

Pemanfaatan Sensor Arus untuk Efektifitas Penggunaan Daya Listrik pada Ruang Kelas Menggunakan Internet of Things

Current Sensor Usage toward Effectiveness of Class Room Electricity Power Consumption Using Internet of Things

Ahmad Khairol Maulidi¹, Fikra Tita Syifa^{2,*}, Gunawan Wibisono³

^{1,2}Prodi S1 Teknik Telekomunikasi, ³Prodi S1 Teknik Elektro
Fakultas Teknik Telekomunikasi dan Elektro, Institut Teknologi Telkom Purwokerto,
Jl. D.I Panjaitan No.128, Purwokerto, Jawa Tengah, Indonesia

^{2,*}Penulis korespondensi: fikra@ittelkom-pwt.ac.id

¹18101145@ittelkom-pwt.ac.id, ³wibisono@ittelkom-pwt.ac.id

Received on 14-09-2022, accepted on 27-01-2023, published on 31-01-2023

Abstrak

Seiring bertambahnya kebutuhan akan peralatan listrik yang digunakan di dalam suatu ruangan kelas maka kebutuhan akan energi listrik serta biaya yang akan dikeluarkan akan semakin besar. Sulitnya memonitor pemakaian perangkat listrik yang ada di kelas berpotensi terjadinya pemakaian perangkat secara berlebihan. Penelitian ini bertujuan merancang sistem *smart class* berbasis modul NodeMCU ESP8266, dengan rancangan ini perangkat yang ada di dalam kelas (*prototype*) dapat dimonitor dan dikendalikan secara jarak jauh. Sehingga dapat dilakukan analisa performansi kinerja saat dilakukan monitoring dan controlling jarak jauh. Termonitornya biaya listrik yang digunakan sehingga dapat mengatur perangkat dengan lebih efisien. Rancangan *prototype* ini menggunakan alat bantu, NodeMCU ESP8266, Modul sensor PZEM-004T, dan relay. Rangkaian tersebut menghasilkan data yang dikirim ke *platform* blynk yang diolah menjadi alat monitor dan kendali jarak jauh yang dapat dijalankan pada aplikasi ios. Hasil pengujian akurasi modul sensor PZEM-004T persentase *error* tegangan rata-rata hanya 0,78% sedangkan pada arus rata-rata persentase *error* hanya mempunyai persentase 12,90% saja. Hasil pengujian *Quality of Service* untuk memonitoring data dari NodeMCU ESP8266 ke *server* blynk mendapatkan *delay* rata-rata 116 ms, *throughput* rata-rata sebesar 5446,900 Kbps dan *packet loss* 0,023%. Berdasarkan parameter tiphon nilai *throughput*, *delay*, *packet loss* termasuk kategori sangat bagus. Pada pengujian fitur hasilnya didapatkan persentase 100% berhasil dari 30 kali pengujian. *prototype smart class* ini berhasil karena berdasarkan pengujian *prototype* ini dapat berjalan dengan baik.

Kata kunci: *Internet of things, NodeMCU ESP8266, PZEM-004T, Smart class.*

Abstract

Electrical energy and costs will increase as the need for electrical equipment in a classroom increases. The difficulty of monitoring the use of electrical devices in the class has the potential for excessive use. This study aims to design a smart class system based on the NodeMCU ESP8266 module. With this design, the class (*prototype*) devices can be monitored and controlled remotely. So that performance analysis can be carried out when monitoring and controlling remotely. Monitored the cost of electricity used so that it could manage devices more efficiently. This prototype design uses tools, NodeMCU ESP8266, PZEM-004T sensor module, and relay. The circuit generates data sent to the Blynk platform, which is processed into a remote monitoring and control tool that can be run on iOS applications. The results of testing the accuracy of the PZEM-004T sensor module have an average error percentage of only 0.78%, while the average current percentage error only has a percentage of 12.90%. The quality of Service

test results for monitoring data from the NodeMCU ESP8266 to the Blynk server gets an average delay of 116 ms, an average throughput of 5446,900 Kbps, and a packet loss of 0.023%. Based on the Typhon parameters, the throughput, delay, and packet loss values are very good. In the feature test, the results obtained a percentage of 100% successful from 30 times testing. This smart class prototype is successful because, based on testing, this prototype can run well.

