

Perancangan *Plant Automation* untuk *Battery Monitoring System* Menggunakan Arduino Uno

Riza Hadi Saputra^{1*}, Kharis Sugiarto¹, Muhammad Faisal Lesmana¹

Program Studi Teknik Elektro, Jurusan Teknologi Industri dan Proses, Institut Teknologi Kalimantan
Jl. Soekarno-Hatta Km. 15, Karang Joang, Balikpapan, Kalimantan Timur
E-mail: riza.saputra@lecturer.itk.ac.id

Naskah Masuk: 09 Februari 2023; Diterima: 23 Agustus 2023; Terbit: 31 Maret 2024

ABSTRAK

Abstrak - BMS (*Battery Monitoring System*) merupakan contoh penerapan otomasi dalam pengawasan dan pemantauan baterai. Dalam perancangan otomasi pada BMS sederhana, terdapat empat parameter yang diawasi, yaitu tegangan, arus, temperatur, dan SOC (*State Of Charge*). Pada parameter tegangan, terdapat selisih antara nilai yang dihasilkan dengan multimeter dan nilai yang ditampilkan pada database, yakni sekitar 0.1 V hingga 0.35 V. Meskipun terdapat perbedaan tersebut, hasil analisis menggunakan Mean Squared Error (MSE) menunjukkan nilai sekitar 0.19, yang mengindikasikan bahwa alat yang dibuat dapat dianggap layak. Parameter arus listrik juga mengalami perbedaan antara nilai yang diukur dan nilai sebenarnya. Nilai arus negatif menunjukkan tidak adanya beban yang terhubung, sedangkan nilai arus positif menunjukkan adanya beban yang terhubung. Terdapat selisih antara nilai arus listrik yang diukur dan nilai sebenarnya sebesar 0.002 A hingga 0.02 A, dengan nilai MSE percobaan sebesar 2. Pada parameter SOC, terdapat kesulitan dalam penggunaan sintaks "map" yang mengakibatkan banyak kesalahan saat terjadi penurunan tegangan. Nilai selisih SOC berkisar antara 1% hingga 10%, dan menghasilkan nilai MSE sebesar 4.39. Terakhir, parameter temperatur juga menunjukkan selisih antara nilai pengukuran dan nilai sebenarnya sebesar 0.021⁰C hingga 0.645⁰C. Hasil analisis MSE akan menjadi dasar untuk mengevaluasi keberhasilan alat otomasi yang dirancang. Evaluasi ini penting dalam menentukan apakah alat tersebut memenuhi standar yang ditetapkan dan dapat diandalkan dalam pemantauan baterai. Dengan demikian, pemahaman terhadap hasil otomasi pada BMS sederhana ini memberikan kontribusi penting dalam pengembangan teknologi otomasi yang lebih efektif di masa depan.

Kata kunci: Otomasi, BMS, SOC, Arduino, Tegangan, Arus

ABSTRACT

Abstract - BMS (*Battery Monitoring System*) is an example of the application of automation in battery supervision and monitoring. In the design of automation in a simple BMS, there are four parameters that are monitored, namely voltage, current, temperature, and SOC (*State Of Charge*). In the voltage parameter, there is a difference between the value generated with a multimeter and the value displayed in the database, which is around 0.1 V to 0.35 V. Despite this difference, the analysis results using Mean Squared Error (MSE) show a value of around 0.19, which indicates that the tool made can be considered feasible. The electric current parameter also experiences a difference between the measured value and the actual value. A negative current value indicates no connected load, while a positive current value indicates a connected load. There is a difference between the measured and actual electric current values of 0.002 A to 0.02 A, with an experimental MSE value of 2. In the SOC parameter, there are difficulties in using the "map" syntax which results in many errors when a voltage drop occurs. The SOC difference values ranged from 1% to 10%, and resulted in an MSE value of 4.39. Finally, the temperature parameter also showed a difference between the measured and actual values of 0.021⁰C to 0.645⁰C. The results of the MSE analysis will be the basis for evaluating the success of the designed automation tool. This evaluation is important in determining whether the tool meets the set standards and is reliable in battery monitoring. Thus, understanding the automation results on this simple BMS makes an important contribution to the development of more effective automation technologies in the future.

Keywords: Automation, BMS, SOC, Arduino, Voltage, Arus

Copyright © 2024 Universitas Muhammadiyah Jember.

1. PENDAHULUAN

Otomasi adalah teknik yang digunakan untuk menggantikan tenaga manusia dalam pengawasan dan menjalankan perangkat, proses, sistem, atau status secara otomatis [1][2]. Hal ini melibatkan penerapan teknologi mekanik, elektronik, dan sistem berbasis komputer seperti PC, PLC, dan mikrokontroler [3]. Tujuan utama dari otomasi adalah memberikan kemudahan dan efisiensi dalam melakukan pekerjaan. Beberapa contoh otomasi sudah banyak sekali digunakan, yang salah satu contohnya yaitu sistem

monitoring baterai pada suatu rangkaian baterai yang dibuat [4]. Tujuan adanya sistem monitoring baterai pada rangkaian baterai yaitu pengoptimalan pengisian dan pengosongan baterai yang mana dapat mengoptimalkan proses pengisian dan pengosongan baterai dengan memperhatikan arus dan tegangan pengisian secara akurat [5][6]. Hal ini juga membantu menjaga kesehatan baterai dan memperpanjang masa pakainya. Sistem otomatis dapat secara otomatis mengatur pengisian baterai sesuai dengan karakteristik baterai yang sedang digunakan, sehingga menghindari pengisian berlebihan atau kurang [7].

