

## BAB III PELAKSANAAN KERJA PROFESI

### 3.1 Bidang Kerja

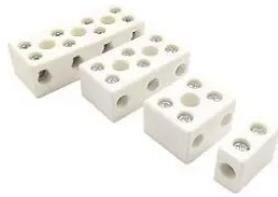
Selama menjalani program kerja profesi di Spora EV, praktikan berperan dalam pengembangan modul *energy meter* untuk cargo bike berbasis sitem hybrid charging. Ciptaan ini berupa suatu program komputer untuk membangun *energy meter*, untuk membaca arus, tegangan, kapasitas baterai, energi yang digunakan, dan waktu operasi secara *real-time*. *Energy meter* ini dipasang di antara baterai dan beban. Fokus utama bidang kerja ini adalah integrasi sistem tertanam untuk mendukung keberlanjutan lingkungan dan efisiensi operasional dalam mobilitas urban.

Praktikan terlibat dalam pembuatan algoritma perhitungan energi, kalibrasi sensor, dan pengembangan antarmuka pengguna berbasis *OLED* untuk menampilkan data energi terkait sebagai bagian dari proyek ini. Komponen yang digunakan dalam modul ini meliputi:

**Tabel 1.1 Komponen yang Digunakan**

No.	Komponen Digunakan	Penjelasan
1.	 <b>Arduino Nano</b>	Sebuah <i>mikrokontroler</i> kecil yang digunakan untuk mengelola seluruh proses pengukuran dan pemrosesan data dalam modul <i>energy meter</i> .
2.	 <b>Step Down</b>	Komponen yang digunakan untuk menurunkan tegangan input agar sesuai dengan kebutuhan daya <i>5angkah nano</i> dan komponen lainnya. Pada modul ini, <i>step down</i> diatur untuk menghasilkan tegangan output sebesar 8 volt.

3.



Komponen yang digunakan sebagai konektor untuk menghubungkan kabel-kabel dari berbagai perangkat ke modul 6angkah6man.

**Terminal Block**

4.



**OLED**

Layar kecil berbasis teknologi *organic light-emitting diode* yang digunakan untuk menampilkan energi secara *real-time*.

5.



**Resistor 33k**

Rangkaian resistor yang dibutuhkan untuk menurunkan tegangan dari baterai ke tingkat yang aman agar dapat dibaca oleh *arduino nano*.



**Resistor 2k2**

6.



**Sensor Arus**

Sensor arus yang digunakan untuk mengukur gelombang listrik yang mengalir melalui rangkaian.

### 3.2 Pelaksanaan Kerja

Pelaksanaan kerja profesi terdiri dari beberapa tahap utama, meliputi perancangan algoritma, implementasi sistem, dan pengujian sistem.

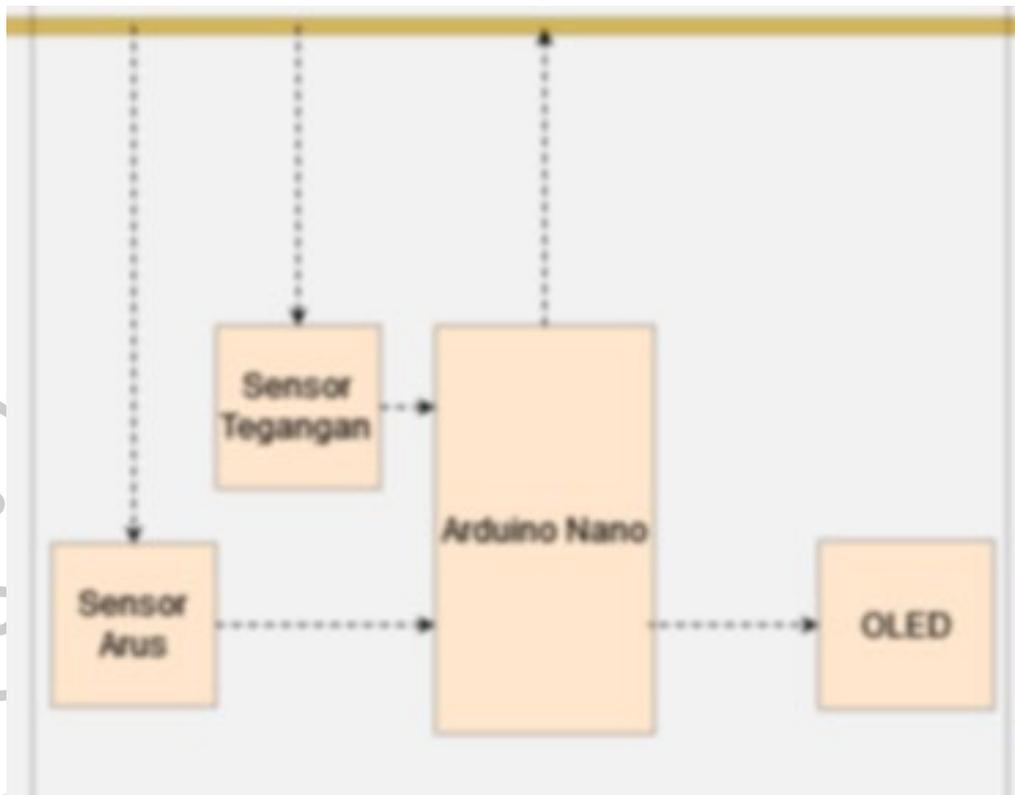
### 3.2.1 Perancangan Sistem

Tahap awal melibatkan desain algoritma yang mendukung pengukuran dan pemantauan energi. Algoritma yang digunakan dalam modul ini adalah sebagai berikut:

1. Melakukan kalibrasi pembacaan arus, tegangan, dan pewaktuan untuk Ah dan Wh.
2. Menetapkan jeda *sampling* 100 ms untuk pembacaan data secara konsisten.
3. Hitung rerata tegangan (V), arus (I), perubahan kapasitas baterai (dAh), dan perubahan energi (dWh) setelah iterasi tertentu.
4. Lakukan integrasi numerik untuk menghitung total kapasitas baterai (Ah) dan energi yang digunakan (Wh).
5. *Booting* logo.
6. Inisialisasi nilai sebelum *void loop*.
7. *Voltage sampling* untuk tiap iterasi.
8. *Current sampling* untuk tiap iterasi.
9. Kalkulasi Ah.
10. Kalkulasi Wh.
11. Tampilkan hasil pengukuran pada layar *OLED* dan cetak ke PC.

### Diagram Blok *Energy Meter*

Berikut adalah diagram blok yang menggambarkan hubungan antar komponen utama dalam modul *energy meter*.



Gambar 3. 1 Diagram Blok *Energy Meter*

**Gambar 3.1** menunjukkan hubungan antar komponen utama pada sistem *energy meter* yang digunakan dalam *cargo bike*. Berikut penjelasannya:

#### 1. Sensor Tegangan:

- Menggunakan rangkaian resistor untuk membaca tegangan baterai.
- Tegangan yang diukur dikirim ke *arduino nano* untuk diproses.

#### 2. Sensor Arus:

- Mendeteksi arus listrik yang mengalir dari baterai ke beban.
- Informasi arus diteruskan ke *arduino nano*.

