IMPLEMENTASI WEBSOCKET UNTUK KONTROLING DAN MONITORING PERANGKAT BERBASIS ARDUINO

Yosia Rimbo Deantama[1)], Adhitya Bhawiyuga, S.Kom, M.S[2)], Rakhmadhany Primananda, S.T, M.Kom[3)]
Program Studi Teknik Informatika
Program Teknologi Informasi dan Ilmu Komputer
Universitas Brawijaya, Malang 65145, Indonesia
email : yosiarimbo[at]gmail.com[1)], bhawiyuga[at]ub.ac.id[2)], rakhmadhany[at]ub.ac.id[3)]

Abstrak

Internet of Things (*IoT*) adalah sebuah konsep dimana suatu objek mati memiliki kemampuan untuk menerima dan mengirimkan sebuah data melalui koneksi jaringan. Dengan konsep *IoT*, manusia memungkinkan untuk mengontrol dan memonitoring sebuah lingkungan. Untuk mengimplementasikan konsep *IoT* maka dibutuhkan peran serta teknologi, tidak tekecuali teknologi jaringan. Tetapi semakin banyaknya protokol jaringan yang muncul, membuat komunikasi antar objek dalam *IoT* menjadi rumit. Maka dari itu dibutuhkan sebuah protokol jaringan yang mudah dikembangkan dan mudah dipahami kebanyakan orang. Pada penelitian ini, penulis menggunakan sebuah protokol jaringan yang mudah dikembangkan untuk mengimplementasikan *IoT* yaitu *Websocket*. Selain *Websocket* yang digunakan untuk jalur komunikasi, penulis juga menggunakan Arduino Uno sebagai mikrokontroler yang digunakan untuk mendapatkan dan memproses data dari sebuah lingkungan dan Redis sebagai media penyimpanan data. Untuk mempraktekkan implementasi Websocket untuk monitoring dan kontroling perangkat berbasis Arduino, maka dilakukan dua implementasi. Impelementasi yang pertama adalah implementasi *Websocket* untuk mengontrol empat lampu *LED* yang terpasang pada Arduino Uno. Implementasi kedua adalah implementasi *Websocket* untuk memonitoring data berupa suhu dan kelembapan dari sensor DHT 11 yang terpasang pada Arduino Uno. Kedua pengimplementasian sama-sama menggunakan metodologi client-server. 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 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*

Abstract

Internet of Things (IoT) is a concept where an object has the ability to receive and transmit data through a network connection. With the concept of IOT, humans can do controlling and monitoring an environment. To implement the concept of IOT it takes the role of technology, include network technology. But the increasing number of network protocols that appear, making communication between objects in the IoT becomes complicated. Therefore it needs a network protocol that is easy to develop and easy to understand for most people. In this study, the authors use a network protocol which is easy to developed to implement IoT, the protocol is Websockets. Besides Websockets which used for communication lines, the author also uses Arduino Uno as a micro-controller that used to acquire and process data from an environmental and Redis as a data storage. To practice Websockets implementation for monitoring and controlling devices based on Arduino, there has the two implementations. The first is Websockets implementation to control four LED lights mounted on the Arduino Uno. The second is Websockets implementation for monitoring temperature and humidity data from DHT 11 which mounted on the Arduino Uno.To perform one process controlling and monitoring devices, time needed is not more than 1 second with a percentage of 100% success. There is a decrease in performance at the time of the request as much as 350 per second or 300 clients are connected simultaneously. With the increasing number of requests and the number of clients connected it will affect the Round Trip Time and requests are able to be serviced by the system.
Keywords : *Websocket, Arduino, Redis, Internet of Things*

.

