

***PROTOTYPE MONITORING KINERJA BATERAI
KENDARAAN BERMOTOR YANG TERINTEGRASI
DENGAN APLIKASI MOBILE***

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat Kelulusan Diploma III Politeknik Manufaktur Negeri Bangka Belitung



Diusulkan oleh

Haris Adi Putra	NIM :	0032211
Putri Yuliana	NIM :	0032222

**POLITEKNIK MANUFAKTUR NEGERI
BANGKA BELITUNG
TAHUN 2025**

LEMBAR PENGESAHAN

JUDUL PROYEK AKHIR

***PROTOTYPE MONITORING KINERJA BATERAI
KENDARAAN BERMOTOR YANG TERINTEGRASI
DENGAN APLIKASI MOBILE***

Oleh :

Haris Adi Putra	NIM	0032211
Putri Yuliana	NIM	0032222

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan Program Dipolma D-III Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

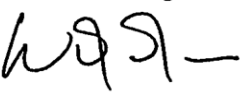
Pembimbing 1


(Zanu Saputra, M.Tr.T)

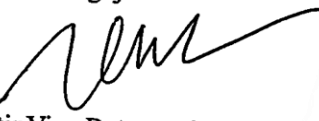
Penguji 1


(Peprizal, M.Pd.T)

Pembimbing 2


(Yus Dwi Yanti, S.P, M.Si)

Penguji 2


(Valentin Vina Ratnapuri, M.Si)

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini :

Nama Mahasiswa 1	: Haris Adi Putra	NIM : 0032211
Nama Mahasiswa 2	: Putri Yuliana	NIM : 0032222

Dengan Judul : *PROTOTYPE MONITORING KINERJA BATERAI KENDARAAN BERMOTOR YANG TERINTEGRASI DENGAN APLIKASI MOBILE*

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sunggailiat, 27 Juni 2025

Nama Mahasiswa

Tanda Tangan

1. Haris Adi Putra



2. Putri Yuliana



ABSTRAK

Kendaraan bermotor, khususnya sepeda motor, memainkan peran penting dalam mobilitas masyarakat modern. Salah satu elemen penting pada kendaraan bermotor adalah baterai, baterai sebagai sumber energi utama sering kali mengalami penurunan kualitas dan usia pakai yang singkat, sehingga memerlukan sistem pemantauan yang efektif. Penelitian ini bertujuan untuk merancang dan mengembangkan sistem pemantauan kinerja baterai yang terhubung dengan aplikasi *mobile*, dengan memanfaatkan *mikrokontroler* NodeMCU ESP32 dan sensor untuk memantau parameter penting seperti tegangan, arus, dan suhu secara *real-time*. Data yang dikumpulkan akan ditampilkan pada layar LCD serta pada aplikasi *mobile* yang dibuat menggunakan *MIT App Inventor*, memberikan notifikasi kepada pengguna tentang kondisi abnormal seperti *overcharging*, *overdischarging*, dan suhu tinggi. Metode penelitian mencakup perancangan sistem, pembuatan *prototype*, dan pengujian untuk mengevaluasi kinerja sistem. Oleh karena itu, proyek ini diharapkan dapat memberikan kontribusi yang berarti dalam meningkatkan kesadaran pengguna terhadap kondisi baterai kendaraan bermotor dan mencegah kerusakan lebih lanjut. Berdasarkan hasil pengujian, rata-rata persentase error sensor tegangan sebesar 4,78%, sensor arus INA219 sebesar 21,2%, dan sensor arus ACS712 sebesar 8,86%. Sementara itu, sensor suhu DS18B20 menunjukkan respons yang baik terhadap perubahan suhu lingkungan. Seluruh nilai error tersebut masih berada dalam batas toleransi yang dapat diterima.

Kata kunci: Aplikasi *Mobile*, Baterai, *NodeMCU* ESP32, Sepeda Motor, Sistem *Monitoring*.

ABSTRACT

Motor vehicles, especially motorcycles, play an important role in the mobility of modern society. One of the important elements in motor vehicles is the battery, which serves as the primary energy source and often experiences a decline in quality and a short lifespan, thus requiring an effective monitoring system. This research aims to design and develop a battery performance monitoring system connected to a mobile application, utilizing the NodeMCU ESP32 microcontroller and sensors to monitor important parameters such as voltage, current, and temperature in real time. The collected data will be displayed on an LCD screen as well as on a mobile application created using MIT App Inventor, providing notifications to users about abnormal conditions such as overcharging, overdischarging, and high temperatures. The research methods include system design, prototype creation, and testing to evaluate the system's performance. Therefore, this project is expected to make a meaningful contribution to raising user awareness about the condition of motor vehicle batteries and preventing further damage. Based on the test results, the average percentage of error for the voltage sensor is 4.78%, the current sensor INA219 is 21.2%, and the current sensor ACS712 is 8.86%. Meanwhile, the temperature sensor DS18B20 shows a good response to changes in the ambient temperature. All of these error values are still within the acceptable tolerance limits.

Keywords: Mobile Application, Battery, NodeMCU ESP32, Motorcycle, Monitoring System.

KATA PENGANTAR

Assalamu'alaikum warahmatullahi wabarakatuh

Segala puji syukur kami panjatkan kepada Allah SWT yang telah memberikan kemudahan, memungkinkan kami menyelesaikan Laporan Proyek Akhir berjudul “*Prototype Monitoring Kinerja Baterai Kendaraan Bermotor yang Terintegrasi dengan Aplikasi Mobile*” tepat waktu. Shalawat dan salam kami sampaikan kepada Nabi Muhammad SAW yang dinanti syafa'atnya di akhirat. Laporan ini merupakan syarat kelulusan Program Studi D-III Teknik Elektronika di Politeknik Manufaktur Negeri Bangka Belitung.

Kami ingin menyampaikan rasa terima kasih kepada pihak-pihak yang telah memberikan banyak bantuan, motivasi, serta memberikan saran dan kritik yang sangat diharapkan dalam menyelesaikan proyek akhir ini. Berikut adalah pihak-pihak yang turut membantu, baik secara langsung maupun tidak langsung, di antaranya:

1. Orang tua dan keluarga penulis yang senantiasa memberikan doa dan dukungan moral serta materi yang sangat berharga.
2. Rekan kerja Proyek Akhir yang telah mendukung dan berjuang bersama sehingga proyek akhir ini dapat diselesaikan dengan sebaik-baiknya.
3. Bapak Zanu Saputra, M.Tr.T. selaku pembimbing institusi dan juga selaku Kepala Jurusan Rekayasa Elektro dan Industri Pertanian sekaligus Dosen Pembimbing 1 yang telah memberikan bimbingan, arahan, dan saran berharga selama pembuatan dan penyusunan laporan proyek akhir ini.
4. Yus Dwi Yanti, S.P., M.Si. selaku Dosen Pembimbing 2 yang telah memberikan bimbingan, arahan, dan saran berharga selama pembuatan dan penyusunan laporan proyek akhir ini.

5. Bapak I Made Andik Setiawan, M.Eng., Ph.D. selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung.
6. Ibu Novitasari, M.Pd., selaku Kepala Prodi D-III Teknik Elektronika.
7. Bapak Ocsirendi, S.ST.,M.T. selaku Dosen Wali penulis di Politeknik Manufaktur Negeri Bangka Belitung.
8. Seluruh Dosen dan PLP yang sudah banyak membantu penulis dalam menyelesaikan proyek akhir ini.
9. Seluruh teman-teman kelas 3 EA yang telah banyak membantu dan kebersamaian 3 tahun ini.
10. Seluruh pihak-pihak terkait yang tidak dapat penulis sebutkan satu per satu.

Penulis berharap agar laporan ini memberikan manfaat dengan meningkatkan wawasan dan pengetahuan yang terkandung dalam proyek akhir. Penulis menyadari bahwa laporan proyek akhir ini masih memiliki kekurangan dan jauh dari kesempurnaan. Oleh karena itu, penulis menghargai kritik dan saran yang konstruktif untuk perbaikan di masa mendatang. Semoga laporan proyek akhir ini dapat dipahami dan bermanfaat bagi semua orang, terutama bagi mereka yang membaca laporan ini.

Sungailiat, 27 Juni 2025

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	i
PERNYATAAN BUKAN PLAGIAT.....	ii
ABSTRAK	iii
<i>ASTRACT</i>	iv
KATA PENGANTAR.....	v
DAFTAR ISI	vii
DAFTAR TABEL	x
DAFTAR GAMBAR.....	xi
LAMPIRAN	xiii
BAB I PENDAHULUAN	
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	3
1.4 Batasan Masalah	3
BAB II DASAR TEORI	
2.1 Baterai Kendaraan Sepeda Motor	4
2.1.1 Aki Kering (<i>Maintenance-Free</i>).....	5
2.2 Sistem Kontrol Baterai Sepeda Motor.....	6
2.2.1 ESP32.....	6
2.2.2 Relay.....	7
2.3 Sistem <i>Monitoring Prototype</i> Baterai yang Terintegrasi dengan Aplikasi	

<i>Mobile</i>	8
2.3.1 Aplikasi <i>Mobile Berbasis MIT App Inventor</i>	9
2.3.2 <i>Voltage Sensor</i>	11
2.3.3 Sensor Arus INA219	11
2.3.4 Sensor Arus ACS712	12
2.3.5 Sensor Suhu DS18B20	13
2.3.6 <i>Real Time Clock (RTC)</i>	13
2.3.7 <i>Liquid Cristal Display (LCD)</i>	14
2.4 Proteksi Sistem <i>Charging</i>	15
2.5 Proteksi Sistem Kelistrikan	16
BAB III METODE PELAKSANAAN	
3.1 Tahapan Perancangan dan Pembuatan Alat	17
3.2 <i>Study literature</i>	17
3.3 Mekanisme <i>Prototype Monitoring</i> Kinerja Baterai Kendaraan Bermotor yang Terintegrasi dengan Aplikasi <i>Mobile</i>	18
3.3.1 Mekanisme Sistem <i>Monitoring</i> Baterai	18
3.3.2 Mekanisme Sistem <i>Charging</i> Baterai	19
3.3.3 Mekanisme Sistem Kelistrikan Baterai.....	21
3.4 Perancangan Konstruksi <i>Prototype Monitoring</i> Kinerja Baterai Kendaraan Bermotor	22
3.5 Perancangan Sistem Kontrol dan <i>Monitoring Prototype Monitoring</i> Kinerja Baterai Kendaraan Bermotor.....	23
3.5.1 Rangkaian Elektrik	24
3.5.2 Pembuatan Sistem Kontrol dan <i>Monitoring Prototype</i> Kinerja Baterai Kendaraan Bermotor	25

3.5.2.1 Pembuatan Sistem Kontrol	25
3.5.2.2 Pembuatan Sistem <i>Monitoring</i>	25
3.5.2.3 Uji Coba Sistem Kontrol Dan Monitoring	26
3.6 Uji Coba Keseluruhan	26
3.7 Penyusunan Laporan Proyek Akhir.....	27
BAB IV PEMBAHASAN	
4.1 Deskripsi Alat	28
4.2 Pengujian Sensor.....	29
4.2.1 Sensor <i>Voltage Divider</i>	29
4.2.2 Sensor Arus INA219	31
4.2.3 Sensor Arus ACS12.....	33
4.2.4 Sensor DS18B20	36
4.3 Hasil Tampilan Pada <i>MIT APP Inventor</i>	37
4.4 Hasil Pengujian Sistem <i>Charging</i>	38
4.5 Hasil Pengujian Sistem Proteksi Kelistrikan.....	40
4.6 Hasil Pengujian Keseluruhan	42
BAB V KESIMPULAN DAN SARAN	
5.1 Kesimpulan	45
5.2 Saran	46
DAFTAR PUSTAKA	50
DAFTAR LAMPIRAN	53

DAFTAR TABLE

Tabel

4.1	Pengujian Sensor <i>Voltage Divider</i>	29
4.2	Pengujian Sensor Arus INA219	31
4.3	Pengujian Sensor Arus ACS 712.....	34
4.4	Pengujian Sensor DS18B20	36
4.5	Data Hasil Pengujian Baterai Normal	42
4.6	Data Hasil Pengujian <i>Charging</i> Baterai.....	43

DAFTAR GAMBAR

2.1	Aki Basah dan Aki Kering	4
2.2	Aki Kering	6
2.3	ESP32.....	7
2.4	Relay.....	8
2.5	Aplikasi <i>Mobile</i>	10
2.6	<i>MIT App Inventor</i>	11
2.7	<i>Voltage Sensor</i>	11
2.8	INA219	12
2.9	ACS712.....	13
2.10	DS18B20.....	13
2.11	<i>Real Time Clock (RTC)</i>	14
2.12	<i>Liquid Cristal Display (LCD)</i>	15
3.1	Tahapan-Tahapan Pelaksanaan	17
3.2	Blok Diagram Mekanisme Sistem <i>Monitoring</i> Baterai.....	19
3.3	Flowchart Mekanisme Sistem <i>Monitoring</i> Baterai	19
3.4	Flowchart Mekanisme Sistem <i>Charging</i> Baterai	20
3.5	Blok Diagram Mekanisme Sistem <i>Charging</i> Baterai.....	21
3.6	Flowchart Mekanisme Sistem Kelistrikan Baterai	22
3.7	Blok Diagram Mekanisme Sistem Kelistrikan Baterai	22
3.8	Desain <i>prototype monitoring</i> kinerja baterai kendaraan bermotor.....	23
3.9	Blok Diagram Sistem <i>Monitoring</i>	23

3.10	Skema Rangkaian Elektrik	24
3.11	Pembuatan Sistem Kontrol.....	25
3.12	Tampilan Aplikasi <i>MIT App Inventor</i>	26
4.1	Keseluruhan Alat.....	28
4.2	Rangkaian Pengujian Sensor <i>Voltage Divider</i>	29
4.3	Rangkaian Pengujian Sensor INA219.....	31
4.4	Rangkaian Pengujian Sensor ACS712	34
4.5	Rangkaian Pengujian Sensor DS18B20.....	36
4.6	Tampilan Antarmuka <i>Firebase MIT APP Inventor</i>	38
4.7	Tampilan Antarmuka Aplikasi <i>MIT APP Inventor</i>	38
4.8	Rangkaian Pengujian Sistem <i>Charging</i>	39
4.9	Grafik <i>Monitoring</i> Proses <i>Charging</i>	40
4.10	Rangkaian Pengujian Sistem Kelistrikan	41
4.11	Grafik Sistem Proteksi Kelistrikan.....	41
4.12	Tampilan di LCD Kondisi Baterai Normal	43
4.13	Tampilan di LCD Kondisi Proses <i>Charging</i> Baterai	44

DAFTAR LAMPIRAN

Lampiran 1 : Daftar Riwayat Hidup	50
Lampiran 2 : Program Keseluruhan.....	53



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Pada zaman modern ini, kendaraan bermotor telah menjadi kebutuhan primer dalam mobilitas sehari-hari. Di Indonesia, dari Januari hingga Agustus 2024, jumlah kendaraan bermotor mencapai 4.342.781 unit. Angka ini meningkat 3,1% dibandingkan 4.211.737 unit pada tahun 2023 (AISI – Asosiasi Industri Sepeda Motor Indonesia). Salah satu komponen penting dalam kendaraan bermotor adalah baterai atau aki, yang berfungsi menyimpan energi listrik dalam bentuk energi kimia untuk menyediakan daya bagi sistem starter, pengapian, dan komponen kelistrikan lainnya (Agustian, 2015). Namun, dalam praktiknya, baterai mengalami penurunan kualitas dan memiliki usia pakai yang relatif singkat, sehingga diperlukan perhatian khusus dalam penggunaannya karena harus diganti secara berkala (Kurniawan dan Amarudin, 2025). Menurut AISI, umur pakai baterai atau aki sepeda motor berkisar antara 2 hingga 5 tahun.

Penggunaan baterai pada sepeda motor kerap menimbulkan berbagai permasalahan, terutama disebabkan oleh kurangnya perawatan dan pemantauan terhadap kondisi sistem kelistrikan. Beberapa faktor yang berkontribusi terhadap kerusakan baterai antara lain *overcharging* yang disebabkan oleh kerusakan pada *regulator* (kiprok), *undercharging* akibat *alternator* yang tidak berfungsi secara optimal, serta penggunaan aksesoris listrik yang melebihi kapasitas aki. Selain itu, kondisi suhu lingkungan yang *ekstrem*, korosi pada terminal, serta frekuensi pemakaian yang rendah juga dapat mempercepat penurunan performa dan umur pakai baterai (Agustian, 2015). Apabila energi dalam baterai digunakan hingga 50% atau lebih dari kapasitas totalnya, maka pengisian baterai akan menjadi tidak efisien, sehingga baterai menjadi lebih rentan terhadap kerusakan. Selain itu, pengguna kendaraan tidak menyadari bahwa kondisi baterai telah mencapai level tegangan di bawah rata-rata 85% dari standar level baterai yang normal (Tamara dan Aji, 2021).

