

**PENGEMBANGAN FIRMWARE SISTEM MONITORING
KONSUMSI LISTRIK MENGGUNAKAN IC RENERGY 8209 PADA
PLATFORM ESP32**

SKRIPSI



Disusun oleh:

MUHAMMAD RASYID
19/443590/TK/48786

**PROGRAM STUDI TEKNIK ELEKTRO
DEPARTEMEN TEKNIK ELEKTRO DAN TEKNOLOGI INFORMASI
FAKULTAS TEKNIK UNIVERSITAS GADJAH MADA
YOGYAKARTA
2023**

HALAMAN PENGESAHAN

PENGEMBANGAN *FIRMWARE SISTEM MONITORING KONSUMSI LISTRIK MENGGUNAKAN IC RENERGY 8209 PADA PLATFORM ESP32*

SKRIPSI

Diajukan sebagai Salah Satu Syarat untuk Memperoleh

Gelar Sarjana Teknik

pada Departemen Teknik Elektro dan Teknologi Informasi Fakultas Teknik

Universitas Gadjah Mada



Dosen Pembimbing I

Dr. I Wayan Mustika, S.T., M.Eng.
NIP. 198109212014041001

Dosen Pembimbing II

Ahmad Ataka Awwalur Rizqi , S.T., Ph.D
NIP. 111199207202108101



SKRIPSI

**PENGEMBANGAN FIRMWARE SISTEM MONITORING KONSUMSI LISTRIK
 MENGGUNAKAN IC RENERGY 8209 PADA PLATFORM ESP32**

Dipersiapkan dan disusun oleh

Muhammad Rasyid
19/443590/TK/48786

Telah dipertahankan di depan dewan penguji
pada tanggal : **26 Juni 2023**

Susunan Dewan Penguji

Pembimbing Utama

Dr. I Wayan Mustika, S.T., M.Eng.

Pembimbing Pendamping

Ahmad Ataka Awwalur Rizqi , S.T., Ph.D

Anggota Dewan Penguji Lain

Skripsi ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana Teknik

Tanggal: 17 Juli 2023
Pengelola Program Studi: Sarjana Teknik Elektro

Dr.Eng. Ir. Adha Imam Cahyadi, S.T., M.Eng., IPM.
NIP 197911022008121001

Mengetahui,
Ketua Departemen Teknik Elektro dan Teknologi Informasi



Prof. Ir. Hanung Adi Nugroho, S.T., M.Eng., Ph.D., IPM., SMIEEE.
NIP 197802242002121001



PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini :

Nama : Muhammad Rasyid
NIM : 19/443590/TK/48786
Tahun terdaftar : 2019
Program Studi : Teknik Elektro
Fakultas : Teknik Universitas Gadjah Mada

Menyatakan bahwa dalam dokumen ilmiah Skripsi 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-plagiasi dan apabila dokumen ilmiah Skripsi ini di kemudian hari terbukti merupakan plagiasi dari hasil karya penulis lain dan/atau dengan sengaja mengajukan karya atau pendapat yang merupakan hasil karya penulis lain, maka penulis bersedia menerima sanksi akademik dan/atau sanksi hukum yang berlaku.

Yogyakarta, 13 Juni 2023




Muhammad Rasyid
19/443590/TK/48786

HALAMAN PERSEMBAHAN

Invictus

by William Ernest Henley

Out of the night that covers me,
Black as the pit from pole to pole,
I thank whatever gods may be
For my unconquerable soul.

In the fell clutch of circumstance
I have not winced nor cried aloud.
Under the bludgeonings of chance
My head is bloody, but unbowed.

Beyond this place of wrath and tears
Looms but the Horror of the shade,
And yet the menace of the years
Finds and shall find me unafraid.

It matters not how strait the gate,
How charged with punishments the scroll,
I am the master of my fate,
I am the captain of my soul.

To Mom and Dad, Alhamdulillah, I finally did it!

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT atas limpahan rahmat, karunia, serta petunjuk-Nya sehingga tugas akhir berupa penyusunan skripsi ini telah terselesaikan dengan baik. Dalam hal penyusunan tugas akhir ini penulis telah banyak mendapatkan arahan, bantuan, serta dukungan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Bapak Ir. Hanung Adi Nugroho, S.T., M.E., Ph.D., IPM. Selaku Ketua Departemen Teknik Elektro dan Teknologi Informasi Fakultas Teknik Universitas Gadjah Mada
2. Bapak Ir. Lesnanto Multa Putranto, S.T., M.Eng., Ph.D., IPM. Selaku Sekretaris Departemen Teknik Elektro dan Teknologi Informasi Fakultas Teknik Universitas Gadjah Mada
3. Bapak Dr. I Wayan Mustika, S.T., M.Eng dan Ahmad Ataka Awwalur Rizqi, S.T., Ph.D. Selaku dosen pembimbing skripsi.
4. Ibu, bapak, adik, dan keluarga saya. *My number one support system and my motivation, always.*
5. Mas Firman Hidayat, Sunandar, Yaenuri, dan Wisnu. Sebagai senior dan kawan diskusi.
6. Mas Airlangga dan Pradhesta. Sebagai rekan seperjuangan, seerbimbangan, dan seperpusungan.
7. Mas Fanuel, Ismail, Mahligai, Martin, Miftahul, Naufal, Rauf, Thoriq dan Mba Aisyah, Angelita, Elisabetha, Filza, Natasha. Sebagai sahabat yang saya sangat syukuri telah dipertemukan oleh Allah SWT.
8. *All my idols, friends, and foes, that I respected.*

Akhir kata penulis berharap semoga skripsi ini dapat memberikan manfaat bagi kita semua, aamiin.

DAFTAR ISI

HALAMAN PENGESAHAN	i
PERNYATAAN BEBAS PLAGIASI	iii
HALAMAN PERSEMPAHAN	iv
KATA PENGANTAR	v
DAFTAR ISI.....	vi
DAFTAR TABEL	viii
DAFTAR GAMBAR	ix
DAFTAR SINGKATAN.....	xi
INTISARI.....	xii
ABSTRACT	xiii
BAB I Pendahuluan	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	3
1.4 Batasan Penelitian	3
1.5 Manfaat Penelitian	4
1.6 Sistematika Penulisan.....	4
BAB II Tinjauan Pustaka dan Dasar Teori	6
2.1 Tinjauan Pustaka	6
2.2 Dasar Teori	7
2.2.1 Sistem <i>Monitoring</i> Konsumsi Listrik	7
2.2.2 IC Renergy 8209	12
2.2.3 Pengembangan <i>Firmware</i> Pada Platform ESP32	23
2.3 Analisis Perbandingan Metode	29
BAB III Metode Penelitian.....	30
3.1 Alat dan Bahan Tugas Akhir	30
3.1.1 Alat Tugas akhir.....	30
3.1.2 Bahan Tugas Akhir.....	30
3.2 Metode yang Digunakan.....	31
3.2.1 Pengembangan <i>Firmware</i> Kalibrasi Pengujian	31
3.2.2 Kalibrasi IC Renergy 8209	34
3.2.3 Pengujian Prototipe Sistem	40
3.2.4 Pengembangan <i>Firmware</i> Final	42
3.3 Alur Tugas Akhir	44
3.4 Keterbatasan Penelitian	47
BAB IV Hasil dan Pembahasan.....	49

4.1	Pembahasan Hasil <i>Firmware</i> Kalibrasi dan Pengujian	49
4.1.1	Fitur Komunikasi UART dengan IC Renergy 8209	51
4.1.2	Fitur Membaca Nilai Register	54
4.1.3	Fitur Memberi Nilai Register	56
4.1.4	Fitur Inisialisasi Konfigurasi.....	58
4.1.5	Fitur Tampilan OLED	59
4.2	Pembahasan Hasil Kalibrasi IC Renergy 8209	60
4.2.1	Konfigurasi Sistem	60
4.2.2	Kalibrasi Daya Aktif	64
4.2.3	Kalibrasi Daya Reaktif.....	66
4.2.4	Kalibrasi RMS	67
4.3	Pembahasan Hasil Pengujian Prototipe Sistem.....	70
4.3.1	Hasil Pengujian Tanpa Beban	70
4.3.2	Hasil Pengujian Arus Mula	70
4.3.3	Hasil Pengujian Variasi Arus	71
4.3.4	Hasil Pengujian Register.....	80
4.4	Pembahasan Hasil Pengembangan <i>Firmware</i> final	82
4.4.1	Fitur yang Diterapkan Kembali	82
4.4.2	Fitur Tombol Interupsi	82
4.4.3	Fitur Penyimpanan Data Pada EEPROM	85
4.4.4	Fitur Tampilan OLED Final	88
4.5	Perbandingan Hasil Penelitian dengan Hasil Terdahulu	92
BAB V	Kesimpulan dan Saran	95
5.1	Kesimpulan	95
5.2	Saran	97
	DAFTAR PUSTAKA	98

DAFTAR TABEL

Tabel 2.1	Jenis arus.....	11
Tabel 2.2	Tabel BKD	11
Tabel 3.1	Konfigurasi pin kalibrasi dan pengujian.....	33
Tabel 3.2	Konfigurasi pin OLED-ESP32.....	34
Tabel 4.1	Inisialisasi konfigurasi	58
Tabel 4.2	Alamat register kalibrasi dan konstantanya	69
Tabel 4.3	Rangkuman pengujian variasi arus	80
Tabel 4.4	Inisialisasi konfigurasi final	84

DAFTAR GAMBAR

Gambar 1.1	Grafik konsumsi listrik di Indonesia	1
Gambar 2.1	Meter kWh PLN	8
Gambar 2.2	Energi meter pabrikan Acrel	9
Gambar 2.3	Segitiga daya	9
Gambar 2.4	Rangkaian aplikasi IC Renergy 8209	13
Gambar 2.5	Diagram blok IC Renergy 8209	14
Gambar 2.6	Diagram blok kalkulasi daya aktif	14
Gambar 2.7	Diagram blok kalkulasi daya reaktif	15
Gambar 2.8	Diagram blok kalkulasi energi	16
Gambar 2.9	Diagram blok kalkulasi tegangan dan arus RMS	16
Gambar 2.10	Diagram blok kalibrasi IC Renergy 8209	17
Gambar 2.11	Diagram blok konfigurasi sistem	17
Gambar 2.12	Diagram blok kalibrasi daya aktif	19
Gambar 2.13	Diagram blok kalibrasi daya reaktif	20
Gambar 2.14	Diagram blok kalibrasi RMS	21
Gambar 2.15	Paket <i>byte</i> komunikasi IC Renergy 8209	22
Gambar 2.16	Urutan paket memberi nilai register	22
Gambar 2.17	Urutan paket membaca nilai register	22
Gambar 2.18	Format bit UART	25
Gambar 2.19	Tampilan papan ESP32	26
Gambar 2.20	Konfigurasi pin ESP32	27
Gambar 2.21	Tampilan awal Arduino IDE	28
Gambar 3.1	<i>Wiring diagram</i> untuk kalibrasi dan pengujian	33
Gambar 3.2	Diagram alir kalibrasi	35
Gambar 3.3	Energi meter kalibrator pabrikan GFUVE	38
Gambar 3.4	<i>Testbench</i> pada laboratorium	39
Gambar 3.5	Diagram blok <i>testbench</i>	39
Gambar 3.6	Diagram alir pengujian	40
Gambar 3.7	<i>Wiring diagram</i> IC Renergy 8209 final	44
Gambar 3.8	Diagram alir penelitian	45
Gambar 4.1	Diagram alir <i>firmware</i> kalibrasi dan pengujian	49
Gambar 4.2	Tata cara pengiriman data dengan UART	52
Gambar 4.3	Tata cara menerima data dengan UART	52
Gambar 4.4	Masalah pada waktu tunggu kurang dari 10 ms	53
Gambar 4.5	Tampilan OLED kalibrasi dan pengujian	59
Gambar 4.6	<i>Error</i> kombinasi gain	61
Gambar 4.7	Distribusi kesalahan pembacaan	62
Gambar 4.8	Mencari konstanta pembagi	68
Gambar 4.9	Pengujian tanpa beban	70
Gambar 4.10	Pengujian arus mula	71
Gambar 4.11	Pengaturan Variac 2	71
Gambar 4.12	Pengujian arus minimal	73
Gambar 4.13	Pengujian arus transisi faktor daya 1	74

Gambar 4.14 Pengujian arus transisi faktor daya 0.5.....	75
Gambar 4.15 Pengujian arus dasar faktor daya 1	76
Gambar 4.16 Pengujian arus dasar faktor daya 0.5.....	77
Gambar 4.17 Pengujian arus maksimal faktor daya 1	78
Gambar 4.18 Pengujian arus maksimal faktor daya 0.5	79
Gambar 4.19 Pengujian register	81
Gambar 4.20 Ilustrasi penyimpanan EEPROM.....	87
Gambar 4.21 Diagram alir OLED final	88
Gambar 4.22 Tampilan OLED pemakaian listrik.....	89
Gambar 4.23 Tampilan OLED estimasi tarif	89
Gambar 4.24 Tampilan OLED daya aktif	90
Gambar 4.25 Tampilan OLED tegangan RMS	90
Gambar 4.26 Tampilan OLED arus RMS	91
Gambar 4.27 Tampilan OLED faktor daya	91
Gambar 4.28 Prototipe sistem <i>monitoring</i> konsumsi listrik.....	92

DAFTAR SINGKATAN

IC	= Integrated Circuit (sirkuit terpadu)
MCU	= Microcontroller Unit
RMS	= Root Mean Square
UART	= Universal Asynchronous Receiver Transmitter
OLED	= Organic Light-Emitting Diode
EEPROM	= Electrically Erasable Programmable Read-Only Memory
EC	= Energy Constant

INTISARI

Permintaan energi listrik dari tahun ke tahun selalu meningkat. Peningkatan tersebut juga dibarengi dengan tarif listrik yang semakin mahal. Mengatasi kondisi tersebut, diperlukan suatu upaya untuk dapat melakukan manajemen penggunaan energi listrik. Informasi dasar yang perlu diketahui dalam melakukan upaya tersebut adalah jumlah konsumsi energi listrik dan hal tersebut dapat diketahui dengan menggunakan suatu perangkat *monitoring* konsumsi listrik yang disebut energi meter.

Penelitian ini bertujuan untuk mengembangkan sistem *monitoring* konsumsi listrik menggunakan IC Renergy 8209 pada platform ESP32, khususnya pada sisi *firmware*. IC tersebut dipilih karena harganya yang relatif murah. Tujuan lain adalah untuk menguji sistem tersebut berdasarkan standar nasional yang diatur dalam SK Ditjen Perlindungan Konsumen dan Tata Niaga Nomor 16 Tahun 2019. Dalam mengembangkannya, terdapat empat tahap yang harus dilalui yaitu pengembangan *firmware* kalibrasi dan pengujian, kalibrasi IC, pengujian sistem, dan pengembangan *firmware* final.

Hasil penelitian menunjukkan tata cara pengembangan *firmware* sistem *monitoring* konsumsi listrik dan fitur-fitur yang harus terdapat didalamnya untuk menampilkan parameter-parameter energi listrik. Kalibrasi membuat IC dapat bekerja pada suatu karakteristik tertentu dan meminimalisir kesalahan pembacaan. Terakhir, berdasarkan pengujian yang dilakukan, dengan mengacu pada standar nasional, sistem ini terbukti secara fungsionalitas bekerja dengan tingkat akurasi pembacaan energi sebesar 0.16%.

Kata kunci : Pengembangan firmware, IC Renergy 8209, sistem *monitoring* listrik, manajemen energi

ABSTRACT

From time to time, electricity demand keeps increasing. This phenomenon is accompanied by rising electricity bills. In overcoming this situation, energy management plays a crucial role. The basic information to do energy management is the amount of electricity consumption and that information can be known by using an energy meter.

This research has the objective to develop an electricity consumption system using IC Renergy 8209 on the ESP32 platform, especially on the firmware side. This IC is chosen for its cheap price compared to other IC on the market. Another objective is to test the system based on the national standard which has been regulated on SK Ditjen Perlindungan Konsumen dan Tata Niaga No.16 Tahun 2019. There are four steps to develop this system which are developing firmware for calibration and testing, IC calibration, system testing, and developing final firmware for users.

This research shows how to develop firmware for an energy monitoring system using IC Renergy 8209 on the ESP32 platform and the features that must be included to display the electricity parameters. Calibration makes the IC respond with the desired characteristics and minimized error readings. Lastly, based on the testing, this system is functionality proven with 0.16% active energy readings accuracy.

Keywords : firmware development, IC Renergy 8209, energy monitoring system, energy management

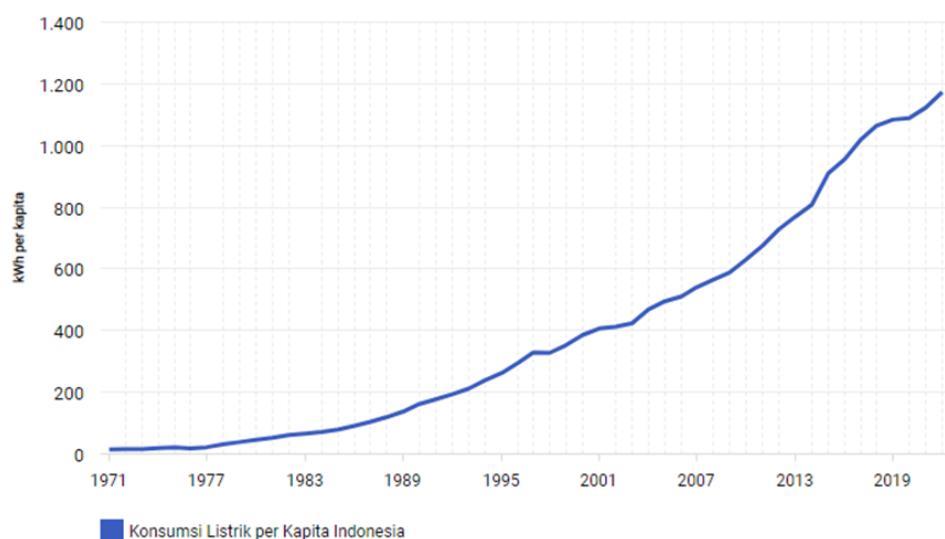
BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam melakukan berbagai kegiatan atau usaha, makhluk hidup membutuhkan energi. Energi tersebut memiliki berbagai macam bentuk seperti energi panas, energi mekanik, hingga energi listrik. Hukum kekekalan energi menyatakan bahwa energi tidak dapat diciptakan maupun dimusnahkan namun dapat berubah bentuknya. Contoh aplikasi dari hukum tersebut adalah ketika ingin menggunakan kipas angin di rumah. Energi listrik yang disuplai oleh PLN kemudian masuk ke rangkaian listrik kipas angin dan diubah bentuknya menjadi energi mekanik dan energi panas.

Diantara bentuk energi yang ada, energi listrik memainkan peran penting pada penggunaannya sehari-hari. Pemilihan energi listrik didasari karena sifatnya yang mudah diubah bentuknya dan dipindahkan dari lokasi satu ke lainnya. Selain itu energi listrik juga dapat disimpan dalam bentuk baterai. Keuntungan yang ditawarkan oleh energi listrik membuat mayoritas peralatan yang digunakan pada kehidupan sehari-hari menggunakan energi listrik sebagai sumber energi [1].



Gambar 1.1. Grafik konsumsi listrik di Indonesia

Di Indonesia sendiri, permintaan energi listrik setiap tahunnya terus bertambah seperti pada Gambar 1.1 [2]. Tercatat pada 2022, konsumsi energi listrik perkapita di Indonesia mencapai 1.1732 kWh/kapita atau nilai tersebut naik sekitar 4% jika dibandingkan pada tahun 2021. Kenaikan konsumsi energi listrik tersebut diakibatkan oleh pertumbuhan penduduk Indonesia yang terus bertambah sehingga pemakaian peralatan yang menggunakan energi listrik sebagai sumbernya pun bertambah.

Peraturan Menteri Energi dan Sumber Daya Mineral (ESDM) Nomor 28 Tahun 2016 mengatur tentang besaran tarif energi listrik yang disesuaikan dengan kategori dan kapasitas pengguna. Sebagai contoh, untuk kategori terkecil yaitu rumah tangga dengan kapasitas 450 VA tarif listriknya adalah Rp 415/kWh. Sementara untuk kategori rumah tangga dengan kapasitas 3500 VA ke atas, tarif listriknya adalah Rp 1699,53/kWh. Perbedaan tarif tersebut dikarenakan untuk kapasitas 450 VA dan 900 VA tarifnya disubsidi oleh pemerintah [3]. Tarif listrik tersebut dari tahun ke tahun akan selalu bertambah karena bahan baku pembangkitan energi listrik yang nilainya juga semakin bertambah.

Dalam menyikapi pertambahan konsumsi listrik dan tarif listrik, diperlukan manajemen penggunaan energi listrik. Dalam melakukan manajemen tersebut, informasi dasar yang harus diketahui adalah seberapa banyak energi listrik yang sudah digunakan. Langkah paling mudahnya untuk mengetahui informasi tersebut adalah dengan memperhatikan kWh meter atau energi meter yang sudah disediakan oleh Perusahaan Listrik Negara (PLN). Cara kerja dari alat tersebut adalah pengguna mendepositkan pulsa kWh yang akan berkurang jika pengguna menggunakan energi listrik.

Perkembangan teknologi memungkinkan manajemen konsumsi energi listrik dilakukan dengan lebih baik lagi yaitu dengan menggunakan energi meter tambahan. Perangkat tersebut dapat melakukan *monitoring* konsumsi listrik dan dapat membatasi pemakaian dengan batas tertentu sebelum tagihan membengkak [4]. Selain itu, menempatkan energi tambahan dapat berguna apabila terjadi konsumsi energi listrik yang tidak diinginkan, pengguna dapat mengetahui lokasi dari masalah tersebut. Ukuran dari energi meter tersebut juga semakin ringkas karena cara perhitungan yang sebelumnya secara elektromekanikal, saat ini telah menjadi digital dengan menggunakan *integrated circuit* (IC) tertentu.

Hal tersebut yang melatarbelakangi penulis dalam menulis skripsi dengan judul Pengembangan *Firmware* Sistem *Monitoring* Konsumsi Listrik Menggunakan IC Renergy 8209 pada Platform ESP32. Penelitian ini secara garis besar akan membahas tentang pengembangan sistem *monitoring* konsumsi listrik khususnya pada sisi *firmware*. IC yang digunakan adalah IC Renergy 8209 yang harganya relatif murah. Kemudian akan dibahas juga mengenai kalibrasi dan pengujian yang harus dilakukan agar menghasilkan pembacaan yang baik menurut standar tertentu. Harapannya penelitian ini dapat mendorong pabrikan dalam negeri untuk mengembangkan perangkat energi meter.

1.2 Rumusan Masalah

Dari latar belakang tersebut, maka rumusan masalah dari penelitian ini adalah sebagai berikut:

1. Dibutuhkan *firmware* dan tahapan pengembangan sistem agar *development board* dapat berkomunikasi dengan IC Renergy 8209 untuk melakukan fungsinya dengan tepat dan akurat.
2. Diperlukan kepastian bahwa prototipe sistem telah memenuhi persyaratan dari suatu acuan tertentu.

1.3 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dari penelitian ini adalah sebagai berikut:

1. Membuat *firmware* dan tahapan pengembangan sistem *monitoring* konsumsi listrik menggunakan metering IC Renergy 8209 pada platform ESP32.
2. Menentukan fitur-fitur minimal yang harus ada dalam *firmware* untuk dapat menampilkan parameter-parameter energi listrik.
3. Mengetahui mekanisme dan melakukan kalibrasi IC Renergy 8209 agar dapat bekerja dengan karakteristik tertentu dan mengurangi kesalahan pembacaan.
4. Menguji sistem *monitoring* konsumsi listrik menggunakan IC Renergy 8209 berdasarkan standar nasional dan mengevaluasi tingkat akurasi pembacaan energi sistem tersebut.

1.4 Batasan Penelitian

Hal-hal yang menjadi batasan dari penelitian ini adalah sebagai berikut:

1. Objek penelitian: Studi pengembangan *firmware* sistem *monitoring* konsumsi energi listrik menggunakan IC Renergy 8209 pada platform ESP32.
2. Metode penelitian: Penelitian pengembangan *firmware* IC Renergy 8209 pada platform ESP32, kalibrasi IC Renergy 8209, dan pengujian sistem berdasarkan standar.
3. Waktu dan tempat penelitian: Waktu penelitian adalah Februari – Juni 2023 di Laboratorium Teknik Tenaga Listrik DTETI FT UGM.
4. Populasi dan sampel: Populasi adalah pembacaan energi dari sistem *monitoring* konsumsi listrik dan sampel adalah pembacaan energi listrik menggunakan IC Renergy 8209.
5. Variabel: Variabel bebas adalah pengembangan *firmware* dan variabel terikat adalah platform ESP32.
6. Hipotesis: Bahwa pengembangan *firmware* sistem monitoring konsumsi listrik menggunakan IC Renergy 8209 pada platform ESP32 dapat memenuhi standar yang berlaku secara nasional.

7. Keterbatasan Penelitian: Keterbatasan penelitian ini adalah kalibrasi dilakukan tanpa menggunakan alat untuk kalibrasi sesungguhnya, dinamika catu daya dan beban, ketelitian dari osiloskop, dan dinamika nilai faktor daya.

1.5 Manfaat Penelitian

Penelitian ini harapannya dapat memberikan manfaat sebagai berikut:

1. Memberikan pemahaman terhadap pengembangan *firmware* dan tahapan pengembangan sistem *monitoring* konsumsi listrik menggunakan IC Renergy 8209 pada platform ESP32.
2. Prototipe sistem *monitoring* konsumsi listrik yang menggunakan IC Renergy 8209 pada platform ESP32 memenuhi standar yang berlaku secara nasional dan dapat dijadikan sebagai acuan untuk fabrikasi.

1.6 Sistematika Penulisan

Dalam menyusun penulisan skripsi ini terdapat pendoman dan struktur yang telah direkomendasikan dari Departemen Teknik Elektro dan Teknologi Informasi Fakultas Teknik Universitas Gadjah Mada. Rekomendasi tersebut penulis terapkan agar penelitian yang dilaksanakan dapat tersampaikan dengan jelas. Selain itu sistematika penulisan yang akan dijabarkan selanjutnya merupakan kerangka kerja yang penulis terapkan dalam melakukan penelitian. Berikut sistematika penulisan yang dilakukan:

BAB I Pendahuluan

Bab pendahuluan merupakan awalan dari penelitian yang dilakukan. Bagian ini berisi latar belakang serta motivasi yang mendasari penelitian, rumusan masalah yang akan diteliti, tujuan yang akan dicapai pada penelitian, batasan dari penelitian, manfaat yang diperoleh dari penelitian, dan sistematika penulisan skripsi.

BAB II Tinjauan Pustaka dan Landasan Teori

Landasan teori memiliki isi rangkuman studi literatur yang terkait pada skripsi ini. Bagian ini diawali dengan menelaah penelitian serupa yang telah dilakukan oleh orang lain. Penelitian tersebut dianalisis dan dijadikan sebagai bahan evaluasi dalam melakukan penelitian pada skripsi ini. Setelah itu, dari judul yang dilakukan pada penelitian ini, dijabarkan lebih lanjut dan dijadikan sebagai kerangka dalam menjelaskan dasar teori.

BAB III Metodologi Penelitian

Bagian ini membahas metodologi yang akan dilakukan pada penelitian skripsi ini. Metodologi penelitian diawali dengan menyebutkan alat dan bahan yang digunakan selama proses penelitian. Setelah itu dijabarkan metode yang digunakan dalam melakukan penelitian serta mekanisme yang akan dilakukan. Selanjutnya alur dari

penelitian dijabarkan lebih jauh pada bab ini. Terakhir, keterbatasan dari penelitian akan lebih dirinci alasannya, dampaknya, dan pendekatan yang dilakukan untuk mengatasi hal tersebut.

BAB IV Hasil dan Pembahasan

Pada setiap metode yang digunakan, akan dibahas hasilnya pada bagian ini. Selain itu hasil pengujian dan analisisnya dijelaskan lebih lanjut. Hasil analisis kemudian akan dibandingkan dengan hasil penelitian serupa yang telah dilakukan sebelumnya.

BAB V Kesimpulan dan Saran

Bagian ini merupakan penutup dari penelitian skripsi yang dilakukan. Kesimpulan berisi jawaban dari rumusan masalah dan tujuan yang telah disebutkan pada bab pendahuluan. Selain itu, kesimpulan juga berisi penemuan lainnya yang menurut penulis cukup signifikan dalam penelitian ini. Terakhir adalah saran yang menurut penulis perlu disampaikan untuk penelitian ke depannya.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Pengembangan sistem *monitoring* konsumsi listrik dapat dilakukan dengan berbagai pendekatan dan tentu apa yang dilakukan pada skripsi ini bukanlah yang pertama kali. Untuk itu penulis akan meninjau beberapa penelitian sejenis yang telah dilakukan sebelumnya. Hal ini untuk melihat potensi pengembangan dan antisipasi masalah yang pernah dihadapi sebelumnya untuk diterapkan pada skripsi ini.

Penelitian yang dilakukan Guimaraes dan kawan-kawan dengan judul “*Smart Energy Monitoring System with ADE7758 IC*” meneliti tentang pengembangan sistem *monitoring* energi pintar dengan IC ADE7758 yang merupakan pabrikan dari Analog Device. Penelitian ini menggunakan arduino sebagai *development board* dan protokol komunikasi SPI. IC ADE7758 dirangkai pada *printed circuit board* yang juga bertindak sebagai *shield* dan dirancang agar bisa ditumpuk langsung diatas arduino. Untuk mendeteksi arus digunakan *current transformer* (CT) dan deteksi tegangan menggunakan transformator tegangan 220V/5V sederhana. Tampilan yang digunakan adalah LCD yang dapat menampilkan informasi konsumsi energi listrik. Pengujian dilakukan selama beberapa jam dengan daya aktif sebesar 600 W dan referensi yang digunakan adalah energi meter Elster A1052. Dari pengujian ini didapatkan kesalahan pembacaan energi (*error*) sebesar 1.183% dan berdasarkan standar yang diterapkan di negara penulis tersebut, yaitu Brazil, nilai kesalahan pembacaan tersebut masih bisa diterima [5].

Mejbaul Haque dan kawan-kawan juga melakukan penelitian serupa dengan judul “*Microcontroller Based Single Phase Digital Prepaid Energy Meter for Improved Metering and Billing System*”. Penelitian tersebut menggunakan mikrokontroller (MCU) Atmega32 dan IC AD7751 pabrikan dari Analog Devices yang dikembangkan menjadi perangkat energi meter prabayar. Deteksi arus menggunakan resistor shunt bernilai 3.335 miliohm dan deteksi tegangan menggunakan pembagi tegangan dengan kombinasi 1 megaohm dan 1 kiloohm resistor yang keduanya akan terhubung ke pin dari IC AD7751. MCU berfungsi untuk berkomunikasi dengan menghitung pulsa energi, kontrol *relay*, dan menampilkan informasi sisa pulsa energi yang dimiliki pada LCD. Sistem ini menggunakan *smart card* untuk mengisi pulsa energi ke energi meter. Apabila pulsa energi telah habis maka *relay* akan terbuka dan listrik tidak bisa mengalir. *Firmware* dari alat ini dikembangkan dengan menggunakan perangkat lunak Win-AVR dan bahasa pemrograman C. Pengujian dilakukan dengan menggunakan beban pemanas elektrik 1200 watt dan diuji selama 120 detik dan dicatat pembacaannya setiap 20 detik.

Hasil pengamatan kemudian dibandingkan dengan perhitungan menggunakan formula dan didapatkan *error* pembacaan pada detik ke 120 sebesar -0.02%. Penelitian ini tidak menjelaskan standar yang digunakan dalam pengembangan sistem *monitoring* konsumsi listrik. Menurut penulis, penggunaan hitungan formula sebagai referensi kurang tepat dilakukan karena berdasarkan keterangan dari penelitian tersebut, tegangan yang diberikan bervariasi antara 210 V hingga 230 V dan sehingga deviasi daya sangat mungkin terjadi namun tidak dipertimbangkan [6].

Penelitian selanjutnya berjudul “*Monitoring Daya Listrik Berbasis IoT*” yang dilakukan oleh Agus Ardiansyah. Berbeda jika dibandingkan dengan dua penelitian sebelumnya yang menggunakan sebuah IC dengan fungsi melakukan perhitungan energi, penelitian ini hanya menggunakan sensor arus dan tegangan yang kemudian perhitungan dilakukan secara komputasi di *development board* yaitu arduino. Sensor arus dan tegangan yang digunakan adalah ACS712 dan ZMPT101B. Untuk mendukung fungsi *internet of things* (IoT), NodeMCU disusun secara seri dengan arduino. NodeMCU menerima data yang ditransmisikan dari arduino untuk kemudian dikirimkan ke *cloud*. Hal tersebut dilakukan karena arduino belum memiliki fitur yang memungkinkan untuk melakukan komunikasi nirkabel seperti Wi-Fi. Pengujian yang dilakukan adalah pengujian sensor tegangan, sensor arus, dan server. Dari pengujian diketahui kesalahan pembacaan sensor tegangan 0.3%, kesalahan pembacaan sensor arus 0.28%, dan waktu tunda pengiriman data ke server 12 detik. Pada penelitian ini, energi listrik tidak diuji padahal variabel tersebut merupakan salah satu yang tertampil pada sisi antarmuka pengguna. Selain itu tidak dijelaskan pengembangan alat tersebut mengacu pada suatu standar tertentu [7].

2.2 Dasar Teori

Bagian ini akan mengulas lebih dalam terkait teori-teori dari variabel yang dikembangkan pada skripsi. Setidaknya terdapat tiga variabel yang akan diulas yaitu sistem *monitoring* konsumsi listrik, IC Renergy 8209, dan pengembangan *firmware* pada platform ESP32. Ketiga variabel tersebut saling berkesinambungan dalam pengembangan yang dilakukan pada skripsi ini.

2.2.1 Sistem *Monitoring* Konsumsi Listrik

Setiap rumah yang menggunakan listrik dari Perusahaan Listrik Negara (PLN) sebagai sumber energi, pasti memiliki sistem *monitoring* konsumsi listrik untuk memantau penggunaan listrik yang kemudian dikonversikan menjadi berapa biaya yang harus dibayar atas penggunaan listrik tersebut. Perangkat yang digunakan pada sistem *monitoring* konsumsi listrik biasa disebut sebagai energi meter atau kWh meter. Secara garis besar, energi meter memiliki pengertian yaitu perangkat untuk mengukur energi

listrik yang berjalan secara non stop dan dapat menyimpan total energi listrik yang terpakai [8].



Gambar 2.1. Meter kWh PLN

Saat ini energi meter yang digunakan oleh PLN telah berganti menjadi digital dari yang sebelumnya analog seperti pada Gambar 2.1 [9]. Berdasarkan penelitian yang dilakukan oleh Enisyah Ie dan kawan-kawan, energi meter digital memiliki keunggulan pada konsumsi daya perangkat yang lebih rendah dan akurasi yang lebih baik daripada energi meter analog. Selain itu prinsip perhitungan konsumsi listrik antara analog dan digital juga berbeda. Sistem analog menggunakan prinsip *electromechanical* yaitu medan listrik yang terjadi akan menghasilkan medan magnet untuk memutar sebuah piringan yang merepresentasikan pertambahan penggunaan energi listrik. Sementara sistem digital menggunakan sebuah IC yang berfungsi sebagai pengukur energi listrik dengan cara mengeluarkan pulsa sebagai penanda pertambahan energi listrik dan mikrokontroller untuk berkomunikasi dengan IC tersebut serta keperluan lainnya [10].

Hadirnya energi meter digital memungkinkan pengembangan perangkat ini menjadi lebih kecil ukurannya. Hal ini karena komponen yang digunakan berukuran lebih kecil daripada komponen yang digunakan pada energi meter analog. Selain itu penggunaan MCU pada energi meter jenis ini juga membuka kemungkinan untuk *monitoring* secara jarak jauh. Contohnya adalah energi meter pabrikan dari Acrel seperti pada Gambar 2.2 [11] yang dapat melakukan *monitoring* konsumsi listrik satu fase dan ukurannya hanya 2 x 9 cm.

