pengalamatan 16-bit atau IEEE 64-bit extended Media Access Control , bandwidth yang kecil sebesar 250 kbps dengan frekuensi berjalan pada rentan 2,4 GHz [7].

Proses kerja dari alat ini adalah pada kedua *node* yang terdiri dari *sensor node* dan juga *sink node* melakukan pembacaan parameter fisik yang berupa suhu dan juga kelembapan dengan menggunakan modul sensor DHT 11. Kedua *node* akan membuat *frame data*, pada *sink node* sama seperti pada *sensor node* tapi yang membedakan yaitu, pada *sink node* menerima paket data dari *sensor node* dan mengirimnya ke *gateway* dengan komunikasi serial USB untuk menampilkan *frame data* yang dikirim oleh *sensor node (client)* dan juga menampilkan *frame*

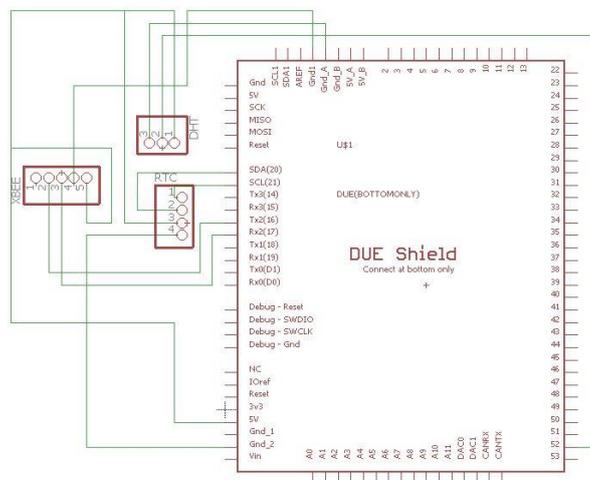
data yang dibuat oleh *sink node (server)* itu sendiri. Selanjutnya, hubungan *gateway* dengan PC menggunakan kabel LAN (RJ45) dan juga perangkat lunak Putty, untuk mengakses *gateway* menggunakan SSH (*Secure Shell*) sebagai perintah untuk mengakses *gateway*. Penggunaan dari *gateway* hanya dibutuhkan pada pengoperasian sistem tidak digunakan pada proses pengujian yang hanya menggunakan 2 buah node. Langsung menghubungkan *sink node* ke laptop. Rancangan sistem ditunjukkan pada diagram blok rancangan sistem pada Gambar 1.



Gambar 1 Diagram blok rancangan sistem

2.2 Rancangan Perangkat Keras

Rancangan perangkat keras dari *sensor node* dan juga *sink node* menggunakan Arduino Mega 2560 sebagai *microcontroller*, modul sensor DHT11 untuk membaca parameter fisik yang berupa suhu dan kelembapan, dan juga modul RTC (*Real Time Clock*) untuk menampilkan . Dimana data dari modul sensor DHT11 dan juga RTC digunakan untuk pembuatan *frame data*. Sebagai komponen utama dalam melakukan komunikasi yang menyediakan standar 802.15.4 sebagai media komunikasi dalam hal ini menggunakan Xbee Series 1 untuk penggunaan protokol 6LoWPAN dan Xbee Series 2 untuk penggunaan protokol Zigbee, kedua modul Xbee sama-sama dihubungkan dengan Xbee Adapter yang dihubungkan dengan pin RX, TX, vcc +5V, GND (*ground*). Semua komponen yang terdiri dari modul DHT 11, modul RTC, dan Xbee Adapter akan dihubungkan dengan pin mikrokontroler Arduino Mega 2560 dengan menggunakan pcb yang didesain seperti *shield*. Skematik pcb sesuai pada Gambar 2.



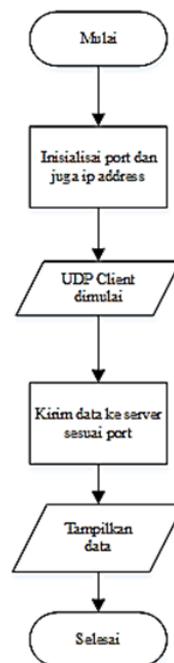
Gambar 2 Skematik pcb

2.3 Rancangan Perangkat Lunak

Rancangan perangkat lunak, meliputi pemrograman menggunakan bahasa C++ /C untuk membuat RIOT OS dan juga membuat program pada Arduino IDE yang akan ditanamkan ke