Rancangan sistem *monitoring* dengan *mobile* dapat digunakan untuk

mengatasi kendala dalam pemantauan kondisi penggunaan baterai. Memanfaatkan sensor dan teknologi komunikasi, sistem ini dapat memantau kondisi baterai secara *real-time*, seperti tegangan, arus, dan kondisi kelayakan baterai *good*, *weak*, atau *dead*.

Berdasarkan penelitian yang dilakukan oleh Agustian (2015), alat pengecekan kondisi kesehatan baterai telah dibuat dengan menggunakan sistem *monitoring* yang menampilkan data arus, tegangan, dan suhu baterai melalui *LCD*, serta dikendalikan oleh *mikrokontroler ATmega16*. Namun, alat tersebut masih memiliki beberapa kelemahan, khususnya dalam sistem perekaman data secara berkala. Beberapa masalah yang muncul antara lain adalah kurangnya pencatatan waktu yang tepat dan tidak adanya fitur penyimpanan data otomatis. Selain itu, perangkat ini juga belum dilengkapi dengan sistem pemantauan jarak jauh atau aplikasi yang terintegrasi dengan *smartphone*, sehingga pengguna tidak dapat memantau kondisi baterai secara *real-time* dan dengan cara yang praktis. Berdasarkan penelitian sebelumnya, diperlukan sebuah sistem yang dapat memantau kondisi kelayakan baterai, termasuk tegangan, arus, dan suhu, secara *real-time*. Sistem ini penting mengingat baterai merupakan komponen vital dalam kendaraan bermotor yang digunakan oleh banyak orang. Selain itu, hasil pemantauan kondisi baterai harus dapat diakses dan dipantau melalui aplikasi *smartphone* secara *real-time*. Oleh karena itu, proyek akhir kami yang memiliki judul **“Prototype Monitoring Kinerja Baterai Kendaraan Bermotor yang Terintegrasi dengan Aplikasi Mobile”** dilengkapi komponen *mikrokontroler NodeMCU* untuk memantau arus, tegangan yang keluar dan masuk secara *real-time* berbasis *mobile*.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini meliputi :

1. Bagaimana cara merancang dan membuat sistem *monitoring* kondisi baterai kendaraan bermotor secara *real-time* menggunakan sensor tegangan, arus, dan suhu?
2. Bagaimana mengintegrasikan sistem *monitoring* baterai kendaraan bermotor

dengan aplikasi *mobile* agar pengguna dapat memantau kondisi baterai dengan mudah?

3. Bagaimana merancang dan mengembangkan aplikasi menggunakan *MIT App Inventor* untuk *prototype* sistem *monitoring* kinerja baterai kendaraan bermotor yang terintegrasi dengan aplikasi *mobile*?

1.3 Tujuan Penelitian

Tujuan dalam penelitian ini adalah :

1. Membuat program keseluruhan dari sistem *monitoring* baterai untuk memantau kondisi baterai kendaraan bermotor secara *real-time*, dengan mengukur parameter-parameter penting seperti suhu, tegangan, arus, dan proses *charging*.
2. Membuat aplikasi yang intuitif dan mudah digunakan untuk memantau semua parameter baterai dalam satu tampilan.
3. Memberikan informasi mengenai kelayakan baterai secara *real-time*.

1.4 Batasan Masalah

Batasan masalah dalam penelitian ini meliputi :

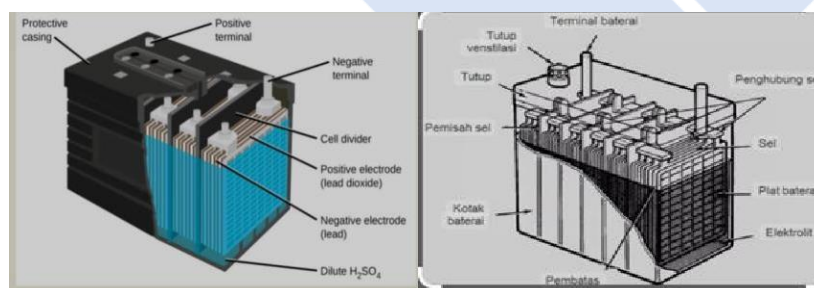
1. *Monitoring* dilakukan melalui aplikasi pada *smartphone* dan tampilan *LCD*.
2. Sistem *monitoring* ini dirancang khusus untuk kendaraan bermotor roda dua (sepeda motor), dengan tujuan untuk memantau parameter-parameter seperti suhu, tegangan, arus, dan proses *charging*.
3. Aplikasi hanya menampilkan status baterai secara *real-time* tanpa fitur penyimpanan data historis atau laporan performa dalam jangka waktu tertentu.

BAB II

DASAR TEORI

2.1 Baterai Kendaraan Sepeda Motor

Baterai kendaraan bermotor memiliki beberapa tipe atau jenis yang digunakan sesuai dengan kebutuhan dan karakteristik kendaraan. Beberapa jenis baterai yang umum digunakan antara lain baterai aki basah, baterai aki kering (*maintenance-free*), dan baterai aki AGM (*Absorbent Glass Mat*). Baterai aki basah merupakan jenis baterai yang menggunakan elektrolit dalam bentuk cair. Jenis ini memerlukan perawatan rutin, seperti pengisian ulang air aki agar tetap berfungsi optimal (Karimah dkk, 2023). Sedangkan baterai aki kering adalah jenis baterai yang tidak memerlukan penambahan air elektrolit secara berkala karena elektrolitnya berbentuk gel atau terserap dalam separator. Hal ini membuat aki kering lebih praktis dan minim perawatan dibandingkan dengan aki basah. Aki kering memiliki ketahanan lebih baik terhadap perubahan. Selain itu, material elektrolitnya mengandung timah kalsium yang membuatnya lebih stabil, tidak mudah menguap, serta lebih aman karena tidak mudah tumpah. Karakteristik ini membuatnya lebih tahan terhadap perubahan suhu dan kondisi lingkungan, sehingga menjadikannya pilihan yang lebih efisien untuk kendaraan (Setiono, 2015).



Gambar 2.1 Aki Basah dan Aki Kering

2.1.1 Aki Kering (*Maintenance-Free*)

Sistem dalam penelitian ini dirancang untuk aki kering karena keunggulannya dalam hal perawatan, keamanan, dan efisiensi pemasangan. Aki kering tidak memerlukan pengisian ulang cairan elektrolit, karena elektrolitnya berbentuk gel atau terserap dalam separator, sehingga tidak mudah menguap atau tumpah. Hal ini membuatnya lebih aman dibandingkan aki basah. Selain itu, desainnya yang ringkas memungkinkan pemasangan yang lebih mudah dan cepat (Sugi dan Sulaiman, 2024). Aki kering memiliki keunggulan dalam ketahanan terhadap guncangan dan posisi miring, sehingga ideal untuk kendaraan bermotor yang sering mengalami getaran atau perubahan posisi. Selain itu, aki ini menghasilkan lebih sedikit korosi karena tidak mengeluarkan uap asam, sehingga area sekitarnya tetap bersih dan komponen lebih awet. Bebasnya aki kering dari perawatan rutin juga memberikan efisiensi waktu dan tenaga bagi pengguna yang mengutamakan kepraktisan (Dhamayanthie dkk, 2023).

Dalam sistem pemantauan aki, parameter utama yang diamati meliputi tegangan, arus, suhu, dan indikator aki. Tegangan dipantau untuk memastikan ketersediaan daya dan mendeteksi kondisi aki, apakah normal, melemah, atau perlu diganti. Arus diamati untuk melihat pola penggunaan daya serta mencegah *over-discharge* yang dapat merusak aki. Suhu digunakan untuk mengidentifikasi potensi *overheating* yang dapat memengaruhi kinerja dan umur aki. Selain itu, indikator aki baik berupa indikator fisik maupun sinyal dari sistem kendaraan memberikan informasi visual mengenai kondisi internal aki. Pemantauan menyeluruh terhadap parameter-parameter ini penting untuk menjaga kesehatan aki kering dan meningkatkan keandalan kendaraan. Menurut studi yang dilakukan oleh Sugi dan Sulaiman pada tahun 2024, pemantauan kondisi aki secara *real-time* dapat meningkatkan efisiensi penggunaan daya hingga 30% dan memperpanjang umur pakai aki hingga 40%.



Gambar 2.2 Aki Kering

2.2 Sistem Kontrol Baterai Sepeda Motor

2.2.1 ESP32

Mikrokontroler ESP32 adalah *mikrokontroler* SoC (*System on Chip*) yang terintegrasi, dilengkapi dengan Wi-Fi 802.11 b/g/n, *Bluetooth* versi 4.2, dan berbagai *peripheral*. ESP32 merupakan *chip* yang cukup komprehensif, yang mencakup prosesor, penyimpanan, dan akses ke GPIO (*General Purpose Input Output*). ESP32 dapat digunakan sebagai pengganti dalam rangkaian Arduino dan memiliki kemampuan untuk terhubung langsung ke Wi-Fi (Wagyuana dan Rahmat, 2019). Spesifikasi dari ESP32 adalah sebagai berikut : *Board* ini tersedia dalam dua versi, yaitu 30 GPIO dan 36 GPIO. Keduanya memiliki fungsi yang serupa, namun versi 30 GPIO dipilih karena dilengkapi dengan dua pin GND. Semua pin diberi label di bagian atas *board*, sehingga mudah dikenali. *Board* ini juga memiliki antarmuka USB to UART yang memudahkan pemrograman menggunakan aplikasi pengembangan seperti Arduino IDE. Sumber daya untuk *board* dapat diberikan melalui konektor micro USB (Nizam dkk, 2022).

Sebagai penguat dasar pemilihan komponen dalam perancangan sistem, salah satu referensi relevan yang penulis temukan dalam jurnal penelitian berjudul “Sistem Kontrol Jarak Jauh PLC Menggunakan ESP32 Berbasis IoT” oleh Sarfini dan Irawan (2024). Dalam jurnal tersebut, *mikrokontroler* ESP32 berfungsi sebagai pengontrol utama. Pemilihan *mikrokontroler* ini didasarkan pada beberapa

pertimbangan. Pertama, karena *mikrokontroler* ini dapat dipantau melalui web yang dapat diakses melalui *smartphone* berkat kemampuan wi-fi yang dimilikinya. Selain itu dapat dikendalikan dengan sistem jarak jauh, dan memiliki konsumsi arus dan daya yang sangat rendah. Terakhir, *mikrokontroler* ini memiliki penyimpanan memori yang memadai.

Berdasarkan penelitian tersebut akan dijadikan sebagai referensi dalam proyek akhir ini, dimana ESP32 akan digunakan sebagai *mikrokontroler* utama yang dapat mendukung dalam pembuatan sistem aplikasi yang terintegrasi dengan android. ESP32 berperan sebagai pusat *monitoring* untuk menampilkan proses *monitoring*, tegangan, arus, suhu, proteksi *charging*, proteksi kelistrikan, dan memproses data yang ditampilkan pada *smartphone*.



Gambar 2.3 ESP32

2.2.2 Relay

Relay adalah saklar (*switch*) yang dapat dioperasikan secara elektrik. Komponen ini bersifat *elektromekanis* dan terdiri dari dua bagian utama, yaitu *elektromagnet (coil)* dan mekanisme kontak saklar. *Elektromagnet* berfungsi sebagai penggerak, sementara itu, kontak saklar berfungsi untuk menghubungkan atau memutuskan arus listrik dalam suatu rangkaian (Saleh dan Haryanti, 2017).

Berdasarkan cara kerjanya, relay dapat dibedakan menjadi beberapa jenis, yaitu *Change Over (CO)*, *Normally Open (NO)*, dan *Normally Closed (NC)*. Relay *Change Over (CO)* memiliki satu kontak saklar yang secara default terhubung ke salah satu terminal, jika kumparan pertama dialiri arus, saklar akan berpindah ke

pengumpulan, peninjauan kembali, pelaporan, dan pengambilan tindakan berdasarkan informasi dari suatu proses yang sedang dilaksanakan (Aditya dkk, 2021).

Dari beberapa definisi di atas, dapat disimpulkan bahwa *monitoring* atau pemantauan adalah proses yang dilakukan secara terus-menerus untuk mengumpulkan data dengan tujuan mengevaluasi pencapaian sasaran. Proses ini mencakup berbagai kegiatan, seperti pengumpulan data, peninjauan ulang, pelaporan, serta tindak lanjut berdasarkan informasi yang diperoleh selama pelaksanaan suatu program atau proyek. *Monitoring* memiliki peran yang krusial dalam memastikan bahwa setiap kegiatan berlangsung sesuai dengan rencana yang telah ditetapkan. Dalam sistem yang penulis kembangkan untuk memantau kondisi kelayakan baterai kendaraan bermotor, NodeMCU ESP32 berfungsi sebagai alat yang mengirimkan semua parameter yang telah diukur ke aplikasi secara *real-time*, dan proses yang akan ditampilkan pada layar LCD maupun aplikasi adalah data tegangan, arus, suhu baterai, dan persentase *charging* baterai.

2.3.1 Aplikasi *Mobile* Berbasis MIT App Inventor

Aplikasi *mobile* terdiri dari dua kata, yaitu aplikasi dan *mobile*. Dalam istilah, aplikasi merujuk pada program yang siap digunakan untuk menjalankan suatu fungsi bagi pengguna atau aplikasi lainnya, sedangkan *mobile* berarti perpindahan dari satu tempat ke tempat lain. Secara lebih rinci, aplikasi *mobile* adalah program yang siap digunakan untuk menjalankan fungsi tertentu yang diinstal pada perangkat *mobile* (Siregar dan Permana, 2016). Aplikasi *mobile* bekerja dengan melibatkan berbagai komponen, seperti antarmuka pengguna (*UI*), *backend server*, *database*, serta koneksi jaringan (internet atau lokal) untuk memastikan kelancaran pertukaran data. Dengan arsitektur yang matang dan optimalisasi pengembangan, aplikasi *mobile* dapat memberikan layanan yang responsif dan efisien, baik di *platform* android maupun iOS, sehingga mampu memenuhi berbagai kebutuhan pengguna di era digital (Harlim dan Setiyawati, 2022).

Aplikasi yang digunakan dalam alat ini dikembangkan dengan menggunakan *MIT App Inventor*, yaitu sebuah *platform* pemrograman visual yang memungkinkan pengguna untuk membuat aplikasi dengan cara yang mudah. Melalui aplikasi ini, pengguna dapat merancang tampilan antarmuka serta menyusun logika pemrograman secara bersamaan, sehingga proses perancangan aplikasi menjadi lebih mudah dan efisien.



Gambar 2.5 Aplikasi *Mobile*

MIT App Inventor adalah salah satu *platform* untuk membuat aplikasi sederhana tanpa perlu mempelajari atau menggunakan banyak bahasa pemrograman. Pengguna dapat merancang aplikasi android sesuai keinginan dan memanfaatkan berbagai *layout* atau komponen yang tersedia. Dengan *MIT App Inventor*, pengguna tidak perlu menguasai bahasa pemrograman, karena pengguna cukup melakukan *drag and drop* objek visual untuk menciptakan aplikasi yang dapat dijalankan di perangkat Android. Salah satu keunggulan *MIT App Inventor* adalah penggunaan tidak perlu mengingat dan menulis instruksi, sehingga tidak akan menimbulkan frustrasi dalam proses pengembangan aplikasi (Edriati dkk, 2021).

Dengan kombinasi kedua *hybrid app* ini, pengguna dapat dengan mudah mengembangkan aplikasi Android yang interaktif dan fungsional sesuai kebutuhan proyek, tanpa harus memahami sintaks pemrograman yang kompleks. Dengan menggunakan aplikasi *MIT App Inventor*, kami merancang tampilan aplikasi yang mampu menampilkan semua parameter yang diperlukan dalam pemantauan kondisi aki, seperti tegangan, arus, suhu, dan persentase pengisian baterai. Dengan demikian, pengguna dapat memantau kondisi baterai dengan lebih baik dan jelas.



Gambar 2.6 MIT App Inventor

2.3.2 Voltage Sensor

Modul sensor tegangan berfungsi untuk mendeteksi dan mengukur tegangan dengan prinsip kerja pembagi tegangan resistor, dimana tegangan *output* modul adalah hasil dari pembagian dengan tegangan *input*. Modul ini memiliki tegangan *input* antara 0-25V DC, tegangan deteksi dari 0.02445-25V DC, ketelitian pengukuran sebesar 0.00489V dan ukuran 25 x 13 mm, sehingga cocok digunakan untuk pemantauan tegangan pada berbagai aplikasi elektronik. Dalam penelitian ini, sensor tegangan digunakan untuk memonitor kondisi tegangan pada baterai guna memastikan kinerjanya tetap optimal dan mendeteksi perubahan tegangan yang dapat mempengaruhi sistem secara keseluruhan, sensor tegangan DC digunakan untuk membaca *output* tegangan dari *thermoelectric generator*. Sensor ini dapat mengukur tegangan maksimum hingga 25 volt dan dipasang secara paralel pada *output* tegangan *thermoelectric* (Fahruri dkk,2021).

