

Implementasi Push Message Dengan Menggunakan Restful Web Service Pada Komunikasi Wireless Sensor

Rino Nugroho¹, Mahar Faiqurahman², Zamah Sari³

^{1,2,3}Teknik Informatika/Universitas Muhammadiyah Malang

rinonugroho46@gmail.com¹, maharf@gmail.com², abdzamahsari@gmail.com³

Abstrak

Wireless Sensor Network (WSN) adalah sebuah jaringan nirkabel yang terdiri dari satu atau lebih node bahkan berjumlah ribuan. Node dalam wireless sensor network (WSN) terdiri dari sensor node dan sink node. Penggunaan wireless sensor pada jaringan dapat membentuk sebuah node yang dapat saling berkomunikasi. Proses komunikasi pada umumnya menggunakan mekanisme pull yang didahului proses permintaan data dari sink node ke sensor node yang menyediakan data hasil sensing. Pada beberapa arsitektur wireless sensor node, mekanisme pull ini dinilai kurang efektif karena sink node harus melakukan permintaan data terlebih dahulu ke sensor node. Sebagai alternatif, mekanisme push message dapat digunakan untuk mengirimkan data hasil sensing dalam interval waktu yang sudah pasti atau ditentukan. Dalam penelitian ini diimplementasikan mekanisme push message dengan menggunakan restful web service pada komunikasi wireless sensor. Hasil pengujian pada pengiriman data secara push didapatkan pengiriman data ke sink node secara bergantian sesuai dengan urutan alamat tujuan yang terdaftar atau tersimpan di memory sensor node. Serta dalam melakukan pengiriman data menjadi efisien tanpa adanya dilakukan permintaan data setiap saat.

Kata Kunci: WSN, Node, Komunikasi

Abstract

Wireless Sensor Network (WSN) is a wireless network consisting of one or more nodes even numbering thousands. The nodes in the wireless sensor network (WSN) consist of sensor nodes and sink nodes. The use of wireless sensors on the network can form a node that can communicate with each other. The communication process generally uses a pull mechanism that precedes the data query process from the node to node sensor that provides sensing data. In some wireless sensor node architecture, this pull mechanism is considered less effective because the node sink must first request data to the sensor node. Alternative, a push message mechanism can be used to transmit sensed data within specified or determined time intervals. In this research is implemented push message mechanism by using restful web service in wireless sensor communications. Test results on the delivery of data by push data transmission obtained to sink nodes alternately in accordance with the order of destination address listed or stored in memory sensor node. And in doing data delivery to be efficient in the absence of data requests at any time.

Keywords: WSN, Node, Communication

1. Pendahuluan

Jaringan sensor nirkabel atau *Wireless Sensor Network (WSN)* adalah sebuah jaringan nirkabel yang terdiri dari satu atau lebih *node* bahkan berjumlah ribuan. Teknologi ini dapat disusun hingga membentuk *node* pada jaringan yang dapat saling berkomunikasi [1]. Pada setiap sensor *node* dalam *Wireless Sensor Network (WSN)* terdiri dari lima komponen yaitu *kontroller* atau *mikrokontroler*, memori, sensor, perangkat komunikasi dan catu daya. *Sensor Node* tersebut memiliki kemampuan untuk menyebarkan data yang dikumpulkan ke *node* lain yang berdekatan [2]. Komponen-komponen *node* dalam jaringan sensor nirkabel dapat direpresentasikan melalui alat arduino dan raspberry, dimana alat tersebut sebagai pendukung dalam komunikasi data sensor.

Arduino adalah pengendali *mikro single-board* yang bersifat *open source*, diturunkan dari *Wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardwarenya* memiliki prosesor Atmel AVR dan *softwarenya* memiliki bahasa pemrograman

sendiri [3]. Sedangkan Raspberry adalah komputer berukuran kartu kredit atau ATM, yang dikembangkan oleh Raspberry Pi Foundation yang berbasis di Inggris. Raspberry Pi (atau Raspi), digunakan untuk menjalankan aplikasi *desktop*, terutama piawai dalam memainkan video definisi tinggi. Raspberry Pi Model B, 512MB RAM. Antarmuka: HDMI, Ethernet & 2 USB ports [4].