Keywords: Internet of things, NodeMCU ESP8266, PZEM-004T, Smart class.

I. PENDAHULUAN

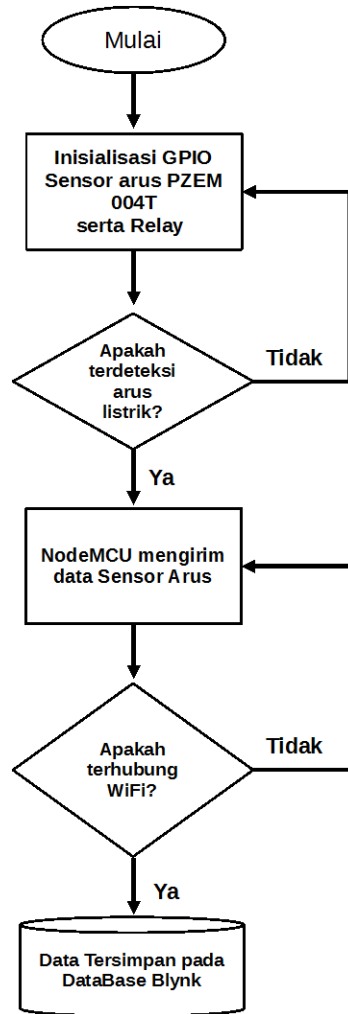
Seiring bertambahnya kebutuhan akan peralatan listrik yang digunakan di dalam suatu ruangan kelas maka kebutuhan akan energi listrik serta biaya yang akan dikeluarkan akan semakin besar. Penggunaan perangkat listrik pada ruang kelas yang berlebihan sering di jumpai di lingkungan kampus, seperti lupa mematikan perangkat listrik yang ada dikelas sehingga membuat kampus harus mengeluarkan biaya listrik lebih atas kelalaian tersebut. Hingga saat ini kebutuhan akan energi di Indonesia tergolong cukup besar yaitu mencapai kisaran angka 221.296 GWh, dan diprediksi akan selalu meningkat tiap tahunnya [1].

Pada ruang yang ada di sebuah kampus terdapat kendala untuk memonitor semua perangkat yang ada di dalam kelas. Selain itu perangkat listrik yang ada di dalam ruang kelas tidak dapat diketahui berapa biaya listrik yang digunakan dan berapa lama penggunaan perangkat listrik tersebut, sehingga akan sulit untuk menentukan prosedur yang tepat untuk dapat melakukan penghematan penggunaan daya listrik ada pada ruang kelas. Kebutuhan akan sebuah sistem yang dapat memonitor dan mengontrol perangkat listrik pada ruang kelas. Pada rancangan sistem ini merancang sebuah prototype yang dapat mengendalikan dan juga memonitor biaya penggunaan listrik pada ruang kelas. Dengan terealisasikannya rancangan prototype ini, maka setiap perangkat listrik yang terhubung dengan rancangan internet of things ini akan dapat dikendalikan menggunakan platform blynk yang dapat diinstal pada perangkat smartphone baik sistem operasi android maupun ios. Internet of things adalah konsep di mana suatu device dapat memiliki kemampuan untuk mengirim data melalui jaringan internet tanpa adanya interaksi manusia ke manusia atau antara manusia ke komputer [2]. Sehingga akan diketahui perangkat apa saja yang sedang aktif dan berapa lama perangkat tersebut aktif dapat diketahui juga dikendalikan apabila terdapat perangkat yang lupa dimatikan atau tidak diperlukan menggunakan perangkat yang terkoneksi dengan jaringan internet. Berdasarkan latar belakang tersebut sehingga diambil penelitian tentang Perancangan Smart Class untuk Efektifitas Penggunaan Daya Listrik Kampus dengan menggunakan Internet Of Things ini.

