

LAPORAN TUGAS AKHIR / *CAPSTONE DESIGN*

EM-IOT: Sistem *Monitoring* Baterai dan *Location Tracking* Berbasis IoT pada Motor Listrik



Penyusun:

Tasya Thifali Salsabila (18524060)

Haikal Faiz Ramadhan (18524064)

Program Studi Teknik Elektro

Fakultas Teknologi Industri

Universitas Islam Indonesia

Yogyakarta

2022

HALAMAN PENGESAHAN

EM-IOT : Sistem *Monitoring* Baterai dan *Location Tracking* Berbasis IoT pada Motor Listrik

Penyusun:

Tasya Thifali Salsabila (18524060)

Haikal Faiz Ramadhan (18524064)

Yogyakarta, 21 Juli 2022

Dosen Pembimbing 1

Dosen Pembimbing 2



Dwi Ana Ratna Wati, S.T, M.Eng.

Iftitah Imawati, S.T, M.Eng.

035240102

215241301

Program Studi Teknik Elektro

Fakultas Teknologi Industri

Universitas Islam Indonesia

Yogyakarta

2022

LEMBAR PENGESAHAN TUGAS AKHIR

EM-IOT : Sistem *Monitoring* Baterai dan *Location Tracking* Berbasis IoT pada Motor

Listrik

ISLAM

Disusun oleh:

Tasya Thifali Salsabila 18524060

Haikal Faiz Ramadhan 18524064

Telah dipertahankan di depan dewan penguji

pada tanggal: 9 Agustus 2022

Susunan Dewan Penguji

Ketua Penguji

: Dwi Ana Ratna Wati, S.T., M.Eng.

Anggota Penguji 1

: Elvira Sukma Wahyuni, S.Pd., M.Eng.

Anggota Penguji 2

: Binar Perdana, S.T.

Tugas akhir ini telah disahkan sebagai salah satu persyaratan

untuk memperoleh gelar Sarjana Teknik

Tanggal: 12 Agustus 2022

Ketua Program Studi Teknik Elektro



Amrullah, S.T., M.Eng., Ph.D.

045240101

PERNYATAAN

Dengan ini kami menyatakan bahwa:

1. Tugas Akhir ini tidak mengandung karya yang diajukan untuk memperoleh gelar kesarjaan di suatu perguruan tinggi lainnya, dan sepanjang pengetahuan kami juga tidak mengandung karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.
2. Informasi dan materi Tugas Akhir yang terkait hak milik, hak intelektual, paten merupakan milik bersama antaraa tiga pihak, yaitu penulis, dosen pembimbing, dan Universitas Islam Indonesia. Dalam hal ini, penggunaan informasi dan materi Tugas Akhir terkait paten maka akan didiskusikan lebih lanjut untuk mendapatkan persetujuan dari ketiga pihak tersebut di atas.

Yogyakarta, 10 Agustus 2022



Tasya Thifali Satsabila (18524060)



Haikal Faiz Ramadhan (18524064)

DAFTAR ISI

HALAMAN PENGESAHAN	2
DAFTAR ISI	5
RINGKASAN TUGAS AKHIR	6
BAB 1 : Definisi Permasalahan	7
BAB 2 : Observasi	9
BAB 3 : Usulan Perancangan Sistem	11
3.1 Usulan Rancangan Sistem	11
3.2 Metode Uji Coba dan Pengujian Usulan Rancangan Sistem	20
BAB 4 : Hasil Perancangan Sistem	21
4.1 Kesesuaian Usulan dan Hasil Perancangan Sistem	21
4.2 Kesesuaian Perencanaan dalam Manajemen Tim dan Realisasinya	22
4.3 Analisis dan Pembahasan Kesesuaian antara Perencanaan dan Realisasi	22
BAB 5 : Implementasi Sistem dan Analisis	24
5.1 Hasil dan Analisis Implementasi	24
5.1.1 Monitoring Baterai	26
5.1.2 Location Tracking	27
5.1.3 Delay Pengiriman Data	28
5.2 Pengalaman Pengguna	29
5.3 Dampak Implementasi Sistem	30
5.3.1 Teknologi/Inovasi	30
BAB 6 : Kesimpulan dan Saran	31
6.1 Kesimpulan	31
6.2 Saran	31
LAMPIRAN – LAMPIRAN	33

RINGKASAN TUGAS AKHIR

Tercatat 18 unit Stasiun Pengisian Kendaraan Listrik Umum atau SPKLU yang tersebar di 16 titik wilayah Jawa Tengah dan D.I. Yogyakarta. Keterbatasan SPKLU berdampak mengganggu perjalanan. Sebab pengisian baterai membutuhkan waktu yang lama dan ketersediaan stasiun pengisian *fast charging* yang masih terbatas. Oleh karena itu, diperlukannya sistem untuk *monitoring* kondisi baterai dan posisi kendaraan. *Monitoring* kondisi baterai diperlukan untuk mengetahui sisa tegangan dan sisa jarak tempuh. Dan *monitoring* lokasi terkini kendaraan diperlukan agar mempermudah mengetahui lokasi terkini kendaraan. Sehingga, bermanfaat untuk manajemen atau mengatur waktu untuk *charging* kendaraan dan mengetahui lokasi dari kendaraan.

Cara kerja EM-IOT diawali dengan pembacaan tegangan dari baterai motor listrik dengan menggunakan rangkaian pembagi tegangan, dan pembacaan nilai *latitude* dan *longitude* untuk mendapatkan titik lokasi menggunakan sensor GPS NEO 6M. Hasil pembacaan tersebut kemudian diolah menggunakan mikroprosesor NodeMCU ESP-12E sehingga didapatkan hasil keluaran berupa tegangan baterai, persentase baterai, sisa jarak tempuh, dan lokasi terkini kendaraan. Hasil keluaran berbentuk data selanjutnya dikirim menuju *Thingspeak* dengan menggunakan koneksi *Wi-Fi*. Data yang telah tersimpan pada *Thingspeak* akan ditampilkan pada aplikasi Android. Pada aplikasi Android terdapat fitur keamanan menggunakan akun dengan data akun yang terdaftar disimpan pada *Firebase Authentication*.

Pada pembuatan sistem EM-IOT terdapat beberapa hal yang tidak sesuai dengan usulan awal perencanaan sistem. Perubahan pada judul didasari aplikasi desain UI tidak dapat mendukung *APIKey openrouteservice* untuk dilakukannya proses navigasi rute. Selain itu perubahan *hardware* dikarenakan tidak dapat memenuhi spesifikasi yang dibutuhkan. Serta untuk menekan biaya dan ruang penyimpanan alat, desain tempat *hardware* diperkecil.

Hasil pengujian *monitoring* baterai dan *location tracking* pada aplikasi EM-IOT telah dapat menampilkan data dengan rata-rata *error* tidak melebihi 5%. Maka sistem dikatakan bekerja dengan baik memenuhi spesifikasi yang telah ditentukan. Data pada aplikasi diperbarui setiap 15 detik. Waktu tersebut merupakan *delay* pengiriman data pada *Thingspeak*.

Alat dan aplikasi EM-IOT memperhatikan aspek teknologi dengan menerapkan teknologi IoT dan aplikasi ditujukan untuk pengguna Android. Selain itu, alat EM-IOT dilengkapi sistem anti air dengan *rating* IP22.

BAB 1 : Definisi Permasalahan

Jumlah kendaraan bermotor konvensional mengalami peningkatan setiap tahunnya. Berdasarkan data Badan Pusat Statistik, tahun 2018 sebanyak 126.508.776 unit kendaraan bermotor yang ada di Indonesia [1]. Satu tahun setelahnya meningkat menjadi 5,8% sehingga pada tahun 2019 jumlah kendaraan bermotor di Indonesia telah mencapai 133.617.012 unit [1].

Meningkatnya jumlah kendaraan bermotor menyebabkan konsumsi bahan bakar fosil ikut meningkat. Penggunaan bahan bakar fosil memiliki dampak buruk khususnya untuk lingkungan. Gas hasil pembakaran bahan bakar pada kendaraan bermotor menghasilkan emisi yang menyebabkan pencemaran udara [2]. Untuk meminimalisir hal tersebut, kini kendaraan bermotor menggunakan energi ramah lingkungan yaitu listrik sebagai energi penggerak [3].

Indonesia tengah mempersiapkan untuk memasuki era kendaraan listrik dengan diterbitkannya Peraturan Presiden Nomor 55 Tahun 2019 tentang Percepatan Program Kendaraan Bermotor Listrik Berbasis Baterai (*Battery Electric Vehicle/BEV*) untuk Transportasi Jalan [4]. Berdasarkan peraturan tersebut kedepannya pemerintah menargetkan pengembangan kendaraan listrik hingga tahun 2035. Pemerintah menargetkan produksi pada tahun 2030 mencapai kurang lebih 3 juta kendaraan listrik.

Selain produksi kendaraan dibutuhkan fasilitas penunjang untuk kendaraan listrik yaitu Stasiun Pengisian Kendaraan Listrik Umum atau yang disebut dengan SPKLU. Menurut data dari Kementerian ESDM tercatat baru 187 unit SPKLU yang beroperasi dan tersebar di 155 lokasi di seluruh Indonesia yang mayoritas berada di Pulau Jawa per-September 2021 [5]. Dengan 18 unit yang tersebar di 16 titik wilayah Jawa Tengah dan D. I. Yogyakarta.

Kendaraan listrik di Indonesia tidak terlepas dari permasalahan. Salah satu permasalahan tersebut adalah jangkauan jarak yang terbatas dan ketersediaan stasiun pengisian listrik. Umumnya kendaraan listrik khususnya mobil listrik dalam sekali pengisian bahan bakar dapat berjalan sejauh 200 hingga 400 kilometer [6]. Hal tersebut memiliki dampak apabila kendaraan listrik digunakan jarak jauh maka diperlukannya persiapan dan penyesuaian energi tersimpan pada baterai agar tidak mengganggu perjalanan. Sebab pengisian baterai membutuhkan waktu yang lama dan ketersediaan stasiun pengisian hingga pengisian *fast charging* saat ini masih terbatas.

Berdasarkan uraian diatas, maka diperlukannya sistem untuk *monitoring* kondisi baterai dan posisi kendaraan. *Monitoring* kondisi baterai diperlukan untuk mengetahui sisa tegangan dan sisa jarak tempuh. Dan *monitoring* lokasi terkini kendaraan diperlukan agar mempermudah mengetahui lokasi terkini kendaraan. Sistem *monitoring* ini terintegrasi IoT. Aplikasi menjadi

media untuk menampilkan hasil *monitoring* kondisi baterai dan lokasi terkini. Karena menggunakan aplikasi *monitoring* dapat dilakukan jarak jauh.

Pembuatan sistem *monitoring* baterai dan *location tracking* kendaraan listrik terbagi menjadi dua yaitu, *hardware* dan *software* dengan batasan realistis sebagai berikut:

1. *Hardware* dirancang tahan terhadap air.
2. Sistem menggunakan baterai dengan tegangan maksimal sebesar 54 volt dan *threshold* baterai 39 volt.
3. Sistem *location tracking* hanya dapat diakses di ruangan terbuka.

Selain batasan realistis, pembuatan sistem ini terdapat batasan masalah sebagai berikut:

1. Sistem dirancang dengan menggunakan teknologi *wireless* [7].
2. Sistem dapat dipantau hanya melalui *smartphone* khusus Android.

Tujuan dari perancangan alat ini adalah memberikan solusi alternatif untuk *monitoring* serta manajemen pada pemakaian baterai dan lokasi terkini kendaraan listrik yang terintegrasi dengan sistem IoT. Sehingga, *monitoring* tersebut dapat diakses oleh pengguna dari jarak jauh dan membantu mengakses lokasi ketika baterai dalam keadaan habis atau keadaan darurat lainnya. Manfaat dari perancangan alat ini adalah mempermudah untuk mengatur waktu *charging* baterai kendaraan dan mengetahui lokasi kendaraan.

BAB 2 : Observasi

Proses observasi memiliki tujuan untuk memastikan bahwa rancangan sistem *monitoring* baterai dan *location tracking* berbasis IoT yang dibuat sesuai dengan batasan yang telah ditentukan sebelumnya. Untuk mencapai tujuan tersebut, diperlukan pengumpulan informasi dari penelitian-penelitian sejenis yang telah dilakukan sebelumnya. Tabel 2.1 mencakup sumber informasi terkait solusi yang memungkinkan untuk membantu melakukan *monitoring* baterai dan *location tracking* berbasis IoT pada motor listrik.

Tabel 2.1. Kumpulan solusi yang identik dengan proyek tugas akhir

Penulis	Usulan Solusi	Hasil / Evaluasi
Adam Zuhdi Sisyadi (2020)[8]	Alat sistem <i>monitoring</i> untuk membaca nilai tegangan, arus, dan kecepatan pada mobil listrik. Data <i>monitoring</i> akan dikirimkan menuju <i>Graphic User Interface</i> (GUI) <i>Thingspeak</i> menggunakan jaringan internet yang berasal dari hotspot <i>handphone</i> .	Hasil perancangan sistem <i>monitoring</i> data tegangan, arus, dan kecepatan menggunakan <i>interface Thingspeak</i> berbasis IoT memiliki <i>delay</i> pembacaan 15 detik. Akan tetapi, sistem memiliki kekurangan yaitu belum adanya <i>monitoring</i> untuk perhitungan jarak tempuh dan konsumsi daya (Wh).
Fitria Dayanti. (2018) [9]	Sistem <i>charging</i> dan <i>monitoring</i> pada pengisian baterai agar dapat terkontrol dengan baik sesuai dengan tegangan pengisian baterai pada baterai level tegangan 12 volt DC berbasis mikrokontroler ATMega16. <i>Monitoring</i> baterai dilakukan untuk mengetahui nilai tegangan masukan baterai, daya baterai, dan sisa daya baterai.	Hasil perancangan sistem <i>charging</i> dan <i>monitoring</i> dapat bekerja untuk memenuhi tegangan minimum dalam melakukan pengisian. Dengan tegangan pengisian 13,38 volt dan didapatkan nilai daya sebesar 16,11 Watt. Pembacaan tegangan pada sistem ini menggunakan sensor tegangan yang tersusun dari rangkaian pembagi tegangan. Sistem <i>monitoring</i> baterai mampu menyediakan informasi untuk mengetahui nilai tegangan <i>input charge controller</i> , <i>output charge controller</i> , arus pengisian baterai, dan daya baterai. Tegangan pengisian diatur dengan nilai antara 13 volt hingga 14 volt dengan menggunakan <i>switching</i> mosfet yang diatur oleh ATMega16. Sistem ini memiliki kekurangan yaitu tidak adanya data <i>logger</i> untuk menyimpan nilai daya baterai.
M. A. Budiman, dkk. (2020)[10]	Sistem pelacak dan pengendali kendaraan jarak jauh dengan menggunakan teknologi komunikasi satelit GPS untuk mendapatkan nilai koordinat lokasi (<i>latitude</i> dan <i>longitude</i>) dengan sensor GPS NEO dan divisualisasikan menggunakan <i>Google Maps</i> secara <i>real time</i> . Sistem pengendali jarak jauh dengan menggunakan modul NodeMCU V3 yang terhubung pada koneksi internet. Sistem dilengkapi sistem keamanan berupa <i>fingerprint</i> .	Hasil perancangan sistem dapat mengetahui status kondisi baterai kendaraan maupun baterai alat pelacak. Aplikasi dapat mengirimkan perintah untuk mengoperasikan mesin dan klakson kendaraan. Sistem ini memiliki kekurangan yaitu diperlukannya koneksi <i>Wi-Fi</i> sebagai sarana komunikasi alat sehingga pengguna harus menggunakan modem supaya alat dapat berfungsi dan mengirimkan data pada server.

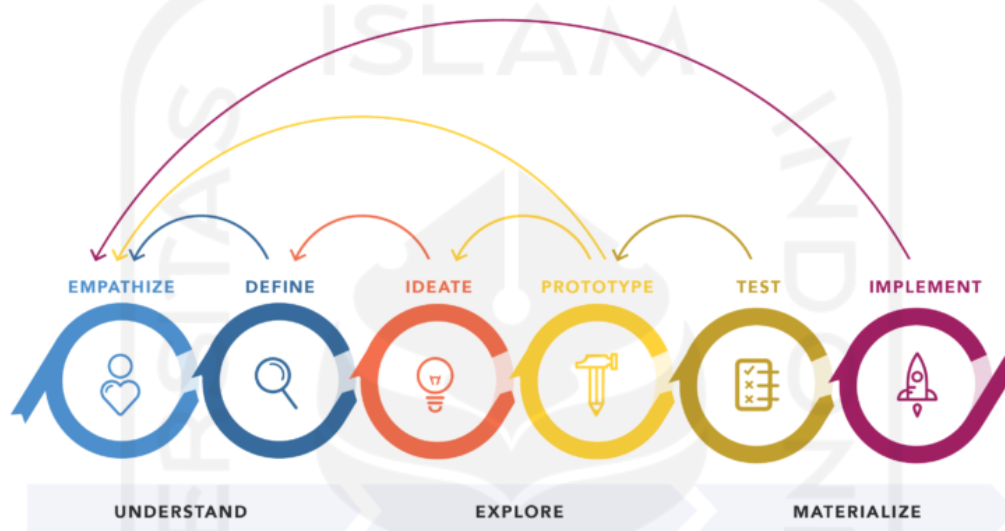
Berdasarkan hasil studi literatur yang termuat pada Tabel 2.1, didapatkan informasi yang dapat dikembangkan menjadi usulan terbaik dari solusi permasalahan yang diangkat, yaitu sistem *monitoring* baterai dan *location tracking* pada motor listrik dengan spesifikasi sebagai berikut:

1. Alat memiliki dimensi berukuran $8,5 \times 10 \times 3,5$ cm.
2. Alat dapat mendeteksi lokasi terkini, sisa tegangan, persentase tegangan, dan sisa jarak tempuh dengan toleransi kesalahan maksimal 5%.
3. Aplikasi dapat diakses melalui *smartphone* khusus Android dengan versi minimum 5.0 – 5.0.2 (API 21) .
4. Sistem komunikasi alat menggunakan protokol *wireless* yaitu *Wi-Fi* untuk mengirimkan data.
5. Pengiriman data dari sensor yang telah terhubung *Wi-Fi* menuju *database* memiliki *delay* maksimal 15 detik.
6. Sumber sistem menggunakan tegangan DC sebesar 5 volt yang bersumber dari baterai motor listrik yang telah di *step down*.
7. Alat memiliki perlindungan dengan *rating* IP22 yaitu kemampuan tahan dari percikan air dan sentuhan jari atau benda serupa.

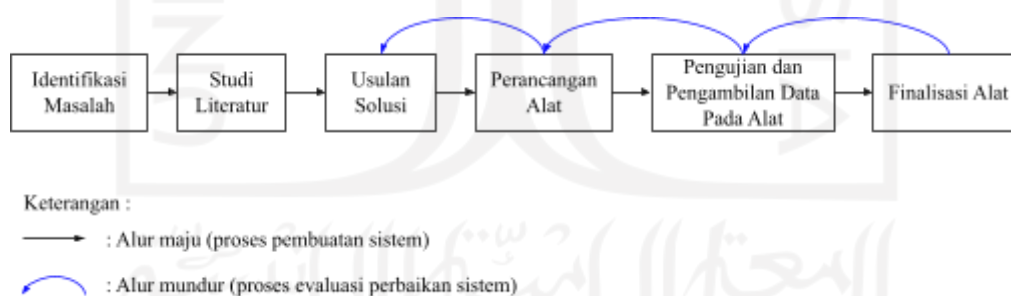
BAB 3 : Usulan Perancangan Sistem

3.1 Usulan Rancangan Sistem

Perancangan sistem dalam *project capstone design* dilakukan berdasarkan metode *design thinking*. Metode *design thinking* memiliki kerangka pemikiran yang terbagi menjadi 3 dasar yaitu, *understand*, *explore*, dan *materialize* yang ditunjukkan pada Gambar 3.1. Dan untuk pembuatan usulan sistem dilakukan implementasi *design thinking* dalam bentuk diagram blok yang terdiri dari proses identifikasi masalah, studi literatur, usulan solusi, perancangan alat, pengujian dan pengambilan data, serta analisa alat yang ditunjukkan pada Gambar 3.2.



Gambar 3.1. Siklus perancangan suatu sistem rekayasa



Gambar 3.2. Diagram blok perancangan sistem

Pada tahapan awal melakukan identifikasi masalah. Diperlukannya informasi mengenai kendaraan listrik di Indonesia. Berdasarkan informasi tersebut, kendaraan listrik di Indonesia sudah mulai berkembang. Akan tetapi, keterbatasan Stasiun Pengisian Kendaraan Listrik Umum (SPKLU) khususnya di wilayah Yogyakarta. Hal tersebut mengakibatkan ruang gerak pengguna kendaraan listrik menjadi terbatas. Dari hal tersebut diangkat permasalahan mengenai pembuatan sistem *monitoring* baterai dan *location tracking* bagi kendaraan listrik.

Tahapan selanjutnya dengan melakukan studi literatur. Studi literatur dilakukan berdasarkan hasil identifikasi permasalahan. Studi literatur memiliki tujuan untuk menentukan komponen dan spesifikasi alat agar mencapai usulan solusi terbaik.

Berdasarkan hasil studi literatur didapatkan beberapa usulan alternatif. Terdapat dua usulan alternatif untuk permasalahan ini, yaitu:

1. *Monitoring* baterai dan estimasi penggunaan daya untuk rute perjalanan berbasis IoT. Protokol komunikasi menggunakan *wireless* yaitu *Wi-Fi*. Hasil produk yang diusulkan berupa *hardware* yaitu alat dan *software* yaitu aplikasi.
2. *Monitoring* baterai dan estimasi penggunaan daya untuk rute perjalanan berbasis IoT. Protokol komunikasi menggunakan *wireless* yaitu *bluetooth*. Hasil produk yang diusulkan berupa *hardware* yaitu alat dan *software* yaitu aplikasi.

