

Implementasi Sistem Monitoring Produksi Energi PLTS Berbasis Protokol Modbus RTU Dan Modbus TCP

Fransiscus Xaverius Ariwibisono¹, Widodo Pudji Muljanto²

^{1,2} Institut Teknologi Nasional Malang

E-mail: ¹ari@lecturer.itn.ac.id, ²widodo_pm@lecturer.itn.ac.id

Abstrak

Pemanfaatan sumber energi baru terbarukan (EBT) saat ini mulai berkembang pesat seiring dengan kebutuhan sumber listrik yang ramah lingkungan. Pembangkit listrik tenaga surya (PLTS) merupakan salah satu pilihan solusi bagi masyarakat terutama yang berada di daerah yang belum terjangkau jaringan listrik. Produksi energi listrik yang dihasilkan sangat tergantung dari kapasitas panel surya yang terpasang. Dalam rangka pemantauan produksi energi listrik PLTS serta beban pemakaian dibutuhkan sarana yang dapat memberikan informasi secara real-time. Selain itu juga dapat dengan mudah diakses dari jarak jauh dengan memanfaatkan jaringan yang menggunakan protokol TCP/IP. Permasalahan yang muncul kemudian bahwa data sensor yang diperoleh dari hasil pembacaan harus sesuai dengan parameter standar yang berlaku untuk semua peralatan sensor. Protokol Modbus RTU dan Modbus TCP merupakan pilihan yang tepat untuk menjawab permasalahan tersebut. Dalam penelitian ini dibahas terkait implementasi protokol Modbus RTU/TCP dalam lingkup PLTS. Dengan demikian, sistem monitoring energi listrik PLTS ini memungkinkan diterapkan pada jaringan yang lokasinya berada yang jauh dari pusat kendali atau monitoring.

Kata Kunci—Modbus RTU, Modbus TCP, RS-485

Abstract

The use of renewable energy sources is currently starting to grow rapidly in line with the need for environmentally friendly power sources. Solar power plants are a choice of solutions for the community, especially those in areas that have not reached by the electricity grid. The production of the resulting electrical energy is highly dependent on the capacity of the installed solar panels. In order to monitor solar power plants electricity production and usage load, a facility is needed that can provide real-time information. Besides that, it can easily accessed remotely by utilizing a network that uses the TCP/IP protocol. The problem that arises then is that the sensor data obtained from the readings must comply with the standard parameters that apply to all sensor equipment. The Modbus RTU and Modbus TCP protocols are the right choice to answer this problem. This research will discuss the implementation of the Modbus RTU/TCP protocol within the scope of solar power plants. Thus, solar power plants electrical energy monitoring system allows it applied to networks that are located far from the control or monitoring center.

Keywords— Modbus RTU, Modbus TCP, RS-485

1. PENDAHULUAN

Dalam manajemen energi ketenagalistrikan dibutuhkan sistem kerja yang efisien serta efektif dalam rangka menciptakan sebuah produktifitas yang

tinggi serta akurasi data yang tinggi pula. Sistem yang ada saat ini di beberapa lokasi produksi ketenagalistrikan masih dipantau secara *offline* atau setempat.

Untuk mengatasi kondisi tersebut tentunya diperlukan sebuah sistem yang mampu mempercepat proses pengambilan

data dan informasi dengan akurasi yang valid dan *real-time*. Dalam penelitian ini bermaksud untuk membuat sebuah sistem monitoring pemakaian energi listrik yang bersumber dari pembangkit listrik tenaga surya (PLTS). Melalui sistem tersebut diharapkan dapat meningkatkan akurasi sekaligus dapat meningkatkan efisiensi operasional karena dapat dilakukan tanpa harus datang ke lokasi PLTS.

Perangkat antarmuka yang digunakan adalah RS-485 dan TCP/IP yang menggunakan protokol Modbus. Adapun protokol yang berjalan dalam komunikasi serial RS-485 yaitu Modbus Rtu, sedangkan Modbus TCP berjalan pada komunikasi TCP/IP.

Penelitian ini akan melakukan analisis parameter yang dihasilkan oleh PLTS *on-grid* dengan kapasitas produksi maksimal sebesar 500kWp yang berada di area kampus II Institut Teknologi Nasional Malang. Penelitian dimulai dari pengumpulan data-data yang diperlukan untuk analisis.

Landasan Teori

1. Penelitian Terkait

Dalam penerapan di industri, komunikasi RS-485 yang menggunakan Modbus RTU dan komunikasi TCP/IP dihasilkan pengujian menunjukkan hasil yang baik.[9] Hasil tersebut diperoleh dari pengujian pada tanda operator pada LED yang menyala pada komunikasi Modbus RTU.

Implementasi protokol Modbus pada power meter SPM 91 dalam rangka mendapatkan data pemakaian daya listrik. [11] Hasilnya diperoleh data perkiraan penggunaan daya listrik yang diharapkan dapat menghemat pemakaian listrik.