### 3. Arduino Nano:

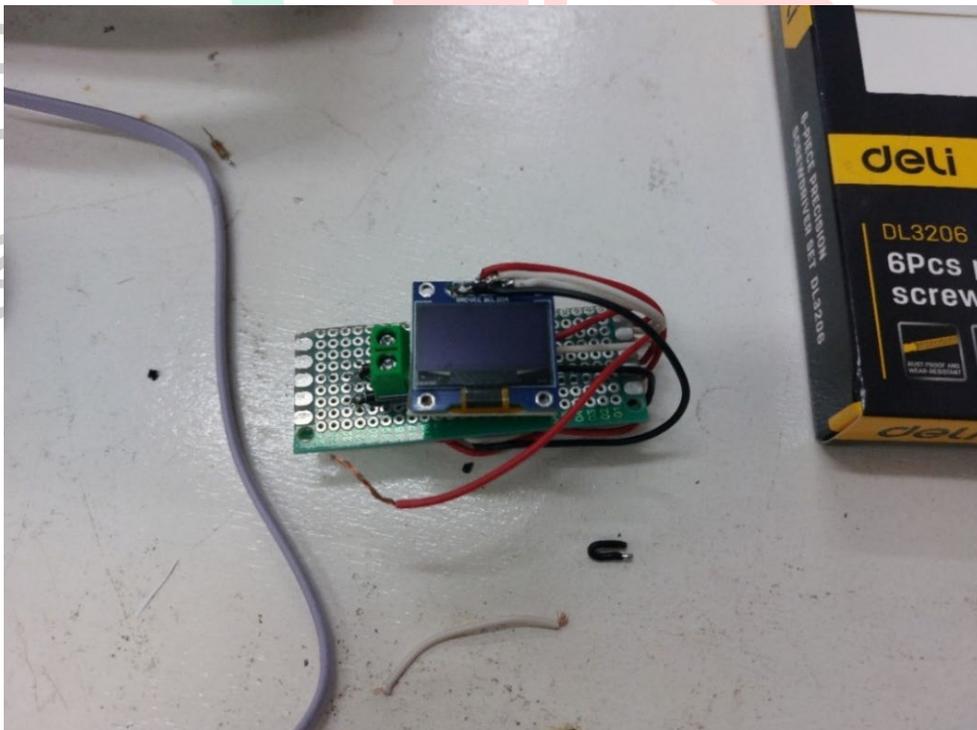
- *Mikrokontroler* utama untuk memproses data dari sensor tegangan dan arus.
- Menghitung parameter energi dan menampilkan hasil di *OLED*.

### 4. OLED:

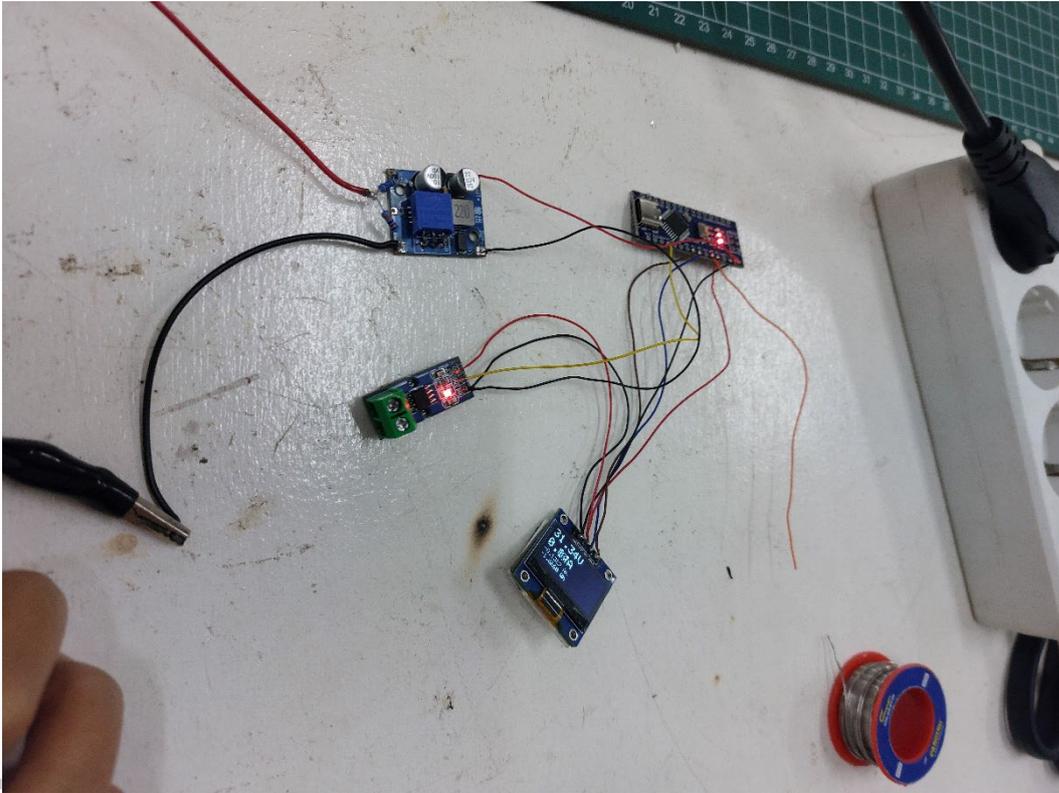
- Layar untuk menampilkan data energi seperti tegangan, arus, dan kapasitas baterai secara *real-time*.

### Tahap Awal Pengembangan Prototipe *Energy Meter*

Pada **Gambar 3.2** dan **3.3**, prototipe *energy meter* dirancang sebagai langkah awal untuk menguji algoritma dan memastikan fungsi komponen utama seperti pembacaan tegangan menggunakan *voltage divider* serta pembacaan arus dengan sensor ACS. Prototipe ini masih dalam bentuk dasar tanpa *casing* dan fokus pada validasi fungsi dasar sistem.



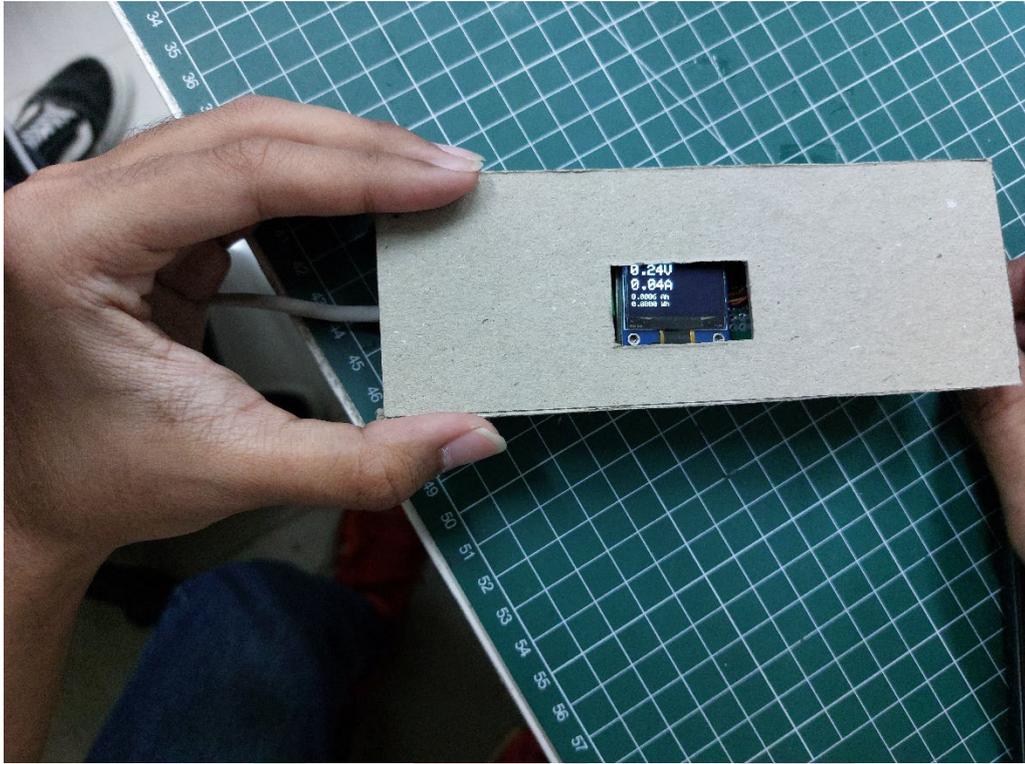
Gambar 3. 2 Tahap Awal (1)



