Modul Komunikasi Modbus RTU *over* RS485 Berbasis Arduino

Indra H Mulyadi[1*], Rahmi Mahdaliza[1], Aditya Gautama[1], Senanjung Prayoga[1], Kamarudin[1]

[1]Politeknik Negeri Batam, Batam, Indonesia

[*]Email: indra@polibatam.ac.id

***Abstract*—Modul akuisisi data dengan menggunakan Modbus *Remote Terminal Unit* (RTU) telah banyak digunakan secara komersial di industri. Namun sayangnya, harga peralatan tersebut tergolong mahal. Protokol komunikasi Modbus RTU ini cukup kompleks untuk dibuat oleh pengguna awam. Dengan latar belakang tersebut, penelitian ini bertujuan untuk membuat modul komunikasi Modbus RTU *over* RS485 yang berbasiskan Arduino sehingga harga pembuatannya menjadi murah dan dapat menjadi sarana pembelajaran di lembaga pendidikan dan pelatihan. Modul yang dibuat ini terdiri atas dua *input* analog dan dua *input*/*output* digital. Sinyal analog yang masuk dikondisikan terlebih dahulu dengan menggunakan *signal conditioner* sebelum diproses lebih lanjut oleh mikrokontroler. Hasil pengujian menggunakan *Modbus Analyzer* menunjukkan bahwa komunikasi Modbus RTU *over* RS485 pada modul yang telah dibuat ini dapat berjalan sesuai dengan *data frame* yang didesain.**

Kata kunci: Modbus RTU, RS485, Arduino

I. PENDAHULUAN

DALAM sebuah *plant* kontrol proses, pengukuran dilakukan untuk memonitor beberapa besaran, seperti tekanan, level, aliran, dan suhu. Proses pengukuran besaran-besaran tersebut merupakan proses akuisisi data. Akuisisi data yang dilakukan dalam lingkunan industri tentunya tidak sederhana, mengingat data yang diakuisisi cukup banyak dan jarak antara sensor yang satu dan lainnya berjauhan.

Salah satu sistem monitoring yang digunakan di industri adalah *Supervisory Control and Data Acquisition* (SCADA) [1]–[3]. Membangun sistem SCADA berarti membangun jaringan komunikasi dan *Human Machine Interface* (HMI) [4]. Data dari sensor pada sebuah *plant* diakuisisi dan dikirimkan ke komputer menggunakan modul komunikasi, baik menggunakan kabel maupun nirkabel.

Peralatan-peralatan yang ada dalam sebuah sistem SCADA seperti *Programmable Logic Controller* (PLC), HMI, *control panel*, *driver*, kendali motor, dan *Input/Output* (I/O) mengunakan protokol komunikasi dalam untuk pengiriman data. Ada beberapa jenis protokol komunikasi untuk menghubungkan peralatan yang ada di dalam sebuah sistem SCADA, seperti Modbus, RP-570, Profibus, dan Conitel [5]. Protokol yang digunakan dalam penelitian ini adalah Modbus, sebuah protokol yang banyak digunakan pada instrumentasi di industri. Ada beberapa jenis Modbus yang dipakai di industri, antara lain Modbus RTU, Modbus ASCII, Modbus TCP/IP, Modbus UDP/IP, Pemex Modbus, Daniel Modbus, dan Enron Modbus. Modbus yang digunakan pada penelitian ini adalah Modbus RTU.

Pada OSI *layer*, Modbus [6] merupakan protokol komunikasi yang terletak pada *application layer*, yakni pada level 7. *Layer* ini yang menyediakan komunikasi *client*/*server* antar peralatan yang saling terkoneksi dengan tipe *bus* dan jaringan yang berbeda.

Modbus bekerja berdasakan pola *request*/*reply*. *Request* dilakukan oleh alat yang berperan sebagai *master*; sedangkan *reply* dilakukan oleh alat yang berperan sebagai *slave*. Secara umum, satu *frame* paket data Modbus terdiri atas *Protocol Data Unit* (PDU) dan *Application Data Unit* (ADU). PDU terdiri atas *function code* dan data, sedangkan ADU terdiri atas PDU ditambah *address* dan *error check*.