Namun, selain hal tersebut, kerentanan terhadap kesalahan sistem otomatis yang kompleks dapat rentan terhadap kesalahan atau kegagalan teknis [8]. Jika sistem otomasi yang mengelola baterai mengalami kerusakan atau mengalami kesalahan dalam mengendalikan pengisian atau pemakaian daya, hal ini dapat menyebabkan kerusakan pada baterai itu sendiri [9]. Dalam kasus seperti ini, diperlukan pemantauan dan pemeliharaan yang cermat untuk memastikan sistem otomasi berfungsi dengan baik dan menghindari potensi kerusakan baterai yang disebabkan oleh kesalahan sistem. Sistem monitoring baterai yang diotomatisasi memiliki beberapa permasalahan yang perlu diperhatikan. Salah satunya adalah pengawasan suhu. Baterai yang beroperasi pada suhu yang terlalu tinggi atau terlalu rendah dapat mengalami degradasi kinerja dan umur baterai yang lebih pendek [10]. Oleh karena itu, sistem otomatis harus mampu memonitor suhu baterai secara terus-menerus dan mengambil tindakan yang tepat untuk menjaga suhu dalam batas yang aman. Permasalahan lainnya adalah pengoptimalan pengisian baterai. Baterai yang diisi dengan arus yang terlalu tinggi atau tegangan yang tidak tepat dapat menyebabkan kerusakan pada baterai dan mengurangi masa pakainya. Sistem otomatis harus dapat mengatur arus dan tegangan pengisian baterai dengan cerdas, sesuai dengan karakteristik baterai yang sedang digunakan [11]. Selain itu, pengawasan penggunaan daya juga menjadi permasalahan dalam otomasi sistem monitoring baterai. Sistem otomatis harus dapat mengelola penggunaan daya yang efisien, mengalokasikan daya dengan bijaksana berdasarkan kebutuhan, dan menghindari pemborosan energi. Semua permasalahan ini memerlukan desain sistem otomasi yang cermat dan algoritma yang tepat untuk mengoptimalkan penggunaan dan umur baterai.

Oleh karenanya, dalam penelitian terkait rancangan baterai, dibuat sistem monitoring baterai berbasis mikrokontroler (Arduino Uno) sebagai sistem otomasi yang dapat bekerja sesuai dengan program yang telah dibuat dengan empat parameter yaitu tegangan, arus, temperatur, dan SOC (State Of Charge). Diharapkan dari penelitian ini dapat dikembangkan lebih lanjut terkait sistem monitoring baterai dan dapat mendukung sistem pada rangkaian baterai agar dapat memperpanjang umur baterai sesuai dengan beban yang diberikan.

2. METODE PENELITIAN

Dalam percobaan ini yang dapat dilihat pada Gambar 1, akan dilakukan perancangan sistem monitoring baterai pada 3 sel baterai tipe lithium-ion dengan masing-masing spesifikasi 3.7 V dan 2200 mAh. Sistem ini bertujuan untuk mengawasi kondisi baterai secara efektif. Alat kendali yang digunakan adalah Arduino tipe Uno, yang akan bertindak sebagai otak sistem dan mengambil data dari sensor-sensor yang terpasang [12]. Metode penelitian yang dapat dilakukan dalam penelitian ini adalah sebagai berikut:

- Perancangan dan konfigurasi sistem: Meliputi perancangan dan konfigurasi perangkat keras dan perangkat lunak yang diperlukan untuk membangun sistem monitoring baterai. Hal ini melibatkan menghubungkan baterai lithium-ion, Arduino Uno, sensor-sensor, dan komponen lainnya sesuai dengan kebutuhan.
- Pengukuran karakteristik baterai: Dilakukan pengukuran dan analisis karakteristik baterai, seperti tegangan, arus, dan temperatur. Hal ini bertujuan untuk mengetahui kondisi baterai secara real-time dan memberikan informasi penting dalam pengelolaan baterai.
- Monitoring arus: Dengan menggunakan sensor arus ACS712-30, dapat dilakukan pengukuran arus yang mengalir melalui baterai. Data arus ini berguna dalam mengawasi pemakaian baterai dan mendeteksi situasi yang tidak normal, seperti arus berlebih atau arus yang tidak stabil.
- Pemantauan suhu: Sensor suhu LM35 dapat digunakan untuk mengukur suhu di sekitar baterai. Pemantauan suhu ini penting karena suhu yang tinggi dapat berdampak negatif pada kinerja dan umur pakai baterai. Data suhu yang terkumpul dapat digunakan untuk mengontrol kondisi operasional dan memberikan peringatan jika suhu mencapai batas yang tidak aman.
- Pengolahan data: Arduino Uno akan berperan dalam pengolahan data dari sensor-sensor, seperti penghitungan daya, pemantauan tegangan, arus, dan suhu. Dengan menggunakan program yang sesuai, Arduino Uno dapat mengawasi pengisian dan pengosongan baterai sesuai dengan kondisi yang diukur.
- Evaluasi performa sistem: Setelah sistem monitoring baterai selesai dirancang dan diimplementasikan, dilakukan evaluasi performa sistem. Hal ini melibatkan pengujian sistem dalam berbagai kondisi operasional dan pembebanan baterai yang telah ditentukan.

Dengan metode penelitian di atas, penelitian ini akan memberikan pemahaman yang lebih baik tentang pengelolaan baterai lithium-ion dengan menggunakan sistem monitoring baterai yang dirancang, serta memberikan informasi yang berguna untuk meningkatkan kinerja dan umur pakai baterai tersebut.