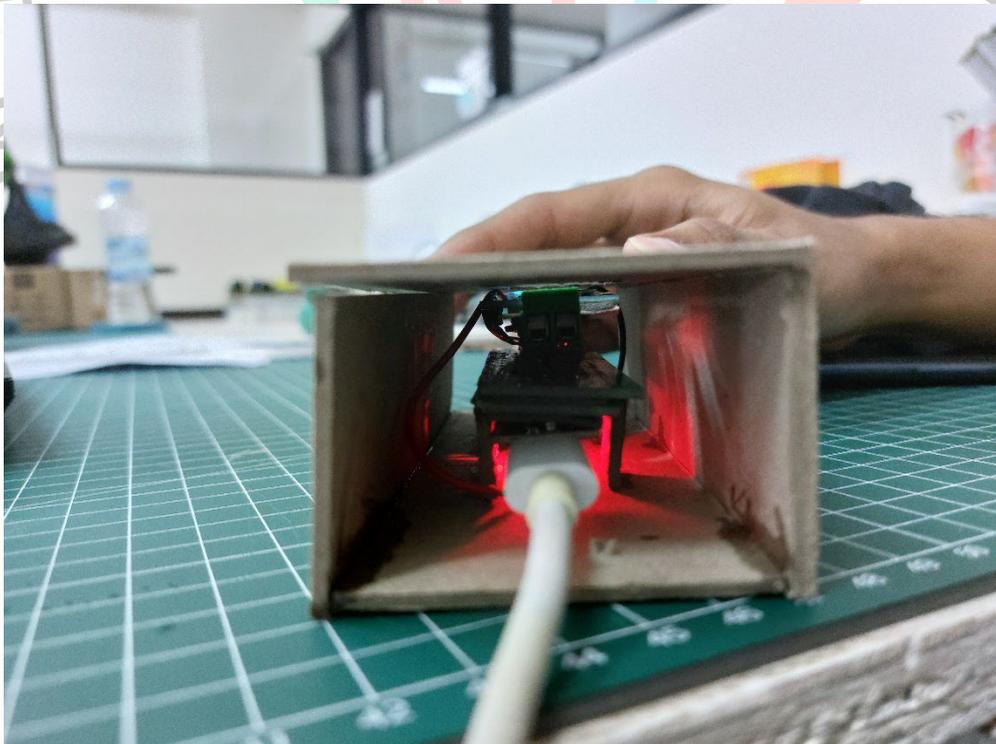
**Gambar 3. 3 Tahap Awal (2)**

Selama proses pengujian, casing awal dimaksudkan untuk melindungi komponen elektronik dasar dan memudahkan akses. Hardboard dipilih sebagai material utama karena ringan, fleksibel, dan memiliki kekuatan yang cukup untuk melindungi bagian-bagian sementara. Selain itu, karena bahan-bahan ini mudah ditemukan dan murah, prototipe dapat dibuat dengan cepat dan efisien tanpa memperumit anggaran atau kerumitan desain. *Casing* dibuat sederhana dengan dimensi yang disesuaikan agar semua komponen di dalamnya tersusun dengan rapi. Metode ini memastikan bahwa modul tetap stabil selama pengujian di laboratorium dan saat dipasang pada *cargo bike*, meningkatkan kenyamanan dan keandalan modul selama setiap tahap pengujian.

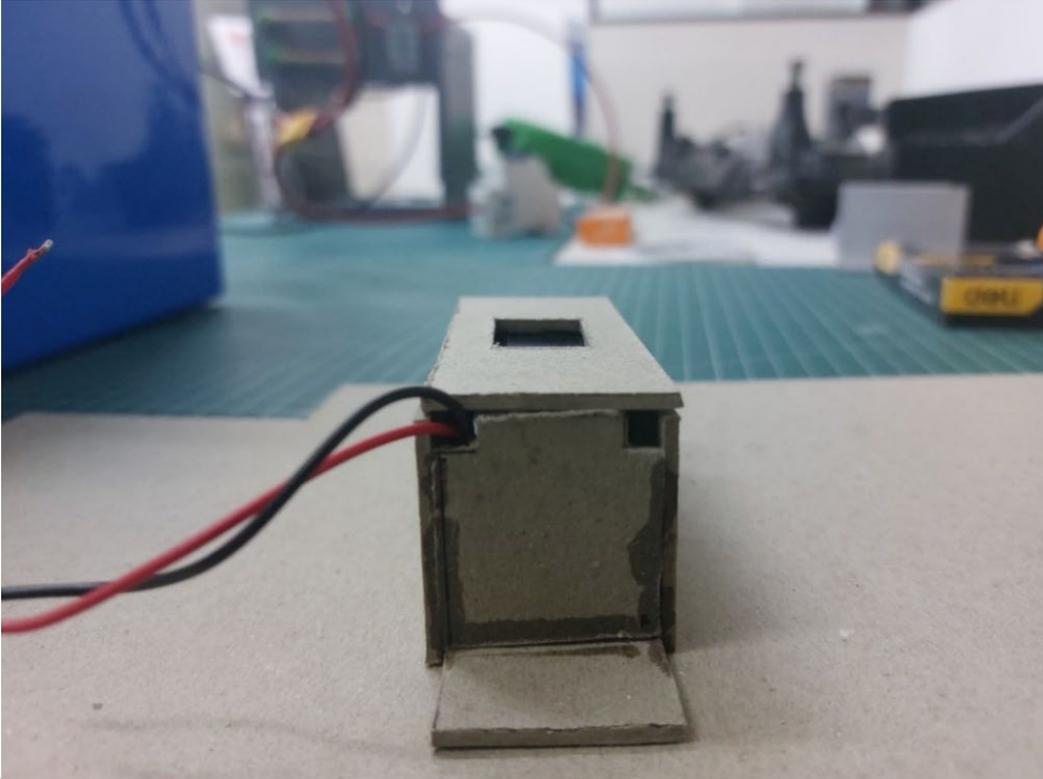
Pada **Gambar 3.4** sampai **3.6** proses *fitting energy meter* ke casing hardboard dilakukan untuk memastikan komponen terpasang dengan kokoh dan casing dapat menampung semuanya dengan baik. *Fitting* ini juga menguji apakah casing memberikan perlindungan yang cukup selama pengoperasian awal pada *cargo bike*.



Gambar 3. 4 Casing Hardboard (1)



Gambar 3. 5 Casing Hardboard (2)



Gambar 3. 6 Casing Hardboard (3)

### 3.2.2 Implementasi Sistem

Perangkat lunak modul ditulis menggunakan Bahasa pemrograman C++, dengan dibantu *library* untuk menjalankan komponennya. Beberapa langkah implementasi meliputi:

1. Inisialisasi sensor tegangan dan arus.
2. Penyiapan tampilan *OLED* untuk menampilkan parameter energi.
3. Integrasi algoritma pengukuran ke dalam *loop* utama *arduino nano*.
4. Pembuatan fungsi monitoring yang memungkinkan data energi diperbarui secara berkala.

#### Kode Program Energy Meter

Kode program untuk modul *energy meter* dirancang untuk membaca data dari sensor arus dan tegangan, memproses informasi tersebut, dan menampilkannya secara real-time pada layar *OLED*. Program ini ditulis dalam bahasa C++ dan diunggah ke mikrokontroler *Arduino Nano*, yang menjadi otak dari sistem *energy meter*. Struktur utama kode program terdiri dari beberapa bagian

utama:

#### 1. Inisialisasi Hardware:

Bagian ini mencakup inisialisasi pin untuk sensor, konfigurasi komunikasi serial, dan pengaturan pustaka untuk layar *OLED*. Contohnya, pustaka *Adafruit GFX* dan *Adafruit SSD1306* digunakan untuk mendukung tampilan grafis.

#### 2. Fungsi Pembacaan Data:

- Fungsi *readCurrent()* bertugas membaca data dari sensor arus ACS712, yang kemudian dikalibrasi untuk menghilangkan noise dan memastikan keakuratan.
- Fungsi *readVoltage()* mengambil data dari sensor tegangan, mengubah nilai *ADC* menjadi tegangan dalam satuan *volt*.

#### 3. Pengolahan Data:

Setelah data dibaca dari sensor, kode program mengolah informasi ini untuk menghitung parameter energi seperti daya, kapasitas baterai, dan konsumsi energi total.

#### 4. Tampilan Data:

Data hasil pengolahan ditampilkan pada layar *OLED* dengan format yang mudah dibaca pengguna. Program dirancang untuk memperbarui layar secara berkala, menggunakan fungsi *display.update()*.

