

Sistem Monitoring Baterai Robot Tel-U Interactive Food Assistant

Muhammad Jibrán Hady
School Of Electrical Engineering
Telkom University
Bandung, Indonesia
jibranhady@student.telkomuniversity.ac.id

Faisal Candrasyah Hasibuan, S.T., M.T
School Of Electrical Engineering
Telkom University
Bandung, Indonesia
faicanhasfcb@telkomuniversity.ac.id

R Rogers Dwiputra Setiady, S.T., M.T.
School Of Electrical Engineering
Telkom University
Bandung, Indonesia
rogerssetiady@telkomuniversity.ac.id

Abstrak — Sistem monitoring baterai pada robot pengantar makanan Tel-U Interactive Food Assistant (TIFA) dirancang untuk meningkatkan efisiensi operasional di Tel-U Coffee dengan mengatasi masalah manajemen daya pada robot. Dalam lingkungan yang sibuk, terutama pada jam-jam sibuk, pemantauan status baterai robot secara real-time menjadi krusial untuk memastikan kelancaran operasional dan mencegah terjadinya gangguan yang disebabkan oleh kehabisan daya. Penelitian ini mengembangkan sistem pemantauan baterai berbasis Internet of Things (IoT), yang memungkinkan barista dan administrator untuk memantau status baterai robot secara jarak jauh melalui aplikasi Android dan web monitoring. Teknologi yang digunakan meliputi ESP32 sebagai mikrokontroler, protokol Socket.IO untuk komunikasi data real-time, dan MySQL untuk penyimpanan data baterai. Sistem ini tidak hanya menampilkan status baterai secara langsung, tetapi juga memberikan notifikasi otomatis jika level baterai mendekati batas minimum, sehingga memungkinkan pengguna untuk melakukan tindakan preventif seperti pengisian ulang. Dengan adanya sistem pemantauan yang terintegrasi ini, TIFA berhasil meningkatkan efisiensi operasional dengan mengurangi downtime akibat baterai habis dan memastikan robot dapat terus beroperasi dalam kondisi optimal. Hasil pengujian menunjukkan tingkat kepuasan pengguna yang tinggi, serta kemampuan sistem dalam menangani lonjakan pesanan dengan memantau dan mengelola penggunaan baterai secara efektif.

Kata kunci— ESP32, IoT, Pemantauan Baterai, Voltage Sensor.

I. PENDAHULUAN

Dalam beberapa tahun terakhir, industri kafe dan restoran mengalami perubahan signifikan dalam cara pelayanan kepada pelanggan, terutama dengan adanya perkembangan teknologi robotika dan Internet of Things (IoT). Salah satu penerapan teknologi tersebut adalah robot pengantar makanan, yang dirancang untuk membantu proses pengantaran pesanan di kafe atau restoran, meningkatkan efisiensi operasional, dan memberikan pengalaman yang lebih baik bagi pelanggan. Namun, meskipun teknologi ini menawarkan potensi besar, tantangan terkait pemantauan status baterai pada robot tetap menjadi hambatan yang signifikan. Pada robot pengantar makanan, terutama yang digunakan dalam lingkungan dengan permintaan tinggi seperti Tel-U Coffee, masalah utama yang sering terjadi adalah kehabisan daya atau gangguan pada pengisian ulang

baterai, yang dapat menyebabkan terhentinya operasional robot dan menurunkan kualitas layanan. Tel-U Coffee menghadapi masalah ini selama jam sibuk (peak hours), ketika volume pesanan meningkat pesat, dan robot harus dapat beroperasi secara terus-menerus tanpa gangguan. Sebelumnya, sistem pemantauan baterai pada robot TIFA tidak cukup efisien untuk memantau status baterai secara real-time. Tanpa adanya pemantauan yang akurat dan pemberitahuan dini, robot sering kali kehabisan daya pada saat yang kritis, menyebabkan keterlambatan dalam pengantaran dan berkurangnya kepuasan pelanggan. Oleh karena itu, pengembangan sistem monitoring baterai yang dapat memberikan informasi status baterai secara real-time, serta notifikasi ketika baterai mencapai batas minimum, menjadi kebutuhan mendesak untuk meningkatkan efisiensi operasional[1],[7].

Penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan sistem pemantauan baterai berbasis Internet of Things (IoT) pada robot pengantar makanan TIFA, yang dapat memantau status baterai robot secara langsung dan memungkinkan tindakan preventif yang cepat. Sistem ini mengintegrasikan teknologi ESP32 sebagai mikrokontroler, Socket.IO untuk komunikasi data, dan MySQL sebagai basis data untuk menyimpan data status baterai. Dengan sistem ini, barista dan administrator dapat memantau status baterai robot dari jarak jauh melalui aplikasi Android dan web monitoring, sehingga dapat mengurangi downtime dan memastikan robot terus beroperasi dengan optimal.

II. KAJIAN TEORI

1. Robot Pengantar Makanan

Robot pengantar makanan adalah aplikasi teknologi robotika yang dirancang untuk mengotomatisasi proses pengantaran pesanan di restoran atau kafe. Teknologi ini bertujuan untuk meningkatkan efisiensi operasional dengan mengurangi beban kerja staf dan mempercepat proses pengantaran, terutama pada jam-jam sibuk. Robot ini dilengkapi dengan berbagai sensor dan sistem kontrol untuk bergerak otonom, mendeteksi rintangan, serta menyesuaikan tray untuk kapasitas pengantaran. Dengan meningkatnya permintaan terhadap layanan yang cepat dan efisien, robot pengantar makanan semakin banyak diterapkan dalam sektor

kuliner. Namun, tantangan utama yang harus dihadapi adalah manajemen daya robot yang bergantung pada kapasitas baterai dan pengisian ulang yang tepat waktu untuk menjaga kelancaran operasional.

2. Pemantauan Baterai pada Robot Pengantar Makanan

Pemantauan baterai robot pengantar makanan sangat penting untuk memastikan bahwa robot dapat beroperasi secara berkelanjutan tanpa gangguan. Salah satu cara untuk memantau status baterai secara real-time adalah dengan menggunakan sensor tegangan (voltage sensor) yang dapat mengukur tingkat daya yang tersisa pada baterai. Tanpa sistem pemantauan yang memadai, robot dapat berhenti beroperasi pada saat yang krusial, yang akan mengurangi efisiensi operasional dan kepuasan pelanggan. Dalam sistem ini, data dari sensor tegangan dikirimkan secara real-time ke platform monitoring yang dapat diakses oleh operator melalui aplikasi Android atau web monitoring. Sistem ini juga dilengkapi dengan notifikasi otomatis yang memberi tahu operator ketika tingkat baterai mencapai level minimum, memungkinkan tindakan preventif seperti pengisian ulang.

3. Penggunaan Voltage Sensor dalam Pemantauan Baterai

Voltage sensor adalah komponen penting dalam sistem pemantauan baterai yang digunakan untuk mengukur tegangan yang ada pada baterai robot. Sensor ini bekerja dengan mengukur tegangan listrik yang ada pada terminal positif dan negatif baterai, kemudian mengirimkan data tersebut ke mikrokontroler untuk diproses. Dalam aplikasi robot pengantar makanan, sensor tegangan digunakan untuk memantau status daya secara akurat dan memberikan informasi yang diperlukan untuk menjaga agar robot tetap dapat beroperasi dengan efisien[2][3]. Penggunaan voltage sensor memungkinkan pengumpulan data secara real-time tentang status baterai robot, serta membantu memprediksi kapan pengisian ulang diperlukan. Hal ini sangat penting untuk menghindari kegagalan operasional akibat kehabisan baterai di tengah tugas pengantaran.