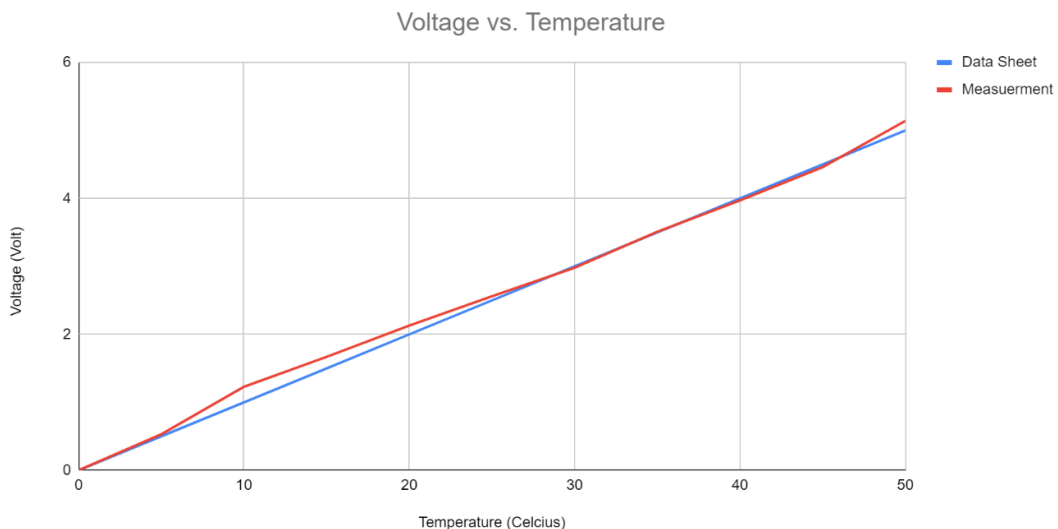
2.3. PENGGUNAAN SENSOR TEMPERATUR LM35

Pada sensor temperatur LM35 yang terlihat pada Gambar 3, terdapat 3 pin yang mempunyai fungsi masing-masing. Pin 1, yang paling kiri, adalah pin Vcc yang dihubungkan pada Arduino dengan tegangan 5V. pin 2, yang ditengah, adalah keluaran tegangan yang akan dibaca pada Arduino dan dihubungkan di Arduino pada pin analog input. Sedangkan pin 3, yang paling kanan, adalah ground yang akan dihubungkan ke pin GND di Arduino. Sensor temperatur LM35 digunakan dalam sistem pemantauan baterai (battery monitoring system) karena memiliki beberapa alasan yang membenarkan penggunaannya. Pertama, LM35 merupakan sensor yang presisi dalam mengukur suhu dengan akurasi yang tinggi. Hal ini penting dalam pemantauan baterai karena suhu yang tinggi dapat mengindikasikan adanya masalah atau kondisi yang tidak normal dalam baterai, seperti overcharging atau overdischarging. Dengan menggunakan sensor temperatur yang presisi seperti LM35, kita dapat mendapatkan informasi yang akurat mengenai suhu baterai. Alasan kedua adalah kemudahan penggunaan LM35 dalam integrasi dengan mikrokontroler seperti Arduino. Sensor LM35 memiliki tiga pin yang telah dijelaskan sebelumnya, dan kompatibilitasnya dengan Arduino membuatnya dapat dengan mudah dihubungkan dan diintegrasikan dengan sistem pemantauan yang sudah ada. Pin 1 (Vcc) dapat dihubungkan langsung ke pin 5V pada Arduino, pin 2 (keluaran tegangan) dapat dihubungkan ke pin analog input pada Arduino, dan pin 3 (ground) dihubungkan ke pin GND pada Arduino. Dengan adanya kemudahan penggunaan ini, implementasi LM35 pada sistem pemantauan baterai menjadi lebih efisien dan dapat dilakukan dengan cepat. Selain itu, LM35 juga memiliki rentang suhu yang luas, yaitu -55°C hingga 150°C . Rentang suhu yang luas ini memungkinkan penggunaan sensor ini pada berbagai jenis baterai yang memiliki suhu operasional yang berbeda. Dengan kemampuan mengukur suhu pada rentang yang luas, sistem pemantauan dapat mendeteksi perubahan suhu yang signifikan dan mengambil tindakan yang sesuai untuk menjaga keandalan dan kinerja baterai. Dalam kesimpulannya, penggunaan sensor temperatur LM35 pada sistem pemantauan baterai menjadi pilihan yang tepat karena presisi pengukuran suhu yang tinggi, kemudahan integrasi dengan mikrokontroler seperti Arduino, dan rentang suhu yang luas. Dengan memantau suhu baterai secara akurat, sistem pemantauan dapat mendeteksi potensi masalah dan mengambil langkah-langkah yang diperlukan untuk menjaga keamanan dan kinerja baterai. Berikut pada Gambar 4 adalah pengujian data sensor temperatur LM35.



Gambar 3. Sensor temperature LM35

(Source: https://miro.medium.com/v2/resize:fit:352/0*rZ8sLFOFU-N4Okxp)

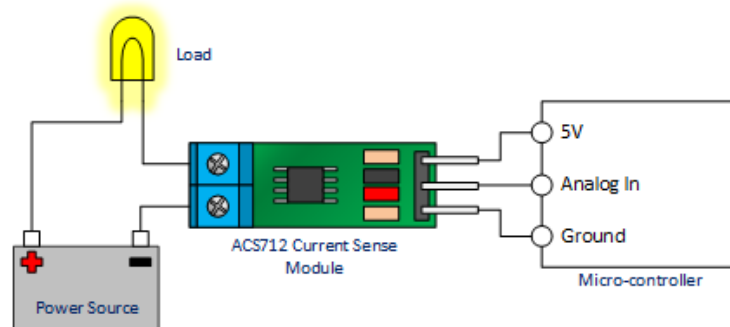


Gambar 4. Pengujian Data Sensor Temperatur LM35

2.4. PENGGUNAAN SENSOR ARUS ACS712-30

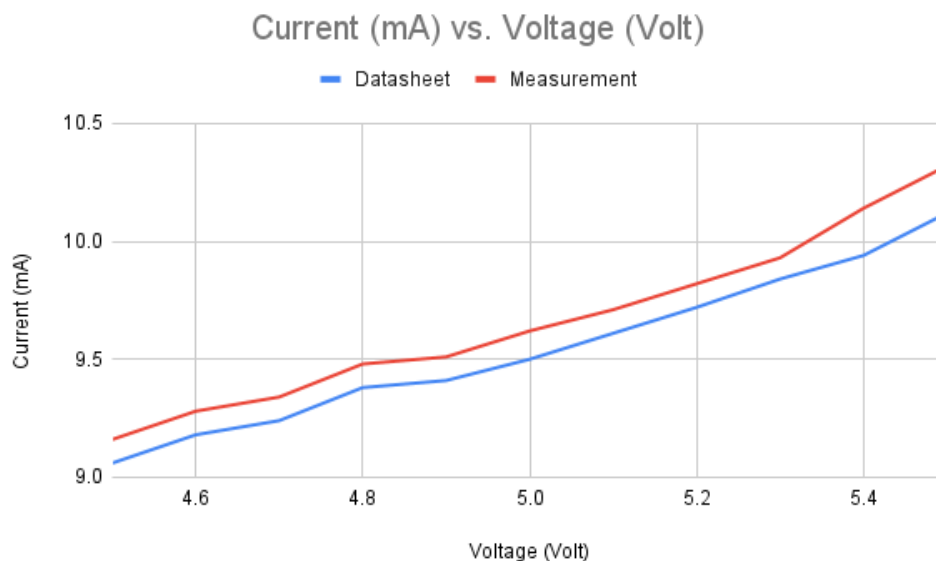
Sensor arus listrik ACS712-30 digunakan dalam sistem pemantauan baterai (*battery monitoring system*) dengan alasan yang tepat dan relevan. Sensor ini memiliki struktur yang mirip dengan sensor temperatur LM35, yaitu terdiri dari tiga pin: Vcc, analog input, dan ground. Pin 1 (Vcc) dihubungkan ke pin 5V pada Arduino, pin 2 (keluaran tegangan) dihubungkan ke pin analog input pada Arduino, dan pin 3 (ground) dihubungkan ke pin GND pada Arduino yang terlihat pada Gambar 5. Perbedaan utama dari sensor ACS712-30 terletak pada objek yang akan dideteksinya. Sensor ini digunakan untuk mengukur arus listrik yang mengalir melalui kabel negatif pada beban dalam sistem pemantauan baterai. Dengan demikian, sensor ACS712-30 mampu memberikan informasi mengenai arus listrik yang masuk atau keluar dari baterai, yang sangat penting dalam pemantauan keadaan baterai. Penggunaan sensor ACS712-30 dalam *battery monitoring system* memiliki beberapa keunggulan. Pertama, sensor ini mampu mengukur arus listrik dengan presisi yang tinggi. Hal ini memungkinkan sistem pemantauan untuk mendapatkan informasi yang akurat mengenai arus yang mengalir pada baterai. Dengan demikian, dapat terdeteksi apakah arus listrik berada dalam rentang normal atau terjadi kelebihan arus atau arus yang terlalu rendah. Selain itu, sensor ACS712-30 memiliki rentang pengukuran yang luas. Dalam kasus ini, model ACS712-30 memiliki kemampuan untuk mengukur arus hingga 30 Ampere. Rentang pengukuran yang luas ini memungkinkan penggunaan sensor ini pada berbagai jenis baterai yang mungkin memiliki kebutuhan arus yang berbeda. Berikut pada Gambar 6 adalah pengujian data sensor arus ACS712-30.