**Gambar 3.7** sampai **3.11** berikut adalah kode program yang digunakan dalam pengembangan modul *energy meter* ini:

```

1 #Program Komputer untuk Energy Meter 50-100v / 0-30A
2
3 #include <SPI.h>
4 #include <Wire.h>
5 #include <Adafruit_GFX.h>
6 #include <Adafruit_SSD1306.h>
7
8 float k_V_in = 0.9836;
9 float k_V_out = 0.9836;
10 float vn = 2.49;
11 float k_I = 0.8803;
12 float v_per_a = 0.066;
13 float dt = 100;
14 float k_dt = 1.0;
15 int n = 36;
16 float t = (n*dt/1000)/3600;
17
18 float Threshold_Charging_atas = 81.0;
19 float Threshold_Charging_bawah = 78.0;
20
21 //Setup Pin-pin Analog dan Digital
22 int VinPin = A6;
23 int VoutPin = A7;
24 int amperePin = A0;
25 int led = LED_BUILTIN;
26 int status_LED = HIGH;
27
28 // Setup OLED
29 #define SCREEN_WIDTH 128

```

Gambar 3. 7 Kode Program (1)

```

28 // Setup OLED
29 #define SCREEN_WIDTH 128
30 #define SCREEN_HEIGHT 64
31 #define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino reset pin)
32 Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET); //Buat objek display dari class adafruit_ssd1306
33
34 // Bitmap untuk Logo
35 const unsigned char upj [] PROGMEM = {
36 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
37 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3f, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
38 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x3f, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
39 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x7c, 0x10, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
40 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x78, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
41 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x78, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
42 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x70, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
43 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x70, 0x03, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
44 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x70, 0x7f, 0xff, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
45 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x70, 0x00, 0x1f, 0xc0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
46 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x30, 0x00, 0x03, 0xf0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
47 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x30, 0x7e, 0x80, 0xf8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
48 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x30, 0x42, 0x80, 0x7c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
49 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x18, 0x5a, 0x80, 0x3e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
50 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x18, 0x52, 0x80, 0x1e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
51 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x18, 0x56, 0x80, 0x1e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
52 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0c, 0x56, 0x88, 0x1e, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
53 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x04, 0x50, 0x88, 0x1c, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
54 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x06, 0x5f, 0x90, 0x38, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
55 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x02, 0x00, 0x10, 0x60, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
56 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x01, 0x00, 0x30, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

```

Gambar 3. 8 Kode Program (2)

```

66   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x0f, 0xe0, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
67   0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x07, 0x80, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
68   };
69
70   void setup() { // put your setup code here, to run once:
71     Serial.begin(9600);
72     // Initialize with the I2C addr 0x3C (for the 128x64)
73     if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) {
74       Serial.println(F("SSD1306 allocation failed"));
75       for(;;) // Don't proceed, loop forever
76     }
77
78     //Booting logo
79     display.clearDisplay();
80     display.drawBitmap(0, 0, upj, 128, 32, 1);
81     display.display();
82     delay(1000);
83
84     pinMode(led, OUTPUT);
85     pinMode(relay, OUTPUT);
86     digitalWrite(led, status_LED);
87     digitalWrite(relay, status_Relay);
88   }
89
90   void loop() { // put your main code here, to run repeatedly:
91
92     //Sampling V_in dan V_out untuk Tiap Iterasi
93     rawValVolt_in = analogRead(VinPin);
94     rawValVolt_out = analogRead(VoutPin);

```

Gambar 3. 9 Kode Program (3)

```

89
90   void loop() { // put your main code here, to run repeatedly:
91
92     //Sampling V_in dan V_out untuk Tiap Iterasi
93     rawValVolt_in = analogRead(VinPin);
94     rawValVolt_out = analogRead(VoutPin);
95     V_sampling_in = k_V_in * 80.0/1023.0 * rawValVolt_in;
96     V_sampling_out = k_V_out * 80.0/1023.0 * rawValVolt_out;
97     V_in = V_in + V_sampling_in;
98     V_out = V_out + V_sampling_out;
99
100    rawValAmp = analogRead(amperePin);
101    v_I = 5.0 / 1023.0 * rawValAmp;
102    I_sampling = k_I * (v_I - vn) / v_per_a;
103    I = I + I_sampling;
104
105    if(counter == n){ 00000000
106      counter = 0;
107      status_LED = !status_LED; //Membuat LED berkedip setiap n sampling
108      digitalWrite(led, status_LED);
109      V_in = V_in/n; V_out = V_out/n; I = I/n;
110
111      //Ampere-hour Calculation
112      dAh = I * t;
113      Ah = Ah + dAh;
114      //Watt-hour Calculation
115      dWh = (V_out * I) * t;
116      kWh = kWh + dWh;
117      //Timer dalam jam

```

Gambar 3. 10 Kode Program (4)

```

115 [] dWh = (V_out * I) * t;
116 [] Wh = Wh + dWh;
117 [] //Timer dalam Jam
118 [] timer = timer + t;
119 []
120 [] //Cetak Hasil ke OLED
121 [] display.clearDisplay(); display.setTextSize(1); display.setTextColor(1);
122 [] display.setCursor(0, 0); display.print(V_in,2); display.print("v"); display.setCursor(0, 16); display.print(timer,3); display.
println("h");
123 [] display.setCursor(42, 0); display.print(I,2); display.print("A"); display.setCursor(42,16); display.print(Ah,2); display.
println("Ah");
124 [] display.setCursor(84, 0); display.print(V_out,2); display.print("v"); display.setCursor(84, 16); display.print(Wh,2); display.
print("Wh");
125 [] display.display();
126 [] //Cetak Hasil ke PC
127 [] Serial.print(V_in,2); Serial.print("v "); Serial.print(timer,3); Serial.println("h");
128 [] Serial.print(I,2); Serial.print("A "); Serial.print(Ah,2); Serial.println("Ah");
129 [] Serial.print(V_out,2); Serial.print("v "); Serial.print(Wh,2); Serial.println("Wh");
130 [] Serial.println();
131 [] //Kembalikan V dan I ke nol namun tidak untuk Ah dan Wh.
132 [] V_in = 0; V_out = 0; I = 0; [] [] [] [] []
133 [] }
134 [] counter = counter + 1;
135 [] delay(k_dt * dt);
136 [] }
137 []

```

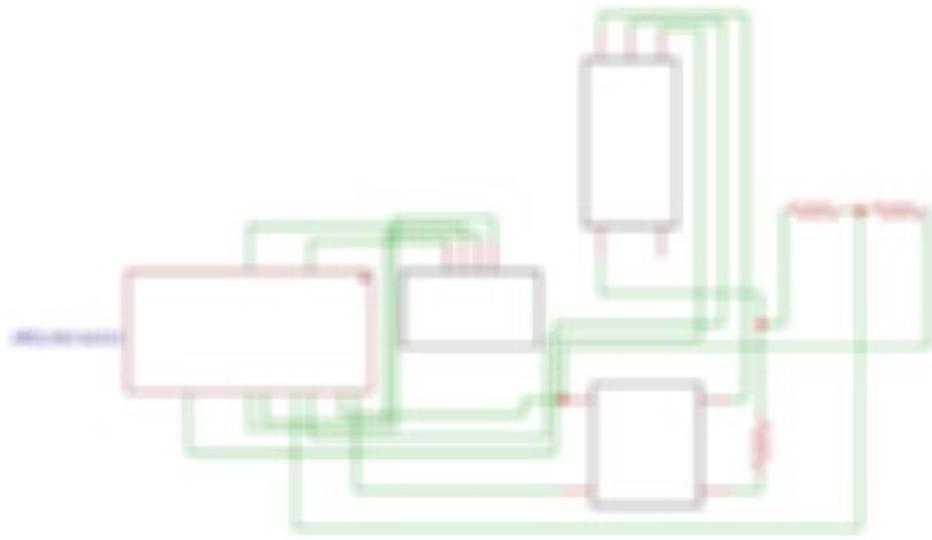
Gambar 3. 11 Kode Program (5)

Kode ini berfungsi untuk:

1. Menginisialisasi tampilan *OLED* dan sensor arus/tegangan.
2. Melakukan pengukuran parameter energi seperti A, V, Ah, dan Wh.
3. Menampilkan hasil pengukuran secara real-time pada layar *OLED*.
4. Mengatur waktu *sampling* dan proses integrasi data energi.

