

## MOTORCYCLES RPM MONITORING USING VOTOL COMMUNICATION PROTOCOL (HEXADECIMAL) CONTROLLER IN REAL TIME BASED ON IOT

Muhammad Alghifari<sup>a,1,\*</sup>, Dr. Ir. Hartoyo, S.Pd., M.Pd., MT.<sup>a,2</sup>

<sup>a</sup> Departement of Electrical and Electronic Engineering, Vocational Faculty, UNY

<sup>1</sup> [muhammadalghifari.2020@student.uny.ac.id](mailto:muhammadalghifari.2020@student.uny.ac.id); <sup>2</sup> [hartoyo@uny.ac.id](mailto:hartoyo@uny.ac.id)

\* Corresponding Author

### ARTICLE INFO

#### Article History

Received Date Month Year

Revised Date Month Year

Accepted Date Month

Year

#### Keywords

Votol

Controller;

Hexadecimal;

Monitoring;

RPM;

### ABSTRACT

This final project aims to develop and evaluate a real-time IoT-based motorcycle RPM monitoring system using the Votol controller communication protocol (Hexadecimal). The project involves functional testing of components like the Votol controller, Node MCU ESP32, and LCD I2C 20x4, and the stages include Requirement Analysis, Tool Design and Planning, Development of the Monitoring System, and System Testing. Conducted at the Satuso EV Workshop in Yogyakarta, the RPM monitoring system helps users monitor voltage, current, and RPM in real-time and analyze performance. Data from the Hexadecimal Votol controller is used to determine battery percentage and convert RPM to speed in km/h. Functional and performance testing showed that all components work normally, with parameter values matching the Votol controller and Thingspeak Dashboard, and an error max range of  $\pm 2,5\%$  according to IEC 13B-23 standards, making the system suitable for use both with and without load, and ensuring consistent data across LCD, Thingspeak, and the Votol controller.

Proyek akhir ini bertujuan untuk mengembangkan dan mengevaluasi sistem pemantauan RPM sepeda motor listrik secara real-time berbasis IoT menggunakan protokol komunikasi controller Votol (Hexadecimal). Proyek ini melibatkan pengujian fungsional komponen seperti controller Votol, Node MCU ESP32, dan LCD I2C 20x4, dengan tahapan meliputi Analisis Kebutuhan, Desain dan Perencanaan Alat, Pengembangan Sistem Monitoring, serta Pengujian Sistem. Dilakukan di Bengkel Satuso EV Yogyakarta, sistem pemantauan RPM ini membantu pengguna memantau tegangan, arus, dan RPM secara real-time serta menganalisis kinerja. Data dari controller Votol Hexadecimal digunakan untuk mengetahui persentase baterai dan mengkonversi RPM menjadi kecepatan dalam km/jam. Pengujian fungsional dan kinerja menunjukkan bahwa semua komponen berfungsi normal, dengan nilai parameter sesuai dengan controller Votol dan Dashboard Thingspeak, serta kisaran maksimal kesalahan  $\pm 2,5\%$  sesuai standar IEC 13B-23. Sistem ini layak digunakan baik tanpa beban maupun dengan beban, dan data pada Thingspeak konsisten dengan LCD dan controller Votol.

This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## 1. Pendahuluan

Konversi motor berbahan bakar minyak menjadi Kendaraan motor listrik sedang dalam masa percepatan. Diketahui kementerian ESDM meningkatkan bantuan konversi dari awalnya 7 juta naik hingga 10 juta yang di atur dalam Peraturan Menteri energi dan sumber daya mineral nomor 13 tahun 2023. Proses konversi motor listrik ini dilakukan dengan cara mengganti komponen-komponen utama dalam kendaraan motor berbahan bakar minyak yaitu, sistem penggerak motor bahan bakar minyak diganti dengan *Brush Less Direct Current* (BLDC) motor yang kapasitasnya dapat di sesuaikan dengan kebutuhan, kemudian menambahkan *controller* yang dirancang khusus untuk motor listrik, baterai sebagai energi utama dari kendaraan motor listrik itu sendiri [1].

Kendaraan motor listrik memiliki unsur penting yaitu controller yang berfungsi untuk kalibrasi, mengontrol keluaran tegangan, arus, dan sinyal yang mengaktifkan serta menjalankan motor listrik. Monitoring kinerja controller motor listrik dapat dilakukan secara konvensional atau jarak jauh dengan perangkat penyimpanan dan pengirim data. Monitoring adalah aktivitas untuk mengevaluasi kinerja suatu sistem dan mendapatkan informasi untuk perbaikan, termasuk dalam predictive maintenance yang mengukur kinerja secara berkelanjutan selama operasional. Monitoring motor listrik menggunakan perangkat software dari controller, namun penggunaannya tidak efektif secara real-time saat berkegiatan sehari-hari karena besaran monitoring cenderung fluktuatif akibat perubahan beban, gangguan, atau kondisi tidak normal pada motor listrik dan controller [2].

Kendaraan motor listrik yang digunakan pada saat kondisi berat, seperti pada saat berbeban atau melawan medan tanjakan, terjadi suatu peristiwa seperti peningkatan penggunaan tegangan, arus, dan daya yang digunakan oleh motor listrik. Meskipun daya atau energi yang digunakan semakin besar, namun hal ini juga mengakibatkan penurunan kecepatan atau putaran per menit (RPM) dan juga penurunan *performa* motor listrik seiring waktu. Penurunan *performa* ini juga dapat disebabkan oleh berbagai faktor yaitu kenaikan penggunaan daya yang energinya di ambil dari baterai, kemudian juga bisa disebabkan oleh peningkatan suhu motor yang menyebabkan penurunan efisiensi secara keseluruhan[3].

