

Rancang Bangun Sistem Monitoring Daya Baterai Pembangkit Listrik Tenaga Surya (PLTS) dan Kontrol Beban Berbasis *Internet of Things*

Riza Alfita*, Koko Joni, Fajar Dwika Darmawan

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Trunojoyo Madura
Jl. Raya Telang, PO.BOX 02 Kamal, Bangkalan, Indonesia 69162

Abstrak

Teknologi *Internet Of Things* dalam penelitian ini dimanfaatkan pada Pembangkit Listrik Tenaga Surya (PLTS) (Studi Kasus : Prodi Teknik Elektro Universitas Trunojoyo Madura) sebagai sistem monitoring daya baterai dan kontrol beban. Semua sistem tersebut dibangun dengan tujuan mempermudah pengguna dalam mengelola konsumsi daya yang digunakan sekaligus mencegah kerusakan baterai sehingga masa pakai baterai dapat bertahan lebih lama dan penggunaan PLTS menjadi lebih optimal. Semua sistem tersebut dirancang menggunakan beberapa komponen yang saling terintegrasi dengan fungsi masing-masing seperti Raspberry sebagai pengolah data, smartphone sebagai interface, dan sensor aktuator sebagai input output. Dari hasil pengujian akurasi monitoring didapatkan nilai error rata-rata sebesar 1,56 %. Setelah memastikan sistem memiliki tingkat akurasi yang tinggi, dilakukan pengujian charge-discharge secara real time selama 7 hari yang menunjukkan bahwa sistem bekerja sesuai tujuan penelitian yang dibuktikan dengan tidak adanya penggunaan daya melebihi batas standar SOC baterai yang digunakan sebesar 30%. Sedangkan untuk pengujian sistem kontrol didapatkan bahwa koneksi wifi memiliki rata-rata delay paling cepat selama 10,30 s, provider A 11,17 s, dan provider B 12,60 s.

Kata kunci: *Internet of Things; monitoring; kontrol; daya; Pembangkit Listrik Tenaga Surya*

Abstract

[Title: Design of Monitoring Battery Solar Power Plant and Load Control System based Internet of Things] *Internet of Things* technology in this research is utilized on solar power plant (Case Study: Electrical Engineering Department of Trunojoyo Madura University) as a battery power monitoring and load control system. All of these systems were built to make it easier for users to manage the power consumption while preventing battery damage so that lifetime can last longer and the use of PLTS than more optimal. All of these systems are designed to use several integrated components with their respective functions, including Raspberry as a data processing, smartphone as an interface, and sensors actuator as input-output. From the results of the monitoring accuracy test, the average error value is 1.56%. After ensuring the system has a high level of accuracy, The charge-discharge test is conducted in real-time for 7 days, which shows that the system works according to the research objectives as evidenced by the nothingness of power consumption exceeding the SOC standard limit battery used by 30%. Meanwhile, for the control system test, the wifi connection has the fastest average delay for 10,30 s, provider A 11,17 s, and provider B 12,60 s.

Keywords: *Internet of Things; monitoring; control; power; solar power plant*

1. Pendahuluan

Letak geografis Indonesia yang berada di ekuator

menjadikan Indonesia salah satu negara yang memiliki nilai surplus sinar matahari. Berdasarkan hal tersebut, sejatinya energi surya diyakini sangat potensial untuk dikembangkan, salah satunya dimanfaatkan sebagai

*) Penulis Korespondensi.

E-mail: riza.alfita@trunojoyo.ac.id

sumber energi pada Pembangkit Listrik Tenaga Surya (PLTS) (Margana, 2019).

PLTS pembangkit yang mengkonversikan energi foton dari surya menjadi energi listrik. Konversi ini terjadi pada panel surya yang terdiri dari sel-sel *photovoltaic*. Sel-sel ini terdiri dari lapisan-lapisan tipis dari silikon (Si) murni dan bahan semikonduktor lainnya. PLTS memanfaatkan energi surya secara langsung untuk menghasilkan listrik DC yang disimpan pada baterai untuk kemudian dengan bantuan *inverter* dapat diubah menjadi listrik AC (Nugraha, Giriantari, & Kumara, 2013).

PLTS sudah mulai berkembang baik skala besar hingga skala rumah tangga. Jika PLTS skala besar secara berkala ada seorang ahli atau teknisi yang bertugas melakukan *maintenance* dan monitoring, maka untuk PLTS skala rumah tangga umumnya hanya melakukan pemakaian tanpa mengetahui berapa kapasitas daya dari baterai yang masih bisa digunakan. *Overdischarge* menjadi salah satu penyebab utama baterai sebagai komponen penyimpan daya memiliki masa pakai yang tidak tahan lama karena seringkali masih digunakan disaat kapasitas sudah pada batas SOC (Salkind *dkk.*, 2001). SOC (*State of Charge*) didefinisikan sebagai rasio total kapasitas energi yang dapat digunakan dari sebuah baterai dengan kapasitas baterai seluruhnya. SOC menggambarkan energi yang tersedia dan dituliskan dalam persentase rentang 0% hingga 100% (Abdi *dkk.*, 2017)

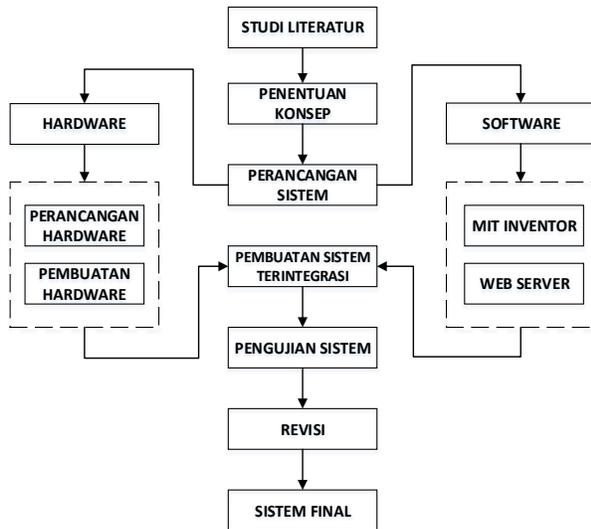
Beberapa penelitian terkait monitoring daya listrik sudah dilakukan. Diantaranya adalah monitoring parameter panel surya menggunakan Arduino dengan output berisi informasi mengenai arus dan tegangan berupa *spreadsheet excel* (Fachri, Sara & Away, 2015), SMS *gateway* (Fitriandi, Komalasari & Gusmadi, 2016) dan format TXT (Suryawinata, Purwanti & Sunardiyo, 2017). Informasi atau data didapatkan melalui sensor yang terhubung pada Arduino untuk diolah dan dihasilkan output data berupa *spreadsheet excel*, TXT, atau SMS *gateway*. Sedangkan pemanfaatan *Internet of Things* untuk monitoring arus dan tegangan yang disimpan dan ditampilkan pada web server Thingspeak diusulkan oleh Aritonang, Maison, Hais (2020).