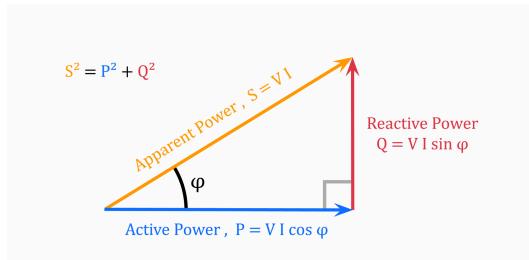
1. Daya dan energi listrik

Listrik memiliki dua komponen utama yaitu tegangan dan arus. Tegangan



Gambar 2.2. Energi meter pabrikan Acrel

adalah beda potensial atau muatan antara dua titik dan arus adalah banyaknya muatan yang mengalir. Perpaduan keduanya disebut energi listrik. Sementara daya listrik adalah besarnya energi listrik yang terukur dalam waktu sesaat [12]. Memahami daya listrik dapat dilakukan dengan melihat segitiga daya, seperti pada Gambar 2.3 [13]. Terdapat tiga jenis daya yang masing-masing dihasilkan oleh beban tertentu.



Gambar 2.3. Segitiga daya

Daya aktif adalah daya yang dihasilkan dari beban resistif dengan satuan W atau watt. Daya ini yang dihitung energinya pada energi meter yang biasa terdapat pada rumah tangga. Formula mencarinya adalah:

$$P = V I \cos(\theta) \quad (2-1)$$

Dengan V adalah tegangan, I adalah arus, dan cos theta merupakan faktor daya.

Kemudian ada daya reaktif yang dihasilkan dari beban reaktif (induktif atau kapasitif) dengan satuan VAR. Daya ini biasa dihasilkan dari perangkat yang memiliki induktor atau kapasitor seperti motor listrik dan lain-lain. Formula

mencarinya adalah:

$$Q = VI \sin(\theta) \quad (2-2)$$

Sisi miring dari segitiga daya adalah daya semu atau *apparent power* dengan satuan VA. Daya semu adalah gabungan dari daya aktif dan reaktif yang mengalir dari sumber ke beban. Daya semu dapat dicari dengan formula berikut :

$$S = VI \quad (2-3)$$

atau

$$S = \sqrt{P^2 + Q^2} \quad (2-4)$$

Seperti yang disebutkan sebelumnya, daya listrik diukur sesaat sementara energi listrik diukur dalam jangka waktu tertentu. Misalnya energi aktif maka perhitungannya adalah total daya listrik dikali lama waktu yang diamati. Pada energi aktif biasanya satuanya adalah kWh, energi reaktif adalah kVARh, dan energi semu adalah kVAh. Energi listrik inilah yang nantinya akan dihitung oleh sistem *monitoring* konsumsi listrik.

2. Standar pada energi meter

Pada pengembangan energi meter terdapat standar yang perlu dipenuhi agar bisa dijual secara komersial. Standar ini menentukan batas maksimal kesalahan pembacaan dari energi meter dan pengklasifikasianya. Ada beberapa standar yang mengatur tentang energi meter baik yang berlaku secara internasional maupun nasional. Untuk yang berlaku secara internasional, lembaga yang mengeluarkannya adalah *International Electrotechnical Commission* (IEC) dengan nomor seri IEC 62053 yang mengatur tentang energi meter statis [14]. Sementara di Indonesia juga memiliki standar untuk energi meter yang mengacu pada Surat Keputusan Direktur Jenderal Perlindungan Konsumen dan Tertib Niaga Nomor 161 Tahun 2019 tentang Syarat Teknis Meter kWh. Pada penelitian ini standar nasional akan diuraikan lebih lanjut karena yang nantinya akan digunakan dipenelitian ini adalah standar tersebut.

SK Ditjen Perlindungan Konsumen dan Tertib Niaga Nomor 161 Tahun 2019 menjadi pedoman dalam melakukan pengujian energi meter yang berlaku di Indonesia. Sebelum masuk ke pengujian, ada beberapa kondisi yang perlu diperhatikan.

Tabel 2.1. Jenis arus

Jenis arus	Pengertian	Nilai
I_{st} (arus mula)	Nilai arus terkecil untuk energi meter mulai menghitung energi	$0.002 \times Id$
I_{min} (arus minimum)	Nilai arus terkecil untuk memenuhi syarat keakurasan	$0.05 \times Id$
I_{tr} (arus transisi)	Nilai arus yang harus berada dalam batas kesalahan maksimum	$0.1 \times Id$

- Frekuensi : toleransi frekuensi dasar (f_n) adalah $\pm 2\%$ dengan nilai f_n ditentukan oleh pabrikan.
- Tegangan : toleransi tegangan dasar (U_n) adalah 10% dengan nilai U_n ditentukan oleh pabrikan.
- Arus : nilai arus dasar (I_d) dan arus maksimal (I_{max}) ditentukan oleh pabrikan dengan pengaturan parameter arus untuk kelas 1 dan 2 tertera pada tabel 2.1.
- Suhu : rentang suhu operasional adalah -10 sampai 50 derajat celcius.

Kemudian untuk persyaratan keakurasan, surat keputusan ini telah menetapkan Batas Kesalahan yang Diizinkan (BKD) pada pembacaan energi untuk kelas 1 dan 2 seperti pada tabel 2.2.

Tabel 2.2. Tabel BKD

Besaran		Arus	
Arus	Faktor daya	1 (B)	2 (A)
$I_{tr} \leq I \leq I_{max}$	1	$\pm 1\%$	$\pm 2\%$
	0.5 induktif - 1	$\pm 1.5\%$	$\pm 2.5\%$
$I_{min} \leq I \leq I_{tr}$	1	$\pm 1.5\%$	$\pm 2.5\%$
	0.5 induktif - 1	$\pm 1.8\%$	$\pm 2.5\%$
$I_{st} \leq I \leq I_{min}$	1	$\pm 1.5\%$	$\pm 2.5\%$

Terdapat empat pengujian yang harus dilakukan untuk setiap energi meter yaitu:

- Pengujian tanpa beban untuk melihat kinerja pada kondisi tanpa beban dan pada kondisi ini perangkat diharuskan untuk tidak

mengukur energi sama sekali.

- Pengujian arus mula (I_{st}) yang bertujuan untuk melihat pembacaan energi meter pada nilai arus terkecil.
- Pengujian variasi arus yang bertujuan untuk mengetahui kesalahan ukur pada variasi arus.
- Pengujian register yang bertujuan untuk mengetahui keakurasian dari perhitungan energi dari energi meter.

Ketika semua pengujian telah dilakukan oleh penguji maka energi meter mendapatkan Surat Keterangan Hasil Pengujian yang didalamnya terdapat keterangan lolos atau tidaknya pengujian tersebut [8].

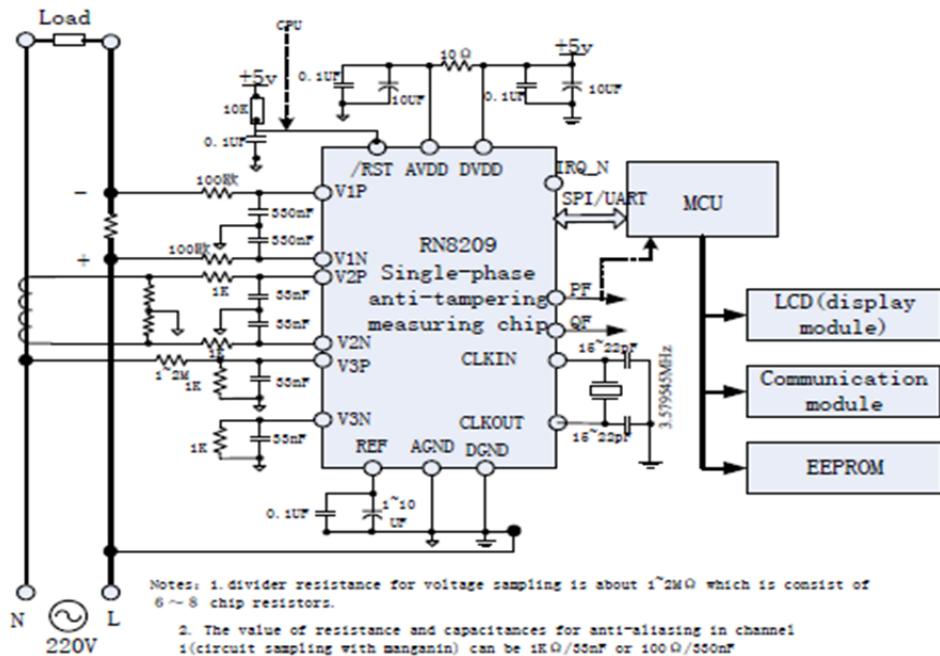
2.2.2 IC Renergy 8209

Bagian ini menjabarkan lebih lanjut tentang IC Renergy 8209 berdasarkan *datasheet* yang telah disediakan oleh pabrikan Renergy [15].

1. Tentang IC Renergy 8209

IC Renergy 8209 merupakan *integrated circuit* pabrikan dari Shenzhen RENERGY Micro-Technology yang memiliki fungsi sebagai penghitung energi aktif dan reaktif. Selain itu IC ini juga dapat melakukan pembacaan tegangan RMS, arus RMS, daya aktif, dan daya reaktif. IC ini dapat memberikan akurasi pembacaan energi aktif, energi reaktif, tegangan RMS (*root mean square*), dan arus RMS sampai dengan 0.1%. Nilai-nilai tersebut dapat diakses dengan memanggil register yang tersedia pada IC sesuai dengan alamatnya masing-masing dan menggunakan protokol komunikasi UART melalui pin Tx dan Rx.

Gambar 2.4 [15] merupakan rangkaian aplikasi dari IC Renergy 8209 untuk bisa beroperasi secara optimal, sesuai yang disarankan oleh pabrikan IC. Terdapat tiga masukan yang diwakili oleh tiga pasang pin yaitu V1P, V1N, V2P, V2N, V3P, dan V3N. Masing-masing pin tersebut mewakili masukan yang berbeda. V1P dan V1N digunakan untuk masukan arus *channel A* yang diambil dari jalur beban secara seri dan ditempatkan resistor shunt dengan fungsi untuk menghasilkan sedikit jatuh tegangan agar dapat dilakukan pembacaan arus. V2P dan V2N digunakan untuk masukan arus *channel B* yang diambil dengan menggunakan *current transformer*, namun pada aplikasi kali ini hanya digunakan resistor shunt. V3P dan V3N digunakan untuk masukan tegangan yang diambil dari jalur beban secara paralel dan terdapat rangkaian pembagi tegangan untuk memperkecil tegangan yang masuk ke IC agar sesuai dengan jangkauan kerja dari IC.



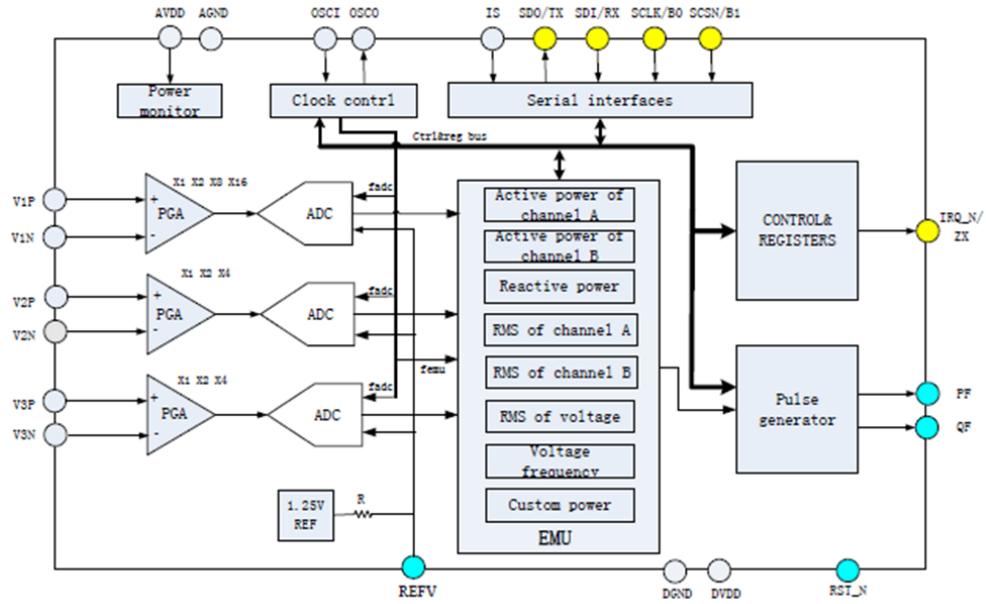
Gambar 2.4. Rangkaian aplikasi IC Renergy 8209

IC Renergy 8209 memiliki tiga keluaran yaitu pin UART TX/RX, PF, dan QF. Pin UART digunakan sebagai pin komunikasi antara mikrokontroller (MCU) dan IC dengan *baud rate* bersifat tetap yaitu sebesar 4800 bps. Pin PF adalah keluaran untuk perhitungan energi aktif menggunakan pulsa atau kedipan listrik yang biasanya digunakan LED. Sementara pin QF sama seperti PF hanya saja pin ini digunakan untuk perhitungan energi reaktif.

Gambar 2.5 [15] adalah blok diagram dari IC Renergy 8209. Ketiga masukkan kemudian nilainya akan diperbesar oleh *operational amplifier* (OpAmp) yang direpresentasikan dengan blok PGA (*Programmable Gain Amplifier*) dan nilai *gain* dari OpAmp dapat dikonfigurasikan melalui register terkait. Setelah itu isyarat analog akan diubah menjadi digital melalui blok ADC (*Analog to Digital Converter*) untuk masuk ke blok EMU atau *Energy Measurement Unit* untuk dilakukan perhitungan agar dapat menghasilkan keluaran energi aktif dan reaktif, proses ini akan dijelaskan pada bagian selanjutnya. EMU akan menghasilkan nilai tegangan, arus, daya aktif, daya reaktif, energi aktif, dan energi reaktif yang kemudian nilai tersebut disimpan pada register terkait.

2. Proses kalkulasi

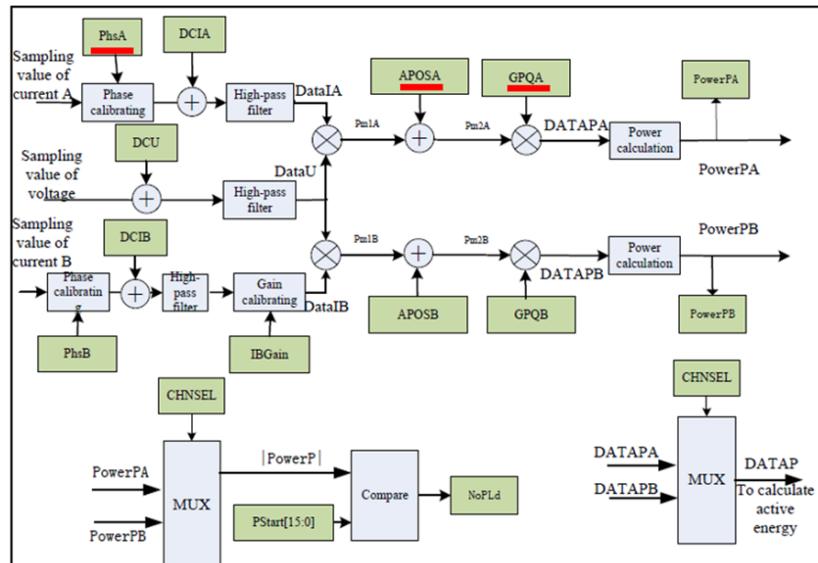
Proses kalkulasi pada IC ini terjadi pada blok EMU. Mengacu pada *datasheet* yang telah disediakan, berikut proses kalkulasi dari IC Renergy 8209 yang direpresentasikan dengan diagram blok. Sebagai tambahan, beberapa blok



Gambar 2.5. Diagram blok IC Renergy 8209

pada figur-figr yang ada selanjutnya ada yang diberi garis merah sebagai tanda bahwa blok tersebut merupakan blok register kalibrasi.

(a) Daya aktif

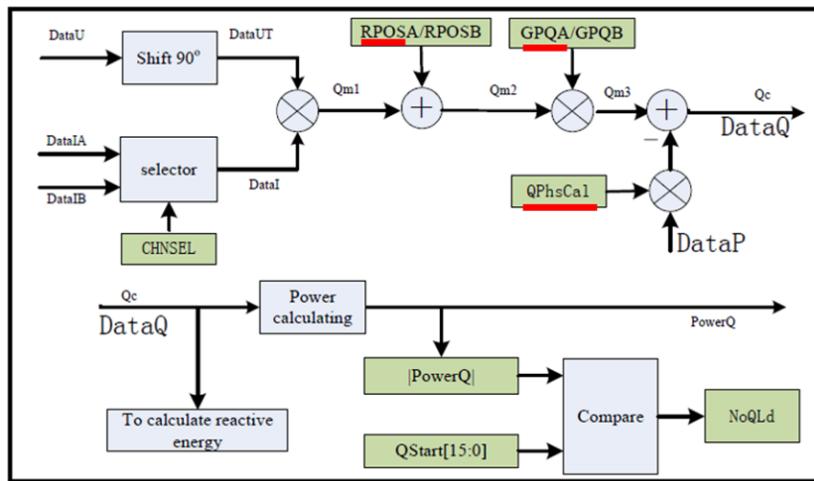


Gambar 2.6. Diagram blok kalkulasi daya aktif

Gambar 2.6 [15] merupakan diagram blok dari proses kalkulasi yang dilakukan oleh IC Renergy 8209. Sebelum ke blok *High-pass filter*, masukan arus yang telah melalui ADC akan diproses pada blok *phase calibrating* untuk koreksi fase apabila *power factor* bernilai rendah dengan konstanta kalibrasi yang disimpan pada alamat register PhsA. Sementara

masukan tegangan langsung diteruskan ke *High-pass filter* untuk menghilangkan derau pada frekuensi rendah. Selanjutnya nilai tegangan dan arus dikali untuk mendapatkan nilai daya aktif. Blok APOSA berfungsi untuk menghilangkan *offset* dan blok GPQA berfungsi untuk kalibrasi *gain channel* A juga mengurangi kesalahan pembacaan energi. Nilai daya aktif yang telah terkalibrasi disimpan pada alamat register PowerPA.

(b) Daya reaktif

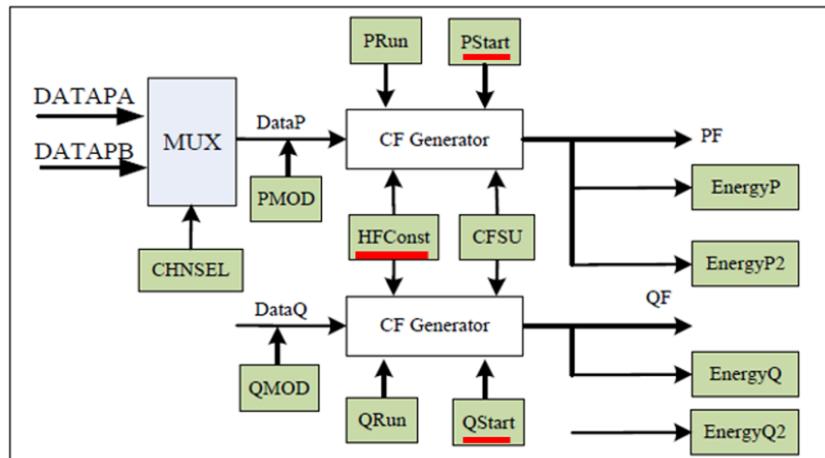


Gambar 2.7. Diagram blok kalkulasi daya reaktif

Dalam melakukan perhitungan daya reaktif, IC Renergy 8209 menggunakan masukan nilai daya aktif yang telah dihitung sebelumnya dan perkalian arus dengan tegangan yang fasenya telah digeser sebesar 90 derajat seperti pada Gambar 2.7 [15]. Metode tersebut dinamakan metode *Time Delay* [16]. Hasil perkalian arus dengan tegangan yang fasenya digeser sebesar 90 derajat kemudian akan dihilangkan *offsetnya* dengan blok RPOSA dan dikalibrasi *gain-nya* dengan blok GPQA. Setelah itu akan dikurang dengan hasil perhitungan daya aktif yang telah dikali dengan blok QphsCal atau register kalibrasi *gain* daya reaktif. Hasil perhitungan daya reaktif dapat diakses pada alamat register PowerQ.

(c) Energi aktif dan reaktif

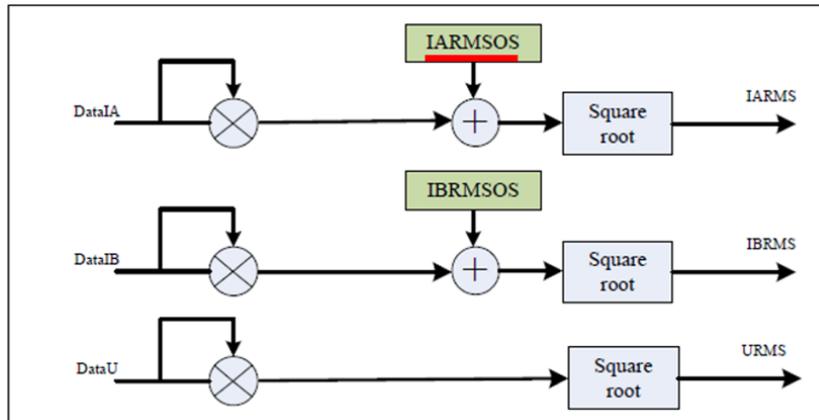
IC Renergy 8209 menggunakan metode pulsa untuk menghitung energi aktif dan reaktif. Cara kerjanya adalah IC akan melakukan *counter* energi sampai batas nilai tertentu yang diatur oleh blok HFCConst dan apabila batas tersebut terpenuhi maka akan mengeluarkan pulsa yang merepresentasikan pertambahan nilai pada register energi sebesar satu, hal tersebut berlaku untuk energi aktif dan reaktif. Pulsa tersebut dapat diamati pada pin PF untuk energi aktif dan QF untuk energi reaktif. Nilai akumulasi pulsa energi dapat diakses



Gambar 2.8. Diagram blok kalkulasi energi

pada alamat register EnergyP dan EnergyQ. Blok PStart dan QStart berfungsi sebagai batas nilai daya untuk mulai melakukan *counter*. Artinya jika nilai daya dibawah dari Pstart dan Qstart maka IC tidak akan menambah nilai pada *counter*. Hal ini dilakukan agar IC tidak menghitung pertambahan energi pada saat kondisi tidak ada beban. Diagram blok prosesnya dapat dilihat pada Gambar 2.8 [15].

(d) Tegangan dan arus RMS

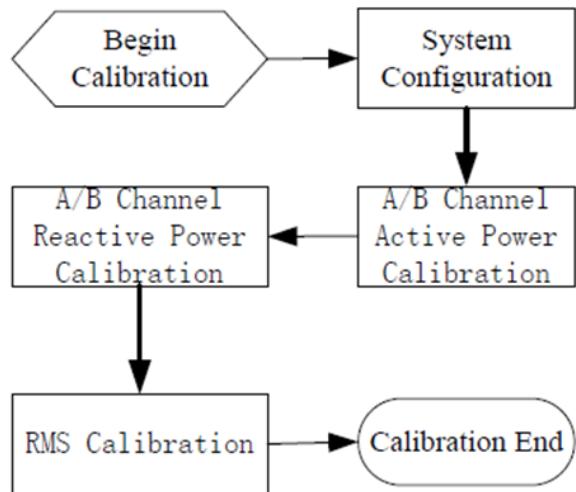


Gambar 2.9. Diagram blok kalkulasi tegangan dan arus RMS

Gambar 2.9 [15] adalah diagram blok perhitungan nilai tegangan RMS dan arus RMS. Nilai arus dan tegangan yang telah melalui blok *phase calibration* dan *high-pass filter* kemudian akan dilakukan perkalian dengan nilainya sendiri dan operasi akar kuadrat untuk mendapatkan nilai RMS. Blok IARMSOS berfungsi untuk mengeliminasi kemungkinan *offset* pada arus *channel A*.

3. Proses kalibrasi

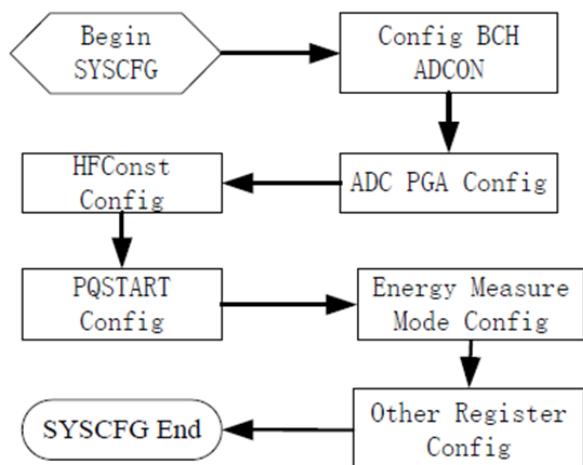
Kalibrasi merupakan proses yang penting dalam pengembangan energimeter agar perangkat dapat menghasilkan nilai yang mendekati nilai sebenarnya. Selain itu, pada proses kalibrasi, pengembang mengatur karakteristik dari sistem *monitoring* konsumsi listrik dengan IC tersebut sesuai dengan keinginan pabrikan. Pada IC Renergy 8209, alur kalibrasi telah disediakan pada *datasheet*.



Gambar 2.10. Diagram blok kalibrasi IC Renergy 8209

Secara garis besar, alur kalibrasi dapat dilihat pada Gambar 2.10 [15]. Pertama perlu dilakukan konfigurasi sistem lalu kalibrasi daya aktif dan reaktif kemudian kalibrasi RMS. Perlu diperhatikan bahwa alur kalibrasi dijalankan secara sekuensial dan alur prosesnya tidak bisa ditukar karena masing-masing blok bergantung pada blok sebelumnya dalam hal perhitungan nilai parameter kalibrasi.

(a) Konfigurasi sistem



Gambar 2.11. Diagram blok konfigurasi sistem

Konfigurasi sistem bertujuan untuk mengatur nilai-nilai register konfigurasi yang terdapat pada IC Renergy 8209 agar karakteristiknya sesuai dengan keinginan pabrikan. Representasi diagram blok dapat dilihat pada Gambar 2.11 [15]. Register konfigurasi yang dimaksud adalah SYSCON, EMUCON, HFConst, Pstart, Qstart, dan EMUCON2. SYSCON berfungsi untuk mengkonfigurasi *gain* OpAmp arus dan tegangan (blok ADC PGA Config), *baud rate* dari IC untuk komunikasi UART, dan pengatur ADC *channel* B (dalam penelitian ini tidak digunakan karena hanya *channel* A saja yang dipakai). EMUCON berfungsi untuk mengkonfigurasi register energi (blok *energy measure mode config*). HFConst berfungsi untuk mengkonfigurasi *energy constant* (EC) atau banyaknya pulsa untuk merepresentasikan 1 kWh dan satunya adalah impuls/kWh (blok HFConst config). Untuk menentukan nilai HFConst digunakan formula sebagai berikut:

$$HFConst = \text{int}\left[\frac{16.1079 \cdot V_u \cdot V_i \cdot 10^{11}}{EC \cdot U_n \cdot I_d}\right]$$

$$V_u = U_d \cdot \text{ratio voltage divider} \cdot U \text{ Gain}$$

$$V_i = I_d \cdot R_{shunt} \cdot ChA \text{ Gain} \quad (2-5)$$

$$EC = \text{meter constant}$$

$$U_n = \text{tegangan dasar}$$

$$I_d = \text{arus dasar}$$

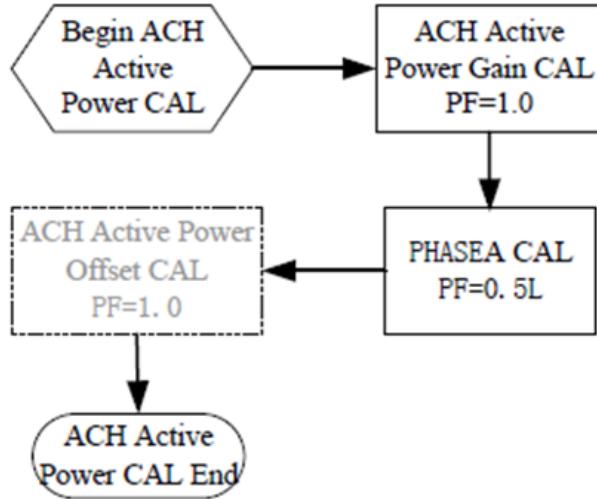
Pstart dan Qstart telah dijelaskan pada bagian sebelumnya (blok PQSTART config) dan EMUCON2 berfungsi untuk memperbaiki lebih jauh koreksi fase.

(b) Kalibrasi daya aktif

Terdapat tiga kondisi yang harus dilakukan untuk melakukan kalibrasi daya aktif sesuai dengan Gambar 2.12 [15]. Kondisi pertama adalah ketika faktor daya bernilai satu dengan arus sebesar arus dasar yang telah ditetapkan. Kondisi tersebut untuk mencari konstanta kalibrasi *gain* daya aktif (GPQA) yang didapatkan dari formula berikut:

$$P_{gain} = \frac{-ep}{1 + ep}$$

$$\text{if}(err \geq 0) \longrightarrow GPQA = \text{int}[P_{gain} \cdot 32768] \quad (2-6)$$



Gambar 2.12. Diagram blok kalibrasi daya aktif

$$\text{otherwise} \longrightarrow GPQA = \text{int}[(Pgain \cdot 32768) + 65536]$$

ep = error pulse

Untuk mencari *error pulse* dapat mengikuti formula berikut:

$$ep = -\frac{T_{terukur} - T_{diharapkan}}{T_{diharapkan}}$$

$$T_{terukur} = \text{waktu antara dua pulsa} \quad (2-7)$$

$$T_{diharapkan} = \frac{EC \cdot Y}{1000 \cdot 3600}$$

$$Y = \text{daya aktif atau daya reaktif}$$

Setelah nilai konstanta kalibrasi *gain* daya aktif ditemukan, nilai tersebut disimpan pada alamat register GPQA IC. Kondisi selanjutnya yang perlu diterapkan adalah tegangan dan arus dasar dengan faktor daya bernilai 0.5. Tujuannya adalah untuk mencari konstanta kalibrasi fase (PhsA) yang didapatkan dari formula berikut:

$$\theta = \arcsin\left(\frac{-ep}{\sqrt{3}}\right)$$

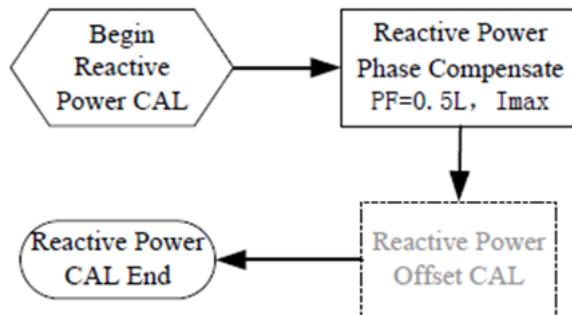
$$0.02^\circ \text{ per LSB untuk } f = 50 \text{ Hz} \quad (2-8)$$

$$\text{if } (\theta \geq 0) \longrightarrow PhsA = \text{int}\left[\frac{\theta}{0.02}\right]$$

$$\text{otherwise} \longrightarrow PhsA = \text{int}\left[\frac{256 + \theta}{0.02}\right]$$

Nilai konstanta kalibrasi fase kemudian disimpan pada alamat register PhsA. Proses selanjutnya adalah mencari nilai konstanta kalibrasi *offset* namun hal ini bersifat opsional dan hanya dilakukan apabila *offset* lebih dari 5% dari arus dasar. Caranya adalah dengan mengkondisikan arus menjadi nol atau tanpa beban. Hasil pembacaan dari IC kemudian dikomplemenkan dan disimpan pada alamat register APOSA dan kalibrasi daya aktif selesai.

(c) Kalibrasi daya reaktif



Gambar 2.13. Diagram blok kalibrasi daya reaktif

Setelah kalibrasi daya aktif selesai, kalibrasi daya reaktif bisa dilakukan. Perlu diketahui bahwa apabila IC Renergy 8209 hanya digunakan untuk menghitung energi aktif saja maka proses ini bisa diabaikan. Terdapat dua kondisi yang dilakukan untuk mencari dua konstanta kalibrasi seperti pada Gambar 2.13 [15]. Kondisi pertama adalah pada tegangan dan arus dasar dengan faktor daya bernilai 0.5. Kondisi ini dilakukan untuk mencari konstanta kalibrasi fase dan nilai kalibrasi tersebut dapat ditentukan dengan formula berikut:

$$\alpha = \frac{ep}{\cot(\theta)}$$

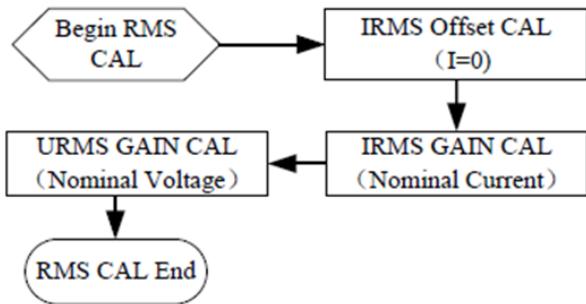
$$if \ PF = 0.5 \ then \ \alpha = ep \cdot 0.5774 \quad (2-9)$$

$$if (\alpha \geq 0) \longrightarrow QPhsCal = int[\alpha \cdot 0.5774]$$

$$otherwise \longrightarrow QPhsCal = int[\alpha \cdot 0.5774 + 65536]$$

Nilai yang didapatkan kemudian disimpan pada alamat register QPhsCal. Kondisi kedua adalah ketika tidak ada beban dan digunakan untuk mencari nilai *offset* daya reaktif. Cara menentukannya sama seperti mencari nilai *offset* pada daya aktif.

(d) Kalibrasi RMS



Gambar 2.14. Diagram blok kalibrasi RMS

Kalibrasi terakhir adalah kalibrasi untuk nilai RMS arus dan tegangan. Representasi diagram blok terlihat pada Gambar 2.14 [15]. Proses yang pertama dilakukan adalah memperbaiki terjadinya *offset* pada arus. Caranya adalah dengan melakukan beberapa kali pembacaan arus pada kondisi tanpa beban kemudian direrata. Hasil tersebut dikuadratkan lalu nilainya dikomplemenkan dan empat digit bilangan heksadesimalnya diambil kemudian disimpan pada alamat register IARMSOS.

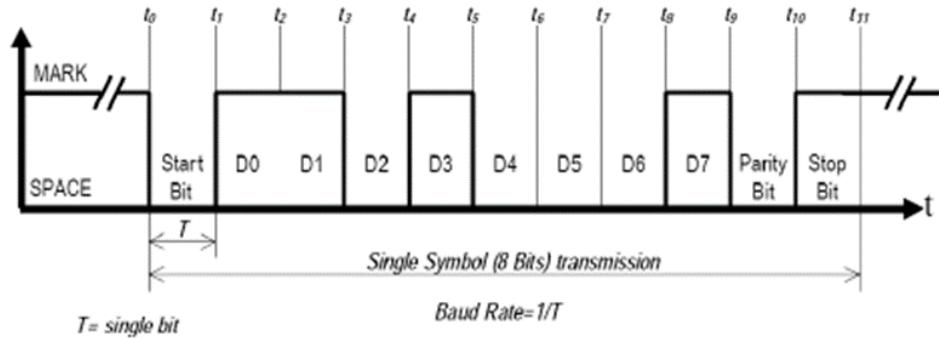
Langkah selanjutnya adalah mencari konstanta konversi arus, tegangan, daya aktif, daya reaktif, energi aktif, dan energi reaktif. Hal ini dilakukan karena hasil pembacaan yang dilakukan oleh IC nilainya sangat besar karena mengalami perbesaran internal. Sehingga perlu dicari suatu konstanta pembagi agar nilainya berada pada jangkauan yang sebenarnya.