1. 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 atau 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 peneltian ini dibutuhkan sebuah protokol yang mempunyai karakteristik *HTTP* yang mampu mengimplementasikan kegiatan kontrolling 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 kontrolling perangakat 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 Kontroling 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 interkasi manusia dengan dengan sebuah perangkat berbasis Arduino dapat terus dikembangkan dan dapat dirasakan manfaatnya bagi para penggunanya

2. DASAR TEORI

2.2 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

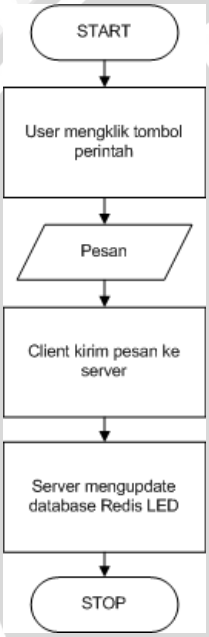
Protokol Websocket mampu memberikan layanan komunikasi 2 arah secara bersamaan. Selain itu Websocket memiliki header yang kecil. Namun Websocket tidak bisa dipertimbangkan untuk menjadi alternatif pengganti TCP, karena Websocket dan TCP bekerja dalam 2 lapisan jaringan yang berbeda dalam model protokol 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 Weebsocket 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. Hali ini sangat efisien untuk mengirimkan informasi secara cepat.

3. PERANCANGAN SISTEM

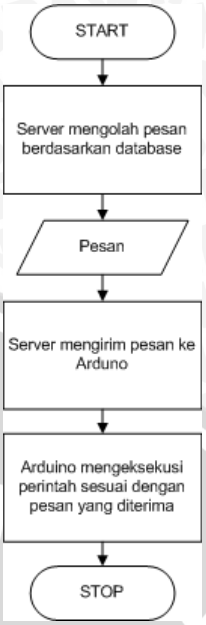
Pada subbab ini akan dibahas mengenai langkah-langkah pengimplementasian Websocket untuk Monitoring dan Kontroling Perangkat Berbasis Arduino. Terdapat dua macam pengimplementasian yaitu pengimplementasian kontroling LED, dan pengimplementasian monitoring suhu dan kelembapan menggunakan DHT 11 . Adapun perangkat keras dan perangkat lunak yang diperlukan oleh kedua sistem yaitu, sensor DHT 11, 4 lampu LED dan Arduino Uno sebagai kebutuhan perangkat keras. Program Arduino, Program server dalam Python, Aplikasi web client, dan database Redis sebagai kebutuhan perangkat lunak

3.1 Perancangan Kontroling Lampu LED



Gambar 3. 1 Diagram Alur Pengiriman Pesan ke Server

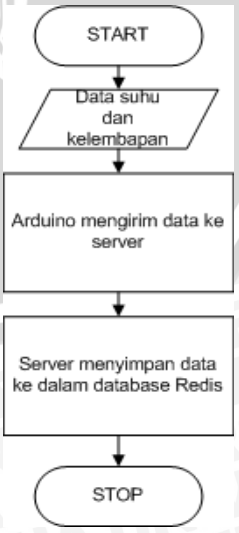
Gambar 3.1 menunjukkan bagaimana proses pengiriman pesan dari client ke server. Pertama client harus terhubung dengan server, setelah berhasil melakukan koneksi, pengguna dapat mengklik tombol untuk mematikan atau menghidupkan lampu LED. Pesan yang dikirimkan ke server sesuai dengan button yang diklik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



Gambar 3.2 Diagram Alur Pengiriman Pesan ke Arduino

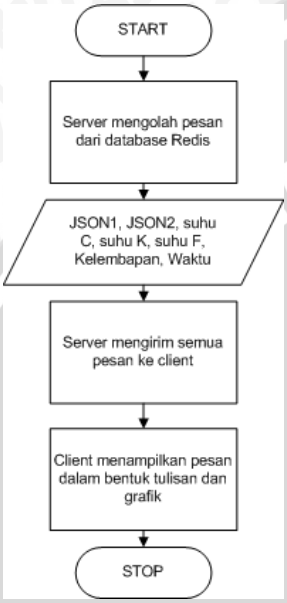
Gambar 3.2 menunjukkan bagaimana proses pengiriman pesan dari server ke Arduino. Server akan membuat pesan berdasarkan semua status LED yang terdapat dalam database Redis. Pesan yang dikirimkan ke Arduino berbentuk integer. Pengiriman pesan integer ini dilakukan karena Arduino hanya bisa membaca pesan 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.