II. METODE PENELITIAN

Gambar 1 merupakan flowchart alur penelitian. Penelitian rancangan sistem ini terdiri dari beberapa tahapan. Tahap pertama studi literatur. Tahap ini dilakukan dengan tujuan untuk mengumpulkan, mempelajari, dan mengolah bahan mengenai rancangan smart class dan internet of things dari berbagai sumber. Sumber yang diambil berasal dari sumber yang dapat dipertanggung jawabkan seperti buku, jurnal, dan artikel lainnya. Tahap kedua perancangan prototype. Perancangan prototype, tahap ini merupakan proses perancangan desain smart class. perancangan prototype ini dilakukan untuk proses menentukan hasil yang bagus. Proses prototype merupakan proses di mana menggabungkan berbagai komponen yang di perlukan untuk perancangan smart class. Komponen yang digunakan antara lain ada sensor arus PZEM 004T V3, Mikrokontroler NodeMCU ESP8266, Module Relay 4 Channel, Kabel jumper dan kabel USB. Penyusunan program script pada arduino IDE yang bertujuan untuk memberikan perintah pada mikrokontroler.

Setelah penyusunan script program telah selesai dilakukan maka kemudian dilakukan tahap selanjutnya pengujian sistem. Pengujian dilakukan untuk menguji apakah program dan komponen yang sudah di rancang dapat berjalan. Jika program dan komponen sudah berjalan sesuai rancangan maka tahap ini berhasil dan dilanjutkan ke tahap selanjutnya, tetapi jika sistem tidak dapat berjalan maka akan dilakukan pengulangan tahap ke tahap sebelumnya yaitu perancangan prototype hardware. Tahap pengambilan data, tahap ini akan dilakukan pengambilan data yang diambil berupa nilai arus, tegangan dan lain lain. Setelah data yang diperlukan sudah terkumpul, maka tahap selanjutnya data akan dilakukan analisa dan tahap terakhir adalah pengambilan kesimpulan yang dihasilkan.



Gambar 1. Flowchart Kinerja Pengiriman Data

A. Perancangan Sistem Perangkat

1. NodeMCU ESP8266

NodeMCU adalah sebuah platform IoT yang bersifat *opensource*. Terdiri dari perangkat keras berupa *System On Chip ESP8266* dari ESP8266 buatan *Espressif System*. NodeMCU bisa dianalogikan sebagai *board arduino* yang terkoneksi dengan ESP8262. NodeMCU telah me-package ESP8266 ke dalam sebuah *board* yang sudah terintegrasi dengan berbagai *feature* selayaknya *microkontroler* dan kapasitas akses terhadap *wifi* dan juga *chip* komunikasi yang berupa *USB to serial*. Sehingga dalam pemrograman hanya dibutuhkan kabel data USB [3].

2. Sensor

Sensor adalah sebuah alat elektronik yang digunakan untuk mendeteksi adanya perubahan lingkungan. Fisik berupa gerakan, cahaya, panas, tekanan, dan fenomena-fenomena lingkungan lainnya. Pada dasarnya sensor dapat digolongkan sebagai transduser karena keluaran dari sensor dapat diubah menjadi tegangan atau besaran listrik. Biasanya sensor digunakan sebagai *input* pada suatu sistem [4].

3. Modul Sensor PZEM-004T

PZEM-004T adalah sensor yang dapat digunakan untuk mengukur tegangan *rms*, arus *rms* dan daya aktif yang dapat dihubungkan melalui *arduino* ataupun platform *open source* lainnya. Dimensi fisik dari papan PZEM-004T adalah $3,1 \times 7,4$ cm. Modul pzem-004t dibundel dengan kumparan trafo arus diameter 3mm yang dapat digunakan untuk mengukur arus maksimal sebesar 100A [5].

4. Relay

Relay adalah komponen elektronika berupa saklar elektronik yang digerakkan oleh arus listrik. Secara prinsip, *relay* merupakan tuas saklar dengan lilitan kawat pada batang besi (solenoid) di dekatnya. Ketika solenoid dialiri arus listrik, tuas akan tertarik karena adanya gaya magnet yang terjadi pada solenoid sehingga kontak saklar akan menutup. Pada saat arus di hentikan, gaya magnet akan hilang, tuas akan kembali ke posisi semula dan kontak saklar kembali terbuka. *Relay* biasanya digunakan untuk menggerakkan arus / tegangan yang besar (misalnya peralatan listrik 4 A / AC 220 V) dengan memakai arus / tegangan yang kecil (misalnya 0.1 A / 12 Volt DC) [6].

