

Implementasi Protokol Komunikasi Modbus Untuk Mini Scada Pada Plant Pengisian Serbuk Temulawak

Jorgi Fauzy Kusuma¹, Muhamad Rifa'i², Imam Saukani²

Politeknik Negeri Malang

Email: jorgifauzy09@gmail.com¹, muh.rifai@polinema.ac.id²,

imam.saukani@polinema.ac.id³

KEYWORDS

Modbus
Communication,
SCADA, RS485,
Haiwell Cloud
SCADA

ABSTRACT

In the modern industrial era industrial control systems have developed rapidly in terms of hardware equipment, software, and communication protocols. Industrial control systems where the role of humans has been largely shifted and replaced by automatic control systems. Where in an automatic control system, to send data from the field (sensor) to the controller, for communication between controllers, to send data from the controller to the computer, all require data communication and are connected to each other. This research was based on the problem, namely how to control and read all data information on the plant or ginger powder filling machine, such as sensor data information, actuator data, as well as data on the results of the processes carried out by the plant to be displayed and recorded. So a communication protocol is needed that can exchange information between master devices and slave devices, namely the Modbus communication protocol and a SCADA system needs to be implemented. Modbus communication implementation is implemented on hardware, namely RTU (remote terminal unit) which is made using an STM32 microcontroller, RS485 module and ENC28J60 module and MTU (master terminal unit) using a PC and the SCADA system is made using the Haiwell Cloud SCADA application. The results of performance testing of the mini SCADA which implements the Modbus communication protocol have succeeded in monitoring all sensor data and information originating from the plant, can control the plant via the HMI interface, and obtained a delivery latency delay of 342.5 ms, then the average delivery data loss is 342.5 ms. 0.625% in the transmission process of all data originating from the RTU to the MTU.

KATA KUNCI

Komunikasi
Modbus, SCADA,
RS485, Haiwell
Cloud SCADA

ABSTRAK

Dalam era industri modern sistem kontrol industri telah berkembang pesat dalam hal peralatan perangkat keras, perangkat lunak, dan protokol komunikasi. Sistem kontrol industri dimana peranan manusia telah banyak digeser dan digantikan oleh sistem kontrol otomatis. Dimana pada sistem kontrol otomatis, untuk mengirimkan data dari lapangan(sensor) ke kontroler, untuk komunikasi antar kontroler, untuk mengirimkan data dari

kontroller ke komputer, semua membutuhkan komunikasi data dan saling terhubung. Penelitian ini dibuat berdasarkan permasalahan yaitu bagaimana melakukan kontroling serta membaca semua informasi data pada plant atau mesin pengisian serbuk temulawak seperti informasi data sensor, data aktuator, maupun data hasil proses yang dilakukan oleh plant untuk ditampilkan dan dicatat. Sehingga diperlukan sebuah protokol komunikasi yang dapat melakukan pertukaran informasi antara perangkat master dan perangkat slave yaitu protokol komunikasi modbus serta perlu diterapkan sistem SCADA. Impelementasi komunikasi modbus diterapkan pada hardware, yaitu RTU(*remote terminal unit*) yang dibuat menggunakan mikrokontroller STM32, modul RS485 serta modul ENC28J60 dan MTU(*master terminal unit*) menggunakan PC dan sistem SCADA dibuat menggunakan aplikasi Haiwell Cloud SCADA. Hasil dari pengujian performa mini SCADA yang mengimplementasikan protokol komunikasi modbus telah berhasil melakukan monitoring semua data sensor serta informasi yang berasal dari plant, dapat melakukan pengontrolan plant melalui *interface* HMI, serta didapatkan delay latensi pengiriman sebesar 342.5 ms, kemudian rata-rata data loss pengiriman sebesar 0.625% dalam proses transmisi seluruh data yang berasal dari RTU menuju MTU.

PENDAHULUAN

Bidang komunikasi di lingkungan industri telah berkembang dan juga dikembangkan lebih lanjut dalam hal peralatan perangkat keras, perangkat lunak, dan protokol komunikasi (Purba, 2019). Standardisasi protokol telah diterapkan dan dioptimalkan untuk berbagai level atau level otomatisasi sistem (Setiawan, 2021). Setiap protokol memiliki karakteristik tertentu dan jika digunakan diluar batasnya, maka kinerjanya akan menurun bahkan berhenti bekerja. Protokol adalah aturan atau norma yang mengatur atau memungkinkan timbulnya hubungan komunikasi, dan transfer data antara dua atau lebih titik perangkat komputer (Cholik, 2021). Dalam era industri modern, sistem kontrol industri biasanya merujuk pada otomatisasi sistem kontrol yang digunakan (Khuluqi & Ratnanto Fitriadi, 2021). Sistem kontrol industri dimana peranan manusia telah banyak digeser dan digantikan oleh sistem kontrol otomatis. Dimana pada sistem kontrol otomatis, untuk mengirimkan data dari lapangan(sensor) ke kontroler, untuk komunikasi antar kontroler, untuk mengirimkan data dari kontroler ke komputer, semua membutuhkan komunikasi data.

Modbus adalah suatu protokol pada sistem komunikasi data yang merupakan standar untuk perangkat elektrik (Ariwibisono & Muljanto, 2023). Pada dasarnya protokol modbus sering digunakan dalam perangkat kendali (Sofyan & Fitriani, 2023). Untuk proses bertukar data, standar komunikasi serial RS-485 menggunakan Modbus RTU (Chadiq Zakaria, 2020). Adapun jenis Modbus terbaru sudah dapat berkomunikasi berbasis TCP atau biasa disebut Modbus TCP/IP (Sofyan & Fitriani, 2023). Fungsi Modbus yaitu sebagai penghubung antara komputer sebagai master dan sensor sebagai slave (Harjanto, 2020). Penelitian sebelumnya dilakukan oleh Muhammad Syariffudin, penelitian menjelaskan pembuatan mini SCADA menggunakan mikrokontroler keluarga AVR yang terdiri dari 2 mikrokontroler sebagai slave dan 1 mikrokontroler sebagai master dimana sistem menggunakan protokol komunikasi modbus RS485 sebagai pengumpul data kemudian data dikirim dan dilakukan monitoring pada komputer yang terintegerasi sistem monitoring dengan menggunakan aplikasi Visual Basic.