Salah satu penelitian tentang RPM Sepeda motor listrik khususnya yang berbasis IoT adalah pengembangan *monitoring* RPM Sepeda motor listrik yang menggunakan sensor *hall effect* yang di *monitoring* secara *real time* dan menggunakan IoT [4]. Penelitian ini menghasilkan sebuah alat yang berbentuk *prototipe* motor listrik yang mampu memantau putaran atau RPM Sepeda motor listrik secara *daring* dan *real time*. Penelitian lain yang membahas mengenai *monitoring* motor listrik adalah alat *monitoring* tegangan, arus, dan juga daya listrik yang digunakan oleh motor listrik tersebut untuk mengetahui kecepatan (RPM) dari motor listrik juga mengetahui berapa energi yang digunakan berbasis IoT yang memanfaatkan sensor tegangan, dan arus [3]. Penelitian tersebut menghasilkan sebuah sistem yang mampu membaca jumlah pemakaian energi dan kecepatan (RPM) yang dihasilkan oleh motor listrik melalui LCD dan juga *Data logger* dari sd card.

Penelitian terkait motor listrik ini masih sangat minim dilakukan, ada beberapa penelitian yang sudah dilakukan terkait sistem *monitoring* motor listrik terkhususkan *monitoring* yang berkaitan dengan performa, dan kecepatan (RPM) motor listrik, namun yang menjadi alasan pentingnya penelitian ini dan yang membedakan dengan penelitian sebelumnya yaitu pada sistem kinerja *monitoring*, yang sistem kinerja alatnya yaitu dengan mengkonversi data *Hexadecimal* yang ada dalam *protocol comucation controller* motor listrik. Data *Hexadecimal* itu kemudian di konversi menjadi sebuah data *decimal* yang dapat dibaca dan juga dapat menjadi data tegangan, arus, daya, dan RPM Sepeda motor listrik yang kemudian menjadi data *monitoring* motor listrik [5]. Berdasarkan gagasan dan permasalahan dari penelitian sebelumnya, penulis membuat, memodifikasi penelitian yang sudah ada menjadi satu kesatuan sistem yang utuh menggunakan konsep *monitoring* motor listrik secara *real time* dan juga jarak jauh. Ide tersebut dituangkan dalam bentuk Proyek Akhir berjudul “*Sistem Monitoring Rpm Sepeda Motor Listrik Menggunakan Protocol Communication (Hexadecimal) Controller Votol Secara Real Time Berbasis Iot*”. Tujuan dari Proyek Akhir ini adalah untuk mengetahui uji fungsional dan kinerja dari keseluruhan sistem yang digunakan pada motor listrik meliputi *controller* motor listrik, motor BLDC melalui perfoma dan juga RPM yang dihasilkan, serta penggunaan tegangan dan arus pada baterai.

## 2. Pendekatan Pemecahan Masalah

### 2.1 Kendaraan Sepeda Motor Listrik

Konversi motor berbahan bakar minyak menjadi motor listrik sedang dipercepat, dengan bantuan konversi dari Kementerian ESDM meningkat dari 7 juta menjadi 10 juta sesuai Peraturan Menteri ESDM nomor 13 tahun 2023. Proses ini melibatkan penggantian komponen utama, seperti sistem penggerak motor bahan bakar minyak dengan motor BLDC (Brushless Direct Current), penambahan controller khusus, dan baterai sebagai sumber energi utama [1]. Motor listrik mengubah energi listrik menjadi energi mekanik berdasarkan prinsip induksi elektromagnetik, dengan jenis motor BLDC yang umum digunakan dalam kendaraan listrik [6].

Controller motor listrik mengendalikan tegangan, arus, kecepatan, dan arah putar motor, memproses sinyal dari sensor dan actuator untuk menghasilkan sinyal kontrol yang sesuai [1]. Controller juga mengatur keluaran dari BLDC dan baterai melalui modul seperti speed control yang menggunakan algoritma PID dan PI, inverter yang mengubah keluaran baterai menjadi 3 fasa DC, dan modul PWM yang memanipulasi lebar sinyal untuk kontrol daya dan tegangan [7].

### 2.2 Data Hexadecimal

Sistem bilangan *Hexadecimal* (*hex*) merupakan sistem bilangan berbasis 16, di mana nilai-nilai diwakili dengan kombinasi angka 0 – 9 dan huruf A – F. Sistem ini banyak digunakan dalam bidang pemrograman dan elektronika karena menawarkan keringkasan penulisan dalam menulis sebuah angka yang mengartikan data atau juga nilai yang sebenarnya kalimat untuk mengetahui eror atau keberhasilan sebuah sistem [5]. Proses pengkonversian bilangan *Hexadecimal* ke bilangan Desimal juga dilakukan dengan cara pembagian berulang terhadap basis 16.

### 2.3 Platform Thingspeak

*Internet of Think* (IoT) merupakan sebuah tren yang muncul pada sebuah perangkat yang terhubung ke internet. Perangkat atau sebuah alat yang terhubung ke internet tersebut digunakan untuk mengirimkan data ke penyimpanan atau *cloud computing* untuk diproses secara lebih dan kemudian di analisis. Salah satu platform IoT yang dapat digunakan untuk melakukan hal tersebut ialah *platform think speak*. Platform ini juga digunakan untuk mengumpulkan dan memvisualisasi, dan menganalisis data langsung pada *cloud* secara instan. Data yang dihasilkan oleh Platform ini juga bisa dihubungkan dengan *software* matlab dan dianalisis secara online [9].

*Think speak* dapat menerima berbagai jenis data yang dikirim dari sebuah program atau juga sebuah mikrocontroller yang kemudian diolah dalam sebuah channel. Pada setiap channel yang ada dapat menerima data yaitu delapan *fields* yang kemudian di tampilkan dalam berbagai bentuk seperti grafik, angka dan juga gauge. Channel yang dibuat juga dapat diakses secara pribadi maupun secara komersial dengan cara mengundang untuk masuk ke dalam channel yang dibuat [10].