Pada penelitian ini, sensor tegangan digunakan untuk mengukur tegangan yang mengalir pada baterai kendaraan bermotor guna memantau kondisi tegangan pada baterai.

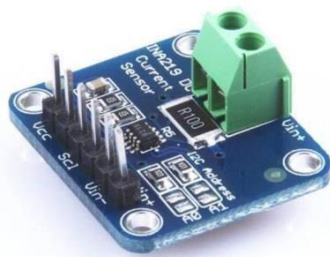


Gambar 2.7 Voltage Sensor

2.3.3 Sensor Arus INA219

Sensor INA219 adalah modul sensor yang dapat mengukur tegangan, arus, dan daya secara bersamaan. Prinsip kerja sensor ini melibatkan deteksi arus listrik

yang mengalir melalui kabel tembaga didalamnya, yang kemudian menghasilkan medan magnet. Medan magnet ini ditangkap oleh IC medan magnet terintegrasi dan diubah menjadi tegangan yang proporsional terhadap arus yang mengalir (Radina dkk, 2022). Dalam penelitian ini, sensor INA219 digunakan untuk mengukur arus dan tegangan yang mengalir pada baterai kendaraan bermotor untuk memantau kondisi dan kinerjanya secara *real-time*. Pemantauan ini bertujuan untuk mencegah terjadinya kelebihan arus serta memastikan kinerja baterai tetap optimal.



Gambar 2.8 INA219

2.3.4 Sensor Arus ACS712

ACS712 adalah sensor arus berbasis *Hall Effect* yang dirancang untuk mengukur arus AC maupun DC dengan tingkat akurasi tinggi. Sensor ini bekerja dengan mendeteksi medan magnet yang dihasilkan oleh aliran arus listrik, lalu mengonversinya menjadi tegangan analog yang sebanding dengan arus yang diukur. ACS712 banyak digunakan dalam berbagai aplikasi, seperti pengendalian motor, pemantauan beban listrik, catu daya beralih (*switched-mode power supply*), serta perlindungan terhadap beban berlebih, sehingga menjadikannya komponen andal dalam sistem pemantauan arus listrik, sensor arus yang digunakan sensor arus ACS712 amper karena sensor ini memiliki ketepatan pembacaan yang cukup baik karena didalamnya terdapat rangkaian *low offset linier hall* dengan satu lintasan yang terbuat dari tembaga (Ratnasari dan Senen, 2017). Pada penelitian ini, sensor ACS712 digunakan untuk mengukur arus yang mengalir pada baterai kendaraan bermotor guna memantau kondisi dan kinerjanya secara *real-time* untuk menghindari terjadinya kelebihan arus.



Gambar 2.9 ACS712

2.3.5 Sensor Suhu DS18B20

DS18B20 adalah sensor suhu digital terbaru dari seri Maxim IC. Sensor ini dapat mengukur suhu dengan ketelitian antara 9 hingga 12-bit, rentang -50°C sampai 125°C dengan ketelitian $\pm 0.1^{\circ}\text{C}$. Sensor suhu umumnya memerlukan ADC dan beberapa pin port pada *mikrokontroler*, tetapi DS18B20 tidak memerlukan ADC untuk berkomunikasi dengan *mikrokontroler* dan hanya membutuhkan satu kabel saja (Saputro dan Prasetya, 2022). Karakteristik sensor ini meliputi penggunaan pada tegangan 3-5V, tingkat akurasi kesalahan sebesar $\pm 0,5\%$ dengan kisaran suhu antara -10°C hingga 85°C . Kabel merah pada sensor DS18B20 digunakan untuk VCC, kabel hitam untuk GND, dan kabel kuning untuk data (Aritonang dkk, 2021).

Berdasarkan penelitian di atas yang menggunakan sensor DS18B20, penulis juga menggunakan sensor tersebut untuk mengetahui kondisi suhu aki. Selain memiliki tingkat ketelitian yang baik, sensor ini juga dapat dirangkai dengan komponen lain tanpa memerlukan ADC untuk berkomunikasi dengan *mikrokontroler* serta tidak membutuhkan banyak pin port pada *mikrokontroler*.



Gambar 2.10 DS18B20

2.3.6 Real Time Clock (RTC)

Real-Time Clock (RTC) adalah sebuah *chip* jam elektronik yang dirancang untuk menghitung dan menyimpan informasi waktu secara *real-time*, mulai dari

detik hingga tahun, dengan tingkat akurasi yang tinggi. RTC tetap dapat menjaga pencatatan waktu meskipun perangkat utama dimatikan, sehingga memastikan data waktu tetap akurat. Setelah proses perhitungan waktu selesai, data yang dihasilkan akan langsung disimpan atau dikirim ke perangkat lain melalui sistem antarmuka untuk digunakan dalam berbagai aplikasi. Modul *Real Time Clock* (RTC) pada penelitian ini berfungsi sebagai *timer* untuk mengatur penghidupan dan pemadaman lampu (Iqbar dan Kartika, 2020). Dalam penelitian ini, RTC digunakan untuk mencatat waktu pengecekan baterai secara berkala dan *real-time*, sehingga memungkinkan pemantauan kondisi baterai dengan lebih sistematis dan efisien.



Gambar 2.11 *Real-Time Clock* (RTC)

2.3.7 *Liquid Cristal Display* (LCD)

Liquid Crystal Display (LCD) adalah salah satu jenis layar elektronik yang memanfaatkan teknologi CMOS *logic* dan beroperasi tanpa menghasilkan cahaya sendiri, melainkan dengan memantulkan cahaya dari lingkungan sekitar (*front-lit*) atau mentransmisikan cahaya dari belakang (*back-lit*). LCD berfungsi sebagai media untuk menampilkan data dalam berbagai bentuk, seperti karakter, huruf, angka, dan grafik, sehingga banyak digunakan dalam berbagai perangkat elektronik (Bawotong dkk, 2015). Dalam penelitian ini, *LCD* digunakan untuk menampilkan seluruh parameter hasil pemantauan baterai secara manual, sebagai alternatif selain menggunakan *smartphone*, sehingga pengguna dapat melihat informasi langsung dari perangkat tanpa perlu mengakses aplikasi seluler.



Gambar 2.12 *Liquid Crystal Display (LCD)*

2.4 Sistem *Charging* Baterai

Proses *charging* baterai merupakan tahapan penting dalam siklus penggunaan baterai yang harus dilakukan dengan tepat agar tidak menimbulkan kerusakan. Selama proses ini, pengisian daya harus dikendalikan agar tidak terjadi *overcharging* yang dapat menyebabkan peningkatan suhu berlebih atau kerusakan sel baterai. Oleh karena itu, pemantauan parameter seperti tegangan, arus, dan suhu sangat penting untuk memastikan pengisian berjalan dalam batas aman. Dalam hal ini, penggunaan perangkat seperti ESP32 menjadi solusi modern yang efisien karena dapat memantau parameter-parameter tersebut secara *real-time*. Dengan sistem pemantauan ini, pengguna dapat mengetahui kondisi pengisian baterai secara aktual dan mengambil langkah-langkah pencegahan jadi terdeteksi adanya penyimpangan dari nilai normal (Prabowo dkk, 2022).

Prinsip kerja dari alat ini adalah memantau kapasitas baterai selama proses pengisian (*charging*) dan pengosongan (*discharging*) dengan cara mengamati nilai tegangan dan arus, serta menghitung *State of Charge* (SOC) dalam satuan persen (%), sesuai dengan algoritma yang telah diprogram. Hasil pemantauan ini ditampilkan secara langsung melalui layar *LCD* dan juga pada *smartphone* pengguna. Estimasi nilai SOC yang akurat dilakukan saat proses *charging* dan *discharging* berlangsung, baik melalui pengujian langsung pada alat maupun dengan menggunakan metode *modified coulomb counting* untuk memperoleh hasil pengukuran yang lebih presisi. Nilai SOC yang direkomendasikan adalah 100% saat baterai dalam kondisi penuh (*charging*), dan 20% sebagai batas minimum saat baterai digunakan (*discharging*) guna menjaga performa dan umur pakai baterai

(Utomo dkk,2023).

2.5 Proteksi Sistem Kelistrikan

Pada sistem kelistrikan motor ini, relay berfungsi sebagai perangkat perlindungan yang bertugas untuk mendeteksi dan memutus aliran listrik saat terjadi gangguan pada rangkaian kelistrikan. Relay adalah komponen elektronik yang beroperasi secara otomatis sebagai saklar yang dikendalikan oleh sinyal listrik. Ketika sistem mendeteksi kondisi abnormal, seperti arus lebih (*overcurrent*) atau hubungan arus pendek (*short circuit*), relay akan segera memutuskan aliran listrik untuk mencegah kerusakan pada komponen lainnya (Turang, 2015).

Selain relay, sistem ini juga dilengkapi dengan fuse (*sekring*) sebagai lapisan perlindungan tambahan terhadap gangguan arus listrik. Fuse dirancang untuk bekerja satu kali, yaitu dengan memutuskan aliran listrik secara permanen saat arus melebihi batas toleransi yang telah ditentukan. Fungsi utama fuse adalah untuk melindungi komponen sensitif dari potensi kerusakan akibat lonjakan arus yang tiba-tiba dan berbahaya (Kharisma dan Nazharullah, 2023).

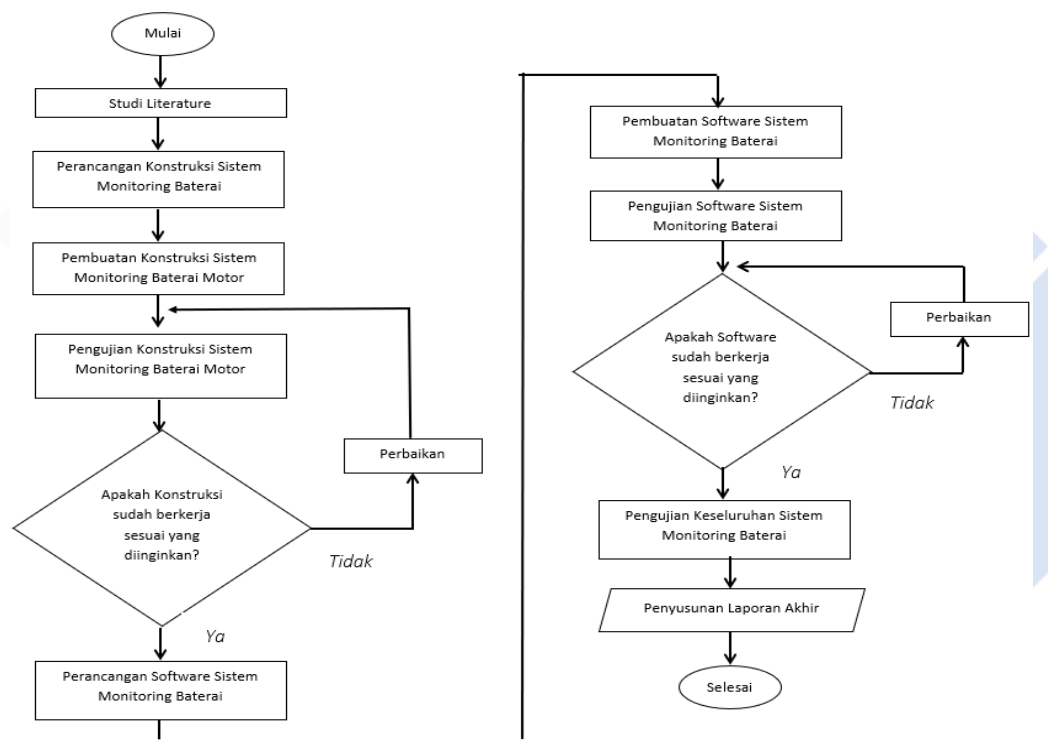
Dalam penelitian ini, dilakukan pengujian terhadap pengaturan (*setting*) relay yang berfungsi untuk melindungi sistem kelistrikan motor dari gangguan arus lebih dan hubungan singkat. Kombinasi antara relay dan fuse membentuk sistem proteksi yang andal dalam menjaga kestabilan dan keamanan kelistrikan. Dengan adanya sistem proteksi ini, apabila terjadi gangguan seperti *short circuit*, maka aliran listrik akan segera terputus sebelum mencapai baterai, sehingga kondisi baterai tetap aman dan tidak terdampak langsung oleh gangguan tersebut.

BAB III

METODE PELAKSANAAN

3.1 Tahapan Perancangan dan Pembuatan Alat

Tahapan-tahapan penelitian yang akan dilakukan meliputi studi literatur, perancangan sistem, pembuatan sistem, pengumpulan data, dan uji coba, yang dapat dilihat pada gambar berikut.



Gambar 3.1 Tahapan-tahapan Pelaksanaan

3.2 Study literature

Studi literature adalah tahap untuk mengumpulkan berbagai informasi yang relevan dalam menyelesaikan proyek akhir ini. Data yang diperoleh kemudian dianalisis guna menentukan aspek-aspek yang akan diterapkan pada proyek akhir. Pada tahap ini, penulis melakukan studi pustaka, yaitu dengan mencari, membaca, dan mempelajari berbagai referensi yang berkaitan dengan proyek akhir. Referensi tersebut meliputi jurnal dan buku yang membahas penelitian terkait

monitoring baterai menggunakan ESP32, pemantauan menggunakan aplikasi *MIT App Inventor*, serta literatur lain yang mendukung dasar teori dalam proyek ini.

3.3 Mekanisme *Prototype Monitoring Kinerja Baterai Kendaraan Bermotor* yang Terintegrasi dengan Aplikasi *Mobile*

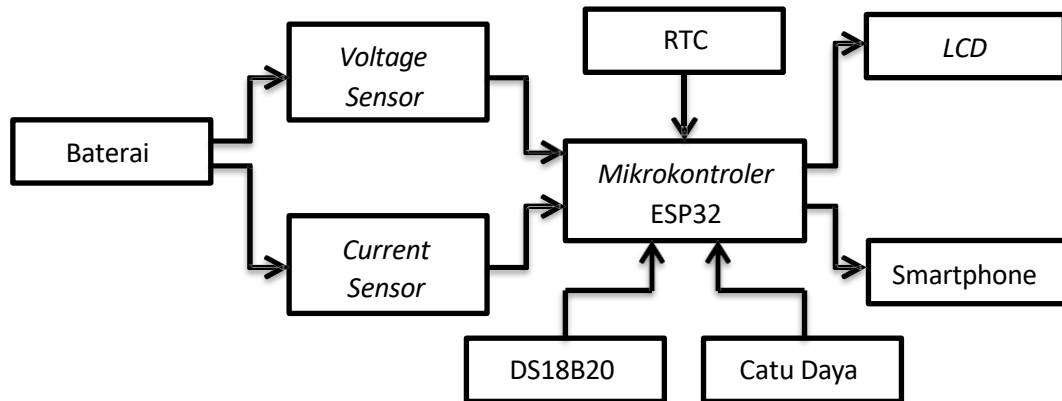
Sistem *monitoring* baterai ini dirancang untuk memantau kondisi kelayakan baterai kendaraan secara *real-time* dengan memanfaatkan sensor tegangan, sensor arus, dan sensor suhu DS18B20. Sensor-sensor ini membaca parameter penting dari baterai, seperti tegangan, arus, dan suhu, lalu mengirimkan data ke *mikrokontroler* ESP32 untuk diproses. Jika terdeteksi kondisi tidak normal, seperti tegangan lemah, arus berlebih, suhu tinggi, atau hubungan arus pendek (*short circuit*), sistem akan memberikan notifikasi langsung kepada pengguna melalui aplikasi *mobile* yang terhubung. Selain notifikasi, sistem juga memiliki proteksi aktif berupa relay yang secara otomatis memutus aliran arus saat terjadi *overcurrent*, *short circuit*, atau *overcharging*, guna mencegah kerusakan lebih lanjut pada baterai. Seluruh data pemantauan ditampilkan secara langsung di aplikasi *smartphone* dan layar *LCD*, sehingga pengguna dapat memantau dan menangani kondisi baterai dengan lebih cepat dan efisien.