3.2 Perancangan Monitoring Suhu dan Kelembapan



Gambar 3.3 Diagram Alur Pengiriman Data ke Server

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 didapatakan 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 Diagram Alur Pengiriman Pesan ke Client

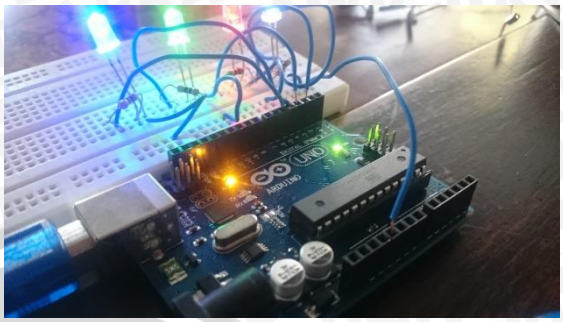
Gambar 3.4 menunjukkan proses pengiriman data dari server ke cleint. Server akan membuat pesan dengan mengambil data dari database Redis. Terdapat 7 pesan yang akan dikirimkan ke client diantaranya 2 bertipe data JSON dan 5 bertipe data string. Data JSON nantinya akan digunakan oleh aplikasi web untuk 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 langung ditampilkan dalam bentuk tulisan dan grafik

4. 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 dak kelembapan.

4.1 Impelementasi Kontroling Lampu LED

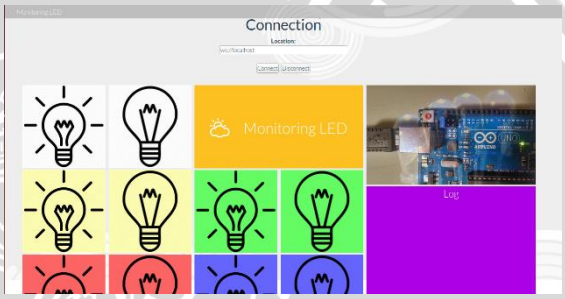
4.1.1 Implementasi Perangkat Keras



Gambar 4.1 Impelementasi Perangkat Keras LED

4.1.2 Impelementasi Aplikasi Web Client

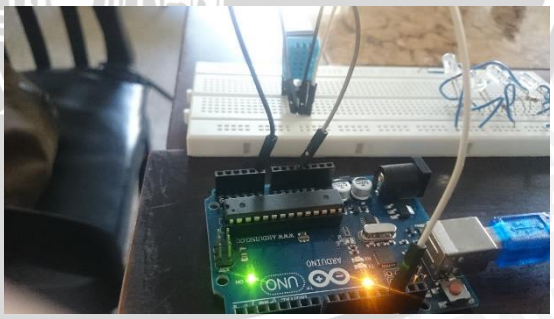
Antarmuka sistem untuk mengontrol lampu LED terdiri dari beberapa fungsi. Tampilan antarmuka sistem untuk mengontrol LED ditunjukkan dalam Gambar 5.2 berikut:



Gambar 4.2 Antarmuka Sistem Kontroling Lampu LED

4.2 Implementasi Monitoring Suhu dan Kelembapan

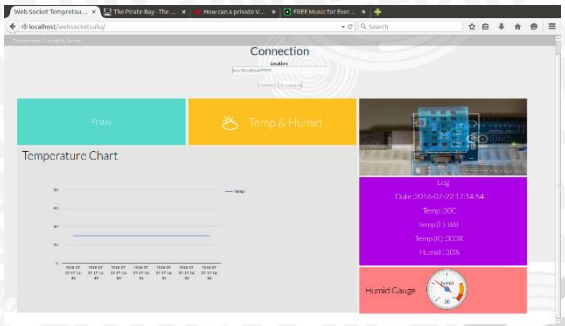
4.2.1 Implementasi Perangkat Keras



Gambar 4.3 Konfigurasi DHT 11

4.2.2 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:

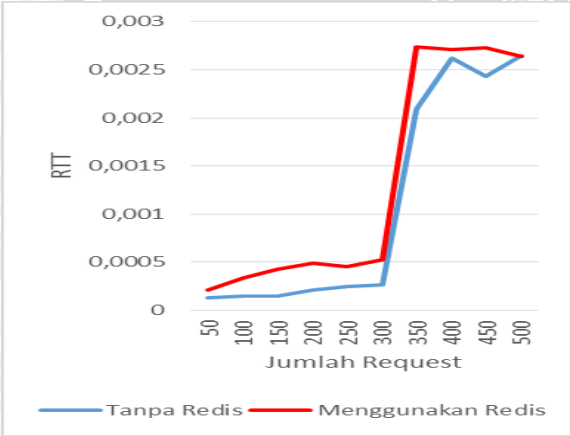


Gambar 4.4 Antarmuka Monitoring Suhu dan Kelembapan

5. Pengujian dan Analisis

5.1 Pengujain 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 dilakuakan 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.



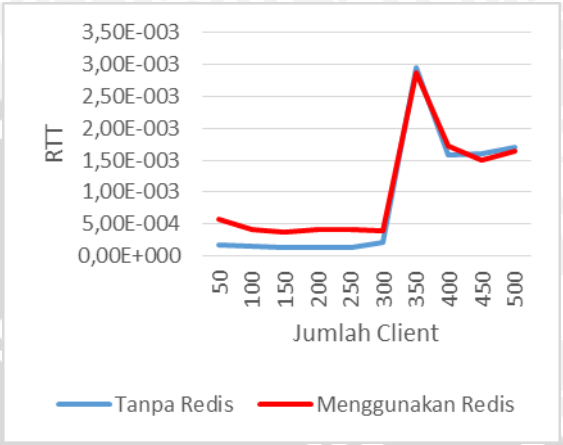
Gambar 5.1 Perbandingan RTT Berdasarkan Jumlah Request

Pada gambar 5.1 menunjukkan bahwa sistem yang tidak menggunakan Redis memiliki waktu RTT yang lebih baik daripada sistem yang memakai Redis. Kedau sistem sama-sama mengalami penurunan performa pada saat menangani 350 request.

Terjadi peningkatan RTT pada saat dikirmkan 350 request hingga seterusnya, hal ini dikarenakan dalam TCP terdapat metode flow control yang digunakan untuk mengosongkan buffer.

5.2 Pengujian Round Time Trip Berdasaran Jumlah Client

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



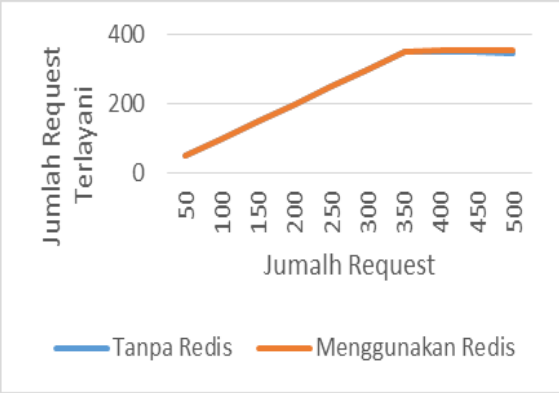
Gambar 5.2 Perbandingan RTT Berdasarkan Jumlah Client

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.

