

MONITORING ENERGI METER BERBASIS MODBUS RTU

PROYEK AKHIR

Laporan akhir ini dibuat dan diajukan untuk memenuhi salah satu syarat kelulusan
Diploma III Politeknik Manufaktur Negeri Bangka Belitung



Disusun oleh :

Gira Yunikristiyanti NIRM : 0031740

Latipa Kusuma Dawe NIRM : 0031719

**POLITEKNIK MANUFAKTUR NEGERI BANGKA
BELITUNG**

2020

LEMBAR PENGESAHAN

MONITORING ENERGI METER BERBASIS MODBUS RTU

Oleh :

Gira Yunikristiyanti NIRM : 0031740
Latipa Kusuma Dawe NIRM : 0031719

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Diploma III Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1



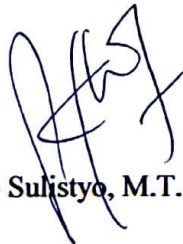
Indra Dwisaputra, M.T.

Pembimbing 2



M. Iqbal Nugraha, M.Eng.

Penguji 1



Eko Sulistyono, M.T.

Penguji 2



Ocsirendi, M.T

Penguji 3



Yudhi, M.T.

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan dibawah ini :

Nama Mahasiswa 1 : Gira Yunikristiyanti

NIRM : 0031740

Nama Mahasiswa 2 : Latipa Kusuma Dawe

NIRM : 0031719

Dengan Judul : Monitoring Energi Meter Berbasis Modbus Rtu

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 18 Agustus 2020

Nama Mahasiswa

Tanda Tangan

1. Gira Yunikristiyanti



2. Latipa Kusuma Dawe



ABSTRAK

Perkembangan sistem kontrol dipengaruhi oleh sistem komunikasi data untuk mengirimkan data ke kontroler. Energi meter berfungsi untuk memonitoring dan membatasi penggunaan listrik secara keseluruhan. Penggunaan Energi Meter yang berbasis *modbus* paling banyak digunakan pada sektor gedung dan industri. Monitoring penggunaan energi listrik tidak cukup hanya dengan menggunakan kwh meter, diperlukan sistem komunikasi data yang dapat monitoring penggunaan energi listrik. *Protocol modbus* adalah salah satu sistem komunikasi data yang banyak digunakan pada sistem kontrol dengan port serial RS-232 dan RS-485. Komunikasi RS-485 digunakan untuk memonitoring energi meter berbasis modbus rtu yang dapat ditampilkan pada web menggunakan *local host*. Alat ini dapat menampilkan hasil pembacaan tegangan, arus, daya aktif, daya aktif total, daya reaktif, daya semu, frekuensi dan grafik. Dari hasil uji coba nilai yang dilakukan pada kwh, pada web dan perhitungan memiliki selisih nilai daya sebesar 0.62 watt dengan persentase error 2.09%. Hal tersebut disebabkan oleh interval waktu yang digunakan saat pengujian. Untuk merefresh hasil monitoring pada web membutuhkan waktu 5 menit. Dengan adanya alat ini penggunaan energi listrik dapat di monitoring dengan baik.

Kata Kunci : Energy Meter, Monitoring, Modbus, RTU, Web

ABSTRACT

The evolution of the control system is influenced by the data communication system to send a data to the controller. The functions of Energy meter to monitoring and limiting all electricity usage. The usage of energy meters based on modbus is most widely used in the building sector and industry. Monitoring electrical energy is not enough just by using a kWh meter, it required a data communication system that can monitor the use of electrical energy. The modbus protocol is a data communication system that widely used in control systems with serial ports RS-232 and RS-485. RS-485 communication is used for monitoring the energy meter based on modbus rtu which can be displayed on the web using a local host. This tool can display the readings of voltage, current, active power, total active power, reactive power, apparent power, frequency and graphs. From the results of the test values performed on kwh, on the web and calculations has a difference in power value 0.62 watts with an error percentage 2.09%. It is caused by the time interval used when testing. To refresh the monitoring results on the web takes 5 minutes. With this tool the use of electrical energy can be monitored properly.

Keywords: Energy Meter, Monitoring, Modbus, RTU, Web

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh. Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyusun laporan proyek akhir ini dengan judul “Monitoring Energi Meter Berbasis Modbus Rtu” sehingga penulis dapat menyelesaikan salah satu syarat dari kelulusan Program Diploma III Politeknik Manufaktur Negeri Bangka Belitung Program Studi Teknik Elektro dan Informatika.

Dalam penyusunan laporan ini banyak pihak yang membantu penulis mulai dari pengerjaan proyek akhir hingga penyusunan laporan proyek akhir dengan tepat waktu dengan itu penulis menyampaikan ucapan terima kasih kepada:

1. Allah SWT yang telah memberikan segala nikmat dan karunianya yang sangat luar biasa untuk penulis sehingga penyusunan ini berjalan dengan baik.
2. Orang tua tercinta serta seluruh keluarga yang dengan penuh keikhlasan dan kesabran memberikan bantuan material dan spiritual untuk penulis.
3. Bapak Sugeng Ariyono, M.Eng, Ph.D. Selaku Direktur di Politeknik Manufaktur Negeri Bangka Belitung tahun 2016-2020 dan Bapak I Made Andik Setiawan, M.Eng, Ph.D Selaku Direktur di Politeknik Manufaktur Negeri Bangka Belitung tahun 2020-2024 memberikan kemudahan dalam menyelesaikan pendidikan.
4. Bapak Indra Dwisaputra, M.T. dan Bapak M. Iqbal Nugraha, M.Eng. Selaku dosen pembimbing yang telah membimbing, mengarahkan dan memberi saran-saran dalam pembuatan dan penyusunan laporan proyek akhir ini.
5. Bapak Yudho Andriwianto selaku Direktur Utama PT. Pratama Motivasi Mandiri yang telah meberikan kesempatan dalam melakukan praktik kerja lapangan sehingga penulis mendapatkan bantuan secara material dan dukungan dalam pengerjaan proyek akhir.
6. Bapak Agus Dwi Yunto sebagai pembimbing Praktik Kerja Lapangan yang telah memberikan banyak ilmu serta bantuan secara material yang mendukung proyek akhir penulis.

7. Bapak Gunawan sebagai engineering manager yang telah memberikan banyak ilmu dan pengetahuan selama Praktik kerja Lapangan sehingga dapat diterapkan pada proyek akhir.
8. Seluruh karyawan PT. Pratama Motivasi Mandiri yang mendukung penuh dalam proses pengerjaan proyek akhir.
9. Serta teman-teman yang selalu memberikan support selama proyek akhir ini yang selalu berjuang bersamasama.

Penulis menyadari bahwa dalam penulisan proyek akhir ini masih memiliki kekurangan dan jauh dari kesempurnaan. Oleh karena itu penulis mengharpakan kritik dan saran yang membangun sehingga apa yang menjadi kesalahan penulis dapat diperbaiki untuk penulisan yang akan datang. Besar harapan penulis semoga makalah ini dapat bermanfaat bagi penulis dan pembaca serta proyek akhir yang dibuat oleh penulis juga dapat bermanfaat untuk perkembangan ilmu teknologi pada umumnya.

Semoga Allah senantiasa melimpahkan taufiq dan hidayah-Nya kepada penulis dan semua pihak yang telah membantu dalam pembuatan proyek akhir dan penulisan makalah proyek akhir ini, Wassalamua'laikum Warahmatullahi Wabarakatuh.

Sungailiat, 18 Agustus 2020

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	i
PERNYATAAN BUKAN PLAGIAT	ii
ABSTRAK	ii
ABSTRACT	iv
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
BAB I	1
PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan	3
BAB II.....	5
DASAR TEORI.....	5
2.1 Energi Meter ABB B23.....	5
2.2 Energi Listrik	9
2.3 Komunikasi Energi Meter Modbus.....	10
2.4 Ethernet Shield.....	11
2.5 RS485.....	12
BAB III.....	13

METODE PELAKSANAAN.....	13
3.1 Rancangan Sistem Hardware	13
3.2 Rancangan Tampilan Sistem Monitoring	14
3.3 Rancangan Kontruksi Mekanik.....	16
3.4 Skematik sistem Elektrik Hardware.....	17
3.5 Sistem Kontroler	18
3.6 Algoritma Sistem	19
4.1 Pembuatan Hardware Mekanik Alat Monitoring.....	21
4.2 Perancangan Sistem Pada Web	24
4.2.1 Menu Utama.....	24
4.2.2 Menu Proyek Akhir.....	24
4.2.3 Daftar Menu Proyek Akhir.....	25
4.2.4 Menu Datalog txt.....	25
4.2.5 Menu Grafik.....	25
4.3 Perancangan dan Pembuatan Software	26
4.4 Pengujian Alat Monitoring Energi Meter Berbasis Modbus RTU	26
4.5 Analisa Hasil Uji Coba.....	35
BAB V.....	39
KESIMPULAN DAN SARAN	39
5.1 Kesimpulan	39
5.2 Saran.....	39
DAFTAR PUSTAKA	40
LAMPIRAN	41
LAMPIRAN 1	42
DAFTAR RIWAYAT HIDUP	42
LAMPIRAN 2	43

PROGRAM..... 43

DAFTAR TABEL

Tabel 3.1 Fungsi Dari Setiap Komponen.....	14
Tabel 4.1 Perbandingan Pada Hasil Uji Coba.....	28
Tabel 4.2 Perbandingan Pada Hasil Uji Coba Daya Semu dan Daya Reaktif	29
Tabel 4.3 Tabel Analisa Perhitungan	38

DAFTAR GAMBAR

Gambar 1.1 Rancang Bangun Monitoring Energi Listrik Menggunakan Sms Berbasis Mikrokontroler Atmega328 (Putra, 2003).....	2
Gambar 2.1 Bagian Energi Meter Modbus ABB (23/24, t.thn.).....	5
Gambar 2.2 Diagram Hardware	11
Gambar 2.3 Ethernet Shield (Putra, p. 2006).....	12
Gambar 2.4 RS485 (Imam, 2016).....	12
Gambar 3.1 Sistem Alat Monitoring.....	13
Gambar 3.2 Rancangan Tampilan Sistem Web	15
Gambar 3.3 Rancangan Kontruksi Mekanik.....	16
Gambar 3.4 Skematik sistem Elektrik Hardware.....	17
Gambar 3.5 sistem kontroler	18
Gambar 3.6 Pcb Pengkabelan	18
Gambar 3.7 Flowchart sistem	20
Gambar 4.1 Panel.....	21
Gambar 4.2 Pemotongan Atau Pengeboran Panel	22
Gambar 4.3 Pemasangan instalasi elektrik hardware.....	22
Gambar 4.4 Pemasangan Komponen pada panel.....	23
Gambar 4.5 Assesmenting Alat	23
Gambar 4.6 Rancangan Tampilan Pada Web	24
Gambar 4.7 Daftar Menu Proyek Akhir.....	24
Gambar 4.8 Daftar Menu Proyek Akhir.....	25
Gambar 4.9 Gambar Menu Datalogtxt.....	25
Gambar 4.10 Grafik	26
Gambar 4.11 Beban 1 Motor 3 Phasa	27
Gambar 4.12 Beban 2 Motor 3 Phasa	27
Gambar 4.13 Beban 3 Motor 1 Phasa	27
Gambar 4.14 Hasil Monitoring Pada Web.....	28
Gambar 4.15 Tegangan Beban R	29

Gambar 4.16 Tegangan Beban S.....	29
Gambar 4.17 Tegangan Beban T	30
Gambar 4.18 Arus Beban R	30
Gambar 4.19 Arus Beban S.....	30
Gambar 4.20 Arus Beban T	31
Gambar 4.21 Daya Aktif R	31
Gambar 4.22 Daya Aktif S.....	31
Gambar 4.23 Daya Aktif T.....	32
Gambar 4.24 Daya Aktif Total.....	32
Gambar 4.25 Frekuensi	32
Gambar 4.26 Daya Reaktif R.....	33
Gambar 4.27 Daya Reaktif S	33
Gambar 4.28 Daya Reaktif T	33
Gambar 4.29 Daya Semu R.....	34
Gambar 4.30 Daya Semu S	34
Gambar 4.31 Daya SemuT.....	34
Gambar 4.32 Power Faktor ($\text{Cos } \varphi$).....	35

BAB I

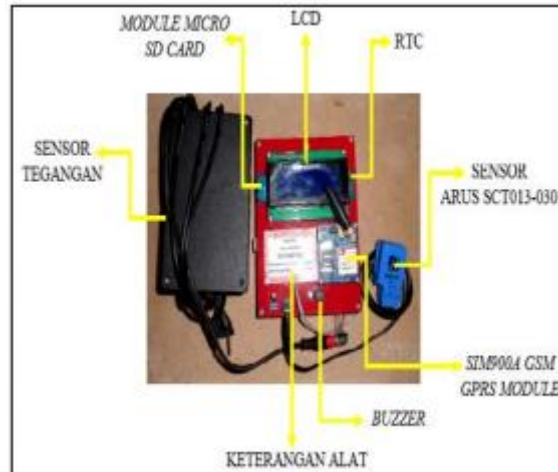
PENDAHULUAN

1.1 Latar Belakang

Energi meter berfungsi untuk memonitor dan membatasi penggunaan listrik secara keseluruhan. Untuk melakukan monitoring arus, tegangan dan penggunaan energi listrik pada jaringan listrik tidak cukup hanya dengan menggunakan kwh meter, sehingga diperlukan sistem informasi yang bisa digunakan untuk monitoring penggunaan energi listrik agar bisa melakukan manajemen listrik yang lebih baik maka diperlukan suatu sistem yang bisa memonitor pemakaian energi listrik. Perkembangan sistem kontrol tidak akan lepas dari sistem komunikasi data. Untuk mengirimkan data ke kontroler, komunikasi antar kontroler, mengirimkan data dari kontroler ke komputer, atau semua yang membutuhkan komunikasi data. Dalam era digital, sistem kontrol biasanya merujuk pada otomatisasi sistem kontrol yang digunakan kini digeser dan digantikan oleh sistem kontrol otomatis. Salah satu sistem komunikasi data yang banyak digunakan pada sistem kontrol adalah *protocol modbus*[1].