dalam Arduino *board*. Pada proses penanaman program pada mikrokontroler Arduino Mega 2560 menggunakan perintah *upload* pada Arduino IDE dan perintah *make flash all term* pada RIOT OS yang sebelumnya dilakukan *compile* menggunakan *compiler gcc*, kedua perangkat lunak tersebut di-*install* pada laptop yang menggunakan sistem operasi Ubuntu 14.10 LTS dan juga Windows 10. Kedua program tersebut akan ditanamkan secara bergantian sesuai dengan penggunaan untuk pengujian, jika ingin melakukan pengujian terhadap protokol Zigbee akan menanamkan program dengan Arduino IDE, jika ingin melakukan pengujian terhadap protokol 6LoWPAN akan menanamkan program RIOT OS. Membuat *frame data* yang menjadi acuan data yang dikirimkan, pengalamatan *node* menggunakan Ipv6 yang berjenis *unicast link-local address*. Selain itu, rancangan pada *gateway* untuk menerima data secara serial dengan USB dari *sink node* yang bertujuan untuk dapat diakses menggunakan perangkat komputer atau pada jaringan komputer. Penggunaan *gateway* sebagai akses terhadap serial monitor untuk memantau proses pengiriman dan penerimaan *frame data* pada laptop atau pada jaringan, peran *gateway* ini juga bisa dimaksud sebagai *ethernet shield* Arduino Mega 2650 untuk memungkinkan akses terhadap jaringan komputer dan juga PC (*Personal Computer*). Perangkat lunak yang terdapat pada *gateway* dalam hal ini adalah Raspberry Pi adalah sistem operasi Raspbian yang merupakan sistem operasi *default* atau bawaan pada Raspberry Pi. Pengaturan pada sistem operasi Raspbian hanya SSH (*Secure Shell*) dengan tujuan untuk dapat mengakses Raspberry Pi.

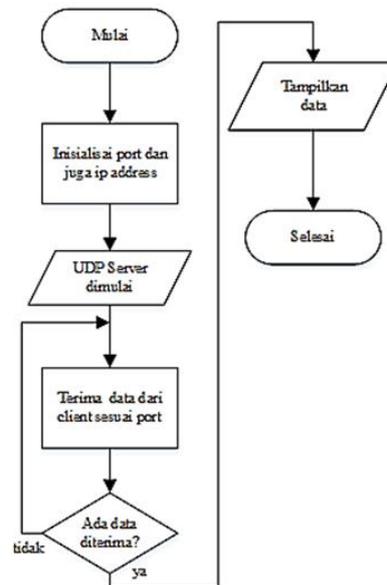
Rancangan dan pemrograman komunikasi protokol 6LoWPAN dengan RIOT OS menggunakan perangkat lunak Gedit dan *compiler gcc* pada Ubuntu 14.04 LTS. Pemrograman untuk pembentukan *Operating System* RIOT OS sudah disediakan pustaka dari pengembang, sehingga hanya menggabungkan pustaka dan pemrograman yang sederhana. Pemrograman dibagi menjadi 2 yaitu, rancangan *sensor node* dan juga *sink node* yang bertindak sebagai *server* dan juga *client*. Gambar 3 menunjukkan diagram alir proses pengiriman langsung *frame data* pada *sensor node* atau *client* menuju ke *sink node* atau *server*.



Gambar 3 Diagram alir *sensor node* protokol 6LoWPAN

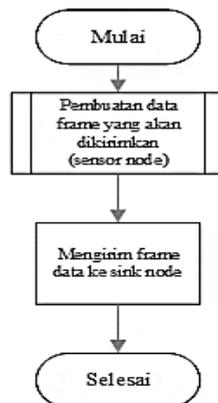
Pemrograman 6LoWPAN dilakukan dengan mengacu pada proses pengiriman UDP yang hanya mengirimkan *frame data* yang dibuat secara terus-menerus. *Sink node* akan langsung dihubungkan dengan laptop tidak melalui *gateway* pada proses pengujian berlangsung. Program dimulai dengan menginisialisasi dari *port client* dan juga *port server* sebagai tujuan pengiriman data, selain itu ip address juga diinisialisasi untuk mengetahui alamat dari *node client*.

Selanjutnya mencetak UDP *client* untuk mengetahui program sudah berjalan apa belum. Proses dilanjutkan dengan dikirimnya *frame data* dari *sensor node* atau *client* menuju ke *sink node* atau *server*. Pengiriman data harus sesuai dengan *port* tujuan. Data yang dikirim akan ditampilkan dengan tujuan mengetahui data seberapa yang dikirim, dikarenakan program UDP ini mengirimkan data secara terus menerus. Diagram alir pada Gambar 4 merupakan diagram alir dari *sink node* atau *server*. Menunjukkan proses yang diawali sama seperti pada *node client*, yaitu dengan menginisialisasi *port* dan juga *ip address* dari *server*. Selanjutnya mencetak tulisan yang menunjukkan bahwa *server* sudah siap untuk menerima data dari *client*. Proses selanjutnya adalah menerima *frame data* dari *sensor node* atau *client*. Jika *frame data* diterima, data akan ditampilkan pada *sink node* atau *server*, jika tidak *sink node* atau *server* akan menunggu sampai ada data yang diterima.



