Volume 18 Nomor 2 (2023) 1-10
E – Link P-ISSN 1858-2109 E-ISSN 2656-5676
Jurnal Teknik Elektro dan Informatika

PERANCANGAN SISTEM SCADA *BATTERY MANAGEMENT SYSTEM* MENGGUNAKAN MYSQL DAN *WONDERWARE INTOUCH*

Riza Hadi Saputra[1)] , Muhammad Agung Nursyeha[2)], Muchammad Chilmi[3)]
[1,2,3)] Program Studi Teknik Elektro, Jurusan Teknologi Industri dan Proses,
Institut Teknologi Kalimantan
Jl. Soekarno-Hatta Km. 15, Karang Joang, Balikpapan, Kalimantan Timur, 76127
E-mail: [1)]riza.saputra@lecturer.itk.ac.id, [2)]agung.nursyeha@lecturer.itk.ac.id,
[3)]04201054@student.itk.ac.id

ABSTRAK

Permasalahan utama pada Battery Management System (BMS) adalah keterbatasan dalam pengumpulan data lapangan dan keterlambatan dalam pengambilan keputusan darurat. Solusi yang telah digunakan sebelumnya adalah penggunaan SCADA untuk mengintegrasikan data dari peralatan yang tersebar. Namun, dalam konteks BMS, diperlukan integrasi data dari berbagai sensor seperti tegangan, arus, dan temperatur. Metode penelitian melibatkan integrasi sensor-sensor relevan ke dalam sistem SCADA yang ada, dengan harapan meningkatkan efisiensi, keamanan, dan responsivitas dalam manajemen baterai, dengan potensi aplikasi dalam berbagai industri seperti kendaraan listrik dan energi terbarukan. Penggunaan BMS sangat penting karena memecahkan beberapa permasalahan kritis dalam manajemen baterai. BMS melindungi baterai dari kondisi berbahaya seperti *overcharging* dan *overheating*, serta memaksimalkan kinerja baterai. Dalam penelitian ini, telah dibuat SCADA menggunakan *WW Intouch* sebagai tampilan dan MySQL sebagai penyimpan data pada baterai lithium yang dihubungkan dengan sensor tegangan, arus, dan temperatur. Hasil dari percobaan ini didapatkan adalah nilai tegangan yang ditampilkan pada *database* berbeda dengan yang dikalibrasi. Besar nilai galatnya yaitu 7% dengan tingkat akurasi sebesar 93%. *Database* dibuat untuk menyimpan data tegangan, arus, temperatur, dan SOC yang telah diperoleh dari Arduino.
Kata kunci : SCADA, BMS, WW Intouch, MySQL, Arduino

ABSTRACT

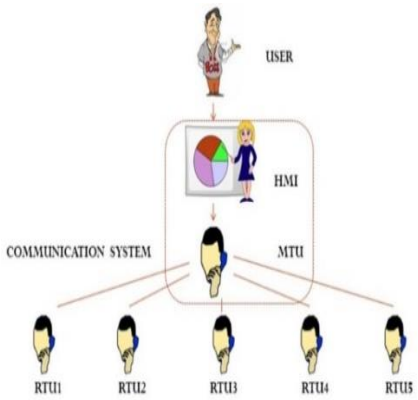
The main problems with the Battery Management System (BMS) are limitations in field data collection and delays in emergency decision making. The solution that has been used before is the use of SCADA to integrate data from dispersed equipment. However, in the context of BMS, integration of data from various sensors such as voltage, current, and temperature is required. The research method involves the integration of relevant sensors into existing SCADA systems, with the hope of improving efficiency, safety, and responsiveness in battery management, with potential applications in various industries such as electric vehicles and renewable energy. The use of BMS is very important because it solves several critical issues in battery management. The BMS protects the battery from dangerous conditions such as overcharging and overheating, and maximizes battery performance. In this experiment, SCADA has been created using WW Intouch as a display and MySQL as data storage on a lithium battery connected to voltage, current, and temperature sensors. The result of this experiment is that the voltage value displayed in the database is different from the calibrated one. The error value is 7% with an accuracy rate of more than 90%. A database is created to store the voltage, current, temperature, and SOC data that has been obtained from the Arduino.
***Keywords**: SCADA, BMS, WW Intouch, MySQL, Arduino*

Diterima Redaksi : 13-01-2023 | Selesai Revisi : 04-10-2023 | Diterbitkan Online : 31-10-2023

1. PENDAHULUAN

Permasalahan dalam pengawasan dan pengendalian sistem yang tersebar di berbagai area adalah adanya keterbatasan dalam pengumpulan informasi keadaan peralatan di lapangan dan keterlambatan dalam pengambilan keputusan saat terjadi masalah atau gangguan [1]. Operator seringkali harus mengumpulkan data secara manual dan mengirimkannya ke pusat, yang tidak hanya memakan waktu tetapi juga berisiko terlambat dalam mengatasi situasi darurat [2].