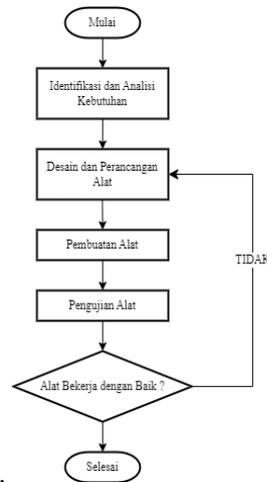
### 2.4 Node MCU ESP32

Node MCU ESP32, adalah sebuah mikrocontroller canggih dari salah satu pengembangan Arduino, mikrocontroller ini menawarkan kemudahan dan efisiensi dalam pengembangan perangkat *Internet of things* (IoT). Keunggulan utama Node MCU terletak pada modul Wi-Fi yang sudah langsung terintegrasi di dalamnya, yang memudahkan pengguna dalam tambahan untuk koneksi internet [11]. Node MCU ESP32 merupakan mikrocontroller yang paling populer. Dikembangkan oleh *espressif sistem* sebagai penerus dari ESP8266, ESP32 juga menawarkan performa dan fleksibilitas yang lebih tinggi. Perangkat ini juga telah banyak digunakan untuk berbagai proyek IoT karena salah satu pilihan yang ideal untuk membangun Solusi IoT yang terintegrasi dan cerdas [12].

Node MCU ESP32 memiliki dual core clock dengan kecepatan tinggi, jumlah pin yang banyak, dan juga flash memory yang besar, membuat ideal untuk menjalankan aplikasi IoT yang kompleks. Perangkat ini mudah digunakan dan di program. Fitur utama dalam Node MCU ESP32 seperti, Wi-Fi 802.11 b/g/n, Bluetooth 4.2, Dual Core clock hingga 240 MHz, 36 Pin GPIO, 16 Kanal PWM, Flash Memory 4MB [13].

## 3. Perancangan Alat

Proses pembuatan sistem monitoring RPM ini terdiri atas beberapa tahap yang terdiri atas analisis kebutuhan, perancangan, pembuatan, dan pengujian. Berikut ini merupakan diagram alir dari proses pembuatan alat

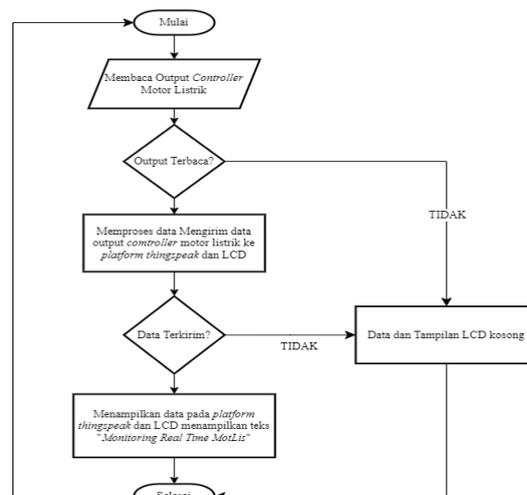


**Fig. 1.** Diagram Alir Penelitian

Analisis kebutuhan merupakan tahapan analisis mengenai urgensi pembuatan sistem monitoring RPM. Tahapan analisis kebutuhan ini adalah dengan melakukan observasi lapangan dan juga studi literatur. Observasi lapangan dilakukan pada kendaraan motor listrik yang ada pada bengkel konversi motor listrik satuso EV Yogyakarta. Proses monitoring motor listrik dengan perangkat keras software dinilai kurang efektif karena tidak bisa dilakukan secara real-time dan tidak berfungsi baik dalam kondisi berat seperti saat berbeban atau melawan tanjakan. Oleh karena itu, diperlukan sistem monitoring berbasis IoT yang dapat memantau parameter komponen motor listrik, termasuk controller motor, motor BLDC, RPM, serta penggunaan tegangan dan arus pada baterai secara real-time. Sedangkan studi literatur dilakukan untuk mencari solusi dari permasalahan tersebut berdasarkan dari berbagai literatur. Literatur yang didapatkan berupa artikel yang didapatkan dari berbagai jurnal.

### 3.1 Diagram Alir Sistem

Berikut ini merupakan *flowchart* Sistem Monitoring Rpm Sepeda Motor Listrik Menggunakan *Protocol Communication (Hexadecimal) Controller Votol Secara Real Time* Berbasis Iot.



**Fig. 2.** Diagram Alir Sistem

Gambar 2 merupakan *flowchart* Sistem Monitoring Rpm Sepeda Motor Listrik Menggunakan

*Protocol Communication (Hexadecimal) Controller Votol Secara Real Time Berbasis Iot.* Berdasarkan flowchart sistem pada Gambar 2, proses kerja alat dimulai dengan pembacaan keluaran controller motor listrik yang berupa data Hexadecimal dan dikonversi menjadi data desimal untuk mendapatkan parameter seperti tegangan, arus, daya, performa, dan kecepatan (RPM) motor listrik. Sistem mendeteksi apakah parameter dari controller terbaca atau tidak; jika tidak terbaca, data pada monitoring akan kosong dan sistem akan mereset ulang, sedangkan jika terbaca, parameter akan diperbarui pada LCD dan dikirim ke Platform Think Speak. Jika data tidak terkirim, tampilan monitoring tetap kosong dan sistem melakukan deteksi ulang. Jika terkirim, data akan ditampilkan pada website monitoring Think Speak, memperbarui tegangan dan arus yang dihasilkan oleh controller.

