MONITORING *AUTOMATIC TRANSFER SWITCH* DAN BATERAI BERBASIS IOT MENGGUNAKAN MIKROKONTROLER AVR DAN ESP32

Sulthan Imani Akbar[1*)], Darjat[2], dan Ajub Ajulian Zahra Macrina[3]

[123]Program Studi Sarjana Departemen Teknik Elektro, Universitas Diponegoro
Jl. Prof. Soedharto, S.H., Kampus UNDIP Tembalang, Semarang 50275, Indonesia

[*)] *E-mail* : *sulthanimani@student.undip.ac.id*

Abstrak

Saat ini, monitoring *Automatic Transfer Switch* (ATS) dan baterai sebagai sumber listrik cadangan dilakukan secara manual dengan pengamatan langsung pada alat saja, monitoring tidak dapat dilakukan secara fleksibel. Hal tersebut sangat diperhitungkan untuk tempat-tempat yang membutuhkan sumber listrik yang bersifat kontinu namun tempat tersebut jarang dikunjungi. Maka dari itu, penulis pada Tugas Akhir ini membuat suatu sistem untuk mempermudah melakukan monitoring ATS dan baterai dengan konsep *Internet of Things* (IoT). Perancangan sistem menggunakan mikrokontroler ATMega2560, ESP32, LCD, *platform* Firebase, dan Aplikasi Andorid. Data yang dikirimkan dari ATS dan baterai berupa nilai tegangan, arus, serta daya yang diteruskan oleh ATS, nilai *State of Charge* (SoC) baterai, kondisi *charging discharging* baterai, dan mode yang digunakan oleh ATS. Berdasarkan pengujian yang telah dilakukan, data yang dikirimkan dapat ditampilkan dengan jelas pada aplikasi Android dan LCD yang memiliki jeda penampilan rata-rata 10,4 detik untuk aplikasi Android dan rata-rata 2,234 detik untuk LCD. Dengan adanya sistem monitoring ini diharapkan dapat membantu dalam pemantauan ATS dan baterai sehingga dapat melakukan tindakan apabila ATS atau baterai tidak bekerja dengan semestinya.

***Kata kunci**: ATS, Baterai, Monitoring, Internet of Things (IoT), Aplikasi Android, LCD*

Abstract

Currently, monitoring of Automatic Transfer Switch (ATS) and batteries as a backup power source is done manually with direct observation of the device only, monitoring cannot be done flexibly. This is very taken into account for places that require a continuous source of electricity but those places are rarely visited. Therefore, the author in this Final Project created a system to make it easier to monitor ATS and batteries by applying the concept of the Internet of Things (IoT). The design of this system uses an ATMega2560, ESP32, LCD, Firebase, and Andorid Application. The data sent from the system is the value of the voltage, current, and power transmitted by the ATS, the value of the SoC, the condition of charging discharging, and the mode used. Based on the tests, the transmitted data can be displayed clearly on the Android application and LCD which has an average display delay of 10,4 seconds for the Android application and an average of 2,234 seconds for the LCD. With this monitoring system, it is hoped that it can assist in monitoring ATS and the battery so that it can take action if the ATS or battery is not working properly.

***Keywords:** ATS, Battery, Monitoring, Internet of Things (IoT), Android Application, LCD*

1. Pendahuluan

Internet of Things (IoT) merupakan teknologi cerdas yang memiliki fungsi untuk melakukan pengaturan otomatis, berbagi dan mengirim data jarak jauh, serta dapat berinteraksi dan bertindak langsung dalam menghadapi situasi atau perubahan pada objek yang telah ditentukan[1]. Hal ini tentunya dapat dimanfaatkan untuk pekerjaan manusia terutama dalam memberikan kemudahan, rasa aman, dan nyaman.

Automatic Transfer Switch (ATS) adalah *switch* elektronik yang mendetekesi ketika sumber listrik utama mengalami gangguan dan secara otomatis mengoperasikan sumber listrik cadangan[2]. Keandalan dari alat ATS diperhitungkan karena memiliki fungsi untuk menjaga agar listrik tetap dapat digunakan oleh konsumen. Pada saat ini pengamatan terhadap kinerja ATS tidak dapat dilakukan secara langsung oleh konsumen ketika tidak berada pada tempat yang aliran listriknya terhubung dengan ATS, maka konsumen tidak dapat mengetahui apakah ATS bekerja dengan baik atau tidak[3]. Kondisi tersebut sangat

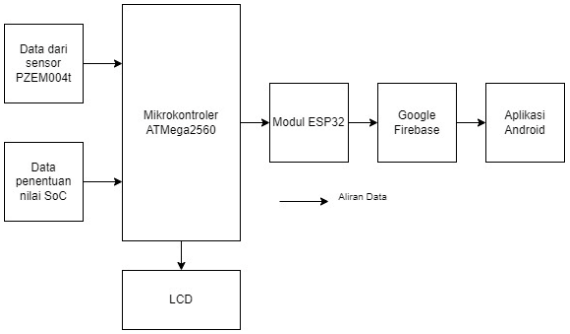
diperhitungkan untuk tempat-tempat yang membutuhkan sumber listrik yang bersifat kontinu namun tempat tersebut jarang dikunjungi orang seperti stasiun pemancar dan gudang penyimpanan. Selain melakukan pengamatan terhadap ATS, kondisi baterai sebagai sumber listrik cadangan juga harus diperhatikan. Apabila sumber utama mengalami gangguan dan kondisi baterai tidak mumpuni untuk digunakan maka kinerja dari ATS akan terhenti dan beban tidak mendapatkan aliran listrik[4].

