

P-27

RANCANG BANGUN WIRELESS BATTERY MONITORING SYSTEM BERBASIS ESP32

DESIGN AND CONSTRUCTION OF WIRELESS BATTERY MONITORING SYSTEM BASED ON ESP32

**Hilmansyah, S.T., M.T.^{1*}, Ir. Restu Mukti Utomo, M.T.², Angga Wahyu Saputra, S.T., M.T.³,
Reva Fauzan Alif⁴**

^{1,2,3,4} Jurusan Teknik Elektro, Politeknik Negeri Balikpapan, Jl. Soekarno-Hatta Km. 8, Balikpapan.

*E-mail: hilmansyah@poltekba.ac.id

Diterima 03-10-2020	Diperbaiki 06-11-2020	Disetujui 7-12-2020
---------------------	-----------------------	---------------------

ABSTRAK

Perkembangan teknologi informasi yang semakin maju memberi banyak dampak pada sektor-sektor kehidupan manusia, salah satunya adalah kemajuan pesat pada sektor industri yang telah memasuki era industri 4.0. Kemudahan akses dan proses menjadi pilar utama dalam industri di era modern saat ini, dengan berkembangnya gadget-gadget canggih memberikan kemudahan bagi manusia untuk dapat mengolah informasi, mengambil dan mengontrol data, serta kegiatan yang serba digital lainnya yang membuat kegiatan industri bisa dilakukan secara mobile dan praktis. Dalam kegiatan industri monitoring diperlukan konsep untuk mempermudah pembacaan data pada suatu objek industri. Konsep yang digunakan untuk sistem monitoring yang dibangun dan diterapkan untuk penggunaan baterai. Penggunaan sensor-sensor yang diantaranya ada sensor tegangan, sensor arus ACS758 dan sensor suhu RTD PT100 untuk membaca data-data yang terdapat pada baterai dan akan diolah oleh mikrokontroler ESP32, mengubah data-data tersebut menjadi nilai yang bisa dibaca oleh manusia dan kemudian akan dikirimkan ke media penyimpanan database menggunakan layanan internet. Kelebihan ESP32 yang dapat mengolah data-data menggunakan pemrograman dan dengan teknologi IoT (modul wifi dan bluetooth) yang tersemat padanya membuat mikrokontroler ini menjadi salah satu komponen IoT yang mendukung industri masa kini. Penggunaan layanan Google Firebase dipilih guna membuat proses kerja pengiriman data ini menjadi lebih praktis dengan fitur-fitur pendukung yang dimilikinya. Mit App Inventor merupakan software pendukung penampilan data pada sistem ini, yang dimana data akan diolah menjadi sebuah bentuk tampilan aplikasi tatap muka yang menampilkan data-data dari firebase serta add-ons dari ThingSpeak, sehingga hasil akhirnya data akan sangat mudah untuk dikolektif.

Kata kunci: Baterai, Internet of Things, ESP32DevKit, MIT App Inventor, ThingSpeak.

ABSTRACT

The development of increasingly advanced information technology has had many impacts on sectors of human life, one of which is the rapid progress in the industrial sector which has entered the industrial era 4.0. Ease of access and process are the main pillars in the industry in today's modern era, with the development of sophisticated gadgets making it easy for humans to be able to process information, retrieve and control data, as well as other all-digital activities that make industrial activities can be carried out via mobile and practical. In industrial monitoring activities, a concept is needed to facilitate reading data on an industrial object. The concept used for a monitoring system that is built and applied to battery usage. The use of sensors including a voltage sensor, ACS758 current sensor and PT100 RTD temperature sensor to read the data contained in the battery and will be processed by the ESP32 microcontroller, converting the data into values that can be read by humans and then sent to the database storage media using internet services. The advantage of ESP32 is that it can process data using programming and with IoT technology (wifi and bluetooth modules) embedded in it, making this microcontroller one of the IoT components that support today's industry. The use of Google Firebase services was chosen to make the work process of sending this data more practical with the supporting features it has. Mit App Inventor is software that supports data display on this system, where the data will be processed into a face-to-face application display that displays data from Firebase and add-ons from ThingSpeak, so that the final data will be very easy to collect.