3.3.1 Mekanisme Sistem *Monitoring Baterai*

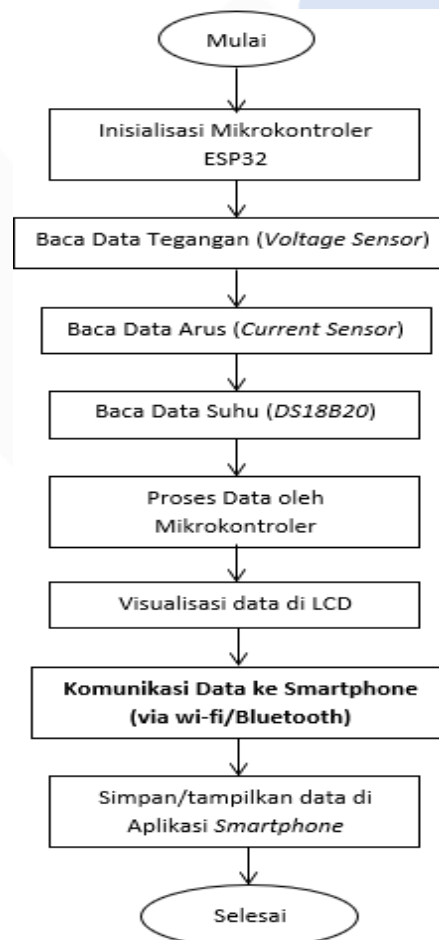
Mekanisme *monitoring* baterai pada sistem ini dimulai ketika sensor tegangan (*voltage sensor*) dan sensor arus (*current sensor*) membaca nilai tegangan serta arus yang berasal dari baterai. Selain itu, sistem ini juga dilengkapi dengan sensor suhu DS18B20 yang berfungsi untuk memantau perubahan suhu pada baterai. Semua data yang diperoleh dari ketiga sensor tersebut kemudian diproses oleh *mikrokontroler* ESP32 (Hilmansyah dkk, 2020).

Data hasil pemrosesan tersebut selanjutnya ditampilkan secara *real-time* melalui dua media, yaitu layar *LCD* yang terpasang pada perangkat serta aplikasi *mobile* yang dikembangkan menggunakan *MIT App Inventor*. Dengan demikian, pengguna dapat memantau kondisi baterai secara langsung, baik melalui perangkat

keras maupun aplikasi *mobile* (Hilmansyah dkk, 2020).



Gambar 3.2 Blok Diagram Mekanisme Sistem *Monitoring* Baterai



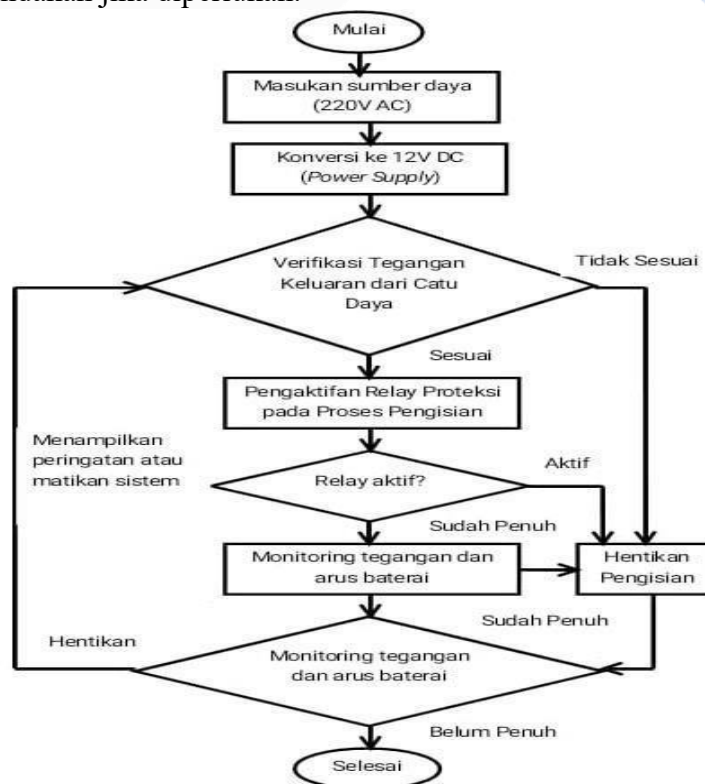
Gambar 3.3 Flowchart Mekanisme Sistem *Monitoring* Baterai

3.3.2 Mekanisme Sistem *Charging* Baterai

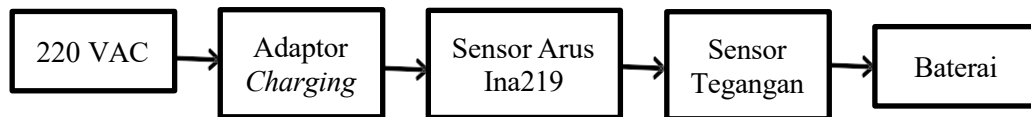
Mekanisme kerja alat ini dimulai dari baterai yang terhubung ke rangkaian

sensor tegangan, arus, dan suhu. Ketiga sensor tersebut secara berlanjut membaca nilai parameter dari baterai selama proses pengisian (*charging*) maupun pengosongan daya (*discharging*). Data yang diperoleh dari sensor selanjutnya dikirim ke *mikrokontroler* ESP32. Di dalam ESP32, data tersebut diproses menggunakan algoritma *modified coulomb counting* untuk menghitung nilai SoC dalam bentuk persentase (%). Nilai SoC digunakan untuk mengetahui kapasitas baterai saat ini, dan sistem akan menetapkan ambang batas aman, yaitu 100% saat *charging* dan 20% saat proses *discharging* (Utomo dkk, 2023).

Jika ESP32 mendeteksi kondisi tegangan atau arus berada di luar batas aman, maka *mikrokontroler* akan segera mengaktifkan sistem proteksi, seperti memutus arus pengisian untuk mencegah *overcharging*, atau memberikan notifikasi ketika terjadi *overdischarging*. Hasil pemantauan ini ditampilkan secara *real-time* melalui layar LCD dan aplikasi *mobile* yang dikembangkan menggunakan *MIT App Inventor*, sehingga pengguna dapat langsung melihat kondisi baterai dan mengambil tindakan jika diperlukan.



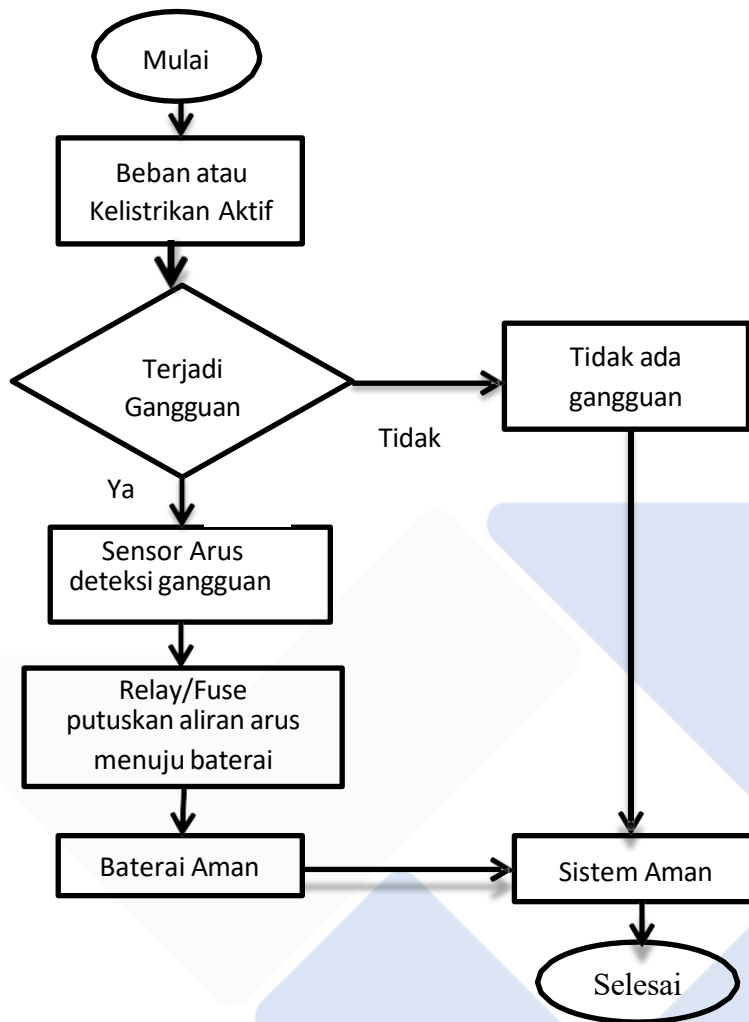
Gambar 3.4 Flowchart Mekanisme Sistem *Charging* Baterai



Gambar 3.5 Blok Diagram Mekanisme Sistem *Charging* Baterai

3.3.3 Mekanisme Sistem Kelistrikan Baterai

Mekanisme sistem kelistrikan baterai pada kendaraan bermotor dirancang untuk menjamin distribusi daya yang aman dan stabil ke seluruh komponen kelistrikan. Baterai berfungsi sebagai sumber utama energi listrik pada sepeda motor. Untuk melindungi sistem dari gangguan kelistrikan seperti arus lebih (*overcurrent*) atau hubungan pendek (*short circuit*), digunakan perangkat proteksi berupa relay dan fuse (*sekring*). Relay berfungsi sebagai saklar otomatis yang akan memutus aliran listrik ketika terdeteksi adanya kondisi abnormal dalam sistem. Sementara itu, fuse bekerja sebagai pemutus arus satu kali pakai yang akan putus secara permanen jika arus melebihi batas toleransi. Ketika terjadi gangguan, relay akan secara otomatis memutus sirkuit untuk mencegah kerusakan lanjutan, dan jika lonjakan arus sangat besar, fuse akan langsung terputus guna melindungi komponen yang lebih sensitif. Sistem ini memastikan bahwa gangguan pada kelistrikan tidak berdampak langsung terhadap kondisi baterai, serta memberikan waktu dan perlindungan yang cukup untuk penanganan lebih lanjut (Otong dan Albantani, 2024).



Gambar 3.6 Flowchart Mekanisme Sistem Kelistrikan Baterai



Gambar 3.7 Blok Diagram Mekanisme Sistem Kelistrikan Baterai

3.4 Perancangan Konstruksi *Prototype Monitoring Kinerja Baterai Kendaraan Bermotor*

Perancangan desain konstruksi *prototype* untuk *monitoring* kinerja baterai kendaraan bermotor merupakan tahapan penting dalam pembuatan alat yang akan digunakan dalam proyek akhir. Proses perancangan ini dilakukan dengan bantuan perangkat lunak yang sesuai, dan bahan yang digunakan untuk pembuatan

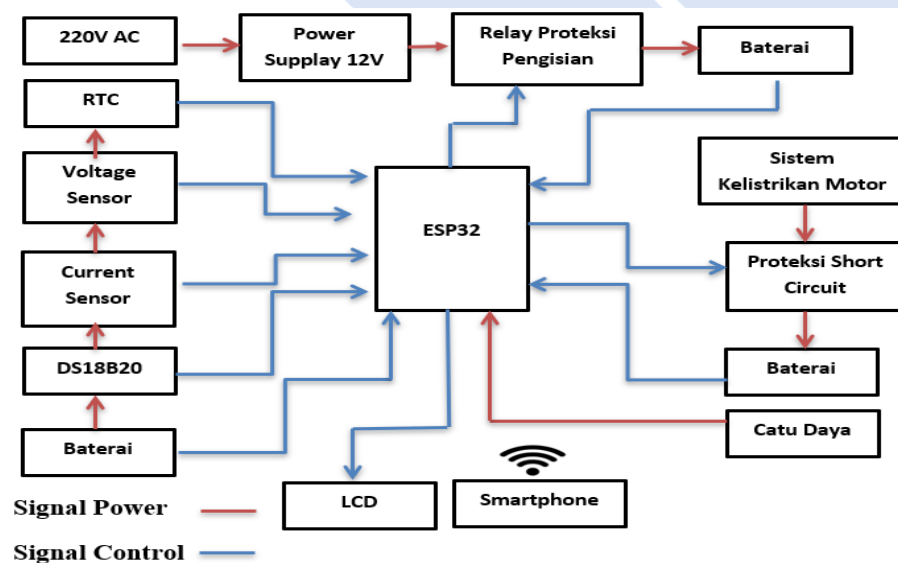
prototype ini adalah bahan plastik. Desain konstruksi *prototype monitoring* kinerja baterai kendaraan bermotor dapat dilihat pada gambar di bawah ini.



Gambar 3.8 Desain *prototype monitoring* kinerja baterai kendaraan bermotor

3.5 Perancangan Sistem Kontrol dan *Monitoring Prototype* Kinerja Baterai

Perancangan sistem kontrol dan pemantauan yang terintegrasi dengan perangkat *mobile* dilakukan menggunakan *software MIT App Inventor*. Data yang diperoleh dari berbagai sensor akan diproses oleh ESP32, dan informasi tersebut kemudian akan dikirim ke *smartphone* untuk diakses oleh pengguna. Selain itu, pengguna juga dapat memantau informasi tersebut melalui tampilan layar LCD. Blok diagram sistem *monitoring* dapat dilihat pada gambar di bawah ini.



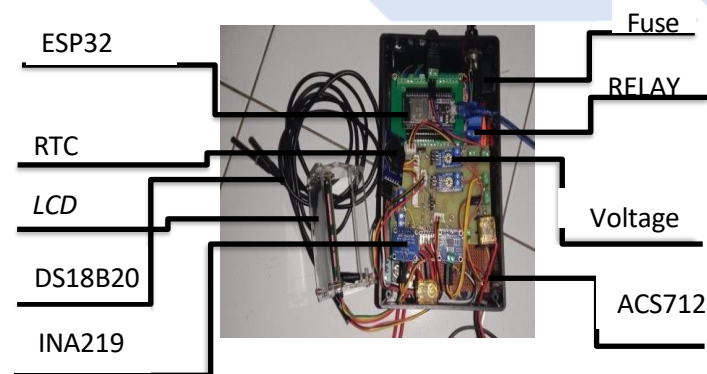
Gambar 3.9 Blok Diagram Sistem *Monitoring*

Keterangan :

1. 220V AC digunakan sebagai sumber energi listrik untuk menyuplai *power supply*.
2. *Power Supply* digunakan untuk mengubah tegangan listrik dari 220V AC menjadi 12V DC.
3. Catu daya berfungsi untuk menyuplai daya ke ESP32.
4. Proteksi *short circuit* dan relay proteksi berfungsi untuk memutus aliran listrik ketika terjadi arus berlebih pada rangkaian kelistrikan motor, serta saat terjadi *overcharging* atau *discharging*.
5. *Voltage sensor* digunakan untuk memantau tegangan pada baterai kendaraan.
6. *Current sensor* digunakan untuk memantau arus pada baterai kendaraan.
7. DS18B20 digunakan untuk memantau suhu pada baterai kendaraan.
8. *RTC (Real-Time Clock)* digunakan untuk menampilkan waktu secara *real-time* dengan akurasi yang tinggi.
9. *LCD* digunakan untuk menampilkan nilai *output* hasil *monitoring*.
10. ESP32 sebagai *mikrokontroler* dan juga sebagai penghubung alat dan *smartphone* melalui jaringan internet.

3.5.1 Rangkaian Elektrik

Pembuatan sistem kontrol dan *monitoring* dilakukan menggunakan pemrograman pada *software tinkercad*. Dengan skema *wiring* sebagai berikut.



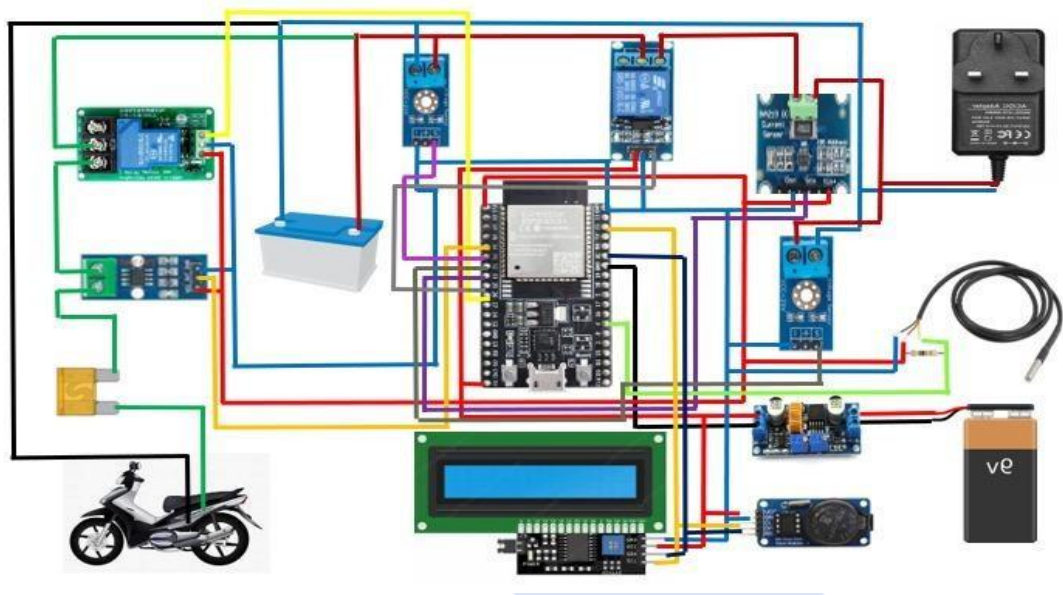
Gambar 3.10 Skema Rangkaian Elektrik

3.5.2 Pembuatan Sistem Kontrol dan *Monitoring Prototype* Kinerja Baterai Kendaraan Bermotor

Program dan rangkaian yang telah dirancang kemudian diimplementasikan dengan memanfaatkan data dari sensor sebagai masukan. Data yang diperoleh diproses dan ditampilkan pada layar LCD serta aplikasi di *smartphone* untuk mempermudah pemantauan.