GAMBAR 1
Voltage Sensor

4. ESP32 sebagai Mikrokontroler dalam IoT

ESP32 adalah mikrokontroler berbasis Wi-Fi dan Bluetooth yang banyak digunakan dalam aplikasi IoT karena kemampuannya dalam menghubungkan perangkat dengan jaringan secara efisien. Dalam sistem pemantauan baterai robot TIFA, ESP32 digunakan untuk mengendalikan sensor tegangan dan mengirimkan data status baterai robot ke server menggunakan Socket.IO[9],[10]. ESP32 mendukung komunikasi data yang cepat dan handal, sehingga memungkinkan pengiriman informasi status baterai secara real-time tanpa gangguan. Keunggulan ESP32 terletak pada konsumsi daya yang rendah, memungkinkan penggunaan perangkat dalam waktu yang lama tanpa perlu khawatir mengenai konsumsi energi yang berlebihan[4].

5. Arsitektur Model-View-Controller (MVC)

Model-View-Controller (MVC) merupakan pola arsitektur perangkat lunak yang memisahkan aplikasi menjadi tiga komponen utama: *Model*, *View*, dan *Controller*. Model

berfungsi sebagai representasi logika data, yang bertugas mengakses dan memanipulasi basis data. View adalah bagian antarmuka pengguna (user interface) yang menyajikan data ke pengguna. Sementara itu, Controller bertindak sebagai penghubung antara Model dan View dengan menangani permintaan pengguna dan logika bisnis aplikasi. Pemisahan tanggung jawab ini memungkinkan pengembangan sistem menjadi lebih modular, terstruktur, dan mudah dalam proses debugging serta pemeliharaan. Dalam konteks aplikasi backend berbasis IoT, seperti sistem pemantauan baterai pada robot, penerapan arsitektur MVC memberikan keuntungan dalam hal pengelolaan endpoint REST API serta integrasi dengan komunikasi real-time. Router akan mengarahkan permintaan dari perangkat (seperti ESP32) ke Controller, yang kemudian memproses data dan meneruskannya ke Model untuk disimpan di basis data, atau menyebarkannya ke client melalui protokol WebSocket menggunakan Socket.IO [9].

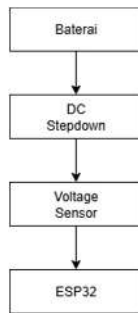


GAMBAR 2
ESP32

III. METODE

1. Desain Sistem Komponen

Dalam pengembangan sistem robot pengantar makanan TIFA, salah satu tantangan utama adalah pemantauan dan pengelolaan daya baterai untuk memastikan robot dapat beroperasi secara berkelanjutan, terutama selama jam-jam sibuk di Tel-U Coffee. Baterai yang digunakan dalam robot memiliki kapasitas tegangan maksimum 53V dan tegangan minimum 49.2V, yang menentukan batas antara status daya penuh (100%) dan kosong (0%). Untuk memastikan sistem dapat beroperasi dengan efisien dan komponen lain dapat berfungsi dengan baik, tegangan dari baterai yang tinggi tersebut perlu diturunkan. Proses ini dilakukan dengan menggunakan DC stepdown converter, yang menurunkan tegangan menjadi 25V, sesuai dengan kebutuhan sistem. Setelah itu, voltage sensor mengukur tegangan ini dan mengonversinya menjadi 5V, yang selanjutnya diteruskan ke mikrokontroler ESP32. Mikrokontroler ini, yang beroperasi pada tegangan 3.3V, mengelola dan memproses data status baterai untuk kemudian diteruskan ke sistem monitoring. Diagram berikut menggambarkan arsitektur sistem pemantauan baterai robot TIFA, yang mencakup interaksi antara komponen utama dalam pengelolaan daya dan pemantauan status baterai secara real-time.



GAMBAR 3
Desain Sistem Komponen

Berdasarkan desain sistem yang telah dibuat, berikut penjelasan rinci mengenai alur dan komponen sistem:

A. Baterai

Baterai memiliki kapasitas maksimum 53V yang setara dengan 100% daya. Ketika tegangan baterai mencapai 49.2V, baterai dianggap mencapai 0% daya. Sistem ini menggunakan rentang tegangan ini untuk menghitung dan memantau status baterai selama operasi robot.

