Jurnal Elektro
p-ISSN: 1979-9780
e-ISSN: 2746-4288

Jurnal Elektro Vol.17 No.2 November 2024
journal homepage: https://ejournal.atmajaya.ac.id/index.php/JTE

Pengembangan Perangkat Lunak Untuk Akuisisi Data Konsumsi Daya Listrik Dari *Power Meter*

Ignatius Ginting[1], Melisa Mulyadi[2*], Linda Wijayanti[3], Ferry Rippun[4]

[1,4]Department of Electrical Engineering, Faculty of Engineering
Atma Jaya Catholic University of Indonesia, Jakarta 12930, INDONESIA
[2,3]Engineer Profession Program, Faculty of Engineering
Atma Jaya Catholic University of Indonesia, Jakarta 12930, INDONESIA

Article Info

Article history:

Received
8 November 2024

Accepted
12 November 2024

Keywords:
Internet, Node MCU, power meter, Python, ThingSpeak, Three phase

Abstract

Technological developments make it possible to monitor electricity consumption in real-time via the internet. The system developed uses a power meter to collect data, Python to process data and IoT connectivity via NodeMCU ESP8266 to send data to the internet. Python is used so that measurement data can be processed and presented in an easy-to-understand format. The test results show that electrical data measurements can be displayed with an error ranging from 0.04% -11.56%. This is because there is a delay from the time the power meter is measured until the data is displayed on ThingSpeak.

Info Artikel

Histori Artikel:

Diterima:
8 November 2024

Disetujui:
12 November 2024

Kata Kunci:
Internet, Node MCU, power meter, Python, ThingSpeak, Tiga fasa

Abstrak

Perkembangan teknologi memungkinkan pemantauan konsumsi listrik secara real-time melalui internet. Sistem yang dikembangkan menggunakan power meter untuk mengumpulkan data, Python untuk mengolah data dan konektivitas IoT melalui NodeMCU ESP8266 untuk mengirimkan data ke internet. Python digunakan agar data pengukuran dapat diolah dan disajikan dalam format yang mudah dipahami. Hasil pengujian menunjukkan pengukuran data besaran listrik dapat ditampilkan dengan kesalahan berkisar 0,04%-11,56%. Hal ini karena adanya delay dari saat pengukuran oleh power meter sampai data ditampilkan di ThingSpeak.

1. PENDAHULUAN

Menurut Kementerian Energi dan Sumber Daya Mineral (ESDM) Republik Indonesia dalam siaran pers nomor: 28. Pers/04/SJI/2024 pada tanggal 15 Januari 2024 diberitakan bahwa konsumsi listrik masyarakat meningkat, pada tahun 2023 mencapai 1.285 kWh/Kapita [1]. Peningkatan penggunaan energi listrik sudah tentu akan berdampak pada pengeluaran biaya penggunaan energi listrik tersebut [2]. Oleh karena itu pengukuran besaran listrik menjadi aspek krusial dalam pemantauan konsumsi energi, baik di sektor domestik maupun industri. Untuk itu diperlukan *power meter* yang dapat mengukur berbagai parameter listrik seperti tegangan, arus, dan daya. Namun, data pengukuran yang dihasilkan oleh *power meter* seringkali membutuhkan pengolahan lebih lanjut agar dapat diinterpretasikan dengan mudah. Pada penelitian ini diimplementasikan bahasa pemrograman *Python* untuk memvisualisasikan data pengukuran dalam bentuk grafik yang lebih mudah dipahami. Selain itu, dengan memanfaatkan teknologi IoT melalui NodeMCU ESP8266, data tersebut dapat diakses secara *real-time* melalui *smart phone*, sehingga memungkinkan pemantauan konsumsi energi menjadi lebih efektif.

2. TINJAUAN PUSTAKA

2.1 Bahasa Pemrograman *Python*

Bahasa pemrograman *Python* telah menjadi salah satu bahasa yang pemrograman yang paling populer dan banyak digunakan di berbagai bidang, termasuk analisis data [3][4].

Python dapat melakukan beberapa instruksi multiguna secara langsung (intepratif) dengan orientasi objek serta menggunakan sistem dinamis untuk mejalankan perintah atau syntaxnya. *Python* juga

Melisa Mulyadi
Email address: melisa.mulyadi@atmajaya.ac.id

memiliki fungsi *backend* yang kompleks untuk dijalankan pada web saat menampilkan informasi kepada penggunanya. Contohnya, situs web harus berinteraksi dengan basis data, berkomunikasi dengan situs web lain, dan melindungi data saat mengirimkannya melalui jaringan. Bahasa pemrograman *python* telah menyediakan semua peralatan yang dibutuhkan untuk membangun aplikasi web dengan cepat dan lebih mudah.