Solusi yang telah ada sebelumnya adalah penggunaan SCADA (*Supervisory Control and Data Acquisition*) untuk mengintegrasikan semua data dari peralatan yang tersebar menjadi terpusat dan dapat diakses secara *real-time* [3]. Namun, dalam beberapa aplikasi seperti *Battery Management System* (BMS), ada kebutuhan untuk mengkombinasikan pengukuran dari berbagai sensor seperti tegangan, arus, dan temperatur untuk mengelola baterai dengan lebih efisien [4]. Berikut pada Gambar 1 adalah struktur SCADA.



Gambar 1. Struktur SCADA

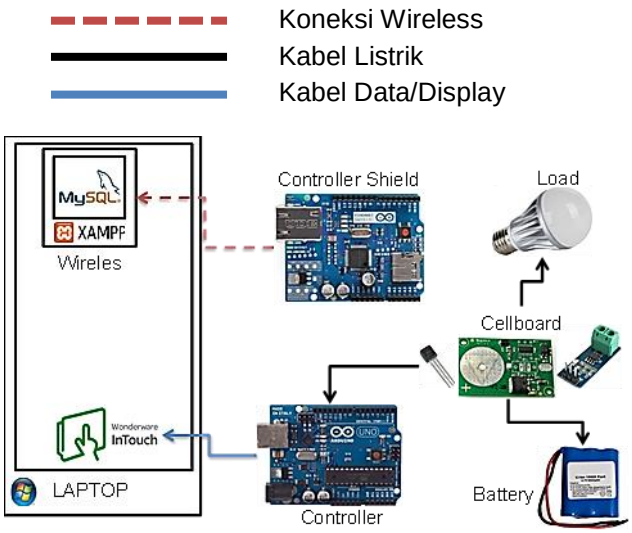
Solusi yang ditawarkan dalam penelitian ini adalah pengembangan SCADA khusus untuk BMS yang menggunakan sensor-sensor tersebut untuk mengambil data penting terkait baterai, seperti tegangan, arus, dan temperatur. Data ini akan dihubungkan ke Arduino sebagai pengambil data dari sensor, dan selanjutnya disimpan dalam *database* MySQL [5]. Pengguna akan dapat memantau dan mengendalikan BMS melalui antarmuka HMI (*Human-Machine Interface*) menggunakan perangkat lunak *Wonderware InTouch* [6]. Berikut pada Gambar 2 yaitu tampilan sistem SCADA pada BMS.

Gambar 2. Tampilan sistem SCADA pada BMS

Metodologi singkat yang digunakan dalam penelitian ini adalah pengintegrasian sensor-sensor yang relevan dengan BMS ke dalam sistem SCADA yang sudah ada, dengan menggunakan Arduino untuk pengambilan data sensor dan MySQL untuk menyimpan data [7]. Antarmuka pengguna yang intuitif akan dibangun dengan *Wonderware InTouch* untuk memantau dan mengontrol BMS secara *real time* [8]. Dengan demikian, pada penelitian ini, solusi ini akan meningkatkan efisiensi dan keamanan dalam pengelolaan baterai dalam berbagai aplikasi, seperti kendaraan listrik dan energi terbarukan. Pada penelitian ini menggunakan MySQL dan *WW Intouch* pada BMS yang akan dibuat [9].

2. METODE PENELITIAN

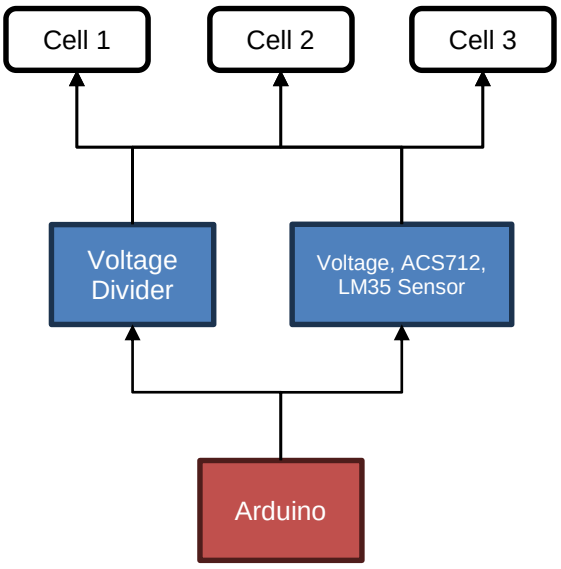
Dalam percobaan ini dilakukan dua tahap percobaan. Tahap pertama yaitu membuat perancangan sistem otomasi baterainya, misalnya perancangan baterai, sensor, dan dihubungkan ke Arduino untuk didapatkan data-data yang dibutuhkan. Selanjutnya tahap kedua, yaitu menyimpan data-data yang telah diakuisisi oleh Arduino ke dalam *database* MySQL dan HMI *WW Intouch*. Pada Gambar 3 adalah skema besar rancangan pada percobaan sistem SCADA untuk baterai.