Dalam kesimpulannya, penggunaan sensor arus listrik ACS712-30 dalam sistem pemantauan baterai merupakan pilihan yang tepat karena kemampuan presisi pengukuran arus listrik, rentang pengukuran yang luas, dan kemudahan integrasinya dengan Arduino. Dengan menggunakan sensor ini, sistem pemantauan dapat memantau dan mengendalikan arus listrik yang masuk atau keluar dari baterai dengan akurat, sehingga memastikan kinerja dan keamanan baterai yang optimal.



Gambar 5. Struktur sensor ACS712

(Source: <https://www.androiderode.com/wp-content/uploads/2018/09/ACS712-testing.png>)



Gambar 6. Pengujian Data Sensor Arus ACS712-30

2.5. SOC DAN FUNGSI MAP

State of Charge (SOC), adalah parameter yang mengukur jumlah energi yang tersimpan dalam baterai pada suatu waktu tertentu. Untuk baterai lithium-ion, perhitungan SOC didasarkan pada hubungan non-linear antara tegangan baterai dan tingkat pengisian. Pada umumnya, perhitungan SOC dilakukan dengan menggunakan metode penggabungan antara tegangan dan kurva pembongkaran kapasitas baterai. Berikut pada Persamaan 1 adalah persamaan yang digunakan.

$$SOC (\%) = \left(\frac{V - V_{min}}{V_{max} - V_{min}} \right) * 100 \quad (1)$$

dimana, V adalah tegangan baterai yang terukur. V_{min} adalah tegangan minimum yang terkait dengan SOC 0%, dan V_{max} adalah tegangan maksimum yang terkait dengan SOC 100%.

SOC sangat penting dalam aplikasi yang menggunakan baterai, termasuk dalam fungsi “map” Arduino. Pemahaman SOC membantu mengelola penggunaan energi baterai dengan efisien dan mencegah baterai dari pengisian atau pengosongan berlebihan yang dapat merusaknya. Dalam fungsi “map” Arduino, SOC dapat diukur menggunakan sensor tegangan baterai atau dengan menggunakan metode perhitungan berdasarkan tegangan baterai yang terbaca. Umumnya, tegangan baterai terkait dengan SOC, yang berarti semakin tinggi tegangan baterai, semakin tinggi pula SOC, dan sebaliknya. Berikut adalah langkah-langkah untuk mengukur SOC pada fungsi “map” menggunakan tegangan baterai:

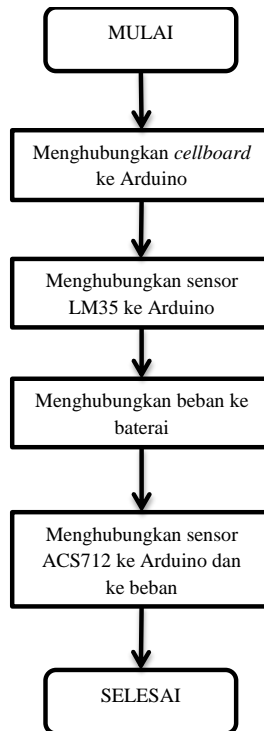
- Menentukan tegangan baterai minimum dan maksimum yang diukur oleh Arduino Uno;
- Membaca tegangan baterai menggunakan pin analog Arduino yang terhubung ke terminal positif baterai. Nilai tegangan tersebut dapat dikonversi menjadi skala 0-1023 dengan menggunakan fungsi “analogRead()” pada Arduino Uno;
- Mengubah tegangan menjadi nilai SOC dengan cara membuat pemetaan antara tegangan baterai yang terukur dengan nilai SOC yang sesuai. Fungsi “map()” pada Arduino dapat digunakan untuk melakukan pemetaan. Sebagai contoh, jika tegangan baterai minimum adalah 3 V dan tegangan baterai maksimum adalah 5 V, maka tegangan tersebut dapat dikonversi menjadi persentase SOC (0-100%). Berikut adalah contoh fungsi “map()” yang digunakan:

```
int batteryVoltage = analogRead(A0); // Baca tegangan baterai
int soc = map(batteryVoltage, 0, 1023, 0, 100); // Ubah tegangan menjadi SOC (0 – 100%)
```

2.6. METODOLOGI PERCOBAAN

Penjelasan pada Gambar 7 dimulai dengan menghubungkan rangkaian *cellboard* ke Arduino yang telah ditentukan letak pinnya [17]. Berdasarkan percobaan dari Maharmi, dkk., [18] *Cellboard* disini berfungsi untuk mengambil data tegangan dari baterai yang telah dipasang dan juga tegangan ini sangat diperlukan sebagai referensi sensor LM35 dan ACS712-30. Setelah menghubungkan *cellboard*, selanjutnya menghubungkan sensor LM35 ke Arduino untuk mendeteksi temperatur. Setelah menghubungkan sensor LM35, selanjutnya menghubungkan beban terlebih dahulu berupa lampu LED 12VDC,3W sebelum menghubungkan sensor ACS712-30. Setelah beban dicoba dan menyala, selanjutnya menghubungkan sensor ACS712-30 diantara beban dan baterai agar terdeteksi arus yang dilaluinya.