2.2 Tkinter

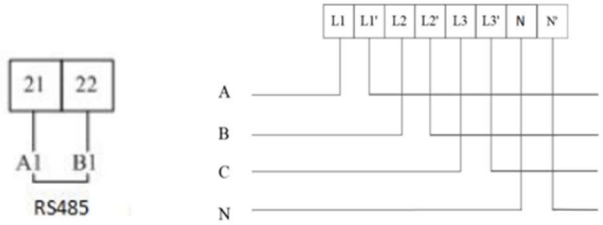
Tkinter merupakan bahasa pemrograman *python* yang membantu pembuatan aplikasi tampilan atau *Graphical User Interface* (GUI). Tkinter merupakan *Library* yang memberikan kemudahan pembuatan program berbasis grafis dan desain. Dalam pembuatan GUI terdapat beberapa GUI atau widget yang akan berinteraksi dengan *user* seperti widget *per user interface* yaitu *button, scrollbar, listbox, check button, radio button, label text* dan lain sebagainya.

2.3 *ThingSpeak*

ThingSpeak adalah *platform open source* untuk *Internet of Things* (IOT) dengan *Application Programming Interface* (API) terbuka yang berguna sebagai *server* basisdata untuk menyimpan data yang diproses oleh mikrokontroler serta berkomunikasi melalui jaringan internet [5]. Data yang dikumpulkan berasal dari perangkat node berupa sensor-sensor yang sudah terhubung ke internet dan juga memungkinkan pengambilan data dari perangkat lunak untuk keperluan visualisasi, notifikasi, kontrol dan analisis historis data.

2.4 *Power Meter*

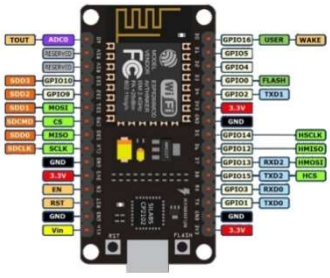
Power meter adalah alat yang berguna untuk membaca dan menampilkan tegangan, arus, frekuensi, faktor daya, dan parameter lainnya [6][7]. Alat ini mengukur daya listrik yang dikonsumsi oleh suatu perangkat atau sistem melalui pengukuran tegangan dan arus listrik yang melewati perangkat atau sistem tersebut. *Power meter* juga dapat mengukur daya listrik yang dihasilkan termasuk daya aktif, daya reaktif, faktor daya dan harmonik. *Power meter* dilengkapi pula dengan kemampuan untuk melakukan pengukuran, pemantauan secara *real time* dan menyimpan data pengukuran yang dapat dianalisis lebih lanjut, misal: identifikasi tren konsumsi energi yang membantu untuk perencanaan dan pengaturan penggunaan energi. Gambar 1 memperlihatkan cara menghubungkan *power meter*.



Gambar 1. Koneksi *power meter*

2.5 NodeMCU

NodeMCU merupakan *platform* berbasis IoT yang bersifat *open source*. Terdiri dari perangkat keras berupa *system on chip* ESP8266 [8]. NodeMCU dapat digunakan untuk berkomunikasi dan bertukar data dengan perangkat lain di jaringan internet. Hal ini membuat NodeMCU sangat cocok untuk pengembangan aplikasi *Internet of Things* (IOT) yang memerlukan konektivitas nirkabel. Modul NodeMCU ditunjukkan pada Gambar 2.