Walaupun secara prinsip kerja semuanya memiliki kesamaan, tapi perbedaan jenis data informasi pada output menjadikan kinerja yang juga beragam. Jika berdasarkan tujuan penelitian yang membutuhkan data informasi secara *real time* maka pemanfaatan *Internet of Things* dirasakan cocok dan efisien. Selain itu *Internet of Things* tidak membutuhkan tambahan perangkat keras lainnya, jika dibandingkan output berupa *spreadsheet*

excel dan TXT yang masih membutuhkan sebuah perangkat keras komputer untuk bekerja serta menyimpan data. Sama halnya dengan SMS *gateway* yang masih membutuhkan perangkat lain berupa modul GSM *shield* serta hanya bekerja sesuai interval yang diinginkan.

Paper ini memaparkan penelitian yang dilakukan untuk membantu pengguna dalam mengatur konsumsi daya yang digunakan sekaligus mencegah kerusakan sehingga masa pakai baterai bisa bertahan lebih lama serta pemakaian PLTS lebih maksimal. Jika penelitian sebelumnya *Internet of Things* yang dimanfaatkan untuk monitoring baterai berupa data pada web server (Rohman & Iqbal, 2016), maka pada penelitian ini dikembangkan berupa aplikasi *smartphone*. Aplikasi pada *smartphone* yang terintegrasi dengan web server dan pengolah data akan menampilkan persentase baterai secara *real time* dan notifikasi alarm pemakaian sehingga pengguna akan mengetahui kapasitas baterai masih bisa digunakan atau tidak sesuai batas SOC sebesar 30%. Untuk mengetahui kapasitas tersebut diperoleh dari sensor tegangan 0-25 V sebagai pengambil data untuk diteruskan pada Raspberry Pi. Sebagai pusat pengolahan data, Raspberry Pi 3B dipilih karena kehandalan dan tetap mampu bekerja secara otomatis tanpa perlu melakukan pemrograman ulang ketika *reboot*. Raspberry Pi 3B sudah terhubung konektivitas jaringan internet tanpa memerlukan modul *wifi* tambahan seperti halnya modul ESP pada Arduino. Selain itu juga dilengkapi slot SD *Card* untuk menyimpan data, baik itu data sistem operasi ataupun hasil pengolahan (Rivaldo, Bustami & Siswanto, 2020).

Kelalaian pengguna dalam menggunakan beban ketika tidak terpakai juga mengakibatkan masa pakai baterai tidak bertahan lama, terlebih malam hari ketika tidak ada daya yang masuk pada baterai. Muzawi, Efendi & Sahrin (2018) membuat sebuah prototype kendali beban berupa lampu memanfaatkan teknologi *Internet of Things*, proses pengendalian tersebut dilakukan menggunakan jaringan lokal melalui web server. Menurut rekomendasi ITU-T (*International Telecommunication Union*) Y.2060 (2012), *Internet of Things* didefinisikan sebagai sebuah penemuan yang mampu menyelesaikan permasalahan yang ada melalui penggabungan teknologi dan dampak sosial, sementara itu jika ditinjau dari standarisasi secara teknik *Internet of Things* dapat digambarkan sebagai infrastruktur global untuk memenuhi kebutuhan masyarakat yang memungkinkan layanan canggih dengan interkoneksi secara fisik dan virtual berdasarkan perkembangan teknologi informasi dan komunikasi yang telah ada.

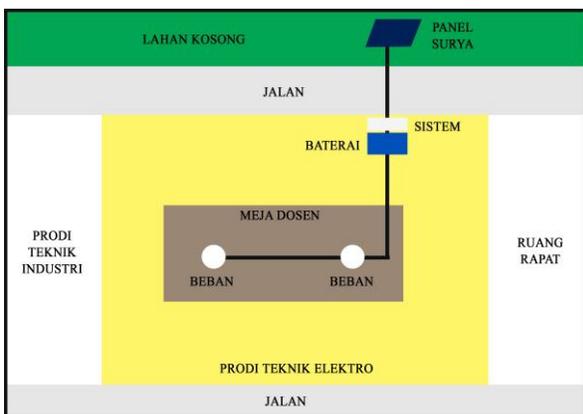


Gambar 1. Metode penelitian

Pengembangan dari penelitian yang sudah ada dilakukan dengan membuat kendali melalui aplikasi *smartphone* yang terintegrasi dengan web server dan pusat pengolahan data. Penambahan sistem kendali atau kontrol beban bertujuan untuk efisiensi waktu dikarenakan mampu mengaktifkan atau menonaktifkan beban dimanapun pengguna berada yang nantinya akan lebih memperpanjang masa pakai baterai apabila terjadi kelalaian dalam pemakaian beban. Seiring dengan penggunaan PLTS pada Ruang Prodi Teknik Elektro UTM, maka sistem yang dihasilkan akan dimanfaatkan secara *real* sesuai tujuan penelitian ini dirancang.

2. Metode Penelitian

Dalam penelitian ini menggunakan metode R&D (*Research and Development*). Metode R&D adalah metode penelitian yang digunakan untuk menghasilkan



Gambar 2. Denah penerapan sistem

produk tertentu dan menguji keefektifan sebuah produk (sistem) tersebut (Sugiyono, 2011). Adapun untuk prosedur metode penelitian seperti yang ditunjukkan pada Gambar 1.

2.1 Studi Literatur

Studi literatur digunakan untuk menemukan serta mempelajari konsep dan landasan teoritis yang menunjang terkait penelitian yang akan dibuat. Studi literatur juga akan memberikan gambaran hasil-hasil penelitian terdahulu yang relevan sebagai bahan perbandingan untuk mengembangkan penelitian tentang perancangan dan pembuatan sistem monitoring daya baterai PLTS dan kontrol beban.

