INTEGRASI *WIRELESS SENSOR NETWORK* PADA SISTEM TERTANAM MENGGUNAKAN METODE *PROXY AGENT LAYER*

Ahmad Heryanto[1)], Siti Nurmaini[2)], Tri Wanda Septian[3)], Ricy Firnando[4)]

[1234)]Sistem Komputer Universitas Sriwijaya
Jl. Lintas Timur km. 32 Indralaya, Ogan Ilir 30662. Indonesia
[1)]email : hery@zipruz.com

ABSTRACT

Wireless sensor network uses a proxy agent layer into a solution for embedded systems to exchange information to the data center. Data center as a collector and processor of information. Data centers can cover the weaknesses of embedded systems for computing and storage. The test results of WSN in embedded systems, Agents can send a good-quality package in real time at the power level RSSI -72 dBm with transmission distance of 40 meters. Meanwhile, the RSSI level of -93 dBm at a distance of 80 meters, the data can still be acceptable, but the communication is not running in real time and sufficient quality. On the label RSSI -98 dBm with a distance of 90 meters, the data received by the Coordinator including the very poor and the communication do not run in real time.

Key words

Sistem tertanam, Wireless Sensor Network, Proxy Agent Layer.

1. Pendahuluan

Aplikasi sistem tertanam muncul dengan beraneka macam bentuk. Papularitas aplikasi tersebut diperoleh karena ukurannya yang kecil, mengkonsumsi daya rendah dan hemat energi. Sistem tertanam menggunakan *Small Board Computer* (SBC) dan mikrokontroler. Produk-produk SBC dan mikrokontroler pun semakin beragam dengan menawarkan kelebihan dan kemudahan yang semakin baik jika dibandingkan dengan komputer konvensional. Namun, masih ada beberapa kekurangan yang dimiliki oleh peralatan sistem tertanam seperti kemampuan pemrosesan data yang rendah, media penyimpanan data yang kecil serta kemampuan komunikasi yang terbatas.

Dengan segala kekurangan tersebut, sistem tertanam telah berkembang dengan sangat pesat. Salah satu *trend* teknologi sistem tertanam adalah *Internet of Thinks* (IoT). Pada konsep IoT, semua perangkat terhubung ke jaringan internet. Pada IoT, sistem tertanam harus mampu melampaui batasan-batasan kelemahan tersebut dengan cara bekerja secara terdistribusi. Sistem terdistribusi tersebut harus memiliki kemampuan seperti berbagi data, membagi beban kerja, *remote control*, *transparant*, dan *fault tolerance*. Pada tahun 2020 diperkirakan akan terdapat lebih dari 50 miliar perangkat yang akan terhubung ke internet, sebagian besar akan didominasi oleh sistem tertanam yang terhubung secara nirkabel/kabel melaui jaringan komputer **[1]**.

Komunikasi *peripheral* sistem tertanam yang ada saat ini kebanyakan dari manusia ke mesin (*human to machine)*. Pada era IoT komunikasi terjadi antara mesin ke mesin atau *machine to machine* (M2M). M2M merupakan konsep teknologi yang menuntut peralatan-peralatan elektronik untuk saling berkomunikasi tanpa perantara dari manusia. Contoh M2M misalnya, alat monitoring kesehatan yang terpasang pada tubuh seorang pasien yang dilengkapi *sensor* medis dan secara kontinu mengirimkan rekam medis kepada *data center* tempat integrasi sistem terdistribusi tersebut, jika pengguna tersebut mengalami gangguan medis seperti serangan jantung maka seketika *peripheral* yang terdapat pada pasien akan menghubungi *data center* dan kemudian *data center* akan mengirimkan informasi ke sistem unit gawat darurat agar segera dikirim ambulan ke lokasi pasien, kemudian dari *data center* juga secara otomatis mencari catatan rekam medis pesien agar dapat digunakan oleh dokter untuk mengambil keputusan terbaik jika pasien tersebut tiba dirumah sakit.