Gambar 2. Modul nodeMCU

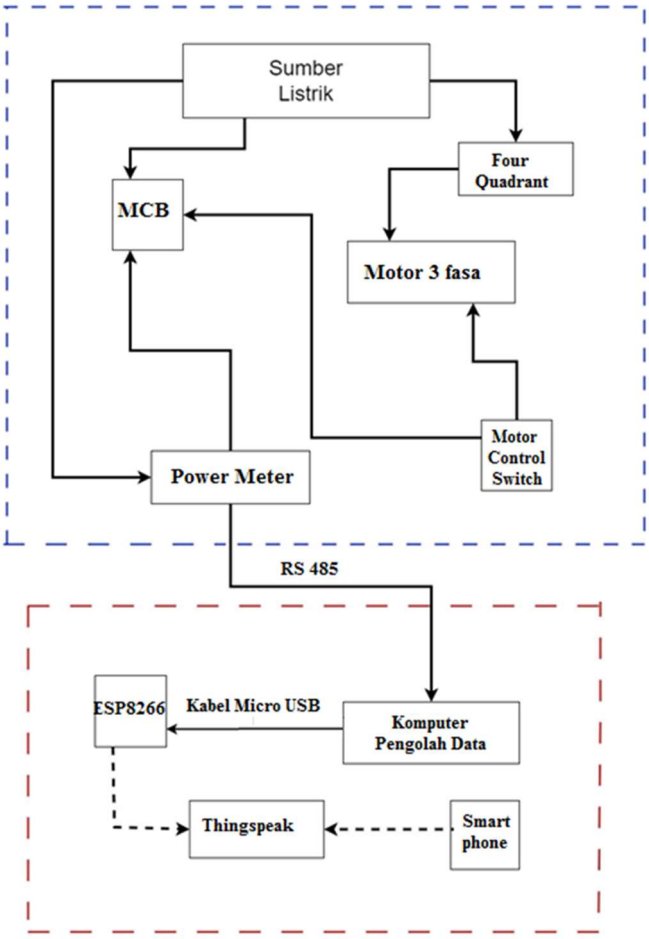
NodeMCU dilengkapi dengan sejumlah pin GPIO yang dapat digunakan untuk mengontrol dan berinteraksi dengan berbagai perangkat eksternal, seperti sensor, aktuator, dan perangkat lainnya.

3. METODOLOGI

Metode penelitian yang digunakan adalah penelitian eksperimental yang diawali dengan studi literasi untuk menginventarisir konsep/teori yang relevan seperti standard yang berlaku, buku pedoman, dan artikel sejenis. Tahap kedua adalah membuat rancangan perangkat lunak antara lain: pemrograman untuk pengolahan data pengukuran dari *power meter* dengan bahasa pemrograman *python* dan pemograman NodeMCU menggunakan perangkat lunak Arduino IDE. Mengintegrasikan antar perangkat melalui komunikasi serial. Data pengukuran yang diterima Arduino kemudian diubah menjadi format JSON [9], [10] agar dapat dibaca oleh *ThingSpeak*. Setelah semua sistem terpasang maka dilakukan pengujian untuk melihat unjuk kerja dari sistem yang dibuat melalui pengujian. Tahapan penelitian diakhiri dengan menganalisis hasil yang sudah didapatkan.

4. HASIL DAN PEMBAHASAN

Sistem yang dirancang menggunakan modul motor listrik dari laboratorium Mesin Listrik yang ada di Universitas Katolik Indonesia Atma Jaya untuk mengukur besaran-besaran listrik dan kinerja beban di motor listrik tersebut. Sistem ini menggunakan tiga komponen utama yang saling berhubungan, yaitu motor sebagai beban, *power meter* untuk membaca besaran listrik, dan program tampilan untuk menampilkan hasil pembacaan *power meter*. Diagram blok sistem dapat dilihat pada Gambar 3.



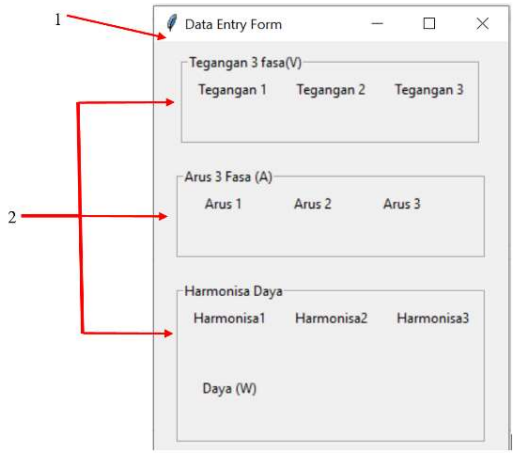
Gambar 3. Diagram Blok Sistem

Dalam perancangan ini, motor induksi tiga fasa menjadi beban utama. Motor akan menghasilkan karakteristik listrik seperti tegangan, arus, dan daya. Untuk mengatur kecepatan putar motor, digunakan inverter atau drive yang memiliki empat kuadran operasi. Motor dikendalikan oleh sebuah saklar kontrol motor. Aliran daya dari sumber menuju motor dilindungi oleh MCB. *Power meter* akan mengukur parameter listrik motor dan mengirimkan data tersebut ke komputer melalui protokol komunikasi RS-485. Data yang telah diolah oleh komputer kemudian dikirimkan ke platform IoT *ThingSpeak* menggunakan perangkat NodeMCU ESP8266.