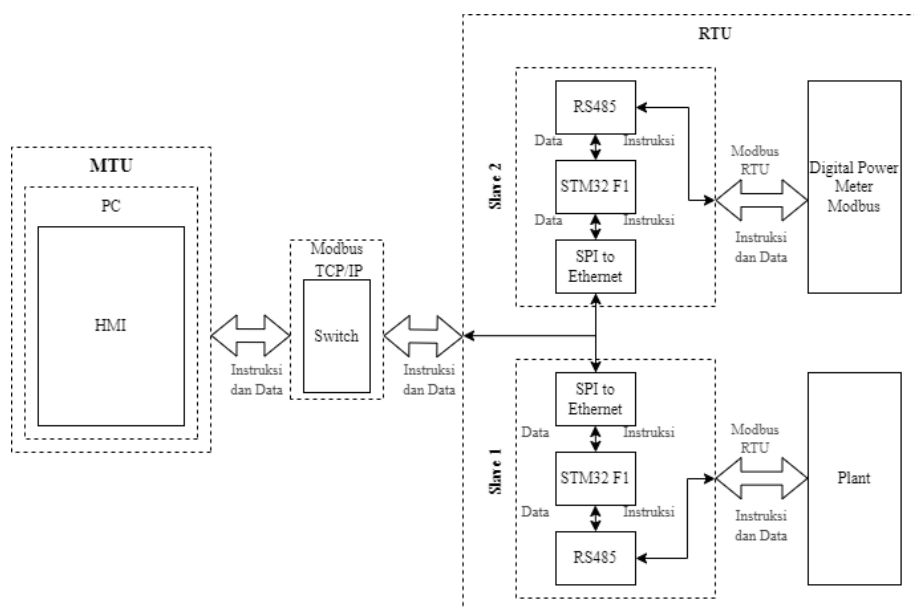
Pada penelitian ini bertujuan merancang sebuah protokol komunikasi yang memiliki standard industri yaitu modbus rs485 atau modbus RTU dan modbus TCP/IP menggunakan mikrokontroller untuk mengumpulkan dan mencatat semua informasi serta melakukan kontrolling pada plant yaitu mesin pengisian serbuk temulawak dengan dilengkapi sistem mini SCADA untuk proses pengawasan, dan pengendalian.

METODE PENELITIAN

Pada metode penelitian akan dijelaskan mengenai tahapan-tahapan pengerjaan meliputi perancangan mekanik, perancangan elektrik, perancangan software dan blok diagram.

1. Blok Diagram

Sistem terdiri dari 1 MTU dan 2 RTU yang di dalamnya terdapat mikrokontroller dan komponen lainnya, juga terdapat bagian yang dapat dikontrol oleh MTU. Berikut merupakan blok diagram sistem secara keseluruhan.



Gambar 1 Diagram Blok Sistem

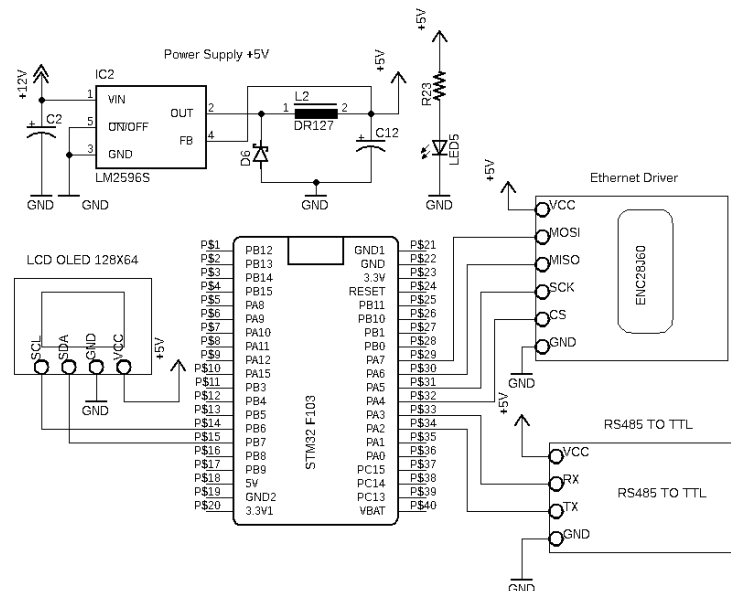
Gambar 1 merupakan sistem kerja dari implementasi perancangan protokol komunikasi modbus untuk pembuatan mini SCADA pada plant pengisian serbuk temulawak. Dalam blok diagram tersebut terdiri dari satu master terminal unit (MTU) yang menggunakan personal computer (PC) atau laptop dimana di dalamnya terdapat human machine interface (HMI) dan sistem supervisory control and data acquisition (SCADA) yang dibuat menggunakan software Haiwell SCADA. Dalam blok diagram tersebut terdapat remote terminal unit (RTU) yang merupakan sekelompok komponen yang berfungsi melakukan akuisi data yang berasal dari plant atau proses untuk dilakukan transfer ke master station (MTU). Komponen RTU pada sistem yang digunakan melibatkan 2 slave dimana setiap slave nya terdapat komponen STM32 F1 sebagai pusat pengendali atau pengolah data, SPI to Ethernet berfungsi untuk mengirimkan data melalui protokol TCP/IP ke master station dan RS485 berfungsi sebagai interface langsung dengan sensor dan aktuator yang terdapat pada plant. Pada slave 1 berfungsi untuk mengambil data variabel elektronik yang berasal dari pembacaan sensor-sensor serta melakukan pengontrolan aktuator yang terdapat pada plant, dimana data yang diakuisi di slave 1 yaitu data sensor loadcell berupa data berat, dan data kecepatan putar motor berupa data RPM. Kemudian pada slave 2 berfungsi untuk mengambil data power meter yang berasal dari