M2M menggunakan berbagai perangkat *tranduser* seperti *sensor* untuk menangkap informasi dari lingkungan (seperti seperti suhu, tingkat kecerahan, tekanan, posisi dan lain-lain). Informasi akan disampaikan melalui jaringan (nirkabel, kabel atau kombinasi) ke *data center*. Sebuah aplikasi khusus dibuat untuk mengolah berbagai informasi yang dihasilkan oleh *peripheral* sistem tertanam dan

selanjutnya dapat dianalisa untuk menghasilkan keputusan yang dapat mempermudah kualitas hidup umat manusia. Namun tantangan besar telah menunggu, dibutuhkan teknik integrasi yang handal dan optimal dalam sistem tertanam. Sering kali sistem tertanam menggunakan *platform* yang berbeda-beda, sehingga jika ada data yang dikirim oleh *peripheral* yang *multiplatform* tidak dapat dipahami oleh *peripheral* yang lain dengan baik. Pada penelitian ini teknik integrasi menggunakan *proxy Agent layer* dimana setiap *Agent -Agent* sistem tertanam yang mengirimkan informasi menggunakan *proxy* yang akan diletakan berada di antara *Agent* dan *data center*. Implementasi model ini terdiri dari *software Agent* dan *web service*[2]. Keunggulan metode ini sistem akan menjadi independen yaitu tidak bergantung terhadap bahasa pemrograman tertentu, sistem operasi, perangkat keras dan komponen pendukung lainnya. *Web service* menggunakan protokol dan format data yang telah terstandarisasi bagi berbagai *platform* sistem tertanam sehingga data dapat di integrasikan secara statik dan dinamik dalam melakukan perpindahan informasi pada *peripheral* yang *multiplatform* [3].

2. Wireless Sensor Network

Wireless Sensor Network (WSN) adalah aplikasi sistem tertanam yang menggunakan satu atau lebih sensor elektronik dan dilengkapi dengan peralatan sistem komunikasi nirkabel. Sensor-sensor digunakan untuk menangkap informasi sesuai dengan karakteristik lingkungan tempat sistem tertanam tersebut berada. Jika rangkaian WSN dihubungkan ke proxy dan proxy tersebut dapat mengakses Internet, maka WSN dapat diakses dan berkolaborasi dengan sistem lain.

WSN yang terkoneksi ke internet dapat membuat penggunaannya melakukan monitoring dimanapun dan kapanpun tanpa harus memiliki akses fisik keperangkat sistem monitoring pada sensor-sensor tersebut.

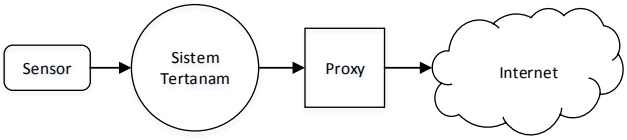
Teknologi WSN banyak digunakan untuk segala bidang. Beberapa aplikasi berikut banyak menggunakan teknologi WSN:

1. Monitoring lingkungan.
2. Tracking System
3. Biomedik
4. Keamanan
5. Perternakan
6. Pertanian
7. Dan lain-lain

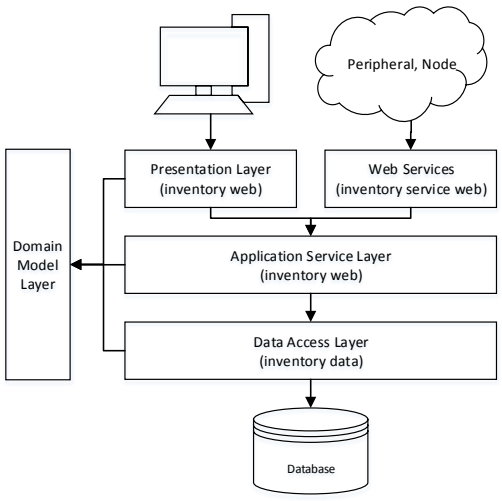
Gambar 1. Komponen Xbee WSN

Setiap node WSN (Agent) akan mengirim data sensor ke suatu Coordinator, dan hasil kumpulan data semuanya akan diolah sehingga akan memberikan suatu informasi. Basis *engine* dari WSN pada penelitian ini menggunakan Xbee S2. Xbee merupakan merek dagang dari digi yang memproduksi *chip* khusus untuk perangkat sistem tertanam yang akan mentransmisikan informasi melalui gelombang. Xbee S2 tampak pada gambar 1. Keunggulan dari Xbee S2 dapat mengimplementasikan *tree network* dan *mesh networking* dengan cara memberikan tugas pada masing-masing *node* dengan *mode Coordinator*, *Router* dan *End-Device* (*Agent*)[4].