Perangkat interface yang sering dipergunakan dalam menghubungkan perangkat software dan hardware yaitu RS-485 dan TCP/IP. Sistem pertukaran data menggunakan modbus. Modbus RTU merupakan protokol modbus yang berjalan pada komunikasi serial RS-485, sedangkan modbus TCP berjalan pada komunikasi TCP/IP.

Penyebutan secara lengkap RS-485 adalah EIA/TIA-485 merupakan *Standard*

for Electrical Characteristics of Generators and Receivers for use in a Balanced Digital Multipoint System yaitu standar komunikasi serial asinkron yang berstandar industri.

Modbus merupakan sebuah protokol komunikasi pada jaringan yang berstandar internasional dimana banyak diterapkan pada industri dan bersifat *open source*. Awal mula Modbus dipublikasikan oleh Modicon pada tahun 1979 yang diimplementasikan pada PLC (*Programmable Logic Controllers*)[10].

1.1. Protokol Modbus

Modbus merupakan protokol komunikasi serial yang dibuat oleh Modicon untuk diterapkan pada PLC. Modbus digunakan untuk mengirimkan sinyal dari perangkat instrumentasi dan kontrol yang kemudian kembali ke *controller* utama. Selain itu juga, Modbus sering dipergunakan untuk menghubungkan komputer pengawasan dengan unit terminal remote (RTU) kontrol pengawasan dan data akuisisi (SCADA).[14] Ada dua versi protokol modbus yaitu Modbus RTU dan Modbus ASCII untuk baris serial sedangkan untuk ethernet disebut Modbus TCP.

Protokol modbus memungkinkan komunikasi dua arah antar perangkat jaringan yang sama. Adapun pertimbangan mengapa menggunakan protokol Modbus adalah sebagai berikut :

- a. *Open protocol*.
- b. Relatif mudah untuk dihubungkan dengan jaringan industri.
- c. Dapat melakukan transfer data *raw bits* atau *words* dengan semua perangkat industri.

1.2. Komunikasi RS-485

Komunikasi RS485 selain dapat digunakan untuk komunikasi *multidrop* yaitu berhubungan secara *one to many* dengan jarak yang jauh teknik ini juga dapat digunakan untuk menghubungkan 32 unit beban sekaligus hanya dengan menggunakan dua buah kabel saja tanpa memerlukan referensi *ground* yang sama antara unit yang satu dengan unit lainnya. [1]

Sistem komunikasi dengan menggunakan RS485 ini dapat digunakan untuk komunikasi data antara 32 unit peralatan elektronik hanya dalam dua kabel saja. Selain itu, jarak komunikasi dapat mencapai 1,6 km dengan digunakannya kabel AWG-24 *twisted pair*.

Komunikasi bus RS-485 memiliki mode transmisi *balanced differential* dimana memiliki dua sinyal yaitu A dan B yang berbeda tengangan antara keduanya. Jika *input low* maka line A akan berfungsi sebagai referensi terhadap B. Setiap kali perangkat (sensor atau *control*) akan mengirimkan data maka harus memeriksa jalur terlebih dahulu. [2] Apakah jalur yang akan digunakan sebagai media pengiriman data tersebut sedang ada trafik data atau tidak. Jika jalur sedang dipergunakan maka peralatan harus menunggu hingga jalur kosong atau tidak ada pengiriman data.

Dalam proses pengiriman data, agar sampai ke peralatan yang dituju maka terlebih dahulu diawali dengan Slave ID kemudian dilanjutkan dengan data yang dikirimkan. Jika peralatan lain menerima data yang tidak memiliki ID yang sama dengan slave ID maka peralatan tersebut akan menolak atau mengabaikan data tersebut. Namun jika slave ID yang dikirimkan sudah sesuai dengan ID dari peralatan tersebut maka data selanjutnya diterima untuk proses selanjutnya.

1.3. Frame Modbus

Address	Function code	Data	error check
---------	---------------	------	-------------

Gambar 1. Frame Modbus

Frame Modbus terdiri dari :

1. Slave Address

Pada byte pertama merupakan alamat *slave* yang terdiri dari 1 byte. Alamat *slave* ditentukan yaitu 1 sampai 247. Sedangkan *address 0* digunakan master untuk semua *slave*.

2. Function Code

Pada *byte* kedua merupakan *Function Code* dimana perintah dari master harus dilakukan oleh *slave*. Adapun daftar

function code terdapat pada Gambar 2 dibawah ini :

Data	Read	Write (1 data)	Write (1 > data)	Start Reg
Coils	FC01	FC05	FC15	00001
Discrete input	FC02			10001
Input register	FC04			30001
Holding register	FC03	FC06	FC16	40001

Gambar 2. Function Code

3. Byte Data

Variasi jumlah byte data tergantung pada jumlah data yang dituliskan ke slave. Adapun byte data berisi alamat register, jumlah data, serta data yang dituliskan alamat *register*.