Keywords: Battery, Internet of Things, ESP32DevKit, MIT App Inventor, ThingSpeak.

PENDAHULUAN

Perkembangan ilmu pengetahuan dan teknologi yang sangat pesat dewasa ini memungkinkan banyaknya perkembangan serta dirancangnya alat-alat yang semakin baik dan efisien untuk membantu mempermudah pekerjaan manusia. Salah satunya terdapat di bidang elektronika dan instrumenisasi, yang biasanya berupa alat-alat sistem elektronik yang terdiri dari sensor-sensor elektronik, pengontrol, dan penampil hasil ukur. Salah satu sistem instrumenisasi yang banyak sekali diterapkan adalah sistem pemantauan baterai (*battery monitoring*). Pemantauan baterai dalam dunia industri dapat dikatakan masih dilakukan secara manual dengan melakukan pengecekan menggunakan alat ukur dengan mendatangi lokasi tempat baterai digunakan. Kondisi ini memungkinkan untuk terjadinya keterlambatan penanganan ketika baterai sedang mengalami gangguan ataupun perlu perbaikan yang dapat menurunkan kuantitas hidup umur baterai.

Teknologi IoT (*Internet of Things*) telah berkembang pesat dan banyak digunakan di berbagai aplikasi untuk mempermudah kebutuhan manusia. IoT merupakan kumpulan dari benda-benda yang dilengkapi dengan sensor-sensor yang sesuai dengan fungsinya dan terhubung melalui jaringan internet dan sering disebut dengan jaringan di dalam jaringan. Sistem IoT berfungsi dengan mengumpulkan data-data yang dihasilkan oleh masing-masing benda yang terhubung ke internet untuk dapat diolah dan dianalisis menjadi informasi yang berguna, sehingga nantinya dapat digunakan untuk mengontrol dan memonitor benda tersebut [1]. Perangkat-perangkat *online* pendukung IoT saat ini, telah banyak bermunculan diantaranya ada Firebase, MIT App Inventor, ThingSpeak, dsb. Firebase merupakan layanan *Backend as a Service* (BaaS) yang saat ini dimiliki oleh Google. Firebase merupakan solusi yang ditawarkan oleh Google untuk mempermudah pengembangan aplikasi *mobile* dengan Dua fitur menarik yang ditawarkan olehnya yaitu *Firebase Remote Config* dan *Firebase Real Time Database*. Selain itu terdapat juga fitur pendukung untuk aplikasi yang memerlukan *push notification* yaitu *Firebase Notification Console*. Ketika data berubah, maka aplikasi yang terhubung dengan Firebase akan meng-*update* secara langsung melalui setiap *device* (perangkat) baik *website* ataupun *mobile* [2]. MIT App Inventor merupakan sebuah platform pengembangan *online* yang dapat digunakan

untuk menyelesaikan masalah pembuatan aplikasi suatu sistem. Platform ini menyediakan editor dengan prinsip "*What You See is What You Get*". Dalam pembuatan aplikasi dengan MIT App, Dikerjakan secara *online* didalam *website* mereka dengan aplikasi yang menjalankan sistem operasi Android dan iOS. Platform ini menggunakan bahasa pemrograman berbasis blok yang dibangun di *Google Blockly* (Fraser, 2013) dan terinspirasi oleh bahasa seperti *StarLogo TNG* (Begel & Klopfer, 2007) dan *Scratch*, mempermudah siapa saja untuk dapat membangun aplikasi ponsel yang dapat memenuhi kebutuhan *project* mereka [3]. ThingSpeak merupakan sebuah layanan internet berbasis *website* yang menyediakan layanan untuk pengaplikasian IoT. Thingspeak merupakan *website* dengan layanan yang berisi aplikasi dan API yang bersifat *open source* untuk menyimpan dan mengambil data dari berbagai perangkat yang menggunakan HTTP (*Hypertext Transfer Protocol*) melalui jaringan internet [4].

