

PERANCANGAN SISTEM *MONITORING* DAN MANAJEMEN BATERAI DENGAN METODE *SWITCHING* BERBASIS MIKROKONTROLER ARDUINO UNO

Joshua Parulian Siahaan ^{*)}, Susatyo Handoko dan Darjat

Departemen Teknik Elektro, Fakultas Teknik, Universitas Diponegoro
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)} E-mail: jpsiahaann@gmail.com

Abstrak

Istilah *charging* dan *discharging* baterai adalah hal yang umum didengar dalam penggunaan baterai. Penyimpanan energi listrik ke dalam baterai sendiri harus di manajemen sedemikian rupa agar proses pengisian baterai (*charging*) dan pengosongan baterai (*discharging*) dapat berlangsung secara baik, tanpa mengakibatkan beban padam karena kehabisan daya baterai. Sistem dikontrol oleh sebuah mikrokontroler Arduino UNO, masukan berupa 3 buah sensor tegangan, 6 buah relai sebagai aktuator, LCD sebagai penampil status baterai, dan *buzzer* sebagai aktuator bunyi peringatan. Sensor tegangan membaca nilai tegangan pada seluruh baterai yang terpasang, kemudian Arduino UNO akan memproses baterai yang memiliki nilai diatas *setpoint* akan disambungkan ke beban melalui relai beban, sedangkan baterai yang tegangannya berada dibawah nilai *setpoint* akan disambungkan ke sistem *charging* melalui relai *charging*. Pada penelitian ini diberi beban bohlam lampu 25 watt, didapatkan hasil baterai 1 menyuplai beban kurang lebih selama 115 menit, kemudian *switching* ke baterai 2 dan baterai 1 di isi ulang, kemudian baterai 2 akan menyuplai beban selama kurang lebih 108 menit kemudian *switching* ke baterai 3 dan baterai 2 di isi ulang, baterai 3 menyuplai beban kurang lebih selama 107 menit. Perancangan sistem *monitoring* dan manajemen baterai dengan metode *switching* berbasis mikrokontroler Arduino UNO telah berhasil direalisasikan.

Kata kunci: Baterai, Charging, Discharging, Switching Baterai, Relai, Arduino UNO

Abstract

The terms of charging and discharging are commonly heard in battery usage. The storage of electrical energy in the battery itself must be managed in such a way so that the battery charging and discharging processes continuously, without causing outage due to lack of battery power. The system controlled by an Arduino UNO, 3 voltage sensors as an input, 6 relays as an actuator, LCD as a battery status display, and a buzzer as a warning sound actuator. The voltage sensor reads all installed batteries voltage, then the Arduino UNO will process the battery that has a value above the setpoint will be connected to the load via load relay, battery which voltage is below the setpoint value will be connected to the charging system via charging relay. In this study, a 25 watt bulb used as a load, as result, battery 1 supplied the load for about 115 minutes, then switching to battery 2 and battery 1 recharged, then battery 2 supplied the load for approximately 108 minutes then switching to battery 3 and battery 2 recharged, battery 3 supplied the load of approximately 107 minutes. The design of the battery monitoring and management system using the Arduino UNO with switching method has been successfully realized.

Keywords: Battery, Charging, Discharging, Battery Switching, Relay, Arduino UNO

1. Pendahuluan

Secara geografis Indonesia berada tepat di garis khatulistiwa dengan rata-rata intensitas radiasi matahari per harinya sekitar 4.8 kWh/m² [1]. Indonesia juga memiliki potensi yang cukup besar dalam energi bayu/angin yaitu sebesar 60,647 MW yang tersebar di 34 provinsi di Indonesia [2]. Menyadari potensi pengembangan energi baru terbarukan yang sangat besar di Indonesia, dan rasio elektrifikasi di Indonesia yang

belum seluruhnya 100% [3], energi terbarukan ini bisa menjadi solusi untuk daerah terpencil dan belum teraliri listrik secara merata.

Pada penggunaannya, sebagai contoh, Pembangkit Listrik Tenaga Surya (PLTS) bergantung bergantung pada besarnya radiasi matahari, jam matahari ekuivalen/*Equivalent Sun Hours* (ESH), sudut lintang [4]. Dikarenakan energi listrik yang dihasilkan terbatas dan tidak stabil, maka terlebih dahulu energi listrik harus

disimpan ke dalam baterai, agar pemakaian di siang hari stabil dan dapat dipakai di malam hari. Energi listrik yang telah disimpan di dalam baterai nantinya akan diubah menggunakan *inverter* menjadi tegangan AC 220 volt [5] agar dapat digunakan ke peralatan rumah tangga atau sejenisnya. Penyimpanan energi listrik ke dalam baterai sendiri harus di manajemen sedemikian rupa agar proses pengisian baterai (*charging*) dan pengosongan baterai (*discharging*) dapat berlangsung secara baik, tanpa mengakibatkan beban padam karena kehabisan daya baterai.

