

LAPORAN PROYEK AKHIR

**IMPLEMENTASI SISTEM PEMANTAUAN PENGGUNAAN ENERGI LISTRIK
RUMAH TANGGA BERBASIS TELEGRAM BOT DAN *WEBHOSTING***



Diajukan oleh :

Riuna Sely

19/447305/SV/16999

PROGRAM SARJANA TERAPAN TEKNOLOGI REKAYASA INTERNET

DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA

SEKOLAH VOKASI

UNIVERSITAS GADJAH MADA

YOGYAKARTA

2023

**IMPLEMENTASI SISTEM PEMANTAUAN PENGGUNAAN ENERGI LISTRIK
RUMAH TANGGA BERBASIS TELEGRAM BOT DAN *WEBHOSTING***

**Proyek Akhir
Program Studi Teknologi Rekayasa Internet**

**Diajukan kepada
Departemen Teknik Elektro dan Informatika
Fakultas Sekolah Vokasi Universitas Gadjah Mada
Sebagai syarat kelengkapan studi jenjang Sarjana Terapan (D-IV)
Dalam memperoleh derajat Sarjana Terapan Teknik
Program Studi Teknologi Rekayasa Internet**

**Oleh :
Riuna Sely
19/447305/SV/16999**

**PROGRAM SARJANA TERAPAN TEKNOLOGI REKAYASA INTERNET
DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA
SEKOLAH VOKASI
UNIVERSITAS GADJAH MADA
YOGYAKARTA
2023**

HALAMAN PENGESAHAN

Judul : Implementasi Sistem Pemantauan Penggunaan Energi Listrik
Rumah Tangga Berbasis Telegram Bot dan *Webhosting*
Nama : Riuna Sely
Program Studi : Teknologi Rekayasa Internet
Dosen Pembimbing : Nur Rohman Rosyid, S.T., M.T., D.Eng
Waktu Ujian : Jumat, 7 Juli 2023

Telah dipertanggungjawabkan dan diuji oleh tim penguji serta disetujui dan disahkan

Sebagai syarat kelengkapan studi jenjang Sarjana Terapan

Program Studi Teknologi Rekayasa Internet

Sekolah Vokasi Universitas Gadjah Mada

Tim Penguji

Ketua Penguji



Dr. Ronald Adrian, S.T., M.Eng.
NIKA. 111199002201608101

Anggota Penguji



Budi Bayu Murti, S.T., M.T.
NIP. 197212231999031001

Sekretaris Penguji



Yuris Mulya Saputra, S.T., M.Sc., Ph.D.
NIP. 111198806201605101

Mengetahui,

a.n Ketua Departemen
Teknik Elektro dan Informatika,
Sekretaris Departemen



Yuris Mulya Saputra, S.T., M.Sc., Ph.D.
NIP. 111198806201605101

Ketua Program Studi
Sarjana Terapan
Teknologi Rekayasa Internet



Dr. Ronald Adrian, S.T., M.Eng.
NIKA. 111199002201608101

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini :

Nama : RIUNA SELY
NIM : 19/447305/SV/16999
Tahun Terdaftar : 2019
Program Studi : Teknologi Rekayasa Internet
Fakultas/Sekolah : Sekolah Vokasi

Menyatakan bahwa dalam dokumen ilmiah Tugas Akhir ini tidak terdapat bagian dari karya ilmiah lain yang telah diajukan untuk memperoleh gelar akademik di suatu lembaga Pendidikan Tinggi, dan juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang/lembaga lain, kecuali yang secara tertulis disitasi dalam dokumen ini dan disebutkan sumbernya secara lengkap dalam daftar pustaka.

Dengan demikian saya menyatakan bahwa dokumen ilmiah ini bebas dari unsur-unsur plagiasi dan apabila dokumen ilmiah Tugas Akhir ini di kemudian hari terbukti merupakan plagiasi dari hasil karya penulis lain dan/atau sengaja mengajukan karya atau pendapat yang merupakan hasil karya penulis lain, maka penulis bersedia menerima sanksi akademik dan/atau sanksi hukum yang berlaku.

Yogyakarta, 09 Juli 2023


Riuna Sely
19/447305/SV/16999

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas rahmat dan hidayah-Nya, penulis selalu mendapatkan pertolongan, kesehatan, dan keselamatan sehingga dapat menyelesaikan proyek akhir dengan judul “Implementasi Sistem Pemantauan Penggunaan Energi Listrik Rumah Tangga Berbasis Telegram Bot dan *Webhosting*”. Adapun tujuan dari penyusunan proyek akhir ini adalah untuk memenuhi salah satu syarat dalam mendapatkan gelar Sarjana Terapan Program Studi Teknologi Rekayasa Internet, Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada, Yogyakarta.

Pada kesempatan ini, penulis menyadari bahwa banyak pihak yang terlibat dalam memberikan dukungan dan bantuan selama proses penyusunan laporan proyek akhir hingga selesai. Oleh karena itu, dengan penuh kehormatan, penulis mengucapkan terima kasih kepada :

1. Bapak Nur Rohman Rosyid, S.T., M.T., D.Eng selaku Ketua Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada yang telah memberikan izin untuk melakukan penelitian serta memberikan persetujuan terhadap laporan proyek akhir ini, sekaligus sebagai Dosen Pembimbing Proyek Akhir yang senantiasa membantu dan meluangkan waktu, tenaga, dan pikiran selama penulis menyusun proyek akhir ini.
2. Bapak Ronald Adrian, S.T., M.Eng. selaku Ketua Program Studi Sarjana Terapan Teknologi Rekayasa Internet, Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada yang telah membimbing dan memberikan pengarahan kepada penulis selama melaksanakan perkuliahan di Program Studi Sarjana Terapan Teknologi Rekayasa Internet.
3. Seluruh dosen, staff pengajar, dan karyawan Program Studi Sarjana Terapan Teknologi Rekayasa Internet, Departemen Teknik Elektro dan Informatika, Sekolah Vokasi, Universitas Gadjah Mada.
4. Bapak Suparno selaku ayah kandung penulis dan Ibu Suliani selaku ibu kandung penulis yang telah memenuhi dan mendukung segala kebutuhan penulis baik secara moril maupun materiil, serta memotivasi dan mengajarkan kepada penulis untuk mampu bertahan hidup di tanah perantauan, sehingga penulis mampu menyelesaikan proyek akhir ini.
5. Rafi Dinno Darmawan selaku adik kandung penulis yang membangkitkan semangat dan saling berbagi suka duka sampai proyek akhir dapat terselesaikan.

6. VDMI *Family* yang telah memberikan dukungan, baik bekal ilmu dan finansial sehingga penulis dapat menyelesaikan proyek akhir tepat waktu.
7. Teman-teman seperjuangan Teknologi Rekayasa Internet yang telah berjuang bersama untuk dapat menyelesaikan proyek akhir ini.
8. Teman-teman KKN YO023 Wonosari 2022 Subunit 4, Wulan, Riris, Kurnia, Rasyad, Winaldy, dan Aldhi yang senantiasa mendukung dan memotivasi, sehingga penulis mampu menyelesaikan proyek akhir ini.
9. Sahabat-sahabat dan saudara, Adelya, Vivi, Amel, Redha, Dhila, Amalia, Mbak Krida, Mbak Erlin, Mas Haqiqie yang telah memberikan dukungan dan motivasi kepada penulis hingga proyek akhir dapat terselesaikan.
10. Berbagai pihak yang tidak dapat disebutkan satu per satu yang telah membantu dalam penyelesaian proyek akhir ini.

Penulis menyadari bahwa Laporan Proyek Akhir ini masih sangat jauh dari kata sempurna. Oleh karena itu, penulis berharap adanya kritik dan saran yang bersifat membangun untuk menyempurnakan penulisan Laporan Proyek akhir di kemudian hari. Akhir kata, semoga Laporan Proyek Akhir ini dapat bermanfaat dan senantiasa dapat dikembangkan untuk penelitian selanjutnya.

Yogyakarta, 09 Juli 2023



Riina Sely
19/447305/SV/16999

DAFTAR ISI

HALAMAN PENGESAHAN.....	ii
PERNYATAAN BEBAS PLAGIASI.....	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
INTISARI.....	xii
<i>ABSTRACT</i>	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian.....	2
1.4 Batasan Masalah	2
1.5 Manfaat Penelitian.....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 Tinjauan Pustaka.....	4
2.2 Dasar Teori	10
2.2.1 <i>Internet of Things</i>	10
2.2.2 NodeMCU ESP 8266	10
2.2.3 Sensor PZEM-004T.....	11
2.2.4 Modul LCD 16x2	13
2.2.5 Modul I2C	13
2.2.6 <i>Webhosting</i>	14
2.2.7 MySQL.....	14
2.2.8 Telegram Bot.....	14
2.3 Hipotesis	14
BAB III METODOLOGI PENELITIAN.....	15
3.1 Alat Penelitian	15
3.2 Bahan Penelitian	15

3.3	Tahapan Penelitian	16
3.3.1	Tahapan Umum	16
3.3.2	Tahapan Teknis	17
3.4	Diagram Blok dan Perancangan Sistem	18
3.5	Instalasi dan Konfigurasi Program pada Arduino IDE.....	19
3.5.1	Instalasi Arduino IDE.....	19
3.5.2	Instalasi <i>Library</i>	22
3.5.3	Pembuatan Program <i>Setup</i>	25
3.5.4	Pembuatan Program <i>Loop</i>	26
3.5.5	Pembuatan Fungsi <i>Reset</i>	30
3.6	Registrasi dan Konfigurasi Webhosting dan <i>Database</i>	30
3.6.1	Registrasi <i>Webhosting</i>	30
3.6.2	Pembuatan <i>Website Domain</i>	31
3.6.3	Pembuatan <i>Database</i> dan Pengguna	32
3.6.4	Pembuatan Skrip HTTP POST Menggunakan PHP	33
3.6.5	Pembuatan Skrip Isi <i>Database</i> Menggunakan PHP	34
3.6.6	Pembuatan Skrip Grafik Visualisasi	35
3.7	Skenario Pengujian	36
3.7.1	Pengujian Fungsional Sistem	37
3.7.2	Pengujian Kinerja Sistem	37
BAB IV HASIL PENELITIAN DAN PEMBAHASAN		39
4.1	Hasil Pengujian Fungsional Sistem	39
4.1.1	Hasil Rangkaian Komponen.....	39
4.1.2	Hasil Pengujian pada Layar LCD.....	39
4.1.3	Hasil Pengujian pada <i>Serial Monitor</i>	40
4.1.4	Hasil Pengujian pada <i>Database</i>	40
4.1.5	Hasil Pengujian Tabel pada Tampilan <i>Webhosting</i>	41
4.1.6	Hasil Pengujian Grafik pada Tampilan <i>Webhosting</i>	41
4.1.7	Hasil Pengujian pada Telegram.....	42
4.2	Hasil Pengujian Kinerja Sistem	43
4.2.1	Hasil Pengujian Akurasi Sistem.....	43
4.2.2	Hasil Pengujian Konektivitas	45

BAB V PENUTUP.....	48
5.1 Kesimpulan.....	48
5.2 Saran	48
DAFTAR PUSTAKA	49
LAMPIRAN.....	51

DAFTAR GAMBAR

Gambar 2. 1 NodeMCU ESP8266	11
Gambar 2. 2 Sensor PZEM-004T	11
Gambar 2. 3 Datasheet Sensor PZEM-004T	12
Gambar 2. 4 Modul LCD	13
Gambar 2. 5 Modul I2C	13
Gambar 3. 1 Tahapan Umum Penelitian.....	16
Gambar 3. 2 Tahapan Teknis Penelitian	17
Gambar 3. 3 Diagram Blok Sistem	18
Gambar 3. 4 Perancangan Sistem Menggunakan Fritzing	18
Gambar 3. 5 Hasil Unduhan Arduino IDE	20
Gambar 3. 6 Halaman License Agreement Arduino IDE	20
Gambar 3. 7 Opsi Instalasi Pengguna Arduino IDE	20
Gambar 3. 8 Lokasi Instalasi Arduino IDE.....	21
Gambar 3. 9 Halaman Awal Arduino IDE	21
Gambar 3. 10 Menu Preferences Arduino IDE	22
Gambar 3. 11 Menu Manage Libraries	22
Gambar 3. 12 Kolom Pencarian Library	23
Gambar 3. 13 Folder Libraries Arduino.....	23
Gambar 3. 14 Library yang Digunakan.....	24
Gambar 3. 15 Program Setup	25
Gambar 3. 16 Fungsi Loop.....	26
Gambar 3. 17 Fungsi If	26
Gambar 3. 18 Fungsi Else If.....	27
Gambar 3. 19 Fungsi Else	27
Gambar 3. 20 Fungsi SendMessageToAll.....	27
Gambar 3. 21 String Message ke Telegram	28
Gambar 3. 22 Notifikasi Peringatan Kondisi	28
Gambar 3. 23 Koneksi ke Protokol HTTPS	28
Gambar 3. 24 Memulai HTTPS dan Menambahkan Header	29
Gambar 3. 25 Cetak Data HTTP dan HTTP Response Code.....	29

Gambar 3. 26 Fungsi If pada HTTP Response Code	29
Gambar 3. 27 Fungsi Reset	30
Gambar 3. 28 Paket Webhosting.....	30
Gambar 3. 29 Halaman Registrasi Webhosting	31
Gambar 3. 30 Halaman Pembuatan Website Baru	31
Gambar 3. 31 Halaman Pembuatan Database Baru	32
Gambar 3. 32 Spesifikasi Database.....	32
Gambar 3. 33 Tabel Database yang Dibuat.....	32
Gambar 3. 34 Folder pada File Manager.....	33
Gambar 3. 35 File Terdaftar	33
Gambar 3. 36 Potongan File post-esp-data.php	33
Gambar 3. 37 Potongan File esp_data.php.....	34
Gambar 3. 38 Potongan File esp_chart.php	35
Gambar 3. 39 Potongan Pengaturan esp_chart.php.....	35
Gambar 3. 40 Skenario Pengujian Sistem	36
Gambar 4. 1 Rangkaian Komponen.....	39
Gambar 4. 2 Tampilan Layar LCD	39
Gambar 4. 3 Tampilan pada Serial Monitor	40
Gambar 4. 4 Tampilan pada Database	40
Gambar 4. 5 Tabel pada Tampilan Webhosting.....	41
Gambar 4. 6 Grafik pada Tampilan Webhosting	41
Gambar 4. 7 Tampilan pada Telegram.....	42
Gambar 4. 8 Notifikasi Peringatan pada Telegram	42
Gambar 4. 9 Perbandingan Pengujian Tegangan	43
Gambar 4. 10 Perbandingan Pengujian Arus	44

DAFTAR TABEL

Tabel 2. 1 Ringkasan Tinjauan Pustaka	7
Tabel 3. 1 Datasheet Port PZEM-004T V3.0 ke NodeMCU ESP8266.....	19
Tabel 3. 2 Datasheet Port I2C ke NodeMCU ESP8266	19
Tabel 3. 3 Kategori Delay Menurut TIPHON	38
Tabel 4. 1 Hasil Pengujian Tegangan dengan Multimeter.....	43
Tabel 4. 2 Hasil Pengujian dengan Sensor	43
Tabel 4. 3 Hasil Pengujian Akurasi Tegangan dan Arus	44
Tabel 4. 5 Hasil Pengujian Delay ke Tabel Webhosting.....	45
Tabel 4. 6 Hasil Pengujian Delay ke Grafik Webhosting	45
Tabel 4. 7 Hasil Pengujian Delay ke Telegram.....	46
Tabel 4. 8 Pengujian Ukuran Data Terpakai ke Tabel Webhosting	46
Tabel 4. 9 Pengujian Ukuran Data Terpakai ke Grafik Webhosting	47
Tabel 4. 10 Pengujian Ukuran Data Terpakai ke Telegram	47

INTISARI

Dalam menjalankan aktivitas sehari-hari, masyarakat tidak terlepas dari penggunaan energi listrik. Walaupun pada penelitian terdahulu telah ditemukan adanya energi alternatif, hal tersebut belum sepenuhnya dapat menggantikan peranan energi listrik. Hingga saat ini, masyarakat mendapatkan suplai energi dari PT Perusahaan Listrik Negara (PLN). Oleh karena itu, untuk menunjang kemudahan masyarakat dalam melakukan pemantauan penggunaan energi listrik rumah tangga, penulis mengusulkan sistem pemantauan energi listrik rumah tangga berbasis *webhosting* dan Telegram Bot menggunakan modul PZEM-004T V3.0 dan NodeMCU ESP8266. Dalam implementasinya, dilengkapi dengan adanya notifikasi dan peringatan dini apabila terdapat kondisi bahaya pada variabel tegangan dan arus agar tetap aman meskipun pengguna berada pada jarak jauh. Hasil pengujian menunjukkan akurasi alat sebesar 99.704% pada pengujian tegangan dan 99.646% pada pengujian arus yang menunjukkan sensor berjalan normal. Kualitas *delay* berkisar antara 150-300 ms atau berada dalam kategori baik dan nilai *delay* terkecil didapatkan dari pengujian pengiriman data Arduino IDE ke Telegram.

