

RANCANG BANGUN SISTEM KONTROL DAN MONITORING BERBASIS WEBSOCKET PADA PERANGKAT ARDUINO

Adhitya Bhawiyuga¹, Rakhmadhany Primananda², Widhi Yahya³, Yosia Rimbo Deantama⁴
Fakultas Ilmu Komputer Universitas Brawijaya^{1,2,3,4}
{bhawiyuga¹, rakhmadhany², widhi.yahya³, yosiarimbo⁴}@ub.ac.id

ABSTRACT

Internet of Things (IOT) is a system that connects the “things” through a computer network to control and monitor an environment. The IoT widely implemented in various fields which has a different environment characteristic for each. This stimulates the emergence of new protocols in computer networks. In this study, we implemented and analyzed a controlling and monitoring system based on microcontroller Arduino Uno with Websockets as communication protocol. The Websockets provide statefull and full-duplex communication on the HTTP protocol which is suitable to be applied for the control and monitoring systems. The Redis is used as a data storage on the data server because it has powerful processing capability. The implemented system is used to control the LED and to monitor the temperature and the humidity. The test results show that, performing process controlling and monitoring in a once, take less than a second with 100% success rate. The system performance decreases when serving 350 requests per second or serving 300 clients simultaneously. The increasing number of requests and the number of connected clients affect the increasing of Round Trip Time and the requests that are able to be serviced by the system.

Keywords: *Websocket, Arduino, Redis, Internet of Things*

ABSTRAK

Internet of Things (IoT) adalah sebuah sistem yang menghubungkan objek-objek mati melalui sebuah jaringan komputer untuk mengontrol dan memonitoring sebuah lingkungan. IoT diimplementasikan diberbagai bidang yang tentunya setiap bidang memiliki karakteristik lingkungan yang berbeda. Hal ini merangsang munculnya protokol-protokol baru dalam jaringan komputer. Pada penelitian ini, diimplementasikan dan dianalisis sebuah sistem kontroling dan monitoring berbasis mikrokontroler Arduino Uno dengan protokol komunikasi menggunakan Websocket. Websocket dipilih karena mampu menyediakan komunikasi *statefull* dan *full-duplex* pada protokol HTTP yang cocok diterapkan untuk sistem kontrol dan monitoring. Pada sisi *server* data digunakan Redis sebagai media penyimpanan data karena memiliki kemampuan pemrosesan yang handal. Pada implementasinya sistem digunakan untuk mengontrol LED serta untuk memonitoring suhu dan kelembapan. Hasil pengujian kinerja menunjukkan bahwa untuk melakukan satu kali proses kontroling dan monitoring perangkat, waktu yang dibutuhkan tidak lebih dari 1 detik dengan persentase keberhasilan 100%. Terdapat penurunan performa pada saat dilakukan request sebanyak 350 request per detiknya atau terdapat 300 client yang terkoneksi secara bersamaan. Dengan semakin banyaknya request dan jumlah client yang terkoneksi maka akan berpengaruh terhadap Round Time Trip dan request yang mampu dilayani oleh sistem.

Kata Kunci: *Websocket, Arduino, Redis, Internet of Things*

PENDAHULUAN

Internet of Things (IoT) adalah sebuah konsep dimana suatu objek mati memiliki kemampuan untuk menerima dan mengirimkan sebuah data melalui koneksi jaringan. Salah satu contoh IoT adalah interaksi manusia dengan sebuah perangkat, dimana perangkat tersebut digunakan untuk mengontrol sebuah lingkungan atau memonitoring kondisi sebuah lingkungan. Selain membutuhkan perangkat seperti sensor dan aktuator, untuk mengimplementasikan konsep IoT dibutuhkan peran serta teknologi lain, seperti teknologi komputer dan teknologi jaringan.

Perkembangan teknologi yang sangat pesat berdampak pada bertambah banyaknya protokol jaringan khususnya protokol jaringan untuk komunikasi client dengan suatu perangkat. Munculnya berbagai jenis protokol tersebut menimbulkan satu masalah yaitu membuat pengembangan sistem menjadi rumit. Untuk mengatasi masalah tersebut dibutuhkan protokol jaringan yang sederhana dan sering digunakan kebanyakan orang.