4.1 Perancangan Tampilan

Dengan memanfaatkan *Library* Tkinter di *Python*, dikembangkan GUI untuk menampilkan hasil pengukuran. Tkinter berfungsi untuk membangun jendela utama (*window*) dan berbagai komponen tampilan lainnya (*widget*). Variabel pengukuran ditampilkan pada label-label yang disusun dalam bentuk *grid* di dalam sebuah *frame* seperti yang ditunjukkan Gambar 4. Setiap baris *grid* mewakili satu jenis pengukuran, sementara setiap kolom berisi nilai pengukuran yang bersesuaian. Konfigurasi *grid* ini memungkinkan tampilan data yang terstruktur dan mudah dibaca.

Aplikasi ini menggunakan *window* tkinter sebagai tampilan utama. Data dari *power meter* (tegangan, arus, harmonisa, daya) akan ditampilkan secara *real-time* dalam kelompok-kelompok *widget* yang berbeda, dengan masing-masing kelompok menunjukkan 3 data pengukuran terbaru.
Contoh tampilan untuk hasil pengukuran ditunjukkan Gambar 4.



Gambar 4. Contoh tampilan hasil pengukuran

Aplikasi ini membaca data pengukuran listrik dari 10 alamat yang dinyatakan dalam format heksadesimal. Data di simpan pada masing-masing variabel berdasarkan alamat yang sudah di tentukan yaitu 0067 untuk daya aktif, 006B untuk daya reaktif, 0076 untuk faktor daya, 006F untuk daya semu. Program akan menjalankan operasi pengukuran dan menampilkan hasilnya setiap 2 detik. Setelah 5 menit (300 detik), program akan mengakhiri eksekusi. Penjelasan alamat yang digunakan ditunjukkan Tabel 1.

Tabel 1. Alamat program

Objek	Alamat (Hex)	Keterangan
Tegangan 3 fasa Tegangan 1 Tegangan 2 Tegangan 3 395.2 395.4 393.3	0x0078	Pengukuran tegangan tiap *line* fasa
Arus 3 Fasa Arus 1 Arus 2 Arus 3 1.04 1.1 1.1	0x0064	Pengukuran arus tiap *line* fasa
Harmonisa1 Harmonisa2 Harmonisa3 1.58 1.3 1.23	0x05DD	Pengukuran harmonisa tegangan tiap *line* fasa
Daya 163	0x006A	Pengukuran daya total

4.2 Perancangan Program *Python*

Perancangan perangkat lunak menggunakan bahasa pemrograman *python* versi 3.11.04 yang dijalankan dengan aplikasi Vscode (Visual Studio Code). Perancangan ini membutuhkan beberapa bantuan *Library* atau pustaka yang disediakan oleh *python* untuk membantu pemrograman agar sintaksnya lebih sederhana dan penulisan code tidak perlu diulang.

Library atau pustaka yang digunakan dalam pembuatan aplikasi pengukuran data yaitu:

1. *Library* pymodbus adalah pustaka yang digunakan untuk protokol komunikasi modbus dengan menggunakan komunikasi asinkronus.
2. *Library* pyserial adalah pustaka yang digunakan untuk koneksi serial dengan menggunakan kabel USB (Rs485) sebagai komunikasi serial data.
3. *Library* Tkinter adalah pustaka yang berfungsi sebagai pembuatan tampilan GUI.
4. Numpy yang digunakan sebagai pustaka *python* dengan fungsi sebagai array dalam operasi perhitungan, pengurutan, statistik dan lain lainnya.

4.3 Perancangan *ThingSpeak*

Data hasil pengukuran dari power meter dikirim secara *real-time* ke *platfor*m *ThingSpeak* melalui modul Wi-Fi NodeMCU ESP8266. Proses pengiriman data ini diprogram menggunakan Arduino IDE, lalu diunggah ke NodeMCU. Tahap awal untuk membuat program di NodeMCU, adalah menghubungkan perangkat NodeMCU ke komputer menggunakan perangkat lunak *Python*. Koneksi ini dilakukan melalui port COM7 pada komputer dengan kecepatan transfer data 115200 bit per detik seperti diperlihatkan Gambar 5.

```
ser = serial.Serial('COM7', 115200)
fmt = 'ffffffff' # fmt:format
```

