



# Plagiarism Checker X Originality Report

**Similarity Found: 26%**

Date: Wednesday, January 25, 2017

Statistics: 431 words Plagiarized / 1688 Total words

Remarks: Medium Plagiarism Detected - Your Document needs Selective Improvement.

---

Implementasi Pull Message Dengan Menggunakan Restful Web Service Pada Komunikasi Wireless Sensor Abstrak Wireless Sensor Network adalah infrastruktur jaringan nirkabel yang terdiri dari beberapa node sensor. Arduino adalah kit elektronik atau papan rangkaian elektronik open source yang di dalamnya terdapat komponen utama yaitu sebuah chip mikrokontroler dengan jenis AVR dari perusahaan Atmel. Raspberry Pi adalah suatu perangkat mini computer berukuran sebesar kartu kredit.

Implementasi penggunaan arduino dan raspberry pi banyak digunakan dalam aplikasi wireless sensor. Restful Web Service pada dasarnya menggunakan metode operasional yang sederhana yang dimiliki oleh protokol HTTP untuk berkomunikasi. Pada penelitian ini digunakan Arduino sebagai sensor node dan Raspberry Pi sebagai sink node untuk komunikasi wireless sensor.

Pada sensor node Multithreading sangat membantu untuk menjalankan fungsi secara paralel, seperti fungsi sensor, memory heap, maupun web server. Hasil pengujian menunjukkan interval sensing, ukuran data, dan jumlah sink node pengaruhnya terhadap free memory heap. Selain itu pengaruh ukuran data terhadap request dan response time.

Kata kunci: Arduino, Raspberry Pi, Multithreading, WSN, heap Abstract Wireless Sensor Network is a wireless network infrastructure that consists of multiple

sensor nodes. Arduino is an electronic kit or open source electronic circuit board in primary component of a chip microcontroller with the AVR type of AVR company. Raspberry Pi is a mini computer device size of a credit card. Implementation of use arduino and raspberry pi widely used in wireless sensor applications.

Restful Web Service basically using simple operational method which is owned by the HTTP protocol to communicate. On this research Arduino use as sensor node and Raspberry Pi as sink node for wireless sensor communication. Multithreading on the sensor node helpful to run in parallel functions such as sensor function, memory heap, and web server.

Result of test showed sensing interval, data size, and the number of sink node influence on free memory heap. Beside the effect of data size on request and response time. Keywords: Arduino, Raspberry Pi, Multithreading, WSN, heap  
1.Pendahuluan Wireless Sensor Network (WSN) adalah infrastruktur jaringan nirkabel yang terdiri dari beberapa node sensor.

Teknologi WSN dapat digunakan untuk memonitor beberapa hal seperti temperatur, kelembapan, jarak benda, cahaya dan lain sebagainya[1]. Wireless Sensor menggunakan 2 standar komunikasi wireless. Protokol IEEE 802.15.4 dirancang untuk keperluan wireless sensor network. Teknologi ini beroperasi pada frekuensi 2,4 GHz untuk 16 kanal. Memiliki jangkauan sekitar 10 – 100 m dengan data rate maksimal 250 Kbit/s.

Salah satu vendornya adalah zigbee. Protokol IEEE 802.11(Wireless LAN) ditransmisikan dengan frekuensi 2.4 – 2.48 GHz. Wireless LAN(802.11b) pertama kali mempunyai bandwidth 11Mbps[2]. Arduino adalah rangkaian elektronik open source yang di dalamnya terdapat komponen utama yaitu sebuah chip mikrokontroler dengan jenis AVR dari perusahaan Atmel.

Terdapat bootloader yang akan menangani upload program. Bahasa yang digunakan adalah bahasa C[3]. Raspberry Pi adalah suatu perangkat mini computer berukuran sebesar kartu kredit. Raspberry Pi memiliki system Broadcom BCM2835 chip(SoC), yang mencakup ARM1176JZF-S 700 MHz processor.

VideoCore IV GPU, dan awalnya dikirim dengan 256 megabyte RAM, kemudian upgrade ke 512MB. Termasuk built-in hard disk atau, menggunakan kartu SD untuk booting dan penyimpanan jangka panjang[4]. Implementasi penggunaan arduino dan raspberry pi banyak digunakan dalam aplikasi wireless sensor.

Digunakannya **single board computer raspberry pi** sebagai gateway atau base station karena dapat menekan biaya pengeluaran[5]. Restful Web Service pada dasarnya menggunakan metode operasional yang sederhana yang dimiliki oleh protokol HTTP untuk berkomunikasi. Pada umumnya API (Application Programming Interface) Restful Web Service menggunakan metoda-metoda standar HTTP itu sebagai parameter-parameter untuk URL (Uniform Resource Locator) yang dikirimkan klien ke server Restful Web Service[6].