Komunikasi jaringan wireless sensor dalam implementasiannya sebagian besar digunakan pada lingkungan terbuka. Dalam hal berkomunikasi dibutuhkan proses pengiriman secara teratur dan efisien dikarenakan pada *wireless sensor* melakukan pengiriman data ke banyak *node*, memiliki *source* dan daya penyimpanan power yang terbatas, serta pengiriman data dengan ukuran yang besar ataupun banyak. Komunikasi yang efisien dapat mempengaruhi dalam menghemat sumber daya terutama daya listrik, mendapatkan data yang stabil sehingga data yang dibutuhkan sesuai dengan kebutuhan.

Restful Web Service pada dasarnya komunikasi yang menggunakan metode operasional yang sederhana yang dimiliki oleh protokol standar *HTTP* sebagai parameter untuk *URL (Uniform Resource Locator)* yang dikirimkan dari *client* ke server. Pada sistem ini memiliki kemudahan dalam mengintegrasikan data dengan baik meskipun berbeda anatra *platform* perangkat keras maupun perangkat lunak (sistem operasi) [5].

Dalam komunikasi data terdapat mekanisme pengiriman pesan *pull* dan *push*. Didalam mekanisme *pull* setiap pengiriman pesan didahului oleh permintaan (*request*) dari *client*. Namun dalam hal komunikasi tersebut diperlukan waktu yang relatif lama untuk memproses komunikasi permintaan (*request*) dan pengiriman (*send*) pesan, serta terjadinya penggunaan penyimpanan sementara (*memory*) yang lebih besar pada komponen pengiriman data dalam mengatur alur komunikasi jaringan. Sedangkan pada mekanisme *push* setiap pengiriman pesan dilakukan proses permintaan (*request*) terlebih dahulu untuk melakukan pendaftaran alamat atau data client, setelah proses tersebut client tidak perlu melakukan proses permintaan kembali. Karena alamat atau data tujuan sudah terekam sehingga data yang dikirimkan langsung atau terjadwal [6].

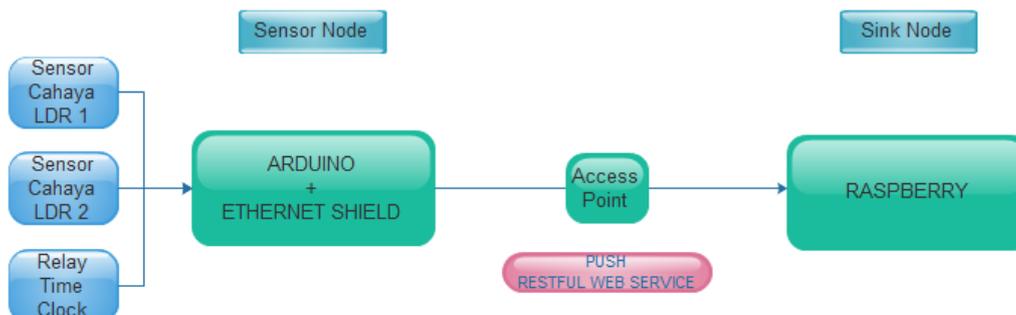
Dari latar belakang di atas maka dalam penelitian ini diimplementasikan *push message* dengan menggunakan *Restful Web Service* pada komunikasi *wireless* sensor antara komponen arduino (*node sensor*) dengan komponen Raspberry pi 2 (*sink node*). Arsitektur *Web Service Restful* ini akan diimplementasikan pada sistem komunikasi antara *node sensor* dan *sink node*. Serta untuk komunikasi antara *node sensor* dan *sink node* menggunakan jaringan *wireless*.