Hal ini yang mendorong penulis untuk membuat suatu sistem yang dapat melakukan monitoring terhadap kondisi ATS dan baterai dengan memberikan informasi secara *real-time*. Pengguna dapat meninjau kondisi alat secara langsung atau melalui gawai yang bersifat *portable* atau dapat dibawa kemana-mana[5]. Data yang ditampilkan pada *smartphone* berupa data yang dapat menyatakan bahwa ATS bekerja dengan baik dan baterai mampu untuk digunakan yaitu berupa tegangan, arus, dan daya yang dihasilkan ATS, sumber listrik yang digunakan ATS, kondisi kapasitas baterai sebagai sumber listrik cadangan berupa *State of Charge* (SoC) baterai, dan indikator *charger* atau *discharger* pada baterai[6]. Selain itu, pada alat ini menggunakan basis data Firebase untuk penyimpanan data.

2. Metode

2.1 Perancangan Sistem Monitoring ATS dan Baterai

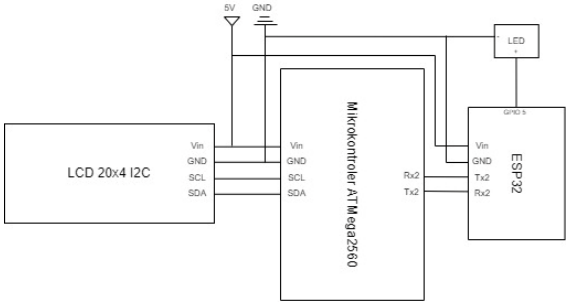
Perancangan sistem monitoring ATS dan baterai menggunakan mikrokontroler AVR dan ESP32 memungkinkan adanya pemantauan ATS dan baterai secara *online* maupun langsung. Pengolahan data akan dilakukan oleh Mikrokontroler ATMega2560, sedangan pengiriman data ke *database* dilakukan menggunakan modul ESP32 sebagai modul *Wi-Fi* dan komunikasi data[7], [8]. Setelah data diterima oleh Firebase, data akan ditampilkan pada *smartphone* melalui aplikasi berbasis Android[9]. Untuk pemantauan secara langsung menggunakan antarmuka LCD I2C 20x4. Perancangan sistem monitoring ATS dan baterai secara keseluruhan ditunjukan pada Gambar 1.



Gambar 1. Diagram Blok Subsistem Monitoring.

2.2 Perancangan Hardware

Gambar 2 menunjukkan *wiring* perancangan *hardware* alat monitoring.



Gambar 2 *Wiring* Perancangan *Hardware* Sistem Monitoring

Sumber tegangan 5V yang dihubungkan ke setiap *port* Vin pada mikrokontroler ATMega2560, modul ESP32, dan LCD, untuk LED menerima sumber dari modul ESP32. Untuk *ground* (GND) juga dihubungkan ke setiap *port* GND pada mikrokontroler ATMega2560, modul ESP32, LCD.

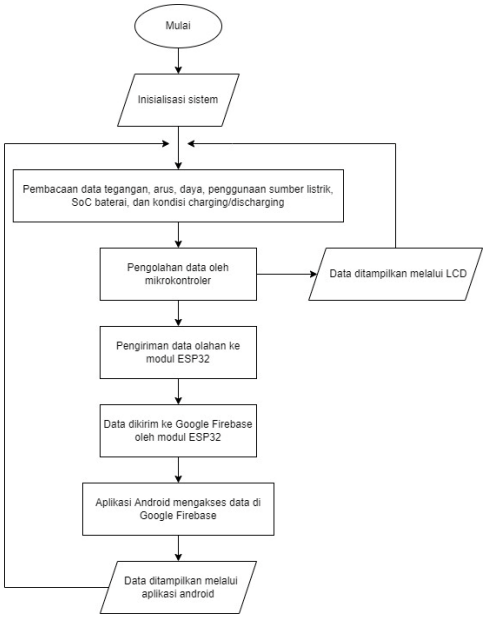
Pemasangan LCD dilakukan sesuai dengan *pin* yang telah ditentukan, untuk *serial data* (SDA) pada LCD dipasangankan dengan *pin* SDA pada mikrokontroler ATMega2560 dan untuk *serial clock* (SCL) pada LCD dipasangkan dengan *pin* SCL pada mikrokontroler ATMega2560[10].

Untuk komunikasi *serial* asinkron UART *pin transmitter* (Tx2) pada mikrokontroler ATMega2560 dihubungkan dengan *pin receiver* (Rx2) modul ESP32, sedangkan *pin trasnsmitter* (Tx2) pada modul ESP32 dihubungkan dengan *pin receiver* (Rx2) pada mikrokontroler ATMega2560[11]. Pemasangan *pin ground* untuk mikrokontroler ATMega2560 dan modul ESP32 dijadikan satu sumber untuk mengurangi *error* pengiriman dan penerimaan data.