Terdapat banyak sekali dari penelitian-penelitian sebelumnya yang merancang sistem untuk monitoring baterai ini. Sistem monitoring yang dibangun dan dirancang dengan menggunakan komponen mikrokontroler dan lcd [5], menggunakan mikrokontroler dan *bluetooth* [6], jua menggunakan protokol *zigbee* dan *point-to-point* dalam pengiriman datanya [7]. Berdasarkan beberapa penelitian sebelumnya, maka diperlukan konsep lebih lanjut untuk sistem monitoring ini, yaitu dengan memanfaatkan konsep IoT. Konsep IoT memberikan beberapa kemudahan bagi pengguna dalam mengakses suatu informasi seperti jumlah rata-rata pengunjung dan *review* suatu tempat wisata [8], dan pemantauan *online* kualitas air waduk [9]. Oleh karenanya, dengan penelitian ini akan dirancang suatu perangkat serta sistem untuk memonitoring tegangan/daya, arus, serta suhu baterai pada mobil listrik secara *real-time* dengan konsep IoT agar perolehan informasi menjadi lebih efektif dan efisien.

METODOLOGI

Metode yang digunakan untuk merancang sistem ini meliputi proses perencanaan, perancangan, implementasi, pengujian alat, pengambilan data, dan analisis. Metode penelitian jenis ini sangat sering digunakan sebagai sistem metodologi dalam bidang penelitian teknik. Alat yang dibangun memiliki fungsi untuk memantau dan

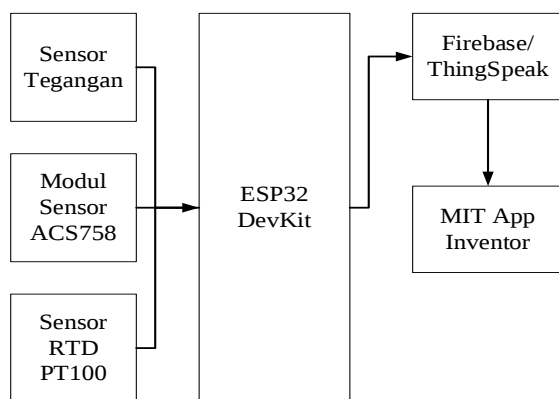
memberikan informasi penggunaan baterai pada mobil listrik dengan parameter daya tegangan pada baterai, arus pada baterai, serta suhu pada baterai berbasis *database* dan tampilan aplikasi *smartphone*.

Penggunaan resistor kotak sebagai sensor pembagi tegangan, modul sensor ACS758 sebagai pengukur arus dan sensor RTD PT100 sebagai pengukur suhu dikategorikan sebagai *input* pada sistem untuk memberikan data pembacaan sensor. Sementara itu, penggunaan Firebase, ThingSpeak dan MIT App Inventor dikategorikan sebagai *output* untuk menampilkan data pembacaan dari sensor-sensor secara *real time*. Desain rancangan untuk sistem ini dapat dilihat pada Gambar 1. Berdasarkan pada gambar, dapat dilihat bahwa sensor tegangan, modul sensor arus ACS758, dan sensor suhu RTD PT100 digunakan sebagai *input*. Untuk proses kontrol, menggunakan *board* ESP32 DevKit. Data-data yang diperoleh dari pembacaan sensor-sensor tersebut akan dikirimkan ke Firebase dan ThingSpeak menggunakan modul WiFi yang tersemat pada *board* ESP32 DevKit.