Gambar 5. Koneksi port nodeMCU

Setelah berhasil menghubungkan NodeMCU ke komputer langkah berikutnya adalah mengintegrasikan keduanya ke dalam satu jaringan Wi-Fi yang sama. Dengan cara ini, data dapat dengan mudah dikirim dan diterima di antara kedua perangkat tersebut, seperti yang terlihat pada Gambar 6.

```
WiFiClient client;///koneksi internet
const char* ssid = "wZpqTx88";
const char* password = "ignaaja22";
```

Gambar 6. Koneksi NodeMCU ke internet

Arduino akan membaca data dari *power meter* dan mengirimkannya ke *ThingSpeak* sesuai dengan perintah yang telah diprogram. Program *Python* akan mengirimkan perintah ke Arduino melalui *port serial* untuk memverifikasi kesiapannya. Setelah mendapat konfirmasi, program *Python* akan mengirimkan data pengukuran dalam format string (teks) ke Arduino. Arduino menerima data ini dan melakukan tindakan yang sesuai berdasarkan perintah yang telah diprogram. Tampilan kode dapat dilihat pada Gambar 7.

```
"" if arduino.readline() == 1:
  arduino.write(bytes(str(tegangan[0]) + '\n', 'utf-8'))
  arduino.write(bytes(str(tegangan[1]) + '\n', 'utf-8'))
  arduino.write(bytes(str(tegangan[2]) + '\n', 'utf-8'))
  arduino.write(bytes(str(arus[0]) + '\n', 'utf-8'))
  arus.write(bytes(str(arus[1]) + '\n', 'utf-8'))
  arduino.write(bytes(str(arus[2]) + '\n', 'utf-8'))
  arduino.write(bytes(str(harmonisa[0]) + '\n', 'utf-8'))
  arduino.write(bytes(str(frequency[0]) + '\n', 'utf-8')) """
```

Gambar 7. Program alamat arduino

Program *Python* akan meminta data dari NodeMCU. Data yang diterima kemudian akan dikirim ke *website ThingSpeak*. Komunikasi antara komputer dan NodeMCU sudah siap untuk mengirim dan menerima data seperti yang ditunjukkan Gambar 8.

```
arduino = serial.Serial(port='COM7', baudrate=115200, timeout=.1)
```

Gambar 8. Koneksi serial arduino

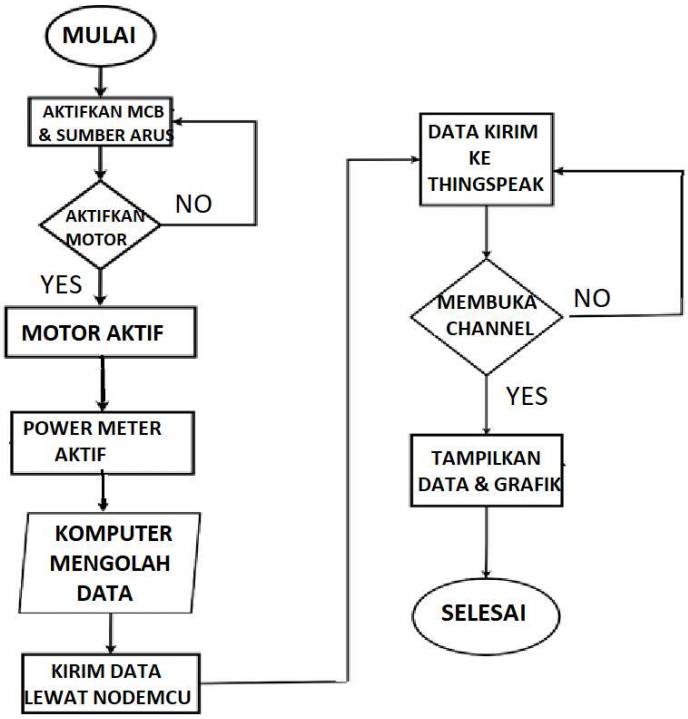
Untuk memvisualisasikan data pengukuran secara online, dibuat sebuah *channel* di *platform ThingSpeak*. *Channel* ini berfungsi sebagai wadah untuk menyimpan data yang dikirim dari Arduino. Proses pembuatan channel meliputi pemberian nama, deskripsi, dan penentuan sejumlah *field* yang akan digunakan untuk menyimpan data pengukuran. Setelah *channel* dibuat maka akan didapatkan *Channel ID* dan *Write* API *key* yang sangat penting untuk menghubungkan Arduino dengan *ThingSpeak* yang ditunjukkan Gambar 9.

```
myChannelNumber = 2185292;
myWriteAPIKey = "LSWHXDR4T3BCYRAY";
```

Gambar 9. *PI key*

Selanjutnya mengintegrasikan Channel ID dan Write API key yang telah didapatkan ke dalam kode program Arduino. Pastikan NodeMCU terhubung ke jaringan internet agar dapat mengirimkan data ke *endpoint* yang telah ditentukan di *ThingSpeak* kemudian jalankan program *Python* untuk memantau aktivitas pengiriman data dan memastikan semuanya berjalan lancar.