Menjelaskan pada Gambar 7 dimulai dengan menghubungkan rangkaian *cellboard* ke Arduino yang telah ditentukan letak pinnya. *Cellboard* disini berfungsi untuk mengambil data tegangan dari baterai yang telah dipasang dan juga tegangan ini sangat diperlukan sebagai referensi sensor LM35 dan ACS712-30. Setelah menghubungkan *cellboard*, selanjutnya menghubungkan sensor LM35 ke Arduino untuk mendeteksi temperatur. Setelah menghubungkan sensor LM35, selanjutnya menghubungkan beban terlebih dahulu berupa lampu LED 12VDC,3W sebelum menghubungkan sensor ACS712-30. Setelah beban dicoba dan menyala, selanjutnya menghubungkan sensor ACS712-30 diantara beban dan baterai agar terdeteksi arus yang lewat disana [19].



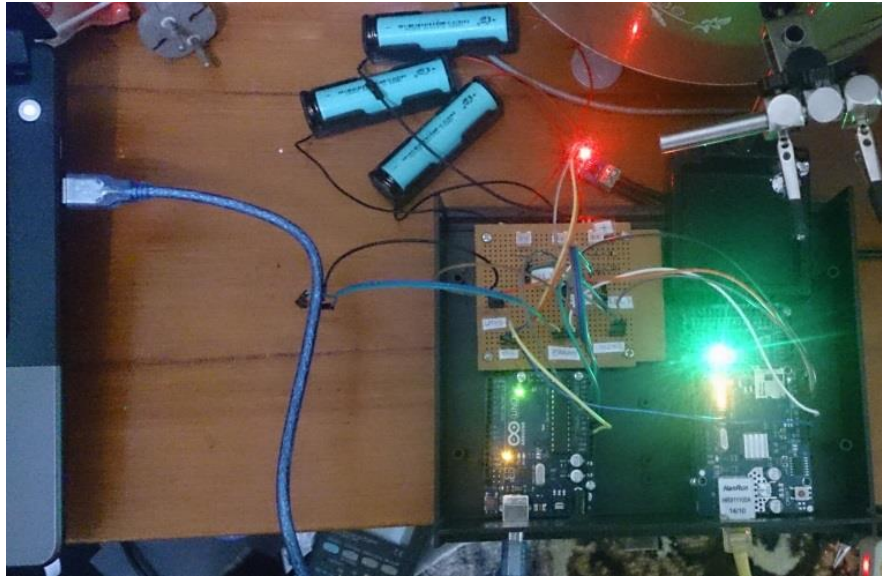
Gambar 7. Metodologi percobaan otomasi pada *battery monitoring system*

3. HASIL DAN PEMBAHASAN

Setelah merancang dan menjalankan sistemnya, maka langkah selanjutnya adalah melihat data yang telah diambil oleh Arduino. Yang pertama ditampilkan adalah data tegangan dari masing-masing baterai dan total tegangan baterai. Data tegangan ini nantinya akan digunakan untuk menentukan kapasitas yang tersisa dalam baterai atau SOC. Selain itu juga, data tegangan termasuk data yang harus ada terlebih dahulu sebelum data arus dan temperatur dikarenakan semua perhitungan nilai tersebut mengacu pada tegangan. Berikut pada Gambar 8 adalah data yang diperoleh setelah dihubungkan dengan database sedangkan Gambar 9 adalah BMS yang telah dibuat.

time	id	device_id	voltage_1	voltage_2	voltage_3	voltage_pack	SOC	current	temperature
2015-05-12 23:44:04	0	1	4.115	4.119	4.152	12.385	72	0.074	18.555
2015-05-12 23:44:01	0	1	4.115	4.119	4.142	12.376	72	0.074	18.555
2015-05-12 23:43:59	0	1	4.115	4.119	4.147	12.381	72	0.074	19.019
2015-05-12 23:43:56	0	1	4.115	4.119	4.147	12.381	72	0.074	18.555
2015-05-12 23:43:53	0	1	4.115	4.119	4.142	12.376	72	0	19.482
2015-05-12 23:43:51	0	1	4.115	4.124	4.147	12.385	72	0	19.019
2015-05-12 23:43:48	0	1	4.115	4.119	4.147	12.381	72	0	18.555
2015-05-12 23:43:46	0	1	4.119	4.124	4.147	12.39	72	0.074	18.555
2015-05-12 23:43:43	0	1	4.115	4.119	4.142	12.376	72	0.074	18.555
2015-05-12 23:43:41	0	1	4.115	4.119	4.147	12.381	72	0.074	19.019
2015-05-12 23:43:38	0	1	4.119	4.124	4.152	12.395	72	0.074	18.555
2015-05-12 23:43:36	0	1	4.119	4.124	4.152	12.395	72	0.074	19.019
2015-05-12 23:43:33	0	1	4.119	4.124	4.156	12.399	72	0.074	19.019
2015-05-12 23:43:31	0	1	4.119	4.124	4.152	12.395	72	0	19.019
2015-05-12 23:43:28	0	1	4.119	4.124	4.152	12.395	72	0.074	18.555
2015-05-12 23:43:26	0	1	4.119	4.124	4.152	12.395	72	0.074	19.019
2015-05-12 23:43:23	0	1	4.156	4.161	4.193	12.51	90	-0.148	19.019
2015-05-12 23:43:21	0	1	4.152	4.156	4.184	12.492	81	-0.074	18.091
2015-05-12 23:43:18	0	1	4.156	4.161	4.184	12.501	90	-0.074	19.019
2015-05-12 23:43:16	0	1	4.156	4.161	4.184	12.501	90	-0.148	18.091
2015-05-12 23:43:13	0	1	4.156	4.161	4.193	12.51	90	-0.074	19.019
2015-05-12 23:43:10	0	1	4.161	4.166	4.193	12.52	90	-0.148	19.482
2015-05-12 23:43:08	0	1	4.161	4.166	4.193	12.52	90	-0.074	19.019
2015-05-12 23:43:05	0	1	4.161	4.166	4.193	12.52	90	-0.148	19.019
2015-05-12 23:43:03	0	1	4.156	4.166	4.189	12.51	90	-0.074	19.019
2015-05-12 23:43:00	0	1	4.161	4.166	4.193	12.52	90	-0.074	18.555