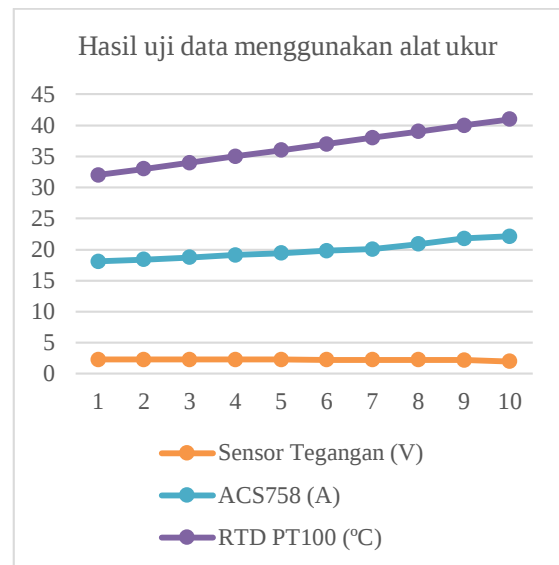
HASIL DAN PEMBAHASAN

A. Pengujian Sensor

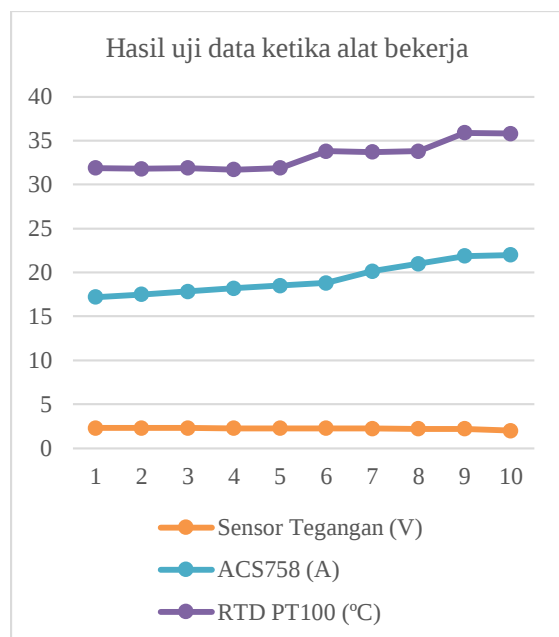
Pengujian bertujuan untuk mengetahui apakah sensor-sensor yang telah terpasang bekerja dengan sangat baik. Dilakukan dengan cara mengukur nilai keluaran sensor ketika alat sedang bekerja, yaitu ketika kondisi motor berubah-ubah kondisinya. Keluaran nilai dari sensor-sensor tersebut diukur dengan menggunakan alat ukur yang diantaranya ada multimeter dan termometer. Untuk data keluaran hasil ukur menggunakan alat ukur, dapat dilihat pada Gambar 2. Dan untuk data keluaran ketika alat sedang bekerja dapat dilihat pada Gambar 3.