1. *Coordinator*
   Xbee S2 mampu mengoperasikan protokol zigbee dengan mode *Coordinator* (ZC). *Coordinator* akan menjadi pusat komunikasi antara router dan *end device* dengan aplikasi pada sistem tertanam. Coordinator router dan *end device* akan membentuk *tree network* yang kepada proxy server. Pada sebuah jaringan WSN yang berbasis Xbee harus memiliki satu node yang mengoperasikan mode *Coordinator*.
2. Router
   Xbee S2 dengan mode router akan menjalankan protocol zigbee dengan mode router (ZR). Router akan bertindak sebagai *forwarder* paket informasi yang dikirim oleh sistem tertanam ke *data center*. Router juga memiliki kemampuan untuk menentukan jalur terbaik dalam komunikasi *end device* dan *Coordinator*.
3. *End device*
   Xbee S2 dengan mode *end device* akan menjadi *Agent* dari sistem tertanam. Pada mode *end device* engine xbee memiliki fitur untuk menghemat energi dari aplikasi yang dibuat karena memiliki kemampuan sleep jika terjadi kondisi idle hingga komunikasi terbentuk antara coordinator dan *end device* terbentuk kembali.



Gambar 2. Sistem tertanam metode *proxy*[5]



Gambar 3. Alur kerja dari *Web Service* [6]

3. Proxy *Agent* Layer

Metode *proxy* adalah bentuk integrasi sistem tertanam yang paling banyak digunakan pada aplikasi *peripheral*. Skema ini memerlukan *proxy server* yang memiliki kinerja tinggi, *server* yang dapat menghubungkan dan melakukan pertukaran informasi dengan beberapa sistem tertanam. Metode ini mudah untuk dikembangkan dan menghubungkan perangkat tertanam ke internet menjadi lebih cepat. Gambar 2 adalah arsitektur skema *proxy layer*. Proxy menyediakan suatu layanan untuk meneruskan setiap permintaan *data center* kepada *agent-agent* yang terdapat di lingkungan atau meneruskan setiap informasi yang dikumpulkan oleh *agent-agent* kepada *data center*. Proxy merupakan suatu server atau program komputer yang mempunyai kemampuan untuk menghubungkan *Agent* kedalam jaringan internet.

4. Perancangan *Hardware, Software* dan Insfratruktur

4.1 Disain *Hardware*

Ada bebeberapa komponen *hardware* yang digunakan pada penelitian ini. Komponen *hardware* tersebut adalah *Data center,* Sistem Tertanam dan Modul *Sensor*.

4.1.1. Data center

Data center akan menggunakan *file system* ext4, dan sistem operasi linux Centos 6. Dan terhubung ke internet dengan menggunakan alamat *IP Static* dari Fakultas Ilmu Komputer Universitas Sriwijaya. Pada *data center* tersebut akan berjalan engine *database* mysql, *database* ini akan merekam setiap data yang dikirimkan oleh *Agent* pada sistem tertanam.

4.1.2. Sistem Tertanam

Sistem tertanam digunakan dua jenis tipe *hardware* yaitu mikrokontroler dan SBC, tujuannya untuk menguji kemampuan dari integrasi sistem tertanam dalam menggunakan model *Proxy Agent Layer* dalam mengirimkan data ke *data center*.

4.1.3. Modul *Sensor*

Modul *sensor* yang digunakan adalah Tiga Sensor yang digunakan pada penelitian ini. Sensor-sensor tersebut adalah sensor Suhu, Kelembapan dan Intensitas Gas. Dimana untuk sensor suhu dan kelembapan menggunakan sensor DHT11 sedangkan untuk sensor Gas menggunakan TGS 2600.

4.2. Disain *Software*

Perangkat lunak yang menggunakan *web service* yang terintegrasi memiliki arsitektur yang terdiri dari lima layer[6]. Layer-layer tersebut seperti tampak pada Gambar 3.