B. DCStepdown

Untuk memastikan komponen-komponen lain yang memerlukan tegangan lebih rendah dapat beroperasi dengan aman, tegangan dari baterai yang berada di kisaran 49.2V hingga 53V diturunkan menggunakan DC stepdown converter. DC stepdown ini mengurangi tegangan menjadi 25V agar sesuai dengan kebutuhan daya dari sensor dan mikrokontroler. Dengan cara ini, sistem menghindari kerusakan pada komponen yang lebih sensitif terhadap tegangan lebih tinggi.

C. VoltageSensor

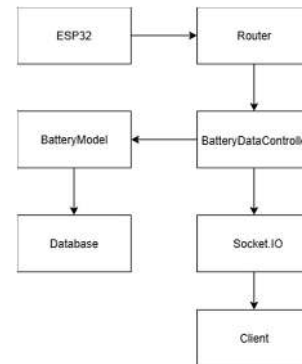
Setelah melalui DC stepdown, tegangan 25V diteruskan ke voltage sensor. Sensor ini bertugas untuk memantau tegangan yang masuk ke sistem dan mengubahnya menjadi data yang dapat diproses oleh mikrokontroler. Pada tahap ini, sensor mengubah tegangan input menjadi 5V, yang merupakan tingkat tegangan yang diperlukan untuk proses lebih lanjut.

D. ESP32

Data tegangan yang sudah diproses oleh voltage sensor kemudian dikirimkan ke ESP32, mikrokontroler yang mengendalikan dan memonitor robot. ESP32 beroperasi pada tegangan 3.3V, sehingga tegangan yang masuk ke mikrokontroler ini telah diturunkan lebih lanjut oleh voltage regulator internal pada ESP32 agar sesuai dengan kebutuhan operasionalnya.

2. Desain Sistem Alur Backend

Untuk mendukung komunikasi antara perangkat keras dan sistem monitoring berbasis web secara efisien, dirancanglah arsitektur *backend* yang mengadopsi prinsip Model-View-Controller (MVC) serta integrasi komunikasi real-time menggunakan Socket.IO[9], [10]. Diagram berikut menggambarkan alur data dari ESP32 sebagai pengirim data tegangan baterai, hingga data tersebut disimpan ke dalam database serta didistribusikan ke client secara real-time.



GAMBAR 4
Desain Sistem Alur Data

A. ESP32

ESP32 berperan sebagai *edge device* yang melakukan pembacaan nilai tegangan baterai melalui sensor, kemudian mengonversi nilai tersebut menjadi persentase kapasitas dengan pendekatan berbasis *ADC scaling*. Data yang telah dikalkulasi dikirimkan dalam bentuk permintaan HTTP POST ke endpoint REST API *backend*, yakni `/api/battery-data`. Permintaan ini membawa payload yang memuat informasi seperti serial, name, persentase, dan timestamp, yang dikemas dalam format JSON[5].

B. Router

Lapisan routing pada sistem diimplementasikan dalam berkas `routes/battery.route.php`, yang bertugas untuk meneruskan setiap permintaan masuk menuju controller yang relevan. Dalam konteks ini, request POST yang diterima akan diarahkan ke method `store()` pada kelas `BatteryDataController`. Mekanisme ini mengikuti prinsip pemisahan tanggung jawab dalam arsitektur MVC, sehingga routing tidak menangani logika bisnis secara langsung.

C. BatteryDataController

Controller bertindak sebagai pusat pemrosesan logika aplikasi. Pada method `store()`, data yang dikirim dari ESP32 akan dibaca dari `php://input`, kemudian di-decode dan divalidasi. Setelah validasi berhasil, controller akan meneruskan data ke `BatteryModel` untuk disimpan ke dalam database menggunakan perintah SQL terstruktur. Selain itu, controller juga menginisiasi proses pemancaran data melalui instance `Socket.IO` dengan cara men-trigger event `batteryUpdate` yang memuat payload persentase baterai terbaru[10]. Proses ini memungkinkan client yang terhubung menerima data baru secara real-time tanpa intervensi pengguna.