### 3.2 Desain Hardware

Proses pembuatan *hardware* terdiri atas dua bagian, yaitu pembuatan desain elektrik dan desain mekanikal. Pembuatan desain elektrik mencakup pembuatan *wiring* diagram kelistrikan, sedangkan pembuatan desain mekanikal mencakup pembuatan desain bentuk fisik alat dan komponen mekanik dari alat yang akan dibuat. Berikut ini merupakan hasil pembuatan desain *hardware*.

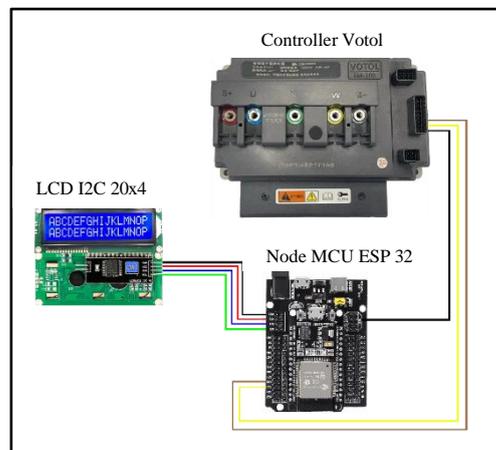


Fig. 3. Wiring Komponen

Gambar 3 di atas merupakan gambar wiring dari setiap komponen yang digunakan pada sistem monitoring RPM. Komponen utama sistem monitoring RPM yang digunakan berdasarkan gambar di atas adalah controller votol, Mikrokontroler Node MCU ESP 32, dan LCD 20x4

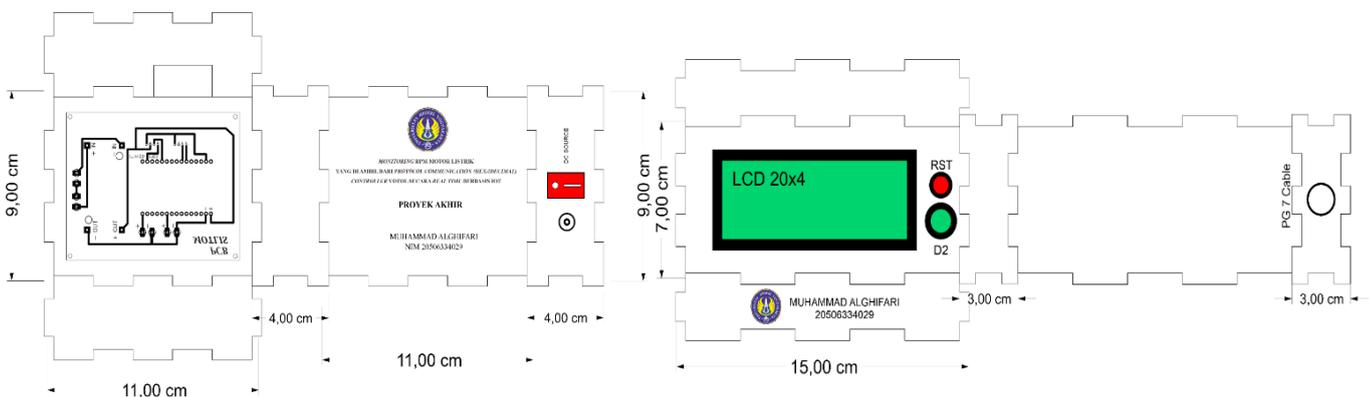


Fig. 4. Desain Alat

Gambar 4 di atas merupakan gambar dari box komponen dan box LCD dari sistem monitoring RPM. Box tersebut digunakan untuk meletakkan komponen PCB dari setiap komponen yang telah

dibuat dan melektakkan LCD dan tombol.

### 3.3 Desain Software

Tahapan ini dilakukan untuk mengetahui proses pembuatan desain *software* dari Sistem *Monitoring Rpm Sepeda Motor Listrik Menggunakan Protocol Communication (Hexadecimal) Controller Votol Secara Real Time Berbasis Iot*. Tahapan dalam perancangan *software* ini dilakukan dalam dua tahap, yaitu tahap pembuatan program atau *coding* di aplikasi Arduino IDE dan pembuatan *dashboard* antar muka pada *platform ThingSpeak*.

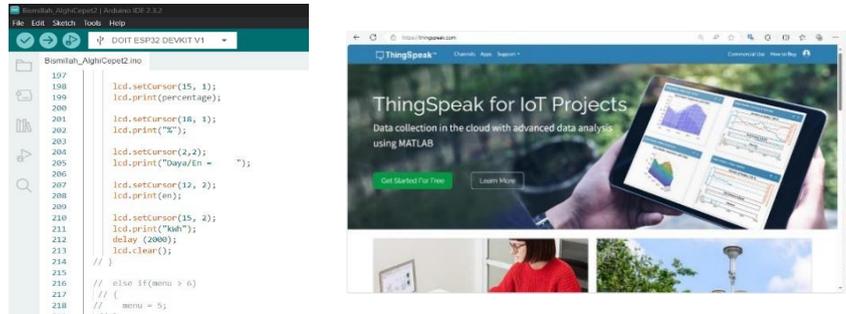


Fig. 5. Aplikasi Arduino IDE dan Platform ThingSpeak

## 4. Hasil dan Pembuatan

### 4.1 Hasil Pembuatan Hardware Sistem

Hasil pembuatan *hardware* Sistem *Monitoring Rpm Sepeda Motor Listrik Menggunakan Protocol Communication (Hexadecimal)* terdiri atas pembuatan sistem elektrikal dan mekanikal. Sistem elektrikal yang terdiri atas beberapa komponen dibuat dalam sebuah PCB agar komponen yang digunakan menjadi lebih rapi dan aman. Setelah itu komponen elektrikal tersebut dimasukkan ke dalam box yang telah dibuat. Berikut ini merupakan gambar dari alat yang telah di buat.