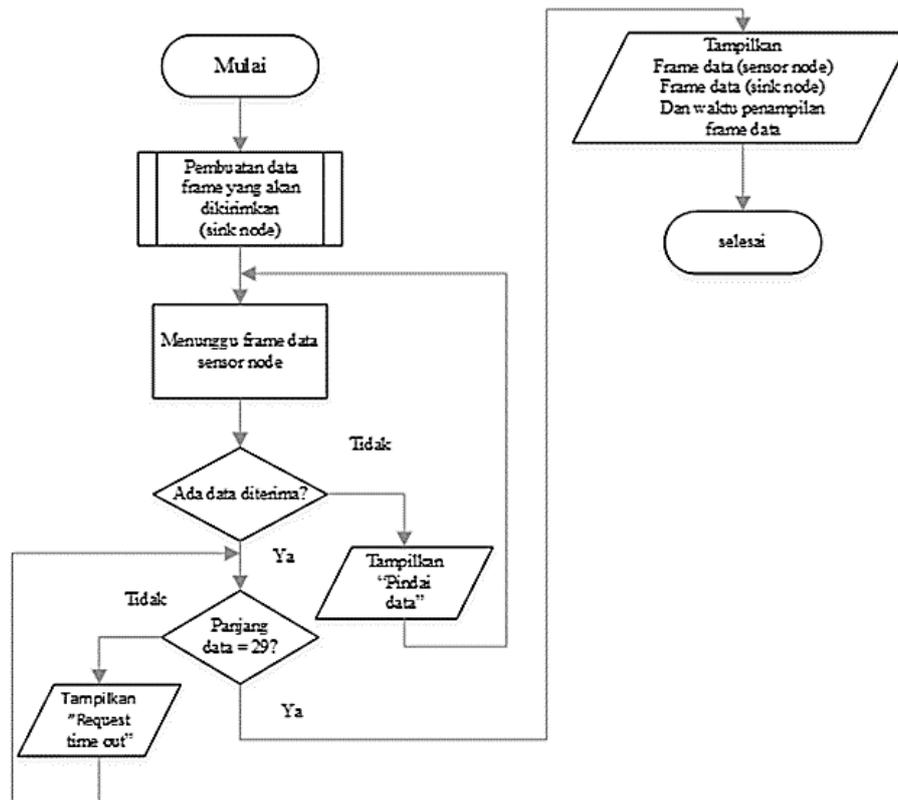
Gambar 4 Diagram alir *sink node* protokol 6LoWPAN

Rancangan dan pemrograman sistem dari komunikasi protokol Zigbee menggunakan perangkat lunak Arduino IDE dibagi menjadi 2 yaitu, rancangan *sensor node* dan juga *sink node*. Gambar 5 merupakan diagram alir *sensor node* dengan proses kerja yang sederhana yaitu dimulainya pembuatan *data frame* untuk dikirimkan ke *sink node*. Setelah dibuatkan *frame data*, selanjutnya melakukan pengiriman *frame data* dengan cara menampilkan pada serial monitor, secara otomatis akan terkirim melalui komunikasi serial Xbee.



Gambar 5 Diagram alir *sensor node* protokol Zigbee

Rancangan pada *sink node* sama seperti *sensor node* yaitu membuat *frame data*. Selanjutnya, *sink node* akan menunggu data dari *sensor node*, jika *sink node* sudah mendapatkan data dari *sensor node* maka selanjutnya akan diproses, jika tidak maka akan mencari terus sampai mendapatkan data dari *sensor node*. Dikarenakan zigbee merupakan protokol yang bisa dirancang maka, proses memeriksa sesuai apa tidak panjang *frame data* sepanjang 29 *byte*, jika sesuai akan ditampilkan di serial monitor *sink node* bersamaan dengan *frame data* dari *sink node*, jika tidak maka akan menampilkan "Request Time Out" yang menjadi acuan jika data yang diterima kurang dari 29 *byte* atau cacat dengan mengulangnya sampai mendapatkan data yang sesuai dengan panjang 29 *byte*. Proses selanjutnya menampilkan *frame data* dari *sensor node* dan juga dari *sink node*. Selain itu, menampilkan juga waktu pada *gateway* untuk mengetahui besaran *delay* pada protokol Zigbee. Gambar 6 merupakan diagram alir dari *sink node*.