4. Error Check (CRC)

Cyclic Redundancy Check (CRC) merupakan algoritma untuk memastikan integritas data serta untuk melakukan pengecekan kesalahan pada suatu data yang akan dikirimkan atau disimpan. CRC terletak pada dua byte terakhir.[7] Byte ini digunakan untuk mendeteksi jika ada kesalahan pada *frame* modbus. Data yang ditransmisikan sangat rentan mengalami kesalahan. Hal ini bias terjadi karena *noise* atau kerusakan pada perangkat keras. CRC bekerja menggunakan perhitungan matematika terhadap sebuah bilangan berdasarkan total bit yang akan ditransmisikan atau disimpan. Cara kerja ini disebut *checksum*.

5. Modbus Exception Response

Response exception merupakan respon yang berasal dari *slave* pada saat terjadi kondisi tidak normal atau *error*. Pada saat *slave* menerima *query*, akan tetapi *slave* tidak dapat menangani perintah tersebut, maka *slave* akan mengirimkan sebuah respon *exception frame* respon.

Address	Function code	Data	error check
Address	Function code + 80	exception code	error check

Gambar 3. Perbandingan frame Modbus normal dengan saat terjadi exception

1.4. Modbus RTU

Protokol Modbus RTU menggunakan komunikasi RS-485 dan

representasi nilai data biner. Format RTU mengikuti request data dengan CRC *checksum* sebagai mekanisme pemeriksaan kesalahan (*error check*) dalam rangka memastikan nilai data. Dalam setiap *message* modbus dibentuk dalam sebuah *frame* yang dipisahkan oleh periode-periode pada saat *idle* (tanpa adanya komunikasi dan ON/OFF port). Komunikasi Modbus RTU dipergunakan dalam sistem pemantauan skala kecil yang menggunakan sensor dan *Human Machine Interface* (HMI) yang letaknya tidak jauh.[12]

Start	Slave ID	Function code	Data	CRC	Stop
3,5 bytes	1 byte	1 byte	n bytes	2 bytes	3,5 bytes

Gambar 4. Format Frame Modbus RTU

Nama Register	Tipe Objek	Akses	Keterangan
Coils	Single bit	Read Write	Master read write coil
Discrete input	Single bit	Read only	Slave update request
Input register	16 bit word	Read only	Slave update request
Holding register	16 bit word	Read Write	Master read write

Gambar 5. Register Modbus RTU

Akses data pada protokol Modbus memiliki 4 jenis yaitu:

a. *Coils*

Register untuk menyimpan nilai diskrit ON atau OFF. Panjang data tiap register 16 bit. Register untuk mengaktifkan *coil* (ON) bernilai 0xFF00. Dan untuk menonaktifkan *coil* (OFF) bernilai 0x0000. Data *coil* bisa disimpan pada register 00000 sampai 09999.

b. *Input Relay*

Input relay merupakan kebalikan dari *coil* dimana digunakan untuk mengetahui status relay dalam kondisi ON atau OFF.[5] Untuk master, input relay bersifat *read only* dan hanya dapat diubah oleh *slave*. Data *input relay* ini disimpan pada register 10001 sampai 19999.

c. *Input Register*

Menyimpan data analog yang bernilai 0 sampai 65535. *Input register* bersifat *read only* untuk master. Data *input register* disimpan pada register 30001 sampai 39999.

d. *Holding Register*

Digunakan untuk menyimpan data rentang 0 sampai 65535 dengan alamat *register* 40001 sampai 49999.

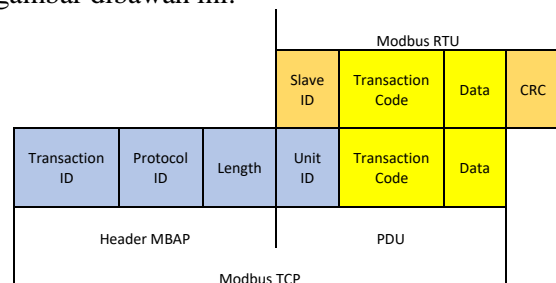
1.5. Modbus TCP

Transmission Control Protocol (TCP) dan Internet Protocol (IP) adalah protokol yang dipergunakan secara bersamaan sebagai protokol *transport* pada internet. Pada saat data Modbus dikirim menggunakan protokol Modbus TCP, [8] maka data akan diteruskan ke TCP melalui alamat IP. Kemudian IP menempatkan data pada paket (*datagram*) untuk dikirim.

Sebelum TCP mengirimkan data, harus terlebih dahulu membuat sambungan (*link*) karena protokol Modbus TCP berbasis koneksi. Master atau *client* pada modbus TCP akan menentukan koneksi dengan *slave* atau server. Kemudian server menunggu *client* untuk *input link* untuk membuat sambungan. Setelah itu server akan respon permintaan dari *client* hingga *status client closed*.

Dalam melakukan transfer data, Modbus TCP/IP lebih cepat dibandingkan dengan Modbus RTU. Dalam implementasi sistem SCADA maupun *automation* dimana tingkat transfer data yang padat maka disarankan menggunakan protokol modbus TCP/IP agar tingkat *real-time* nya lebih tinggi.