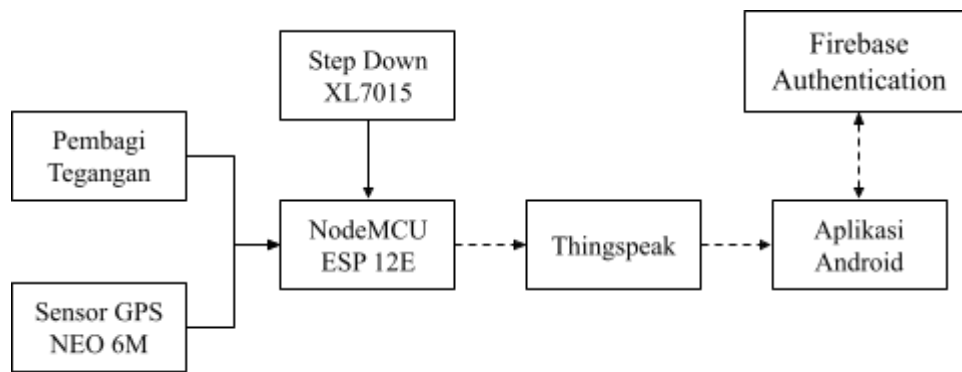
Usulan pertama dipilih sebagai usulan solusi untuk mengatasi permasalahan yang dibahas. Hal tersebut dikarenakan *Wi-Fi* memiliki kecepatan transfer data yang lebih tinggi, jarak transmisi yang lebih jauh, serta *bandwidth* yang lebih besar. Akan tetapi pada proses realisasi terjadi perubahan usulan menjadi *monitoring* baterai dan lokasi terkini dari kendaraan listrik. *Monitoring* baterai dilakukan untuk mengetahui tegangan, persentase tegangan, serta sisa jarak tempuh.

Perancangan alat berdasarkan hasil usulan solusi. Sistem alat dirancang terbagi menjadi dua, yaitu *hardware* dan *software*. *Hardware* maupun *software* dirancang untuk memenuhi spesifikasi yang telah ditentukan. Perancangan sistem juga harus memenuhi standar keteknikan. Standar keteknikan dalam perancangan ini adalah tegangan yang masuk ke dalam NodeMCU ESP 12E tidak melebihi tegangan operasi yaitu 3,3 volt. Maka, sistem dirancang menggunakan rangkaian pembagi tegangan.

Tahapan pengujian dan pengambilan data dilakukan bertujuan memastikan alat telah sesuai dengan spesifikasi dan menyelesaikan permasalahan yang ada. Pengujian dilakukan dengan membandingkan hasil aplikasi dengan parameter pembanding. Hal tersebut dilakukan untuk mengetahui nilai akurasi sistem berdasarkan nilai *error*.

Tahapan akhir yaitu melakukan finalisasi alat. Finalisasi alat dengan melakukan analisa. Analisa dilakukan berdasarkan data hasil pengujian.

Sistem yang dirancang bernama EM-IOT merupakan sistem monitoring baterai dan lokasi terkini kendaraan listrik. Monitoring baterai pada sistem EM-IOT menampilkan tegangan, persentase baterai, dan sisa jarak tempuh. Sistem ini dirancang berbasis IoT. Berikut merupakan diagram blok sistem EM-IOT yang ditunjukkan dengan Gambar 3.3.



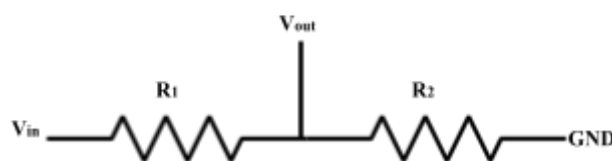
Keterangan :

1. ———> : Wire
2. - - - -> : Wireless
3. <- - - -> : Komunikasi dua arah secara wireless

Gambar 3.3. Diagram blok sistem

EM-IOT memiliki cara kerja yang diawali dengan pembacaan tegangan dari baterai motor listrik dengan menggunakan rangkaian pembagi tegangan, dan pembacaan nilai *latitude* dan *longitude* untuk mendapatkan titik lokasi dengan menggunakan sensor GPS NEO 6M. Dari hasil pembacaan tersebut diolah menggunakan mikroprosesor NodeMCU ESP 12E. Hasil keluaran dari NodeMCU ESP 12E berupa tegangan baterai tersisa, persentase baterai, sisa jarak tempuh, dan lokasi terkini kendaraan. Hasil keluaran tersebut selanjutnya dikirim menuju *Thingspeak* dengan menggunakan koneksi *Wi-Fi*. Data yang telah tersimpan pada *Thingspeak* akan ditampilkan pada aplikasi Android. Pada aplikasi Android terdapat fitur keamanan menggunakan akun. Data akun yang terdaftar disimpan pada *Firebase Authentication*.

Rangkaian pembagi tegangan terdiri dari dua buah resistor seperti pada Gambar 3.4. Rangkaian pembagi tegangan berfungsi untuk menurunkan *input* tegangan dengan rentang 0 hingga 54 volt. Penurunan nilai tegangan *input* dilakukan untuk menyesuaikan tegangan operasi dari mikrokontroler yang digunakan. Hal ini dilakukan agar tidak melebihi tegangan operasi tersebut. Oleh karena itu, dilakukan perhitungan nilai resistor pada pembagi tegangan melalui Persamaan 3.1 dan Persamaan 3.2.



Gambar 3.4. Rangkaian pembagi tegangan

$$V_{out} = \frac{R_2}{R_1 + R_2} \times V_{in} \quad (3.1)$$

Keterangan :

V_{out} = Tegangan keluaran (volt).

V_{in} = Tegangan input (volt).

R_1 = Resistor 1 (ohm).

R_2 = Resistor 2 (ohm).

Maka, untuk mendapatkan nilai R_1 dan R_2 sebagai berikut.

$$\text{Saat } V_{in} = 54 \text{ volt.} \quad (3.2)$$

$$\begin{aligned} V_{out} &= \frac{3300}{56000 + 3300} \times 54 \\ &= 3,00 \text{ volt} \end{aligned}$$

Saat $V_{in} = 0$ volt.

$$\begin{aligned} V_{out} &= \frac{3300}{56000 + 3300} \times 0 \\ &= 0 \text{ volt} \end{aligned}$$

Hasil pembacaan tegangan baterai diolah menjadi persentase. Perhitungan persentase baterai memperhatikan dua parameter, yaitu tegangan maksimal baterai sebesar 54 volt dan *threshold* baterai sebesar 39 volt. Persentase baterai dapat diperoleh dengan Persamaan 3.3.

$$\begin{aligned} persen &= \frac{V - V_{th}}{V_{max} - V_{th}} \times 100 \\ &= \frac{V - 39}{54 - 39} \times 100 \\ &= \frac{V - 39}{15} \times 100 \end{aligned} \quad (3.3)$$

Keterangan :

Persen = Persentase tegangan (%).

V_{th} = Tegangan threshold (volt).

V = Tegangan terbaca (volt).

V_{max} = Tegangan maksimal (volt).

Hasil pembacaan tegangan baterai diolah menjadi sisa jarak tempuh. Perkiraan ini didasari data dari pengujian motor listrik EMPUS yang digunakan sebagai objek uji coba. Motor listrik ini menghabiskan 0,4 volt untuk 1 km perjalanan. Perkiraan sisa jarak aplikasi EM-IOT diperoleh dengan Persamaan 3.4.

$$\begin{aligned} D &= \frac{V - V_{th}}{k} \\ &= \frac{V - 39}{0,4} \end{aligned} \quad (3.4)$$

Keterangan :

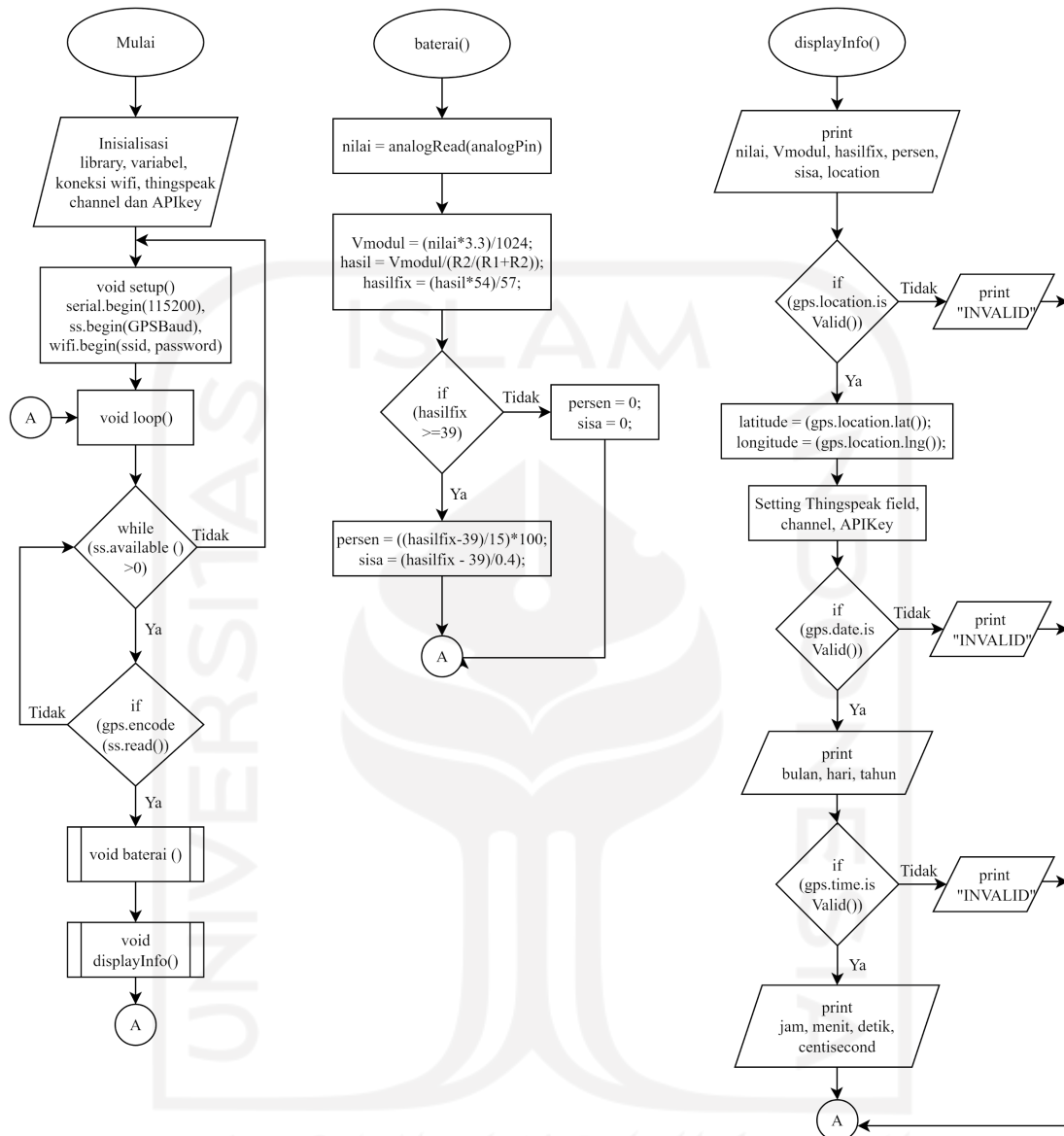
D = Sisa jarak tempuh (km).

V_{th} = Tegangan threshold (volt).

V = Tegangan terbaca (volt).

k = Koefisien hasil pengujian motor listrik EM-PUS (volt/km).

Berdasarkan uraian diatas, sistem EM-IOT memiliki *flowchart* program yang ditunjukkan pada Gambar 3.5. Serta rangkaian skematik *hardware* EM-IOT pada Gambar 3.6. Dan desain kemasan alat EM-IOT ditunjukkan pada Gambar 3.7.



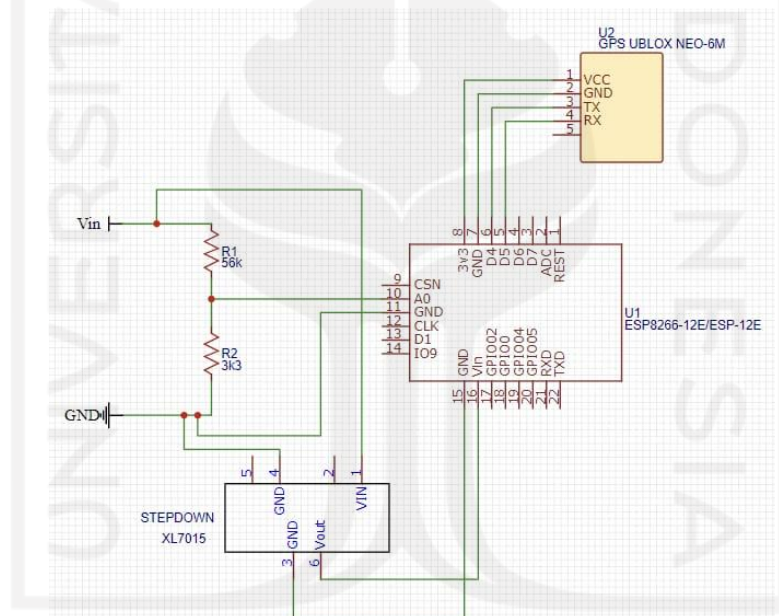
Gambar 3.5. *Flowchart* program

Berdasarkan Gambar 3.5, EM-IOT memiliki alur pemrograman yang diawali dengan inisialisasi *library*, *variable*, koneksi *Wi-Fi*, *thingspeak channel* dan *APIKey*. Kemudian proses *setup* untuk mengatur *baudrate* yang digunakan dan *Wi-Fi*. Proses selanjutnya adalah *void loop*. Apabila GPS aktif maka dilakukan proses *encode* GPS. Jika *encode* GPS terbaca, dilakukan proses perhitungan pada *void* baterai dan proses *display info*.

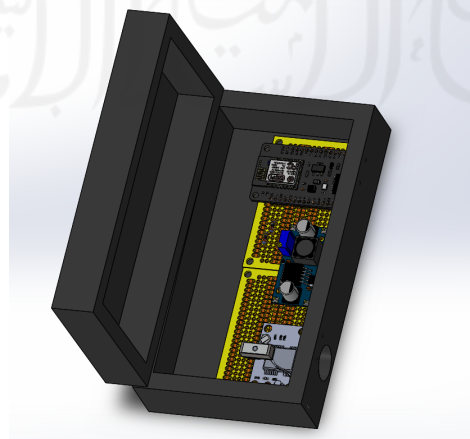
Proses pada *void* baterai dilakukan pembacaan nilai ADC yang terdeteksi oleh mikrokontroler. Nilai ADC tersebut kemudian dikonversi menjadi nilai tegangan yang masuk ke mikrokontroler yaitu *Vmodul*. Untuk menampilkan nilai tegangan sesungguhnya, nilai *Vmodul* akan dibagi dengan resistor pembagi tegangan. Hasil tersebut kemudian dilakukan kalibrasi

dengan menggunakan perbandingan. Perbandingan dilakukan dengan cara nilai hasil dikali dengan tegangan maksimal baterai yaitu 54 volt kemudian dibagi dengan 57 volt yang merupakan tegangan maksimal yang terdeteksi di mikrokontroler. Hasil kalibrasi tersebut dimasukkan dalam nilai *hasilfix*. Apabila *hasilfix* lebih dari sama dengan 39 volt maka dilakukan proses perhitungan persentase dan sisa jarak tempuh. Jika tidak maka persentase dan sisa jarak tempuh akan sama dengan nol. Sebab 39 volt merupakan ambang batas aman atau *threshold* dari baterai.

Proses pada *void displayInfo* adalah menampilkan keluaran dari nilai, *Vmodul*, hasil, *hasilfix*, persen, sisa, dan lokasi. Apabila GPS lokasi *valid* maka nilai *latitude* dan *longitude* terdeteksi. Kemudian, dilakukan pengaturan *field*, *channel*, dan *APIKey* dari *thingspeak*. Apabila GPS tanggal *valid* maka ditampilkan bulan, hari, dan tahun. Dan apabila GPS waktu *valid* maka ditampilkan jam, menit, detik, dan centidetik.



Gambar 3.6. Skematik rangkaian

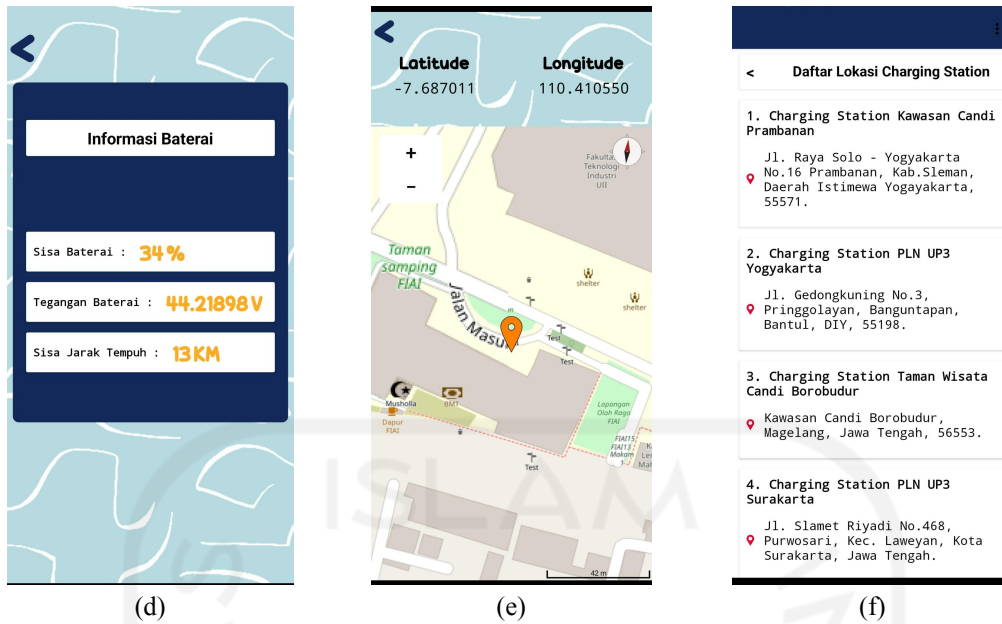


Gambar 3.7. Desain kemasan alat

Berdasarkan Gambar 3.6 merupakan desain skematik rangkaian EM-IOT yang terdiri dari berupa jalur input tegangan fasa dan *ground*, dua buah resistor untuk pembagi tegangan, modul *step down* tegangan, modul GPS, dan NodeMCU ESP12-E. Desain skematik tersebut kemudian dibuat menjadi PCB yang diletakkan pada kemasan alat. Desain kemasan alat yang ditunjukkan pada Gambar 3.7 berukuran $8,5 \times 10 \times 3,5$ cm. Terdapat satu lubang berdiameter 1 cm untuk jalur *input* baterai. Kemasan alat dibuat dengan bahan akrilik 3 mm.

Hasil pembacaan *monitoring* baterai dan *location tracking* dari *hardware* ditampilkan melalui *software*. Tampilan atau *User Interface* yang dirancang berbentuk aplikasi Android. Desain *User Interface* dirancang menggunakan *Kodular*. Aplikasi EM-IOT terdiri dari sistem keamanan dengan pembuatan akun. Terdapat tampilan *home* untuk memilih fitur dan *logout*. Terdapat tiga fitur yaitu, baterai, *maps*, dan daftar SPKLU. Pada fitur baterai menampilkan tegangan baterai, persentase baterai, dan sisa jarak. Terdapat notifikasi apabila baterai akan habis dan baterai dalam keadaan penuh. Pada fitur *maps* menampilkan nilai *latitude* dan *longitude* serta peta lokasi dari kendaraan. Dan fitur daftar SPKLU menampilkan daftar-daftar SPKLU. Daftar tersebut merupakan simulasi lokasi SPKLU. Berikut desain aplikasi yang ditunjukkan pada Gambar 3.8.





Gambar 3.8. Usulan rancangan aplikasi untuk pengguna. (a) Tampilan *sign up* aplikasi, (b) tampilan *login* aplikasi (c) tampilan pilihan fitur, (d) tampilan *monitoring* baterai, (e) tampilan *location tracking*, (f) tampilan daftar SPKLU.

Pada fitur monitoring baterai terdapat notifikasi peringatan. Saat persentase baterai sebesar 50% aplikasi EM-IOT menampilkan notifikasi peringatan baterai akan habis yang ditunjukkan pada Gambar 3.9. Kemudian saat persentase baterai sebesar 34% aplikasi EM-IOT menampilkan notifikasi untuk *charging* (Gambar 3.10). Berdasarkan hasil pengujian dengan menggunakan motor listrik EM-PUS didapati tarikan motor listrik sangat lemah sehingga tidak mampu untuk dilakukan perjalanan. Dan saat persentase baterai menunjukkan 100% aplikasi EM-IOT menampilkan notifikasi bahwa baterai penuh (Gambar 3.11).



Gambar 3.9. Tampilan notifikasi saat persentase baterai 34%



Gambar 3.10. Tampilan notifikasi saat persentase baterai 50%



Gambar 3.11. Tampilan notifikasi saat persentase baterai 100%

Berdasarkan usulan desain dan spesifikasi yang telah ditentukan, maka diperlukan inventarisasi kebutuhan sistem perangkat keras. Tabel 3.1 memperlihatkan kebutuhan sistem sesuai usulan dan spesifikasi yang dibutuhkan.