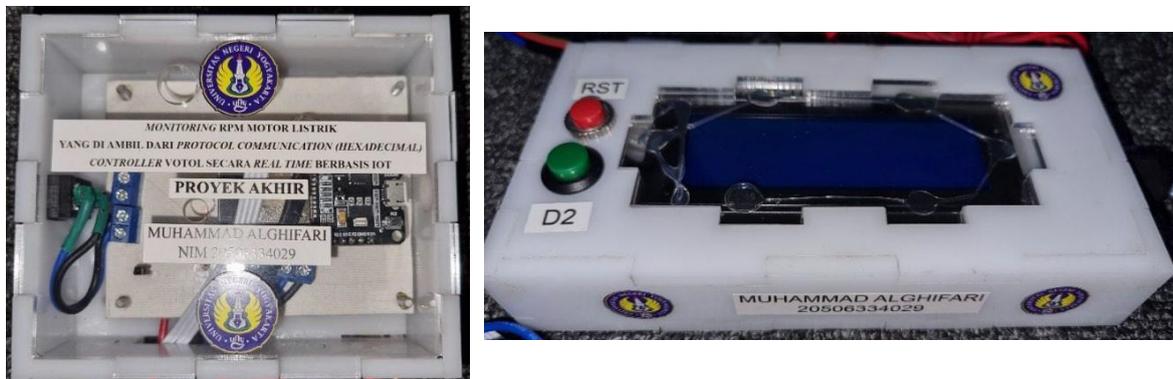


Fig. 6. Hasil Sistem monitoring RPM yang telah dibuat

Gambar 6 di atas merupakan hasil rancang bangun Sistem *Monitoring Rpm Sepeda Motor Listrik Menggunakan Protocol Communication (Hexadecimal)* yang telah dibuat. Berdasarkan tampak depan gambar tersebut, sistem monitoring terdapat 2 box yang pertama ada box untuk meletakkan PCB untuk Node MCU ESP32 kemudian box kedua untuk meletakkan LCD dan tombol reset dan display.

### 4.2 Hasil Pembuatan Software Sistem

Pembuatan *softwares* sistem ini terdiri atas pembuatan program pada mikrokontroler Node MCU ESP 32 yang dibuat dengan menggunakan *software* Arduino IDE. Mikrokontroler Node MCU ESP 32 ini berfungsi untuk mengontrol setiap komponen-komponen yang digunakan. Selain itu, mikrokontroler ini memiliki fitur *Wi-Fi* yang dapat digunakan untuk mengirimkan data pembacaan sensor ke *platform IOT*. Berikut ini merupakan hasil dari pembuatan program di Node MCU ESP 32.

```

1 #include <SPI.h>
2 #include "ThingSpeak.h"
3 #include <Wire.h>
4 #include <LiquidCrystal_I2C.h>
5
6 //Inisialisasi
7 #define ROTARY_A 32
8 #define ROTARY_B 33
9 #define PRESSED LOW
10
11 char ssid[] = "Indersky";
12 char pass[] = "minalmota1";
13 int keyIndex = 0;
14 WiFiClient client;
15
16 unsigned long myChannelNumber = 2489633;
17 const char *myAuthToken = "26687206c0a15";
18
19 LiquidCrystal_I2C lcd(0x27, 20, 4);
20
21 // Variabel waktu
22 unsigned long previousMillis = 0;
23 unsigned long previousMillis2 = 0;
24
25 const unsigned long Interval = 1500;
26 const unsigned long Interval2 = 3000;
27
28 float tegatt;
29 float arusatt;
30 int persentase;
31 int em;
32 int jarak = 0;
33 int jarak = 0;
    
```

Fig. 7. Program Arduino IDE

Setelah membuat program pada Arduino IDE, tahapan selanjutnya yaitu membuat *dashboard* tampilan pada *platform ThingSpeak*. *Dashboard* ini digunakan untuk menampilkan data hasil pengukuran berbagai parameter yang dikirimkan dari sistem monitoring RPM yang telah dibuat. Sehingga data yang telah didapatkan dapat dipantau secara *real-time* dari jarak jauh. Data yang akan ditampilkan dalam *dashboard* ini di antaranya adalah tegangan, Arus, RPM, dan Kecepatan. Tampilan yang di sediakan pada *dashboard* ini juga ada beberapa macam, dan dalam sistem monitoring ini menggunakan tampilan grafik dan gauge. Selain itu data yang telah dikirimkan juga dapat di unduh dalam bentuk *CSV*, sehingga dapat di buka pada *Microsoft Excel*.

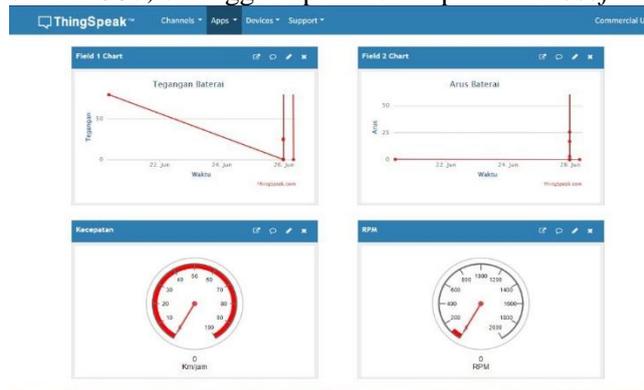


Fig. 8. Dashboard Tampilan pada Platform ThingSpeak

### 4.3 Hasil Pengujian Kecepatan Berbeban

Pengujian kecepatan dilakukan pada kondisi berbeban pada sistem monitoring RPM yang dimaksud dengan kondisi berbeban ialah motor listrik dalam kondisi berpenumpang dan di damping dengan motor BBM vario 125 untuk melihat parameter kecepatan yang ada. Berikut tabel pengujian sistem yang telah dilakukan.