Alat monitoring energi meter sebelumnya telah ada dengan judul proyek akhir Rancang Bangun Alat Penghitung Biaya Pemakaian Listrik Secara Digital yang dibuat oleh Muhamad Fajri dan Riska Paresda Mahasiswa Politeknik Manufaktur Negeri Bangka Belitung. Alat ini dibuat untuk memonitoring penggunaan energi listrik dengan mengkalkulasikannya pada biaya media pada monitoring ini menggunakan SMS. Pemakaian energi listrik dengan kalkulasi biaya penggunaan akan dikirimkan ke ponsel melalui via SMS. Selain itu alat yang serupa dengan judul Rancang Bangun Monitoring Energi Listrik Menggunakan Sms Berbasis Mikrokontroler Atmega328 yang dibuat oleh Raden Ajeng Gusti Ramadhianti, Ir. Cok Gede Indra Partha, I Gusti Agung Pt Raka Agung Program Studi Teknik Elektro, Fakultas Teknik Universitas Udayana. Berikut ini merupakan gambar dari

alat Rancang Bangun Monitoring Energi Listrik Menggunakan Sms Berbasis Mikrokontroler Atmega328



Gambar 1.1 Rancang Bangun Monitoring Energi Listrik Menggunakan Sms Berbasis Mikrokontroler Atmega328 (Putra, 2003)

Alat monitoring energi listrik menggunakan SMS berbasis mikrokontroler Atmega 328 telah dapat dibangun dan sistem pembacaan notifikasi penggunaan energi listrik sudah dapat dilakukan melalui SMS. Peralatan terdiri dari rangkaian utama yaitu rangkaian mikrokontroler Atmega 328 sebagai pengolah data, module GSM sebagai pengirim dan penerima instruksi melalui SMS, module RTC sebagai penunjuk tanggal dan waktu secara real time, module micro SD card sebagai datalogger, rangkaian sensor arus sebagai pembaca nilai arus dan rangkaian sensor tegangan sebagai pembaca nilai tegangan. Semua peralatan sudah bekerja sesuai dengan perencanaan. Data logger yang dirancang menyimpan data arus, tegangan, energi terpakai dan biaya yang harus dibayarkan selama 24 jam, file hasil data logger tersimpan oleh SD card berkapasitas 2 GB dengan ukuran file sebesar 3 KB. Sedangkan alat monitoring yang penulis buat merupakan alat monitoring energi listrik yang ditampilkan pada sebuah web dengan menggunakan lokal *address* yang memonitoring tegangan, arus, daya aktif, daya aktif total, daya reaktif, daya semu, frekuensi, dan juga grafik dengan menggunakan energi meter modbus.[6]

Protocol modbus dibuat oleh perusahaan PLC bernama Modicon tahun 1979 dan sampai sekarang menjadi salah satu protokol komunikasi standar yg dipakai

dalam otomatisasi pengelolaan gedung, proses industri dan lain-lain. Modbus juga merupakan metode yang digunakan untuk mengirimkan data atau informasi dari suatu alat dan kemudian diterima oleh kontroler, pada saat itulah terjadi komunikasi di antara kedua peralatan tersebut.[1]

Penggunaan Energi Meter yang berbasis Modbus biasanya paling banyak digunakan pada gedung dan sektor industri. Dimana pada sistem kontrol otomatis ini mutlak menggunakan sistem komunikasi data. Sehingga di perlukan alat untuk mengirimkan data ke kontroler atau komunikasi antar kontroler.[2]

1.2 Rumusan Masalah

Adapun rumusan pada proyek akhir ini adalah sebagai berikut :

1. Bagaimana merancang sistem komunikasi energi meter Modbus ?
2. Bagaimana membuat tampilan sistem monitoring pada web ?
3. Bagaimana cara mengakses penggunaan energi meter di web ?
4. Bagaimana cara menyimpan hasil monitoring pada sd card ?

1.3 Batasan Masalah

Adapun batasan masalah pada proyek akhir ini adalah sebagai berikut :

1. Menggunakan beban 2 buah motor 3 phasa dan 1 buah motor 1 phasa Tipe Y90S-2 hubungan Y 380 Volt, 3.4 Amper untuk motor 3 phasa, untuk tipe 1 phasa SE2662-3F
2. Diakses dengan menggunakan *local address*
3. Memonitoring tegangan, arus, daya aktif , daya aktif total, daya Reaktif, daya semu, frekuensi, dan juga grafik.
4. Tampilan grafik merupakan hasil dari penggunaan daya aktif total.

1.4 Tujuan

Alat ini digunakan untuk mendata dan menganalisa penggunaan energi listrik yang digunakan. Adapun hal-hal yang akan menjadi tujuan penulis dalam membuat alat ini ialah sebagai berikut :

1. Membuat hardware sistem monitoring energi meter yang berbasis modbus RTU.
2. Memonitoring penggunaan energi listrik.

3. Membuat tampilan nilai pemakaian daya listrik dalam bentuk web.
4. Penggunaan daya listrik dapat di akses dengan menggunakan sistem *ip address* di PC.

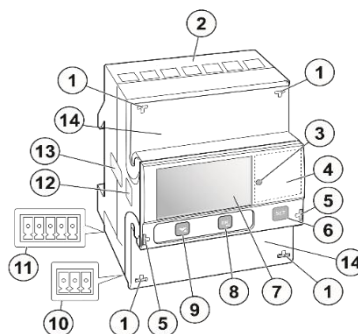
BAB II

DASAR TEORI

2.1 Energi Meter ABB B23

ABB B23 adalah alat pengukur listrik 3 fase yang dilengkapi dengan layar backlit yang memberikan informasi tentang energi aktif, arus, tegangan, faktor daya dan frekuensi. Dengan menggunakan ABB B23 bersama-sama dengan Modul Konektivitas Meter eksternal CMeX20 atau CMeX20w Elvaco, data meter dapat dikonversi ke M-Bus / Nirkabel M-Bus dan dikirimkan ke sistem penerima. Untuk mengukur besaran daya aktif digunakan alat ukur yang bernama Kwh meter/energi meter. Sehingga saat ini sistem monitoring sudah semakin maju seiring dengan berkembangnya teknologi. Modbus merupakan serial komunikasi yang terdiri dari komunikasi RTU dan juga TCP IP. RTU merupakan komunikasi serial yang diakses dengan lokal address sedangkan TCP IP serial komunikasi yang diakses dengan jangkauan yang luas menggunakan jaringan internet atau wifi.[7]

Salah satunya pada penggunaan mikrokontroler yang semakin pesat untuk pengaplikasiannya dalam memonitoring. Sistem pemantau tersebut dilakukan bertujuan untuk dapat mengawasi segala aktifitas atau kegiatan yang terjadi pada suatu ruangan atau daerah tertentu. Pentingnya sistem monitoring energi listrik sebagai support system (sistem pendukung) bahwa pemakaian listrik sangat perlu dikendalikan sehingga bisa menggunakan listrik secara efisien dan hemat. Berikut ini gambar 2.1 bagian dari energi meter modbus ABB.[3]



Gambar 2.1 Bagian Energi Meter Modbus ABB (23/24, t.thn.)

Tabel 2.1 Penjelasan Bagian Dari Energi Meter ABB

Item	Deskripsi	komments
1	Poin penyegelan	Benang segel digunakan untuk menyegel meteran.
2	Blok terminal	Terminal untuk semua tegangan dan arus
3	LED	Berkedip sebanding dengan energi yang diukur
4	Data produk	Berisi data tentang tipe meteran
5	Titik penyegelan untuk penutup yang dapat disegel	Benang segel digunakan untuk menutup penutup.
6	Setel tombol	Masuk ke mode konfigurasi
7	Tampilan	LCD untuk pembacaan meter
8	Tombol OK / Keluar	Tekan untuk melakukan tindakan atau untuk memilih menu. Tekan dan tahan untuk keluar ke menu sebelumnya atau untuk berganti antara menu default dan utama.
9	Tombol OK / Keluar	Toggle down / up (beralih / kanan di menu utama): tekan untuk turun, tekan dan tahan untuk naik.
10	Terminal untuk koneksi komunikasi	
11	Terminal untuk koneksi input / output	
12	Antarmuka komunikasi optik	Untuk komunikasi IR
13	Label penyegelan	Di kedua sisi meteran
14	Penutup terminal	Tutup pelindung dengan diagram kabel tercetak di bagian dalam.

Voltage/current inputs	
Tegangan nominal	3x230/400 VAC
Rentang tegangan	3x220–240 VAC (-20% – +15%)
Sirkuit tegangan disipasi daya	1.6 VA (0.7 W) total
Sirkuit arus disipasi daya	0.007 VA (0.007 W) per phase at 230 VAC and Ib
Basis saat Ib	5 A
Referensi saat Iref	5 A
Itr Transisi	0.5 A
Imax maksimum	65 A
Minimum Imin	0.25 A
Mulai Ist	< 20 mA
Area kawat terminal	1–25 mm ²
Torsi pengetatan yang disarankan	3 Nm
Data umum	
Frekuensi	50 or 60 Hz ± 5%
Kelas Akurasi	B (Cl. 1) and Reactive Cl. 2
Energi aktif	1%
Tampilan energi	7-digit LCD
Mekanis	
Bahan	Polycarbonate dalam kaca depan transparan. Kaca polikarbonat yang diperkuat di kasing bawah dan kasing atas. Polikarbonat dalam penutup terminal.
Bobot	
Lingkungan	
Suhu Operasional	-40°C to +70°C
Suhu penyimpanan	-40°C to +85°C
Kelembaban	75% yearly average, 95% on 30 days/year
Tahan api dan panas	Terminal 960°C, cover 650°C (IEC 60695-2-1)

Ketahanan terhadap air dan debu	IP 20 pada blok terminal tanpa penutup pelindung dan IP 51 pada penutup pelindung, menurut IEC 60529.
Lingkungan mekanik	Kelas M1 sesuai dengan Petunjuk Instrumen Pengukuran (MID), (2004/22 / EC).
Electromagnetic environment	Kelas E2 sesuai dengan Petunjuk Instrumen Pengukuran (MID), (2004/22 / EC).
Outputs	
Arus	2–100 mA
Tegangan	24 VAC–240 VAC, 24 VDC–240 VDC. For meters with only 1 output 5–40 VDC.
Frekuensi keluaran pulsa	Prog. 1–999999 imp/MWh, 1–999999 imp/kWh, 1–999999 imp/Wh
Panjang pulsa	10–990 ms
Area kawat terminal	0.5–1 mm ²
Torsi pengetatan yang disarankan	0.25 Nm
Inputs	
Tegangan	0–240 V AC/DC
Off	0–12 V AC/DC
ON	57–240 V AC/24–240 V DC
Min. panjang pulsa	30 ms
Area kawat terminal	0.5–1 mm ²
Torsi pengetatan yang disarankan	0.25 Nm
Communication	
Area kawat terminal	0.5–1 mm ²
Torsi pengetatan yang disarankan	0.25 Nm
M-Bus	EN 13757-2, EN 13757-3
Modbus	Modbus Application Protocol Specification V1.1b
EQ-Bus	IEC 62056-42, 62056-46, 62056-53, 62056-61, 62056-62

Pada data tabel tersebut merupakan spesifikasi dari Kwh Modbus ABB 23 data tersebut diambil dari datasheet kwh Modbus ABB 23/24 .

