

APLIKASI MONITORING SMART CHARGER ANDROID BERBASIS MIKROKONTROLER ESP8266 MENGGUNAKAN FLUTTER

MONITORING APPLICATION OF SMART CHARGER ANDROID BASED ON MICROCONTROLLER ESP8266 USING FLUTTER

Asvian Sulaeman¹⁾, Angelina Pramana Thenata²⁾, Bhustomy Hakim³⁾

^{1,2,3)}Fakultas Teknologi dan Desain, Program Studi Informatika, Universitas Bunda Mulia
Jl. Jalur Sutra Barat No Kav 7-9, Kota Tangerang, Banten, 15143

Diterima 06 Januari 2023 / Disetujui 12 Februari 2023

ABSTRACT

Smartphones are devices that are commonly used by people in this modern era. Based on the operating system, smartphones are divided into two types, namely IOS and Android. To perform its function, the smartphone requires power that it draws from the battery. Batteries used in smartphones require recharging using a charger. While recharging can take a long time so that charging often continues even though the battery is fully charged which often results in overcharge. Based on the identification of these problems, this study aims to design a smart charger using an ESP8266 microcontroller and a monitoring application that can read and display battery data from an Android Smartphone and send a signal to the smart charger to connect/disconnect the current that is connected when charging the battery based on the battery data provided. read by the researcher's Android application. This study uses the Flutter framework as the Android dan Web User Interface, because Flutter has started to support multi-platform programming. This research was conducted to test the tool using comparative experiments with ordinary chargers with smart chargers made by researchers and Blackbox on web dan android applications. The results of this study are a web dan android application that can read dan display battery data, a smart charger that can connect to the internet and connect / disconnect the charging current based on the data sent by the application using the MQTT Protocol. As for the results of testing this application using comparative experiments that prove the success of the smart charger to stop the charging current when the battery is fully charged, and Blackbox against Web dan Android applications that have been declared successful. Based on the results of this study, it can be concluded that the smart charger is designed to be used to prevent overcharging by disconnecting the battery charging current when the battery is fully charged, and displaying the battery information using the Android dan Web application. The smart charger can extend battery life on Android Smartphones.

Keywords: Android, ESP8266, Flutter, Monitoring, Smart Charger

ABSTRAK

Smartphone merupakan perangkat yang sudah umum digunakan oleh masyarakat di zaman modern ini. Berdasarkan sistem operasinya smartphone terbagi menjadi dua jenis, yaitu IOS dan Android. Untuk menjalankan fungsinya, smartphone membutuhkan daya yang diambilnya dari baterai. Baterai yang digunakan didalam smartphone memerlukan pengisian ulang menggunakan charger. Sedangkan pengisian ulang tersebut bisa memakan waktu yang cukup lama sehingga pengisian daya sering kali tetap berlanjut walaupun baterai sudah terisi penuh yang sering mengakibatkan overcharge. Berdasarkan identifikasi masalah tersebut, penelitian ini bertujuan untuk merancang smart charger menggunakan mikrokontroler ESP8266 dan aplikasi monitoring yang dapat membaca dan menampilkan data baterai dari Android Smartphone dan mengirimkan sinyal kepada smart charger untuk menyambungkan / memutuskan arus yang terhubung saat melakukan pengisian daya baterai berdasarkan data baterai yang dibaca oleh aplikasi Android yang dibuat peneliti. Penelitian ini menggunakan framework Flutter sebagai User Interface Android dan Web, karena Flutter sudah mulai mendukung pemrograman multi-platform. Penelitian ini dilakukan pengujian terhadap alat menggunakan eksperimen perbandingan dengan charger biasa dengan smart charger yang dibuat peneliti dan blackbox pada aplikasi web

dan android. Hasil penelitian ini adalah aplikasi web dan android yang dapat membaca dan menampilkan data baterai, smart charger yang dapat terhubung dengan internet dan menyambungkan / memutus arus charging berdasarkan data yang dikirim oleh aplikasi menggunakan MQTT Protocol. Adapun juga hasil pengujian aplikasi ini menggunakan eksperimen perbandingan yang membuktikan keberhasilan smart charger untuk menghentikan arus pengisian daya saat baterai sudah terisi penuh, dan blackbox terhadap aplikasi Web dan Android yang sudah dinyatakan berhasil. Berdasarkan hasil penelitian tersebut, dapat disimpulkan bahwa smart charger dirancang dapat digunakan untuk mencegah overcharging dengan cara memutuskan arus pengisian daya baterai saat baterai sudah terisi penuh, serta menampilkan informasi baterai tersebut menggunakan aplikasi Android dan Web. Adapun smart charger dapat memperpanjang umur baterai pada Android Smartphone.