Salah satu perbedaan yang sangat mendasar antara Modbus RTU dengan Modbus TCP ada pada *frame* data yang dikirimkan. Hal tersebut dapat dilihat pada gambar dibawah ini:



Gambar 6. Format Frame Modbus TCP

1.6. Modbus Power Meter

Alat ini digunakan untuk mengukur total energi listrik yang dikonsumsi oleh peralatan yang diambil dari energi listrik dari satu daya utama. KWH meter series ini banyak diterapkan pada sistem kontrol, sistem SCADA, dan energi sistem manajemen, otomatisasi gardu transformator, mendistribusikan otomatisasi bersih, monitor daya listrik komunitas tempat tinggal dll.



Gambar 7. Intelligent Meter TOKY

Power Meter ini mempunyai fitur, antara lain:

- 1) Dapat mengukur Item: 3 phase power jaringan Tegangan, Arus, Daya Aktif, Daya Reaktif / Frekuensi, Power Factor.
- 2) Dua/OFF input dan 2 ON/OFF output (opsional empat ON/OFF input).
- 3) RS485 antarmuka protokol komunikasi Modbus RTU.
- 4) Active energy pulse output.

1.7. Data Register

No.	reflection add.	Variable name	register	Data type	read/write	unit	note
1	0x4000	Phase voltage A	2	long	R	0,1V	
2	0x4002	Phase voltage B	2	long	R	0,1V	
3	0x4004	Phase voltage C	2	long	R	0,1V	
4	0x4006	Line voltage AB	2	long	R	0,1V	
5	0x4008	Line voltage BC	2	long	R	0,1V	
6	0x400a	Line voltage CA	2	long	R	0,1V	
7	0x400c	Phase current A	2	long	R	0,001A	
8	0x400e	Phase current B	2	long	R	0,001A	
9	0x4010	Phase current C	2	long	R	0,001A	
10	0x4012	Active power A	2	long	R	0,1W	
11	0x4014	Active power B	2	long	R	0,1W	
12	0x4016	Active power C	2	long	R	0,1W	
13	0x4018	Total active power	2	long	R	0,1W	
14	0x401a	Reactive power A	2	long	R	0,1var	
15	0x401c	Reactive power B	2	long	R	0,1var	
16	0x401e	Reactive power C	2	long	R	0,1var	
17	0x4020	Total reactive power	2	long	R	0,1var	
18	0x4022	Apparent power A	2	long	R	0,1VA	
19	0x4024	Apparent power B	2	long	R	0,1VA	
20	0x4026	Apparent power C	2	long	R	0,1VA	
21	0x4028	Total apparent power	2	long	R	0,1VA	
22	0x402a	Power factor A	2	long	R	0,001	
23	0x402c	Power factor B	2	long	R	0,001	
24	0x402e	Power factor C	2	long	R	0,001	
25	0x4030	Total power factor	2	long	R	0,001	
26	0x4032	Frequency	2	long	R	0,01Hz	
27	0x4034	Total Kwh	2	long	R	0,01kWh	
28	0x4036	Total Kvarh	2	long	R	0,01kvarh	
29	0x4038	Forward Kwh	2	long	R	0,01kWh	LED display type power meter do not have this function
30	0x403a	Backward Kwh	2	long	R	0,01kWh	
31	0x403c	Forward Kvarh	2	long	R	0,01kvarh	
32	0x403e	Backward Kvarh	2	long	R	0,01kvarh	

Tabel 1. Data Register.

Slave memiliki alamat yang unik, dimana digunakan untuk mengirim perintah dari master sesuai dengan alamat yang dituju. Hanya alamat yang dituju yang dapat menerima perintah, meskipun perintah tersebut dikirim ke slave yang lain.[3] Jaringan Modbus dapat menghubungkan sampai dengan 247 slave dengan alamat slave yang berbeda pada 1 master. Setiap perintah yang akan dikirimkan memiliki mekanisme pemeriksaan data untuk memastikan data yang dikirim tidak terjadi kerusakan atau kesalahan. Perintah pada Modbus dapat berisi perintah untuk mengubah nilai register, membaca nilai register, maupun membaca port I/O. Modbus master dapat mengirimkan perintah dalam dua mode, yaitu unicast dan broadcast. Pada mode unicast, perangkat master mengirim perintah ke satu slave, sedangkan mode broadcast perangkat master mengirim perintah ke semua slave.

1.8. Modbus RTU to TCP Converter

Konverter ini memiliki fungsi merubah jenis komunikasi dari Modbus RTU ke Modbus TCP berbasis IP address. Fitur setup dapat diakses melalui browser yang tertanam dalam perangkat ini diantaranya setup IP static dan dinamik.