5. Blynk

Blynk merupakan *platform* sistem operasi *Android* maupun *iOS* sebagai kendali pada modul *Arduino*, *Raspberry Pi*, *ESP8266* dan perangkat sejenis lainnya melalui internet. Aplikasi *blynk* ini sangat mudah digunakan meskipun baru menggunakannya, untuk bisa terhubung dengan *board arduino blynk* dibutuhkan sebuah kode yang disebut *auth token*. Kode tersebut akan dapat dilihat melalui dasbor *blynk* dan dimasukkan ke dalam program yang telah dibuat, Selain itu juga dibutuhkan instalasi *library blynk* untuk dapat menggunakannya [7].

6. Quality of Service (QoS)

Quality of Service adalah teknik untuk mengelola *bandwidth*, *delay*, dan *packet loss* untuk aliran dalam jaringan. Tujuan dari mekanisme *QoS* adalah mempengaruhi setidaknya satu diantara empat parameter dasar *QoS* yang telah ditentukan. *QoS* didesain untuk membantu *end user (client)* menjadi lebih produktif dengan memastikan bahwa *user* mendapatkan *performansi* yang handal dari aplikasi-aplikasi berbasis jaringan. *QoS* mengacu pada kemampuan jaringan untuk menyediakan layanan yang lebih baik pada *trafik* jaringan tertentu melalui teknologi yang berbeda-beda. *QoS* merupakan suatu tantangan yang besar dalam jaringan berbasis IP dan internet secara keseluruhan [8].

7. Throughput

Throughput merupakan kinerja jaringan yang terukur. *Throughput* Merupakan jumlah total kedatangan paket yang berhasil yang diamati pada *destination* selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut [8]. Rumus yang digunakan untuk mencari *throughput* adalah:

$$\text{Throughput} = \frac{\text{Jumlah Paket}}{\text{Waktu}} \quad (1)$$

Tabel 1. Parameter throughput tiphon [9].

Kategori throughput	Presentase	Indeks
<i>Bad</i>	0 - 338 kbps	0
<i>Poor</i>	338 - 700 kbps	1
<i>Fair</i>	700 - 1200 kbps	2
<i>Good</i>	1200 kbps - 2,1 Mbps	3
<i>Excelent</i>	>2,1 Mbps	4

8. Packet loss

Packet Loss merupakan suatu parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang, dapat terjadi karena *collision* atau tabrakan antar paket dan *congestion* atau penuhnya *trafik* data pada jaringan dan hal ini berpengaruh pada semua aplikasi karena *retransmisi* (pengiriman kembali) akan mengurangi efisiensi jaringan secara keseluruhan meskipun jumlah *bandwidth* cukup tersedia untuk aplikasi-aplikasi tersebut. Umumnya perangkat jaringan memiliki *buffer* untuk menampung data yang diterima. Jika terjadi kongesti yang cukup lama, *buffer* akan penuh, dan data baru tidak akan diterima [10].

Tabel 2. Parameter packet loss tiphon [9].

Categories of Degradation	Packet Loss
<i>Very good</i>	0% ≤ 3%
<i>Good</i>	3% ≤ 15%

Good Enough	$15\% \leq 25\%$
Ugly	$>25\%$

9. Delay

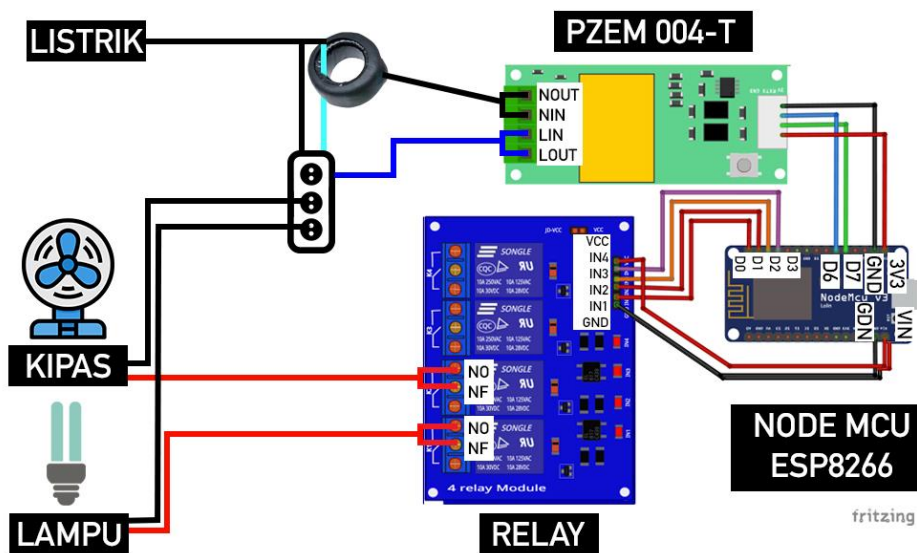
Delay (Latency) merupakan waktu yang dibutuhkan data untuk menempuh jarak dari asal ke tujuan. Delay dapat mempengaruhi oleh jarak, media fisik, kongesti atau juga waktu proses yang lama [10]. Untuk menghitung nilai Delay (Latency) digunakan persamaan sebagai berikut :