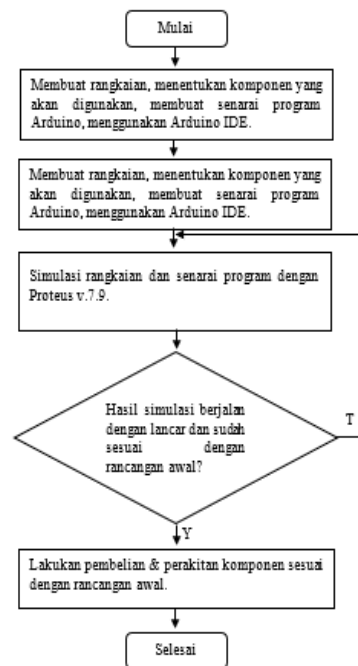
Berdasarkan masalah pada manajemen baterai yang telah disebutkan diatas, beberapa penelitian mengenai manajemen baterai dan sejenisnya telah dilakukan, secara singkat penelitian yang telah dilakukan ada yang menggunakan Arduino UNO dan aktuatornya berupa *Smart Relay Zelio* [6], namun didapatkan bahwa harga dari *Smart Relay Zelio* cukup mahal dan ukurannya relatif besar namun dapat digunakan untuk beban yang besar serta belum adanya LCD untuk mengetahui status tegangan dari baterai yang sedang digunakan dan yang sedang diisi ulang. Kemudian ada juga penelitian yang menggunakan Mikrokontroler AVR ATmega 8535 [7], yang menggunakan baterai Li-Po sebagai media penyimpanan energi listrik.

Berdasarkan kedua penelitian yang telah disebutkan, pada Tugas Akhir ini dilakukan pengembangan yang lebih berfokus pada perancangan sistem *switching*, yaitu dengan menggunakan modul relay elektro mekanik 5 volt yang harganya jauh lebih murah daripada *Smart Relay Zelio*, kemudian menggunakan Arduino UNO sebagai mikrokontroler yang lebih mudah dipahami bahasa pemrogramannya, dan lebih banyak digunakan saat ini, beberapa tambahan seperti LCD 2x16 sebagai penampil status tegangan dari baterai yang digunakan dan yang sedang diisi dan *buzzer* sebagai keluaran bunyi, menggunakan 3 buah baterai kering [8] [9] 12 volt 5 ah sebagai media penyimpanan energi listrik, serta *fuse*/pelebur sebagai pengaman alat dari arus lebih akibat beban berlebih maupun hubung singkat. Pada penelitian ini dibuatlah suatu alat yang dapat menghubungkan sistem pengisian (*charging*) dan pengosongan (*discharging*) pada tiga buah baterai. Proses awalnya mikrokontroler Arduino UNO melalui sensor tegangan [10] akan membaca kondisi tegangan ketiga baterai yang terpasang, kemudian sistem akan memilih baterai mana yang memiliki tegangan $\geq 11,5$ volt dan apabila terdapat lebih dari satu baterai yang memiliki tegangan $\geq 11,5$ volt, maka sistem akan memilih baterai dengan urutan yang lebih kecil terlebih dahulu, kemudian baterai akan menyuplai beban, dan tegangannya menurun hingga melewati batas tegangan $< 11,5$ volt nantinya baterai tersebut akan dilepas dari beban dan akan diisi hingga mencapai tegangan ≥ 13 volt, kemudian beban akan disambungkan ke baterai berikutnya yang tegangannya $\geq 11,5$ volt, sistem otomatis ini disebut dengan *switching* yang nantinya akan diatur oleh Arduino

UNO [11] dan aktuatornya adalah relay elektromekanik. Kemudian, apabila tegangan dari ketiga baterai tidak ada yang $\geq 11,5$ volt ataupun tidak ada baterai yang terpasang yang mengakibatkan pembacaan tegangan bernilai 0 volt, Arduino UNO akan mengaktifkan *buzzer* [12] sebagai peringatan bahwa seluruh baterai tidak dapat dipakai, pada sistem ini digunakan juga LCD 2x16 [13] sebagai penampil *real time* kondisi tegangan baterai & LED yang menampilkan baterai mana yang sedang digunakan dan yang sedang diisi. Sistem pengisian dan penggunaan baterai akan berlangsung secara bergantian (salah satu diisi dan lainnya dipakai ke beban, dan bergantian) seperti yang telah dijelaskan diatas diharapkan membuat pemakaian energi listrik dapat digunakan jauh lebih lama dibandingkan jika baterai diisi secara serempak dan digunakan sekaligus (baterai diisi penuh, kemudian dipakai sekaligus hingga habis, dan diisi kembali)