2. Metode Penelitian

Metode Penelitian akan menjelaskan tentang perancangan sistem, perancangan perangkat dan perancangan alur kerja sistem.

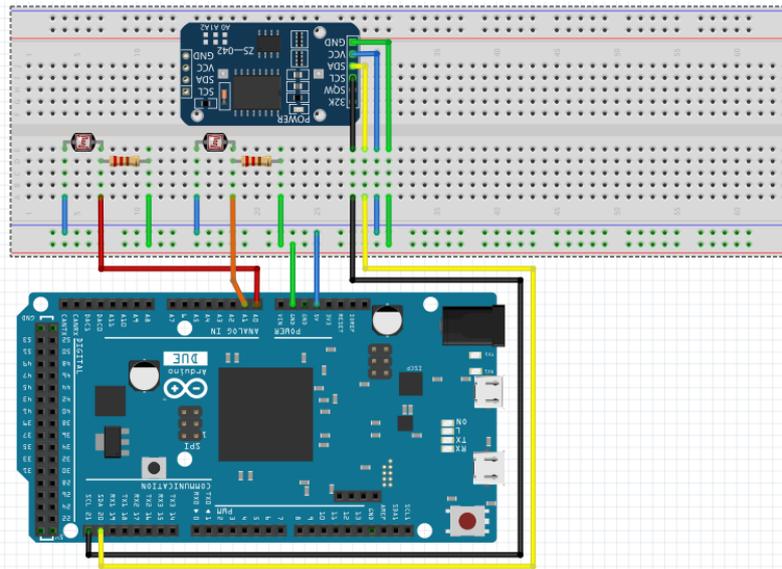
2.1. Perancangan Sistem

Pada Gambar 1 dijelaskan terdiri dari dua buah sensor Cahaya LDR, Relay Time Clock (RTC), Arduino dan Ethernet Shield yang berfungsi sebagai *sensor node*. Access Point untuk media pengiriman menggunakan komunikasi *wireless*. Raspberry sebagai *sink node*. Alur dari kerja tersebut adalah kedua sensor melakukan proses pengambilan data sensor (*sensing*), lalu *sensor node* akan menyimpan data di SD Card, setelah itu *sensor node* akan mengirimkan data ke *sink node*. Pada *sink node* akan melakukan proses penyimpanan data sensor berupa nilai jumlah cahaya yang diterima. Sedangkan pada Relay Time Clock berfungsi sebagai penyimpanan waktu seperti hari, tanggal dan jam.



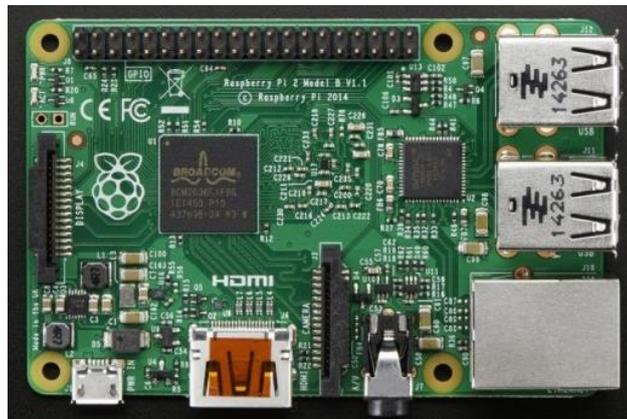
Gambar 1. Alur Umum Sistem

2.2. Perancangan Perangkat



Gambar 2. Perancangan Perangkat Arduino

Perancangan perangkat pada Gambar 2 berperan sebagai *sensor node* yang terdiri dari Arduino Due, dua buah sensor cahaya LDR, ethernet shield, dan Relay Time Clock (RTC) Sehingga dapat berfungsi sebagai proses data, pengiriman data dan penerimaan koneksi dari *sink node*.



