

Implementasi *Constrained Application Protocol* pada *Internet of Things* dengan *Constrained RESTful Environments* berbasis *Constrained Device*

Sri Suryani¹, Ubaidah², Hendra Widodo³

^{1,2,3} Prodi Teknik Elektro, Fakultas Teknik
Universitas Muhammadiyah Lampung
Jl. H. Zainal Abidin Pagar Alam No.14 Bandar Lampung 35142
E-mail: srisuryani@mail.uml.ac.id,
ubaidah@uml.ac.id,
hendrawidodo@mail.uml.ac.id

Abstrak

This study discusses the implementation of Constrained Application Protocol (CoAP) with Constrained RESTful Environments (CoRE) in RFC 7252 which is used as a research parameter. Implementation of this Constrained Application Protocol uses Internet of Things (IoT) technology. The testing technique is carried out offline and the devices used are based on constrained devices. Network performance test parameters in this study are UDP throughput, UDP delay, UDP packet loss and UDP packet delivery ratio. Testing network performance with LED and Buzzer outputs produces the largest average UDP throughput of 4.5737 kbps while the smallest average throughput is 1.2293 kbps, the largest average UDP delay is 2 seconds and the smallest average is 0.6 seconds, then the average result of UDP packet loss is 0% while the average success of packet delivery ratio is 100%. From the results of this test, the Constrained Application Protocol (CoAP) protocol has low network performance results compared to the HyperText Transfer Protocol (HTTP) protocol to be implemented in Internet of Things (IoT) technology.

Keywords — CoAP, Internet of Things, Network performance, Arduino mega 2560, constrained device.

1. Pendahuluan

Internet of Things adalah sebuah paradigma baru yang dengan pesat mendapatkan tempat dalam sebuah konsep pada telekomunikasi wireless modern. Ide dasar dari konsep ini ialah hadirnya penyebaran di masyarakat tentang berbagai macam hal atau objek, seperti *Radio Frequency Identification (RFID)*, *tags*, sensor, aktuator, ponsel, dan sebagainya. Dimana, melalui skema pengalamatan yang unik, mampu berinteraksi satu sama lain dan bekerja sama antar sesama untuk mencapai tujuan yang sama [1]. *Internet of Things* memiliki visi untuk menghubungkan miliaran perangkat ke internet. Namun, banyak dari perangkat ini diketahui sebagai objek yang *smart* sehingga memiliki keterbatasan *power supply*, pengolahan data dan memori [2].

Perangkat kecil yang dilengkapi Control Processing Unit, memori dan sumber daya listrik merupakan bagian dari constrained device. Constrained device menggunakan sensor atau aktuator, smart objects atau

smart devices membentuk sebuah jaringan, untuk kemudian dijadikan constrained nodes dalam jaringan tersebut [3]. Untuk mengatasi constrained device serta dapat memberikan alternative yang standar untuk protokol proprietary, *Internet Engineering Task Force (IETF)* telah memperkenalkan *Constrained Application Protocol (CoAP)* untuk perangkat *Internet of Things*. *Constrained Application Protocol (CoAP)* dikembangkan sebagai alternatif pengganti *HyperText Transfer Protocol (HTTP)* untuk menghubungkan perangkat dengan komputasi terbatas ke web. Dibandingkan dengan HTTP, CoAP memiliki less state, yang dapat diterapkan dengan keadaan memori yang lebih kecil dan memiliki komunikasi overhead serta delay yang lebih kecil [2].

Internet Engineering Task Force (IETF) juga telah mengusulkan metode CoRE (*Constrained RESTful Environments*) untuk mengatasi *constrained device* [4].

2. Metodologi

Penelitian ini menggunakan protokol *Constrained Application Protocol* (CoAP) sebagai protokol alternatif pengganti *HyperText Transfer Protokol* (HTTP) pada teknologi *Internet of Things* (IoT). Beberapa tahapan dalam penelitian ini terdiri dari : (1) penentuan komponen. (2) perancangan sistem yang akan dibuat berdasarkan dengan arsitektur *Internet of Things* (3) skenario pengujian pengambilan dataset.