3.5.2.1 Pembuatan Sistem Kontrol

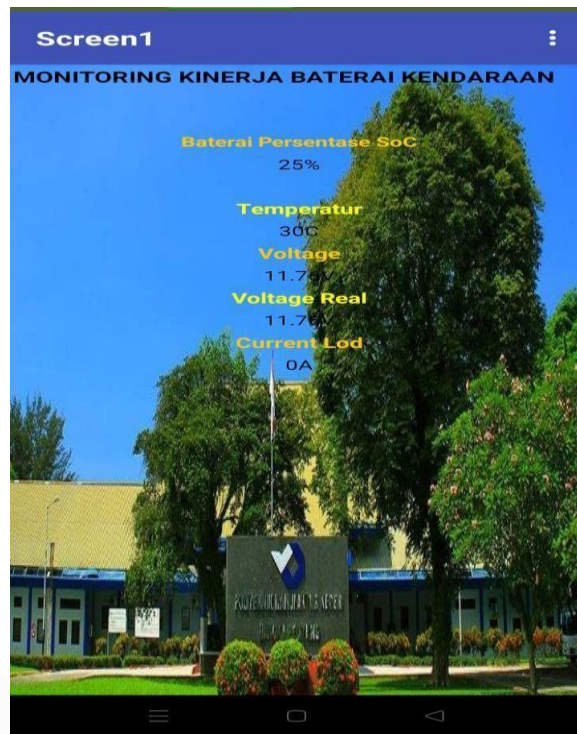
Sistem kontrol dibuat dengan cara menghubungkan komponen-komponen seperti ESP32, sensor tegangan, sensor INA219, sensor ACS712, sensor DS18B20, RTC, *LCD*, sesuai rangkaian yang telah dibuat.



Gambar 3.11 Pembuatan Sistem Kontrol

3.5.2.2 Pembuatan Sistem *Monitoring*

Sistem *monitoring* dibuat dengan menggunakan aplikasi *MIT App Inventor*. Antarmuka pada aplikasi dirancang sesuai dengan kebutuhan, seperti nilai tegangan, arus, suhu, dan persentase baterai. Gambar hasil perancangan sistem *monitoring* pada aplikasi *MIT App Inventor* dapat dilihat pada gambar berikut.



Gambar 3.12 Tampilan Aplikasi *MIT App Inventor*

3.5.2.3 Uji Coba Sistem Kontrol Dan Monitoring

Pengujian sistem kontrol dan *monitoring* dilakukan untuk menilai kinerja sistem secara akurasi data yang diperoleh. Proses pengujian ini melibatkan pengukuran besaran pada masing-masing sensor menggunakan alat ukur manual, kemudian hasilnya dibandingkan dengan data yang ditampilkan pada aplikasi yang dikembangkan menggunakan *MIT App Inventor*. Perbandingan ini bertujuan untuk memastikan bahwa sistem mampu memberikan informasi yang tepat dan dapat diandalkan dalam memantau kondisi aki kendaraan.

3.6 Uji Coba Keseluruhan

Setelah seluruh proses konstruksi, sistem kontrol, dan *monitoring* pada *prototype* kinerja baterai kendaraan bermotor yang terintegrasi dengan aplikasi *mobile* selesai, dilakukan uji coba untuk mengevaluasi kinerja sistem secara keseluruhan. Uji coba ini bertujuan untuk memastikan bahwa setiap komponen, baik perangkat keras maupun perangkat lunak, telah berfungsi sesuai dengan

spesifikasi desain yang sudah direncanakan. Selain itu, pengujian ini juga dilakukan untuk menilai keandalan serta akurasi sistem dalam memantau kondisi baterai kendaraan secara *real-time*.

3.7 Penyusunan Laporan Proyek Akhir

Penyusunan laporan proyek akhir ini adalah tahap akhir dalam pelaksanaan proyek yang bertujuan untuk mendokumentasikan seluruh proses dan hasil penelitian secara sistematis dan komprehensif. Laporan ini mencakup berbagai aspek penting, seperti latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, landasan teori, metode penelitian, pembahasan, hasil yang diperoleh, serta kesimpulan dan saran. Dengan adanya laporan ini, semua tahapan penelitian dapat dipertanggungjawabkan secara ilmiah dan menjadi referensi untuk penelitian selanjutnya.

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini, penulis menjelaskan dan menguraikan langkah-langkah dalam pembuatan penelitian tugas akhir yang berjudul “*Prototype Monitoring Kinerja Baterai Kendaraan Bermotor yang Terintegrasi dengan Aplikasi Mobile*”. Penulis memaparkan proses pembuatan dan pengujian alat, baik pada sistem perangkat keras maupun perangkat lunak, sebagai bagian dari proyek penelitian akhir.

4.1 Deskripsi Alat

Sistem *monitoring* aki motor ini dirancang untuk memantau empat parameter utama, yaitu arus, tegangan, suhu baterai, dan proses pengisian daya (*charging*). Pemantauan arus dan tegangan dilakukan dengan menggunakan sensor INA219 dan sensor *voltage divider*, sementara suhu baterai dipantau menggunakan sensor DS18B20.

Selain melakukan *monitoring*, sistem ini juga dilengkapi dengan fitur proteksi terhadap gangguan kelistrikan, seperti hubung singkat (*short circuit*) dan arus berlebih, yang dikendalikan menggunakan fuse dan relay sebagai sistem pengaman. Seluruh data hasil pemantauan diproses oleh *mikrokontroler* ESP32, kemudian ditampilkan melalui dua media, yaitu layar LCD untuk tampilan visual langsung dan aplikasi *mobile* berbasis *MIT App Inventor* untuk pemantauan melalui *smartphone*. Agar pemantauan dapat dilakukan secara *real-time*, sistem ini dilengkapi dengan modul RTC (*Real-Time Clock*).



Gambar 4.1 Keseluruhan Alat

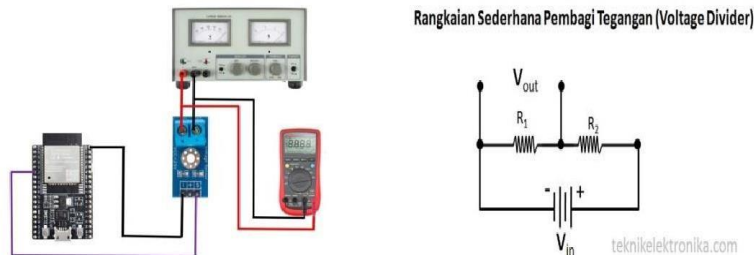
4.2 Pengujian Sensor

4.2.1 Sensor *Voltage Divider*

Pengujian sensor tegangan ini dilakukan untuk memastikan bahwa sensor tegangan yang digunakan beroperasi dengan baik. Pengujian sensor ini dilakukan dengan cara menghubungkan modul sensor tegangan dengan *power supply* sebagai alat yang di uji tegangannya. Hasil pengujian sensor tegangan dibandingkan dengan nilai tegangan asli yang berada pada *power supply* dengan serial monitor hasil dari pembacaan sensor, serta pada pengujian sensor tegangan ini juga menggunakan pembagi tegangan dengan menggunakan rumus perhitungan pembagi tegangan berikut.

Rumus Pembagi Tegangan : $V_{out} = V_{in} \times (R_2 / (R_1 + R_2))$

Rumus Pembagi Arus : $I_x = \frac{R_t}{R_x + R_T} I_T$



Gambar 4.2 Rangkaian Pengujian Sensor *Voltage Divider*

Data hasil pengukuran yang menggunakan sensor tegangan dengan variasi nilai tegangan dapat dilihat pada tabel 4.1 berikut.

Tabel 4.1 Pengujian Sensor *Voltage Divider*

<i>Power Supply</i> (V)	Multimeter (V)	Sensor Tegangan (V)	<i>Error</i> (%)
2V	2,00	1,47	26,5
3V	3,08	2,71	13,65
4V	4,05	3,84	5,47
5V	5,00	4,98	0,40
6V	6,00	6,17	2,76

7V	7,00	7,17	2,37
8V	8,00	8,06	0,74
9V	9,00	9,05	0,55
10V	10,00	9,99	0,10
11V	11,00	11,00	0
12V	12,00	12,00	0

Dari data hasil pengujian pada tabel 4.1, terlihat bahwa terdapat perbedaan antara tegangan dari *power supply*, hasil pengukuran menggunakan multimeter, dan nilai yang terbaca oleh sensor. Oleh karena itu, perlu dilakukan perhitungan persentase kesalahan dari sensor yang digunakan. Berikut adalah rumus yang digunakan untuk menghitung persentase kesalahan.

$$\%Error = \left| \frac{\text{Hasil pembacaan sensor} - \text{Hasil pengukuran multimeter}}{\text{Hasil pembacaan sensor}} \right| \times 100\%$$

Berikut perhitungan persentase *error* dari tabel 4.1

1. Perhitungan persentase *error* pada tegangan 2V

$$\%Error = \left| \frac{1,47 - 2,00}{1,47} \right| \times 100\% = 26,5\%$$

2. Perhitungan persentase *error* pada tegangan 3V

$$\%Error = \left| \frac{2,71 - 3,08}{2,71} \right| \times 100\% = 13,65\%$$

3. Perhitungan persentase *error* pada tegangan 4V

$$\%Error = \left| \frac{3,84 - 4,05}{3,84} \right| \times 100\% = 5,47\%$$

4. Perhitungan persentase *error* pada tegangan 5V

$$\%Error = \left| \frac{4,98 - 5,00}{4,98} \right| \times 100\% = 0,40\%$$

5. Perhitungan persentase *error* pada tegangan 6V

$$\%Error = \left| \frac{6,17 - 6,00}{6,17} \right| \times 100\% = 2,76\%$$

6. Perhitungan persentase *error* pada tegangan 7V

$$\%Error = \left| \frac{7,17 - 7,00}{7,17} \right| \times 100\% = 2,37\%$$

7. Perhitungan persentase *error* pada tegangan 8V

$$\%Error = \left| \frac{8,06 - 8,00}{8,06} \right| \times 100\% = 0,74\%$$

8. Perhitungan persentase *error* pada tegangan 9V

$$\%Error = \frac{|9,05-9,00|}{9,05} \times 100\% = 0,55 \%$$

9. Perhitungan persentase *error* pada tegangan 10V

$$\%Error = \frac{|9,99-10,00|}{9,99} \times 100\% = 0,10 \%$$

10. Perhitungan persentase *error* pada tegangan 11V

$$\%Error = \frac{|11,00-11,00|}{11,00} \times 100\% = 0\%$$

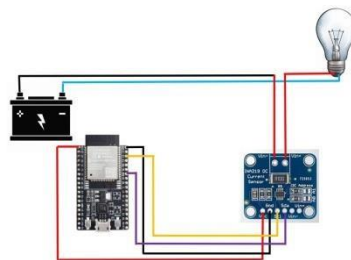
11. Perhitungan persentase *error* pada tegangan 12V

$$\%Error = \frac{|12,00-12,00|}{12,00} \times 100\% = 0\%$$

Berdasarkan data hasil pengujian sensor tegangan pada tabel 4.1, diperoleh rata-rata persentase *error* sebesar 4,78%. Persentase *error* tersebut disebabkan oleh ketidakstabilan nilai tegangan sumber yang dibaca oleh sensor. Meskipun demikian, sensor pembagi tegangan ini masih bisa digunakan karena persentase kesalahannya tetap di bawah 10%.

4.2.2 Sensor Arus INA219

Pada proyek akhir ini, sensor arus INA219 digunakan untuk mengukur arus yang mengalir selama proses pengisian (*charging*). Dalam pengujian, sensor dipasang secara seri dengan beban, sehingga arus yang mengalir dapat diukur secara langsung. Beban yang digunakan dalam pengujian ini adalah sebuah lampu, yang berfungsi sebagai elemen resistif agar nilai arus yang mengalir dapat dihitung dan dianalisis dengan akurat. Rangkaian uji coba sensor arus INA219 dapat dilihat pada Gambar 4.3 berikut.



Gambar 4.3 Rangkaian Pengujian Sensor INA219

Data hasil pengujian sensor arus INA219 dapat dilihat pada tabel 4.2 di bawah ini.

Tabel 4.2 Pengujian Sensor Arus INA219

<i>Power Suplly (V)</i>	<i>Ampermeter (A)</i>	<i>Mikro (A)</i>	<i>Error (%)</i>
7,5	0,000	0,003	100
8,4	0,01	0,04	75
8,5	0,04	0,05	20
8,6	0,07	0,08	12,5
9	0,13	0,126	3,17
9,2	0,15	0,149	0,67
9,5	0,21	0,207	1,45
9,7	0,24	0,24	0
10	0,28	0,28	0
11	0,35	0,35	0

Dari data hasil pengujian pada tabel 4.2, terlihat bahwa terdapat perbedaan nilai arus antara ampermeter dan hasil pembacaan sensor. Oleh karena itu, perlu dilakukan perhitungan persentase *error* dari sensor arus yang digunakan. Berikut adalah rumus dan perhitungan yang digunakan untuk menghitung persentase *error*.

$$\%Error = \left| \frac{\text{Hasil pembacaan sensor} - \text{Hasil pengukuran Ampermeter}}{\text{Hasil pembacaan sensor}} \right| \times 100\%$$

Berikut perhitungan persentase *error* dari tabel 4.2

1. Perhitungan persentase *error* pada tegangan 7,5V

$$\%Error = \left| \frac{0,003 - 0,000}{0,003} \right| \times 100\% = 100\%$$

2. Perhitungan persentase *error* pada tegangan 8,4V

$$\%Error = \left| \frac{0,04 - 0,01}{0,04} \right| \times 100\% = 75\%$$

3. Perhitungan persentase *error* pada tegangan 8,5V

$$\%Error = \left| \frac{0,05 - 0,04}{0,05} \right| \times 100\% = 20\%$$

4. Perhitungan persentase *error* pada tegangan 8,6V

$$\%Error = \left| \frac{0,08 - 0,07}{0,08} \right| \times 100\% = 12,5\%$$

5. Perhitungan persentase *error* pada tegangan 9V

$$\%Error = \left| \frac{0,126-0,13}{0,126} \right| \times 100\% = 3,17\%$$

6. Perhitungan persentase *error* pada tegangan 9,2V

$$\%Error = \left| \frac{0,149-0,15}{0,149} \right| \times 100\% = 0,67\%$$

7. Perhitungan persentase *error* pada tegangan 9,5V

$$\%Error = \left| \frac{0,207-0,21}{0,207} \right| \times 100\% = 1,45\%$$

8. Perhitungan persentase *error* pada tegangan 9,7V

$$\%Error = \left| \frac{0,24-0,24}{0,24} \right| \times 100\% = 0\%$$

9. Perhitungan persentase *error* pada tegangan 10V

$$\%Error = \left| \frac{0,28-0,28}{0,28} \right| \times 100\% = 0\%$$

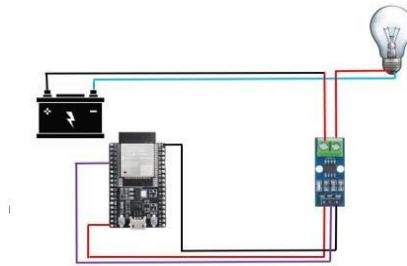
10. Perhitungan persentase *error* pada tegangan 11V

$$\%Error = \left| \frac{0,35-0,35}{0,35} \right| \times 100\% = 0\%$$

Berdasarkan data hasil pengujian sensor arus INA219 pada tabel 4.2, diperoleh rata-rata persentase *error* sebesar 21,2%. Persentase *error* ini disebabkan oleh ketidakstabilan nilai tegangan sumber yang dibaca oleh sensor, serta respons sensor terhadap arus kecil. Meskipun demikian, sensor arus INA219 ini masih dapat digunakan karena meskipun terdapat deviasi dalam pengujian, data yang diperoleh menunjukkan respons *error* yang semakin kecil, bahkan mencapai 0%. Semakin kecil persentase *error*, maka semakin akurat pembacaan sensor tersebut.

4.2.3 Sensor Arus ACS712

Pada proyek akhir ini, sensor arus ACS712 digunakan untuk mengukur arus yang mengalir pada sistem kelistrikan. Dalam pengujian, sensor dipasang secara seri dengan beban, sehingga arus yang mengalir dapat diukur secara langsung. Beban yang digunakan dalam pengujian ini adalah sebuah lampu, yang berfungsi sebagai elemen resistif agar nilai arus yang mengalir dapat dihitung dan dianalisis dengan akurat. Rangkaian uji coba sensor arus ACS712 dapat dilihat pada gambar 4.4 berikut.