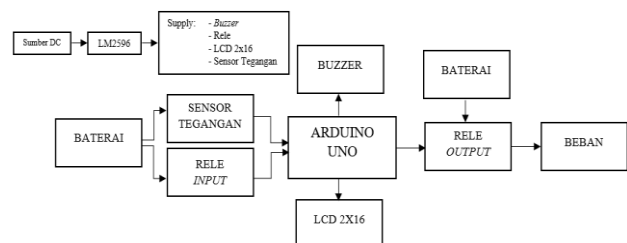
2. Metode

2.1. Perancangan Sistem



Gambar 1. Diagram alir perancangan sistem

2.2. Rancang Blok Diagram

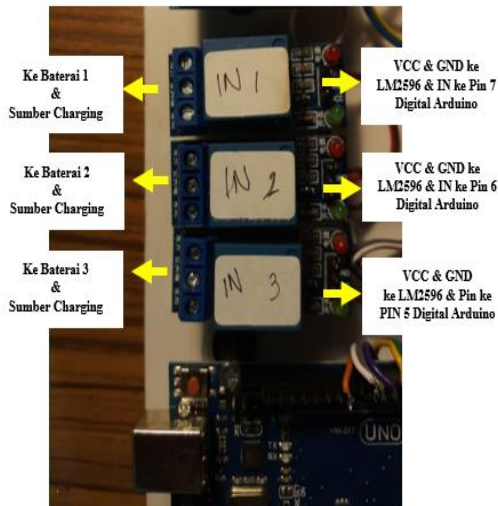


Gambar 2. Blok diagram sistem

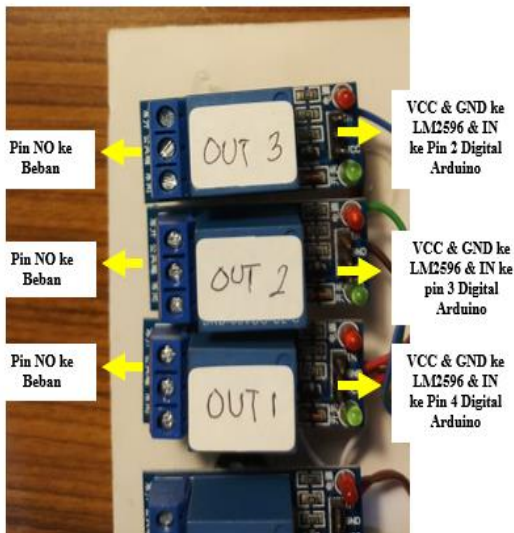
6. *Buzzer* sebagai aktuator bunyi apabila semua baterai tidak ada yang bernilai diatas *setpoint*, LM2596 sebagai sumber tegangan *buzzer*.

2.4.1. Perancangan Mikrokontroler Arduino UNO dan Relai *Input & Output*

Berikut gambar 5 dan 6 merupakan rangkaian Arduino UNO dan modul relai *input & output*.



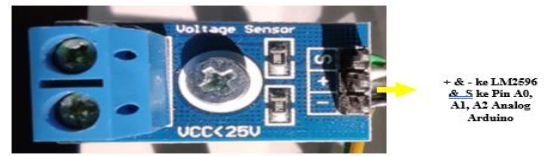
Gambar 5. Rangkaian relai *input*



Gambar 6. Rangkaian relai *output*

2.4.2. Perancangan Mikrokontroler Arduino UNO dan Modul Sensor Tegangan

Pada perancangan Tugas Akhir ini, sensor tegangan pada gambar 7 digunakan untuk mendeteksi tegangan pada baterai yang terpasang.



Gambar 7. Rangkaian modul sensor tegangan

2.4.3. Perancangan Mikrokontroler Arduino UNO dan Modul *Buzzer*

Pada perancangan tugas akhir ini, modul *buzzer* pada gambar 8 digunakan sebagai peringatan apabila tidak ada baterai yang memiliki tegangan diatas nilai *setpoint*.



Gambar 8. Rangkaian modul *buzzer*

2.4.4. Perancangan LCD 2x16 dan Modul I2C

Pada perancangan Tugas Akhir ini, LCD 2x16 yang terhubung dengan modul I2C [14] pada gambar 9 digunakan sebagai penampil status tegangan baterai, baterai mana yang sedang dipakai, baterai mana yang sedang diisi.



Gambar 9. Rangkaian modul LCD 2x16 dan I2C

2.4.5. Modul *stepdown* LM2596

Pada perancangan Tugas Akhir ini, modul *stepdown* LM2596 pada gambar 10 digunakan untuk menyuplai tegangan ke Relai, *buzzer*, LCD 2x16, sensor tegangan.



Gambar 10. Rangkaian modul *stepdown* LM2596

2.4.6. Beban

Beban yang digunakan berupa sebuah bohlam lampu motor merek STANLEY dengan spesifikasi 12 volt, 25 watt/ 25 watt seperti pada gambar 11, terdapat dua kumparan wolfram pada lampu ini, namun yang dipakai hanya satu sehingga beban bohlam lampu hanya 25 watt. Penggunaan bohlam lampu dipilih karena memiliki watt yang cukup besar, sehingga waktu yang diperlukan untuk menurunkan tegangan baterai tidak terlalu lama.



Gambar 11. Bohlam lampu 25 watt

2.5. Perancangan Perangkat Lunak

Pada Tugas Akhir ini dibutuhkan beberapa perangkat lunak dalam pembuatannya, yaitu sebagai berikut:

a. Proteus v7.9

Proteus adalah *software* yang dapat digunakan untuk merancang rangkaian dan menyimulasikannya sebelum dicetak di *Printed Circuit Board (PCB)* agar dapat diketahui apakah rangkaian sudah bekerja dengan baik atau belum. [15]

b. Arduino IDE

Arduino IDE adalah *software* yang disediakan di situs arduino.cc yang bertujuan sebagai perangkat pengembang *sketch* yang digunakan sebagai program dipapan Arduino [10].