2.2 Penentuan Konsep

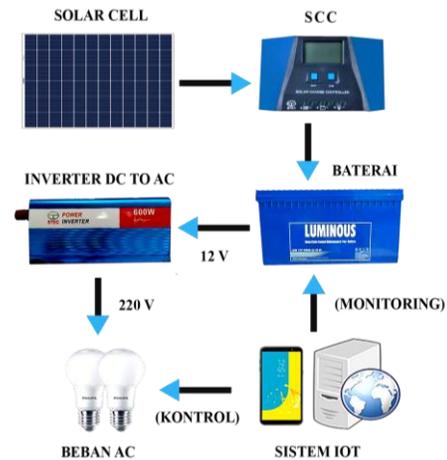
Tahap ini dilakukan untuk menentukan konsep serta tujuan berdasarkan literasi-literasi yang dijadikan acuan perancangan. Adapun denah penerapan dan konsep sistem masing-masing dijelaskan pada Gambar 2 dan Gambar 3.

2.2.1. Konsep Sistem Monitoring

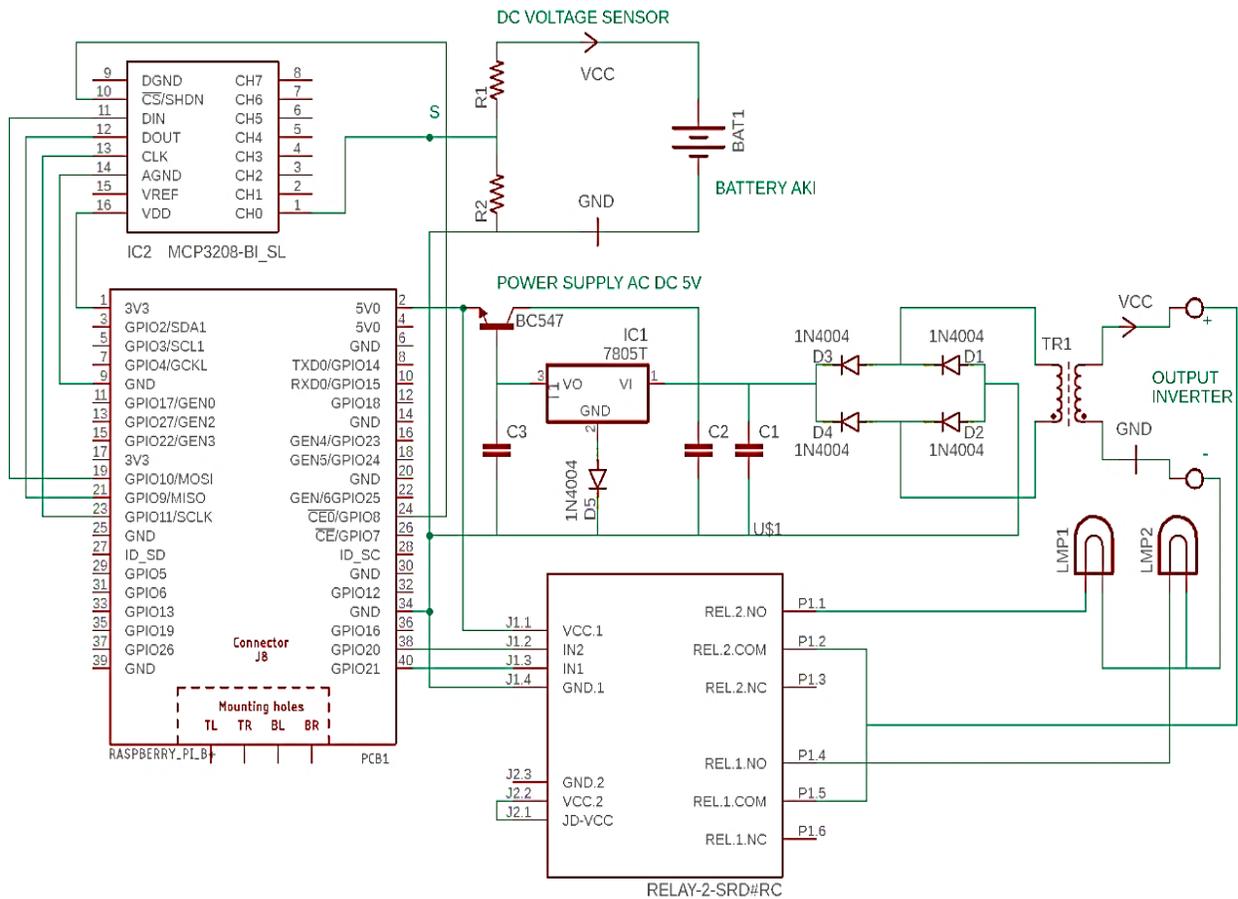
Sistem ini dibangun untuk memberi informasi mengenai sisa daya yang masih mengalir pada baterai panel surya dalam bentuk persentase (%) dan notifikasi peringatan pemakaian dengan *interface* pada sebuah aplikasi *smartphone*. Tujuan dari sistem monitoring adalah membantu pengguna dalam mengatur konsumsi daya yang digunakan sehingga masa pakai baterai bisa bertahan lebih lama serta pemakaian PLTS lebih maksimal.

2.2.2. Konsep Sistem Kontrol

Sistem ini dibuat untuk mengontrol beban listrik menggunakan aplikasi *smartphone*. Sistem ini juga dapat meminimalisir penggunaan saklar listrik karena kendali sudah dapat diakses pada sebuah aplikasi. Kelebihan dari sistem ini adalah dapat mengaktifkan serta menonaktifkan beban secara manual melalui *smartphone* dari jarak jauh jika pengguna lalai dalam melakukan pemakaian beban yang akan mengakibatkan



Gambar 3. Konsep sistem



Gambar 4. Rangkaian elektronik

berkurangnya masa pakai baterai karena *overdischarge*.

2.3 Perancangan Sistem

Perancangan sistem pada penelitian terdapat 2 tahap bagian yaitu perancangan *hardware* dan *software*. Perancangan *hardware* bertujuan untuk menganalisa, menyiapkan kebutuhan, serta merancangya sebelum masuk pada tahap pembuatan sistem. Sedangkan perancangan *software* bertujuan untuk menyiapkan kebutuhan *software* sebelum diintegrasikan dengan *hardware*.