D. BatteryModel

Model bertanggung jawab sebagai abstraksi lapisan data yang menghubungkan controller dengan sistem manajemen basis data. Pada sistem ini, model menggunakan pendekatan PHP Data Objects (PDO) untuk melakukan eksekusi query parameterized yang aman terhadap injeksi SQL[6].

E. Database

Basis data MySQL digunakan untuk menyimpan data historis baterai secara terstruktur. Tabel `battery_data` mencatat setiap entri data berdasarkan waktu masuk, nilai kapasitas, dan identitas perangkat. Penyimpanan ini menyediakan landasan bagi analisis tren jangka panjang,

pemodelan prediktif, serta sebagai fallback apabila koneksi real-time tidak tersedia.

F. Socket.IO

Socket.IO digunakan sebagai middleware real-time communication yang berjalan dalam instance Node.js terpisah. Setelah controller memproses dan menyimpan data baru, event batteryUpdate dikirim melalui server Socket.IO, yang kemudian diteruskan ke semua client yang sedang terhubung. Hal ini memungkinkan dashboard monitoring untuk menampilkan data baterai terbaru secara instan tanpa perlu melakukan polling berkala.

G. Client

Client merupakan aplikasi web yang terhubung ke *backend* melalui dua jalur: REST API dan WebSocket. Untuk kebutuhan historis, client melakukan request GET terhadap endpoint tertentu untuk menampilkan grafik atau log data. Sedangkan untuk pemantauan instan, client membuka koneksi WebSocket melalui Socket.IO dan mendengarkan event batteryUpdate. Ketika event diterima, antarmuka pengguna langsung diperbarui tanpa intervensi tambahan, sehingga memberikan pengalaman pengguna yang lebih responsif dan efisien.

3. Pengolahan ADC Raw Data ke Persentase Baterai

Dalam sistem pemantauan baterai robot TIFA, proses konversi sinyal analog dari sensor tegangan menjadi data digital yang dapat diproses oleh mikrokontroler memerlukan penggunaan ADC (Analog-to-Digital Converter). Mikrokontroler, seperti ESP32, memiliki kemampuan untuk membaca nilai tegangan baterai melalui sensor menggunakan ADC, yang kemudian diubah menjadi nilai digital. Rumus pertama yang digunakan dalam sistem ini adalah untuk menghitung nilai tegangan berdasarkan nilai yang dibaca oleh ADC (Analog-to-Digital Converter):

$$V_{ADC} = \frac{ADC_{Value} \times V_{Ref}}{ADC_{Max}}$$

Rumus ini digunakan untuk mengonversi nilai ADC yang terukur menjadi tegangan yang sesungguhnya pada sensor. Nilai ADC value dibaca oleh mikrokontroler, dan dengan menggunakan V_{ref} serta ADC_{max} , kita dapat menghitung tegangan yang sebenarnya yang diterima oleh sensor.

$$Divider\ Ratio = \frac{V_{Aktual\ Baterai}}{V_{Masuk\ ESP}}$$

Rasio pembagi tegangan digunakan untuk menghubungkan tegangan yang diterima oleh mikrokontroler dengan tegangan asli yang diukur menggunakan multimeter pada baterai. Rasio ini penting untuk mengetahui berapa besar perbedaan tegangan yang diterima mikrokontroler dibandingkan dengan tegangan aslinya.

$$V_{Baterai} = V_{ADC} \times Divider\ Ratio$$

Setelah menghitung nilai tegangan ADC dan rasio pembagi tegangan, kita dapat menghitung tegangan aktual baterai ($V_{Baterai}$). Dengan mengalikan hasil perhitungan ADC dengan rasio pembagi tegangan, kita mendapatkan nilai tegangan baterai yang sesungguhnya.

