

Prototype Smart Charging Untuk Sepeda Listrik Berbasis Internet of Things

Gillang Saputra¹, Ares Setiawan¹, Eko Sulistyo¹, Dora Palupi¹

¹Politeknik Manufaktur Negeri Bangka Belitung, Sungailiat

*E-mail : saputragillang06@gmail.com

Received: 12 Januari 2026; Received in revised form: 14 Januari 2026;

Accepted: 14 Januari 2026

Abstract

The increase in the use of electric bicycles has led to the need for an efficient and safe charging system. This study designed a prototype of a smart charging system for electric bicycles based on the Internet of Things (IoT) to facilitate real-time and remote monitoring of charging. This system uses an ESP32 microcontroller, a 24V 15A power supply, 400W and 1200W step-up converters, and ACS712 30A voltage and current sensors. Charging data is displayed through the Blynk application to facilitate user monitoring. Test results show that the voltage increases until it stabilizes, while the current decreases over time during charging, indicating that the system works safely and efficiently (voltage 45–57 V, current approximately 2 A to nearly 0 A, time 0 to 353 minutes). Electricity consumption increases at the start of charging and then decreases, while charging costs continue to accumulate, indicating that the system is capable of calculating energy and estimating costs in real-time (maximum energy approximately 0.0018 kWh, cost approximately Rp1,200–Rp1,300).

Keywords: Smart Charging, Electric Bicycle, IoT, ESP32, Blynk

Abstrak

Peningkatan penggunaan sepeda listrik mendorong perlunya sistem pengisian daya yang efisien dan aman. Penelitian ini merancang prototype smart charging sepeda listrik berbasis Internet of Things (IoT) untuk memungkinkan pemantauan pengisian daya secara real-time dan jarak jauh. Sistem menggunakan mikrokontroler ESP32, power supply 24V 15A, konverter step-up 400W dan 1200W, serta sensor tegangan dan arus ACS712 30A. Data pengisian ditampilkan melalui aplikasi Blynk untuk memudahkan pemantauan pengguna. Hasil pengujian menunjukkan tegangan meningkat hingga stabil, sementara arus menurun seiring waktu pengisian, menandakan sistem bekerja secara aman dan efisien (tegangan 45–57 V, arus kurang lebih 2 A hingga mendekati 0 A, waktu 0 sampai 353 menit). Energi listrik meningkat pada awal pengisian lalu menurun, sedangkan biaya pengisian terus bertambah secara akumulatif, menunjukkan sistem mampu menghitung energi dan estimasi biaya secara real-time (energi maksimal kurang lebih 0,0018 kWh, biaya kurang lebih Rp1.200–Rp1.300).

Kata kunci: Pengisian Daya Cerdas, Sepeda Listrik, IoT, ESP32, Blynk

1. PENDAHULUAN

Perkembangan kendaraan listrik di Indonesia menunjukkan tren peningkatan yang signifikan seiring dengan meningkatnya kesadaran masyarakat terhadap pentingnya energi ramah lingkungan dan efisiensi biaya operasional. Kendaraan listrik dinilai mampu mengurangi emisi gas buang serta ketergantungan terhadap bahan bakar fosil, sehingga menjadi salah satu solusi transportasi berkelanjutan di masa depan.[1] Salah satu bentuk kendaraan listrik yang banyak digunakan pada skala personal adalah sepeda listrik, khususnya untuk aktivitas

jarak pendek seperti keperluan harian di kawasan perumahan.[2]

Meskipun penggunaan sepeda listrik semakin meluas, fasilitas pendukung berupa sistem pengisian daya yang aman, efisien, dan informatif masih terbatas. Sebagian besar pengguna masih mengandalkan pengisian daya konvensional menggunakan stop kontak rumah tangga tanpa adanya pemantauan kondisi baterai secara *real-time*. Metode ini memiliki beberapa kelemahan, antara lain risiko pengisian berlebih (*overcharging*), tidak adanya informasi arus dan tegangan selama pengisian, serta ketidak efisienan

penggunaan energi listrik yang dapat berdampak pada penurunan umur baterai dan potensi gangguan keselamatan.[3]

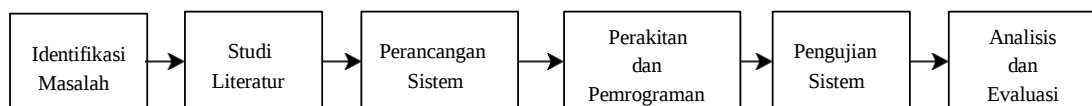
Beberapa penelitian sebelumnya telah membahas sistem pengisian baterai sepeda listrik berbasis mikrokontroler dan IoT. Budianto et al. mengembangkan sistem pengisian baterai sepeda listrik berbasis IoT yang mampu menampilkan parameter dasar pengisian, namun belum dilengkapi dengan perhitungan energi dan estimasi biaya secara *real-time*. [4] Kami telah merancang pengendali *charger* sepeda listrik dengan pengaturan tegangan, tetapi sistem tersebut masih bersifat lokal dan belum terintegrasi dengan pemantauan jarak jauh [3]. Penelitian lain memanfaatkan platform Blynk untuk monitoring perangkat IoT, namun penerapannya belum secara spesifik difokuskan pada sistem pengisian daya sepeda listrik dengan metode CC–CV yang terintegrasi.[5]