Kata kunci : *Internet of Things*, sensor PZEM-004T, NodeMCU ESP8266, energi listrik, *webhosting*, *mySQL*, Telegram Bot

ABSTRACT

In daily activities, people cannot be separated from the use of electrical energy. Although previous studies have found alternative energy, it has not completely replaced the role of electrical energy. Until now, the community gets their energy supply from PT Perusahaan Listrik Negara (PLN). Therefore, to support the convenience of the community in monitoring household electrical energy use, the authors propose a web hosting and Telegram Bot-based household electrical energy monitoring system using the PZEM-004T V3.0 module and NodeMCU ESP8266. In this system, Telegram Bot can send notifications and early warnings if there are dangerous conditions on the voltage and current variables so that they remain safe even though the user is at a distance. The test results show that the accuracy of the tool is 99.704% in the voltage test and 99.646% in the current test which shows the sensor is running normally. The quality of the delay ranges from 150-300 ms or is in the good category and the smallest delay value is obtained from testing Arduino IDE data sending to Telegram.

Keywords: Internet of Things, PZEM-004T sensor, NodeMCU ESP8266, electrical energy, webhosting, mySQL, Telegram Bot

BAB I PENDAHULUAN

1.1 Latar Belakang

Energi listrik menjadi salah satu kebutuhan yang paling mendasar dan cukup krusial bagi kehidupan masyarakat dalam menjalankan aktivitas sehari-hari, mulai dari skala rumahan hingga instansi maupun tempat umum. Terbukti dari data statistik PLN tahun 2021, disebutkan bahwa jumlah energi listrik yang terjual sebesar 257.634,25 GWh. Dari data tersebut, sebanyak 115.370,05 GWh atau sebesar 44,78% dari total keseluruhan digunakan oleh keperluan listrik rumah tangga (Data Statistik PLN, 2021). Meskipun beberapa penelitian telah menemukan energi alternatif, namun belum dapat untuk menggantikan posisi penggunaan energi listrik secara maksimal (Agung, 2013). Berdasarkan fakta tersebut, masyarakat harus berfokus pada optimalisasi konsumsi energi listrik dengan cara mendukung keberlanjutan dan efisiensi energi listrik.

Masyarakat dapat memanfaatkan kemajuan zaman sebagai upaya untuk mendukung keberlanjutan dan efisiensi dalam menggunakan energi listrik. Salah satu teknologi yang dapat dimanfaatkan adalah IoT (*Internet of Things*). Penggunaan IoT dapat berupa pemanfaatan teknologi informasi, konektivitas jaringan internet, dan sensor untuk memungkinkan perangkat saling terintegrasi satu sama lain melalui jaringan internet secara *real-time*, dimana perangkat nantinya akan berfungsi sebagai penghasil data, pengirim dan penerima data, serta pengumpul data (Prayitno, B., 2019). Semua mesin yang menggunakan identitas berupa IP *address* dan menggunakan jaringan internet sebagai media komunikasi. Hal tersebut akan mempermudah masyarakat dalam melakukan interaksi dan integrasi dengan perangkat. Adapun pengembangan dari penggunaan IoT akan membawa dampak besar bagi masyarakat dengan peran di bidang manajemen ekonomi, operasi produksi, sosial, politik, hingga kehidupan pribadi, terutama perihal efisiensi dan efektivitas pekerjaan (Ritonga, A. F., 2020).

Oleh karena itu, penulis mengusulkan solusi berupa sistem pemantauan energi listrik rumah tangga berbasis *Internet of Things* melalui *webhosting* dan notifikasi melalui Telegram Bot. Sistem nantinya dapat memantau penggunaan energi listrik dengan memanfaatkan kinerja dari sensor PZEM-004T yang memungkinkan pengguna untuk melakukan pemantauan tegangan, arus, daya, dan energi listrik, serta

penggunaan mikrokontroler NodeMCU ESP8266 yang dilengkapi dengan modul WiFi untuk dapat dilakukan pemantauan secara *real-time* melalui *webhosting* maupun Telegram Bot. Sistem juga dilengkapi dengan adanya notifikasi peringatan dini ketika nilai tegangan dan arus yang tidak stabil melalui Telegram Bot yang memungkinkan pengguna untuk memantau jarak jauh dan mengantisipasi munculnya risiko bahaya. Adapun lingkup penelitian akan berfokus pada beberapa pengujian, yaitu pengujian akurasi sensor dan pengujian *Quality of Services* (QoS) pada parameter *delay* dan kapasitas data yang terpakai untuk dibandingkan dan dilakukan analisis serta pembahasan.

1.2 Rumusan Masalah

Rumusan masalah yang ditinjau dari penelitian proyek akhir ini adalah bagaimana cara merancang sistem pemantauan energi listrik skala rumah tangga untuk mengetahui akurasi dan implementasi sistem berbasis *webhosting* serta dapat mengirimkan notifikasi peringatan dini mengenai informasi arus dan tegangan melalui Telegram Bot.

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian proyek akhir ini adalah mempermudah pengguna ketika melakukan pemantauan energi listrik dalam jarak jauh melalui *webhosting* dan notifikasi peringatan dini Telegram Bot dengan akurasi sistem yang akan dilakukan pengujian.

1.4 Batasan Masalah

Dalam penelitian proyek akhir ini, terdapat beberapa batasan masalah diantaranya :

1. Pengujian dilakukan dalam lingkup penggunaan listrik rumah tangga.
2. Sistem memberikan notifikasi peringatan kepada pengguna terkait variabel tegangan dan arus.
3. Sistem menggunakan mikrokontroler NodeMCU ESP8266 yang dilengkapi modul Wi-Fi agar sistem dapat dipantau jarak jauh melalui *webhosting* dan Telegram Bot.
4. Pengujian tidak meneliti keamanan sistem dan arsitektur *cloud*.
5. Pengujian akurasi hanya menggunakan dua variabel yaitu *tegangan* dan *arus*.
6. Sistem hanya memberikan notifikasi pada ID Telegram pengguna terdaftar.

1.5 Manfaat Penelitian

Berikut ini merupakan beberapa manfaat yang dapat diambil dari penelitian yang dilakukan oleh penulis :

Bagi Penulis

1. Menambah pengetahuan mengenai pemanfaatan IoT untuk kebutuhan sehari-hari, khususnya penggunaan energi listrik.
2. Menambah pengetahuan mengenai pentingnya pemantauan nilai tegangan, arus, dan energi listrik.
3. Mengembangkan keilmuan yang telah didapatkan selama masa perkuliahan.
4. Sebagai salah satu pesyaratan kelulusan dalam mendapatkan gelar Sarjana Terapan pada program studi Teknologi Rekayasa Internet.

Bagi Masyarakat

1. Penelitian dapat digunakan sebagai salah satu referensi penelitian mengenai implementasi pemantauan sistem pada penggunaan energi listrik.
2. Digunakan sebagai bahan kajian dari penelitian yang telah dilakukan sebelumnya.
3. Hasil dari penelitian dapat digunakan untuk melakukan penelitian selanjutnya, baik pengembangan maupun berbentuk perancangan.
4. Memberikan alternatif solusi untuk upaya optimalisasi penggunaan energi listrik skala rumah tangga.

BAB II TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Pemanfaatan IoT untuk sistem pemantauan energi listrik rumah tangga sudah pernah dilakukan sebelumnya dengan menggunakan modul yang beragam, sehingga dapat menjadi bahan pembelajaran dan tinjauan untuk dapat diadaptasi ke dalam sistem pemantauan energi listrik rumah tangga yang dibangun.

Penelitian terkait sistem pemantauan kWh meter yang pertama ditinjau berjudul “Rancang Bangun kWh Meter Digital sebagai Penghitung Biaya Pemakaian Energi Listrik Berbasis Arduino Uno R3” (Wahyu, M. I. L., 2018), menggunakan modul mikrokontroler berupa Arduino Uno R3 untuk pengendali utama. Rangkaian kWh meter digital memakai daya sebesar 2 Watt dengan catu daya tambahan yang bekerja ketika adanya pemadaman listrik. Hasil penelitian berupa alat yang dapat mengukur tegangan listrik, arus yang mengalir, daya yang digunakan, besar energi listrik yang terpakai, konversi ke rupiah, dan sistem *reset* otomatis dengan tingkat persentase ketelitian dan keberhasilan sebesar 95% dan galat 5%.

Penelitian kedua yang ditinjau berjudul “Sistem *Monitoring* Pemakaian Daya Listrik pada kWh Meter Menggunakan Arduino dan SMS *Gateway*” (Wirasasmita, R. H., 2022), menggunakan rangkaian modul berupa Arduino Uno R3 sebagai mikrokontroler, modul GSM SIM800L, sensor tegangan AC ZMPT101B, dan sensor arus ACS712-30A. Rangkaian sistem pemantauan pemakaian daya listrik pada kWh meter 1 fase ini dilengkapi dengan *dayabank* untuk kondisi listrik padam atau tegangan trip selama lebih dari 3 jam. Hasil penelitian berupa alat yang dapat mengukur pemakaian daya listrik dengan persentase kesalahan pada sensor tegangan ZMPT101B sebesar 0,0055% dan sekitar 0,00195% pada sensor arus ACS172-30.

Penelitian ketiga yang ditinjau berjudul “Prototipe Sistem *Monitoring* Konsumsi Energi Listrik serta Estimasi Biaya pada Peralatan Rumah Tangga Berbasis *Internet of Things*” (Adiwiranto, M. N., 2022), menggunakan rangkaian modul berupa NodeMCU ESP8266 sebagai mikrokontroler, sensor PZEM-004T, dan LCD 16x2 untuk menampilkan hasil pengukuran. Hasil penelitian berupa alat yang dapat mengukur nilai tegangan, arus, daya aktif, faktor daya, konsumsi energi listrik, dan estimasi biaya.

Sistem memiliki nilai akurasi terhadap alat ukur standar dengan persentase pada tegangan sebesar 98,94%, pada arus 99,18%, daya aktif sebesar 98,87%, faktor daya sebesar 98,44%, dan pada perhitungan total konsumsi energi listrik sebesar 97,98%.

Pada penelitian selanjutnya yang telah ditinjau berjudul “Alat *Monitoring* Pemakaian Energi Listrik Berbasis Android Menggunakan Modul PZEM-004T” (Habibi, F. N., 2017), alat pemantauan pemakaian energi listrik secara *real-time* menggunakan modul PZEM-004T, Raspberry Pi, dan Android dapat mengukur dan menampilkan nilai arus, tegangan, daya, serta akumulasi energi. Alat ukur tersebut memiliki besar simpangan tegangan sebesar 1,07%, arus 3,45%, daya nyata 3,93%, dan pada energi sebesar 18,6%. Pada beberapa sampel percobaan, pengukuran kurang akurat, sebab pendekatan angka meteran tidak terdapat desimal di belakangnya.

Penelitian berikutnya yang ditinjau adalah berjudul “*Smart* Sistem Untuk Pemantauan dan Pengisian Pulsa Listrik Prabayar *Via Smartphone*” (Giarniasih, H. N., 2019), telah dibuat rancang bangun sistem pemantauan dan pengisian kWh meter prabayar menggunakan NodeMCU sebagai mikrokontroler, sensor arus ACS172, *buzzer*, Firebase, dan MIT App Inventor. Hasil penelitian berupa sistem yang mampu melakukan pemantauan dan pengisian pulsa dari jarak jauh dengan selalu terkoneksi internet.

Pada penelitian yang telah ditinjau berjudul “Perancangan Sistem *Monitoring* Penggunaan Air PAM Berbasis IoT dengan Telegram Bot” (Maulidin, M. A. R., 2020), menggunakan Arduino Uno sebagai mikrokontroler, sensor *flow* meter, RTC, modul WiFi ESP8266, dan Telegram Bot. Adapun hasil penelitian adalah perhitungan debit air yang dipantau melalui Telegram Bot, sehingga masyarakat mampu melakukan estimasi biaya langganan air dalam jangka waktu tertentu. Dari kelima data yang diuji, ditemukan nilai *error* sebesar 12,48%, 7,80%, 6,65%, 5,30%, 4,47%. Hasil tersebut menunjukkan bahwa semakin besar volume air yang diamati, maka nilai *error* yang didapatkan juga akan semakin kecil.

Penelitian lain berjudul “Prototipe Sistem *Monitoring* dan Pengisian Token Listrik Prabayar Menggunakan Arduino Uno Berbasis *Website*” (Triyanto, D., 2020), telah dibuat sistem yang dapat mempermudah pengguna listrik prabayar dalam melakukan pengisian dan pemantauan token listrik dari jarak jauh dengan menggunakan modul

Arduino Uno, sensor tegangan ZMPT101B, sensor arus ACS172, modul *relay* 1 *channel*, modul GSM SIM 800L, MCB, LCD, dan Telegram API. Hasil dari penelitian adalah sistem yang terintegrasi dengan *website* untuk mengontrol token listrik pada kWh meter prabayar dengan jangka waktu setiap 5 menit. Adapun pengujian yang dilakukan sebanyak 15 kali dengan hasil yang sesuai antara sisa token yang tertera pada *website*, Telegram, dan prototipe.

Pada penelitian berjudul “Rancang Bangun Sistem *Monitoring* Peringatan Dini Bencana Banjir Berbasis IoT Menggunakan Protokol MQTT dengan Notifikasi Telegram Bot” (Bustomi, M. A. Y., 2021), menggunakan 2 komponen mikrokontroler NodeMCU dan Wemos sebagai pengontrol komponen IoT, dan informasi kepada masyarakat sekitar tentang peringatan dini bencana banjir melalui grafik secara *real-time*, GIS lokasi alat, dan peringatan dari Telegram Bot dan prototipe IoT yang telah dibuat. Adapun informasi peringatan akan dikirimkan oleh Telegram Bot secara otomatis setiap 1 jam dengan mengirimkan pesan “status”. Apabila terjadi kenaikan air yang menjadi tanda bahaya, *buzzer* pada alat juga akan berbunyi, sehingga mampu meminimalisir kerugian materi dan korban jiwa akibat bencana banjir.

Penelitian berjudul “Rancang Bangun Alat Pemantauan Biaya Pemakaian Energi Listrik menggunakan sensor PZEM-004T dan aplikasi Blynk” (Chairunnisa, I., 2022), menggunakan sensor PZEM-004T sebagai pembaca nilai arus dan tegangan, WeMos D1 Mini Pro sebagai mikrokontroler sehingga diketahui nilai energi dan biaya yang digunakan dan akan dipantau melalui *software* Blynk. Pengujian dilakukan dengan menyalakan alat selama 2 jam dan diperoleh hasil bahwa nilai *error* yang dimiliki oleh alat adalah memiliki persentase *error* sebesar 0,1% pada nilai arus dan 0,004% pada nilai tegangan.

Penelitian terakhir yang ditinjau berjudul “Pemrograman Sistem *Monitoring* Panel Tegangan Rendah Berbasis IoT” (Aziz, N. M., 2021), sistem menggunakan komponen berupa NodeMCU ESP8266, sensor PZEM-004T, Google Sheets, dan Telegram. Adapun hasil penelitian berupa sistem pemantauan arus, tegangan, daya, dan frekuensi pada sisi tegangan rendah dan telah dilakukan pengujian akses dari berbagai tempat dengan menggunakan bantuan internet. Pada pengujian menggunakan Telegram, pengiriman waktu pesan yang didapatkan adalah kurang dari 20 detik tergantung kecepatan internet dan sistem *reset* data.