2.6. Peletakan Alat

Seluruh peralatan pada penelitian tugas akhir ini diletakkan pada sebuah kotak kayu berukuran 25 cm x 20 x 4 cm.



Gambar 12. Realisasi keseluruhan rangkaian setelah dirakit



Gambar 13. Tampak atas peralatan setelah dirakit

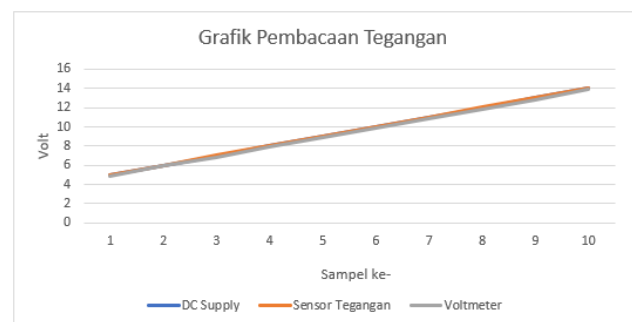
3. Hasil dan Pembahasan

3.1. Pengujian Sensor Tegangan

Tabel 1. Hasil pengujian sensor tegangan

No.	Voltmeter DC Power Supply	Tegangan yang Diamati (V)		Error (%) Voltmeter Dan Sensor Tegangan
		Voltmeter	Sensor Tegangan	
1	5,00	5,02	4,91	0,02
2	6,00	6,02	5,91	0,01
3	7,00	7,02	6,88	0,02
4	8,00	8,03	7,88	0,01
5	9,00	9,03	8,88	0,01
6	10,00	10,04	9,89	0,01
7	11,00	11,04	10,86	0,01
8	12,00	12,05	11,86	0,01
9	13,00	13,05	12,86	0,01
10	14,00	14,06	13,86	0,01
Rata-rata Error				0,011

Data Tabel 1 merupakan hasil pengujian sensor tegangan. Jika pembacaan sensor tegangan dibandingkan dengan alat ukur voltmeter didapatkan nilai *error* yang terbilang sangat kecil, grafik pembacaan tegangan dapat dilihat pada gambar 16.



Gambar 14. Grafik perbandingan pembacaan sensor tegangan, voltmeter, dan DC Supply

3.2. Pengujian Kontak Relai

Pengujian kontak relai ini bertujuan untuk mengetahui apakah relai dapat bekerja dengan baik atau tidak. Dari hasil pengujian relai dapat dilihat pada Tabel 2. Relai telah bekerja sesuai dengan perintah pada senarai program.

Tabel 2. Hasil Pengujian Kontak Relai

No.	Instruksi Program	Kontak NC	Kontak NO	Kondisi Lampu	Keterangan
1	LOW	Non-aktif	Aktif	Menyala	Benar
2	HIGH	Aktif	Non-Aktif	Padam	Benar

3.3. Pengujian Buzzer

Pengujian ini bertujuan untuk mengetahui apakah modul *buzzer* berfungsi dengan baik atau tidak. Pengujian ini dilakukan dengan cara memasukkan perintah dalam senarai program untuk menyalakan *buzzer*.

Tabel 3. Hasil Pengujian Buzzer

No.	Instruksi Program	Kondisi buzzer	Keterangan
1	LOW	Bunyi	Benar
2	HIGH	Diam	Benar

Tabel 3 merupakan hasil pengujian *buzzer* untuk mengetahui apakah *buzzer* mengeluarkan suara ketika dipicu pada senarai program.

3.4. Pengujian Sistem Switching Baterai

Secara sederhana cara kerja sistem *switching charging* dan *discharging* baterai adalah mengatur baterai mana yang akan dihubungkan ke beban, dan baterai mana yang akan di isi. Pengaturan ini berdasarkan pembacaan dari sensor tegangan, bila tegangan baterai $\geq 11,5$ volt maka baterai dapat disalurkan ke beban, sebaliknya bila tegangan baterai $< 11,5$ volt maka baterai akan di isi ulang, dan beban akan diteruskan ke baterai berikutnya yang memiliki tegangan $\geq 11,5$ volt, begitu seterusnya hingga sistem dipadamkan, atau ketika tidak ada baterai yang dalam kondisi siap, sehingga alarm hidup, dan sistem akan mengisi terlebih dahulu hingga ada baterai yang siap untuk dipakai. Terdapat beberapa kondisi baterai yang mungkin terjadi pada saat peralatan pertama kali dihidupkan:

Tabel 4. Kondisi baterai dan respon sistem

No	Baterai 1	Baterai 2	Baterai 3	Respon Sistem
1	1	1	1	Beban ke baterai 1
2	0	1	1	Beban ke baterai 2, isi ulang baterai 1
3	1	0	1	Beban ke baterai 1, isi ulang baterai 2
4	1	1	0	Beban ke baterai 1, isi ulang baterai 3
5	0	0	1	Beban ke baterai 3, isi ulang baterai 1
6	1	0	0	Beban ke baterai 1, isi ulang baterai 2
7	0	1	0	Beban ke baterai 2, isi ulang baterai 1.
8	0	0	0	Alarm bunyi, isi ulang baterai 1.