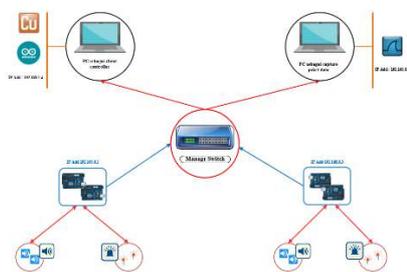
A. Penentuan Komponen

Penelitian ini menentukan komponen yang diperoleh berdasarkan referensi rfc 7252 yaitu komponen untuk *constrained device* menggunakan node dengan kapasitas 8-bit mikrokontroler dengan RAM dan ROM dalam jumlah kecil [5].

B. Perancangan Sistem

Perancangan sistem ini terdiri dari beberapa tahapan, diantaranya penentuan komponen, penentuan perangkat lunak dan perangkat keras, pembuatan topologi, serta skenario pengambilan dataset.

Pembuatan prototype atau topologi ini menggunakan beberapa device yaitu mikrokontroler arduino mega 2560 dengan output led dan buzzer sebagai informasi jika data sudah berhasil dikirim.



Gambar 1. Topologi pembuatan IoT dengan protokol CoAP

C. Skenario Pengujian Pengambilan Dataset

Skenario pengambilan data dilakukan secara berurutan untuk setiap payload dengan jumlah yang sama yaitu sepuluh kali

pengujian, untuk skenario pengambilan data dapat dilihat pada tabel 1.

Tabel 1. Skenario Pengambilan Data

Dataset	Jumlah Pengujian	Metode CoRE yang digunakan	Keterangan
Payload 100	10 kali pengujian	PUT	Led/buzzer pada arduino akan hidup jika diberi data led/buzzer bernilai 1 dan akan mati jika inputan data led/buzzer bernilai tidak sama dengan 1. Sedangkan pada web/client akan mendapatkan respon balik berupa nilai RTT/Latency.
Payload 200	10 kali pengujian		
Payload 300	10 kali pengujian		
Payload 400	10 kali pengujian		
Payload 500	10 kali pengujian		
Payload 600	10 kali pengujian		
Payload 700	10 kali pengujian		
Payload 800	10 kali pengujian		
Payload 900	10 kali pengujian		
Payload 1000	10 kali pengujian		
Payload 1100	10 kali pengujian		
Payload 1200	10 kali pengujian		
Payload 1300	10 kali pengujian		
Payload 1461	10 kali pengujian		

3. Hasil dan Pembahasan

Hasil pada penelitian di dapat berdasarkan hasil pengujian yang telah dilakukan berdasarkan topologi yang telah dirancang. Hasil pengujian ini ada beberapa parameter, yaitu *Throughput*, *packet loss*, *delay* dan *packet delivery ratio* (PDR).

Tabel 2. Nilai Rata-Rata Hasil Pengujian Dengan Output LED

Parameter Payload	Rata-rata Throughput (kbps)	Rata-rata Delay (detik)	Rata-rata Packet Loss (%)	Rata-rata Packet Delivery Ratio (%)
100	1,2257	0,6971	0%	100%
200	1,491	0,8191	0%	100%
300	1,5718	0,9949	0%	100%
400	1,9128	1,0254	0%	100%
500	1,9414	1,2656	0%	100%
600	2,1204	1,2796	0%	100%
700	2,1023	1,5481	0%	100%
800	2,4644	1,4861	0%	100%
900	2,4421	1,7255	0%	100%
1000	2,5035	1,8616	0%	100%
1100	2,3676	2,0909	0%	100%
1200	4,3537	1,5369	0%	100%
1300	4,4602	1,6115	0%	100%
1461	4,5737	1,8012	0%	100%

Berdasarkan tabel 2 nilai rata-rata terbesar *throughput* dengan output LED adalah 4,5737 kbps dan nilai rata-rata terkecil *throughput* 1,2257 kbps. Nilai rata-rata terbesar *delay* dengan output LED yaitu 2,0909 detik sedangkan nilai rata-rata terkecilnya 0,6971 detik. Untuk *paket loss* dan *paket delivery ratio* dengan output LED nilai rata-rata yang dihasilkan 0% untuk *packet loss* dan keberhasilan *packet delivery ratio* adalah 100%. Hasil tersebut diperoleh karena pengujian yang dilakukan dalam jaringan lokal yang menggunakan dua node untuk pengujiannya sehingga performa jaringan yang dihasilkan tidak begitu besar.