Kata Kunci: Android, ESP8266, Flutter, Monitoring, Smart Charger

*Korespondensi Penulis:

E-mail: angelina.pramana31@gmail.com

PENDAHULUAN

A. Latar Belakang

Berkembangnya ilmu pengetahuan dan teknologi informasi telah mengubah perspektif dan aktivitas masyarakat Indonesia (Budiman, 2017). Dewasa ini teknologi telah berkembang. Sebagian besar peralatan menggunakan listrik sebagai sumber dayanya (Putra & Mukhaiyar, 2020). Listrik adalah energi alam yang digunakan masyarakat. Meningkatnya jenis kebutuhan masyarakat mengakibatkan kebutuhan listrik menjadi kebutuhan utama (Dwijantari, 2017). Sekarang listrik dapat disimpan didalam baterai. Baterai merupakan alat untuk menyimpan energi listrik dalam bentuk kimia yang selanjutnya akan diubah kembali untuk memperoleh arus listrik yang dibutuhkan (Dwijantari, 2017).

Smartphone menjadi ikon dari 'gaya hidup' masyarakat (Dwijantari, 2017). Dalam implementasinya smartphone menggunakan sistem operasi, salah satunya adalah android. Android merupakan sistem operasi yang berjalan di mobile yang mencakup sistem operasi, middleware dan aplikasi (Kuswanto & Radiansah, 2018). Banyak masyarakat yang memiliki kebiasaan meninggalkan smartphone-nya terisi, walaupun baterai dari smartphone tersebut sudah penuh. Hal tersebut dapat menyebabkan baterai

bocor/kehilangan kapasitas aslinya. Beberapa solusi sudah ditemukan untuk mengatasi masalah tersebut, seperti rangkaian circuit cut-off dapat menghentikan keluaran voltase dan ampere sesuai nilai yang ditentukan (Prihananto et al., 2019). Selain rangkaian circuit cut-off, ada juga PMIC (Power Management Integrated Circuit) adalah sirkuit terintegrasi yang digunakan untuk konversi tegangan, regulasi tegangan, dan manajemen baterai. PMIC menangani pengurutan energi, suplai daya ke banyak beban, dan mencegah kondisi voltase terlalu besar, voltase terlalu kecil, arus lebih, dan gangguan panas/thermal (Schnauffer et al., 2020).

Dari beberapa solusi tersebut masih ada beberapa kelemahan seperti PMIC yang akan memicu efek trickle saat melakukan isi ulang baterai. Trickle charging merupakan charging dengan implementasi arus konstan. Besarnya arus yang digunakan merupakan hasil dari rata rata arus yang diperlukan sampai baterai terisi penuh (full charge) dan arus kompensasi untuk melayani beban. Umumnya trickle charging digunakan pada baterai yang tidak sering dikosongkan (discharge) (Purba, 2020). Metode ini juga dikenal dengan sebutan CCTC (Constant Current Trickle Charge), CCTC memiliki kekurangannya sendiri, yaitu saat pengisian daya baterai dilakukan

terlalu lama walaupun baterai sudah terisi penuh yang dikenal juga sebagai overnight charges. Metode CCTC ini secara tidak langsung memperpendek umur baterai smartphone (Fiantis, 1967).