Gambar 8. RS-485 to Ethernet Converter

2. METODE PENELITIAN

1. Pendahuluan,

Dalam tahapan penelitian ini dilakukan observasi ke lapangan dengan tujuan untuk menganalisa kebutuhan sistem yang terdiri dari kebutuhan fungsional dan non-fungsional dan juga kebutuhan perangkat keras dan perangkat lunak. Kemudian selanjutnya adalah melakukan perancangan sistem yang meliputi pemuatan arsitektur sistem, blok diagram, skematik rangkaian

serta desain alat. Adapun bahan-bahan yang digunakan dalam penelitian ini adalah power meter Toky DS9L serias, koverter *RS-485 to Ethernet Modbus gateway* tipe *USR-DR302* serta infrastruktur jaringan *Local Area Network (LAN)*. [6]

2. Metode Penelitian

Dalam tahapan metode penelitian diperlukan kerangka kerja yaitu yang terkait secara sistematis. Hal ini dilakukan untuk mempermudah dalam rangka melakukan penelitian. Sebelumnya harus membuat kerangka penelitian ini terlebih dahulu untuk menganalisa topik penelitian.

Analisa penelitian menjelaskan bagaimana proses dalam mengambil data yang dipergunakan untuk penelitian ini. Metode pengumpulan data dilakukan melalui observasi dan verifikasi. Baik data yang langsung terbaca dalam Modbus maupun secara pengukuran riil di titik tempat sensor ditempatkan. Penulis mengupayakan hal tersebut agar data yang disajikan maupun diolah nantinya dapat dipertanggungjawabkan secara ilmiah.

Dalam tahapan analisa penelitian, penulis menjelaskan proses dalam rangka pengambilan data yang diperlukan untuk penelitian ini.

Arsitektur sistem yang dibuat terdiri dari aplikasi desktop sebagai master Modbus dan *slave* pada power meter untuk mendapatkan data tegangan, arus, serta daya energi listrik. [4]

3. HASIL DAN PEMBAHASAN

Berikut merupakan langkah-langkah yang dilaksanakan dalam penelitian ini:

- a. Menyiapkan peralatan seperti converter *RS-485 to Ethernet*, Power Meter Toky DS9L, sensor arus CT, jaringan *Local Area Network (LAN)*.
- b. Menyiapkan perangkat lunak untuk mendukung interface dashboard parameter yang akan ditampilkan yaitu Haiwell Cloud SCADA dan Node Red

Sebagai tambahan informasi bahwa sumber energi listrik kampus II ITN Malang,

yaitu untuk memenuhi kebutuhan operasional kampus yang berasal dari sumber PLN 20 kV kemudian diturunkan menjadi 400 V menggunakan transformator *Step Down* dengan daya 400 kVa. Selain itu, sumber energi listrik kampus II ITN Malang juga di supply dari *PLTS on Grid*, dibangun diatas lahan seluas setengah hektar, dimana terdapat sekitar 1.114 panel surya ditargetkan dapat menghasilkan listrik 0,5 Mega Watt peak (MWp) atau 500 kilo watt peak, yang di interkoneksi (*On Grid*) dengan sumber yang dari PLN.

Variable name	Register type	Address format	Register address
1 PhaseA	4X(Analog output,Function code 03 06)	Decimal	16384
2 PhaseA_1	4X(Analog output,Function code 03 06)	Decimal	16385
3 Freq	4X(Analog output,Function code 03 06)	Decimal	16434
4 Freq1	4X(Analog output,Function code 03 06)	Decimal	16435
5 Arus	4X(Analog output,Function code 03 06)	Decimal	16396
6 Arus1	4X(Analog output,Function code 03 06)	Decimal	16397
7 Power	4X(Analog output,Function code 03 06)	Decimal	16402
8 P_Factor	4X(Analog output,Function code 03 06)	Decimal	16426
9 P_Factor1	4X(Analog output,Function code 03 06)	Decimal	16427
10 Power1	4X(Analog output,Function code 03 06)	Decimal	16403
11 PhaseB	4X(Analog output,Function code 03 06)	Decimal	16386
12 PhaseB_1	4X(Analog output,Function code 03 06)	Decimal	16387
13 ArusB	4X(Analog output,Function code 03 06)	Decimal	16398
14 ArusB_1	4X(Analog output,Function code 03 06)	Decimal	16399
15 P_FactorB	4X(Analog output,Function code 03 06)	Decimal	16428
16 P_FactorB_1	4X(Analog output,Function code 03 06)	Decimal	16429
17 PowerB	4X(Analog output,Function code 03 06)	Decimal	16404
18 PowerB_1	4X(Analog output,Function code 03 06)	Decimal	16405
19 PhaseC	4X(Analog output,Function code 03 06)	Decimal	16388
20 PhaseC_1	4X(Analog output,Function code 03 06)	Decimal	16389