Gambar 3. Perangkat Raspberry

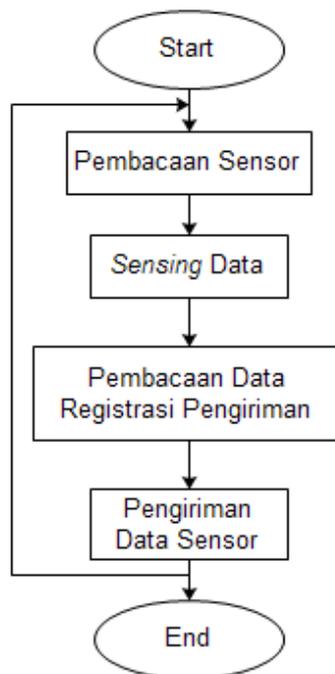
Pada Gambar 3 adalah Raspberry berperan sebagai *sink node* yang berfungsi sebagai penerima data dari *sensor node* dan kirim permintaan koneksi ke *sensor node*. Berikut komponen pendukung utama raspberry adalah:

- Kabel Power Listrik 5V=2A
- Memory SD card dengan Tipe Micro SD (class 10, 8 Gb)
- Kabel HDMI to VGA
- Keyboard dan Mouse Wireless
- USB Wireless Wi-fi
- Sistem Operasi (OS) Raspbian

2.3. Perancangan proses

A. Sensor Node

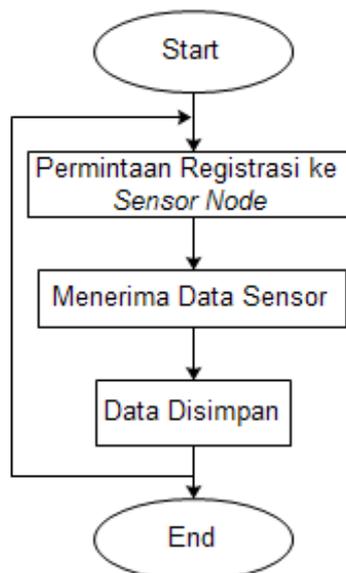
Tahapan alur proses pada *sensor node* meliputi *sensing* data sensor, pengiriman data sensor, dan proses *response*. Pada Gambar 4 dijelaskan alur proses *sensing* dan pengiriman data sensor.



Gambar 4. Flowchart Alur Proses pada Sensor Node

B. Sink Node

Tahapan alur proses *sink node* adalah melakukan proses permintaan registrasi ke *sensor node* dan menerima data sensor dari *sensor node*. Gambar 5 berikut *flowchart* alur proses dari *sink node*.

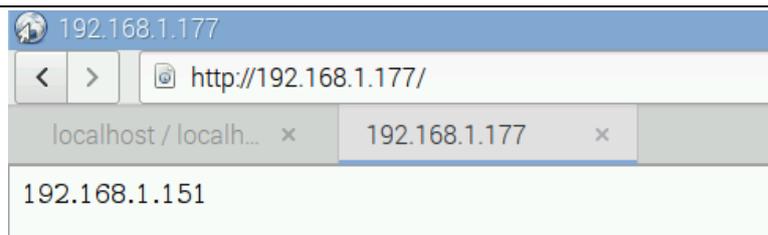


Gambar 5. Flowchart Alur Proses pada Sink Node

3. Hasil Penelitian dan Pembahasan

3.1. Pengujian Koneksi Antara Sensor Node Dan Sink Node.

Pada tahap pengujian ini dilakukan koneksi dengan wireless, yang bertujuan *sensor node* dengan *sink node* saling berkomunikasi, dimana *sensor node* berperan sebagai pengirim data serta *sink node* berperan sebagai penerima data. Hasil dari pengujian ini adalah *sensor node* akan mengirimkan respon berupa pengiriman *ip address sink node* yang telah disimpan pada SD card *sensor node* sebelumnya, seperti pada Gambar 6.