2.3 Perancangan Software

Perancangan perangkat lunak (*software*) pada Tugas Akhir ini merupakan perancangan senarai program pada mikrokontroler, modul, LCD, aplikasi Android, dan pengaturan pada sistem Firebase. Gambar 3 menunjukkan

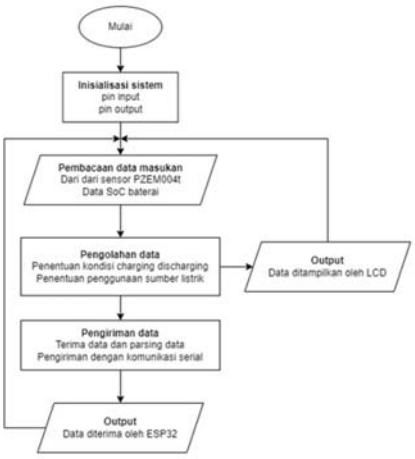
diagram alir (*flowchart*) dari perancangan *software* subsistem moniotoring secara keseluruhan.



Gambar 3. Diagaram Alir (*Flowchart*) *Software* Subsistem Monitoring Secara Keseluruhan

2.3.1 Perancangan Software pada Mikrokontroler ATMega 2560

Perancangan *software* sistem tertanam mikrokontroler ATMega2560 terbagi menjadi lima bagian yaitu inisialisasi sistem, pembacaan data masukan, proses pengolahan data, pengiriman data, dan pengaturan *output* menggunakan antarmuka LCD I2C maupun pengaturan *output* ke modul ESP32. Gambar 4 menunjukkan diagram alir (*flowchart*) dari perancangan *software* ATMega2560.



Gambar 4. Diagram Alir (*Flowchart*) Dari Perancangan *Software* ATMega2560

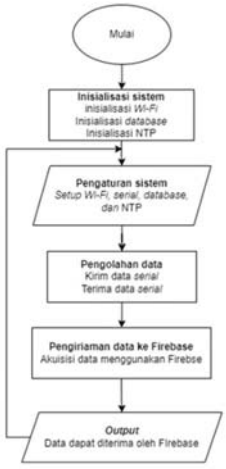
Pada proses pengolahan data, dilakukan penulisan kondisi *charging* atau *discharging* baterai yang digunakan untuk ditampilkan pada LCD maupun aplikasi Android dan penentuan penggunaan sumber listrik yang digunakan. Tabel 1 menjelaskan penentuan kondisi *charging discharging* baterai dan penentuan penggunaan sumber listrik yang digunakan.

Tabel 1. Penentuan kondisi *charging discharging* baterai dan penggunaan sumber listrik

Mode	Sumber listrik utama	SoC	Kondisi baterai	Penggunaan sumber listrik
Otomatis	Tidak ada	>20 %	*Discharging*	Mode Otomatis dengan Sumber Baterai
Otomatis	Tidak ada	<=20 %	Baterai habis, PLN gangguan	Mode Otomatis
Otomatis	Ada	<=55 %	*Charging*	Mode Otomatis dengan Sumber PLN
Otomatis	Ada	>55 %	Baterai tidak digunakan	Mode Otomatis dengan Sumber PLN
Otomatis	Ada	>=20 % atau <=55% atau <=20%	*Charging*	Mode Otomatis dengan Sumber PLN
Otomatis	Ada	<100 %	*Charging*	Mode Otomatis dengan Sumber PLN
Otomatis	Ada	>=100 %	Baterai tidak digunakan	Mode Otomatis dengan Sumber PLN
Otomatis	Ada	>20 % dan <=55 %	*Charging*	Mode Otomatis dengan Sumber PLN
Otomatis	Ada	<=20 %	*Charging*	Mode Otomatis dengan Sumber PLN
Manual	Tidak ada	>20 %	*Discharging*	Mode Manual dengan Sumber Baterai
Manual	Tidak ada	<=20 %	Baterai habis, PLN Gangguan	Mode Otomatis
Manual	Ada	>55 %	*Dicharging*	Mode Manual dengan Sumber Baterai
Manual	Ada	>20 % dan <=55 %	*Dicharging*	Mode Manual dengan Sumber Baterai

2.3.2 Perancangan Software pada Modul ESP32

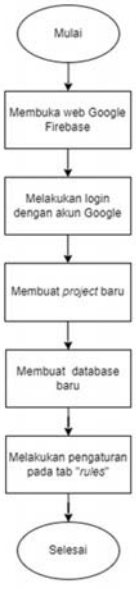
Perancangan perangkat lunak untuk modul ESP32 terbagi menjadi lima bagian yaitu: inisialisasi sistem, pengaturan sistem, pengolahan data, pengiriman data ke firebase, dan keluaran atau output yang diinginkan. Gambar 5 menunjukkan diagram alir (*flowchart*) perancangan softawre modul ESP32.