Untuk mencari konstanta pembagi arus, tegangan, dan daya aktif dilakukan pada kondisi faktor daya sebesar 1 dan arus dasar. Hasil pembacaan oleh register terkait yaitu IARMS, URMS, dan PowerPA kemudian dibagi dengan nominal pada referensi. Untuk mencari konstanta daya reaktif dilakukan pada kondisi faktor daya sebesar 0.5 dan arus maksimal. Kemudian hasil pembacaan dari register daya reaktif yaitu PowerQ dibagi dengan nominal pada referensi. Sementara untuk energi aktif dan reaktif nilai konstanta konversinya adalah EC dibagi 1000. Dengan selesainya proses ini maka kalibrasi dari IC Renergy 8209 telah selesai.

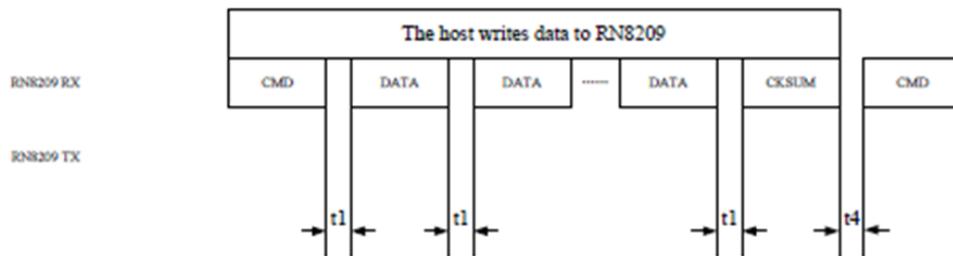
4. Format komunikasi

Seri IC Renergy 8209 yang dipakai untuk skripsi kali ini hanya bisa diakses menggunakan UART sebagai protokol komunikasinya dan *baud rate* nya bersifat *fixed* pada 4800 bps. Format paket yang digunakan berukuran 11 bit dengan urutan *start bit* (1 bit), *data frame* (8 bit), *even parity* bit (1 bit), dan *stop bit* (1 bit). Ilustrasi paket dapat dilihat pada Gambar 2.15 [15].

Untuk memberi nilai pada register, tata caranya seperti Gambar 2.16 [15].



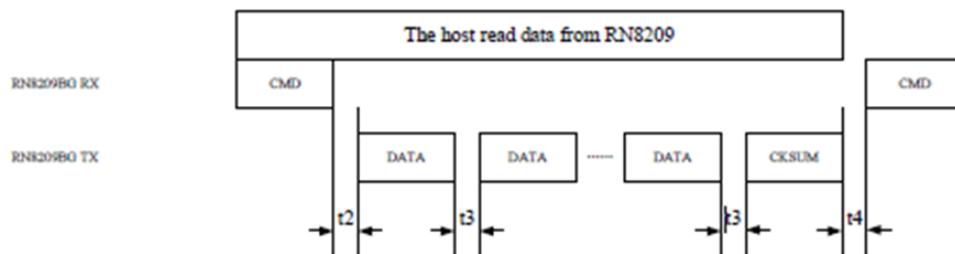
Gambar 2.15. Paket byte komunikasi IC Renergy 8209



Gambar 2.16. Urutan paket memberi nilai register

Komunikasi diinisiasi dengan mengirimkan byte CMD yang berisi alamat register dengan *most significant bit* (MSB) bernilai 1. Kemudian diikuti oleh paket-paket data yang akan diisi ke dalam alamat register. Terakhir ditutup dengan byte CKSUM untuk memastikan integritas data yang ditransmisikan. Cara perhitungan CKSUM adalah sebagai berikut:

$$CKSUM[7 : 0] = \sim (CMD[7 : 0] + DATA_n[7 : 0] + \dots + DATA_1[7 : 0]) \quad (2-10)$$



Gambar 2.17. Urutan paket membaca nilai register

Sementara untuk membaca nilai pada alamat register di IC Renergy 8209, mempunya tata caranya tersendiri seperti pada Gambar 2.17 [15]. Pertama MCU harus mengirimkan byte CMD berisi alamat register yang dituju dengan MSB

bernilai 0. Kemudian IC akan mengirimkan data yang terdapat pada alamat register tersebut dan ditutup dengan *byte CKSUM*.

Format komunikasi ini perlu diperhatikan dalam pengembangan *firmware* dari IC Renergy 8209. Apabila tata caranya tidak sesuai maka IC bisa saja menerima atau mengirim data yang tidak tepat.

2.2.3 Pengembangan *Firmware* Pada Platform ESP32

Seperti yang sudah disebutkan pada bab sebelumnya dibagian tujuan penelitian, mengembangkan *firmware* merupakan salah satu tujuan dari skripsi ini. Pada implementasinya setiap perangkat yang menggunakan mikrokontroller akan menggunakan *firmware* untuk mengoperasikannya. *Firmware* ini termasuk ke dalam ciptaan yang memiliki hak cipta dan perlindungannya diatur dalam UU Nomor 19 Tahun 2002 tentang Hak Cipta.

1. Pengembangan *firmware*

Firmware atau bisa disebut sebagai perangkat lunak sistem tertanam merupakan perangkat lunak yang dirancang untuk sistem tertanam (*embedded system*) dan disimpan pada memori non volatil seperti ROM, EPROM, EEPROM, atau FLASH. *Firmware* juga bisa diartikan sebagai kode program untuk melakukan perintah pada sebuah prosesor. Secara garis besar, *firmware* digunakan untuk mengontrol perangkat keras dengan memberi dan membaca nilai dari suatu register (akan dibahas lebih lanjut pada bagian lainnya) dan memberikan respon terhadap permintaan interupsi dari perangkat keras.

Pada pengembangan *firmware*, orang yang mengembangkan *firmware engineer* atau *firmware developer*. Seorang *firmware engineer* akan berhubungan erat dengan *hardware engineer* atau orang yang mengembangkan perangkat keras dari suatu produk sistem tertanam karena masing-masing akan membutuhkan satu sama lain. Sebuah *firmware* tidak bisa digunakan jika tidak ada perangkat kerasnya dan perangkat keras tidak akan bisa beroperasi apabila tidak terdapat *firmware* didalamnya.

Gary Stringham dalam bukunya tentang *Hardware and Firmware Interface Design* menyatakan ada tujuh prinsip dalam pengembangan produk sistem tertanam yang mencakup *firmware* dan *hardware* [17].

- (a) Kolaborasi : *firmware* dan *hardware engineer* saling berkoordinasi dalam pengembangan produk, biasanya dalam bentuk dokumentasi dari pekerjaannya dan presentasi berkala.
- (b) Mengikuti suatu standar : prinsip ini diterapkan untuk menjaga suatu kualitas

produk yang dikembangkan.

- (c) Beban kerja merata : antara *firmware* dan *hardware engineer* memiliki beban kerja yang sama dengan memperhatikan produk dan *constraints*.
- (d) Kompabilitas : versi apapun dari *firmware* yang dikembangkan akan bekerja pada semua versi *hardware* dari produk yang sama.
- (e) Antisipasi : dalam pengembangan produk selalu memikirkan dampak yang mungkin terjadi dalam perancangan *firmware* dan *hardware*.
- (f) Kontingensi : mempersiapkan keamanan *firmware* dan *hardware* untuk kemungkinan masalah yang akan muncul ke depannya.
- (g) Terencana : pengembangan produk memiliki rencana dan tujuan yang mengacu pada kerangka kerja tertentu.

2. Menghubungkan ke *hardware*

Firmware yang dirancang salah satunya memiliki tugas utama untuk berkomunikasi dengan perangkat keras. Untuk bisa menjalankan tugas tersebut dibutuhkan suatu tata cara atau protokol komunikasi. Selain itu ada juga komponen alamat yang akan dikomunikasikan atau dalam perancangan produk sistem tertanam disebut sebagai register.

(a) Protokol komunikasi UART

Salah satu protokol komunikasi yang paling banyak dipakai adalah UART atau *Universal Asynchronous Receiver-Transmitter*. Secara definisi UART adalah protokol komunikasi secara serial asinkron dengan konfigurasi kecepatan transmisi data tertentu. Asinkron artinya adalah tidak ada *clock* untuk sinkronisasi bit keluaran dari pengirim ke penerima. Maka agar bisa terjadi komunikasi, pihak pengirim dan penerima perlu mendeklarasikan kecepatan transmisi data atau *baud rate* dengan satuan bit per sekon (bps) [18].

Dalam berkomunikasi menggunakan UART, ada perangkat yang bertindak sebagai *master* dan ada juga yang bertindak sebagai *slave* atau satu perangkat dapat juga memegang kedua peran. Dalam pengembangan produk sistem tertanam, mikrokontroller bertindak sebagai *master* yang mengirimkan perintah dan IC bertindak sebagai *slave* yang menerima perintah. Pada konteks penelitian ini, ESP32 berperan sebagai *master* dan IC Renergy 8209 berperan sebagai *slave*. Agar bisa terhubung, terdapat antarmuka berupa sepasang pin Tx dan Rx pada kedua pihak. Pin Tx berguna untuk mengirim data ke penerima dan pin Rx adalah pin untuk

menerima data dari pin Tx.

Start Bit (1 bit)	Data Frame (5 to 9 Data Bits)	Parity Bits (0 to 1 bit)	Stop Bits (1 to 2 bits)
------------------------	------------------------------------	-------------------------------	------------------------------

Gambar 2.18. Format bit UART

Protokol komunikasi UART mengirim data dalam bentuk paket dengan format tertentu yang disesuaikan pada perangkat. Paket tersebut memiliki format seperti Gambar 2.18 [18]. Paket diawali dengan bit mulai (*start bit*) sebagai tanda bahwa bit selanjutnya terdapat informasi. Selanjutnya adalah *frame* data yang berisi informasi dengan ukuran lima sampai sembilan bit tergantung dari format yang berlaku pada perangkat. Dibelakangnya terdapat bit paritas atau *parity* bit yang berguna untuk mendeteksi terjadinya kesalahan pada pengiriman yang biasanya disebabkan oleh radiasi elektromagnetik, perbedaan *baud rate*, atau lintasan transmisi data yang panjang sehingga ada perubahan dari bit data. Bit paritas ini bersifat opsional dan tidak semua perangkat menggunakan bit ini. Terakhir adalah bit stop yang menandai akhir dari paket.

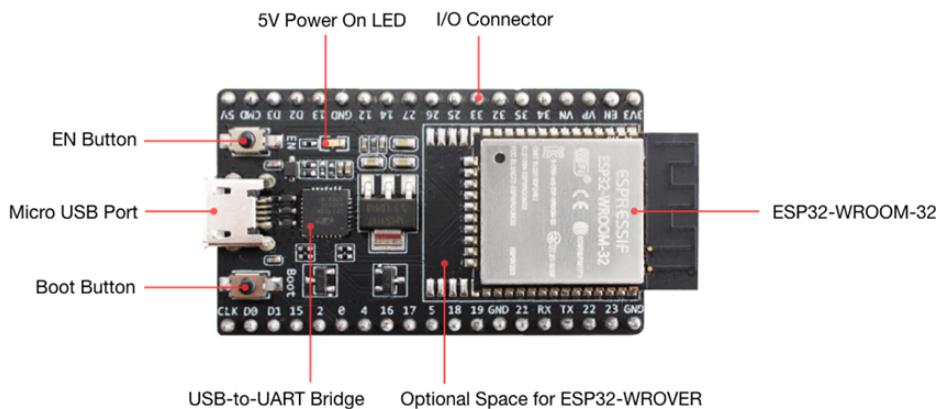
(b) Register

Register adalah antarmuka yang disediakan oleh perangkat keras untuk diakses oleh *firmware*. *Firmware* akan berkomunikasi dengan perangkat keras pada alamat yang disebut register. Cara yang dilakukan adalah dengan membaca nilai register untuk mendapatkan informasi dan memberi nilai register untuk memberi perintah atau mengubah konfigurasi pada perangkat keras. Perangkat keras memberikan informasi dengan menyimpan informasi tersebut pada register juga menerima informasi dari *firmware* di alamat register yang tersedia [17].

Register dapat dianalogikan sebagai sebuah ruangan dalam gedung. Ruangan tersebut memiliki nomor atau alamat agar dapat membedakan antara satu ruang dan ruang lainnya. Kemudian ruangan tersebut memiliki ukuran tertentu yang ukurannya tergantung dari pabrikan dan berisi informasi tertentu, ada yang satu *byte*, 2 *byte*, dan lain-lain. Terakhir setiap ruangan memiliki fungsinya masing-masing untuk menunjang operasi pada gedung tersebut. Artinya setiap register pada suatu perangkat keras memiliki tugas dan perannya masing-masing dalam perangkat keras.

3. ESP32 *development board*

ESP32 adalah *development board* pabrikan dari Espressif System yang didalamnya terdapat mikrokontroller, pin masukan dan keluaran, hingga Wi-Fi.



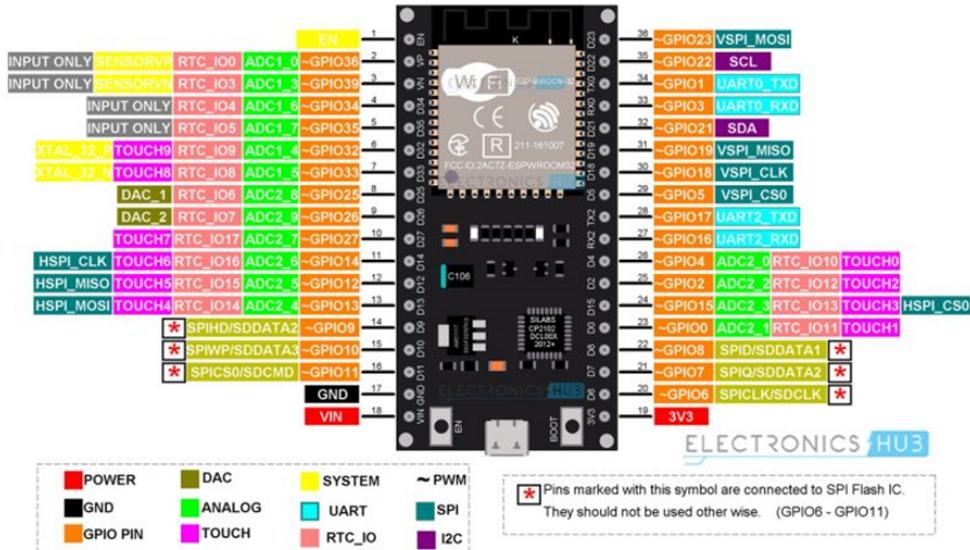
Gambar 2.19. Tampilan papan ESP32

ESP32 berfungsi sebagai perangkat yang memuat semua perintah dari *firmware* yang telah dirancang sekaligus menyimpannya untuk didistribusikan ke perangkat keras yang telah terkoneksi. Keuntungan yang ditawarkan dari *development board* ini adalah harganya yang murah, komunitas yang luas, fitur Wi-Fi dan *bluetooth*, dan lain-lain [19]. Tampilan dari *board* ESP32 bertipe ESP32-DevKitC V4 bisa dilihat pada Gambar 2.19 [20].

ESP32 DevKit *Board* merupakan jenis *development board* ESP32 yang akan digunakan pada skripsi ini dan memiliki beberapa komponen kunci yaitu:

- ESP32-WROOM-32 : modul ESP32 yang didalamnya terdapat mikrokontroller.
- EN : tombol untuk melakukan reset pada board
- Boot : tombol untuk mengunduh *firmware* dari *port serial*.
- USB-to-UART Bridge : jembatan penghubung antara komunikasi UART menuju USB.
- Port micro USB : antarmuka untuk *micro* USB sebagai jalur komunikasi antara pengguna dan/atau catu daya.
- LED : indikator ketika catu daya terhubung.
- Pin I/O : pin masukan dan keluaran.

Pada papan ESP32 DevKit terdapat 36 pin dengan fungsinya masing-masing seperti pada Gambar 2.20 [21]. Secara umum konfigurasi pin dari ESP32 DevKit mencakup sebagai berikut:



Gambar 2.20. Konfigurasi pin ESP32

- 34 pin GPIO yang dapat diprogram.
- 18 pin ADC 12 bit.
- 2 pin DAC 8 bit.
- 16 pin PWM.
- 3 pin antarmuka UART.
- 3 pin antarmuka SPI.
- 2 pin antarmuka I2C.
- 2 pin antarmuka I2S.

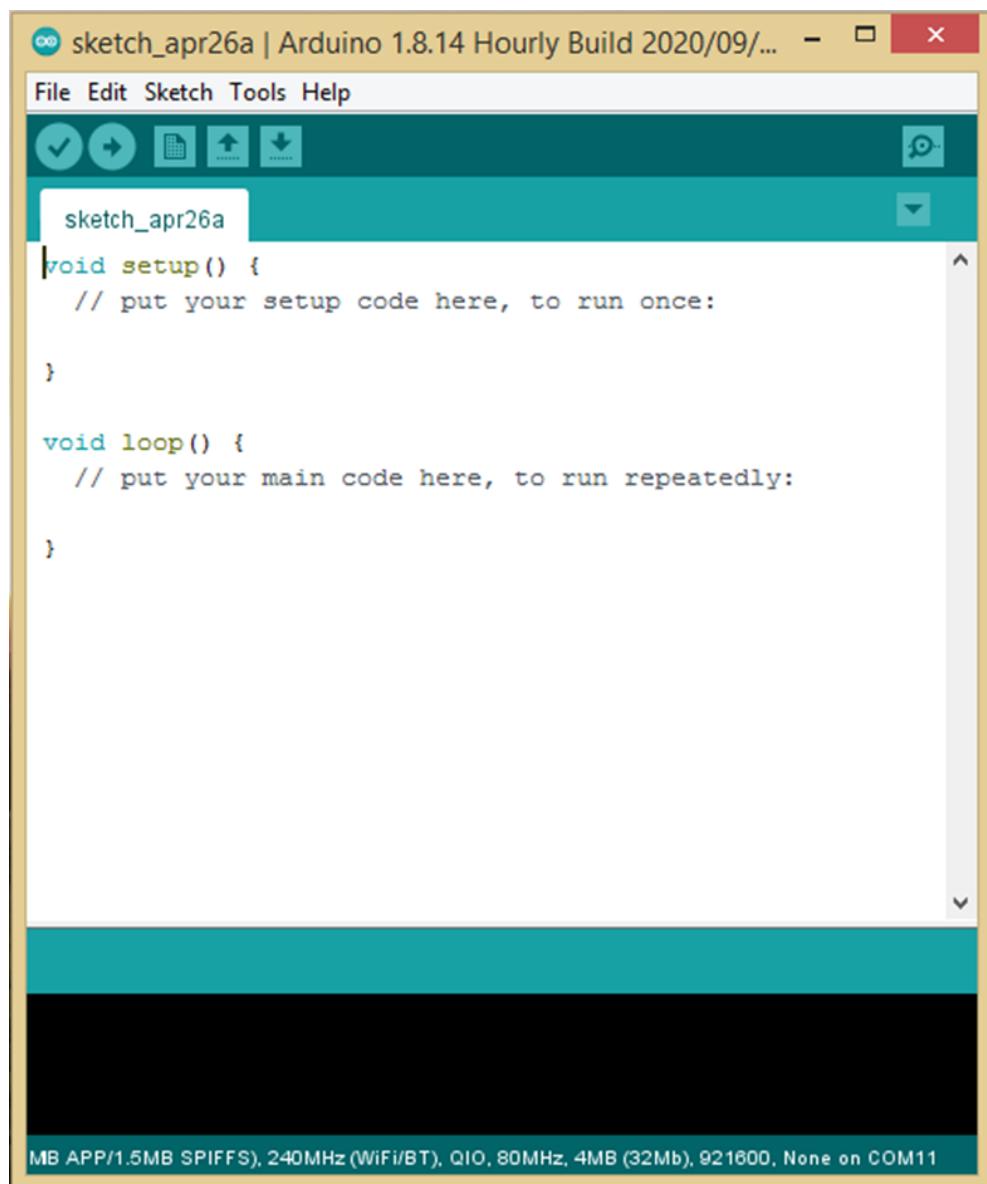
Spesifikasi yang dimiliki oleh ESP32 DevKit adalah sebagai berikut:

- Prosesor : Tensilica 32-bit single/dual core CPU Xtensa LX6
- Memori : 520 KB SRAM
- Konektivitas nirkabel : Wi-Fi 802.11 b/g/n, Bluetooth v4.2

4. Arduino IDE

Dalam mengembangkan sebuah *firmware* diperlukan sebuah bahasa pemrograman tertentu dan *compiler* untuk mengubah bahasa pemrograman menjadi bahasa mesin. Pemilihan bahasa pemrograman dan *compiler* disesuaikan dengan jenis development board yang dipakai pada proses pengembangan. Lingkungan yang terintegrasi untuk mengembangkan sebuah sistem tertanam disebut sebagai *Integrated Development Environment* (IDE) dan salah satu yang paling banyak penggunaannya adalah Arduino IDE.

Sebenarnya arduino IDE diperuntukkan *development board* pabrikan dari perusahaan Arduino. Akan tetapi seiring berjalananya waktu banyak orang lain yang juga mengembangkan *library* agar *development board* pabrikan lain dapat menggunakan Arduino IDE. Hal ini bukan menjadi masalah karena Arduino bersifat *open source* [22]. Termasuk pengembangan menggunakan ESP32 yang dapat dilakukan pada IDE ini.



Gambar 2.21. Tampilan awal Arduino IDE

Gambar 2.21 merupakan tampilan mula ketika menjalankan aplikasi Arduino IDE. Pada bagian *toolbar* terdapat beberapa menu yang bisa dipilih seperti membuat *file* baru pada menu *file*, menambahkan *library* baru pada menu *sketch*, mengkonfigurasi *port* dan jenis papan pada menu *tools*, dan lain-lain. Dibawah *toolbar* terdapat beberapa ikon yang berguna untuk melakukan *compiling*, *uploading*, membuat *file* baru, membuka *file* lain, menyimpan *file*, dan

membuka serial monitor. Untuk serial monitor sendiri merupakan fitur yang disediakan oleh Arduino IDE untuk memantau keluaran dari MCU dan memberi masukan ke MCU. Bagian putih yang ada di tengah merupakan tempat untuk menulis kode program atau *firmware*. Terakhir kotak hitam yang berada dibawah adalah *log* komunikasi dari PC ke MCU.

Arduino IDE menggunakan bahasa pemrograman arduino yang berbasis pada C/C++ yang telah dimodifikasi untuk keperluan sistem tertanam [23]. Pada gambar tampilan mula dari Arduino IDE, terdapat dua fungsi yang langsung tertulis pada kanvas yaitu *void setup* dan *void loop*. Dalam pemrograman arduino, kedua fungsi tersebut wajib untuk digunakan. Fungsi *setup* akan dijalankan sekali dan pertama kali saat MCU direset. Fungsi ini biasanya akan diisi kode inisialisasi dari perangkat. Sementara fungsi *loop* akan dijalankan secara berulang-ulang sampai reset selanjutnya dilakukan. Di dalam fungsi inilah segala aktivitas pada perangkat akan dijalankan secara berulang.

2.3 Analisis Perbandingan Metode

Berdasarkan tinjauan pustaka telah diulas tiga penelitian yang memiliki pendekatannya masing-masing dalam melakukan penelitiannya. Ketiga penelitian tersebut memiliki satu kesamaan yaitu dalam melakukan pengujian tidak disebutkan mengacu pada standar tertentu. Walau penelitian yang dilakukan oleh Guimaraes sudah membandingkan kesalahan pembacaan dengan batas kesalahan yang diperbolehkan pada standar yang berlaku di Brazil, namun tidak disebutkan tata cara pengujian berdasarkan standar yang disebutkan. Mengacu pada suatu standar tertentu merupakan salah satu dari prinsip pengembangan produk sistem tertanam yang dikemukakan oleh Gary Stringham. Tujuannya adalah untuk menjaga kualitas dari produk yang dikembangkan. Sehingga hal tersebut menjadi catatan yang perlu diperhatikan pada penelitian yang penulis lakukan. Dalam melakukan pengujian nanti, penulis akan mengikuti tata cara sesuai dengan SK Ditjen Perlindungan Konsumen dan Tertib Niaga Nomor 161 Tahun 2019.

BAB III

METODE PENELITIAN

3.1 Alat dan Bahan Tugas Akhir

Untuk menunjang penelitian yang dilakukan pada skripsi ini, digunakan beberapa alat dan bahan dengan rincian pada bagian selanjutnya

3.1.1 Alat Tugas akhir

Mayoritas peralatan yang akan digunakan pada penelitian ini sudah tersedia di Laboratorium Sistem Tenaga Listrik DTETI FT UGM. Rincian dari peralatan yang dipakai adalah sebagai berikut:

1. Variac 1
2. Variac 2
3. Osiloskop
4. Catu daya
5. Lampu pijar (beban resistif)
6. Induktor (beban induktif)
7. Voltech PM100 *single phase power analyzer* sebagai referensi
8. ESP32
9. Modul OLED
10. Rangkaian modul IC Renergy 8209
11. Saklar
12. Perangkat lunak Arduino IDE
13. Kabel USB mikro

3.1.2 Bahan Tugas Akhir

Bahan-bahan yang digunakan untuk keperluan penelitian yang dilakukan pada skripsi ini rinciannya adalah sebagai berikut:

1. Resistor
2. *Push button*
3. LED
4. Kabel dan jumper

3.2 Metode yang Digunakan

Penulis sudah melakukan studi pustaka terkait pengembangan *firmware* pada platform ESP32, sistem *monitoring* konsumsi listrik, dan IC Renergy 8209. Selain itu, penulis juga melakukan kajian lebih lanjut dari penelitian serupa yang pernah dilakukan sebelumnya. Mempertimbangkan hal tersebut, penulis telah memilih untuk melakukan metode atau pendekatan yang akan dilakukan pada skripsi ini. Terdapat empat bagian yang paling signifikan dalam pengembangan *firmware* sistem *monitoring* konsumsi listrik menggunakan IC Renergy 8209 pada platform ESP32. Keempat bagian ini disusun secara sekuensial dan masing-masing dari bagian tersebut memiliki metodenya dalam melaksanakannya, yang akan dijelaskan lebih lanjut kemudian.

3.2.1 Pengembangan *Firmware* Kalibrasi Pengujian

Pengembangan *firmware* kalibrasi dan pengujian dilakukan untuk keperluan kalibrasi IC Renergy 8209 dan pengujinya yang akan dilakukan setelahnya. *Firmware* ini berisi perintah dan fungsi yang utamanya hanya untuk keperluan tersebut. Dalam pengembangan *firmware* ini memperhatikan tujuh prinsip pengembangan produk sistem tertanam yang telah dijabarkan sebelumnya. *Firmware* kalibrasi dan pengujian bukanlah *firmware* final karena peruntukannya berbeda sehingga fitur yang dirancang juga berbeda. *Firmware* kalibrasi diperuntukkan untuk pengembang atau orang yang akan mengkalibrasi sementara *firmware* final diperuntukkan kepada pengguna yang akan menggunakan sistem *monitoring* konsumsi listrik yang dirancang. Pada *firmware* ini, digunakan Arduino IDE dengan ESP32 sebagai *development board* dan fitur yang akan dirancang adalah sebagai berikut :

1. Komunikasi UART dengan IC Renergy 8209

Fitur ini berisi tata cara untuk mengirim dan menerima data dari IC Renergy 8209 dalam bentuk fungsi. Pada pembuatan fitur komunikasi UART dengan IC, perlu memperhatikan aturan yang telah diberikan oleh pabrikan IC. Aturan tersebut dapat diketahui dengan melihat *datasheet* IC ini. Pada bab sebelumnya juga telah diulas mengenai aturan atau tata cara yang diterapkan oleh IC Renergy 8209.

2. Membaca nilai register

Terdapat urutan yang harus diikuti dalam membaca nilai register di IC Renergy 8209, jika tidak maka hasil yang didapat boleh jadi bukan yang diharapkan atau bahkan IC tidak merespon apa-apa. Urutan yang dimaksud dapat dilihat pada Gambar 2.17. Untuk memastikan integritas dari data yang diterima, nantinya juga akan dibuat rutin untuk menghitung CKSUM secara komputasi dan membandingkannya dengan byte CKSUM yang diterima dari IC. Apabila CKSUM komputasi dan yang diterima dari IC tidak sama, maka MCU akan

melakukan permintaan ulang ke IC.

3. Memberi nilai register

Sama seperti membaca nilai register, memberi nilai register pada IC Renergy 8209 juga memiliki urutannya tersendiri. Urutan tersebut telah diulas pada bab sebelumnya dan gambaran umumnya bisa dilihat pada Gambar 2.16. Tidak semua register di IC Renergy 8209 bisa diberi nilai. Beberapa register hanya bisa dibaca. Namun setiap register yang bisa diberi nilai sudah pasti bisa dibaca oleh MCU. Semua register yang bisa diberi nilai oleh MCU bersifat terproteksi dan memiliki tata cara khusus untuk bisa mengaksesnya. Secara *default*, register yang bisa diberi nilai, terproteksi pada saat reset dilakukan. Untuk membukanya perlu mengirim alamat register dan data khusus membuka proteksi register ke IC Renergy 8209. Setelah proteksi terbuka, barulah data bisa mulai diberi pada alamat register yang dituju.

Register yang telah diberi nilainya kemudian dibaca nilai yang ada di dalamnya untuk memastikan data telah tersimpan. Apabila data gagal tersimpan, akan terdapat pemberitahuan bahwa data gagal tersimpan dan secara otomatis akan melakukan pengiriman ulang sampai jumlah tertentu. Jika masih gagal juga, maka proses dihentikan dan terdapat pemberitahuan bahwa terjadi kegagalan pengiriman data pada register yang dimaksud. Fungsi ini diperlukan untuk memastikan data telah tersimpan dan memberi tahu bahwa jika ada register yang gagal menyimpan data.

Setelah operasi memberi nilai register selesai, proteksi diaktifkan kembali untuk mencegah perubahan nilai ketika *looping* dijalankan. Caranya adalah dengan mengirim alamat register dan data khusus untuk mengaktifkan proteksi ke IC Renergy 8209.

4. Inisialisasi konfigurasi IC

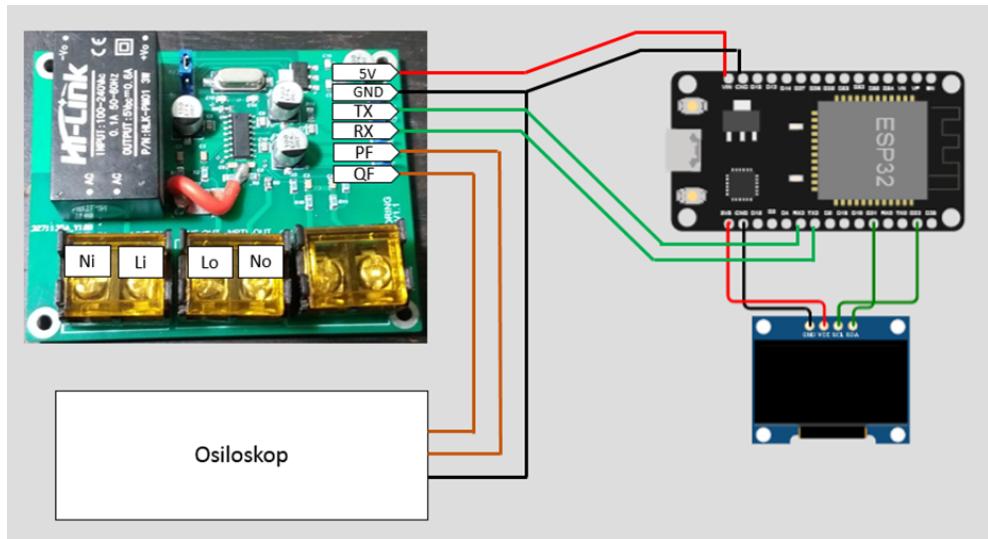
Pada proses kalibrasi yang dilakukan nanti, akan ada banyak register yang harus diberi nilai untuk mengatur konfigurasi IC Renergy 8209. Kemudahan untuk mengubah dan memberi nilai pada setiap register yang diinginkan menjadi tujuan dari fitur ini supaya dapat meminimalisir *human error*. Setiap register yang dituju diberi keterangan nama dari alamat register yang dituju dengan pemberian nama variabel yang mudah dimengerti.

5. Tampilan pada OLED

Fitur terakhir yang perlu ditambahkan adalah fitur untuk menampilkan data-data yang diperlukan pada OLED. Hal ini dilakukan untuk memudahkan pembacaan yang telah dilakukan oleh MCU tanpa perlu melihatnya pada serial monitor. Kemudahan lainnya yang bisa didapatkan adalah sifat modularitas dari

OLED itu sendiri sehingga pada saat pengujian dan ingin membandingkannya dengan referensi, hal itu bisa dilakukan dengan OLED. Data yang ditampilkan pada OLED adalah tegangan RMS, arus RMS, daya aktif, daya reaktif, energi aktif, energi reaktif, dan faktor daya. Fitur ini akan membantu pada saat pengujian.

Sebagai langkah untuk memudahkan kolaborasi berbagai pihak, pemberian keterangan pada bagian-bagian *firmware* juga akan dilakukan. Langkah ini juga memudahkan orang lain yang ingin menggunakan *firmware* kalibrasi dan pengujian dapat dengan lebih cepat mengerti *firmware* yang telah dibuat.



Gambar 3.1. *Wiring diagram* untuk kalibrasi dan pengujian

Gambar 3.1 merupakan *wiring diagram* dari prototipe energi meter di sisi perangkat keras. Agar *firmware* yang dikembangkan bisa berkomunikasi dengan IC Renergy 8209, perlu memperhatikan konfigurasi pin dari perangkat keras. Terdapat enam pin keluaran dari IC Renergy 8209 dengan hubungannya ke ESP32 dan osiloskop dengan konfigurasi seperti pada tabel 3.1.

Tabel 3.1. Konfigurasi pin kalibrasi dan pengujian

IC Renergy 8209	ESP32	Osiloskop
5V	Vin	-
GND	GND	GND
TX	RX2	-
RX	TX2	-
PF	-	Ch 1
QF	-	Ch 2

Untuk pin PF dan QF terhubung ke osiloskop karena nantinya akan digunakan

untuk mengamati pulsa daya aktif dan reaktif yang dikeluarkan oleh IC Renergy 8209. Pulsa ini diamati untuk proses selanjutnya yaitu kalibrasi.

Sementara konfigurasi pin-pin pada oled yang terkoneksi pada ESP32 adalah seperti tabel 3.2.

Tabel 3.2. Konfigurasi pin OLED-ESP32

OLED	ESP32
GND	GND
VCC	3.3 V
SCL	D22
SDA	D21

Pada rangkaian modul IC Renergy 8209, terminal Ni dan Li akan terhubung masukan dari catu daya. Sementara terminal No dan Lo terhubung ke beban.

3.2.2 Kalibrasi IC Renergy 8209

Pada bab sebelumnya telah dijelaskan tentang mekanisme kalibrasi yang diterapkan IC Renergy 8209. Metode yang digunakan dalam kalibrasi IC Renergy 8209 menggunakan metode yang sama seperti yang telah dicantumkan pada *datasheet*. Jika direpresentasikan dengan diagram alir, kalibrasi yang akan dilakukan secara garis besar sesuai dengan Gambar 3.2. Selanjutnya akan dijelaskan lebih teknis terkait tahapan kalibrasi yang akan dilakukan.