$$\text{Delay} = \text{Waktu packet dikirim} - \text{Waktu packet diterima} \tag{2}$$

Tabel 3. Parameter delay tiphon [9].

Categories of Degradation	Delay/Latency
Very Good	$< 150 \text{ ms}$
Good	$150 \text{ s.d. } 300 \text{ ms}$
Ugly	$300 \text{ s.d. } 450 \text{ ms}$
Tidak dapat diterima	$>450\text{ms}$

B. Skematik Rangkaian



Gambar 2. Skematik Rangkaian

Pada gambar 2 NodeMCU ESP8266 berfungsi sebagai mikrokontroler yang menerima data dari sensor arus PZEM 004T yang kemudian data tersebut dikirim melalui fitur wifi yang tersedia pada NodeMCU ke dalam cloud platform blynk. Sensor arus PZEM 004T ini berfungsi untuk mendeteksi , energy, voltase, current perangkat listrik yang terhubung dengan sensor. Pin MI GND dari sensor PZEM 004T terhubung dengan NodeMCU ESP8266 pin GND. Pin MI Rx dari sensor PZEM 004T terhubung dengan NodeMCU ESP8266 pin D7. Pin MI Tx dari sensor PZEM 004T terhubung dengan NodeMCU ESP8266 pin D6. Pin MI VCC dari sensor PZEM 004T terhubung dengan NodeMCU ESP8266 pin 3V3. Pin LOUT dari sensor PZEM 004T terhubung dengan stop kontak dan lampu dengan cara di splitter. Pin LIN dari sensor PZEM 004T terhubung dengan stop kontak dan lampu dengan cara di splitter. Pin NIN dari sensor PZEM 004T terhubung dengan splitter. Pin NOUT dari sensor PZEM 004T terhubung dengan splitter. Relay 5V 4 Channel. Relay yang digunakan pada rancangan sistem ini menggunakan relay 5V 4 Channel, Relay ini

berfungsi sebagai penggerak serta memutus aliran listrik pada lampu. Pin NO dari *Relay 5V 4 Channel* terhubung dengan steker. Pin COP dari *Relay 5V 4 Channel* terhubung dengan steker. Pin GND dari *Relay 5V 4 Channel* terhubung dengan *pin GND* pada *NodeMcu ESP8266*. Pin In1 dari *Relay 5V 4 Channel* terhubung dengan *pin d0* pada *NodeMcu ESP8266*. Pin In2 dari *Relay 5V 4 Channel* terhubung dengan *pin d1* pada *NodeMcu ESP8266*. Pin In3 dari *Relay 5V 4 Channel* terhubung dengan *pin d2* pada *NodeMcu ESP8266*. Pin In4 dari *Relay 5V 4 Channel* terhubung dengan *pin d3* pada *NodeMcu ESP8266*. Pin *Vcc* dari *Relay 5V 4 Channel* terhubung dengan *pin vin* pada *NodeMcu ESP8266*.