Secara *de facto*, Modbus mulai dipakai sejak 1979. Saat ini, jutaan peralatan otomasi mengunakan protokol ini. Beberapa peralatan peralatan dalam sistem SCADA seperti PLC, HMI, *Control panel*, *driver*, kendali motor, dan I/O menggunakan protokol Modbus ini untuk saling terhubung jarak jauh. Agar dapat digunakan pada jarak hingga lebih dari 1 km, maka modul yang dibuat pada penelitian ini menggunakan RS485 [7].

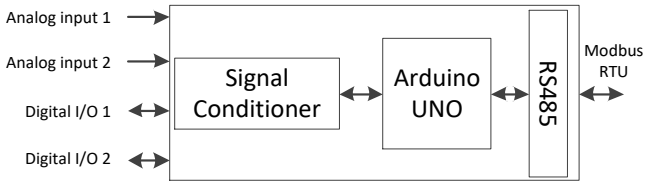
Modul akuisisi data dengan menggunakan Modbus RTU telah banyak digunakan di industri. Namun sayangnya, harga peralatan tersebut tergolong mahal. Protokol komunikasi Modbus RTU cukup kompleks untuk dibuat oleh pengguna awam. Dengan latar belakang itulah penelitian ini membuat modul komunikasi Modbus RTU *over* RS485 yang berbasiskan Arduino sehingga harga pembuatannya menjadi murah, di samping dapat menjadi pembelajaran bagi pengguna awam, seperti lembaga pendidikan dan pelatihan.

II. METODE

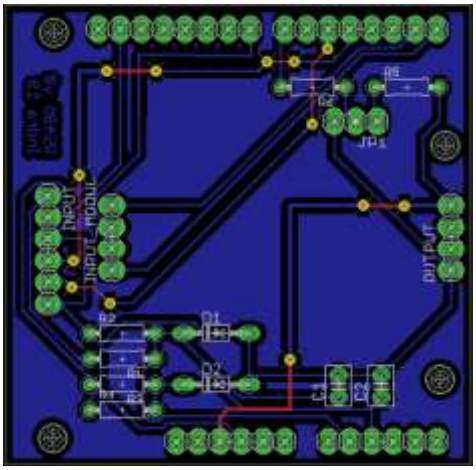
A. Perancangan Elektronik

Diagram blok sistem elektronik pada modul yang dibuat ini dapat dilihat pada Gambar 1. *Input* analog terdiri atas 2 *channel*. Digital *input*/*output* terdiri atas 2 *channel*. Sebelum masuk ke

Analog-to-Digital converter (ADC) milik Arduino, sinyal analog dimasukkan terlebih dahulu ke *signal conditioner* berupa pembagi tegangan, dioda Zener, dan kapasitor. Hal ini berfungsi untuk *filtering* sinyal yang masuk dan mencegah *over voltage*. Gambar 2 memperlihatkan layout *Printed Circuit Board* (PCB) yang dibuat. PCB ini bersifat *plug-and-play* untuk dipasang di atas Arduino Uno (menjadi *Arduino shield*).



Gambar 1. Diagram blok modul



Gambar 2. *Layout* PCB

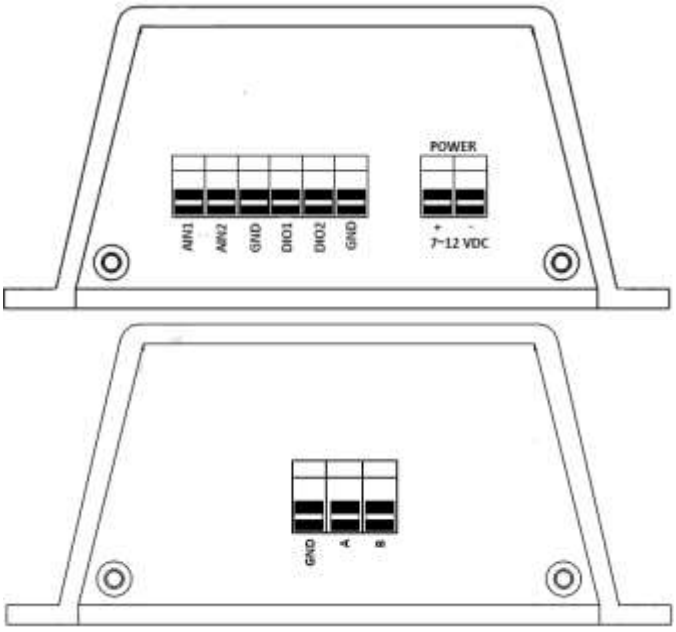
Modul ini dilengkapi dengan dua buah *Light Emitting Diode* (LED). LED berwarna merah adalah indikasi *power*, yang akan menyala ketika ada *power* yang masuk. LED berwarna biru adalah indikator yang akan memberitahukan bahwa komunikasi antara Modbus *slave* dan Modbus *master* sedang berlangsung.