Berdasarkan celah penelitian tersebut, kebaruan (*novelty*) pada penelitian ini

terletak pada perancangan prototype *smart charging* sepeda listrik berbasis *Internet of Things* yang tidak hanya memantau tegangan dan arus secara *real-time*, tetapi juga mampu menghitung konsumsi energi serta estimasi biaya pengisian, serta menerapkan metode pengisian *Constant Current–Constant Voltage (CC–CV)* secara terkontrol. Sistem ini dikembangkan menggunakan ESP32 dan aplikasi Blynk sehingga memungkinkan monitoring dan pengendalian jarak jauh melalui *smartphone*. Dengan demikian, penelitian ini diharapkan memberikan kontribusi berupa solusi pengisian daya sepeda listrik yang lebih aman, efisien, informatif, dan aplikatif pada skala personal.

2. METODE PENELITIAN

Metode pelaksanaan yang digunakan dalam proses pengerjaan penelitian ini dijelaskan melalui susunan blok diagram sebagaimana ditunjukkan pada Gambar 1.



Gambar 1. Diagram Alir Metode Penelitian

2.1. Pengambilan dan Pengolahan Data

Pengambilan dan pengolahan data dilakukan untuk mengetahui kinerja sistem *smart charging* sepeda listrik yang telah dirancang. Proses pengambilan data dimulai saat sistem diaktifkan dan proses pengisian daya berlangsung. Data tegangan dan arus diperoleh secara langsung dari komponen sensor yang terpasang pada sistem, yaitu sensor arus ACS712 30A dan rangkaian pembagi tegangan. Sensor ACS712 digunakan untuk mendeteksi besarnya arus yang mengalir selama proses pengisian, sedangkan pembagi tegangan berfungsi untuk menurunkan tegangan keluaran agar dapat dibaca dengan aman oleh pin ADC pada mikrokontroler ESP32.

Seluruh data hasil pembacaan sensor kemudian diproses oleh ESP32 sebagai pusat kendali sistem. Berdasarkan data tersebut, ESP32 mengatur kerja relay dan solid state relay (SSR) yang berfungsi sebagai sakelar elektronik untuk menghubungkan dan memutuskan daya pada rangkaian pengisian. Relay dan SSR

mengendalikan dua buah DC-DC step-up converter, yaitu *step-up* 400 W yang digunakan untuk menaikkan tegangan awal serta *step-up* 1200 W yang digunakan untuk menyesuaikan tegangan pengisian baterai sepeda listrik sesuai kebutuhan. Dengan pengendalian ini, proses pengisian dapat berlangsung secara bertahap dan lebih aman.

Data yang telah diproses oleh ESP32 selanjutnya dikirimkan secara *real-time* ke media monitoring, baik melalui LCD maupun aplikasi berbasis *Internet of Things*. Data tersebut meliputi nilai tegangan, arus, energi yang digunakan, serta estimasi biaya pengisian. Pada tahap pengolahan data, hasil pengukuran dianalisis untuk melihat kestabilan sistem, pola perubahan arus dan tegangan selama pengisian, serta kesesuaian kerja komponen dengan metode pengisian yang diterapkan. Data kemudian disajikan dalam bentuk tabel dan grafik untuk mempermudah proses analisis dan evaluasi kinerja sistem secara keseluruhan.

1. Mikrokontroler ESP32 Wroom

ESP32 merupakan mikrokontroler yang digunakan sebagai pusat kendali pada sistem *smart charging* sepeda listrik. Komponen ini berfungsi untuk mengumpulkan data dari sensor arus dan sensor tegangan, kemudian mengolah data tersebut guna memantau kondisi pengisian baterai.[6] Selain itu, ESP32 mengatur proses pengisian dengan mengendalikan relay dan solid state relay sehingga aliran daya dapat dikontrol secara aman. Dengan adanya konektivitas WiFi, ESP32 juga memungkinkan sistem melakukan monitoring dan pengendalian pengisian daya secara jarak jauh melalui aplikasi berbasis *Internet of Things*.