Implementasi Protokol Komunikasi Modbus Untuk Mini SCADA Pada Plant Pengisian Serbuk Temulawak

pembacaan digital power meter PZEM-016 yang digunakan untuk mengetahui parameter listrik yang dipakai di plant utama dengan menggunakan modbus RTU atau RS485, dimana data yang diakusisi di slave 2 yaitu data tegangan, arus, daya dan energi yang terpakai. Terdapat komponen Switch yang berfungsi menghubungkan slave 1 dan slave 2 ke master station (MTU) yang memungkinkan untuk melakukan pertukaran data melalui protokol komunikasi modbus TCP/IP. HMI berfungsi sebagai master yang berarti sebagai pusat pengontrolan terhadap sistem yang dilengkapi dengan sistem SCADA, master akan melakukan polling data terhadap slave – slave yaitu mikrokontroler yang terdapat RTU untuk melakukan permintaan data pengukuran berbagai sensor yang selanjutnya data akan kembali dikirimkan ke MTU dan ditampilkan ke HMI dalam bentuk informasi data yang dikirim dari RTU. Kemudian dalam proses pengontrolan HMI akan memberi perintah untuk RTU, kemudian mikrokontroler pada RTU akan mengeksekusi sesuai permintaan dari HMI.

2. Perancangan Elektrik

Dalam perancangan elektrik ini terdiri dari perancangan power supply, perancangan driver ethernet, perancangan RS485, dan perancangan LCD Oled 128x64



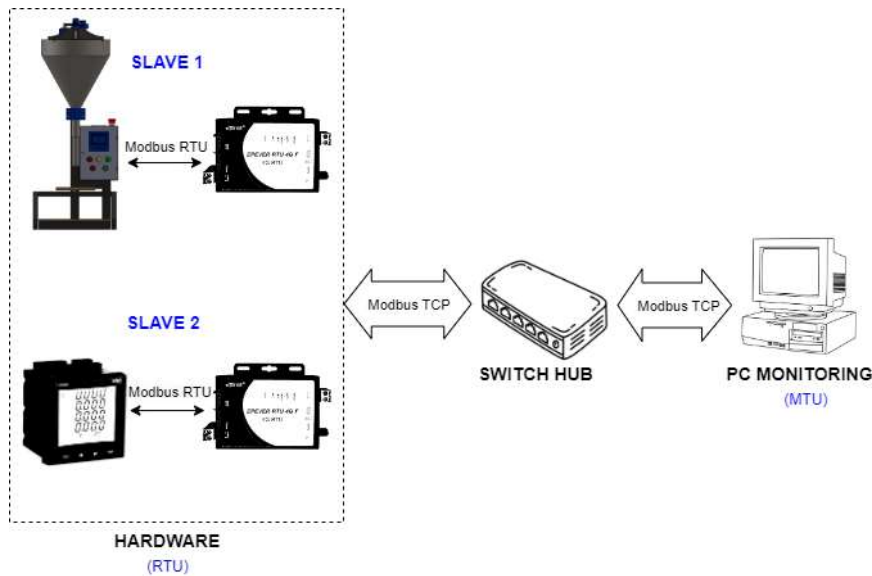
Gambar 2 Rangkaian Elektrik RTU(remote terminal unit)

Gambar 2 Power supply digunakan sebagai catu daya sistem, dengan menggunakan catu daya sebesar 5V untuk sistem keseluruhan yaitu untuk mikrokontroler, RS485, ENC28J60 serta LCD Oled. Dalam perancangan power supply, jenis power supply yang digunakan adalah power supply switching atau buck konverter dengan menggunakan jenis IC LM2596. Spesifikasi power supply yaitu untuk tegangan input sebesar 12V dari power adaptor, output tegangan output yaitu sebesar 5V dan arus keluaran sebesar 3A max. Perancangan rangkaian rs485 digunakan sebagai *hardware* komunikasi untuk protokol komunikasi modbus RTU atau rs485. Pada Gambar 2 pin yang digunakan untuk berkomunikasi dengan rs485 yaitu pin PA2 dari STM32 yang terhubung dengan pin TX rs485, pin PA3 terhubung dengan pin RX rs485, pin VCC rs485 terhubung dengan 5V dan pin GND terhubung dengan ground. Pada perancangan rangkaian ethernet ini berfungsi sebagai *hardware* komunikasi untuk protokol modbus TCP/IP. Pada Gambar 2 pin yang digunakan driver ethernet yaitu CS terhubung dengan PA4, SCK terhubung dengan PA5, MISO terhubung dengan PA6, dan MOSI terhubung dengan PA7. Pada perancangan LCD berfungsi sebagai penampil data sensor PZEM-016 untuk menampilkan parameter listrik. LCD dihubungkan dengan mikrokontroler STM32 dengan

menggunakan komunikasi I2C dimana pin yang digunakan yaitu pin PB6 terhubung dengan pin SCL LCD, pin PB7 STM32 terhubung dengan pin SDA LCD serta catu daya +5V.

3. Perancangan Komunikasi

Jenis protokol komunikasi yang digunakan pada penelitian ini adalah modbus RTU (RS485) dan modbus TCP/IP dengan media perantara kabel LAN dan switch hub.