2.3.1. Perancangan Hardware

Berdasarkan analisa kebutuhan, komponen-komponen yang digunakan diantaranya baterai VRLA (*Valve Regulated Lead Acid*) Luminous 200 Ah untuk menyimpan dan mengalirkan daya dari PLTS sekaligus komponen yang akan dimonitoring, Raspberry Pi 3B sebagai pusat pengolahan data, sensor tegangan 0-25 V untuk mengambil data tegangan baterai, *microchip* MCP3208 untuk merubah data analog sensor tegangan menjadi data digital, *relay 2 channel* sebagai aktuator untuk mengaktifkan dan menonaktifkan beban, *power supply* 5 V sebagai catu daya keseluruhan sistem, dan *smartphone* sebagai *interface* monitoring daya baterai

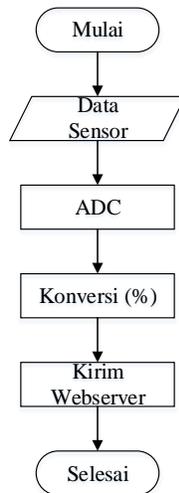
dan pusat kontrol beban. Semua komponen yang digunakan tersebut dirangkai menjadi satu kesatuan sesuai pada Gambar 4.

2.3.2. Perancangan Software

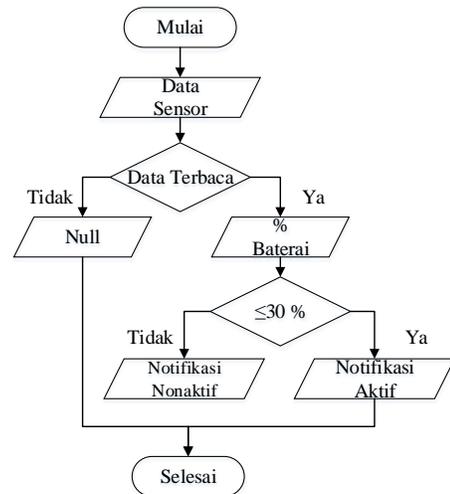
Perancangan *software* meliputi perencanaan web server melalui Thingspeak. Thingspeak merupakan web server berbasis *open source Internet of Things* aplikasi dan API untuk menyimpan dan mengambil data yang beragam dengan keluaran berupa grafik (Pasha, 2016). Pada penelitian ini Thingspeak mengambil, menyimpan, menganalisis, mengamati dan bekerja pada data yang didapatkan dari sensor yang terhubung ke pengolah data. Sedangkan pembuatan *interface* berupa aplikasi yang dibuat menggunakan MIT App Inventor. MIT App Inventor adalah *platform* online yang dirancang untuk mengajarkan konsep pemikiran komputasi melalui pengembangan aplikasi seluler (Patton, Tissenbaum, & Harunani, 2019). Pada aplikasi yang dirancang terdapat 2 *main interface* didalamnya yaitu monitoring, dan kontrol.

1. Monitoring

Alur Gambar 5 dimulai dari sensor tegangan sebagai input mendapatkan sebuah data berupa analog. Data tersebut dirubah menjadi digital memanfaatkan



Gambar 5. Flowchart monitoring (Raspberry)



Gambar 6. Flowchart monitoring (Aplikasi)

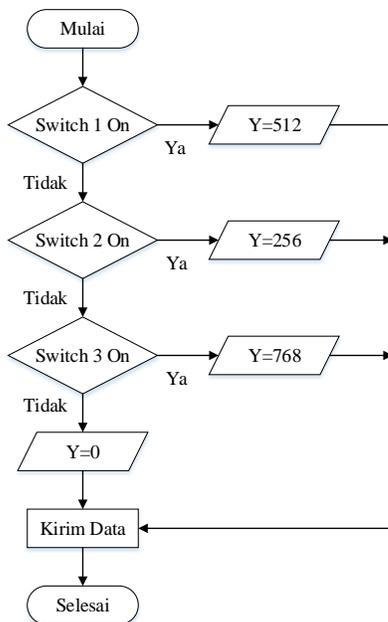
ADC. Data yang sudah berupa digital diolah oleh Raspberry Pi menggunakan rumus yang sudah ada di dalamnya. Hasil pengolahan berupa persentase selanjutnya dikirim secara *real time* melalui web server pada sebuah *interface* aplikasi yang sudah dibuat sebelumnya.

Gambar 6 menjelaskan bahwa sistem aplikasi terlebih dahulu menerima data dari web server melalui alamat *api key* yang digunakan. Jika data belum dikirim oleh web server maka aplikasi akan menampilkan karakter “null”. Namun jika sudah dikirim maka akan terbaca dan menampilkan data *real time* berupa

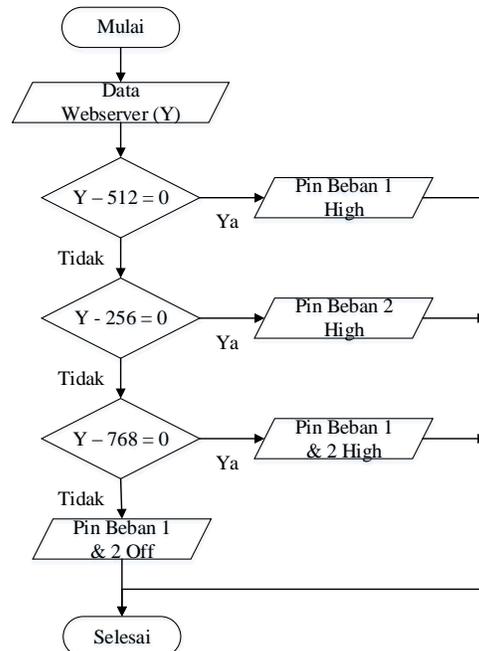
persentase yang menandakan sistem bekerja dengan baik. Selanjutnya jika persentase yang terbaca berada pada batas atau dibawah SOC (30%), maka akan menampilkan notifikasi peringatan. Namun apabila persentase daya baterai terbaca masih diatas batas SOC, maka notifikasi peringatan nonaktif.