1. *Web Service Layer*
   Web Service Layer berperan dalam implementasi *web service* yang dihasilkan oleh sistem tertanam.
2. *Presentation Layer*
   Presentation Layer diimplementasikan pada komponen *user interface.*
3. *Application Service Layer*
   Application Service Layer berisi implementasi *logic* dari suatu aplikasi.
4. *Domain Model layer*
   Domain Model Layer bertanggung jawab pada *class* dan data *transfer object* yang digunakan.
5. *Data Access Layer*
   Data Access Layer berhubungan langsung dengan *database*.

4.3 Disain Insfratruktur

Model Jaringan distribusi *Agent* yang digunakan dapat menggunakan tipe-tipe insfrastruktur sebagai berikut:

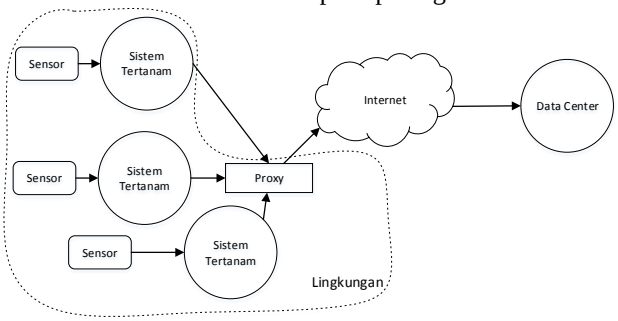
1. Sistem *client-server*
   Sistem *client-server* adalah model sistem terdistribusi yang membagi jaringan berdasarkan pemberi layanan (*server*) dan penerima jasa layanan (*client*).
2. Sistem *point to point*
   Sistem *point to point* adalah model sistem terdistribusi dimana sistem node berfungsi sebagai *client* maupun *server*.
3. Sistem *Cluster*
   Sistem *Cluster* adalah gabungan dari beberapa sistem node yang dikumpulkan pada suatu lokasi, saling berbagi tempat penyimpanan data (*storage*), dan

saling terhubung dalam jaringan lokal (*Local Area Network*).

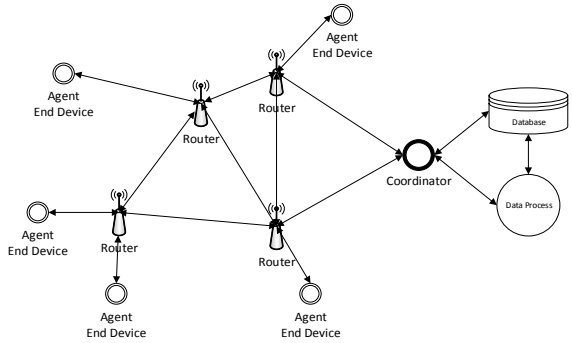
5. Hasil Percobaan

Desain sistem ini terdiri dari desain *hardware* maupun *software*. *Hardware* yang akan dipakai adalah Arduino dan Raspberry sebagai kontroler utama. Sensor suhu dan kelembapan menggunakan sensor DHT11 sedangkan untuk sensor gas menggunakan TGS 2600. *Data center* yang dibangun menggunakan Sistem Operasi Linux Centos 6 dengan Mysql sebagai *database*, serta menggunakan PHP sebagai antar muka data dari sistem ke pengguna. Pengunaan *Proxy Agent Layer* yang paling baik adalah menggunakan *web service*. *Web service* akan dipasang pada masing-masing *node* sistem tertanam. Sistem tertanam akan membentuk data berupa XML/JSON yang selanjutnya berdasarkan interval waktu/momen tertentu akan mengirimkan data ke *data center*.