Tabel 3.1. Inventarisasi kebutuhan usulan sistem alat

No	Nama Alat	Keterangan
1	Perangkat untuk kemasan alat	Dibuat untuk menempatkan rangkaian elektronis yang telah dibuat agar terlindungi dari panas dan tetesan air dengan <i>rating IP22</i> . Memiliki dimensi $8,5 \times 10 \times 3,5$ cm.
2	Mikrokontroler NodeMCU ESP 12E	Memiliki fungsi untuk mengolah data dari rangkaian pembagi tegangan dan sensor GPS sebelum di upload ke <i>cloud server</i> yang digunakan yaitu <i>Thingspeak</i> . NodeMCU ESP 12E dipilih sebab memiliki harga yang relatif murah dengan kemampuan <i>module Wi-Fi</i> .
3	Resistor	Resistor yang dirangkai seri memiliki fungsi sebagai rangkaian pembagi tegangan pada sumber tegangan yang bertujuan untuk menurunkan tegangan agar dapat diolah pada mikrokontroler.

No	Nama Alat	Keterangan
4	Sensor GPS NEO 6M	Memiliki fungsi untuk membaca titik koordinat suatu lokasi dengan menggunakan dua parameter, yaitu <i>latitude</i> (garis lintang) dan <i>longitude</i> (garis bujur). Sensor ini memiliki akurasi posisi mencapai 2 meter dengan dimensi yang tidak terlalu besar yaitu: <i>GPS Board</i> sebesar 22×30×4 mm, dan antena 20×5×5 mm.
5	Modul <i>Step Down</i> XL7015	Memiliki fungsi untuk menurunkan tegangan baterai maksimal 54 volt hingga 5 volt yang digunakan sebagai sumber daya utama alat. Modul <i>Step Down</i> XL7015 dipilih sebab memiliki tegangan operasi maksimum 80 volt dengan tegangan keluaran yang dapat diatur dari 1,25 volt hingga 20 volt.

3.2 Metode Uji Coba dan Pengujian Usulan Rancangan Sistem

Pengujian terbagi menjadi dua bagian yaitu *monitoring* baterai dan *location tracking*. Pengujian sistem dilakukan untuk memastikan sistem dapat bekerja dengan baik. Sistem bekerja dengan baik apabila memenuhi toleransi *error* 5% pada hasil pembacaan tegangan, persentase, sisa jarak, *latitude*, dan *longitude*. Pelaksanaan pengujian dilakukan pada area Kampus Terpadu Universitas Islam Indonesia. Pengambilan data dilakukan saat awal dan akhir setiap rute pengujian.

Pengujian *monitoring* baterai menggunakan baterai dengan tegangan maksimal 54 volt dan *threshold* 39 volt. Pengukuran tegangan dilakukan dengan *voltmeter* dan pembacaan pada aplikasi EM-IOT. Kemudian, dilakukan perhitungan *error* dari hasil tersebut. Data persentase dan sisa tegangan pada aplikasi kemudian dibandingkan berdasarkan perhitungan. Berdasarkan data tersebut, perhitungan *error* juga dilakukan.

Pengujian *location tracking* terbagi menjadi dua, yaitu nilai *latitude* dan *longitude*. *Latitude* dan *longitude* merupakan koordinat letak geografis yang selalu terdapat pada setiap lokasi. *Latitude* merupakan garis bujur yang mana terbentang dari utara hingga selatan, sedangkan *longitude* merupakan garis lintang yang mana terbentang dari barat hingga timur. Pengujian dilakukan dengan melakukan penentuan lokasi menggunakan *Google Maps*. Berdasarkan lokasi tersebut, dilakukan pengambilan data dengan menggunakan aplikasi EM-IOT. Hasil tersebut kemudian dilakukan perhitungan *error*.

BAB 4 : Hasil Perancangan Sistem

4.1 Kesesuaian Usulan dan Hasil Perancangan Sistem

Proses realisasi sistem terdapat perubahan dari rencana perancangan. Perubahan dilakukan sebab alat tidak dapat bekerja dengan baik. Dilakukan perubahan rancangan yang agar sistem dapat bekerja dengan baik. Adapun perbedaan antara rancangan usulan dan realisasi hasil rancangan yang termuat pada Tabel 4.1. Dan realisasi desain alat berdasarkan usulan desain ditunjukkan pada Gambar 4.1.



Gambar 4.1 Desain alat. (a) Usulan desain, (b) Realisasi desain

Tabel 4.1 Perbandingan usulan dan hasil perancangan sistem

No	Spesifikasi	Usulan	Realisasi
1	Mikrokontroler	NodeMCU ESP32	NodeMCU ESP 12E
2	Sensor untuk <i>monitoring</i> baterai	Sensor tegangan DC dan sensor arus ACS712.	Sensor tegangan berupa rangkaian pembagi tegangan.
3	Sensor GPS	NEO 6M.	NEO 6M.
4	Konektivitas	<i>Wi-Fi</i> .	<i>Wi-Fi</i> .
5	Dimensi	15 × 10 × 6 cm .	8,5 × 10 × 3,5 cm.
6	Pembuatan aplikasi	<i>Kodular</i> .	<i>Kodular</i> .
7	Sumber tegangan alat	Baterai motor listrik EM-PUS yang telah di <i>step down</i> menjadi 5V.	Baterai motor listrik EM-PUS yang telah di <i>step down</i> menjadi 5V.
8	Database	<i>Thingspeak</i> dan <i>firebase</i>	<i>Thingspeak</i> dan <i>firebase</i>

4.2 Kesesuaian Perencanaan dalam Manajemen Tim dan Realisasinya

Pengerjaan usulan rancangan sistem EM-IOT terdapat kemunduran waktu dikarenakan pada implementasi dan pengujian sistem terdapat perubahan. Perubahan tersebut meliputi penggantian sistem yang digunakan sehingga memakan waktu pengerjaan menjadi lebih lama dari perkiraan sebelumnya. Hal tersebut juga mempengaruhi Rancangan Anggaran Biaya (RAB) yang telah diusulkan sebelumnya. Maka, Tabel 4.2 dan Tabel 4.3 memuat perbandingan antara usulan waktu dengan realisasi pelaksanaan dan RAB usulan dengan RAB realisasi.

Tabel 4.2 Kesesuaian antara usulan dan realisasi *timeline* pengerjaan Tugas Akhir 2

No	Kegiatan	Usulan waktu	Realisasi Pelaksanaan
1	Pembelian alat dan bahan	Februari	Februari – Mei
2	Perancangan sistem dengan usulan	Maret – April	Maret – Mei
3	Pengujian Sistem	April	Mei – Juni

Tabel 4.3 Kesesuaian RAB Tugas Akhir antara usulan dan realisasi

No	Jenis Pengeluaran	Usulan Biaya		Realisasi Biaya	
		Kuantitas	Total Harga	Kuantitas	Total Harga
1	NodeMCU ESP32	1 pcs	Rp 85.000,-	0 pcs	Rp 0,-
2	Module GPS APM 2.5 NEO-6M	1 pcs	Rp 50.000,-	1 pcs	Rp 65.000,-
3	Sensor ACS712 30A	1 pcs	Rp 25.000,-	0 pcs	Rp 0,-
4	Sensor Tegangan DC 25V	1 pcs	Rp 10.000,-	0 pcs	Rp 0,-
5	Jasa Cetak Case	1 pcs	Rp 350.000,-	1 pcs	Rp 35.000,-
6	NodeMCU ESP 12E	0 pcs	Rp 0,-	1 pcs	Rp 50.000,-
7	Resistor	0 pcs	Rp 0,-	2 pcs	Rp 300,-
8	Module Step Down XL7015	0 pcs	Rp 0,-	1 pcs	Rp 18.000,-
9	Jasa Cetak PCB	1 pcs	Rp 100.000,-	1 pcs	Rp 87.000,-
	Total		Rp 595.000,-		Rp 255.000,-

4.3 Analisis dan Pembahasan Kesesuaian antara Perencanaan dan Realisasi

Berdasarkan hasil perancangan, pembuatan, dan pengujian alat terdapat perbaikan. Perbaikan bertujuan agar sistem yang dibuat tetap sesuai dengan tujuan awal yaitu memberikan solusi alternatif untuk mempermudah pemantauan dan manajemen penggunaan kendaraan listrik.

Perubahan judul disebabkan oleh perubahan rancangan sistem yang semula dapat melakukan estimasi daya suatu rute perjalanan yang akan ditempuh. Hal tersebut tidak dapat direalisasikan karena keterbatasan *Kodular* yang tidak dapat mengakses *APIKey* dari *openrouteservice* untuk dapat menampilkan navigasi rute.

Berdasarkan Tabel 4.1 terdapat perubahan usulan antara lain:

1. Mikrokontroler

Perubahan penggunaan mikrokontroler yang sebelumnya diusulkan menggunakan NodeMCU ESP32 menjadi NodeMCU ESP 12E. Hal ini disebabkan program yang telah dibuat tidak dapat dijalankan dikarenakan kualitas komponen dan fabrikasi, sehingga NodeMCU ESP 12E dipilih sebagai solusi alternatif.

2. Sensor untuk monitoring baterai

Perubahan sensor tegangan menjadi rangkaian pembagi tegangan. Hal ini dikarenakan sensor tegangan hanya dapat membaca tegangan maksimal 25 volt. Maka, rangkaian pembagi tegangan dipilih agar sistem dapat membaca tegangan baterai yang terpasang pada motor listrik dengan kapasitas 54 volt. Dan sensor ACS712 yang merupakan sensor arus tidak digunakan pada realisasi karena perubahan judul yang semula terdapat estimasi daya menjadi tidak ada sehingga tidak diperlukannya sensor arus dalam realisasi.

3. Dimensi

Perubahan desain tempat *hardware*. Desain *hardware* semula berukuran 15×10×6 cm menjadi 8,5×10×3,5 cm. Perubahan tersebut dilakukan agar memudahkan menyimpan alat pada motor listrik dan menghemat biaya.

BAB 5 : Implementasi Sistem dan Analisis

5.1 Hasil dan Analisis Implementasi

Pengujian dilakukan pada tanggal 20 Juni 2022 di area Kampus Terpadu Universitas Islam Indonesia dengan lima rute perjalanan yang bervariasi antara lain rute pertama (Gedung Olahraga UII – Gedung Olahraga UII) pada Gambar 5.1, rute kedua (Gedung Fakultas Teknologi Industri UII– Gedung Fakultas Hukum UII) pada Gambar 5.2, rute ketiga (Gedung Fakultas Hukum UII – Gedung Fakultas Hukum UII) pada Gambar 5.3, rute keempat (Gerbang Utama UII – Gerbang Utama UII) pada Gambar 5.4, dan rute kelima (Gedung Perpustakaan UII – Gedung Perpustakaan UII) pada Gambar 5.5. Berdasarkan hasil pengujian kemudian dilakukan perhitungan nilai *error* dengan menggunakan Persamaan 5.1.

$$\%error = \frac{approximate - exact}{exact} \times 100 \quad (5.1)$$

Keterangan :

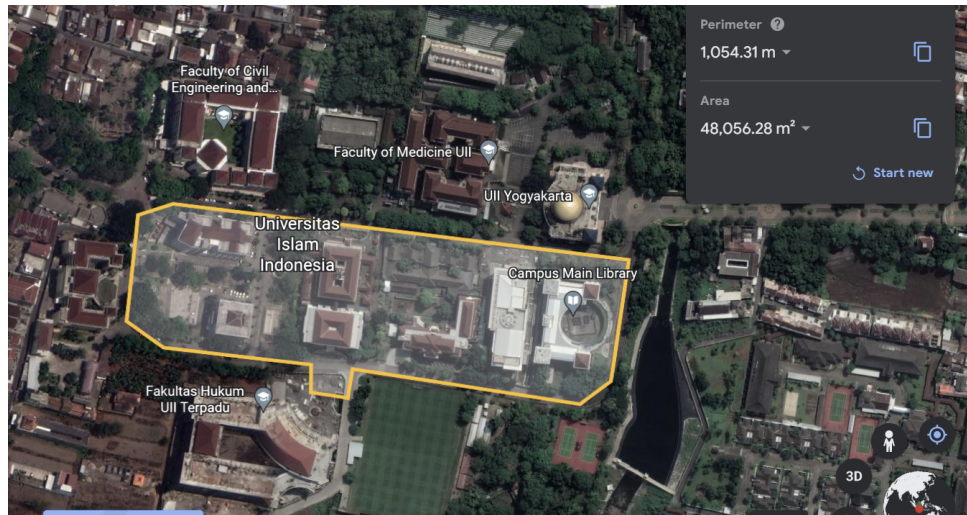
- Error* = Kesalahan (%).
- Approximate* = Nilai aktual pengamatan.
- Exact* = Nilai eksak atau asli



Gambar 5.1. Rute pertama (Gedung Olahraga UII – Gedung Olahraga UII)



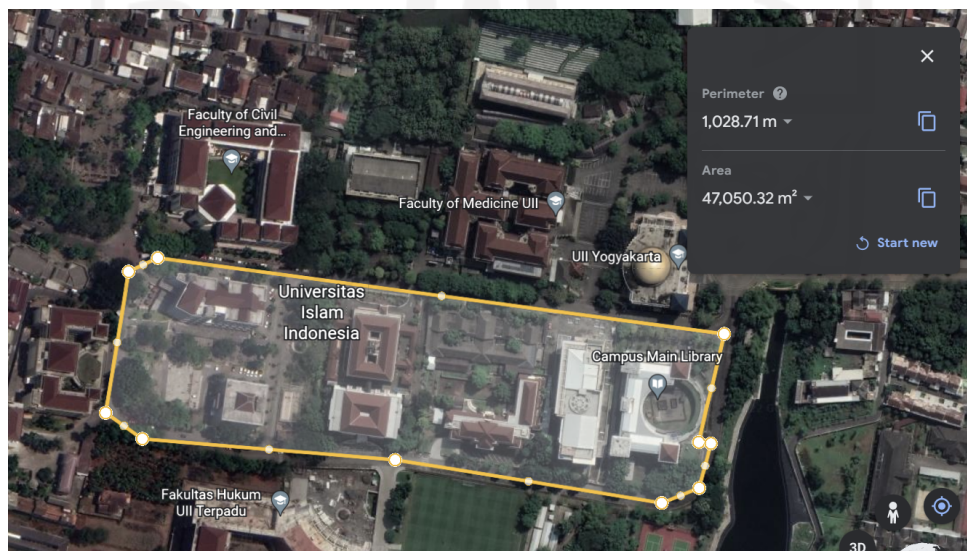
Gambar 5.2. Rute kedua (Gedung Fakultas Teknologi Industri UII– Gedung Fakultas Hukum UII)



Gambar 5.3. Rute ketiga (Gedung Fakultas Hukum UII – Gedung Fakultas Hukum UII)



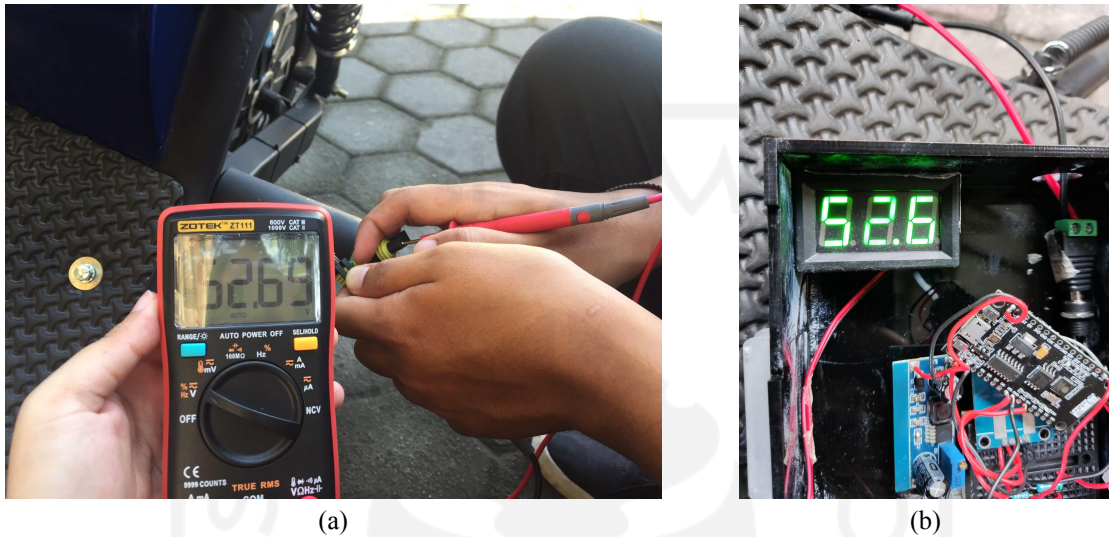
Gambar 5.4. Rute keempat (Gerbang Utama UII – Gerbang Utama UII)



Gambar 5.5. Rute kelima (Gedung Perpustakaan UII – Gedung Perpustakaan UII)

5.1.1 Monitoring Baterai

Pengujian *monitoring* tegangan baterai dilakukan pengambilan data pada titik awal dan titik akhir di setiap rute perjalanan. Sebelum melakukan pengujian dilakukan *voltmeter* digital dengan *multimeter* sebagai alat ukur tegangan (Gambar 5.6). Hasil pengujian tegangan baterai ditunjukkan pada Tabel 5.1 dan pengiriman data nilai tegangan pada *thingspeak* pada Gambar 5.7.



Gambar 5.6. Perbandingan pengukuran antara multimeter dengan *voltmeter* digital. (a) Pengukuran tegangan baterai dengan multimeter, (b) pengukuran tegangan baterai dengan *voltmeter* digital

Tabel 5.1 Hasil uji coba *monitoring* tegangan baterai

Rute	Tegangan yang Terbaca pada <i>voltmeter</i> (Volt)		Tegangan yang Terbaca pada EM-IOT (Volt)		Error (%)	
	Awal	Akhir	Awal	Akhir	Awal	Akhir
1	52,2	51,8	53,16	52,72	1,83	1,78
2	51,8	51,4	52,66	52,33	1,66	1,8
3	51,4	51,0	52,33	51,78	1,8	1,53
4	50,7	49,8	51,40	50,80	1,38	2
5	49,8	49,4	50,74	50,25	1,89	1,72
	Rata-Rata Error				1,71	1,77

Hasil pembacaan tegangan baterai yang tertampil pada aplikasi EM-IOT diolah menjadi persentase. Hasil pengujian persentase baterai termuat pada Tabel 5.2 dan pengiriman data persentase baterai pada *thingspeak* termuat pada Gambar 5.8.

Tabel 5.2 Hasil uji coba persentase tegangan baterai

Rute	Tegangan (volt)		Persentase Tertampil pada Aplikasi (%)		Perhitungan Persentase (%)		Error (%)	
	Awal	Akhir	Awal	Akhir	Awal	Akhir	Awal	Akhir
1	53,16	52,72	94	91	94,4	91,47	0,42	0,51
2	52,66	52,33	91	88	91,07	88,87	0,08	0,98
3	52,33	51,78	88	85	88,87	85,20	0,98	0,23
4	51,40	50,80	82	78	82,67	78,67	0,81	0,85
5	50,74	50,25	78	75	78,27	75,00	0,34	0
	Rata-Rata Error						0,53	0,52

Hasil pembacaan tegangan baterai yang tertampil pada aplikasi EM-IOT diolah menjadi sisa jarak tempuh. Hasil pengujian termuat pada Tabel 5.3 dan pengiriman data sisa jarak tempuh pada *thingspeak* termuat pada Gambar 5.9.

Tabel 5.3 Hasil uji coba sisa jarak

Rute	Tegangan (volt)		Sisa Jarak Tempuh Tertampil pada Aplikasi (km)		Perhitungan Jarak Tempuh (km)		Error (%)	
	Awal	Akhir	Awal	Akhir	Awal	Akhir	Awal	Akhir
1	53,16	52,72	35	34	35,4	34,3	1,13	0,87
2	52,66	52,33	34	33	34,15	33,33	0,44	0,98
3	52,33	51,78	33	31	33,33	31,95	0,98	2,97
4	51,40	50,80	31	29	31,00	29,50	0	1,69
5	50,74	50,25	29	28	29,35	28,13	1,19	0,44
	Rata-Rata Error						0,75	1,39

Hasil pengujian *monitoring* baterai yang terdiri dari tegangan, persentase baterai, dan sisa jarak tempuh yang tertampil pada aplikasi EM-IOT telah dapat menampilkan data dengan rata-rata *error* tidak melebihi 5%. Maka sistem dikatakan bekerja dengan baik memenuhi spesifikasi yang telah ditentukan.

5.1.2 Location Tracking

Pengujian *location tracking* dilakukan dengan penentuan lokasi awal dan akhir rute dengan menggunakan *Google Maps* untuk dapat mengetahui nilai *latitude* dan *longitude* yang sesungguhnya. Pembacaan nilai *latitude* dan *longitude* menggunakan penulisan jenis DD (Derajat Desimal). Pengambilan data dilakukan pada titik awal dan akhir di setiap rute

perjalanan yang telah ditentukan. Hasil uji coba nilai *latitude* dan *longitude* pada awal rute perjalanan termuat pada Tabel 5.4, akhir rute perjalanan pada Tabel 5.5, dan pengiriman data nilai *latitude* dan *longitude* pada *thingspeak* termuat pada Gambar 5.10.

Tabel 5.4 Hasil uji coba *location tracking* saat awal rute perjalanan

Rute	Titik Lokasi Sebenarnya		Titik Lokasi EM-IOT		Error (%)	
	<i>Latitude</i>	<i>Longitude</i>	<i>Latitude</i>	<i>Longitude</i>	<i>Latitude</i>	<i>Longitude</i>
1	-7.686477	110.409726	-7.686530	110.409733	0,0007	0,000006
2	-7.686700	110.410362	-7.686714	110.410372	0,0002	0,000009
3	-7.688885	110.413215	-7.688910	110.413198	0,0003	0,000015
4	-7.687679	110.418630	-7.687731	110.418542	0,0007	0,00008
5	-7.688699	110.415420	-7.688617	110.415496	0,0011	0,000069
Rata-Rata Error					0,0006	0,00004

Tabel 5.5 Hasil uji coba *location tracking* saat akhir rute perjalanan

Rute	Titik Lokasi Sebenarnya		Titik Lokasi EM-IOT		Error (%)	
	<i>Latitude</i>	<i>Longitude</i>	<i>Latitude</i>	<i>Longitude</i>	<i>Latitude</i>	<i>Longitude</i>
1	-7.686477	110.409726	-7.686532	110.409772	0,0007	0,000042
2	-7.689126	110.413310	-7.689292	110.413271	0,0022	0,000035
3	-7.688885	110.413215	-7.688910	110.413198	0,0003	0,000015
4	-7.687679	110.418630	-7.687731	110.418542	0,0007	0,00008
5	-7.688699	110.415420	-7.688642	110.415534	0,0007	0,000103
Rata-Rata Error					0,0009	0,00006

Hasil pengujian *location tracking* yang terdiri dari nilai *latitude* dan *longitude* yang tertampil pada aplikasi EM-IOT telah dapat menampilkan data dengan rata-rata *error* tidak melebihi 5%. Maka sistem dikatakan bekerja dengan baik memenuhi spesifikasi yang telah ditentukan.