2. Kontrol

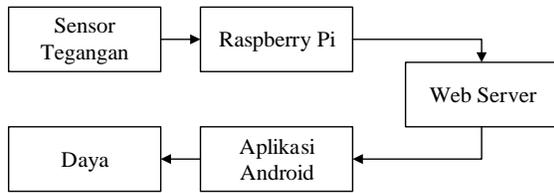
Alur pada Gambar 7 dimulai ketika pengguna menginputkan perintah (*switch*) pada aplikasi. Aplikasi akan mengirimkan data perintah yang berupa inialisasi nilai tersebut pada Raspberry melalui web server. Terdapat 3 inialisasi nilai untuk masing-masing *switch*,



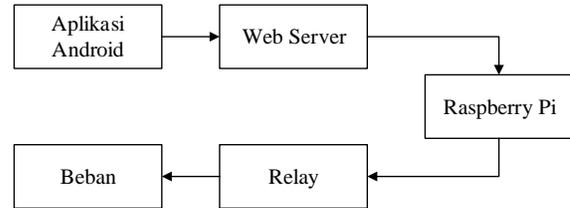
Gambar 7. Flowchart kontrol (Aplikasi)



Gambar 8. Flowchart kontrol (Raspberry)



Gambar 9. Diagram alur sistem monitoring



Gambar 10. Diagram alur sistem kontrol

yaitu *switch* 1 dengan nilai inialisasi 512, *switch* 2 dengan nilai 256, dan *switch* 3 dengan nilai 768. Sesuai dengan alur Gambar 8, Raspberry Pi akan membaca dan mengolah data yang sudah diterima untuk selanjutnya akan mengirim perintah kepada relay sebagai aktuator. Jika data yang masuk dikurangi 512 bernilai 0, maka pin beban 1 bernilai *high* dan mengaktifkan beban 1. Jika data yang masuk dikurangi 256 bernilai 0, maka pin beban 2 bernilai *high* dan mengaktifkan beban 2. Jika data yang masuk dikurangi 768 bernilai 0, maka pin beban 1 dan 2 bernilai *high* dan mengaktifkan beban 1 dan 2 secara bersamaan.

2.4 Pembuatan Sistem Terintegrasi

Tahap ini merupakan tahap dimana hasil perancangan *hardware* diintegrasikan dengan hasil perancangan *software*. Berdasarkan diagram alur sistem monitoring pada Gambar 9, sensor tegangan sebagai input akan mengambil data tegangan dari baterai yang digunakan pada PLTS. Proses baru akan dijalankan apabila data dari sensor tegangan sudah diterima oleh Raspberry Pi. Selanjutnya data tersebut diolah dan hasil pengolahan akan dikirim ke aplikasi *smartphone* melalui penyedia layanan *Internet of Things* berupa web server. Proses tersebut akan dijalankan terus menerus secara *real time* namun untuk mengirim data koneksi internet

harus stabil. Output yang dihasilkan yaitu berupa persentase sisa daya baterai dan notifikasi peringatan apabila mencapai batas SOC yang ditampilkan melalui *interface* aplikasi *smartphone*.

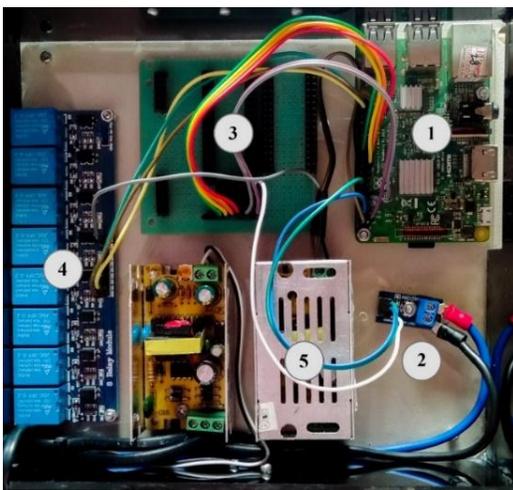
Seperti yang ditunjukkan diagram alur sistem kontrol pada Gambar 10, aplikasi *smartphone* sebagai pengontrol akan mengaktifkan atau menonaktifkan *switch* yang ada sesuai perintah untuk diteruskan kepada web server. Proses baru akan dijalankan ketika web server sudah menerima data perintah dari aplikasi *smartphone* untuk selanjutnya dikirim kepada Raspberry Pi. Kemudian data tersebut akan diolah untuk memberikan perintah kepada relay sebagai aktuator beban. Output yang dihasilkan berupa beban yang aktif atau nonaktif sesuai perintah. Hasil dari keseluruhan sistem yang telah terintegrasi berupa *hardware* seperti pada Gambar 11 dan sebuah aplikasi seperti pada Gambar 12.

2.5 Pengujian Sistem

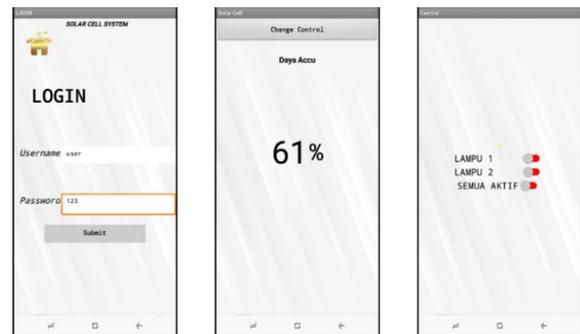
Setelah alat (*hardware*) dan aplikasi (*software*) terintegrasi, tahap selanjutnya yang dilakukan adalah pengujian masing-masing sistem. Dari pengujian yang dilakukan akan didapatkan analisa serta kesimpulan mengenai kinerja dari keseluruhan sistem.

3. Hasil dan Pembahasan

Seluruh sistem yang telah selesai dirancang selanjutnya akan melalui tahap pengujian. Pengujian yang dilakukan diantaranya adalah fungsionalitas dan kehandalan sistem, pembebanan, notifikasi peringatan, *charge-discharge*, dan delay kontrol.



Gambar 11. Hardware terintegrasi



Gambar 12. Interface aplikasi terintegrasi

Tabel 1. Akurasi sistem monitoring

Data	Aplikasi	Perhitungan	Error
1	39 %	38 %	2,56 %
2	46 %	45 %	2,17 %
3	57 %	58 %	2,56 %
4	62 %	62,5 %	0,8 %
5	65 %	66 %	1,5 %
6	70 %	72 %	2,7 %
7	78 %	77,5 %	0,64 %
8	84 %	82 %	2,38 %
9	83 %	83 %	0 %
10	99 %	99 %	0 %
Rata-Rata Error			1,53 %

3.1 Pengujian Fungsionalitas dan Kehandalan Sistem

Pengujian fungsionalitas sistem dilakukan dengan menjalankan aplikasi untuk memastikan bahwa sistem berjalan sesuai dengan fungsinya masing-masing. Pengujian meliputi aplikasi antarmuka pada monitoring dan kontrol.