2.2 Energi Listrik

Sumber listrik PLN merupakan sumber energi listrik dengan arus bolak – balik atau Alternating Current (AC) yang dihasilkan dari generator AC, baik dari pembangkit Tenaga Diesel (PLTD) maupun dari Tenaga Air (PLTA) maupun dari pembangkit – pembangkit lainnya yang menghasilkan sumber listrik dengan arus bolak balik. Daya listrik didefinisikan sebagai laju hantaran energi listrik dalam rangkaian listrik. Daya listrik dibagi menjadi tiga, yaitu : daya aktif, daya reaktif, dan daya semu.[5]

1. Daya Aktif

Daya dengan satuan joule/detik atau watt disebut sebagai daya aktif. Daya aktif adalah daya sebenarnya yang dibutuhkan oleh beban. Daya aktif dapat dihitung dengan persamaan

$$P = V.I. \cos\phi \quad \dots\dots\dots(2.1)$$

P = Daya

V = Tegangan

I = Arus

Cos ϕ = Faktor Daya

2. Daya Reaktif

Daya reaktif adalah daya yang dibutuhkan untuk pembentukan medan magnet atau daya yang ditimbulkan oleh beban yang bersifat induktif. Satuan daya reaktif adalah VAR (Volt.Amper Reaktif). Rumus daya reaktif ini bisa didapatkan dengan menggunakan rumus

$$Q = V.I. \sin \phi \quad \dots\dots\dots(2.2)$$

Q = Daya Semu (VAR)

V = Tegangan (V)

I = arus listrik (A)

Sin ϕ = Faktor Reaktif

3. Daya Semu

Daya semu adalah daya yang dihasilkan dari perkalian tegangan dan arus listrik. Daya semu merupakan daya yang diberikan oleh PLN kepada konsumen. Satuan daya semu adalah VA (Volt-Ampere). Rumus daya semu ini bisa didapatkan dengan menggunakan rumus

$$S = V \times I \quad \dots\dots\dots(2.3)$$

S = Daya Semu (VA)

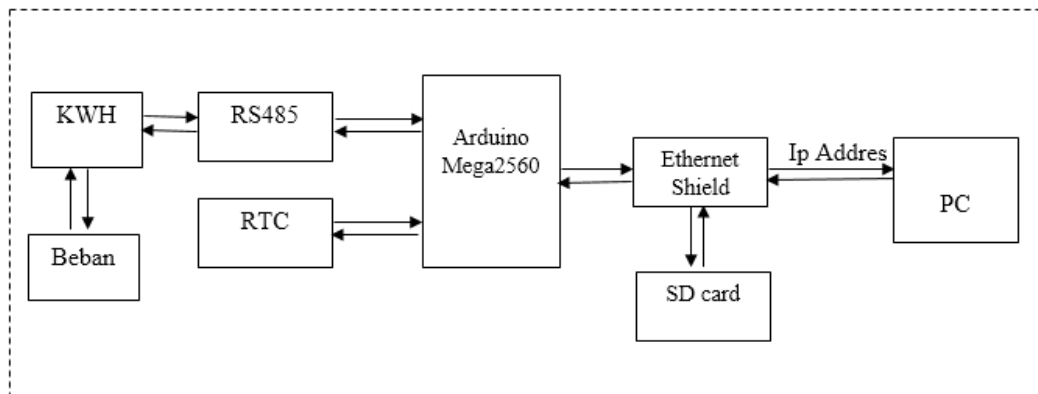
V = Tegangan (V)

I = Arus listrik (A)

2.3 Komunikasi Energi Meter Modbus

Modbus adalah protokol komunikasi serial yang diterbitkan oleh Modicon pada 1979 untuk diaplikasikan pada programmable logic controller (PLC). Kemudian protokol ini telah menjadi standar de facto protokol komunikasi di industri, dan sekarang modbus merupakan protokol komunikasi dua-arah yang paling umum digunakan sebagai media penghubung dengan perangkat industri atau media elektronik lainnya dengan komputer. Komunikasi dengan menggunakan protokol modbus bisa melalui perantara port serial (RS-232, RS485, FO), bisa juga melalui ethernet (LAN) dan jaringan lainnya yang mendukung protokol internet.[3]

Pada proyek akhir ini penulis menggunakan *energi meter* ABB dengan tipe B23 312-100 dengan protokol modbus yang melalui port serial RS485 dan juga ethernet (LAN) . KWh meter memberikan informasi secara real time konsumsi daya aktif yang digunakan pada suatu sistem kelistrikan. Secara umum jenis kwh meter dibagi menjadi dua jenis, yang pertama adalah kwh meter analog dan yang kedua adalah kwh meter digital. Power meter digital yang digunakan pada proyek akhir ini adalah komunikasi serial RS485 dengan protokol Modbus RTU. Data Modbus dikirim melalui port serial RS485 untuk berkomunikasi. Komunikasi modbus dapat dilakukan dengan menggunakan sarana kabel ethernet atau jaringan wifi. Data Modbus dapat dikirim di dalam jaringan lokal maupun internet. Berikut ini adalah diagram hardware dari alat monitoring energi meter berbasis modbus RTU.[4]



Gambar 2.2 Diagram Hardware

Pada diagram tersebut, kwh meter akan membaca beban yang digunakan kemudian akan mengirimkan data ke arduino melalui RS485 dan data hasil pembacaan kwh akan diakses ke PC dengan menggunakan ethernet shield melalui *IP Address*. Sedangkan RTC digunakan untuk pemrograman waktu dan sd card digunakan untuk menyimpan data hasil monitoring penggunaan energi meter.

2.4 Ethernet Shield

Ethernet Shield adalah modul yang digunakan untuk mengkoneksikan Arduino mega 2560 dengan internet menggunakan kabel (Wired). Arduino Ethernet Shield dibuat berdasarkan pada Wiznet W5100 ethernet chip. Ethernet Shield bekerja dengan cara memberikan layanan ip address pada arduino dan pc agar dapat terhubung Cara menggunakan cukup mudah yaitu hanya dengan menghubungkan Arduino Ethernet Shield dengan board Arduino lalu akan disambungkan ke jaringan internet. Cukup memasukkan module ini ke board Arduino Selain itu module ini juga terdapat sebuah onboard micro-SD slot, yang dapat digunakan untuk menyimpan file dan data. Module Ethernet Shield bisa digunakan dengan board Arduino Uno dan Mega. Dan dapat bekerja dengan baik pada kedua Arduino tersebut.[5]

Energi meter yang dibaca oleh arduino tidak bisa langsung disimpan pada database server, diperlukan modul ethernet shield untuk arduino agar arduino bisa terhubung ke jaringan komputer dengan media kabel UTP dan konektor RS485 Ethernet shield merupakan modul yang dipasang diatas arduino yang berfungsi

menghubungkan arduino dengan jaringan komputer melalui media kabel UTP sehingga dibaca oleh arduino. dapat dilihat gambar 2.3 adalah ethernet shield.[5]

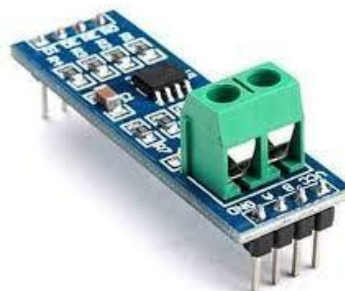


Gambar 2.3 Ethernet Shield (*Putra, p. 2006*)

Gambar 2.3 merupakan gambar dari ethernet shield yang digunakan dan dipasang di atas arduinomega2560 untuk menghubungkan komputer/PC ke arduino dengan menggunakan kabel LAN.

2.5 RS485

RS-485 adalah protokol komunikasi serial asinkron yang tidak memerlukan pulsa clock. Komunikasi ini menggunakan teknik yang disebut sinyal diferensial untuk mentransfer data biner dari satu perangkat ke perangkat lainnya. Untuk menggunakan RS-485 di Arduino, modul yang disebut 5V MAX485 TTL ke RS485 yang didasarkan pada Maxim MAX485 IC. Berikut ini adalah gambar dari RS485.[3]



Gambar 2.4 RS485 (*Imam, 2016*)

Modul RS485 digunakan untuk komunikasi antara Kwh dengan arduino pada modul RS485 dihubungkan dengan arduinomega2560 dan juga kwh.

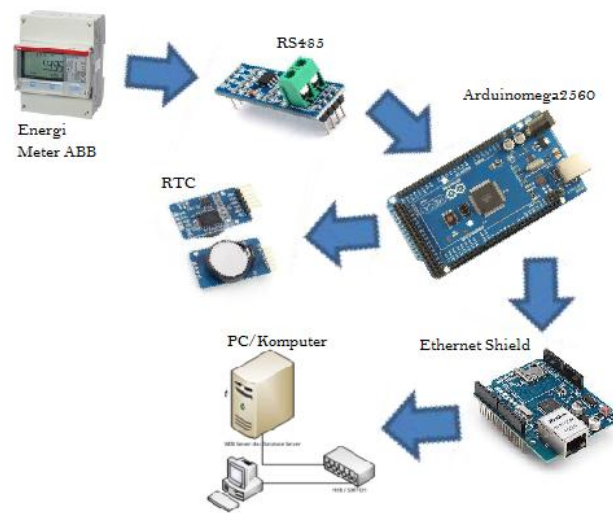
BAB III

METODE PELAKSANAAN

Perencanaan pada proyek akhir ini berawal saat penulis melakukan praktek kerja lapangan di PT. Pratama Motivasi Mandiri sebuah perusahaan yang bergerak pada bidang Sistem Otomasi Gedung Keselamatan & Keamanan juga menjual Desain Teknik, Instalasi, Pelatihan dan Pengembangan. Sehingga semua kebutuhan Proyek Akhir akan di fasilitasi oleh perusahaan. Pada perencanaan proyek akhir penulis di arahkan oleh pembimbing praktik kerja lapangan serta rekan-rekannya di PT. Pratama Motivasi Mandiri.

3.1 Rancangan Sistem Hardware

Sistem alat monitoring energi meter berbasis modbus RTU ini dirancang untuk memonitoring energi listrik yang digunakan dengan menampilkan hasil pembacaan pada web melaalui ip address dengan menentukan komponen yang akan digunakan seperti arduino mega, ethernet shield, Rs485, RTC, KWH meter, dan juga MCB. Berikut ini adalah gambaran sistem monitoring energi meter berbasis modbus RTU. Berikut ini merupakan gambar dari sistem alat monitoring.



Gambar 3.1 Sistem Alat Monitoring

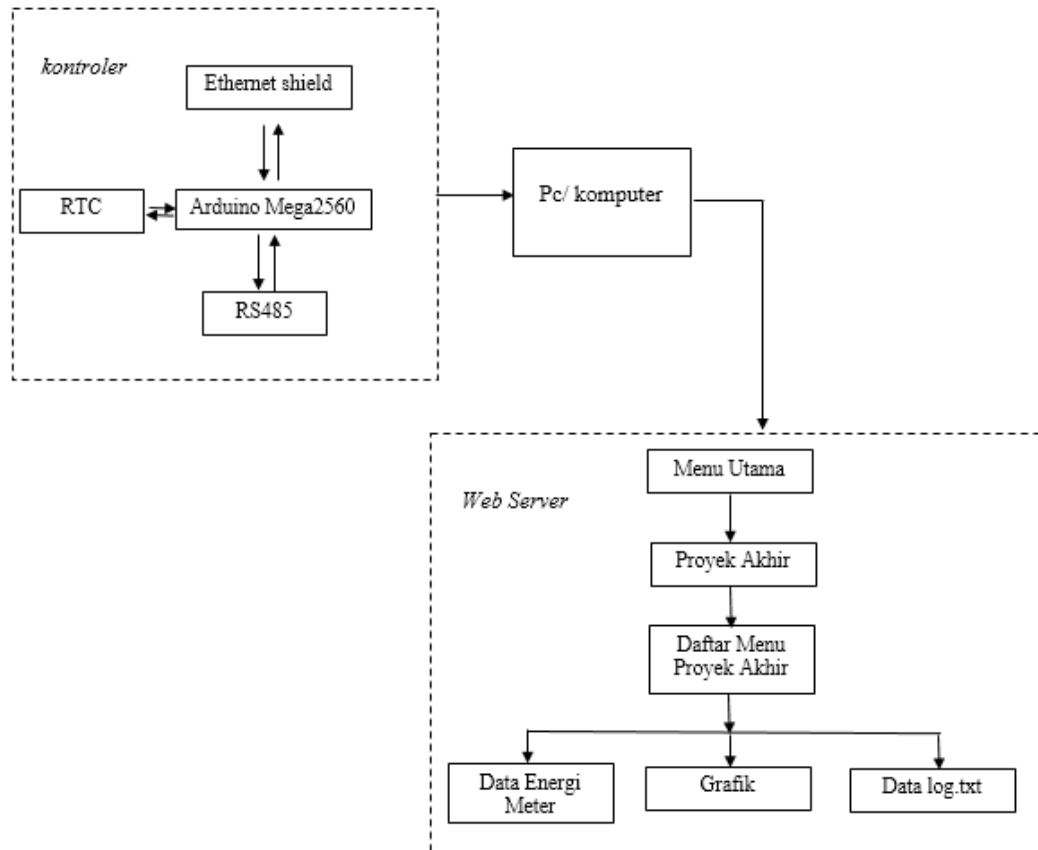
Tabel 3.1 Fungsi Dari Setiap Komponen

No	Komponen	Keterangan
1.	Energi Meter ABB	Energi meter berfungsi untuk membaca penggunaan energi listrik pada beban.
2.	Modul RS485	RS485 berfungsi sebagai konektor antara energi meter dengan arduino mega untuk mengirimkan data penggunaan energi listrik yang digunakan oleh beban.
3.	Arduino mega 2560	Arduino digunakan sebagai media pengirim data dengan menggunakan sistem <i>coding/programing</i> .
4.	RTC	Pengunaan RTC untuk menampilkan waktu pada saat memonitoring penggunaan enenrgi listrik.
5.	Ethernet Shield	Sebagai media yang digunakan untuk mengkoneksikan antara Arduino dengan internet menggunakan kabel LAN.
6.	SD Card	Sebagai media penyimpanan data.
7.	PC/ Komputer	Sebagai media penampil saat memonitoring penggunaan daya listrik yang dihubungkan oleh ethernet shield dengan menggunakan kabel Lan dan diakses dengan menggunakan ip address.