Gambar 4.4 Rangkaian Pengujian Sensor ACS712

Data hasil pengujian sensor arus ACS712 dapat dilihat pada tabel 4.3 di bawah ini.

Tabel 4.3 Pengujian Sensor Arus ACS712

<i>Power Supply (V)</i>	<i>Multimeter (A)</i>	<i>Sensor ACS712 (A)</i>	<i>Error (%)</i>
7,5	0,03	0,02	33,3
8,4	0,07	0,09	22,2
8,5	0,10	0,11	9,09
8,6	0,14	0,15	6,7
9	0,20	0,22	9,09
9,2	0,22	0,24	8,3
9,5	0,25	0,25	0
9,7	0,27	0,27	0
10	0,30	0,30	0
11	0,37	0,37	0

Dari data hasil pengujian pada tabel 4.3, terlihat bahwa terdapat perbedaan nilai arus antara multimeter dan hasil pembacaan sensor. Oleh karena itu, perlu dilakukan perhitungan persentase *error* dari sensor arus yang digunakan. Berikut adalah rumus dan perhitungan yang digunakan untuk menghitung persentase *error*.

$$\%Error = \left| \frac{\text{Hasil pembacaan sensor} - \text{Hasil pengukuran multimeter}}{\text{Hasil pembacaan sensor}} \right| \times 100\%$$

Berikut perhitungan persentase *error* dari tabel 4.3

1. Perhitungan persentase *error* pada tegangan 7,5V

$$\%Error = \left| \frac{0,03-0,02}{0,03} \right| \times 100\% = 33,3\%$$

2. Perhitungan persentase *error* pada tegangan 8,4V

$$\%Error = \left| \frac{0,09-0,07}{0,09} \right| \times 100\% = 22,2\%$$

3. Perhitungan persentase *error* pada tegangan 8,5V

$$\%Error = \left| \frac{0,11-0,10}{0,11} \right| \times 100\% = 9,09\%$$

4. Perhitungan persentase *error* pada tegangan 8,6V

$$\%Error = \left| \frac{0,15-0,14}{0,15} \right| \times 100\% = 6,7\%$$

5. Perhitungan persentase *error* pada tegangan 9V

$$\%Error = \left| \frac{0,22-0,20}{0,22} \right| \times 100\% = 9,09\%$$

6. Perhitungan persentase *error* pada tegangan 9,2V

$$\%Error = \left| \frac{0,24-0,22}{0,24} \right| \times 100\% = 8,3\%$$

7. Perhitungan persentase *error* pada tegangan 9,5V

$$\%Error = \left| \frac{0,25-0,25}{0,25} \right| \times 100\% = 0\%$$

8. Perhitungan persentase *error* pada tegangan 9,7V

$$\%Error = \left| \frac{0,27-0,27}{0,27} \right| \times 100\% = 0\%$$

9. Perhitungan persentase *error* pada tegangan 10V

$$\%Error = \left| \frac{0,30-0,30}{0,30} \right| \times 100\% = 0\%$$

10. Perhitungan persentase *error* pada tegangan 11V

$$\%Error = \left| \frac{0,37-0,37}{0,37} \right| \times 100\% = 0\%$$

Berdasarkan data hasil pengujian sensor tegangan pada tabel 4.3, diperoleh rata-rata persentase *error* sebesar 8,86%. Persentase *error* tersebut disebabkan oleh ketidakstabilan nilai tegangan sumber yang dibaca oleh sensor. Meskipun demikian, sensor pembagi tegangan ini masih bisa digunakan karena persentase kesalahannya tetap di bawah 10%.

4.2.4 Sensor Suhu DS18B20

Pada proyek akhir ini, digunakan sensor suhu digital DS18B20 untuk mendeteksi perubahan suhu pada baterai kendaraan. Pengujian sensor dilakukan dengan mengamati suhu ruangan sebagai kondisi awal, kemudian memberikan stimulus panas menggunakan sumber panas eksternal seperti korek api. Tujuan dari pengujian ini adalah untuk menilai kinerja sensor dalam merespons perubahan suhu dengan akurat. Rangkaian uji coba sensor suhu DS18B20 dapat dilihat pada gambar 4.4 di bawah ini.



Gambar 4.5 Rangkaian Pengujian Sensor DS18B20

Data hasil pengujian sensor arus DS18B20 dapat dilihat pada tabel 4.4 berikut.

Tabel 4.4 Pengujian Sensor DS18B20

Waktu Respon (detik)	DS18B20 (C ⁰)	Thermometer (C ⁰)	Selisih (C ⁰)	Kondisi Lingkungan	Ket
0	30.0 °C	30 °C	0.0 °C	Suhu Ruangan	Kondisi awal stabil
5	34.5 °C	33 °C	1.5 °C	Setelah diberi panas (korek api)	Sensor merespons kenaikan suhu
5	35.5 °C	35 °C	0.5 °C	Panas ditingkatkan lagi	Sensor tetap responsif
10	32.5 °C	31 °C	1.5 °C	Setelah panas dimatikan	Suhu mulai menurun kembali

Dari data hasil pengujian pada tabel 4.4, dapat dilihat bahwa terdapat perbedaan antara suhu yang terbaca oleh sensor DS18B20 dan suhu yang ditunjukkan oleh termometer sebagai alat ukur acuan. Perbedaan ini menunjukkan adanya selisih pembacaan antara sensor dan termometer, yang dapat digunakan untuk mengevaluasi tingkat akurasi sensor DS18B20. Oleh karena itu, perlu dilakukan perhitungan selisih suhu (ΔT) antara kedua alat tersebut. Berikut rumus yang digunakan untuk menghitung selisih suhu:

$$\Delta T = T_{DS18B20} - T_{Thermometer}$$

Berikut perhitungan persentase *error* dari tabel 4.3

1. Perhitungan selisih suhu pada waktu 0 detik, suhu ruangan

$$\Delta T = T_{30.0} - T_{30} = 0.0 \text{ }^{\circ}\text{C}$$

2. Perhitungan selisih suhu pada waktu 5 detik, setelah diberi panas (korek api)

$$\Delta T = T_{34.5} - T_{33} = 1.5 \text{ }^{\circ}\text{C}$$

3. Perhitungan selisih suhu pada waktu 5 detik, panas ditingkatkan lagi

$$\Delta T = T_{35.5} - T_{35} = 0.5 \text{ }^{\circ}\text{C}$$

4. Perhitungan selisih suhu pada waktu 10 detik, setelah panas dimatikan

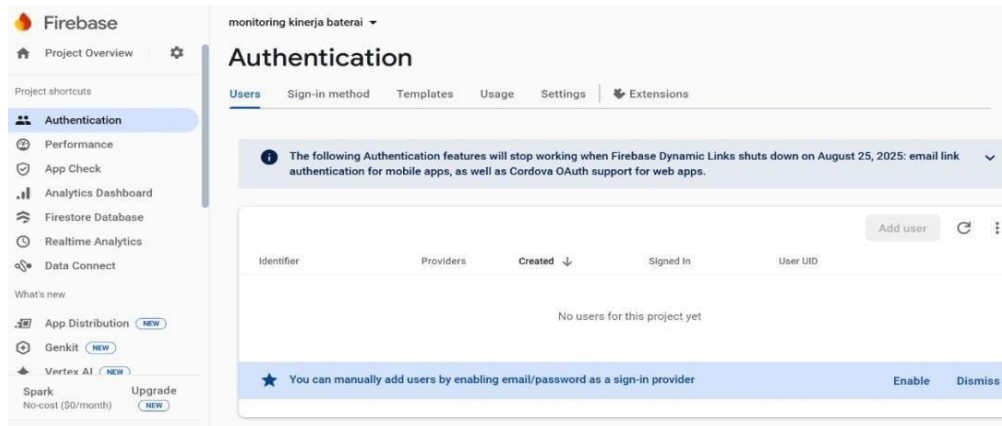
$$\Delta T = T_{32.5} - T_{31} = 1.5 \text{ }^{\circ}\text{C}$$

Berdasarkan data hasil pengujian yang terdapat pada Tabel 4.3, rata-rata selisih suhu antara sensor DS18B20 dan termometer acuan adalah $0,87^{\circ}\text{C}$. Selisih ini disebabkan oleh perbedaan karakteristik alat ukur dan kondisi lingkungan. Meskipun demikian, sensor DS18B20 tetap layak digunakan karena deviasinya masih dalam batas toleransi pengukuran.

4.3 Hasil Tampilan pada *MIT APP Inventor*

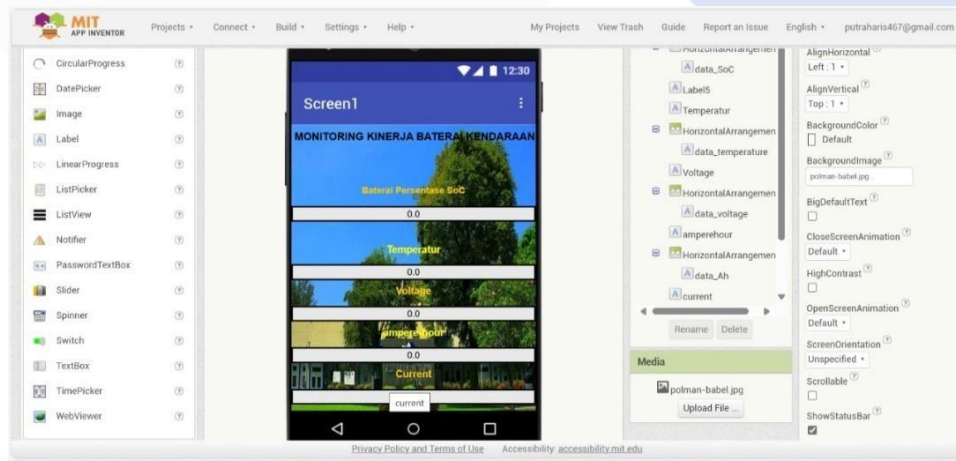
Desain tampilan di *firebase* dan tampilan aplikasi *MIT App Inventor* dapat

dilihat pada gambar 4.6 dan gambar 4.7 di bawah ini.



Gambar 4.6 Tampilan Antarmuka *Firease MIT APP Inventor*

Gambar 4.6 menunjukkan sistem kontrol yang berupa tampilan parameter sensor dan pemantauan di *firebase*, yang mencakup parameter sensor serta sistem pengisian daya baterai.



Gambar 4.7 Tampilan Antarmuka Aplikasi *MIT APP Inventor*

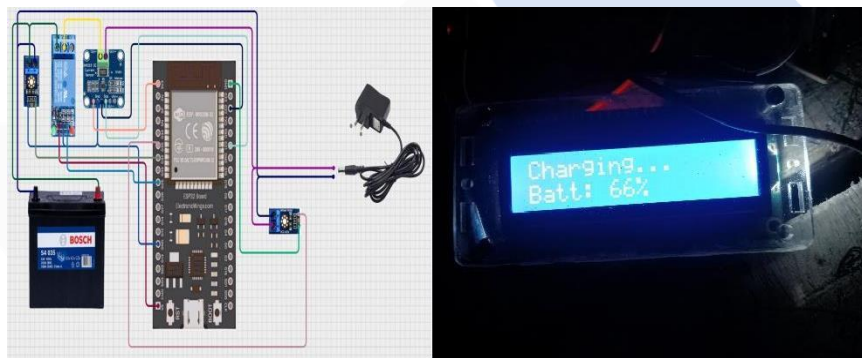
Gambar 4.7 menunjukkan sistem kontrol dan *monitoring* baterai pada aplikasi *MIT App Inventor*, berupa tampilan parameter hasil pengukuran dari sensor tegangan, sensor arus, sensor suhu, serta status pengisian daya (*charging*).

4.4 Hasil Pengujian Sistem *Charging*

Pengujian dilakukan menggunakan aki dalam kondisi hampir kosong. Proses *charging* menggunakan adaptor sebagai penurun dan penyearah tegangan dari AC ke DC, dengan pengaturan waktu oleh RTC untuk memutus arus setiap 3

menit melalui relay. Tujuannya adalah membaca tegangan asli baterai tanpa adanya arus masuk. Sensor INA219 digunakan untuk membaca arus dan tegangan pengisian secara akurat melalui komunikasi I2C, sehingga lebih stabil dan presisi dibandingkan sensor analog biasa.

RTC mengatur agar relay OFF secara berkala, memungkinkan pembacaan tegangan baterai tanpa beban, yang lebih mencerminkan kondisi tegangan asli. Pembacaan data dilakukan secara berkala dan ditampilkan di serial monitor. Rangkaian uji coba sistem *charging* baterai, dapat dilihat pada gambar 4.8 di bawah ini.



Gambar 4.8 Rangkaian Pengujian Sistem *Charging*

Data hasil pengujian sistem *charging* dapat dihitung dengan menggunakan *State of Charge (SoC)* untuk menunjukkan seberapa banyak kapasitas baterai yang telah terisi.

$$SoC_{(t)} = SoC_{(t_0)} - \frac{1}{C_n} \int_{t_0}^t I dt$$

Keterangan :

SoC_{t_0} = SoC awal (sebelum terjadi proses pengisian)

C_n = Kapasitas maksimum baterai (Ah)

$I(t)$ = Arus baterai masuk atau keluar (A)

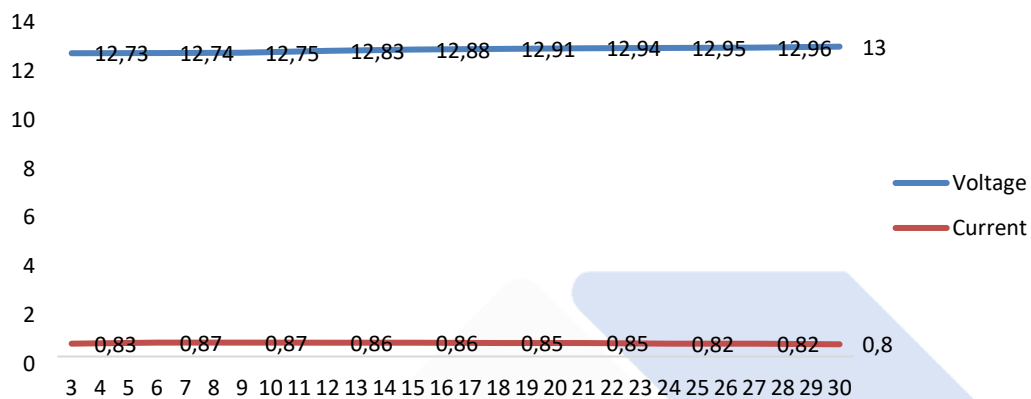
t = Waktu operasi

Berikut adalah grafik tegangan dan arus yang diperoleh dari data

monitoring baterai menggunakan ESP32:

- Garis biru menunjukkan nilai tegangan baterai (Volt).
- Garis merah menunjukkan nilai arus pengisian (mA).

Grafik Hasil Uji Monitoring Charging Baterai

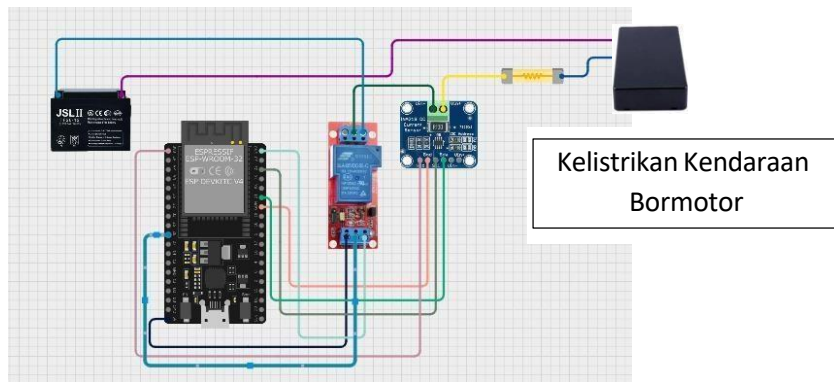


Gambar 4.9 Grafik *Monitoring* Proses *Charging*

Grafik ini menggambarkan dinamika proses pengisian atau pemantauan baterai secara *real-time*. Terlihat bahwa pada awalnya tegangan sekitar 12,73V, lalu meningkat hingga mencapai 12.96 V. Arus juga bervariasi, menandakan fluktuasi beban atau perubahan kondisi pengisian.