3.1.1 Monitoring

Pengujian fungsionalitas sistem monitoring dilakukan untuk memastikan sistem bekerja sesuai fungsinya dan mengetahui tingkat akurasi dengan cara membandingkan hasil nilai pada aplikasi dan perhitungan manual. Nilai pada aplikasi didapatkan dari proses yang terintegrasi antara Raspberry, web server, dan aplikasi dengan input data dari sensor tegangan yang terhubung dengan baterai. Sedangkan nilai hasil perhitungan manual didapatkan dari perhitungan menggunakan Persamaan 1 dengan input data dari avometer (voltmeter) yang terhubung dengan baterai. Hasil dari perbandingan didapatkan sebuah nilai error dalam bentuk persentase. Untuk mengetahui persentase error antara hasil nilai pada aplikasi dengan perhitungan manual menggunakan Persamaan 2.

$$Persentase = \frac{DT - V_{min}}{V_{max} - V_{min}} \times 100\% \quad (1)$$

Di mana DT merupakan data tegangan baterai (V); Vmin adalah tegangan batas bawah baterai (10,5 V); dan Vmax adalah tegangan batas atas baterai (13,8 V).

$$Error = \frac{N_{max} - N_{min}}{N_{max}} \times 100\% \quad (2)$$

Di mana Nmax merupakan nilai terbesar diantara data perbandingan; dan Nmin adalah nilai terkecil diantara data perbandingan.

Berdasarkan hasil pengujian dari 10 kali sampel yang tertera pada Tabel 1, pengambilan data didapatkan error terbesar senilai 2,7 % dan terkecil senilai 0 %.

Tabel 2. Input pengujian sistem kontrol

Data	Switch 1	Switch 2	Switch 3
1	On	Off	Off
2	On	On	Off
3	Off	On	Off
4	Off	Off	On
5	Off	Off	On
6	Off	Off	Off
7	Off	On	Off
8	On	On	Off
9	Off	Off	On
10	On	Off	Off

Sedangkan untuk rata-rata error senilai 1,53 %. Dengan rendahnya nilai error tersebut, maka fungsionalitas sistem monitoring akan valid sebagai acuan apabila sistem diterapkan secara real.

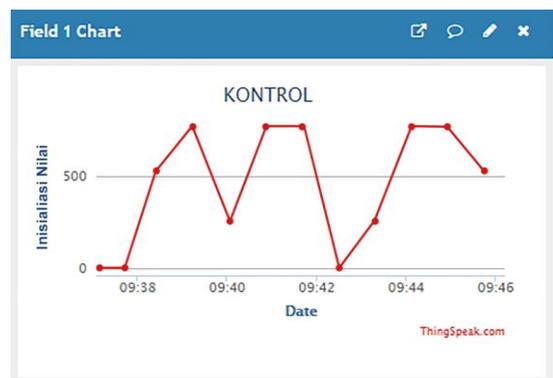
3.1.2 Kontrol

Pengujian fungsionalitas sistem kontrol dilakukan dengan tujuan mengetahui bahwa kinerja sistem sudah bekerja sesuai fungsi yang diharapkan. Hasil pengujian diperoleh data berupa Tabel 2 dan grafik thingspeak pada Gambar 13.

Berdasarkan hasil pengujian dari 10 kali sampel pengambilan data secara acak didapatkan tingkat keberhasilan mencapai 100%. Tingkat akurasi keberhasilan tersebut diperoleh dari data tabel 2 sebagai input dan gambar 13 sebagai output telah menunjukkan kesesuaian, seperti halnya ketika switch 1 aktif maka inialisasi nilai pada grafik akan bernilai 512, switch 2 aktif bernilai 256, dan switch 3 aktif bernilai 768. Dengan tingkat akurasi tersebut maka bisa disimpulkan bahwa masing-masing switch pada aplikasi valid untuk dijadikan acuan keberhasilan penerapan secara langsung.

3.2 Pengujian Pembebanan

Pengujian dilakukan dengan menguji daya baterai yang dibebani beban sebesar rentang 86,15 – 91,30 W



Gambar 13. Grafik output kontrol thingspeak

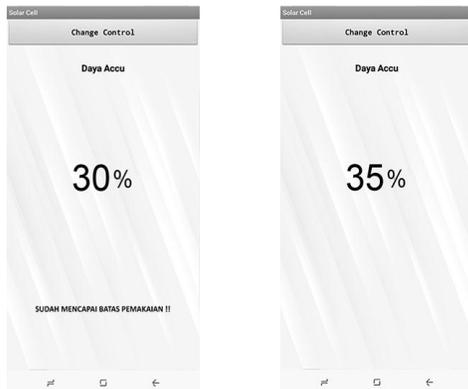
Tabel 3. Pengujian ketika pembebanan

Tanggal	Waktu	Persentase	Tegangan
02.02.2020	21.30.00	41 %	11,88
02.02.2020	21.40.00	40 %	11,84
02.02.2020	21.50.00	39 %	11,8
02.02.2020	22.00.00	38 %	11,76
02.02.2020	22.10.00	37 %	11,72
02.02.2020	22.20.00	36 %	11,68
02.02.2020	22.30.00	35 %	11,64
Perubahan Daya Baterai			- 6 %

selama 60 menit. Hasil pengujian diperoleh data seperti yang tertera pada tabel 3. Berdasarkan hasil pengujian pada Tabel 3, terdapat perubahan daya baterai yang mulanya 41 % turun menjadi 35 %. Terjadinya penurunan tersebut diakibatkan adanya pembebanan yang konstan disaat bersamaan tidak adanya daya yang masuk pada baterai karena pengujian dilakukan pada malam hari. Dengan kondisi dan besar beban yang sama, maka diperkirakan baru akan mencapai batas SOC setelah 11 jam pembebanan apabila daya awal baterai 100%. Namun jika pembebanan lebih besar maka baterai akan lebih cepat mencapai batas SOC.