Gambar 5. Diagram Alir (*Flowchart*) Perancangan *Software* Modul ESP32

2.3.3 Perancangan Firebase Relatime Database

Perancangan Firebase *Realtime Database* dilakukan pada website Firebase dengan terlebih dahulu melakukan login menggunakan akun Google. Gambar 6 menunjukkan diagram alir (*flowchart*) dari perancangan pada Firebase *Realtime Database.*



Gambar 6. Diagram Alir (*Flowchart)* Perancangan Firebase *Realtime Database*

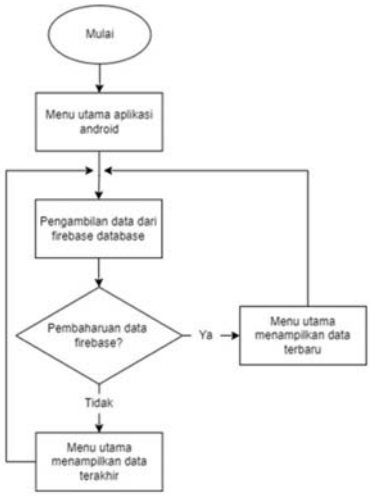
Langkah awal setelah masuk pada *website* Firebase dan melakukan login adalah membuat *project* baru. Setelah project baru telah terbentuk langkah selanjutnya adalah membuka menu Firebase *Realtime Database* untuk membuat *database* baru. Langkah terakhir dalam perancangan Firebase adalah melakukan pengaturan pada menu *Realtime Database* tab bertuliskan "*Rules*".

2.3.4 Peancangan Aplikasi Android

Pada aplikasi Android ini perancangan aplikasi terbagi menjadi empat bagian penyusunan yaitu: perancangan cloud server, perancangan sistem database, perancangan sistem monitoing, perancangan tampilan aplikasi android. Gambar 7. menunjukkan diagram alir (*flowchart*) dari perancangan aplikasi Monitoring ATS Baterai.

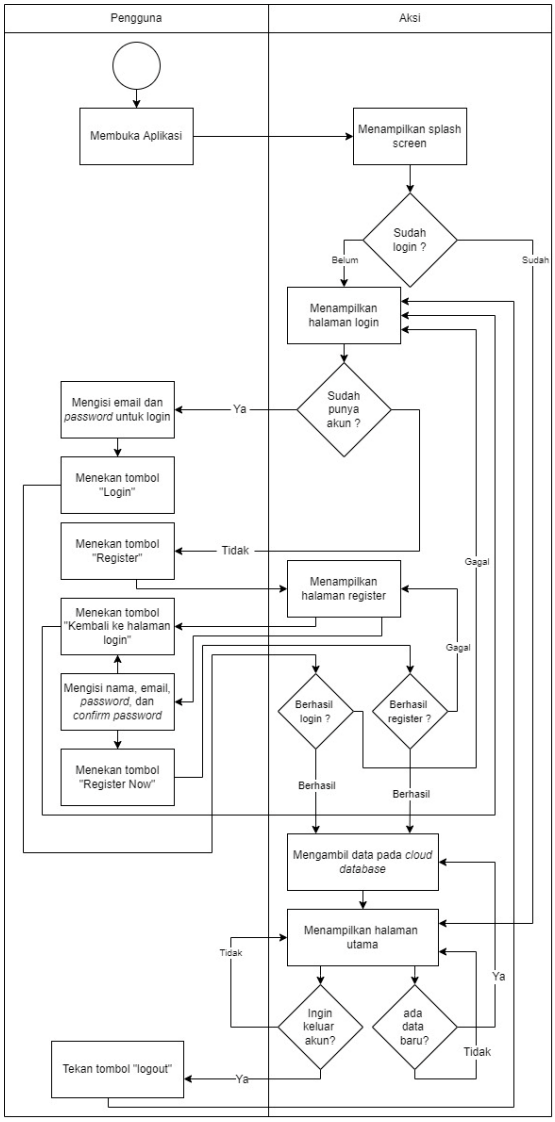
Gambar 7. Diagram Alir (*Flowchart*) Perancangan Aplikasi Android

Pada perancangan sistem monitoring diharapkan dapat menampilkan data terbaru yang dikirimkan ke Firebase secara *real-time*[14]. Gambar 8 menunjukkan diagram alir (*flowchart*) dari perancangan sistem monitoring.



Gambar 8. Diagram Alir (*Flowchart*) Dari Perancangan Sistem Monitoring

Perancangan tampilan Aplikasi Android akan membahas terkait diagram *activity* dan perancangan tampilan *interface* untuk tiap *activity*. Gambar 9 menunjukkan diagram *activity* dari perancangan tampilan aplikasi Android.



Gambar 9. Diagram *Activity* Perancangan Tampilan Aplikasi Android

3. Metodologi Pengujian

Pengujian dan analisis pada sistem monitoring ATS dan baterai berbasis IoT dengan menggunakan mikrokontroler Atmega2560 dan modul ESP32 meliputi pengujian perangkat keras (hardware), dan pengujian perangkat lunak (software).

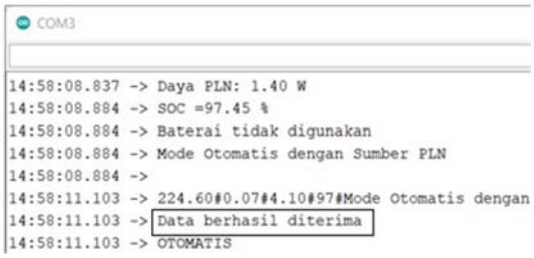
3.1 Pengujian Perangkat Keras

3.1.1 Pengujian Mikrokontroler ATMega 2560

Pengujian mikrokontroler ATMega2560 meliputi pengujian komunikasi *serial* penerimaan dan pengiriman data dari dan ke modul ESP32.