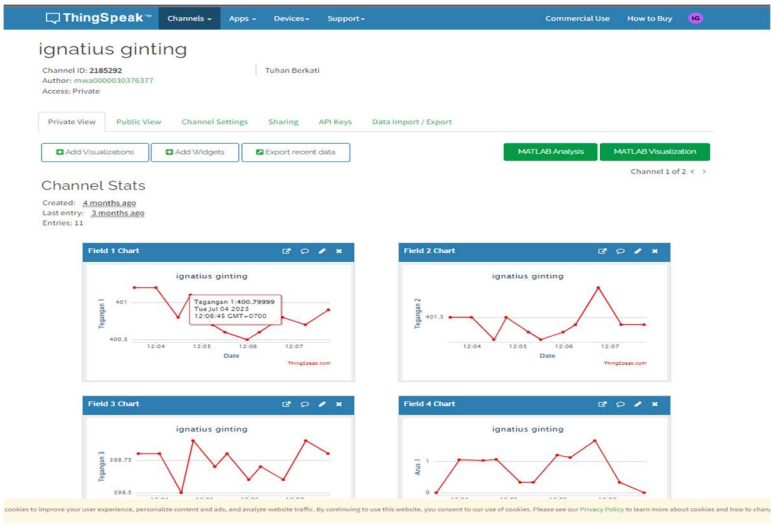
Diagram alir keseluruhan dari sistem yang dibuat diperlihatkan pada Gambar 10.



Gambar 10. Diagram alir keseluruhan sistem

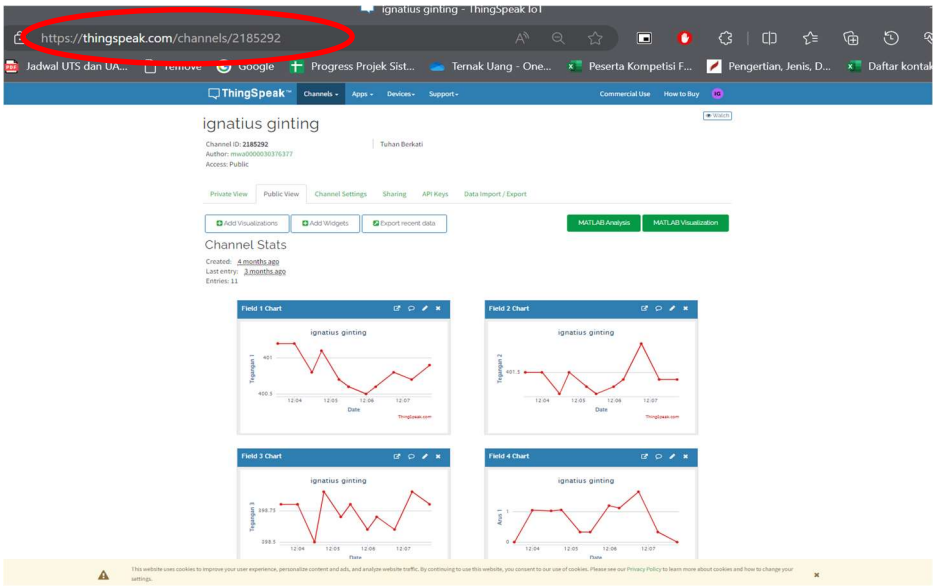
4.4 Pengujian Sistem

Pada pengujian ini, komunikasi perangkat lunak beroperasi di komputer dan mentransmisikan data. ESP8266 mengirimkan data melalui internet yang terhubung dengan perangkat lunak. Hasil pengukuran ditampilkan di halaman *ThingSpeak*, seperti yang terlihat pada Gambar 11.