B. Perancangan Mekanik

Gambar 3 menunjukkan desain *casing* yang dibuat menggunakan 123D Design (Autodesk, San Rafael, Amerika Serikat), sebuah perangkat lunak gratis untuk desain tiga-dimensi (3D). Pembuatan *casing* dilakukan menggunakan mesin 3D *printer*.

Gambar 4 menunjukkan konektor untuk akuisisi data dan koneksi untuk komunikasi menggunakan RS485. Konektor untuk akuisisi data terdiri atas enam pin (*analog input* #1, *analog input* #2, *digital* I/O #1, *digital* I/O #2, dan dua *ground*). Selain itu ditambahkan pula dua pin untuk *input* suplai 7~12 VDC. Konektor untuk komunikasi terdiri atas tiga pin, yakni A, B, dan GND, sesuai dengan standar RS485. Gambar 5 memperlihatkan modul yang telah dibuat. Modul ini diberi nama MR48519A.

Gambar 3. Desain *casing*: sisi *input* (kiri) dan sisi *output* (kanan)



Gambar 4. Konektor untuk akuisisi data (atas) dan konektor untuk komunikasi menggunakan RS485 (bawah)

Gambar 5. Modul MR48519A

C. Data Frame

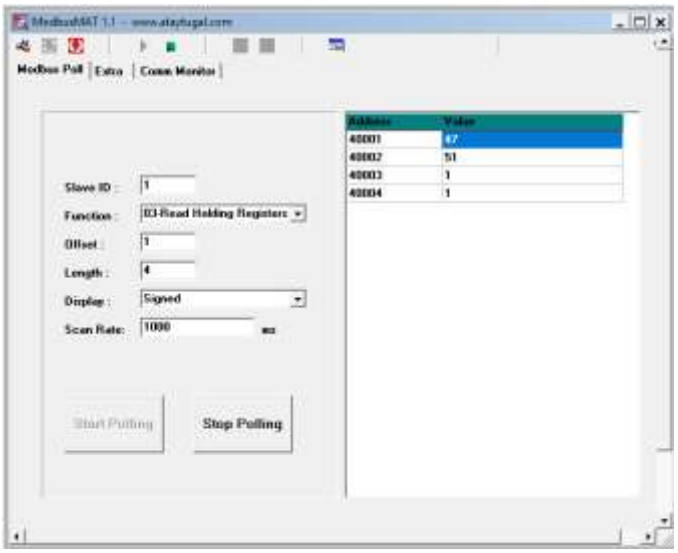
Modul RTU yang dibuat ini berperan sebagai Modbus *slave* dengan *data frame* sesuai pada Gambar 6. *Data frame* terdiri atas *slave* ID, *function code*, panjang data, data, dan *Cyclic Redundancy Check* (CRC). *Slave* ID pada penelitian ini diset 0x01, *function code* yang digunakan adalah 0x03 (*read holding registers*). Panjang data adalah 8 byte. Data terdiri atas *input* analog (4 byte) dan *input*/*output* digital (4 byte). Data analog menggunakan dua channel. Masing-masing *channel* memerlukan 2 byte, sehingga total data yang diperlukan untuk membaca data analog adalah 4 byte. Data digital merupakan dua channel *input*/*output* yang masing-masing menggunakan 2 byte, sehingga total alokasi yang diperlukan untuk data digital adalah 4 byte. Paket terakhir adalah CRC-16 (2 byte).