5.1.3 Delay Pengiriman Data

Data yang dihasilkan oleh NodeMCU ESP12E dikirim menuju *thingspeak*. Pengiriman data tersebut terdapat *delay*. Pada Tabel 5.6 memuat *delay* pengiriman data pada *thingspeak*.

Tabel 5.6 Pengiriman data pada *thingspeak*

No	Tanggal/Waktu	Tegangan (Volt)	Persentase (%)	Sisa Jarak (km)	Latitude	Longitude
1	17/07/2022 15:53:40	44,49	36	13	-7.686660	110.409906
2	17/07/2022 15:53:55	44,27	35	13	-7.686700	110.409907
3	17/07/2022 15:54:10	44,27	35	13	-7.686692	110.409907
4	17/07/2022 15:54:25	44,22	34	13	-7.686715	110.409940
5	17/07/2022 15:54:40	43,89	32	12	-7.686704	110.409970
6	17/07/2022 15:54:55	43,23	28	10	-7.686878	110.410586
7	17/07/2022 15:55:10	44,16	34	12	-7.687016	110.410867
8	17/07/2022 15:55:25	44,11	34	12	-7.686973	110.410791
9	17/07/2022 15:55:40	44,16	34	12	-7.687004	110.410550
10	17/07/2022 15:55:55	44,22	34	13	-7.687011	110.410550

Berdasarkan Tabel 5.6 data akan diperbarui pada *thingspeak* setiap 15 detik. *Delay* tersebut merupakan ketetapan dari platform *thingspeak*. Sehingga hal tersebut tidak bisa diminimalisir. Akan tetapi *delay* 15 detik masih dapat dikatakan baik dan tidak membahayakan untuk *monitoring*. Maka dengan *delay* 15 detik telah memenuhi spesifikasi yang ditentukan.

5.2 Pengalaman Pengguna

Uji penggunaan aplikasi EM-IOT dilakukan kepada partisipan. Hal tersebut bertujuan untuk memperoleh masukan terkait aplikasi yang telah dibuat dari partisipan sebagai pengguna. Tabel 5.7 merupakan hasil tanggapan dan masukan dari pengguna setelah menggunakan aplikasi.

Tabel 5.7 Pengalaman pengguna

No	Fitur/Komponen	Capaian	Aksi/Perbaikan
1	Fungsi	Fungsi sebagai <i>monitoring</i> tegangan baterai, persentase baterai, sisa jarak, dan lokasi terkini yang ditampilkan melalui aplikasi sudah berjalan dengan baik.	Fungsi <i>monitoring</i> baterai yang terdiri dari tegangan baterai, persentase baterai, dan sisa jarak serta <i>monitoring</i> lokasi terkini dapat dipertahankan. Karena sistem tersebut dapat berjalan atau berfungsi dengan baik. Hal ini ditunjukkan dengan hasil pengujian nilai error tidak melebihi 5%.
2	Kemudahan	Aplikasi mudah dipahami	Tindakan yang diberikan adalah mempertahankan. Karena aplikasi yang dibuat ramah pengguna.
3	Keamanan	Keamanan data sudah cukup baik karena perlu <i>login</i> sebelum mengakses aplikasi	Sistem keamanan dipertahankan. Karena pentingnya kerahasiaan data pengguna khususnya lokasi terkini kendaraan.

5.3 Dampak Implementasi Sistem

5.3.1 Teknologi/Inovasi

Hasil perancangan EM-IOT didasari oleh observasi untuk mengumpulkan referensi teknologi terdahulu yang digunakan untuk mengembangkan EM-IOT sesuai dengan kebutuhan. Hasil dari implementasi observasi alat termuat pada Tabel 5.8.

Tabel 5.8 Perbandingan dengan alat yang sudah ada

No	Fitur/Komponen	Sistem yang dibuat	Adam Zuhdi Sisyadi	Fitria Dayanti	M. A. Budiman, dkk
1	Fitur Sistem	<i>Monitoring</i> tegangan baterai, persentase baterai, sisa jarak, lokasi terkini kendaraan.	<i>Monitoring</i> nilai tegangan, arus, dan kecepatan.	<i>Monitoring charging</i> dan pengisian baterai.	Pelacak dan pengendali kendaraan jarak jauh.
2	Ketahanan Sistem	Anti air (<i>IP22</i>).	Tidak ada.	Tidak ada.	Tidak ada.
3	Aplikasi	Android <i>App</i> .	Tidak ada.	Tidak ada.	Android <i>App</i> .

BAB 6 : Kesimpulan dan Saran

6.1 Kesimpulan

Tercatat 18 unit SPKLU yang tersebar di 16 titik wilayah Jawa Tengah dan D.I. Yogyakarta. Keterbatasan SPKLU berdampak mengganggu perjalanan. Sebab pengisian baterai membutuhkan waktu yang lama dan ketersediaan stasiun pengisian *fast charging* yang masih terbatas. Oleh karena itu, diperlukannya sistem untuk *monitoring* kondisi baterai dan posisi kendaraan. *Monitoring* kondisi baterai diperlukan untuk mengetahui sisa tegangan dan sisa jarak tempuh. Dan *monitoring* lokasi terkini kendaraan diperlukan agar mempermudah mengetahui lokasi terkini kendaraan. Sehingga, bermanfaat untuk manajemen atau mengatur waktu untuk *charging* kendaraan dan mengetahui lokasi dari kendaraan.

Berdasarkan hasil perancangan, pembuatan, dan pengujian pada EM-IOT diperoleh beberapa kesimpulan sebagai berikut:

1. Sistem EM-IOT dapat berfungsi dengan baik untuk melakukan *monitoring* tegangan baterai, persentase baterai, sisa jarak, dan lokasi terkini kendaraan.
2. Sistem memiliki rata-rata *error* pembacaan tegangan awal rute 1,71 dan akhir rute 1,77.
3. Sistem memiliki rata-rata *error* pembacaan persentase awal rute 0,53 dan akhir rute 0,52.
4. Sistem memiliki rata-rata *error* pembacaan sisa jarak tempuh awal rute 0,75 dan akhir rute 1,39.
5. Sistem memiliki rata-rata *error* pembacaan *latitude* awal rute 0,0006 dan akhir rute 0,0009.
6. Sistem memiliki rata-rata *error* pembacaan *longitude* awal rute 0,00004 dan akhir rute 0,00006.
7. Pada persentase baterai akan menunjukkan 0% saat tegangan baterai memasuki ambang batas (*threshold*) sebesar 39 volt, dan 100% saat tegangan baterai maksimal sebesar 54 volt.
8. Diperlukan waktu 15 detik untuk data dapat diperbarui, hal ini disebabkan terjadi *delay* pengiriman data menuju *Thingspeak*.

6.2 Saran

Berdasarkan pembuatan dan pengujian EM-IOT memiliki beberapa kelemahan. Oleh karena itu terdapat beberapa saran yang diberikan untuk penelitian berikutnya sebagai berikut:

1. Aplikasi yang dibuat mampu beroperasi di semua jenis *operating* sistem.
2. Mengurangi nilai *error* pembacaan nilai tegangan.
3. Mengurangi *delay* pengiriman data.

Daftar Pustaka

- [1] Badan Pusat Statistik, “Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis (Unit), 2018-2020”. Available:
<https://www.bps.go.id/indicator/17/57/1/jumlah-kendaraan-bermotor.html> (accessed Feb. 23, 2022).
- [2] Dwiki Muda Yulanto dan Henry Iskandar, “Studi Analisis Perkembangan Teknologi Kendaraan Listrik Hibrida,” *J. Automot. Technol. Vocat. Educ.*, vol. 02, no. 1, p. 14, Apr. 2021.
- [3] V. Tulus Pangapoi Sidabutar, “Kajian pengembangan kendaraan listrik di Indonesia: prospek dan hambatannya,” *J. Paradig. Ekon.*, vol. 15, no. 1, pp. 21–38, May 2020, doi: 10.22437/paradigma.v15i1.9217.
- [4] Kementerian Perindustrian RI, “Siap Masuki Era Kendaraan Listrik, Indonesia Fokus Bangun Ekosistem,” Oktober 2021. Available:
<https://kemenperin.go.id/artikel/22865/Siap-Masuki-Era-Kendaraan-Listrik,-Indonesia-Fokus-Bangun-Ekosistem> (accessed Feb. 23, 2022).
- [5] Kementerian ESDM RI, “187 SPKLU Siap Layani Kendaraan Listrik, Ini Sebarannya,” Oktober 2021. Available:
<https://www.esdm.go.id/id/media-center/arsip-berita/187-spklu-siap-layani-kendaraan-listrik-ini-sebarannya> (accessed Feb. 23, 2022).
- [6] Pratomo FJ, “5 Masalah Paling Umum yang Terjadi pada Mobil Listrik I Carro.id,” *CARRO*, Aug. 16, 2021. Available:
<https://carro.id/blog/berita/5-masalah-paling-umum-yang-terjadi-pada-mobil-listrik/5627/> (accessed Feb. 23, 2022).
- [7] M. H. A. Wahab *et al.*, “IoT-Based Battery Monitoring System for Electric Vehicle,” *Int. J. Eng.*, p. 6, 2018, doi: 10.14419/ijet.v8i1.4.25472.
- [8] Adam Zuhdi Sisyadi, “Rancang Bangun Sistem Monitoring Penggunaan Energi pada Mobil Listrik Universitas Jember,” Universitas Jember, Jember, 2020.
- [9] F. Dayanti, “Perancangan Sistem Charging dan Monitoring pada Baterai Level Tegangan 12 Volt DC Berbasis Mikrokontroler ATMEGA 16,” Institut Teknologi Sepuluh Nopember Surabaya, Surabaya, 2018. [Online]. Available:
https://repository.its.ac.id/58424/1/10511500000034-Non_Degree.pdf
- [10] Budiman, Muhammad Arif, Harefa, Auli Zatulo, dan Sakti, Dolly Virgian Shaka Yudha, “Perancangan Sistem Pelacak GPS dan Pengendali Kendaraan Jarak Jauh Berbasis Arduino,” Jul. 2020, doi: 10.5281/ZENODO.4323233.

LAMPIRAN – LAMPIRAN

- *Logbook* Kegiatan Selama Proses Tugas Akhir 2

Hari, Tanggal	Deskripsi Kegiatan
08 April 2022	Melakukan diskusi terkait spesifikasi sistem
09 April 2022	Melakukan diskusi terkait spesifikasi sistem
27 April 2022	Diskusi dengan dosen pembimbing
28 April 2022	Membuat skematik rancangan alat
29 April 2022	Membuat skematik rancangan alat
8 Mei 2022	Membuat kode pemrograman
9 Mei 2022	Membuat kode pemrograman
11 Mei 2022	Merancang aplikasi EMIOT
13 Mei 2022	Merancang aplikasi EMIOT
19 Mei 2022	Pembuatan laporan TR-201
20 Mei 2022	Pengumpulan laporan TR-201
25 Mei 2022	Pembelian komponen elektronik
27 Mei 2022	Pembelian komponen elektronik
3 Juni 2022	Melakukan uji coba dan pengambilan data alat
5 Juni 2022	Melakukan uji coba dan pengambilan data alat
7 Juni 2022	Melakukan perhitungan error data
9 Juni 2022	Melakukan perhitungan error data
12 Juni 2022	Pembuatan laporan TR-202
13 Juni 2022	Pembuatan laporan TR-202
15 Juni 2022	Pengumpulan laporan TR-202
17 Juni 2022	Pembuatan revisi laporan TR-202
20 Juni 2022	Pembuatan revisi laporan TR-202
21 Juni 2022	Pembuatan laporan Tugas Akhir 2
25 Juni 2022	Pembuatan laporan Tugas Akhir 2
27 Juni 2022	Pembuatan laporan Tugas Akhir 2
10 Juli 2022	Pembuatan laporan Tugas Akhir 2
18 Juli 2022	Pengumpulan laporan Tugas Akhir 2

▪ Dokumen TA201 dan TA202

DEPARTMENT OF ELECTRICAL ENGINEERING
Capstone Design 2021 - 2022

TECHNICAL REPORT

IDENTITAS

Dokumentasi Proses	TA201
Topik / Judul Capstone Design	EM-IoT - Sistem Monitoring Persentase Baterai dan Estimasi Penggunaan Daya untuk Rute Perjalanan Berbasis IoT pada Motor Listrik
Nama Lengkap	Tasya Thafiq Sahabilla
No. Induk Mahasiswa (NIM)	18224069 18224069
Dosen Pembimbing 1	Hifah Inayah S.T. M.Eng
Dosen Pembimbing 2	Dwi Ann Rama Wati S.T. M.Eng

Spesifikasi Sistem

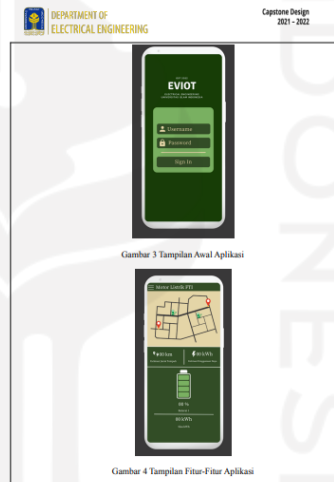
Berdasarkan hasil studi literatur dan dasar teori yang telah disusun, dapat ditentukan spesifikasi sistem dan pengaruh luar terhadap rancangan sistem sebagai berikut:

1. Alat dapat mengukur pengisian baterai dengan sensor tegangan dan arus dengan toleransi kesalahan maksimal 9%.
2. Estimasi perjalanan yang dapat ditempuh dengan sisa ketersediaan daya baterai.
3. Data monitoring dan estimasi daya dapat diakses melalui smartphone khusus Android.
4. Terdapat indikator peringatan ketika daya kendaraan tidak mencukupi rencana tujuan perjalanan.
5. Alat (Hardware) memiliki kemampuan tahan air dengan rating IP22.
6. Komunikasi sensor dan mikrokontroler dengan menggunakan protokol wireless.
7. Mikrokontroler mengkonversi data analog dari sensor menjadi data digital menggunakan ADC built-in dengan resolusi 10 bit.
8. Konektivitas alat agar terhubung ke Smartphone menggunakan wireless.
9. Sumber daya listrik sebesar 12V.

DEPARTMENT OF ELECTRICAL ENGINEERING
Capstone Design 2021 - 2022

Desain rancangan awal

Berdasarkan spesifikasi sistem yang telah ditentukan sebelumnya, usulan desain pertama yang disusun adalah membuat sistem IoT untuk monitoring baterai dan estimasi penggunaan daya untuk rute perjalanan dengan menggunakan aplikasi yang terhubung dengan smartphone terkhusus pada Android. Usulan ini bertujuan untuk mempermudah pengendara kendaraan listrik untuk melakukan pemantauan terkait kondisi baterai secara *real-time* dan estimasi baterai yang akan dipakai untuk rute perjalanan yang akan ditempuh. Hal tersebut dapat mempermudah pengendara untuk mengatur waktu pengisian ulang baterai atau penarikan baterai. Usulan sistem IoT ini terbagi menjadi dua yaitu hardware dan juga software. Hardware dirancang dengan spesifikasi IP rating 22 yang berarti memiliki proteksi dari masuknya jari dan tidak akan rusak apabila terkena tetesan air secara vertikal IP rating 22 merupakan persyaratan minimum untuk desain aksesoris listrik untuk penggunaan indoor. Dan hardware akan diletakkan pada kendaraan listrik berdekatan dengan penempatan baterai dari kendaraan tersebut. Pada kendaraan motor listrik yang akan digunakan untuk implementasi hardware akan diletakkan tepat dibawah jok motor seperti pada Gambar 1 dan desain hardware ditunjukkan dengan Gambar 2. Sedangkan software dirancang dengan menggunakan aplikasi yang hanya dapat diakses oleh pengguna smartphone Android dengan akun yang terkoneksi pada motor listrik. Pembuatan aplikasi dengan menggunakan Kotlin. Aplikasi dirancang dengan keamanan dengan menggunakan akun dengan tampilan seperti pada Gambar 3. Pada aplikasi terdapat indikator kapasitas baterai, peta, kapasitas daya yang tersisa dan estimasi jarak tempuh yang ditunjukkan dengan Gambar 4. Dan pada aplikasi ini terdapat notifikasi atau pemberitahuan yang akan diberikan apabila daya baterai tidak mencukupi untuk dilakukan perjalanan seperti pada Gambar 5.



DEPARTMENT OF ELECTRICAL ENGINEERING
Capstone Design 2021 - 2022

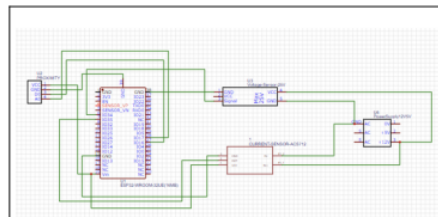
Gambar 6 Blok Diagram Desain Sistem Usulan 1.

Berdasarkan blok diagram pada Gambar 4.7, penggunaan perangkat keras dan lunak yang dipilih dengan penjelasan penggunaannya ditunjukkan pada Tabel 1.

Tabel 1 Penjelasan Perangkat Pada Desain Sistem Usulan 1

No	Nama Perangkat	Keterangan
1	Sensor ACS712 30A	Sensor ini berfungsi sebagai pembaca dan pengukur arus yang dapat mendeteksi arus hingga 30 A. Sensor ini akan terhubung dengan baterai kendaraan listrik dengan tujuan untuk mendeteksi arus yang mengalir pada baterai.
2	Rangkaian Pembagi Tegangan	Rangkaian pembagi tegangan digunakan untuk menurunkan tegangan output baterai untuk disalurkan kepada sensor tegangan DC. Hal ini disebabkan kapasitas sensor tegangan DC yang hanya dapat mendeteksi 25VDC sehingga tegangan output baterai terlebih dahulu diturunkan dengan menggunakan rangkaian pembagi tegangan.
3	Sensor Tegangan DC	Sensor tegangan ini memiliki fungsi sebagai pembaca dan pengukur tegangan DC dengan kapasitas 25 VDC. Sensor ini akan terhubung dengan rangkaian pembagi tegangan.

4	ESP32	ESP32 digunakan sebagai mikrokontroler dengan tujuan untuk komunikasi dan pemrosesan data dari hasil pembacaan sensor yang terkoneksi pada internet sehingga dapat mengirimkan data pembacaan menuju cloud.
6	Module GPS	Module GPS digunakan sebagai pembacaan lokasi untuk memudahkan penentuan titik tujuan maupun lokasi terkini dari kendaraan.
7	Cloud	Cloud digunakan untuk menyimpan dan mengirimkan data menuju aplikasi
8	Aplikasi	Merupakan platform dari hasil sistem.



Gambar 7 Skema Rangkaian Desain Sistem Usulan 1

Berdasarkan blok diagram sistem pada Gambar 6, proses pengambilan data dilakukan dengan menggunakan sensor yang terhubung dengan baterai. Sensor arus untuk mendeteksi arus pada baterai yang dipilih adalah sensor ACS712. Kemudian sebelum sensor tegangan bekerja terlebih dahulu tegangan dari baterai diturunkan tegangannya dengan menggunakan pembagi tegangan. Digunakannya pembagi tegangan sebab jenis sensor yang dipilih hanya dapat mendeteksi tegangan mencapai 25 V sedangkan baterai memiliki kapasitas tegangan hingga mencapai 48V dengan menggunakan Persamaan (2.2)

$$V_{out} = \frac{R_2}{R_1 + R_2} \times V_{in} \quad (2.2)$$

Saat $V_{in} = 48 \text{ V}$

$$V_{out} = \frac{1,1 \text{ k}\Omega}{10\text{k}\Omega + 1,1\text{k}\Omega} \times 48 = 4,8 \text{ V}$$

Saat $V_{in} = 0 \text{ V}$

$$V_{out} = \frac{1,1 \text{ k}\Omega}{10\text{k}\Omega + 1,1\text{k}\Omega} \times 0 = 0 \text{ V}$$

Sehingga tegangan yang akan dibaca oleh sensor memiliki range 0-4,8 V. Tegangan yang telah diturunkan kemudian akan dilakukan pembacaan tegangan oleh sensor tegangan dengan spesifikasi sensor yang ditunjukkan pada Tabel 2.1. Selain digunakan sensor untuk pengambilan data, digunakan modul GPS APM2.5 NEO-6M pengumpulan data untuk mengetahui lokasi terkini dan lokasi tujuan yang digunakan untuk mengetahui jarak yang akan ditempuh suatu perjalanan. Data yang didapatkan oleh sensor dan modul GPS akan dikirim menuju mikrokontroler dengan menggunakan protokol *WiFi*.