1= Baterai $\geq 11,5$ volt

0= Baterai $< 11,5$ volt

Pada tabel 4, nomor 1 adalah kondisi ketika sistem berjalan dengan kondisi awal baterai seluruhnya berada diatas tegangan *setpoint* sehingga baterai satu akan menyuplai beban hingga tegangannya dibawah tegangan *setpoint*. Nomor 2-4 pada tabel 4 adalah kondisi ketika sistem dijalankan dengan kondisi awal salah satu tegangan baterai berada dibawah tegangan *setpoint*, maka baterai yang memiliki tegangan diatas *setpoint* akan menyuplai beban dan baterai yang memiliki tegangan dibawah *setpoint* akan diisi ulang. Sedangkan pada nomor 5-7 adalah kondisi ketika sistem berjalan dengan kondisi awal terdapat dua buah baterai yang berada dalam kondisi dibawah tegangan *setpoint*, maka baterai yang urutannya lebih kecil akan diisi terlebih dahulu (mis. tegangan baterai 1 dan 2 berada dibawah tegangan *setpoint*, maka baterai 1 akan diisi ulang terlebih dahulu) dan baterai yang berada dalam kondisi diatas tegangan *setpoint* akan menyuplai beban. Dan nomor 8 adalah kondisi ketika sistem berjalan dengan kondisi awal seluruh baterai berada dibawah tegangan *setpoint* ataupun tidak ada baterai yang terpasang, maka *buzzer* akan berbunyi menandakan tidak ada baterai yang siap untuk menyuplai beban dan LCD akan menampilkan peringatan seperti pada gambar 17.



Gambar 15. Tampilan peringatan di LCD

Namun, apabila ternyata ada baterai yang terpasang, namun seluruhnya berada di bawah tegangan *setpoint* maka baterai dengan urutan paling kecil akan terlebih dahulu diisi ulang (mis. tegangan baterai 1, 2 dan 3 berada dibawah tegangan *setpoint*, maka baterai 1 akan diisi ulang terlebih dahulu).

Pada percobaan yang dilakukan, ketiga baterai yang digunakan sedang dalam kondisi baik, sehingga ketiganya dalam kondisi *full charge* yang berarti seluruhnya berada diatas tegangan *setpoint*, sehingga baterai 1 akan terlebih dahulu menyuplai beban.



Gambar 16. Arus terukur saat dibebani bohlam lampu



Gambar 17. Tampilan LCD saat baterai 1 di isi ulang dan baterai 2 menyuplai beban

Pada percobaan yang dilakukan, baterai satu terlebih dahulu menyuplai beban hingga baterai melewati batas tegangan *setpoint*, kemudian sistem akan *switching* suplai beban ke baterai dua, dan baterai satu akan di isi ulang dan akan berhenti otomatis saat telah mencapai batas tegangan *setpoint*, seperti yang terlihat pada gambar 20.



Gambar 18. Tampilan LCD saat baterai 1 di isi ulang dan baterai 2 menyuplai beban

Kemudian pada percobaan yang dilakukan, baterai 2 habis terlebih dahulu sebelum baterai 1 mencapai tegangan *setpoint* sehingga baterai 1 tetap dalam proses pengisian ulang, sehingga sistem akan *switching* suplai beban ke baterai 3. Ketika baterai 1 telah mencapai tegangan *setpoint*, maka seperti pada gambar 21 proses pengisian akan dilanjutkan ke baterai 2 dan baterai 1 dalam posisi *stand by*.



Gambar 19. Tampilan LCD saat baterai 2 di isi ulang dan baterai 3 menyuplai beban

Kemudian pada percobaan yang dilakukan, ketika baterai 3 sedang menyuplai beban, baterai 2 telah mencapai tegangan *setpoint*, sehingga baterai 2 *stand by* dan sistem

hanya bekerja untuk menyuplai beban dari baterai 3 saja seperti pada gambar 22.



Gambar 20. Tampilan LCD saat baterai 3 menyuplai beban

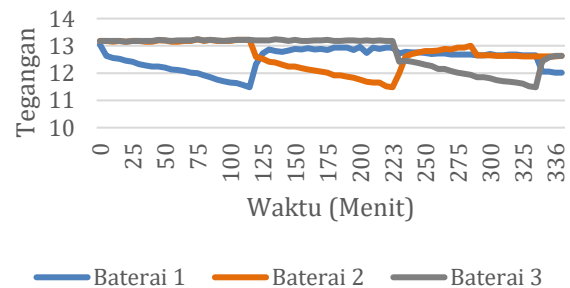
Peralatan akan mengulangi proses yang sama terus menerus, sesuai dengan kondisi baterai dan tetap mengacu pada tabel 4 sebagai aturan utama. Proses sistem pada alat ini akan berhenti jika saklar ditekan atau sumber catu daya sistem hilang. Pada pengujian sistem *switching* baterai, nilai tegangan dibaca oleh Arduino UNO setiap 10 detik yang kemudian dapat ditampilkan pada fitur *serial monitor* di Arduino IDE, namun karena banyaknya data yang diperoleh, maka pada tabel 5, data yang ditampilkan adalah nilai tegangan tiap baterai per 5 menit sekali dan saat proses *switching* terjadi:

Tabel 5. Nilai pembacaan tegangan tiap baterai

No	Waktu	Baterai 1 (V)	Baterai 2 (V)	Baterai 3 (V)
1	17:37:56 (Awal)	13,05 (Stand By)	13,18 (Stand By)	13,18 (Stand By)
2	17:38:26	12,63 (On Load)	13,18 (Stand By)	13,18 (Stand By)
3	17:43:06	12,55 (On Load)	13,15 (Stand By)	13,18 (Stand By)
4	17:48:07	12,52 (On Load)	13,18 (Stand By)	13,18 (Stand By)
5	17:53:07	12,46 (On Load)	13,1 (Stand By)	13,15 (Stand By)
6	17:58:07	12,42 (On Load)	13,18 (Stand By)	13,18 (Stand By)
7	18:03:07	12,33 (On Load)	13,18 (Stand By)	13,18 (Stand By)
8	18:08:08	12,28 (On Load)	13,15 (Stand By)	13,18 (Stand By)
9	18:13:08	12,24 (On Load)	13,15 (Stand By)	13,18 (Stand By)
10	18:18:08	12,24 (On Load)	13,20 (Stand By)	13,22 (Stand By)
11	18:23:08	12,20 (On Load)	13,20 (Stand By)	13,20 (Stand By)
12	18:28:09	12,13 (On Load)	13,15 (Stand By)	13,18 (Stand By)
13	18:33:09	12,11 (On Load)	13,15 (Stand By)	13,20 (Stand By)
14	18:38:09	12,07 (On Load)	13,18 (Stand By)	13,20 (Stand By)
15	18:43:00	12,02 (On Load)	13,18 (Stand By)	13,20 (Stand By)
16	18:48:00	12,00 (On Load)	13,24 (Stand By)	13,24 (Stand By)
17	18:53:00	11,92 (On Load)	13,18 (Stand By)	13,20 (Stand By)
18	19:03:01	11,85 (On Load)	13,22 (Stand By)	13,22 (Stand By)
19	19:08:01	11,76 (On Load)	13,18 (Stand By)	13,20 (Stand By)
20	19:13:02	11,70 (On Load)	13,18 (Stand By)	13,18 (Stand By)

21	19:18:02	11,65 (On Load)	13,18 (Stand By)	13,20 (Stand By)	57	22:18:09	12,68 (Stand By)	12,94 (Charging)	11,98 (On Load)
22	19:23:03	11,63 (On Load)	13,22 (Stand By)	13,22 (Stand By)	58	22:23:09	12,68 (Stand By)	13,00 (Charging)	11,94 (On Load)
23	19:28:03	11,57 (On Load)	13,22 (Stand By)	13,22 (Stand By)	59	22:28:00	12,65 (Stand By)	12,65 (Stand By)	11,85 (On Load)
24	19:33:03	11,48 (Switching)	13,22 (On Load)	13,22 (Stand By)	60	22:33:00	12,65 (Stand By)	12,65 (Stand By)	11,85 (On Load)
25	19:38:04	12,33 (Charging)	12,59 (On Load)	13,20 (Stand By)	61	22:38:00	12,70 (Stand By)	12,65 (Stand By)	11,81 (On Load)
26	19:43:04	12,72 (Charging)	12,52 (On Load)	13,20 (Stand By)	62	22:43:00	12,65 (Stand By)	12,63 (Stand By)	11,74 (On Load)
27	19:53:05	12,87 (Charging)	12,42 (On Load)	13,20 (Stand By)	63	22:48:01	12,65 (Stand By)	12,63 (Stand By)	11,70 (On Load)
28	19:58:05	12,81 (Charging)	12,39 (On Load)	13,24 (Stand By)	64	22:53:01	12,68 (Stand By)	12,63 (Stand By)	11,68 (On Load)
29	20:03:05	12,78 (Charging)	12,31 (On Load)	13,22 (Stand By)	65	22:58:01	12,68 (Stand By)	12,63 (Stand By)	11,65 (On Load)
30	20:08:05	12,83 (Charging)	12,24 (On Load)	13,18 (Stand By)	66	23:03:01	12,65 (Stand By)	12,61 (Stand By)	11,61 (On Load)
31	20:13:06	12,89 (Charging)	12,24 (On Load)	13,22 (Stand By)	67	23:08:02	12,65 (Stand By)	12,61 (Stand By)	11,52 (On Load)
32	20:18:06	12,87 (Charging)	12,18 (On Load)	13,18 (Stand By)	68	23:11:22	12,65 (On Load)	12,61 (Stand By)	11,48 (Switching)
33	20:23:06	12,92 (Charging)	12,13 (On Load)	13,18 (Stand By)	69	23:12:02	12,05 (On Load)	12,61 (Stand By)	12,44 (Charging)
34	20:28:07	12,87 (Charging)	12,09 (On Load)	13,20 (Stand By)	70	23:13:02	12,05 (On Load)	12,61 (Stand By)	12,57 (Charging)
35	20:33:07	12,89 (Charging)	12,05 (On Load)	13,20 (Stand By)	71	23:14:02	12,02 (On Load)	12,61 (Stand By)	12,63 (Charging)
36	20:38:07	12,85 (Charging)	12,02 (On Load)	13,22 (Stand By)	72	23:15:03	12,02 (On Load)	12,63 (Stand By)	12,63 (Charging)
37	20:43:08	12,94 (Charging)	11,92 (On Load)	13,18 (Stand By)					
38	20:48:08	12,94 (Charging)	11,92 (On Load)	13,18 (Stand By)					
39	20:53:08	12,94 (Charging)	11,87 (On Load)	13,20 (Stand By)					
40	20:58:08	12,85 (Charging)	11,83 (On Load)	13,20 (Stand By)					
41	21:03:09	12,96 (Charging)	11,76 (On Load)	13,18 (Stand By)					
42	21:08:09	12,74 (Charging)	11,68 (On Load)	13,20 (Stand By)					
43	21:13:09	12,94 (Charging)	11,65 (On Load)	13,18 (Stand By)					
44	21:18:00	12,89 (Charging)	11,65 (On Load)	13,20 (Stand By)					
45	21:23:00	12,94 (Charging)	11,52 (On Load)	13,18 (Stand By)					
46	21:26:30	12,94 (Charging)	11,48 (Switching)	13,18 (On Load)					
47	21:28:06	12,72 (Stand By)	12,00 (Charging)	12,42 (On Load)					
48	21:33:06	12,78 (Stand By)	12,61 (Charging)	12,46 (On Load)					
49	21:38:06	12,76 (Stand By)	12,70 (Charging)	12,42 (On Load)					
50	21:43:07	12,76 (Stand By)	12,76 (Charging)	12,37 (On Load)					
51	21:48:07	12,74 (Stand By)	12,81 (Charging)	12,31 (On Load)					
52	21:53:07	12,70 (Stand By)	12,81 (Charging)	12,26 (On Load)					
53	21:58:08	12,72 (Stand By)	12,83 (Charging)	12,15 (On Load)					
54	22:03:08	12,72 (Stand By)	12,89 (Charging)	12,15 (On Load)					
55	22:08:08	12,68 (Stand By)	12,87 (Charging)	12,07 (On Load)					
56	22:13:09	12,68 (Stand By)	12,94 (Charging)	12,02 (On Load)					