Gambar 6. Hasil Pengujian Koneksi

3.2. Pengujian Pengiriman Data Menggunakan Tiga Buah Sink Node

Tahapan pengujian ini dilakukan dengan melakukan koneksi dan pengiriman data ke tiga buah *sink node*. Hasil dari pengujian tersebut, *sensor node* akan melakukan bergantian pengiriman sesuai urutan *ip address* pada penyimpanan *SD Card*. Proses pengiriman data sensor dari satu *sink node* ke *sink node* lainnya membutuhkan waktu sekitar 1 sampai 2 detik. Namun apabila dalam proses pengiriman data mengalami putus koneksi maka dibutuhkan 2 sampai 3 detik untuk pengiriman data ke *ip address sink node* berikutnya. Berikut Gambar 7 monitoring pada sensor node saat melakukan proses pengiriman data sensor ke tiga buah sink node.

```

Initializing SD card...initialization done.
server is at 192.168.1.177
dsensor.txt exists.
Removing dsensor.txt...
ldr 1: 41 ldr 2: 0
ldr 1: 437 ldr 2: 974
ldr 1: 901 ldr 2: 967
ldr 1: 760 ldr 2: 393
ldr 1: 232 ldr 2: 0
ldr 1: 347 ldr 2: 516
ldr 1: 131 ldr 2: 1023
ldr 1: 751 ldr 2: 954
ldr 1: 972 ldr 2: 742
ldr 1: 955 ldr 2: 643
data sudah tersimpan
dsensor.txt exists.
Pembacaan data di file address.txt
Isi= ip.-192.168.1.151.-192.168.1.152.-192.168.1.153.-

waktu = 16:51:55
server2 = 192.168.1.151
connected
dataString= [[41,0],[437,974],[901,967],[760,393],[232,0],[347,516],[131,1023],[751,954],[972,742],[955,643]]
GET /ldr/add.php?dsensor=[[41,0],[437,974],[901,967],[760,393],[232,0],[347,516],[131,1023],[751,954],[972,742],[955,643]]
Free memory = 77615

waktu = 16:51:56
server2 = 192.168.1.152
connected
dataString= [[41,0],[437,974],[901,967],[760,393],[232,0],[347,516],[131,1023],[751,954],[972,742],[955,643]]
GET /ldr/add.php?dsensor=[[41,0],[437,974],[901,967],[760,393],[232,0],[347,516],[131,1023],[751,954],[972,742],[955,643]]
Free memory = 77615

waktu = 16:51:57
server2 = 192.168.1.153
connected
dataString= [[41,0],[437,974],[901,967],[760,393],[232,0],[347,516],[131,1023],[751,954],[972,742],[955,643]]
GET /ldr/add.php?dsensor=[[41,0],[437,974],[901,967],[760,393],[232,0],[347,516],[131,1023],[751,954],[972,742],[955,643]]
Free memory = 77615