Gambar 3. Skema rancangan sistem SCADA pada *Battery Management System*

2.1. Perancangan Sistem Otomasi BMS

Tahap pertama akan dilakukan perancangan sistem *Battery Management System* pada 3 sel baterai tipe lithium-ion dengan masing-masing spesifikasi 3.7-volt dan 2200 mAh. Alat kendali yang digunakan adalah Arduino tipe Mega2560 yang terkenal dengan banyaknya pin I/O. Sensor arus yang digunakan ACS712-30 yang bisa mendeteksi arus hingga 30 Ampere dan sensor temperatur menggunakan LM35. Pada Gambar 4 adalah skema *Battery Management System* yang telah dibuat [10].

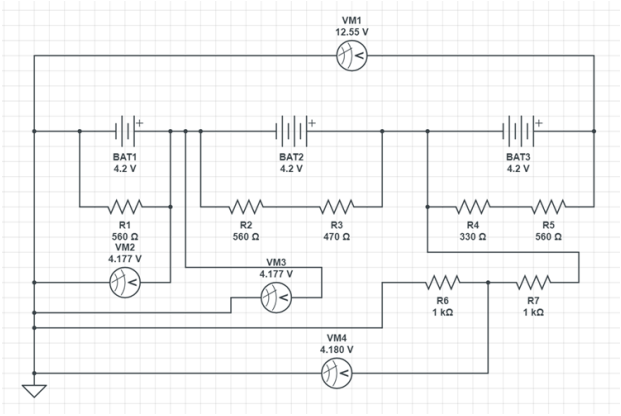


Gambar 4. Skema percobaan *Battery Management System*

Arduino berfungsi sebagai kontrol semua sensor yang terhubung seperti ACS712-30 dan LM35 untuk mendeteksi arus dan temperatur. Untuk mendeteksi sensor tegangan, ada di rangkaian *cellboard*. Rangkaian *cellboard* sendiri terdiri dari rangkaian pembagi tegangan atau *voltage divider* karena tegangan yang masuk ke Arduino hanya diperbolehkan 0 – 5 Volt. Baterai yang dipakai adalah 3 buah lithium-ion dengan masing-masing tegangan minimum 3.7 Volt dan tegangan maksimal 4.2 Volt yang dihubungkan secara seri sehingga menghasilkan tegangan total 12.6 Volt. Tegangan total ini diperlukan untuk menyalakan lampu LED dengan tegangan 12 VDC, 3 W yang diantaranya ada sensor arus ACS712-30 untuk mendeteksi arus yang lewat jika ada beban atau tidak. Untuk sensor LM35 diperlukan mendeteksi temperatur lingkungan sekitar baterai agar kondisi baterai tetap optimal.

2.1.1. *Cellboard*

Cellboard adalah sebuah papan sirkuit elektronik yang dirancang untuk memantau atau mengendalikan sel-sel baterai dalam sebuah paket baterai. *Cellboard* biasanya terdiri dari sensor suhu, sensor tegangan, dan mikrokontroler yang memungkinkan pengguna untuk memantau kondisi setiap sel baterai dan menentukan pengisian dan pengosongan baterai secara tepat. Dengan adanya *cellboard*, pengguna dapat mengoptimalkan kinerja dan daya tahan baterai serta menjaga keamanan selama penggunaan. *Cellboard* sering digunakan dalam aplikasi kendaraan listrik, sistem penyimpanan energi, dan peralatan medis. *Cellboard* dalam percobaan ini menggunakan rangkaian pembagi tegangan atau *voltage divider* pada setiap sel baterai [11]. Tegangan minimum pada baterai adalah 3.7 Volt dan tegangan maksimalnya adalah 4.2 Volt. Dalam rangkaian pembagi tegangan, kita harus menentukan nilai resistor yang akan dipakai dalam percobaan ini dengan spesifikasi tegangan baterai yang telah disebutkan. Pada Gambar 5 adalah skematik rangkaian pembagi tegangan pada *cellboard* yang telah dibuat.