Berikut ini merupakan tabel ringkasan tinjauan pustaka yang digunakan oleh penulis :

Tabel 2. 1 Ringkasan Tinjauan Pustaka

No.	Judul Penelitian	Pokok Bahasan	Tipe Penelitian	Tools/Library	Penulis	Tahun
1.	Rancang Bangun kWh Meter Digital Sebagai Penghitung Biaya Pemakaian Energi Listrik Berbasis Arduino Uno R3	kWh meter digital sebagai penghitung biaya pemakaian energi listrik melalui pengukuran tegangan dan arus	Rancang Bangun	Sensor tegangan ZMPT101B, sensor arus ACS172, Arduino Uno R3, RTC, <i>seven segment</i> , <i>keypad</i>	M. Ilham Ludya Wahyu, dkk	2018
2.	Sistem <i>Monitoring</i> Pemakaian Daya Listrik pada kWh Meter Menggunakan Arduino dan SMS Gateway	Pemantauan pemakaian daya listrik pada kWh meter 1 fase dan SMS Gateway berisi total biaya yang harus dibayarkan oleh pengguna dalam 1 hari	Implementasi Sistem	Arduino Uno, LCD 16x2, SMS gateway, sensor ACS712-304, sensor AC ZMPT101B, modul GSM SIM800L, kWh meter digital	Rasyid Hardi W., dkk	2022
3.	Prototipe Sistem <i>Monitoring</i> Konsumsi Energi Listrik serta Estimasi Biaya pada Peralatan Rumah Tangga Berbasis <i>Internet of Things</i>	Pemantauan nilai akurasi tegangan, arus, faktor daya, dan konsumsi energi terhadap beban pada peralatan rumah tangga berbasis IoT	Rancang Bangun	Sensor PZEM-004T, NodeMCU ESP8266, Ubidots Clouds, laptop/smartphone, LCD 16x2	Mohamad Nursamsi A., dkk	2022
4.	Alat <i>Monitoring</i> Pemakaian Energi Listrik Berbasis Android Menggunakan Modul PZEM-004T	Pemantauan energi listrik berbasis Android melalui pengujian tegangan, arus, daya nyata, dan energi	Implementasi Sistem	Smartphone Energy Meter, sensor PZEM-004T, <i>smartphone</i> Android, Python, Raspberry Pi	Fatoni Nur Habibi, dkk	2017

No.	Judul Penelitian	Pokok Bahasan	Tipe Penelitian	Tools/Library	Penulis	Tahun
5.	<i>Smart</i> Sistem Untuk Pemantauan dan Pengisian Pulsa Listrik Prabayar <i>Via Smartphone</i>	Pemantauan konsumsi energi listrik melalui pengujian arus, daya, dan Firebase	Implementasi Sistem	Sensor ACS712, <i>buzzer</i> , Firebase, MIT App Inventor	Hasna Naufal G., dkk	2019
6.	Perancangan Sistem <i>Monitoring</i> Penggunaan Air PAM Berbasis IoT dengan Telegram Bot	Pemantauan nilai debit air dari pulsa frekuensi rotor pada sensor arus air berbasis Iot dengan Telegram Bot	Rancang Bangun	Arduino Uno, sensor <i>flow</i> meter, modul WiFi ESP8266, Telegram Bot	Muhammad Abdur Rofi M., dkk	2020
7.	Prototipe Sistem <i>Monitoring</i> dan Pengisian Token Listrik Prabayar Menggunakan Arduino Uno Berbasis <i>Website</i>	Pemantauan penggunaan arus dan tegangan serta pengisian token listrik prabayar menggunakan Arduino Uno yang terintegrasi <i>website</i>	Rancang Bangun	Arduino Uno, LCD, LCD <i>backpack</i> , sensor arus ACS712, sensor tegangan ZMPT101B, modul GSM SIM 800L, modul <i>relay</i> 1 <i>channel</i> , MCB 2A	Dedi Triyanto, dkk	2020
8.	Rancang Bangun Sistem <i>Monitoring</i> Peringatan Dini Bencana Banjir Berbasis IoT Menggunakan Protokol MQTT dengan Notifikasi Telegram Bot	Pemantauan peringatan dini bencana banjir melalui uji coba sensor ketinggian air berbasis IoT menggunakan notifikasi dari Telegram Bot	Rancang Bangun	Sensor ketinggian air, protokol MQTT, Telegram Bot, <i>website</i>	M. Abi Yazid Al Bustomi, dkk	2021

No.	Judul Penelitian	Pokok Bahasan	Tipe Penelitian	Tools/Library	Penulis	Tahun
9.	Rancang Bangun Alat Pemantau Biaya Pemakaian Energi Listrik Menggunakan Sensor PZEM-004T dan Aplikasi Blynk	Pemantauan biaya pemakaian energi listrik rumah tangga melalui Blynk	Rancang Bangun	Blynk, sensor PZEM004T, <i>smartphone</i> , Wemos D1 Mini Pro	Indah Chairunnisa, dkk	2022
10.	Pemrograman Sistem <i>Monitoring</i> Panel Tegangan Rendah Berbasis IoT	Pemantauan nilai tegangan dan arus pada 3 fase yang dapat diakses tanpa keterbatasan jarak dan waktu	Implementasi Sistem	Gardu distribusi, modul NodeMCU ESP8266, sensor PZEM-004T, Google Sheets, Telegram	Nofvan Maghresa Aziz, dkk	2021

Berdasarkan tabel ringkasan tinjauan pustaka, penelitian yang akan dilakukan oleh penulis tidak hanya memiliki persamaan, namun juga terdapat beberapa perbedaan seperti penggunaan *webhosting* yang dapat memudahkan pengguna dalam mengoperasikan sistem. Pengguna tidak perlu melakukan proses login dan dapat diakses darimana saja, meskipun berada dalam jarak yang jauh. Sistem juga dapat mengirimkan notifikasi peringatan dini terkait kondisi arus dan tegangan pada sistem apabila ditemukan potensi bahaya, sehingga pengguna mampu melakukan pemantauan dan menentukan tindakan lanjut dari kondisi yang dikirimkan. Selain itu, sistem juga tersimpan di beberapa lokasi seperti *webhosting*, *database*, dan juga Telegram Bot, sehingga apabila ada kehilangan data, data dapat dicadangkan, dan juga dapat digunakan untuk fungsi pelaporan.

2.2 Dasar Teori

Pada bagian ini, penulis akan menguraikan teori-teori yang berkaitan dengan penelitian sebagai acuan atau dasar teori dalam melakukan penelitian.

2.2.1 *Internet of Things*

Menurut rekomendasi ITU-T Y.2060, *Internet of Things* (IoT) menjadi salah satu penemuan untuk menyelesaikan permasalahan melalui kolaborasi dari teknologi dengan dampak sosial. IoT juga berperan sebagai infrastruktur yang bersifat universal untuk memenuhi layanan informasi yang dibutuhkan oleh masyarakat dan mampu mengkoneksikan layanan secara fisik maupun virtual berdasarkan perkembangan informasi dan teknologi komunikasi (ICT).

Pengertian IoT ditinjau dari *e-book* berjudul “*Making Sense of IoT*” oleh Kevin Ashton, bahwa IoT merupakan sebuah atau kumpulan sensor yang terhubung ke internet dan memungkinkan adanya pertukaran data dan informasi antar perangkat komputer serta menjadi bagian dari kehidupan manusia (Yudho&Abdul, 2019).

2.2.2 **NodeMCU ESP 8266**

NodeMCU ESP8266 merupakan suatu *platform* yang bersifat *open source* untuk IoT yang menggunakan pemrograman LUA atau *sketch* Arduino IDE dalam membantu *programmer* dalam menyelesaikan prototipe atau proyek yang berkaitan dengan IoT. Adapun pengembangan pada modul ESP8266 yang mengintegrasikan beberapa pin seperti GPIO, PWM, I2C, L-wire, dan ADC dalam satu *board* yang memiliki ukuran dimensi panjang 4.83 cm dan lebar 2.54 cm. Meskipun tergolong kecil, *board* NodeMCU telah dilengkapi dengan adanya fitur WiFi dan *firmware* yang sifatnya *open source* (Suhartono, 2021).

Berikut ini merupakan spesifikasi NodeMCU menurut *datasheet* yang telah dibuat oleh Handson Technology :

1. Tegangan : 3.3V
2. WiFi *direct* (P2P), soft-AP
3. Konsumsi arus : 10uA~170mA
4. *Flash memory* : maksimal 16MB (normal 512K)
5. Terintegrasi protokol TCP/IP
6. Prosesor : Tensilica L106 32-bit
7. Kecepatan prosesor : 80~160 MHz
8. RAM : 32K + 80K
9. GPIO : 17 (multiplex)
10. *Analog to digital* : 1 input dengan 1024 resolusi step
11. Daya output : +19.5dBm pada mode 802.11b
12. 802.11 *support* : b/g/n
13. Conarus maksimal pada koneksi TCP : 5



Gambar 2. 1 NodeMCU ESP8266

2.2.3 Sensor PZEM-004T

Sensor PZEM-004T merupakan sensor multifungsi yang digunakan untuk mengukur tegangan AC, arus, daya aktif, frekuensi, faktor daya, dan mampu untuk membaca data melalui *port* TTL. Pada sensor PZEM-004T, batas pengukuran arus adalah 10-100A, sedangkan pengukuran tegangan 80-380V (Walid, 2022).



Gambar 2. 2 Sensor PZEM-004T

Sensor PZEM-004T memiliki tiga komponen dengan perannya masing-masing, seperti optocoupler yang berfungsi sebagai pengisolasi sinyal DC dan AC untuk memastikan bahwa sirkuit pengukuran dalam keadaan aman dan stabil. Komponen kedua yang bernama current transformator yang berfungsi sebagai pendeteksi dan pengukur arus yang mengalir. Komponen ketiga berupa shunt resistor yang berfungsi sebagai pengukur tegangan yang sedang melintas pada sirkuit. Dari ketiga komponen tersebut, menentukan cara kerja sensor PZEM-004T yaitu berdasarkan variasi tegangan listrik yang sedang berjalan di sistem.

Berikut *datasheet* dari sensor PZEM-004T :

1.1 Voltage
1.1.1 Measuring range: 0V~200V
1.1.2 Resolution: 0.1V
1.1.3 Measurement accuracy: 0.5%
1.2 Current
1.2.1 Measuring range: 0~50A(PZEM-004T-00A), 0~100A(PZEM-004T-100A)
1.2.2 Starting measure current: 0.01A(PZEM-004T-00A), 0.02A(PZEM-004T-100A)
1.2.3 Resolution: 0.001A
1.2.4 Measurement accuracy: 0.5%
1.3 Active power
1.3.1 Measuring range: 0~2.5kW(PZEM-004T-10A), 0~11.5kW(PZEM-004T-100A)
1.3.2 Starting measure power: 0.001W
1.3.3 Resolution: 0.1W
1.3.4 Display format:
< 1000W, the display unit is mW, such as: 999.9W
> 1000W, the display unit is kW, such as: 0.001W
1.3.5 Measurement accuracy: 0.2%
1.4 Power factor
1.4.1 Measuring range: 0.00~1.00
1.4.2 Resolution: 0.01
1.4.3 Measurement accuracy: 1%
1.5 Frequency
1.5.1 Measuring range: 45Hz~65Hz
1.5.2 Resolution: 0.1Hz
1.5.3 Measurement accuracy: 0.2%
1.6 Active energy
1.6.1 Measuring range: 0~9999.99kWh
1.6.2 Resolution: 1Wh
1.6.3 Measurement accuracy: 0.2%
1.6.4 Display format:
< 10kWh, the display unit is kWh(1/1000Wh), such as: 9999Wh
> 10kWh, the display unit is kWh, such as: 9999.99kWh
1.6.5 Reset strategy: via software to reset.

Gambar 2. 3 Datasheet Sensor PZEM-004T

2.2.4 Modul LCD 16x2

Liquid Crystal Display (LCD) 16x2 merupakan modul IoT yang berfungsi untuk menampilkan data dalam bentuk karakter. Data yang ditampilkan adalah berupa data ASCII (*American Standard Code International Interchange*) yang dapat mengkonversi nilai alfabet dan numerik menjadi data digital. Kode ASCII terdiri dari 10 bit, yaitu 1 bit *start*, 1 bit *stop*, dan 8 bit data (Wiratman, 2008).



Gambar 2. 4 Modul LCD

2.2.5 Modul I2C

Modul *Inter Integrated Circuit* (I2C) mendukung jenis komunikasi *multiple bus master* menggunakan protokol *plug and play* yang terdiri dari dua sinyal yaitu SDA (*Serial Data*) dan SCL (*Serial Clock*), dimana kedua sinyal tersebut bersifat *bi-directional*. SCL berfungsi untuk *clock and wait*, sedangkan SDA digunakan untuk proses mengirim data dan alamat. Seluruh perangkat I2C memiliki alamat yang spesifik. Tujuannya untuk membedakan antar perangkat dalam bus I2C yang sama (Valdez&Becker, 2015).



Gambar 2. 5 Modul I2C

2.2.6 Webhosting

Webhosting merupakan *platform* analisis yang mampu memudahkan pengguna dalam melakukan visualisasi data pada sebuah *dashboard*. Dalam implementasinya, terdapat data *source* yang dapat ditambahkan untuk mendukung dalam menampilkan data yang diinginkan oleh pengguna (Shilpi, 2017).

2.2.7 MySQL

MySQL merupakan *database server* relasional atau RDBMS (*Relational Database Management System*) yang bersifat *open source* untuk menangani data bervolume besar. *Software* ini memiliki beberapa fasilitas yang memudahkan praktisi untuk membangun suatu proyek. Salah satunya adalah dengan memanfaatkan API (*Application Programming Interface*) yang memungkinkan banyak aplikasi lain maupun bahasa pemrograman untuk saling terintegrasi (Wahana Komputer, 2010).

2.2.8 Telegram Bot

Telegram merupakan aplikasi *chatting* yang menggunakan nomor seluler pribadi sebagai identitas dan verifikasi akun serta memungkinkan pengguna untuk mengirimkan pesan dengan enkripsi *end-to-end*. Dalam Telegram terdapat fitur bernama Bot yang dapat menjalankan pesan secara instan tanpa adanya nomor tambahan sebagai syarat mengirimkan pesan (Alwizain, 2023).

2.3 Hipotesis

Berdasarkan uraian yang mengacu pada tinjauan pustaka dan dasar teori, maka telah diperoleh beberapa tahapan yang dapat diadaptasi dari penelitian sebelumnya untuk merancang sistem pemantauan penggunaan energi listrik rumah tangga berbasis *webhosting* dan dapat mengirimkan notifikasi melalui Telegram Bot serta tahapan-tahapan untuk pengujiannya.

BAB III METODOLOGI PENELITIAN

3.1 Alat Penelitian

Dalam melaksanakan penelitian, adapun peralatan yang dibutuhkan untuk menunjang penelitian berupa perangkat keras. Peralatan tersebut akan digunakan untuk melakukan perakitan komponen agar menjadi sistem dengan kesatuan yang utuh. Adapun hasil dari perakitan komponen yang telah dibuat didapatkan dari perancangan sistem yang telah dilakukan. Beberapa komponen perangkat keras tersebut adalah sebagai berikut :

1. Satu buah NodeMCU ESP8266
2. Satu buah sensor PZEM-004T V3.0
3. Satu buah modul I2C
4. Satu buah LCD 16x2
5. Satu buah laptop
6. Satu buah steker dan terminal
7. Satu buah kabel USB
8. Beberapa beban berupa peralatan rumah tangga

3.2 Bahan Penelitian

Dalam melakukan penelitian, adapun bahan penelitian berupa perangkat lunak yang akan berperan dalam proses pengiriman data ketika sistem telah terkoneksi jaringan internet. Adapun perangkat lunak yang digunakan diantaranya sebagai berikut :

1. Arduino IDE
2. MySQL *Database*
3. *Webhosting* 000webhostapp
4. Telegram Bot

3.3 Tahapan Penelitian

3.3.1 Tahapan Umum



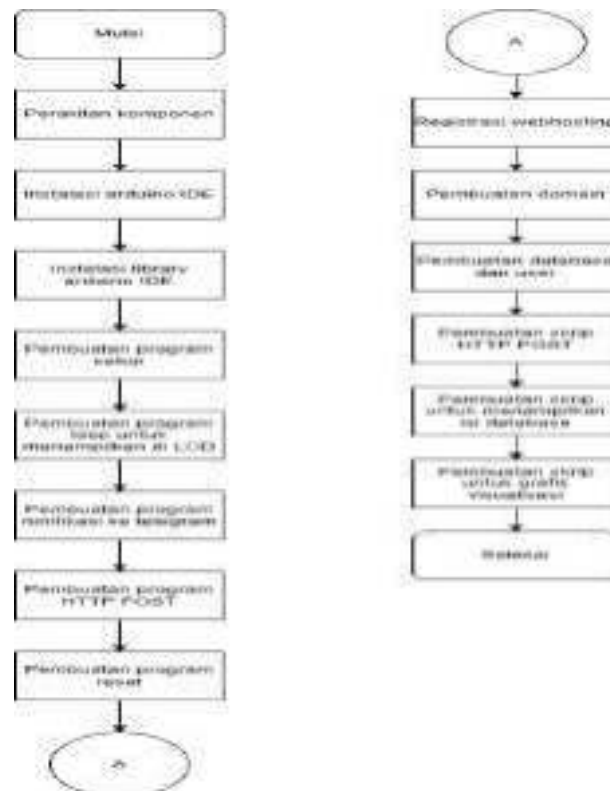
Gambar 3. 1 Tahapan Umum Penelitian

Tahapan penelitian proyek akhir secara umum terdiri dari, identifikasi masalah, studi literatur, analisis kebutuhan sistem, perancangan arsitektur dan sistem, implementasi, pengujian dan pengambilan data, analisis dan pembahasan hasil, serta dokumentasi dan pelaporan. Adapun pada tahapan identifikasi masalah, diperlukan pengkajian dan penjabaran, serta usulan solusi dari masalah yang ditemukan. Tahap studi literatur dilakukan dengan mencari referensi dari jurnal, artikel, buku, dan internet sebagai landasan teori dan kerangka berpikir dalam sebuah penelitian. Tahapan analisis kebutuhan sistem terdiri dari kebutuhan perangkat keras yang akan dirangkai dan kebutuhan perangkat lunak. Selanjutnya pada tahapan perancangan arsitektur dan sistem digunakan *software* Fritzing untuk desain.