3.2 Rancangan Tampilan Sistem Monitoring

Desain tampilan adalah bagian dari sistem yang menghubungkan pengguna dengan komputer sehingga dapat membentuk sebuah dialog. Dengan perancangan tampilan yang dilakukan maka pengguna akan mendapatkan kemudahan memanfaatkan fasilitas-fasilitas yang disediakan oleh sistem. Rancangan tampilan

ini menggunakan media web browser sebagai terminal bagi pengguna. Pada halaman web dari sistem terdiri dari bagian-bagian sebagai berikut :



Gambar 3.2 Rancangan Tampilan Sistem Web

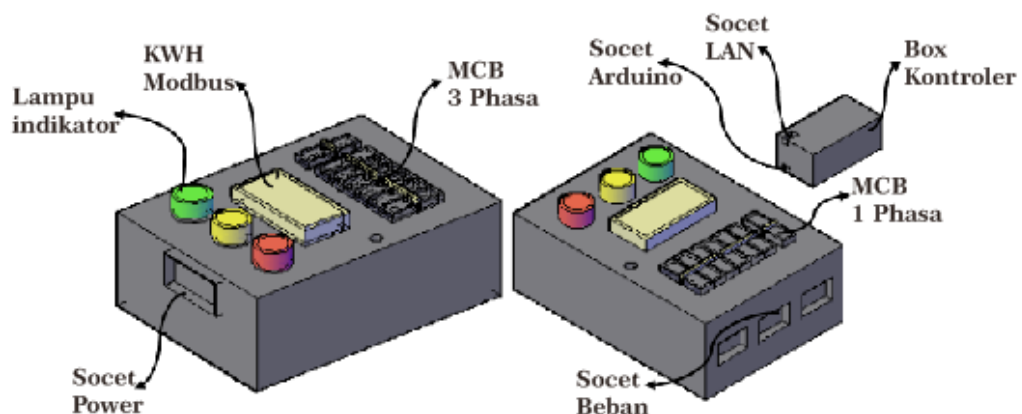
Pada Rancangan Tampilan Sistem Monitoring yang menggunakan sistem web saat memonitoring energi listrik, ketika pengguna ingin mengakses dengan *ip address* maka fasilitas tampilan yang diberikan akan seperti Gambar 3.2 pada bagian tersebut terdiri dari :

1. Saat pengguna mengakses dengan *IP Address* maka pengguna akan diarahkan ke menu utama. Pada menu utama terdapat judul proyek akhir, lambang universitas/politeknik, lambang industri tempat penulis praktek kerja lapangan, nama penulis beserta NIM, serta menu Proyek Akhir. Pada menu ini terdapat lambang dari industri tempat praktek kerja lapangan penulis karena proyek akhir ini difasilitasi dan direkomendasikan oleh industri tempat penulis saat melakukan praktik kerja lapangan.

2. Proyek Akhir merupakan halaman untuk menampilkan daftar menu proyek akhir. Pada halaman ini pengguna akan diarahkan memilih menu pilihan untuk memonitoring penggunaan energi listrik menu yang terdapat pada halaman ini ialah data energi meter, grafik, datalog.txt. Pada menu ini pengguna akan memilih menu pilihan setelah dipilih maka pengguna dapat melihat hasil dari menu yang dipilih.
3. Pada menu data energi meter, pengguna dapat melihat data yang dimonitoring oleh kwh, mulai dari voltage,current,frekuensi,daya aktif,daya aktif total,daya reaktif serta daya semu.
4. Pada menu grafik pengguna dapat melihat tampilan grafik. Tampilan grafik merupakan data daya aktif total yang dimonitoring oleh kwh.
5. Datalog.txt merupakan data yang disimpan pada SD Card yang merupakan hasil dari data monitoring kwh.

3.3 Rancangan Kontruksi Mekanik

Rancangan kontruksi mekanik pada sistem monitoring energi meter berbasis modbus RTU ini merupakan langkah awal dalam pembuatan kontruksi mekanik dimana rancangannya dibuat sesederhana mungkin dan menggunakan komponen yang mudah didapatkan. Secara umum dapat dilihat pada gambar 3.3 berikut.



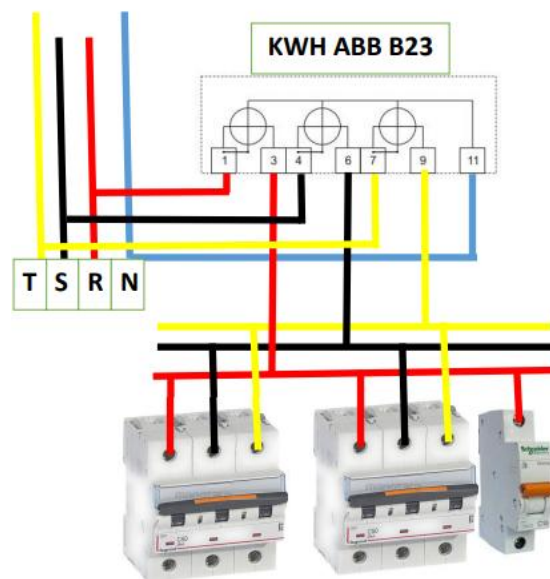
Gambar 3.3 Rancangan Kontruksi Mekanik

Perancangan kontruksi mekanik menggunakan autocad pada alat ini menggunakan box panel dengan ukuran 35 cm x 25 cm x 12 cm, *pilot lamp*

digunakan sebagai lampu indikator. Penggunaan komponen pada alat ini menggunakan sumber standar industri seperti soket power 3 fasa, box kontroler merupakan tempat penyimpanan sistem kontrol komunikasi pada box ini terdapat modul arduino mega 2560, rs485, ethernet shield, rtc serta pcb yang digunakan sebagai pengkabelan dan dipasang dengan cara ditumpukan untuk mempermudah dan mengefesiensi box kontroler.

3.4 Skematik sistem Elektrik Hardware

Skematik sistem *hardware* pada proyek ini merupakan langkah dalam menentukan input dan output pada alat dengan menggunakan komponen yang telah ditentukan. Skematik sistem elektrik pada alat ini dapat dilihat pada gambar 3.4 berikut.



Gambar 3.4 Skematik sistem Elektrik Hardware

Pada skematik ini sumber yang digunakan pada KWH ABB 23 ialah 3 fasa dengan garis merah (R), hitam (S), dan kuning (T) sumber tersebut akan masuk pada input kwh abb 23 lalu output dari kwh abb akan masuk ke input mcb 3 fasa, input mcb 1 fasa serta 1 output pada kwh abb yang di koneksi dengan arduino

melalui Rs485. Output dari mcb 3 phasa dan 1 phasa akan di sambungkan pada beban yang digunakan.

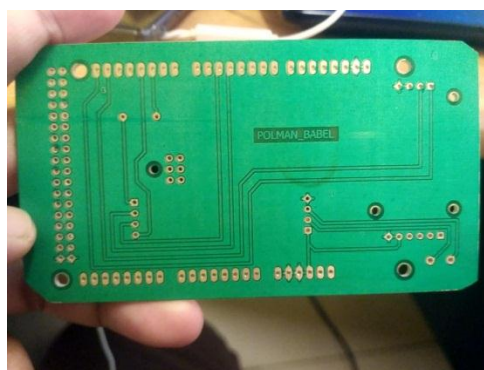
3.5 Sistem Kontroler

Sistem kontroler merupakan sistem komunikasi yang digunakan sebagai penghubung antara alat dengan alat lainnya. Pada sistem ini menggunakan arduino mega 2560 yang dipasang bertumpuk dengan RS485, RTC, dan juga ethernet shield. Berikut ini gambar dari sistem kontroler yang digunakan penulis dalam membuat proyek akhir. Berikut gambar untuk kontroler yang digunakan.



Gambar 3.5 sistem kontroler

Pada sistem kontrol ini penulis menggunakan pcb sebagai media pengkabelan untuk memudahkan dalam pengolahan, selain itu pcb juga lebih memperindah tampilan sistem kontroler. Berikut ini gambar pcb yang digunakan dalam media pengkabelan.

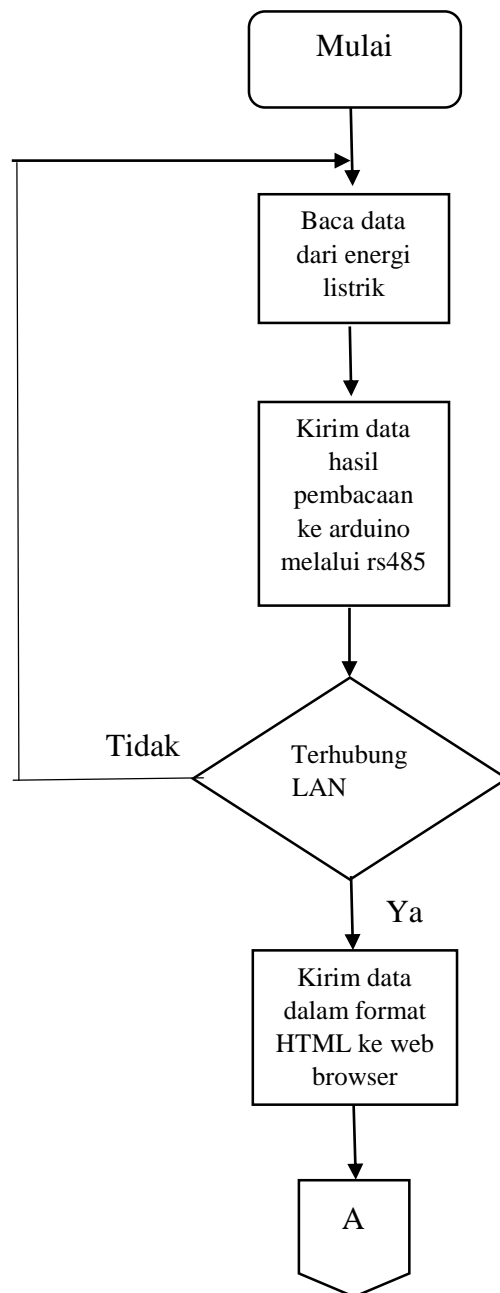


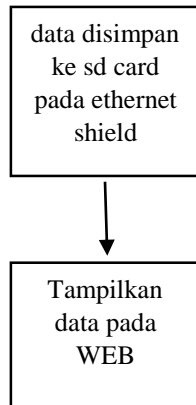
Gambar 3.6 Pcb Pengkabelan

Pada pcb ini terdapat modul RS485 dengan RTC yang dipasang dengan cara ditumpuk dengan tujuan agar penggunaan pada pcb lebih rapi.

3.6 Algoritma Sistem

Flowchart sistem berfungsi untuk mempermudah memahami cara kerja program pada sistem yang dibuat. Flowchart sistem dapat dilihat pada gambar 3.4 dibawah ini





Gambar 3.7 Flowchart sistem

BAB IV

PEMBAHASAN

Pada bab ini merupakan uraian dari proses pembuatan proyek akhir yang berjudul “Monitoring Energi Meter Berbasis Modbus RTU”.

4.1 Pembuatan Hardware Mekanik Alat Monitoring

Adapun tahap-tahap pembuatan konstruksi Alat Monitoring Energi Meter Berbasis Modbus RTU yaitu :

1. Membuat sketsa/pola untuk semua komponen yang akan digunakan pada panel
Berikut gambar 4.1 Panel.



Gambar 4.1 Panel

Panel yang digunakan pada alat ini memiliki ukuran 35 cm x 25 cm x 12 cm penulis akan membuat pola pada panel untuk memasang kwh, mcb 3 phasa, mcb 1 phasa, lampu indikator, soket beban serta soket power dengan ukuran yang sesuai dengan ukuran komponen

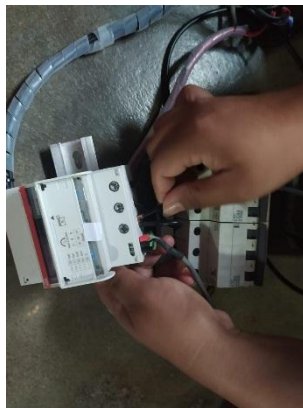
2. Pemotongan atau pengeboran sketsa pola. Berikut gambar dari 4.2 Pemotongan Atau Pengeboran Panel.



Gambar 4.2 Pemotongan Atau Pengeboran Panel

Pola yang telah dibuat pada panel akan di lobangi atau di potong agar sesuai dengan ukuran komponen yang digunakan.