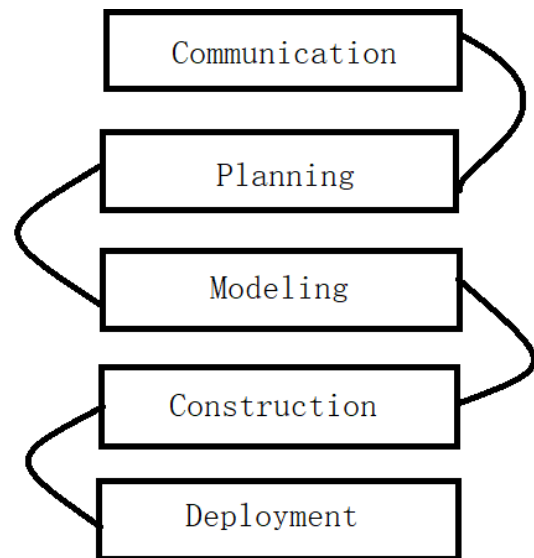
B. Rumusan Masalah

Berdasarkan pada latar belakang di atas, identifikasi masalah dalam penelitian ini sebagai berikut :

- a) Bagaimana merancang smart charger menggunakan ESP8266?
- b) Bagaimana cara membuat aplikasi Android menggunakan Flutter untuk mendeteksi status baterai dan mengirimkan sinyal ke smart charger untuk memutus arus listrik sehingga mencegah overcharge?
- c) Bagaimana cara membuat aplikasi web menggunakan Flutter untuk dapat melakukan monitoring dari browser.
- d) Bagaimana cara mengirimkan data dari Android dan Smart Charger ke aplikasi Monitoring setiap detiknya?

METODE PENELITIAN

Penelitian ini mengumpulkan berbagai informasi dari internet dan buku untuk mendapatkan teori yang berkaitan dengan topik penelitian. Metode pengembangan sistem yang digunakan pada penelitian ini adalah metode waterfall (waterfall model) seperti gambar 1 yang terdiri dari tahapan (1) Communication (Project Initialization and Requirements Gathering), (2) Planning (Estimating, Scheduling, and Tracking), (3) Modeling (Analysis and Design), (4) Construction (Code and Test), (5) Deployment (Delivery, Support, and Feedback).



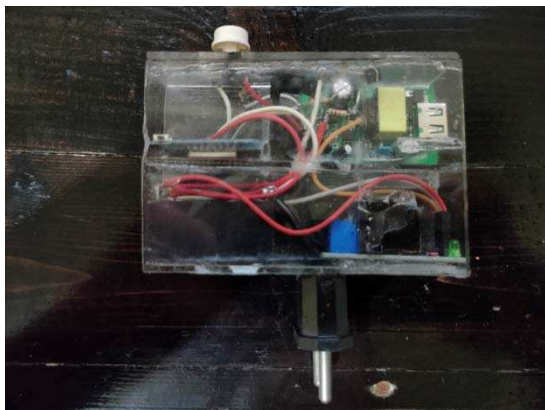
Gambar 1 Waterfall Model

Metode ini dipilih karena penelitian ini dilakukan oleh peneliti sendiri dari pengumpulan konsep sampai implementasinya. Tahapan communication adalah tahapan peneliti melakukan komunikasi untuk memahami tujuan yang ingin dicapai dalam perancangan sistem ini. Hasil dari komunikasi yang dilakukan adalah inisialisasi proyek, seperti menganalisis permasalahan yang ada mengenai overcharge, dsb, serta mendefinisikan sistem dan fungsi fungsi sistem yang akan dibuat untuk mengatasi permasalahan yang telah dianalisis tersebut. Tahapan planning adalah tahapan perencanaan yang membahas estimasi tugas teknis yang akan dilakukan, analisis resiko, sumber daya yang diperlukan, produk yang ingin dihasilkan, dan penjadwalan kerja yang dilaksanakan. Tahapan Modeling adalah tahapan yang membahas perancangan dan permodelan arsitektur sistem yang berfokus pada perancangan sistem, tampilan interface, dsb. Pada tahapan ini peneliti melakukan desain schematic diagram, flowchart diagram, desain user interface. Tahapan construction adalah tahapan yang membahas proses coding untuk mengubah desain yang sudah dibuat peneliti menjadi program / produk yang diinginkan, setelah proses coding selesai akan dilakukan pengujian terhadap