Pada tahapan implementasi akan merujuk pada tahapan secara teknis yang dimulai dari perakitan komponen hingga visualisasi hasil data. Tahapan selanjutnya adalah pengambilan data yang digunakan untuk bahan pengujian dan analisis perbandingan hasil. Tahapan terakhir yang dilakukan adalah proses dokumentasi pada seluruh komponen dan program untuk dikumpulkan sebagai pelaporan akhir.

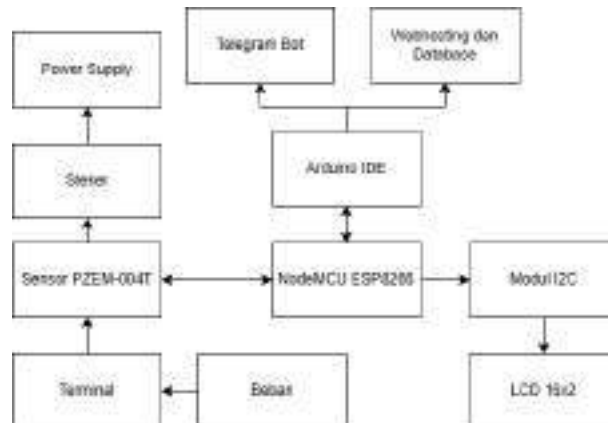
3.3.2 Tahapan Teknis

Tahapan teknis pada penelitian proyek akhir terdiri dari perakitan komponen, instalasi arduino IDE, instalasi *library* pada arduino IDE, pembuatan program *setup*, pembuatan program *loop* untuk menampilkan hasil data di LCD, notifikasi ke Telegram, fungsi HTTP POST, pembuatan program *reset*, registrasi *webhosting*, pembuatan *website* dan *domain*, pembuatan *database* dan *user*, pembuatan skrip HTTP POST menggunakan PHP dan HTML, pembuatan skrip untuk menampilkan isi dari *database* yang telah dibuat, dan tahapan terakhir adalah pembuatan skrip untuk grafik visualisasi. Berikut merupakan diagram alir dari tahapan yang sudah dirancang :



Gambar 3. 2 Tahapan Teknis Penelitian

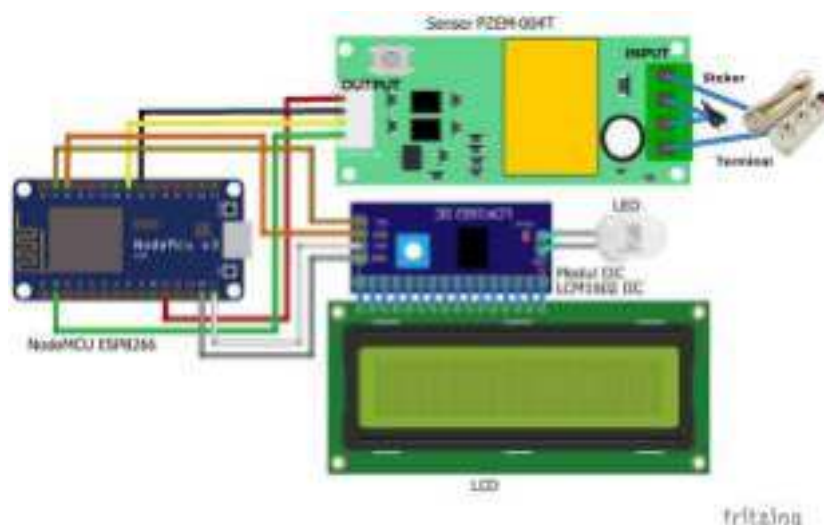
3.4 Diagram Blok dan Perancangan Sistem



Gambar 3. 3 Diagram Blok Sistem

Diagram blok sistem berisi tentang penjelasan sistem secara komprehensif meliputi fungsi dan hubungan masukan dan keluaran. Dalam penelitian ini, pusat pengendali sistem berada pada komponen NodeMCU ESP8266 sebagai mikrokontroler. Komponen ini akan terhubung dengan modul I2C sebagai keluaran dan sensor PZEM-004T serta Arduino IDE sebagai masukan data. Sensor PZEM-004T memiliki masukan data ke steker dan terminal untuk dapat diuji menggunakan beban serta keluaran ke modul NodeMCU ESP8266. Modul I2C akan terhubung ke LCD 16x2 untuk menampilkan data. Arduino IDE memiliki masukan data dari NodeMCU ESP8266 dan keluaran ke *webhosting* dan Telegram Bot.

Perancangan sistem berisi tentang desain perangkat keras yang digunakan dalam penelitian proyek akhir. Berikut ini merupakan gambar rangkaian sistem yang dilakukan dalam penelitian proyek akhir :



Gambar 3. 4 Perancangan Sistem Menggunakan Fritzing

Perancangan sistem dilakukan dengan menggunakan *software* Fritzing yang terdiri dari empat komponen yaitu, modul NodeMCU ESP82266, LCD 16x2, I2C, dan sensor PZEM-004T. Pada sensor PZEM-004T, bagian input akan terhubung ke steker dan terminal, sedangkan pada bagian output akan terhubung ke NodeMCU ESP8266 yang mendapatkan suplai tegangan dari laptop pengguna. Berikut merupakan *datasheet* pada masing-masing modul :

Tabel 3. 1 Datasheet Port PZEM-004T V3.0 ke NodeMCU ESP8266

Pin PZEM-004T V3.0	Pin ESP8266	Fungsi
GND	GND	Input
TX	D5	Input
RX	D6	Input
5V	3V3	Input

Rangkaian sistem terdiri dari beberapa komponen seperti, NodeMCU ESP8266, sensor PZEM-004T V3.0, dan LCD I2C 16x2. Tabel 3.1 merupakan *datasheet port* yang menghubungkan NodeMCU ESP8266 dengan sensor PZEM-004T V3.0. Pin D5 pada ESP8266 akan terhubung pada pin TX pada sensor PZEM-004T V3.0 sebagai pengirim. Sedangkan pin D6 pada ESP8266 terhubung dengan pin RX pada sensor PZEM-004T V3.0 sebagai penerima.

Tabel 3. 2 Datasheet Port I2C ke NodeMCU ESP8266

Pin I2C	Pin ESP8266	Fungsi
GND	GND	Output
VCC	VIN	Output
SDA	D2	Output
SCL	D1	Output

Pada tabel 3.2 merupakan *datasheet port* yang menghubungkan NodeMCU ESP8266 dengan I2C pada LCD 16 x 2. Pin D2 pada ESP8266 terhubung dengan pin SDA pada I2C, sedangkan pin D1 pada ESP8266 akan terhubung dengan pin SCL pada I2C. Untuk melakukan fungsi *reset*, digunakan pin RST pada ESP8266.

3.5 Instalasi dan Konfigurasi Program pada Arduino IDE

3.5.1 Instalasi Arduino IDE

Arduino IDE merupakan perangkat lunak yang berfungsi untuk menulis, menyunting, dan mengunggah program ke papan Arduino. Untuk melakukan instalasi perangkat lunak Arduino IDE dapat diunduh melalui situs *website* resmi <https://www.arduino.cc/en/main/software> dan memilih

sesuai spesifikasi pada perangkat pengguna. Pada penelitian ini digunakan Arduino IDE dengan versi 2.0.4.



Gambar 3. 5 Hasil Unduhan Arduino IDE

Kemudian memilih tombol *I Agree* pada jendela *License Agreement* untuk melanjutkan pada langkah instalasi selanjutnya.



Gambar 3. 6 Halaman License Agreement Arduino IDE

Selanjutnya memilih opsi instalasi dan pada penelitian ini memilih *Only for me (Admin)*.



Gambar 3. 7 Opsi Instalasi Pengguna Arduino IDE

Langkah selanjutnya memilih lokasi folder instalasi dan menekan tombol *Install*.



Gambar 3. 8 Lokasi Instalasi Arduino IDE

Setelah proses instalasi berhasil, selanjutnya menuju ke halaman awal pada papan Arduino IDE.



Gambar 3. 9 Halaman Awal Arduino IDE

Pada menu *Preferences*, pengguna harus menambahkan *link board manager* untuk mengakses *board* NodeMCU ESP8266 yaitu http://arduino.esp8266.com/stable/package_esp8266com_index.json.



Gambar 3. 10 Menu Preferences Arduino IDE

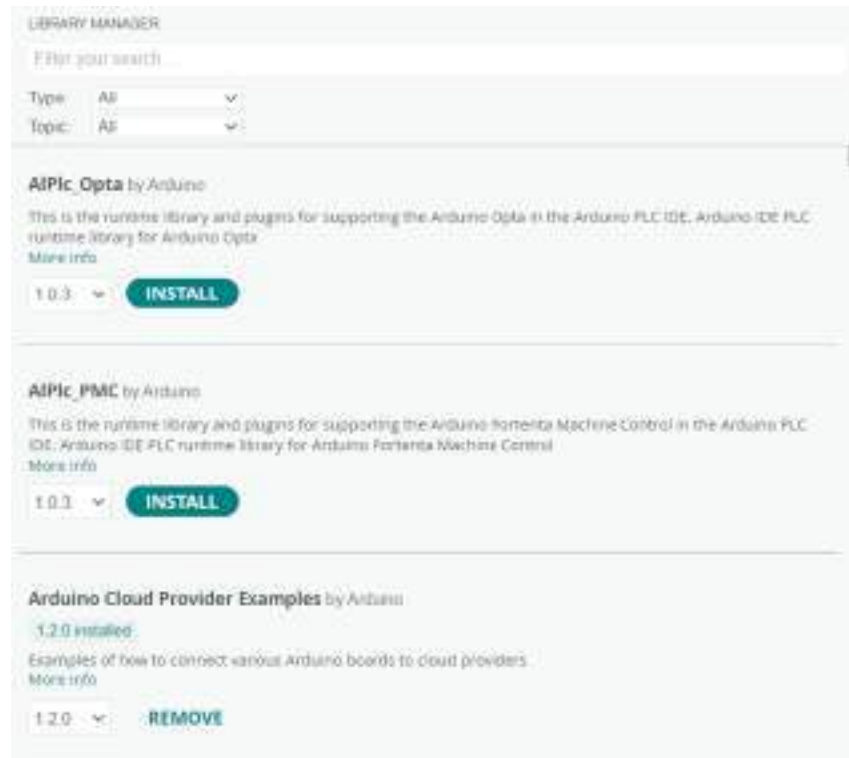
3.5.2 Instalasi *Library*

Untuk dapat menjalankan program pada Arduino IDE, maka digunakan berbagai macam *library* agar komponen eksternal dan internal yang terhubung ke Arduino dapat berfungsi dengan baik. Untuk melakukan instalasi pada aplikasi Arduino IDE, memilih menu *Tools* kemudian *Manage Libraries*.



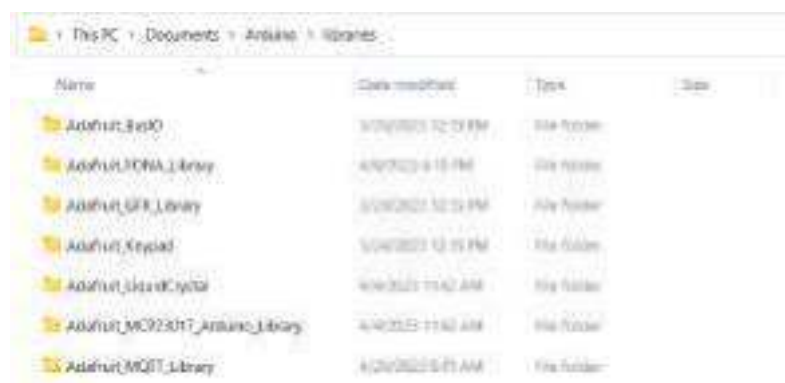
Gambar 3. 11 Menu Manage Libraries

Berikut ini merupakan daftar *library* dalam pencarian, apabila *library* yang akan digunakan telah ditemukan, maka memilih tombol *install* untuk melakukan instalasi *library*.



Gambar 3. 12 Kolom Pencarian Library

Terdapat cara lain untuk menambahkan *library* yaitu dengan mengunduh folder berisi *library* yang diperlukan, kemudian memindahkan ke dalam folder *libraries* di dalam folder Arduino.



Gambar 3. 13 Folder Libraries Arduino

Berikut ini merupakan daftar *library* yang digunakan dalam pemrograman sistem :

```
1  #include <ESP8266WiFi.h>
2  #include <ESP8266HTTPClient.h>
3  #include <WiFiClientSecure.h>
4  #include <UniversalTelegramBot.h>
5  #include <ESP8266WebServer.h>
6  #include <LiquidCrystal_I2C.h>
7  #include <PZEM004Tv30.h>
8  #include <SoftwareSerial.h>
9  #include <Wire.h>
```

Gambar 3. 14 Library yang Digunakan

- Library ESP8266WiFi.h digunakan untuk menghubungkan, mengatur, dan mengontrol koneksi jaringan WiFi pada modul ESP8266.
- Library ESP8266HTTPClient.h berfungsi sebagai penyedia koneksi HTTP pada modul ESP8266 yang memungkinkan pengguna untuk mengirim permintaan HTTP ke server atau menerima respons dari server.
- Library WiFiClientSecure.h merupakan perluasan dari library WiFiClient.h yang memiliki dukungan keamanan tambahan untuk membuat koneksi melalui protokol HTTPS pada modul ESP8266.
- Library UniversalTelegramBot.h memungkinkan pengguna untuk menghubungkan mikrokontroler ESP8266 dengan *platform* Telegram melalui API Telegram Bot, sehingga pengguna dapat membuat Telegram Bot untuk menerima dan mengirim pesan atau notifikasi.
- Library ESP8266WebServer.h menyediakan fungsi-fungsi yang memungkinkan pengguna untuk membuat server web pada modul ESP8266, membuat rute HTTP, mengelola permintaan HTTP, dan memberikan respons kepada pengguna yang terhubung.
- Library LiquidCrystal_I2C.h menyediakan fungsi-fungsi yang digunakan untuk mengontrol dan mengendalikan tampilan pada layar LCD dengan menggunakan koneksi I2C pada *platform* Arduino IDE.
- Library PZEM004Tv30.h digunakan untuk mengakses dan membaca data dari modul PZEM-004T V3.0 pada platform Arduino IDE yang

memiliki fungsi untuk mengukur parameter tegangan, arus, daya, energi, frekuensi, dan faktor daya.

- Library SoftwareSerial.h berfungsi untuk mengkoneksikan *port* serial tambahan menggunakan pin digital pada mikrokontroler ESP8266.
- Library Wire.h digunakan oleh protokol I2C untuk mengontrol perangkat I2C seperti sensor, modul, dan mikrokontroler.