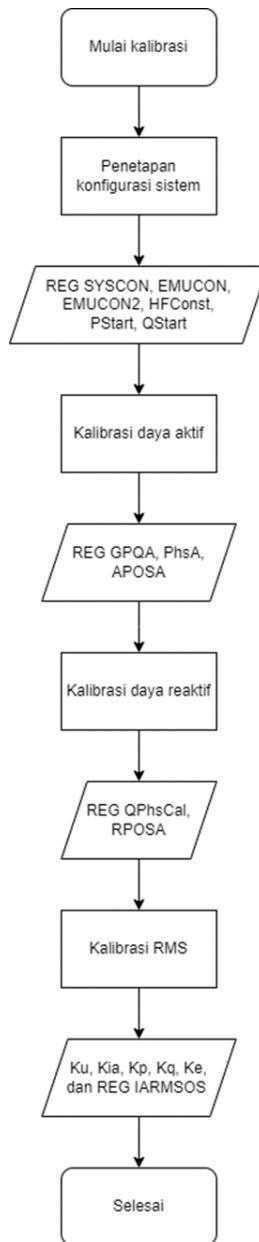
1. Konfigurasi sistem

IC Renergy 8209 memiliki beberapa opsi konfigurasi yang memungkinkan perancangnya untuk menentukan karakteristik dari IC ini. Konfigurasi yang dipilih akan berpengaruh terhadap perilaku IC saat digunakan. Untuk melakukan konfigurasi terdapat beberapa register yang harus diberi nilai sesuai dengan ketentuan yang terdapat pada *datasheet* terkait data yang harus diberi.

Sebelum membahas konfigurasi yang digunakan, terdapat kondisi yang akan diterapkan pada pengembangan sistem *monitoring* konsumsi listrik ini yaitu :

- (a) Arus dasar (I_d): 2 A
- (b) Tegangan dasar (U_n): 220 V
- (c) *Energy constant* (EC): 2000 imp/kWh

Selanjutnya ini adalah konfigurasi yang akan diterapkan pada masing-masing register konfigurasi yang digunakan untuk penelitian ini:



Gambar 3.2. Diagram alir kalibrasi

(a) Register SysCon

- i. *Baud rate*: 4800 bps
- ii. ADC saluran arus B: OFF (karena *channel* B tidak digunakan)
- iii. *Gain* saluran arus A: 8
- iv. *Gain* saluran tegangan : 2

(b) Register EMUCon

- i. Akumulasi energi: ON
- ii. *High-pass filter* saluran arus B: OFF

- iii. Mode akumulasi energi reaktif: energi reaktif bernilai positif dan negatif diakumulasi dengan $EQ = |EnergyQ|$
- iv. Mode akumulasi energi aktif: energi reaktif bernilai positif dan negatif diakumulasi dengan $EP = |EnergyP|$
- v. *High-pass filter* saluran arus A: ON
- vi. *High-pass filter* saluran tegangan: ON
- vii. Pulsa energi reaktif: ON
- viii. Pulsa energi aktif: ON

(c) Register EMUCon2

- i. Koreksi fase lanjut saluran arus A: OFF

(d) Register HFConst

Menentukan nilai dari register HFConst digunakan formula 2-5. Untuk mencapai EC yang diinginkan, perhitungan HFConst sebagai berikut:

$$HFConst = \text{int}\left[\frac{16.1079 \cdot 220 \cdot 0.001 \cdot 1 \cdot 2 \cdot 0.001 \cdot 8 \cdot 10^{11}}{2000 \cdot 220 \cdot 2}\right]$$

$$HFConst = 6443 \text{ (DEC)} \text{ atau } 0x192B \text{ (HEX)}$$

(e) Register PStart

Register ini berisi nilai yang dikeluarkan oleh register PowerPA ketika energi meter dialiri arus sebesar arus mula.

(f) Register QStart

Register ini berisi nilai yang dikeluarkan oleh register PowerQ ketika energi meter dialiri arus sebesar arus mula.

2. Kalibrasi daya aktif

Proses kalibrasi daya aktif dilakukan pada tiga kondisi dengan ketentuan yang telah dijelaskan pada bab sebelumnya. Ketiga kondisi tersebut bertujuan untuk mencari ketiga nilai konstanta kalibrasi dari daya aktif yaitu:

(a) Register GPQA

Dilakukan pada kondisi U_n , I_d , dan faktor daya 1. Formula mencari nilai yang akan diberi ke register tersebut dapat mengacu pada formula 2-6.

(b) Register PhsA

Dilakukan pada kondisi U_n , I_d , dan faktor daya 0.5 induktif. Formula mencari nilai yang akan diberi ke register tersebut dapat mengacu pada formula 2-8.

(c) Register APOSa

Dilakukan pada kondisi U_n dan tanpa beban dengan cara mencari nilainya sesuai dengan yang telah dijelaskan pada bab sebelumnya bagian kalibrasi daya aktif.

3. Kalibrasi daya reaktif

Untuk kalibrasi daya reaktif terdapat dua kondisi yang harus dilakukan untuk mencari dua nilai konstanta kalibrasi dari register kalibrasi daya reaktif.

(a) Register QPhsCal

Dilakukan pada kondisi U_n , I_d , dan faktor daya 0.5 induktif. Formula mencari nilai yang akan diberi ke register tersebut dapat mengacu pada formula 2-9.

(b) Register RPOSa

Mekanisme pencarian nilainya sama seperti cara mencari nilai register APOSa.

4. Kalibrasi RMS

Nilai yang dikeluarkan register tegangan, arus saluran A, daya aktif, daya reaktif, dan energi merupakan nilai yang mengalami perbesaran di dalam perhitungan internal IC. Oleh karena itu diperlukan konstanta pembagi agar nilai yang ditampilkan menjadi mendekati nilai aslinya dengan cara mencarri seperti yang telah dijelaskan pada bab sebelumnya. Selain itu apabila terdapat *offset* pada saluran arus A maka diperlukan pengaturan untuk mengatasi *offset* tersebut dengan memberi nilai pada IARMSOS. Jika dirangkum, pada bagian ini konstanta kalibrasi dan pembagi yang akan dicari adalah sebagai berikut:

- (a) Register IARMSOS
- (b) Konstanta konversi tegangan (Ku)
- (c) Konstanta konversi arus saluran A (Kia)
- (d) Konstanta konversi daya aktif (Kp)
- (e) Konstanta konversi daya reaktif (Kq)
- (f) Konstanta energi (Ke)

Jika melihat formula-formula yang diberikan dari *datasheet*, terdapat

variabel *error* yang perlu dicari nilainya. *Error* yang dimaksud bukanlah *error* relatif terhadap nilai referensi. Akan tetapi *error* tersebut diberi nama *error pulse* (ep) dengan cara perhitungan seperti pada formula 2-7.

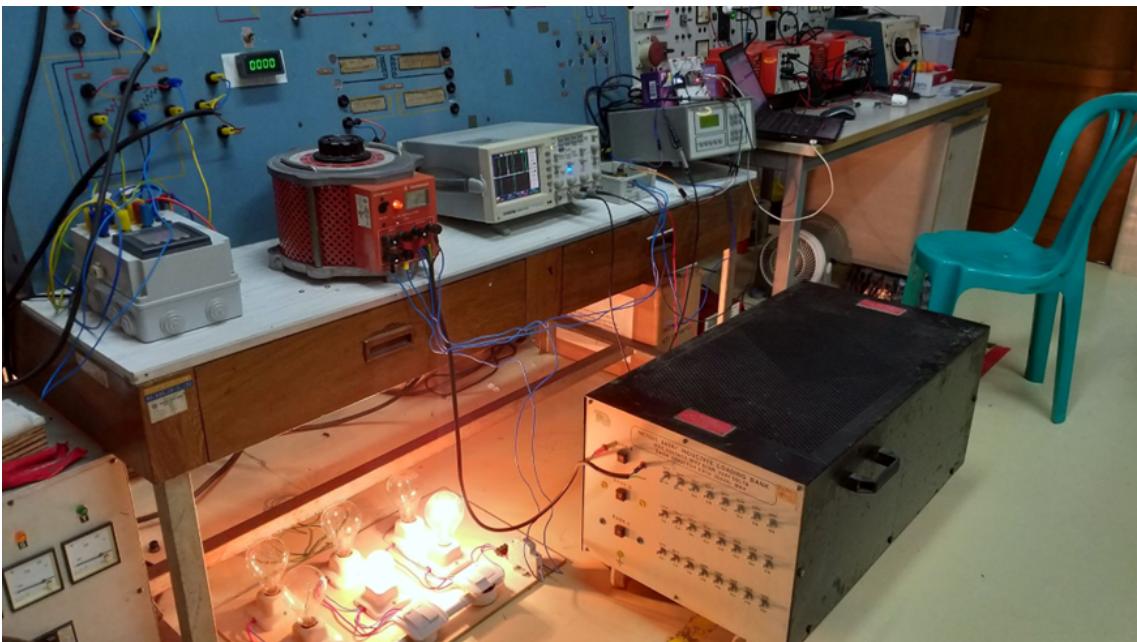


Gambar 3.3. Energi meter kalibrator pabrikan GFUVE

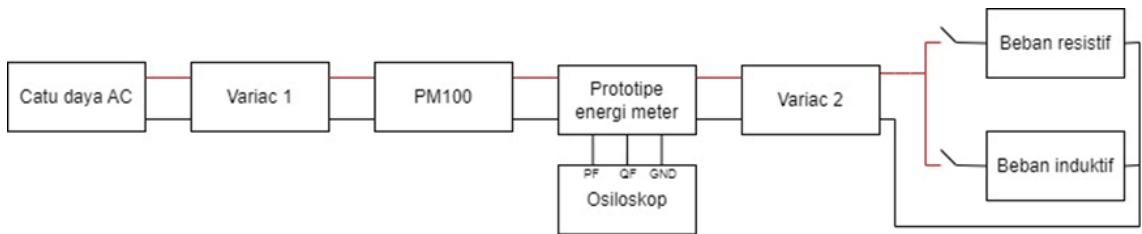
Pada proses kalibrasi untuk energi meter terdapat alat yang berfungsi khusus untuk mengkalibrasi energi meter yaitu energi meter kalibrator. Alat tersebut memiliki fitur yang disediakan khusus untuk kalibrasi. Seperti mengukur parameter energi listrik, menampilkan ep secara otomatis, dan lain-lain. Contoh tampilan dari energi meter kalibrator terdapat pada Gambar 3.3 [24].

Kalibrasi akan dilakukan di Laboratorium Sistem Tenaga Listrik DTETI FT UGM. Laboratorium tersebut belum memiliki alat khusus untuk melakukan kalibrasi energi meter. Untuk mengatasi hal tersebut, penulis berusaha agar kalibrasi tetap bisa dilakukan dengan cara menggunakan peralatan yang dapat mendekati alat energi meter kalibrator. Susunan peralatan yang digunakan untuk kalibrasi dan pengujian penulis sebut sebagai *testbench* yang tampilannya pada Gambar 3.4. *Testbench* tersebut bisa direpresentasikan dengan diagram blok di Gambar 3.5.

Catu daya AC digunakan untuk memberikan listrik AC yang terhubung langsung dengan Variac 1. Fungsi dari Variac 1 adalah sebagai pengatur besar dan kecilnya tegangan AC yang diberikan. PM100 merupakan *power meter analyzer* yang digunakan

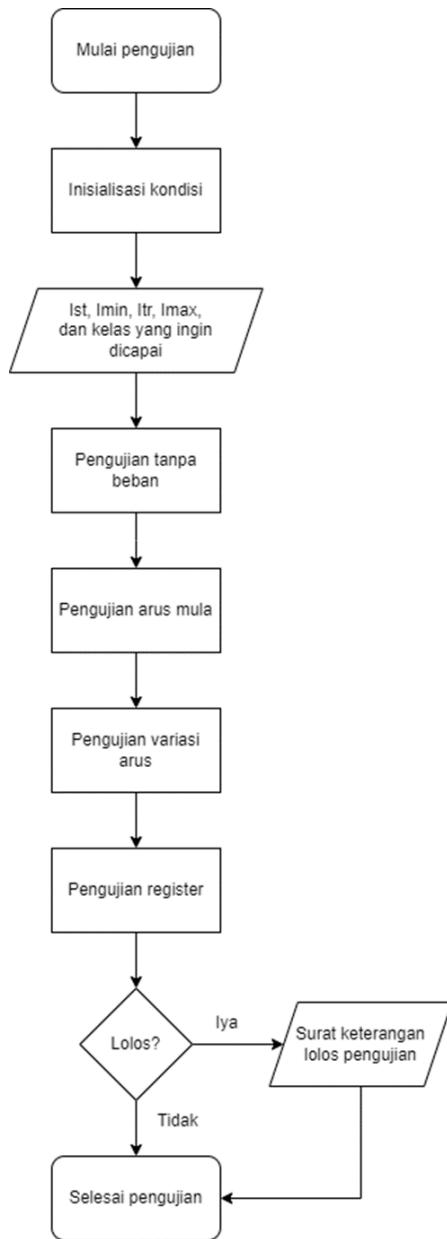


Gambar 3.4. *Testbench* pada laboratorium



Gambar 3.5. Diagram blok *testbench*

sebagai referensi pembacaan tegangan, arus, daya aktif, daya reaktif, energi aktif, energi reaktif, dan faktor daya. Setelah itu, barulah terhubung ke prototipe energi meter yang terhubung ke osiloskop untuk mengamati sinyal pulsa. Bagian selanjutnya adalah Variac 2 yang berfungsi sebagai pengatur arus yang akan masuk ke beban. Terdapat dua jenis beban yang digunakan dan disusun secara paralel yaitu beban resistif berupa lampu pijar dan beban induktif berupa induktor. Beban resistif tersebut memiliki daya maksimal sampai sekitar 850-900 W atau besar arus sekitar 4 A. Untuk beban induktor memiliki rating sampai 6000 VA namun penggunaannya hanya dibatasi sampai 750 VA. Dengan menggunakan rancangan *testbench* ini, semua kebutuhan untuk melakukan kalibrasi bisa dilakukan. Sementara PM100, sebagai referensi sudah memenuhi standar syarat akurasi untuk pengujian kWh meter kelas 1 (B) berdasarkan Surat Keputusan Direktur Jenderal Perlindungan Konsumen dan Tertib Niaga nomor 24 tahun 2021 tentang Syarat Teknis Standar Ukuran Metrologi Legal Besaran Energi Listrik [25].



Gambar 3.6. Diagram alir pengujian

3.2.3 Pengujian Prototipe Sistem

Proses pengujian dilakukan di Laboratorium Sistem Tenaga Listrik DTETI FT UGM dengan menggunakan *testbench* yang sama seperti proses kalibrasi. Pengujian dilakukan dengan mengacu pada standar yang telah ditetapkan pada Surat Keputusan Direktur Jenderal Perlindungan Konsumen dan Tertib Niaga Nomor 161 Tahun 2019 tentang Syarat Teknis Meter kWh. Secara garis besar, pengujian yang dilakukan dapat direpresentasikan dengan diagram alir pada Gambar 3.6. Berdasarkan tabel Batas Kesalahan yang Diizinkan, penulis akan mencoba mengembangkan sistem konsumsi energi listrik kelas 2 (A). Dengan menggunakan arus dasar (I_d) sebesar 2 A, maka besaran jenis arus lainnya adalah sebagai berikut:

1. $I_{st} : 0.05 \text{ A}$
2. $I_{min} : 0.1 \text{ A}$
3. $I_{tr} : 0.2 \text{ A}$
4. $I_{max} : 4 \text{ A}$

Arus mula yang digunakan berdasarkan tabel jenis arus seharusnya bernilai 0.004 A. Akan tetapi karena keterbatasan perlengkapan dari laboratorium, digunakan arus 0.05 A. Hal tersebut masih diperbolehkan berdasarkan standar yang berlaku selama besaran arus mula lebih kecil daripada arus minimum.

1. Pengujian tanpa beban
 - (a) Pengujian dilakukan pada tegangan dasar (U_n) dan tidak ada arus yang mengalir pada beban atau tanpa beban.
 - (b) Pada kondisi tersebut, energi meter tidak boleh menghasilkan satu pun pulsa.
 - (c) Pengujian dilakukan dengan waktu minimal yang ditentukan dengan formula berikut:

$$t = \frac{100 \cdot 10^3}{b \cdot EC \cdot m \cdot P_{min}}$$

$b = \text{batas kesalahan maksimum sesuai BKD pada } I_{min}$ (3-1)

$m = \text{jumlah perangkat}$

Dengan formula tersebut maka didapatkan waktu minimal pengujian dengan perhitungan sebagai berikut:

$$t = \frac{100 \cdot 10^3}{1 \cdot 2000 \cdot 1 \cdot 440} = 0.91 \text{ jam (54.6 menit)}$$

2. Pengujian arus mula
 - (a) Pengujian dilakukan pada U_n , I_{st} , dan faktor daya 1.
 - (b) Pada kondisi tersebut, batas kesalahan maksimal adalah 2.5%
 - (c) Waktu antara dua pulsa dihitung dengan formula berikut:

$$\tau = \frac{3.6 \cdot 10^6}{m \cdot EC \cdot P_{st}} \quad (3-2)$$

3. Pengujian variasi arus

- (a) Pengujian dilakukan pada U_n dengan titik uji wajib variasi arus dan faktor daya seperti pada daftar berikut:
 - i. I_{min} dengan faktor daya 1
 - ii. I_{tr} dengan faktor daya 1
 - iii. I_{tr} dengan faktor daya 0.5 induktif
 - iv. I_d dengan faktor daya 1
 - v. I_d dengan faktor daya 0.5 induktif
 - vi. I_{max} dengan faktor daya 1
 - vii. I_{max} dengan faktor daya 0.5 induktif
- (b) Pada kondisi-kondisi tersebut, batas kesalahan maksimal sesuai pada tabel 2.2.

4. Pengujian register

- (a) Pengujian dilakukan pada U_n dan I_d .
- (b) Pada kondisi tersebut, batas kesalahan maksimal adalah 0.2%.
- (c) Menggunakan metode komparator yaitu menghitung *error* relatif antara pembacaan IC Renergy 8209 (register) dan referensi.

3.2.4 Pengembangan *Firmware Final*

Seperti yang sudah dijelaskan pada bagian pengembangan *firmware* kalibrasi dan pengujian, *firmware* final memiliki fitur yang berbeda. *Firmware* final ditujukan untuk kenyamanan pengguna yang akan menggunakan sistem *monitoring* konsumsi listrik yang dikembangkan. Beberapa fitur yang telah dikembangkan pada *firmware* kalibrasi dan pengujian tetap dipertahankan dengan sedikit dimodifikasi untuk disesuaikan dengan kebutuhan pengguna. Fitur-fitur yang akan dikembangkan pada *firmware* adalah sebagai berikut:

1. Fitur yang diterapkan kembali

Fitur komunikasi UART, dan memberi nilai register tidak mengalami perubahan dari yang sudah dikembangkan pada *firmware* kalibrasi dan pengujian. Untuk fitur membaca nilai register, terdapat modifikasi. Pada *firmware* final jumlah register yang dibaca disesuaikan dengan kebutuhan yaitu register yang menyimpan nilai tegangan, arus, daya aktif, dan energi aktif. Sementara pada fitur inisialisasi konfigurasi, alamat register yang digunakan tetap sama. Akan tetapi

data yang akan diberikan pada alamat register terkait adalah data yang didapatkan ketika melakukan kalibrasi IC Renery 8209.

2. Tampilan OLED.

Tampilan pada OLED sedikit berbeda dengan *firmware* kalibrasi dan pengujian. Tampilan OLED pada *firmware* kalibrasi dan pengujian menampilkan semua data yang dibaca untuk memudahkan pengamatan. Sementara pada tampilan OLED *firmware* final tujuannya adalah kemudahan pengguna untuk membaca data-data. Sehingga untuk kepentingan tersebut, ukuran karakter dibuat lebih besar dan hanya ada satu data yang tertampil pada satu layar OLED beserta keterangannya. Untuk mengganti data yang ingin dilihat dapat menekan tombol yang telah disediakan. Data-data yang akan ditampilkan adalah sebagai berikut:

- (a) Energi aktif (kWh)
- (b) Estimasi biaya tagihan listrik (Rp)
- (c) Arus (A)
- (d) Tegangan (V)
- (e) Daya aktif (W)

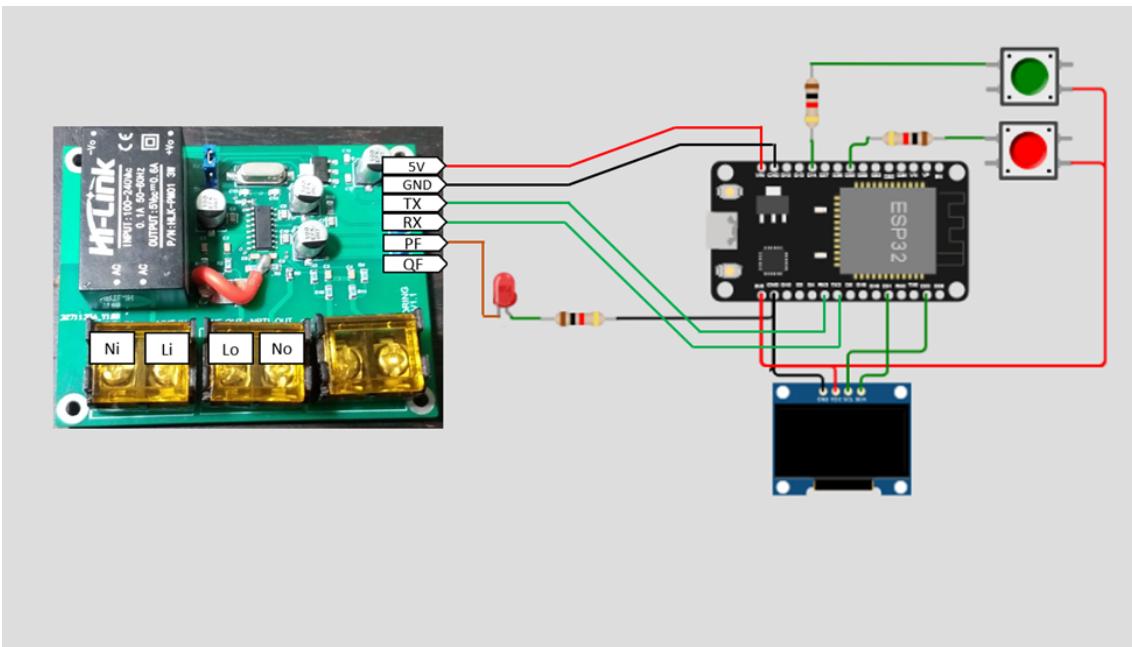
3. Interupsi

Fungsi interupsi akan terkoneksi dengan *push button*. Apabila tombol tersebut ditekan maka rutin interupsi akan dijalankan. Terdapat dua jenis interupsi yang akan dikembangkan. Interupsi pertama berfungsi untuk mengubah tampilan data pada OLED. Interupsi lainnya berfungsi sebagai tombol reset untuk penggunaan energi listrik dan estimasi biaya. Dua interupsi tersebut direpresentasikan dengan dua tombol *pushbutton*

4. Penyimpanan data pada EEPROM

Fitur penyimpanan data pada EEPROM adalah langkah antisipasi apabila tidak suplai daya ke MCU atau kondisi listrik padam. EEPROM nantinya akan menyimpan pembacaan terakhir yang dilakukan MCU agar pada saat suplai daya kembali mengalir, perhitungan energi listrik dan estimasi biaya tidak mengulang dari nol. Apabila pengguna ingin mengulang perhitungan energi listrik dan estimasi biaya, dilakukan dengan menekan tombol reset. Sehingga dengan fitur ini, keputusan melakukan reset perhitungan energi berada pada kendali penuh pengguna.

Gambar 3.7 merupakan *wiring diagram* dari prototipe sistem *monitoring* konsumsi listrik yang akan diberikan *firmware* final. Perbedaan *wiring* dari yang



Gambar 3.7. *Wiring diagram IC Renergy 8209 final*

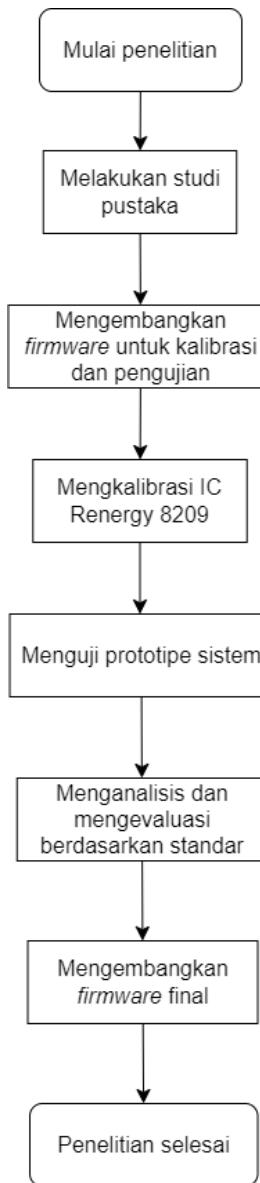
sebelumnya adalah terdapat penambahan dua *push button* yang berfungsi sebagai pengganti tampilan dan melakukan reset. Selain itu, pin PF dan QF kini tidak lagi terhubung ke osiloskop. Melainkan akan terhubung ke LED untuk memberi tahu apabila terdapat penambahan pulsa energi maka LED akan berkedip. Selain dari pin PF yang sekarang terhubung ke LED, konfigurasi koneksi antara rangkaian IC Renergy 8209 dan ESP32 sama dengan prototipe sebelumnya. Konfigurasi koneksi antara OLED dan ESP32 juga sama.

3.3 Alur Tugas Akhir

Dalam mengerjakan penelitian dan melakukan pengembangan *firmware* sistem *monitoring* konsumsi listrik menggunakan IC Renergy 8209 terdapat alur yang telah penulis buat. Alur penelitian ini dibuat setelah melakukan studi pustaka. Sehingga didapatkan alur penelitian yang dilakukan seperti diagram alur yang terdapat pada Gambar 3.8. Urutan dari penelitian yang akan dilakukan adalah studi pustaka, mengembangkan *firmware* untuk kalibrasi dan pengujian, mengkalibrasi IC Renergy 8209, menguji IC Renergy 8209 dan pengambilan data, menganalisis data dan mengevaluasi berdasarkan standar yang telah ditentukan, dan mengembangkan *firmware* final. Penjelasan lebih lanjut dari setiap tahapan akan dijabarkan selanjutnya.

1. Studi pustaka

Sebelum memulai lebih jauh, penulis melakukan studi pustaka terlebih dahulu. Penulis mengawali dengan mempelajari *datasheet* dari IC Renergy 8209. Bagian yang perlu dipahami antara lain adalah spesifikasi dan fitur, tata cara



Gambar 3.8. Diagram alir penelitian

komunikasi dengan IC, cara mengkonfigurasi, prosedur kalibrasi, kegunaan setiap alamat register, dan lain-lain. Setelah itu penulis mencari penelitian serupa yang telah dilakukan sebelumnya oleh orang lain. Dengan mempelajari penelitian serupa, penulis mendapat gambaran *firmware* dan sistem yang akan dikembangkan seperti apa. Termasuk juga kelebihan yang dapat diterapkan dan kekurangan yang dapat diantisipasi dan dikembangkan lebih baik. Kemudian penulis mulai mempelajari tentang pengembangan *firmware* secara umum dan prinsip-prinsip dalam melakukannya.

2. Mengembangkan *firmware* untuk kalibrasi dan pengujian

Setelah penulis sudah merasa cukup dengan pengetahuan yang didapatkan saat studi pustaka, *firmware* untuk kalibrasi dan pengujian mulai dikembangkan.

Pada saat pengembangan ini, penulis juga mencoba beberapa konfigurasi dari IC Renergy 8209 untuk melihat bagaimana respon IC ketika konfigurasi itu diterapkan. Untuk memastikan segala fitur yang dikembangkan pada sisi *firmware* telah bekerja dengan baik, penulis mencoba menjalankan prototipe sistem *monitoring* konsumsi listrik di Laboratorium Sistem Tenaga Listrik DTETI FT UGM. Apabila fitur-fitur yang dikembangkan pada *firmware* prototipe sistem *monitoring* konsumsi listrik telah bekerja dengan baik maka proses kalibrasi bisa dijalankan.

3. Mengkalibrasi IC Renergy 8209

Proses kalibrasi diawali dengan memastikan seluruh kabel dan koneksi telah terpasang dengan baik untuk meminimalisir risiko karena daya yang dipakai cukup tinggi dan bisa menyebabkan kecelakaan kerja. Setelah memeriksa keamanan, prosedur kalibrasi mulai dijalankan. Prosedur tersebut dijalankan secara sekuensial sesuai dengan tata cara yang terdapat pada *datasheet*. Setiap nilai konstanta kalibrasi ditemukan, nilai tersebut langsung dimasukkan ke *firmware* dan MCU direset. Setiap data yang ditemukan akan dicatat agar dapat terdokumentasi dengan baik. Pencatatan tersebut dilakukan dengan bantuan *google sheet*.

4. Menguji prototipe sistem

Pengujian yang dilakukan akan mengikuti pedoman dari Surat Keputusan Direktur Jenderal Perlindungan Konsumen dan Tertib Niaga Nomor 161 Tahun 2019 tentang Syarat Teknis Meter kWh. Dokumen tersebut telah menjabarkan secara lengkap tentang panduan pengujian energi meter. Penulis akan membuat lembar uji yang mengacu pada lembar uji yang telah disediakan pada dokumen tersebut. Data pengujian kemudian akan dicatat pada *google sheet* untuk dianalisis dan dievaluasi.

5. Menganalisis dan mengevaluasi berdasarkan standar

Setelah dilakukan pengujian dan data sudah dikumpulkan, hasil pengujian dan data yang terkumpul akan dianalisis dan dievaluasi. Analisis dilakukan untuk melihat apakah prototipe sistem *monitoring* konsumsi listrik telah berfungsi dengan tepat. Sementara evaluasi berdasarkan standar dilakukan untuk menilai prototipe sistem *monitoring* konsumsi listrik telah memenuhi persyaratan dari standar tersebut. Selain itu evaluasi berdasarkan standar juga untuk menentukan tingkat akurasi dari prototipe sistem.

6. Mengembangkan *firmware* final

Seperti yang sudah dijelaskan, *firmware* final akan berfokus pada fitur-fitur yang ditujukan untuk pengguna sistem *monitoring* konsumsi listrik. Sementara

firmware kalibrasi dan pengujian ditujukan untuk kalibrator dan penguji untuk memudahkan mereka dalam melakukan kalibrasi dan pengujian sistem *monitoring* konsumsi listrik. Setelah semua fitur yang dikembangkan pada *firmware* final bekerja dengan baik, penelitian ini bisa dikatakan selesai karena sistem telah dikalibrasi, diuji, dan terbukti dapat bekerja dengan baik.

3.4 Keterbatasan Penelitian

Pengembangan *firmware* sistem *monitoring* konsumsi listrik menggunakan IC Renergy 8209 pada platform ESP32 dilakukan sebisa mungkin mendekati kondisi ideal karena mencapai kondisi ideal dengan segala keterbatasan yang ada merupakan hal yang mustahil. Terdapat beberapa keterbatasan dalam pengembangan ini.

Keterbatasan pertama telah disebutkan pada bab ini yaitu kalibrasi dilakukan tanpa menggunakan alat untuk kalibrasi energi meter sesungguhnya yaitu energi meter kalibrator. Kondisi tersebut membuat penulis mencoba mencari cara agar fungsi dari energi meter kalibrator bisa tercapai dengan menggunakan alat yang tersedia pada Laboratorium Sistem Tenaga Listrik DTETI FT UGM. Hal tersebut mempengaruhi tingkat ketelitian.

Keterbatasan kedua adalah dinamika pada catu daya dan beban. Pada penelitian ini diinginkan catu daya yang dapat memberikan tegangan 220 V secara konstan namun hal tersebut sulit untuk bisa dilakukan karena tegangan yang diberikan PLN bersifat cukup dinamis. Mengatasi hal tersebut, penulis menggunakan variac untuk mengatur keluaran tegangan catu daya. Akan tetapi pendekatan tersebut juga tidak memberikan garansi tegangan akan konstan 220 V dan dimungkinkan deviasi masih terjadi namun lebih terminimalisir. Dari sisi beban resistif, arus yang dihasilkan juga tidak bisa konstan karena deviasi tegangan yang terjadi dan panas yang dihasilkan akibat penggunaan dalam jangka waktu tertentu akan membuat arus menjadi dinamis. Selain itu terkadang terjadi deviasi pada frekuensi listrik dan hal tersebut mempengaruhi deviasi pada sisi beban juga. Untuk mengatasi hal tersebut, penulis menggunakan variac lainnya untuk mengatur besar kecilnya arus agar sesuai yang diharapkan.

Keterbatasan ketiga adalah ketelitian dari osiloskop. Dalam mengamati pulsa yang keluar dari IC Renergy 8209, digunakan osiloskop. Resolusi waktu dapat dibesar-kecilkan sesuai dengan kebutuhan namun apabila resolusi dibesarkan maka ketelitian dari osiloskop memburuk dan berlaku sebaliknya. Pada kasus skripsi ini, waktu yang dibutuhkan untuk melakukan pengujian arus mula mencapai 82 detik. Tentu dengan lama waktu tersebut ketelitian dari osiloskop menjadi semakin buruk dan hal tersebut mempengaruhi perhitungan kesalahan. Selain itu pada kondisi tegangan dasar, arus dasar, dan faktor daya 1, berdasarkan perhitungan, periode pulsa adalah sekitar 3.9 detik. Dengan waktu tersebut, *time/div* yang digunakan pada osiloskop adalah 0.5 detik.

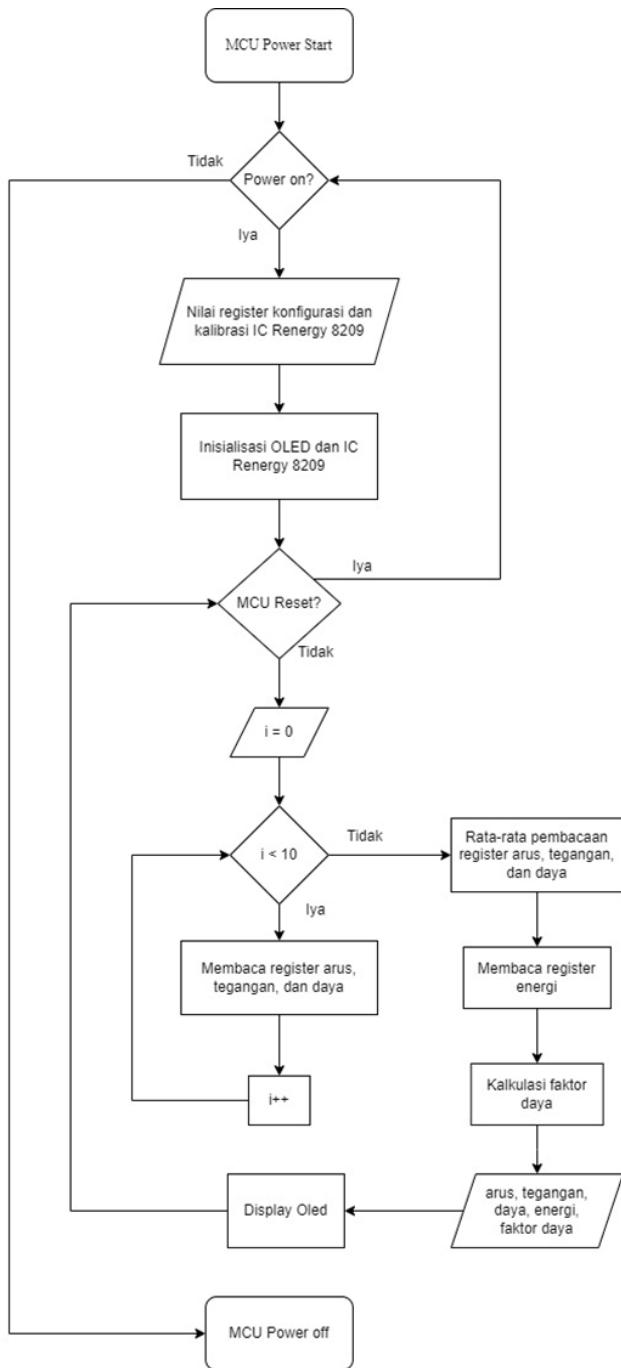
Pada konfigurasi resolusi tersebut, kursor dari osiloskop memberikan *step* 0.02 detik. Hal tersebut akan mempengaruhi nilai *error pulse*. Prototipe ini akan berikut pada nilai kesalahan kurang dari 5% bahkan kurang dari 1% dengan nilai parameter yang kecil. Sehingga penyimpangan sebesar 0.02 akan cukup mempengaruhi perhitungan kesalahan.