Desain rancangan akhir
Terdapat perubahan berdasarkan hasil desain rancangan awal, yaitu desain sistem perangkat keras. Pada desain rancangan akhir dipilih Arduino UNO sebagai mikrokontroler dan modul SIM800L sebagai media untuk mendukung pengiriman data dengan memanfaatkan SIM Card. Dan pengukuran tegangan pada baterai langsung memanfaatkan rangkaian pembagi dengan batas maksimal tegangan masukan yang masuk adalah 60 V dan batas minimal 0 V dengan perhitungan menggunakan Persamaan (2.3). Sehingga, alur kerja sistem ditunjukkan dengan diagram blok pada Gambar 8.

$$V_{out} = \frac{R_2}{R_1 + R_2} \times V_{in} \quad (2.3)$$

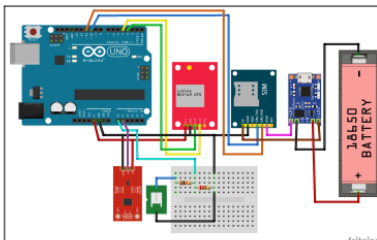
Saat $V_{in} = 60 \text{ V}$

$$V_{out} = \frac{220 \Omega}{10\text{k}\Omega + 220 \Omega} \times 60 = 1,29 \text{ V}$$

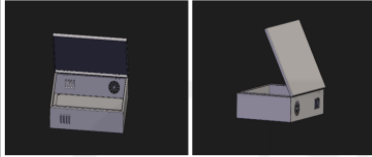
Saat $V_{in} = 0 \text{ V}$

$$V_{out} = \frac{220 \Omega}{10\text{k}\Omega + 220 \Omega} \times 0 = 0 \text{ V}$$

Gambar 8 Blok Diagram Desain Akhir.

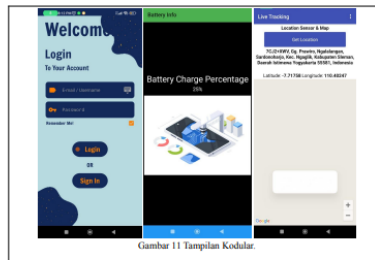


Gambar 9 Rangkaian Skematik Desain Akhir.



Gambar 10 Desain Case Hardware.

Desain sistem untuk perangkat lunak menggunakan aplikasi smartphone berbasis Android yang bernama Kodlar. Kodlar dipilih dengan alasan fitur-fitur di dalamnya memudahkan dalam membuat desain *User Interface* yang diinginkan, seperti ketersediaan layout, widget, hingga font.



Gambar 11 Tampilan Kodlar.

Production Costs

No	Item/Pengeluaran	Satuan	Harga Satuan	Jumlah	Harga Total
1.	Arduino UNO + kabel	pcs	Rp 80.000	1	Rp 80.000
2.	Module GPS AMP 2.5 NEO-M8	pcs	Rp 83.000	2	Rp 166.000
3.	Sensor ACS712 30A	pcs	Rp 20.000	2	Rp 40.000
4.	Resistor 10 kOhm	pcs	Rp 50	3	Rp 150
5.	Resistor 220 Ohm	pcs	Rp 50	3	Rp 150
6.	Kabel Jumper	set	Rp 27.000	1	Rp 27.000
7.	Module Sim800l	pcs	Rp 86.500	2	Rp 173.000
8.	Telkomsel IoT Sim Card	pcs	Rp 35.000	2	Rp 70.000
9.	LIBLOX GPSV5 NEO-M8N	pcs	Rp 214.000	1	Rp 214.000

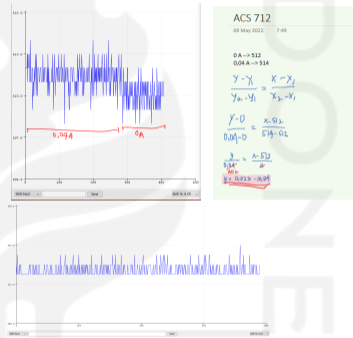
10.	Tanah Solder	pcs	Rp 20.000	1	Rp 20.000
11.	Panah Solder	pcs	Rp 45.000	1	Rp 45.000
12.	Li-ion Charger-Dual	pcs	Rp 40.000	1	Rp 40.000
13.	Box baterai	pcs	Rp 5.500	1	Rp 5.500
14.	Kawat pin konektor	pcs	Rp 200	20	Rp 40.000
15.	Black housing 1 pin	pcs	Rp 300	20	Rp 60.000
16.	Kabel pita	meter	Rp 8000	5	Rp 40.000
	TOTAL				Rp 895.800

Log / Catatan Aktivitas (meliputi perencanaan, aktivitas/tugas, dan capaiannya)
 Capaian yang telah tm lakukan yaitu:
 1. Pembelian peralatan.
 2. Kalibrasi peralatan.
 3. Pambuatan sistem.
 Pada proses ini terdapat kendala yang dihadapi, yaitu keterbatasan akses modul GPS yang harus pada lingkungan *outdoor*, hasil pembacaan tegangan dan arus pada baterai saat kode program digabungkan masih belum sesuai, dan modul sim800l mengalami kendala sistem yaitu *rebooting* terus-menerus ketika akan terhubung ke jaringan.

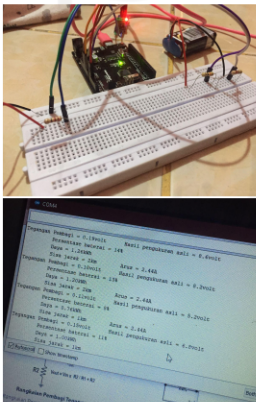
Catatan Lambaian

LAMPIRAN-LAMPIRAN

Sertakan dokumen bukti-bukti pendukung dari deskripsi/penjelasan laporan ini (jika ada)
 Proses regresi sensor arus.



Mengabungkan sensor arus dan pembagi tegangan




Presi GPS



TECHNICAL REPORT

IDENTITAS

Dokumentasi Proses	TA202
Topik / Judul Capstone Design	EM-IOT : Sistem Monitoring Persentase Baterai dan Location Tracking Berbasis IoT pada Motor Listrik
Nama Lengkap	1. Tasya Thifali Salehila 2. Haikal Faiz Ramadhan
No. Induk Mahasiswa (NIM)	1. 18524060 2. 18524064
Dosen Pembimbing 1	 Hifah Imawati S. T. M. Eng
Dosen Pembimbing 2	Dwi Ana Ratna Wati S. T. M. Eng

Metode / Rancangan Pengujian Sistem

Metode atau rancangan pengujian sistem dilakukan untuk memastikan sistem dapat dengan baik. Pengujian terbagi menjadi dua bagian yaitu monitoring baterai dan location tracking. Pengujian dilakukan baik apabila memenuhi tiga parameter pengukuran yaitu tegangan, latitude, dan longitude. Pelaksanaan pengujian dilakukan pada area Kampus Terpadu Universitas Islam Indonesia. Mekanisme perancangan pengujian sistem dengan tiga parameter tersebut terbagi menjadi dua kondisi pengujian yaitu saat awal pengujian dan akhir pengujian untuk setiap rutenya.

Pengujian untuk monitoring baterai menggunakan baterai dengan spesifikasi tegangan maksimal sebesar 54 volt dan threshold sebesar 39 volt. Pengujian ini menggunakan satu parameter pengujian yaitu tegangan. Pada pelaksanaan pengujian dilakukan pengukuran tegangan pada awal rute perjalanan dengan menggunakan volt meter digital yang terhubung langsung pada baterai dan tampilan pada aplikasi. Dan pada akhir perjalanan dilakukan pengukuran tegangan seperti awal rute perjalanan.

Pengujian kedua yang dilakukan untuk location tracking menggunakan dua parameter pengujian yaitu latitude dan longitude. Latitude dan longitude merupakan koordinat letak geografis yang selalu terdapat pada setiap lokasi. Latitude merupakan garis bujur yang mana terbentang dari utara hingga selatan, sedangkan longitude merupakan garis lintang yang mana terbentang dari barat hingga timur. Pada pelaksanaan pengujian dilakukan perbandingan nilai latitude dan longitude pada titik awal dan akhir dari setiap rute perjalanan. Perbandingan dilakukan menggunakan titik lokasi berdasarkan nilai latitude dan longitude yang tertera di Google Maps dengan titik lokasi serta nilai latitude dan longitude yang terlampir pada aplikasi.

Metode Pengukuran untuk pengujian Sistem

Metode pengukuran untuk pengujian sistem dengan lima rute perjalanan yang bervariasi antara lain rute pertama (Gedung Olah Raga UII - Gedung Olah Raga UII), rute kedua (Gedung Fakultas Teknologi Industri UII - Gedung Fakultas Hukum UII), rute ketiga (Gedung Fakultas Hukum UII - Gedung Fakultas Hukum UII), rute keempat (Gerbang Utama UII - Gerbang Utama UII), dan rute kelima (Gedung Perpustakaan UII - Gedung Perpustakaan UII), dan perhitungan nilai error hasil pembacaan pengujian. Selain itu, dilakukan perbandingan titik lokasi antara aplikasi dengan Google Maps. Serta dilakukan perhitungan sisa jarak yang tersisa dengan setiap 1 km menghabiskan 0,4 volt dan perhitungan nilai error untuk pembacaan hasil pengukuran tegangan dengan parameter perbandingan antara aplikasi dan realita dengan menggunakan persamaan persentase kesalahan.

$$\%error = \frac{aplikasi - realita}{realita} \times 100 \quad (1)$$



Hasil Pengujian Sistem

Adapun hasil pengujian yang telah dilakukan pada 20 Juni 2022 dengan lima rute perjalanan yang bervariasi sebagai berikut.

- Pengujian pertama dilakukan dari Gedung Olah Raga UII dan berakhir di lokasi yang sama dengan jarak tempuh sebesar 987,34 m.



Gambar 1.1 Rute Persejataan Pertama.

Gambar 1.2 Sisa baterai awal pada voltmeter (a) dan pada aplikasi EM-IOT di rute pertama.

Hasil pengujian yang ditunjukkan dengan Gambar 1.2 terdapat perbedaan pembacaan antara voltmeter dengan aplikasi, oleh sebab itu didapatkan persentase kesalahan pembacaan sebagai berikut.

$$\%error = \frac{aplikasi - voltmeter}{voltmeter} \times 100 \quad (1.1)$$

$$\%error = \frac{53,8 - 54,2}{54,2} \times 100$$

$$\%error = 1,83$$

Persentase baterai yang terlampir pada aplikasi menggunakan perhitungan dengan persamaan sebagai berikut.

$$\%persentase = \frac{tegangan \text{ terdapat di aplikasi} - \text{threshold}}{\text{tegangan maksimal} - \text{threshold}} \times 100 \quad (1.2)$$

$$\%persentase = \frac{53,8 - 39}{54 - 39} \times 100$$

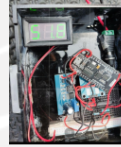

$$\%persentase = 94,4$$

Sisa jarak yang dapat ditempuh dengan sisa baterai terlampir pada aplikasi menggunakan perhitungan dengan persamaan sebagai berikut.

$$\text{sisa jarak} = \frac{\text{tegangan terdapat di aplikasi} - \text{threshold}}{0,4} \quad (1.3)$$

$$\text{sisa jarak} = \frac{53,8 - 39}{0,4}$$

$$\text{sisa jarak} = 35,4 \text{ km}$$

Gambar 1.3 Sisa baterai akhir pada voltmeter (a) dan pada aplikasi EM-IOT di rute pertama.

Hasil pengujian yang ditunjukkan dengan Gambar 1.3 terdapat perbedaan pembacaan antara voltmeter dengan aplikasi, oleh sebab itu didapatkan persentase kesalahan pembacaan sebagai berikut.

$$\%error = \frac{aplikasi - voltmeter}{voltmeter} \times 100 \quad (1.4)$$

$$\%error = \frac{53,9 - 54,4}{54,4} \times 100$$

$$\%error = 1,78$$

Persentase baterai yang terlampir pada aplikasi menggunakan perhitungan dengan persamaan sebagai berikut.

$$\%persentase = \frac{\text{tegangan terdapat di aplikasi} - \text{threshold}}{\text{tegangan maksimal} - \text{threshold}} \times 100 \quad (1.5)$$

$$\%persentase = \frac{53,9 - 39}{54 - 39} \times 100$$

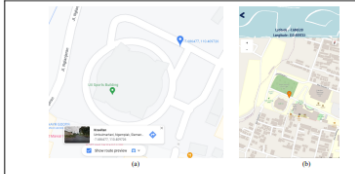
$$\%persentase = 91,5$$

Sisa jarak yang dapat ditempuh dengan sisa baterai terlampir pada aplikasi menggunakan perhitungan dengan persamaan sebagai berikut.

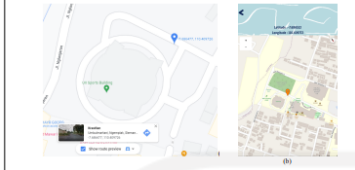
$$\text{sisa jarak} = \frac{\text{tegangan terdapat di aplikasi} - \text{threshold}}{0,4} \quad (1.6)$$

$$\text{sisa jarak} = \frac{53,9 - 39}{0,4}$$

$$\text{sisa jarak} = 34,3 \text{ km}$$



Gambar 1.4 Perbandingan titik lokasi Gedung Olah Raga UII berdasarkan Google Maps (a) dan aplikasi EMOT (b) pada awal nilai pertama.



Gambar 1.5 Perbandingan titik lokasi Gedung Olah Raga UII berdasarkan Google Maps (a) dan aplikasi EMOT (b) pada nilai rata-rata.

Berdasarkan hasil pengujian titik lokasi pada Gedung Olah Raga UII yang ditunjukkan pada Gambar 1.4 dan Gambar 1.5 dapat diketahui nilai kesalahan persentase kesalahan sebagai berikut.

$$\%error = \frac{|\text{latitudemenuju} - \text{sekarang}|}{\text{sekarang}} \times 100 \quad (1.7)$$

$$\%error = \frac{-7.268129 - (-7.2684777)}{7.2684777} \times 100$$

$$\%error = 0,0007$$

Kesalahan nilai latitude akhir rata-rata

$$\%error = \frac{|\text{latitudemenuju} - \text{sekarang}|}{\text{sekarang}} \times 100$$

$$\%error = \frac{-7.268129 - (-7.2684777)}{7.2684777} \times 100$$

$$\%error = 0,0007$$

Kesalahan nilai longitude awal rata-rata

$$\%error = \frac{|\text{longitudemenuju} - \text{sekarang}|}{\text{sekarang}} \times 100$$

$$\%error = \frac{118.410713 - 118.407728}{118.407728} \times 100$$

$$\%error = 0,000006$$

Kesalahan nilai longitude akhir rata-rata

$$\%error = \frac{|\text{longitudemenuju} - \text{sekarang}|}{\text{sekarang}} \times 100$$

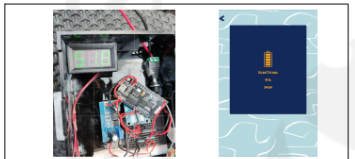
$$\%error = \frac{118.410713 - 118.407728}{118.407728} \times 100$$

$$\%error = 0,000042$$



Gambar 2.1 Rute Perbaikan Kuda

2. Pengujian kuda dilakukan dari Gedung Fakultas Teknologi Industri UII dan berakhir di Gedung Fakultas Hukum UII dengan jarak tempuh sebesar 1.015 m.



Gambar 2.2 Sisa baterai awal pada voltmeter (a) dan pada aplikasi EMOT (b) nilai kedua.

Hasil pengujian yang ditunjukkan dengan Gambar 2.2 terdapat perbedaan pembacaan antara voltmeter dengan aplikasi, oleh sebab itu didapatkan persentase kesalahan pembacaan sebagai berikut.

$$\%error = \frac{|\text{sekarang} - \text{sekarang}|}{\text{sekarang}} \times 100 \quad (2.1)$$

$$\%error = \frac{|\text{sekarang} - \text{sekarang}|}{\text{sekarang}} \times 100$$

$$\%error = 1,60$$

Persentase baterai yang tertampil pada aplikasi menggunakan perhitungan dengan persamaan sebagai berikut.

$$\%persentase = \frac{|\text{sekarang} - \text{sekarang}|}{\text{sekarang}} \times 100 \quad (2.2)$$

$$\%persentase = \frac{|\text{sekarang} - \text{sekarang}|}{\text{sekarang}} \times 100$$

$$\%persentase = 91,1$$

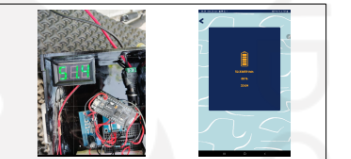
Sisa jarak yang dapat ditempuh dengan sisa baterai tertampil pada aplikasi menggunakan perhitungan dengan persamaan sebagai berikut.

$$\text{sisa jarak} = \frac{|\text{sekarang} - \text{sekarang}|}{\text{sekarang}} \quad (2.3)$$

$$\text{sisa jarak} = \frac{|\text{sekarang} - \text{sekarang}|}{\text{sekarang}}$$

$$\text{sisa jarak} = \frac{34}{1}$$

$$\text{sisa jarak} = 34,2 \text{ km}$$



Gambar 2.3 Sisa baterai akhir pada voltmeter (a) dan pada aplikasi EMOT (b) nilai kedua.

Hasil pengujian yang ditunjukkan dengan Gambar 2.3 terdapat perbedaan pembacaan antara voltmeter dengan aplikasi, oleh sebab itu didapatkan persentase kesalahan pembacaan sebagai berikut.

$$\%error = \frac{|\text{sekarang} - \text{sekarang}|}{\text{sekarang}} \times 100 \quad (2.4)$$

$$\%error = \frac{|\text{sekarang} - \text{sekarang}|}{\text{sekarang}} \times 100$$

$$\%error = 1,8$$

Persentase baterai yang tertampil pada aplikasi menggunakan perhitungan dengan persamaan sebagai berikut.

$$\%persentase = \frac{|\text{sekarang} - \text{sekarang}|}{\text{sekarang}} \times 100 \quad (2.5)$$

$$\%persentase = \frac{|\text{sekarang} - \text{sekarang}|}{\text{sekarang}} \times 100$$

$$\%persentase = 88,87$$

Sisa jarak yang dapat ditempuh dengan sisa baterai tertampil pada aplikasi menggunakan perhitungan dengan persamaan sebagai berikut.

$$\text{sisa jarak} = \frac{|\text{sekarang} - \text{sekarang}|}{\text{sekarang}} \quad (2.6)$$

$$\text{sisa jarak} = \frac{|\text{sekarang} - \text{sekarang}|}{\text{sekarang}}$$

$$\text{sisa jarak} = \frac{33,33}{1}$$

$$\text{sisa jarak} = 33,33 \text{ km}$$



Gambar 2.4 Perbandingan titik lokasi Gedung Fakultas Teknologi Industri UII berdasarkan Google Maps (a) dan aplikasi EMOT (b) pada awal nilai kedua.

Berdasarkan hasil pengujian titik lokasi pada Gedung Fakultas Teknologi Industri UII yang ditunjukkan pada Gambar 2.4 dapat diketahui nilai kesalahan persentase kesalahan sebagai berikut.

$$\text{Kesalahan nilai latitude awal rata-rata} \quad (2.7)$$

$$\%error = \frac{|\text{latitudemenuju} - \text{sekarang}|}{\text{sekarang}} \times 100$$

$$\%error = \frac{-7.268129 - (-7.2684777)}{7.2684777} \times 100$$

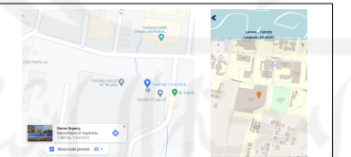
$$\%error = 0,0002$$

$$\text{Kesalahan nilai longitude awal rata-rata}$$

$$\%error = \frac{|\text{longitudemenuju} - \text{sekarang}|}{\text{sekarang}} \times 100$$

$$\%error = \frac{118.410713 - 118.410713}{118.410713} \times 100$$

$$\%error = 0,000009$$



Gambar 2.5 Perbandingan titik lokasi Gedung Fakultas Hukum UII berdasarkan Google Maps (a) dan aplikasi EMOT (b) pada akhir nilai kedua.

Berdasarkan hasil pengujian titik lokasi pada Gedung Fakultas Hukum UII yang ditunjukkan pada Gambar 2.5 dapat diketahui nilai kesalahan persentase kesalahan sebagai berikut.

$$\text{Kesalahan nilai latitude akhir rata-rata} \quad (2.8)$$

$$\%error = \frac{|\text{latitudemenuju} - \text{sekarang}|}{\text{sekarang}} \times 100$$

$$\%error = \frac{-7.268129 - (-7.2684777)}{7.2684777} \times 100$$

$$\%error = 0,0022$$

$$\text{Kesalahan nilai longitude akhir rata-rata}$$

$$\%error = \frac{|\text{longitudemenuju} - \text{sekarang}|}{\text{sekarang}} \times 100$$

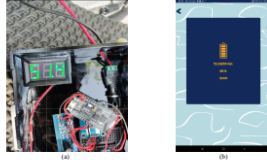
$$\%error = \frac{118.410713 - 118.410713}{118.410713} \times 100$$

$$\%error = 0,000035$$

3. Pengujian ketiga dilakukan dari Gedung Fakultas Hukum UII dan berakhir di lokasi yang sama dengan jarak tempuh sebesar 1.054 m.