3.5.3 Pembuatan Program Setup

```
void setup() {  
  Serial.begin(115200);  
  //pinSerial.begin(9600);  
  wifiClient.setInsecure();  
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);  
  while (WiFi.status() != WL_CONNECTED) {  
    delay(1000);  
  }  
  lcd.begin();  
  lcd.backlight();  
  lcd.setCursor(0, 0);  
  lcd.print("initializing...");  
  delay(2000); lcd.clear();  
  String message = "Ish Monitoring Bot is started!";  
  sendMessageToAll(message);  
  delay(2000); lcd.clear();  
  float energy1 = pmm.resetEnergy();  
}
```

Gambar 3. 15 Program Setup

Fungsi setup() pada Arduino IDE digunakan untuk melakukan inisialisasi awal sebelum program loop dijalankan. Fungsi setup() dieksekusi sebanyak satu kali ketika program dinyalakan atau dikembalikan ke pengaturan awal. Di dalam fungsi setup(), pengguna dapat menambahkan program untuk mengkonfigurasi perangkat keras yang sudah terangkai, seperti berikut :

- Serial.begin(baudRate) untuk memulai komunikasi serial pada kecepatan aliran data yang telah ditentukan.
- WiFi.begin(WIFI_SSID, WIFI_PASSWORD) digunakan untuk memulai koneksi WiFi dengan nama jaringan dan kata sandi yang telah ditentukan.
- Lcd.begin() berfungsi sebagai fungsi inisialisasi modul LCD dengan jumlah baris dan kolom yang telah didefinisikan.

- Pzem.resetenergy() digunakan untuk mengatur ulang pengukuran energi menjadi bernilai 0 pada modul PZEM-004T V3.0.

Selain itu, pada fungsi setup() juga terdapat fungsi yang memanggil parameter bernama message untuk dikirimkan sebagai notifikasi pada Telegram dan fungsi pada LCD untuk menampilkan inisialisasi awal sebelum program utama dijalankan.

3.5.4 Pembuatan Program *Loop*

a. Fungsi Menampilkan Data di Layar LCD

```
void loop() {
  float energy = pzem.energy();
  float harga = energy * 1352.00;
  float voltage = pzem.voltage();
  float current = pzem.current();
  float power = pzem.power();
}
```

Gambar 3. 16 Fungsi Loop

Fungsi loop() dalam Arduino IDE merupakan fungsi utama yang akan dieksekusi secara berulang hingga Arduino dihentikan atau direset. Fungsi ini berjalan setelah fungsi setup() berhasil dieksekusi. Pada gambar 3.15 mendefinisikan fungsi variabel terukur dari sensor PZEM-004T V3.0 yaitu tegangan, arus, daya, energi, dan harga untuk mengambil nilai pada perangkat yang terhubung. Dalam variabel tersebut, dideklarasikan sebagai tipe data *float* yang hasil perhitungannya akan ditampilkan pada layar LCD, *serial monitor*, dan pengiriman notifikasi pada Telegram Bot.

```
if (energy >= 0.000) {
  lcd.setCursor(0, 0);
  lcd.print("Energy : ");
  lcd.print(energy);
  lcd.print(" kWh");
  lcd.setCursor(0, 1);
  lcd.print("harga : ");
  lcd.print("Rp ");
  lcd.print(harga);
}
```

Gambar 3. 17 Fungsi If

Gambar 3.16 merupakan fungsi *if* yang akan menampilkan perhitungan energi dan harga pada layar LCD dari sensor PZEM-004T V3.0.


```

else if (energy != NAN) {
    lcd.setCursor(0, 0);
    lcd.print("Energy : ");
    lcd.print(energy);
    lcd.print(" kWh");
    lcd.setCursor(0, 1);
    lcd.print("Harga : ");
    lcd.print("Rp ");
    lcd.print(harga);
}

```

Gambar 3. 18 Fungsi Else If

Pada gambar 3.18 mendefinisikan fungsi *else if* yang akan menampilkan nilai *NAN* pada layar LCD karena berbagai macam kondisi seperti rangkaian yang salah.

```

else {
    lcd.setCursor(0, 0);
    lcd.print("Error reading energy");
}

```

Gambar 3. 19 Fungsi Else

Gambar 3.19 menunjukkan fungsi *else* yang akan menampilkan nilai *error* pada layar LCD karena komponen yang rusak atau tidak dapat membaca hasil perhitungan dari sensor.

b. Fungsi Tampilan Notifikasi ke Telegram

```

void sendMessageToAll(string message) {
    for (int i = 0; i < static_cast<int>(sizeof(chatIds) / sizeof(chatIds[0])); i++) {
        bot.sendMessage(String(chatIds[i]), message);
    }
}

```

Gambar 3. 20 Fungsi SendMessageToAll

Fungsi untuk mengirimkan notifikasi ke Telegram terbagi menjadi beberapa bagian dengan penjelasan pada gambar 3.20 berisi fungsi untuk menghitung ukuran *array* *chatIds* dalam *byte* dan mengkonversi menjadi tipe data *int*. Sedangkan fungsi *sizeof()* digunakan untuk menghitung memori yang terpakai oleh suatu tipe data atau objek tertentu.

```
String message = "Voltage: " + String(voltage) + " V\n" +
  "Current: " + String(current) + " A\n" +
  "Power: " + String(power) + " W\n" +
  "Energy: " + String(energy) + " kWh\n" +
  "Harga: " + "Rp " + String(harga);
sendMessageToAll(message);
delay(60000); lcd.clear();

Serial.println(message);
```

Gambar 3. 21 String Message ke Telegram

Gambar 3.21 mendeklarasikan dan menginisialisasi variabel *message* dengan nilai tertentu dalam bentuk tipe data string. Fungsi ini digunakan untuk format pengiriman notifikasi data ke Telegram Bot dari hasil perhitungan sensor dengan jeda waktu setiap 1 menit sekali.

```
if (voltage <= 198.00) {
  String message = "Voltage is Undervoltage";
  sendMessageToAll(message);
}
if (voltage >= 242.00) {
  String message = "Voltage is Overvoltage";
  sendMessageToAll(message);
}
if (current >= 4.00) {
  String message = "Current is Inrush";
  sendMessageToAll(message);
}
```

Gambar 3. 22 Notifikasi Peringatan Kondisi

Gambar 3.22 mendeklarasikan fungsi peringatan notifikasi yang akan dikirim ke Telegram Bot sesuai kondisi tertentu, seperti pengaturan batas nilai tegangan dan arus yang aman, sehingga pengguna dapat melakukan pemantauan meskipun berada pada jarak yang jauh.

c. Fungsi HTTP POST

```
if (WiFi.status() == WL_CONNECTED) {
  HTTPClient https;
```

Gambar 3. 23 Koneksi ke Protokol HTTPS

Pada program pembuatan fungsi HTTP POST, terdapat beberapa bagian seperti gambar 3.23 yang berisi fungsi *if* yang digunakan untuk memeriksa status koneksi WiFi. Apabila status koneksi WiFi WL_CONNECTED, maka kode akan dieksekusi karena koneksi telah terhubung.

3.5.5 Pembuatan Fungsi *Reset*

```
if (digitalRead(RESET_PIN) == LOW) {  
    delay(2000); lcd.clear();  
}
```

Gambar 3. 27 Fungsi Reset

Pada pembuatan fungsi *reset*, `digitalRead(pin)` digunakan untuk membaca status logika dari pin digital tertentu yang dalam hal ini adalah `RESET_PIN`. Fungsi ini akan mengembalikan nilai `LOW` (logika rendah) apabila tegangan pada pin tersebut rendah atau mendekati nilai 0. Fungsi tersebut akan berjalan jika pengguna menekan tombol RST pada modul ESP8266.

3.6 Registrasi dan Konfigurasi Webhosting dan *Database*

3.6.1 Registrasi *Webhosting*

Untuk dapat menggunakan 000webhostapp sebagai pemantauan sistem, maka pengguna terlebih dahulu melakukan registrasi pada platform 000webhostapp dengan memasukkan email dan kata sandi, serta memilih paket yang diperlukan.



Gambar 3. 28 Paket Webhosting

Gambar 3. 29 Halaman Registrasi Webhosting

3.6.2 Pembuatan *Website Domain*

Tahapan selanjutnya membuat *domain website* dengan nama *kwhmonitoringtri* dan mengisi kata sandi sesuai ketentuan dan kolom yang tersedia.

Gambar 3. 30 Halaman Pembuatan Website Baru

3.6.3 Pembuatan *Database* dan Pengguna

Membuat *database* baru dengan memasukkan nama *database*, nama pengguna, dan kata sandi sesuai ketentuan pada kolom yang telah disediakan.



Gambar 3. 31 Halaman Pembuatan Database Baru

Database telah berhasil dibuat dengan spesifikasi seperti berikut :



Name	Pengguna	Host	Tabel	Ukuran	
SECONDARY_esp_data	SECONDARY_root	localhost	1	OK!	

Gambar 3. 32 Spesifikasi Database

Menuju phpMyAdmin untuk membuat tabel bernama *sensor_data* dengan kolom berisikan variabel yang akan dipantau.



Tabel	Tipe	Struktur	Coll	Unsigned	Not Null	Primary	Index	Size	Index
sensor_data	text	45	utf8mb4_0900_ai_ci	0	1	1	1	19.0 MB	1
waktu	datetime	19	utf8mb4_0900_ai_ci	0	1	1	1	19.0 MB	1

Gambar 3. 33 Tabel Database yang Dibuat

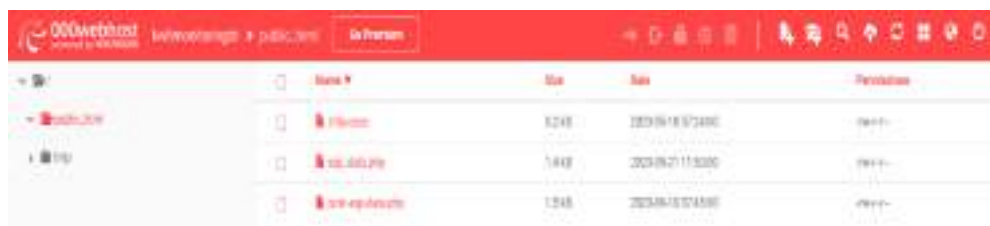
3.6.4 Pembuatan Skrip HTTP POST Menggunakan PHP

Untuk membuat skrip HTTP POST, terlebih dahulu menuju ke *File Manager*, kemudian membuka folder *public_html*.



Gambar 3. 34 Folder pada File Manager

Membuat *file* bernama *post-esp-data.php*.



Gambar 3. 35 File Terdaftar

Pada program *post-esp-data.php*, memasukkan nama server, nama *database*, nama pengguna, kata sandi, dan *API key*. Fungsi *if* (`$_SERVER["REQUEST_METHOD"] == "POST"`) digunakan untuk memeriksa metode permintaan yang diterima oleh server dalam PHP. Kemudian membuat koneksi dan query *INSERT INTO* sesuai tabel *database* *phpMyAdmin* yang telah dibuat sebelumnya.

```
/public_html/post-esp-data.php
1 <?php
2
3 $servername = "localhost";
4 $dbname = "id2876117_esp_data";
5 $username = "id2876117_root";
6 $password = "123456789";
7
8 $conn_key_value = "k7mAT34b0j7FF";
9
10 $api_key = $voltage = $current = $power = $energy = $harge = "";
11
12 if ($_SERVER["REQUEST_METHOD"] == "POST") {
13     $api_key = test_input($_POST["api_key"]);
14     if($api_key == $conn_key_value) {
15         $voltage = test_input($_POST["voltage"]);
16         $current = test_input($_POST["current"]);
17         $power = test_input($_POST["power"]);
18         $energy = test_input($_POST["energy"]);
19         $harge = test_input($_POST["harge"]);
20
21         // Create connection
22         $conn = mysqli_connect($servername, $username, $password, $dbname);
23         // Check connection
24         if (!$conn) {
25             die("Connection failed: " . mysqli_connect_error());
26         }
27
28         $sql = "INSERT INTO sensor_data (voltage, current, power, energy, harge)
29 VALUES ('" . $voltage . "', '" . $current . "', '" . $power . "', '" . $energy . "', '" . $harge . "')";
30
31         if (mysqli_query($conn, $sql)) {
32             echo "Record inserted successfully";
33         } else {
34             echo "Error: " . mysqli_error($conn);
35         }
36     }
37 }
```

Gambar 3. 36 Potongan File *post-esp-data.php*

3.6.5 Pembuatan Skrip Isi Database Menggunakan PHP

Pada skrip `esp_data.php` berfungsi untuk menampilkan isi *database* dengan menggunakan PHP dan HTML yang dapat diakses oleh pengguna melalui *link website*. Dalam skrip ini digunakan *query* SELECT yang menampilkan urutan data menggunakan ID dengan urutan *ascending*. Tabel yang ditampilkan juga menunjukkan waktu (*timestamp*) ketika data masuk saat sistem dijalankan.

```
/public_html/esp_data.php
1 <?php
2
3 $servername = "localhost";
4 $username = "root";
5 $password = "root";
6
7 // Create connection
8 $conn = new mysqli($servername, $username, $password, $dbname);
9 // Check connection
10 if ($conn->connect_error) {
11     die("Connection failed: " . $conn->connect_error);
12 }
13
14 $sql = "SELECT id, voltage, current, power, energy, charge, reading_time FROM sensor_data ORDER BY id ASC";
15
16 echo '
17 <table border="1" cellpadding="5" cellspacing="5"
18 <tr bgcolor="#f2f2f2" align="center"
19 <td>ID</td>
20 <td>Voltage</td>
21 <td>Current</td>
22 <td>Power</td>
23 <td>Energy</td>
24 <td>Charge</td>
25 <td>Timestamp</td>
26 </tr>';
27
28 if ($result = $conn->query($sql)) {
29     while ($row = $result->fetch_assoc()) {
30         $row_id = $row["id"];
31         $row_voltage = $row["voltage"];
32         $row_current = $row["current"];
33         $row_power = $row["power"];
34         $row_energy = $row["energy"];
35         $row_charge = $row["charge"];
36         $row_reading_time = $row["reading_time"];
37         $row_reading_time = date("Y-m-d H:i:s", strtotime($row_reading_time + "7 hours"));
38
39         echo '
40 <tr align="center"
41 <td>'. $row_id . '</td>
42 <td>'. $row_voltage . '</td>
43 <td>'. $row_current . '</td>
44 <td>'. $row_power . '</td>
45 <td>'. $row_energy . '</td>
46 <td>'. $row_charge . '</td>
47 <td>'. $row_reading_time . '</td>
48 </tr>';
49     }
50     $result->free();
51 }
```

Gambar 3. 37 Potongan File `esp_data.php`

3.6.6 Pembuatan Skrip Grafik Visualisasi

Pada pembuatan grafik visualisasi, skrip yang dibuat bernama `esp_chart.php`. Grafik dibuat dengan alamat URL <https://code.highcharts.com/highcharts.js> yang merupakan skrip JavaScript Highcharts untuk membuat grafik interaktif di halaman *webhosting*. Grafik terdiri dari lima variabel tertampil yaitu tegangan, arus, daya, energi, dan harga dengan sumbu x pada grafik adalah waktu (*timestamp*).

```
openk_tes/esp_chart.php
1 <?php
2 $servername = "localhost";
3 $username = "root@localhost";
4 $password = "root@localhost";
5 $servername = "localhost";
6 $username = "root@localhost";
7 $password = "root@localhost";
8
9 // Create connection
10 $conn = new mysqli($servername, $username, $password, $dbname);
11 // Check connection
12 if ($conn->connect_error) {
13     die("Connection failed: " . $conn->connect_error);
14 }
15
16 $sql = "SELECT id, voltage, current, power, energy, price, reading_time FROM sensor_data ORDER BY reading_time DESC LIMIT 200";
17 $result = $conn->query($sql);
18
19 while ($data = $result->fetch_assoc()) {
20     $rows[] = $data;
21 }
22
23 $reading_time = $rows[0]['reading_time'];
24
25 $i = 0;
26 while ($reading_time < $reading_time) {
27     $reading_time[$i] = date("Y-m-d H:i:s", strtotime("$reading_time + 1 hour"));
28     $i++;
29 }
30
31 $voltage = json_encode(array_reverse(array_column($rows, 'voltage')), JSON_NUMERIC_CHECK);
32 $current = json_encode(array_reverse(array_column($rows, 'current')), JSON_NUMERIC_CHECK);
33 $power = json_encode(array_reverse(array_column($rows, 'power')), JSON_NUMERIC_CHECK);
34 $energy = json_encode(array_reverse(array_column($rows, 'energy')), JSON_NUMERIC_CHECK);
35 $price = json_encode(array_reverse(array_column($rows, 'price')), JSON_NUMERIC_CHECK);
36 $reading_time = json_encode(array_reverse(array_column($rows, 'reading_time')), JSON_NUMERIC_CHECK);
```