Gambar 1. Blok Diagram perancangan alat



Gambar 2. Hasil uji data menggunakan alat ukur

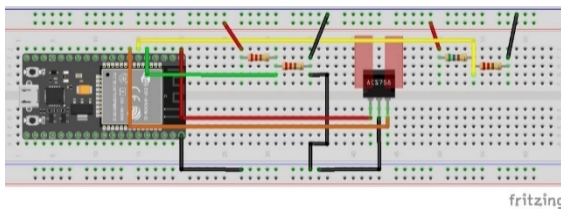


Gambar 3. Hasil uji data ketika alat bekerja

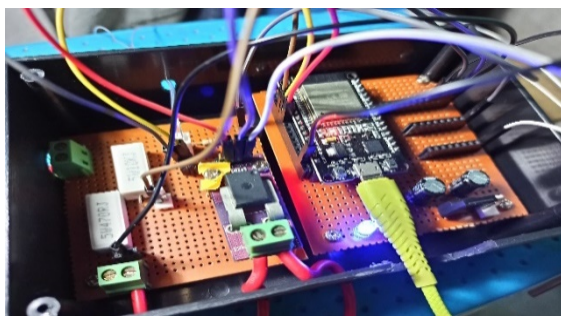
Pengujian dilakukan dengan maksud sebagai pembandingan atau menjadi standar nilai keluaran dari sensor agar mencapai nilai keluaran alat yang diinginkan. Pada data hasil uji rangkaian tegangan, nilai tegangan yang dihasilkan sebesar di bawah kurang dari 5V yang dimana hasil nilai ini merupakan hasil konversi dari nilai tegangan pada baterai yang sebenarnya dengan nilai tegangan sebesar 51V, pengkonversian dilakukan dengan menggunakan rumus $V_{out} = V_{in} \times (R1 / (R1+R2))$. Dalam pengujian ini, sensor-sensor yang telah terhubung dan terintegrasi satu sama lain, dengan masing-masing keluaran nilai data masuk kedalam *board* ESP32 DevKit yang kemudian, data-data tersebut diolah

menggunakan program yang sesuai dengan parameter-parameter ukurnya sehingga menghasilkan nilai keluaran yang sesuai dengan nilai yang didapat menggunakan alat ukur. Gambar 4 dan Gambar 5 menunjukkan alur rangkaiannya.

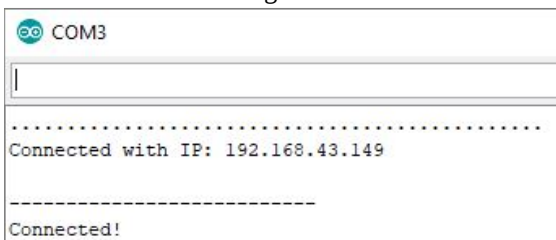
Untuk pemrograman pada *board* ESP32 DevKit, dapat dilakukan dengan bantuan *software* Arduino IDE yang diinstal pada komputer dengan menggunakan kabel USB sebagai media komunikasinya. Untuk penyesuaian program-program dengan desain alat yang dirancang, dilakukan dengan menggunakan *library-library* yang mendukung dan sesuai dengan sistem kinerja alat. Salah satu *library* yang digunakan adalah *library* WiFi, yang membantu komunikasi antara *board* ESP32 DevKit dengan Akses Poin yang digunakan. Gambar 6 adalah ketika *board* telah terkoneksi dengan Akses Poin.



Gambar 4. Skematik rangkaian sensor-sensor pada alat



Gambar 5. Rangkaian sistem alat



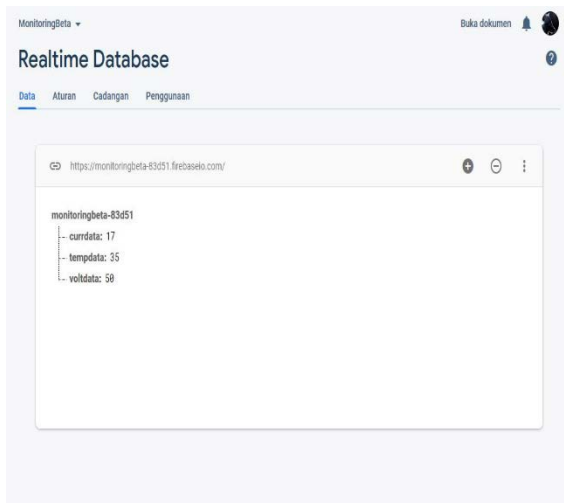
Gambar 6. Tampilan ketika board telah terhubung dengan Akses Poin