Gambar 9 Mapping Modbus Variabel

Pada saat data sensor *slave* mengirimkan parameter ke master, data yang diterima seringkali masih merupakan data yang belum melalui proses kalibrasi. Oleh karena itu perlu adanya kalibrasi pada aplikasi sebelum ditampilkan. [13] Adapun script yang dipergunakan untuk proses kalibrasi tegangan, arus, daya dan faktor daya seperti berikut ini:

```

%InternalVariable_1.Phase_A=(%Modbus_1.PhaseA*65535)+(%Modbus_1.PhaseA_1)
%InternalVariable_1.Freq=(%Modbus_1.Freq*65535)+(%Modbus_1.Freq1)
%InternalVariable_1.Arus=(%Modbus_1.Arus*65535)+(%Modbus_1.Arus1)
%InternalVariable_1.Power=(%Modbus_1.Power*65535)+(%Modbus_1.Power1)
%InternalVariable_1.P_Factor=(%Modbus_1.P_Factor*65535)+(%Modbus_1.P_Factor1)

```

Gambar 10. Script Kalibrasi

Memperhatikan *script* diatas bahwa setiap parameter Modbus harus dikalibrasi dengan sebuah parameter yang nilainya 65535 agar sesuai dengan nilai yang

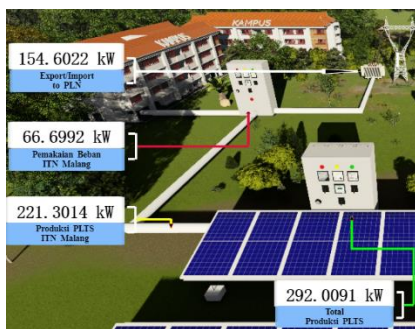
sebenarnya ketika dilakukan test secara manual.

Tahapan pendefinisian Modbus *variable* seperti pada Gambar 11 diperlukan untuk mengetahui posisi alamat data Modbus yang tersimpan dalam *register* sesuai petunjuk yang tercantum dalam panduan produk power meter Toky DS9L. Disamping itu juga perlunya format tipe data yang akan dibaca.

Variable nam	Register type	Address format	Register address	Address length	Data type
1 Arus	W	Decimal	4	2	Long integer
2 ArusB	W	Decimal	12	2	Long integer
3 ArusC	W	Decimal	20	2	Long integer
4 Freq	W	Decimal	2	2	Long integer
5 P_Factor	W	Decimal	8	2	Long integer
6 P_FactorB	W	Decimal	14	2	Long integer
7 P_FactorC	W	Decimal	22	2	Long integer
8 Phase_A	W	Decimal	0	2	Long integer
9 PhaseB	W	Decimal	10	2	Long integer
10 PhaseC	W	Decimal	18	2	Long integer
11 Power	W	Decimal	6	2	Long integer
12 Power_Active	W	Decimal	30	2	Long integer
13 PowerB	W	Decimal	16	2	Long integer
14 PowerC	W	Decimal	24	2	Long integer
15 Total_Energy	W	Decimal	28	2	Long integer
16 Total_pemakaia	W	Decimal	32	2	Long integer
17 Total_Power	W	Decimal	26	2	Long integer

Gambar 11. Mapping Modbus Internal Variabel

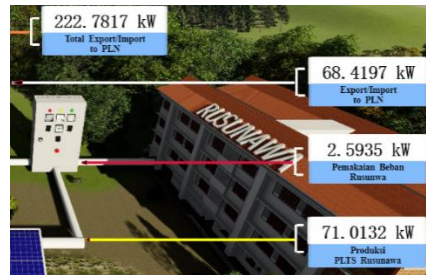
Dalam aplikasi Haiwell Cloud SCADA, selain mapping Modbus variable diperlukan juga pendefinisian internal variable sebagai kelanjutan dari Modbus variable yang sebelumnya telah dilakukan pendefinisannya. Hal itu ditunjukkan seperti pada Gambar 11.



Gambar 12. Dashboard Energi Listrik Sisi Kampus

Pada tampilan Gambar 12 menunjukkan bahwa total produksi energi listrik PLTS saat itu adalah 292,0091 kW

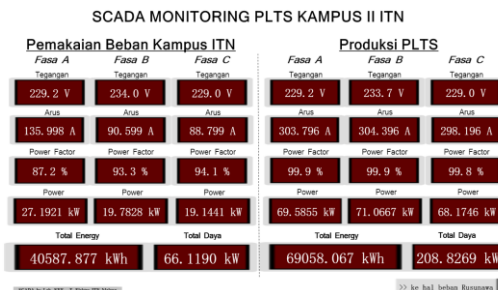
dimana sebagian terdistribusi ke jaringan listrik kampus sebesar 221,3014 kW. Di sisi lain pemakaian energi listrik pada semua gedung yang ada di area kampus menunjukkan daya sebesar 66,6992 kW. Sehingga dengan demikian sisa daya yang tidak terserap sebesar 154,602 kW.



Gambar 13. Dashboard Energi Listrik Sisi Rusunawa

Gambar 13 menampilkan data Modbus dari power meter pada gedung Rusunawa dimana daya listrik yang tersalurkan sebesar 71,0132 kW dari PLTS. Pada operasionalnya menyerap daya listrik sebesar 2,5935 kW. Sehingga masih terdapat sisa daya listrik sebesar 68,4197 kW.