```

Gambar 7. Monitoring Proses Pengiriman Data Sensor

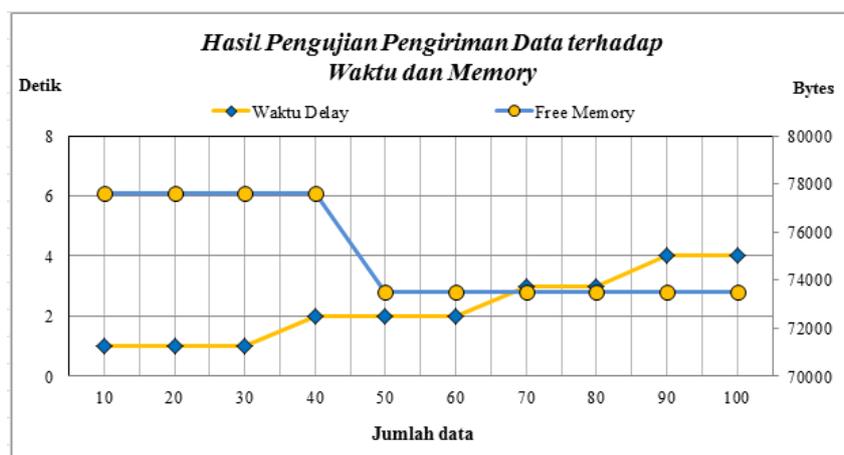
3.3. Pengujian Delay Pengiriman Data

Tahapan pengujian ini dilakukan dengan mengirimkan data dengan jumlah yang berbeda dari *sensor node* ke *sink node*. Jumlah data yang akan dilakukan percobaan adalah mulai dari jumlah data 10, 20, 30, 40, 50, 60, 70, 80, 90, dan 100.

Pada pengujian ini bertujuan untuk mengetahui waktu *delay* pengiriman dengan penerimaan data. Serta mengetahui jumlah *free memory* pada *sensor node*. berikut Tabel 1 hasil pengujian.

Tabel 1. Hasil Pengujian

No.	Jumlah Data	Waktu Delay (Detik)	Jumlah Free Memory(Bytes)
1.	10	1	77615
2.	20	1	77615
3.	30	1	77615
4.	40	2	77615
5.	50	2	73519
6.	60	2	73519
7.	70	3	73519
8.	80	3	73519
9.	90	4	73519
10.	100	4	73519



Gambar 8. Grafik Hasil Pengujian

Pada Gambar 8 dapat dijelaskan bahwa grafik waktu *delay* menunjukkan peningkatan waktu yang diperlukan untuk pengiriman data. Banyaknya data dapat mempengaruhi lamanya waktu yang diperlukan untuk pengiriman data dari *sensor node* ke *sink node*. Sedangkan grafik *free memory* menunjukkan penurunan kapasitas *memory* pada *sensor node* saat dilakukan pengiriman data, dikarenakan semakin banyak data yang dikirim maka kapasitas *memory sensor node* mengalami pengurangan jumlah *memory* pada *sensor node*.

3.4. Pengujian Penggunaan MultiThread

Pada pengujian ini dilakukan menjalankan instruksi multithread dengan pemanggilan *ip address sensor node* dari browser *sink node*. Setelah dilakukan pengujian maka didapatkan hasil bahwa selama tidak ada proses permintaan koneksi dari *sink node* ke *sensor node* maka akan terjadi proses *push* dimana terjadi proses *sensing* sensor dan melakukan pengiriman ke *sink node* sesuai urutan penyimpanan *ip address* di *Sd card sensor node*. Apabila ada proses koneksi dari *sink node*, maka *sensor node* akan melakukan interupsi untuk melakukan proses respon permintaan koneksi dari *sink node*. Akan tetapi proses interupsi tersebut dilakukan setelah proses *sensing* dan penyimpanan data sensor selesai dilakukan seperti pada Gambar 9.

```

ldr 1: 751 ldr 2: 899
ldr 1: 764 ldr 2: 927
data sudah tersimpan
dsensor.txt exists.
new client
GET / HTTP/1.1
Host: 192.168.1.177
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (Macintosh; ARM Mac OS X) AppleWebKit/538.15 (KH
Accept-Encoding: gzip, deflate
Connection: Keep-Alive