Keterbatasan keempat adalah menghasilkan faktor daya yang diinginkan. Untuk mengatur faktor daya dengan besar arus yang diinginkan dilakukan kombinasi antara beban resistif dan induktif dan pengaturan arus dengan menggunakan variac. Beban resistif dan induktif yang digunakan bukan sebuah beban yang dapat diatur nilainya dengan jangkauan tertentu dan memiliki *step* yang kecil. Melainkan beban yang digunakan adalah lampu pijar dan induktor yang nilai hambatan dan induktansinya bernilai tetap. Hal tersebut akan berpengaruh terhadap nilai faktor daya yang muncul. Misalnya ketika diinginkan faktor daya bernilai 0.5, pengaturan beban yang dilakukan mungkin tidak bisa menghasilkan nilai 0.5 melainkan nilainya disekitar 0.5, bisa kurang atau bisa lebih.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pembahasan Hasil *Firmware* Kalibrasi dan Pengujian



Gambar 4.1. Diagram alir *firmware* kalibrasi dan pengujian

Firmware sistem *monitoring* konsumsi listrik menggunakan IC Renergy 8209 pada platform ESP32 dirancang menggunakan bahasa pemrograman arduino dengan Arduino IDE. *Firmware* ini dirancang untuk kebutuhan kalibrasi dan pengujian sehingga

yang menjadi sasaran pengguna adalah kalibrator dan penguji. Apabila *firmware* diubah, tidak akan mempengaruhi karakteristik pembacaan dari metering IC Renergy 8209 karena karakteristik IC ini ditentukan oleh nilai register konfigurasi dan konstanta kalibrasi terkait.

Secara garis besar, cara kerja *firmware* kalibrasi dan pengujian yang dirancang dapat direpresentasikan dengan diagram alir pada Gambar 4.1. Untuk mulai menjalankan *firmware* tersebut, ESP32 perlu diberi catu daya terlebih dahulu agar mulai bekerja. Kemudian nilai-nilai register konfigurasi dan konstanta kalibrasi akan dimasukkan dengan cara mengubah nilai pada kode program di Arduino IDE. Agar tidak tertukar antara alamat register satu dan lainnya, kode program diberi keterangan alamat dan nama registernya.

Nilai-nilai register konfigurasi dan konstanta kalibrasi kemudian dikirim oleh ESP32 ke IC Renergy 8209 dengan menggunakan protokol komunikasi UART. Langkah tersebut merupakan inisialisasi dari IC Renergy 8209 agar sesuai dengan karakteristik yang diinginkan. Selain itu, tampilan OLED juga perlu diinisialisasi sesuai dengan ketentuan dari OLED yang dipakai. Untuk memudahkan proses komunikasi dengan tampilan OLED, digunakan *library* untuk OLED tersebut yaitu Adafruit SH110G dan Adafruit GFX. Tampilan OLED menggunakan protokol komunikasi SPI untuk mengirim data. Bagian tersebut dalam struktur *firmware* termasuk dalam *setup* artinya perintah yang terdapat pada bagian ini hanya dijalankan sekali yaitu ketika ESP32 mengalami interupsi reset.

Blok-blok selanjutnya adalah proses yang akan dilakukan secara berulang-ulang atau *looping* sampai ESP32 dilakukan interupsi reset atau catu daya dimatikan. Proses pertama pada *looping* adalah membaca nilai register arus, tegangan, dan daya aktif. Setiap pembacaan nilai tersebut dilakukan sebanyak 10 kali kemudian nilainya dirata-rata. Metode tersebut biasa disebut sebagai *avarage sum*. Hal ini dilakukan karena setiap ESP32 meminta nilai-nilai register terkait, nilai register tersebut mengalami deviasi atau tidak selalu sama. Sehingga *avarage sum* dilakukan untuk mencari nilai tengah dari pembacaan yang dilakukan.

Proses pembacaan energi tidak menggunakan metode *avarage sum* karena register energi pada IC Renergy 8209 merupakan sebuah register *counter* yang menghitung sudah berapa banyak pulsa yang dihasilkan. IC Renergy 8209 tidak mengkalkulasikan faktor daya, melainkan hanya bisa mendeteksi pergeseran fase yang terjadi. Sehingga untuk menentukan faktor daya perlu dilakukan kalkulasi pada sisi kode program dengan mencari nilai arccos dari pembagian antara daya aktif dan arus dikali tegangan.

Data-data pembacaan arus, tegangan, daya aktif, energi aktif, dan faktor daya

kemudian disimpan dalam variabel masing-masingnya. Untuk menampilkan data-data tersebut pada OLED, variabel kemudian dipanggil dengan konfigurasi tampilan OLED yang telah ditentukan pada kode program. Setelah itu proses berulang lagi ke *avarage sum* pembacaan nilai register arus, tegangan, dan daya aktif sampai ESP32 diberi perintah reset atau catu daya dimatikan.

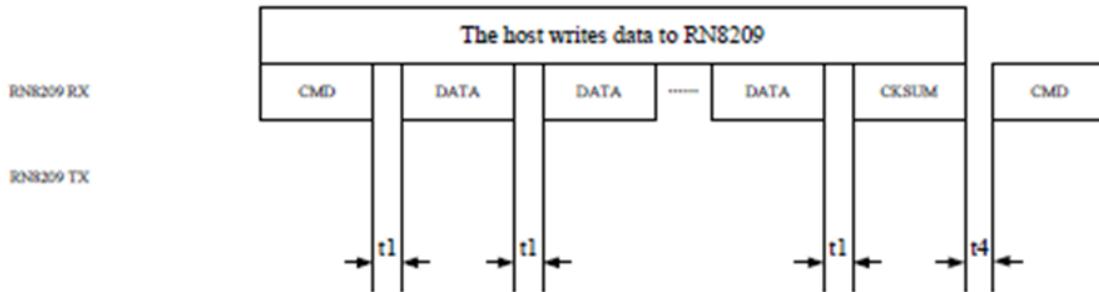
4.1.1 Fitur Komunikasi UART dengan IC Renergy 8209

Fitur komunikasi UART dengan IC Renergy berguna untuk melakukan komunikasi UART dengan IC Renergy 8209 sesuai dengan tata cara yang telah ditetapkan oleh pabrikan. Fitur ini diawali dengan deklarasi *setup* pin TX dan RX ESP32 dan *baud rate* yang digunakan. Pada penelitian ini digunakan pin TX pada GPIO17 dan pin RX pada GPIO16. Sementara *baud rate* bersifat fixed pada 4800 bps karena RN8209C hanya menggunakan *baud rate* tersebut. Fitur ini terdiri dari dua fungsi yaitu untuk mengirim data *host* ke *slave* melalui pin TX ESP32 (uart_TX) dan menerima data dari *slave* ke *host* melalui pin RX ESP32 (uart_RX) yang secara garis besar dapat direpresentasikan oleh Algorithm 1.

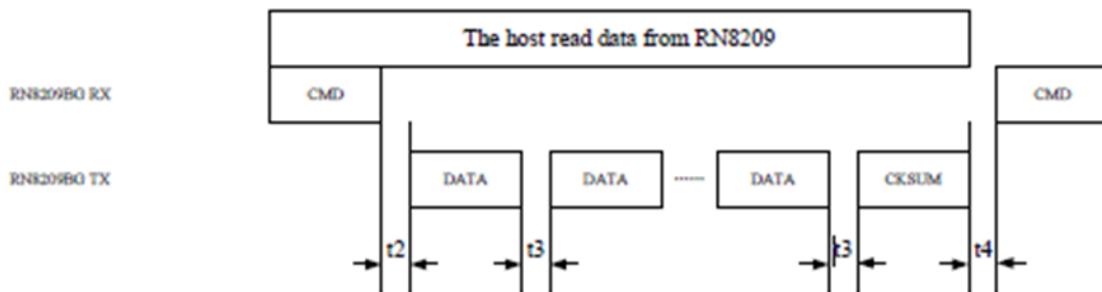
Algorithm 1 Komunikasi UART dengan IC

```
1: function UART_TX(data, len = len(data))
2:   for int i = 0; i < len+2; i++ do
3:     Serial2.write(datain[i])
4:     delay(10);
5:   end for
6: end function
7:
8: function UART_RX(reg address, buffer)
9:   int i=0
10:  Serial.write(reg address)
11:  while Serial2.available() > 0 do
12:    buffer[i] = Serial2.read()
13:    delay(10)
14:  end while
15: end function
```

Fungsi pertama adalah mengirim data dari *host* ke *slave* melalui pin TX ESP32 dengan tata cara seperti Gambar 4.2 [15]. Fungsi ini membutuhkan dua argumen yaitu *array* data yang ingin dikirim dan panjang *array* data tersebut. Bahasa pemrograman arduino telah memiliki perintah untuk melakukan transmisi data yaitu *Serial.write*. Perintah tersebut akan mengirimkan data melalui pin TX namun data tersebut perlu dicacah menjadi 8 bit atau 1 *byte*. Mengatasi hal tersebut, data disimpan dalam suatu *array data frame* yang telah dimasukkan sebagai argumen. Perintah *Serial.write* kemudian dipanggil secara berulang sebanyak panjang dari *array data frame*. Setiap memanggil perintah *Serial.write* diberi tunggu sebesar 10 ms.



Gambar 4.2. Tata cara pengiriman data dengan UART



Gambar 4.3. Tata cara menerima data dengan UART

Fungsi kedua adalah menerima data *slave* ke *host* melalui pin RX ESP32 dengan tata cara seperti Gambar 4.3 [15]. Fungsi ini memerlukan dua argumen yaitu alamat register yang dituju dan *array data frame* untuk menyimpan data yang diterima dari IC Renergy 8209. Untuk menerima data, *host* perlu mengirim perintah terlebih dahulu ke *slave*. Perintah tersebut merupakan data 8 bit yang terdiri 1 bit perintah menulis atau membaca register dan 7 bit alamat register yang dituju. Dalam fungsi ini bit perintah bernilai 0 atau perintah untuk membaca nilai yang terdapat pada register terkait.

Data yang diterima dari IC Renergy 8209 tidak langsung masuk ke alamat variabel yang telah dibuat. Namun ESP32 memiliki alamat penyimpanan yang berfungsi sebagai ruang singgah data yang diterima melalui pin RX. Mekanisme seperti ini meminimalisir data terpotong pada saat proses transmisi dari *slave* ke *host*. Untuk memeriksa apakah ada data yang diterima oleh ESP32, dipanggil perintah *Serial.available*. Perintah tersebut menjadi argumen dari fungsi *while*. Artinya selama terdapat data pada ruang singgah maka perintah di dalam *while* akan diberikan. Perintah yang digunakan untuk memindahkan data dari ruang singgah tersebut ke alamat *array data frame* adalah *Serial.read*. Ketika *Serial.read* dipanggil data dari ruang singgah akan berpindah ke alamat yang baru sehingga ruangan tersebut kosong dan siap menerima data selanjutnya.

Terdapat penjelasan pada pemilihan waktu tunggu sebesar 10 ms. Pemilihan ini berasal dari percobaan penulis ketika mencari waktu tunggu paling minimal yang

```

Read 0 : 82
Read 1 : 9
Read 2 : 0
Read 3 : F5
End

Reactive Energy :
Read 0 : 0
Read 1 : 0
Read 2 : 0
Read 3 : D6
End

Device ID :
Read 0 : 0
Read 1 : 0
Read 2 : 0
Read 3 : D3
End

Active Energy :
Read 0 : 82
Read 1 : 9
Read 2 : 0
Read 3 : F5
End

```

Gambar 4.4. Masalah pada waktu tunggu kurang dari 10 ms

dibutuhkan dalam melakukan transmisi dan menerima data ke dan dari IC Renergy 8209. Ketika waktu tunggu kurang dari 10 ms terjadi pergeseran data dalam tampilan serial monitor seperti pada Gambar 4.4.

Pada awalnya penulis tidak memberikan waktu tunggu antar perintah transmisi atau menerima data. Fenomena yang terjadi adalah nilai yang tertampil pada bagian "Active Energy" seharusnya berada pada bagian "Device ID". Nilai yang tertampil pada bagian "Reactive Energy" seharusnya berada pada bagian "Active Energy". Sementara nilai yang tertampil pada bagian "Device ID" seharusnya berada pada bagian "Reactive Energy". Kemudian penulis menambahkan waktu tunggu 1 ms dan masalah tersebut bisa terselesaikan untuk transmisi dan menerima data dari tiga nilai register.

Ketika ditambahkan lebih banyak lagi data yang harus ditransmisi dan diterima, fenomena tersebut kembali terjadi. Kemudian penulis menambahkan terus waktu tunggu hingga 10 ms. Pada waktu tunggu tersebut akhirnya komunikasi UART dapat berjalan lancar dan fenomena tersebut tidak terjadi kembali.

4.1.2 Fitur Membaca Nilai Register

Pada fitur membaca nilai register tidak semua alamat register akan dibaca. Fitur ini berfokus pada alamat-alamat register yang berisi hasil pembacaan IC Renergy 8209 yang merupakan parameter-parameter dari energi listrik. Parameter tersebut antara lain tegangan, arus, daya aktif, dan energi aktif. Setiap pembacaan parameter dibuat satu fungsi sendiri seperti pada Algorithm 2.

Fungsi pertama adalah membaca nilai register tegangan (read_voltage). Tegangan yang terbaca merupakan tegangan RMS yang tersedia pada register URMS dengan alamat 0x24. Untuk mendapatkan nilai tersebut, dipanggil fungsi mengirim data pada fitur komunikasi UART dengan argumen alamat register adalah 0x24. Alamat register tersebut memiliki ukuran 3 *byte* yang dalam pengiriman datanya tercacah cacah menjadi 3 bagian. Nilai yang tercacah kemudian digabungkan dan dimasukkan pada variabel dengan tipe data uint_32t yang memiliki ukuran 4 *byte*. Dengan begitu data yang diterima kini telah menjadi utuh. Operasi tersebut kemudian diulang sebanyak 10 kali untuk dilakukan metode *avarage sum*. Hasil rerata masih berupa nilai *raw* dari register yang sangat besar akibat perbesaran pada internal IC sehingga perlu dibagi dengan scaling constant Ku. Nilai tegangan kemudian dijadikan sebagai nilai *return* dari fungsi ini.

Fungsi kedua adalah membaca nilai register arus (read_current). Arus yang terbaca merupakan arus RMS yang tersedia pada register IARMS dengan alamat 0x23. Urutan perintah dari fungsi ini secara garis besar sama seperti pada saat membaca nilai register tegangan. Perbedaan keduanya terdapat pada alamat register dan nilai konstanta pembagi yang digunakan.

Fungsi ketiga adalah membaca nilai register daya aktif (read_power). Hasil pembacaan daya aktif dari IC Renergy 8209 disimpan pada register PowerPA dengan alamat 0x26. Alamat register PowerPA berukuran 4 *byte* dan nilai yang disimpan merupakan nilai komplemen dari pembacaan yang sebenarnya. Dalam melakukan pembacaan, urutan operasi yang dilakukan sama seperti pada fungsi membaca nilai register tegangan namun setiap data yang telah diterima dan disimpan pada variabel bertipe data uint32_t, nilainya perlu dikomplemenkan terlebih dahulu. Setelah itu barulah dilakukan *avarage sum*, dibagi dengan konstanta pembagi daya aktif, dan dijadikan nilai *return* dari fungsi ini.

Fungsi keempat adalah membaca nilai register energi aktif (read_energy). Data energi aktif yang terhitung diambil dari register EnergyP dengan alamat 0x29 yang memiliki ukuran 3 *byte*. Seperti yang sudah dijelaskan sebelumnya, register ini merupakan *counter* energi aktif yang menghitung pulsa yang telah dikeluarkan oleh IC Renergy 8209. Setiap pulsa tersebut merepresentasikan banyaknya energi aktif yang

Algorithm 2 Membaca Nilai Register

```
1: function READ_VOLTAGE_OR_CURRENT
2:     uint8_t databuf[4];
3:     uint32_t temp[10],total=0;
4:     float num;
5:     for int i=0;i<10;i++ do
6:         uart_rx(REG_URMS or REG_IARMS,databuf);
7:         temp[i] = (databuf[0]<<16) + (databuf[1]<<8) + (databuf[2]);
8:     end for
9:     for int i=0; i<10; i++ do
10:        total += temp[i];
11:    end for
12:    total /= 10;
13:    num = (float)total/konstanta_pembagi; return num;
14: end function
15:
16: function READ_POWER
17:     uint8_t databuf[5];
18:     uint32_t temp[10],total=0;
19:     double num;
20:     for int i=0;i<10;i++ do
21:         uart_rx(REG_PowerQ,databuf);
22:         temp[i] = (databuf[0]<<24) + (databuf[1]<<16) + (databuf[2]<<8) + (databuf[3]);
23:     end for
24:     for int i=0; i<10; i++ do
25:        total += temp[i];
26:     end for
27:     if total&0x80000000 then
28:         total = total;
29:         total += 1;
30:     end if
31:     total /= 10;
32:     num = (float)total/konstanta_pembagi; return num;
33: end function
34:
35: function READ_ENERGY
36:     uint8_t databuf[4];
37:     uint32_t temp;
38:     float num;
39:     uart_rx(REG_EnergyP,databuf);
40:     temp = (databuf[0]<<16) + (databuf[1]<<8) + (databuf[2]);
41:     num = (float)temp/konstanta_pembagi; return num;
42: end function
```

telah digunakan dan representasi nilai per pulsanya didapatkan dengan membagi EC dengan 1000. Selain itu untuk fungsi ini tidak digunakan metode *avarage sum*, hanya perlu dipanggil fungsi menerima data dengan alamat tersebut dan ukuran data 3 byte kemudian nilai register dibagi dengan konstanta pembagi dan dijadikan nilai *return* dari fungsi ini.

4.1.3 Fitur Memberi Nilai Register

Fungsi memberi nilai register akan berhubungan dengan fitur inisialisasi konfigurasi IC. Fitur ini yang akan melakukan perintah pengiriman data ke IC Renergy 8209 dan memeriksa apakah data tersebut telah tersimpan pada alamat yang dituju. Apabila data belum tersimpan, fungsi ini akan secara otomatis mengirim ulang data sampai 5 kali dan jika masih belum berhasil juga akan mengirimkan pesan kalau data tidak berhasil terkirim. Secara garis besar algoritma dari fitur ini seperti *pseudocode* pada Algorithm 3.

Pada fitur ini terdapat *write buffer* sebagai ruang singgah data yang akan dikirim ke IC Renergy 8209. Argumen alamat akan diperiksa apakah alamat tersebut merupakan register perintah spesial atau bukan. Register perintah spesial tersebut memiliki beberapa perintah seperti membuka kunci register dan mengunci kembali register tersebut agar nilai yang berada didalamnya tidak berubah. Apabila register tersebut bukan register perintah spesial, bit terbesar akan ditambah 1 sebagai representasi perintah untuk memberi nilai pada register. Namun jika register tersebut adalah register perintah spesial maka operasi tersebut tidak perlu dilakukan. Kemudian nilai itu dimasukkan pada *write buffer* pada index 0.

Selanjutnya argumen *data buffer* dimasukkan pada *write buffer* pada indeks-indeks selanjutnya sejumlah ukuran dari *data buffer*. Nilai alamat register beserta data-datanya kemudian dilakukan operasi perhitungan CKSUM dan hasil perhitungannya dijadikan sebagai nilai pada *write buffer* indeks terakhir sebagai penanda akhir dari paket data yang akan dikirimkan.

Untuk memeriksa register perintah spesial telah diterima oleh IC Renergy 8209, ESP32 akan membaca nilai dari alamat register SysStatus. Apabila bit kedua bernilai satu artinya register terkunci, jika bernilai nol artinya kunci register terbuka. Sementara memeriksa pengiriman data pada register lainnya dilakukan dengan membaca nilai yang tersimpan pada alamat nilai register tersebut. Kemudian data yang dikirim dibandingkan dengan data yang diterima, apabila tidak sama maka terdapat pesan pada data ke berapa yang tidak terkirim dan melakukan pengiriman ulang sampai 5 kali. Jika sudah data berhasil terkirim sebelum lima kali, pesan yang diterima adalah “Data berhasil terkirim” namun jika masih belum berhasil juga, maka pesan yang diterima adalah “Data gagal terkirim”.

Algorithm 3 Memberi Nilai Register

```
1: function WRITE REGISTER(reg address, data buffer, len = len(data buffer))
2:   if reg address == 0xEA then
3:     write buffer[0] = reg address
4:   else
5:     write buffer[0] = reg address + 0x80
6:   end if
7:
8:   CKSUM += write buffer
9:
10:  for i = 1; i < len; i++ do
11:    write buffer[i] = data buffer[i]
12:    CKSUM += data buffer[i]
13:  end for
14:
15:  write buffer[len data+1] = ~CKSUM
16:  write write buffer to IC Renergy
17:
18:  if reg address == 0xEA then
19:    read SysStatus register and store at read buffer
20:    temp = read buffer[0]
21:    if data buffer == 0xE5 then
22:      if !(temp and 0x10) then
23:        print "Write still protected"
24:      end if
25:    else
26:      if data buffer == 0xDC then
27:        if temp and 0x10 then
28:          print "Write still unprotected"
29:        end if
30:      end if
31:    end if
32:  else
33:    for i = 0; i < 5; i++ do
34:      for j = 0; j < len; j++ do
35:        read reg address and store at read buffer
36:        if reg address == SysCon and j == 0 then
37:          temp = 0
38:        end if
39:        if temp not equal to data buffer[j+1] then
40:          print "data i failed to write"
41:        else
42:          print "Write success"
43:          j = 4
44:        end if
45:      end for
46:    end for
47:  end if
48: end function
```

Tabel 4.1. Inisialisasi konfigurasi

No	Nama Register	Alamat Register	Data yang dikirim	Keterangan
1	Perintah spesial	0xEA	0xE5	Membuka kunci register
2	Perintah spesial	0xEA	0xFA	Soft reset
3	Perintah spesial	0xEA	0x5A	Set <i>channel</i> arus A
4	SysCon	0x00	0x1602	Konfigurasi sistem
5	EMUCon	0x01	0x2803	Konfigurasi perhitungan 1
6	EMUCon2	0x17	0x0000	Konfigurasi perhitungan 2
7	HFCConst	0x02	0x192B	Pengatur EC
8	Pstart	0x03	0x0000	<i>No-load threshold</i> daya aktif
9	GPQA	0x05	0x0000	Konstanta kalibrasi <i>gain</i>
10	PhsA	0x07	0x00	Konstanta kalibrasi fase
11	IARMSOS	0x0E	0x0000	Konstanta kalibrasi <i>offset</i> arus channel A
12	APOSA	0x0A	0x0000	Konstanta kalibrasi <i>offset</i> daya aktif
13	Qstart	0x04	0x0000	<i>No-load threshold</i> daya reaktif
14	QPhsCal	0x09	0x0000	Konstanta kalibrasi fase daya reaktif
15	RPOSA	0x0C	0x0000	Konstanta kalibrasi <i>offset</i> daya reaktif
16	Perintah spesial	0xEA	0xDC	Mengunci register konfigurasi dan kalibrasi

4.1.4 Fitur Inisialisasi Konfigurasi

Fitur inisialisasi konfigurasi IC termasuk dalam struktur *setup* pada *firmware* yang dikembangkan artinya hanya sekali dijalankan pada saat ESP32 dilakukan interupsi reset. Fitur ini berisi alamat-alamat register konfigurasi dan kalibrasi. Urutan dan daftar alamat tersebut tersedia pada tabel 4.1. Secara garis besar, kode program dapat direpresentasikan pada algorithm 4.

Algorithm 4 Inisialisasi Konfigurasi

```

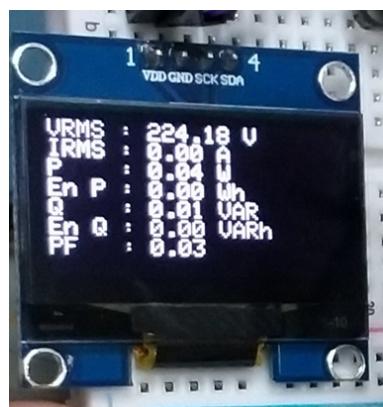
1: function INIT
2:   uint8_t rBuf[6]
3:   int i = 0
4:   rBuf[i++] = 0x00
5:   uint8_t checksum = 0
6:   delay(10)
7:   rBuf[i++] = data[i]
8:   ...
9:   uart_write(reg address, rBuf, len = len(rBuf))
10: end function

```

Urutan dari register perintah spesial dan yang dikirim tidak bisa ditukar karena data-data tersebut merepresentasikan suatu perintah yang telah disebutkan pada keterangan. Inisialisasi diawali dengan membuka kunci register. Kemudian dilakukan *soft* reset, seperti yang dianjurkan pada *datasheet* IC ini. Urutan ketiga merupakan perintah untuk menetapkan kalau arus *channel* A yang akan digunakan. Urutan-urutan selanjutnya merupakan alamat register konfigurasi dan kalibrasi, urutan ini boleh

ditukar-tukar. Data yang dikirim untuk register SysCon, EMUCon, EMUCon2, dan HFConst sudah ditetapkan dari awal karena register tersebut merupakan register konfigurasi. Sementara alamat register sisanya merupakan alamat register kalibrasi yang nilainya baru akan ditemukan ketika proses kalibrasi dilakukan dan nilai defaultnya diatur 0. Pada setiap alamat register yang diinisialisasi diberikan keterangan alamat register agak tidak tertukar satu sama pada saat kalibrator memberi nilai konstanta kalibrasi.

4.1.5 Fitur Tampilan OLED



Gambar 4.5. Tampilan OLED kalibrasi dan pengujian

Fitur tampilan pada OLED mengatur data apa saja yang ditampilkan dan bentuk tampilannya akan seperti apa. Fitur ini menjadi jembatan antara pembacaan yang dilakukan oleh IC Renergy 8209 kepada kalibrator dan penguji. Tampilan yang dikembangkan seperti pada Gambar di atas. Data utama yang ditampilkan adalah tegangan RMS (VRMS), arus RMS (IRMS), daya aktif (P), energi aktif (En P), dan faktor daya (PF). IC Renergy 8209 juga dapat menghitung daya reaktif (Q) dan energi reaktif (En Q). Untuk *firmware* kalibrasi dan pengujian, data tersebut ditampilkan sebagai keperluan untuk memeriksa apakah IC ini dapat membaca daya reaktif dan energi reaktif mendekati nilai sebenarnya.

Fitur ini diawali dengan memanggil fungsi dari fitur membaca nilai register terkait data yang ingin ditampilkan. Setiap nilai *return* dari fungsi tersebut kemudian disimpan pada variabel yang telah dideklarasikan dan diberi nama sesuai dengan datanya. Variabel-variabel tersebut nanti akan dipanggil untuk mengisi nilai pada tampilan OLED.

Agar data tersebut bisa tertampil, terdapat tata cara yang harus diikuti. Pertama berikan perintah *clearDisplay* untuk menghapus tampilan sebelumnya dan menggantinya dengan yang baru. Kemudian ukuran dan warna teks diatur dengan fungsi *setTextSize* dan *setTextColor*. Posisi dari teks diatur dengan perintah *setCursor*. Perintah-perintah selanjutnya adalah teks yang ingin ditampilkan. Hal tersebut dilakukan dengan perintah

Algorithm 5 Tampilan OLED

```
1: function OLED
2:   display.clearDisplay
3:   display.textDisplay
4:   display.setTextColor
5:   display.setCursor
6:   display.print(text)
7:   ...
8:   display.display
9: end function
```

print. Urutan terakhir dari tata cara tampilan adalah memanggil perintah *display* untuk menampilkan tampilan yang sudah diatur ke OLED. Secara garis besar kode program dapat direpresentasikan pada Algorithm 5.

Firmware kalibrasi dan pengujian dirancang khusus untuk keperluan kalibrasi dan pengujian dengan sasaran pengguna adalah pengembang dan kalibrator. Fitur-fitur yang dikembangkan pada *firmware* ini dibuat untuk kemudahan pengembang dan kalibrator mengakses fitur dari IC Renergy 8209. Selain itu, *firmware* ini dikembangkan agar IC Renergy 8209 dapat dikonfigurasikan menuju karakteristik yang diinginkan. Pembacaan parameter dan perhitungan energi listrik ditampilkan untuk keperluan evaluasi kesalahan yang terjadi agar pengembang dapat mengoreksi kesalahan tersebut dan nilainya masih di bawah standar yang diinginkan.

4.2 Pembahasan Hasil Kalibrasi IC Renergy 8209

Saat tahap pengembangan firmware kalibrasi dan pengujian, firmware tersebut sudah diuji fungsionalitasnya. Hal tersebut agar saat tahap kalibrasi dilakukan, tidak ada kendala berarti. Tahap kalibrasi ini dilakukan secara sekuensial dan antar bagiannya tidak boleh ditukar. Pada setiap bagian kalibrasi dilakukan pada kondisi yang berbeda-beda, sesuai yang telah dijelaskan pada bab sebelumnya. Penjelasan lebih lanjut pada bagian-bagian selanjutnya.

4.2.1 Konfigurasi Sistem

Kalibrasi diawali dengan mengkonfigurasi IC Renergy 8209 untuk mengatur karakteristik yang diinginkan dari IC tersebut. Pengaturan konfigurasi dilakukan dengan mengatur nilai dari enam register konfigurasi yaitu SysCon, EMUCon, EMUCon2, HFConst, Pstart, dan Qstart. Untuk empat register pertama telah ditentukan nilainya dari awal.

1. Register SysCon

Cara menguji register ini telah bekerja atau belum adalah dengan melihat

hasil pembacaan dari register pembacaan. Hal itu dilakukan untuk melihat apakah *gain* yang telah diatur sudah benar terpilih. Pada pengaturan *gain* arus terdapat pilihan 2, 4, 8, dan 16 untuk mengatur blok *programmable gain amplifier* (PGA). Sementara untuk *gain* tegangan pilihannya ada tiga yaitu 2, 4, dan 8.

Dalam melakukan pengaturan *gain* tersebut, penulis melakukan percobaan untuk melihat pembacaan tegangan, arus, dan daya aktif dari masing-masing *gain* tersebut. Percobaan dilakukan dengan memvariasikan tegangan dari 180 V sampai 230 V dengan step 10 V dan variasi beban dari 4 lampu pijar dengan spesifikasi 150 W.

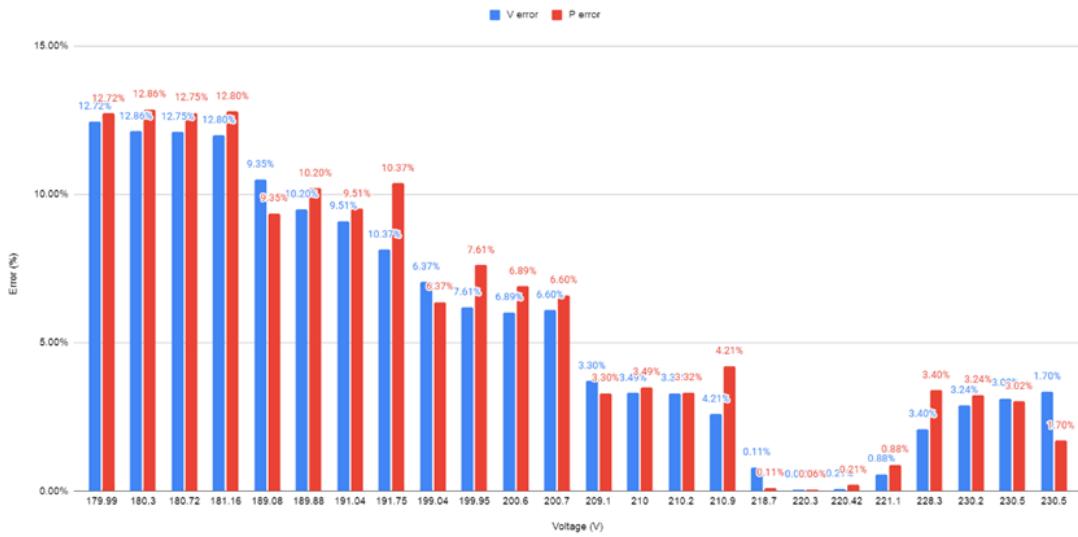
<i>Gain</i>		Kesalahan relatif pembacaan		
<i>Gain U</i>	<i>Gain Ch A</i>	V	I	P
1	1	0.09%	0.31%	0.24%
1	2	0.11%	0.29%	0.27%
1	8	0.08%	0.18%	0.16%
1	16	0.15%	0.38%	0.32%
2	1	0.07%	0.29%	0.29%
2	2	0.07%	0.29%	0.21%
2	8	0.07%	0.46%	0.46%
2	16	0.10%	0.32%	0.35%
4	1	5.68%	0.30%	6.09%
4	2	5.14%	0.28%	5.91%
4	8	5.68%	0.47%	6.57%
4	16	6.15%	0.22%	6.48%

Gambar 4.6. *Error* kombinasi *gain*

Gambar 4.6 merupakan hasil percobaan untuk melihat korelasi antara kombinasi *gain* tegangan dan arus dengan rata-rata kesalahan relatif pembacaan tegangan, arus, dan daya aktif. Dari percobaan tersebut terjadi fenomena ketika *gain* tegangan ditetapkan sebesar 4, kesalahan pembacaan tegangan dan daya aktif meningkat jika dibandingkan dengan *gain* tegangan lainnya. Distribusi kesalahan pembacaan tegangan ketika tegangan divariasikan pada *gain* tegangan 4 bisa dilihat pada Gambar 4.7.

Semakin menjauh dari nilai 220 V, kesalahan pembacaan semakin meningkat. Hanya pada nilai tegangan 220 V kesalahan pembacaan tegangan bernilai relatif kecil. hal tersebut dikarenakan dalam kalibrasi dan pengujian ini digunakan tegangan dasar yaitu 220 V dan konstanta pembagi dicari saat kondisi tersebut. Fenomena ini menunjukkan ketika *gain* tegangan bernilai 4, register URMS menghasilkan nilai yang jangkauan pembacannya kecil. Fenomena ini juga mematahkan asumsi penulis yaitu semakin besar *gain* maka resolusi pembacaan akan semakin baik dan kesalahan pembacaan akan semakin kecil.

V and P error at Gain U = 4



Gambar 4.7. Distribusi kesalahan pembacaan

Pengaturan *gain* lainnya memiliki distribusi kesalahan ketika tegangan dan arus divariasikan yang hampir mirip. Sehingga dapat dianggap semua *gain* akan menghasilkan pembacaan yang hampir serupa dan penulis memilih kombinasi *gain* tegangan bernilai 1 dan *gain* arus bernilai 8. Dengan konfigurasi tersebut, nilai yang harus diberi untuk alamat register SysCon adalah 0x1602 (HEX).

2. Register EMUCon

Register ini mengatur beberapa karakteristik dan pengaturan yang dipilih sudah dijelaskan pada bab sebelumnya. Pengaktifan register energi agar mengakumulasi pembacaan berdampak pada respon register EnergyP dan EnergyQ dalam menghitung energi. Apabila akumulasi energi dinonaktifkan, setiap alamat register ini diambil nilainya oleh ESP32, maka nilai yang berada pada alamat register ini kembali menjadi nol. Sementara jika diaktifkan, alamat register ini akan terus mengakumulasi energi walau nilainya terus menerus diambil dan akan kembali menjadi nol apabila IC direset.

Mode akumulasi energi berguna untuk mengatur apabila nilai energi aktif dan reaktif memiliki tanda negatif, maka nilai tersebut akan dimutlakkan menjadi positif agar tidak mengurangi akumulasi yang telah dilakukan. Hal tersebut dilakukan karena pada implementasinya, perhitungan pada energi meter yang beredar tidak mengizinkan pengurangan perhitungan energi.

High-pass filter pada saluran arus A dan tegangan diaktifkan untuk mengeliminasi derau pada frekuensi rendah. Hal ini juga sekaligus sebagai cara untuk meminimalisir *offset*. Apabila pengaturan ini dinonaktifkan, nilai

pembacaan menjadi lebih besar dan mengurangi konsistensi pembacaan.