A. Pengujian Penerimaan Data *Serial*

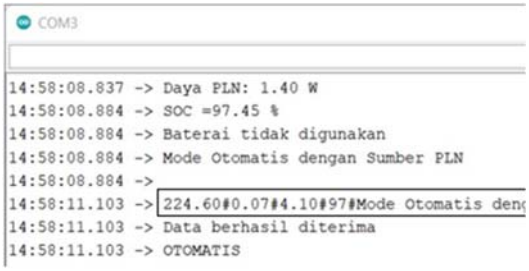
Apabila data telah diterima dapat ditandai dengan adanya tulisan "Data berhasil diterima" pada tampilan *serial monitor*. Gambar 10 menunujukan tampilan *serial monitor* ketika berhasil menerima data.



Gambar 10. Tampilan *Serial Monitor* Ketika Mikrokontroler ATMega2560 Berhasil Menerima Data

B. Pengujian Pengiriman Data *Serial*

Apabila data telah berhasil dikirimkan dapat ditandai dengan adanya tulisan hasil parsing data pada tampilan *serial monitor*. Gambar 11 menunujukan tampilan *serial monitor* ketika berhasil mengirimkan data.



Gambar 11. Tampilan *Serial Monitor* Ketika Mikrokontroler Atmega2560 Berhasil Mengirimkan Data

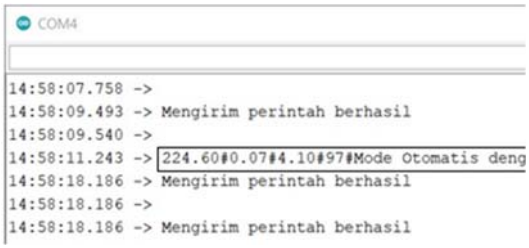
3.1.2 Pengujian Modul ESP32

Pengujian modul ESP32 meliputi pengujian komunikasi *serial* penerimaan dan pengiriman data dari dan ke mikrokontroler ATMega2560 serta pengujian terhadap koneksi internet.

A. Pengujian Penerimaan Data *Serial*

Apabila data telah diterima dapat ditandai dengan adanya tulisan hasil parsing data pada tampilan *serial monitor*.

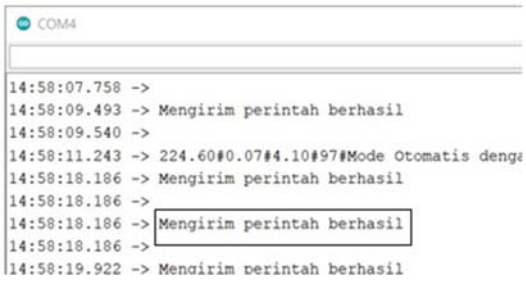
Gambar 12 menunujukan tampilan *serial monitor* ketika berhasil menerima data.



Gambar 12. Tampilan *Serial Monitor* Ketika Modul ESP32 Berhasil Menerima Data

B. Pengujian Pengiriman Data *Serial*

Apabila data telah berhasil dikirimkan dapat ditandai dengan adanya tulisan "Mengirim perintah berhasil" pada tampilan *serial monitor*. Gambar 13 menunujukan tampilan *serial monitor* ketika berhasil mengirimkan data.



Gambar 13. Tampilan *Serial Monitor* Ketika Modul ESP32 Berhasil Mengirimkan Data

C. Pengujian Koneksi Internet

Kondisi dapat terbagi menjadi dua bagian yaitu kondisi tidak terhubung koneksi internet dan kondisi terhubung koneksi internet. Pada *serial monitor* akan menampilkan tulisan *"WiFi connected"* apabila modul telah terhubung dengan koneksi internet. Gambar 14 menunjukkan tampilan *serial monitor* ketika pengujian koneksi internet berhasil dilakukan.

Gambar 14 Tampilan *Serial Monitor* Ketika Pengujian Koneksi Internet Berhasil Dilakukan

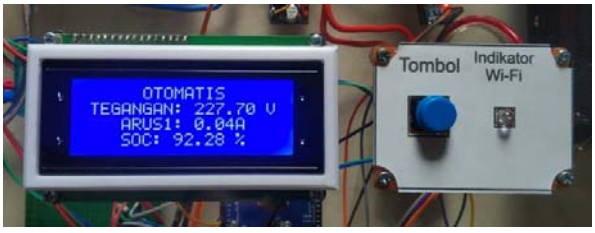
Untuk LED, apabila modul ESP32 telah terhubung dengan koneksi internet maka LED akan menyala. Gambar 15 menunjukkan LED menyala apabila terhubung dengan koneksi internet.

Gambar 15 Menunjukkan LED Menyala Apabila Terkoneksi Internet

Apabila modul ESP32 tidak terhubung dengan internet maka pada *serial monitor* akan menampilkan tulisan *"Connecting to Wi-Fi ….."*. Hal ini terjadi ketika sistem awal mula dihidupkan dan ketika sistem menghubungkan kembali koneksi internet. Jika telah terhubung kembali maka pada *serial monitor* akan menampilkan tulisan *"WiFi connected"*. Gambar 16 menunjukkan tampilan *serial monitor* ketika menghubungkan kembali koneksi internet berhasil dilakukan.

Gambar 16. Tampilan *Serial Monitor* Ketika Menghubungkan Kembali Koneksi Internet Berhasil Dilakukan.

LED akan padam apabila modul ESP32 tidak tehubung dengan koneksi internet. Gambar 17 menunjukkan LED padam apabila tidak terhubung dengan koneksi internet.