$$StepPercent = \frac{100}{voltMax - voltMin} \times (V_{Baterai} - voltMin)$$

Untuk menghitung persentase kapasitas baterai, kita pertama-tama menghitung langkah perbedaan antara $voltMax$ dan

$voltMin$, yang mengindikasikan rentang tegangan yang dipakai untuk mengukur kapasitas baterai (antara 0% dan 100%). Kemudian, kita mengalikan perbedaan tegangan yang terukur ($V_{Baterai}$) dengan langkah tersebut untuk mendapatkan persentase kapasitas baterai.

Dalam sistem pemantauan baterai robot TIFA, perubahan tegangan baterai dapat terjadi secara fluktuatif, terutama ketika beban atau penggunaan robot berubah-ubah. Perubahan tegangan yang tidak stabil atau terlalu cepat dapat menyebabkan pembacaan kapasitas baterai yang tidak akurat dan membingungkan pengguna. Untuk mengatasi masalah ini, perhitungan Step Percent diterapkan. Dengan menggunakan Step Percent, tujuan utamanya adalah untuk menghindari fluktuasi tegangan yang terlalu tajam atau perubahan yang terlalu sering pada persentase kapasitas baterai. Persentase kapasitas baterai hanya akan berubah setiap terjadi perubahan tegangan minimal 0.1V. Dengan cara ini, pembacaan kapasitas baterai menjadi lebih stabil dan terkontrol, memberikan informasi yang lebih akurat dan dapat diandalkan bagi pengguna, serta memungkinkan mereka untuk merencanakan pengisian ulang baterai dengan lebih tepat.

$$step = \frac{V_{Baterai} - V_{Minimum}}{0.1}$$

Langkah ini menghitung perbedaan tegangan yang dapat terjadi per setiap perubahan kecil dalam kapasitas baterai. Ini mengukur seberapa besar perubahan tegangan yang diperlukan untuk mencapai perubahan kapasitas yang signifikan, dalam hal ini 0.1V.

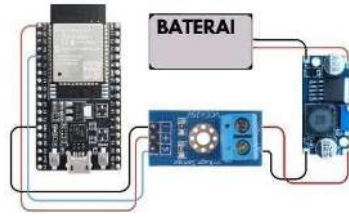
$$Percentage = step \times 2.65\%$$

Berdasarkan perhitungan langkah sebelumnya, kita mengalikan hasilnya dengan faktor 2.65% untuk mendapatkan persentase kapasitas baterai. Faktor ini merupakan konversi dari langkah yang dihitung dalam langkah sebelumnya (step) menjadi persentase kapasitas baterai yang bisa dibaca oleh pengguna.

IV. HASIL DAN PEMBAHASAN

Setelah merancang dan mengembangkan sistem pemantauan baterai untuk robot pengantar makanan TIFA, langkah selanjutnya adalah menguji dan mengevaluasi kinerja sistem tersebut. Pada bagian ini, akan dijelaskan mengenai hasil implementasi sistem yang telah dibangun, termasuk integrasi antara komponen perangkat keras dan perangkat lunak yang digunakan. Pengujian dilakukan untuk memastikan bahwa seluruh sistem berfungsi sesuai dengan tujuan yang telah ditetapkan, yaitu memantau tegangan baterai robot secara real-time dan memberikan notifikasi yang tepat saat level daya mencapai ambang batas yang kritis. Hasil dari pengujian ini memberikan gambaran tentang seberapa efektif sistem dalam mendukung keberlanjutan operasional robot tanpa gangguan akibat kehabisan daya, serta memberikan informasi yang lebih stabil dan akurat bagi pengguna.