III. HASIL DAN PEMBAHASAN

A. Pengujian Modul Sensor PZEM004-T

Pengujian modul sensor PZEM004-T ini bertujuan untuk mengetahui seberapa akurat sensor ini bekerja. Pengujian yang dilakukan dengan cara membandingkan hasil data dari PZEM004-T dengan multimeter digital membandingkan hasil data dari kedua alat tersebut sehingga diketahui selisih dari nilai yang didapat. Dengan demikian akan diketahui seberapa akurat atau seberapa tinggi tingkat *error* alat ini. Pengujian alat ini ada 2 pengujian yaitu pengujian arus dan pengujian ampere berikut data dari hasil pengujian tersebut:

1. Pengujian Tegangan

Pengujian tegangan dilakukan dengan menggunakan alat auto trafo yang mempunyai kemampuan mengatur tegangan sampai dengan 250 volt. Penggunaan auto trafo diperlukan untuk mendapatkan nilai tegangan yang diperlukan, pada pengujian ini tegangan yang diuji dari rentan 200 volt sampai dengan 230 volt.

Tabel 4. Hasil Pengujian rata-rata tegangan

Pengujian ke-	Nilai tegangan yang diuji	Rata-Rata Multi Meter	Rata-Rata PZEM 004T	Error
1	200	200,4	201,7	0,70%
2	201	201,2	202,8	0,70%
3	202	202,3	203,8	0,69%
4	203	203,2	204,6	0,74%
5	204	204,2	205,6	0,78%
6	205	205,1	206,6	0,73%
7	206	206,2	207,8	0,87%
8	207	207,1	208,6	0,77%
9	208	208,1	209,6	0,72%
10	209	209,3	210,9	0,67%
11	210	210,4	212,1	0,81%
12	211	211,4	213,0	0,85%
13	212	212,2	213,7	0,80%
14	213	213,5	215,2	0,94%

Tabel 5. Hasil Pengujian rata-rata tegangan (lanjutan)

Pengujian ke-	Nilai tegangan yang diuji	Rata-Rata Multi Meter	Rata-Rata PZEM 004T	Error
15	214	214,4	216,0	0,79%
16	215	215,3	217,0	0,79%
17	216	216,3	218,1	0,88%
18	217	217,2	219,1	0,92%
19	218	218,2	219,9	0,83%
20	220	220,0	221,6	0,77%
21	221	221,3	222,9	0,81%
22	222	222,4	223,7	0,31%
23	223	223,4	225,0	0,76%
24	224	224,4	225,9	0,71%

25	225	225,2	227,1	0,89%
26	226	226,3	228,0	0,75%
27	227	227,4	229,2	0,79%
28	228	228,2	230,1	0,92%
29	229	229,1	230,8	0,87%
30	230	230,4	232,4	0,91%
Rata-rata				0,78%

Pada tabel di atas persentase *error* dari nilai tegangan yang dibaca sensor dan alat ukur multimeter dibandingkan. Sehingga diketahui selisih nilai yang dihasilkan dan diuji sebanyak 30 kali selama 3 kali. Nilai *error* dari modul sensor ini bisa dikatakan sangat akurat karena persentase *error* dari hasil pengujian ini mendapatkan nilai persentase *error* tertinggi hanya 0,94%. Modul sensor PZEM 004T masih sangat mempunyai untuk digunakan pada rancangan sistem ini.

2. *Pengujian Arus*

Pada pengujian ini dilakukan sehingga diketahui selisih nilai dan persentase *error* dari pembacaan nilai rancangan sistem ini. tuk dapat mencapai nilai 10A.

Tabel 6. Hasil Pengujian Arus.

No.	Arus (A)			
	Rata-rata Tang Ampere	Rata-rata PZEM	Rata-rata Selisih	Error
1	1,47	1,662	0,192	13%
2	1,51	1,230	0,276	17%
3	1,76	1,769	0,026	2%
4	2,89	1,530	1,357	48%
5	5,37	5,540	0,447	9%
6	2,43	2,705	0,275	12%
7	3,94	4,326	0,386	10%
8	6,74	7,271	0,531	8%
9	7,71	8,264	0,558	8%
10	10,84	11,315	0,478	4%
Rata-rata			0,453	12,90%

B. *Quality of Service (QoS)*

Pengujian *QoS* ini dilakukan untuk mengetahui kualitas jaringan alat. Pengujian dilakukan dengan menggunakan *software wireshark* dengan cara membuat sebuah jaringan *hotspot* dari laptop yang mana jaringan hotspot tersebut digunakan untuk menghubungkan modul *wifi NodeMCU ESP8266* sehingga dapat terhubung ke internet dan dapat dimonitor *packetnya*. Skenario pengujian dilakukan dengan 8 variabel yang berbeda.