### **Energy Meter Schematic Diagram**

Kemudian **Gambar 3.12**, berikut ada *schematic diagram* yang menunjukkan koneksi antar komponen utama:

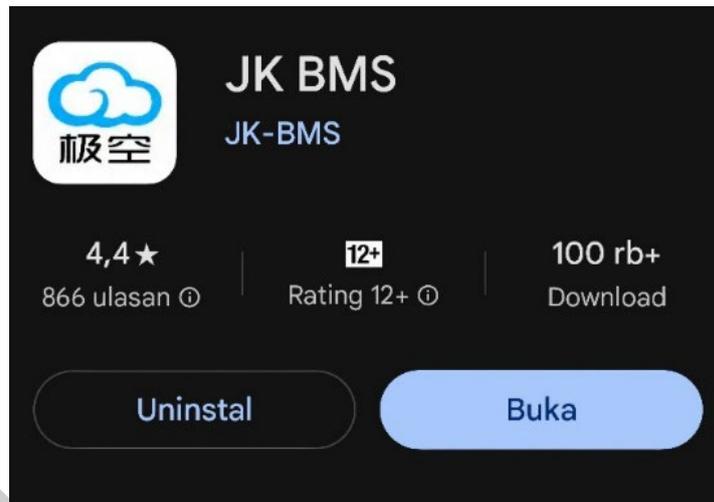


**Gambar 3. 12 Schematic Diagram**

### **3.2.3 Pengujian Sistem**

Aplikasi *JK BMS* digunakan untuk melakukan pengujian sistem, yang dihubungkan ke *fixed* baterai menggunakan *bluetooth*. Aplikasi ini memungkinkan pengguna untuk membaca data secara *real-time* seperti tegangan, arus, kapasitas baterai, dan status pengisian daya. Data dari aplikasi ini menjadi acuan utama untuk semua pengujian yang dilakukan.

Pada **Gambar 3.13** memperlihatkan aplikasi yang digunakan pada saat pengujian sistem.



Gambar 3. 13 JK BMS

Pada **Gambar 3.14** Antarmuka aplikasi dirancang agar intuitif, dengan indikator yang jelas untuk setiap parameter. Hal ini memudahkan teknisi untuk memantau kondisi baterai selama pengujian berlangsung. Selain itu, data yang ditampilkan oleh aplikasi digunakan sebagai acuan utama dalam analisis performa modul energy meter. Parameter yang dipantau meliputi:

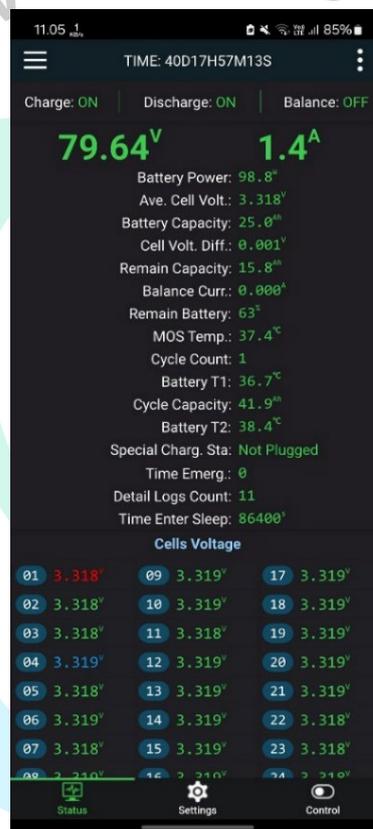
- **Tegangan:** Menunjukkan tingkat daya yang tersisa dalam baterai.
- **Arus:** Mencerminkan beban yang ditarik oleh sistem.
- **Kapasitas Baterai:** Mengindikasikan kapasitas yang tersisa atau telah digunakan selama pengujian.
- **Status Pengisian:** Memberikan informasi apakah baterai sedang dalam proses pengisian atau discharging.

Fitur dan Manfaat Penggunaan Aplikasi *JK BMS* Penggunaan aplikasi ini memberikan beberapa keuntungan utama:

1. **Pemantauan Real-Time:** Data yang diperoleh memungkinkan teknisi untuk segera mendeteksi anomali atau ketidaksesuaian dalam performa baterai.
2. **Kemudahan Akses:** Dengan koneksi Bluetooth, aplikasi dapat digunakan tanpa perlu menghubungkan perangkat secara fisik ke modul energy meter.

3. **Efisiensi Analisis Data:** Data yang dikumpulkan dapat disimpan untuk dianalisis lebih lanjut, membantu dalam proses pengambilan keputusan teknis.

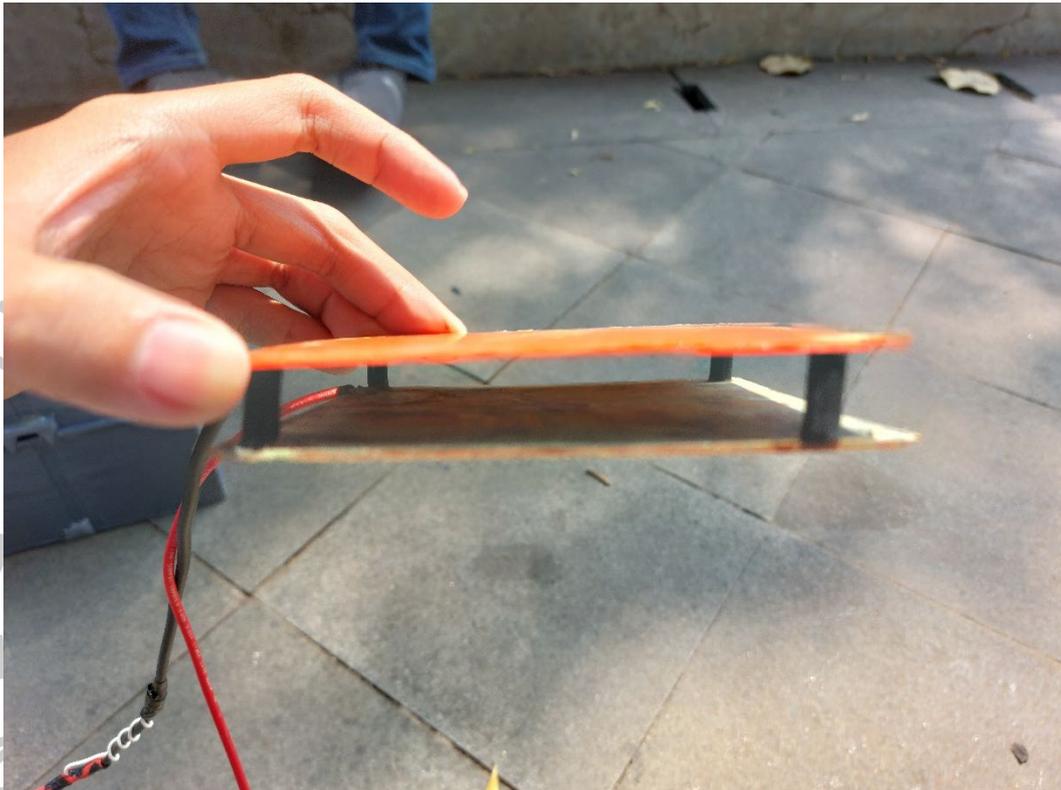
Dengan memanfaatkan aplikasi ini, proses pengujian menjadi lebih sistematis, efisien, dan akurat. Data yang diperoleh tidak hanya mendukung evaluasi kinerja baterai tetapi juga memberikan wawasan untuk optimasi sistem di masa mendatang.



Gambar 3. 14 Tampilan Aplikasi

Selain itu, **Gambar 3.15** menjelaskan *dummy load* yang digunakan untuk pengujian terdiri dari dua papan PCB polos yang dipisahkan oleh selongsong kabel setinggi 5 cm. Kemudian salah satu sisi dari dummy load dihubungkan menggunakan kabel positif dan lainnya negatif untuk mengalirkan arus selama pengujian. Desain ini dipilih untuk mensimulasikan beban nyata yang akan dihadapi oleh baterai selama operasional. Selain itu, penggunaan papan PCB polos memberikan fleksibilitas dalam menyusun konfigurasi beban sesuai dengan

kebutuhan spesifik pengujian. Metode *elektrolisis* menggunakan larutan air garam diterapkan untuk menciptakan beban yang konsisten (Hamid et al., 2017). Larutan ini menghasilkan resistensi yang diperlukan untuk meniru beban nyata selama pengujian discharging dan pengujian *error*.