Gambar 6 Diagram alir *sink node* protokol Zigbee

Frame data yang digunakan pada pengiriman dan penerimaan dibuat sendiri yang terdiri dari bagian *header*, *separator*, *source node*, *destination node*, dan data sensor DHT dan RTC. Format data yang dikirim pada bagian pemancar (*sensor node*) dan penerima (*sink node*) menggunakan header "H" yang menunjukkan bahwa setiap *frame data* dimulai dengan "H", serta setiap parameter dipisahkan dengan tanda "#" untuk mengetahui *frame data* secara terpisah, dan diakhiri dengan tanda "!" yang menunjukkan bahwa *frame data* akan berakhir. Susunan dari *frame data* merupakan susunan yang dibuat sendiri dengan alasan dapat melihat besaran suhu, kelembapan, dan juga waktu yang ditunjukkan pada saat proses pengiriman dan penerimaan data berlangsung, dengan adanya pemisah dan juga mengetahui awal dan akhir dari *frame data*, panjang data yang dikirimkan adalah sebesar 29 *byte*. Tabel 1 menjelaskan susunan dari *frame data* yang digunakan dan dibuat sendiri.

Perancangan juga dilakukan pada pengalamatan kedua *node* yang menggunakan ipv6 dengan format *link-local unicast* dikarenakan komunikasi dilakukan secara *point-to-point* secara

langsung antara dua node dalam suatu jaringan. Penggunaan IPv6 sebagai pengalamatan merupakan hal yang penting pada 6LoWPAN. Pengalamatan pada IPv6 memiliki panjang 128 bit dan terbagi atas 64 bit *prefix* dan 64 bit *interface Identifier* (IID) [8]. Dengan ketentuan $fe80::/10$ atau memiliki sepuluh bit alamat *prefix*, dan sisa panjang alamat akan mengacu pada *serial number* pada Xbee.

Tabel 1 Format *frame data*

Data ke-	<i>Sensor node</i>		<i>Sink node</i>	
	Keterangan	Nilai	Keterangan	Nilai
1	<i>Header</i>	H	<i>header</i>	H
2	Pemisah	#	pemisah	#
3	ID (<i>sensor node</i>)	1	ID(<i>sink node</i>)	2
4	Pemisah	#	pemisah	#
5	pengukuran DHT 11	(suhu)	pengukuran DHT 11	(suhu)
6	Pemisah	#	pemisah	#
7	pengukuran DHT 11	(kelembapan)	pengukuran DHT 11	(kelembapan)
8	Pemisah	#	pemisah	#
9	nilai RTC	(thn)	nilai RTC	(thn)
10	Pemisah	#	pemisah	#
11	nilai RTC	(bln)	Nilai RTC	(bln)
12	Pemisah	#	pemisah	#
13	nilai RTC	(tgl)	nilai RTC	(tgl)
14	Pemisah	#	pemisah	#
15	nilai RTC	(jam)	nilai RTC	(jam)
16	Pemisah	#	pemisah	#
17	nilai RTC	(menit)	nilai RTC	(menit)
18	Pemisah	#	pemisah	#
19	nilai RTC	(detik)	nilai RTC	(detik)
20	Pemisah	#	pemisah	#
21	Akhir	!	akhir	!

2.4 Implementasi

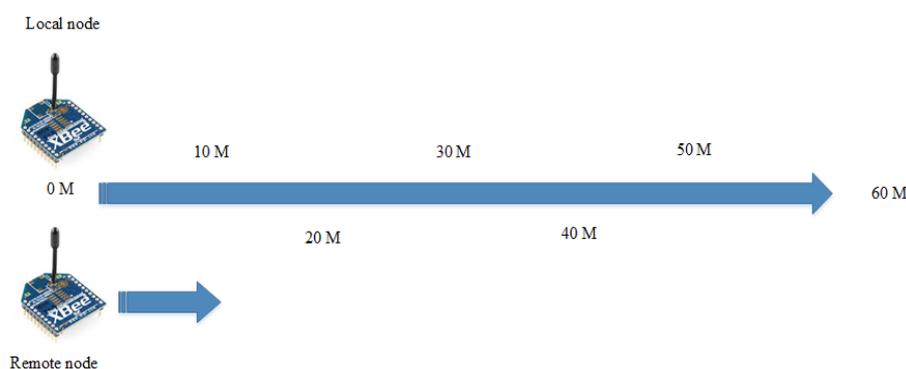
Gambar 7 merupakan gambar *sensor node* dan juga *sink node* yang telah diimplementasikan. Tiap *node* terdiri dari mikrokontroler Arduino Mega 2560, Xbee, modul sensor DHT11, dan juga RTC.