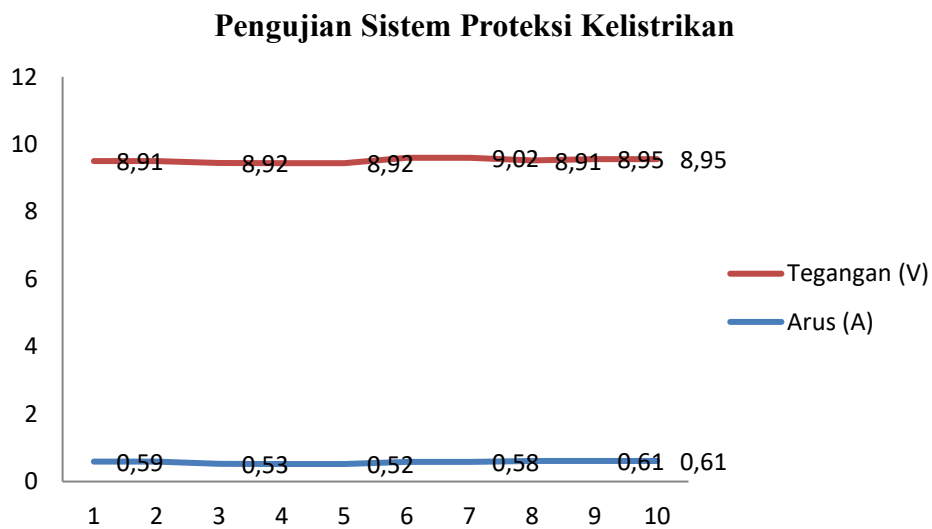
4.5 Hasil Pengujian Sistem Proteksi Kelistrikan

Pada pengujian sistem proteksi kelistrikan dilakukan dengan menggunakan beban kecil berupa lampu dan motor DC bertegangan 9 volt, dengan batas maksimal 30 ampere. Sistem menggunakan sensor INA219 untuk mendeteksi besarnya arus yang mengalir, kemudian *mikrokontroler* mengontrol relay sebagai pemutus arus secara elektronik. Selain itu, fuse dipasang sebagai pengaman tambahan yang berfungsi memutus arus secara langsung jika terjadi arus lebih yang *ekstrem*. Rangkaian pengujian sistem proteksi kelistrikan dapat dilihat pada gambar 4.10 berikut.



Gambar 4.10 Rangkaian Pengujian Sistem Kelistrikan

Pada gambar 4.11 berikut ini disajikan data hasil pengujian sistem kelistrikan dalam bentuk grafik, yang mencakup variabel-variabel penting seperti tegangan, arus, dan waktu pengujian. Grafik ini memberikan gambaran yang jelas mengenai hubungan antara ketiga parameter tersebut selama proses pengujian, serta memudahkan analisis terhadap performa sistem kelistrikan yang diuji.



Gambar 4.11 Grafik Sistem Proteksi Kelistrikan

Grafik hasil uji coba menunjukkan bahwa sistem berhasil mendeteksi kenaikan arus untuk mencegah kerusakan pada komponen. Sementara itu, fuse tetap berfungsi sebagai proteksi cadangan. Proteksi ini bekerja dengan baik, menjaga arus dan tegangan tetap dalam batas aman. Hal ini sangat penting untuk memperpanjang umur aki dan mencegah kerusakan.

4.6 Hasil Pengujian Keseluruhan

Pengujian secara keseluruhan ini adalah pengujian proses pemantauan baterai kendaraan bermotor dengan menggunakan sistem kontrol dan sistem *monitoring*. Terhadap dua jenis data pengujian yang dilakukan, yaitu pengujian saat baterai normal dan pengujian pada saat proses *charging* baterai. Data hasil pengujian saat baterai normal dapat dilihat pada tabel 4.5 berikut.

Tabel 4.5 Data Hasil Pengujian Baterai Normal

<i>Voltage Battery</i>	<i>Current Load</i>	<i>Temperature</i>	<i>Battery (%)</i>
(V)	(mA)	(°C)	
10,5	169	30,0	0%
11,0	210	30,5	0%
11,2	223	30,0	6%
11,4	229	30,0	12%
11,5	233	30,5	16%
11,7	245	30,0	24%
11,8	251	30,0	25%
12	254	30,0	33%
12,24	259	30,0	41%
12,37	261	30,0	45%

Berdasarkan data yang di hasilkan pada tabel 4.5, pengujian dilakukan saat kondisi baterai normal dengan tegangan yang yang dimasukkan terus ditingkatkan maka terjadi perubahan arus beban pada sistem baterai dan juga kapasitas persentase kondisi baterai juga semakin meningkat, dari kondisi tegangan 10,5V hingga 12,37V. Rata-rata arus beban yang masuk yaitu 233,4mA. Pada rangkaian pengujian juga menggunakan lampu sebagai pengujian arus beban pada rangkaian. Tampilan pada LCD dapat dilihat pada Gambar 4.12 di bawah ini.



Gambar 4.12 Tampilan di LCD Kondisi Baterai Normal

Berdasarkan Gambar 4.12, tegangan dan arus yang ditampilkan merupakan tegangan baterai dan arus saat baterai berada dalam kondisi normal. Sementara itu, temperatur dan SoC baterai menunjukkan suhu baterai dalam kondisi normal serta persentase pembacaan baterai.

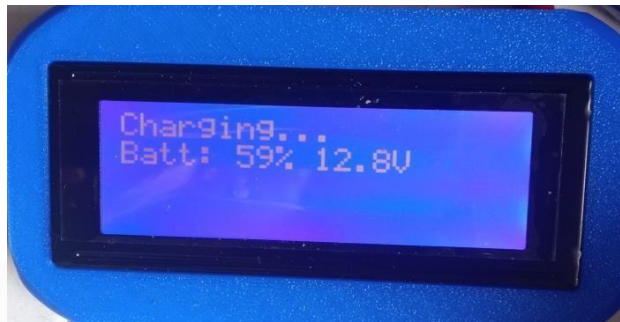
Pengujian baterai pada saat proses *charging* dapat dilihat pada tabel 4.6 di bawah ini.

Tabel 4.6 Data Hasil Pengujian *Charging* Baterai

Waktu (menit)	<i>Voltage</i> <i>Battery</i> (V)	<i>Voltage</i> <i>Charger</i> (V)	<i>Current</i> <i>Charging</i> (mA)	<i>Temperature</i> (°C)	<i>Battery</i> <i>SoC</i> (%)
3	12,73	14,98	83	31,7	52
6	12,74	14,97	87	31,5	54
9	12,75	14,96	87	30,5	55
12	12,83	14,95	86	30,5	56
15	12,88	14,98	86	30,5	58
18	12,91	14,98	85	30,5	59
21	12,94	14,95	85	30,5	60
24	12,95	14,99	82	30,5	60
27	12,96	14,97	82	30,5	61
30	13,00	14,96	80	31,0	62

Berdasarkan data yang terdapat pada tabel 4.6, pengujian proses pengisian (*charging*) baterai dilaksanakan selama 30 menit. Setiap kenaikan persentase SoC

baterai membutuhkan waktu sekitar 3 menit. Selama proses pengisian, tegangan baterai yang masuk mengalami peningkatan sesuai dengan pembacaan sensor tegangan. Proses pengisian menunjukkan kenaikan SoC sebesar 10% dalam durasi 30 menit. Rata-rata tegangan dari *charger* yang mengalir selama pengisian berlangsung adalah sebesar 14,97 V. Tampilan pada LCD dapat dilihat pada gambar 4.13 berikut.



Gambar 4.13 Tampilan di LCD Kondisi Proses *Charging* Baterai

Berdasarkan gambar 4.13, kondisi pengisian baterai sedang berlangsung dengan tegangan terukur sebesar 12,8V. Selama proses ini, kapasitas baterai terus meningkat, dengan peningkatan sebesar 1% setiap interval waktu 3 menit. Hal ini menunjukkan efektivitas proses pengisian dan memberikan gambaran yang jelas mengenai perkembangan kapasitas baterai seiring berjalannya waktu.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah merancang dan membangun *Prototype Monitoring Kinerja Baterai Kendaraan Bermotor* yang Teintegrasi dengan Aplikasi *Mobile*, serta melanjutkan proses pengujian, diperoleh data yang menjadi dasar penarikan kesimpulan sebagai berikut:

1. Sistem *monitoring* baterai kendaraan bermotor berhasil dirancang dan diterapkan menggunakan *mikrokontroler* ESP32 yang terintegrasi dengan berbagai sensor, seperti sensor tegangan, sensor arus INA219, sensor arus ACS712, dan sensor suhu DS18B20. Proses pengisian daya (*charging*) dapat dipantau secara *real-time* dengan akurat melalui aplikasi *mobile* berbasis *MIT App Inventor*, yang menampilkan parameter penting seperti tegangan, arus, suhu, dan status pengisian baterai secara informatif dan interaktif.
2. Hasil pengujian menunjukkan bahwa rata-rata persentase *error* sensor tegangan adalah sebesar 4,78%, sensor arus INA219 sebesar 21,2%, sensor ACS712 sebesar 8,86% serta, sensor suhu DS18B20 menunjukkan respons yang baik terhadap perubahan suhu lingkungan. Semua nilai *error* masih berada dalam batas toleransi yang dapat diterima.
3. Sistem *charging* bekerja sesuai perancangan, dengan kenaikan rata-rata baterai 1% setiap durasi 3 menit, serta sistem proteksi kelistrikan terbukti efektif dalam mendeteksi gangguan seperti arus lebih dan hubungan singkat, serta mampu memutus arus melalui relay dan fuse sebagai lapisan pengaman tambahan. Hal ini berfungsi menjaga kesehatan baterai dan keselamatan sistem secara keseluruhan.

5.2 Saran

Proyek ini masih memiliki beberapa kelemahan dan dapat dikembangkan lebih lanjut. Oleh karena itu, penulis memberikan beberapa saran sebagai berikut:

1. Dalam rangka untuk meningkatkan akurasi pembacaan sensor, disarankan melakukan kalibrasi rutin terhadap sensor arus dan tegangan, khususnya saat digunakan dalam jangka panjang atau di lingkungan *ekstrem*.
2. Penggunaan baterai dan adaptor yang berkualitas tinggi perlu dipertimbangkan agar hasil pengukuran dan kinerja sistem lebih stabil dan tahan lama.
3. Sistem proteksi dapat ditingkatkan dengan penambahan proteksi berbasis *software* seperti pengendalian PWM untuk pengaturan arus pengisian secara lebih presisi.
4. Tampilan antarmuka pada *MIT App Inventor* dapat dibuat lebih *user-friendly*, misalnya dengan menambahkan grafik historis tegangan dan arus, serta informasi prediksi waktu penuh baterai.

DAFTAR PUSTAKA

- Aditya, R., Pranatawijaya, V., & Putra, P. (2021), "Rancang Bangun Aplikasi Monitoring Kegiatan Menggunakan Metode Prototype", *Journal of Information Technology and Computer Science*, vol.1, no 1, pp. 48-57.
- Agustian, L. (2015), "Rancang Bangun Sistem Monitoring Kondisi Aki Pada Kendaraan Bermotor", *Tesis*, Universitas Tanjungpura Pontianak.
- Alviero, A., & Nugroho, D. (2023), "Pengaplikasian Sensor Arus ACS712 Sebagai Sistem Proteksi Pada Alat Penghitung Kertas Otomatis Berbasis Io", *Metrotech*, vol. 2 , no. 1, pp. 1-7.
- Aritonang, W., Bangsa, I., & Rahmadewi, R. (2021), "Implementasi Sensor Suhu DS18B20 dan Sensor Tekanan MPX5700AP menggunakan Mikrokontroler Arduino Pada Alat Pendeteksi Tingkat Stress", *Jurnal Ilmiah Wahana Pendidikan*, vol. 7, no. 1, pp. 153-160.
- Bawotong, V., Mamahit, D., & Sompie, S. (2015), "Rancang Bangun Uninterruptible Power Supply Menggunakan Tampilan LCD Berbasis Mikrokontroler", *E-journal Teknik Elektro dan Komputer*, pp. 1-7.
- Dhamayanthie, I., Solikha, D., & Nurjanah , A. (2023), "Studi Pengolaan Limbah Aki Kering Dan Aki Basah (Studi Kasus DI Indramayu)", *Jurnal Migasian*, vol. 7, no.1, pp. 28-41.
- Edriati, S., Husnita, L., Amri, E., Samudra, A., & Kamil, N. (2021), "Penggunaan Mit App Inventor untuk Merancang Aplikasi Pembelajaran Berbasis Android", *Jurnal Pengabdian kepada Masyarakat*, vol.12, no. 4, pp. 652-657.
- Fahruri, H., Aribowo, W., Widyartono, M., & Hermawan, A. (2021), "Monitoring Arus, Tegangan, Dan Suhu Pada Prototype Thermoelectric Generator Berbasis IoT", *Tesis*, Universitas Negeri Surabaya.
- Febriansa, I., & Sandika, L. (2023), " Stasiun Pengisian Baterai Berbasis IoT" *Proyek Akhir*, Politeknik Manufaktur Negeri Bangka Belitung, Sungailiat.
- Harlim, K., & Setiyawati, N. (2022), "Perancangan User Experience Aplikasi Mobile Majuli Menggunakan Metode Design Thinking", *Journal of Information Technology Ampera*, vol. 3, no. 2, pp. 108-123.
- Hilmansyah, Utomo, R., Saputra, A., & Alif , R. (2020), "Rancang Bangun Wireless battery Monitoring System Berbasis ESP32", *SNITT- Politeknik Negeri*

Balikpapan, pp. 194-199.

- Iqbar, M., & Kartika, K. (2020), "Rancang Bangun Lampu Portabel Otomatis Menggunakan RTC Berbasis Arduino", *Jurnal Ilmiah Teknik Informatika*, vol. 14. No.1, pp. 61-72.
- Karimah, C., Zain, A., & Nofiansyah, A. (2023), "Analisa Baterai Sebagai Sumber Kelistrikan Kendaraan Roda Dua Ditinjau Dari Kapasitas dan Efisiensi", *Jurnal Teknik Terapan*, vol. 2, no.1, pp. 01-11.
- Kurniawan, I., & Amarudin, R. (2025), "Pengendalian Pengisian Baterai Menggunakan Boost Converter Berbasis Logika Fuzzy", *Jurnal Riset Sains dan Teknologi*, vol. 9, no.1, pp. 73-80.
- Nizam, M., Yuana, H., & Wulansari, Z. (2022), "Mikrokontroler ESP 32 Sebagai Alat Monitoring Pintu Berbasis Web", *Jurnal Mahasiswa Teknik Informatika*, vol. 6, no.2, pp. 767-772.
- Otong, M., & Albantani, M. (2024), "Analisis Sistem Proteksi Motor Induksi 6KV Pada Circulating Water Pump (CWP) Menggunakan Relay Overcurrent dan Short Circuit Unit 1-4 PT. PLN Indonesia Power UBP Suralaya", *Jurnal Ilmiah Setrum*, vol.13, no.2, pp.150-156.
- Prabowo, Y., Broto, S., Wisnuadji, T., & Siswanto. (2022), "Analisa Power Mode ESP32 Untuk Catu Daya Pada Sistem Berbasis IoT", *Prosiding Seminar Nasional Sistem Informasi dan Teknologi (SISFOTEK)*, pp.150-154.
- Pratomo, D., & Fahreza, R. (2024), "Prototype Pembangkit Listrik Tenaga Pikohidro Menggunakan Turbin Archimedes Screw Bilah Lima Berbasis IoT", *Proyek Akhir*, Politeknik Manufaktur Negeri Bangka Belitung, Sungailiat.
- Radina, M., Arinto S, F., & Sumadi. (2022). "Sistem Kontrol Beban dan Monitoring Daya Baterai Pada Panel Surya 50WP untuk Aplikasi Penerangan Berbasis Internet of Things". *JITET (Jurnal Informatika dan Teknik Elektro Terapan*, vol. 10, no.3, pp. 167-172.
- Ratnasari, T., & Senen, A. (2017), "Perancangan Prototype Alat Ukur Arus Listrik AC Dan DC Berbasis Mikrokontroler Arduino Dengan Sensor Arus ACS-712 30 Ampere", *Jurnal Sutet*, vol. 7, no.2, pp. 28-33.
- Rifansyah , F. (2023), "Analisis State Of Charge (SoC) Menggunakan Algoritma Coulomb Counting dan State of Health (SoH) Untuk Mendukung Kendaraan Listrik", *Jurnal Mahasiswa Institut Teknologi Telkom Surabaya Fakultas Teknologi Elektro Dan Industri Cerdas*, pp.1-10.
- Saleh, M., & Haryati, M. (2017), "Rancang Bangun Sistem Keamanan Rumah

- Menggunakan Relay”, *Jurnal Teknologi Elektro*, vol.8, no.2, pp. 87-94.
- Saputro, A., & Prasetya, D. (2022), “Rancang Bangun Thermopen Sebagai Pengukur Suhu Menggunakan Sensor DS18B20 Dalam Internet of Things”, *Jurnal Teknik Elektro*, vol. 4. no. 1, pp.32-40.
- Sarfini, A., & Irawan, D. (2024), "Sistem Kontrol Jarak Jauh PLC Menggunakan ESP32 Berbasis IoT", *Jurnal Amplifier : Jurnal Ilmiah Bidang Teknik Elektro dan Komputer*, vol. 14, no. 1, pp.51-55.
- Setiono, I. (2015), “Akumulator, Pemakaian dan Perawatannya”, *Metana*, vol. 11 no.1, pp.31-36.
- Siregar, M., & Permana, I. (2016), “Rancang Bangun Aplikasi Berbasis Mobile Untuk Navigasi Ke Alaman Pelanggan TV Berbayar”, *Jurnal Rekayasa dan Manajemen Sistem Informasi*, vol. 2, no. 1, pp. 82-94.
- Sucipto, Hutabarat, M., & Pangaribuan, M. (2023), “Perancangan Sistem Kendali Saklar on off Kelistrikan Mesin Sepeda Motor Jarak Jauh Melalui SMS”, *Jurnal Teknologi Informasi dan Industri*, vol. 4, no.1, pp. 11-20.
- Sugi, O., & Sulaiman. (2024), “Sistem Pengisian Dan Monitoring Daya Aki Berbasis Smartphone” *Journal Of Social Science Research*, vol. 4, no. 5, pp. 9860-9871.
- Tamara, N., & Aji, W. (2021), “Monitoring Tegangan Aki Kendaraan Berbasis Smartphone Andorid”, *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 3, no.3, pp. 202- 209.
- Turang, D. (2015), “Pengembangan Sistem Relay Pengendalian dan Penghematan Lampu Berbasi Mobile”, *Seminar nasional Informatika (semnasIF) 2015*, pp. 75-85.
- Utomo, M., Nugrahanto, I., & Sungkono. (2023), “Sistem Penyimpanan Energi Menggunakan Baterai Sel Sekunder Pada Photovoltaic”, *Jurnal Elkolind*, vol. 10, no.1, pp. 85-93.
- Wagyana, A., & Rahmat. (2019), “Prototipe Modul Praktik untuk Pengembangan Aplikasi Internet of Things (IoT)”, *Jurnal Ilmiah Setrum*, vol. 8, no. 1, pp. 238-247.