Gambar 5. Skematik rangkaian pembagi tegangan atau *cellboard*

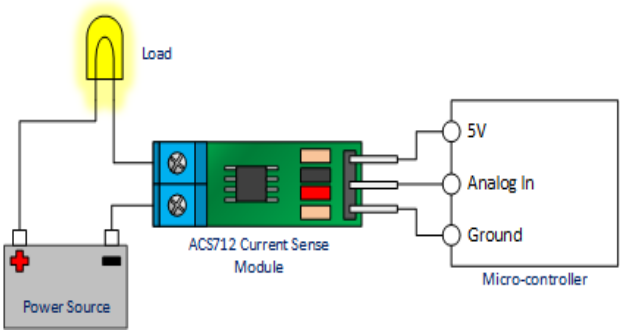
2.1.2. LM35

Sensor LM35 adalah sensor suhu analog yang dirancang khusus untuk mengukur suhu dalam rentang -55 derajat Celsius hingga 150 derajat Celsius. Karakteristik penting dari sensor LM35 adalah akurasi dan kestabilan suhu yang tinggi, serta output yang linier dengan perubahan suhu. Output sensor LM35 dinyatakan dalam bentuk tegangan, dengan kenaikan tegangan sebesar 10 mV per derajat Celsius. Sensor LM35 sangat penting dalam banyak aplikasi yang memerlukan pengukuran suhu yang akurat dan stabil, seperti sistem pendingin dan pemanas ruangan, pengendalian suhu pada industri makanan dan minuman, serta dalam pengukuran suhu pada kendaraan dan mesin industri. Karena ukurannya yang kecil dan penggunaannya yang mudah, sensor LM35 juga digunakan dalam banyak proyek DIY dan eksperimen elektronika. Pada sensor temperatur LM35 terdapat 3 pin yang mempunyai fungsi masing-masing. Pin 1, yang paling kiri, adalah pin Vcc yang dihubungkan pada Arduino dengan tegangan 5 Volt. 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.

Gambar 6. Sensor temperatur LM35

2.1.3. ACS712-30

Sensor ACS712 adalah sensor arus DC yang dirancang untuk mengukur arus listrik dalam rentang 5 Ampere hingga 30 Ampere. Karakteristik penting dari sensor ACS712 adalah keakuratan pengukuran yang tinggi, isolasi galvanik yang memungkinkan pengukuran arus secara aman tanpa perlu memutuskan sambungan kabel, dan output tegangan yang linier dengan arus yang diukur. Output sensor ACS712 dinyatakan dalam bentuk tegangan analog, dengan rentang tegangan keluaran yang proporsional dengan arus yang diukur. Sensor ACS712 sangat penting dalam banyak aplikasi yang memerlukan pengukuran arus listrik yang akurat, seperti dalam sistem pengisian baterai, pengendalian motor, dan pengukuran daya pada sistem pembangkit listrik tenaga surya dan angin. Karena ukurannya yang kecil dan penggunaannya yang mudah, sensor ACS712 juga sering digunakan dalam proyek DIY dan eksperimen elektronika untuk memonitor arus listrik pada rangkaian elektronik. Struktur hampir sama dengan sensor temperatur LM35 yaitu terdiri dari 3 pin yaitu Vcc, analog input, dan *ground*. Pin 1, yang paling kiri, adalah pin Vcc yang dihubungkan pada Arduino dengan tegangan 5V. pin 2, yang di tengah, 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. Namun yang membedakannya adalah objek yang akan dideteksinya. Untuk sensor ACS712-30 dihubungkan diantara kabel negatif pada beban, seperti pada Gambar 7.



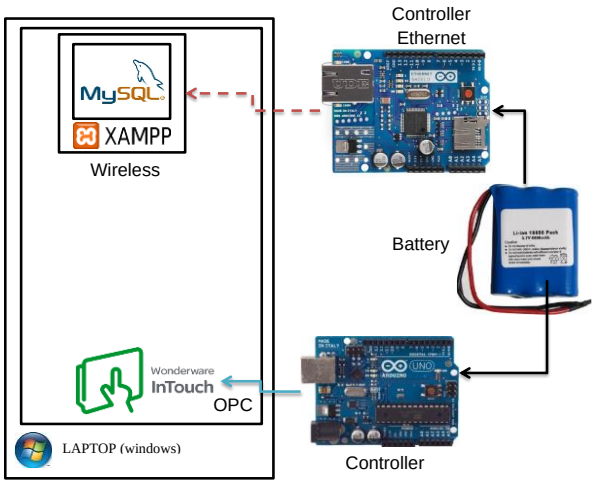
Gambar 7. Struktur sensor ACS712

2.2.Perancangan Sistem Informasi BMS

Dalam percobaan ini akan menghubungkan antara Arduino dengan Arduino IDE agar data yang diperoleh dari baterai dapat diakuisisi oleh Arduino. Selanjutnya data yang telah diperoleh akan disimpan di dalam *database* MySQL melalui aplikasi XAMPP dan ditampilkan pada *WW Intouch* sebagai tampilannya.