Gambar 3. 15 Dummy Load

### Pengujian Error Arus dan Tegangan Energy Meter

Sebelum dipasang pada *cargo bike* dan menggunakan *casing 3D print*, dilakukan pengujian *error* arus dan tegangan modul menggunakan *fixed* baterai untuk memastikan performa sistem dalam pembacaan data.

Jenis *error* yang digunakan dalam pengujian **Gambar 3.16** dan **3.17** adalah *relative error* yang dihitung berdasarkan persentase perbedaan antara hasil pengujian dengan nilai rujukan (Bashyal, 2023). Rumus perhitungan error adalah sebagai berikut:

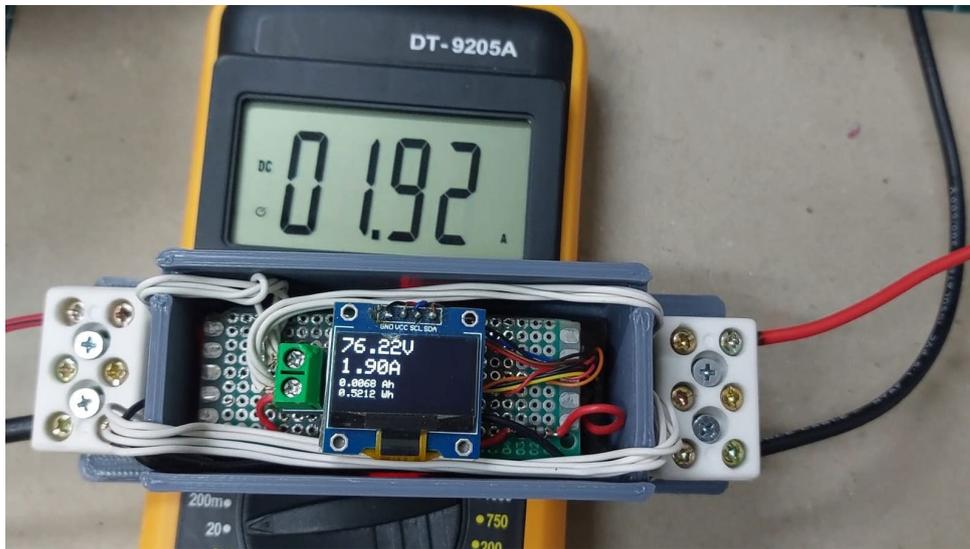
$$Error (\%) = \frac{(\text{Nilai Hasil Pegujian} - \text{Nilai Hasil Rujukan})}{\text{Nilai Rujukan}} \times 100\%$$

Hasil Uji Error Arus EM				
No.	Waktu	Arus	Hasil	Error (%)
1	04/11/2024	1,44	1,43	1%
2	04/11/2024	1,92	1,9	1%
3	04/11/2024	2,85	2,96	4%
4	04/11/2024	3,56	3,1	13%
5	04/11/2024	4,4	4,16	5%
6	04/11/2024	6,08	5,63	7%
7	04/11/2024	7,17	6,89	4%
8	04/11/2024	7,67	7,02	8%
9	04/11/2024	8,55	7,96	7%
10	04/11/2024	10,12	9,35	8%

Gambar 3. 16 Tabel Uji Error Arus

Hasil Uji Error Tegangan EM				
No.	Waktu	Tegangan	Hasil	Error (%)
1	04/11/2024	78,5	78,51	0%
2	04/11/2024	77	76,93	0%
3	04/11/2024	78,4	78,27	0%
4	04/11/2024	78,4	78,2	0%
5	04/11/2024	78,5	78,51	0%
6	04/11/2024	76,9	76,38	1%
7	04/11/2024	76,9	76,3	1%
8	04/11/2024	76,9	76,38	1%
9	04/11/2024	76,8	76,38	1%
10	04/11/2024	76,7	76,06	1%

Gambar 3. 17 Tabel Uji Error Tegangan



Gambar 3. 18 Pengujian Arus Energy Meter



Gambar 3. 19 Pengujian Tegangan Energy Meter

Hasil pengujian dari **Gambar 3.18** dan **3.19** menunjukkan bahwa modul memiliki tingkat akurasi yang memadai dengan kesalahan minimal pada pengukuran arus dan tegangan.

### Pengujian Discharging dan Charging Fixed Baterai

Pengujian *discharging* menggunakan dummy load, sedangkan pengujian *charging fixed* baterai ini dilakukan menggunakan 5 solar panel 30wp dibikin seri, pengujian ini dilakukan untuk mengetahui performa baterai ketika digunakan dalam kondisi beban tertentu. Berikut adalah datanya:

Tabel 3. 1 Discharging Baterai dengan Dummy Load (1)

No.	Waktu	V bat (v)	I (A)
1	03/10/2024 08.36	78.89	1.2
2	03/10/2024 08.40	78.59	1.1
3	03/10/2024 08.43	78.50	1.0

Tabel 3. 2 Discharging Baterai dengan Dummy Load (2)

No.	Waktu	V bat (v)	I (A)
1	04/10/2024 13.10	78.52	0.8
2	04/10/2024 13.20	78.45	0.8
3	04/10/2024 13.31	78.48	0.8

Tabel 3. 3 Discharging Baterai dengan Dummy Load (3)

No.	Waktu	V bat (v)	I (A)
1	07/10/2024 08.35	78.45	0.8
2	07/10/2024 08.58	78.45	0.8
3	07/10/2024 09.09	78.48	0.8

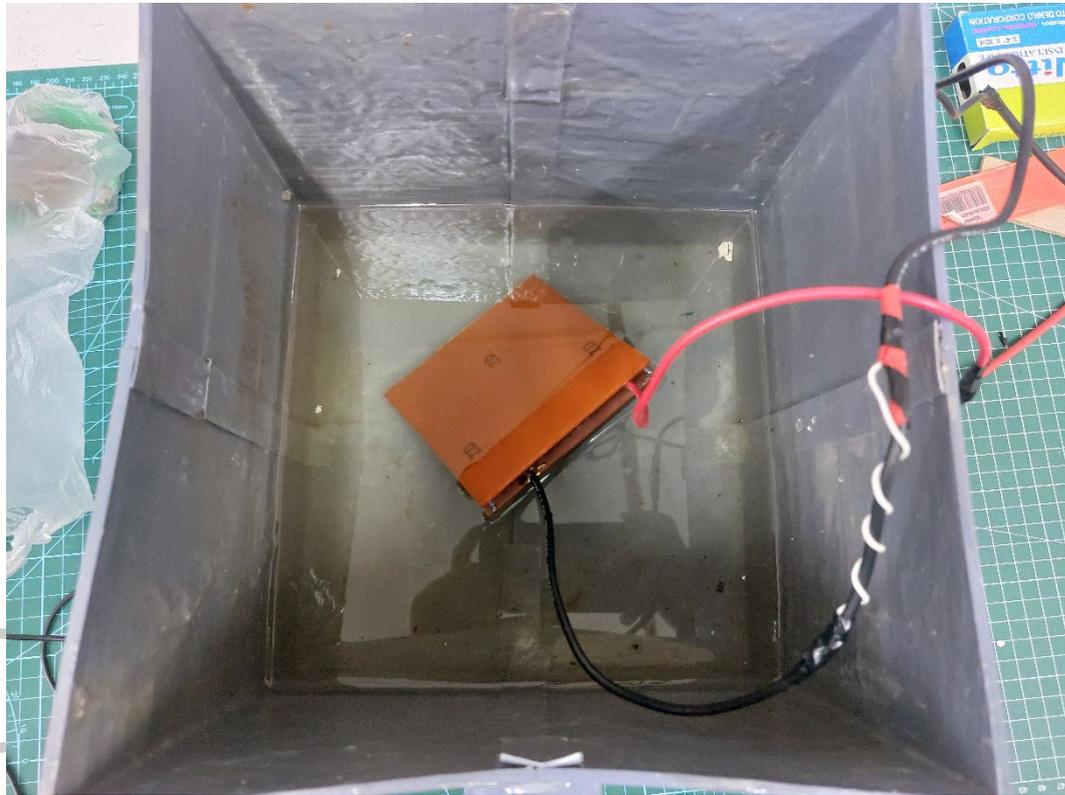
Tabel 3. 4 Discharging Baterai dengan Dummy Load (4)