## 2. Metode Penelitian

### 2.1. Arsitektur Sistem

Gambar 2.1.

Arsitektur **Sistem secara umum** Pada gambar 2.1 dapat dilihat bahwa komponen sistem yang dibangun terdiri dari sensor node, sink node dan server. Fungsi pada masing-masing device tetaplah sama. Setiap Sink Node dapat melakukan request ke sensor node. Dan sensor node hanya melakukan respon kepada Sink **Node yang melakukan** request. Komponen server pada gambar 2.1 tidak dalam penelitian ini.

Namun pada kasus nyata server ini berfungsi sebagai pengelola data sensor yang didapat dari sink node. Gambar 2.2. Arsitektur Sistem Sensor Node berfungsi sebagai node yang digunakan untuk melakukan pemindaian atau sense. Sink atau gateway berfungsi sebagai **keluar masuknya data yang akan dikirim** ke server.

Access Point disini berfungsi sebagai perantara komunikasi untuk menghubungkan antara sensor node dan gateway.

## 2.2. Perancangan Hardware

Gambar 2.3. Sensor node Arduino menjadi sensor node jika terdapat beberapa sensor. Oleh sebab itu ketiga rangkaian antara sensor ultrasonic, sensor cahaya, dan RTC dikombinasikan menjadi satu node.

Rangkaian inilah yang nantinya akan berfungsi sebagai pemindai atau sense untuk mendapatkan data. Gambar 2.4. Sink node (gateway) Raspberry Pi berperan sebagai gateway, berfungsi sebagai pintu gerbang untuk **keluar masuknya data yang di** dapat dari sensor node dengan konsep Restful Web Service menggunakan teknik Pull Message. Selain itu dibutuhkan 3 buah Raspberry Pi untuk melakukan pengujian pada **tugas akhir ini.**

Untuk mendukung **kinerja Raspberry Pi sebagai** gateway.

## 2.3. Perancangan Software

Proses data Arduino dibagi menjadi 2 proses. Yaitu proses sensing dan proses penanganan request. Proses pada Arduino dapat dijelaskan pada gambar 2.5 dan 2.6 di bawah ini. Gambar 2.5. Proses sensing Gambar 2.6.

Proses penanganan request Fungsi dari Raspberry Pi seperti yang sudah dijelaskan pada arsitektur sistem adalah sebagai sink node yang mempunyai peran untuk melakukan request data pada sensor node. Penjelasan alur kerja sistem akan dijelaskan pada gambar 2.7 flowchart berikut ini. Gambar 2.7. Alur sistem Raspberry Pi 3. Hasil **Penelitian dan Pembahasan** 3.1. Pengaruh Interval Waktu Sensing Terhadap Memory Heap Tujuan **pengujian ini untuk mengetahui apakah interval waktu sensing pada sensor node (Arduino) berpengaruh terhadap memory heap.** Skenario pada pengujian **ini akan dilakukan dengan** beberapa percobaan pada interval waktu sensing.

Sensing akan dijalankan pada setiap interval 1 detik, 3 detik, 5 detik, 7 detik, 9 detik, 11 detik, 13 detik, 15 detik, 17 detik, dan 19 detik. Tabel 3.1. Hasil pengujian interval waktu No. Interval waktu sensing (detik) Free memory heap (bytes) 1. 1 6148 2. 3 6148 3. 5 6148 4. 7 6148 5. 9 6148 6. 11 6148 7. 13 6148 8. 15 6148 9. 17 6148 10. 19 6148 Gambar 3.1.

Grafik hasil pengujian interval waktu Hasil analisis **yang didapat dari pengujian interval waktu sensing terhadap memory heap, bahwa interval waktu sensing tidak berpengaruh terhadap memory heap.** Dikarenakan ketika saat proses sensing. 3.2. Pengaruh Ukuran **Data Yang Dikirim** Terhadap Memory Heap Tujuan **pengujian ini untuk mengetahui apakah ukuran data yang dikirim akan berpengaruh terhadap memory heap.**

Skenario pengujian ini dengan dilakukan beberapa percobaan pulling data dengan ukuran data yg berbeda dimulai dari 10 KB, 20 KB, 30 KB, 40 KB, 50 KB, 60 KB, 70 KB, 80 KB, 90 KB, dan 100 KB. Tabel 3.2. Hasil pengujian ukuran data No. Ukuran Data (Kilobyte) Free Memory heap (bytes) 1. 10 6148 2. 20 6148 3. 30 6148 4. 40 6148 5. 50 6148 6. 60 6148 7. 70 6148 8. 80 6148 9. 90 6148 10. 100 6148 Gambar 3.2.