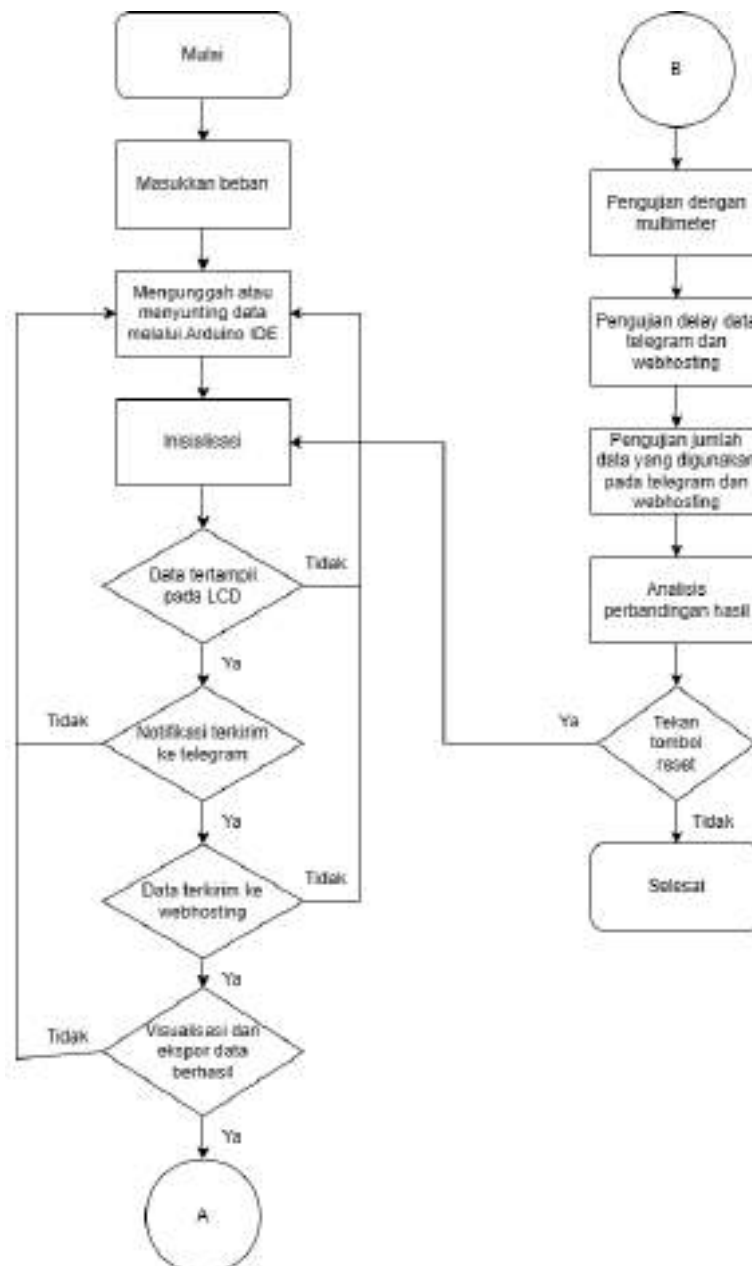
Gambar 3. 38 Potongan File `esp_chart.php`

```
<!DOCTYPE html>
<html>
<meta name="viewport" content="width=device-width, initial-scale=1">
<script src="https://code.highcharts.com/highcharts.js"></script>
<style>
body {
    min-width: 320px;
    max-width: 1300px;
    height: 500px;
    margin: 0 auto;
}
h2 {
    font-family: Arial, Helvetica;
    font-size: 36px;
    text-align: center;
}
</style>
<body>
<h1>IoT Monitoring System - Chart Mode</h1>
<div id="chart-voltage" class="container"></div>
<div id="chart-current" class="container"></div>
<div id="chart-power" class="container"></div>
<div id="chart-energy" class="container"></div>
<div id="chart-price" class="container"></div>
</body>
</html>
<script>
var voltage = <?php echo $voltage; ?>
var current = <?php echo $current; ?>
var power = <?php echo $power; ?>
var energy = <?php echo $energy; ?>
var price = <?php echo $price; ?>
var reading_time = <?php echo $reading_time; ?>
```

Gambar 3. 39 Potongan Pengaturan `esp_chart.php`

3.7 Skenario Pengujian

Pengujian sistem dilakukan menggunakan dua skenario berupa pengujian fungsional dan kinerja (performa). Pengujian fungsional merupakan proses yang digunakan untuk menjalankan rangka penelitian dengan spesifikasi uji perangkat keras maupun lunak pada suatu sistem. Sedangkan pengujian kinerja sistem berdasarkan pada pemeliharaan sistem dan memperbaiki kerusakan apabila ditemukan suatu masalah sebelum layak digunakan oleh pengguna. Berikut ini merupakan skenario pengujian yang akan dilakukan :



Gambar 3. 40 Skenario Pengujian Sistem

3.7.1 Pengujian Fungsional Sistem

Skenario pengujian fungsional terdiri dari beberapa tahapan meliputi, memasukkan beban yang diuji, mengunggah program melalui Arduino IDE, melakukan inisialisasi, data tertampil pada layar LCD, muncul notifikasi pada Telegram Bot, data terkirim ke *webhosting*, dan tertampil pada tabel, *database*, dan grafik sebagai visualisasi. Setelah proses inisialisasi, apabila data tidak dapat ditampilkan, maka dilakukan penyuntingan program kembali dan mengunggah program untuk dijalankan ulang.

3.7.2 Pengujian Kinerja Sistem

a. Pengujian Akurasi dengan Multimeter dan Tang Ampere

Pengujian akurasi dilakukan dengan cara membandingkan hasil perhitungan dari sensor dengan hasil perhitungan menggunakan multimeter dan tang ampere. Variabel yang diukur adalah tegangan dan arus. Hasil dari perhitungan multimeter dan tang ampere akan dibandingkan dengan *datasheet* yang dimiliki oleh sensor, kemudian menghitung tingkat akurasi.

b. Pengujian Konektivitas

- Pengujian *Delay* dan Rata-Rata *Delay*

Pengujian *delay* dilakukan dengan bantuan *platform* Wireshark. *Delay* merupakan waktu yang diperlukan oleh suatu data untuk mengirim seluruh jaringan. Untuk dapat melakukan pengujian waktu *delay*, maka harus menentukan alamat asal dan tujuan. Kemudian, mencari selisih waktu dari tujuan ke asal. Rumus perhitungan *delay* sebagai berikut :

$$\text{Delay} = \text{Waktu paket ke- (n+1)} - \text{waktu paket ke- n}$$

Selanjutnya, melakukan perhitungan rata-rata *delay* dengan menggunakan statistik yang ada di Wireshark dengan membagi total *delay* dengan total *packet* yang terima. Berikut merupakan rumus perhitungan dari rata-rata *delay* :

$$\text{Rata-rata Delay} = \frac{\text{Total paket yang diterima}}{\text{total waktu pengiriman}}$$

Terakhir, menentukan tingkat kualitas *delay* yang dihasilkan sesuai ketentuan yang bersumber dari TIPHON. Adapun TIPHON (*Telecommunications and Internet Protocol Harmonization Over Networks*) menjadi standarisasi QoS (*Quality of Services*) yang diluncurkan oleh ETSI (*European Telecommunications Standards Institute*) pada tahun 1998. Berikut merupakan tabel kategori *delay* dengan sumber TIPHON :

Tabel 3. 3 Kategori Delay Menurut TIPHON

Kategori Delay	Delay (ms)
<i>Excellent</i>	<150
<i>Good</i>	150 – 300
<i>Moderate</i>	300 – 450
<i>Poor</i>	>450

- Pengujian Ukuran Data yang Terpakai

Pengujian untuk melihat ukuran data yang terpakai dilakukan menggunakan Wireshark. Pengguna harus menentukan terlebih dahulu alamat asal dan tujuan yang akan diuji. Kemudian melihat ukuran data yang dikirimkan atau diterima untuk dianalisis.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

4.1 Hasil Pengujian Fungsional Sistem

4.1.1 Hasil Rangkaian Komponen

Berikut ini merupakan hasil dari komponen yang telah dirangkai sesuai desain sistem yang dibuat sebelumnya :



Gambar 4. 1 Rangkaian Komponen

4.1.2 Hasil Pengujian pada Layar LCD

Hasil pembacaan data pada layar LCD terdiri dari dua variabel, yaitu energi dan harga. Adapun energi terletak pada baris pertama dan harga pada baris kedua.



Gambar 4. 2 Tampilan Layar LCD

4.1.3 Hasil Pengujian pada *Serial Monitor*

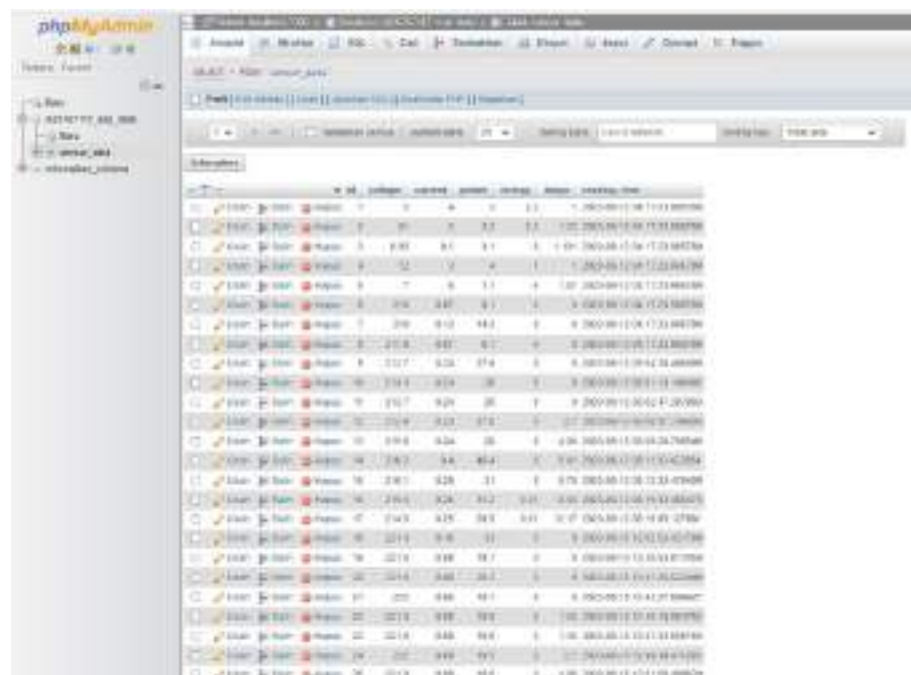
Hasil pembacaan data pada *serial monitor* terdiri dari IP yang terkoneksi Arduino IDE, hasil pembacaan variabel dari sensor, format pengiriman data ke *webhosting*, serta jumlah HTTP *response code*. Adapun IP yang digunakan adalah 192.168.1.49. Kemudian untuk jumlah HTTP *response code*, apabila 200 maka data berhasil dikirimkan, sedangkan apabila tertera *error code* = -1, maka data tidak dapat dikirimkan ke *webhosting*.

```
Calitip: 0.00 N  
Power: 13.70 W  
Energy: 0.10 kWh  
Range: Rp 0.00  
HTTP request Data: api_key=IPRAT5M35Tr3qvoInsgs*222.20&lat=5.034&power=13.70&energy=0.004&range=0.00  
HTTP response code: 200
```

Gambar 4. 3 Tampilan pada Serial Monitor

4.1.4 Hasil Pengujian pada *Database*

Berikut ini merupakan hasil pembacaan data pada *database*. Disini terdapat perbedaan *timestamp* dengan yang tertera pada *webhosting*, karena pada *webhosting* menggunakan waktu zona sesuai lokasi pengguna. Sedangkan pada *database* menggunakan *timestamp* UTC (*Universal Time Coordinated*) yang menjadi standar waktu internasional sebagai referensi umum dalam mengkoordinasikan waktu di seluruh dunia.



id	lokasi	energi	power	range	status	timestamp
1	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
2	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
3	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
4	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
5	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
6	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
7	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
8	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
9	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
10	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
11	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
12	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
13	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
14	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
15	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
16	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
17	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
18	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
19	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
20	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
21	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
22	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
23	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
24	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000
25	192.168.1.49	0.00	13.70	0.00	1	2020-09-11 08:11:33.000000

Gambar 4. 4 Tampilan pada Database

4.1.5 Hasil Pengujian Tabel pada Tampilan *Webhosting*

Hasil pembacaan tabel pada tampilan *webhosting* dapat diakses melalui link https://kwhmonitoringtri.00webhostapp.com/esp_data.php. Tabel disortir berdasarkan ID dengan urutan *ascending*. Pada kolom *timestamp*, disesuaikan dengan lokasi sehingga sama dengan waktu yang ada pada PC pengguna. Pembacaan data akan terus-menerus melakukan pembaruan hingga sistem dihentikan atau direset.

kWh Monitoring System

ID	Voltage	Current	Power	Energy	Hours	Timestamp
1	5	4	3	2.2	1	2023-06-12 11:17:23
2	51	3	2.2	2.2	1.22	2023-06-12 11:17:23
3	885	8.1	3.1	9	3.188	2023-06-12 11:17:25
4	12	3	4	1	1	2023-06-12 11:17:29
5	7	8	3.1	8	1.87	2023-06-12 11:17:29
6	218	8.97	6.1	9	0	2023-06-12 11:17:29
7	218	8.13	14.3	9	0	2023-06-12 11:17:29
8	217.9	8.97	6.1	9	0	2023-06-12 11:17:29
9	218.7	8.24	37.3	9	0	2023-06-12 12:04:28
10	214.3	8.24	28	9	0	2023-06-12 13:01:33

Gambar 4. 5 Tabel pada Tampilan *Webhosting*

4.1.6 Hasil Pengujian Grafik pada Tampilan *Webhosting*

Hasil pembacaan tabel pada tampilan *webhosting* dapat diakses melalui link https://kwhmonitoringtri.00webhostapp.com/esp_chart.php. Tabel disortir berdasarkan *reading_time* atau *timestamp*. Tampilan grafik menggunakan grafik garis dengan sumbu x sebagai *timestamp* dan sumbu y merupakan nilai dari variabel yang sedang diuji. Adapun pembacaan data akan terus-menerus melakukan pembaruan hingga sistem dihentikan atau direset.



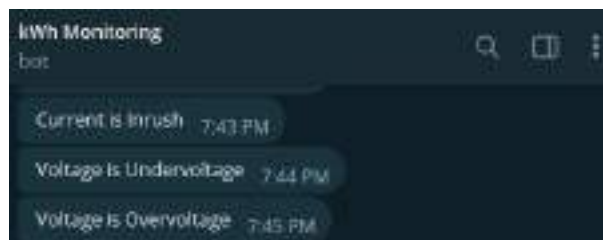
Gambar 4. 6 Grafik pada Tampilan *Webhosting*

4.1.7 Hasil Pengujian pada Telegram

Data yang dikirim dari sensor berhasil masuk ke Telegram sebagai notifikasi yang diberi jeda waktu setiap satu menit sekali sesuai ID yang terdaftar. Dimulai dari munculnya notifikasi “*kWh Monitoring Bot is started.*” Kemudian akan mengirimkan notifikasi dari perhitungan data sensor berupa variabel tegangan, arus, daya, energi, dan harga. Apabila kapasitas tegangan dan arus tidak sesuai dengan batas yang ditentukan, maka terkirim peringatan dini seperti pada gambar 4.7.