Table 1 Hasil Pengujian Kecepatan Berbeban

Variasi Speed Mode	Hasil Pembacaan		Selisih Pembacaan (Kecepatan)	Persentase Error (%)	Kriteria Ketelitian
	LCD berdasarkan data Hexadecimal (Km/jam)	Km/jam (Motor Vario 125)			
Mode 1	32	33	1	3,03	Gol IV
Mode 2	48	49	1	2,04	Gol IV

Mode 3	54	55	1	1,81	Gol IV
Rata-Rata			1	2,29	Gol IV

Pada Tabel 1 di atas adalah hasil pengujian dengan metode *blacbox testing*. Hasil pengujian dapat diketahui jika km/jam dari nilai RPM pada sepeda motor listrik terlihat mengalami kenaikan yang sangat signifikan pada saat perubahan speed mode. Pada saat speed mode 1 ke 2 yaitu 16 km/jam kenaikannya dan pada saat speed mode 2 ke 3 yaitu 6 km/jam kenaikannya. Pada pembacaan di sepeda motor honda vario 125 juga mengalami kenaikan yang berbeda, juga terlihat kenaikan yang berbeda yaitu beda 1 km/jam pada setiap speed mode. Dari hasil pengujian tersebut pembacaan kecepatan (km/jam) dari nilai RPM yang dilakukan sistem *monitoring* ini mendapatkan hasil normal karena Berdasarkan standar IEC (*Internasional Electrotechnical Commission*) 13B-23 standar pengujian kecepatan masih  $\pm 2,5\%$  Gol IV sehingga dapat dikatakan bahwa sistem *monitoring* RPM yang digunakan masih layak.

#### 4.4 Hasil Pengujian Tanpa Beban

Pengujian tanpa beban merupakan tahapan pengujian kinerja sistem monitoring RPM pada saat motor listrik tidak berpenumpang. Berikut ini tabel-tabel yang merupakan hasil pengujian sistem yang telah dilakukan.

Table 2 Hasil Pengujian Tegangan Tanpa Beban

Kondisi <i>Speed Mode</i>	Hasil Pembacaan		Selisih Pembacaan (V)	Persentase <i>Error (%)</i>	Hasil
	LCD berdasarkan data <i>Hexadecimal</i> (V)	Alat Ukur Multimeter (V)			
20%	79,0	79,7	0,7	0,87	Gol III
40%	78,5	79,0	0,5	0,63	Gol III
60%	78,0	78,7	0,7	0,88	Gol III
80%	77,0	78,0	1,0	1,28	Gol III
100%	76,0	77,0	1,0	1,29	Gol III
Rata-Rata			0,78	0,99	Gol III

Pada Tabel 2 di atas, hasil pengujian dengan metode blackbox testing menunjukkan pengujian tegangan tanpa beban pada controller Votol yang menggunakan protokol komunikasi (Hexadecimal) terhubung dengan sistem monitoring RPM sepeda motor listrik. Pengujian ini membandingkan parameter tegangan pada sistem monitoring RPM dengan multimeter, dengan current limiting diubah dari 60A menjadi 20A dan menggunakan 5 kondisi speed mode (20%, 40%, 60%, 80%, dan 100%). Hasilnya menunjukkan tegangan mengalami penurunan rata-rata 0,78 volt pada setiap perubahan speed mode, tetapi perubahan ini tidak signifikan. Berdasarkan standar IEC 13B-23, pengujian tegangan berada dalam kisaran  $\pm 1,0\%$  Gol III, sehingga sistem monitoring RPM masih termasuk presisi rendah Gol III.

Table 3 Hasil Pengujian Arus Tanpa Beban

Kondisi <i>Speed Mode</i>	Hasil Pembacaan		Selisih Pembacaan (A)	Persentase <i>Error (%)</i>	Hasil
	LCD berdasarkan data <i>Hexadecimal</i> (A)	Alat Ukur Ampermeter DC (A)			
20%	4	4	0	0	Gol I
40%	8	8	0	0	Gol I
60%	11,9	11,9	0	0	Gol I
80%	16	16	0	0	Gol I

100%	20,6	20,6	0	0	Gol I
Rata-Rata			0	0	Gol I

Pada Tabel 23 di atas adalah hasil pengujian Arus dengan metode *blacbox testing*. Hasil pengujian dapat diketahui jika arus pada sepeda motor listrik terlihat mengalami kenaikan pada saat perubahan kondisi speed mode dan terlihat perubahannya yaitu  $\pm 4$  Ampere dari perpindahan kondisi *speed mode* 1 ke 2 dan seterusnya. Dari hasil pengujian tersebut pembacaan arus yang dilakukan sistem *monitoring* RPM mendapatkan hasil normal karena Berdasarkan standar IEC (*Internasional Electrotechnical Commission*) 13B-23 standar pengujian arus masih  $\pm 0,01\%$  Gol I sehingga dapat dikatakan bahwa sistem *monitoring* RPM yang digunakan masih termasuk kedalam golongan kepresisian tinggi.

Table 4 Hasil Pengujian RPM Tanpa Beban

Kondisi <i>Speed Mode</i>	Hasil Pembacaan		Selisih Pembacaan (RPM)	Persentase <i>Error (%)</i>	Hasil
	LCD berdasarkan data <i>Hexadecimal</i> (RPM)	Alat Ukur Tachometer (RPM)			
20%	1536	1552	16	1,03	Gol I
40%	3439	3455	16	0,46	Gol I
60%	3376	3397	21	0,61	Gol I
80%	3381	3410	29	0,85	Gol I
100%	5144	5165	21	0,4	Gol I
Rata-Rata			20,6	0,67	Gol I

Pada Tabel 4 di atas adalah hasil pengujian RPM dengan metode *blacbox testing*. Hasil pengujian dapat diketahui jika RPM pada sepeda motor listrik terlihat mengalami kenaikan yang tidak terlalu signifikan pada saat perubahan kondisi *speed mode*. Rata-rata selisih RPM yaitu 20,6 RPM, karena RPM pada saat perubahan kondisi sangat signifikan di kondisi 1 ke 2 dan kondisi 4 ke 5. Dari hasil pengujian tersebut pembacaan RPM yang dilakukan sistem *monitoring* RPM mendapatkan hasil normal karena Berdasarkan standar IEC (*Internasional Electrotechnical Commission*) 13B-23 standar pengujian RPM masih  $\pm 0,01\%$  Gol I sehingga dapat dikatakan bahwa sistem *monitoring* RPM yang digunakan masih termasuk kedalam golongan kepresisian tinggi.