Gambar 3.1 Ratu Pelemban Kutaja



Gambar 3.2 Sisa baterai awal pada voltmeter (a) dan pada aplikasi EMOT (b) rute kutaja. Hasil pengujian yang ditunjukkan dengan Gambar 3.2 terdapat perbedaan pembacaan antara voltmeter dengan aplikasi, oleh sebab itu didapatkan persentase kesalahan pembacaan sebagai berikut.

$$\%error = \frac{|\text{aplikasi} - \text{voltmeter}|}{\text{voltmeter}} \times 100$$

$$\%error = \frac{33.33 - 33.33}{33.33} \times 100$$

$$\%error = 1,8$$

Persentase baterai yang tertampil pada aplikasi menggunakan perhitungan dengan persamaan sebagai berikut.

$$\%persentase = \frac{\text{tegangan terbaca di aplikasi} - \text{threshold}}{\text{baterai maksimum} - \text{threshold}} \times 100$$

$$\%persentase = \frac{3.4 - 2.7}{3.6 - 2.7} \times 100$$

$$\%persentase = 88,87$$

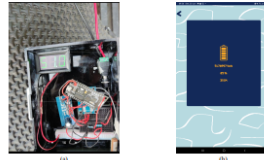
$$\%persentase = 88,87$$

Sisa jarak yang dapat ditempuh dengan sisa baterai tertampil pada aplikasi menggunakan perhitungan dengan persamaan sebagai berikut.

$$\text{sisa jarak} = \frac{\text{tegangan terbaca (volt)} - \text{threshold}}{0,4 \text{ volt/lkm}} \times 100$$

$$\text{sisa jarak} = \frac{3.4 - 2.7}{0,4} \times 100$$

$$\text{sisa jarak} = 33,33 \text{ km}$$



Gambar 3.3 Sisa baterai akhir pada voltmeter (a) dan pada aplikasi EMOT (b) rute kutaja.

Hasil pengujian yang ditunjukkan dengan Gambar 3.3 terdapat perbedaan pembacaan antara voltmeter dengan aplikasi, oleh sebab itu didapatkan persentase kesalahan pembacaan sebagai berikut.

$$\%error = \frac{|\text{aplikasi} - \text{voltmeter}|}{\text{voltmeter}} \times 100$$

$$\%error = \frac{34 - 34}{34} \times 100$$

$$\%error = 1,53$$

Persentase baterai yang tertampil pada aplikasi menggunakan perhitungan dengan persamaan sebagai berikut.

$$\%persentase = \frac{\text{tegangan terbaca di aplikasi} - \text{threshold}}{\text{baterai maksimum} - \text{threshold}} \times 100$$

$$\%persentase = \frac{3.4 - 2.7}{3.6 - 2.7} \times 100$$

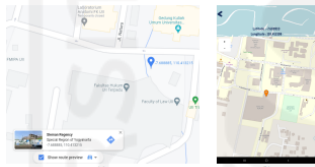
$$\%persentase = 85,2$$

Sisa jarak yang dapat ditempuh dengan sisa baterai tertampil pada aplikasi menggunakan perhitungan dengan persamaan sebagai berikut.

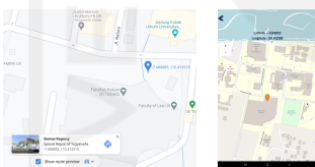
$$\text{sisa jarak} = \frac{\text{tegangan terbaca (volt)} - \text{threshold}}{0,4 \text{ volt/lkm}} \times 100$$

$$\text{sisa jarak} = \frac{3.4 - 2.7}{0,4} \times 100$$

$$\text{sisa jarak} = 31,95 \text{ km}$$



Gambar 3.4 Perbandingan titik lokasi Gedung Fakultas Hukum UI berdasarkan Google Maps (a) dan aplikasi EMOT (b) pada awal rute kutaja



Gambar 3.5 Perbandingan titik lokasi Gedung Fakultas Hukum berdasarkan Google Maps (a) dan aplikasi EMOT (b) pada akhir rute kutaja

Berdasarkan hasil pengujian titik lokasi pada Gedung Fakultas Hukum UI yang ditunjukkan pada Gambar 3.4 dan Gambar 3.5 dapat diketahui nilai kesalahan persentase kesalahan sebagai berikut.

$$\%error = \frac{|\text{aplikasi} - \text{google}|}{\text{google}} \times 100$$

$$\%error = \frac{31.95 - 31.95}{31.95} \times 100$$

$$\%error = 0,0003$$

$$\%error = \frac{|\text{aplikasi} - \text{google}|}{\text{google}} \times 100$$

$$\%error = \frac{31.95 - 31.95}{31.95} \times 100$$

$$\%error = 0,000015$$

4. Pengujian kecepatan dilakukan dari gerbang utama UIH dan berakhir di lokasi yang sama dengan jarak tempuh sebesar 1,750 m.



Gambar 4.1 Ratu Pelemban Kutaja



Gambar 4.2 Sisa baterai awal pada voltmeter (a) dan pada aplikasi EMOT (b) rute kecupat. Hasil pengujian yang ditunjukkan dengan Gambar 4.2 terdapat perbedaan pembacaan antara voltmeter dengan aplikasi, oleh sebab itu didapatkan persentase kesalahan pembacaan sebagai berikut.

$$\%error = \frac{|\text{aplikasi} - \text{voltmeter}|}{\text{voltmeter}} \times 100$$

$$\%error = \frac{31.4 - 31.4}{31.4} \times 100$$

$$\%error = 1,38$$

Persentase baterai yang tertampil pada aplikasi menggunakan perhitungan dengan persamaan sebagai berikut.

$$\%persentase = \frac{\text{tegangan terbaca di aplikasi} - \text{threshold}}{\text{baterai maksimum} - \text{threshold}} \times 100$$

$$\%persentase = \frac{3.4 - 2.7}{3.6 - 2.7} \times 100$$

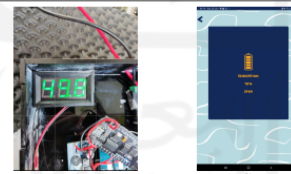
$$\%persentase = 82,67$$

Sisa jarak yang dapat ditempuh dengan sisa baterai tertampil pada aplikasi menggunakan perhitungan dengan persamaan sebagai berikut.

$$\text{sisa jarak} = \frac{\text{tegangan terbaca (volt)} - \text{threshold}}{0,4 \text{ volt/lkm}} \times 100$$

$$\text{sisa jarak} = \frac{3.4 - 2.7}{0,4} \times 100$$

$$\text{sisa jarak} = 31 \text{ km}$$



Gambar 4.3 Sisa baterai akhir pada voltmeter (a) dan pada aplikasi EMOT (b) rute kecupat.

Hasil pengujian yang ditunjukkan dengan Gambar 4.3 terdapat perbedaan pembacaan antara voltmeter dengan aplikasi, oleh sebab itu didapatkan persentase kesalahan pembacaan sebagai berikut.

$$\%error = \frac{|\text{aplikasi} - \text{voltmeter}|}{\text{voltmeter}} \times 100$$

$$\%error = \frac{29.5 - 29.5}{29.5} \times 100$$

$$\%error = 2,0$$

Persentase baterai yang tertampil pada aplikasi menggunakan perhitungan dengan persamaan sebagai berikut.

$$\%persentase = \frac{\text{tegangan terbaca di aplikasi} - \text{threshold}}{\text{baterai maksimum} - \text{threshold}} \times 100$$

$$\%persentase = \frac{3.08 - 2.7}{3.6 - 2.7} \times 100$$

$$\%persentase = 78,67$$

Sisa jarak yang dapat ditempuh dengan sisa baterai tertampil pada aplikasi menggunakan perhitungan dengan persamaan sebagai berikut.

$$\text{sisa jarak} = \frac{\text{tegangan terbaca (volt)} - \text{threshold}}{0,4 \text{ volt/lkm}} \times 100$$

$$\text{sisa jarak} = \frac{3.08 - 2.7}{0,4} \times 100$$

$$\text{sisa jarak} = 29,5 \text{ km}$$



Gambar 4.4 Perbandingan titik lokasi Gedung Uti berdasarkan Google Maps (a) dan aplikasi EMOT (b) pada awal rute kecampur.



Gambar 4.1 Perbandingan titik lokasi Gedung Uti berdasarkan Google Maps (a) dan aplikasi EMOT (b) pada akhir rute kecampur.

Berdasarkan hasil pengujian titik lokasi pada Gedung Uti yang ditunjukkan pada Gambar 4.4 dan Gambar 4.5 dapat diketahui nilai kesalahan persentase kesalahan sebagai berikut.

$$\% \text{error} = \frac{\text{titik lokasi awal} - \text{titik lokasi akhir}}{\text{titik lokasi awal}} \times 100$$

$$\% \text{error} = \frac{-2,687711 - (-2,687676)}{-2,687676} \times 100 \quad (4.7)$$

$$\% \text{error} = \frac{38,74 - 38,84}{38,74} \times 100$$

$$\% \text{error} = -1,99$$

Persentase baterai yang tertampil pada aplikasi menggunakan perhitungan dengan persamaan sebagai berikut.

$$\% \text{persentase} = \frac{\text{tegangan terbaca di aplikasi} - \text{tegangan baterai maksimal}}{\text{tegangan baterai maksimal}} \times 100 \quad (5.2)$$

$$\% \text{persentase} = \frac{3,4 - 3,4}{3,4} \times 100$$

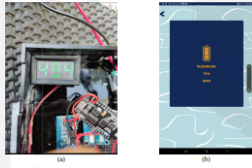
$$\% \text{persentase} = 78,27$$

Sisa jarak yang dapat ditempuh dengan sisa baterai tertampil pada aplikasi menggunakan perhitungan dengan persamaan sebagai berikut.

$$\text{sisa jarak} = \frac{\text{tegangan terbaca (volt)} - \text{tegangan baterai minimal}}{\text{tegangan baterai minimal}} \times 100 \quad (5.3)$$

$$\text{sisa jarak} = \frac{3,4 - 3,4}{3,4} \times 100$$

$$\text{sisa jarak} = 29,35 \text{ km}$$



Gambar 5.3 Sisa baterai akhir pada voltmeter (a) dan pada aplikasi EMOT (b) rute kelima. Hasil pengujian yang ditunjukkan dengan Gambar 5.3 didapat perbedaan pembacaan antara voltmeter dengan aplikasi, oleh sebab itu didapatkan persentase kesalahan pembacaan sebagai berikut.

$$\% \text{error} = \frac{\text{tegangan terbaca} - \text{tegangan baterai maksimal}}{\text{tegangan baterai maksimal}} \times 100 \quad (5.4)$$

$$\% \text{error} = \frac{3,4 - 3,4}{3,4} \times 100$$

$$\% \text{error} = 1,72$$

Persentase baterai yang tertampil pada aplikasi menggunakan perhitungan dengan persamaan sebagai berikut.

$$\% \text{persentase} = \frac{\text{tegangan terbaca di aplikasi} - \text{tegangan baterai maksimal}}{\text{tegangan baterai maksimal}} \times 100 \quad (5.5)$$

Gambar 5.5 Perbandingan titik lokasi Gedung Perpustakaan Pusat Uti berdasarkan Google Maps (a) dan aplikasi EMOT (b) pada akhir rute kelima.

Berdasarkan hasil pengujian titik lokasi pada Gedung Perpustakaan Uti yang ditunjukkan pada Gambar 5.4 dan Gambar 5.5 dapat diketahui nilai kesalahan persentase kesalahan sebagai berikut.

$$\% \text{error} = \frac{\text{titik lokasi awal rute} - \text{titik lokasi akhir rute}}{\text{titik lokasi awal rute}} \times 100 \quad (5.7)$$

$$\% \text{error} = \frac{-7,688477 - (-7,688417)}{-7,688417} \times 100$$

$$\% \text{error} = 0,00011$$

$$\% \text{error} = \frac{\text{titik lokasi awal rute} - \text{titik lokasi akhir rute}}{\text{titik lokasi awal rute}} \times 100$$

$$\% \text{error} = \frac{-7,688442 - (-7,688409)}{-7,688409} \times 100$$

$$\% \text{error} = 0,00007$$

$$\% \text{error} = \frac{\text{titik lokasi awal rute} - \text{titik lokasi akhir rute}}{\text{titik lokasi awal rute}} \times 100$$

$$\% \text{error} = \frac{110,415496 - 110,415420}{110,415420} \times 100$$

$$\% \text{error} = 0,000069$$

$$\% \text{error} = \frac{\text{titik lokasi awal rute} - \text{titik lokasi akhir rute}}{\text{titik lokasi awal rute}} \times 100$$

$$\% \text{error} = \frac{110,415554 - 110,415520}{110,415520} \times 100$$

$$\% \text{error} = 0,000103$$

Tabel 1. Hasil Pengujian Monitoring Baterai

Rate	Tegangan Real (Volt)		Tegangan yang Terbaca pada Aplikasi (Volt)		Kesalahan Pembacaan (%)	
	Awal	Akhir	Awal	Akhir	Awal	Akhir
1	52,2	51,8	53,16	52,72	1,83	1,78
2	51,8	51,4	52,66	52,33	1,66	1,8
3	51,4	51,0	52,33	51,78	1,8	1,53
4	50,7	49,8	51,40	50,80	1,38	2
5	49,8	49,4	50,74	50,25	1,89	1,72

$$\% \text{error} = 0,0007$$

Kesalahan nilai longitude awal dan akhir rute.

$$\% \text{error} = \frac{\text{titik lokasi awal} - \text{titik lokasi akhir}}{\text{titik lokasi awal}} \times 100$$

$$\% \text{error} = \frac{110,415557 - 110,415520}{110,415520} \times 100$$

$$\% \text{error} = 0,00008$$

5. Pengujian kelima dilakukan dari Gedung Perpustakaan Uti dan berakhir di lokasi yang sama dengan jarak tempuh sebesar 1,024 m.



Gambar 5.1 Rute Perpustakaan Kelima



Gambar 5.2 Sisa baterai awal pada voltmeter (a) dan pada aplikasi EMOT (b) rute kelima.

Hasil pengujian yang ditunjukkan dengan Gambar 5.2 didapat perbedaan pembacaan antara voltmeter dengan aplikasi, oleh sebab itu didapatkan persentase kesalahan pembacaan sebagai berikut.

$$\% \text{error} = \frac{\text{tegangan terbaca} - \text{tegangan baterai maksimal}}{\text{tegangan baterai maksimal}} \times 100 \quad (5.1)$$

$$\% \text{persentase} = \frac{\text{tegangan terbaca} - \text{tegangan baterai maksimal}}{\text{tegangan baterai maksimal}} \times 100$$

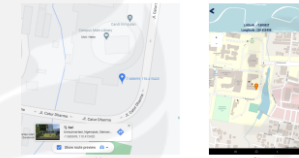
$$\% \text{persentase} = 75$$

Sisa jarak yang dapat ditempuh dengan sisa baterai tertampil pada aplikasi menggunakan perhitungan dengan persamaan sebagai berikut.

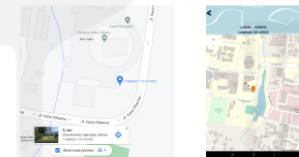
$$\text{sisa jarak} = \frac{\text{tegangan terbaca (volt)} - \text{tegangan baterai minimal}}{\text{tegangan baterai minimal}} \times 100 \quad (5.6)$$

$$\text{sisa jarak} = \frac{3,4 - 3,4}{3,4} \times 100$$

$$\text{sisa jarak} = 28,125 \text{ km}$$



Gambar 5.4 Perbandingan titik lokasi Gedung Perpustakaan Pusat Uti berdasarkan Google Maps (a) dan aplikasi EMOT (b) pada awal rute kelima.



Hasil pengujian yang tertera pada Tabel 1, maka dapat disimpulkan bahwa pembacaan tegangan baterai pada aplikasi yang dirancang memiliki kesalahan pembacaan maksimal 2% dibandingkan dengan tegangan real pada baterai. Oleh sebab itu, sistem aplikasi yang dirancang berjalan dengan baik dan memenuhi spesifikasi sistem dengan batas kesalahan pembacaan maksimal yang diharapkan adalah 2%.

Tabel 2. Hasil Pengujian Lokasi Lokasi

Rate	Titik lokasi sebenarnya				Titik lokasi pada aplikasi EMOT				Kesalahan Pembacaan (%)			
	Awal (latitudo, longitude)	Akhir (latitudo, longitude)	Awal (latitudo, longitude)	Akhir (latitudo, longitude)	Awal (latitudo, longitude)	Akhir (latitudo, longitude)	Latitudo	Longitude	Latitudo	Longitude		
1	-7,688477 110,409728	-7,688477 110,409726	-7,688436 110,409733	-7,688532 110,409772	0,0007	0,00006	0,0007	0,00042				
2	-7,688706 110,410192	-7,688726 110,411108	-7,688714 110,410172	-7,688292 110,413271	0,0002	0,000009	0,0022	0,00015				
3	-7,688885 110,413215	-7,688885 110,413215	-7,688910 110,413198	-7,688910 110,413198	0,0003	0,000115	0,0003	0,00011				
4	-7,687879 110,413030	-7,687879 110,413030	-7,687731 110,413042	-7,687731 110,413042	0,0007	0,00008	0,0007	0,00008				
5	-7,688609 110,415420	-7,688609 110,415420	-7,688617 110,415496	-7,688642 110,415334	0,0011	0,000069	0,0007	0,000103				

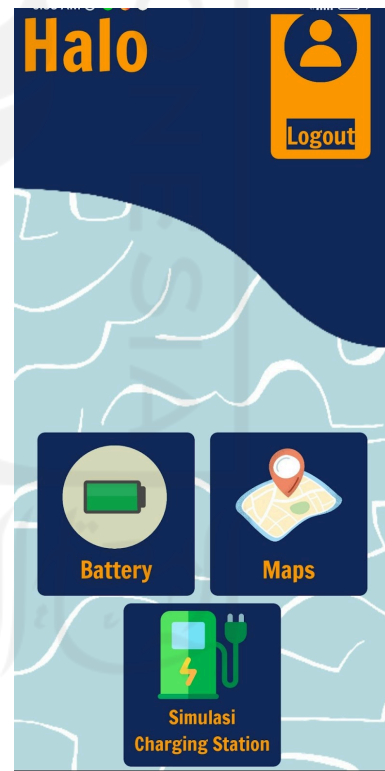
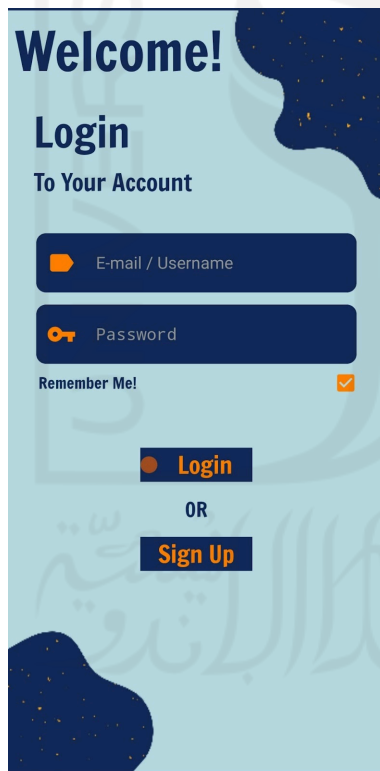
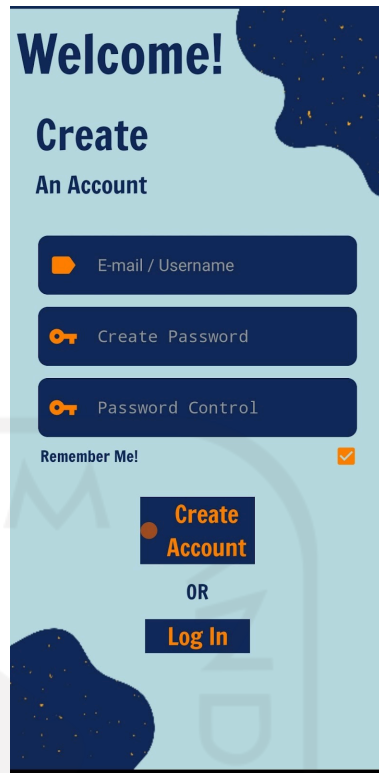
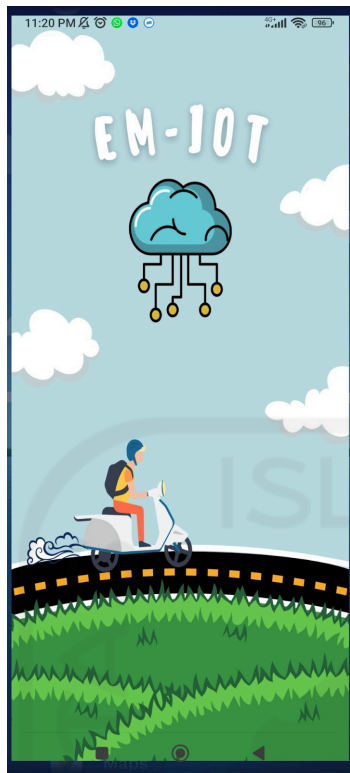
Hasil pengujian yang tertera pada Tabel 2, dapat disimpulkan bahwa pembacaan titik lokasi (latitudo dan longitude) pada aplikasi yang dirancang memiliki kesalahan pembacaan maksimal sebesar 0,0011% dibandingkan dengan titik lokasi sebenarnya. Oleh sebab itu, sistem aplikasi yang dirancang berjalan dengan baik dan memenuhi spesifikasi sistem dengan batas kesalahan pembacaan maksimal yang diharapkan adalah 5%.