Gambar 17. Menunjukkan LED Padam Apabila Tidak Terhubung dengan Koneksi Internet

3.2 Pengujian Perangkat Lunak

3.2.1 Pengujian Pengiriman Data ke Cloud Database

Respon Pengiriman data yang telah berhasil dikirmkan dapat dilihat dari tampilan Firebase *Realtime Database*. Gambar 18 menunjukkan tampilan data yang diterima oleh Firebase *Relatime Database*.

Gambar 18. Tampilan Firebase *Realtime Database*

Dalam satu kali pengiriman ke *cloud database* memuat 5 data sesuai dengan runtutan senarai program pada modul ESP32. Pada Firebase *Realtime Database* penamaan *parent* menggunakan waktu *real-time* saat data dikirimkan sehingga mempermudah pembacaan *database*.

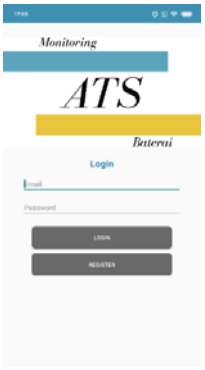
3.2.2 Pengujian Aplikasi Android

Device yang digunakan untuk pengujian penampilan aplikasi Android adalah Xiaomi Redmi 6 Pro dengan *Operating System* (OS) Android 8.1 (Oreo), layar 5,84 inci resolusi 1080 x 2280 piksel, CPU Octa-core 2,0 GHz Cortex-A53. Indikator keberhasilan pengujian yaitu seluruh halaman aplikasi dapat terbuka dengan baik pada *device.*

A. Pengujian *Splash Screen*
Splash screen hanya berupa gambar logo yang ditampilkan selama 3 detik sebelum menuju Halaman *Login*. Gambar 19 menunjukkan tampilan pada *splash screen*.

Gambar 19. Tampilan *Splash Screen*

B. Pengujian Halaman *Login*
Gambar 20 menunjukkan tampilan pada halaman *Login*. Tampilan halaman *Login* terdapat logo sebagai *branding*, dua kolom untuk mengisi email dan *password* sesuai dengan kolom yang ditentukan, dan dua tombol untuk melakukan *login* atau registrasi



Gambar 20. Tampilan Halaman *Login*

C. Pengujian Halaman *Register*
Gambar 21 menunjukkan tampilan pada halaman *register*. Tampilan halaman *register* terdapat tuliskan judul yaitu "Registrasi", terdapat empat kolom untuk dimasukkan data yang diperlukan seperti nama lengkap, email, *password*, dan kolom untuk melakukan konfirmasi *password*, lalu terdapat dua tombol yang digunakan untuk registrasi sekarang atau kembali ke halaman *login.*

Gambar 21. Tampilan Halaman *Register*

D. Pengujian Halaman Utama
Pada halaman utama terdapat judul di bagian atas bertuliskan "Monitoring ATS dan Baterai", nama pengguna, tanggal dan hari. Bagian tengah terdapat tulisan dengan latar belakang biru muda untuk menampilkan informasi data mode yang digunakan ATS, dan terdapat tulisan dengan latar belakang kuning digunakan untuk menampilkan informasi kondisi *charging discharging* baterai. Pada bagian tengah kebawah terdapat tulisan untuk menampilkan nilai tegangan, arus, daya, dan nilai SoC baterai. Terdapat satu tombol di bagain tengah bawah

untuk *logout* dan dibawah tombol terdpat tulisan "Teknik Elektro Univeristas Diponegoro" sebagai *branding*. Gambar 22 menunjukkan tampilan dari halaman utama.

Gambar 22. Tampilan Halaman Utama

3.2.3 Pengujian Penampilan Data pada LCD dan Aplikasi Android

Pengujian penampilan data pada LCD dan aplikasi Android dilakukan dengan mengambil sampel data pada kondisi yang telah ditentukan pada Tabel 1. Respon tampilan data dapat dilihat dari layar LCD dan aplikasi Android untuk setiap kondisinya.

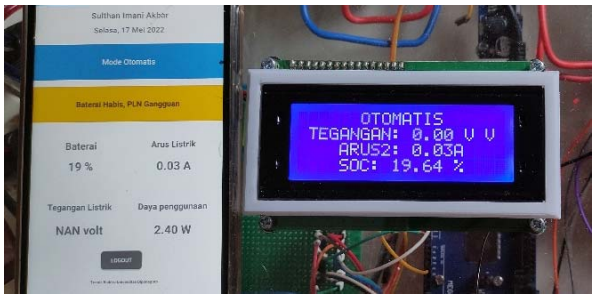
A. Kondisi Pertama

Kondisi pertama adalah ketika alat menggunakan mode otomatis namun sumber listrik utama tidak ada dan SoC baterai > 20%. Gambar 23 menunjukkan tampilan layar LCD dan aplikasi Android untuk kondisi pertama.

Gambar 23. Tampilan Data pada LCD dan Aplikasi Android untuk Kondisi Pertama

B. Kondisi Kedua

Alat menggunakan mode otomatis namun sumber listrik utama tidak ada dan SoC baterai <= 20%. Gambar 24 menunjukkan tampilan layar LCD dan aplikasi Android untuk kondisi kedua.