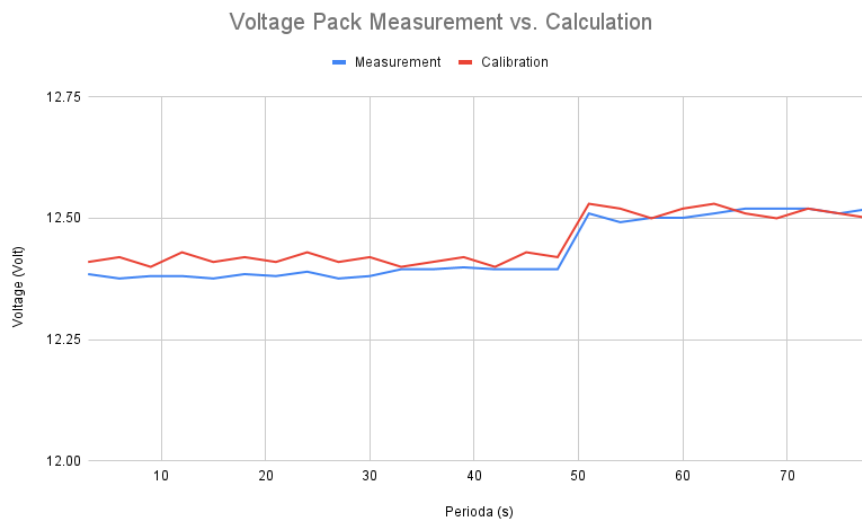
Gambar 8. Tampilan data yang ditampilkan pada database



Gambar 9. BMS yang telah dibuat

3.1. HASIL PENGAMBILAN DATA TEGANGAN

Dapat dilihat pada Gambar 8 bahwa tegangan masing-masing baterai terbaca hampir sama, sekitar 4.173 volt. Namun setelah dikalibrasi dengan multimeter, ada tegangan berlebih sedikit yang ditampilkan di database. Di database terlihat 4.173 volt, sedangkan saat dikalibrasi dengan multimeter, tegangan baterai adalah 4.201 volt. Hal ini disebabkan oleh pemilihan tegangan referensi yang digunakan di Arduino. Pada kolom selanjutnya, ada total tegangan yaitu 12.52 volt yang didapatkan dari penjumlahan tegangan ketiga sel baterai. Untuk lebih jelasnya, dapat dilihat pada Gambar 10 dengan nilai MSE (Mean Square Error) yang didapatkan yaitu 0.19.

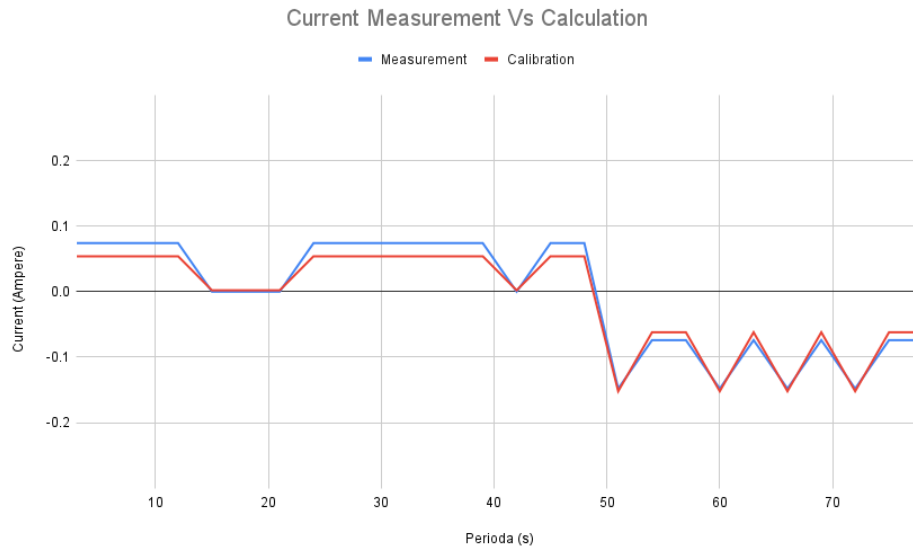


Gambar 10. Perbandingan Nilai Tegangan Total dengan Kalibrasi

3.2. HASIL PENGAMBILAN DATA ARUS

Selanjutnya adalah data arus yang didapatkan dari sensor ACS712-30. Data arus ini didapatkan dari perhitungan data dari tegangan yaitu tegangan dikurangi dengan *offset* lalu dibagi dengan 66mV untuk sensor ACS712-30. Dapat dilihat pada Gambar 8 bahwa arus yang terdeteksi ada dua jenis, positif dan negatif. Jika nilai arus positif artinya ada beban yang tersambung dengan baterai atau lampu LED menyala. Jika arus bernilai negatif, maka tidak ada beban yang tersambung dengan baterai atau lampu LED padam. Karena lampu yang digunakan berspesifikasi rendah yaitu 12VDC, 3W, maka arus yang dibutuhkan juga kecil. Ini alasan utama mengapa arus positif saat dihubungkan beban bernilai 0.074 atau

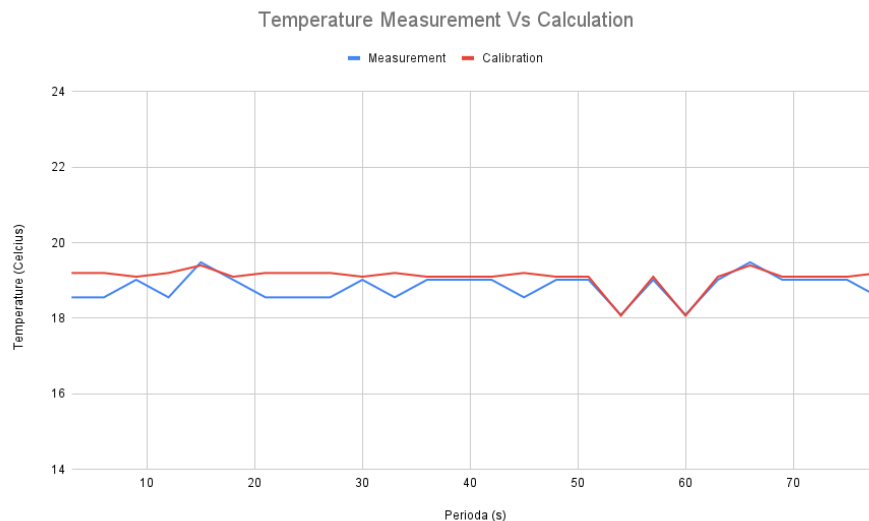
hampir mendekati 0. Berikut pada Gambar 11 yaitu perbandingan nilai arus listrik yang dihasilkan dari BMS yang dibandingkan dengan kalibrasi atau multimeter dengan nilai MSE 2.