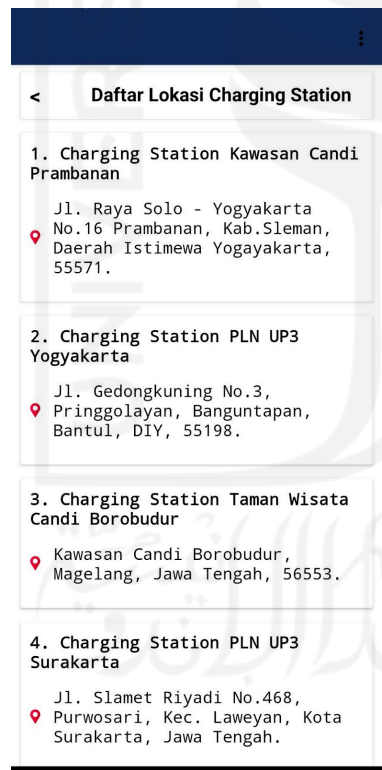
Log / Catatan Aktivitas (meliputi perencanaan, aktivitas, tugas, dan capaiannya)

1. Pembuatan hardware
2. Membuat program dengan menggunakan Arduino IDE
3. Membutuhkan User Interface aplikasi yang dirancang
4. Melakukan perbaikan sistem di area Kampus Terpadu Uti



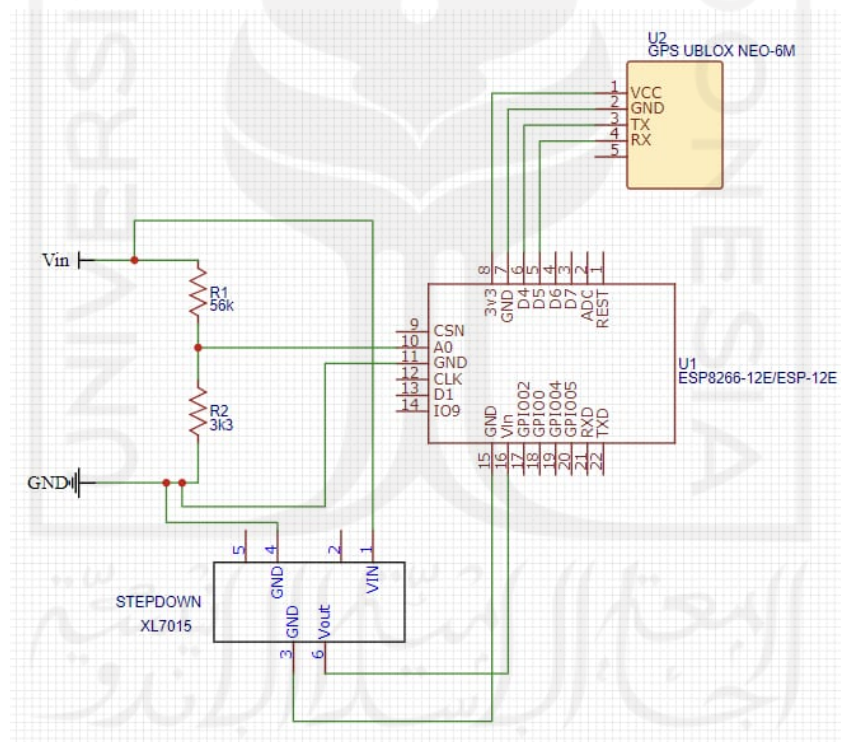
- Desain model/produk/sistem termasuk aplikasi jika ada







- Skematik elektronik keseluruhan



- Kode program alat

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>
#include "ThingSpeak.h"
#include <ESP8266WiFi.h>
```

```
static const int RXPin = 14, TXPin = 2;
static const uint32_t GPSBaud = 9600;
```

```

// repace your wifi username and password
const char* ssid = "GTFOFMH";
const char* password = "QWERTY123";

unsigned long myChannelNumber = 1765778;
const char * myWriteAPIKey = "EVB5ITDKOWRMFIK4";

int analogPin = A0;
float Vmodul = 0.0;
float hasil = 0.0;
float R1 = 56000.0;
float R2 = 3300.0;
float nilai;
int persen = 0;
int sisa = 0;
float hasilfix = 0;

// The TinyGPS++ object
TinyGPSPlus gps;
WiFiClient client;

// The serial connection to the GPS device
SoftwareSerial ss(TXPin, RXPin);

void setup()
{
  Serial.begin(115200);
  ss.begin(GPSBaud);

  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(15000);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
  Serial.print("Netmask: ");
  Serial.println(WiFi.subnetMask());
  Serial.print("Gateway: ");
  Serial.println(WiFi.gatewayIP());
  ThingSpeak.begin(client);
}

void baterai ()
{
  nilai = analogRead(analogPin);

```

```

Vmodul = (nilai * 3.3) / 1024;
hasil = Vmodul / (R2/(R1+R2));
hasilfix = (hasil*54)/57;
  if (hasilfix >= 39){
    persen = ((hasilfix-39)/15)*100;
    sisa = (hasilfix - 39)/0.4;
  }
  else {
    persen = 0;
    sisa = 0;
  }
}

void loop()
{
  // This sketch displays information every time a new sentence is correctly encoded.
  while (ss.available() > 0)
    if (gps.encode(ss.read()))
      baterai();
      displayInfo();

  if (millis() > 5000 && gps.charsProcessed() < 10)
  {
    Serial.println(F("No GPS detected: check wiring."));
    while(true);
  }
}

void displayInfo()
{
  Serial.print("\t value = ");
  Serial.print(nilai);
  Serial.print("\t Tegangan Pembagi = ");
  Serial.print(Vmodul,2);
  Serial.println("volt");
  Serial.print("\t Hasil pengukuran asli = ");
  Serial.print(hasilfix,1);
  Serial.println("volt");
  Serial.print("\t Persentase baterai = ");
  Serial.print(persen);
  Serial.println("%");
  Serial.print("\t Sisa jarak = ");
  Serial.print(sisa);
  Serial.println("km");
  Serial.print(F("Location: "));
  if (gps.location.isValid())
  {

    double latitude = (gps.location.lat());
    double longitude = (gps.location.lng());

```

```

String latbuf;
latbuf += (String(latitude, 6));
Serial.println(latbuf);

String lonbuf;
lonbuf += (String(longitude, 6));
Serial.println(lonbuf);

ThingSpeak.setField(1, hasilfix);
ThingSpeak.setField(2, persen);
ThingSpeak.setField(4, sisa);
ThingSpeak.setField(5, latbuf);
ThingSpeak.setField(6, lonbuf);
ThingSpeak.writeFields(myChannelNumber, myWriteAPIKey);
delay(5000);
}
else
{
Serial.print(F("INVALID"));
}

Serial.print(F(" Date/Time: "));
if (gps.date.isValid())
{
Serial.print(gps.date.month());
Serial.print(F("/"));
Serial.print(gps.date.day());
Serial.print(F("/"));
Serial.print(gps.date.year());
}
else
{
Serial.print(F("INVALID"));
}

Serial.print(F(" "));
if (gps.time.isValid())
{
if (gps.time.hour() < 10) Serial.print(F("0"));
Serial.print(gps.time.hour());
Serial.print(F(":"));
if (gps.time.minute() < 10) Serial.print(F("0"));
Serial.print(gps.time.minute());
Serial.print(F(":"));
if (gps.time.second() < 10) Serial.print(F("0"));
Serial.print(gps.time.second());
Serial.print(F("."));
if (gps.time.centisecond() < 10) Serial.print(F("0"));
Serial.print(gps.time.centisecond());
}
}

```

```

}
else
{
  Serial.print(F("INVALID"));
}

Serial.println();
}

```

▪ Kode program kodular

```

when Label_signin .Click
do open another screen screenName SignIn

when Text_Box_email .Got Focus
do set Label_email .Visible to true

when Text_Box_pass .Got Focus
do set Label_pass .Visible to true

when Text_Box_email .Lost Focus
do set Label_email .Visible to false

when Text_Box_pass .Lost Focus
do set Label_pass .Visible to false

when Label_email .Click
do call Screen1 .Hide Keyboard

when Label_pass .Click
do call Screen1 .Hide Keyboard

when Label_login .Click
do if not contains text Text_Box_email .Text
    piece '@'
then call Notifier1 .Show Alert
    notice 'Your e-mail address is not contains '@'
else call Firebase_Authentication1 .Email Password Login
    email Text_Box_email .Text
    password Text_Box_pass .Text
call Notifier1 .Show Progress Dialog
    message 'We are entering to your account.'
    title 'Please Wait'
set Clock1 .Timer Enabled to true

when Screen1 .Initialize
do call Label_login .Set Animation Style
    style DoubleBounce
    position Left
    size 50
    color orange

when Clock1 .Timer
do call Notifier1 .Dismiss Progress Dialog
open another screen with start value screenName home
startValue split text Text_Box_email .Text
    at '@'

when Firebase_Authentication1 .Login Success
provider user ID name email phone Number profile Picture
do open another screen screenName home

when Label_login .Click
do open another screen screenName Screen1

when Text_Box_email .Got Focus
do set Label_email .Visible to true

when Text_Box_pass .Got Focus
do set Label_pass .Visible to true

when Text_Box_email .Lost Focus
do set Label_email .Visible to false

when Text_Box_pass .Lost Focus
do set Label_pass .Visible to false

when Label_email .Click
do call SignIn .Hide Keyboard

when Label_pass .Click
do call SignIn .Hide Keyboard

when Firebase_Authentication1 .Sign Up Success
provider user ID name email phone Number profile Picture
do open another screen screenName Screen1

when Label_createacc .Click
do if not contains text Text_Box_email .Text
    piece '@'
then call Notifier1 .Show Alert
    notice 'Your e-mail address is not contains '@'
else if not Text_Box_pass .Text == Text_Box_pass_cek .Text
then call Notifier1 .Show Alert
    notice 'Your passwords is not same.'
else call Firebase_Authentication1 .Email Password Sign Up
    email Text_Box_email .Text
    password Text_Box_pass .Text
call Notifier1 .Show Progress Dialog
    message 'Please Wait'
    title 'We are creating your account.'
set Clock1 .Timer Enabled to true

when SignIn .Initialize
do call Label_createacc .Set Animation Style
    style DoubleBounce
    position Left
    size 50
    color orange

```



```

when Label_logout .Click
do call Firebase_Authentication1 .Logout
  open another screen screenName " Screen1 "

```

```

when Card_View_maps .Click
do open another screen screenName maps

```

```

when Card_View_batre .Click
do open another screen screenName batre

```

```

when Image_maps .Click
do open another screen screenName maps

```

```

when Label_batre .Click
do open another screen screenName batre

```

```

when Label_maps .Click
do open another screen screenName maps

```

```

when Image_batre .Click
do open another screen screenName batre

```

```

when Label1 .Click
do open another screen screenName home

```

```

initialize global tegangan to 0

```

```

initialize global persen to 0

```

```

initialize global sisa to 0

```

```

initialize global link to " https://api.thingspeak.com/channels/1765778/feed... "

```

```

initialize global read_API to " N1HEUYCA3I4M7XRC "

```

```

to baca
do set Web1 . URL to join get global link get global read_API
  call Web1 .Get

```

```

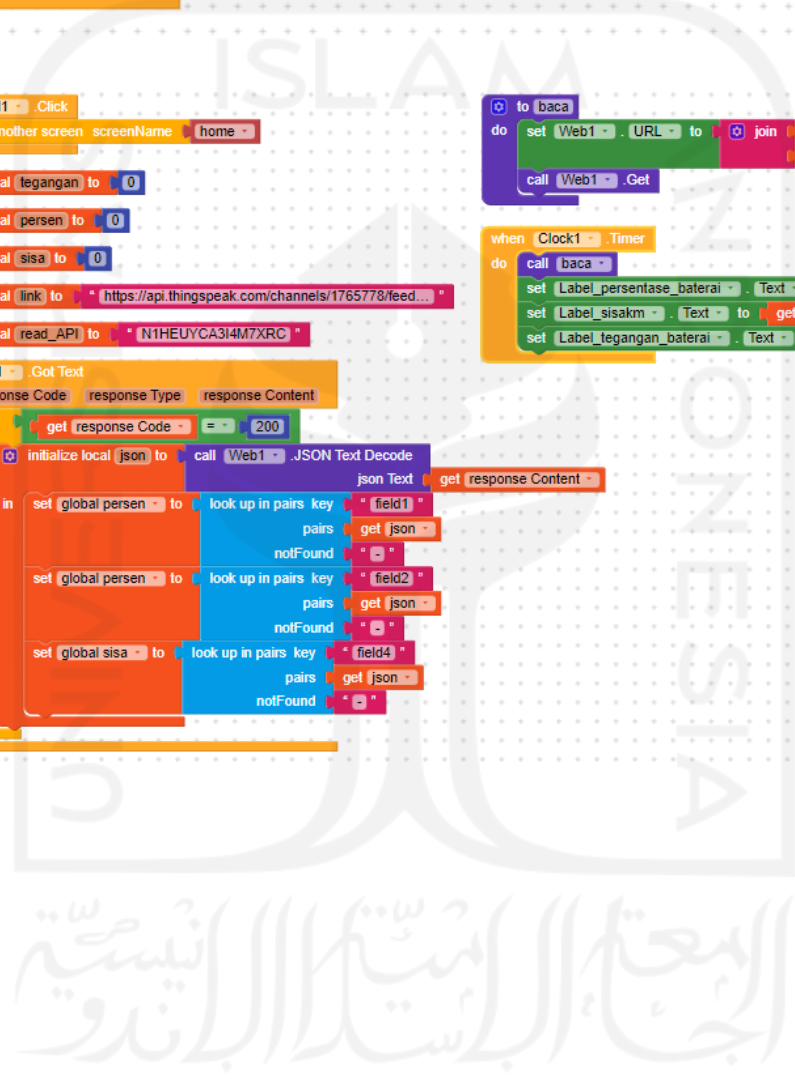
when Clock1 .Timer
do call baca
  set Label_persentase_baterai .Text to get global persen
  set Label_sisakm .Text to get global sisa
  set Label_tegangan_baterai .Text to get global tegangan

```

```

when Web1 .Got Text
url response Code response Type response Content
do if get response Code = 200
  then initialize local json to call Web1 .JSON Text Decode
    in set global persen to look up in pairs key " field1 "
      pairs get json
      notFound " "
      set global persen to look up in pairs key " field2 "
        pairs get json
        notFound " "
      set global sisa to look up in pairs key " field4 "
        pairs get json
        notFound " "

```



```

when Clock2.Timer
do
  if get global persen = 34
  then
    call Notifier1.Show Image Dialog
    id 1
    title "Baterai Habis!!"
    image "status.png"
    button Text "Segera Charge Baterai!!"
  else if get global persen = 50
  then
    call Notifier1.Show Image Dialog
    id 2
    title "Baterai Tersisa 50%"
    image "half-battery.png"
    button Text "Baterai Tersisa 50%"
  else if get global persen = 100
  then
    call Notifier1.Show Image Dialog
    id 3
    title "Baterai Penuh!! 100%"
    image "status.png"
    button Text "Baterai Penuh!! 100%"
  
```

```

when Label1.Click
do
  open another screen screenName "home"

  initialize global lat to 0
  initialize global lon to 0

  initialize global link to "https://api.thingspeak.com/channels/1765778/feed..."
  initialize global read_API to "N1HEUYCA314M7XRC"

  to koneksi
  do
    set Web1.URL to join get global link
    get global read_API
    call Web1.Get
  
```