Hyper Text Transfer Protocol (HTTP) adalah salah satu protokol yang paling banyak digunakan dalam internet, dengan HTTP yang bekerja di layer TCP membuat HTTP menjadi protokol yang mampu mengirim berita, video, dan melayani banyak aplikasi web (Megyesi, Kramer & Molnar, 1997). Dalam sebuah situs, yaitu dzone.com membuat sebuah survei tentang penggunaan IoT. Di dalam artikel tersebut, sebanyak 265 responden dari 528 responden menjawab HTTP merupakan protokol yang paling banyak digunakan dalam IoT. Dari paparan di atas, maka dalam penelitian ini dibutuhkan sebuah protokol yang mempunyai karakteristik HTTP yang mampu mengimplementasikan kegiatan controlling dan monitoring perangkat berbasis Arduino.

Websocket adalah protokol yang mampu menyediakan komunikasi full-duplex pada protokol HTTP dengan menggunakan satu TCP socket saja. Dengan websocket memungkinkan untuk dibuatnya sebuah aplikasi berbasis HTTP yang real time. (Srinivasan, Scharnagl & Schilling, 2013). Dari penjelasan websocket di atas, menunjukkan bahwa websocket merupakan sebuah teknologi yang berjalan pada protokol HTTP. Dikarenakan adanya kesinambungan pada pengertian Websocket dengan permasalahan yang dijelaskan sebelumnya, maka dalam penelitian ini penulis menggunakan protokol Websocket untuk memonitoring dan controlling perangkat berbasis Websocket.

Selain membutuhkan teknologi jaringan, agar perangkat seperti aktuator atau sensor dapat bekerja maka dibutuhkan sebuah teknologi komputer berupa mikrokontroler yang berfungsi sebagai pengolah data baik itu data yang akan dikirimkan ke perangkat atau data yang diterima dari perangkat.

Dalam penelitian *Experiences on Using Arduino for Laboratory Experiments of Automatic Control and Robotics* (Candelas, dkk, 2015). Dilakukan analisis tentang mikrokontroler Arduino dimana disimpulkan Arduino adalah sebuah mikrokontroler yang mudah dipelajari, memiliki harga yang murah serta memiliki free software. Maka dari itu, selain menggunakan protokol Websocket penulis juga menggunakan Arduino sebagai mikrokontroler.

Mengacu pada permasalahan yang telah disampaikan, judul yang diambil dalam skripsi ini adalah "Implementasi Websocket untuk Monitoring dan Controlling Perangkat Berbasis Arduino". Dimana untuk pengimplementasian monitoring menggunakan perangkat 4 lampu LED yang terpasang di Arduino.

Dan untuk pengimplementasian kontroling menggunakan sensor DHT 11 yang terpasang di Arduino. Diharapkan tema skripsi yang diangkat akan menjadi salah satu solusi supaya interaksi manusia dengan dengan sebuah perangkat berbasis Arduino dapat terus dikembangkan dan dapat dirasakan manfaatnya bagi para penggunanya

DASAR TEORI

Websocket

Protokol Websocket merupakan protokol yang mulai dikembangkan pada HTML5. Protokol ini adalah hasil peningkatan dari HTTP dan berdiri pada protokol TCP. TCP adalah protokol inti yang mengatur jalan pengiriman informasi dalam internet

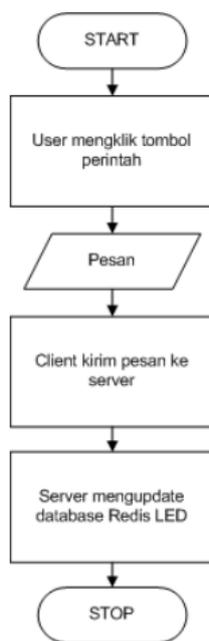
Protokol Websocket mampu memberikan layanan komunikasi 2 arah secara bersamaan. Selain itu Websocket memiliki header yang kecil. Namun Websocket tidak bisa dipertimbangkan untuk menjadi alternatif pengganti TCP, karena Websocket dan TCP bekerja dalam 2 lapisan jaringan yang berbeda dalam model protokol internet. Untuk dapat menggunakan protokol Websocket, diperlukan proses handshake agar client dapat terhubung dengan server.

Untuk membuka koneksi Websocket, client harus membuka koneksi jaringan protokol TCP ke server terlebih dahulu. HTTP merupakan protokol standar saat koneksi TCP terbuka. Setelah itu, client meminta server untuk mengubah protokol HTTP ke protokol Websocket. Setelah server merespon permintaan client, maka koneksi Websocket terbuka. Koneksi Websocket ini akan terus terbuka hingga ada perintah untuk memutuskan koneksi. Pengiriman informasi dari server ke client dalam protokol ini tidak memerlukan lagi pengenalan client, sehingga ini tidak memerlukan lagi pengenalan client, sehingga jumlah ukuran paket yang dikirimkan lebih sedikit. Hal ini sangat efisien untuk mengirimkan informasi secara cepat.