Slave ID	*Function Code*	Panjang data	data (4 x 2 byte)	CRC-16

Gambar 6. *Data frame* Modbus RTU

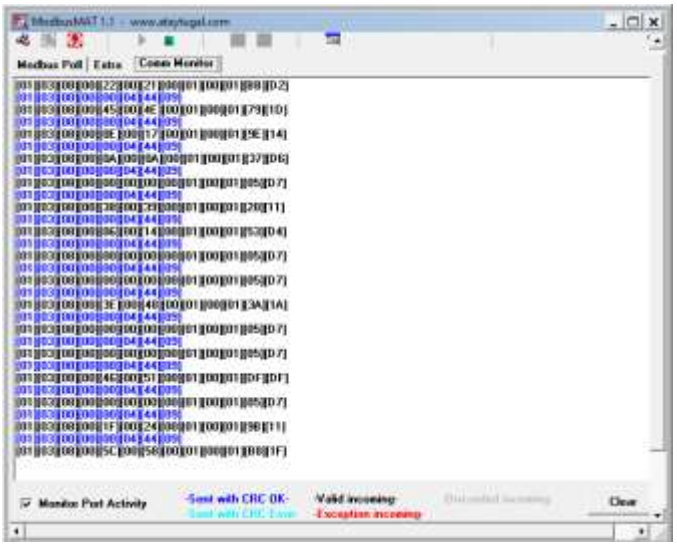
III. HASIL DAN PEMBAHASAN

Pada pengujian ini, kami menggunakan perangkat lunak ModbusMAT v1.1, sebuah *Modbus analyzer* yang digunakan sebagai Modbus *server*. Cara kerja *analyzer* ini adalah memberikan perintah kepada Modbus *slave* dan membaca data yang masuk dari *slave* sebagaimana cara kerja standar sebuah Modbus *server*. Pengujian dilakukan dengan menghubungkan *master* dan *slave* menggunakan koneksi RS485. Pembacaan data dari modul *slave* dilakukan secara secara berkala per 1000 dan 500 ms. Perhatikan Gambar 7.



Gambar 7. Hasil pembacaan data yang masuk menggunakan Modbus RTU

ModbusMAT menyediakan menu *port monitor* untuk menganalisis aliran data, baik dari *master* ke *slave* maupun sebaliknya. Perhatikan Gambar 8. Data berwarna biru merupakan data dari *master* ke *slave*, sedangkan yang berwarna hitam adalah aliran data dari *slave* ke *master*.



Gambar 8. Analisis data yang masuk menggunakan Modbus RTU

Sebagai contoh kasus, perhatikan baris ke-2 pada aliran data dalam Gambar 8, yakni 0x01, 0x03, 0x00, 0x00, 0x00, 0x04, 0x44, 0x09, yang merupakan *data frame master-to-slave*. Dengan menggunakan kalkulator, diperoleh CRC-16 dari 0x010300000004 adalah 0x0944. Hal ini sesuai dengan CRC-16 pada *data frame*. *Data frame master-to-slave* ini diuraikan lebih detail dalam Tabel I.

TABEL I
DATA FRAME MASTER-TO-SLAVE

Data frame	Keterangan
0x01	*Slave* ID = 1
0x03	*Function code* = 3 (*Read holding register*)
0x00	Data
0x00	Data
0x00	Data
0x04	Data
0x44 0x09	CRC-16 = 0x0944

TABEL II
DATA FRAME SLAVE-TO-MASTER

Data frame	Keterangan
0x01	Slave ID = 1
0x03	Function code = 3 (Read holding register)
0x08	Panjang data = 8 byte
0x00	Data
0x45	Data
0x00	Data
0x4E	Data
0x00	Data
0x01	Data
0x00	Data
0x01	Data
0x79 0x1D	CRC-16 = 0x1D79

Data frame pada Tabel I diterima oleh *slave* untuk diproses. Kemudian *slave* merespon dengan mengirimkan *data frame* ke *master* sesuai pada baris ke-3 pada aliran data dalam Gambar 8, yakni 0x01, 0x03, 0x08, 0x00, 0x45, 0x00, 0x4E, 0x00, 0x01, 0x00, 0x01, 0x79, 0x1D. Dengan menggunakan kalkulator, diperoleh CRC-16 dari 0x0103080045004E00010001 adalah

0x1D79. Hal ini sesuai dengan CRC-16 pada *data frame*. Data frame *slave-to-master* ini diuraikan lebih detail dalam Tabel II.