Gambar 11. Tampilan Channel *ThingSpeak*

Pengukuran yang ditampilkan di *ThingSpeak* bersifat *real-time* dan diperbaharui setiap 5 menit dengan interval 1 menit antara setiap perubahan pengukuran. *ThingSpeak* juga dapat diakses melalui perangkat keras lain, seperti *smartphone* yang memungkinkan monitoring dari jarak jauh. Akses melalui *smartphone* dapat dicapai dengan berbagi URL (*Uniform Resource Locator*) sebagai suatu alamat sumber dari *ThingSpeak* seperti yang ditunjukkan dalam Gambar 12.



Gambar 12. URL dalam *ThingSpeak*

URL yang disalin ke *browser* pada perangkat keras seluler akan menampilkan grafik yang identik dengan tampilan di *ThingSpeak* pada komputer, seperti dalam Gambar 13.

Gambar 13. Tampilan pada *smartphone*

4.5 Pengujian Perbandingan Hasil Pengukuran Power Meter dengan Tampilan *ThingSpeak*

Pengujian dilakukan dengan memutar potensiometer pada modul *four quadrant*. Perubahan posisi potensiometer menyebabkan kecepatan motor dan arus listrik berfluktuasi, naik turun secara bergantian. Harmonisa juga menunjukkan pola yang sama. Namun, tegangan dan daya cenderung tetap stabil. Hasil pengukuran oleh power meter dapat dilihat pada Tabel 2 sedangkan tampilan hasil pengukuran pada *ThingSpeak* ditunjukkan Tabel 3.

Tabel 2. Hasil pengukuran power meter

No	V_1	V_2	V_3	I_1	I_2	I_3	Frequency	Daya aktif	Daya reaktif	Daya semua
1	397.700	397.899	396.100	0.00	0.00	0.000	49.950	0.0	0.0	0.0
2	397.899	398.200	396.399	0.00	0.00	0.000	50.030	0.0	0.0	0.0
3	397.799	398.100	396.200	0.00	0.00	0.000	50.040	0.0	0.0	0.0
4	397.500	397.899	396.100	1.06	1.11	1.09	50.040	53.0	282.0	142.0
5	397.500	397.899	396.100	1.06	1.11	1.09	50.040	63.0	713.0	731.0
6	397.600	398.000	396.200	1.06	1.11	1.10	50.040	60.0	761.0	781.0
7	397.899	398.200	396.200	1.07	1.11	1.10	50.030	68.0	730.0	756.0
8	397.700	398.100	396.100	1.07	1.11	1.09	50.020	61.0	785.0	834.0
9	397.500	397.899	395.899	1.06	1.11	1.09	49.990	63.0	747.0	764.0

Tabel 3 Hasil pengukuran yang ditampilkan pada *ThingSpeak*

created_at	entry_id	V_1	V_2	V_3	I_1	I_2	I_3	Harmonisa daya
2023-11-23T05:27:16+00:00	1	397.299	399.200	396.399	0.000	0.000	0.000	114.000
2023-11-23T05:27:54+00:00	2	397.600	399.600	396.899	0.000	0.000	0.000	144.000
2023-11-23T05:28:10+00:00	3	397.700	399.700	397.000	0.000	0.000	0.000	144.000
2023-11-23T05:28:40+00:00	4	397.700	399.700	396.899	1.080	1.120	1.120	140.000
2023-11-23T05:29:09+00:00	5	397.399	399.299	396.399	1.090	1.120	1.130	127.000
2023-11-23T05:29:24+00:00	6	397.700	399.500	396.899	1.130	1.160	1.150	149.000
2023-11-23T05:29:40+00:00	7	397.600	399.500	396.600	1.060	1.090	1.070	136.000
2023-11-23T05:30:09+00:00	8	397.399	399.299	396.399	1.090	1.130	1.120	144.000
2023-11-23T05:30:24+00:00	9	397.399	399.299	396.600	1.030	1.080	1.060	122.000

Dari Tabel 2 dan Tabel 3 dapat dilihat ada sedikit perbedaan nilai data yang ditampilkan *ThingSpeak* dengan nilai yang diukur *power meter*. Hal ini disebabkan adanya *delay* waktu pengiriman data untuk sampai ditampilkan pada *ThingSpeak*. Tabel 4 menunjukkan persentase kesalahan tersebut.