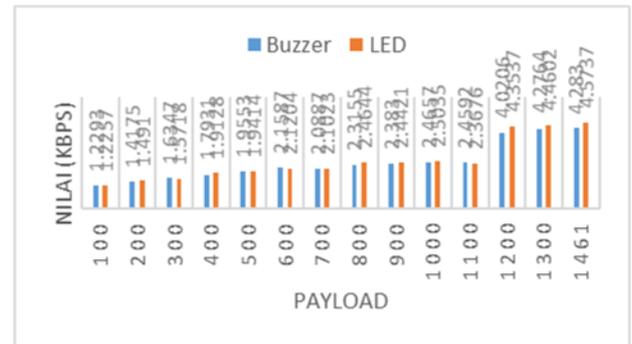
Tabel 3. Nilai Rata-Rata Hasil Pengujian Dengan Output Buzzer

Parameter Payload	Rata-rata Throughput (kbps)	Rata-rata Delay (detik)	Rata-rata Packet Loss (%)	Rata-rata Packet Delivery Ratio (%)
100	1,2293	0,6622	0%	100%
200	1,4175	0,8945	0%	100%
300	1,6347	0,9897	0%	100%
400	1,7931	1,1745	0%	100%
500	1,9553	1,2651	0%	100%
600	2,1587	1,3251	0%	100%
700	2,0887	1,5623	0%	100%
800	2,3155	1,5854	0%	100%
900	2,383	1,7032	0%	100%
1000	2,4657	1,9437	0%	100%
1100	2,4592	2,0327	0%	100%
1200	4,0206	1,7097	0%	100%
1300	4,2764	1,7288	0%	100%
1461	4,283	1,9411	0%	100%

Berdasarkan tabel 3 nilai rata-rata terbesar *throughput* dengan output buzzer adalah 4,283 kbps dan nilai rata-rata terkecilnya 1,2293 kbps. Untuk nilai rata-rata terbesar *delay* dengan output buzzer ialah 2,0327 detik dan nilai rata-rata terkecilnya 0,6622 detik. Jadi, jika nilai rata-rata kedua output

dibandingkan maka nilai rata-rata *throughput* pada LED lebih besar dibandingkan buzzer, sedangkan nilai rata-rata *delay* pada LED lebih kecil dibandingkan buzzer.

A. *Throughput*

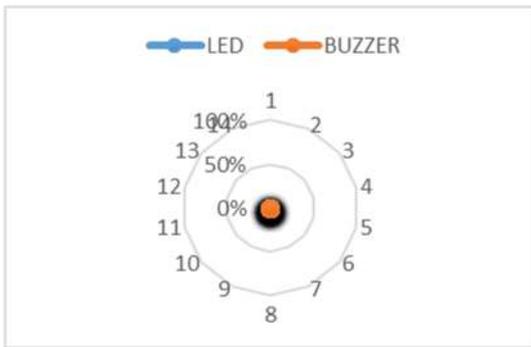


Gambar 2. grafik perbandingan *Throughput* dengan output LED dan Buzzer

Berdasarkan hasil pengujian UDP *Throughput* dengan jaringan lokal, berbasis pada *constrained device* yang mengacu pada tabel 2 dan tabel 3, nilai rata-rata UDP *throughput* yang dihasilkan dengan output led lebih besar dibandingkan nilai rata-rata yang dihasilkan dengan output buzzer. Hasil tersebut diperoleh berdasarkan rumus *throughput* yang sudah dijelaskan pada halaman sebelumnya, dimana jumlah data yang diterima oleh buzzer lebih kecil dibandingkan jumlah data yang diterima oleh led dan waktu pengiriman data oleh buzzer lebih besar dibandingkan dengan waktu pengiriman data oleh led.

Nilai terkecil UDP *throughput* dengan output LED adalah 1.096 kbps dan nilai terkecil UDP *throughput* dengan output buzzer adalah 0.969 kbps, sedangkan untuk nilai terbesar UDP *throughput* dengan output LED adalah 4.733 kbps dan nilai terbesar UDP *throughput* dengan output buzzer adalah 4.487 kbps.