Tabel I dan Tabel II menunjukkan bahwa Modbus *master* berhasil mengirimkan perintah ke *slave* dengan menggunakan *data frame* yang benar. Pada arah sebaliknya (*slave-to-master*), data (berupa analog dan digital) berhasil dikirimkan oleh *slave* ke Modbus *master* sesuai dengan *data frame* standar Modbus yang telah didesain. Perhitungan CRC-16 dapat dilakukan dengan tepat. Dengan demikian, dapat disimpulkan bahwa komunikasi Modbus berjalan sesuai dengan *data frame* yang didesain.

Modul yang dibuat ini masih dalam tahap pengembangan sehingga memiliki beberapa keterbatasan. Ke depannya, modifikasi protokol perlu dilakukan untuk memastikan *data reliability*. Pengujian jarak maksimal penggunaan modul ini perlu dilakukan sesuai dengan standar RS485. *Safety* dan performansi dari modul ini perlu diuji lebih lanjut sesuai Standar Nasional Indonesia (SNI) dan International Electrotechnical Commission (IEC). Kalibrasi juga perlu dilakukan untuk memastikan kebenaran pembacaan data.

IV. KESIMPULAN

Modul Modbus RTU *over* RS485 berbasis Arduino telah dibuat. Modul ini berbiaya murah dan diharapkan sesuai untuk sarana pembelajaran bagi pengguna yang awam, seperti lembaga pendidikan dan pelatihan. Hasil pengujian menunjukkan bahwa komunikasi Modbus dapat berjalan sesuai dengan *data frame* yang didesain. Penelitian selanjutnya perlu dilakukan untuk memastikan *data reliability*. Pengujian jarak maksimal, *safety*, dan performansi dari modul ini perlu diuji lebih lanjut sesuai standar SNI dan IEC. Demikian halnya dengan proses kalibrasi.

UCAPAN TERIMA KASIH

Penelitian ini dibiayai oleh Pusat Penelitian dan Pengabdian Masyarakat Politeknik Negeri Batam melalui penelitan internal dengan skema Penelitian Madya.

REFERENSI

[1] D. Bailey and E. Wright, *Practical SCADA for Industry*. Burlington, USA: Elsevier, 2003.

[2] M. Bagajewicz, A. Fuxman, and A. Uribe, “Instrumentation network design and upgrade for process monitoring and fault detection,” *AIChE J.*, vol. 50, no. 8, pp. 1870–1880, Aug. 2004, doi: 10.1002/aic.10279.

[3] S. K. Mittal, M. Singh, P. Kapur, B. K. Sharma, and M. A. Shamshi, “Design and development of instrumentation network for landslide monitoring and issue an early warning,” *J. Sci. Ind. Res. (India).*, vol. 67, no. 5, 2008.

[4] F. Ponci, A. Sadu, R. Uhl, M. Mirz, A. Angioni, and A. Monti, “Instrumentation and measurement testing in the real-Time lab for automation of complex power systems,” *IEEE Instrum. Meas. Mag.*, vol. 21, no. 1, 2018, doi: 10.1109/MIM.2018.8278805.

[5] D. J. Kang and R. J. Robles, “Compartmentalization of protocols in SCADA communication,” *Int. J. Adv. Sci. Technol.*, vol. 8, pp. 27–36, 2009.

[6] Modbus, *Modbus*. Hopkinton, MA, USA: Modbus Organization, Inc., 2012.

[7] *RS-422 and RS-485 Applications Ebook: A Practical Guide to Using RS-422 and RS-485 Serial Interfaces*. Ottawa, Illinois, USA: Advantech B+B SmartWorx, 2010.