Gambar 7 *sensor node* dan *sink node*

3. HASIL DAN PEMBAHASAN

Rencana pengujian dibagi menjadi 3 yaitu, pengujian kekuatan sinyal pada Xbee, pengujian komunikasi dengan protokol Zigbee, dan pengujian komunikasi dengan protokol 6LoWPAN. Pengujian *signal strength* atau melihat kekuatan sinyal antar Xbee (RSSI). RSSI (*Received Signal Strength Indication*) adalah sebuah ukuran kekuatan sinyal radio yang diterima

oleh *receiver* dengan kata lain dapat mengetahui besar kemampuan sejauh mana *transmitter* dan *receiver* dalam melakukan komunikasi [9]. Jarak yang diuji adalah sama seperti pengujian komunikasi protokol yaitu, 10 sampai dengan 60 meter dengan interval 10 meter, jarak. Pemilihan jarak ini merupakan asumsi dari jarak efektif yang bisa dilakukan oleh perangkat Xbee. Pengujian RSSI menggunakan 2 jenis *node* yaitu, *local node* dan juga *remote node*, maksud dari *node* ini adalah modul Xbee bukan menggunakan *node* yang menggunakan Arduino Mega 2560. Perangkat Xbee yang digunakan adalah perangkat Xbee S2 dengan asumsi hasil RSSI dari Xbee S1 sama, dikarenakan dilakukan pada tempat dan jarak yang sama dengan pengujian komunikasi protokol. Dimaksud dengan *local node* adalah *node* yang akan mengirimkan dan menerima paket data, sedangkan *remote node* merupakan *node* yang meneruskan dan mengembalikan menuju *local node* paket data yang sudah dikirimkan oleh *local node*. Pada *local node* menggunakan mode API (*Application Programming Interface*), sedangkan *remote node* menggunakan mode *Transparent* (AT). Perangkat Xbee bekerja pada gelombang 2,4 GHz dan bekerja sama dengan sinyal Wi-Fi, maka dari itu interferensi gangguan terhadap pengujian RSSI berpengaruh. Pada Gambar 8 mengilustrasikan pengujian RSSI Xbee.



Gambar 8 Ilustrasi pengujian RSSI Xbee

Pengujian dilakukan di perangkat lunak XCTU yang merupakan perangkat lunak khusus disediakan oleh Digi Software yang merupakan perusahaan pembuatan Xbee. Dengan *tools* yang disediakan dapat langsung mengetahui besar dari RSSI dengan satuan dBm.

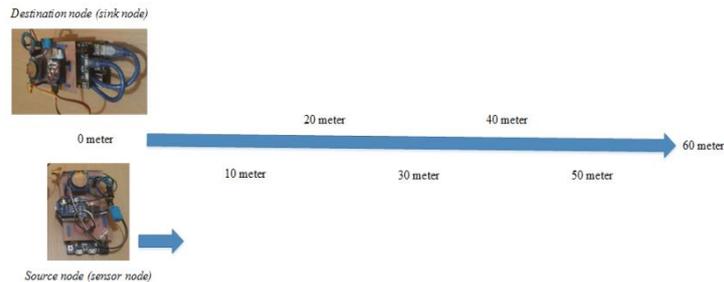
Pengujian komunikasi dibagi menjadi 2 yaitu, pengujian komunikasi dengan protokol 6LoWPAN dan komunikasi dengan protokol Zigbee. Pengujian dilakukan dengan melihat nilai end to end delay dan packet loss. *End-to-end delay* adalah waktu yang diperlukan oleh suatu paket data yang berasal dari *source node* hingga mencapai *destination node*. Dalam hal ini ini *source node* merupakan *sensor node* dan *destination node* merupakan *sink node*. Pengertian *packet loss* merupakan banyaknya paket yang hilang atau tidak diterima selama proses pengiriman dari *source node* ke *destination node*. Perhitungan *packet loss* menggunakan rumus yang terdapat pada persamaan (1).

$$\text{Packet loss} = \frac{(a-b)}{b} \times 100\% \quad (1)$$

a = paket data yang dikirim b = paket data yang diterima

Pengujian komunikasi dilakukan dengan memberikan variasi pada jarak dan juga *baudrate*. Jarak yang diberikan pada pengujian sebesar 10 meter sampai dengan 60 meter dengan interval sebesar 10 meter, sedangkan pada *baud rate* yang diujikan sebesar 1200 bps, 4800 bps, 9600 bps, 19200 bps, 38400 bps, 56900 bps, dan 115200 bps. Pengujian ini dilakukan 7 skenario, dengan mengacu pada *baud rate* dan pada masing-masing *baud rate* akan ada variasi jarak yang sudah dijelaskan sebelumnya. Selama dilakukan pengujian terhadap komunikasi protokol tidak menggunakan *gateway*, tetapi langsung menghubungkan *sink node* langsung ke laptop. Alasan tidak menggunakan *gateway* pada pengujian adalah fungsi dari *gateway* hanya sebagai *ethernet*

shield atau hanya melakukan monitoring terhadap proses pengiriman dan penerimaan data dari komputer atau jaringan dengan menggunakan fungsi dari SSH tidak termasuk kedalam salah satu pengujian terhadap komunikasi pada kedua protokol. Gambar 9 merupakan ilustrasi dari pengujian komunikasi protokol Zigbee dan protokol 6LoWPAN.