5.3 Pengujian Kehandalan Sistem Melayani Request 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.

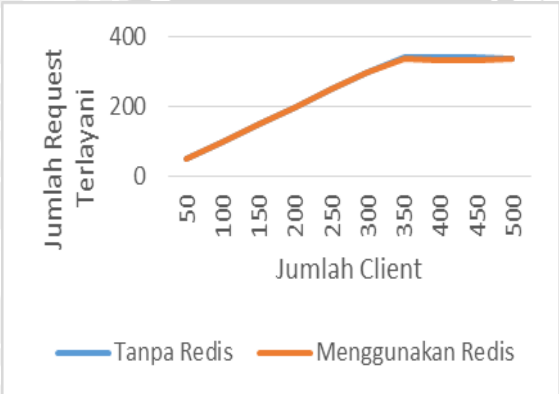


Gambar 5.3 Perbandingan Jumlah Request yang Terlayani Berdasarkan Jumlah Request

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 meiliki 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.

5.4 Pengujian Kehandalan Sistem Melayani Request Berdasarkan Jumlah Client

Skenario pengujian hampir sama dengan pengujian Kehandalan Sistem Melayani Request sebelumnya, hanya saja parameter jumlah request diganti dengan jumlah client.



Gambar 5.4 Perbandingan Jumlah Request Terlayani Berdasarkan Jumlah Client

Gambar 5.4 menunjukkan hasil perbandingan analisis jumlah request yang terlayani antara sistem yang menggunakan redis dengan sistem yang tidak menggunakan redis dengan jumlah client yang berbeda-beda. Dari grafik dapat dijelaskan bahwa kedua sistem meiliki 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.

6. PENUTUP

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

[1] B. W. Evans, Arduino Programming Notebook, 1 penyunt., California: Creative Commons, 2007.

[2] W. Durfee, Arduino Microcontroller Guide, University of Minnesota, 2011.

[3] 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.

[4] D-Robotics, DHT11 Humidity & Temperature Sensor, UK, 2010.

[5] P. Megyesi, Z. Kramer dan S. Molnar, Comparison of Web Transfer Protocols, Budapest: High Speed Network Laboratory.

[6] J. Heidemann, K. Obraczka dan J. Touch, Modeling the Performance of HTTP Over Several Transport Protocols, vol. 5, IEEE, 1997.

[7] M. Miftakhuddin, W. Suadi dan B. A. Pratomo, Implementasi Key-Value Store dengan Struktur Data List dan Tree Menggunakan Python, Institut Teknologi Sepuluh November.

[8] A. McCurdy dan R. v. Hattem, Redis-py Documentation, 2016.

[9] K. Seguin, Little Redis Book.

[10] K. Ma dan R. Sun, Introducing Websocket-Based Real Time Monitoring System for Remote Intelligent Buildings, vol. 2013, Jinan: University of Jinan, 2013.

[11] L. Srinivasan, J. Scharnagl dan S. Klaus, Analysis of Websocket as the New Age Protocol for Remote Robot Tele-operation, Seoul: 3rd IFAC, 2013.

[12] W. Zhang, P. Passow, E. Jovanov, R. Stoll dan K. Thurow, A Seccure and Scalable Telemonitoring System Using Ultra Low Energy Wireless Sensor Interface for Long Term Monitoring In Life Science Applications, Rosctock: University of Rosctock.

[13] C. Salzmann, G. Sten, W. Halimi dan D. Gillet, The Smart Device Specification for Remote Labs, Lausame: EPFL.

[14] I. Pette dan A. Melkinov, Websocket Protocol, IETF, 2011.

[15] T. M. Labs, Twisted Documentation, 2016.

[16] V. Pimentel dan B. Nickerson, Web Display of Real-Time Wind Sensor Data, Fredericton: University of New Brunswick, 2011.

[17] R. Weber, Internet of Things: Privacy Issues Revisited, Zurich: University of Zurich, 2015.

[18] 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.

[19] E. Frecon, Web Like Protocols for the Internet of Things, ICE.