Tabel 4 Perbedaan Hasil Pengukuran data

No.	V_1	V_2	V_3	I_1	I_2	I_3	Harmonisa daya	Total Daya
1	0,10%	0,33%	0,08%	0	0	0	34,20%	0
2	0,08%	0,35%	0,13%	0	0	0	6,25%	0
3	0,02%	0,45%	0,20%	0	0	0	6,25%	0
4	0,05%	0,45%	0,20%	1,85%	0,89%	2,68%	9,28%	6,70%
5	0,03%	0,35%	0,08%	2,75%	0,89%	3,54%	6,29%	2,39%
6	0,03%	0,38%	0,18%	6,19%	4,31%	5,22%	9,38%	4,19%
7	0,08%	0,33%	0,10%	0,94%	1,83%	1,87%	12,50%	4,34%
8	0,08%	0,30%	0,08%	1,83%	1,77%	2,77%	12,50%	4,73%
9	0,03%	0,35%	0,18%	5,83%	4,63%	5,66%	7,37%	0,61%
Rata-rata error	0,04%	0,36%	0,13%	2,16%	2,39%	3,62%	11,56%	2,55%

Hasil pengukuran tegangan, arus, harmonisa, dan daya total memiliki tingkat keakuratan yang berbeda-beda, berkisar antara 0,04% hingga 11,56%. Perbedaan akurasi ini disebabkan oleh jeda waktu antara pengukuran satu dengan yang lainnya, yaitu sekitar 30 hingga 60 detik.

5. SIMPULAN

Hubungan komunikasi antara power meter dengan komputer, NodeMCU dan *ThingSpeak* dapat berjalan dengan baik hal ini dibuktikan dengan data dapat ditampilkan oleh *ThingSpeak*. Hasil pengukuran data bisa ditampilkan dengan kesalahan berkisar 0,04% - 11,56%. Hal ini karena adanya *delay* dari saat pengukuran sampai data ditampilkan di *ThingSpeak*.

REFERENSI

[1] PRIYOMBODO, "Target Konsumsi Listrik Masyarakat Tahun 2024 Meningkat," *www.kompas.id*, 2024.

[2] F. M. Farhan, E. Rosdiana, I. W. Fathonah, R. A. Sala, "Sistem Pemantauan Dan Pengendalian Daya Listrik secara Real Time Berbasis Mikrokontroler," Jurnal TEKTRIKA, vol. 7, no. 1, pp. 8-13, 2022.

[3] I. Grout, W. A. P. de Ferreira, and A. C. R. da Silva, "On-Line Electrical Supply Generation Fuel Mix Data Analysis using *Python* and TensorFlow," in *2019 International Conference on Power, Energy and Innovations (ICPEI)*, 2019, pp. 20–23, doi: 10.1109/ICPEI47862.2019.8944972.

[4] R. Lo, dkk, "Penggunaan Bahasa Pemrograman *Python* dalam Menganalisis Hubungan Kualitas Kopi dengan Lokasi Pertanian Kopi," Jurnal Publikasi Teknik Informatika, vol. 2, No.2, pp. 100-109, 2023.

[5] M. H. Kurnia, R. E. Saputra, and C. Setianingsih, "High-Low Detection of Sea Water Waves With Multi-Sensor System Based on Iot," e-Proceeding Eng., vol. 8, no. 5, pp. 6175–6182, 2021.

[6] L. Zhou *et al.*, "Development and design of a data acquisition device for cold standby power meter," in *2024 IEEE 2nd International Conference on Control, Electronics and Computer Technology (ICCECT)*, 2024, pp. 322–325, doi: 10.1109/ICCECT60629.2024.10545776.

[7] H. W., Pradana, S. I., Haryudo, "Model Power Metering Berbasis Haiwell Cloud Scada Untuk Efisiensi Energi Listrik," Jurnal Teknik Elektro, vol. 13, no. 1, pp. 10-20, 2023.

[8] A. B. P. Manullang, Y. Saragih, R. Hidayat, " Implementasi Nodemcu Esp8266 Dalam Rancang Bangun Sistem Keamanan Sepeda Motor Berbasis IOT," Jurnal Informatika & Rekayasa Elektronika, vol. 4, no. 2,. 163-170.

[9] Z. Song, Y. Wu, C. Tian, C. Liu, and W. Wang, "The Test Data of Multi-source Transmission and Transformation Equipment are Unified JSON Format File Generation," in *2021 International Conference on Advanced Electrical Equipment and Reliable Operation (AEERO)*, 2021, pp. 1–4, doi: 10.1109/AEERO52475.2021.9708143.

[10] R. Jeansoulin, "Data Served by JSONP," in *JavaScript and Open Data*, 2018, pp. 229–234.