sistem dan program yang sudah dibuat. Pada tahapan ini peneliti membuat kode untuk program smart charger menggunakan Arduino IDE dengan bahasa C++, dan kode untuk aplikasi Android dan Browser Monitoring. Tahapan Deployment adalah tahapan yang membahas implementasi produk / sistem yang dibuat kepada target subyek yang sudah direncanakan. Pada tahapan ini peneliti melakukan implementasi dengan menampilkan hasil sistem yang dibuat kepada dosen pembimbing dan dosen penyidang untuk memberikan semua informasi hasil penelitian yang telah dibuat peneliti. Pengukuran keberhasilan pada penelitian menggunakan blackbox untuk aplikasi android dan web untuk melakukan pengecekan terhadap setiap fitur yang telah dibuat peneliti berjalan sesuai ekspektasi. Pengukuran keberhasilan pada smart charger menggunakan experiment dengan beberapa test subject yang telah dibuat peneliti.

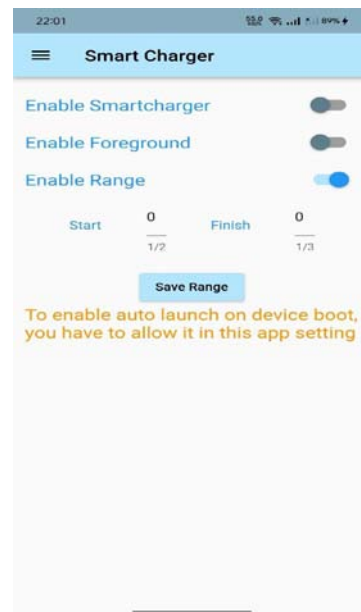
HASIL DAN PEMBAHASAN

Implementasi alat pada penelitian ini berupa rancangan smart charger yang telah dibuat peneliti. Peneliti membuat rancangan dengan bentuk charger pada umumnya menggunakan bahan akrilik. Gambar 2 menunjukkan alat yang sudah dirancang peneliti dalam keadaan mati karena tidak terhubung dengan listrik. Alat akan menyala jika sudah terhubung dengan sumber listrik AC.



Gambar 1 Alat Keadaan Mati

Implementasi Antarmuka Pengguna pada penelitian ini merupakan pembuatan tampilan antarmuka pengguna yang telah dirancang peneliti pada Bab 3. Terdapat 2 jenis tampilan, yaitu tampilan masukan dan tampilan keluaran yang diimplementasikan dalam aplikasi Android dan Web menggunakan Flutter.



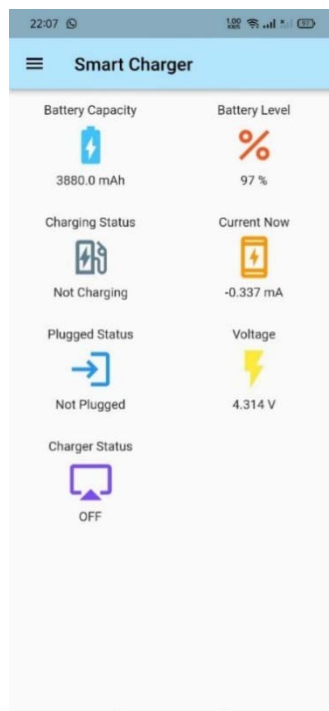
Gambar 2 Tampilan Masukan Aplikasi Android

Gambar 3 menunjukkan tampilan masukan aplikasi Android yang telah dibuat oleh peneliti untuk menentukan pilihan pengguna, terdapat 3 pilihan switch button sebagai berikut :

- Enable Smart Charger untuk menyalakan fitur smart charger, yaitu fitur untuk mengirimkan sinyal kepada smart charger untuk memutuskan arus charger sehingga baterai Android tidak terus terisi saat dayanya sudah penuh
- Enable Foreground untuk menyalakan fitur foreground service, yaitu fitur untuk tetap menjalankan fungsi aplikasi walaupun aplikasi ditutup / di-kill.
- Enable Range untuk menyalakan fitur Range Charging, yaitu fitur untuk menyalakan dan mematikan charger saat arus mencapai angka persentase tertentu.