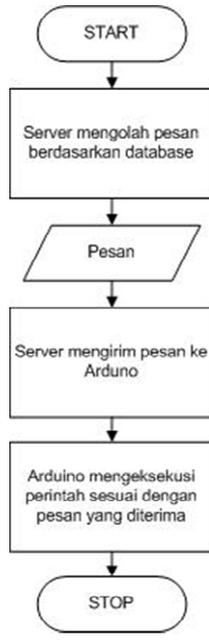
PERANCANGAN SISTEM

Bab ini menguraikan hasil penelitian yang telah diperoleh dimana peneliti harus mampu menunjukkan tercapai tidaknya tujuan penelitian yang telah ditetapkan. Hasil penelitian sebaiknya ditampilkan dalam bentuk grafik ataupun tabel. Tujuan pembahasan adalah menjawab masalah penelitian, menafsirkan temuan-temuan, mengintegrasikan temuan dari penelitian ke dalam kumpulan pengetahuan yang telah ada dan menyusun teori baru atau memodifikasi teori yang sudah ada. Pembahasan penelitian harus dikembalikan kepada teori yang menjadi sandaran penelitian ilmiah yang telah dilakukan.

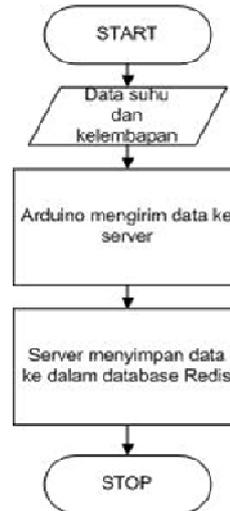
Perancangan Kontroling Lampu LED



Gambar 3.1
 Diagram Alur
 Pengiriman Pesan ke
 Server



Gambar 3.2
 Diagram Alur
 Pengiriman Pesan
 ke Arduino



Gambar 3.3
 Diagram Alur
 Pengiriman Data
 ke Server



Gambar 3.4 Diagram
 Alur Pengiriman
 Pesan ke Client

Gambar 3.1 menunjukkan bagaimana proses pengiriman pesan dari client ke server. Pertama client harus terhubung dengan server, setelah berhasil melakukan koneksi, pengguna dapat mengklik tombol untuk mematikan atau menghidupkan lampu LED. Pesan yang dikirimkan ke server sesuai dengan button yang diklik oleh pengguna. Pengiriman pesan dilakukan menggunakan protokol Websocket, dimana pada bagian server menggunakan library twisted txws. Selanjutnya berdasarkan pesan tersebut, server akan mengupdate database Redis.

Gambar 3.2 menunjukkan bagaimana proses pengiriman pesan dari server ke Arduino. Server akan membuat pesan berdasarkan semua status LED yang terdapat dalam database Redis. Pesan yang dikirimkan ke Arduino berbentuk integer. Pengiriman pesan integer ini dilakukan karena Arduino hanya bisa membaca pesan dalam bentuk bytes. Setelah pesan berhasil dibentuk, server akan mengirimkan pesan ke Arduino menggunakan komunikasi Serial. Arduino akan langsung mengeksekusi perintah saat pesan berhasil diterima.

Gambar 3.3 menunjukkan bagaimana proses pengiriman data suhu dan kelembapan yang didapatkan sensor DHT 11 ke server. Pertama Arduino akan mentrigger sensor DHT 11 untuk mendapatkan data suhu dan kelembapan. Setelah itu data tersebut langsung dikirimkan ke server menggunakan komunikasi serial. Data yang telah didapatkan oleh server akan langsung disimpan ke dalam database Redis bertipe data lists, dimana data suhu disimpan pada lists 'temp' dan data kelembapan disimpan pada lists 'humid'.