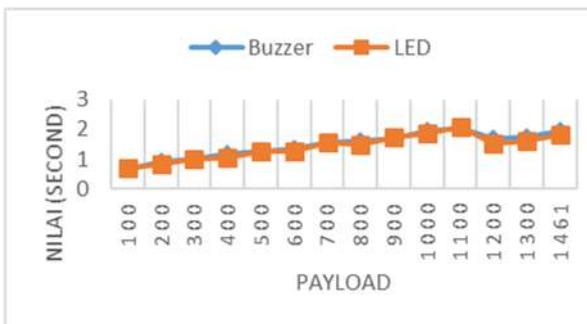
B. Packet loss



Gambar 3. grafik perbandingan Packet Loss dengan output LED dan Buzzer

Berdasarkan hasil pengujian UDP Packet Loss yang mengacu pada tabel 2 dan 3 nilai rata-rata yang diperoleh sama untuk kedua output tersebut yaitu 0%. Hal tersebut dikarenakan jaringan yang digunakan adalah jaringan lokal, teknik pengujian yang dilakukan secara offline dan perangkat yang digunakan dalam kondisi prima sehingga untuk kehilangan paket pada percobaan ini sangat kecil kemungkinannya.

C. Delay



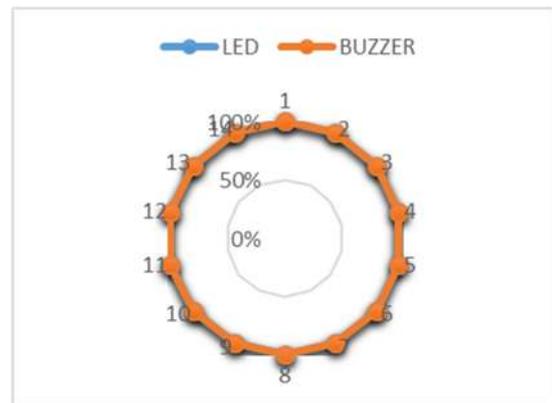
Gambar 4. grafik perbandingan Delay dengan output LED dan Buzzer

Berdasarkan pengujian delay dengan mengacu pada hasil tabel 2 dan 3 maka nilai rata-rata delay pada output buzzer lebih besar dibandingkan dengan delay yang dihasilkan pada output led. Hal tersebut disebabkan karena durasi waktu yang digunakan untuk mengirim paket pada output buzzer lebih besar sehingga ketika durasi waktu dibagi dengan jumlah paket yang diterima hasilnya akan semakin besar, karena rata-rata paket yang diterima hampir

sama jumlahnya untuk setiap payload akan tetapi durasi waktunya yang berbeda.

Untuk nilai rata-rata delay terbesar adalah 2.0327 detik terdapat pada payload 1100, sedangkan nilai rata-rata terkecil adalah 0.6622 detik terdapat pada payload 100. Untuk payload 1100 lebih besar nilai rata-rata delaynya dibandingkan dengan payload 1200 hingga 1461 itu disebabkan karena protokol Constrained Application Protocol (CoAP) memiliki karakteristik yang sama dengan protokol UDP yaitu connectionless.

D. Packet Delivery Ratio (PDR)



Gambar 5. grafik perbandingan Packet Delivery Ratio dengan output LED dan Buzzer

Berdasarkan hasil pengujian packet delivery ratio yang mengacu pada tabel 2 dan 3 maka nilai rata-rata PDR adalah 100%, keberhasilan tersebut di dapat berdasarkan hasil packet loss, karena pada pengujian ini tidak terdapat paket yang hilang dan keberhasilan PDR mencapai 100%. Faktor yang mempengaruhi keberhasilan PDR tersebut selain hasil paket loss yaitu jaringan yang digunakan adalah jaringan lokal, teknik pengujian secara offline dan kondisi perangkat prima sehingga kecil kemungkinan terjadi gangguan pada percobaan ini.