Gambar 11. Perbandingan Nilai Arus Listrik dengan Kalibrasi

3.3. HASIL PENGAMBILAN DATA TEMPERATUR

Data temperatur juga berdasarkan data tegangan. Sensor LM35 mengubah satuan tegangan menjadi derajat celcius adalah 10C/10mV. Untuk pemrograman di Arduinonya yaitu tegangan dikalikan tegangan referensi dikalikan dengan nilai maksimal dari temperatur yang akan dideteksi. Untuk data yang didapatkan dan dibandingkan dengan alat ukur, dapat dilihat pada Gambar 12. Pada Gambar 12, didapatkan nilai MSE 2.



Gambar 12. Perbandingan Nilai Temperatur dengan Kalibrasi

3.4. HASIL PENGOLAHAN DATA SOC

Data yang diambil selanjutnya yaitu SOC atau kapasitas pada baterai. SOC digunakan sebagai kapasitas indikator pada baterai yang sedang diawasi. Dalam percobaan ini, SOC yang dihitung adalah paket atau total dari 3 baterai yang telah disusun secara seri dan data ditampilkan pada Gambar 8. Dapat dilihat dari data diatas bahwa SOC semakin menurun ketika tegangan menurun. Pada percobaan ini untuk estimasi SOC tidak digunakan metode apapun. Hanya syntax program dari Arduino yaitu “map”. “map” adalah salah satu syntax di dalam Arduino yang berfungsi merubah satu nilai besaran ke besaran lainnya

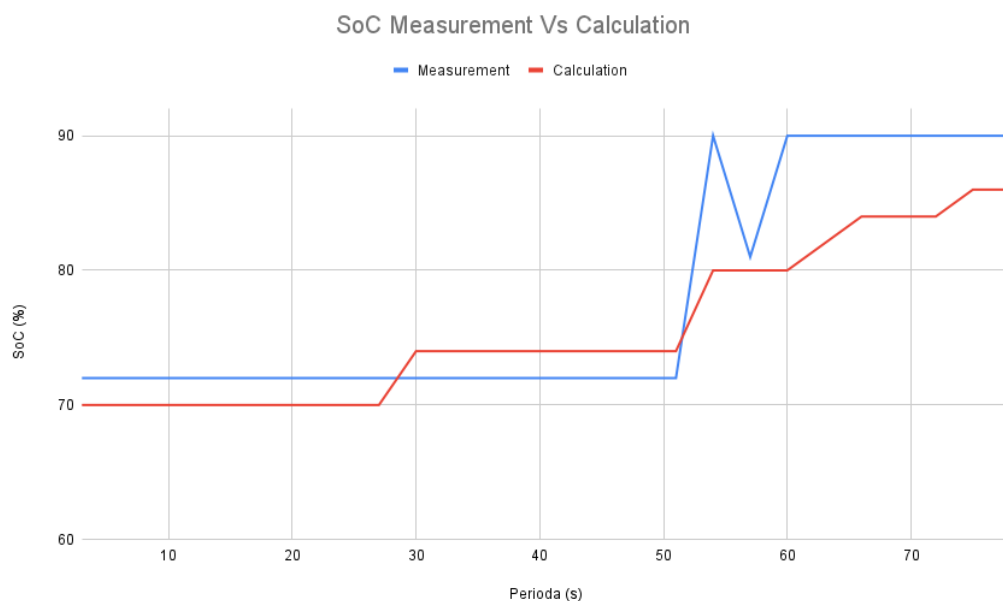
dalam bentuk range. Syntax ini bisa dimanfaatkan untuk menentukan kondisi baterai dalam persen dari nilai tegangan atau arus yang ada pada baterai. Misalnya dalam kasus ini kita merubah nilai tegangan (menurut referensi) ke dalam SOC (dalam persen).

$$SOC = \text{map}(\text{tegangan}, \text{mintegref}, \text{maxtegregf}, \text{minsoc}, \text{maxsoc});$$

Program diatas menunjukkan nilai tegangan yang akan dirubah ke dalam nilai SOC. Tegangan didapatkan dari jumlah tegangan pada baterai yaitu sekitar 12VDC secara seri. *Min_teg_ref* adalah nilai minimal total tegangan saat semua baterai dalam keadaan kosong sedangkan untuk *max_teg_ref* adalah nilai maksimum total tegangan saat semua baterai dalam keadaan penuh. Untuk syntax *min_soc* dan *max_soc* adalah nilai SOC yang akan mengonversi dari nilai tegangan yang terdeteksi. Berikut adalah contoh nilai SOC pada tegangan 12VDC.

```
float data6 = 0;
data6 = map((data5 * 10), 115, 126, 0, 100);
```

Data6 adalah nilai SOC yang akan dicari. Sedangkan data5 adalah nilai tegangan yang akan berubah-ubah sesuai kondisi baterai saat itu. 115 adalah nilai minimal tegangan referensi. 126 adalah nilai maksimal tegangan referensi. 0 adalah nilai minimal SOC. 100 adalah nilai maksimal SOC. Mengapa dikali 10? Nilai asli dari 115 adalah 11.5 volt dan nilai asli 126 adalah 12.6 volt. Jika yang direferensikan adalah 11.5 dan 12.6, maka range nilai atau span tersebut sangatlah kecil dibandingkan dengan 115 dan 126. Hal ini berguna memperkecil error pada skala SOC 0-100. Berikut pada Gambar 13 yaitu perbandingan nilai SOC yang dihasilkan dari BMS yang dibandingkan dengan perhitungan dengan nilai MSE 4.39.