Dalam kasus komunikasi dari Arduino ke *database* ada beberapa tahap yaitu komunikasi antara *battery pack* dengan Arduino, lalu komunikasi Arduino dengan HMI, dan komunikasi Arduino dengan *database* MySQL. Dalam percobaan ini, kita menggunakan dua Arduino. Arduino Mega2560 dengan Ethernet *shield* berfungsi untuk mengirimkan data dari *battery pack* berupa tegangan, arus, dan temperatur ke *database* MySQL melalui *router* yang berfungsi sebagai terminal bagi Arduino Mega2560 dan laptop [12]. Arduino Mega2560 dihubungkan dengan kabel ethernet ke *router* dan laptop dihubungkaan melalui nirkabel atau *wireless* dari *router*, sedangkan Arduino Uno berfungsi menampilkan data dari *battery pack* dengan *WW Intouch* melalui OPC dengan kabel serial. Selanjutnya, untuk Arduino Mega2560 dengan ethernet *shield* menyimpan data yang diperoleh ke MySQL dalam aplikasi XAMPP. Sedangkan untuk Arduino Uno hanya menampilkan data yang diperoleh pada *WW Intouch*. Sistem operasi yang digunakan pada percobaan ini adalah *Windows* 7. Pada Gambar 8 adalah skema rancangan komunikasi yang telah dibuat.

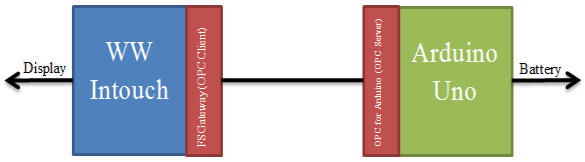
Ada dua jenis komunikasi Arduino yang dilakukan pada percobaan ini yang dapat dilihat pada Gambar 9. Pertama adalah komunikasi dengan ethernet yang berfungsi menyimpan data yang telah diperoleh ke dalam *database*. Sedangkan yang kedua adalah komunikasi dengan OPC Arduino, yang berfungsi menampilkan data yang telah diperoleh dari *battery pack*.



Gambar 8. Skema rancangan komunikasi *Battery Management System*

Setelah membuat program pada Arduino, selanjutnya adalah komunikasi antara program Arduino ke bahasa PHP agar dapat dibaca oleh MySQL dan data yang telah diambil dapat disimpan dalam *database*. Tujuan perubahan bahasa ini hanyalah konversi dari bahasa Arduino ke dalam bahasa PHP. Selanjutnya yang kedua adalah Arduino yang dihubungkan ke HMI atau *WW Intouch* melalui OPC serial. OPC adalah perangkat lunak untuk menghubungkan satu alat dengan yang lainnya walaupun berbeda jenis. OPC juga bisa disebut penyamaan perangkat lunak dari perangkat lunak yang berbeda-beda. Pada Arduino Uno digunakan komunikasi OPC. OPC adalah kependekan dari *OLE for Process Control* sedangkan OLE kependekan dari *Object Linking and Embedding*. OLE dimiliki oleh perusahaan perangkat lunak terbesar didunia yaitu Microsoft. Jadi pengertian dari OPC adalah sebuah perangkat lunak yang menghubungkan beberapa macam perangkat lunak lainnya yang digunakan pada industri khususnya otomasi agar dapat menghubungkan antar perangkat lunak yang berbeda merek, seperti mengubungkan perangkat lunak PLC Siemens dengan HMI WW *Intouch*. Namun berkembangnya teknologi dan

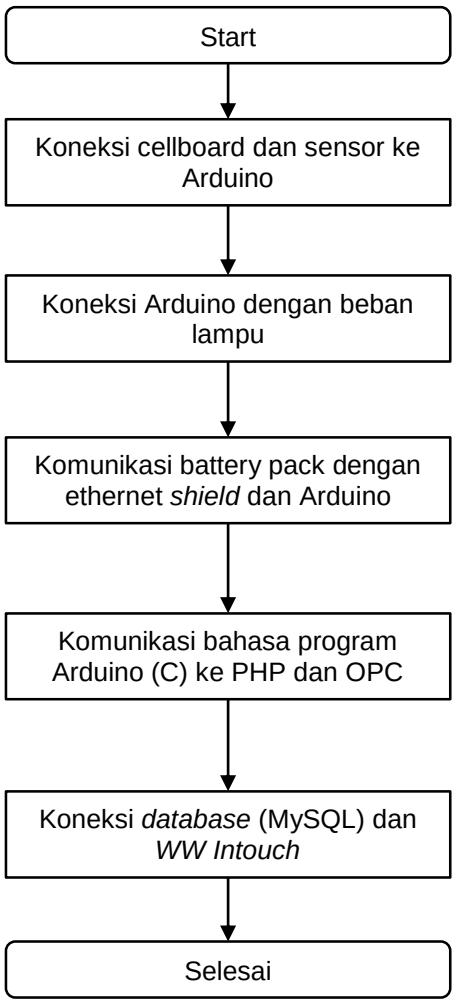
banyaknya industri yang memakainya, maka sekarang OPC merupakan singkatan dari *Open Platform Communication* [13]. Dalam percobaan ini digunakan *OPC for Arduino* yang berfungsi menghubungkan antara Arduino IDE dengan *WW Intouch* atau sebagai *OPC Server* pada percobaan ini [14]. Di dalam program WW *Intouch* ada sebuah program OPC yang diberi nama *FSGateway* yang berfungsi sebagai *OPC Client* yang berfungsi menerima data yang diberikan oleh *OPC Server*.