Gambar 9 Ilustrasi pengujian komunikasi protokol

Pada pengujian komunikasi protokol 6LoWPAN digunakan perangkat lunak Wireshark yang merupakan perangkat lunak yang akan merekam semua proses pengiriman dan penerimaan data. Protokol UDP, tidak menyediakan *packet loss*, dikarenakan tidak menyediakan *acknowledgement* dan juga *sequence number* [10]. Pengujian protokol Zigbee menggunakan serial monitor dari perangkat lunak Arduino IDE yang menampilkan aktifitas pengiriman dan penerimaan data. Pendeteksian *packet loss* pada pengujian protokol Zigbee akan terlihat pada *serial monitor* dengan menampilkan “Request Time Out” yang menunjukkan data yang diterima memiliki panjang data kurang dari 29 *byte*.

Pada hasil proses kerja alat dibahas berjalannya alat dari awal proses, pengiriman data penerimaan data. Pada Gambar 10 dan Gambar 11 merupakan proses awal dari berjalannya program *sensor node* dan juga *sink node*. Proses kerja alat hanya terfokus pada 6LoWPAN.

```
Starting 'UDP client process'
UDP client process started nbr:10 routes:10
client IPv6 address: fd00::0013:A200:40A0:AF1E
fe80::0013:A200:40A0:AF1E
Created a connection with the server :: local/remote port 8765/5678
```

Gambar 10 Proses awal pada sensor node 6LoWPAN

```
Starting 'UDP server process'
UDP server process started nbr:10 routes:10
client IPv6 address: fd00::0013:A200:40A0:AF97
fe80::0013:A200:40A0:AF97
Created a connection with the server :: local/remote port 8765/5678
```

Gambar 11 Proses awal pada sink node 6LoWPAN

Pada Gambar 12 merupakan proses pengiriman *frame data* dari *sensor node* menuju *sink node*, sedangkan pada Gambar 13 merupakan proses penerimaan *frame data* dari *sensor node* menuju *sink node*.

```
DATA dikirim ke 2 'H#1#bb#cc#dd#ee#ff#gg#hh#ii#!'
DATA dikirim ke 2 'H#1#bb#cc#dd#ee#ff#gg#hh#ii#!'
DATA dikirim ke 2 'H#1#bb#cc#dd#ee#ff#gg#hh#ii#!'
```

Gambar 12 Pengiriman frame data pada *sensor node*

DATA diterima 'H#1#bb#cc#dd#ee#ff#gg#hh#ii#!'dari 1
DATA diterima 'H#1#bb#cc#dd#ee#ff#gg#hh#ii#!'dari 1
DATA diterima 'H#1#bb#cc#dd#ee#ff#gg#hh#ii#!'dari 1

Gambar 13 Penerimaan frame data pada *sink node*

Hasil pengujian dari protokol 6LoWPAN yang dibandingkan dengan protokol Zigbee dari segi *end-to-end delay* dan juga *packet loss* dari 7 skenario menurut *baud rate* dan juga jarak. Tabel 2 dan Tabel 3 merupakan hasil dari pengujian *end-to-end delay* dari kedua protokol.

Tabel 2 Hasil pengujian *end-to-end delay* protokol 6LoWPAN

<i>end-to-end delay</i> 6LoWPAN (milidetik)						
Jarak <i>Baud rate</i>	10 meter	20 meter	30 meter	40 meter	50 meter	60 meter
1200 bps	3337	1784	1923	1775	2136	2067
4800 bps	1720	1713	1555	2242	1301	369
9600 bps	2153	2168	2226	2594	2419	1941
19200 bps	2396	2396	1859	1915	1908	1872
34800 bps	2623	2024	2000	2152	2116	2183
57600 bps	1079	174	1056	1545	1051	1153
115200 bps	2066	1763	1770	1782	1494	1538

Tabel 3 Hasil pengujian *end-to-end delay* protokol Zigbee

<i>end-to-end delay</i> Zigbee (milidetik)						
Jarak <i>Baud rate</i>	10 meter	20 meter	30 meter	40 meter	50 meter	60 meter
1200 bps	167	326	365	171	114	628
4800 bps	273	70	368	161	368	549
9600 bps	499	254	150	592	185	349
19200 bps	147	287	238	336	1085	658
34800 bps	141	233	212	137	181	198
57600 bps	1079	174	1056	1545	1051	1153
115200 bps	334	421	445	154	252	602

Sedangkan pada hasil *packet loss*, protokol 6LoWPAN tidak menyediakan *packet loss* dikarenakan tidak menyediakan *sequence number* dan juga *acknowledgement*. Protokol Zigbee memiliki *packet loss* yang terdapat pada Tabel 4.