No.	Waktu	V bat (v)	I (A)
1	08/10/2024 10.15	76.45	0.9

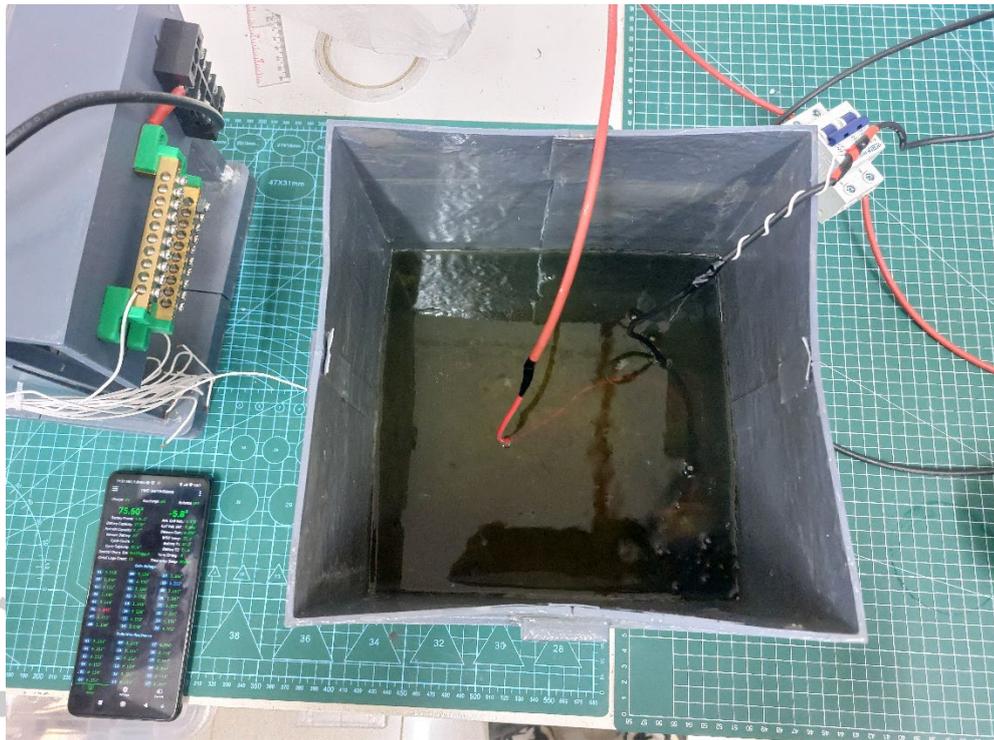
Pengujian *discharging* dilakukan menggunakan *dummy load* dengan metode *elektrolisis* untuk mensimulasikan beban nyata. Dari **Tabel 3.1** sampai **3.4**, pengujian mencatat perubahan tegangan baterai (V bat) terhadap waktu. Secara keseluruhan, tegangan baterai mengalami penurunan dari 78.89 V pada awal pengujian (03/10/2024 jam 08.36) hingga mencapai 76.45 V di akhir pengujian (08/10/2024 jam 10.15).

Penurunan ini mencerminkan kapasitas baterai yang terkuras secara bertahap selama proses *discharging*. Waktu pengujian juga menunjukkan bahwa beban konsisten sehingga proses pengosongan daya berjalan stabil.

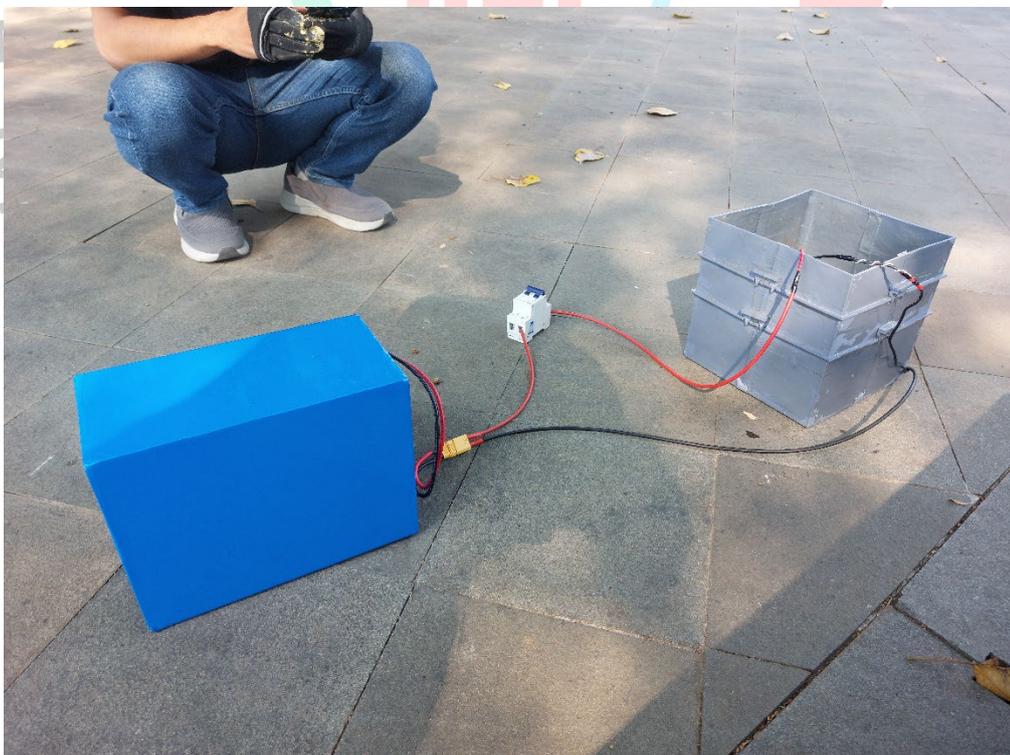
**Gambar 3.20** sampai **3.23** menunjukkan proses pengujian *discharging* baterai menggunakan metode *elektrolisis* dengan *dummy load*. Pengaturan beban yang konsisten memastikan hasil pengujian yang akurat.



**Gambar 3. 20 Discharging dengan Metode Elektrolisis (1)**



Gambar 3. 21 Discharging dengan Metode Elektrolisis (12)



Gambar 3. 22 Discharging dengan Metode Elektrolisis (3)



Gambar 3. 23 Discharging dengan Metode Elektrolisis (4)

Tabel 3. 5 Charging Baterai dengan Solar Panel (1)

No.	Waktu	V bat (v)	I (A)
1	10/10/2024 10.55	76.05	1.1
2	10/10/2024 10.59	76.51	1.6
3	10/10/2024 11.05	77.02	0.9

Tabel 3. 6 Charging Baterai dengan Solar Panel (2)

No.	Waktu	V bat (v)	I (A)
1	11/10/2024 13.23	77.26	1.2
2	11/10/2024 13.34	77.38	0.1
3	11/10/2024 13.41	77.56	0.5

Tabel 3. 7 Charging Baterai dengan Solar Panel (3)

No.	Waktu	V bat (v)	I (A)
1	15/10/2024 10.43	77.04	1.2
2	15/10/2024 10.49	77.33	0.8
3	15/10/2024 10.54	77.60	1.4

**Tabel 3. 8 Charging Baterai dengan Solar Panel (4)**

No.	Waktu	V bat (v)	I (A)
1	17/10/2024 10.17	77.80	1.4
2	17/10/2024 10.26	78.47	1.2
3	17/10/2024 10.33	78.72	1.4

**Tabel 3. 9 Charging Baterai dengan Solar Panel (5)**

No.	Waktu	V bat (v)	I (A)
1	22/10/2024 13.35	79.54	0.2
2	22/10/2024 13.39	79.69	1.4
3	22/10/2024 13.49	79.70	1.0

Pengujian *charging* menggunakan solar panel yang disusun seri untuk meningkatkan tegangan keluaran. Dari **Tabel 3.5** sampai **Tabel 3.9**, tercatat peningkatan tegangan baterai dari 76.05 V (10/10/2024 jam 10.55) sampai mencapai 79.70 V (22/10/2024 jam 13.49). Pengujian ini menunjukkan bahwa baterai dapat diisi ulang dengan efisiensi tinggi menggunakan tenaga surya, di mana tegangan meningkat secara bertahap seiring waktu. Data arus (I) pada tabel juga mengalami perubahan kecil selama proses pengisian, kemungkinan disebabkan oleh intensitas matahari yang berubah.