Pengaturan pulsa energi aktif dan reaktif mengatur pin PF dan QF dari IC Renergy 8209. Apabila dinonaktifkan, pin tersebut tidak akan menghasilkan pulsa yang merepresentasikan pertambahan energi. Kedua pin tersebut berperan dalam proses kalibrasi untuk menghitung *error pulse* yang dihasilkan dan nilai tersebut akan berguna dalam menentukan nilai konstanta kalibrasi.

3. Register EMUCon2

Pada register ini hanya terdapat satu pengaturan yang diperlukan yaitu koreksi fase lanjut pada saluran arus A. Hal ini dilakukan karena koreksi tersebut sebenarnya sudah dilakukan oleh register PhsA yang berguna untuk mengkalibrasi fase. Selain itu koreksi fasa pada register ini bernilai hanya 0.01 derajat sehingga koreksi tersebut tidak terlalu berdampak signifikan. Akan tetapi pada penggunaan yang membutuhkan akurasi tinggi, pengaturan ini mungkin diperlukan.

4. Register HFConst

Dengan kombinasi gain tegangan sebesar 1, gain arus sebesar 8, rasio rangkaian pembagi tegangan 0.001, dan EC 2000 imp/kWh, maka nilai yang perlu diberikan untuk register HFConst adalah 0x3256 (HEX). Apabila nilainya dibiarkan default, dari percobaan yang penulis lakukan, nilai tersebut akan mengatur EC mendekati 1000 imp/kWh.

Register HFConst berperan untuk mengarahkan IC Renergy 8209 untuk mendekati target EC yang diinginkan. Hal tersebut diketahui ketika penulis melakukan percobaan dengan memberikan tegangan dan arus pada saat sebelum IC dikalibrasi. Didapatkan daya sebesar 446.5 W dengan waktu antara pulsa berdasarkan formula adalah 4.031 sekon. Pada kondisi tersebut, IC Renergy 8209 mengeluarkan pulsa dengan jarak sekitar 3.9 sekon. Artinya *error pulse* yang terjadi adalah 3.368%. Nilai tersebut sudah cukup kecil namun masih belum cukup untuk kelas energi yang ingin dicapai.

5. PStart dan QStart

Nilai kedua register ini dicari ketika IC Renergy 8209 diberikan arus mula pada tegangan dasar. Hal itu dilakukan agar pada saat implementasinya nanti, IC baru akan menghitung energi ketika nilai arus lebih dari arus mula yang telah ditetapkan, yang merupakan fungsi dari register ini. Nama umum untuk fungsi register ini adalah *no-load threshold*.

Pada saat IC diberikan arus mula, nilai yang terbaca oleh register PowerPA adalah sekitar 1500 (DEC) atau 0x5DC (HEX) dan nilai tersebut yang dimasukkan ke dalam register Pstart dan Qstart. Dengan nilai tersebut harapannya, IC baru

mulai menghitung energi ketika daya aktif bernilai lebih 10 W.

Pada saat dilakukan pengujian arus mula, tidak ada pulsa energi yang dikeluarkan dari IC ini. Hal tersebut menjadikan IC ini pada saat itu tidak lolos pengujian arus mula. Kemudian penulis mengurangi nilai dari register Pstart dan pulsa energi masih tidak juga keluar. Akhirnya penulis memilih untuk membuat alamat register ini bernilai nol.

4.2.2 Kalibrasi Daya Aktif

Terdapat tiga nilai konstanta kalibrasi yang harus dicari pada proses kalibrasi daya aktif. Ketiga nilai tersebut dicari dengan kondisi yang berbeda-beda seperti yang telah dijelaskan pada bab sebelumnya. Kemudian ketiga nilai tersebut akan dimasukkan ke dalam register kalibrasi daya aktif terkait. Bagian ini menjelaskan lebih lanjut hasil kalibrasi daya aktif yang telah dilakukan.

1. Register GPQA

Pada saat melakukan konfigurasi HFConst, telah dilakukan percobaan untuk mengetahui efek dari konfigurasi tersebut. Diketahui bahwa *error pulse* yang terjadi adalah sebesar 3.368%. Nilai *error pulse* tersebut yang akan digunakan untuk menghitung nilai konstanta kalibrasi gain berdasarkan formula yang sudah dijelaskan sebelumnya. Perhitungan nilai konstanta kalibrasi gain akan menjadi seperti berikut:

$$P_{gain} = \frac{-3.368\%}{1 + 3.368\%} = -0.03258$$

$$GPQA = \text{int}[(-0.03258 \cdot 32768) + 65536]$$

$$GPQA = 64468 (\text{DEC}) \text{ atau } 0xFBD4 (\text{HEX})$$

Dari hasil perhitungan tersebut didapatkan nilai konstanta kalibrasi *gain* bernilai 64468 dalam basis desimal atau 0xFBD4 dalam basis heksadesimal. Nilai tersebut yang akan dimasukkan ke dalam alamat register GPQA. Caranya dengan mengubah nilai register GPQA pada fitur inisialisasi konfigurasi.

Setelah alamat register GPQA telah terisi dengan nilai tersebut, penulis melakukan percobaan ulang untuk melihat dampak yang ditimbulkan oleh register ini. Percobaan dilakukan dengan memberikan arus dan tegangan dasar. Didapatkan daya aktif sebesar 447.6 W, artinya diperkirakan waktu antara dua pulsa adalah sebesar 4.02145 sekon. Sementara pengukuran yang dilakukan pada osiloskop menunjukkan IC Renergy 8209 membutuhkan waktu 4.02 sekon di

antara dua pulsa. Maka *error pulse* yang terjadi setelah nilai konstanta kalibrasi *gain* adalah sebesar 0.04%. Penulis menduga *error pulse* yang terjadi bisa saja lebih kecil daripada itu namun karena keterbatasan resolusi dari osiloskop yaitu *step* waktu hanya 0.02 sekon sehingga didapatkan nilai tersebut.

Dari percobaan tersebut dapat diketahui bahwa alamat register GPQA dapat mempercepat atau memperlambat keluarnya pulsa dari IC Renergy 8209.

2. Register PhsA

Untuk proses ini, kondisi faktor daya diubah menjadi 0.5 induktif pada arus dan tegangan dasar. Sebelum melakukan perhitungan, dicari terlebih dahulu *error pulse* yang terjadi pada kondisi ini. Didapatkan daya aktif sebesar 210.8 W pada kondisi tersebut, artinya diperkirakan waktu antara dua pulsa adalah sebesar 8.5389 sekon. Sementara waktu antara dua pulsa yang dikeluarkan oleh IC Renergy 8209 dan terukur pada osiloskop adalah 8.48 sekon. Maka *error pulse* yang terjadi pada kondisi ini adalah sebesar 0.69%. Jika dilihat nilai *error pulse* tersebut, terdapat peningkatan kesalahan dari kondisi tegangan dan arus dasar pada faktor daya bernilai 1 setelah nilai register GPQA dimasukkan.

Dari nilai *error pulse* dapat dilakukan perhitungan untuk mencari nilai konstanta kalibrasi fase PhsA. Perhitungannya sebagai berikut:

$$\theta = \arcsin\left(\frac{-0.69\%}{\sqrt{3}}\right) = -0.22976^\circ$$

$$PhsA = \text{int}\left[\frac{256 - 0.22976}{0.02}\right]$$

$$PhsA = 244 (\text{DEC}) \text{ atau } 0xF4 (\text{HEX})$$

Nilai yang didapatkan dari perhitungan tersebut adalah 244 dalam basis desimal atau 0xF4 dalam basis heksadesimal. Nilai tersebut akan dimasukkan ke dalam alamat register PhsA. Caranya adalah dengan mengubah nilai alamat register Phs A pada fitur inisialisasi konfigurasi.

Setelah nilai konstanta kalibrasi PhsA dimasukkan, dilakukan percobaan untuk melihat dampak yang diberikan setelah alamat register PhsA diberi nilai sesuai dengan perhitungan yang telah dilakukan. Percobaan dilakukan pada tegangan dan arus dasar dengan faktor daya bernilai 0.5. Dengan pengaturan kondisi tersebut didapatkan daya aktif sebesar 217 W artinya diperkirakan waktu antara dua pulsa terjadi dalam waktu 8.2949 sekon. Sementara waktu antara dua pulsa yang dikeluarkan oleh IC Renergy 8209 dan terukur oleh osiloskop adalah sebesar 8.3 sekon. Maka *error pulse* yang terjadi setelah alamat register PhsA

diisi nilai konstanta kalibrasi sesuai perhitungan adalah sebesar -0.06%.

Dari proses ini terdapat dua hal yang dapat diketahui. Pertama, ketika kondisi faktor daya tidak bernilai satu, terjadi pertambahan *error pulse*. Alamat register PhsA dapat meminimalisir *error pulse* yang terjadi dengan perhitungan yang tepat. Hal tersebut merupakan hal kedua yang diketahui dari proses ini.

3. Register APOSA

Proses mencari nilai register APOSA dilakukan dalam kondisi tanpa beban. Pada kondisi ini seharusnya nilai pembacaan bernilai nol. Akan tetapi apabila IC Renergy 8209 tetap menghasilkan nilai pembacaan daya aktif walau kondisi tanpa beban, selama nilai tersebut masih di bawah 1000, maka proses ini bisa dilewati.

Pada penelitian kali ini, nilai pembacaan daya aktif yang dikeluarkan oleh IC Renergy 8209 dengan kondisi tanpa beban adalah sekitar 500. Maka nilai konstanta register APOSA dibuat nol dan nilai tersebut yang dimasukkan ke register tersebut dalam fitur inisialisasi konfigurasi.

4.2.3 Kalibrasi Daya Reaktif

Proses kalibrasi daya reaktif bertujuan untuk meminimalisir kesalahan pembacaan daya reaktif dan energi reaktif oleh IC Renergy 8209. Kedua parameter tersebut pada penggunaannya nanti tidak diperlukan karena energi meter yang dirancang pada penelitian kali ini ditujukan untuk skala rumah tangga. Pada energi meter skala rumah tangga, energi reaktif tidak masuk ke dalam perhitungan tarif listrik melainkan hanya energi aktifnya saja. Hal tersebut karena pada penggunaan skala rumah tangga, beban yang bersifat induktif atau kapasitif tidak begitu dominan sehingga energi reaktif yang dihasilkan tidak banyak. Dengan kondisi seperti itu, kalibrasi daya reaktif sebenarnya bisa dilewatkan. Akan tetapi untuk keperluan penelitian, kalibrasi daya reaktif tetap dilakukan.

1. Register QPhsCal

Alamat register pertama yang perlu dicari nilainya dalam proses kalibrasi daya reaktif adalah register QPhsCal. Untuk mencari nilai konstanta kalibrasi fase daya reaktif, kondisi yang perlu diatur adalah pada tegangan dasar dan arus maksimal dengan faktor daya bernilai 0.5.

Pin yang berfungsi untuk mengeluarkan pulsa energi reaktif berbeda dengan energi aktif. Pulsa energi reaktif dikeluarkan melalui pin QF sementara energi aktif melalui pin PF. Oleh karena itu sebelum dilakukan percobaan, *probe* osiloskop perlu dipindah ke pin QF dan diamati apakah koneksi tersebut sudah terkoneksi dengan baik atau belum.

Pada kondisi yang telah disebutkan di atas didapatkan daya reaktif sebesar 734 VAR dan faktor daya 0.527, artinya diperkirakan waktu antara dua pulsa akan keluar dalam waktu 2.4523 sekon. Sementara waktu antara dua pulsa yang dikeluarkan oleh IC Renergy 8209 dan terukur pada osiloskop adalah 2.48 sekon. Dengan kedua nilai tersebut, *error pulse* yang terjadi adalah sebesar -1.12%. Nilai *error pulse* tersebut dipakai untuk perhitungan nilai konstanta kalibrasi fase daya reaktif dengan formula yang telah dijelaskan pada bab sebelumnya. Maka perhitungannya adalah sebagai berikut:

$$\theta = \frac{-1.12\%}{\cot(\arccos(0.527))} = -0.018$$

$$QPhsCal = \text{int}[-0.018 \cdot 0.5774 + 65536]$$

$$PhsA = 64946 (\text{DEC}) \text{ atau } 0xFBD2 (\text{HEX})$$

Dari perhitungan didapatkan nilai konstanta kalibrasi daya reaktif sebesar 64946 dalam basis desimal atau 0xFBD2 dalam basis heksadesimal. Nilai tersebut kemudian dimasukkan ke alamat register QPhsCal dengan cara mengubah nilai data pada register tersebut di fitur inisialisasi konfigurasi.

Selanjutnya untuk melihat dampak dari pemberian nilai ke alamat register QPhsCal, dilakukan percobaan yang sama dengan sebelumnya. Kali ini didapatkan daya reaktif sebesar 737.7 VAR, artinya diperkirakan waktu antara dua pulsa adalah 2.44 sekon. Sementara waktu antara dua pulsa yang dikeluarkan oleh IC Renergy 8209 dan terukur oleh osiloskop adalah 2.42 sekon. Maka didapatkan *error pulse* bernilai 0.83 %. Terjadi perbaikan *error pulse* setelah nilai konstanta kalibrasi daya reaktif dimasukkan ke dalam alamat register QPhsCal.

2. Register APOS

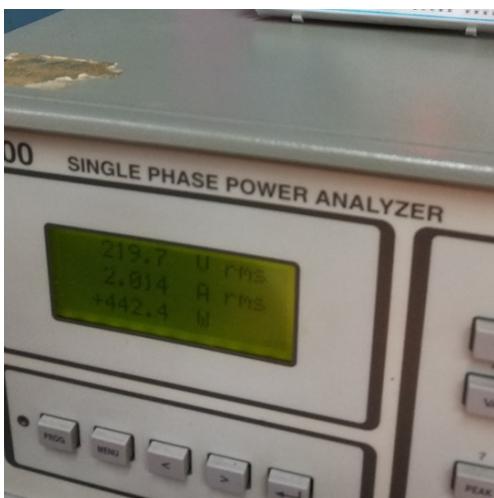
Mencari nilai konstanta *offset* daya reaktif sama caranya seperti mencari nilai konstanta *offset* daya aktif. Dilakukan pada kondisi yang juga sama, yaitu pada kondisi tanpa beban. Apabila nilai pembacaan bernilai kurang dari 1000, maka bagian ini bisa diabaikan. Ketika diberikan kondisi tersebut, pembacaan yang dikeluarkan oleh register daya reaktif bernilai sekitar 500. Artinya bagian ini dapat diabaikan dan nilai yang diberikan untuk alamat register RPOSA adalah 0.

4.2.4 Kalibrasi RMS

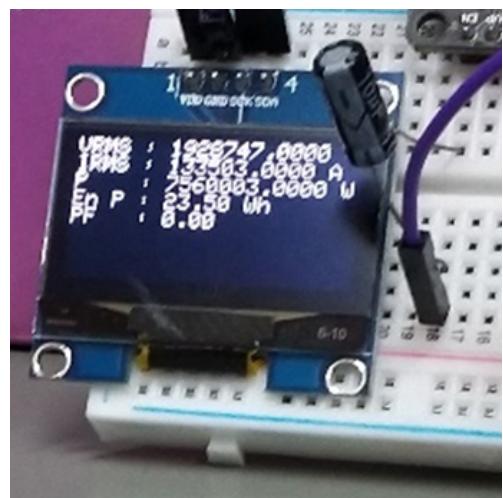
Pada proses kalibrasi RMS, terdapat dua bagian yang perlu dilakukan dengan tujuannya masing-masing. Dua bagian tersebut dilakukan pada kondisi yang berbeda. Bagian pertama adalah mencari nilai konstanta *offset* arus *channel A* dan yang kedua

adalah mencari nilai scaling constant untuk pembacaan tegangan, arus, daya aktif, dan energi aktif.

Mencari nilai konstanta *offset* arus *channel A* dilakukan pada kondisi tanpa beban. Bagian ini bersifat opsional atau boleh dilewatkan. Apabila pembacaan arus *channel A* pada kondisi tanpa beban bernilai kurang dari 1000, maka bagian ini bisa diabaikan. Pada penelitian kali ini, IC Renergy 8209 mengeluarkan nilai pembacaan arus *channel A* pada kondisi tanpa beban disekitar angka 500. Oleh karena itu penulis memilih memberikan nilai nol untuk konstanta *offset* arus *channel A* yang akan dimasukkan ke alamat register IARMSOS.



(a) Tampilan referensi



(b) Tampilan OLED

Gambar 4.8. Mencari konstanta pembagi

Selanjutnya adalah mencari konstanta pembagi pembacaan arus, tegangan, daya aktif, dan energi aktif. Hal ini dilakukan karena hasil pembacaan yang dikeluarkan oleh register terkait nilainya sangat besar karena mengalami perbesaran internal pada saat IC melakukan komputasi. Semakin besar gain yang dipilih maka nilai yang dikeluarkan register pembacaan akan semakin tinggi. Oleh karena itu dibutuhkan suatu konstanta untuk membagi nilai yang besar tersebut ke dalam jangkauan nilai yang sebenarnya.

Untuk mencari nilai tersebut, atur kondisi menjadi tegangan dan arus dasar dengan faktor daya bernilai 1 seperti pada Gambar 4.8. Kemudian catat nilai yang terbaca pada referensi dan yang dikeluarkan oleh IC Renergy 8209. Pada penelitian kali ini, referensi menunjukkan pembacaan tegangan RMS 219.7 V, arus RMS 2.014 A, dan daya aktif 442.4 W seperti pada Gambar 4.8. Sementara IC Renergy 8209 menunjukkan pembacaan tegangan RMS 1928747, arus RMS 133503, dan daya aktif 7560003. Kemudian nilai pembacaan IC dibagi dengan nilai dari referensi. Maka didapatkan konstanta pembagi tegangan RMS 8779, arus RMS 66287, dan daya aktif 17089. Sementara konstanta pembagi energi aktif didapatkan dengan membagi EC dengan

Tabel 4.2. Alamat register kalibrasi dan konstantanya

No	Nama Register	Alamat Register	Data yang dikirim
1	SysCon	0x00	0x1602
2	EMUCon	0x01	0x2803
3	EMUCon2	0x17	0x0000
4	HFCConst	0x02	0x192B
5	Pstart	0x03	0x0000
6	GPQA	0x05	0xFBD4
7	PhsA	0x07	0xF4
8	IARMSOS	0x0E	0x0000
9	APOSA	0x0A	0x0000
10	Qstart	0x04	0x0000
11	QPhsCal	0x09	0xFBD2
12	RPOSA	0x0C	0x0000
13	Kia	-	66287 (DEC)
14	Ku	-	8779 (DEC)
15	Kp	-	17089 (DEC)
16	Ke	-	2 (DEC)

1000. Didapatkan konstanta pembagi energi aktif sebesar 2. Keempat nilai tersebut akan dimasukkan sebagai konstanta pembagi pada fitur membaca nilai register sesuai dengan parameternya masing-masing.

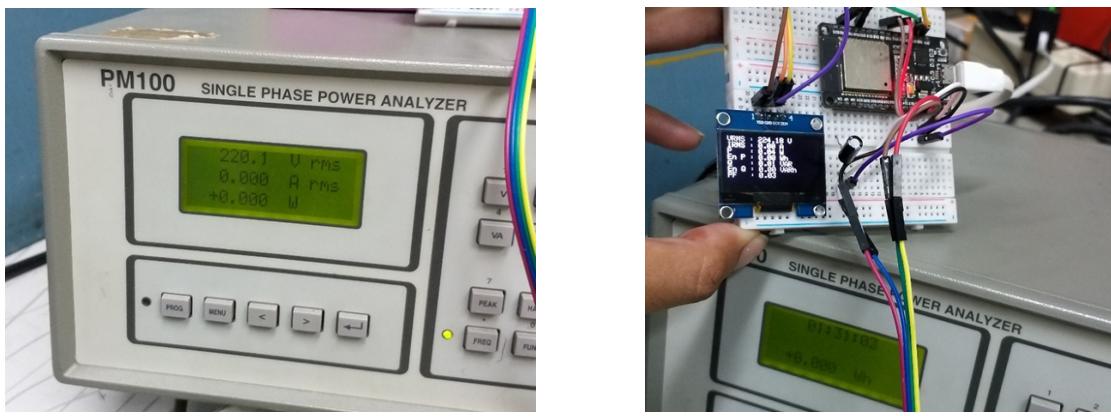
Proses kalibrasi IC Renergy 8209 sudah selesai. Jika dirangkum, hasil yang didapatkan dari proses kalibrasi ini adalah pada tabel 4.2. Tabel tersebut berisi nama register, alamatnya, dan data yang akan dikirim berapa. Data-data tersebut kemudian akan dimasukkan pada fitur inisialisasi kondisi sesuai dengan alamat registernya masing-masing.

Melakukan kalibrasi untuk IC Renergy 8209 membuat IC ini akan berfungsi sebagaimana diinginkan oleh pengembang. Dilakukan konfigurasi untuk menentukan karakteristik dari IC. Karakteristik tersebut yang nantinya akan menjadi perilaku IC dalam melakukan perhitungan energi listrik dan melakukan pembacaan parameter-parameternya. Untuk meminimalisir kesalahan pembacaan dan perhitungan energi, dilakukan proses kalibrasi. Jika tidak dilakukan rangkaian kalibrasi, IC masih tetap berfungsi namun tidak sebagaimana mestinya atau tidak menunjukkan hasil yang sebenarnya. Dengan melakukan kalibrasi, kesalahan pembacaan parameter dan perhitungan energi dapat diminimalisir agar selanjutnya dapat lolos ke tahap pengujian.

4.3 Pembahasan Hasil Pengujian Prototipe Sistem

Bagian ini akan membahas tentang hasil dari pengujian yang telah dilakukan berdasarkan standar yang berlaku secara nasional. Standar tersebut diatur dalam Surat Keputusan Direktur Jenderal Perlindungan Konsumen dan Tertib Niaga Nomor 161 Tahun 2019. Terdapat empat pengujian yang harus dilakukan dan agar dinyatakan terbukti secara fungsionaitas bekerja, keempat pengujian tersebut harus dinyatakan lolos.

4.3.1 Hasil Pengujian Tanpa Beban



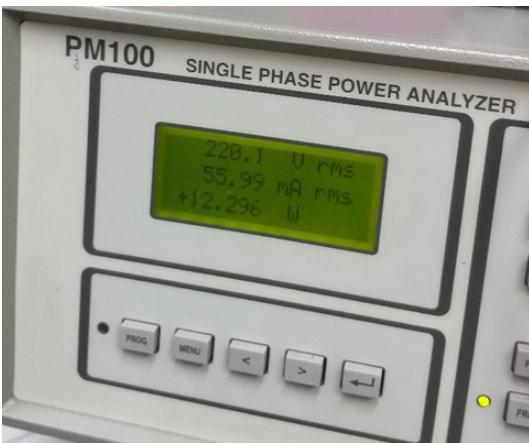
Gambar 4.9. Pengujian tanpa beban

Pengujian tanpa beban dilakukan dengan memberikan tegangan dasar dan tanpa ada arus yang mengalir. Kondisi ini diberikan selama 0.91 jam atau 54.6 menit sesuai dengan perhitungan yang telah dilakukan pada bab sebelumnya. Selama kondisi tersebut, agar dinyatakan lolos, energi meter yang diuji tidak boleh mengeluarkan satu pun pulsa.

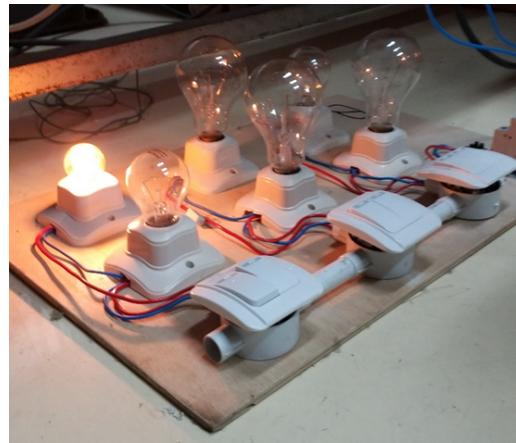
Pada pengujian yang dilakukan pada skripsi ini, didapatkan tegangan sebesar 220.1 V dan arus 0 A atau tidak ada arus yang mengalir sama sekali seperti pada Gambar 4.9 (a). Pengujian dilakukan selama 1 jam 31 menit dan referensi tidak menghitung satu pun Wh. Dari sisi energi meter yang diuji juga tidak mengeluarkan satu pun pulsa atau menghitung energi seperti pada Gambar 4.9 (b). Dengan begitu energi meter ini lolos pengujian tanpa beban.

4.3.2 Hasil Pengujian Arus Mula

Seperti pada pengertiannya yang telah dijelaskan sebelumnya, arus mula merupakan arus terkecil dari energi meter yang dirancang untuk mulai menghitung energi. Berdasarkan perhitungan, besaran arus mula pada energi meter yang diuji adalah 0.004 A. Akan tetapi karena keterbatasan peralatan, arus mula yang didapatkan adalah 56 mA atau 12.296 W dengan menyalakan bohlam kecil dengan rating sekitar 12 W



(a) Referensi



(b) Pengaturan beban

Gambar 4.10. Pengujian arus mula

seperti pada Gambar 4.10. Untuk pengujian ini batas kesalahan yang diizinkan adalah sebesar 2.5%.

Dengan daya aktif sebesar 12.296 W maka waktu antara dua pulsa yang seharusnya dikeluarkan adalah 146.389 sekon. Sementara waktu antara dua pulsa yang dikeluarkan oleh IC Renergy 8209 dan terukur di osiloskop adalah sebesar 149.6 sekon. *Error pulse* yang terjadi pada pengujian ini adalah 2.19%. Nilai tersebut masih dibawah dari batas kesalahan yang diizinkan yaitu 2.5%. Sehingga pengujian arus mula dinyatakan lolos.

4.3.3 Hasil Pengujian Variasi Arus



(a) Kondisi faktor daya 1



(b) Kondisi faktor daya 0.5

Gambar 4.11. Pengaturan Variac 2

Pengujian variasi arus dilakukan pada tegangan dasar yaitu 220 V dengan beberapa besaran arus dengan faktor daya tertentu yang titik uji wajibnya telah

ditentukan oleh standar yang digunakan. Titik uji wajib tersebut telah disebutkan pada bab sebelumnya. Setiap titik uji wajib harus dibaca setidaknya tiga kali untuk mendapatkan rata-rata kesalahan pembacaan. Batas kesalahan yang diizinkan dari setiap titik uji wajib tersebut berbeda-beda, sesuai dengan besaran arus dan faktor dayanya.

Untuk mendapatkan arus dan faktor daya yang diinginkan, penulis mengkombinasikan beberapa lampu pijar dengan *rating* tertentu dan induktor. Terdapat perbedaan koneksi pada Variac 2 apabila diinginkan beban resistif atau beban induktif. Perbedaan tersebut seperti Gambar 4.11. Ketika diinginkan beban yang faktor dayanya 1 atau resistif, jalur beban dihubungkan pada bagian *input* Variac 2. Pada kondisi tersebut, tidak ada arus yang mengalir melalui Variac 2. Hal tersebut dilakukan karena pada internal Variac 2 terdapat induktor yang dapat menghasilkan daya reaktif dan kondisi tersebut tidak diinginkan ketika hanya daya aktif saja yang ingin dihasilkan.

Jika diinginkan faktor daya bernilai 0.5, maka jalur beban ditempatkan pada pin *output* pada variac 2 atau arus dialirkan terlebih dahulu melalui variac. Dengan begitu Variac 2 dapat berfungsi sebagai pengatur arus yang akan masuk ke beban. Ketika ingin mencari arus dan faktor daya tertentu, kombinasi lampu pijar dan induktor diatur. Variac 2 harus ditempatkan pada kondisi nol saat catu daya dihubungkan. Hal tersebut untuk menghindari *inrush current* atau arus instan yang nilainya besar dan dapat menyebabkan *trip* pada MCB.

Bagian-bagian berikutnya adalah penjelasan lebih detil terkait pengujian variasi arus yang telah dilakukan.

1. Arus minimal

Arus minimal yang perlu dicapai pada pengujian kali ini adalah sebesar 0.1 A. Agar tercapai besaran arus tersebut digunakan dua buah lampu pijar kecil yang dapat menghasilkan arus sebesar 107.81 mA, seperti pada Gambar 4.12. Dengan arus tersebut didapatkan daya aktif sebesar 23.7 W, artinya diperkirakan waktu antara dua pulsa yang dibutuhkan adalah sebesar 75.9494 sekon. Pada kondisi arus dan faktor daya tersebut, batas kesalahan yang diizinkan adalah 2.5%. Nilai daya tersebut tidak konstan, melainkan dinamis dan setiap cuplikan yang akan diukur nanti akan bervariasi waktu yang diharapkannya. Akan tetapi dinamika yang terjadi relatif kecil.

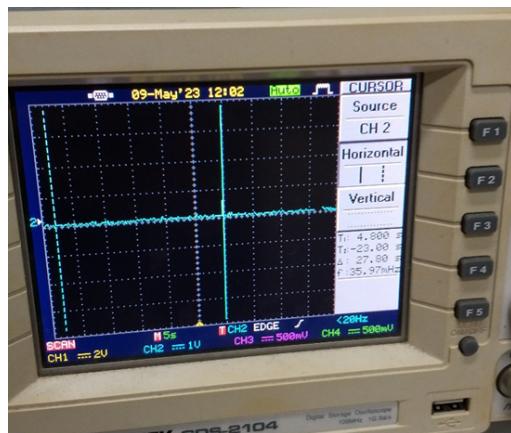
Pada pembacaan pertama dan kedua didapatkan waktu terukur pada osiloskop sebesar 77.8 sekon dan waktu diharapkan sebesar 75.9494 sekon, maka kesalahannya adalah 2.44%. Pada pembacaan ketiga, didapatkan waktu terukur pada osiloskop sebesar 77.4 sekon dan waktu yang diharapkannya sebesar 75.6302 sekon, maka kesalahannya adalah 2.34%. Rata-rata dari ketiga pembacaan tersebut adalah 2.4% dan nilai tersebut masih dibawah batas.



(a) Referensi



(b) Pengaturan beban



(c) Waktu terukur pada osiloskop

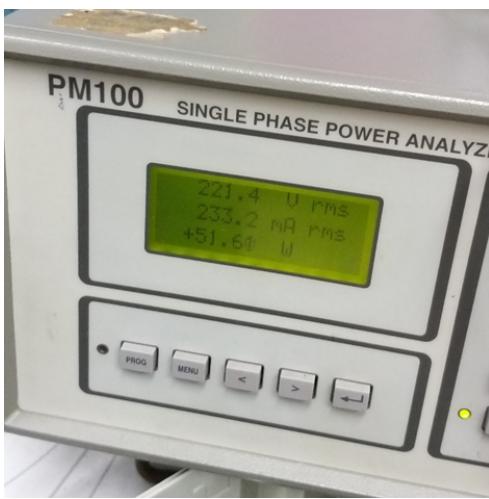
Gambar 4.12. Pengujian arus minimal

2. Arus transisi

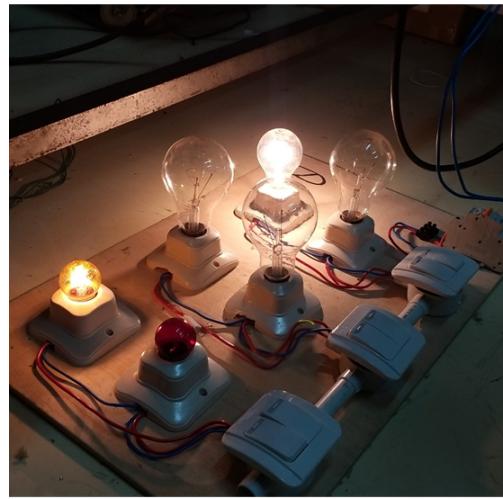
(a) Faktor daya 1

Arus transisi yang perlu dicapai adalah 0.2 A. Untuk mendapatkan nilai arus tersebut digunakan satu lampu pijar kecil dan satu lampu pijar dengan *rating* 40 W. Pengaturan tersebut mampu menghasilkan arus sekitar 233 mA atau daya aktif sekitar 51.61 W seperti pada Gambar 4.13 (a). Besar arus tersebut memiliki batas kesalahan yang diizinkan yaitu sebesar 2.5%. Kemudian dengan daya aktif tersebut maka diperkirakan waktu antara dua pulsa adalah sebesar 34.8769 sekon.

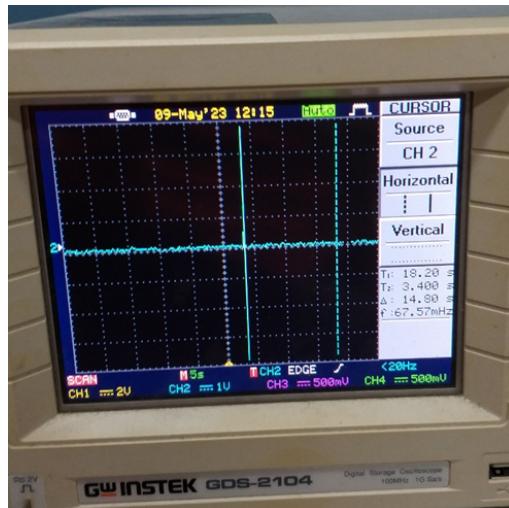
Pada pembacaan pertama dan kedua didapatkan waktu terukur di osiloskop sebesar 35.2 sekon yang ditunjukkan pada Gambar 4.13 (c). Terlihat waktu yang terukur pada osiloskop tersebut adalah 14.8 sekon. Hal tersebut karena keterbatasan dari osiloskop yang hanya bisa mengukur pada jangkauan 50 sekon dengan *time/div* sebesar 5 sekon. Untuk mendapatkan waktu antara dua pulsa perlu mengurangi 50 sekon dengan 14.8 sekon,



(a) Referensi



(b) Pengaturan beban



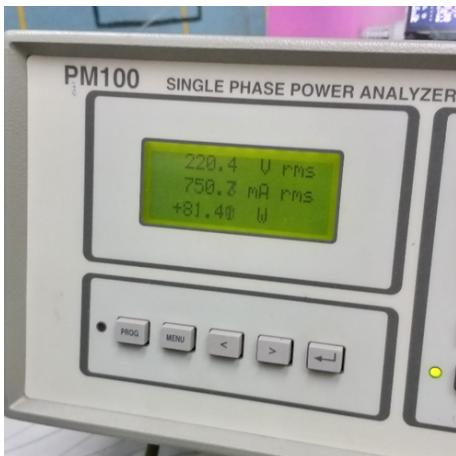
(c) Waktu terukur pada osiloskop

Gambar 4.13. Pengujian arus transisi faktor daya 1

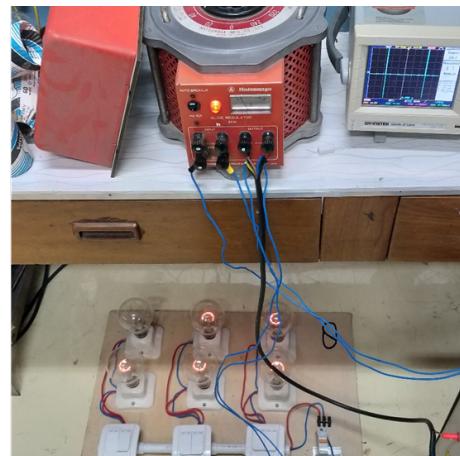
didapatkan waktu sebesar 35.2 sekon. Untuk waktu yang diharapkan adalah sebesar 34.8769 sekon, maka kesalahannya adalah sebesar 0.93%. Pada pembacaan ketiga, terdapat kenaikan daya aktif menjadi 51.69 W, maka waktu yang diharapkan adalah 34.8229 sekon. Sementara waktu yang terukur pada osiloskop adalah sebesar 35.2 sekon. Maka kesalahan pembacaan ketiga adalah 1.08%. Rata-rata kesalahan dari ketiga pembacaan tersebut adalah 0.98%, yang masih dibawah batas kesalahan yang diizinkan yaitu 2.5%.