Gambar 9. Skema rancangan OPC pada Arduino Uno dan *WW Intouch*

2.3.Metodologi Percobaan

Dapat dilihat pada Gambar 10 yaitu 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 12 VDC, 3 W sebelum menghubungkan sensor ACS712-30. Setelah beban dicoba dan menyala, selanjutnya menghubungkan sensor ACS712-30 diantara beban dan baterai agar terdeteksi arus yang lewat. Selanjutnya adalah pekerjaan paralel yaitu menghubungkan *battery pack* dengan Arduino Mega2560 dan Arduino Uno. Arduino Mega2560 digunakan untuk mengambil data lalu disimpan ke dalam *database* sedangkan Arduino Uno mengambil data lalu ditampilkan tanpa menyimpannya. Selanjutnya menentukan bahasa komunikasi untuk kedua Arduino. Untuk Arduino Mega2560 mengubah bahasa Arduino ke dalam bahasa PHP sedangkan Arduino Uno menggunakan tambahan deklarasi komunikasi OPC pada bahasa Arduino. Selanjutnya adalah menyambungkan Arduino Mega2560 ke *database* atau MySQL pada aplikasi XAMPP di laptop dan Arduino Uno disambungkan ke WW Intouch melalui *OPC Server* (*OPC for Arduino*) dan *OPC Client* (*FSGateway*).



Gambar 10. Diagram Alir Sistem SCADA pada *Battery Management System*

3. HASIL DAN DISKUSI

Setelah merancang dan menjalankan sistemnya, maka langkah selanjutnya adalah melihat data yang telah diambil oleh Arduino. Pertama ditampilkan adalah data tegangan dari masing-masing baterai dan total tegangan baterai.

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 11. Tangkapan layar data yang ditampilkan pada *database*

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 11 adalah tangkapan layar data yang diperoleh setelah dihubungkan dengan *database*.

3.1.Hasil Pengambilan Data Tegangan

Dapat dilihat pada Gambar 11 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 sebelahnya, ada total tegangan yaitu 12.52 Volt yang didapatkan dari penjumlahan tegangan ketiga sel baterai.

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 66 mV untuk sensor ACS712-30. Berikut pada Persamaan 1 adalah persamaan matematika untuk sensor ACS712-30.

$$Vout = \frac{Vcc}{2} + Sensitivity * I \qquad (1)$$

Dalam Persamaan 1 dapat dijabarkan sebagai Vout adalah tegangan keluaran sensor (dalam Volt), Vcc adalah tegangan catu daya yang diberikan ke sensor (biasanya 5 Volt), dan sensitivity adalah sensitivitas sensor ACS712-30 yang biasanya adalah 66 mV per Ampere (0,066 V/A). Sedangkan I adalah arus listrik yang diukur (dalam Ampere). Dapat dilihat pada Gambar 11 bahwa arus yang terdeksi 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. Lampu yang digunakan berspesifikasi rendah yaitu 12 VDC, 3 W, maka arus yang dibutuhkan juga kecil. Ini alasan utama mengapa arus positif saat dihubungkan beban bernilai 0.074 atau hampir mendekati 0.

3.3.Hasil Pengambilan Data Temperatur

Data temperatur juga berdasarkan data tegangan. Sensor LM35 mengubah satuan tegangan menjadi derajat celcius adalah 10^0 C / 10 mV. Untuk pemrograman di Arduino yaitu tegangan dikalikan tegangan referensi dikalikan dengan nilai maksimal dari temperatur yang akan dideteksi. Untuk data yang didapatkan dapat dilihat pada Gambar 11.

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 11. 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).

$$SOC = map(tegangan, mintegref, maxtegref, minsoc, maxsoc);$$

Program diatas menunjukkan nilai tegangan yang akan dirubah ke dalam nilai SOC. Tegangan didapatkan dari jumlah tegangan pada baterai yaitu sekitar 12 VDC 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 mengubah dari nilai tegangan yang terdeteksi. Berikut adalah contoh nilai SOC pada tegangan 12 VDC.

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

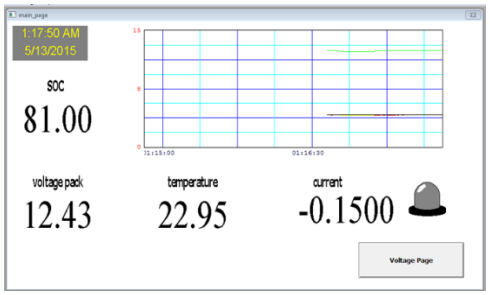
Data6 adalah nilai SOC yang akan dicari. Sedangkan data5 adalah nilai tegangan yang akan berubah-ubah sesuai kondisi baterai saat itu. 115 adalah nilai minimal tegangan referensi. 126 adalah nilai maksimal tegangan referensi. 0 adalah nilai minimal SOC. 100 adalah nilai maksimal SOC. Mengapa dikali 10? 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.