2. Sensor ACS712 30A

Sensor ACS712 30A digunakan untuk mengukur besar arus listrik yang mengalir selama proses pengisian baterai. Sensor ini bekerja dengan mendeteksi medan magnet yang dihasilkan oleh arus listrik, sehingga dapat memantau arus secara *real-time* tanpa mengganggu rangkaian utama.[7] Data arus ini digunakan untuk memastikan proses pengisian berlangsung aman dan sesuai dengan batas yang ditentukan.

3. Pembagi Tegangan

Pembagi tegangan berfungsi untuk menurunkan tegangan keluaran dari rangkaian pengisian agar berada pada batas aman untuk dibaca oleh mikrokontroler ESP32.[8] Dengan adanya rangkaian ini, tegangan baterai yang relatif tinggi dapat dipantau secara akurat tanpa merusak pin input mikrokontroler.

4. Converter step-up DC-DC 400W

Converter step-up DC-DC 400W digunakan untuk menaikkan tegangan dari sumber daya awal ke tingkat tegangan menengah.[9] Modul ini berperan sebagai tahap awal dalam proses pengaturan tegangan sebelum disesuaikan kembali sesuai kebutuhan pengisian baterai bank.

5. Converter step-up DC-DC 1200W

Converter step-up DC-DC 1200W berfungsi untuk menaikkan dan menstabilkan tegangan hingga mencapai tegangan yang dibutuhkan oleh baterai sepeda listrik. Modul ini mampu menangani daya yang lebih besar sehingga mendukung proses pengisian yang efisien dan stabil.[9]

6. Relay

Relay digunakan sebagai sakelar elektromekanis yang berfungsi untuk menghubungkan dan memutus aliran listrik pada sistem pengisian.[10] Komponen ini dikendalikan oleh ESP32 dan berperan dalam mengatur proses pengisian agar dapat dinyalakan atau dimatikan sesuai kondisi sistem.

7. Solid State Relay (SSR) 60V

Solid State Relay 60V berfungsi sebagai sakelar elektronik tanpa komponen mekanis yang digunakan untuk pengendalian beban dengan respon cepat dan tingkat keandalan tinggi. SSR digunakan untuk meningkatkan keamanan dan kestabilan sistem, terutama saat pengendalian daya pada tegangan dan arus yang relatif besar.[10]

8. Internet of Things (IoT)

Internet of Things (IoT) memungkinkan sistem *smart charging* terhubung ke jaringan internet sehingga data pengisian dapat dipantau secara *real-time*. Dengan teknologi ini, pengguna dapat mengetahui kondisi pengisian baterai tanpa harus berada langsung di dekat alat.

9. Blynk App

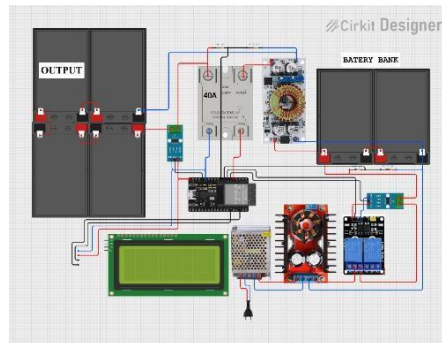
Blynk merupakan platform aplikasi IoT yang digunakan sebagai antarmuka monitoring dan kontrol sistem. Melalui aplikasi Blynk, pengguna dapat melihat informasi tegangan, arus, energi, serta status pengisian secara langsung, sekaligus mengendalikan sistem pengisian melalui *smartphone* dengan mudah dan praktis.

2.2. Perancangan Hardware Prototype *Smart Charging* Untuk Sepeda Listrik Berbasis *Internet of Things*

Desain elektrikal pada sistem *smart charging* sepeda listrik dirancang menggunakan web *circuit designer* sebagai media simulasi dan perancangan rangkaian sebelum dilakukan perakitan secara fisik. Penggunaan web *circuit designer* bertujuan untuk memvisualisasikan hubungan antar komponen, seperti ESP32, sensor arus ACS712 30A, rangkaian pembagi tegangan, relay, solid state relay, serta DC-DC step-up converter 400 W dan 1200 W, sehingga alur daya dan sinyal dapat dipahami dengan lebih jelas. Melalui perancangan ini, jalur tegangan dan kontrol dapat disusun secara sistematis untuk

memastikan setiap komponen bekerja sesuai fungsinya. Selain itu, desain elektrik berbasis web *circuit designer* membantu mengurangi kesalahan perakitan, memudahkan proses evaluasi rangkaian,

serta memastikan sistem memiliki tingkat keamanan dan keandalan yang baik sebelum diimplementasikan ke dalam bentuk *hardware* sebenarnya.



Gambar 2. Desain Rangkaian Elektrikal

Perancangan Sistem Otomatis dan Integrasi IoT.