Dari pengujian yang telah dilakukan, sistem *switching* baterai telah bekerja dengan cukup baik, sesuai dengan apa yang dirancang dalam senarai program. Dari tabel 5 dapat dibuat grafik seperti pada gambar dibawah:



Gambar 21. Grafik proses *Stand By*, *On Load*, dan *Switching* pada baterai 1, 2 dan 3

Dari data yang didapat pada tabel 5, dan juga pada gambar 23 dapat dilihat bahwa proses *switching* dari baterai 1 ke baterai 2 terjadi pada menit ke-115. Nilai tegangan baterai 1 perlahan turun hingga <math><11,5</math> volt selama kurang lebih 115 menit, kemudian baterai 1 akan diputus dari beban dan akan di isi ulang, dapat dilihat kenaikan tegangan pada baterai 1 mulai dari menit ke-115 hingga menit ke-223. Sesaat setelah baterai 1 mencapai tegangan ≥ 13 volt, tegangan baterai akan stabil di kisaran 12,68 volt.

Pada saat baterai 1 diputus dari beban, baterai 2 langsung disambung ke beban, terlihat pada menit ke-115 hingga

menit ke-223 kurang lebih selama 108 menit nilai tegangan baterai 2 turun hingga <11,5 volt, kemudian baterai 2 akan diputus dari beban dan akan diisi ulang, dapat dilihat kenaikan tegangan baterai 2 pada menit ke-223 hingga menit ke-285. Sesaat setelah baterai 2 mencapai tegangan ≥ 13 volt, tegangan baterai akan stabil di kisaran 12,65 volt. Pada saat baterai 2 diputus dari beban, baterai 3 langsung disambung ke beban, terlihat pada menit ke-223 hingga menit ke-330 kurang lebih selama 107 menit baterai 3 mengalami penurunan tegangan hingga melewati 11,5 volt, kemudian baterai 3 akan diputus dari beban dan akan diisi ulang hingga mencapai tegangan ≥ 13 volt.

Pada saat baterai 3 diputus dari beban, baterai 1 langsung disambung ke beban, dan sistem akan berulang terus (*looping*) hingga saklar dimatikan, ataupun sumber catu daya sistem dilepas.

Pada proses *switching* baterai, beban berupa bohlam lampu sama sekali tidak padam dalam proses *switching*, hal ini disebabkan karena proses perpindahan dari kontak NC ke NO pada relai berlangsung sangat cepat, sehingga beban tidak sempat bahkan tidak sampai padam.