4. Kesimpulan

Hasil pengujian yang di dapat dari penelitian ini berdasarkan parameter yang telah di tentukan yaitu Throughput, Delay, Packet Loss dan Packet Delivery Ratio (PDR)

mendapatkan hasil yang baik untuk performa jaringan dengan menggunakan protokol CoAP sebagai protokol alternatif pengganti protokol HTTP yang di implementasikan pada *Internet of Things*. Berdasarkan tabel 2 nilai rata-rata terbesar *throughput* dengan output LED adalah 4,5737 kbps dan nilai rata-rata terkecil *throughput* 1,2257 kbps. Nilai rata-rata terbesar *delay* dengan output LED yaitu 2,0909 detik sedangkan nilai rata-rata terkecilnya 0,6971 detik. Untuk *paket loss* dan *paket delivery ratio* dengan output LED nilai rata-rata yang dihasilkan 0% untuk *packet loss* dan keberhasilan *packet delivery ratio* adalah 100%. Hasil tersebut diperoleh karena pengujian yang dilakukan dalam jaringan lokal yang menggunakan dua node untuk pengujiannya sehingga performa jaringan yang dihasilkan tidak begitu besar. Berdasarkan tabel 3 nilai rata-rata terbesar *throughput* dengan *output* buzzer adalah 4,283 kbps dan nilai rata-rata terkecilnya 1,2293 kbps. Untuk nilai rata-rata terbesar *delay* dengan *output* buzzer ialah 2,0327 detik dan nilai rata-rata terkecilnya 0,6622 detik. Jadi, jika nilai rata-rata kedua output dibandingkan maka nilai rata-rata *throughput* pada LED lebih besar dibandingkan buzzer, sedangkan nilai rata-rata *delay* pada LED lebih kecil dibandingkan buzzer.

Daftar Pustaka

- S. Li, L. Da Xu, and S. Zhao, (2015) "The internet of things: a survey," *Inf. Syst. Front.*, vol. 17, no. 2, pp. 243–259.
- M. Ilaghi, T. Leva, and M. Komu, (2014) "Techno-economic feasibility analysis of constrained application protocol," 2014 IEEE World Forum Internet Things, WF-IoT 2014, pp. 153–158.
- C. Bormann, (2014) "Terminology for Constrained-Node Networks," *Internet Eng. Task Force*, pp. 1–17.
- A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, (2015) "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376.
- IETF RFC 7252, (2014) *The Constrained Application Protocol (CoAP)*.
- I. Ishaq, J. Hoebeke, F. Van Den Abeele, I. Moerman, and P. Demeester, (2013) "Group Communication in Constrained Environments using CoAP-based Entities," pp. 345–350.
- D. Garcia-carrillo and R. Marin-lopez, (2016) "Lightweight CoAP-Based Bootstrapping Service for,"
- Rafiullah Khan, Salmad Ullah Khan, Rifaqat Zaheer, Shahid Khan, (2012) "Future Internet : The Internet of Things Architecture, Possible Application and Key Challenges", *International Conference on Frontiers of Information Technology*.
- Z. Shelby, (2014) "The Constrained Application Protocol (CoAP)," *Internet Eng. Task Force*, pp. 1–112.
- Nurul Hikmah, (2016) "Implementasi WSN Pada Pendeteksi Dini Kebakaran Dengan Metode Fuzzy Logic Sugeno", *Tugas Akhir, Fakultas Ilmu Komputer, Universitas Sriwijaya*.
- D. H. Networks, *Network Infrastructure And Architecture Designing High-Availability Networks*. 2008.
- J. Postel, (1980) "User Datagram Protocol," no. August, pp. 1–3.
- I. Oktariawan, Martinus, and Sugiyanto, (2013) "Pembuatan Sistem Otomasi Dispenser Menggunakan Mikrokontroler Arduino Mega 2560," *Jurnal FEMA, Volume 1, Nomor 2, April 2013*.

- M. Dan, (2016) "Modul praktikum mikroprosesor dan mikrokontroler," vol. 4, no. 24.
- J. Iptek and V. No, "Perancangan Prototype Sistem Kontrol Dan Monitoring Pembatas Daya Listrik Berbasis Mikrokontroler," vol. 16, no. 1, 2012.
- F..S.A.forouzan, Behrouz, College, De Anza, (2007) Data Communications And Networking_Forouzan_2.pdf.
- V. Mehta and N. Gupta, (2012) "Performance Analysis of QoS Parameters for Wimax Networks," vol. 1, no. 5, pp. 105-110.
- M. C. H. D. A. N. H and V. Call, "Analisis kualitas layanan," no. 1, pp. 138-143.
- M. B. Tamboli and D. Dambawade, (2016) "Secure and efficient CoAP based authentication and access control for Internet of Things (IoT)," 2016 IEEE Int. Conf. Recent Trends Electron. Inf. Commun. Technol., pp. 1245-1250.