Tahapan awal dalam pembuatan sistem monitoring berbasis *smartphone* adalah melakukan pemrograman mikrokontroler menggunakan *software*

Arduino IDE. Pada tahap ini, program disusun untuk mengatur proses pembacaan data sensor, pengolahan data, serta pengiriman informasi ke aplikasi monitoring melalui koneksi *Internet of Things*.

```

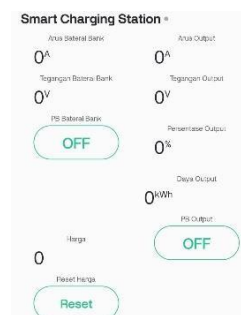
1 // Arduino IDE - Smart Charging Station
2 // Arduino IDE - Smart Charging Station
3 // Arduino IDE - Smart Charging Station
4 // Arduino IDE - Smart Charging Station
5 // Arduino IDE - Smart Charging Station
6 // Arduino IDE - Smart Charging Station
7 // Arduino IDE - Smart Charging Station
8 // Arduino IDE - Smart Charging Station
9 // Arduino IDE - Smart Charging Station
10 // Arduino IDE - Smart Charging Station
11 // Arduino IDE - Smart Charging Station
12 // Arduino IDE - Smart Charging Station
13 // Arduino IDE - Smart Charging Station
14 // Arduino IDE - Smart Charging Station
15 // Arduino IDE - Smart Charging Station
16 // Arduino IDE - Smart Charging Station
17 // Arduino IDE - Smart Charging Station
18 // Arduino IDE - Smart Charging Station
19 // Arduino IDE - Smart Charging Station
20 // Arduino IDE - Smart Charging Station
21 // Arduino IDE - Smart Charging Station
22 // Arduino IDE - Smart Charging Station
23 // Arduino IDE - Smart Charging Station
24 // Arduino IDE - Smart Charging Station
25 // Arduino IDE - Smart Charging Station
26 // Arduino IDE - Smart Charging Station
27 // Arduino IDE - Smart Charging Station
28 // Arduino IDE - Smart Charging Station
29 // Arduino IDE - Smart Charging Station
30 // Arduino IDE - Smart Charging Station
31 // Arduino IDE - Smart Charging Station
32 // Arduino IDE - Smart Charging Station
33 // Arduino IDE - Smart Charging Station
34 // Arduino IDE - Smart Charging Station
35 // Arduino IDE - Smart Charging Station
36 // Arduino IDE - Smart Charging Station
37 // Arduino IDE - Smart Charging Station
38 // Arduino IDE - Smart Charging Station
39 // Arduino IDE - Smart Charging Station
40 // Arduino IDE - Smart Charging Station
41 // Arduino IDE - Smart Charging Station
42 // Arduino IDE - Smart Charging Station
43 // Arduino IDE - Smart Charging Station
44 // Arduino IDE - Smart Charging Station
45 // Arduino IDE - Smart Charging Station
46 // Arduino IDE - Smart Charging Station
47 // Arduino IDE - Smart Charging Station
48 // Arduino IDE - Smart Charging Station
49 // Arduino IDE - Smart Charging Station
50 // Arduino IDE - Smart Charging Station
51 // Arduino IDE - Smart Charging Station
52 // Arduino IDE - Smart Charging Station
53 // Arduino IDE - Smart Charging Station
54 // Arduino IDE - Smart Charging Station
55 // Arduino IDE - Smart Charging Station
56 // Arduino IDE - Smart Charging Station
57 // Arduino IDE - Smart Charging Station
58 // Arduino IDE - Smart Charging Station
59 // Arduino IDE - Smart Charging Station
60 // Arduino IDE - Smart Charging Station
61 // Arduino IDE - Smart Charging Station
62 // Arduino IDE - Smart Charging Station
63 // Arduino IDE - Smart Charging Station
64 // Arduino IDE - Smart Charging Station
65 // Arduino IDE - Smart Charging Station
66 // Arduino IDE - Smart Charging Station
67 // Arduino IDE - Smart Charging Station
68 // Arduino IDE - Smart Charging Station
69 // Arduino IDE - Smart Charging Station
70 // Arduino IDE - Smart Charging Station
71 // Arduino IDE - Smart Charging Station
72 // Arduino IDE - Smart Charging Station
73 // Arduino IDE - Smart Charging Station
74 // Arduino IDE - Smart Charging Station
75 // Arduino IDE - Smart Charging Station
76 // Arduino IDE - Smart Charging Station
77 // Arduino IDE - Smart Charging Station
78 // Arduino IDE - Smart Charging Station
79 // Arduino IDE - Smart Charging Station
80 // Arduino IDE - Smart Charging Station
81 // Arduino IDE - Smart Charging Station
82 // Arduino IDE - Smart Charging Station
83 // Arduino IDE - Smart Charging Station
84 // Arduino IDE - Smart Charging Station
85 // Arduino IDE - Smart Charging Station
86 // Arduino IDE - Smart Charging Station
87 // Arduino IDE - Smart Charging Station
88 // Arduino IDE - Smart Charging Station
89 // Arduino IDE - Smart Charging Station
90 // Arduino IDE - Smart Charging Station
91 // Arduino IDE - Smart Charging Station
92 // Arduino IDE - Smart Charging Station
93 // Arduino IDE - Smart Charging Station
94 // Arduino IDE - Smart Charging Station
95 // Arduino IDE - Smart Charging Station
96 // Arduino IDE - Smart Charging Station
97 // Arduino IDE - Smart Charging Station
98 // Arduino IDE - Smart Charging Station
99 // Arduino IDE - Smart Charging Station
100 // Arduino IDE - Smart Charging Station