Table 5 Hasil Pengujian Pada Thingspeak Tanpa Beban

Kondisi Speed Mode	Hasil Pembacaan <i>Thingspeak</i>			
	Tegangan (V)	Arus (A)	RPM	Hasil
20%	79,0	4	1536	Normal Sama Dengan Tampilan LCD I2C 20x4
40%	78,5	8	3439	
60%	78,0	11,9	3376	
80%	77,0	16	3381	
100%	76,0	20,6	5144	

Tabel 5 di atas merupakan hasil dari uji kinerja pada sistem *monitoring* RPM sepeda motor listrik menggunakan *protocol communication (Hexadecimal)* pada hasil pembacaan *Thingspeak*. Hasil pengujian di atas mengatakan bahwa pembacaan pada *Thingspeak* dan juga pembacaan pada

LCD I2C 20x4 sudah sesuai dan sama dengan pembacaan Tabel 1, Tabel 2, Tabel 3. Pembacaan pada *Thingspeak* ini berfungsi untuk memantau performa pada sepeda motor listrik setiap bulan atau tahunnya jika pengguna ingin mengetahui bagaimana performa dari sepeda motor listrik apakah performanya menurun atau performanya tetap stabil dari awal pembelian sampai pada saat pengecekan secara berkala. Berikut adalah Gambar 10 data file CSV hasil pengujian tanpa beban yang di export dari web *Thingspeak*.

	created_at	entry_id	Tegangan	Arus	RPM	Km/jam
168	22/05/2024 12:01	1208	7920000	130000	84300000	6300000
169	22/05/2024 12:02	1209	7920000	120000	84300000	6300000
170	22/05/2024 12:02	1210	7920000	130000	84300000	6300000
171	22/05/2024 12:02	1211	7920000	130000	84200000	6300000
172	22/05/2024 12:02	1212	7920000	120000	84300000	6300000
173	22/05/2024 12:03	1213	7920000	120000	84300000	6300000
174	22/05/2024 12:03	1214	7920000	130000	84300000	6300000
175	22/05/2024 12:03	1215	7920000	120000	84300000	6300000
176	22/05/2024 12:04	1216	7920000	120000	84300000	6300000
177	22/05/2024 12:04	1217	7920000	120000	84300000	6300000
178	22/05/2024 12:04	1218	7920000	120000	84300000	6300000
179	22/05/2024 12:04	1219	7920000	120000	84300000	6300000
180	22/05/2024 12:05	1220	7930000	120000	82600000	6200000
181	22/05/2024 12:05	1221	7920000	120000	84300000	6300000
182	22/05/2024 12:05	1222	7920000	120000	84300000	6300000
183	22/05/2024 12:05	1223	7920000	120000	84300000	6300000
184	22/05/2024 12:06	1224	7920000	120000	84300000	6300000
185	22/05/2024 12:06	1225	7920000	120000	84300000	6300000
186	22/05/2024 12:06	1226	7920000	120000	84300000	6300000
187	22/05/2024 12:06	1227	7950000	0	0	2000000
188	22/05/2024 12:07	1228	7940000	80000	60300000	4500000
189	22/05/2024 12:07	1229	7940000	80000	60300000	4500000
190	22/05/2024 12:07	1230	7930000	130000	83800000	6300000
191	22/05/2024 12:07	1231	7920000	190000	118300000	8800000
192	22/05/2024 12:08	1232	7910000	190000	118500000	8800000

Fig. 10. Data file CSV hasil pengujian tanpa beban dari platform *Thingspeak*

## 5. Kesimpulan

Proyek akhir ini berjudul “Sistem *Monitoring* RPM Sepeda Motor Listrik Menggunakan *Protocol Communication (Hexadecimal)* Secara *Real Time* Berbasis *IoT*” yang fungsinya untuk mempermudah pengguna dalam *me-monitoring* sepeda motor listrik. Parameter yang diukur pada sistem *monitoring* RPM sepeda motor listrik ini adalah tegangan, arus, RPM. Hasil pembacaan RPM tersebut digunakan juga untuk menghitung nilai km/jam yang digunakan pada pengujian berbeban parameter RPM. Sistem *monitoring* ini juga menggunakan mikrocontroller Node MCU ESP 32. Node MCU ESP 32 ini mempunyai fitur Wi-Fi yang berfungsi untuk mengirimkan data para meter ke Platform *Thingspeak* supaya dapat di *monitoring* secara *real time* dan dapat menganalisis data tersebut melalui file csv yang di *export* dari Platform *Thingspeak*. Pengujian dari pemecahan data data *Hexadecimal* juga berhasil yang diartikan dengan mendapatkan balasan dari *controller* votol. Balasan data *Hexadecimal* tersebut adalah 23 data *byte* yang dapat di artikan sebagai parameter sistem *monitoring* RPM sepeda motor listrik, dan di artikan menjadi desimal yaitu sebagai berikut,

- 1) Data Hex *byte* ke 5 dan 6 mengartikan sebagai Tegangan baterai
- 2) Data Hex *byte* ke 7 dan 8 mengartikan sebagai Arus
- 3) Data Hex *byte* ke 14 dan 15 mengartikan sebagai RPM

Hasil dari pengujian fungsional yang diperuntukkan untuk komponen yang ada pada sistem *monitoring* RPM sepeda motor listrik. Dari pengujian fungsional menunjukkan bahwa masing-masing komponen dapat berfungsi dengan normal tanpa ada kendala, seperti pada komponen *controller* votol yang dapat berfungsi dengan baik pada saat di beri tegangan 72 v kemudian di hubungkan dengan BLDC motor dan *throttle body electric*. Hasil pengujian Node MCU ESP 32 dan LCD I2C 20x4 juga berfungsi dengan baik pada saat di berikan *input* program yang di upload melalui *software* Arduino IDE.