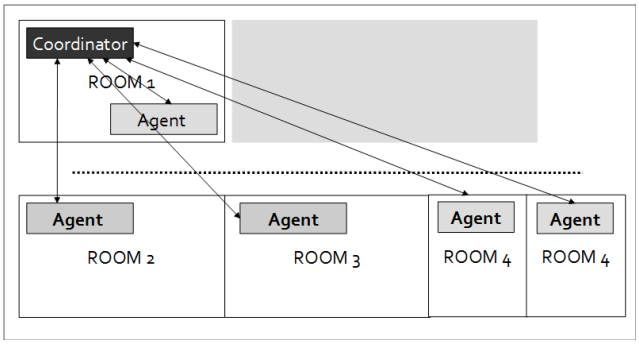
Bentuk topologi dari motode *proxy* menggunakan sistem *client-server* dan *cluster* yang ditunjukan seperti Gambar 4. *Service Agent* yang berada di dalam node sistem tertanam akan menerima HTTP *request* dari dari *data center* dan mengambil obyek *sensor* yang membentuk suatu parameter lingkungan dari suatu sistem tertanam. Lalu obyek akan dikirim ke *data center* melalui protokol komunikasi yang sama yaitu HTTP *replay*. Bentuk komunikasi divisualisasikan seperti pada gambar 5.



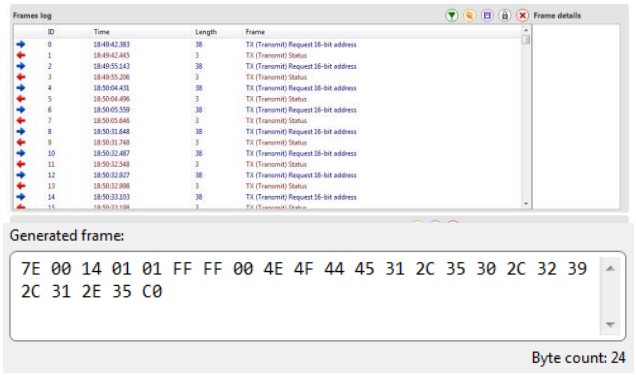
Gambar 4. *Client-Server* dan *Cluster*



Gambar 5. *Proxy Agent Layer*



Gambar 6. Disain ruangan percobaan



Gambar 7. Ganerate Frame Pada *Agent*

Data yang diperoleh dari sistem tertanam dapat diolah lebih lanjut dengan menerapkan *style* pada tiap obyek tersebut, serta dapat melakukan *encoding* hasilnya ke dalam format data HTTP yang dapat dikembalikan kepada *client*. Pengguna juga dapat memanfaatkan data yang dihasilkan oleh *service Agent* tanpa langsung terhubung dengan sistem tertanam.

Pengujian dilakukan dengan melakukan monitoring suhu, kelembapan dan intensitas gas di dalam beberapa ruangan yang berbeda. Skema ruangan yang digunakan tampak seperti pada Gambar 6. Format data yang dihasilkan oleh *Agent* adalah sebagai berikut:

<AGENT>, <KELEMBAPAN>, <SUHU>, <GAS>

Sebagai contoh data yang diperoleh pada *Agent* yaitu dengan nama Node 1,sensor tekanan adalah 50, suhu 29, dan intensitas gas adalah 1.5. Sehingga data yang terbentuk untuk salah satu *Agent* adalah sebagai berikut

NODE 1,50,29,1.5

Selanjutnya sistem tertanam akan mengirimkan string tersebut kedalam modul XBee dan ditransmisikan ke modul XBee penerima (*Coordinator*). Gambar 5.3 menunjukkan koneksi antara arduino dengan XBee. Data frame 7E 00 14 01 01 FF 00 4E 4F 44 45 31 2C 35 30 2C 32 39 2C 31 2E 35 C0 adalah frame yang ditransmisikan oleh data yang dibentuk oleh *Agent*. Komunikasi sistem

tertanam dengan data center dapat berlangsung secara kontinu hingga ada intruksi selanjutnya dari *Coordinator* untuk menghentikan pengiriman data..

Setelah komunikasi serial selesai terjadi antara *Coordinator* dan *Agent* akan diperoleh informasi frame seperti pada tabel 1. Selain data frame juga akan dapat diketahui RSSI dari *Agent*, dimana untuk frame data pada tabel 2. akan di dapatkan data *Received Signal Strength Indicator* (RSSI) sebesar -38db dan *payload data*. Data RSSI berbanding terbalik dengan jarak antara *Agent* dan *Coordinator*, semakin dekat jarak antara *Coordinator* dan *Agent* maka nilai RSSI akan semakin besar, namun semakin jauh jarak *Agent* dan *Coordinator* maka akan menyebabkan nilai RSSI akan semakin kecil. Data hasil yang dikirim kepada *Coordinator* selanjutnya diproses dengan melakukan parsing dan melakukan *decode* data *payload* dari format heksadesimal menjadi format data ASCII.