3.5.Hasil Tampilan *Database*

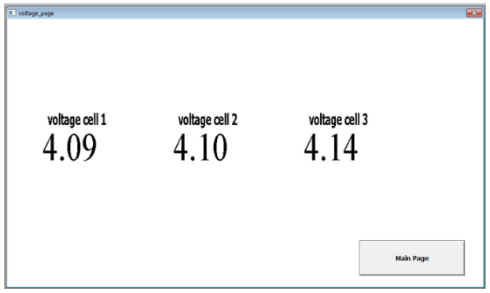
Setelah masalah komunikasi selesai, maka selanjutnya memperlihatkan hasil *database* yang telah dibuat. *Database* dibuat di MySQL pada program XAMPP dengan data yang didapatkan dari Arduino, seperti data tegangan, arus, dan temperatur. Untuk gambar, telah ditampilkan sebelumnya pada Gambar 11.

3.6.Hasil HMI pada WW Intouch

Bersamaan dengan komunikasi ke *database*, juga dilakukan komunikasi ke *WW Intouch* untuk menampilkan data dari Arduino Uno. Data yang diambil dari Arduino Uno tidak disimpan ke dalam *database* melainkan langsung ditampilkan ke *WW Intouch* karena Arduino Mega2560 dengan *ethernet shield* sudah menyimpannya pada MySQL. Pada Gambar 12 dan Gambar 13 adalah HMI yang berhasil dibuat pada penelitian ini.



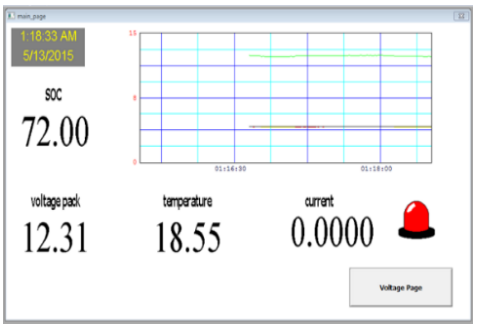
Gambar 12. Tampilan halaman utama *WW Intouch* yang telah dibuat



Gambar 13. Tampilan halaman tegangan sel baterai *WW Intouch* yang telah dibuat

Gambar 12 dan 13 adalah hasil tampilan dari WW Intouch saat program telah dijalankan. Gambar 12 adalah tampilan dari halaman utama yaitu menampilkan data tegangan total, temperatur, arus, grafik tegangan, dan indikator beban. Untuk Gambar 13 adalah halaman kedua, yaitu halaman tegangan semua sel yang sedang diawasi. Dalam percobaan ini ada 3 sel baterai yang sedang diawasi. Dalam Gambar 12 indikator beban sedang tidak menyala atau tidak ada lampu yang terhubung. Program yang terkait dengan indikator beban bergantung pada arus yang sedang dideteksi. Mengacu pada *database*, saat arus bernilai negatif, maka tidak ada beban dan jika arus bernilai positif, maka ada beban yang terhubung. Berikut pada Gambar 14 adalah saat beban menyala dan arus menjadi

positif. Selanjutnya yaitu program alarm jika kapasitas baterai mulai menipis. Dalam percobaan ini dicoba jika baterai dibawah 80% maka aka nada peringatan kelap-kelip bertulisan "*must be charged*" yang mengingatkan kita bahwa baterai sudah dibawah 80%. Berikut pada Gambar 15 untuk tampilannya dan Gambar 16 untuk BMS yang telah dibuat.



Gambar 14. Tampilan halaman utama *WW Intouch* saat lampu menyala

Gambar 15. Tampilan halaman utama *WW Intouch* saat alarm baterai menyala

Gambar 16. BMS yang telah dibuat

4. KESIMPULAN DAN SARAN

Kesimpulan yang didapatkan dari percobaan ini yaitu telah dibuat BMS sederhana untuk mendeteksi tegangan, arus, temperatur, dan SOC dan juga *database* pada MySQL dan HMI pada *WW Intouch*. Nilai tegangan yang ditampilkan pada *database* tidak sama dengan nilai yang dikalibrasi menggunakan multimeter dikarenakan tidak tepatnya pemilihan tegangan referensi. Besar nilai galatnya yaitu 7% dengan tingkat akurasi sebesar 93%. Selain itu, jika nilai arus bernilai negatif, maka tidak ada beban yang terhubung, sedangkan jika arus bernilai positif, maka ada beban yang terhubung. Hasil SOC menggunakan *syntax* "map" banyak mengalami *error* setiap penurunan tegangan. Data yang disimpan dan ditampilkan pada *database* adalah tegangan, arus, temperatur, dan SOC. Data yang diambil dari Arduino Uno tidak disimpan ke dalam *database* melainkan langsung ditampilkan ke *WW Intouch* karena Arduino Mega2560 dengan *ethernet shield* sudah menyimpannya pada MySQL.