3. Pemasangan instalasi elektrik hardware. Berikut gambar 4.3 Pemasangan instalasi elektrik pada hardware



Gambar 4.3 Pemasangan instalasi elektrik hardware

Pemasangan instalasi elektrik pada setiap komponen menggunakan kabel dengan ukuran 4 mm pemasangan instalasi elektrik sesuai dengan rancangan elektrik pada hardware.

4. Pemasangan Komponen. Berikut ini gambar 4.4 Pemasangan Komponen pada panel.



Gambar 4.4 Pemasangan Komponen pada panel

Pemasangan komponen dilakukan setelah pola yang dipotong dan dibor selesai dan juga instalasi elektrik pada komponen kemudian di rekatkan dengan menggunakan besi perekat atau rel.

5. *Assembling* keseluruhan kerangka alat. Berikut gambar 4.5 *assembling* alat.



Gambar 4.5 *Assembling* Alat

Proses *Assembling* adalah proses penggabungan seluruh komponen dan merupakan akhir dari pengerjaan hardware proyek akhir. Pada proses ini akan dilakukan pengecekan mulai dari instalasi elektrik komponen dengan menggunakan multimeter untuk mengecek sambungan dari setiap input dan output komponen, setelah melakukan pengecekan sambungan alat ini diberikan stiker identitas nama komponen serta nama dari penulis. Setelah proses pemasangan stiker selesai alat ini kemudian dicoba dengan cara pemberian sumber agar tidak ada yang salah dalam pemasangan instalasi elektrik hardware.

4.2 Perancangan Sistem Pada Web

Rancangan Tampilan Pada Web alat monitoring energi meter berbasis modbus RTU dapat dilihat pada gambar berikut :

4.2.1 Menu Utama

Menu utama merupakan menu tampilan awal pada web. Untuk tampilan dari menu utama dapat dilihat pada gambar 4.6.



Gambar 3.6 Rancangan Tampilan Pada Web

Tampilan menu utama pada web pada menu ini akan menampilkan judul proyek akhir, lambang perguruan tinggi, nama penulis serta NIM.

4.2.2 Menu Proyek Akhir

Adapun gambar tampilan dari menu proyek akhir ditunjukkan oleh gambar

4.7



Gambar 4.7 Daftar Menu Proyek Akhir

Pada gambar 4.7 merupakan tampilan untuk daftar menu proyek akhir. Pada halaman ini pengguna dapat memilih menu yang terdapat pada bagian atas halaman dengan cara di klik. Jika pengguna telah memilih menu pada halaman daftar menu proyek akhir maka pengguna akan diarahkan ke halaman yang dipilih.

4.2.3 Daftar Menu Proyek Akhir

Berikut ini tampilan untuk menu data energi meter dapat dilihat pada gambar 4.8

Energy Meter Real Value

Descripsi 1
Descripsi 2

VOLTAGE(V)	CURRENT(Amp)	FREQUENCY(Hz)	DAYA AKTIF(Watt)	DAYA AKTIF TOTAL(Watt)	DAYA REAKTIF(VAR)	DAYA SEMU(VA)
Voltage 1 : 232.50	Current 1 : 2.34	Frekuensi : 50.00	Daya Aktif 1 : 87.64	Daya Aktif Total : 265.80	Daya Reaktif 1 : 537.40	Daya Semu 1 : 545.70
Voltage 2 : 229.50	Current 2 : 2.39		Daya Aktif 2 : 81.59		Daya Reaktif 2 : 548.70	Daya Semu 2 : 552.70
Voltage 3 : 233.30	Current 3 : 2.36		Daya Aktif 3 : 92.30		Daya Reaktif 3 : 550.20	Daya Semu 3 : 558.40

[Go back to Page2](#)

Gambar 4.8 Daftar Menu Proyek Akhir

Pada menu ini akan menampilkan pembacaan tegangan, arus, daya aktif, daya aktif total, daya reaktif, daya semu juga frekuensi. Tampilan menu data energi meter menampilkan hasil pembacaan penggunaan energi listrik

4.2.4 Menu Datalog txt

Pilihan menu data datalog.txt dapat dilihat pada gambar 4.9.

Voltage 1 : 0.00 Current 1 : 0.00 Frekuensi : 0.00 Daya 1 : 0.00 Daya total : 0.00 Voltage 2 : 0.00 Current 2 : 0.00 Daya 2 : 0.00 Voltage 3 : 0.00 Current 3 : 0.00 Daya 3 : 0.00 Voltage 1 : 228.30 Current 1 : 2.35 Frekuensi : 50.02 Daya 1 : 137.31 Daya total : 137.31 Voltage 2 : 228.30 Current 2 : 0.00 Daya 2 : 0.00 Voltage 3 : 227.90 Current 3 : 0.00 Daya 3 : 0.00 Voltage 1 : 229.10 Current 1 : 6.54 Frekuensi : 50.01 Daya 1 : 307.35 Daya total : 630.31 Voltage 2 : 228.30 Current 2 : 4.88 Daya 2 : 147.83 Voltage 3 : 228.00 Current 3 : 4.78 Daya 3 : 165.43 Voltage 1 : 0.00 Current 1 : 0.00 Frekuensi : 0.00 Daya 1 : 0.00 Daya total : 0.00 Voltage 2 : 0.00 Current 2 : 0.00 Daya 2 : 0.00 Voltage 3 : 0.00 Current 3 : 0.00 Daya 3 : 0.00 Voltage 1 : 229.31 Current 1 : 2.36 Frekuensi : 50.00 Daya 1 : 80.23 Daya total : 239.86 Voltage 2 : 228.70 Current 2 : 2.32 Daya 2 : 79.98 Voltage 3 : 228.20 Current 3 : 2.28 Daya 3 : 82.47 Voltage 1 : 0.00 Current 1 : 0.00 Frekuensi : 0.00 Daya aktif 1 : 0.00 Daya aktif total : 0.00 Daya Reaktif 1 : 0.00 Daya Semu 1 : 0.00 Voltage 2 : 0.00 Current 2 : 0.00 Daya Aktif 2 : 0.00 Daya Reaktif 2 : 0.00 Daya Semu 2 : 0.00 Voltage 3 : 0.00 Current 3 : 0.00 Daya Aktif 3 : 0.00 Daya Reaktif 3 : 0.00 Daya Semu 3 : 0.00 Voltage 1 : 232.50 Current 1 : 2.34 Frekuensi : 50.00 Daya aktif 1 : 87.64 Daya aktif total : 265.80 Daya Reaktif 1 : 537.40 Daya Semu 1 : 545.70 Voltage 2 : 229.50 Current 2 : 2.39 Daya Aktif 2 : 81.59 Daya Reaktif 2 : 548.70 Daya Semu 2 : 552.70 Voltage 3 : 233.30 Current 3 : 2.36 Daya Aktif 3 : 92.30 Daya Reaktif 3 : 550.20 Daya Semu 3 : 558.40

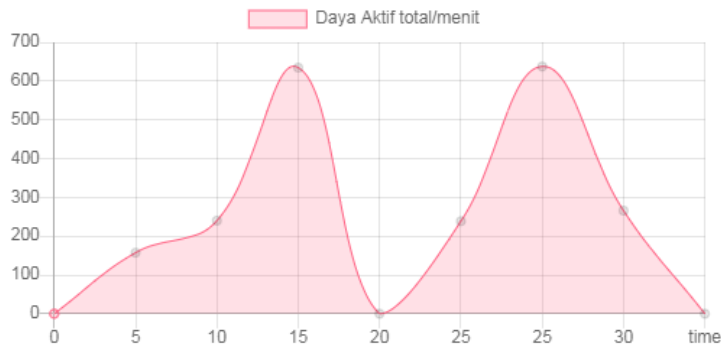
Gambar 4.9 Gambar Menu Datalogtxt

Pada menu ini akan menampilkan hasil pembacaan yang disimpan pada sd card.

4.2.5 Menu Grafik

Rancangan Tampilan grafik alat monitoring energi meter berbasis modbus RTU dapat dilihat pada gambar 4.10 berikut:

Grafik



Gambar 4.10 Grafik

Tampilan grafik pada web merupakan data yang akan diambil dari data hasil penggunaan daya dengan sumbu y merupakan daya aktif total dan sumbu x merupakan waktu. Pada pengambilan data grafik sumbu x menunjukkan waktu yang digunakan mulai dari 5 menit sampai dengan 15 menit dengan gelombang grafik naik dan pada menit ke 20 grafik kembali turun hal tersebut disebabkan oleh *autorefresh* saat pengambilan data. Dan grafik kembali naik ke menit 25. Pada menit ke 30 grafik akan kembali turun.

4.3 Perancangan dan Pembuatan Software

Dalam pengerjaan *Software* alat proyek akhir Monitoring Energi Meter Berbasis Modbus RTU untuk perancangannya menggunakan pemrograman dengan aplikasi sublime. Aplikasi ini digunakan untuk membuat pengcodangan tampilan pada web (HTML).

4.4 Pengujian Alat Monitoring Energi Meter Berbasis Modbus RTU

Setelah pembuatan *software* selesai maka selanjutnya melakukan uji coba menggunakan alat Monitoring Energi Meter Berbasis Modbus RTU dengan beban 2 buah motor 3 phasa dan 1 buah motor 1 phasa beban yang digunakan pada alat ini dapat dilihat pada gambar-gambar berikut :



Gambar 4.11 Beban 1 Motor 3 Phasa



Gambar 4.12 Beban 2 Motor 3 Phasa



Gambar 4.13 Beban 3 Motor 1 Phasa

Pada pengujian alat monitoring penulis menggunakan beban motor yang terdapat pada laboratorium sistem tenaga listrik di kampus Polman Babel dengan percobaan yang dilakukan penulis mendapatkan hasil monitoring dari alat ini yang di tampilkan pada web. Hasil tersebut ditunjukkan oleh gambar 4.15

Energy Meter Real Value

Desripsi 1

Desripsi 2

VOLTAGE(V)	CURRENT(Amp)	FREQUENCY(Hz)	DAYA AKTIF(Watt)	DAYA AKTIF TOTAL(Watt)	DAYA REAKTIF(VAR)	DAYA SEMU(VA)
Voltage 1 : 232.50	Current 1 : 2.34	Frekuensi : 50.00	Daya Aktif 1 : 87.64	Daya Aktif Total : 265.80	Daya Reaktif 1 : 537.40	Daya Semu 1 : 545.70
Voltage 2 : 229.50	Current 2 : 2.39		Daya Aktif 2 : 81.59		Daya Reaktif 2 : 548.70	Daya Semu 2 : 552.70
Voltage 3 : 233.30	Current 3 : 2.36		Daya Aktif 3 : 92.30		Daya Reaktif 3 : 550.20	Daya Semu 3 : 558.40

Go back to [Page 2](#)

Gambar 4.14 Hasil Monitoring Pada Web

Pada gambar 4.14 dapat dilihat merupakan hasil dari uji coba yang dilakukan di laboratorium STL Polman Babel bentuk dari monitoring pada web ditampilkan dengan tabel mulai dari tabel Voltage, Current, Frekuensi, Daya Aktif, Daya Aktif Total, Daya Reaktif, serta Daya Semu

Untuk nilai perbandingan pada hasil uji coba yang dilakukan dapat dilihat pada tabel 4.1 berikut ini.

Tabel 4.1 Perbandingan Pada Hasil Uji Coba

BEBAN	Hasil Monitoring									
	WEB					KWH				
	Volt (V)	Amp (A)	Daya Aktif (W)	Daya Aktif Total (W)	Freq (Hz)	Volt (V)	Amp (A)	Daya Aktif (W)	Daya Aktif Total (W)	Freq (Hz)
R	232.50	2.34	87.64	265.80	50.00	232.4	2.36	88.9	268.1	50.02
S	229.50	2.39	81.59			229.6	2.41	82.3		
T	223.30	2.36	92.30			233.4	2.39	94.8		

Tabel 4.2 Perbandingan Pada Hasil Uji Coba Daya Semu dan Daya Reaktif

BEBAN	Hasil Monitoring			
	WEB		KWH	
	Daya Reaktif (VAR)	Daya Semu (VA)	Daya Reaktif (VAR)	Daya Semu (VA)
R	537.40	545.70	537.4	546.7
S	548.70	552.70	549.5	553.0
T	550.20	558.40	551.8	559.1

Dari hasil uji coba data yang didapatkan antara web dengan kwh memiliki nilai yang hampir sama, sehingga pada uji coba antara web dengan nilai yang terdapat pada kwh dinyatakan sama. Untuk hasil nilai tegangan, arus, daya aktif, daya aktif total, daya reaktif, daya semu dan juga frekuensi pada Kwh dapat dilihat pada gambar-gambar berikut ini :



Gambar 4.15 Tegangan Beban R

Pada gambar 4.15 merupakan hasil dari monitoring Voltage pada beban R dengan hasil yang ditunjukkan 232.4V.



Gambar 4.16 Tegangan Beban S

Pada gambar 4.16 merupakan hasil dari monitoring Voltage pada beban S dengan hasil yang ditunjukkan 229.6V.