Setelah proses parsing telah berhasil dilakukan maka akan diperoleh data-data RSSI, tekanan, suhu dan intensitas gas pada suatu ruangan. Sebagai contoh , salah satu data pada tabel 3, data ke 3 jika dilakukan parsing akan diperoleh informasi sebagai berikut:

38 NODE1,50,29,1.5

Dimana kombinasi string tersebut mengandung informasi-informasi

1. RSSI : -38 (*Received_Signal_Strength_Indicator*)
2. *Agent* : *Node* 1
3. Tekanan: 50
4. Suhu : 29
5. Intensitas Gas: 1.5

Tabel 1. Frame serial *Agent*

frame data [0] is F	frame data [10] is 35
frame data [1] is FF	frame data [11] is 30
frame data [2] is 26	frame data [12] is 2C
frame data [3] is 2	frame data [13] is 32
frame data [4] is 4E	frame data [14] is 39
frame data [5] is 4F	frame data [15] is 2C
frame data [6] is 44	frame data [16] is 31
frame data [7] is 45	frame data [17] is 2E
frame data [8] is 31	frame data [18] is 35
frame data [9] is 2C	

Tabel 2. Payload pada frame data *Agent*

payload [0] is 4E	payload [7] is 30
payload [1] is 4F	payload [8] is 2C
payload [2] is 44	payload [9] is 32
payload [3] is 45	payload [10] is 39
payload [4] is 31	payload [11] is 2C
payload [5] is 2C	payload [12] is 31
payload [6] is 35	payload [13] is 2E
payload [14] is 35	

Tabel 3. Parsing frame data *Agent*

Data	Frame
1	#RSSI1#50#RSSI2#, #PAYLOAD1#4E4F4445312C35302C32392C312E35#PAYLOAD2#
2	#RSSI1#51#RSSI2#, #PAYLOAD1#4E4F4445312C35302C32392C312E35#PAYLOAD2#
3	#RSSI1#38#RSSI2#, #PAYLOAD1#4E4F4445312C35302C32392C312E35#PAYLOAD2#
4	#RSSI1#41#RSSI2#, #PAYLOAD1#4E4F4445312C35302C32392C312E35#PAYLOAD2#
5	#RSSI1#50#RSSI2#, #PAYLOAD1#4E4F4445312C35302C32392C312E35#PAYLOAD2#
6	#RSSI1#50#RSSI2#, #PAYLOAD1#4E4F4445312C35302C32392C312E35#PAYLOAD2#
7	#RSSI1#38#RSSI2#, #PAYLOAD1#4E4F4445312C35302C32392C312E35#PAYLOAD2#

Tabel 4. Pengamatan *Agent*

No	RSSI (dBm)	Delay (ms)	Jitter (ms)	Packet Loss (%)	Jarak (m)
1	-20	0	0	0%	1
2	-41	0	0		10
3	-55	0	0		20
4	-68	0	0		30
5	-72	0	0		40
6	-77	78	5		50
7	-86	84	6		60
8	-89	93	19		70
9	-93	262	45		80
10	-98	437	67		90

Tabel 5. Kualitas Layanan Data ITU

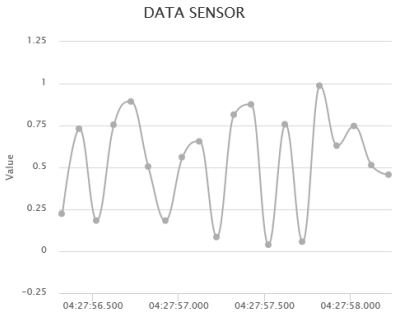
	Delay (ms)	Jitter(ms)	Packet Loss (%)
Baik	<150	<20	<1
Cukup	150 – 300	20-50	1-2
Buruk	> 300	>50	>2

Tabel 6.Data tekanan, suhu dan gas dari *Agent*

Agent	Sensor Tekanan	Sensor Suhu	Sensor Gas
***Agent* 1**	50	29.4	1.5
***Agent* 2**	50	29.7	1.4
***Agent* 3**	50	29.1	1.6
***Agent* 4**	50	28.9	1.5
***Agent* 5**	50	30	1.5