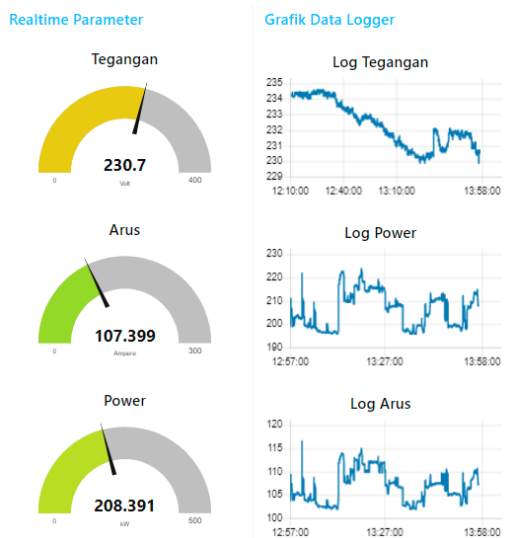
Kapasitas daya yang tersisa dari sisi kampus dan Rusunawa akan di gabungan untuk *export* ke jala-jala PLN sebesar 222,7817 kW. Dengan demikian dapat diketahui secara pasti jumlah daya yang di keluarkan setelah memenuhi kebutuhan internal kampus.



Gambar 14. Tampilan Keseluruhan Parameter

Pada penelitian ini, penulis juga melakukan implementasi menggunakan aplikasi Node Red untuk menampilkan parameter Modbus TCP/IP.[15] Node Red merupakan platform berbasis web, dimana pengguna dapat melakukan monitoring

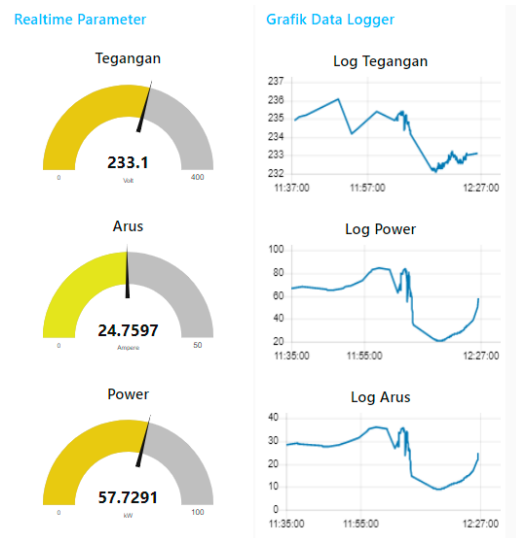
melalui browser. Berikut ini merupakan tampilan yang dihasilkan menggunakan Node Red:



Gambar 15 Node Red Interface Parameter Fasa A

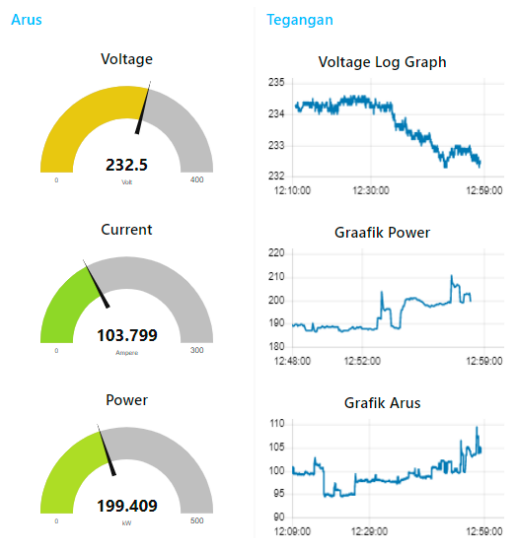
Gambar 15 menunjukkan tampilan parameter pada Fasa A yaitu tegangan, arus, daya dalam bentuk gauge dan tren grafik. Data log yang ditampilkan dapat menginformasikan parameter history dalam rentang waktu tertentu.

Data log yang ditampilkan dapat menginformasikan parameter history dalam rentang waktu tertentu.



Gambar 17. Node Red Interface Parameter Fasa C

Gambar 17 menunjukkan tampilan parameter pada Fasa A yaitu tegangan, arus, daya dalam bentuk gauge dan tren grafik. Data log yang ditampilkan dapat menginformasikan parameter history dalam rentang waktu tertentu.



Gambar 16. Node Red Interface Parameter Fasa B

Gambar 16 menunjukkan tampilan parameter pada Fasa B yaitu tegangan, arus, daya dalam bentuk gauge dan tren grafik.

4. KESIMPULAN

Berdasarkan dari hasil penelitian yang telah dilaksanakan menghasilkan akurasi data yang sangat baik. Hal ini dikarenakan parameter yang diperoleh telah memenuhi standar industri yaitu *protocol* Modbus, baik Modbus RTU maupun Modbus TCP. Sehingga parameter tersebut sudah tidak perlu lagi dilakukan kalibrasi dalam rangka membandingkan hasil pembacaan secara manual atau pengukuran langsung.