Grafik hasil pengujian ukuran data Hasil analisis **yang didapat dari pengujian ukuran data terhadap memory heap, bahwa besarnya data yang dikirim tidak berpengaruh terhadap memory heap.** Dikarenakan ketika sink node **melakukan request dan** sensor node melakukan response, fungsi untuk menampilkan memory heap tidak **berada di antara** proses **request dan response**, sehingga data yang ditampilkan oleh fungsi memory heap adalah data setelah terjadinya proses request dan response selesai.

Oleh sebab itu memory heap seolah tidak mengalami perubahan karena penggunaan memory sudah di release . 3.3. Pengaruh Jumlah Sink Yang

Mengrequest Terhadap Memory Heap Tujuan pengujian ini untuk mengetahui apakah jumlah sink node (Raspberry Pi) yang me-request berpengaruh terhadap memory heap.

Skenario pada pengujian ini sink node akan melakukan request ke sensor node dimulai dengan satu sink node, dua sink node, dan tiga sink node bersamaan akan melakukan request data. Tabel 3.3. Hasil pengujian sink node No. Jumlah Sink Node Free Memory heap (bytes) 1. 1 6148 2. 2 6148 3. 3 6148 Gambar 3.3 Grafik pengujian sink node Hasil analisis yang didapat dari pengujian jumlah sink node terhadap memory heap, bahwa jumlah atau banyaknya sink node yang merequest tidak berpengaruh terhadap memory heap.

Sama seperti pengujian sebelumnya dikarenakan ketika sink node melakukan request dan sensor node melakukan response, fungsi untuk menampilkan memory heap tidak berada di antara proses request dan response, sehingga data yang ditampilkan oleh fungsi memory heap adalah data setelah terjadinya proses request dan response selesai. Oleh sebab itu memory heap seolah tidak mengalami perubahan penggunaan memory sudah di release. 3.4. Pengaruh besar data terhadap request response time.

Tujuan Pengujian ini untuk mengetahui apakah besarnya data akan berpengaruh terhadap request response time. Skenario pada pengujian ini sink node akan melakukan pulling data sensor pada ukuran data 10 KB, 20 KB, 30 KB, 40 KB, 50 KB, 60 KB, 70 KB, 80 KB, 90 KB, dan 100 KB. Tabel 3.4. Tabel hasil pengujian request response time No. Ukuran Data (Kilobyte) Waktu (detik) 1. 10 3 2. 20 6 3. 30 10 4. 40 13 5. 50 17 6. 60 20 7.

70 23 8. 80 26 9. 90 29 10. 100 33 Gambar 3.4. Grafik hasil pengujian request response time Hasil analisis yang didapat dari pengujian pengaruh besar data terhadap request response time, bahwa terdapat beberapa jeda waktu antar proses, artinya semakin banyak data yang tertumpuk di SD Card maka semakin lama pula waktu yang dibutuhkan untuk melakukan request dan response.

Diketahui juga setiap ukuran data kelipatan 10 KB sink node melakukan proses pulling data sensor membutuhkan waktu kurang lebih 3 sampai 4 detik. Ketika proses request dan respons berjalan proses yang lain mengalami interrupt, proses terjadinya interrupt dapat dilihat pada gambar 3.5 berikut. Gambar 3.5. Serial Monitor Interrupt 4. Kesimpulan Berdasarkan hasil implementasi dan pengujian yang telah dilakukan dari tugas akhir yang berjudul "Implementasi Pull Message Dengan Menggunakan Restful Web Service Pada Komunikasi Wireless Sensor"

dapat disimpulkan sebagai berikut: 1. Dapat diimplementasikan sistem pull message dengan menggunakan **restful web service** pada komunikasi wireless sensor dapat dilakukan.

Sensor Node (Arduino) **dapat digunakan sebagai web server dan** bisa menangani banyak client. Sink Node (Raspberry Pi) dapat melakukan proses pull message dengan menggunakan **Restful Web Service** pada komunikasi Wireless Sensor. 2. Free memory heap digunakan untuk mengukur kinerja dari sensor node yang memanfaatkan metode pull message dengan protokol restful.

Hasil pengujian di dapat bahwa memory heap tidak terpengaruh dengan proses sensing, ukuran data, dan jumlah sink node. Namun dalam pengujian **request dan response time**, ukuran **data yang akan dikirim sangat berpengaruh terhadap waktu request dan response** dari sink node ke sensor node. Dalam implementasi tugas akhir ini masih jauh dari sempurna dan **masih terdapat kekurangan. Oleh karena itu** perlu dilakukan pengembangan dan penyempurnaan.

Adapun saran untuk pengembangan atau penyempurnaan sebagai berikut:

1. Untuk pengembangannya perlu untuk menambahkan metode atau solusi yang dapat mengefisiensikan pulling data sehingga waktu **request dan response tidak berpengaruh terhadap** besarnya ukuran data.