3.3 Pengujian Notifikasi Peringatan

Penelitian ini dibangun dengan tujuan untuk membantu pengguna dalam mengatur konsumsi daya yang digunakan sekaligus mencegah kerusakan sehingga masa pakai baterai bisa bertahan lebih lama serta pemakaian PLTS lebih maksimal. Untuk mencapai tujuan tersebut maka pada sistem ini selain menampilkan persentase daya baterai, juga akan menampilkan notifikasi peringatan apabila pemakaian sudah melebihi batas SOC. Notifikasi peringatan tersebut berfungsi untuk mengingatkan pengguna agar tidak melakukan pemakaian atau pembebanan yang berlebihan.



Gambar 14. Sampel pengujian peringatan notifikasi

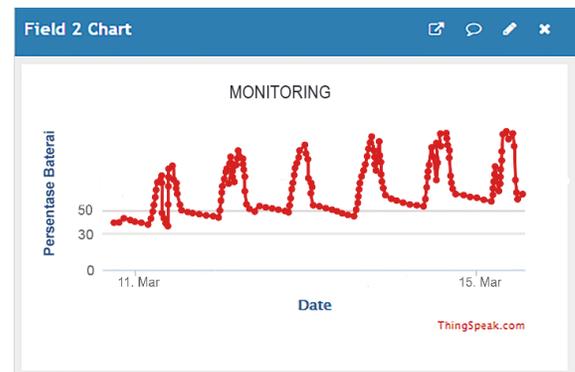
Tabel 4. Pengujian notifikasi peringatan

Data	Tegangan	Persentase	Notifikasi
1	11,84	40 %	-
2	11,64	35 %	-
3	11,52	31 %	-
4	11,49	30 %	Aktif
5	11,46	29 %	Aktif
6	11,6	33 %	-
7	11,68	36 %	-
8	11,49	30 %	Aktif
9	11,84	40 %	-
10	11,98	45 %	-

Pengujian notifikasi peringatan dilakukan sebanyak 10 sampel persentase daya baterai. Batas SOC sudah ditentukan diangka 30%, oleh karenanya jika persentase daya baterai sama atau lebih dari batas tersebut akan mengakibatkan aktifnya notifikasi peringatan seperti yang ditunjukkan pada Tabel 4 dan Gambar 14.

3.4 Pengujian Charge-Discharge

Pengujian *charge-discharge* dilakukan secara *real time* di Ruang Prodi Teknik Elektro Universitas Trunojoyo Madura dengan estimasi 7 hari mulai tanggal 10 Maret 2020 pukul 17.30 WIB hingga tanggal 16 Maret 2020 pukul 16.30 WIB. Grafik pada Gambar 15 menunjukkan adanya fluktuasi naik turun daya baterai yang disebabkan aktifitas masuknya daya dari panel surya dan keluarnya daya melalui beban. Kenaikan daya baterai terjadi seiring dengan masuknya daya dari panel disaat bersamaan tidak ada pembebanan atau kecilnya pembebanan. Sebaliknya, daya baterai akan mengalami penurunan ketika kecil atau tidak adanya daya dari panel yang masuk disaat bersamaan ada pembebanan. Daya baterai akan stabil ketika tidak adanya daya yang masuk dari panel begitupun dengan pembebanan. Selain itu, berdasarkan grafik pengujian *charge-discharge* selama 7



Gambar 15. Grafik *output* monitoring thingspeak

hari tersebut menunjukkan bahwa sistem bekerja sesuai tujuan penelitian yang dibuktikan dengan tidak adanya penggunaan daya melebihi batas standar SOC baterai yang digunakan sebesar 30%.

3.5 Pengujian Delay Kontrol

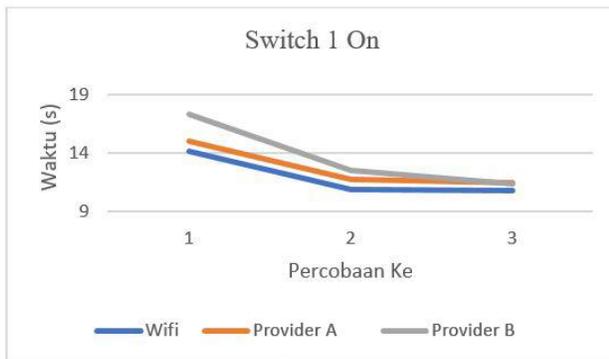
Kehandalan sistem juga diukur dari delay waktu ketika sistem bekerja. Pengujian dilakukan dengan cara menguji setiap masing-masing *switch* yang ada pada aplikasi. Terdapat 3 *switch* yang di uji dengan masing-masing menggunakan 3 koneksi sebagai perbandingan. 3 koneksi tersebut yaitu koneksi *wifi* yang ada pada ruang prodi teknik elektro, koneksi *mobile data provider A* , dan koneksi *mobile data provider B*. Data delay waktu diperoleh dari selisih waktu ketika *switch* pada aplikasi aktif hingga relay sebagai aktuator bekerja.

Koneksi internet sangat berpengaruh terhadap respon sistem khususnya pengiriman data dari aplikasi pada web server dan web server pada Raspberry. Semakin cepat koneksi internet maka semakin cepat respon yang diberikan untuk mengaktifkan relay sebagai aktuator beban. Kestabilan pengiriman dan pemrosesan data bergantung pada besarnya *bandwidth* data yang tersedia di jaringan. Lebar suatu *bandwidth* dalam jaringan dapat merepresentasikan kecepatan pengiriman data (*data rate*) karena semakin besar ukuran *bandwidth* maka semakin banyak paket data yang dapat dilewatkan ke jaringan sehingga semakin besar pula kecepatan pengiriman

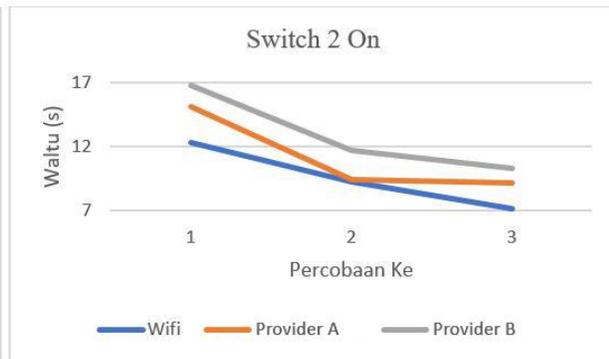
datanya (Mahanta, Ahmed & Bora, 2013). Dari grafik pengujian *switch 1*, *switch 2*, dan *switch 3* yang ditunjukkan pada Gambar 16, Gambar 17, dan Gambar 18 didapatkan bahwa koneksi *wifi* memiliki rata-rata delay paling cepat yaitu 10,30 s, *provider A* 11,17 s, dan *provider B* 12,60 s. Koneksi *wifi* memiliki respon paling cepat disebabkan memiliki *bandwith* yang lebih besar jika dibandingkan *mobile data provider* biasa.