```

Gambar 3. Tampilan pada Arduino IDE

Setelah program berhasil diunggah ke mikrokontroler ESP32, sistem dikonfigurasi untuk terhubung ke jaringan WiFi dan selanjutnya mengakses server Blynk. Melalui koneksi ini, data hasil

pembacaan sensor dapat dikirimkan dan ditampilkan secara *real-time* pada aplikasi Blynk di *smartphone*. Tampilan antarmuka monitoring pada aplikasi Blynk ditunjukkan pada Gambar 4.



Gambar 5. Tampilan Blynk pada Smartphone

2.3. Pembuatan Laporan Akhir

Pembuatan laporan akhir merupakan tahap penutup dari seluruh rangkaian kegiatan proyek akhir. Laporan ini disusun untuk mendokumentasikan proses perancangan, pembuatan, dan implementasi sistem *smart charging* sepeda listrik berbasis *Internet of Things* secara sistematis dan terstruktur.

3. HASIL DAN PEMBAHASAN

Flowchart ini menggambarkan alur kerja prototype *smart charging* sepeda listrik berbasis *Internet of Things*. Sistem dimulai dengan inisialisasi ESP32 dan koneksi ke aplikasi Blynk, kemudian membaca data

tegangan dan arus dari baterai. Berdasarkan kondisi tegangan, sistem mengatur proses pengisian melalui relay dan *solid state relay*. Selain itu, laporan akhir juga menyajikan hasil pengujian serta analisis kinerja sistem berdasarkan data yang diperoleh. Penyajian data dalam bentuk uraian, tabel, dan grafik bertujuan untuk mempermudah pemahaman serta menjadi bahan evaluasi dan referensi bagi pengembangan sistem selanjutnya.

Untuk mengisi baterai bank dan baterai sepeda listrik secara bertahap. Selama proses berlangsung, data pengisian dikirim dan ditampilkan secara *real-time* pada aplikasi Blynk di *smartphone*, sehingga pengguna dapat memantau kondisi pengisian dengan mudah.



Gambar 6, Flowchart Sistem Kerja

3.1. Deskripsi Alat

Alat yang dikembangkan pada penelitian ini merupakan *prototype smart charging* untuk sepeda listrik yang dirancang untuk meningkatkan keamanan dan efisiensi proses pengisian baterai. Sistem ini dikembangkan sebagai solusi atas metode pengisian konvensional yang masih mengandalkan *stop kontak* rumah tangga tanpa adanya pemantauan kondisi pengisian secara langsung, sehingga pengguna tidak dapat mengetahui status baterai maupun konsumsi energi yang digunakan selama proses pengisian berlangsung.

Prototype smart charging untuk sepeda listrik ini menggunakan mikrokontroler

ESP32 sebagai pusat kendali utama yang mengatur seluruh alur kerja sistem. ESP32 terintegrasi dengan sensor arus ACS712 dan rangkaian pembagi tegangan untuk memantau nilai arus dan tegangan secara *real-time*. Informasi hasil pemantauan ditampilkan melalui LCD dan dikirimkan ke aplikasi berbasis *Internet of Things*, sehingga pengguna dapat memantau status pengisian, konsumsi energi, serta estimasi biaya pengisian melalui *smartphone*, baik secara lokal maupun dari jarak jauh. Dengan adanya sistem ini, proses pengisian daya sepeda listrik menjadi lebih praktis, transparan, dan mudah dikendalikan. Gambar 3. Menunjukkan Tampak Luar dan Dalam pada Alat.

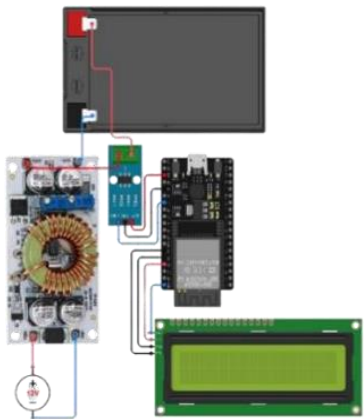


Gambar 3. Tampak Luar dan Dalam pada Alat

3.2. Pengujian Pada Sensor ACS712 30A

Uji coba sensor arus ACS712 30A dilakukan untuk mengetahui kemampuan sensor dalam membaca arus listrik yang mengalir pada sistem pengisian daya.[3] Sensor dipasang secara seri pada jalur arus antara sumber daya dan beban, kemudian dihubungkan ke ESP32 sebagai pengolah data. Nilai arus yang terbaca oleh sensor diproses oleh ESP32 dan ditampilkan melalui LCD sebagai hasil pemantauan.