B. Pengujian Pengiriman Data ke Firebase dan ThingSpeak

Dengan menggunakan fitur *real time database* dari Firebase serta tampilan *real time channel* dari ThingSpeak yang telah dibuat, dapat dilihat pada Gambar 7 dan Gambar 8, terlihat bahwa data tertampil dengan baik dan sesuai dengan parameter yang telah ditentukan. Pengujian *cloud system* yang dimiliki oleh Firebase dan ThingSpeak dilakukan dengan cara menghubungkan alat ke jaringan internet dengan bantuan Akses Poin, sehingga membuat data-data sensor dapat dipantau jarak jauh. Pengujian *cloud system* ini juga bertujuan untuk meninjau pembacaan data-data pada sensor terhadap objek uji coba. Data-data yang telah terkirim sebelumnya, akan disimpan pada *database* Firebase dan *webserver* ThingSpeak, yang selanjutnya data-data tersebut akan diambil oleh aplikasi Android melalui koneksi internet. Dengan demikian, hasil akhirnya adalah data pada aplikasi Android sama dengan data yang terdapat pada Firebase dan ThingSpeak.

C. Pengujian Aplikasi dari MIT App Inventor

Dalam pembuatan aplikasi yang telah dibangun menggunakan MIT App Inventor, perlu disematkannya beberapa blok program guna sesuai dengan kebutuhan sistem yang telah ditentukan, salah satunya blok sistem untuk komunikasi antara Firebase dan ThingSpeak ke aplikasi yang dibangun. Hasil dari aplikasi yang telah dibangun pada penelitian ini berfungsi sebagai media untuk memonitor data yang ditampilkan dalam bentuk angka dan juga grafik. Tujuan dilakukannya pengujian ini adalah untuk mengetahui kondisi terakhir dari objek uji coba dengan menampilkan data secara *real time* ke smartphone Android pengguna. Membuat pengguna dapat memantau riwayat penggunaan baterai dari segi penggunaan daya, arus dan suhu.



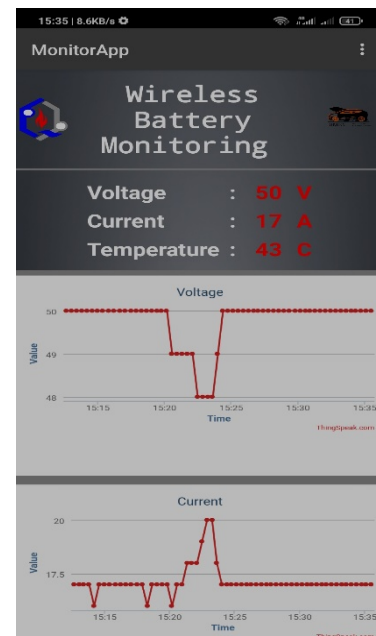
Gambar 7. Tampilan data secara *real time* di Firebase



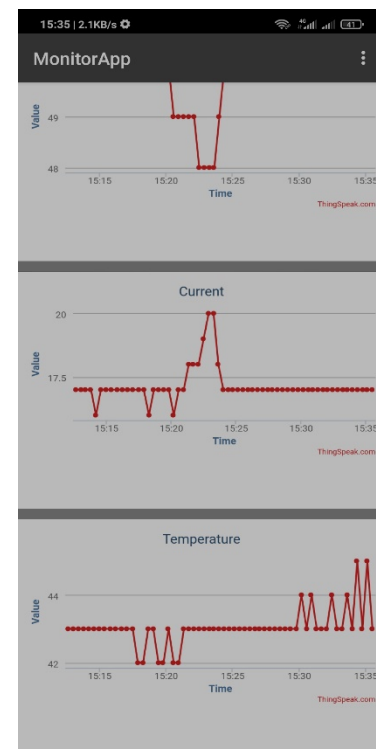
Gambar 8. Tampilan grafik secara *real time* di ThingSpeak

Pada Gambar 9 merupakan tampilan dari desain aplikasi Android yang dibangun menggunakan *software* MIT App Inventor. Pada Gambar 9 terlihat bahwa data-data yang berupa angka pada tampilan aplikasi, merupakan data-data yang diambil oleh aplikasi dari *database* yang terdapat di Firebase melalui jaringan internet, dengan bantuan blok program komunikasi antara aplikasi dan Firebase membuat data yang diperoleh sesuai dengan data yang terdapat pada *database* Firebase.