Gambar 3 Perancangan Sistem Komunikasi

Gambar 3 merupakan sebuah hubungan antara *hardware* yang dapat diintegrasikan dengan sistem SCADA yaitu saling berkirir dan bertukar data maka komunikasi yang digunakan yaitu jenis protokol komunikasi modbus, dalam penelitian ini menggunakan jenis protokol komunikasi modbus RTU yang digunakan berkomunikasi antara perangkat yang berada di lapangan pada Gambar 15 yaitu antara *slave 1* dengan plant dan *slave 2* dengan sensor PZEM-016 atau digital power meter, kemudian di setiap perangkat RTU baik *slave 1* maupun *slave 2* dilengkapi dengan *hardware* komunikasi yang dapat digunakan untuk mentransmisikan data melalui metode TCP/IP atau dalam penelitian ini menggunakan protokol komunikasi modbus TCP/IP dengan perantara kabel UTP (*unshield twisted pair*).

4. Perancangan Sistem SCADA

Terdapat beberapa bagian dalam proses pembuatan sistem berbasis SCADA yaitu alokasi memori register, tampilan HMI untuk MTU.

a. Alokasi Memori Register

Gambar 4 Alokasi memori digunakan untuk menyimpan semua data yang berasal dari RTU *slave 1* atau yang berasal dari plant, dan untuk jenis register type, register address, tipe data dan aksi sehingga semua alokasi memori yang digunakan pada rtu *slave 1* harus sama dengan alokasi memori pada sistem SCADA.

Implementasi Protokol Komunikasi Modbus Untuk Mini SCADA Pada Plant Pengisian Serbuk Temulawak

Variable name	Register type	Address format	Register address	Bit address	Address length	Data type	The mode of reading and writing
1 HASL	4X(Analog output,Function code 03 06)	Decimal	0			1 Integer	Read and write
2 RPM	4X(Analog output,Function code 03 06)	Decimal	1			1 Integer	Read and write
3 PEDAL	4X(Analog output,Function code 03 06)	Decimal	2	0		1 Bool	Read and write
4 SP_1	4X(Analog output,Function code 03 06)	Decimal	4			1 Integer	Read and write
5 SP_2	4X(Analog output,Function code 03 06)	Decimal	5			1 Integer	Read and write
6 SP_DISPLAY	4X(Analog output,Function code 03 06)	Decimal	6			1 Integer	Read and write
7 CLEANING	4X(Analog output,Function code 03 06)	Decimal	3	1		1 Bool	Read and write
8 EMERGENCY	4X(Analog output,Function code 03 06)	Decimal	7	2		1 Bool	Read and write
9 SP_ANY	4X(Analog output,Function code 03 16)	Decimal	8			1 Integer	Read and write
10 INDIKATOR_MOTOR	4X(Analog output,Function code 03 06)	Decimal	9	0		1 Bool	Read and write
11 WRITE_PEDAL_HMI	4X(Analog output,Function code 03 06)	Decimal	10	0		1 Bool	Read and write
12 TOTAL_100	4X(Analog output,Function code 03 06)	Decimal	11			1 Integer	Read and write
13 TOTAL_250	4X(Analog output,Function code 03 06)	Decimal	12			1 Integer	Read and write

Gambar 4 Alokasi Memori SCADA RTU

Variable name	Register type	Address format	Register address	Address length	Data type	The mode of reading and writing
1 TEGANGAN	4X(Analog output,Function code 03 06)	Decimal	10		1 Integer	Read and write
2 ARUS	4X(Analog output,Function code 03 06)	Decimal	11		1 Integer	Read and write
3 DAYA	4X(Analog output,Function code 03 06)	Decimal	13		1 Integer	Read and write
4 ENERGI	4X(Analog output,Function code 03 06)	Decimal	15		1 Integer	Read and write
5 FREKUENSI	4X(Analog output,Function code 03 06)	Decimal	17		1 Integer	Read and write

Gambar 5 Alokasi Memori SCADA RTU

b. Tampilan HMI SCADA

Dalam pembuatan sistem SCADA selalu melibatkan HMI (*human machine interface*) yang berfungsi sebagai antarmuka yang terjadi antara manusia dengan mesin yang digunakan dalam proses yang berkaitan dengan otomasi. Pada sistem SCADA yang dibuat tampilan HMI dibuat untuk menjembatani antara pengguna atau manusia untuk mengawasi serta pengendalian terhadap plant yaitu mesin pengisian temulawak. Dari sistem SCADA yang dirancang semua data yang berasal dari RTU akan dibaca dan ditampilkan pada tampilan HMI untuk proses pengawasan dan juga analisa sehingga memudahkan operator dalam proses pengawasan dan pengendalian. Gambar 6 merupakan tampilan HMI yang digunakan pada penelitian ini.