Gambar 4.17 Tegangan Beban T

Pada gambar 4.17 merupakan hasil dari monitoring Voltage pada beban T dengan hasil yang ditunjukkan 233.4V



Gambar 4.18 Arus Beban R

Pada gambar 4.18 merupakan hasil dari monitoring Current pada beban R dengan hasil yang ditunjukkan 2.36A



Gambar 4.19 Arus Beban S

Pada gambar 4.19 merupakan hasil dari monitoring Current pada beban S dengan hasil yang ditunjukkan 2.41A



Gambar 4.20 Arus Beban T

Pada gambar 4.20 merupakan hasil dari monitoring Current pada beban T dengan hasil yang ditunjukkan 2.39A



Gambar 4.21 Daya Aktif R

Pada gambar 4.21 merupakan hasil dari monitoring Daya Aktif pada beban R dengan hasil yang ditunjukkan 88.9W.



Gambar 4.22 Daya Aktif S

Pada gambar 4.22 merupakan hasil dari monitoring Daya Aktif pada beban S dengan hasil yang ditunjukkan 82.3W



Gambar 4.23 Daya Aktif T

Pada gambar 4.23 merupakan hasil dari monitoring Daya Aktif pada beban T dengan hasil yang ditunjukkan 94.8W



Gambar 4.24 Daya Aktif Total

Pada gambar 4.24 merupakan hasil dari monitoring Daya Aktif Total pada beban R , S, dan T dengan hasil yang ditunjukkan 268.1W



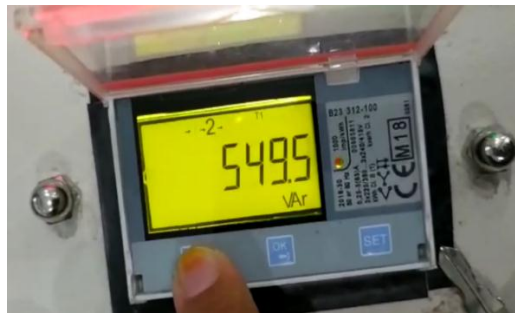
Gambar 4.25 Frekuensi

Pada gambar 4.25 merupakan hasil dari monitoring Daya pada beban R dengan hasil yang ditunjukkan 50.02 Hz.



Gambar 4.26 Daya Reaktif R

Pada gambar 4.26 merupakan hasil dari monitoring Daya Reaktif pada beban R dengan hasil yang ditunjukkan 537.4 VAR.



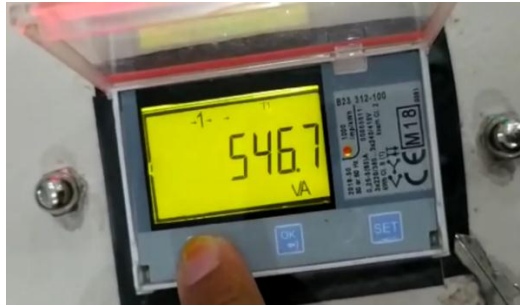
Gambar 4.27 Daya Reaktif S

Pada gambar 4.27 merupakan hasil dari monitoring Daya Reaktif pada beban S dengan hasil yang ditunjukkan 549.5 VAR.



Gambar 4.28 Daya Reaktif T

Pada gambar 4.28 merupakan hasil dari monitoring Daya Aktif pada beban T dengan hasil yang ditunjukkan 551.8 VAR.



Gambar 4.29 Daya Semu R

Pada gambar 4.29 merupakan hasil dari monitoring Daya Semu pada beban R dengan hasil yang ditunjukkan 546.7 VA.



Gambar 4.30 Daya Semu S

Pada gambar 4.30 merupakan hasil dari monitoring Daya Semu pada beban S dengan hasil yang ditunjukkan 553.0 VA.



Gambar 4.31 Daya SemuT

Pada gambar 4.31 merupakan hasil dari monitoring Daya Semu pada beban T dengan hasil yang ditunjukkan 559.1 VA.

4.5 Analisa Hasil Uji Coba

Pada hasil uji coba alat nilai yang ditampilkan pada web dan kwh memiliki rentang nilai masing masing beban. Untuk memperkuat data dari hasil uji coba dapat dilihat pada analisa hasil uji coba untuk daya sebagai berikut :



Gambar 4.32 Power Faktor (Cos φ)

Daya dapat di nyatakan dengan persamaan :

$$P = V.I. \cos\varphi \quad \dots\dots\dots(4.1)$$

P = Daya

V = Tegangan

I = Arus

Cosφ = Faktor Daya

Sehingga untuk perhitungan daya penurunan rumus daya untuk $\cos\varphi$ sebagai berikut :

$$\cos\varphi = \frac{p}{V \times I} \quad \dots\dots\dots(4.2)$$

Untuk membuktikan nilai daya pada setiap beban dapat dilihat pada analisa perhitungan seperti berikut :

4.5.1 Beban Untuk R

$$V = 232.50V$$

$$I = 2.34A$$

$$P = 87,64 W$$

$$P = V * I * \cos\varphi \quad \dots\dots\dots(4.3)$$

$$\cos\varphi = \frac{p}{V \times I}$$

$$\cos\varphi = \frac{87.64 \text{ watt}}{232.50V \times 2.34A}$$

$$\cos\varphi = 0.16$$

$$\varphi = \text{COS}^{-1} (0.16)$$

$$\varphi = 80.79^\circ$$

- $P = V \times I \times \cos\varphi$
 $= 232.50\text{V} \times 2.34\text{A} \times 0.16$
 $= 87.04 \text{ Watt}$
- $Q = V \times I \times \text{Sin } \varphi$
 $= 232.50\text{V} \times 2.34\text{A} \times \sin (80.79^\circ)$
 $= 538.61 \text{ VAR}$
- $S = V \times I$
 $= 232.50\text{V} \times 2.34\text{A}$
 $= 544.05\text{VA}$

4.5.2 Beban Untuk S

$$V = 229.50\text{V}$$

$$I = 2.39\text{A}$$

$$P = 81.59 \text{ W}$$

$$P = V * I * \cos\varphi \dots\dots\dots(4.4)$$

$$\cos\varphi = \frac{p}{V \times I}$$

$$\cos\varphi = \frac{81.59 \text{ watt}}{229.50\text{V} \times 2.39\text{A}}$$

$$\cos\varphi = 0.15$$

$$\varphi = \text{COS}^{-1} (0.15)$$

$$\varphi = 81.37^\circ$$

- $P = V \times I \times \cos \varphi$
 $= 229.50\text{V} \times 2.39\text{A} \times 0.15$
 $= 82.28 \text{ Watt}$
- $Q = V \times I \times \text{Sin } \varphi$
 $= 229.50\text{V} \times 2.39\text{A} \times \sin (81.37^\circ)$
 $= 542.29 \text{ VAR}$

- $S = V \times I$
 $= 229.50V \times 2.39A$
 $= 548.51 \text{ VA}$

4.5.3 Beban Untuk T

$$V = 233.30V$$

$$I = 2.38A$$

$$P = 92.30 \text{ W}$$

$$P = V * I * \cos\varphi \dots\dots\dots(4.5)$$

$$\cos\varphi = \frac{p}{V \times I}$$

$$\cos\varphi = \frac{92.30 \text{ watt}}{233.30V \times 2.38A}$$

$$\cos\varphi = 0.17$$

$$\varphi = \cos^{-1} (0.17)$$

$$\varphi = 80.21^\circ$$

- $P = V \times I \times \cos \varphi$
 $= 233.30V \times 2.38A \times 0.17$
 $= 94.39 \text{ Watt}$

- $Q = V \times I \times \sin \varphi$
 $= 233.30V \times 2.38A \times \sin (80.21^\circ)$
 $= 547.17 \text{ VAR}$

- $S = V \times I$
 $= 233.30V \times 2.38A$
 $= 555.25 \text{ VA}$

Tabel 4.3 Tabel Analisa Perhitungan

Beban	Hasil Analisa Monitoring Daya		
	Perhitungan	Kwh	Web
R	87.04W	88.9 W	87.64 W
S	82.28W	82.3 W	81.59 W
T	94.39 W	94.8 W	92.30 W

Pada data tabel hasil perhitungan yang menunjukkan nilai yang dihasilkan pada tabel 4.3 memiliki selisih $\pm 0.62W$. Hal tersebut dipengaruhi oleh interval waktu yang digunakan pada saat melakukan pengambilan data dengan persentase *error* sebesar 2.09%. Perhitungan persentase *error* sebagai berikut :

Keterangan :

Nilai A = Nilai maksimaal dari tabel 4.3.

Nilai B = Nilai minimal dari tabel 4.3.

Nilai Max = Nilai maksimal dari tabel 4.3.

$$\%Error = \frac{\text{Nilai A} - \text{Nilai B}}{\text{Nilai Max}} \times 100\%$$

$$\%Error = \frac{88,9W - 87,04W}{88,9W} \times 100\%$$

$$\%Error = 2.09\%.$$

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian dan analisa dari proyek akhir yang berjudul “Monitoring Energi Meter Berbasis Modbus RTU” ini, maka dapat ditarik kesimpulan bahwa :

1. Alat dapat memonitoring energi listrik berupa Tegangan, Arus, Daya Aktif, Daya Aktif Total, Daya Reaktif, Daya Semu, Frekuensi serta Grafik.
2. Alat dapat memonitoring energi listrik dan ditampilkan pada media Web.
3. Data hasil monitoring dapat disimpan pada *SD Card*.
4. Penyimpanan data di *SD Card* masih kurang rapi jika di tampilkan pada *card reader*.
5. Monitoring energi listrik pada web bersifat auto refresh sesuai dengan *delay* yang di tentukan pada program.
6. Pada alat ini menggunakan 3 beban dengan 2 beban 3 phasa dan 1 beban untuk 1 phasa.
7. Persentase *error* pada alat monitoring sebesar 2.09%.

5.2 Saran

Untuk penyempurnaan alat monitoring perlu adanya perbaikan yang harus dilakukan, yaitu dengan penambahan beberapa metode yang akan lebih mendukung kesempurnaan dari alat sebelumnya penambahan tersebut ialah sebagai berikut :

1. Monitoring penggunaan energi listrik dapat dilakukan dengan jarak jauh.
2. Hasil dari monitoring penggunaan energi listrik dapat disimpan pada dokumen dengan tampilan yang baik dan dapat di print out.
3. Pada saat pengujian pembacaan data pada beban di kwh dan web terkadang memiliki nilai yang tidak sama, hal tersebut di pengaruhi oleh interval waktu yang digunakan. Sehingga untuk memperkecil selisih pembacaan maka gunakan *delay* yang kecil pada pemrograman.

DAFTAR PUSTAKA

- [1]. Huda Miftahul, *Protokol Komunikasi Modbus RTU* pada Sistem Otomasi Industri, LPP Kampus Yogyakarta.
- [2]. Minamitsumori, Nishinari-ku, *Modbus Protocol Reference Guide*, EM-5650 Rev.10, MSystem Co., LTD.
- [3]. Imam, Azhar Arifin (2016) *Sistem Monitoring Energi Menggunakan Energi Meter SPM91 Berbasis Protocol Modbus dan Serial Komunikasi RS485*. Diploma thesis, Universitas Andalas. (Putra, 2003)
- [4]. Jonathan Teng, Julius Setiadji, Resmana Lim, *Sistem Pembacaan Data Power Meter Dengan Komunikasi Modbus Secara Terpusat*, Teknik Elektronika, Universitas Kristen Petra Surabaya.
- [5]. Eko Kurniawanto Putra, *Sistem Informasi Energi Listrik 3 Fasa Menggunakan Energi Meter SPM93 Berbasis Web*, Institut Teknologi Padang.
- [6]. Raden Ajeng Gusti Ramadhianti , Ir. Cok Gede Indra Partha , I Gusti Agung Pt Raka Agung, *Rancang Bangun Monitoring Energi Listrik Menggunakan Sms Berbasis Mikrokontroler Atmega328*, Program Studi Teknik Elektro, Fakultas Teknik Universitas Udayana.
- [7]. Data Sheet Energi Meter Modbus ABB 23/24.