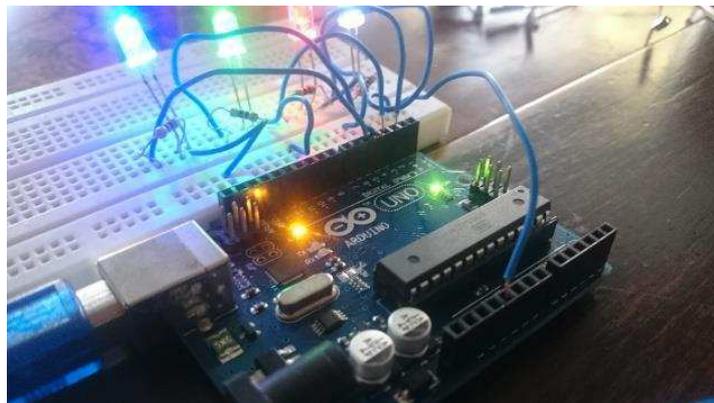
Gambar 3.4 menunjukkan proses pengiriman data dari server ke client. Server akan membuat pesan dengan mengambil data dari database Redis. Terdapat 7 pesan yang akan dikirimkan ke client diantaranya 2 bertipe data JSON dan 5 bertipe data string. Data JSON nantinya akan digunakan oleh aplikasi web untuk diubah menjadi grafik garis dan grafik speedometer. Sedangkan data string akan langsung ditampilkan dalam bentuk tulisan pada aplikasi web. Setelah pesan yang diperlukan sudah terbentuk, pesan akan dikirimkan ke client menggunakan protokol Websocket. Pesan yang diterima oleh client akan langsung ditampilkan dalam bentuk tulisan dan grafik.

IMPLEMENTASI

Sesuai dengan perancangan yang sudah dijelaskan sebelumnya, implementasi dalam penelitian ini dilakukan sebanyak 2 kali yaitu implementasi untuk mengontrol LED dan implementasi untuk monitoring suhu dan kelembapan.

Implementasi Kontroling Lampu LED

Implementasi Perangkat Keras



Gambar 4.1 Implementasi Perangkat Keras LED

Implementasi Aplikasi Web Client

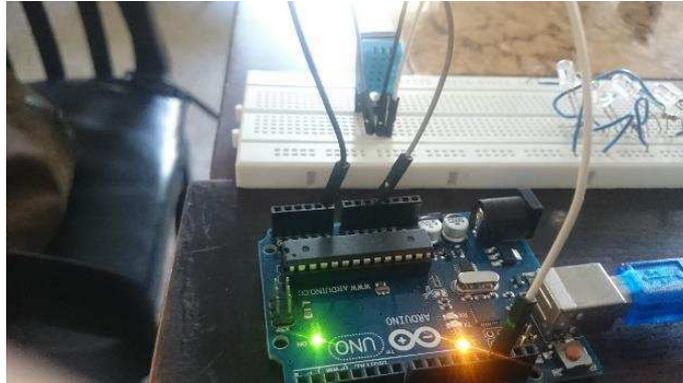
Antarmuka sistem untuk mengontrol lampu LED terdiri dari beberapa fungsi. Tampilan antarmuka sistem untuk mengontrol LED ditunjukkan dalam Gambar 4.2 berikut:



Gambar 4.2 Antarmuka Sistem Kontroling Lampu LED

Implementasi Monitoring Suhu dan Kelembapan

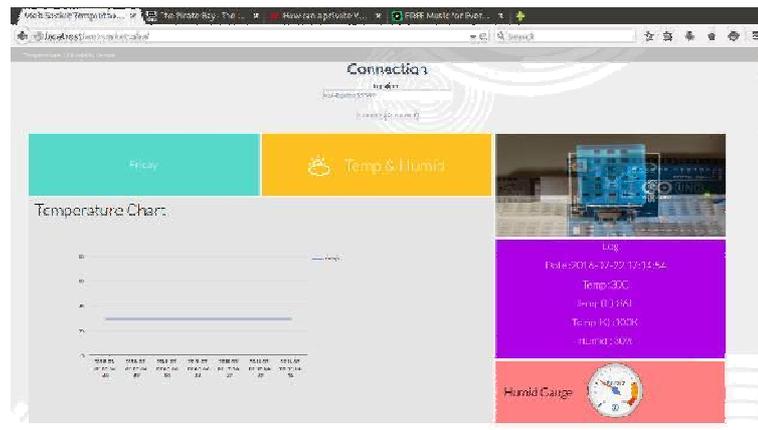
Implementasi Perangkat Keras



Gambar 4.3 Konfigurasi DHT 11

Implementasi Aplikasi Web Client

Antarmuka sistem untuk monitoring suhu dan kelembapan terdiri dari beberapa fungsi. Tampilan antarmuka sistem untuk mengontrol LED ditunjukkan dalam Gambar 5.4 berikut:

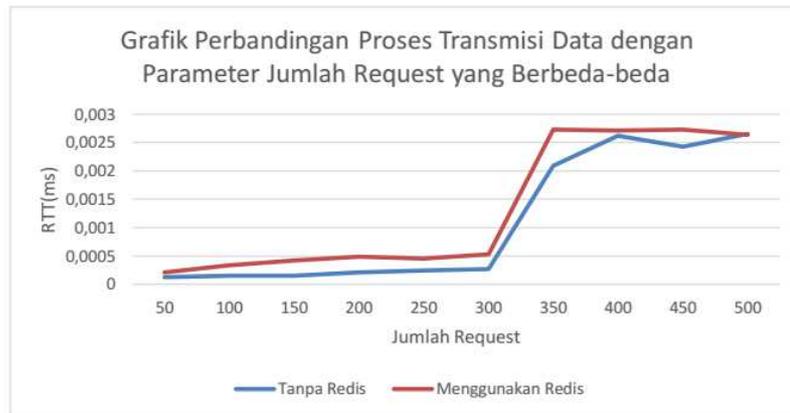


Gambar 4.4 Antarmuka Monitoring Suhu dan Kelembapan

PENGUJIAN DAN ANALISIS

Pengujian Round Time Trip Berdasarkan Jumlah Request

Pengujian dilakukan untuk mengetahui seberapa cepat sebuah data dapat mengalir dari titik awal ke titik akhir sampai menuju titik awal lagi. Pengujian dilakukan pada pengimplementasian monitoring suhu dan kelembapan. RTT didapatkan dengan jumlah request yang bervariasi yaitu 50, 100, 150, 200, 250, 300, 350, 400, 450, 500. Selain itu pengujian dilakukan pada sistem yang menggunakan Redis dan sistem yang tidak menggunakan Redis.

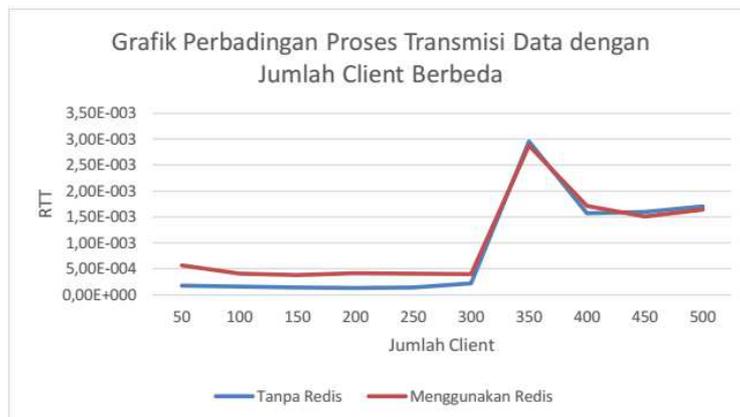


Gambar 5.1 Perbandingan RTT Berdasarkan Jumlah Request

Pada gambar 5.1 menunjukkan bahwa sistem yang tidak menggunakan Redis memiliki waktu RTT yang lebih baik daripada sistem yang memakai Redis. Kedua sistem sama-sama mengalami penurunan performa pada saat menangani 350 request. Terjadi peningkatan RTT pada saat dikirimkan 350 request hingga seterusnya, hal ini dikarenakan dalam TCP terdapat metode flow control yang digunakan untuk mengosongkan buffer.

Pengujian Round Time Trip Berdasarkan Jumlah Client

Skenario pengujian hampir sama dengan pengujian RTT sebelumnya, hanya saja parameter jumlah request diganti dengan jumlah client.



Gambar 5.2 Perbandingan RTT Berdasarkan Jumlah Client

Dari Gambar 5.2 dapat ditarik kesimpulan waktu proses transmisi data tidak menggunakan Redis lebih unggul daripada waktu proses transmisi data menggunakan Redis. Perbedaan RTT yang dihasilkan dari 2 pengujian tidak jauh berbeda satu sama lain. Hanya terjadi perbedaan signifikan pada saat pengujian mencapai 350 *client*.

Terjadi peningkatan RTT pada saat terdapat 350 client, hal ini dikarenakan ketika server mengirimkan data ke salah satu client, client tersebut sudah terputus koneksinya dengan server. Sehingga server menunggu sampai pesan tersebut melewati batas timed out.