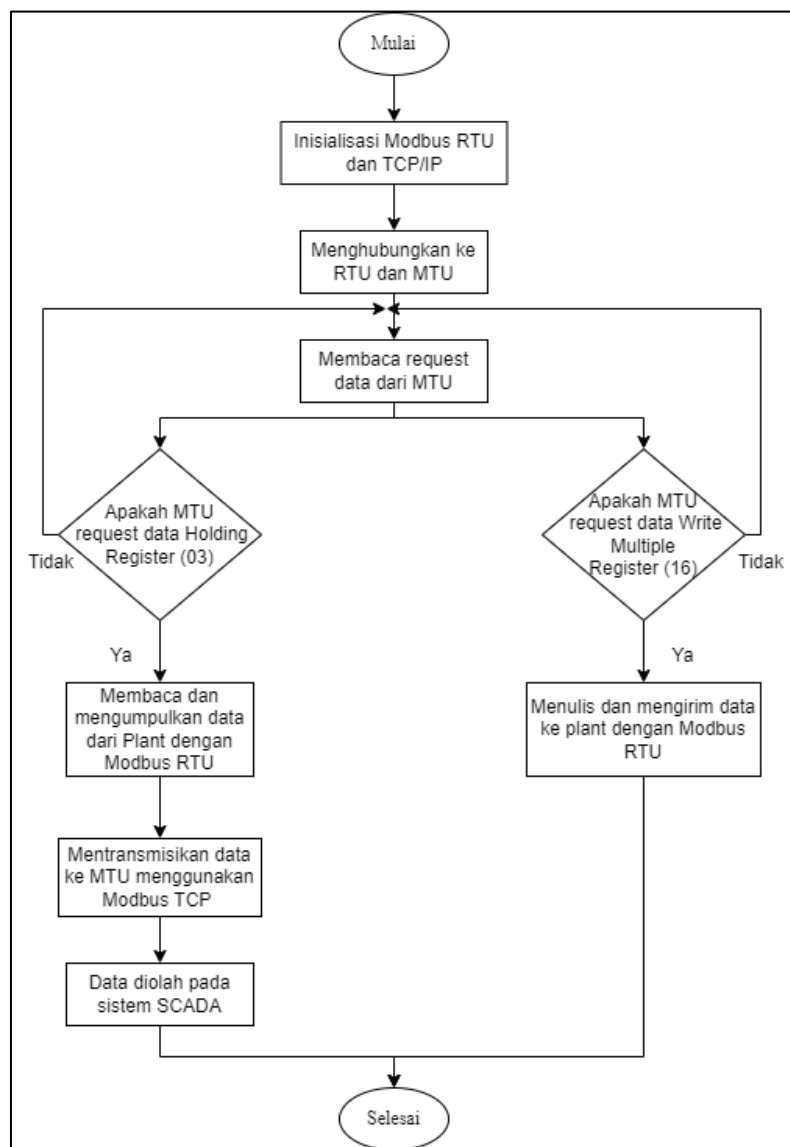


Gambar 6 Tampilan HMI SCADA

5. Flowchart

Perancangan software bertujuan untuk membuat flowchart jalannya alat yang telah dibuat. Adapun Deskripsi Flowchart Sistem pada Gambar 7 yaitu:

1. Mulai, mempersiapkan alat yang sudah dirancang
2. Inisialisasi input/output, dan komunikasi modbus
3. Membaca *request* data yang berasal dari MTU(*master terminal unit*)
4. Terdapat 2 jenis function code yang di *request* MTU, holding register(03) dan write multiple register(16)
5. Jika request berupa holding register maka, RTU akan memberikan atau membaca data yang berasal dari plant berupa informasi sensor dan aktuator menggunakan komunikasi modbus RTU , kemudian data di transmisikan kembali ke MTU menggunakan modbus TCP/IP dan data ditampilkan pada sistem SCADA, jika tidak maka sistem akan kembali melakukan scanning ..
6. Jika request berupa write multiple register maka, RTU akan menerima data berasal dari MTU berupa perintah melakukan aksi terhadap plant, kemudian plant akan melakukan aksi sesuai perintah yang diberikan MTU, jika tidak maka sistem akan kembali melakukan scanning .
7. Selesai.



Gambar 7 Flowchart Program

HASIL DAN PEMBAHASAN

Setelah dilakukan perencanaan dan pembuatan langkah selanjutnya yaitu melakukan pengujian. Pengujian dilakukan bertujuan untuk mengetahui apakah sistem tersebut sudah bekerja sesuai dengan apa yang diharapkan atau tidak, apabila masih belum memenuhi tujuan yang diharapkan maka dibahas apa saja yang menjadi penyebab dari sistem belum bekerja sesuai dengan apa yang diharapkan.

1. Pengujian Komunikasi Modbus RTU

Tujuan pengujian komunikasi modbus RTU adalah menguji modul RTU yang telah dibuat untuk melakukan komunikasi menggunakan protokol modbus untuk melakukan pengiriman data dan penerimaan data apakah bekerja dengan baik dan sesuai perancangan. Tabel 1, Tabel 2, dan Tabel 3 merupakan tabel hasil pengujian Komunikasi Modbus.

Tabel 1. Pengujian Komunikasi 1

<i>Frame Data Request</i>	<i>Frame Data Response</i>	<i>Nilai Data</i>
01 10 00 00 00 01 67 90	01 10 02 00 01 7D 00	1
01 10 00 00 00 02 27 91	01 10 02 00 02 3D 01	2
01 10 00 00 00 03 E6 51	01 10 02 00 03 FC C1	3
01 10 00 00 00 04 A7 93	01 10 02 00 04 BD 03	4
01 10 00 00 00 05 66 53	01 10 02 00 05 7C C3	5

Tabel 2. Pengujian Komunikasi 2

<i>Frame Data Request</i>	<i>Frame Data Response</i>	<i>Nilai Data</i>
01 03 00 00 00 01 84 0A	01 03 02 00 64 B9 AF	100
01 03 00 00 00 01 84 0A	01 03 02 00 C8 B9 D2	200
01 03 00 00 00 01 84 0A	01 03 02 01 2C B8 09	300
01 03 00 00 00 01 84 0A	01 03 02 01 90 B9 B8	400
01 03 00 00 00 01 84 0A	01 03 02 01 F4 B8 53	500

Tabel 3. Pengujian Komunikasi 3

<i>Frame Data Request</i>	<i>Frame Data Response</i>	<i>Nilai Data</i>
01 04 00 00 00 01 31 CA	01 04 02 00 0A 39 37	10
01 04 00 00 00 01 31 CA	01 04 02 00 14 B9 3F	20
01 04 00 00 00 01 31 CA	01 04 02 00 1E 39 38	30
01 04 00 00 00 01 31 CA	01 04 02 00 28 B9 2E	40
01 04 00 00 00 01 31 CA	01 04 02 00 32 38 E5	50