LAMPIRAN

LAMPIRAN 1

DAFTAR RIWAYAT HIDUP

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama lengkap : Gira Yunikristiyanti
Tempat & tanggal lahir : Sungailiat, 25 Mei 1997
Alamat rumah : Jl. Nelayan 2 Gg. Matti Netta
No. 29A rt 01 Sungailiat, Bangka.
Hp. 0895602941551
Email : giraredmi@gmail.com
Jenis kelamin : Perempuan
Agama : ISLAM



2. Riwayat Pendidikan

SD NEGERI 6 Sungailiat	Lulus	2010
SMP NEGERI 1 Sungailiat	Lulus	2013
SMA NEGERI 1 Sungailiat	Lulus	2016

3. Pendidikan Non Formal

Sungailiat, 18 Agustus 2020

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama lengkap : Latipa Kusuma Dawe
Tempat & tanggal lahir : Sungailiat, 12 November 1999
Alamat rumah : RSS, JL.Garuda blok 2B no 15,
Kelurahan Bukit Betung, Sungailiat,
Bangka.
Hp. 0895621892557
Email : latipakusuma12@gmail.com
Jenis kelamin : Perempuan
Agama : Islam



2. Riwayat Pendidikan

SD NEGERI 15 Sungailiat	Lulus	2011
SMP NEGERI 2 Sungailiat	Lulus	2014
SMA NEGERI 1 Sungailiat	Lulus	2017

3. Pendidikan Non Formal

Sungailiat, 18 Agustus 2020

LAMPIRAN 2
PROGRAM

Program HTML Menu Utama

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>TUGAS AKHIR 2020</title>
5
6      </head>
7      <style>
8          body {
9              background-color: lightblue;
10             background-image: url(blue.png);
11         }
12
13
14     </style>
15     <body>
16     <style>
17     .main {
18     width: 80px;
19     box-sizing: border-box;
20     float: left;
21     }
22
23     .dua {
24     width: 260px;
25     box-sizing: border-box;
26     float: right;
27     }
28     .cf:before,
29     .cf:after {
30     content: " "; /* 1 */
31     display: table; /* 2 */
```

```
32 }
33
34 .cf:after {
35     clear: both;
36 }
37
38
39 .cf {
40     *zoom: 1;
41 }
42 </style>
43     <div class="conten cf">
44         <div class="main">
45             
46         </div>
47
48         <div class="dua">
49             
50         </div>
51
52     </div>
53
54     <h1><center><font color="black" size="6px" face="Arial">MONITORING ENERGY METER BERBASIS MODBUS RTU</font></center> </h1>
55     <h1><center><font color="black" size="6px" face="Arial"> TAHUN 2020</font></center> </h1>
56
57     <style>
58     .jam-digital {
59         width: 150px;
60         height: 50px;
61         margin: 20px auto;
```

```

62     border: 4px solid black;
63 }
64 ▼ .kotak{
65     float: left;
66     width: 50px;
67     height: 50px;
68     background-color: white;
69 }
70 ▼ .jam-digital p {
71     color: black;
72     font-size: 20px;
73     text-align: center;
74     margin-top: 16px;
75 }
76
77
78 </style>
79 ▼ <div class="jam-digital">
80     <div class="kotak">
81         <p id="jam"></p>
82     </div>
83     <div class="kotak">
84         <p id="menit"></p>
85     </div>
86     <div class="kotak">
87         <p id="detik"></p>
88     </div>
89 </div>
90
91 ▼ <script>
92     window.setTimeout("waktu()", 1000);

```

```

93
94 ▼ function waktu() {
95     var waktu = new Date();
96     setTimeout("waktu()", 1000);
97     document.getElementById("jam").innerHTML = waktu.getHours();
98     document.getElementById("menit").innerHTML = waktu.getMinutes();
99     document.getElementById("detik").innerHTML = waktu.getSeconds();
100 }
101 ▼ </script>
102
103     <p><center>Disusun oleh : </center></p>
104     <p><center>*Gira Yunikristiyanti 0031740</center></p>
105     <p><center>*Latipa Kusuma Dawe 0031719</center></p>
106     <br><p><center><a href="page2.htm"> PROYEK AKHIR </center> </a>.</p></br>
107
108 </body>
109 </html>

```

Program Daftar Menu Proyek Akhir

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>DAFTAR MENU</title>
5
6   </head>
7   <body>
8     <style>
9       body {
10        font: 16px/28px arial, sans-serif;
11        background-color: #eaeaea;
12        background-image: url(sativa.png);
13        color : #333;
14      }
15    </style>
16
17    <style>
18      .container {
19        width: 980px;
20        margin: auto;
21        background-color: #ffffe6;
22      }
23    </style>
24    <div class= "container">
25
26      <style>
27        .header {
28          padding:30px;
29        }
30        .header .judul {
31
32          font-size: 40px;
33          font-weight:bold;
34
35        }
36        .header h1 {
37          text-align: center;
38        }
39
40        .header ul li {
41          font-size: 20px;
42          display: inline-block;
43          margin-top: 30px;
44          margin-right: 5px;
45        }
46
47        .header a {
48          text-decoration: none;
```

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>DAFTAR MENU</title>
5
6   </head>
7   <body>
8     <style>
9       body {
10        font: 16px/28px arial, sans-serif;
11        background-color: #eaeaea;
12        background-image: url(sativa.png);
13        color : #333;
14      }
15    </style>
16
17    <style>
18      .container {
19        width: 980px;
20        margin: auto;
21        background-color: #ffffe6;
22      }
23    </style>
24    <div class= "container">
25
26    <style>
27      .header {
28        padding:30px;
29      }
30      .header .judul {
31
32        font-size: 40px;
33        font-weight:bold;
34      }
35    }
36    .header h1 {
37      text-align: center;
38    }
39
40    .header ul li {
41      font-size: 20px;
42      display: inline-block;
43      margin-top: 30px;
44      margin-right: 5px;
45    }
46
47    .header a {
48      text-decoration: none;
```

```

97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
<style>
  .main {
    width:480px;
    padding: 20px;
    box-sizing:border-box;
    float:left;
  }
  .main h2 {
    font-size: 30px;
    font-weight: bold;
    text-align: center;
  }
  .main .abb a {
    color: #333;
    text-decoration: none;
  }
  .main .penulis {
    font-size:14px;
    margin-top:-5px;
    text-align: center;
  }
  .main .penulis a {
    color: blue;
  }
  .main p {
    margin-bottom: 20px;
    font: 18px arial, sans-serif;
  }
</style>

<div class="main">
  <a class="abb"><a href="#">ABB B23</a></td>
  <td class="penulis"><a href="#">24 Agustus 2020</a>. </td>
  <td>ABB B23 adalah alat pengukur listrik 3 fase, dilengkapi dengan layar backlight yang memberikan informasi tentang energi aktif, arus, tegangan, faktor daya dan frekuensi. data meter dapat dikonversi ke M-Bus / Nirkabel M-Bus dan dikirimkan ke sistem penerima. Untuk mengukur besaran daya aktif digunakan alat ukur yang bernama kWh meter/energi meter. Pentingnya sistem monitoring energi listrik sebagai support system (sistem pendukung), bahwa pemakaian listrik sangat perlu dikendalikan sehingga akan sangat bisa menggunakan listrik secara efisien dan hemat.</td>
  <td>Modbus adalah protokol komunikasi serial yang diterbitkan oleh Modicon pada 1979 untuk diaplikasikan pada programmable logic controller (PLC). Kemudian protokol ini telah menjadi standar de facto protokol komunikasi di industri, dan sekarang modbus merupakan protokol komunikasi dua-arah yang paling umum digunakan sebagai media penghubung dengan perangkat industri atau media elektronik lainnya dengan komputer. komunikasi dengan menggunakan protokol modbus bisa melalui perantara port serial (RS-232, RS485, FO), bisa juga melalui ethernet (LAN) dan jaringan lainnya yang mendukung protokol internet. </td>
</div>

```

```

137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
<style>
  .sidebar {
    width: 5px;
    float:left;
    padding: 10px;
  }
</style>
<div class="sidebar">
  
</div>
</div>
</body>
</html>

```


Program Arduino

```
#include <SPI.h>
#include <Ethernet.h>
#include <ModbusMaster.h>
#include <SD.h>
#define REQ_BUF_SZ 20
#define MAX485_DE 3
#define MAX485_RE_NEG 2
#define LEVEL_HIGH (1)
#define LEVEL_LOW (0)

const int chipSelect = 4;
ModbusMaster node;
void preTransmission()
{
    digitalWrite(MAX485_RE_NEG, 1);
    digitalWrite(MAX485_DE, 1);
}
void postTransmission()
{
    digitalWrite(MAX485_RE_NEG, 0);
    digitalWrite(MAX485_DE, 0);
}
// Inisialisasi IP Adres
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
byte dns[] = {0,0,0,0};
byte gw[] = {0,0,0,0};
byte sm[] = {255,255,255,0};
IPAddress ip(192, 168, 100, 110);
// Initialize the Ethernet server library
EthernetServer server(80);
File webFile;
char HTTP_req[REQ_BUF_SZ] = {0}; // buffered HTTP
request stored as null terminated string
```

```

char req_index = 0;           // index into HTTP_req
buffer

EthernetClient client;

void setup() {
  pinMode(MAX485_RE_NEG, OUTPUT);
  pinMode(MAX485_DE, OUTPUT);
  // Init in receive mode
  digitalWrite(MAX485_RE_NEG, 0);
  digitalWrite(MAX485_DE, 0);
  // Modbus communication runs at 9600 baudrate
  Serial.begin(9600);
  Serial2.begin(9600);
  // Modbus slave ID 1 (slave ID dari abb b23)
  node.begin(1, Serial2);
  // Callbacks allow us to configure the RS485
  transceiver correctly
  node.preTransmission(preTransmission);
  node.postTransmission(postTransmission);
  // Open serial communications and wait for port to open:
  while (!Serial) {
    ; // wait for serial port to connect. Needed for
    native USB port only
  }
  Serial.println("Ethernet WebServer Example");
  Serial.print("Initializing SD card...");
  delay(500);
  if (!SD.begin(chipSelect)) {
    Serial.println("Card failed, or not present");
    while (1);
  }
  Serial.println("card initialized.");
  Ethernet.begin(mac, ip, dns, gw, sm);
  // start the server

```

```

server.begin();
Serial.print("server is at ");
Serial.println(Ethernet.localIP());
pinMode(10, OUTPUT);
digitalWrite(10, HIGH);
// Check for Ethernet hardware present
if (Ethernet.hardwareStatus() == EthernetNoHardware)
{
    Serial.println("Ethernet shield was not found.
Sorry, can't run without hardware. :(");
    while (true) {
        delay(1); // do nothing, no point running without
Ethernet hardware
    }
}
if (Ethernet.linkStatus() == LinkOFF) {
    Serial.println("Ethernet cable is not connected.");
}
Serial.println("Initializing SD card...");
if (!SD.begin(4)) {
    Serial.println("ERROR - SD card initialization
failed!");
    return;    // init failed
}

Serial.println("SUCCESS - SD card initialized.");
// check for index.htm file
if (!SD.exists("index.htm")) {
    Serial.println("ERROR - Can't find index.htm
file!");
    return;    // can't find index file
}
Serial.println("SUCCESS - Found index.htm file.");
if (!SD.exists("page2.htm")) {

```

```
        Serial.println("ERROR - Can't find page2.htm
file!");
        return; // can't find page2 file
    }
    Serial.println("SUCCESS - Found page2.htm file.");
    if (!SD.exists("grafik.htm")) {
        Serial.println("ERROR - Can't find grafik.htm
file!");
        return; // can't find grafik file
    }
    Serial.println("SUCCESS - Found grafik.htm file.");
    if (!SD.exists("Chart.js")) {
        Serial.println("ERROR - Can't find Chart.js
file!");
        return; // can't find js file
    }
    Serial.println("SUCCESS - Found Chart.js file.");
    if (!SD.exists("datalog.txt")) {
        Serial.println("ERROR - Can't find datalog.txt
file!");
        return; // can't find datalog file
    }
    Serial.println("SUCCESS - Found datalog.txt file.");

    if (!SD.exists("picture1.png")) {
        Serial.println("ERROR - Can't find picture1.png
file!");
        return; // can't find logo polman file
    }
    Serial.println("SUCCESS - Found picture1.png
file.");
    if (!SD.exists("blue.png")) {
        Serial.println("ERROR - Can't find blue.png
file!");
        return; // can't find background main page file
```

```

    }
    Serial.println("SUCCESS - Found blue.png file.");
    if (!SD.exists("pt.png")) {
        Serial.println("ERROR - Can't find pt.png
file!");
        return; // can't find logo pt pmm file
    }
    Serial.println("SUCCESS - Found pt.png file.");
    if (!SD.exists("abb.jpg")) {
        Serial.println("ERROR - Can't find abb.jpg
file!");
        return; // can't find kwh abb file
    }
    Serial.println("SUCCESS - Found abb.jpg file.");
    if (!SD.exists("sativa.png")) {
        Serial.println("ERROR - Can't find sativa.png
file!");
        return; // can't find background page2 file
    }
    Serial.println("SUCCESS - Found sativa.png file.");
    if (!SD.exists("mcb.jpg")) {
        Serial.println("ERROR - Can't find mcb.jpg
file!");
        return; // can't find mcb photo file
    }
    Serial.println("SUCCESS - Found mcb.jpg file.");
}

```

```

void loop() {
    static uint32_t i;
    i++;
    node.setTransmitBuffer(0, lowWord(i));
    node.setTransmitBuffer(1, highWord(i));
}

```

```

EthernetClient client = server.available();
if (client) { // got client?
  Serial.println("new client");
  // an http request ends with a blank line
  boolean currentLineIsBlank = true;
  while (client.connected()) {

    if (client.available()) {
      char c = client.read();
      // buffer first part of HTTP request in HTTP_req
      array (string)
      // leave last element in array as 0 to null
      terminate string (REQ_BUF_SZ - 1)
      if (req_index < (REQ_BUF_SZ - 1)) {
        HTTP_req[req_index] = c;           // save
        HTTP request character
        req_index++;
      }
      Serial.write(c);
      if (c == '\n' && currentLineIsBlank) {