1. Implementasi Antar Komponen



Gambar 5 Wiring Diagram

Pada implementasi sistem pemantauan baterai untuk robot TIFA, sejumlah komponen elektronik saling terhubung untuk mendukung pengukuran tegangan dan pengelolaan daya. Sistem ini menggunakan ESP32 sebagai mikrokontroler utama untuk mengendalikan proses pemantauan dan komunikasi data. Untuk mendeteksi tegangan baterai, sistem ini menggunakan voltage sensor, yang terhubung ke pin 34 pada ESP32 untuk menerima input analog. Komponen voltage sensor dihubungkan ke pin 3.3V pada ESP32 untuk memberi daya pada sensor, sedangkan pin GND pada sensor dihubungkan dengan pin GND ESP32 untuk memastikan jalur ground yang terhubung. Tegangan yang diukur oleh sensor, yang berkaitan dengan tegangan baterai, diteruskan ke input DC stepdown converter untuk menurunkan tegangan yang diterima ke level yang lebih aman, agar sesuai dengan kebutuhan sensor dan mikrokontroler. Output dari DC stepdown ini mengalir ke sensor dan ESP32, dengan memastikan tegangan yang stabil dan terkontrol. Seluruh sistem diberi daya oleh baterai, yang terhubung langsung ke input DC stepdown converter, baik pada sisi positif (+) maupun negatif (-).

2. Pengujian

Dalam rangka melakukan evaluasi kinerja sistem pemantauan baterai, tabel berikut menyajikan hasil pengujian yang menunjukkan hubungan antara persentase kapasitas baterai, nilai tegangan (V Battery), nilai ADC yang terukur (Raw ADC), dan tegangan yang diterima oleh ESP32 (V ESP32). Setiap penurunan tegangan dicatat pada interval 0.2V, dengan setiap penurunan tegangan tersebut menghasilkan pengurangan persentase kapasitas baterai sebesar 5.3%. Tabel ini akan memberikan gambaran yang lebih jelas tentang bagaimana kapasitas baterai berkurang seiring waktu dan bagaimana nilai ADC serta tegangan ESP32 berubah sesuai dengan penurunan tersebut.

TABEL 1
Pengujian Sistem Monitoring Baterai

Persentase (%)	V Battery (V)	Raw ADC (12-bit)	V ESP32 (V)
100.0	53.0	4109	>3.30
95	52.8	4087	3.29
89	52.6	4065	3.26
84	52.4	4042	3.23
79	52.2	4020	3.19
73	52.0	3998	3.16
68	51.8	3975	3.13
63	51.6	3953	3.10
58	51.4	3931	3.08
52	51.2	3908	3.05

47	51.0	3886	3.02
42	50.8	3864	2.99
36	50.6	3841	2.97
31	50.4	3819	2.94
26	50.2	3797	2.92
20	50.0	3774	2.89
15	49.8	3752	2.87
10	49.6	3730	2.85
5	49.4	3707	2.82
0	49.2	3685	<2.80

V. KESIMPULAN

Penelitian ini telah berhasil merancang dan mengimplementasikan sistem monitoring baterai berbasis Internet of Things (IoT) pada robot pengantar makanan Tel-U Interactive Food Assistant (TIFA). Sistem ini mengintegrasikan ESP32 sebagai mikrokontroler, sensor tegangan sebagai pengukur level daya, protokol Socket.IO untuk komunikasi data real-time, serta MySQL sebagai penyimpanan data historis. Melalui pendekatan ini, pengguna dapat memantau status baterai secara langsung melalui aplikasi Android dan platform web.

Hasil pengujian menunjukkan bahwa sistem mampu memberikan informasi tegangan dan kapasitas baterai dengan akurasi yang memadai. Implementasi metode perhitungan kapasitas baterai berbasis step voltage sebesar 0.2V, yang dikonversikan menjadi penurunan kapasitas sebesar 5.3%, terbukti efektif dalam menghindari fluktuasi data yang tidak perlu, sehingga menghasilkan pembacaan yang stabil dan representatif. Selain itu, sistem juga memberikan notifikasi ketika tegangan baterai mencapai batas kritis (49.2V), memungkinkan tindakan preventif dilakukan secara tepat waktu.