Hasil pengujian kinerja dari sistem *monitoring* RPM sepeda motor listrik yang sudah dilakukan juga berjalan dengan baik sesuai fungsi yang diinginkan. Nilai parameter yang dihasilkan juga sesuai dan sama dengan yang ada pada *controller* votol. Parameter yang tampil pada LCD I2C 20x4 juga sama dengan yang tampil pada *Dashboard Thingspeak*. Nilai eror pada setiap parameter yang ada

berdasarkan standar IEC (*Internasional Electrotechnical Commission*) 13B-23 standar pengujian dari nilai RPM  $\pm 0,01\%$  Gol I sehingga termasuk kedalam golongan kepresisian tinggi, selanjutnya pada parameter tegangan berdasarkan IEC (*Internasional Electrotechnical Commission*) 13B-23 standar pengujian tegangan masih  $\pm 1,0\%$  Gol III sehingga termasuk kedalam tingkat presisi lebih rendah dari gol II, dan pada parameter arus yaitu  $\pm 0,01\%$  Gol I sehingga dapat dikatakan bahwa sistem *monitoring* RPM yang digunakan masih terasuk kedalam golongan kepresisian tinggi.

Pada pengujian kecepatan berbeban hasil nilai rata-rata eror adalah 2,29%. Berdasarkan standar IEC (*Internasional Electrotechnical Commission*) 13B-23 standar pengujian Kecepatan masih  $\pm 2,5\%$  Gol IV sehingga dapat dikatakan bahwa sistem *monitoring* RPM yang digunakan masih layak.

## Referensi

- [1] P. R. L. Silalahi, "MODIFIKASI SEPEDA MOTOR HONDA BEAT KARBU 110 CC MENJADI ELECTRIC VEHICLE 2 kW," Tarakan, Oct. 2023.
- [2] D. Despa, G. F. Nama, M. A. Muhammad, and K. Anwar, "The Implementation Internet of Things(IoT) Technology in Real Time Monitoring of Electrical Quantities," *IOP Conf Ser Mater Sci Eng*, vol. 335, no. 1, Apr. 2018, doi: 10.1088/1757-899X/335/1/012063.
- [3] R. Fili *et al.*, "PERANCANGAN SISTEM MONITORING MOBIL LISTRIK," *Jurnal SIMETRIS*, vol. 12, no. 2, 2021.
- [4] U. Primadi.R, "SISTEM MONITORING RPM MOTOR LISTRIK MELALUI PERANGKAT TELEPON PINTAR BERBASIS IOT," Surakarta, Jan. 2019.
- [5] Tim Fisher, "What Is Hexadecimal? How to count in the hexadecimal number system using 0–9 and A-F," <https://www.lifewire.com/what-is-hexadecimal-2625897>.
- [6] M. R. Ramadhan, D. A. Asfani, I. Made Yulistya Negara, and Y. U. Nugraha, "Performance Comparison of BLDC and SRM on Electric Motorcycle," in *7th International Conference on Electrical, Electronics and Information Engineering: Technological Breakthrough for Greater New Life, ICEEIE 2021*, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/ICEEIE52663.2021.9616708.
- [7] N. Masudi, "DESAIN CONTROLLER MOTOR BLDC UNTUK MENINGKATKAN PERFORMA ( DAYA OUTPUT ) SEPEDA MOTOR LISTRIK," Surabaya, Jul. 2014.
- [8] M. Afsari, A. Damaiyanti, and N. Sa'adah, "Implementasi Mode Operasi Kombinasi Cipher Block Chaining dan Metode LSB-1 Pada Pengamanan Data text," *Jurnal Pendidikan Sains dan Komputer*, vol. 2, no. 1, pp. 2809–476, 2022, doi: 10.47709/jpsk.v2i1.1381.
- [9] Learn More About ThingSpeak, "Learn More About ThingSpeak."
- [10] D. Nettikadan and S. Raj, "Smart Community Monitoring System using Thingspeak IoT Plaform," 2018. [Online]. Available: <http://www.ripublication.com>
- [11] Muh. A. J. Hidayat and A. Z. Amrullah, "SISTEM KONTROL DAN MONITORING TANAMAN HIDROPONIK BERBASIS INTERNET OF THINGS (IoT) MENGGUNAKAN NODEMCU ESP32," *Saintekom*, vol. 12, no. 1, Mar. 2022.
- [12] S. Shafiudin, F. J. Rohma, A. E. Prasetya, and R. Firmansyah, "Pemantauan Ruang Inkubator Penetasan Telur Ayam Dengan Berbasis Telemetri Menggunakan Arduino Uno R3," *JURNAL NASIONAL TEKNIK ELEKTRO*, vol. 5, no. 1, Mar. 2016, doi: 10.20449/jnte.v5i1.181.
- [13] M. Babiuch, P. Folytnek, and P. Smutny, "Using the ESP32 Microcontroller for Data Processing," *IEEE*, May 2019, Accessed: Apr. 01, 2024. [Online]. Available: DOI: 10.1109/CarpathianCC.2019.8765944