Pada **Gambar 3.24** dan **3.25** sedang memperlihatkan kegiatan untuk mengukur tegangan dan arus yang keluar dari solar panel.



**Gambar 3. 24 Foto Charging Baterai dengan Solar Panel (1)**



**Gambar 3. 25 Foto Charging Baterai dengan Solar Panel (2)**

### Tahap Instalasi

Pemasangan modul *energy meter* pada *cargo bike* dilakukan untuk memastikan integrasi fisik dan operasional modul bekerja dengan baik. Proses ini mencakup langkah-langkah berikut:

1. Penentuan lokasi pemasangan pada kerangka *cargo bike* untuk memastikan modul dapat diakses dengan mudah dan terlindungi.
2. Pengeboran lubang pada kerangka untuk menempatkan modul dengan aman.
3. Pemasangan modul menggunakan sekrup dan *bracket* yang dirancang khusus,
4. Dan lain-lain.

### Pengeboran pada Kerangka

Pada **Gambar 3.26**, terlihat proses pengeboran pada kerangka *cargo bike*. Proses pengeboran ini dilakukan untuk membuat lubang pemasangan modul *energy meter* pada kerangka *cargo bike*. Langkah ini bertujuan untuk memastikan modul dapat dipasang dengan kokoh dan tidak bergeser saat digunakan dalam kondisi operasional.



**Gambar 3. 26** Pengeboran pada Kerangka Cargo Bike

### Instalasi Energy Meter

Pada **Gambar 3.27** dan **3.28**, setelah proses pengeboran selesai, modul *energy meter* dipasang menggunakan mur dan baut. Modul ini ditempatkan di

lokasi yang strategis untuk memudahkan pembacaan data sekaligus melindunginya dari potensi kerusakan.



**Gambar 3. 27 Instalasi Energy Meter (1)**



**Gambar 3. 28 Instalasi Energy Meter (2)**

### **Instalasi Solar Panel**

Pada **Gambar 3.29** memperlihatkan kegiatan mengupas kabel, mengupas kabel dilakukan untuk menghubungkan ke lima solar panel secara seri. Penyambungan ini bertujuan untuk meningkatkan tegangan output dari panel surya sehingga sesuai dengan kebutuhan pengujian *fixed* baterai.



**Gambar 3. 29 Mengupas Kabel untuk Solar Panel**

- Pada **Gambar 3.30** dan **Gambar 3.32**, memperlihatkan kegiatan pemasangan solar panel ke *bracket*, proses pemasangan solar panel dilakukan dengan mempertimbangkan sudut panel terhadap matahari untuk mendapatkan efisiensi energi yang maksimal.



**Gambar 3. 30 Memasang Solar Panel (1)**



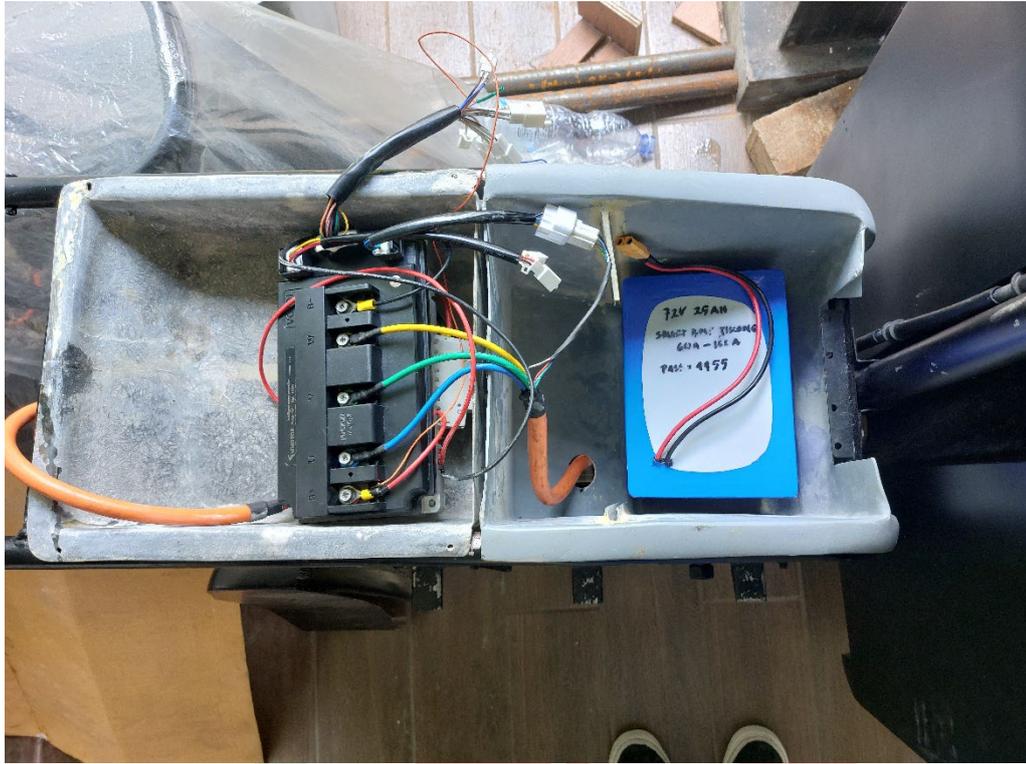
**Gambar 3. 31 Memasang Solar Panel (2)**



Gambar 3. 32 Memasang Solar Panel (3)

#### Fitting Fixed Baterai

Pada **Gambar 3.33** dan **Gambar 3.35** dilakukan untuk memastikan *fixed* baterai terpasang dengan baik pada *cargo bike*. Langkah ini mencakup pengaturan posisi baterai dan penguncian menggunakan bracket untuk menghindari pergeseran saat kendaraan digunakan.



Gambar 3. 33 Fitting Fixed Baterai (1)



Gambar 3. 34 Fitting Fixed Baterai (2)



Gambar 3. 35 Fitting Fixed Baterai (3)

### 3.3 Kendala Yang Dihadapi

Selama pelaksanaan kerja profesi, praktikan menghadapi beberapa kendala, diantaranya:

1. **Kalibrasi Sensor:** Untuk memastikan pembacaan data yang akurat, proses kalibrasi sensor arus dan tegangan membutuhkan ketelitian yang tinggi. Ketidaksesuain dalam faktor kalibrasi membuat hasil pengukuran menjadi tidak akurat, sehingga memerlukan beberapa iterasi pengujian.
2. **Keterbatasan Waktu:** Pengembangan *energy meter* dan pengujian sistem harus dilakukan bersamaan dengan proses integrasi selama program kerja profesi yang terbatas.
3. **Komponen Casing:** Desain casing harus diubah berkali-kali karena kendala teknis terkait dengan desain awal yang tidak sepenuhnya sesuai dengan dimensi modul.

4. **Integrasi Solar Panel:** Penyusunan solar panel secara seri memerlukan pengaturan kabel yang presisi untuk mencegah kehilangan daya dan memastikan efisiensi maksimal selama pengujian.

### 3.4 Cara Mengatasi Kendala

Untuk mengatasi kendala tersebut, beberapa langkah solusi telah diimplementasikan, yaitu:

1. **Kalibrasi Ulang:** Modul diuji berulang kali guna memastikan faktor kalibrasi yang tepat. Setiap iterasi didokumentasikan untuk membantu proses validasi.
2. **Manajemen Waktu:** Praktikkan membuat jadwal kerja yang terorganisir untuk membagi waktu secara efektif antar proses desain, pengembangan, dan pengujian.
3. **Perbaikan Desain Casing:** Praktikkan memperbaiki desain casing dengan menggunakan perangkat lunak desain 3D. Desain yang baru diuji pada prototipe sebelum dilakukan pencetakan ulang.
4. **Pengelolaan Kabel Solar Panel:** Menggunakan terminal block untuk menyambungkan kabel secara rapi dan aman. Proses ini juga memastikan pengurangan risiko *short circuit*.