Gambar 13. Perbandingan Nilai SOC dengan Perhitungan

4. KESIMPULAN

Kesimpulan yang didapatkan dari penelitian ini yaitu telah dibuat perancangan otomasi pada BMS sederhana untuk mendeteksi empat parameter yaitu tegangan, arus, temperatur, dan SOC. Yang pertama pada nilai tegangan yang ditampilkan pada database tidak sama dengan nilai yang dikalibrasi menggunakan multimeter dikarenakan ada selisih sekitar 0.1 – 0.35 V yang terjadi karena beberapa faktor. Untuk hasil nilai MSE pada Tegangan Total sekitar 0.19 yang mana hasil ini dianggap layak untuk alat yang telah dibuat. Yang kedua yaitu arus listrik yang mana jika nilai arus bernilai negatif, maka tidak ada beban yang terhubung, sedangkan jika arus bernilai positif, maka ada beban yang terhubung. Selisih pada nilai arus listrik berkisar antara 0.002 – 0.02 A dengan nilai MSE pada percobaan arus listrik yaitu 2. Yang ketiga untuk hasil SOC dengan menggunakan syntax “map” banyak mengalami *error* setiap penurunan tegangan. Nilai selisih untuk SOC yaitu berkisar 1 – 10% yang mana menghasilkan nilai MSE sebesar 4.39. yang keempat yaitu nilai temperatur yang mengalami selisih dengan pengukuran sebesar 0.021 – 0.645°C. hasil dari nilai MSE untuk temperatur sebesar 2. Oleh karenanya, hasil dari seluruh parameter untuk nilai MSE dibawah 5, maka alat BMS

dapat dikatakan berhasil dan mendekati dengan nilai aslinya. Adapun saran – saran yang dapat dilakukan yaitu menggunakan pin Aref pada Arduino sebagai tegangan referensi agar tegangan yang ditampilkan dan yang dikalibrasi bernilai sama serta menghitung SOC harus menggunakan metode agar nilai yang ditampilkan tidak mengalami error setiap penurunan tegangan.

REFERENSI

- [1] Risfendra and H. Setiawan, “Otomasi Industri Dengan Arduino Outrseal PLC,” *UNP Press*, 2020.
- [2] R. N. Indah, R. Z. A. Syam, and U. Aulia, “Dampak Perubahan Sistem Otomasi Slims Ke Inlislite Di Perpustakaan SMK Negeri 9 Bandung,” *Tibanndaru J. Ilmu Perpust. dan Inf.*, vol. 5, no. 1, pp. 148-158, 2021.
- [3] R. Abdul, “Studi Kasus Kesiapan Pelaksanaan Uji Kompetensi Mata Pelajaran Plc Pada Kompetensi Keahlian Teknik Otomasi Industri SMK Negeri 2 Pati,” *Diss. UNY*, vol. 3, no. 3, p. 1, 2015.
- [4] W. Raharjo, “Rancang Bangun Alat Trainer Otomasi Sebagai Media Pembelajaran Mata Kuliah Otomasi Industri Program Studi Teknik Industri Universitas Muhammadiyah Surakarta,” 2018.
- [5] H. Rahadian and M. A. Heryanto, “Pengembangan Human Machine Interface (HMI) pada Simulator Sortir Bola sebagai Media Pembelajaran Otomasi Industri,” *J. Nas. Tek. ELEKTRO*, vol. 9, no. 2, p. 84-92, 2020.
- [6] A. Julisman, I. D. Sara, and R. H. Siregar, “Prototipe Pemanfaatan Panel Surya Sebagai Sumber Energi Pada Sistem Otomasi Atap Stadion Bola,” *KITEKTRO J. Online Tek. Elektro*, vol. 2, no. 1, pp. 35–42, 2017.
- [7] I. Puspasari, Y. Triwidyastuti, and H. Harianto, “Otomasi Sistem Hidroponik Wick Terintegrasi pada Pembibitan Tomat Ceri,” *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 7, no. 1, pp. 1–10, 2018.
- [8] N. D. Setiawan, “Otomasi Pencampur Nutrisi Hidroponik Sistem NTF (Nutrient Film Technique) Berbasis Arduino Mega 2560,” *J. Tek. Inform. Unika St. Thomas*, vol. 03, no. 2, pp. 78–82, 2018.
- [9] A. Sahara, R. H. Saputra, and F. Oktafiani, “Sistem Smart Garden dalam Ruang Berbasis Arduino UNO Microcontroller ATmega 328,” *PETROGAS J. Energy Technol.*, vol. 1, no. 1, pp. 1–12, 2019.
- [10] A. Taufik, R. H. Saputra, and A. M. M. Huda, “Estmasi State of Charge Baterai Regulated Lead Acid Deep-Cycle 12V Dengan Metode Coulomn Counting,” *J. JIEOM*, vol. 2, no. 1, pp. 6–9, 2019.
- [11] B. Sugeng and R. H. Saputra, “Estimasi State-Of-Charge Menggunakan Simulink Pada Baterai Pembangkit Listrik Tenaga Surya,” *J. ELTIKOM*, vol. 3, no. 1, pp. 1–8, 2019.
- [12] B. G. Carkhuff, P. A. Demirev, and R. Srinivasan, “Impedance-Based Battery Management System for Safety Monitoring of Lithium-Ion Batteries,” *IEEE Trans. Ind. Electron.*, vol. 65, no. 8, pp. 6497–6504, 2018.
- [13] M. U. Ali, A. Zafar, S. H. Nengroo, S. Hussain, M. J. Alvi, and H. J. Kim, “Towards a smarter battery management system for electric vehicle applications: A critical review of lithium-ion battery state of charge estimation,” *Energies*, vol. 12, no. 3. pp. 1-33, 2019.
- [14] Y. Wang *et al.*, “A comprehensive review of battery modeling and state estimation approaches for advanced battery management systems,” *Renewable and Sustainable Energy Reviews*, vol. 131. pp. 23–34, 2020.
- [15] R. Xiong, Y. Zhang, J. Wang, H. He, S. Peng, and M. Pecht, “Lithium-Ion Battery Health Prognosis Based on a Real Battery Management System Used in Electric Vehicles,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4110–4121, 2019.
- [16] H. Gabbar, A. Othman, and M. Abdussami, “Review of Battery Management Systems (BMS) Development and Industrial Standards,” *Technologies*, vol. 9, no. 2, pp. 1-23, 2021.
- [17] M. Lelie *et al.*, “Battery management system hardware concepts: An overview,” *Appl. Sci.*, vol. 8, no. 4, pp. 1–12, 2018.
- [18] B. Maharmi, F. Ferdian, and F. Palaha, “Sistem Akuisisi Data Solar Cell Berbasis Mikrokontroler dan Labview,” *SainETIn*, vol. 4, no. 1, pp. 19–24, 2019.
- [19] B. Wu, W. D. Widanage, S. Yang, and X. Liu, “Battery digital twins: Perspectives on the fusion of models, data and artificial intelligence for smart battery management systems,” *Energy AI*, vol. 1, no. 2, pp. 1–10, 2020.