(b) Faktor daya 0.5

Untuk arus transisi dengan faktor daya 0.5, penulis menyalakan lima buah lampu yang memiliki *rating* 150 W dan Variac 2 yang terhubung. Dengan kombinasi perangkat tersebut didapatkan arus sebesar 750.7 mA



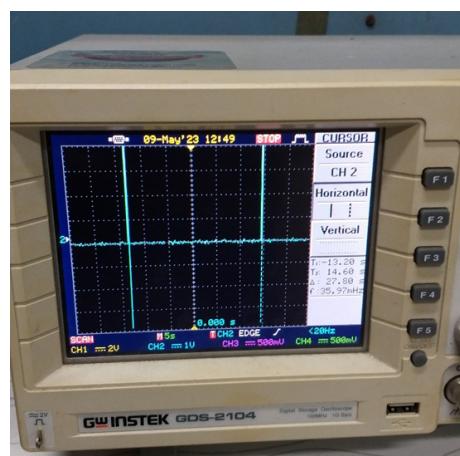
(a) Referensi



(b) Pengaturan beban



(c) Faktor daya



(d) Waktu terukur pada osiloskop

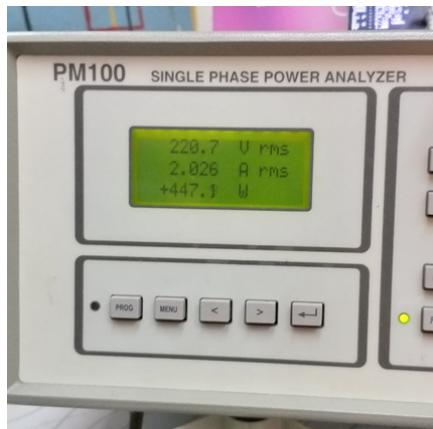
Gambar 4.14. Pengujian arus transisi faktor daya 0.5

dan daya aktif sebesar 81.4 W seperti pada Gambar 4.14. Pada nilai arus tersebut, batas kesalahan yang diizinkan adalah sebesar 2.5%. Kemudian waktu antara dua pulsa yang diharapkan dari nilai daya aktif tersebut adalah sekitar 22.113 sekon.

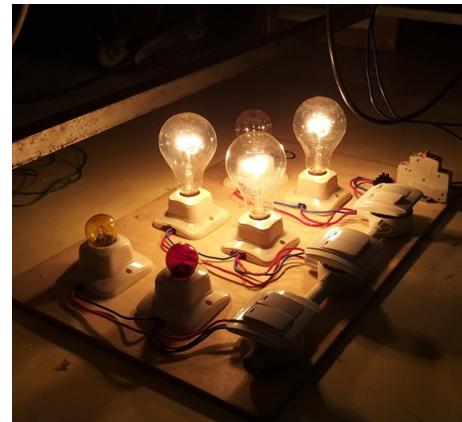
Pada ketiga pembacaan, waktu antara dua pulsa yang dihasilkan oleh IC Renergy 8209 dan terukur oleh osiloskop adalah 22.2 sekon. Namun daya aktif yang terukur oleh referensi mengalami deviasi dari nilai 81.4 W sehingga waktu yang diharapkan berubah-ubah menjadi 22.113 sekon, 22.1103 sekon, dan 22.1402 sekon. Maka kesalahan pembacaan yang terjadi adalah 0.39%, 0.41%, dan 0.27%. Rata-rata kesalahan pembacaan yang didapatkan adalah 0.36% dan masih di bawah batas kesalahan yang diizinkan.

3. Arus dasar

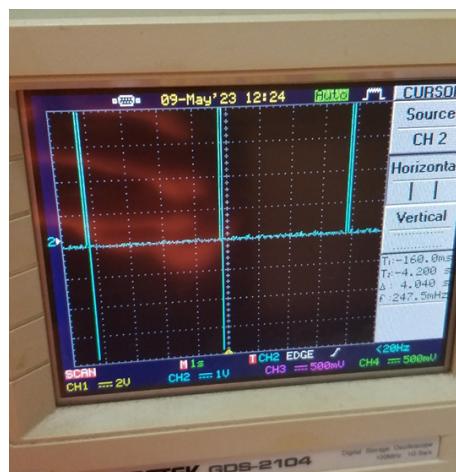
(a) Faktor daya 1



(a) Referensi



(b) Pengaturan beban



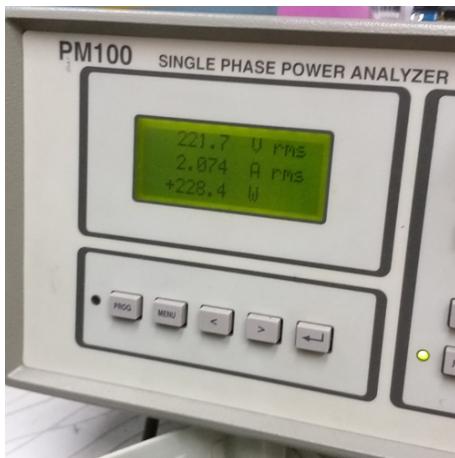
(c) Waktu terukur pada osiloskop

Gambar 4.15. Pengujian arus dasar faktor daya 1

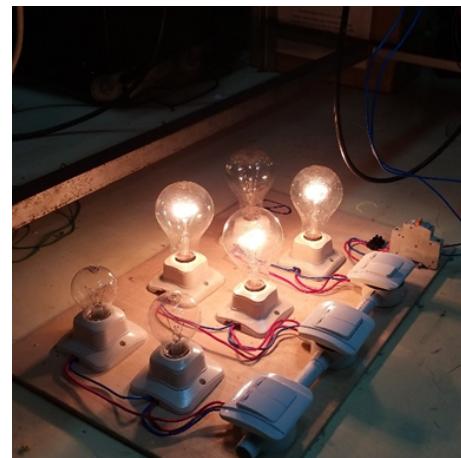
Kondisi arus dasar atau 2 A dengan faktor daya bernilai 1, didapatkan dengan menyalakan tiga buah lampu pijar yang memiliki *rating* 150 W. Pada pengaturan tersebut didapatkan arus sebesar sekitar 2.026 A dan daya aktif sekitar 447 W seperti pada Gambar 4.15. Untuk kondisi arus ini, batas kesalahan yang diizinkan adalah sebesar 2%. Sementara untuk daya aktif tersebut, maka diperkirakan waktu antara dua pulsa adalah 4.0268 sekon.

Pada ketiga pembacaan yang dicuplik, pulsa yang keluar dari IC Renergy 8209 konsisten diwaktu 2.02 sekon walau pada sisi daya aktif terjadi deviasi yang relatif kecil dari 447 W. Dengan mempertimbangkan deviasi tersebut, kesalahan pembacaan yang terjadi berturut-turut adalah 0.33%, 0.26%, dan 0.35%. Sehingga jika ketiga nilai tersebut dirata-rata akan menghasilkan rata-rata kesalahan pembacaan sebesar 0.31% dan nilai tersebut masih di bawah dari batas kesalahan yang diizinkan.

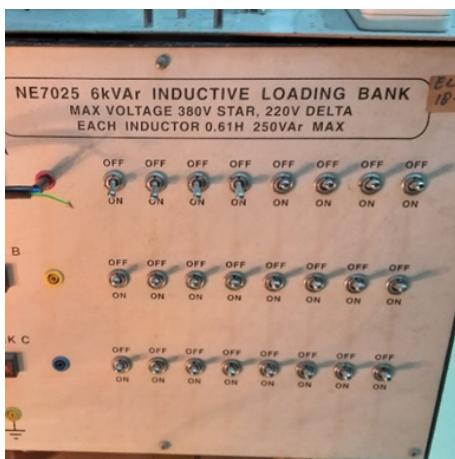
(b) Faktor daya 0.5



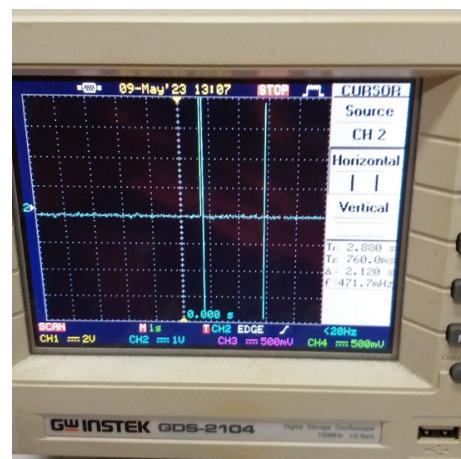
(a) Referensi



(b) Pengaturan beban resistif



(c) Pengaturan beban induktif



(d) Waktu terukur pada osiloskop

Gambar 4.16. Pengujian arus dasar faktor daya 0.5

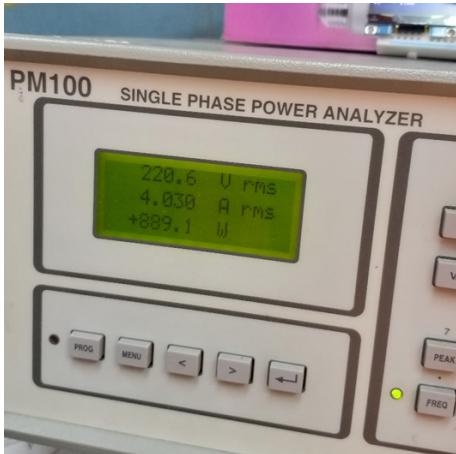
Arus dasar dengan faktor daya sebesar 0.5 didapatkan dengan mengkombinasikan 4 buah induktor bernilai 0.61 H dan tiga buah lampu pijar yang memiliki *rating* sebesar 150 W. Dengan pengaturan tersebut didapatkan faktor daya sebesar 0.508, arus 2.023 A, dan daya aktif 228.4 W seperti pada Gambar 4.16. Besaran arus dasar dan faktor daya tersebut memiliki batas kesalahan yang diizinkan sebesar 2.5%. Sementara besaran daya aktif tersebut diperkirakan akan membutuhkan waktu 7.8809 sekon untuk jarak waktu antara dua pulsa.

Sama seperti ketika kondisi faktor daya bernilai 1, pada titik uji wajib ini waktu antara dua pulsa yang dihasilkan oleh IC Renergy 8209 dan terukur oleh osiloskop konsisten diangka 7.88 sekon. Mempertimbangkan deviasi yang terjadi, kesalahan pembacaan yang dihasilkan dari ketiga pembacaan berturut-turut adalah sebesar 0.01%, 0.03%, dan 0.12%. Maka rata-rata kesalahan untuk titik uji ini adalah sebesar 0.05% dan angka

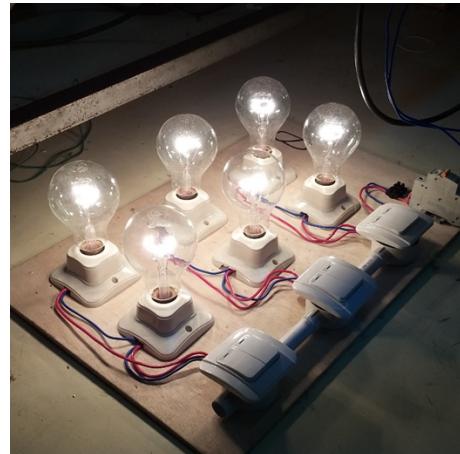
tersebut masih dibawah dari batas kesalahan yang diizinkan yaitu 2.5%.

4. Arus maksimal

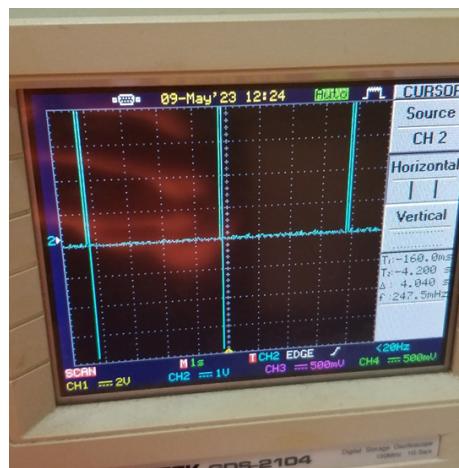
(a) Faktor daya 1



(a) Referensi



(b) Pengaturan beban



(c) Waktu terukur pada osiloskop

Gambar 4.17. Pengujian arus maksimal faktor daya 1

Titik uji arus maksimal adalah sebesar 4 A. Untuk mendapatkan arus tersebut, pengaturan yang dilakukan adalah dengan menyalakan enam buah lampu pijar dengan *rating* 150 W. Didapatkan arus sekitar 4.03 A dan daya aktif sekitar 889.1 W seperti pada Gambar 4.17. Pada nilai arus tersebut batas kesalahan yang diizinkan adalah sebesar 2%. Kemudian diperkirakan dengan nilai daya aktif tersebut, waktu yang dibutuhkan untuk pulsa muncul adalah sekitar 2.0245 sekon.

Ketiga pembacaan dilakukan untuk titik uji ini dan IC Renergy 8209 mengeluarkan waktu pulsa yang konsisten untuk ketiga pembacaan tersebut yaitu di angka 2.02 sekon. Kesalahan pembacaan yang terjadi dengan

mempertimbangkan deviasi daya aktif berturut-turut adalah 0.22%, 0.2%, dan 0.23%. Maka rata-rata kesalahan dari ketiga pembacaan di titik uji ini adalah sebesar 0.22% dan angka tersebut masih di bawah batas kesalahan yang diizinkan yaitu 2%.

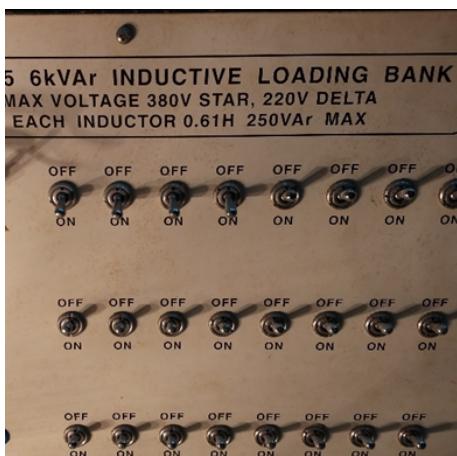
(b) Faktor daya 0.5



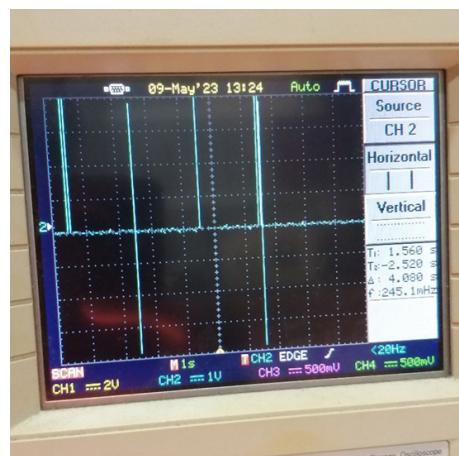
(a) Referensi



(b) Pengaturan beban resistif



(c) Pengaturan beban induktif



(d) Waktu terukur pada osiloskop

Gambar 4.18. Pengujian arus maksimal faktor daya 0.5

Titik uji arus maksimal dengan faktor daya 0.5 menjadi titik uji terakhir dalam pengujian variasi arus. Untuk mencapai besaran arus dan faktor daya tersebut, pengaturan yang dilakukan adalah dengan menyalakan tiga buah lampu yang memiliki *rating* 150 W, dua buah lampu kecil, dan empat buah induktor 0.61 H. Pada pengaturan tersebut didapatkan faktor daya sebesar 0.48, arus sebesar 4.031 A dan daya aktif 430 W seperti pada Gambar 4.18. Besaran arus dan faktor daya tersebut memiliki batas kesalahan yang diizinkan yaitu 2.5%.

Sama seperti dititik uji wajib sebelumnya, IC Renergy 8209

Tabel 4.3. Rangkuman pengujian variasi arus

Arus Uji	Faktor Daya	Waktu Diharapkan	Waktu Terukur	Kesalahan	Rata-rata kesalahan	BKD
I_{min}	1	75.94936709	77.8	2.44%	2.40%	2.50%
		75.94936709	77.8	2.44%		
		75.6302521	77.4	2.34%		
I_{tr}	1	34.87696183	35.2	0.93%	0.98%	2.50%
		34.87696183	35.2	0.93%		
		34.82298317	35.2	1.08%		
	0.5 induktif	22.11302211	22.2	0.39%	0.36%	2.50%
		22.11030586	22.2	0.41%		
		22.1402214	22.2	0.27%		
I_d	1	4.026845638	4.04	0.33%	0.31%	2.00%
		4.029550034	4.04	0.26%		
		4.025944979	4.04	0.35%		
	0.5 induktif	7.880910683	7.88	0.01%	0.05%	2.50%
		7.877461707	7.88	0.03%		
		7.870572803	7.88	0.12%		
I_{max}	1	2.024519177	2.02	0.22%	0.22%	2.00%
		2.02406387	2.02	0.20%		
		2.024746907	2.02	0.23%		
	0.5 induktif	4.109589041	4.08	0.72%	0.76%	2.50%
		4.109589041	4.08	0.72%		
		4.114285714	4.08	0.83%		

menghasilkan pulsa dengan jarak waktu konsisten diangka 4.08 sekon. Kesalahan pembacaan yang terjadi dengan mempertimbangkan deviasi faktor daya dan daya aktif berturut-turut adalah 0.72%, 0.72%, dan 0.83%. Didapatkan rata-rata kesalahan pembacaan yaitu sebesar 0.76% dan nilai tersebut masih di bawah batas kesalahan yang diizinkan yaitu 2.5 %.

Rangkuman dari pengujian variasi arus dapat dilihat pada tabel 4.3. Dari rata-rata kesalahan yang didapatkan, semua variasi arus masih di bawah batas kesalahan yang diizinkan (BKD). Artinya dapat dikatakan sistem *monitoring* konsumsi listrik menggunakan IC Renergy 8209 pada platform ESP32 lolos pengujian variasi arus.

4.3.4 Hasil Pengujian Register

Energi meter yang dikembangkan memiliki EC sebesar 2000 imp/kWh, artinya resolusi energi dari perangkat ini adalah 0.5. Kemudian pengujian register dilakukan pada arus uji sekitar 2 A. Batas kesalahan yang diizinkan dari pengujian ini adalah sepersepuluh dari batas kesalahan arus yang digunakan, artinya batas kesalahan yang diizinkan untuk pengujian ini adalah 0.2 %. Pada pengujian ini, terdapat energi minimum yang harus diuji dengan mengikuti formula berikut:

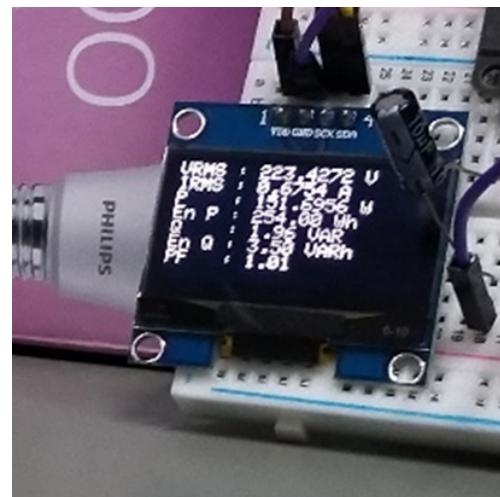
$$E_{min} = \frac{1000 \cdot resolusi}{BKD(\%)} \quad (4-1)$$

Dengan menggunakan formula tersebut, perhitungan energi minimum yang harus diuji adalah sebagai berikut:

$$E_{min} = \frac{1000 \cdot 0.5}{2} = 259Wh$$



(a) Tampilan referensi



(b) Tampilan OLED

Gambar 4.19. Pengujian register

Maka energi minimum yang harus diujikan adalah sebesar 250 Wh. Dari referensi terbaca energi aktif yang terhitung adalah sebesar 253.6 Wh seperti pada Gambar 4.19. Sementara dari sisi IC Renergy 8209 menunjukkan pembacaan sebesar 254 Wh. Perbedaan relatif yang terjadi adalah sebesar 0.16 % dan nilai tersebut masih dibawah batas kesalahan yaitu 0.2 %. Maka pengujian register dinyatakan lolos pengujian.

Semua pengujian telah dilakukan dan semuanya dinyatakan lolos. Sehingga bisa dipastikan bahwa prototipe sistem *monitoring* konsumsi listrik menggunakan IC Renergy 8209 pada platform ESP32, secara fungsionalitas, dapat memenuhi standar yang berlaku secara nasional. Walaupun masih terdapat kesalahan pembacaan dan perhitungan, namun hal tersebut masih dibawah standar yang menjadi acuan. Karena sejatinya kesalahan pembacaan merupakan hal yang tidak bisa dihilangkan dari suatu produk yang tujuannya untuk mengukur sesuatu. Lolosnya pengujian yang dilakukan dapat dijadikan acuan untuk pengembangan perangkat lebih lanjut hingga fabrikasi untuk keperluan komersial.

4.4 Pembahasan Hasil Pengembangan *Firmware* final

Setelah fungsionalitas dari sistem *monitoring* konsumsi listrik menggunakan IC Renergy 8209 terbukti bekerja berdasarkan standar yang berlaku secara nasional, langkah selanjutnya adalah melakukan pengembangan *firmware* final. Seperti yang telah disebutkan pada bab sebelumnya, sasaran pengguna dari *firmware* ini adalah pengguna sistem *monitoring* konsumsi listrik. Sehingga fitur-fitur yang dirancang disesuaikan dengan kebutuhan. Akan tetapi hal tersebut tidak membuat semua fitur yang telah dikembangkan sebelumnya pada *firmware* kalibrasi dan pengujian menjadi tak berguna. Beberapa fitur digunakan kembali sama seperti sebelumnya, fitur lainnya diperbarui untuk disesuaikan dengan kebutuhan, dan terdapat beberapa fitur baru yang dikembangkan. Lebih detilnya akan dijelaskan pada bagian-bagian selanjutnya.

4.4.1 Fitur yang Diterapkan Kembali

Seperti yang telah disebutkan sebelumnya, terdapat fitur-fitur yang tetap dipertahankan untuk *firmware* final. Fitur-fitur ini dipertahankan karena fungsi dari fitur-fitur tersebut masih relevan dengan tujuan dikembangkannya *firmware* final. Fitur-fitur tersebut antara lain adalah:

1. Fitur komunikasi UART dengan IC Renergy 8209
2. Fitur membaca nilai register
3. Fitur memberi nilai register
4. Fitur inisialisasi konfigurasi

Fitur komunikasi UART dengan IC Renergy 8209 dan fitur memberi nilai register tidak mengalami perubahan kode program. Untuk fitur membaca nilai register pada *firmware* final telah ditambahkan konstanta pembagi. Sementara fitur inisialisasi konfigurasi yang terdapat pada *firmware* final sudah terdapat nilai-nilai konstanta kalibrasi yang telah dicari sebelumnya. Alamat register dan data yang dimasukkan pada fitur inisialisasi konfigurasi tersedia pada tabel 4.4.

4.4.2 Fitur Tombol Interupsi

Fitur tombol interupsi merupakan fitur baru yang dikembangkan untuk *firmware* final. Terdapat dua tombol yang disediakan dengan fungsinya masing-masing. Fungsi tombol pertama adalah untuk mengganti tampilan pada OLED dengan metode *counter*. Sementara fungsi tombol kedua adalah untuk membuat perhitungan energi menjadi nol kembali dan melakukan *soft* reset untuk IC Renergy 8209. Fitur tombol interupsi memanfaatkan fungsi interupsi yang telah disediakan oleh ESP32. Kode program dari fitur ini dapat dijelaskan secara garis besar dengan *pseudocode* pada Algorithm 6.

Algorithm 6 Interups

```
1: function BUTTON1ISR
2:   if millis() - button1PressTime > debounceDelay then
3:     button1Pressed = true
4:     button1PressTime = millis()
5:   end if
6: end function
7:
8: function BUTTON2ISR
9:   if millis() - button2PressTime > debounceDelay then
10:    button2Pressed = true
11:    button2PressTime = millis()
12:  end if
13: end function
14:
15: At initialization
16: pinMode(button1, inputPullup)
17: attachInterrupt(button1, button1ISR, Falling)
18: pinMode(button2, inputPullup)
19: attachInterrupt(button2, button2ISR, Falling)
20:
21: At loop
22: if button1Pressed then
23:   button1Pressed = false
24:   counter++
25:   if counter > 5 then
26:     counter = 0
27:   end if
28: end if
29: if button2Pressed then
30:   button2Pressed = false
31:   reset rn8209
32:   lastValue = 0
33:   initialValue = 0
34:   writeDataToEEPROM(lastValue, ADDR1)
35:   writeDataToEEPROM(initialValue, ADDR2)
36: end if
```

Tabel 4.4. Inisialisasi konfigurasi final

No	Nama Register	Alamat Register	Data yang dikirim	Keterangan
1	Perintah spesial	0xEA	0xE5	Membuka kunci register
2	Perintah spesial	0xEA	0xFA	<i>Soft</i> reset
3	Perintah spesial	0xEA	0x5A	<i>Set channel</i> arus A
4	SysCon	0x00	0x1602	Konfigurasi sistem
5	EMUCon	0x01	0x2803	Konfigurasi perhitungan 1
6	EMUCon2	0x17	0x0000	Konfigurasi perhitungan 2
7	HFConst	0x02	0x192B	Pengatur EC
8	Pstart	0x03	0x0000	<i>No-load threshold</i> daya aktif
9	GPQA	0x05	0xFBD4	Konstanta kalibrasi gain
10	PhsA	0x07	0xF4	Konstanta kalibrasi fase
11	IARMSOS	0x0E	0x0000	Konstanta kalibrasi <i>offset</i> arus channel A
12	APOSA	0x0A	0x0000	Konstanta kalibrasi <i>offset</i> daya aktif
13	Qstart	0x04	0x0000	<i>No-load threshold</i> daya reaktif
14	QPhsCal	0x09	0xFBD2	Konstanta kalibrasi daya reaktif
15	RPOSA	0x0C	0x0000	Konstanta kalibrasi <i>offset</i> daya reaktif
16	Perintah spesial	0xEA	0xDC	Mengunci register konfigurasi dan kalibrasi

Terdapat dua fungsi yang dibuat pada fitur ini yaitu Button1ISR dan Button2ISR. ISR sendiri memiliki kepanjangan yaitu *Interrupt Service Routine* atau rutin untuk layanan interupsi. Fungsi ini dibuat untuk mendeteksi adanya interupsi dari tombol-tombol terkait. Selain itu, fungsi ini juga untuk meminimalisir efek *physical* dari *push button* yaitu *bouncing* dengan cara memberikan waktu tunda sebesar 200 milisekon.

Pada bagian *setup* dari struktur kode program *firmware* final, dilakukan inisialisasi dari pin digital yang terhubung dengan *push button* dan jenis masukannya. Digunakan jenis masukan *inputPullup* yang artinya akan memberikan nilai satu atau *HIGH* apabila terdapat tegangan yang masuk. Kemudian dilakukan juga inisialisasi interupsi dengan fungsi *attachInterrupt*.

Pada bagian *loop* dari struktur kode program *firmware* final, terdapat dua kondisi *if* yang berguna untuk mengatur perintah apabila salah satu tombol ditekan oleh pengguna. Kondisi *if* pertama adalah untuk mengatur apabila tombol satu ditekan. Kondisi dan tombol ini berfungsi untuk melakukan *counter* sampai nilainya lima kemudian kembali lagi ke nol. Nilai *counter* ini kemudian akan dijadikan argumen pada tampilan OLED yang akan dijelaskan pada bagian lainnya. Untuk kondisi *if* kedua adalah untuk mengatur apabila tombol dua ditekan. Kondisi dan tombol ini berfungsi untuk membuat perhitungan pemakaian energi menjadi nol kembali sekaligus melakukan reset untuk IC Renergy 8209. Dilakukan reset terlebih dahulu untuk IC. Hal ini dilakukan agar alamat register perhitungan energi yaitu EnergyP nilainya kembali menjadi nol. Kemudian variabel *lastValue* dan *initialValue* diatur nilainya agar nol dan

kedua variabel tersebut dikirim ke alamat EEPROMnya masing-masing. Dengan begitu nilai perhitungan konsumsi energi akan kembali menjadi nol sesuai keinginan dari pengguna.

4.4.3 Fitur Penyimpanan Data Pada EEPROM

Sama seperti fitur tombol interupsi, fitur penyimpanan data pada EEPROM merupakan fitur yang baru dikembangkan pada *firmware* final. Pengembangan fitur ini didasari dari kondisi yaitu ketika catu daya terputus dari ESP32, maka IC Renergy 8209 akan kehilangan daya dan mengakibatkan alamat register EnergyP akan mengulang kembali perhitungan konsumsi energi listrik dari nol. Kondisi tersebut tidak diinginkan karena pada penerapannya sistem *monitoring* konsumsi listrik harus tetap bisa melanjutkan perhitungan konsumsi listrik walau catu daya terputus kecuali pengguna menginginkan perhitungan diulang kembali dari nol. Dari kondisi tersebut, fitur penyimpanan data dikembangkan.

ESP32 memiliki memori internal yang bersifat *non volatile* atau memori yang dapat diisi dan dihapus data didalamnya namun akan tetap tersimpan walau catu daya terputus dari ESP32. Memori tersebut bernama EEPROM atau *Electrically Erasable Programmable Read-Only Memory*. EEPROM akan tetap menyimpan data yang telah diberikan walau catu daya terputus. Terdapat 512 *byte* memori yang telah disediakan oleh ESP32 dan dapat digunakan oleh pengembang untuk menyimpan data [26]. 512 *byte* tersebut tersedia pada 512 alamat EEPROM, artinya setiap alamat EEPROM memiliki ukuran 1 *byte*. Nantinya alamat-alamat tersebut akan digunakan untuk menyimpan perhitungan energi listrik yang telah dilakukan.

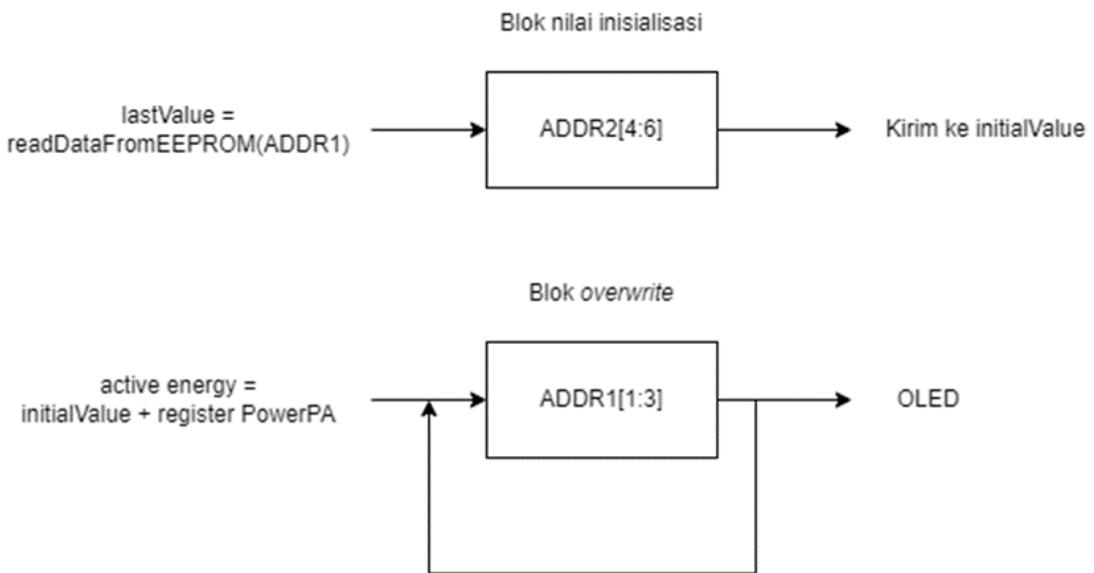
Kode program dari fitur penyimpanan data pada EEPROM dapat direpresentasikan pada *pseudocode* yang tertera pada Algorithm 7. Terdapat dua fungsi yang dibuat pada fitur ini yaitu *writeDataToEEPROM* dan *readDataFromEEPROM*. Seperti namanya, fungsi pertama bertugas untuk memberi nilai pada EEPROM dan fungsi kedua bertugas untuk membaca nilai yang terdapat pada alamat EEPROM. Fungsi ini perlu dibuat karena setiap alamat EEPROM hanya mampu menyimpan data sebesar satu *byte* sementara alamat register EnergyP memiliki ukuran empat *byte*. Fungsi tersebut masing-masing akan membaca dan memberi nilai pada EEPROM dengan melakukan *shifting* pada data dari alamat register EnergyP. Argumen yang dibutuhkan untuk fungsi *writeDataToEEPROM* ada dua yaitu data yang akan disimpan pada alamat EEPROM dengan tipe data *unsigned long* berukuran 32 bit dan alamat EEPROM yang dituju. Sementara argumen yang dibutuhkan untuk fungsi *readDataFromEEPROM* hanya satu yaitu alamat EEPROM yang akan dibaca nilai di dalamnya.

Pada bagian *setup* dalam struktur *firmware* final, terdapat variabel *lastValue*

Algorithm 7 Penyimpanan Data Pada EEPROM

```
1: function WRITEDATATOEEPROM(data, address)
2:     EEPROM.write(address, data » 16)
3:     EEPROM.write(address+1, data » 8)
4:     EEPROM.write(address+2, data)
5:     EEPROM.commit()
6: end function
7:
8: function READDATAFROMEEPROM(address)
9:     data = 0
10:    data |= EEPROM.read(address) « 16
11:    data |= EEPROM.read(address+1) « 8
12:    data |= EEPROM.read(address+2)
13:    return data
14: end function
15:
16: At initialization
17: lastValue = readDataFromEEPROM(ADDR1)
18: writeDataToEEPROM(lastValue, ADDR2)
19: initialValue = readDataFromEEPROM(ADDR2)
20:
21: At loop
22: activeEnergy = initialValue + registerPowerPA
23: writeDataToEEPROM(activeEnergy, ADDR1)
24: displayEnergy = readDataFromEEPROM(ADDR1)
25:
26: displayEnergy to OLED
```

untuk membaca nilai dari alamat EEPROM ADDR1. Kemudian nilai dari variabel tersebut diberi ke alamat EEPROM ADDR2. Terakhir variabel *initialValue* bertugas untuk menyimpan nilai dari alamat EEPROM ADDR2. Sementara pada bagian *loop* terdapat variabel *activeEnergy* yang bertugas untuk mengakumulasikan pembacaan energi listrik dengan melakukan penjumlahan dari variabel *initialValue* dan pembacaan yang dilakukan oleh IC Renergy 8209 saat itu. Kemudian nilai variabel itu disimpan ke dalam alamat EEPROM ADDR1 sebelum ditampilkan pada OLED.