Dengan demikian, sistem yang dikembangkan mampu meningkatkan efisiensi operasional robot, meminimalisasi downtime akibat kehabisan daya, serta memberikan pengalaman pengguna yang lebih baik di lingkungan layanan intensif seperti Tel-U Coffee. Sistem ini juga dapat dijadikan dasar untuk pengembangan lebih lanjut pada manajemen daya robotik berbasis IoT dalam konteks industri pelayanan lainnya.

Pengembangan sistem monitoring baterai pada robot TIFA di masa mendatang dapat diarahkan pada peningkatan akurasi dan kecerdasan dalam pemantauan performa baterai melalui integrasi metode *State of Charge* (SoC), yaitu estimasi kapasitas energi yang tersisa di dalam baterai yang dinyatakan dalam bentuk persentase. Berbeda dengan metode pengukuran berbasis tegangan yang digunakan saat ini, pendekatan SoC mampu memberikan estimasi yang lebih stabil dan akurat, terutama ketika baterai sedang dalam kondisi terpakai atau sedang diisi daya[2],[6]. Salah satu metode yang umum digunakan untuk menghitung SoC adalah *Coulomb Counting*, yaitu dengan mengakumulasi jumlah arus listrik yang keluar dan masuk dari baterai seiring waktu. Dengan mengadopsi metode ini, sistem diharapkan dapat tidak hanya memantau sisa kapasitas baterai secara lebih presisi, tetapi juga memprediksi estimasi waktu pengisian ulang secara real-time[7],[8]. Pengembangan ini akan meningkatkan efisiensi operasional robot, terutama dalam

perencanaan pengisian daya tanpa mengganggu alur pelayanan, serta menjadi dasar untuk sistem manajemen baterai cerdas berbasis Internet of Things (IoT) di masa depan.

REFERENSI

1. **F. A. A. Fernandes et al.**, "Development of a Real-Time Battery Monitoring System for Mobile Robots Using IoT", *Sensors*, vol. 20, no. 17, 2020. [DOI: 10.3390/s20174795]
2. **L. Yu et al.**, "Battery Management System for Electric Vehicles – Modeling, State Estimation and Balancing", *Energies*, vol. 11, no. 1, 2018. [DOI: 10.3390/en11010024]
3. **M. Al-Awami et al.**, "Low-Cost Wireless Monitoring System for Lead-Acid Batteries Using IoT Technology", *IEEE Access*, vol. 7, pp. 27659–27670, 2019. [DOI: 10.1109/ACCESS.2019.2901215]
4. **J. Wang et al.**, "A Review of Battery Management System: Topology, Function, and Communication Technology", *Renewable and Sustainable Energy Reviews*, vol. 91, pp. 1–14, 2018.
5. **M. Chen, Y. Hao, Y. Li, C. F. Lai, and D. Wu.**, "On the computation offloading at ad hoc cloudlet: Architecture and service models," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 18–24, 2015.
6. **G. Liu et al.**, "A Review of State of Charge Estimation Techniques for Lithium-Ion Batteries in Electric Vehicles," *Journal of Power Sources*, vol. 348, pp. 39–49, 2017.
7. **R. B. R. Gonçalves et al.**, "Battery Monitoring System Using Internet of Things Devices," *IEEE Latin America Transactions*, vol. 18, no. 5, pp. 831–837, 2020.
8. **P. Thulasiraman and M. Dhanasekaran.**, "Design and Implementation of Real Time Monitoring System for Industrial Battery Using IoT," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 1, pp. 942–949, 2020.
9. **K. A. Kermani and R. Lotfi.**, "MVC Architecture for IoT Web-Based Applications," *Journal of Web Engineering*, vol. 20, no. 7, pp. 2101–2115, 2021.
10. **T. D. Nguyen et al.**, "Efficient WebSocket Communication in IoT Networks for Real-Time Monitoring," *Sensors*, vol. 21, no. 4, pp. 1341, 2021.