Pengujian dilakukan terhadap masing-masing *function code* yaitu dengan melakukan *request* data terhadap perangkat *slave*, dimana setiap *function code*, *request* data dilakukan sebanyak 5 kali dan juga mendapatkan *response* data sebanyak 5 kali. Dari ketiga tabel hasil, setiap tabel memiliki *message request* dan *message respon*, dimana masing-masing message memiliki modbus data frame masing-masing. Dari hasil pengujian dari Tabel 1, Tabel 2 dan Tabel 3 memiliki *frame message request* yang sama. Contoh pada Tabel 1 nomor 1 master

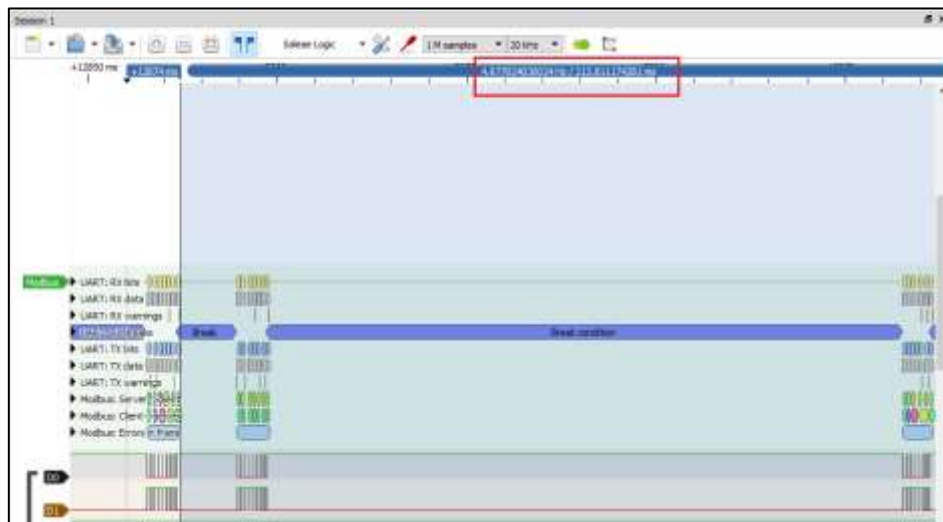
Implementasi Protokol Komunikasi Modbus Untuk Mini SCADA Pada Plant Pengisian Serbuk Temulawak

melakukan request data terhadap *slave* dengan mengirimkan data **01 03 00 00 00 01 84 0A** (heksadesimal) artinya :

- **01** : merupakan alamat *slave*
- **03** : merupakan *function code* untuk holding register
- **00 00** : merupakan alamat register yang diambil nilainya yaitu 40001(holding register)
- **00 01** : merupakan jumlah register yang diminta yaitu sebanyak 1 register
- **84 0A** : merupakan nilai CRC yang digunakan sebagai metode pengecekan *error* data

2. Pengujian Latensi Komunikasi

Tujuan dari pengujian latensi komunikasi yaitu untuk mengetahui *delay* atau waktu yang dibutuhkan perangkat pengirim pada saat proses pengiriman data menuju ke perangkat penerima.



Gambar 8 Waktu Data Transfer Modbus RTU

Dari hasil pengujian untuk mengetahui *delay* pengiriman data yang terjadi pada komunikasi modbus RTU yaitu dengan cara menggunakan logic analyzer yaitu dengan melihat periode pulsa atau *timing pulse* yaitu dengan cara menghubungkan pin A modul RTU dan pin B modul RTU pada probe logic analyzer. Kemudian untuk melihat hasil gelombang atau pulsa yaitu dengan menggunakan aplikasi desktop sigrok. Dapat dilihat hasil pengukuran *delay* atau *latency* data seperti pada Gambar 8 dimana hasil yang terukur yaitu sebesar 213,8 ms dimana pengukuran dihitung sejak perangkat master atau modul RTU mulai melakukan pengiriman data *request* ke *slave*, dan kemudian *slave* mengirimkan data *response* ke *master*. Pada penelitian yang sudah ada (Bonanza Yoma Pratama, dan Efi anisa, 2023) modbus RTU dan ASCII memiliki *delay* ± 1 Detik dengan komunikasi serial.



Gambar 9 Waktu Data Transfer Modbus TCP/IP

Kemudian, hasil pengujian untuk mengetahui *delay* pengiriman data yang terjadi pada komunikasi modbus TCP/IP yaitu dengan menggunakan aplikasi desktop wireshark. Pada pengujian ini MTU atau laptop harus terhubung dengan modul RTU menggunakan kabel UTP/LAN dengan protokol ethernet atau TCP/IP seperti Gambar 9 merupakan hasil pengujian yang dilakukan. Pengujian dilakukan dengan mengambil sampel data sebanyak 1000 data kemudian untuk mencari *delay* yaitu dengan menggunakan rumus Persamaan (1):

$$delay = data\ time\ 2 - data\ time\ 1 \quad (1)$$

$$rata - rata\ delay = \frac{\sum_{Delay=0}^{1000} Delay}{jumlah\ data(1000)} \quad 1 \quad (2)$$

$$rata - rata\ delay = \frac{127,413}{1000}$$

$$rata - rata\ delay = 0,129\ s$$

Untuk mencari *delay* pengiriman data pertama yaitu harus menghitung interval pengiriman data antara waktu pengiriman data kedua dikurangi data pertama, kemudian setelah semua data diketahui interval atau nilai *delay* per data kemudian dilakukan pencarian rata-rata *delay* yaitu dengan menjumlahkan semua data yang telah dihitung intervalnya yaitu kemudian dibagi jumlah data yaitu 1000 data sehingga dapat diketahui hasil *delay* data untuk pengiriman yaitu sebesar 0,129 s. Sehingga untuk menghitung waktu yang diperlukan untuk melakukan pengiriman data dari MTU ke RTU yaitu dengan menjumlahkan *delay* data yang terjadi pada pengiriman modbus TCP/IP ditambah *delay* data yang terjadi pada modbus RTU yaitu sebesar:

$$total\ delay\ data = delay\ MTU + delay\ RTU \quad (3)$$

$$total\ delay\ data = 213,3\ ms + 0,129\ s$$

$$total\ delay\ data = 342.8\ ms$$

Sehingga total *delay* yaitu sebesar 342.8 ms. Adanya *delay* terjadi karena data yang dikirimkan membutuhkan tahapan untuk memproses sinyal menjadi data digital (waktu *preprocessing*), waktu tunggu, waktu untuk melakukan pengiriman data dari satu komputer ke komputer lain, dan waktu *post-processing*.