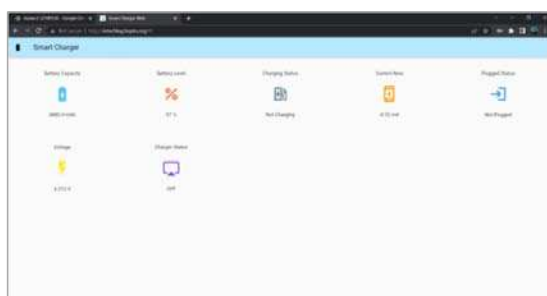
Terdapat juga text “To enable auto launch on device boot, you have to allow it in this app setting” Maksud dari tulisan ini

adalah untuk mengarahkan pengguna jika ingin fungsi aplikasi ini berjalan setelah device menyala secara otomatis tanpa membuka aplikasi ini, pengguna perlu memberikan izin permission auto-launch di pengaturan aplikasi. Hal ini perlu dilakukan karena Android tidak memberikan izin untuk menyalakan auto-launch secara otomatis oleh aplikasi secara langsung.



Gambar 3 Tampilan Keluaran Aplikasi Android

Gambar 4 Menunjukkan Tampilan Keluaran Aplikasi Android yang telah peneliti rancang pada Bab 3. Tampilan ini sudah terintegrasi dengan fungsi untuk membaca data baterai yang dapat diambil langsung dari library Android Baterai Manager.



Gambar 4 Tampilan Keluaran Aplikasi Web

Gambar 5 menunjukkan Tampilan Keluaran Aplikasi Web yang sudah dibuat oleh peneliti menggunakan Flutter sesuai dengan rancangan peneliti sebelumnya pada Bab 3. Tampilan ini sudah terintegrasi dengan MQTT Client untuk menerima data baterai dari android device dan smart charger untuk dapat ditampilkan pada web browser sehingga pengguna dapat melakukan monitoring pada devicenya dari jarak jauh.

Implementasi pengujian pada penelitian ini digunakan untuk menguji semua sistem agar berjalan seperti yang diperlukan. Terdapat 2 implementasi pengujian dari penelitian ini, yaitu implementasi pengujian alat dan implementasi pengujian black box. Implementasi Pengujian Alat digunakan untuk menguji fungsi alat smart charger yang telah dibuat peneliti agar bisa berfungsi sesuai keperluan yang telah dirancang sebelumnya. Pengujian dilakukan menggunakan smart charger yang dibuat peneliti semua pengujian dilakukan menggunakan device yang dimiliki peneliti Hasil pengujian alat pada penelitian ini adalah sebagai berikut.

Tabel 1Hasil Pengujian Smart Charger

Test Subject	Penjelasan	Hasil
Mengisi data baterai secara normal	Peneliti melakukan pengisian baterai secara normal menggunakan <i>smart charger</i> yang dibuat peneliti	<i>Smart charger</i> penelitian berhasil melakukan pengisian daya baterai secara normal
Berhenti melakukan pengisian saat baterai sudah terisi penuh	Peneliti melakukan pengujian pada smart charger penelitian dengan cara observasi untuk membuktikan smart charger penelitian dapat berhenti saat sudah penuh	Smart charger penelitian berhenti melakukan pengisian baterai saat persentase baterai mencapai 100%
Mengisi baterai berdasarkan range yang ditetapkan pengguna	Peneliti melakukan pengujian pada smart charger penelitian dengan cara observasi	Smart charger penelitian berhasil melakukan pengisian daya baterai dalam

Test Subject	Penjelasan	Hasil
	untuk membuktikan Pengisian baterai android berhenti saat persentase baterai mencapai batas atas dan menyala saat persentase baterai android menyentuh batas bawah	range yang diinput user
Arus rata rata yang diterima baterai saat sudah terisi penuh	Peneliti melakukan pengujian pada smart charger penelitian dengan cara pemasangan code pada aplikasi untuk membuktikan Hasil rata rata arus yang diterima smartphone satu menit pertama setelah baterai terisi penuh	Arus rata-rata yang di terima smartphone android saat melakukan pengisian daya baterai menggunakan smart charger penelitian adalah - 0.359mA
Pengisian baterai terhadap multiple devices menggunakan USB Extender	Peneliti melakukan pengujian pada smart charger penelitian dengan cara observasi untuk mendapatkan hasil dari multiple device charging	Arus rata-rata yang di terima smartphone android saat melakukan pengisian daya baterai menggunakan smart charger penelitian adalah - 0.359mA
Pengecekan arus menggunakan multimeter dan nilai yang ditampilkan aplikasi monitoring	Peneliti melakukan pengujian pada smart charger penelitian dengan cara observasi untuk membuktikan Perbandingan toleransi arus yang keluar dari smart charger menggunakan multimeter dengan nilai yang ditampilkan aplikasi monitoring	Arus pada multimeter dan arus yang ditampilkan aplikasi memiliki nilai yang berbeda dengan toleransi selisih 0.2A – 0.3A