Tabel 7. *Delay, Jitter, Packet Loss Agent*

No	Agent 1			Agent 2			Agent 3			Agent 4		
	Delay (ms)	Jitter (ms)	Paket Loss	Delay (ms)	Jitter (ms)	Paket Loss	Delay (ms)	Jitter (ms)	Paket Loss	Delay (ms)	Jitter (ms)	Paket Loss
1	179	8		172	11		180	3		1003	10	
2	171	9		183	12		177	3		1013	10	
3	180	8		171	1		180	11		1003	10	
4	172	5		172	39		169	50		1013	1	
5	177	4		133	87		119	121		1014	1	
6	173	8		220	48		240	70		1013	10	
7	181	11		172	9		170	9		1003	10	
8	170	2		181	2		179	9		1013	9	
9	172	8		179	8		170	10		1004	9	
10	180	0	0%	171	10	0%	180	9	0%	1013	0	0%
11	180	10		181	11		171	2		1013	9	
12	170	1		170	11		173	8		1004	19	
13	171	9		181	9		181	11		1023	20	
14	180	9		172	1		170	2		1003	0	
15	171	10		171	10		172	9		1003	11	
16	181	11		181	11		181	1		1014	12	
17	170	1		170	10		180	10		1002	11	
18	171	9		180	10		170	12		1013	1	
19	180	1		170	1		182	13		1014	11	
20	179	9		171	10		169	11		1003	20	



Gambar 8. Visualisasi dari Sensor pada salah satu *agent*

Data hasil tersebut selanjutnya dikirim ke *data center* melalui protocol TCP/IP. Selanjutnya pada *database* akan disimpan setiap informasi yang dikirim agar dapat dianalisas lebih lanjut pada tahapan selanjutnya. Bentuk visualisasi presentasi dari data dapat dilihat pada gambar 8.

Pada percobaan, *Coordinator* dan *Agent* diletakan pada satu ruangan yang sama dan kondisi LoS. Sistem tertanam melakukan pertukaran informasi data lingkungan. Pada skenario tersebut dihasilkan data pengamatan pada tabel 4.

Berdasarkan standar kualitas layanan data pada jaringan komputer yang dibuat oleh ITU yang di tunjukan pada tabel 5. Hasil pengujian skenario di atas, *Agent* dapat mengirimkan paket berkualitas baik secara *real time* pada level daya RSSI -72 dBm dengan jarak transmisi 70 meter. Pada level RSSI terima -93 dBm dengan jarak 80, data masih dapat diterima namun komunikasi tidak berjalan secara *real time* dan berkualitas cukup. Pada lebel RSSI -98 dBm dengan jarak 90, data yang diterima oleh *Coordinator* termasuk sangat buruk dan komunikasi tidak berjalan secara real time.

Berdasarkan hasil percobaan dari *Agent* pada kondisi LoS seperti skenario sebelumnya maka didapatkan jarak ideal dari *Coordinator* dan *Agent*, yaitu 70 m. Namun pada skenario yang kedua, Seperti tampak pada gambar 6. masing-masing *Agent* disebar ke dalam *room* 1, *room* 2, *room3*, dan *room* 4 dengan jarak antara *Agent* ke *Coordinator* bervariasi dan beberapa *Agent* lebih dari 70 m, sehingga digunakan router sebagai *forwarding* informasi sehingga informasi dapat dikirimkan ke *Coordinator* walaupun melebihi jarak 70 m. Berdasarkan tabel 7. tersebut dapat ditarik analisis bahwa, semakin jauh jarak antara *Agent* dan *Coordinator* akan berpengaruh terhadap kualitas layanan dari sistem tertanam. Hal ini dibuktikan dengan *delay* rata-rata yang diperoleh *Agent* 1 adalah 175.4 ms, *Agent* 2 adalah 175.5 ms, *Agent* 3 adalah 175,65 ms dan *Agent* 4 adalah 1009.2 ms. Sedangkan *jitter* rata-rata yang diperoleh *Agent* 1 adalah 6.65 ms, *Agent* 2 adalah 15.55 ms, *Agent* 3 adalah 18.7 ms dan *Agent* 4 adalah 9.2 ms. *PacketLoss* untuk *Agent* 1, *Agent* 2, *Agent* 3, dan *Agent* 4 adalah 0%. Berdasarkan data tersebut hanya *Agent* 4 yang memiliki kualitas layanan yang rendah, hal ini terjadi karena kondisi ruangan yang tidak LoS dan jarak yang jauh dari *Agent* 4 ke *Coordinator* walaupun antara *Agent* tersebut dengan *Coordinator* telah dijembatani oleh Router seperti pada gambar 5, namun tetap saja akan terdapat penurunan performa pada *room* 4.