1. *Throughput*

Pengujian *Throughput* dilakukan dengan tujuan untuk mengetahui nilai dari jumlah *packet* yang berhasil diterima. Pengujian *throughput* ini dilakukan sebanyak 8 kali dengan menggunakan 8 variabel yang berbeda dan di uji sekitar 2 menit. Hasil dari *throughput* perhitungan tersebut didapatkan melalui perhitungan dengan dengan rumus:

$$Throughput = \frac{Jumlah\ Paket}{Waktu} \times 8 \tag{3}$$

Tabel 7. Hasil pengujian *throughput*.

Pengujian	Beban yang digunakan	Jumlah <i>packet</i>	<i>Throughput (Kbps)</i>
-----------	----------------------	----------------------	--------------------------

1	Solder	1060	5370
2	Bor	1079	5514
3	Charger Laptop dan Smartphone	1043	5243
4	Lampu dan Solder	1104	5613
5	Charger Laptop	849	4323
6	Charger Smartphone	1063	5415
7	Lampu	1023	6259
8	Kipas Angin	1133	5838
Rata-rata			5446,900

Tabel 6 merupakan tabel pengujian *QoS* dengan pengujian sebanyak 8 kali percobaan. Data diambil dari saat alat berjalan dan data yang terbaca dikirimkan ke *database blynk* yaitu *blynk-cloud.com* lalu ditampilkan ke aplikasi *blynk* yang telah terinstal pada *ios*. Berdasarkan tabel 6 diatas rata nilai *throughput* didapatkan nilai rata-rata sebesar 5446,900 *Kbits/Second*. Dimana menurut standar *tiphon* nilai tersebut sangat bagus.

2. Delay

Berikut hasil dari pengujian *delay* dari *NodeMCU ESP8266* ke *database blynk-cloud.com*, hasil data tersebut didapatkan melalui hasil perhitungan menggunakan rumus berikut :

$$Delay = Waktu\ packet\ dikirim - Waktu\ packet\ diterima \quad (4)$$

Tabel 8. Hasil pengujian delay.

Pengujian	Beban yang digunakan	Jumlah <i>packet</i>	<i>Delay</i> (s)
1	Solder	1060	0,115022
2	Bor	1079	0,11263543
3	Charger Laptop dan Smartphone	1043	0,11752704
4	Lampu dan Solder	1104	0,11008481
5	Charger Laptop	849	0,14297491
6	Charger Smartphone	1063	0,1127094
7	Lampu	1023	0,11229393
8	Kipas Angin	1133	0,10760956
Rata-rata			0,116357

3. Packet Loss

Pada pengujian ini akan dilakukan *capture* jaringan menggunakan *wireshark* selama sekitar 2 menit selama 8 kali menggunakan 8 variasi beban yang berbeda. Beban tersebut meliputi kipas angin, lampu, bor, solder dan kombinasi dari keempat alat tersebut

Tabel 9. pengujian *packet loss*

Pengujian	Beban yang digunakan	Jumlah <i>packet</i>	Jumlah <i>packet loss</i>
1	Solder	1060	0%
2	Bor	1079	0%
3	Charger Laptop dan Smartphone	1043	0%
4	Lampu dan Solder	1104	0%
5	Charger Laptop	849	0%
6	Charger Smartphone	1063	0,094%
7	Lampu	1023	0,093%
8	Kipas Angin	1133	0%
Rata-rata			0,023%

IV. KESIMPULAN

Berdasarkan hasil pengujian yang telah dijabarkan pada buku tugas akhir bahwa dapat disimpulkan: Perancangan sistem *smart class* dikatakan berhasil karena alat dapat membaca dan mengirimkan nilai arus (A), tegangan (V), daya aktif (watt) dan energi (kWh) ke platform *blynk* sehingga data tersebut dapat ditampilkan pada aplikasi *ios* yang telah dirancang. Setelah dilakukan pengujian tegangan dan arus untuk mengetahui akurasi modul sensor PZEM-004T, didapatkan hasil pada pengujian tegangan persentase *error* tertinggi sebesar 0,94% sedangkan pada pengujian arus didapatkan hasil persentase tertinggi *error* sebesar 13,10%. Hasil pengujian *Quality of Service* pada *throughput* dikategorikan sangat bagus karena berdasarkan parameter *tiphon* nilai *throughput* diatas 2100 *Kbps* sedangkan nilai rata-rata sebesar 5446,9 *Kbps*. Pada pengujian delay didapatkan nilai rata-rata 116 ms nilai ini dikategorikan sangat bagus begitu pun dengan nilai *packet loss* juga dikategorikan sangat bagus karena didapatkan hasil total rata-rata *packet loss* sebesar 0,023%.