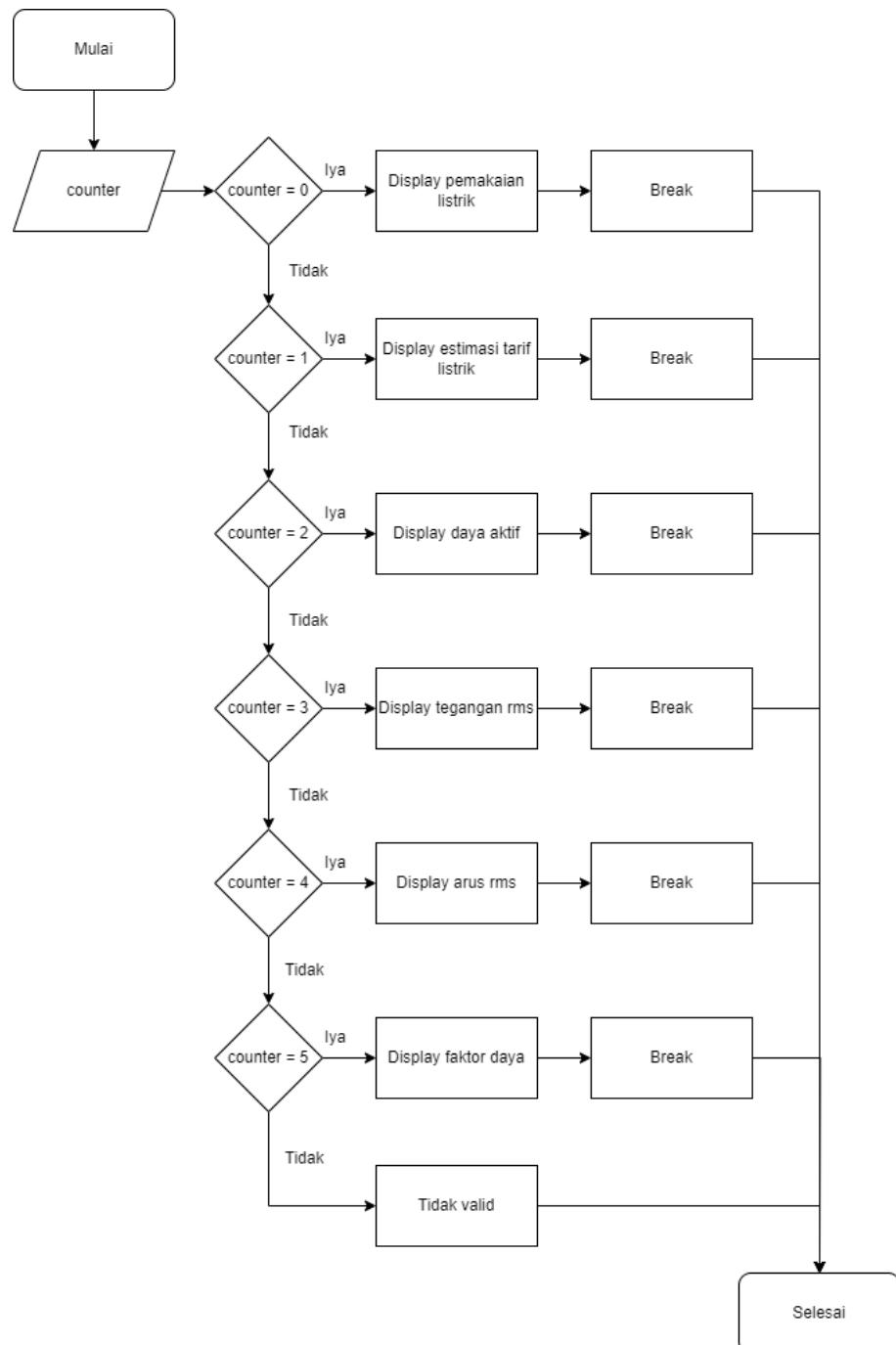


Gambar 4.20. Ilustrasi penyimpanan EEPROM

Untuk menggambarkan secara umum pendekatan yang dilakukan untuk penyimpanan yang dilakukan dapat dilihat pada Gambar 4.20. Terdapat dua blok penyimpanan yang merepresentasikan alamat EEPROM yaitu blok nilai inisialisasi dan blok *overwrite*. Blok nilai inisialisasi bersifat statis atau nilainya tetap dan hanya akan berubah ketika catu daya terputus, ESP32 mengalami reset, atau pengguna memutuskan untuk melakukan reset lewat tombol interupsi kedua. Nilai yang terdapat pada blok ini didapatkan dari pembacaan terakhir dari blok *overwrite* sebelum catu daya terputus. Kemudian nilai yang terdapat blok nilai inisialisasi akan dibaca dan disimpan pada variabel *initialValue*. Blok kedua adalah blok *overwrite* yang bersifat aktif atau nilainya selalu berubah setiap kali *looping* terjadi. Nilai yang disimpan pada blok ini adalah hasil penjumlahan antara variabel *initialValue* dan pembacaan energi oleh IC Renergy 8209 yang tersedia pada alamat register EnergyP. Kemudian nilai tersebut dibaca dan dikirim untuk ditampilkan pada OLED. Dengan pendekatan seperti ini, masalah hilangnya perhitungan energi listrik ketika catu daya terputus atau terjadi reset pada sisi ESP32 dapat teratasi. Sehingga keputusan untuk melakukan perhitungan ulang energi listrik menjadi kendali penuh dari pengguna.

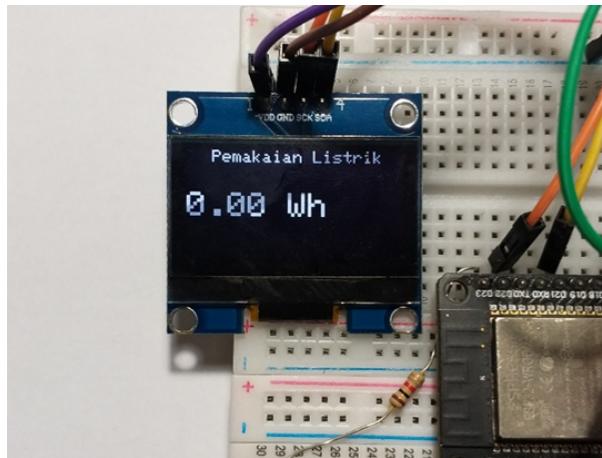
4.4.4 Fitur Tampilan OLED Final

Fitur tampilan OLED merupakan fitur sebelumnya sudah ada pada *firmware* kalibrasi dan pengujian namun terdapat penyesuaian pada tampilannya. Jika pada *firmware* sebelumnya semua parameter energi listrik tertampil pada satu layar, kali ini satu layar hanya terdapat satu tampilan parameter saja. Hal ini bertujuan agar angka dan keterangan parameter energi listrik lebih mudah dibaca. Secara garis besar fitur ini dapat digambarkan dengan diagram alir pada Gambar 4.21.



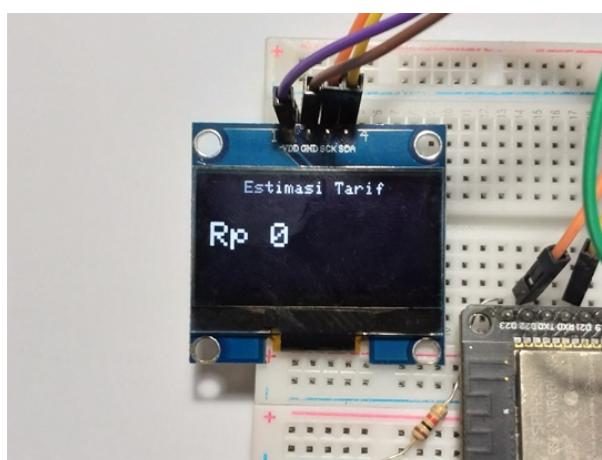
Gambar 4.21. Diagram alir OLED final

Terdapat tombol yang berfungsi untuk mengganti tampilan. Tombol tersebut merupakan sebuah interupsi yang setiap kali dipencet akan menambah nilai variabel *counter*. Apabila nilai *counter* lebih dari 5, maka nilai variabel akan menjadi nol kembali. Nilai variabel *counter* tersebut yang menjadi argumen dari fitur ini. Nilai tersebut akan menjadi parameter dari *switch case* yang dibuat.



Gambar 4.22. Tampilan OLED pemakaian listrik

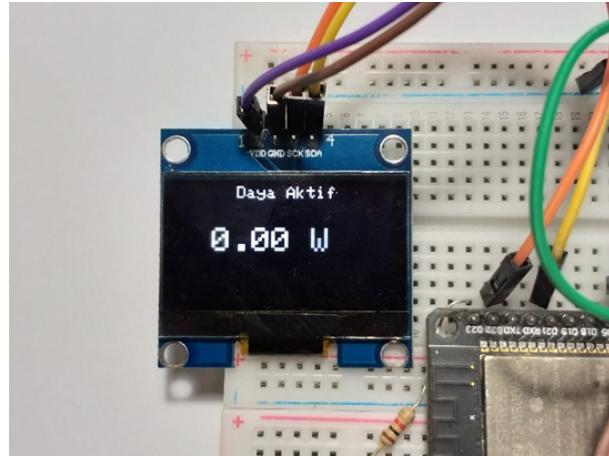
Apabila nilai *counter* adalah nol, parameter energi listrik yang ditampilkan adalah pemakaian energi listrik. Parameter ini ditampilkan pada urutan pertama karena perangkat yang dikembangkan merupakan sistem *monitoring* konsumsi listrik yang memiliki tujuan utama menghitung pemakaian listrik. Data pemakaian energi listrik didapatkan dari alamat ADDR2 EEPROM yang merupakan kombinasi antara variabel *lastValue* dan hasil pembacaan alamat register EnergyP. Tampilan yang dirancang seperti pada Gambar 4.22.



Gambar 4.23. Tampilan OLED estimasi tarif

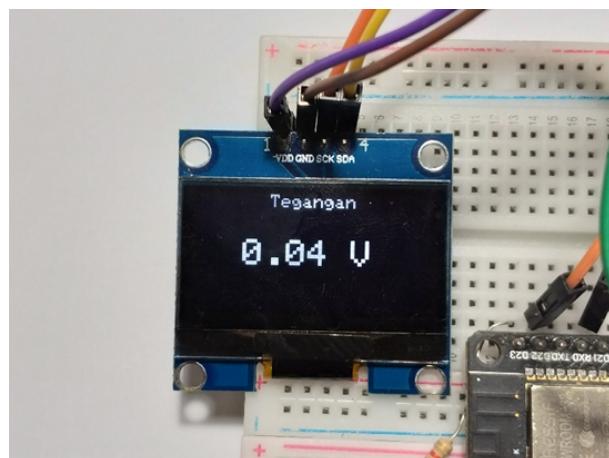
Untuk nilai variabel *counter* adalah satu, data yang ditampilkan adalah estimasi tarif listrik. Perhitungan yang dilakukan adalah dengan mengalikan pemakaian energi listrik dan tarif yang berlaku. Tarif yang digunakan adalah Rp 791/kWh yang termasuk

dalam golongan rumah tangga dengan batas daya 900 VA untuk rumah tangga mampu. Alasan pemilihannya adalah sistem *monitoring* konsumsi listrik dengan menggunakan IC Renergy 8209 yang dikembangkan untuk skripsi ini, diuji dengan arus maksimal sebesar 4 A atau sekitar 900 VA. Untuk tampilan yang dirancang seperti pada Gambar 4.23.



Gambar 4.24. Tampilan OLED daya aktif

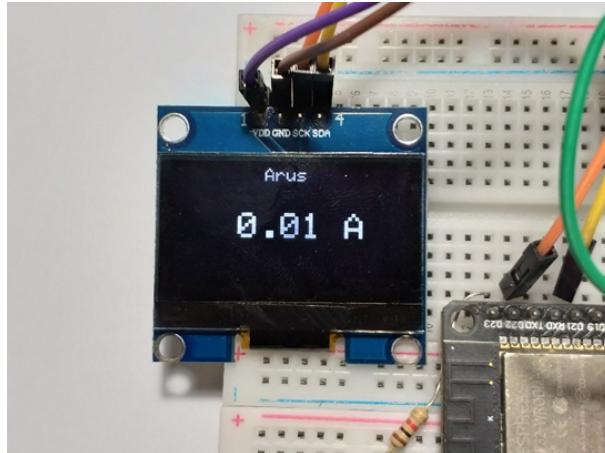
Nilai variabel *counter* bernilai dua, OLED akan menampilkan parameter daya aktif. Nilai dari parameter tersebut didapatkan dari pembacaan yang telah dilakukan oleh IC Renergy 8209 dan dapat diakses pada alamat register PowerPA. Hasil pembacaan tersebut telah dibagi dengan konstanta pembagi agar jangkauan pembacaannya menjadi mendekati yang sebenarnya. Tampilan yang dirancang untuk menampilkan daya aktif dapat dilihat pada Gambar 4.24.



Gambar 4.25. Tampilan OLED tegangan RMS

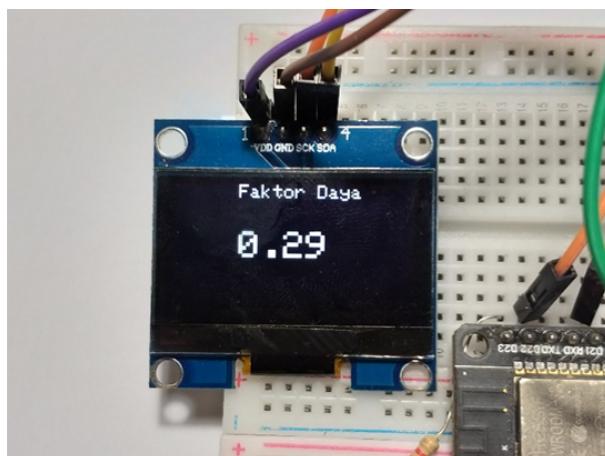
Ketika nilai variabel *counter* adalah tiga, parameter energi listrik yang tertampil adalah tegangan RMS. Data tegangan tersebut diambil dari pembacaan yang telah dilakukan oleh IC Renergy 8209. Data tersebut tersedia pada alamat register URMS. Ukuran huruf keterangan "Tegangan" dibuat berukuran satu. Sementara untuk besaran

nilai tegangan dan satuannya dibuat dengan ukuran dua, seperti pada Gambar 4.25. Pengaturan tersebut dibuat agar pengguna lebih mudah membaca nilai dari tegangan.



Gambar 4.26. Tampilan OLED arus RMS

Kemudian untuk nilai variabel *counter* empat, parameter energi listrik yang ditampilkan adalah arus RMS. Nilai arus RMS tersebut didapatkan dari hasil pembacaan yang dilakukan oleh IC Renergy 8209. Hasil pembacaan tersebut diakses dari alamat register IARMS. Fungsi untuk mengakses alamat register tersebut tersedia pada bagian *looping* agar pembacaan dari arus terus terbarui setiap kali *looping* dilakukan. Tampilan yang telah dirancang untuk parameter arus RMS terlihat pada Gambar 4.26.



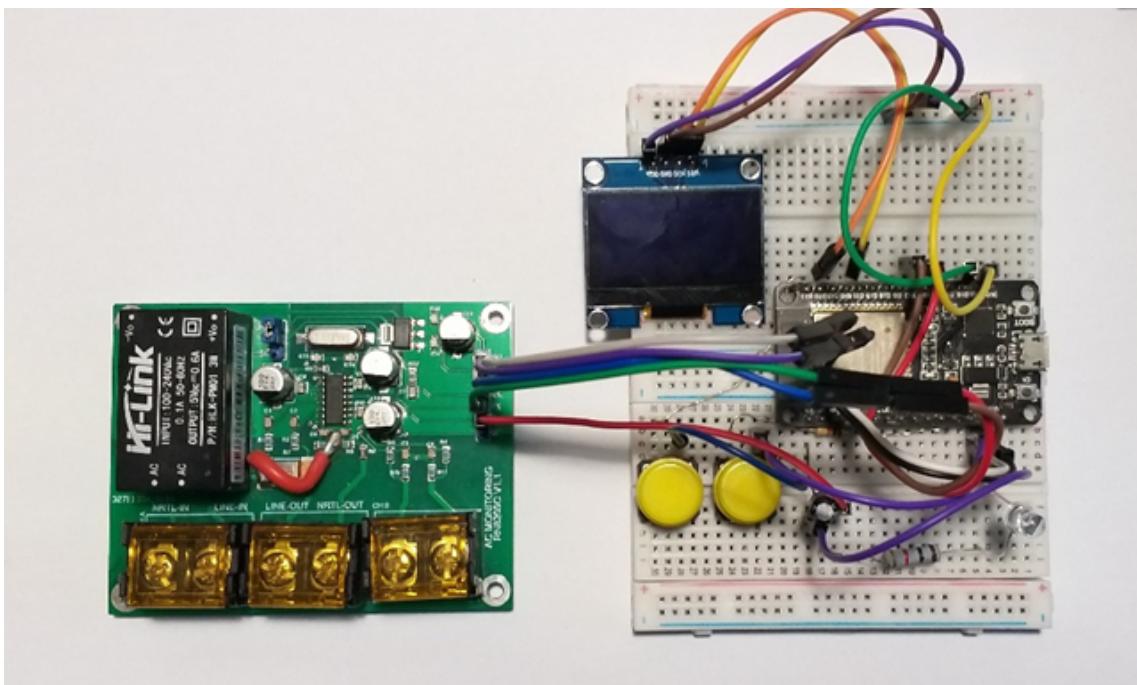
Gambar 4.27. Tampilan OLED faktor daya

Terakhir yaitu ketika nilai variabel *counter* bernilai lima, parameter energi listrik yang akan tertampil pada OLED adalah faktor daya. IC Renergy 8209 tidak menyediakan alamat register yang menyimpan hasil pembacaan faktor daya. Akan tetapi IC Renergy 8209 dapat mendeteksi adanya perbedaan fase antara tegangan dan arus yang menyebabkan adanya faktor daya. Pencarian nilai faktor daya dilakukan dengan formula (4-2). Dengan menggunakan formula tersebut, faktor daya dapat diketahui

dengan cara komputasi. Untuk tampilan dari parameter faktor daya akan seperti pada Gambar 4.27.

$$\text{Faktor daya} = \frac{\text{Daya aktif}}{\text{Tegangan} \cdot \text{Arus}} \quad (4-2)$$

Gambar 4.28 merupakan prototipe dari sistem *monitoring* konsumsi listrik menggunakan IC Renergy 8209 yang dikembangkan pada skripsi kali ini. Prototipe tersebut mengikuti *wiring* diagram yang telah dirancang sebelumnya pada bab sebelumnya. Prototipe yang dibuat sudah dapat merepresentasikan fungsionalitas dari sistem *monitoring* konsumsi listrik yang beredar di pasaran.



Gambar 4.28. Prototipe sistem *monitoring* konsumsi listrik

Firmware final sistem *monitoring* konsumsi listrik menggunakan IC Renergy 8209 pada platform ESP32 telah dikembangkan untuk mengakses fitur-fitur dari IC tersebut. *Firmware* ini dirancang dengan sasaran pengguna yang akan menggunakan sistem ini. Selain itu, *firmware* final yang telah dikembangkan pada skripsi ini, dapat dijadikan acuan untuk pengembangan lebih lanjut hingga skala komersial.

4.5 Perbandingan Hasil Penelitian dengan Hasil Terdahulu

Dalam mengembangkan suatu sistem *monitoring* konsumsi listrik digital, terdapat berbagai jenis IC yang dapat dipilih untuk menjalankan tugas tersebut. IC tersebut berasal dari pabrikan yang berbeda-beda. IC Renergy 8209, pabrikan dari Shenzen RENERGY Micro-Technology, merupakan salah satu dari berbagai IC *monitoring* konsumsi listrik

lainnya. IC lainnya yang berfungsi untuk *monitoring* konsumsi listrik antara lain adalah STPM32 dari pabrikan STMicroelectronics dan ADE7953 dari pabrikan Analog Device.

IC Renergy 8209 dipilih karena harganya yang relatif murah di pasaran. IC Renergy 8209 memiliki harga satuannya berkisar 0.75 dolar amerika. Sementara IC STPM32 dan ADE7953 masing-masing memiliki harga satuannya yaitu 3.48 dan 5.48 dolar amerika. Jika harga tersebut dibandingkan, IC Renergy 8209 memiliki selisih harga yang cukup jauh diantara ketiganya. Pertimbangan harga IC menjadi hal yang cukup berpengaruh karena IC adalah komponen utama dalam sistem *monitoring* konsumsi listrik. Apabila IC tersebut memiliki harga yang mahal, maka produk jadi yang dihasilkan akan lebih mahal pula. Melalui penelitian yang dilakukan pada skripsi ini, membuktikan bahwa dengan menggunakan IC yang lebih murah, sistem *monitoring* konsumsi listrik yang dikembangkan masih dapat memenuhi standar yang berlaku di Indonesia.

Dibalik harga IC Renergy 8209 yang murah, terdapat kompromi yang harus diterima jika dibandingkan dengan dua IC lainnya. IC STPM32 dan ADE7953 memiliki perangkat lunak yang dapat mengakses fitur-fitur serta melakukan kalibrasi pada IC tersebut. Sementara IC Renergy 8209 tidak memiliki dukungan semacam itu. Kemudian IC STPM32 dan ADE7953 memiliki *evaluation board* yang telah disediakan oleh pabrikannya. *Evaluation board* akan berguna bagi pengembang untuk mempelajari IC sebelum pada akhirnya akan memasuki tahap pengembangan lebih lanjut. *Evaluation board* tersebut berisi komponen elektronis untuk mendukung IC dan pin keluaran sebagai tempat keluar pembacaan yang dilakukan IC. Lagi-lagi, IC Renergy 8209 tidak memiliki hal tersebut. Terakhir, IC STPM32 dan ADE7953 memiliki dokumentasi yang cukup lengkap. Misalnya dari *datasheet* IC STPM32 mencapai 91 halaman [27] sementara *datasheet* IC Renergy 8209 hanya 47 halaman. Kuantitas halaman tersebut menunjukkan keterangan-keterangan yang diberikan oleh pabrikan lebih lengkap. Selain itu terdapat dokumen-dokumen pendukung lainnya seperti tata cara dan contoh kalibrasi yang perlu dilakukan dan penjelasan mengenai protokol komunikasi dari IC tersebut. Sementara dokumentasi IC Renergy 8209 hanya berupa *datasheet* saja.

Penelitian pengembangan sistem *monitoring* konsumsi listrik menggunakan IC ADE7758 pada platform Arduino yang dilakukan Guimaraes dan kawan-kawan memiliki kesalahan pembacaan energi sebesar 1.183%. Angka tersebut didapatkan dengan melakukan pengujian pada daya aktif sebesar 600 W. Sementara IC Renergy 8209 yang dikembangkan *firmwarenya* pada skripsi ini, pada daya aktif berkisar 890 W, kesalahan pembacaan yang terjadi adalah 0.22%. Dari segi hasil akhir dari penelitian tersebut dan penelitian yang dilakukan pada skripsi ini menunjukkan bahwa sistem *monitoring* konsumsi listrik yang menjadi penelitian pada skripsi ini memiliki pembacaan yang lebih baik. Akan tetapi kesalahan pembacaan tidak serta merta

merupakan kesalahan dari sisi IC, kalibrasi yang dilakukan juga berperan dalam meminimalisir kesalahan pembacaan. Selain itu, dari sisi harga, IC Renergy 8209 memiliki harga yang lebih murah dari IC ADE7758. ADE7758 memiliki harga satuan 19.1 dolar amerika atau sekitar 25 kali lipat dari harga satuan IC Renergy 8209. Akan tetapi ADE7758 dapat digunakan untuk kebutuhan listrik tiga fase. Sementara IC Renergy 8209 hanya bisa pada listrik satu fase saja. Jika dilihat dari sisi *development board*, penggunaan Arduino sebagai *development board* pada penelitian tersebut belum memiliki fitur WiFi sementara ESP32 yang digunakan pada skripsi ini sudah memiliki fitur tersebut.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pengembangan *firmware* sistem *monitoring* konsumsi listrik menggunakan *metering IC* Renergy 8209 pada platform ESP32 dilakukan dengan empat tahap. Pertama, mengembangkan *firmware* kalibrasi dan pengujian. Langkah tersebut dilakukan sebagai langkah awal dalam melakukan komunikasi antara IC dan pengembang. Selain itu, *firmware* kalibrasi dan pengujian memiliki sasaran pengguna yaitu pengembang dan kalibrator untuk memudahkan tugasnya dalam melakukan kalibrasi IC Renergy 8209. *Firmware* kalibrasi dan pengujian ini juga dirancang agar pengembang dan kalibrator dapat mengeksplorasi fitur yang dimiliki oleh IC ini. Pada *firmware* ini, fitur tersebut adalah komunikasi UART dengan IC Renergy 8209, membaca dan memberi nilai register, inisialisasi konfigurasi, dan tampilan pada OLED.

Tahap kedua adalah kalibrasi IC Renergy 8209. Kalibrasi diawali dengan melakukan konfigurasi untuk menentukan mekanisme pembacaan dan karakteristik yang diinginkan dari sistem *monitoring* konsumsi listrik yang akan dikembangkan. Karakteristik tersebut diatur melalui alamat-alamat register konfigurasi dari IC Renergy 8209.

Setelah itu dilakukan kalibrasi daya aktif yang terdiri dari tiga proses yaitu kalibrasi *gain*, fase, dan *offset* daya aktif. Kalibrasi *gain* daya aktif mampu meminimalisir kesalahan pembacaan energi listrik dengan memberikan suatu konstanta kalibrasi *gain* dengan cara perhitungan yang telah ditentukan oleh pabrikan IC pada *datasheet*. Perbaikan kesalahan yang terjadi dari semula 3.368% menjadi 0.04%. Kemudian pada nilai faktor daya yang semakin rendah, terjadi peningkatan kesalahan pembacaan energi listrik. Untuk menanggulangi hal tersebut, dilakukan kalibrasi fase yang akan menghasilkan suatu konstanta kalibrasi fase daya aktif. Pada kondisi tersebut perbaikan kesalahan yang terjadi dari semula 0.69% menjadi -0.06%. Proses ketiga adalah kalibrasi *offset* daya aktif karena ada kemungkinan terjadi *offset* pada pembacaan yang dilakukan oleh IC Renergy 8209. Akan tetapi pada skripsi ini tidak terdapat *offset* yang nilainya melebihi batas yang telah ditentukan sehingga konstanta kalibrasi *offset* daya aktif dibuat nol. Ketiga nilai konstanta tersebut akan disimpan ke dalam alamat register kalibrasi daya aktif terkait.

Kalibrasi daya reaktif adalah kalibrasi yang dilakukan setelah kalibrasi daya aktif. Terdapat dua proses dalam melakukan kalibrasi daya reaktif yaitu kalibrasi fase dan *offset* daya reaktif. Kalibrasi fase daya reaktif dilakukan untuk meminimalisir kesalahan pembacaan daya dan energi reaktif. Kalibrasi tersebut akan menghasilkan

konstanta kalibrasi fase daya reaktif. Dengan memasukkan konstanta tersebut, perbaikan yang terjadi dari semula -1.12% menjadi 0.83%. Kemudian apabila terdapat *offset* pada pembacaan daya reaktif, kalibrasi *offset* perlu dilakukan. Akan tetapi pada penelitian kali ini, nilai *offset* masih di bawah batas yang ditentukan sehingga nilai konstanta *offset* daya reaktif diatur menjadi nol. Kedua nilai tersebut disimpan ke dalam alamat register kalibrasi daya reaktif terkait.

Kalibrasi terakhir adalah kalibrasi RMS. Pada *channel* arus A, dimungkinkan terjadi *offset* dan apabila hal itu terjadi, dilakukan kalibrasi *offset channel* arus A. Akan tetapi *offset* yang terjadi masih di bawah batas yang telah ditentukan, maka konstanta kalibrasi *offset* arus diatur nilainya menjadi nol. Kemudian hasil pembacaan parameter-parameter energi listrik yang dikeluarkan oleh IC Renergy 8209 nilainya sangat besar karena mengalami perbesaran internal dari *gain* yang telah diatur pada konfigurasi dan perkalian atau penjumlahan dengan konstanta-konstanta kalibrasi. Oleh karena itu pada kalibrasi RMS, akan dicari konstanta pembagi agar pembacaan yang dilakukan IC Renergy 8209 berada pada jangkauan nilai yang sesungguhnya. Kalibrasi telah selesai.

Tahap ketiga adalah melakukan pengujian prototipe sistem berdasarkan standar yang berlaku di Indonesia. Standar tersebut diatur dalam Surat Keputusan Direktur Jenderal Perlindungan Konsumen dan Tata Niaga Nomor 161 Tahun 2019 tentang Syarat Teknis Meter kWh. Dalam dokumen tersebut disebutkan ada empat pengujian yang harus dilakukan agar sistem *monitoring* konsumsi listrik terbukti secara fungsionalitas bekerja. Keempat pengujian tersebut adalah pengujian tanpa beban, arus mula, variasi arus, dan register. Keempat pengujian tersebut memiliki batas kesalahan yang diizinkan (BKD) yang nilainya disesuaikan dengan jenis dan besaran arus yang digunakan pada pengujian tersebut. Berdasarkan pengujian register, didapatkan akurasi pembacaan energi listrik 0.16%. Keempat pengujian yang telah dilakukan, semuanya dinyatakan lolos dengan tingkat akurasi kelas 2 (A). Lulosnya tahap ini memberikan kepastian bahwa prototipe sistem dapat memenuhi standar nasional dan dapat dilakukan pengembangan lebih lanjut.

Tahap keempat atau tahap terakhir adalah pengembangan *firmware* final. *Firmware* ini memiliki sasaran yaitu pengguna yang akan menggunakan sistem *monitoring* konsumsi listrik ini ke depannya. Dari *firmware* pengujian kalibrasi dan pengujian terdapat fitur-fitur yang kembali diterapkan, disesuaikan dengan target pengguna, dan baru dikembangkan pada *firmware* final. Fitur yang disesuaikan adalah tampilan OLED yang menampilkan parameter-parameter energi listrik menjadi satu layar satu parameter. Sementara fitur yang baru dikembangkan bertujuan untuk mengganti tampilan dengan cara menekan tombol, menyimpan nilai pembacaan energi listrik pada memori agar tidak hilang walau catu daya terputus, dan tombol untuk

melakukan reset pembacaan energi listrik.

Dengan melakukan keempat tahap tersebut prototipe sistem *monitoring* konsumsi listrik menggunakan *metering* IC Renergy 8209 pada platform ESP32 dapat dikembangkan. Selain itu dalam empat tahap tersebut, kalibrasi telah dilakukan agar pembacaan yang dilakukan oleh IC menjadi tepat dan akurat. Pengujian juga dilakukan dengan mengacu pada standar yang berlaku di Indonesia untuk memberikan kepastian bahwa sistem telah layak untuk dikembangkan lebih lanjut.

5.2 Saran

Penulis menyadari setidaknya terdapat tiga hal yang dapat dikembangkan lebih lanjut dalam penelitian ini namun tidak bisa dilakukan karena keterbatasan dari penelitian yang dilakukan oleh penulis. Pertama adalah otomasi kalibrasi. Agar hal tersebut bisa dilakukan, perlu dikembangkan *testbench* yang dapat beroperasi secara otomatis sesuai dengan urutan kalibrasi yang dilakukan. *Testbench* tersebut dapat mengatur arus dan faktor daya sesuai dengan yang diperlukan oleh IC. Kemudian dapat menghitung nilai-nilai konstanta kalibrasi dan mengirim nilai tersebut secara otomatis ke sistem yang dikembangkan. Jika hal tersebut dikembangkan, maka akan sangat bermanfaat pada pengembangan untuk keperluan komersil.

Hal kedua yang dapat dikembangkan adalah penerapan konsep *internet of things* ke dalam sistem *monitoring* konsumsi listrik. Agar konsep tersebut dapat diterapkan, dari sisi *firmware* perlu membuat fitur yang menghubungkan pembacaan yang dilakukan oleh perangkat keras dapat terkirim datanya ke *cloud* atau *database*. Kemudian dari *cloud* atau *database*, data tersebut diolah dan dikirim ke suatu antarmuka pengguna agar pengguna dapat mengakses dan mengontrol sistem *monitoring* konsumsi listrik dari jarak jauh.

Hal ketiga adalah *hardware* sistem *monitoring* konsumsi listrik ukurannya dapat dibuat menjadi lebih kecil. Perengkasan ukuran sistem dilakukan agar pada penggunaannya nanti, pengguna dapat lebih mudah menempatkan alat tersebut. Pembuatan *hardware* sistem yang lebih kecil dilakukan dengan merancang ulang rangkaian elektronika dan komponen-komponennya. *Hardware* tersebut kemudian dapat ditempatkan pada suatu *casing* agar tahan dari debu dan benturan.

Ketiga hal tersebut merupakan yang penulis temukan selama penelitian pada skripsi ini dan dapat dikembangkan untuk penelitian-penelitian selanjutnya terkait dengan topik yang penulis angkat.

DAFTAR PUSTAKA

- [1] University of Calgary. (2013) Energy Education. [Online]. Available: https://energyeducation.ca/encyclopedia/Electrical_energy
- [2] A. Ahdiat. (2023) databoks. KataData. [Online]. Available: <https://databoks.katadata.co.id>
- [3] Kementerian ESDM RI, “Peraturan menteri esdm nomor 28 tahun 2016,” Tech. Rep., 2016.
- [4] T. Nusa, S. Sompie, and M. Rumbayan, “Sistem Monitoring Konsumsi Energi Listrik Secara Real Time Berbasis Mikrokontroller,” *E-journal Teknik Elektro dan Komputer*, vol. 4, no. 5, pp. 19–26, 2015.
- [5] A. M. F. Guimarães, T. T. Freitas, H. Griner, and T. H. S. de Almeida, “Smart Energy Monitoring System with ADE7758 IC,” *2015 5th International Youth Conference on Energy (IYCE)*, 2015.
- [6] M. M. Haque, M. K. Hossain, M. M. Ali, and M. R. I. Sheikh, “Microcontroller Based Single Phase Digital Prepaid Energy Meter for Improved Metering and Billing System,” *International Journal of Power Electronics and Drive System (IJPEDS)*, vol. 1, no. 2, pp. 139–147, 2011.
- [7] Ardiansyah, “Monitoring Daya Listrik Berbasis IoT (Internet of Things),” Master’s thesis, Universitas Islam Indonesia, Yogyakarta, 2020.
- [8] Direktur Jenderal Perlindungan Konsumen dan Tertib Niaga, “Surat Keputusan Direktur Jenderal Perlindungan Konsumen dan Tertib Niaga Nomor 161 Tahun 2019 Tentang Syarat Teknis Meter kWh,” Kementerian Perdagangan Republik Indonesia, Jakarta, Tech. Rep., 2019.
- [9] TSE Indonesia. (2019) Kwh meter. [Online]. Available: <http://tse-indonesia.id/kwh-meter/>
- [10] E. Ie, A. P. Launuru, and J. Tupalessy, “Analisis Akurasi kWh Meter Analog Pasca Bayar dan kWh Meter Digital Prabayar,” in *Seminar Nasional Terapan Riset Inovatif ke 8*, vol. 8, no. 1, 2022, pp. 198–205.
- [11] Acrel, *ADL10-e*, Acrel, Shanghai, 2019.
- [12] R. D. Knight, *Fundamentals of Circuits*. Illinois: Pearson Education, 2013.
- [13] CalcKit. (2023) Power Triangle. [Online]. Available: <https://calckit.io/tool/electronics-power-triangle>
- [14] International Electrotechnical Commission, “Electricity metering equipment - Particular requirements - Part 21: Static meters for AC active energy (classes 0,5, 1 and 2),” *IEC 62053-21*, 2020.
- [15] Shenzen RENERGY Micro-Technology Co, *RN8209C/RN8209D User Manual*, Shenzen RENERGY Micro-Technology Co, Shenzen, 2016.

- [16] E. Moulin, “Measuring Reactive Power in Energy Meters,” *Metering International*, vol. 1, pp. 52–54, 2002.
- [17] G. Stringham, *Hardware/Firmware Interface Design : Best Practices for Improving Embedded Systems Development*. Massachusetts: Newnes, 2010.
- [18] E. Peña and M. G. Legaspi, “UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter,” *Analog Dialogue*, vol. 54, no. 4, pp. 1–5, 2020.
- [19] V. O. Oner, *Developing IoT Projects with ESP32*. Birmingham: Packt Publishing, 2021.
- [20] Espressif. (2023) ESP32 : ESP-IDF Programming Guide. [Online]. Available: <https://docs.espressif.com/>
- [21] R. Teja. (2021) ESP32 Pinout | ESP-WROOM-32 Pinout. [Online]. Available: <https://www.electronicshub.org/esp32-pinout/>
- [22] Arduino. (2021) About arduino. [Online]. Available: <https://www.arduino.cc/en/about>
- [23] Y. Dinata, *Arduino itu Pintar*. Jakarta: Elex Media Komputindo, 2016.
- [24] GFUVE, *GF312B*, Beijing GFUVE Electronics Co.,Ltd., Beijing, 2023.
- [25] Direktur Jenderal Perlindungan Konsumen dan Tertib Niaga, “Surat Keputusan Direktur Jenderal Perlindungan Konsumen dan Tertib Niaga Nomor 24 Tahun 2021 Tentang Syarat Teknis Standar Ukuran Metrologi Legal Besaran Energi Listrik,” Kementerian Perdagangan Republik Indonesia, Jakarta, Tech. Rep., 2021.
- [26] R. Santos. (2018) ESP32 Flash Memory – Store Permanent Data (Write and Read). [Online]. Available: randomnerdtutorials.com/esp32-flash-memory/
- [27] STMicroelectronics, *STPM32, STPM33, STPM34 Datasheet*, Geneva, 2022.