Gambar 4. 7 Tampilan pada Telegram



Gambar 4. 8 Notifikasi Peringatan pada Telegram

4.2 Hasil Pengujian Kinerja Sistem

4.2.1 Hasil Pengujian Akurasi Sistem

Berikut merupakan hasil pengujian menggunakan multimeter dan tang ampere dalam satuan Volt untuk parameter tegangan dan Ampere untuk parameter arus :

Tabel 4. 1 Hasil Pengujian Tegangan dengan Multimeter

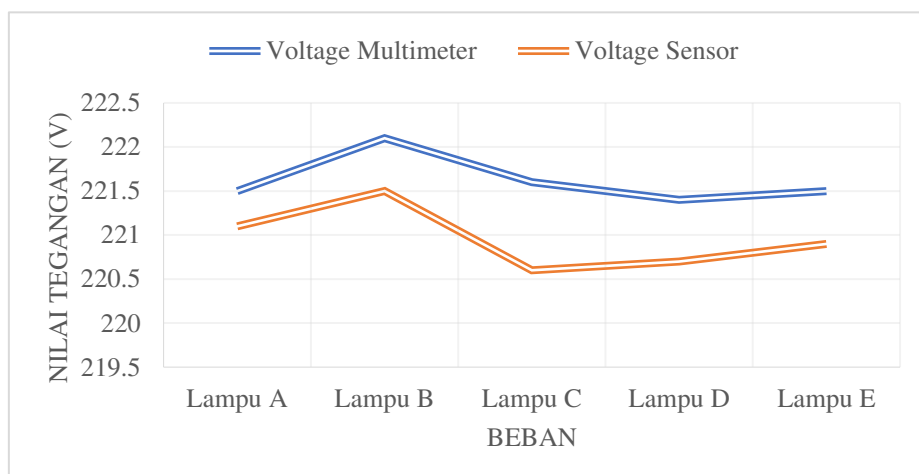
Beban	Nilai Tegangan (V)	Nilai Arus (A)
Lampu A	221.5	2.57
Lampu B	222.1	3.03
Lampu C	221.6	4.1
Lampu D	221.4	4.57
Lampu E	221.5	5.57

Berikut merupakan hasil pengujian menggunakan sensor dalam satuan Volt untuk parameter tegangan dan Ampere untuk parameter arus :

Tabel 4. 2 Hasil Pengujian dengan Sensor

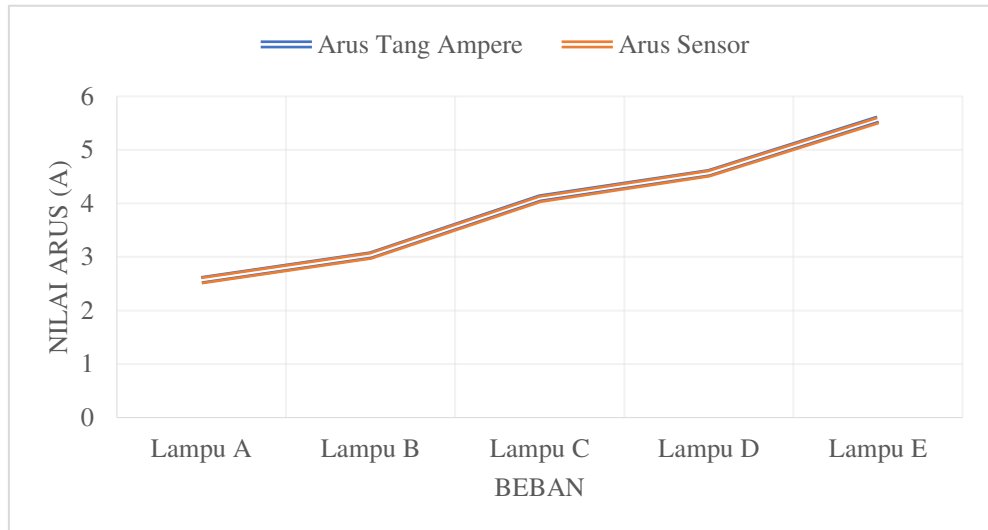
Beban	Nilai Tegangan (V)	Nilai Arus (A)
Lampu A	221.1	2.56
Lampu B	221.5	3.02
Lampu C	220.6	4.08
Lampu D	220.7	4.56
Lampu E	220.9	5.55

Berikut ini merupakan perbandingan hasil pengujian variabel tegangan ketika menggunakan multimeter dengan pengujian menggunakan sensor dengan tampilan berbasis grafik dengan sumbu x merupakan beban yang diuji dan sumbu y merupakan nilai tegangan yang teruji:



Gambar 4. 9 Perbandingan Pengujian Tegangan

Sedangkan perbandingan hasil pengujian variabel arus ketika menggunakan tang ampere dengan pengujian menggunakan sensor dengan tampilan berbasis grafik dengan sumbu x merupakan beban yang diuji dan sumbu y merupakan nilai arus yang teruji adalah sebagai berikut :



Gambar 4. 10 Perbandingan Pengujian Arus

Dari pengujian di atas, dilakukan perhitungan akurasi sebagai berikut :

Tabel 4. 3 Hasil Pengujian Akurasi Tegangan dan Arus

Beban	Selisih Voltage	Persentase Error Voltage	Selisih Arus	Persentase Error Arus
Lampu A	0.4	0.18%	0.01	0.39%
Lampu B	0.6	0.27%	0.01	0.33%
Lampu C	1	0.45%	0.02	0.48%
Lampu D	0.7	0.31%	0.01	0.21%
Lampu E	0.6	0.27%	0.02	0.36%
Rata-Rata Error	0.66	0.296%	0.0014	0.354%
Nilai Akurasi	99.704%		99.646%	

Berdasarkan hasil pengujian akurasi yang didapatkan, sensor berjalan dengan normal karena nilai *error* yang didapatkan kurang dari 0.5% yaitu memiliki rata-rata *error* sebesar 0.296% pada pengujian tegangan dan 0.354% pada pengujian arus. Dengan demikian, nilai akurasi yang dimiliki sistem sebesar 99.704% pada pengujian tegangan dan 99.646% pada pengujian arus.

4.2.2 Hasil Pengujian Konektivitas

a. Hasil Pengujian *Delay* dan Rata-Rata *Delay*

- Hasil Pengujian *Delay* dari Arduino IDE ke Tabel *Webhosting*

Tabel 4. 4 Hasil Pengujian Delay ke Tabel Webhosting

Alamat IP Asal	Alamat IP Tujuan	Packet ke-	Waktu (s)
192.168.1.59	52.182.143.210	799	128.701344
52.182.143.210	192.168.1.59	800	128.703181
Total Delay			0.001837

Dari pengujian *delay* Arduino IDE ke tabel *webhosting*, didapatkan *delay* sebesar 0.001837 s atau 1.837 ms. *Delay* tersebut masuk ke dalam kategori *excellent* berdasarkan tabel TIPHON, karena besaran *delay* kurang dari 150 ms.

Rata-rata *delay* yang didapatkan dari statistik :

$$\begin{aligned}\text{Total Delay} / \text{Total Packet yang Diterima} &= 189.249 / 1239 \\ &= 0.15274 \text{ s} \\ &= 152.74 \text{ ms}\end{aligned}$$

Dari hasil rata-rata *delay* yang didapatkan yaitu sebesar 152.74 ms, rata-rata tersebut masuk ke dalam kategori *good*, karena besaran *delay* berada pada rentang 150 – 300 ms.

- Hasil Pengujian *Delay* dari Arduino IDE ke Grafik *Webhosting*

Tabel 4. 5 Hasil Pengujian Delay ke Grafik Webhosting

Alamat IP Asal	Alamat IP Tujuan	Packet ke-	Waktu (s)
192.168.1.59	20.195.114.44	376	85.366711
20.195.114.44	192.168.1.59	377	85.370850
Total Delay			0.004139

Dari pengujian *delay* Arduino IDE ke grafik *webhosting*, didapatkan *delay* sebesar 0.004139 s atau 4.139 ms. *Delay* tersebut masuk ke dalam kategori *excellent* berdasarkan tabel TIPHON, karena besaran *delay* kurang dari 150 ms.

Rata-rata *delay* yang didapatkan dari statistik :

$$\begin{aligned}\text{Total Delay} / \text{Total Packet yang Diterima} &= 212.254 / 1236 \\ &= 0.17172 \text{ s} \\ &= 171.72 \text{ ms}\end{aligned}$$

Dari hasil rata-rata *delay* yang didapatkan yaitu sebesar 171.72 ms, rata-rata tersebut masuk ke dalam kategori *good*, karena besaran *delay* berada pada rentang 150 – 300 ms.

- Hasil Pengujian *Delay* dari Arduino IDE ke Telegram

Tabel 4. 6 Hasil Pengujian Delay ke Telegram

Alamat IP Asal	Alamat IP Tujuan	Packet ke-	Waktu (s)
192.168.1.59	204.79.197.254	1871	80.333688
204.79.197.254	192.168.1.59	1872	80.335275
Total Delay			0.001587

Dari pengujian *delay* Arduino IDE ke tabel *webhosting*, didapatkan *delay* sebesar 0.001587 s atau 1.587 ms. *Delay* tersebut masuk ke dalam kategori *excellent* berdasarkan tabel TIPHON, karena besaran *delay* kurang dari 150 ms.

Rata-rata *delay* yang didapatkan dari statistik :

$$\begin{aligned} \text{Total Delay / Total Packet yang Diterima} &= 101.625 / 2713 \\ &= 0.03746 \text{ s} \\ &= 37.46 \text{ ms} \end{aligned}$$

Dari hasil rata-rata *delay* yang didapatkan yaitu sebesar 37.46 ms, rata-rata tersebut masuk ke dalam kategori *excellent*, karena besaran *delay* kurang dari 150 ms.

b. Hasil Pengujian Ukuran Data Terpakai

- Hasil Pengujian Ukuran Data Terpakai dari Arduino IDE ke Tabel *Webhosting*

Tabel 4. 7 Pengujian Ukuran Data Terpakai ke Tabel Webhosting

Alamat Asal	192.168.1.59	52.182.143.210
Alamat Tujuan	52.182.143.210	192.168.1.59
Port Asal	52689	443
Port Tujuan	443	52689
Ukuran Data Terpakai (bytes)	66	1506
Ukuran Data Terpakai (bits)	528	12048

Tabel tersebut menunjukkan hasil ukuran data yang terpakai dari Arduino IDE ke tampilan tabel pada *webhosting* maupun sebaliknya. Adapun hasil yang didapatkan adalah sebesar 528 bits untuk pengiriman data dan 12048 bits untuk penerimaan data, karena data dari HTTPS dengan port 443 berbasis gambar.

- Hasil Pengujian Ukuran Data Terpakai dari Arduino IDE ke Grafik *Webhosting*

Tabel 4. 8 Pengujian Ukuran Data Terpakai ke Grafik Webhosting

Alamat Asal	192.168.1.59	20.195.114.44
Alamat Tujuan	20.195.114.44	192.168.1.59
Port Asal	63800	443
Port Tujuan	443	63800
Ukuran Data Terpakai (bytes)	66	1506
Ukuran Data Terpakai (bits)	528	12048

Tabel tersebut menunjukkan hasil ukuran data yang terpakai dari Arduino IDE ke tampilan grafik pada *webhosting* maupun sebaliknya. Adapun hasil yang didapatkan adalah sebesar 528 bits untuk pengiriman data dan 12048 bits untuk penerimaan data, karena data dari HTTPS dengan port 443 berbasis gambar.

- Hasil Pengujian Ukuran Data Terpakai dari Arduino IDE ke Telegram

Tabel 4. 9 Pengujian Ukuran Data Terpakai ke Telegram

Alamat Asal	192.168.1.59	204.79.197.254
Alamat Tujuan	204.79.197.254	192.168.1.59
Port Asal	63875	443
Port Tujuan	443	63875
Ukuran Data Terpakai (bytes)	66	254
Ukuran Data Terpakai (bits)	528	2032

Tabel tersebut menunjukkan hasil ukuran data yang terpakai dari Arduino IDE ke Telegram Bot maupun sebaliknya. Adapun hasil yang didapatkan adalah sebesar 528 bits untuk pengiriman data dan 2032 bits untuk penerimaan data.

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan oleh penulis, didapatkan beberapa kesimpulan sebagai berikut :

1. Sistem berhasil dijalankan baik dari tampilan LCD, notifikasi Telegram, maupun melalui *webhosting* dengan tabel dan grafik untuk pemantauan energi listrik skala rumah tangga.
2. Sistem mampu mengirimkan notifikasi peringatan dini terkait kondisi arus dan tegangan ketika terdapat potensi bahaya.
3. Tingkat *error* yang dimiliki oleh sensor sebesar 0.296% pada pengujian tegangan dan 0.354% pada pengukuran arus atau tingkat akurasi sebesar 99.704% pada pengujian tegangan dan 99.646% pada pengujian arus yang menunjukkan kinerja baik karena tidak melebihi nilai *error* pada *datasheet* sebesar 0.5% pada masing-masing variabel.
4. Konektivitas yang dimiliki oleh sistem berjalan dengan stabil, *delay* memiliki kualitas *excellent* atau pada rata-rata *delay* memiliki kualitas *good*.
5. *Delay* paling sedikit ketika Arduino IDE mengirimkan data ke Telegram yaitu sebesar 1.587 ms.

5.2 Saran

Berdasarkan hasil penelitian yang telah dianalisis oleh penulis, dapat disimpulkan bahwa penelitian masih membutuhkan beberapa saran pengembangan sebagai berikut :

1. Pengembangan dalam tampilan sistem pemantauan energi listrik rumah tangga yang dibuat, khususnya pada tampilan *webhosting*.
2. Diperlukan perbaikan dalam penggunaan jaringan agar dapat terkoneksi baik menggunakan SSO atau *tethering*.
3. Perlunya pengembangan terkait model pengujian yang dilakukan, tidak terbatas pada variabel tegangan dan arus saja.

DAFTAR PUSTAKA

- Adiwiranto, Mohammad Nursamsi, et al. "Prototipe Sistem *Monitoring* Konsumsi Energi Listrik serta Estimasi Biaya pada Peralatan Rumah Tangga Berbasis *Internet of Things*." *Jurnal Edukasi Elektro*, vol. 06, no. 1, 2022, pp. 36-41. Diakses pada 15 Maret 2023.
- Al Bustomi, M. Abi Yazid, and Asmunin. "Rancang Bangun Sistem *Monitoring* Peringatan Dini Bencana Banjir Berbasis IoT Menggunakan Protokol MQTT dengan Notifikasi Bot Telegram." Universitas Negeri Surabaya, 2021. Diakses pada 18 Maret 2023.
- Aziz, Nofvan Maghresa. "Pemrograman Sistem Monitoring Panel Tegangan Rendah Berbasis IoT." Politeknik Negeri Jakarta, 2021, p. 62. Diakses pada 18 Maret 2023.
- Chairunnisa, Indah, and Wildian. "Rancang Bangun Alat Pemantau Biaya Pemakaian Energi Listrik Menggunakan Sensor PZEM-004T dan Aplikasi Blynk." *Jurnal Fisika Unand*, vol. 11, no. 2, 2022, pp. 253-255. Diakses pada 18 Maret 2023.
- Giarniasih, Hasna Naufal, and Trie Maya Kadarina. "*Smart* Sistem Untuk Pemantauan Dan Pengisian Pulsa Listrik Prabayar *Via Smartphone*." *Jurnal Teknologi Elektro Universitas Mercu Buana*, vol. 10, no. 2, 2019, pp. 101-102. Diakses pada 15 Maret 2023.
- Habibi, Fatoni Nur, et al. "Alat *Monitoring* Pemakaian Energi Listrik Berbasis Android Menggunakan Modul PZEM-004T." *Prosiding Seminar Nasional Teknologi Elektro Terapan*, vol. 01, no. 1, 2017, pp. 160-162. Diakses pada 15 Maret 2023.
- Komputer, Wahana. *Panduan Belajar MySQL Database Server*. Diedit oleh Sudarma S., 1 ed., Jakarta Selatan, MediaKita, 2010. Diakses pada 21 Maret 2023.
- Maulidin, Muhammad Abdur Rofi, et al. "Perancangan Sistem *Monitoring* Penggunaan Air PAM Berbasis IoT dengan Bot Telegram." *Indonesian Journal of Technology, Informatics, and Science*, vol. 02, no. 1, 2020, p. 49. Diakses pada 18 Maret 2023.
- Raed, Abd-Alhameed, et al., *editors*. *Imdc-Ist 2021: Proceedings of 2nd International Multi-Disciplinary Conference Theme: Integrated Sciences and Technologies, IMDC-IST 2021, 7-9 September 2021, Sakarya, Turkey*. 2 ed., Sakarya, EAI Publishing, 2022. Diakses pada 20 Maret 2023.