        // send a standard http res ponse header
        client.println("HTTP/1.1 200 OK");
        client.println("Content-Type: text/html");
        client.println("Connection: close"); // the
        connection will be closed after completion of the
        response

        client.println();

        Serial.println("<-----str contains executed-
        ----->");
        // untuk membuka web halaman pertama
        if (StrContains(HTTP_req, "GET / ")
        || StrContains(HTTP_req, "GET /index.htm")) {

```

```

        webFile = SD.open("index.htm");
    }
    else if (StrContains(HTTP_req, "GET
/page2.htm")) {
        webFile = SD.open("page2.htm"); // untuk
membuka web halaman kedua
        Serial.println("<-----page2.htm executed-----
--->");
    }
    else if (StrContains(HTTP_req, "GET
/grafik.htm")) {
        webFile = SD.open("grafik.htm"); // untuk
membuka web halaman kedua/grafik
        Serial.println("<-----grafik.htm executed----
---->");
    }
    else if (StrContains(HTTP_req, "GET /Chart.js"))
{
        webFile = SD.open("Chart.js"); // untuk
membuka web halaman kedua/js grafik
        if (webFile) {
            Serial.println("<-----found Chart.js -----
-->");
        }
        Serial.println("<----- Chart.js executed ----
---->");
    }
    else if (StrContains(HTTP_req, "GET
/datalog.txt")) {
        webFile = SD.open("datalog.txt"); //untuk
melihat tampilan dari data kita saat disimpan di sd card
    }
    else if (StrContains(HTTP_req, "GET
/picture1.png")) {
        webFile = SD.open("picture1.png");
//gambar logo polman pada menu utama web

```

```

}
    else if (StrContains(HTTP_req, "GET /blue.png"))
    {
        webFile          =          SD.open("blue.png");
//background
    }
    else if (StrContains(HTTP_req, "GET /pt.png"))
    {
        webFile = SD.open("pt.png");          //gambar
logo pt pada menu utama web
    }
    else if (StrContains(HTTP_req, "GET /abb.jpg"))
    {
        webFile = SD.open("abb.jpg");          //gambar
kwh meter
    }
    else if (StrContains(HTTP_req, "GET
/sativa.png")) {
        webFile          =          SD.open("sativa.png");
//background
    }
    else if (StrContains(HTTP_req, "GET /mcb.jpg"))
    {
        webFile = SD.open("mcb.jpg");          //gambar
mcb
    }

    else if (StrContains(HTTP_req, "GET /realdata"))
//menampilkan Data pada tabel
    {
        //program arduino untuk tampilan pada web
        client.println("<HEAD>");
        client.println("<META          http-equiv=refresh
content = 300>"); // refresh data setiap 5 menit
        client.println("<meta          name=viewport
content=width=device-width, initial-scale=1>");

```



```

client.println("<style>");
    client.println("body { background-image:
blue.png;}");
    client.println("table {border-collapse:
collapse;border-spacing: 0;width: 100%;border: 2px solid
grey;}");
    client.println("th,      td      {text-align:
center;padding: 10px;}");
    client.println("tr:nth-
child(even){background-color: lightgrey}");
    client.println("</style>");
    client.println("</HEAD>");
    client.println("<body>");
    client.println("<center><h1>Energy Meter Real
Value</h1></center>");
    client.println("<p>Desripsi 1</p>");
    client.println("<p>Desripsi 2</p>");
    client.println("<div      style=overflow-
x:auto;>");
    client.println("<table border=1>");
    client.println("<tr>");
    client.println("<th>VOLTAGE (V) </th>");
    client.println("<th>CURRENT (Amp) </th>");
    client.println("<th>FREQUENCY (Hz) </th>");
    client.println("<th>DAYA AKTIF (Watt) </th>");
    client.println("<th>DAYA      AKTIF
TOTAL (Watt) </th>");
    client.println("<th>DAYA REAKTIF (VAR) </th>");
    client.println("<th>DAYA SEMU (VA) </th>");
    client.println("</tr>");

uint8_t result;
uint16_t data[6];

```

```

char StrVoltage[6];

webFile = SD.open("datalog.txt", FILE_WRITE);
node.preTransmission(preTransmission);
result = node.readHoldingRegisters(23296, 16);
node.postTransmission(postTransmission);
Serial.println("Node Result: ");
Serial.print(result);

if (result == node.ku8MBSuccess) {
    client.println("<tr>");
    client.println("<td>");
    client.print("Voltage 1 : ");

client.println(node.getResponseBuffer(0x01)/10.0f);
    delay(1000);
    client.println("</td>");

        if (webFile) {
            webFile.println("Voltage 1 : ");

webFile.println(node.getResponseBuffer(0x01)/10.0f);
        }

        client.println("<td>");
        client.print("Current 1 : ");

client.println(node.getResponseBuffer(0x0D)/100.00f);
    delay(1000);
    client.println("</td>");

```

```

        if (webFile) {
            webFile.println("Current 1 : ");
webFile.println(node.getResponseBuffer(0x0D)/100.00f);
        }

        client.println("<td>");

        result = node.readHoldingRegisters(4165,
16);
        client.print("Frekuensi : ");

client.println(node.getResponseBuffer(0x02)/1000.00f);
        delay(1000);
        client.println("</td>");

        if (webFile) {
            webFile.println("Frekuensi : ");
webFile.println(node.getResponseBuffer(0x02)/1000.00f);
        }
        client.println("<td>");

        result = node.readHoldingRegisters(23316,
16);
        client.print("Daya Aktif 1 : ");

client.println(node.getResponseBuffer(0x03)/100.00f);
        delay(1000);
        client.println("</td>");

        if (webFile) {

```

```

webFile.println("Daya Aktif 1 : ");

webFile.println(node.getResponseBuffer(0x03)/100.00f);
    }

        client.println("<td>");
        client.print("Daya total : ");

client.println(node.getResponseBuffer(0x01)/100.00f);
    delay(1000);
    client.println("</td>");

        if (webFile) {
            webFile.println("Daya total : ");
webFile.println(node.getResponseBuffer(0x01)/100.00f);
    }

        client.println("<td>");
        result = node.readHoldingRegisters(23326,
16);
        client.print("Daya Reaktif 1 : ");

client.println(node.getResponseBuffer(0x01)/100.00f);
    delay(1000);
    client.println("</td>");

        if (webFile) {
            webFile.println("Daya Reaktif 1 : ");
webFile.println(node.getResponseBuffer(0x01)/100.00f);

```

```
}

    client.println("<td>");
    result = node.readHoldingRegisters(23334,
16);
    client.print("Daya Semu 1 : ");

client.println(node.getResponseBuffer(0x01)/100.00f);
    delay(1000);
    client.println("</td>");
    client.println("</tr>");

        if (webFile) {
            webFile.println("Daya Semu 1 : ");
webFile.println(node.getResponseBuffer(0x01)/100.00f);
        }
        client.println("<tr>");

        result = node.readHoldingRegisters(23296, 16);
        client.println("<td>");
        client.print("Voltage 2 : ");

client.println(node.getResponseBuffer(0x03)/10.0f);
    delay(1000);
    client.println("</td>");

        if (webFile) {
            webFile.println("Voltage 2 : ");
```

```
webFile.println(node.getResponseBuffer(0x03)/10.0f);
    }

    client.println("<td>");
    client.print("Current 2 : ");

client.println(node.getResponseBuffer(0x0F)/100.0f);
    delay(1000);
    client.println("</td>");

    if (webFile) {
        webFile.println("Current 2 : ");
webFile.println(node.getResponseBuffer(0x0F)/100.0f);
    }

    client.println("<td>");
    client.println("</td>");

    client.println("<td>");
    result = node.readHoldingRegisters(23322,
16);
    client.print("Daya Aktif 2 : ");

client.println(node.getResponseBuffer(0x01)/100.00f);
    delay(1000);
    client.println("</td>
```

```
");

        if (webFile) {
            webFile.println("Daya Aktif 2 : ");
webFile.println(node.getResponseBuffer(0x01)/100.00f);
        }

        client.println("<td>");
        client.println("</td>");

        client.println("<td>");
        result = node.readHoldingRegisters(23328,
16);
        client.print("Daya Reaktif 2 : ");

        client.println(node.getResponseBuffer(0x01)/100.00f);
        delay(1000);
        client.println("</td>");

        if (webFile) {
            webFile.println("Daya Reaktif 2 : ");
webFile.println(node.getResponseBuffer(0x01)/100.00f);
        }

        client.println("<td>");
```

```
result = node.readHoldingRegisters(23336, 16);
    client.print("Daya Semu 2 : ");

client.println(node.getResponseBuffer(0x01)/100.00f);
    delay(1000);
    client.println("</td>");
    client.println("</tr>");

        if (webFile) {
            webFile.println("Daya Semu 2 : ");

webFile.println(node.getResponseBuffer(0x01)/100.00f);
        }
    result = node.readHoldingRegisters(23296,
16);
    client.println("<td>");
    client.print("Voltage 3 : ");

client.println(node.getResponseBuffer(0x05)/10.0f);
    delay(1000);
    client.println("</td>");
        if (webFile) {
            webFile.println("Voltage 3 : ");

webFile.println(node.getResponseBuffer(0x05)/10.0f);
```



```

}

        result = node.readHoldingRegisters(23310,
16);
        client.println("<td>");
        client.print("Current 3 : ");

client.println(node.getResponseBuffer(0x03)/100.00f);
        delay(1000);
        client.println("</td>");

        if (webFile) {
            webFile.println("Current 3 : ");
webFile.println(node.getResponseBuffer(0x03)/100.00f);
        }
        client.println("<td>");
        client.println("</td>");

        client.println("<td>");
        result = node.readHoldingRegisters(23320,
16);
        client.print("Daya Aktif 3 : ");

client.println(node.getResponseBuffer(0x01)/100.00f);
        delay(1000);
        client.println("</td>");

client.println("<td>");
        result = node.readHoldingRegisters(23330,
16);
        client.print("Daya Reaktif 3 : ");

```

```
client.println(node.getResponseBuffer(0x01)/100.00f);
    delay(1000);
    client.println("</td>");

        if (webFile) {
            webFile.println("Daya Reaktif 3 : ");
webFile.println(node.getResponseBuffer(0x01)/100.00f);
        }

        client.println("<td>");

result = node.readHoldingRegisters(23338, 16);
    client.print("Daya Semu 3 : ");

client.println(node.getResponseBuffer(0x01)/100.00f);
    delay(1000);
    client.println("</td>");
    client.println("</tr>");
    client.println("</table>");
    client.println("</div>");
    client.println("</body>");
    client.println("</html>");
if (webFile) {
    webFile.println("Daya Semu 3 : ");
    webFile.println(node.getResponseBuffer(0x01)/100.00f);
    }
        client.println("<p><center>Go back to <a
href=page2.htm> Page2 </center></a>.</p>");
    }
```

```

else if (result ==
node.ku8MBInvalidSlaveID)
{
    client.print("Modbus Error : ");

client.println("Slave Failure");
}
else if (result ==
node.ku8MBIllegalDataValue)
{
    client.print("Modbus Error : ");
    client.println("Illegal Value");
}
else if (result ==
node.ku8MBIllegalDataAddress)

{
    client.print("Modbus Error : ");
    client.println("Illegal Address");
}
delay(2);
Serial.println("<-----end bacaregister-
---->");

}
// send web page to client
if (webFile) {
while(webFile.available())
{
    client.write(webFile.read());
}
webFile.close();
}

```

```

0         // reset buffer index and all buffer elements to

req_index = 0;
StrClear(HTTP_req, REQ_BUF_SZ);
break;
    }
client.print("Slave Invalid");
    }
    else if (result ==
node.ku8MBResponseTimedOut)
    {
        client.print("Modbus Error : ");
        client.println("Time Out");
    }
    else if (result ==
node.ku8MBInvalidFunction)
    {
        client.print("Modbus Error : ");
        client.println("Invalid Function");
    }
    else if (result == node.ku8MBInvalidCRC)
    {
        client.print("Modbus Error : ");
        client.println("CRC");
    }
    else if (result ==
node.ku8MBSlaveDeviceFailure)

```

```

        currentLineIsBlank = true;
    }

    else if (c != '\r') {
        // a text character was received from
client
        currentLineIsBlank = false;
    }
    } // end if (client.available())
} // end while (client.connected())

// give the web browser time to receive the data
delay(1);
//dataFile.close();
client.stop(); // close the connection:
char found = 0;
char index = 0;
char len;
len = strlen(str);
if (strlen(sfind) > len) {
return 0;
}
while (index < len) {
    if (str[index] == sfind[found]) {
        found++;
        if (strlen(sfind) == found) {
            return 1;
        }
    }
}

```

```
    else {  
        found = 0;  
    }  
    index++;  
}  
return 0;
```