Gambar 24. Tampilan Data pada LCD dan Aplikasi Android untuk Kondisi Kedua

C. Kondisi Ketiga

Alat menggunakan mode otomatis terdapat sumber listrik utama dan SoC <=55%. Gambar 25 menunjukkan tampilan layar LCD dan aplikasi Android untuk kondisi ketiga.

Gambar 25. Tampilan Data pada LCD dan Aplikasi Android untuk Kondisi Ketiga

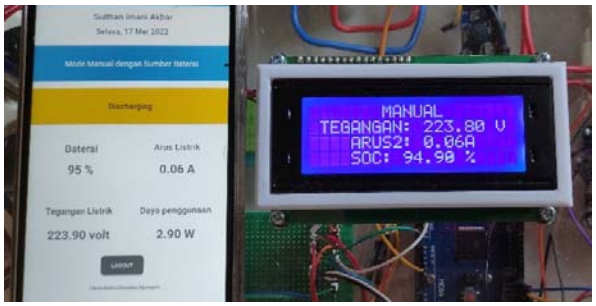
D. Kondisi Keempat

Alat menggunakan mode otomatis terdapat sumber listrik utama dan SoC baterai > 55%. Gambar 26 menunjukkan tampilan layar LCD dan aplikasi Android untuk kondisi keempat.

Gambar 26. Tampilan Data pada LCD dan Aplikasi Android untuk Kondisi Keempat

E. Kondisi Kelima

Alat menggunakan mode manual namun sumber listrik utama tidak ada dan SoC baterai > 20%. Gambar 27 menunjukkan tampilan layar LCD dan aplikasi Android untuk kondisi kelima.



Gambar 27. Tampilan Data pada LCD dan Aplikasi Android untuk Kondisi Kelima

F. Kondisi Keenam

Alat menggunakan mode manual namun sumber listrik utama tidak ada dan SoC <= 20%. Gambar 28 menunjukkan tampilan layar LCD dan aplikasi Android untuk kondisi kesebelas.

Gambar 28. Tampilan Data pada LCD dan Aplikasi Android untuk Kondisi Keenam

Berdasarkan hasil pengujian yang dilakukan tampilan pada LCD berupa mode yang digunakan, nilai tegangan, nilai arus, dan nilai SoC. Untuk tampilan aplikasi android berupa kondisi baterai, kondisi penggunaan sumber listrik atau mode, nilai tegangan, nilai arus, nilai daya penggunaan, dan nilai SoC. Apabila tampilan pada layar LCD pada baris kedua bertuliskan "ARUS1" artinya ATS menggunakan PLN sebagai sumber listrik, jika bertuliskan "ARUS2" artinya ATS menggunakan baterai sebagai sumber listrik.. Perbedaan nilai yang ditampilkan pada layar LCD dengan aplikasi android disebabkan oleh waktu jeda atau *delay* dikarenakan komunikasi *serial* dan koneksi internet yang digunakan. Pada pengujian diatas juga membuktikan bahwa data yang ditampilkan pada aplikasi Android telah sesuai data pada ATS dengan LCD sebagai acuan.

3.2.4 Pengujian Waktu Jeda Penampilan Layar LCD dan Aplikasi Android

Pengujian waktu jeda penampilan untuk tampilan LCD diperoleh dari tampilan *serial monitor*. Hasil pengujian waktu perubahan tampilan untuk antarmuka LCD ditunjukan pada Tabel 2.

Tabel 2. Hasil Pengujian Waktu Perubahan Tampilan Antarmuka LCD

Pengujian ke -	Waktu perubahan (sekon)	Pengujian ke -	Waktu perubahan (sekon)
1	2,268	11	2,220
2	2,203	12	2,251
3	2,250	13	2,188
4	2,238	14	2,266
5	2,234	15	2,235
6	2,236	16	2,220
7	2,219	17	2,250
8	2,204	18	2,219
9	2,267	19	2,238
10	2,235	20	2,232

Waktu rata-rata yang diperoleh adalah 2,234 sekon. Hal ini berarti diperlukan waktu 2,234 sekon untuk melakukan perubahan tampilan layar LCD ke data yang terbaru. Waktu perubahan ini sangat berpengaruh pada keberjalanan program dan hasil pembacaan sensor yang diperoleh mikrokontroler.

Pengujian waktu jeda penampilan untuk aplikasi Android diperoleh dari Firebase *Realtime Database*. Hasil pengujian waktu perubahan tampilan untuk antarmuka aplikasi Android ditunjukan pada Tabel 3.

Tabel 3. Hasil Pengujian Waktu Perubahan Tampilan Antarmuka Aplikasi Android

Pengujian ke -	Waktu perubahan (sekon)	Pengujian ke -	Waktu perubahan (sekon)
1	11	11	10
2	13	12	9
3	13	13	12
4	9	14	8
5	7	15	11
6	8	16	8
7	7	17	10
8	9	18	9
9	9	19	20
10	18	20	7

Waktu rata-rata yang diperoleh adalah 10,4 sekon. Hal ini berarti diperlukan waktu 10,4 sekon untuk melakukan perubahan tampilan layar LCD ke data yang terbaru. Waktu perubahan ini sangat berpengaruh pada komunikasi *serial* dan koneksi internet yang terhubung dengan modul ESP32 maupun gawai yang digunakan.