Pada aplikasi Android sistem ini juga terdapat tampilan grafik yang ditunjukkan Gambar 10 dari data-data sensor yang diambil dari *channel* yang terdapat pada *webservice* ThingSpeak. Tampilan grafik yang diambil oleh aplikasi Android sinkron dengan tampilan grafik yang terdapat pada *channel webservice* ThingSpeak sesuai dengan yang telah diprogram.



Gambar 9. Tampilan data & grafik aplikasi



Gambar 10. Tampilan grafik aplikasi

KESIMPULAN

Firebase dan ThingSpeak pada sistem monitoring ini dirancang agar proses pemantauan sumber daya baterai dapat dilakukan kapan saja dan dimana saja melalui *smartphone* yang terhubung ke jaringan internet. Alat ini didesain dan dirancang menggunakan *board* mikrokontroler ESP32 DevKit yang dapat mengirimkan data melalui jaringan internet menggunakan modul WiFi yang terhubung dengan sebuah Akses Poin.

Penggunaan Firebase dan ThingSpeak sebagai tempat penyimpanan data dan juga aplikasi dari MIT App Inventor sebagai *interface* data pada *smartphone* Android.

SARAN

Masih terdapat banyak hal-hal yang perlu dikembangkan dari penelitian sistem ini, sehingga diharapkan agar penelitian selanjutnya dapat mengembangkan sistem pembaruan yang dapat mengontrol penggunaan daripada baterai agar sistem dapat menjadi lebih sempurna dan membuat hasil olah data menjadi lebih efisien daripada sebelumnya.

UCAPAN TERIMA KASIH

Terima kasih peneliti ucapkan sebesar-besarnya kepada Tuhan Yang Maha Esa atas karunia-Nya yang membuat peneliti mampu menyelesaikan penelitian ini secara baik, serta ucapan terima kasih kepada kerabat-kerabat dekat yang senantiasa memberikan dukungan selama proses pengerjaan penelitian ini.

DAFTAR PUSTAKA

- [1] G. C. H., "Internet of Things," vol. 06, no. 3, pp. 35–41.
- [2] C. Khawas, "Application of Firebase in Android App Development-A Study," no. June, (2018).
- [3] S. Kong and H. Abelson, *Computational Thinking Education*.
- [4] M. Chwalisz, "ThingSpeak Documentation," *SL Thingspeak*, 2016.
- [5] B. S. Putra, A. Rusdinar, and E. Kurniawan, "Desain Dan Implementasi Sistem Monitoring Dan Manajemen Baterai Mobil Listrik," *eProceedings Eng.*, vol. 2, no. 2, (2015).
- [6] A. Jamaluddin, F. A. Perdana, A. Supriyanto, A. Purwanto, and M. Nizam, "Development of Wireless Battery Monitoring for Electric Vehicle," in *2014 International Conference on Electrical Engineering and Computer Science (ICEECS)*, (2014), pp. 147–151.
- [7] A. Rahman, M. Rahman, and M. Rashid, "Wireless Battery Management System of Electric Transport," in *IOP Conference Series: Materials Science and Engineering*, (2017), vol. 260, no. 1, p. 12029.
- [8] G. R. Payara and R. Tanone, "Penerapan Firebase Realtime Database Pada Prototype Aplikasi Pemesanan Makanan Berbasis Android," *J. Tek. Inform. dan Sist. Inf.*, vol. 4, no. 3, 2018.: pp. 397–406,
- [9] S. A. Akbar, D. B. Kalbuadi, and A. Yudhana, "Online Monitoring Kualitas Air Waduk Berbasis Thingspeak," *Transmisi*, vol. 21, no. 4, pp. 109–115.