LAMPIRAN

Lampiran 1.1

DAFTAR RIWAYAT HIDUP

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama lengkap : Haris Adi Putra
Tempat & tanggal lahir : Kijang, 30 Januari 2004
Alamat rumah : KP. Budi Mulya, Kel.
Kijang Kota, Kec. Bintan
Timur, Kab. Bintan
Kepulauan Riau
Email : putraharis467@gmail.com
Jenis kelamin : Laki-laki
Agama : Islam



2. Riwayat Pendidikan

SD Negeri 010 Bintan Timur	Lulus 2016
SMP Negeri 1 Bintan	Lulus 2019
SMK Negeri 3 Tanjungpinang	Lulus 2022
Politeknik Manufaktur Negeri Bangka Belitung	2022- Sekarang

3. Pendidikan Non-Formal

Praktik Kerja Lapangan di PT Yokogawa Manufacturing
Batam Praktik Kerja Lapangan di PT PLN ULP Kijang

Sungailiat, 27 Juni 2025

Haris Adi Putra

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama lengkap : Putri Yuliana
Tempat & tanggal lahir : Prabumulih, 13 Juli 2004
Alamat rumah : Kel. Tanjung Rambang,
Kec. Rambang Kapak Tengah,
Kota Prabumulih, Sumatera Selatan
Email : putriyuliana130704@gmail.com
Jenis kelamin : Perempuan
Agama : Islam



2. Riwayat Pendidikan

SD Negeri 01 Rambang Kuang	Lulus 2016
SMP Negeri 01 Rambang Kuang	Lulus 2019
SMA Negeri 4 Prabumulih	Lulus 2022
Politeknik Manufaktur Negeri Bangka Belitung	2022- Sekarang

3. Pendidikan Non-Formal

Praktik Kerja Lapangan di PT Pratama Motivasi Mandiri (PMM)

Sungailiat, 27 Juni 2025

Putri Yuliana

LAMPIRAN 2

Lampiran 2.1

PROGRAM KESELURUHAN

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <RTCLib.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <Adafruit_INA219.h>
#include <WiFi.h>
#include <HTTPClient.h>

#define SDA_PIN 21
#define SCL_PIN 22
#define RELAY_CHARGING1 25
#define RELAY_PROTECTION2 26
#define ONE_WIRE_BUS 4
#define VOLTAGE_DIVIDER_BATTERY1 35
#define VOLTAGE_DIVIDER_CHARGER2 32

#define R1 30000.0
#define R2 7500.0
#define MAX_ADC 4095.0
#define REF_VOLTAGE 3.3
#define CALIBRATION 1.030

const float maxCurrentLimit = 30.0;

LiquidCrystal_I2C lcd(0x27, 20, 4);
RTC_DS3231 rtc;
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
Adafruit_INA219 ina219Charging(0x40);

float temperature = 0.0;
float batteryVoltageReal = 0.0;
int batteryPercentage = 0;
bool chargingDisplayed = false;
bool protectionActive = false;

const float chargerThresholdOn = 13.0;
const float chargerThresholdOff = 12.9;

bool chargerBelowThreshold = true;
bool chargingActive = false;
bool relayChargingOffForCheck = false;

unsigned long lastSwitchTime = 0;
unsigned long lastCheckTime = 0;
```

```

unsigned long relayOffStartTime = 0;

const unsigned long checkInterval = 180000;
const unsigned long relayOffDuration = 10000;

int displayIndex = 0;

// === WiFi dan Firebase ===
const char* ssid = "Oppo";
const char* password = "Hari1234";
const char* firebaseHost = "monitoring-kinerja-baterai-default-
rtadb.firebaseio.com";
const char* firebaseAuth =
"6JY9VInIsVY0jw8GPnK4upc7CfAaNZXaQ2jUCify";

// === Telegram ===
const char* telegramToken =
"8149167318:AAFMVfqP6zlolA0qu1XdOJmATFWS2yYPbsc";
const char* telegramChatID = "6711085874";
bool telegramNotified = false;
bool lowBatteryNotified = false;

byte fullBlock[8] = {
    0b111111, 0b111111, 0b111111,
    0b111111, 0b111111, 0b111111,
    0b111111, 0b111111
};

// === ACS712 Configuration ===
#define ACS_PIN 34

float bacaArusACS712(int pin) {
    const int jumlahData = 100;
    const int adcZero = 2928; // Nilai offset ADC saat arus = 0A (kalibrasi manual)
    const float slope = 0.21; // Sensitivitas ACS712-30A dalam Volt per Ampere
    (66mV/A)
    long total = 0;

    for (int i = 0; i < jumlahData; i++) {
        total += analogRead(pin);
        delayMicroseconds(100);
    }

    float rata2 = total / (float)jumlahData;

```

```

// Konversi ADC ke tegangan (sensor diberi 5V)
float voltage = rata2 * (4.94 / 4095.0);
float voltageZero = adcZero * (4.94 / 4095.0);

// Hitung arus berdasarkan offset 0A = voltageZero dan slope (66 mV/A)
float arus = (voltage - voltageZero) / slope;

// Arus negatif dianggap 0
if (arus < 0.1) {
    arus = 0.0;
}

return arus;
}

float bacaTegangan(int pin) {
    float totalADC = 0;
    for (int i = 0; i < 50; i++) {
        totalADC += analogRead(pin);
        delay(1);
    }
    float rataADC = totalADC / 50.0;
    float teganganADC = (rataADC / MAX_ADC) * REF_VOLTAGE;
    return teganganADC * ((R1 + R2) / R2) * CALIBRATION;
}

void kalibrasiOtomatis(float voltageActual, float voltageMeasured) {
    float error = voltageActual - voltageMeasured;
    if (abs(error) > 0.1) {
        Serial.print("Kalibrasi Baru: ");
        Serial.println(voltageActual / voltageMeasured, 4);
    }
}

void printBar(float percentage, int row, int startCol) {
    const int barWidth = 14;
    int filled = round(percentage / 100.0 * barWidth);
    lcd.setCursor(startCol, row);
    lcd.write('[');
    for (int i = 0; i < barWidth; i++) {
        lcd.write(i < filled ? byte(0) : ' ');
    }
    lcd.write(']');
}

void kirimKeFirebase(float voltage, int batteryPercent, float currentA, float

```

```

temperatureC, float voltageReal) {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        String url = "https://" + String(firebaseHost) + "/battery_monitor.json?auth=" +
firebaseAuth;
        http.begin(url);
        http.addHeader("Content-Type", "application/json");

        String jsonData = "{}";
        jsonData += "\"voltage\": " + String(voltage, 2) + ",";
        jsonData += "\"battery\": " + String(batteryPercent) + ",";
        jsonData += "\"current\": " + String(currentA, 2) + ",";
        jsonData += "\"temperature\": " + String(temperatureC, 1) + ",";
        jsonData += "\"voltage_real\": " + String(voltageReal, 2);
        jsonData += "}";

        int httpCode = http.PUT(jsonData);
        if (httpCode > 0) {
            Serial.println("✅ Data terkirim ke Firebase!");
        } else {
            Serial.print("❌ Gagal kirim data: ");
            Serial.println(httpCode);
        }
        http.end();
    } else {
        Serial.println("⚠️ WiFi tidak terhubung.");
    }
}

void kirimPesanTelegram(String pesan) {
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;
        String url = "https://api.telegram.org/bot" + String(telegramToken) +
"/sendMessage?chat_id=" + String(telegramChatID) +
"&text=" + urlencode(pesan);
        http.begin(url);
        int httpCode = http.GET();
        if (httpCode > 0) {
            Serial.println("📩 Notifikasi Telegram terkirim!");
        } else {
            Serial.print("❌ Gagal kirim Telegram: ");
            Serial.println(httpCode);
        }
        http.end();
    }
}

```

```

}

String urlencode(String str) {
    String encoded = "";
    char c;
    char code0, code1;
    for (int i = 0; i < str.length(); i++) {
        c = str.charAt(i);
        if (isalnum(c)) {
            encoded += c;
        } else {
            code0 = (c >> 4) & 0xF;
            code1 = c & 0xF;
            encoded += '%';
            encoded += "0123456789ABCDEF"[code0];
            encoded += "0123456789ABCDEF"[code1];
        }
    }
    return encoded;
}

void setup() {
    Serial.begin(115200);
    Wire.begin(SDA_PIN, SCL_PIN);
    lcd.init();
    lcd.backlight();
    lcd.createChar(0, fullBlock);
    sensors.begin();
    ina219Charging.begin();
    ina219Charging.setCalibration_16V_400mA();

    pinMode(RELAY_CHARGING1, OUTPUT);
    pinMode(RELAY_PROTECTION2, OUTPUT);
    digitalWrite(RELAY_CHARGING1, LOW);
    digitalWrite(RELAY_PROTECTION2, HIGH);

    pinMode(ACS_PIN, INPUT);

    lcd.setCursor(2, 1);
    lcd.print("System Starting...");
    delay(2000);
    lcd.clear();

    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
    }
}

```

```

    Serial.print(".");
}
Serial.println("\nWiFi Connected.");
}

void loop() {
    float voltageBattery = bacaTegangan(VOLTAGE_DIVIDER_BATTERY1);
    float voltageCharger = bacaTegangan(VOLTAGE_DIVIDER_CHARGER2);
    float currentCharging = ina219Charging.getCurrent_mA() / 1000.0;
    float currentLoad = bacaArusACS712(ACS_PIN);

    sensors.requestTemperatures();
    float temperatureRaw = sensors.getTempCByIndex(0);
    if (temperatureRaw != -127.0) {
        temperature = temperatureRaw;
    }

    if (currentLoad > maxCurrentLimit) {
        if (!protectionActive) {
            Serial.println(" ⚠️ Arus beban tinggi! Proteksi aktif.");
            digitalWrite(RELAY_PROTECTION2, LOW);
            lcd.clear();
            lcd.setCursor(0, 0); lcd.print(" ⚠️ Overcurrent Load!");
            lcd.setCursor(0, 1); lcd.print("Relay Proteksi OFF");
            protectionActive = true;
        }
        delay(3000);
        return;
    } else if (protectionActive) {
        digitalWrite(RELAY_PROTECTION2, HIGH);
        protectionActive = false;
        lcd.clear();
    }

    if (voltageCharger > chargerThresholdOn) {
        chargerBelowThreshold = false;
    } else if (voltageCharger < chargerThresholdOff) {
        chargerBelowThreshold = true;
    }
    chargingActive = !chargerBelowThreshold;

    if (chargingActive) {
        digitalWrite(RELAY_PROTECTION2, LOW); // 🔴 Proteksi off saat
        charging
        if (!relayChargingOffForCheck) {
            digitalWrite(RELAY_CHARGING1, HIGH);

```

```

    if (millis() - lastCheckTime >= checkInterval) {
        Serial.println("⚡ Putus relay untuk cek tegangan asli...");
        digitalWrite(RELAY_CHARGING1, LOW);
        relayChargingOffForCheck = true;
        relayOffStartTime = millis();
        lastCheckTime = millis(); // ✅ Set di sini agar tidak terulang sebelum
waktunya
    }
    } else {
        if (millis() - relayOffStartTime >= relayOffDuration) {
            batteryVoltageReal = bacaTegangan(VOLTAGE_DIVIDER_BATTERY1);
            kalibrasiOtomatis(12.5, batteryVoltageReal);
            digitalWrite(RELAY_CHARGING1, HIGH);
            relayChargingOffForCheck = false;
            Serial.println("✅ Selesai cek tegangan asli baterai.");
        }
    }

    if (!chargingDisplayed) {
        lcd.clear();
        chargingDisplayed = true;
        if (!telegramNotified) {
            kirimPesanTelegram("🔌 Charging dimulai. Baterai saat ini: " +
String(batteryPercentage) + "% (" + String(batteryVoltageReal, 2) + "V");
            telegramNotified = true;
        }
    }

    batteryPercentage = map(batteryVoltageReal * 100, 1100, 1400, 0, 100);
    batteryPercentage = constrain(batteryPercentage, 0, 100);
    lcd.setCursor(0, 0); lcd.print("Status: Charging ");
    lcd.setCursor(0, 1); lcd.print("Batt: ");
    lcd.print(batteryPercentage); lcd.print("% ");
    lcd.setCursor(0, 2); lcd.print("Voltage: ");
    lcd.print(batteryVoltageReal, 2); lcd.print("V ");
    printBar(batteryPercentage, 3, 0);
} else {
    digitalWrite(RELAY_CHARGING1, LOW);
    digitalWrite(RELAY_PROTECTION2, HIGH); // 🟢 Proteksi aktif kembali
    relayChargingOffForCheck = false;
    chargingDisplayed = false;
    telegramNotified = false;
    batteryVoltageReal = voltageBattery;
    batteryPercentage = map(batteryVoltageReal * 100, 1100, 1400, 0, 100);
    batteryPercentage = constrain(batteryPercentage, 0, 100);

```

```

if (millis() - lastSwitchTime > 3000) {
    displayIndex = (displayIndex + 1) % 2;
    lastSwitchTime = millis();
    lcd.clear();
}
if (displayIndex == 0) {
    lcd.setCursor(0, 0); lcd.print("Voltage : ");
    lcd.print(voltageBattery, 2); lcd.print("V ");
    lcd.setCursor(0, 1); lcd.print("Current : ");
    lcd.print(currentLoad, 2); lcd.print("A ");
    lcd.setCursor(0, 2); lcd.print("Temp  : ");
    lcd.print(temperature, 1); lcd.print((char)223); lcd.print("C ");
    lcd.setCursor(0, 3);
    lcd.print("Status : ");
    if (batteryVoltageReal >= 12.0) {
        lcd.print("Good ");
    } else if (batteryVoltageReal >= 10.8) {
        lcd.print("Weak ");
    } else {
        lcd.print("Dead ");
    }
} else {
    lcd.setCursor(0, 0); lcd.print("Battery SoC:");
    lcd.setCursor(0, 1); lcd.print(batteryPercentage); lcd.print("% ");
    printBar(batteryPercentage, 2, 0);
}
}
// 📡 Notifikasi Telegram jika baterai low (<10V)
if (batteryVoltageReal < 10.0 && !lowBatteryNotified) {
    String pesanLow = " ⚠️ Baterai LOW!\n"
        "SoC: " + String(batteryPercentage) + "%\n"
        "Tegangan: " + String(batteryVoltageReal, 2) + "V\n"
        "Segera lakukan pengisian daya!";
    kirimPesanTelegram(pesanLow);
    lowBatteryNotified = true;
} else if (batteryVoltageReal > 10.5 && lowBatteryNotified) {
    // Reset status jika tegangan sudah normal
    lowBatteryNotified = false;
}

kirimKeFirebase(voltageBattery, batteryPercentage, currentLoad, temperature,
batteryVoltageReal);
delay(1000);
}

```