Berdasarkan hasil dua pengujian waktu jeda penampilan, terdapat perbedaan selisih antara waktu jeda penampilan LCD dengan waktu jeda penampilan aplikasi Android. Hal ini mengakibatkan perbedaan data antara keduanya. Hasil pengujian selisih perbedaan nilai tegangan pada LCD dengan aplikasi android pada waktu bersamaan ditunjukan pada Tabel 4.

Tabel 4. Hasil Pengujian Selisih Perbedaan Nilai Tegangan pada LCD dengan Aplikasi Android

Tegangan pada LCD (V)	Tegangan pada aplikasi (V)	Error (%)
225,00	225,10	0,044
223,90	224,00	0,045
223,80	223,90	0,045
223,40	223,50	0,045
223,70	223,90	0,089
224,40	224,40	0,000
224,50	224,70	0,089
224,20	224,20	0,000
224,00	224,20	0,089
225,90	226,00	0,044
Rata-rata error		0,049

Pada indikator tegangan terdapat tujuh data berbeda dari 10 pengujian dengan rata-rata *error* sebesar 0,049%. Hal ini membuktikan bahwa sistem monitoring yang dibuat dapat berkerja dengan baik. Selanjutnya adalah hasil pengujian selisih perbedaan nilai arus pada LCD dengan aplikasi Android pada waktu bersamaan ditunjukan pada Tabel 5.

Tabel 5. Hasil Pengujian Perbedaan Nilai Arus pada LCD dengan Aplikasi Adnroid

Arus pada LCD (A)	Arus pada aplikasi (A)	Error (%)
0,06	0,06	0
0,03	0,03	0
0,12	0,12	0
0,12	0,13	0,077
0,03	0,03	0
0,13	0,13	0
0,13	0,13	0
0,05	0,05	0
0,04	0,04	0
0,13	0,13	0
Rata-rata error		0,008

Pada indikator arus terdapat satu data berbeda dari 10 pengujian yang dilakukan dengan selisih 0,01A, sehingga didapatkan rata-rata *error* sebesar 0,008%. Hal ini membuktikan bahwa sistem monitoring yang dibuat dapat berkerja dengan baik.

4. Kesimpulan

Telah dikembangkan sistem monitoring ATS dan baterai berbasis IoT dengan menggunakan mikrokontroler AVR dan ESP32. Komunikasi data antara mikrokontroler ATMega2560, ESP32, Firebase, dan aplikasi Android berhasil dibuat. Pengujian penampilan data pada LCD dan aplikasi adnroid berhasil dibuat dan dapat berfungsi dengan baik. Terdapat perbedaan jeda waktu antara penampilan data pada LCD dengan penampilan data pada aplikasi android sehingga terdapat perbedaan nilai ketika kedua antarmuka ditampilan bersama, perbedaan nilai tegangan memiliki rata-rata *error* sebesar 0,049% sedangkan untuk nilai arus memiliki rata-rata *error* 0,008%. Dari nilai rata-rata *error* yang sangat kecil membuktikan bahwa sistem monitoring yang dibuat dapat bekerja dengan baik.

Referensi

[1] S. Madakam, R. Ramaswamy, and S. Tripathi, "Internet of Things (IoT): A Literature Review," *J. Comput. Commun.*, vol. 03, no. 05, pp. 164–173, 2015, doi: 10.4236/jcc.2015.35021.

[2] B. A. Farhan, "Design and Implementation of an Automatic Transfer Switch for a Single Phase Power Generator," no. 7, pp. 16–20, 2021.

[3] I. A. Lazuardi, I. W. Farid, and W. Priananda, "Automatic Transfer Switch Dilengkapi Fitur Monitoring Website pada On-Grid Solar Home System," vol. 10, no. 2, 2021.

[4] M. Q. Azeem, Habib-Ur-Rehman, S. Ahmed, and A. Khattak, "Design and analysis of switching in automatic transfer switch for load transfer," *ICOSST 2016 - 2016 Int. Conf. Open Source Syst. Technol. Proc.*, no. June 2018, pp. 129–134, 2017, doi: 10.1109/ICOSST.2016.7838589.

[5] Z. H., H. A., and M. M., "Internet of Things (IoT): Definitions, Challenges and Recent Research Directions," *Int. J. Comput. Appl.*, vol. 128, no. 1, pp. 37–47, 2015, doi: 10.5120/ijca2015906430.

[6] M. Murnane, "A Closer Look at State of Charge (SOC) and State of Health (SOH) Estimation Techniques for Batteries," *Analog devices*, p. 8, 2017.

[7] A. Appiani, "Arduino ® MEGA 2560 Rev3 Features." Arduino, Italy, pp. 1–18, 2022, [Online]. Available: www.arduino.cc.

[8] E. Systems, "ESP32 Series." Espressif Systems, Shanghai, p. 68, 2022, [Online]. Available: www.espressif.com.

[9] B. Kirthika, S. Prabhu, and S. Visalakshi, "Android Operating System A Review," *Int. J. Trend Res. Dev.*, vol. 2, no. 5, pp. 260–264, 2015, [Online]. Available: www.ijtrd.com.

[10] A. Tyagi and S. Chatterjee, "Liquid Crystal Display : Environment & Technology," vol. 1, no. 7, pp. 110–123, 2013.

[11] Y. Y. Fang and X. J. Chen, "Design and simulation of UART serial communication module based on VHDL," Shanghai, China, 2011. doi: 10.1109/ISA.2011.5873448.