Hasil pengujian menunjukkan bahwa sensor ACS712 30A mampu membaca perubahan arus dengan baik dan menampilkan nilai yang stabil selama proses pengujian. Sensor ini dapat digunakan secara efektif untuk memantau arus pengisian daya secara *real-time*, sehingga mendukung sistem *monitoring* dan pengendalian pengisian daya sepeda listrik secara lebih aman dan efisien.



Gambar 4. Rangkaian Pengujian Sensor ACS712 30A

Tabel 1. Hasil Pengujian Sensor ACS712

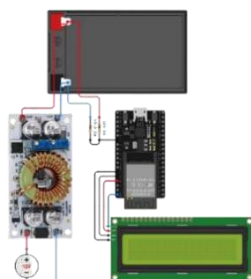
30A Tegangan Input (Volt)	Arus (Ampere)
14.01 Volt	0.52 Ampere
15.05 Volt	0.63 Ampere
15.51 Volt	0.88 Ampere
16.01 Volt	1.00 Ampere
16.51 Volt	1.13 Ampere
17.01 Volt	1.50 Ampere
17.50 Volt	1.94 Ampere

3.3. Pengujian Pembagi Tegangan

Rangkaian uji coba pembagi tegangan digunakan untuk menguji kemampuan sistem dalam membaca tegangan keluaran yang lebih tinggi agar dapat dipantau dengan aman oleh mikrokontroler ESP32.[2] Pada rangkaian ini, sumber tegangan berasal dari modul DC-DC yang terhubung ke baterai, kemudian sebagian tegangan diturunkan menggunakan rangkaian pembagi tegangan sebelum masuk ke pin analog ESP32. Tujuan pengujian ini adalah memastikan bahwa tegangan yang masuk ke

mikrokontroler berada pada batas aman serta dapat dibaca secara akurat.

Hasil uji coba menunjukkan bahwa rangkaian pembagi tegangan mampu menurunkan tegangan sesuai dengan perbandingan nilai resistor yang digunakan. Tegangan hasil pembagi dapat dibaca dengan stabil oleh ESP32 dan ditampilkan melalui LCD. Hal ini menunjukkan bahwa rangkaian pembagi tegangan bekerja dengan baik dan dapat digunakan sebagai bagian dari sistem *monitoring* tegangan pada *prototype smart charging* untuk sepeda listrik ini.



Gambar 5. Rangkaian Pengujian Pembagi Tegangan

Tabel 2. Hasil Pengujian Rangkaian Pembagi Tegangan

Tegangan Input (Volt)	Multimeter (Volt)	Pembagi Tegangan (Volt)	Error (%)
20 Volt	20.27 Volt	20.05 Volt	1.09 %
20.5 Volt	20.51 Volt	20.42 Volt	0.44 %
21 Volt	21.06 Volt	21.05 Volt	0.05 %
21.5 Volt	21.51 Volt	21.52 Volt	0.05 %
22 Volt	22 Volt	22.03 Volt	0.14 %
22,5 Volt	22.50 Volt	22.63 Volt	0.58 %
23 Volt	23.01 Volt	23.21 Volt	0.87 %
23.5 Volt	23 .51 Volt	23.51 Volt	0.00 %
24 Volt	24.02 Volt	24.34 Volt	1.33 %
24.5 Volt	24.53 Volt	24.67 Volt	0.57 %

Untuk menghitung *error* pada pengujian rangkaian pembagi tegangan di gunakanlah rumus pembagi tegangan digunakan untuk menghitung besarnya tegangan keluaran yang masuk ke mikrokontroler berdasarkan perbandingan nilai resistor yang digunakan. Perhitungan ini

bertujuan untuk memastikan bahwa tegangan hasil pembagian berada pada batas aman dan sesuai dengan kemampuan pembacaan pin analog, sehingga pengukuran tegangan dapat dilakukan secara akurat. Berikut rumus untuk perhitungan *error* pada pengujian pembagi tegangan.

$$Error(\%) = \left| \frac{\text{Hasil Pembacaan Sensor} - \text{Hasil Pembacaan Multimeter}}{\text{Hasil Pembacaan Multimeter}} \right| \times 100\%$$