```

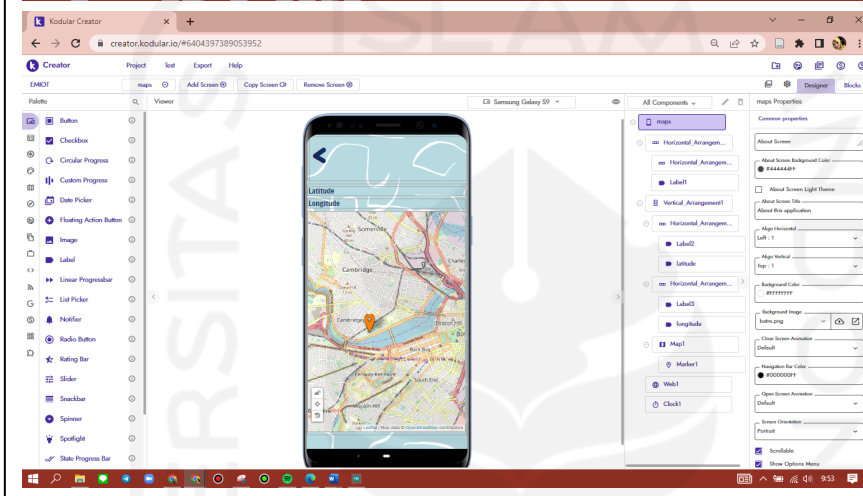
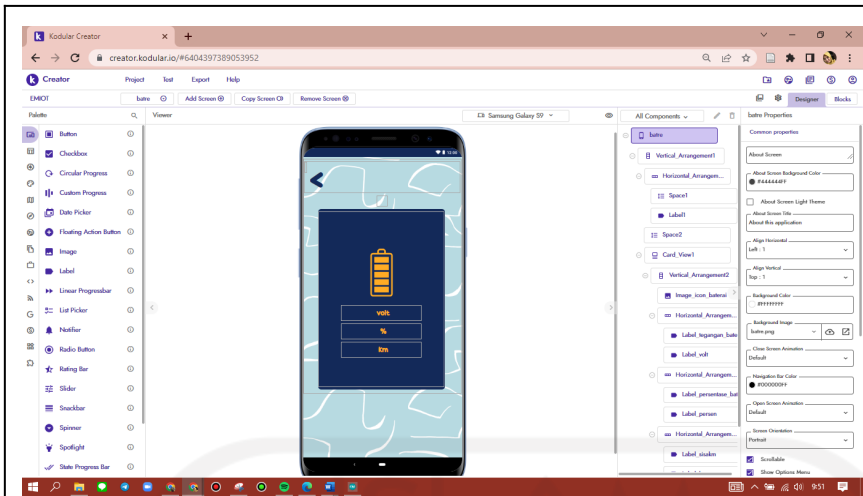
when Clock1.Timer
do
  call koneksi
  set latitude.Text to get global lat
  set longitude.Text to get global lon
  set Map1.Center From String to join get global lat
  get global lon
  set Marker1.Latitude to get global lat
  set Marker1.Longitude to get global lon

```

الجامعة الإسلامية
الاستد الاندو

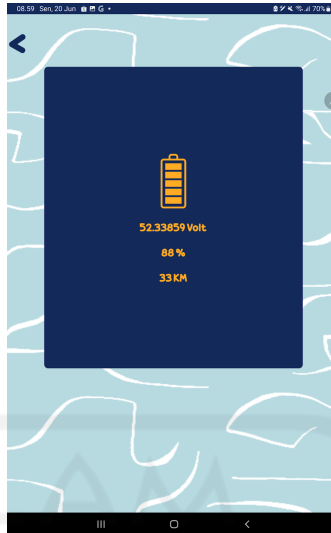
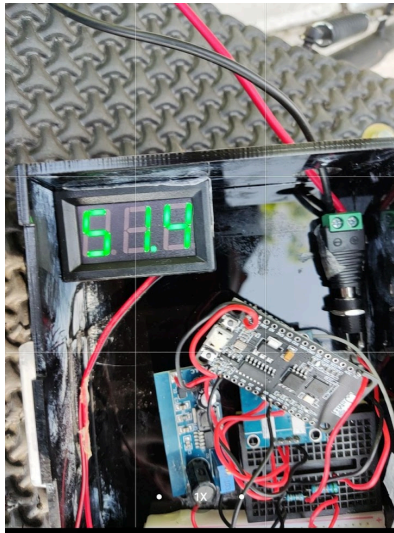
- Dokumentasi proses pengerjaan

Gambar	Keterangan
	<p>Pembuatan desain UI aplikasi pada Kodular</p>

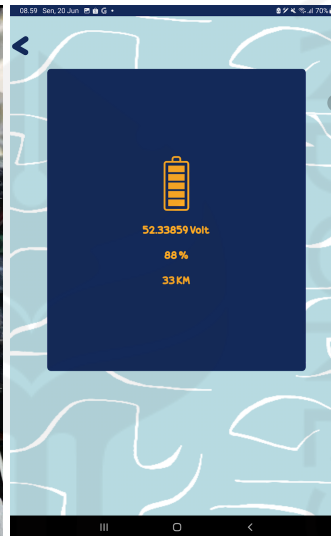
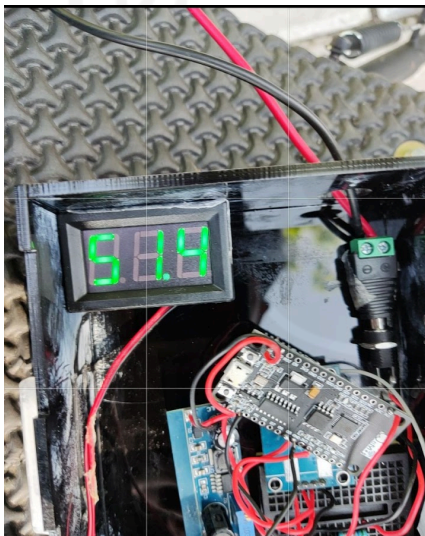


Pengukuran tegangan baterai sebelum uji coba

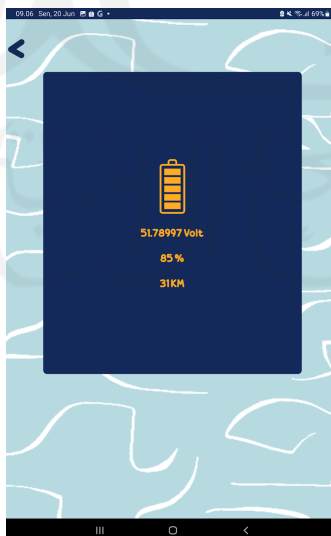
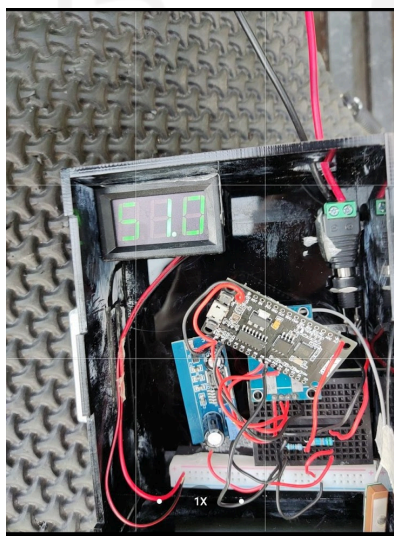
		<p>Hasil uji coba sisa baterai awal rute 1</p>
		<p>Hasil uji coba sisa baterai akhir rute 1</p>
		<p>Hasil uji coba sisa baterai awal rute 2</p>



Hasil uji coba sisa baterai akhir rute 2

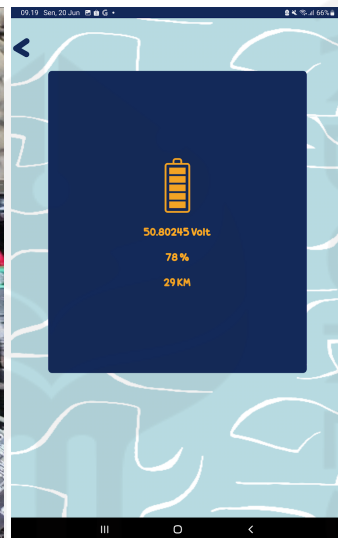
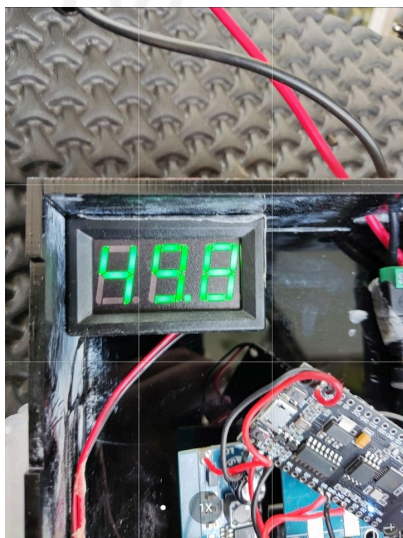
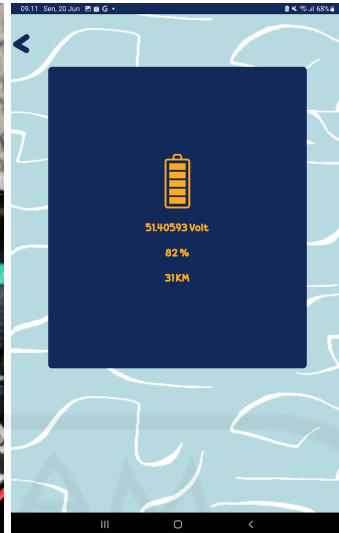
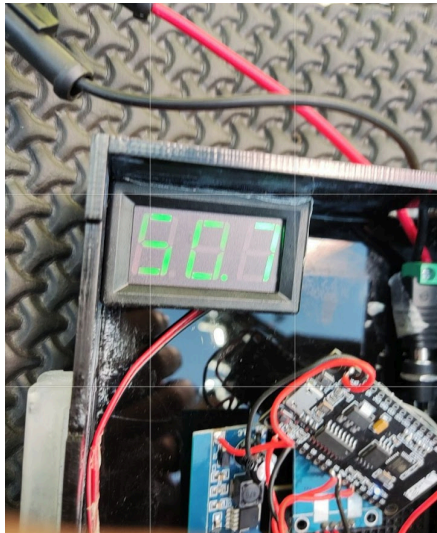


Hasil uji coba sisa baterai awal rute 3

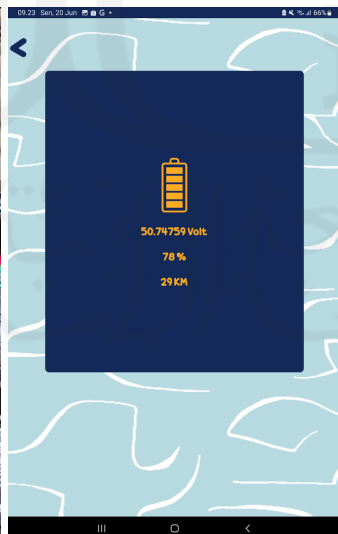
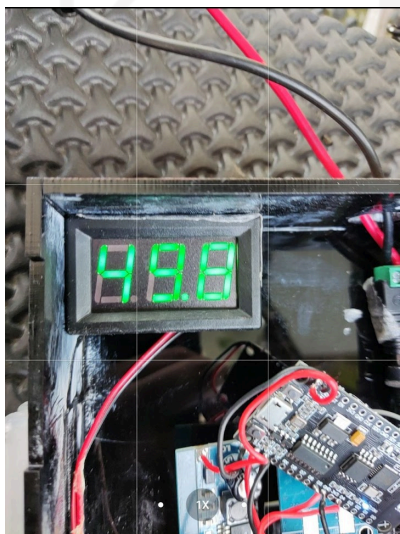


Hasil uji coba sisa baterai akhir rute 3

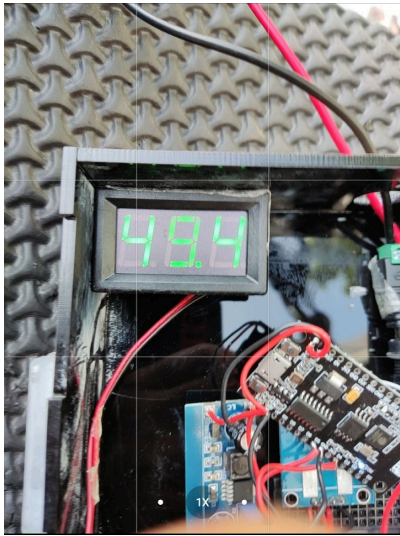
Hasil uji coba sisa baterai awal rute 4



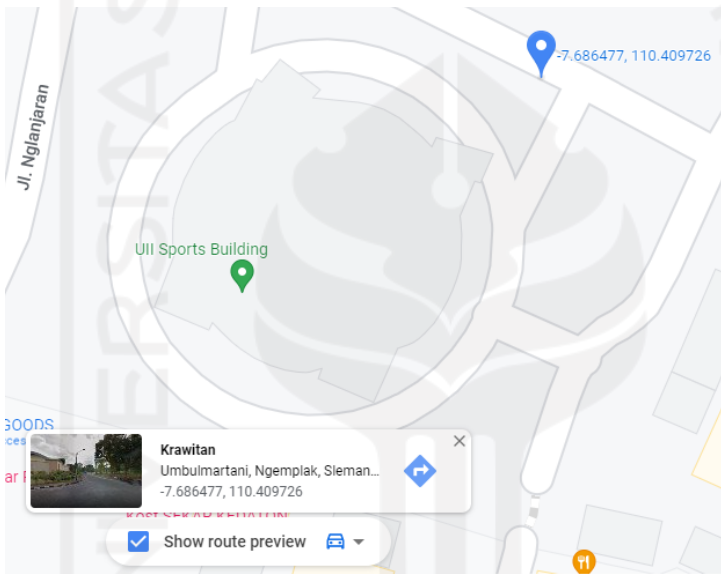
Hasil uji coba sisa baterai akhir rute 4



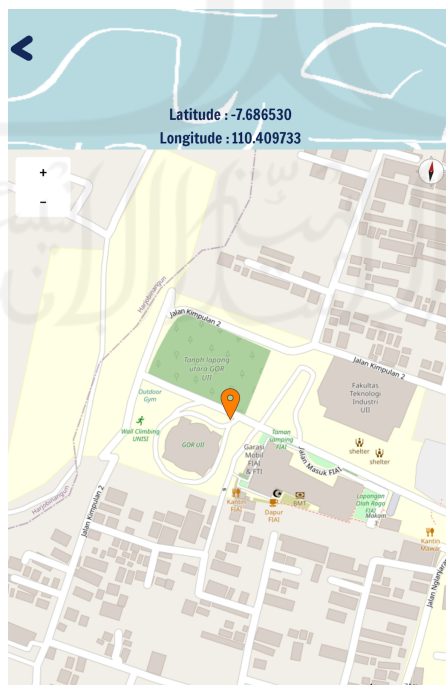
Hasil uji coba sisa baterai awal rute 5

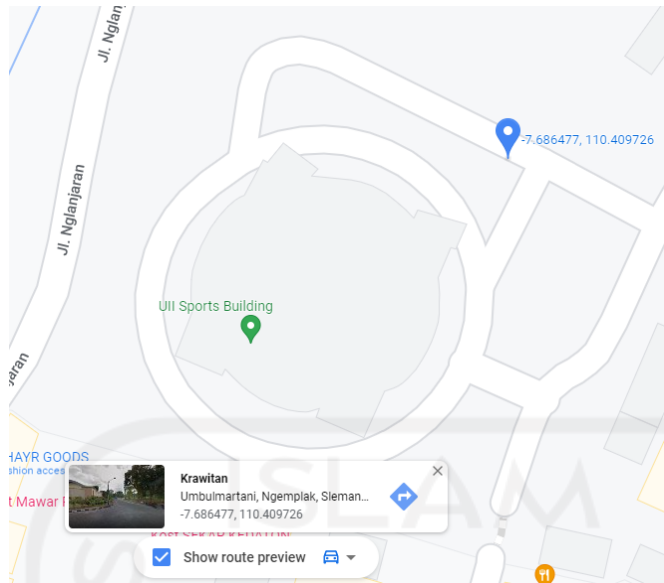


Hasil uji coba sisa baterai akhir rute 5



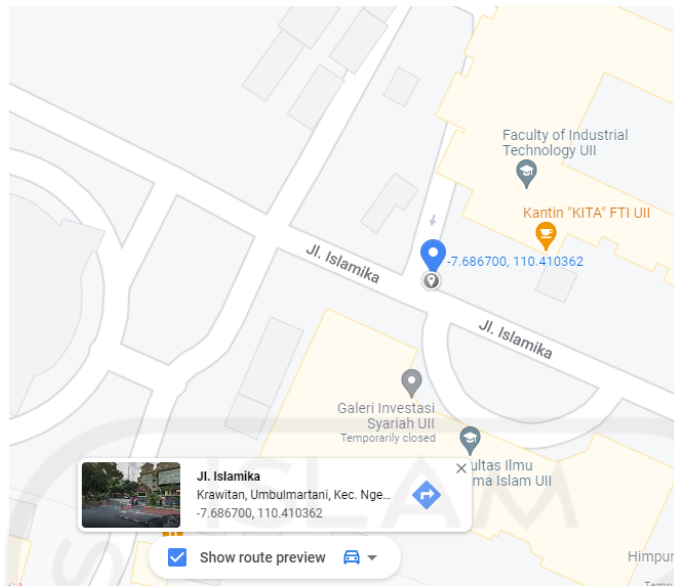
Hasil uji coba lokasi terkini awal rute 1





Hasil uji coba lokasi terkini akhir rute 1





Hasil uji coba lokasi terkini awal rute 2



الجامعة الإسلامية
الاستد الاندو

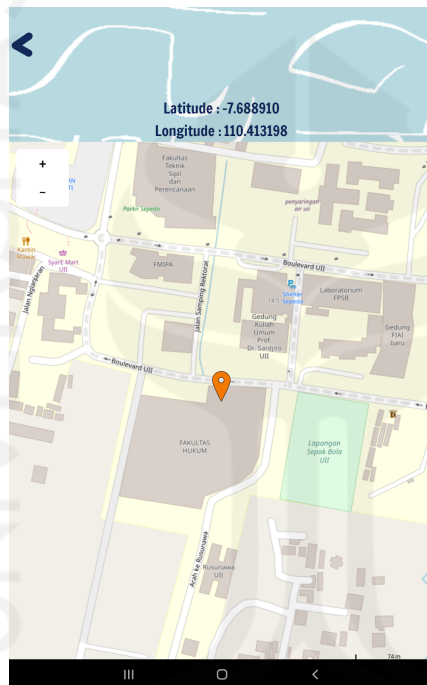
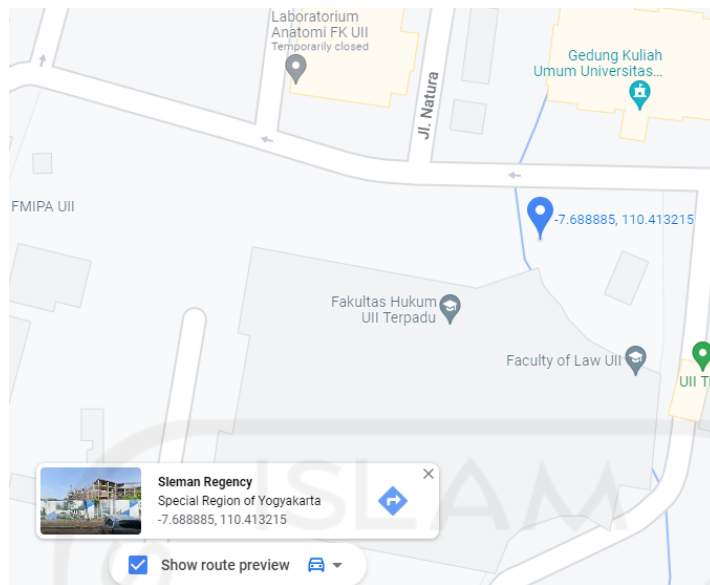


Hasil uji coba lokasi terkini akhir rute 2



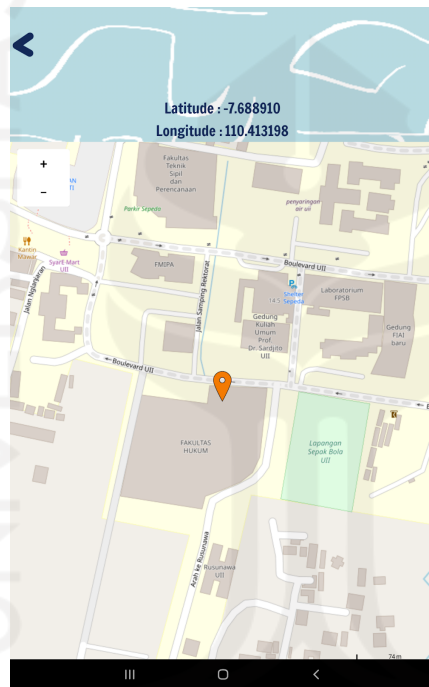
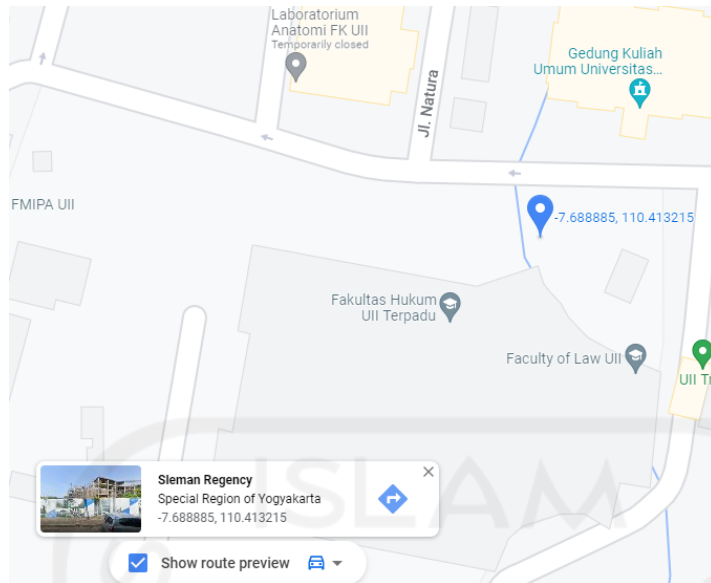
الجامعة الإسلامية
الاستدائات

Hasil uji coba lokasi terkini awal rute 3



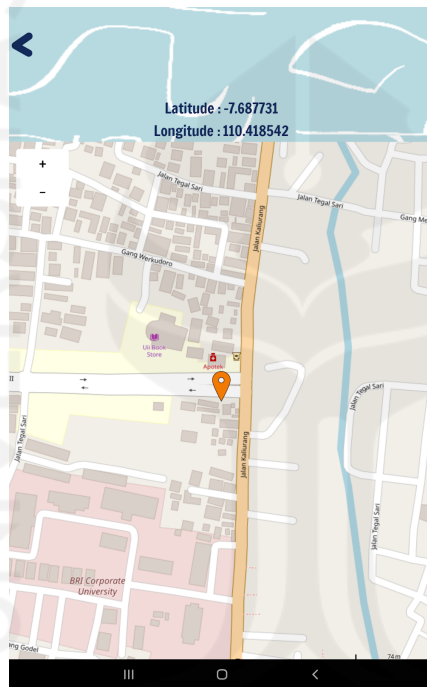
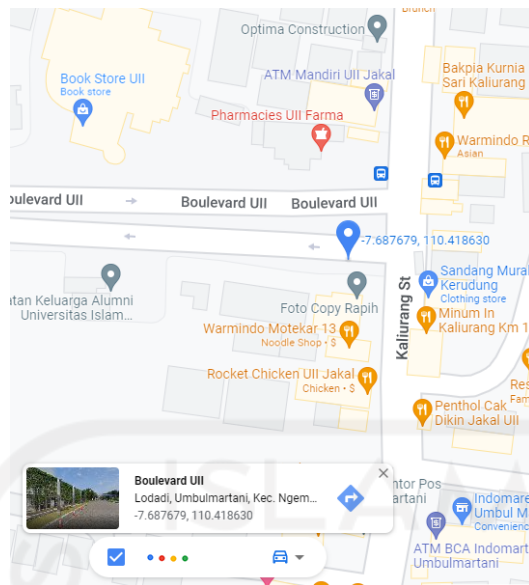
الجامعة الإسلامية
الاستدراكية

Hasil uji coba lokasi terkini akhir rute 3



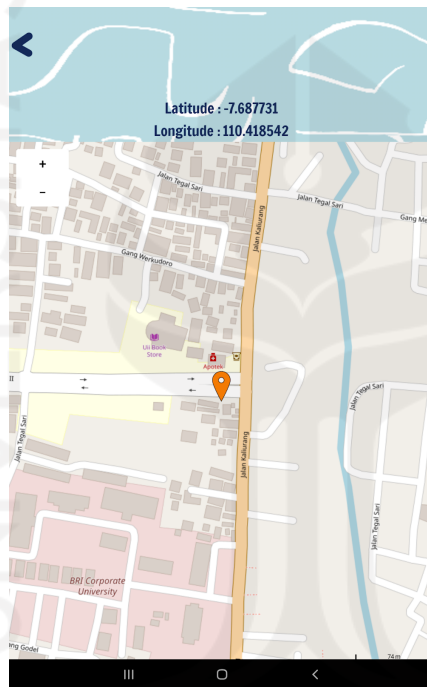
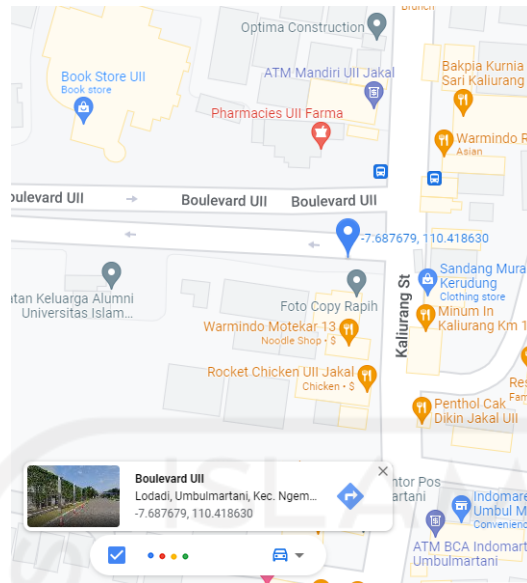
جامعة البستان الاندونيسية

Hasil uji coba lokasi terkini awal rute 4



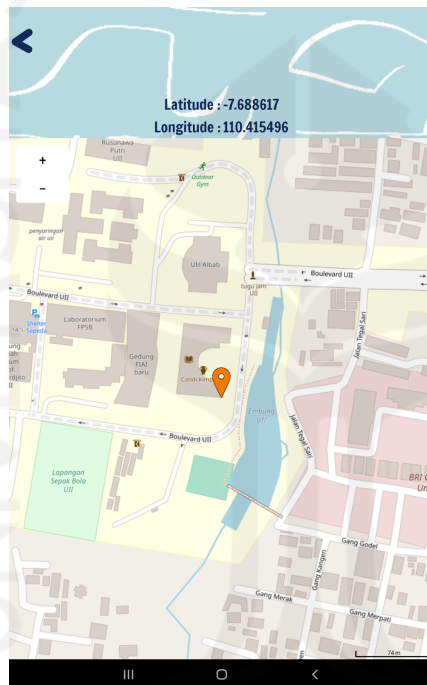
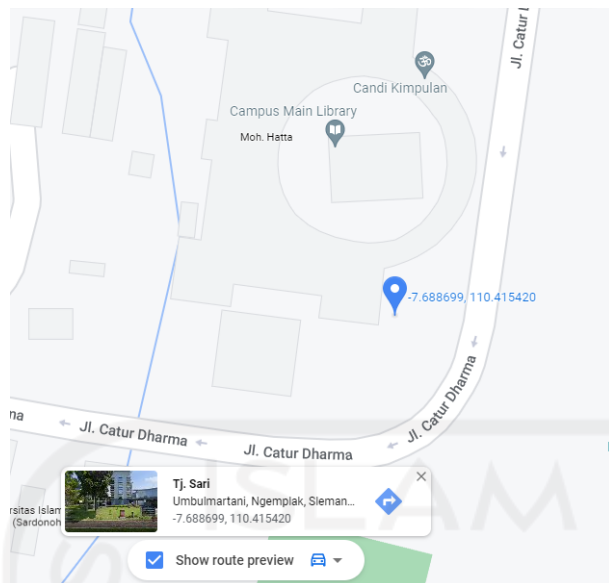
الجامعة الإسلامية
الاستد الاندو

Hasil uji coba lokasi terkini akhir rute 4



الجامعة الإسلامية
الاستد الاندو

Hasil uji coba lokasi terkini awal rute 5



الجامعة الإسلامية
الاستد الاندو



Hasil uji coba lokasi terkini akhir rute 5

▪ Dokumentasi keuangan

No	Jenis Pengeluaran	Kuantitas	Total Harga
1	Module GPS APM 2.5 NEO-6M	1 pcs	Rp 65.000,-
2	Jasa Cetak Case	1 pcs	Rp 35.000,-
3	NodeMCU ESP 12E	1 pcs	Rp 50.000,-
4	Resistor	2 pcs	Rp 300,-
5	Module Step Down XL7015	1 pcs	Rp 18.000,-
6	Jasa Cetak PCB	1 pcs	Rp 87.000
	Total		Rp 255.300,-