Tabel 4 *Packet loss* protokol Zigbee

<i>Packet loss</i> Zigbee (%)						
Jarak <i>Baud rate</i>	10 meter	20 meter	30 meter	40 meter	50 meter	60 meter
1200 bps	0	0	53	20	60	13
4800 bps	13	33	27	13	60	60
9600 bps	20	20	13	40	13	33
19200 bps	20	40	7	27	40	13
34800 bps	0	27	7	13	40	53
57600 bps	47	33	13	20	60	47
115200 bps	13	13	20	20	13	13

Analisis terhadap sistem adalah pemilihan terhadap variasi terhadap *baud rate* dan juga jarak antar *node*. Pada *baud rate* hubungannya adalah pada penggunaan Xbee, dimana Xbee akan dihubungkan dengan mikrokontroler Arduino Mega 2560 melalui komunikasi serial. Pengaruh *end-to-end delay* dengan *baud rate*, jika menggunakan *baud rate* yang kecil akan

menghasilkan *end-to-end delay* yang besar, sebaliknya jika menggunakan *baud rate* yang besar akan menghasilkan *end-to-end delay* yang kecil, dikarenakan peran UART untuk menerjemahkan bit-bit serial menjadi data yang utuh akan mengalami perlambatan dan menghasilkan *end-to-end* yang besar jika menggunakan *baud rate* yang kecil, sebaliknya UART yang menerjemahkan bit-bit serial akan mengalami percepatan pada proses pengiriman data sehingga menghasilkan *delay* yang kecil. Hubungan jarak dengan *delay* adalah pengaruh dengan RSSI (*Received Signal Strenght Indication*), semakin jauh maka akan menghasilkan *delay* yang begitu besar, pengaruh juga pada interferensi terhadap sistem dikarenakan pengujian dilakukan di lingkungan kampus maka akan menghasilkan *delay* yang tidak menentu. Pada 6LoWPAN menggunakan UDP (*User Datagram Protocol*) dengan melakukan komunikasi secara sederhana dengan makeisme yang sangat minimal. Sederhana yang dimaksud adalah pada UDP komunikasi hanya mengirimkan data secara terus menerus tanpa adanya komunikasi awal [11], memang penggunaan UDP dirasa tepat untuk mengirimkan informasi atau data yang lebih menekankan kecepatan. Hasil yang didapat menunjukkan bahwa Zigbee memiliki waktu *delay* yang kecil, dikarenakan protokol Zigbee merupakan protokol, yang lebih sederhana dari pada protokol 6LoWPAN yang tidak memiliki *transport layer* pada OSI layer [12]. Semua proses pengiriman dan penerimaan data mengacu pada OSI layer dari *application layer* hingga PHY layer.

Dari segi *packet loss*, seperti sebelumnya yang dijelaskan bahwa 6LoWPAN tidak menyediakan *packet loss*, dikarenakan tidak memiliki *acknowledgement* dan juga *sequence number* untuk mendeteksi *packet loss*. Protokol Zigbee merupakan protokol yang dalam pengoperasiannya dapat dikembangkan dan dirancang, sehingga dapat mendeteksi *packet loss*. Hubungan *packet loss* terhadap pengujian adalah pada *baud rate* yang berhubungan dengan UART yang bertugas sebagai menerjemahkan bit-bit serial menjadi data yang utuh, jika menggunakan *baud rate* yang kecil maka akan menghasilkan *packet loss* yang kecil dikarenakan UART akan menerjemahkan dengan pelan dan menghasilkan *error* pada saat penerjemahannya kecil, sebaliknya jika menggunakan *baud rate* yang besar akan menghasilkan *error* yang besar pada saat menerjemahkan dan mengakibatkan *packet loss* besar. Pengaruh jarak sama seperti *delay* akan menghasilkan data yang tidak menentu dan mengakibatkan *packet loss*.

Hasil dari RSSI (*Received Signal Strenght Indicationi*) menunjukkan bahwa jarak maksimal adalah 60 meter, sistem masih bisa berjalan pada jarak antara *source node* dan juga *destination node* sampai dengan jarak 60 meter. Tabel 5 merupakan hasil pengukuran dari RSSI.