3. Pengujian Keseluruhan

Tujuan pengujian keseluruhan sistem yaitu untuk mengetahui kinerja seluruh sistem mulai dari *hardware*, sistem komunikasi data dan sistem SCADA apakah bekerja sesuai dengan perancangan dan dapat berjalan dengan baik. Tabel 12 dan Tabel 13 merupakan tabel hasil pengujian keseluruhan sistem.

Tabel 4. Pengujian Peforma Sistem SCADA

Pengujian Ke-	Jarak Pengujian	Jumlah Paket Data	Delay (ms)	Packet Loss(%)	Keterangan
1	1 meter	528 bit	342.5	1.4	Pengujian keseluruhan urutan proses plant dari awal sampai akhir
2	2 meter	528 bit	341.9	0	
3	3 meter	528 bit	342	0.9	
4	4 meter	528 bit	342.4	0	
5	5 meter	528 bit	343.1	2.7	
6	6 meter	528 bit	342.4	0	
7	7 meter	528 bit	342.6	0	
8	8 meter	528 bit	343.1	0	
Rata-rata delay transfer data					342.5 ms
Rata rata paket data loss(%)					0.625 %

Tabel 5. Pengujian Penggunaan Daya Listrik

No	Berat Temulawak	Parameter	Sensor	Alat ukur	Error(%)
1	2 Kg	Tegangan (V)	210.5	209.5	0.004773
		Arus (A)	1.72	1.6	0.075
		Daya (W)	362.06	335.2	0.080131
2	1.5 Kg	Tegangan (V)	214.3	212.8	0.007049
		Arus (A)	1.1	1.02	0.078431
		Daya (W)	235.73	217.056	0.086033
3	0.5 Kg	Tegangan (V)	211	209.3	0.008122
		Arus (A)	0.47	0.4	0.175
		Daya (W)	99.17	83.72	0.184544
Rata-Rata Error(%)				Tegangan (V)	0.006648
				Arus (A)	0.109477
				Daya (W)	0.116903

Dari hasil pengujian keseluruhan, dilakukan pengujian terhadap dua paramter yaitu pengujian peforma SCADA dan pengujian penggunaan daya listrik pada plant. Pada pengujian peforma SCADA dilakukan dengan cara melakukan pengujian *delay* pengiriman atau latensi yang terjadi pada saat pengiriman data yang berasal dari RTU atau *field device* dengan maksud menguji seberapa cepat atau *realtime* sistem SCADA yang telah dibuat dengan memanfaatkan sistem komunikasi modbus. Pada Tabel 4 didapatkan empat parameter penting yaitu jarak

Implementasi Protokol Komunikasi Modbus Untuk Mini SCADA Pada Plant Pengisian Serbuk Temulawak

komunikasi dari field device menuju MTU atau PC kemudian jumlah data yang ditransfer, *delay* pengiriman, dan jumlah paket data yang *loss* atau yang mengalami kegagalan. Untuk jarak komunikasi menggunakan 8 sampel jarak yaitu mulai dari 1 meter hingga 8 meter, kemudian jumlah frame data yang ditransfer yaitu sebesar 528 bit atau 66 byte data. Kemudian untuk menghitung *delay* data menggunakan persamaan (3), kemudian untuk menghitung paket data yang *loss* atau gagal yaitu dengan menggunakan persamaan (4).

$$\text{paket data loss} = \frac{(\text{data yang dikirim} - \text{data yang diterima})}{\text{data yang dikirim}} \times 100\% \quad (4)$$

Delay rata-rata yang dihasilkan yaitu sebesar 342.5 ms, Kemudian paket *loss* data yang dihasilkan rata-rata yaitu sebesar 0.6 yang artinya memiliki nilai sangat kecil hal tersebut diakibatkan karena posisi kabel pada saat pengujian terdapat kesalahan peletakan instalasi atau terdapat *noise* pada saat pengiriman data.

Kemudian pengujian penggunaan daya listrik yang digunakan oleh plant untuk mengetahui konsumsi daya listrik yang digunakan selama proses dari awal sampai akhir. Pengujian dilakukan dengan melakukan pengisian serbuk temulawak menggunakan mesin pengisian serbuk temulawak. Berat temulawak atau isi hooper yaitu sebesar 2kg dengan 3 sampel pengambilan yaitu pada berat 2 kg, 1.5 kg, dan 0.5 kg, dari masing-masing sampel dilakukan pengukuran konsumsi daya listrik dengan hasil seperti pada tabel 4.10 Dari hasil pengukuran didapatkan nilai ukur dari tiga sampel yang berasal dari pembacaan alat ukur dan pembacaan sensor. Pada berat temulawak 2 kg menghasilkan konsumsi daya sebesar 362 Watt, pada berat 1.5 kg menghasilkan 235 Watt, dan pada 0.5 kg menghasilkan daya sebesar 99 Watt. Sehingga bisa disimpulkan bahwa semakin berkurang berat dari serbuk temulawak yang terdapat pada hooper mesin maka konsumsi daya pada saat proses pengisian juga akan semakin berkurang.