Berdasarkan Tabel 1 pengujian smart charger yang dibuat peneliti sudah berjalan

sesuai kebutuhan yang diperlukan dalam penelitian ini, smart charger peneliti berhasil melakukan pengisian baterai seperti charger pada umumnya, dan tidak melakukan pengisian baterai saat baterai Android sudah terisi penuh, smart charger juga berhasil melakukan pengisian baterai berdasarkan range yang ditetapkan pengguna dan arus yang diterima baterai saat sudah terisi penuh adalah -0.359mA sehingga baterai tidak melakukan pengisian daya saat baterai sudah penuh. Smart charger penelitian juga dirancang untuk melakukan pengisian daya baterai pada single device dan arus yang dites menggunakan multimeter dengan yang ditampilkan aplikasi monitoring memiliki nilai yang sama

Tabel 2Hasil Blackbox Testing pada Aplikasi Android

Nama	Ekspektasi	Hasil	Sesuai	
			T	F
Monitoring View	Menampilka n data baterai dan charger	Data baterai dan charger berhasil ditampilkan dengan baik dan lengkap	✓	
Control View	Menampilka n pilihan kontrol enable smarcharger , enable foreground, dan enable range untuk dipilih pengguna	Pilihan kontrol enable smarcharger , enable foreground, dan enable range berhasil ditampilkan	✓	
Enable Smart Charger	Aplikasi dapat memutus arus smart charger yang dibuat peneliti sehingga pengisian daya baterai berhenti saat baterai sudah terisi penuh	Arus smart charger berhasil terputus dan baterai berhenti melakukan pengisian daya saat sudah terisi penuh	✓	
Enable Foreground	Aplikasi dapat tetap berjalan mengawasi	Aplikasi tetap berjalan, smart	✓	

Nama	Ekspektasi	Hasil	Sesuai	
			T	F
	baterai dan dapat menjalankan smart charging dan range charging berdasarkan opsi yang dipilih pengguna walaupun aplikasi ditutup dan di kill dari task	charging dan range charging tetap berfungsi untuk mematikan smart charger berdasarkan kontrol yang dipilih pengguna		
Set Range	Menampilkan text field untuk pengguna mengisi batas bawah dan batas atas pengisian baterai.	Text field untuk batas bawah dan batas atas pengisian baterai berhasil ditampilkan.	✓	
Range Charging	Aplikasi dapat menyalakan arus smart charger saat persentase gawai pengguna mencapai angka batas bawah yang dituliskan pengguna sebelumnya dan dapat mematikan arus smart charger saat persentase gawai pengguna mencapai batas atas yang dituliskan pengguna sebelumnya	Arus smart charger berhasil dinyalakan saat persentase baterai mencapai batas bawah yang pengguna isi sebelumnya dan arus berhasil dimatikan saat persentase gawai pengguna mencapai batas atas yang pengguna isi sebelumnya	✓	

Berdasarkan Tabel 2 semua kebutuhan aplikasi dapat terpenuhi dengan baik, monitoring view, dan control view berhasil ditampilkan sesuai dengan kebutuhan yang diperlukan. Fitur smart charger, foreground, dan range charging juga berhasil memutus

dan menyalakan arus smart charger yang dibuat peneliti dengan baik

Tabel 3 Hasil Blackbox testing pada aplikasi Web

Nama	Ekspektasi	Hasil	Sesuai	
			T	F
Monitoring View	Menampilkan data baterai dan charger	Data baterai dan charger berhasil ditampilkan dengan baik dan lengkap	✓	
Reliability	Website dapat diakses menggunakan internet menggunakan url mtechlog.hopto.org dari mana saja	Website berhasil diakses menggunakan url mtechlog.hopto.org	✓	

Berdasarkan Tabel 3 semua fungsi aplikasi web yang dibuat peneliti berjalan sesuai dengan kebutuhan yang diperlukan. Website mtechlogs.hopto.org dapat diakses menggunakan internet sehingga pengguna dapat menampilkan monitoring view dari web tersebut untuk melakukan monitoring pada smartphone-nya saat melakukan pengisian daya baterai.

KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan peneliti dalam penelitian smart charger ini, dapat disimpulkan beberapa hal sebagai berikut :

1. Smart Charger dibuat peneliti menggunakan modul Wemos D1 Mini yang didalamnya terdapat board ESP8266 sehingga dapat terhubung ke internet melalui Wi-Fi. Aliran listrik AC di konversi menggunakan power supply module untuk memberi arus listrik DC kepada mikrokontroler dan relay.
2. Aplikasi Android dibuat menggunakan Flutter dan library Battery Manager yang disediakan oleh Android API untuk mendapatkan informasi baterai untuk dilakukan pengecekan terhadap status persentase dan pengisian daya baterai sebagai decision maker saat mengirimkan sinyal ke smart charger. Aplikasi ini juga dibuat untuk dapat berjalan di background service sehingga bisa tetap menjalankan

fungsinya walaupun aplikasi tidak sedang dibuka.

3. Aplikasi web dibuat menggunakan Flutter yang sudah mendukung pemrograman multi-platform untuk dapat menyusun dan membuat aplikasi web. Aplikasi web akan menerima data menggunakan MQTT Client dari Android dan smart charger untuk ditampilkan saat dibuka oleh pengguna.
4. Pengiriman dan penerimaan data dilakukan menggunakan MQTT Protocol digabungkan menggunakan timer untuk melakukan publish pada data berdasarkan topic yang sudah ditentukan. Data akan diterima oleh MQTT Client yang terhubung dengan MQTT Broker/Server dan sudah melakukan subscribe pada topic tersebut.

Mikrokontroler Atmega328.

- Putra, D. A., & Mukhaiyar, R. (2020). Monitoring Daya Listrik Secara Real Time. *Voteteknika (Vocational Tek. Elektron. Dan Inform., 8(2), 26.* <https://doi.org/10.24036/voteteknika.v8i2.109138>
- Schnauffer, D., Briggs, N. T. D., & Day, M. (2020). Power Management Fundamentals For Dummies®, Qorvo Special Edition. *John Wiley & Sons, Inc.* <https://borecraft.com/files/qorvo-power-management-fundamentals-for-dummies.pdf>

DAFTAR PUSTAKA

- Budiman, H. (2017). Peran Teknologi Informasi Dan Komunikasi Dalam Pendidikan. *Al-Tadzkiyyah J. Pendidik. Islam, 8(1), 31.* <https://doi.org/10.24042/atjpi.v8i1.2095>
- Dwijantari, S. (2017). *Peramalan Konsumsi Listrik Pada Segmen Rumah Tangga PT PLN (Persero) Distribusi Jawa Timur.*
- Fiantis, D. (1967). Desain Dan Implementasi Charger Baterai Lithium-Ion Menggunakan Constant Current Pulse Charging 500 Watt Untuk Sepeda Motor Listrik. *Angew. Chemie Int. Ed. 6, 1, 951–952.*
- Kuswanto, J., & Radiansah, F. (2018). Media Pembelajaran Berbasis Android Pada Mata Pelajaran Sistem Operasi Jaringan Kelas XI. *J. Media Infotama, 14(1).* <https://doi.org/10.37676/jmi.v14i1.467>
- Prihananto, P. L. D., Facta, M., & Sudjadi, S. (2019). Perancangan Cut Off Sebagai Pemutus Tegangan Dan Arus Sistem Charging Baterai. *Transient, 7(4), 911.* <https://doi.org/10.14710/transient.7.4.911-817>
- Purba, I. P. (2020). *Rancang Bangun Sistem Pengecasan Baterai Bertenaga Surya Metode Pwm 5a Berbasis*