4. Kesimpulan

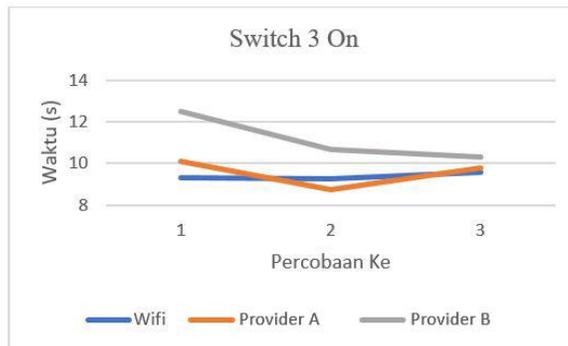
Sistem monitoring daya baterai PLTS dan kontrol beban berbasis *Internet of Things* telah dibangun dan diuji coba secara langsung pada Ruang Prodi Teknik Elektro UTM. Pada sistem monitoring menampilkan persentase daya baterai dan notifikasi peringatan jika pemakaian sudah berada pada batas SOC. Sedangkan pada sistem kontrol difungsikan untuk mengendalikan beban dari jarak jauh apabila terjadi kelalaian oleh pengguna yang mengakibatkan *overdischarge*. Adapun koneksi internet menjadi faktor lain keberhasilan baik monitoring ataupun kontrol. Keseluruhan sistem yang dibangun tersebut bekerja sesuai tujuan penelitian yaitu membantu pengguna dalam mengatur konsumsi daya yang digunakan sekaligus mencegah kerusakan sehingga masa pakai baterai bisa bertahan lebih lama serta pemakaian PLTS lebih maksimal.



Gambar 16. Grafik pengujian *switch 1*



Gambar 17. Grafik pengujian *switch 2*



Gambar 18. Grafik pengujian *switch 3*

Ucapan Terima Kasih

Terima kasih kepada Program Studi Teknik Elektro Universitas Trunojoyo Madura yang telah memberikan kemudahan dan mendukung penelitian ini.

Daftar Pustaka

- Abdi, H., Ivatlo, B. M., Javedi, S., Khodaei, A. R., & Dehnavi, E. (2017). Energy Storage Systems. In G. B. Gharehpetian & S. M. M. Agah (Eds.), *Distributed Generation Systems* (333–368). Oxford: Butterworth-Heinemann.
- Aritonang, C. L., Maison, & Hais, Y. F. (2020). Sistem Monitoring Tegangan, Arus, dan Intensitas Cahaya pada Panel Surya dengan Thingspeak. *Jurnal Engineering*, 2(1), 11–24.
- Fachri, M. R., Sara, I. D., & Away, Y. (2015). Pemantauan Parameter Panel Surya Berbasis Arduino secara Real Time. *Jurnal Rekayasa Elektrika*, 11(4), 123.
- Fadlur, R., & Iqbal, M. (2016). Implementasi Iot Dalam Rancang Bangun Sistem Monitoring Panel Surya Berbasis Arduino. *Prosiding Seminar Nasional Teknologi Dan Informatika* 3, 189–196.
- Fitriandi, A., Komalasari, E., & Gusmedi, H. (2016). Rancang Bangun Alat Monitoring Arus dan Tegangan Berbasis Mikrokontroler dengan SMS Gateway. *Jurnal Electrician*, 10(2), 87–98.
- International Telecommunication Union. (2012). *An overview of internet of things* (ITU-T Y.20). Geneva: International Telecommunication Union.
- Salkind, A. J. S., Hammel, R. O., Cannone, A. G., & Trumbore, F. A. (2001). Valve Regulated Lead-Acid Batteries. In D. Linden & T. B. Reddy (Eds.), *Handbook of Batteries Third Edition* (24.1-24.46). New York: McGraw-Hill.
- Mahanta, D., Ahmed, M., & Bora, J. (2013). A study of Bandwidth Management in Computer Networks. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 2(2), 69–73.
- Margana. (2019). Solar Tracking Dual - Axis Berbasis Arduino Uno dengan menggunakan Lensa Fresnel guna Meningkatkan Efisiensi Pengfokusan Cahaya Matahari. *Jurnal Teknik Energi*, 15(2), 77–80.
- Muzawi, R., Efendi, Y., & Sahrin, N. (2018). Prototype Pengendalian Lampu Jarak Jauh Dengan Jaringan Internet Berbasis Internet of Things(IoT) Menggunakan Raspberry Pi 3. *Jurnal INFORM*, 3(1), 46–50.
- Nugraha, I. M. A., Giriantari, I. A. D., & Kumara, I. N. S. (2013). Studi Dampak Ekonomi dan Sosial PLTS Sebagai Listrik Pedesaan Terhadap Masyarakat Desa Ban Kubu Karangasem. *Prosiding Conference on Smart-Green Technology in Electrical and Information Systems, A-010*, 43–46.
- Pasha, S. (2016). Thingspeak Based Sensing and Monitoring System for IoT with Matlab Analysis. *IJNTR (International Journal of New Technology and Research)*, 2(6), 19–23.
- Patton, E. W., Tissenbaum, M., & Harunani, F. (2019). MIT App Inventor: Objectives, Design, and Development. In S. C. Kong & H. Abelson (Eds.), *Computational Thinking Education* (pp. 31–49). Singapore: Springer
- Rivaldo, R., Bustami, I., & Siswanto, A. (2020). Perancangan Keamanan Pintu Gudang Menggunakan Rfid Dan Kamera Berbasis Raspberry Pi (Studi Kasus: Gudang V-Tech Jambi). *Jurnal Ilmiah Mahasiswa Teknik Informatika*, 2(2), 96–106.
- Sugiyono. (2011). *Pendekatan Kuantitatif, Kualitatif dan R&D*. Bandung: Alfabeta.
- Suryawinata, H., Purwanti, D., & Sunardiyo, S. (2017). Sistem Monitoring Pada Panel Surya Menggunakan Data Logger Berbasis Atmega 328 Dan Real Time Clock DS1307. *Jurnal Teknik Elektro*, 9(1), 30–36.