Contoh perhitungan *error* pada tegangan 20 Volt

$$Error = \left| \frac{20.05 - 20.27}{20.27} \right| \times 100\% = 1,09\%$$

3.4. Hasil Tegangan dan Arus Selama Pengisian Berlangsung

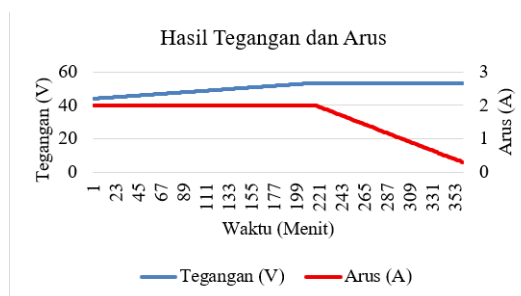
Berdasarkan hasil pengujian, perubahan tegangan dan arus selama proses pengisian sepeda listrik menunjukkan karakteristik yang sesuai dengan metode pengisian *Constant Current–Constant Voltage* (CC–CV). Pada tahap awal pengisian, yaitu sekitar menit ke-1 hingga menit ke 55, tegangan baterai masih berada pada nilai rendah dan meningkat secara bertahap, sementara arus pengisian dijaga pada nilai yang relatif konstan.

Memasuki tahap pengisian berikutnya, dari sekitar menit ke 55 hingga menit ke 199, tegangan terus meningkat secara stabil hingga mendekati nilai maksimum. Pada fase ini, sistem masih bekerja pada mode arus konstan (*constant current*) untuk mempercepat proses pengisian energi ke dalam baterai. Arus yang stabil menunjukkan bahwa sistem mampu menjaga arus pengisian sesuai dengan parameter yang telah ditentukan.

Setelah tegangan mencapai nilai

mendekati maksimum pada sekitar menit ke 199 hingga menit ke 217, sistem beralih ke mode tegangan konstan (*constant voltage*). Pada fase ini, tegangan dijaga tetap stabil, sedangkan arus pengisian mulai menurun secara bertahap. Penurunan arus ini menandakan bahwa kapasitas baterai sudah mendekati penuh dan sistem secara otomatis menyesuaikan arus untuk menjaga keamanan baterai.

Pada tahap akhir pengisian, yaitu dari sekitar menit ke 217 hingga menit ke 343, arus pengisian terus menurun hingga mencapai nilai yang sangat kecil, sementara tegangan tetap stabil. Kondisi ini menunjukkan bahwa proses pengisian telah selesai secara bertahap dan aman. Hasil ini membuktikan bahwa sistem pengisian daya yang dirancang mampu menerapkan metode CC–CV dengan baik, sebagaimana ditunjukkan pada grafik hasil pengujian tegangan dan arus. Gambar 6. menunjukkan hasil tegangan dan arus selama pengisian berlangsung.



Gambar 6. Hasil Tegangan dan Arus Selama Pengisian Berlangsung

3.5. Hasil Energi dan Biaya Selama Pengisian Berlangsung

Berdasarkan hasil pengujian, konsumsi energi dan biaya pengisian sepeda listrik menunjukkan pola perubahan yang sejalan dengan proses pengisian menggunakan metode *Constant Current–Constant Voltage* (CC–CV). Pada tahap awal pengisian, yaitu sekitar menit ke 1 hingga menit ke 55, energi listrik yang digunakan masih relatif kecil dan meningkat secara perlahan seiring dengan bertambahnya waktu pengisian.

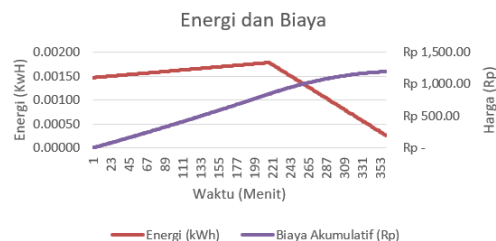
Memasuki tahap pengisian selanjutnya, dari sekitar menit ke 55 hingga menit ke 199, energi yang terserap oleh baterai meningkat secara lebih signifikan. Hal ini disebabkan oleh kondisi pengisian

pada mode arus konstan, di mana daya yang disalurkan ke baterai relatif stabil. Seiring dengan peningkatan energi yang digunakan, biaya pengisian juga meningkat secara bertahap sesuai dengan konsumsi energi listrik. Setelah mencapai sekitar menit ke 199 hingga menit ke 217, laju peningkatan energi mulai melambat. Pada fase ini, sistem mulai beralih ke mode tegangan konstan, sehingga arus pengisian menurun. Penurunan arus menyebabkan pertambahan energi tidak lagi secepat tahap sebelumnya, meskipun proses pengisian masih terus berlangsung.