DAFTAR PUSTAKA

- [1] E. Penggunaan, E. Listrik Herpendi, A. Noor, and R. Sayyidati, "Pengembangan Asisten TV Berbasis Internet of Things (IoT) untuk Efisiensi Penggunaan Energi Listrik," *eksplora.stikom-bali.ac.id*, 2019, doi: 10.30864/eksplora.v9i2.270.
- [2] S. Kendali *et al.*, "Sistem Kendali Peralatan Elektronik Rumah Tangga Berbasis Internet Of Things Menggunakan Protokol MQTT," *journal.universitashumigora.ac.id*, vol. 2, no. 1, 2020, doi: 10.30812/bite.v2i1.799.
- [3] A. Romoadhon, D. A.- Rekeyasa, and undefined 2017, "Sistem Kontrol Peralatan Listrik pada Smart Home Menggunakan Android," *journal.trunojoyo.ac.id*, vol. 10, no. 2, pp. 116–122, 2017, Accessed: Feb. 01, 2022. [Online]. Available: <https://journal.trunojoyo.ac.id/rekeyasa/article/view/3613>
- [4] A. Sanad, S. Sumaryo, P. S1, and T. Elektro, "Perancangan Sistem Dan Monitoring Penerangan Lampu Otomatis Di Tempat Parkir Berbasis Internet Of Things (iot)," ... *telkomuniversity.ac.id*, vol. 5, no. 3, p. 4100, 2018, Accessed: Aug. 01, 2022. [Online]. Available: <https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/8149>
- [5] S. Anwar, T. Artono, N. Nasrul, ... D. D.-P. S., and undefined 2019, "Pengukuran Energi Listrik Berbasis PZEM-004T," *e-jurnal.pnl.ac.id*, vol. 3, no. 1, 2019, Accessed: Feb. 01, 2022. [Online]. Available: <http://e-jurnal.pnl.ac.id/semnaspnl/article/view/1694>
- [6] M. Mantasia, N. Nurhayati, and & Y. Y.-E. T., "RANCANG BANGUN PENGATUR WAKTU OTOMATIS UNTUK PERALATAN LISTRIK RUMAH TANGGA," *ojs.unm.ac.id*, Accessed: Feb. 01, 2022. [Online]. Available: <https://ojs.unm.ac.id/JETC/article/view/24318>
- [7] I. Santoso, M. Farid Adiwisastro, B. Kelana Simpony, D. Supriadi, and D. Silvi Purnia, "Implementasi NodeMCU dalam Home Automation dengan Sistem Kontrol Aplikasi Blynk," *repository.bsi.ac.id*, vol. 9, no. 1, p. 2021, 2021, Accessed: Feb. 01, 2022. [Online]. Available: <https://repository.bsi.ac.id/index.php/unduh/item/312933/10459-29882-1-PB.pdf>
- [8] A. Budiman, M. Duskamaen, H. A.-P. J. Pendidikan, and undefined 2020, "Analisis Quality of Service (Qos) Pada Jaringan Internet Smk Negeri 7 Jakarta," *journal.unj.ac.id*, Accessed: Aug. 04, 2022. [Online]. Available: <http://journal.unj.ac.id/unj/index.php/pinter/article/view/18964>
- [9] "Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); General aspects of Quality of Service (QoS)," 1999, Accessed: Aug. 10, 2022. [Online]. Available: <http://www.etsi.org>
- [10] Y. E.-J. I. I. K. F. Ilmu and undefined 2018, "Internet of Things (IOT) sistem pengendalian lampu menggunakan Raspberry PI berbasis mobile," *ejournal.fikom-unasman.ac.id*, vol. 4, no. 1, 2018, Accessed: Aug. 10, 2022. [Online]. Available: <http://ejournal.fikom-unasman.ac.id/index.php/jikom/article/view/41>