Saran pada percobaan ini yaitu menggunakan pin Aref pada Arduino sebagai tegangan referensi agar tegangan yang ditampilkan dan yang dikalibrasi bernilai sama. Selain itu, menghitung SOC harus menggunakan metode agar nilai yang ditampilkan tidak mengalami *error* setiap penurunan tegangan. Membuat HMI pada web atau dengan bahasa PHP agar menggunakan satu Arduino saja. Khususnya membuat laporan khusus harian atau bulanan agar mudah menganalisis keadaan baterai

5. DAFTAR PUSTAKA

[1] T. An *et al.*, "Self-powered gold nanowire tattoo triboelectric sensors for soft wearable human-machine interface," *Nano Energy*, vol. 77, no. 2, pp. 1–12, 2020, doi: 10.1016/j.nanoen.2020.105295.

[2] K. Friansa, I. N. Haq, B. M. Santi, D. Kurniadi, E. Leksono, and B. Yuliarto, "Development of Battery Monitoring System in Smart Microgrid Based on Internet of Things (IoT)," in *Procedia Engineering*, 2017, vol. 170, pp. 482–487. doi: 10.1016/j.proeng.2017.03.077.

[3] R. Lin *et al.*, "Wireless battery-free body sensor networks using near-field-enabled clothing," *Nat. Commun.*, vol. 11, no. 1, pp. 1–9, 2020, doi: 10.1038/s41467-020-

14311-2.

[4] Y. T. Kwon *et al.*, "All-printed nanomembrane wireless bioelectronics using a biocompatible solderable graphene for multimodal human-machine interfaces," *Nat. Commun.*, vol. 11, no. 1, pp. 1–10, 2020, doi: 10.1038/s41467-020-17288-0.

[5] T. Mulyana, H. Setiawan, and R. Ibrahim, "Investigation on Simulation of Wind and Solar Power Hybrid Systems through Human Machine Interface by InTouch Wonderware Software," in *Atlantis*, 2019, pp. 1–12. doi: 10.2991/icoiese-18.2019.44.

[6] Y. T. Kwon, H. Kim, M. Mahmood, Y. S. Kim, C. Demolder, and W. H. Yeo, "Printed, Wireless, Soft Bioelectronics and Deep Learning Algorithm for Smart Human-Machine Interfaces," *ACS Appl. Mater. Interfaces*, vol. 12, no. 44, pp. 49398–49406, 2020, doi: 10.1021/acsami.0c14193.

[7] J. Czarnowski, A. Dąbrowski, M. Maciaś, J. Główka, and J. Wrona, "Technology gaps in Human-Machine Interfaces for autonomous construction robots," *Autom. Constr.*, vol. 94, no. 3, pp. 179–190, 2018, doi: 10.1016/j.autcon.2018.06.014.

[8] Y. Yu *et al.*, "Biofuel-powered soft electronic skin with multiplexed and wireless sensing for human-machine interfaces," *Sci. Robot.*, vol. 5, no. 41, pp. 23–33, 2020, doi: 10.1126/SCIROBOTICS.AAZ7946.

[9] P. Asabere, F. Sekyere, W. O.-I. J. of, and undefined 2020, "Wireless biometric fingerprint attendance system using Arduino and MySQL *database*," *papers.ssrn.com*, vol. 7, no. 11, pp. 433–436, 2018, Accessed: Sep. 18, 2023. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3523688

[10] I. González, A. J. Calderón, and J. M. Portalo, "Innovative multi-layered architecture for heterogeneous automation and monitoring systems: Application case of a photovoltaic smart microgrid," *Sustain.*, vol. 13, no. 4, pp. 1–24, 2021, doi: 10.3390/su13042234.

[11] R. Balakrishnan and A. Senthilnathan, "PLC Based Frequency Control of Induction Motor in Sugar Mills," in *8th International Conference on Advanced Computing and Communication Systems, ICACCS 2022*, 2022, pp. 1219–1224. doi: 10.1109/ICACCS54159.2022.9785018.

[12] I. Tomar, I. Sreedevi, and N. Pandey, "PLC and SCADA based Real Time Monitoring and Train Control System for the Metro Railways Infrastructure," *Wirel. Pers. Commun.*, vol. 2, no. 3, pp. 123–140, 2022, doi: 10.1007/s11277-022-10109-1.

[13] G. Mahalakshmi, S. Sangeetha, Maladhi, and T. Bhavatharini, "Automation of industrial drives using PLC and SCADA," in *Proceeding of 2nd International Colloquium on Computational & Experimental Mechanics (ICCEM 2021)*, 2022, vol. 2545, p. 040009. doi: 10.1063/5.0108098.

[14] M. O. Qays *et al.*, "Monitoring of renewable energy systems by IoT-aided SCADA system," *Energy Sci. Eng.*, vol. 10, no. 6, pp. 1874–1885, Jun. 2022, doi: 10.1002/ese3.1130.