KESIMPULAN

Setelah dilakukan perancangan pemodelan sistem dan juga pengujian pemodelan sistem didapatkan beberapa kesimpulan diantaranya adalah sebagai berikut: Desain perangkat RTU dibuat dengan menggunakan mikrokontroler STM32F1 Bluepill sebagai pusat pengendali kemudian dilengkapi dengan modul RS485 dan ENC28J60 sebagai hardware komunikasi untuk modbus RTU dan modbus TCP/IP. Untuk bisa mendesain mini SCADA maka diperlukan sistem utama penyusunnya yaitu perangkat RTU (*remote terminal unit*), *Field device* dimana pada penelitian ini menggunakan mesin serbuk temulawak, MTU (*master terminal unit*) yaitu menggunakan sebuah laptop yang telah didesain untuk sistem SCADA dengan tampilan HMI sebagai *user interface*, Komunikasi data dimana pada penelitian ini menggunakan komunikasi modbus RTU dan modbus TCP/IP sebagai protokol untuk mentransmisikan data. Dalam mendesain atau merancang mini SCADA diperlukan sebuah *software* Haiwell Cloud SCADA. Dengan menggunakan protokol komunikasi modbus maka sistem SCADA yang telah berhasil melakukan *Telemetry* atau Pengiriman hasil pengukuran besaran tenaga listrik, *Telesignaling* atau dapat melakukan pengiriman sinyal indikasi atau berupa status, kemudian *Telecontrolling* dapat melakukan pengendalian jarak jauh. Dalam melakukan transmisi data protokol komunikasi memiliki *delay* pengiriman data dimana ketika MTU melakukan *request* data sampai RTU merespon dengan mengirimkan data yang berasal dari plant mengalami *delay* atau *latency* data sebesar 342.5 ms dan untuk rata-rata data *loss* sebesar 0.625.

DAFTAR PUSTAKA

- Ariwibisono, Fransiscus Xaverius, & Muljanto, Widodo Pudji. (2023). Implementasi Sistem Monitoring Produksi Energi PLTS Berbasis Protokol Modbus RTU Dan Modbus TCP. *NUANSA INFORMATIKA*, 17(2), 109–118.
- Chadiq Zakaria, Muchamad. (2020). SISTEM MONITORING INSTRUMENT AIR COMPRESSOR (IAC) BERBASIS SCADA DENGAN KOMUNIKASI MODBUS RTU RS485 DI PLTU PACITAN. Universitas Muhammadiyah Ponorogo.
- Cholik, Cecep Abdul. (2021). Perkembangan Teknologi Informasi Komunikasi/ICT dalam Berbagai Bidang. *Jurnal Fakultas Teknik Kuningan*, 2(2), 39–46.
- Harjanto, Imadudin. (2020). IoT Gateway Menggunakan Protokol MQTT pada Perangkat Kendali Berbasis Modbus-RTU. *JITEK (Jurnal Ilmiah Teknosains)*, 6(1), 12–19.
- Khuluqi, Muh Khassin, & Ratnanto Fitriadi, S. T. (2021). Perancangan dan Pembuatan Trainer-Kit Programmable Logic Control (PLC) untuk Media Pembelajaran Sistem Otomasi Industri (Studi Kasus: Jurusan Teknik Industri UMS). Universitas Muhammadiyah Surakarta.
- Purba, Minda Mora. (2019). Pemanfaatan Teknologi Informasi Dalam Bidang Industri Otomotif. *JSI (Jurnal Sistem Informasi) Universitas Suryadarma*, 6(1), 160–170.
- Setiawan, Pandu. (2021). Otomasi konfigurasi sistem informasi berdasarkan aliran data. Universitas Islam Negeri Maulana Malik Ibrahim.
- Sofyan, Yusuf, & Fitriani, Sofy. (2023). Rancang Bangun Konverter Modbus RTU RS485 ke Modbus TCP/IP Berbasis ATMEGA2560. *Journal of Computer System and Informatics (JoSYC)*, 4(3), 470–477.
- Andrian, Mohammad, Ari Kurniawan, dan Imam Saukani. (2024). Sistem Kendali Suhu Menggunakan Metode PID dalam Proses Deasetilasi Kitin. *TEKTONIK: Jurnal Ilmu Teknik 1.2*: 131-137.
- Ardiansyah, Andri, dan Priatmadja, Rizally. (2015). Rancang Bangun Protokol Modbus Pada KWH Meter Elektronik Tipe ION 8600 Untuk Memonitoring Besaran Energi Listrik Trafo Dengan Menggunakan Aplikasi Citect SCADA. *Jurnal Teknologi Elektro, Universitas Mercu Buana*, 6(1), 38-54.
- Simanullang, Sandro Andre, Santi, Rudati Paula, dan Feriyonika. (2017). Sistem PID Pengendali Level Ketinggian Air Berbasis Modbus/TCP – LCU dan Industrial Field Control Node – RTU. *Industrial Research Workshop and National Seminar*.
- Sofyan, Yusuf dan Fitriani, Sofi. (2023). Rancang Bangun Konverter Modbus RTU RS485 ke Modbus TCP/IP Berbasis ATMEGA2560. *Journal of Computer System and Informatics (JoSYC)*, vol. 4.
- Suwarno, Djoko Untoro dan Erikson. (2021). Sistem Monitoring Untuk Berbagai Variabel Elektronis Menggunakan Protokol Modbus Dan Komunikasi RS485. *Prosiding Seminar Nasional Riset dan Teknologi Terapan (RITEKTRA)*.
- Syariffudin, Muhammad dan Kusriyanto, Medilla. *Mini SCADA Berbasis Mikrokontroler ATMEGA 32 Dengan Komunikasi Modbus RS485 Dan Sistem Monitoring Menggunakan Visual Basic. Seminar Nasional*.
- Tiyono, Agus, Sudjadi, dan Setiawan, Iwan. *Sistem Telekontrol SCADA Dengan Fungsi Dasar Modbus Menggunakan Mikrokontroler AT89S51 Dan Komunikasi Serial RS485*.