6. Kesimpulan

1. Protokol Zigbee dapat menjadi basis *engine* dari *Wireless Sensor Network*.
2. *Proxy Agent Layer* mampu menghubungkan sistem tertanam dengan *jaringan Internet Protocol.*
3. Jarak maksimum antara *Coordinator* dan *Agent* yang memiliki kualitas delay baik berdasarkan standar dari ITU pada hasil pengujian adalah 70 meter.
4. Nilai RSSI untuk kualitas layanan maksimal dari WSN adalah > -89 dBm

REFERENSI

[1] Martin Keen, “The protocol that powers the Internet of Things - IBM Impact Blog,” 04-Apr-2014. [Online]. Available: https://www-304.ibm.com/connections/blogs/aim/entry/protocol_that_powers_the_internet_of_things?lang=en_us. [Accessed: 27-Apr-2015].

[2] A. S. Namin, W. Shen, and H. Ghenniwa, “Web Services/*Agent* -Based Model for Inter-Enterprise Collaboration,” in *Emerging Solutions for Future Manufacturing Systems*, L. M. Camarinha-Matos, Ed. Springer US, 2005, pp. 231–240.

[3] P. S. Saravanakumar, N. Mehanathen, and M. R. Rajagopalan, "Heterogeneous synchronization layer for web services," in *2014 16th International Conference on Advanced Communication Technology (ICACT)*, 2014, pp. 1270–1273.
[4] Y. Liu, Z. Wang, J. Zhao, and Z. Sun, "A wireless sensor network node design based on ZigBee protocol," in *International Conference on Automatic Control and Artificial Intelligence (ACAI 2012)*, 2012, pp. 325–328.
[5] H. Fang and W. Xuechun, "Research on embedded network scheme based on quantum key distribution," in *Cross Strait Quad-Regional Radio Science and Wireless Technology Conference (CSQRWC), 2011*, 2011, vol. 2, pp. 1689–1691.
[6] Safuwan, "Integrasi Perangkat Lunak Enterprise Resource Planning (Erp) Dengan Menggunakan Metode Service Oriented Architecture (Soa)," *Integrasi Perangkat Lunak Enterp. Resour. Plan. Erp Dengan Menggunakan Metode Serv. Oriented Archit. Soa Enterp. Resour. Plan. Erp Softw. Integr. Using Serv. Oriented Archit. Soa Method Oriented Archit. Soa*, vol. 0, no. 0, Mar. 2010.

Ahmad Heryanto, memperoleh gelar S.Kom. dari Universitas Sriwjaya Palembang pada tahun 2010 dan gelar M.T. dari Institut Teknologi Sepuluh Nopember Surabaya pada tahun 2014. Saat ini sebagai Staf Pengajar program studi Sistem Komputer Universitas Sriwijaya.

Siti Nurmaini, memperoleh gelar S.T. dari Universitas Sriwijaya Palembang pada tahun 1991 dan gelar M.T. dari Institut Teknologi Bandung pada tahun 1998 Serta gelar Dr. dari Universiti Teknologi Malaysia pada tahun 2011. Saat ini sebagai staf Pengajar Program Studi Sistem Komputer Universitas Sriwijaya.

Tri Wanda Septian, memperoleh gelar S.Kom. dari Universitas Sriwjaya Palembang pada tahun 2013. Saat ini sebagai Staf Pengajar program studi Sistem Komputer Universitas Sriwijaya.

Ricy Firnando, memperoleh gelar S.Kom. dari Universitas Sriwjaya Palembang pada tahun 2012. Saat ini sebagai Staf Pengajar program studi Sistem Komputer Universitas Sriwijaya.