```

Gambar 9. Proses Interupsi Saat Adanya Koneksi

Setelah proses tersebut maka melakukan proses respon koneksi, dengan cara melakukan perbandingan terlebih dahulu bahwa *ip address* tersebut sudah ada atau belum pada *Sd Card sensor node*. Selanjutnya akan terjadi proses penyimpanan *ip address*, dan melakukan pengiriman *HTTP respon*. Proses pengiriman data sensor setelah adanya interupsi tidak langsung dikirimkan melainkan melakukan proses *sensing* data sensor kembali, dikarenakan pada arduino memiliki *resource* dan proses yang terbatas. Proses di arduino akan melakukan terlebih dahulu menyelesaikan proses inputan. Meskipun waktu interupsi digunakan, arduino akan melakukan proses interupsi dahulu. Berikut Gambar 10 yang mempresentasikan hasil pengujian diatas yang dapat dilihat pada Serial Monitor Arduino IDE.

```

Waiting...
data ketemu, tidak tulis

client disconnected
dsensor.txt exists.
Removing dsensor.txt...
ldr 1: 289 ldr 2: 134
ldr 1: 423 ldr 2: 331

server2 = 192.168.1.152
Connection Failed

waktu = 20:07:48
server2 = 192.168.1.151
connected
dataString= [[316,148],[339,143],[818,916],[530,486],[243,10],[793,852],[703,771],[262,22],[737,783],[828,914]]
Free memory = 77631
dsensor.txt exists.
Removing dsensor.txt...
ldr 1: 351 ldr 2: 73
ldr 1: 837 ldr 2: 883

```

Gambar 10. Proses Respon Koneksi, Sensing Sensor dan Pengiriman Data

4. Kesimpulan

Setelah dilakukan implementasi dan pengujian dari tugas akhir yang berjudul *Implementasi Push Message dengan Menggunakan Restful Web Service pada Komunikasi Wireless Sensor* maka didapatkan kesimpulan bahwa mekanisme *push message* menggunakan *restful web service*, dapat dilakukan dengan proses pengiriman data hasil sensing oleh *sensor node* melalui *HTTP GET*. Proses pengiriman data sensor dari *sensor node* ke *sink node* berdasarkan *ip address* yang sudah tersimpan didalam *memory* yang terdapat pada *sensor node*. Sedangkan dari hasil pengujian proses pengiriman data sensor dari *sensor node* ke *sink node* melalui mekanisme *push message*, didapatkan kesimpulan bahwa waktu *delay* pengiriman data dan penggunaan *memory* pada *sensor node* dipengaruhi oleh jumlah data sensor dari hasil sensing. Sehingga semakin banyak jumlah data yang dikirimkan maka waktu pengiriman data akan semakin lama dan kapasitas *memory* yang tersisa semakin berkurang.

Dalam Implementasi penelitian tugas akhir ini masih belum sempurna dan masih terdapat kekurangan. Oleh sebab itu diperlukan pengembangan lebih lanjut. Adapun saran yang dapat dilakukan pengembangan adalah melakukan analisa, solusi dalam mengurangi pemakaian *memory* pada *sensor node* yang tidak akan mempengaruhi pada pengiriman jumlah data. Serta pengembangan dalam metode lain dan penggunaan komponen lainnya.

Referensi

- [1] M. Fuad *et al.*, "Analisis Performansi Protokol Routing Gpsr Pada Jaringan Sensor Nirkabel Performance Analysis of Gpsr Routing Protocol in Wireless Sensor Networks," vol. 2, no. 2, pp. 6336–6352, 2015.
- [2] Kautsar MS, "Bahasa Pemrograman Arduino," *Stikom*, pp. 6–35.
- [3] A. Fatoni and D. B. Rendra, "Perancangan Prototipe Sistem Kendali Lampu Menggunakan Handphone Android Berbasis Arduino," *J. PROSISKO Vol. 1 Sept. 2014*, vol. 1, no. September, pp. 23–29, 2014.

-
- [4] Nurfalah, M. (2014), Mikrokontroler_makalah_arduino_and_raspb.
- [5] A. Nugroho and K. Mustofa, "Perbandingan antara 'Big' Web Service dengan Restful Web Service untuk Integrasi Data Berformat GML," *J. Inform.*, vol. 11, no. 1, pp. 8–16, 2012.
- [6] J. Cao, X. Feng, J. Lu, H. C. B. Chan, and S. K. Das, "Reliable message delivery for mobile agents: Push or pull?," *IEEE Trans. Syst. Man, Cybern. Part A Systems Humans.*, vol. 34, no. 5, pp. 577–587, 2004.