4. Kesimpulan

Telah berhasil dibuat suatu sistem *switching* baterai otomatis berbasis Arduino UNO dengan aktuator berupa relai, LCD 2x16 sebagai penampil status tegangan baterai, dan *buzzer* sebagai sumber bunyi peringatan dan 3 buah baterai sebagai media penyimpanan energi listrik. Dalam 10 kali pengujian, sensor tegangan memiliki tingkat akurasi pembacaan tegangan yang cukup akurat dengan rata-rata error sebesar 0,011. Pada pengujian berbeban bohlam lampu 25 watt, baterai 1 menyuplai beban kurang lebih selama 115 menit, dan tegangan turun hingga <11,5 volt, kemudian terjadi *switching* beban ke baterai 2, dan baterai 1 diisi ulang mulai dari menit ke-115 hingga menit ke-223, sesaat setelah baterai 1 mencapai tegangan ≥ 13 volt, tegangan baterai akan stabil di kisaran 12,68 volt. Kemudian baterai 2 menyuplai beban kurang lebih selama 108 menit, dan tegangan turun hingga <11,5 volt, kemudian terjadi *switching* beban ke baterai 3, dan baterai 2 diisi ulang mulai dari menit ke-223 hingga menit ke-28, sesaat setelah baterai 2 mencapai tegangan ≥ 13 volt, tegangan baterai akan stabil di kisaran 12,65 volt. Kemudian baterai 3 menyuplai beban selama kurang lebih 107 menit, dan tegangan turun hingga <11,5 volt, kemudian terjadi *switching* beban ke baterai 1, dan baterai 3 diisi ulang hingga mencapai tegangan ≥ 13 volt. Sistem akan bekerja terus (*looping*) hingga saklar dimatikan, ataupun sumber catu daya sistem dilepas. Pada proses *switching* baterai, beban berupa bohlam lampu sama sekali tidak padam dalam proses *switching*, hal ini disebabkan karena proses perpindahan dari kontak NC ke NO pada relai berlangsung sangat cepat, sehingga beban tidak sempat bahkan tidak sampai padam. Sistem ini dapat

dikembangkan lagi dengan menambahkan logika *fuzzy* dan fitur monitoring berbasis Android ataupun IoT

Referensi

- [1]. I. Rahardjo and I. Fitriana, "Analisis Potensi Pembangkit Listrik Tenaga Surya di Indonesia".
- [2]. Lampiran I Peraturan Presiden Republik Indonesia Nomor 22 Tahun 2017.
- [3]. Admin, "Matahari Untuk PLTS di Indonesia," [Online]. Tersedia: <https://www.esdm.go.id/id/media-center/arsip-berita/matahari-untuk-plts-di-indonesia> [Diakses 9 September 2020].
- [4]. Bachtiar, Muhammad, "Prosedur Perancangan Sistem Pembangkit Listrik Tenaga Surya untuk Perumahan (*Solar Home System*)", SMARTek, vol.4, no.3, pp. 176-182, 2006.
- [5]. S. M. Suroso, I. Setiawan, dan B. Winardi, "PERANCANGAN INVERTER SATU FASA OFF-GRID MENGGUNAKAN DSPIC30F4011 DENGAN KONTROL ARUS METODE PROPORTIONAL RESONANT," *Transient: Jurnal Ilmiah Teknik Elektro*, vol. 7, no. 3, pp. 753-760, Mei. 2019
- [6]. M. P. Aidina, "System Control Switching Automatic Charging Pada Pompong Listrik Hybrid," Laporan Tugas Akhir, Politeknik Negeri Batam, Program Studi Teknik Elektronika, 2014.
- [7]. M. A. P. Pradana, "Kontrol Pengisian Baterai Otomatis Pada Sistem Pembangkit Listrik Alternatif," Laporan Tugas Akhir, Universitas Sanata Dharma, Fakultas Sains dan Teknologi, Jurusan Teknik Elektro, Program Studi Teknik Elektro, 2015.
- [8]. J. P. Dunlop, *Batteries and Charge Control in Stand-Alone Photovoltaic Systems Batteries and Charge Control in Stand-Alone Photovoltaic Systems Fundamentals and Application*. Florida: Florida Solar Energy Center, 1997.
- [9]. Admin, "Baterai VRLA," [Online]. Tersedia: <https://galanghakim.wordpress.com/2016/10/11/baterai-vrla/> [Diakses 9 September 2020].
- [10]. R. H. Sianipar, "Analisis Rangkaian Listrik Sederhana," *Rangkaian Listrik*, Bandung: Rekayasa Sains, 2015, 45-47.
- [11]. Abdul Kadir, *Paduan Praktis Mempelajari Aplikasi Mikrokontroler dan Pemrogramannya menggunakan Arduino*, Jogjakarta: Andi, 2012.
- [12]. Admin, "pengertian buzzer" [Online]. Tersedia: <https://indrahajra.wordpress.com/2012/01/07/pengertian-buzzer/> [Diakses 9 September 2020].
- [13]. Admin, "LCD (Liquid Cristal Display)," [Online]. Tersedia: <https://elektronika-dasar.web.id/lcd-liquid-cristal-display/> [Diakses 9 September 2020].
- [14]. Khoiril Iman, "LCD dengan I2C Module untuk Arduino," [Online]. Tersedia: <https://khoiruliman.wordpress.com/2016/06/07/lcd-dengan-i2c-module-untuk-arduino/> [Diakses 9 September 2020].
- [15]. M. H. Widiyanto, "Proteus sebagai aplikasi software pengendali mikrokontroler," [Online]. Tersedia: <https://binus.ac.id/bandung/2020/03/proteus-sebagai-aplikasi-software-pengendali-mikrokontroler/> [Diakses 9 September 2020].