Pada tahap akhir pengisian, yaitu dari sekitar menit ke 217 hingga menit ke 353, pertambahan energi cenderung semakin kecil, sedangkan biaya pengisian

mencapai nilai akumulatif akhir. Kondisi ini menunjukkan bahwa sebagian besar energi telah terserap oleh baterai dan sistem hanya menyuplai energi tambahan dalam jumlah kecil untuk menjaga tegangan tetap stabil. Pola perubahan

energi dan biaya ini menunjukkan bahwa sistem mampu menghitung konsumsi energi serta estimasi biaya pengisian secara akurat, sebagaimana ditunjukkan pada gambar 7.



Gambar 7. Hasil Energi dan Biaya Selama Pengisian Berlangsung

4. SIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa *prototype smart charging* untuk sepeda listrik berbasis *Internet of Things* berhasil dikembangkan dan berfungsi dengan baik. Sistem mampu melakukan proses pengisian daya secara aman dan terkontrol dengan menerapkan metode pengisian *Constant Current–Constant Voltage*, sehingga tegangan dan arus dapat disesuaikan dengan kondisi baterai selama proses pengisian berlangsung. Hasil pengujian menunjukkan bahwa sistem mampu memantau tegangan, arus, konsumsi energi, serta estimasi biaya pengisian secara *real-time*. Integrasi ESP32 dengan sensor arus dan tegangan, serta *platform* IoT, memungkinkan proses *monitoring* dan pengendalian dilakukan baik secara lokal melalui LCD maupun dari jarak jauh melalui *smartphone*. Hal ini memberikan kemudahan dan transparansi bagi pengguna dalam mengetahui kondisi pengisian baterai.

Selain itu, sistem pengisian daya yang dikembangkan terbukti lebih efisien dibandingkan metode pengisian konvensional, karena mampu menyesuaikan parameter pengisian sesuai kondisi baterai dan mencegah terjadinya pengisian berlebih. Dengan demikian, alat ini berpotensi menjadi solusi alternatif yang aplikatif dan terjangkau dalam mendukung penggunaan sepeda listrik serta pengembangan infrastruktur pengisian daya di lingkungan masyarakat.

DAFTAR PUSTAKA

- [1] A. P. Junfithrana, "Aplikasi smart home node mcu iot untuk blynk," *J. Rekayasa Teknol. Nusa Putra*, 2021.
- [2] S. Fuada et al., "Analisis rangkaian pembagi tegangan dan perbandingan hasil simulasinya menggunakan simulator offline," *CIRCUIT J. Ilm. Pendidik. Tek. Elektro*, vol. 6, no. 1, pp. 28–46, 2022.
- [3] A. K. Lubis and D. Sawitri, "Desain dan perancangan alat pantau energi listrik di rumah jarak jauh berbasis IoT," *MeSTERI J.*, vol. 1, no. 1, pp. 46–53, 2022.
- [4] A. Mubarak'Aafi, J. Jamaaluddin, and I. Anshory, "Implementasi sensor PZEM-017 untuk monitoring arus, tegangan dan daya pada instalasi panel surya dengan sistem data logger menggunakan Google Spreadsheet dan smartphone," in *Prosiding Seminar Nasional Teknik Elektro, Sistem Informasi, dan Teknik Informatika (SNESTIK)*, 2022, pp. 191–196.
- [5] S. A. Arrahma and R. Mukhaiyar, "Pengujian Esp32-Cam Berbasis Mikrokontroler ESP32," *JTEIN J. Tek. Elektro Indones.*, vol. 4, no. 1, pp. 60–66, 2023.
- [6] A. S. Ananda, L. N. H. Hayati, and I. As'ad, "Stasiun Pengisian Energi Baterai Ramah Lingkungan Berbasis Panel Surya," *J. Resist. (Rekayasa Sist. Komputer)*, vol. 5, no. 2, pp. 150–158, 2022.
- [7] A. Saputra, H. Khumaini, and A. Azkiya, "Perancangan Alat Monitoring Arus Pada Circuit Breaker Dengan Sensor Acs712 Menggunakan Tampilan Lcd," *Inf. a TIK a*, vol. 14, no. 2, p. 86, 2023.

- [8] M. S. D. Utomo *et al.*, "Analisis perhitungan teori dengan menggunakan variasi simulator online pada rangkaian pembagi tegangan," *Telecommun. Networks, Electron. Comput. Technol.*, vol. 1, no. 2, pp. 61–70, 2021.
- [9] E. Ginanjar, A. Mashar, and W. B. Mursanto, "Perancangan Buck Boost Converter Pada Sistem Pengisian Baterai Untuk Panel Surya Kapasitas 50 Wp," in *Prosiding Industrial Research Workshop and National Seminar*, 2022, pp. 517–524.
- [10] K. D. Pane, M. E. Dalimunthe, and Y. Saragih, "Pengendali Pintar Dalam Pengontrolan Lampu Dan Kipas Menggunakan Bluetooth," *Aisyah J. Informatics Electr. Eng.*, vol. 6, no. 2, pp. 160–169, 2024.