Tabel 5 Hasil pengukuran RSSI

Jarak (meter)	Persentase (%)
10	60,1
20	50,2
30	44,0
40	39,8
50	36,8
60	33,5

4. KESIMPULAN

Berdasarkan hasil penelitian dan juga hasil pengujian terhadap sistem perangkat lunak dan perangkat keras, maka dapat diambil kesimpulan sebagai berikut:

1. 6LoWPAN memberikan dukungan protokol jaringan IPv6 pada *sensor node* dan juga *sink node*, yang terdiri dari:
 - Memiliki rata-rata *end-to-end delay* dari semua skenario sebesar 1879 milidetik yang lebih besar dari protokol Zigbee yang hanya sebesar 422 milidetik.
 - Protokol 6LoWPAN tidak menyediakan *packet loss* dikarenakan menggunakan UDP (*User Datagram Protocol*) yang tidak memiliki pengkoreksian *packet loss*

seperti *acknowledgement* dan juga *sequence number*. Sedangkan protokol Zigbee memiliki rata-rata *packet loss* sebesar 26%.

- Mengirim pesan yang berupa *frame data* secara *unicast*.
2. Kemampuan yang dimiliki oleh sistem terdiri dari:
 - Jarak maksimal antara *sensor node* dengan *sink node* yaitu 60 meter.
 - Dapat mengirimkan *frame data* sepanjang 29 byte dari *sensor node* menuju *sink node*.

5. SARAN

Penelitian yang telah dilakukan masih memiliki beberapa kekurangan dan memungkinkan untuk dilakukan pengembangan lebih lanjut. Diantaranya:

1. Menggunakan TCP pada *transport layer* 6LoWPAN guna mendapatkan koneksi yang handal agar bisa mendeteksi paket yang hilang (*packet loss*).
2. Menggunakan format pengiriman *multicast* atau *broadcast* pada proses pengiriman data berlangsung.

DAFTAR PUSTAKA

- [1] Sang-Joong, J., 2013, *Personal Machine-to-Machine (M2M) Healthcare System with Mobile Device in Global Networks*, Department of Electrical Engineering Universitas Ouluensis, Oulu.
- [2] Toscano, E. dan Bello, L. L., 2012, Comparative assessment of IEEE 802.15.4/Zigbee and 6LoWPAN for low-power industrial WSN in realistic scenarios, *IEEE 9th International Workshop on Factory Communication*, Lemgo, NRW, 21-24 Mei 2012.
- [3] Abdullah, H., Juhana, T., Lidyawati, L., 2013, Simulasi dan Analisis Transmisi Video Streaming pada Jaringan Wifi dengan Menggunakan Opnet Modeler 14.5, *Jurnal Online Institut Teknologi Nasional*, No.1, Vol.1 57-67.
- [4] Bobanto, W. S., 2015, Analisis Kualitas Layanan Jaringan Internet, *e-journal Teknik Elektro dan Komputer*, No.1, Vol.4 80-87.
- [5] Ele, L.S., Kothari, Rao, M., 2014, 6LoWPAN Based Wireless Sensor Network to Monitor Temperature, *International Journal of Advanced Electronics and communication Engineering*, No 1, Vol 1 1-6.
- [6] Sulistyanto, M. P. dan Nugraha, D. A., 2015, Implementasi IoT (Internet of things) dalam pembelajaran di Universitas Kanjuhuran Malang, *SMARTICS Journal*, No.1, Vol.1 20-23.
- [7] Latre, B., Pieter De, M., 2006, Throughput and Delay Analysis of Unslotted IEEE 802.15.4., *Journal of Network*, No.1, Vol.1 20-28
- [8] Sakhtivel, C. dan Kalaiprasath, R., 2016, COMPARATIVE REVIEW ON INTERNET PROTOCOL VERSION 6 (IPV6), *International Journal of Advanced Research*, No.2, Volume 4 1076-1078.
- [9] Palaha, Fadli. dan Zaini, 2014, PROPAGASI INDOOR GELOMBANG RADIO PERANGKAT XBEE, *Jurnal Nasional Teknik Elektro*, No.2, Vol.3 148-156.
- [10] Behrouz A. Forouzan, 2007, *Data Communications And Networking*, Ed.4, Mc Graw Hill.
- [11] Sarma, P.M., 2013, Performance Measurement of TCP and UDP Using Different Queuing Algorithm in High Speed Local Area Network, *Internatinal Journal of Future Computer and Communication*, No 6, Vol. 2 682-686.
- [12] Vishwakarma, D.D., 2012, IEEE 802.15.4 and ZigBee: A Conceptual Study, *International Journal of Advance Research In Computer and Communication Engineering*, No 7, Vol 1 477-480.