Pengujian Kehandalan Sistem Melayani Request Berdasarkan Jumlah Request



Gambar 5.3 Perbandingan Jumlah Request yang Terlayani Berdasarkan Jumlah Request

Pengujian dilakukan untuk mengetahui seberapa handal sistem dalam melayani request cepat. Pengujian dilakukan pada pengimplementasian monitoring suhu dan kelembapan. Jumlah request yang terlayani didapatkan dengan jumlah request yang bervariasi yaitu 50, 100, 150, 200, 250, 300, 350, 400, 450, 500. Selain itu pengujian dilakukan pada sistem yang menggunakan Redis dan sistem yang tidak menggunakan Redis.

Gambar 5.3 menunjukkan hasil perbandingan analisis jumlah request yang terlayani antara sistem yang menggunakan redis dengan sistem yang tidak menggunakan redis dengan jumlah request yang berbeda. Dari grafik dapat dijelaskan bahwa kedua sistem memiliki kehandalan melayani request yang hampir sama. Kedua sistem memiliki kehandalan yang hampir sama, hal ini dikarenakan database Redis tersimpan dalam memori. Sehingga untuk mengambil nilai yang terletak dalam database tidak perlu memerlukan waktu yang lama.

KESIMPULAN

Implementasi protokol Websocket kedalam kontroling dan monitoring perangkat berbasis Arduino dapat dilakukan dengan menggunakan bahasa Python. Selain itu framework twisted txws juga digunakan untuk memudahkan pengimplementasian. Untuk komunikasi antara server dengan Arduino menggunakan komunikasi Serial. Dari hasil pengujian yang dilakukan Round Time Trip yang didapatkan kurang dari 1 detik baik itu sistem yang menggunakan Redis atau yang tidak menggunakan Redis. Hal ini menandakan dengan mengimplementasikan Websocket, pengiriman data dari Arduino sampai ke client dapat dilakukan secara real time. Selain itu dalam faktor kehandalan sistem melayani request, sistem dianggap cukup handal dalam melayani request. Penurunan performa baru dirasakan ketika terdapat 300 sampai 350 client yang melakukan koneksi secara bersamaan.

DAFTAR PUSTAKA

- B. W. Evans, *Arduino Programming Notebook*, 1 penyunt., California: Creative Commons, 2007.
- W. Durfee, *Arduino Microcontroller Guide*, University of Minnesota, 2011.
- F. Candelas, G. Garcia, S. Puente, J. Pomares, C. Jara, J. Perez, D. Mira dan F. Torres, *Experiences of Using Arduino for Laboratory Experiments of Automatic Control and Robotics*, Alicante: IFAC, 2015, pp. 105-110.
- D-Robotics, *DHT11 Humidity & Temperature Sensor*, UK, 2010.
- P. Megyesi, Z. Kramer dan S. Molnar, *Comparison of Web Transfer Protocols*, Budapest: High Speed Network Laboratory.
- J. Heidemann, K. Obraczka dan J. Touch, *Modeling the Performance of HTTP Over Several Transport Protocols*, vol. 5, IEEE, 1997.
- M. Miftakhuddin, W. Suadi dan B. A. Pratomo, *Implementasi Key-Value Store dengan Struktur Data List dan Tree Menggunakan Python*, Institut Teknologi Sepuluh November.
- L. Srinivasan, J. Scharnagl dan S. Klaus, *Analysis of Websocket as the New Age Protocol for Remote Robot Tele-operation*, Seoul: 3rd IFAC, 2013.
- W. Zhang, P. Passow, E. Jovanov, R. Stoll dan K. Thurow, *A Secure and Scalable Telemonitoring System Using Ultra Low Energy Wireless Sensor Interface for Long Term Monitoring In Life Science Applications*, Rosctock: University of Rosctock.
- C. Salzmann, G. Sten, W. Halimi dan D. Gillet, *The Smart Device Specification for Remote Labs*, Lausame: EPFL.
- I. Pette dan A. Melkinov, *Websocket Protocol*, IETF, 2011.
- T. M. Labs, *Twisted Documentation*, 2016.
- V. Pimentel dan B. Nickerson, *Web Display of Real-Time Wind Sensor Data*, Fredericton: University of New Brunswick, 2011.
- R. Weber, *Internet of Things: Privacy Issues Revisited*, Zurich: University of Zurich, 2015.
- G. P. Nugroho, A. Mazharuddin dan H. Studiawan, *Sistem Pendeteksi Banjir Menggunakan Sensor Kecepatan Air dan Sensor Ketinggian Air pada Mikrokontroler Arduino*, vol. 2, *Jurnal Teknil Pomits*, 2013.
- E. Frecon, *Web Like Protocols for the Internet of Things*, ICE.