#### INTERNET SOURCES:

---

-

<1% - Empty

1% - <http://prima.lecturer.pens.ac.id/riset/a>

1% - <http://www.robotic-id.org/2014/11/mengen>

<1% - <http://dokumen.tips/documents/catatan2-e>

1% - <http://www.mikron123.com/index.php/vmchk>

2% - <http://www.academia.edu/11005188/PERBAND>

<1% - <http://ejurnal.its.ac.id/index.php/tekni>

<1% - <http://wisuda.unud.ac.id/pdf/1008605028->

<1% - <https://www.arduino.cc/>

<1% - <http://iosrjournals.org/iosr-jece/papers>

<1% - <https://www.scribd.com/document/14562581>

<1% - <http://thenewstack.io/tutorial-prototipi>

<1% - <https://es.scribd.com/doc/173103678/OPC->

<1% - <http://www.sciencedirect.com/science/art>  
1% - <http://ejournal-s1.undip.ac.id/index.php>  
<1% - <http://dokumen.tips/documents/sistem-sen>  
1% - <http://dokumen.tips/documents/makalah-pe>  
<1% - <https://iwankwahyum.wordpress.com/2014/0>  
<1% - <https://id.scribd.com/document/328457695>  
1% - <https://id.scribd.com/doc/193846102/Maka>  
1% - <http://hardi-santosa.blog.ugm.ac.id/2012>  
3% - <http://dokumen.tips/documents/cara-membu>  
3% - <http://dokumen.tips/documents/cara-membu>  
3% - <http://dokumen.tips/documents/cara-membu>  
1% - <http://elib.unikom.ac.id/files/disk1/632>  
1% - <http://www.mikron123.com/index.php/vmchk>  
<1% - <http://repository.maranatha.edu/view/yea>  
2% - <http://www.academia.edu/11005188/PERBAND>  
1% - <https://repository.ugm.ac.id/33039/1/jii>  
1% - <https://repository.ugm.ac.id/33039/1/jii>  
<1% - [http://www.academia.edu/8921282/Sistem\\_P](http://www.academia.edu/8921282/Sistem_P)  
<1% - <http://jurnal.stikom.edu/index.php/jcone>  
<1% - <https://tugasdenny.wordpress.com/page/15>  
<1% - <http://www.academia.edu/6426036/SeminarN>  
<1% - <http://www.slideshare.net/fitriaafebrii/>  
1% - [http://winarnotugas.blogspot.co.id/2010\\_](http://winarnotugas.blogspot.co.id/2010_)  
<1% - <http://cerpen-makalah.blogspot.com/>  
<1% - <https://jurnal.ugm.ac.id/ijeis/article/d>  
<1% - <https://www.scribd.com/doc/286651488/Buk>  
1% - <https://fr.scribd.com/doc/296117222/dasa>  
<1% - <https://pt.scribd.com/document/322796763>  
<1% - <https://es.scribd.com/doc/314762339/Joki>  
1% - <https://pt.scribd.com/document/327556600>  
<1% - <http://dokumen.tips/documents/tki-bidang>  
1% - <http://digilib.uinsby.ac.id/9219/6/bab4>  
<1% - <https://www.scribd.com/document/21093910>  
<1% - <https://www.scribd.com/document/43572841>  
<1% - <https://id.scribd.com/doc/24903072/PENGE>  
<1% - <https://pt.scribd.com/doc/117593257/jurn>  
<1% - <https://www.scribd.com/doc/239720803/Tek>  
<1% - <https://ra3miza.wordpress.com/2010/12/10>  
<1% - <https://www.scribd.com/doc/176579092/Ref>  
1% - <http://lib.unnes.ac.id/6266/1/4549A.pdf>

<1% - <https://id.scribd.com/document/241547423>  
<1% - <http://documents.tips/documents/sistemop>  
<1% - <https://www.scribd.com/doc/238207586/Pen>  
<1% - <https://id.scribd.com/document/176454444>  
<1% - [https://www.researchgate.net/profile/M\\_R](https://www.researchgate.net/profile/M_R)  
<1% - [http://indoaplikasi.com/tanya\\_jawab.php](http://indoaplikasi.com/tanya_jawab.php)  
<1% - <https://pt.scribd.com/doc/293395008/Modu>  
<1% - <https://www.scribd.com/document/32179995>  
1% - <http://dokumen.tips/documents/rancang-ba>  
<1% - <http://www.academia.edu/23310766/Arduino>  
<1% - <https://pt.scribd.com/doc/84610052/Siste>  
1% - <https://id.scribd.com/doc/130873742/PERA>  
1% - <http://dir.unikom.ac.id/s1-final-project>  
1% - <http://sinta.ukdw.ac.id/sinta/resources/>