Dari hasil uraian serta pengujian terhadap penelitian ini maka dapat diambil beberapa point kesimpulan sebagai berikut:

1. Penggunaan perangkat sensor yang menggunakan standar protokol Modbus sangat mendukung dalam rangka mendapatkan hasil parameter yang sesuai dengan standar industri.

2. Penggunaan protokol Modbus sangat membantu dalam menurunkan biaya investasi setup karena secara topologi nya dapat dipasang secara paralel dengan memanfaatkan ID perangkat yang berbeda-beda antara perangkat satu dengan yang lainnya. Sehingga kemungkinan data tersebut akan tertukar akan sangat kecil sekali bahkan dapat dikatakan tidak terjadi.
3. Melalui sistem monitoring energi PLTS dapat memantau secara real time produksi energi yang dihasilkan maupun beban yang ada. Sehingga dapat dipergunakan untuk keperluan analisa lebih lanjut secara komprehensif.
4. Kecepatan komunikasi data Modbus TCP sangat tergantung pada kondisi traffic jaringan yang ada sehingga akan berpengaruh pada waktu transmisi data dari *slave* ke master Modbus.
5. Kalibrasi merupakan hal yang sangat penting dilakukan untuk menjamin akurasi data yang sesuai dengan nilai parameter.

5. SARAN

Jaringan komunikasi data Modbus dibangun secara khusus atau *private*. Jika kondisi topologi jaringan LAN tidak memungkinkan untuk dilakukan penambahan instalasi fisik maka dapat memanfaatkan teknologi *Virtual Private Network* (VPN) atau *Zero Trust Network Access* (ZTNA). Sehingga trafik paket data Modbus tidak terganggu dengan trafik paket data lainnya.

DAFTAR PUSTAKA

- [1] A. Salam and T. Sucita. (2012). *Rancang Bangun Sistem Jaringan Multidrop Menggunakan RS-485*. Vol 11, No.22, 1-11.
- [2] A. Zainuri, *Aplikasi Sistem Komunikasi Serial Multipoint RS-485 Pada Kontrol Crane Barang*. (2010). Universitas Brawijaya, Malang.
- [3] I. A. S. MODICON, Inc. (1996) *Modicon Modbus Protocol Reference Guide Modicon Modbus 91 Protocol Reference Guide*, Int. Bus.
- [4] J. K. R. Sastry, A. Suresh, S. J. S. Bhanu. (2015). *Building heterogeneous distributed embedded systems through rs485 communication protocol*. ARPN J. Eng. Appl. Sci., vol. 10, no. 16, 6793–6803.
- [5] L. Hui, Z. Hao, P. Daogang. (2012) *Design and Application of Communication Gateway of EPA and MODBUS on Electric Power System*, Energy Procedia. vol. 17, 286–292.
- [6] L. Xuan, L. Yongzhong. (2019) *Research and Implementation of Modbus TCP Security Enhancement Protocol*. J. Phys. Conf. Ser., vol. 1213, no. 5.
- [7] L. Zhao, R. Liang, and J. Zhang. (2014) *The Solving of Bias Resistor and Its Effect on the RS485 Fieldbus*, J. Adv. Computer Networks, vol. 2, no. 1, 71–75.
- [8] L. Zhao, R. Liang, J. Zhang. (2015) *Solving for the Best Value of Bias Resistor to Promote Stability of RS485 Fieldbus*, Int. J. Futur. Gener. Communication Network. vol. 8, no. 3, 89–96.
- [9] Mulyana Agus, Tosin. (2021) *Perancangan dan Implementasi Komunikasi RS-485 Menggunakan Protokol Modbus RTU dan Modbus TCP pada Sistem Pick-by-Light*, Komputika Jurnal Sistem Komputer, Vol. 10, 85-91.
- [10] Mamay Syani and Rizqi Mahestro Tresna. (2022). *Penerapan Network Access Control Autentikasi Internal Network Security Protokol 802.1 x*,

Vol. 16 No.2, Nuansa Informatika, 77-86.

- [11] Mamay Syani and Chika Santika Nurathilla. (2023) *Penerapan Mini Internet of Things (IoT) Board Berbasis Mikrokontroler Untuk Monitoring Kesehatan Lansia*, Vol 17 No.1, Nuansa Informatika, 67-74.
- [12] Muhammad Khaerudin and Andy Achmad. (2023). *Rancang Bangun Pintu Otomatis Berbasis Arduino (Studi Kasus TK. Bina Mulia)*, Vol 17 No.1, Nuansa Informatika, 111-118.
- [13] Nurpadmi, *Studi Tentang Modbus Protokol pada Sistem Kontrol*. (2010). Swara Patra, vol. 01, no. 2.
- [14] Qamara Zulham, Dirgantoro. (2018) *Implementasi Protokol Modbus Pada Power Meter SPM 91 Untuk Penerapan Monitoring Daya Listrik Rumah Tangga*, Vol. 5 No.3, *Proceeding of Engineering*.
- [15] T. Kugelstadt. (2008). *The RS-485 Design Guide Application Report The RS-485 Design Guide*, no. October, pp. 1-10.