- Suhartono, et al. Desain Prototipe Reaktor Plasma Untuk *Plasma Activated Water* (PAW) sebagai Pupuk Cair Nitrogen Menggunakan *Dielectric Barrier Discharge* (DBD), *Internet of Things* (IoT) dan Logika Kabur. Diedit oleh Totok Chamidy, 1 ed., Malang, Academia Publication, 2021. Diakses pada 20 Maret 2023.
- Trigreisian, Alwizain Almas, and Nisa Hanum Harani. Telegram Bot Wawancara Kerja dengan Algoritma *Long Short Term Memory*. Diedit oleh Cahyo Priyanto, 1 ed., Bandung, Penerbit Buku Pedia, 2023. Diakses pada 21 Maret 2023.
- Triyanto, Dedi, et al. “*Prototype Sistem Monitoring dan Pengisian Token Listrik Prabayar Menggunakan Arduino Uno Berbasis Website.*” *Coding : Jurnal Komputer dan Aplikasi*, vol. 08, no. 3, 2020, pp. 64-72. Diakses pada 18 Maret 2023.
- Wahyu, M. Ilham Ludya, et al. “Rancang Bangun kWh Meter Digital Sebagai Penghitung Biaya Pemakaian Energi Listrik Berbasis Arduino Uno R3.” *Jurusan Teknik Elektro Fakultas Teknik Universitas Tanjungpura*, 2018, pp. 6-7. Diakses pada 15 Maret 2023.
- Wangsadinata, Wiratman, and G. Suprayitno, *editors*. Roosseno, Jembatan dan Menjembatani. 1 ed., Jakarta, Yayasan Obor Indonesia, 2008. Diakses pada 20 Maret 2023.
- Wirasmita, Rasyid Hardi, et al. “Sistem *Monitoring* Pemakaian Daya Listrik pada kWh Meter Menggunakan Arduino dan SMS *Gateway*.” *Jurnal Disprotek*, vol. 13, no. 1, 2022, pp. 69-72. Diakses pada 15 Maret 2023.
- Yudhanto, Yudho, and Abdul Azis. Pengantar Teknologi Internet of Things (IoT). Diedit oleh Eko Harry Pratisto, 1 ed., Surakarta, UNSPress, 2019. Diakses pada 20 Maret 2023.
- Agung, Achmad Imam. “Potensi Sumber Energi Alternatif dalam Mendukung Kelistrikan Nasional.” *Jurnal Pendidikan Teknik Elektro*, 2013, pp. 892-895. Diakses pada 25 Maret 2023.
- Prayitno, Budi, et al. “Prototipe Sistem *Monitoring* Penggunaan Daya Listrik Peralatan Elektronik Rumah Tangga Berbasis *Internet of Things*.” *Jurnal Petir*, vol. 12, no. 1, 2019, pp. 73-74. Diakses pada 25 Maret 2023.
- Ritonga, Ahmad Fitra, et al. “Implementasi *Internet of Things* (IoT) untuk Meningkatkan Kompetensi Siswa SMK Jakarta 1.” *Risenologi (Jurnal Sains, Teknologi, Sosial, Pendidikan, dan Bahasa)*, vol. 5, no. 1, 2020, pp. 1-2. Diakses pada 25 Maret 2023.

LAMPIRAN

Lampiran 1.1 *Datasheet* Sensor PZEM-004T V3.0

PZEM-004T V3.0 User Manual

Overview

This document describes the specification of the PZEM-004T AC communication module, the module is mainly used for measuring AC voltage, current, active power, frequency, power factor and active energy, the module is without display function, the data is read through the TTL interface.

PZEM-004T-10A: Measuring Range 10A (Built-in Shunt)

PZEM-004T-100A: Measuring Range 100A (external transformer)

I. Function description

1.1 Voltage

- 1.1.1 Measuring range: 80~260V
- 1.1.2 Resolution: 0.1V
- 1.1.3 Measurement accuracy: 0.5%

1.2 Current

- 1.2.1 Measuring range: 0~10A (PZEM-004T-10A); 0~100A (PZEM-004T-100A)
- 1.2.2 Starting measure current: 0.01A (PZEM-004T-10A); 0.02A (PZEM-004T-100A)
- 1.2.3 Resolution: 0.001A
- 1.2.4 Measurement accuracy: 0.5%

1.3 Active power

- 1.3.1 Measuring range: 0~2.3kW (PZEM-004T-10A); 0~23kW (PZEM-004T-100A)
- 1.3.2 Starting measure power: 0.4W
- 1.3.3 Resolution: 0.1W
- 1.3.4 Display format:
 - <1000W, it display one decimal, such as: 999.9W
 - ≥1000W, it display only integer, such as: 1000W
- 1.3.5 Measurement accuracy: 0.5%

1.4 Power factor

1.4.1 Measuring range: 0.00~1.00

1.4.2 Resolution: 0.01

1.4.3 Measurement accuracy: 1%

1.5 Frequency

1.5.1 Measuring range: 45Hz~65Hz

1.5.2 Resolution: 0.1Hz

1.5.3 Measurement accuracy: 0.5%

1.6 Active energy

1.6.1 Measuring range: 0~9999.99kWh

1.6.2 Resolution: 1Wh

1.6.3 Measurement accuracy: 0.5%

1.6.4 Display format:

<10kWh, the display unit is Wh(1kWh=1000Wh), such as: 9999Wh

≥10kWh, the display unit is kWh, such as: 9999.99kWh

1.6.5 Reset energy: use software to reset.

Lampiran 1.2 Program Arduino IDE

```
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClientSecure.h>
#include <UniversalTelegramBot.h>
#include <ESP8266WebServer.h>
#include <LiquidCrystal_I2C.h>
#include <PZEM004Tv30.h>
#include <SoftwareSerial.h>
#include <Wire.h>

#define WIFI_SSID "****"
#define WIFI_PASSWORD "****"
#define BOT_TOKEN "5874143638:AAHf2PoFx1xg3l5zAXKVFaRZPKlwKTlKE_k"
#define serverName https://kwhmonitoringtri.000webhostapp.com/post-esp-data.php

byte mac[] = {0xB4, 0x0E, 0xDE, 0xA4, 0x0B, 0x05};
PZEM004Tv30 pzem(D5, D6);
LiquidCrystal_I2C lcd(0x27, 16, 2);
WiFiClientSecure wifiClient;
UniversalTelegramBot bot(BOT_TOKEN, wifiClient);
IPAddress ip;

const int RESET_PIN = 0;
// Masukkan ID user dan admin
int chatIds[] = {1388217255, 774702581};
String api_key_value = "tPmAT5Ab3j7F9";
char server[] = "localhost";
const int port = 80;

void sendMessageToAll(String message) {
  for (int i = 0; i < static_cast<int>(sizeof(chatIds) / sizeof(chatIds[0])); i++) {
    bot.sendMessage(String(chatIds[i]), message);
  }
}
```

```

void setup() {
  Serial.begin(115200);
  //pzemSerial.begin(9600);
  wifiClient.setInsecure();
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
  }
  ip = WiFi.localIP();
  Serial.print("Connected to: ");
  Serial.println(ip);

  lcd.begin();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Initializing...");
  delay(2000); lcd.clear();
  String message = "kWh Monitoring Bot is started!";
  sendMessageToAll(message);
  delay(2000); lcd.clear();
  float energy1 = pzem.resetEnergy();
}

void loop() {
  float energy = pzem.energy();
  float harga = energi * 1352.00;
  float voltage = pzem.voltage();
  float current = pzem.current();
  float power = pzem.power();

```

```

if (energy >= 0.000) {
    lcd.setCursor(0, 0);
    lcd.print("Energy : ");
    lcd.print(energy);
    lcd.print(" kWh");
    lcd.setCursor(0, 1);
    lcd.print("Harga : ");
    lcd.print("Rp ");
    lcd.print(harga);
}
else if (energy != NAN) {
    lcd.setCursor(0, 0);
    lcd.print("Energy : ");
    lcd.print(energy);
    lcd.print(" kWh");
    lcd.setCursor(0, 1);
    lcd.print("Harga : ");
    lcd.print("Rp ");
    lcd.print(harga);
}

else {
    lcd.setCursor(0, 0);
    lcd.print("Error reading energy");
}

String message = "Voltage: " + String(voltage) + " V\n" +
    "Current: " + String(current) + " A\n" +
    "Power: " + String(power) + " W\n" +
    "Energy: " + String(energy) + " kWh\n" +
    "Harga: " + "Rp " + String(harga);
sendMessageToAll(message);
delay(60000); lcd.clear();
Serial.println(message);

```

```

if (digitalRead(RESET_PIN) == LOW) {
    delay(2000); lcd.clear();
}

if (voltage <= 198.00) {
    String message = "Voltage is Undervoltage";
    sendMessageToAll(message);
}
if (voltage >= 242.00) {
    String message = "Voltage is Overvoltage";
    sendMessageToAll(message);
}
if (current >= 4.00) {
    String message = "Current is Inrush";
    sendMessageToAll(message);
}

if (WiFi.status() == WL_CONNECTED) {
    HTTPClient https;
    https.begin(wifiClient, serverName);
    https.addHeader("Content-Type", "application/x-www-form-urlencoded");

    String httpRequestData =
"api_key="+api_key_value+"&voltage="+voltage+"&current="+current+"&power="+power+"&en
ergy="+energy+"&harga="+harga;
    Serial.print("httpRequestData: ");
    Serial.println(httpRequestData);

    int httpResponseCode = https.POST(httpRequestData);

    if (httpResponseCode > 0) {
        Serial.print("HTTP Response Code: ");
        Serial.println(httpResponseCode);
    }
}

```

```
else {  
    Serial.print("Error code: ");  
    Serial.println(httpResponseCode);  
}  
https.end();  
}  
else {  
    Serial.println("WiFi Disconnected");  
}  
}
```

Lampiran 1.3 Program PHP dan HTML

1. post-esp-data.php

```
<?php

$servername = "localhost";
$dbname = "id20767117_esp_data";
$username = "id20767117_root";
$password = "root";

$api_key_value = "tPmA'f5Ab3j7l-9";

$api_key= $voltage = $current = $power = $energy = $harga = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $api_key = test_input($_POST["api_key"]);
    if($api_key == $api_key_value) {
        $voltage = test_input($_POST["voltage"]);
        $current = test_input($_POST["current"]);
        $power = test_input($_POST["power"]);
        $energy = test_input($_POST["energy"]);
        $harga = test_input($_POST["harga"]);

        // Create connection
        $conn = new mysqli($servername, $username, $password, $dbname);
        // Check connection
        if ($conn->connect_error) {
            die("Connection failed: " . $conn->connect_error);
        }

        $sql = "INSERT INTO sensor_data (voltage, current, power, energy, harga)
VALUES ('" . $voltage . "', '" . $current . "', '" . $power . "', '" . $energy . "', '" . $harga . "')";

        if ($conn->query($sql) === TRUE) {
            echo "New record created successfully";
        }
    }
}
```



```

else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
}
else {
    echo "Wrong API Key provided.";
}
}

else {
    echo "No data posted with HTTP POST.";
}

function test_input($data) {
    $data = trim($data);
    $data = stripslashes($data);
    $data = htmlspecialchars($data);
    return $data;
}

```

2. esp_data.php

```
<!DOCTYPE html>
<html>
<head>
    <title>kWh Monitoring System</title>
</head>
<body>
    <h1>kWh Monitoring System</h1>
<?php

$servername = "localhost";
$dbname = "id20767117_esp_data";
$username = "id20767117_root";
$password = "123456";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT id, voltage, current, power, energy, harga, reading_time FROM sensor_data
ORDER BY id ASC";

echo '<table border="1" cellpadding="5" cellspacing="5">
    <tr bgcolor="#F1F1F1" align="center">
        <td><b>ID</b></td>
        <td><b>Voltage</b></td>
        <td><b>Current</b></td>
        <td><b>Power</b></td>
        <td><b>Energy</b></td>
        <td><b>Harga</b></td>
        <td><b>Timestamp</b></td>
    </tr>';
```

```

if ($result = $conn->query($sql)) {
    while ($row = $result->fetch_assoc()) {
        $row_id = $row["id"];
        $row_voltage = $row["voltage"];
        $row_current = $row["current"];
        $row_power = $row["power"];
        $row_energy = $row["energy"];
        $row_harga = $row["harga"];
        $row_reading_time = $row["reading_time"];
        $row_reading_time = date("Y-m-d H:i:s", strtotime("$row_reading_time + 7 hours"));
        echo '<tr align="center">
            <td>' . $row_id . '</td>
            <td>' . $row_voltage . '</td>
            <td>' . $row_current . '</td>
            <td>' . $row_power . '</td>
            <td>' . $row_energy . '</td>
            <td>' . $row_harga . '</td>
            <td>' . $row_reading_time . '</td>
        </tr>';
    }
    $result->free();
}

$conn->close();
?>
</table>
</body>
</html>

```

3. esp_chart.php

```
</php

$servername = "localhost";
$dbname = "id20767117_esp_data";
$username = "id20767117_root";
$password = "****";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "SELECT id, voltage, current, power, energy, harga, reading_time FROM sensor_data
order by reading_time desc limit 100";

$result = $conn->query($sql);

while ($data = $result->fetch_assoc()){
    $sensordata[] = $data;
}

$readings_time = array_column($sensordata, 'reading_time');
$i = 0;
foreach ($readings_time as $reading){
    $readings_time[$i] = date("Y-m-d H:i:s", strtotime("$reading + 7 hours"));
    $i += 1;
}
```

```

$ voltage = json_encode(array_reverse(array_column($sensordata, 'voltage')),
JSON_NUMERIC_CHECK);
$ current = json_encode(array_reverse(array_column($sensordata, 'current')),
JSON_NUMERIC_CHECK);
$ power = json_encode(array_reverse(array_column($sensordata, 'power')),
JSON_NUMERIC_CHECK);
$ energy = json_encode(array_reverse(array_column($sensordata, 'energy')),
JSON_NUMERIC_CHECK);
$ harga = json_encode(array_reverse(array_column($sensordata, 'harga')),
JSON_NUMERIC_CHECK);
$ reading_time = json_encode(array_reverse($readings_time), JSON_NUMERIC_CHECK);

/*echo $voltage;
echo $current;
echo $power;
echo $energy;
echo $harga;
echo $reading_time;*/

$result->free();
$conn->close();
?>

<!DOCTYPE html>
<html>
<meta name="viewport" content="width=device-width, initial-scale=1">
<script src="https://code.highcharts.com/highcharts.js"></script>
<style>
body {
    min width: 310px;
    max width: 1300px;
    height: 500px;
    margin: 0 auto;
}

```

```

h2 {
    font-family: Arial Black;
    font-size: 5rem;
    text-align: center;

}
</style>
<body>

    <h1>kWh Monitoring System - Chart Mode</h1>
    <div id="chart-voltage" class="container"></div>
    <div id="chart-current" class="container"></div>
    <div id="chart-power" class="container"></div>
    <div id="chart-energy" class="container"></div>
    <div id="chart-harga" class="container"></div>

<script>

var voltage = <?php echo $voltage; ?>;
var current = <?php echo $current; ?>;
var power = <?php echo $power; ?>;
var energy = <?php echo $energy; ?>;
var harga = <?php echo $harga; ?>;
var reading_time = <?php echo $reading_time; ?>;

```

```

var chartV = new Highcharts.Chart({
  chart: { renderTo: 'chart-voltage' },
  title: { text: 'PZEM004T Voltage' },
  series: [{
    showInLegend: false,
    data: tegangan
  }],
  plotOptions: {
    line: { animation: false,
      dataLabels: { enabled: true }
    },
    series: { color: '#663399' }
  },
  xAxis: {
    type: 'datetime',
    categories: reading_time
  },
  yAxis: {
    title: { text: 'Voltage (Volt)' }
  },
  credits: { enabled: false }
});

```

```

var chartC = new Highcharts.Chart({
  chart:{ renderTo:'chart-current' },
  title: { text: 'PZEM004T Current' },
  series: [{
    showInLegend: false,
    data: arus
  }],
  plotOptions: {
    line: { animation: false,
      dataLabels: { enabled: true }
    },
    series: { color: '#B22222' }
  },
  xAxis: {
    type: 'datetime',
    categories: reading_time
  },
  yAxis: {
    title: { text: 'Current (Ampere)' }
  },
  credits: { enabled: false }
});

```



```

var chartP = new Highcharts.Chart({
  chart:{ renderTo:'chart-power' },
  title: { text: 'PZEM004Tv30 Power' },
  series: [{
    showInLegend: false,
    data: daya
  }],
  plotOptions: {
    line: { animation: false,
      dataLabels: { enabled: true }
    },
    series: { color: '#4169E1' }
  },
  xAxis: {
    type: 'datetime',
    categories: reading_time
  },
  yAxis: {
    title: { text: 'Power (Watt)' }
  },
  credits: { enabled: false }
});

```

```

var chartE = new Highcharts.Chart({
  chart:{ renderTo : 'chart-energy' },
  title: { text: 'PZEM004Tv30 Energy' },
  series: [{
    showInLegend: false,
    data: energi
  }],
  plotOptions: {
    line: { animation: false,
      dataLabels: { enabled: true }
    },
    series: { color: '#DB7093' }
  },
  xAxis: {
    type: 'datetime',
    categories: reading_time
  },
  yAxis: {
    title: { text: 'Energy (kWh)' }
  },
  credits: { enabled: false }
});

```

```

var chartH = new Highcharts.Chart({
  chart:{ renderTo : 'chart-harga' },
  title: { text: 'PZEM004Tv30 Harga' },
  series: [{
    showInLegend: false,
    data: harga
  }],
  plotOptions: {
    line: { animation: false,
      dataLabels: { enabled: true }
    },
    series: { color: '#FF6347' }
  },
  xAxis: {
    type: 'datetime',
    categories: reading_time
  },
  yAxis: {
    title: { text: 'Harga (Rp)' }
  },
  credits: { enabled: false }
});

```

```

</script>

```

```

</body>

```

```

</html>

```