

Perancangan dan Implementasi Komunikasi RS-485 Menggunakan Protokol Modbus RTU dan Modbus TCP pada Sistem *Pick-by-Light*

Agus Mulyana¹, Tosin^{2*}

^{1,2}Program Studi Teknik Komputer, Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia
Jl. Dipati Ukur No. 112 - 116, Bandung, Indonesia 40132

*email: tosin@mahasiswa.unikom.ac.id

(Naskah masuk: 29 Agustus 2020; diterima untuk diterbitkan: 02 Nopember 2020)

ABSTRAK – Proses di industri membutuhkan produktivitas yang tinggi dalam setiap aktivitasnya. Salah satu contohnya adalah proses pengambilan barang di tempat penyimpanan. Sistem yang ada sekarang adalah operator mengambil barang dengan mencari secara manual ke tempat penyimpanan. Hal tersebut akan menimbulkan masalah yaitu proses pengambilan barang membutuhkan waktu yang lama atau terjadi kesalahan dalam pengambilan barang. Tujuan dari penelitian ini adalah merancang sistem pick-by-light dengan komunikasi serial RS-485 dan TCP/IP menggunakan protokol Modbus untuk mendukung aktivitas produksi yang efisien. Pick-by-light adalah sistem yang terdiri dari perangkat lampu sebagai tanda barang tersebut harus diambil, tombol sebagai konfirmasi bahwa barang sudah diambil, dan perangkat pemroses untuk memproses data. Hasil dari pengujian menunjukkan bahwa sistem pick-by-light dapat berjalan dengan baik dengan menerapkan komunikasi serial RS-485 dengan protokol Modbus RTU dan komunikasi TCP/IP dengan protokol Modbus TCP. Hasil ini didapat dari pengujian yang telah dilakukan dengan komunikasi yang berbeda, yaitu menggunakan komunikasi serial RS-485 dan TCP/IP. Dengan demikian, sistem ini dapat diterapkan pada industri manufaktur maupun pergudangan dalam meningkatkan produktivitas pengambilan barang.

Kata Kunci – pick-by-light; RS-485; TCP/IP; Modbus RTU, Modbus TCP.

Design and Implementation of RS-485 Communication Using Modbus RTU and Modbus TCP Protocol on Pick-by-Light System

ABSTRACT – The industrial process requires high productivity in every activity. An example is a process of taking items in storage area. The current system is that the operator takes the items by manually searching in the storage. It will cause problems, namely, the process of taking items takes a long time or occurred a mistake in taking items. The purpose of this research is to design a pick-by-light system with RS-485 serial communication and TCP/IP using the Modbus protocol to support efficient production activities. Pick-by-light is a system consisting of a lightweight device as a sign that an item must be taken, a button to confirm that an item has been taken, and a processing device to process data. The result of this research shows that the pick-by-light system can work properly by implementing RS-485 serial communication with the Modbus Protocol and TCP/IP communication with the Modbus TCP protocol. These results are obtained from tests that have been carried out with different communications using RS-485 serial communication and TCP/IP. It can be applied to manufacturing and warehousing industries to increasing the productivity of items taking.

Keywords - pick-by-light; RS-485; TCP/IP; Modbus RTU, Modbus TCP.

1. PENDAHULUAN

Aktivitas dunia industri membutuhkan sistem kerja yang efisien dan efektif untuk menciptakan produktivitas yang tinggi. Salah satu contoh aktivitas

industri manufaktur maupun pergudangan adalah sistem pengambilan barang. Sistem yang ada sekarang operator akan mengambil barang sesuai daftar barang dengan mencari lokasi barang pada

tempat penyimpanan secara manual. Masalah yang muncul dari aktivitas tersebut adalah waktu yang diperlukan dalam pencarian barang menjadi lama sehingga menurunkan produktivitas produksi.

Untuk mengatasi masalah tersebut diperlukan sebuah sistem yang mampu mempercepat proses pengambilan barang. Pada penelitian ini bermaksud untuk membuat sistem *pick-by-light* yang berfungsi untuk memberikan tanda kepada operator berupa lampu yang menyala pada tempat penyimpanan barang[1]. Dengan sistem tersebut dapat meningkatkan akurasi sekaligus meningkatkan produktivitas dengan mengurangi waktu pencarian[2]. Tujuan penelitian ini adalah menerapkan protokol Modbus RTU dan Modbus TCP pada sistem *pick-by-light* menggunakan komunikasi RS-485 dan TCP/IP.

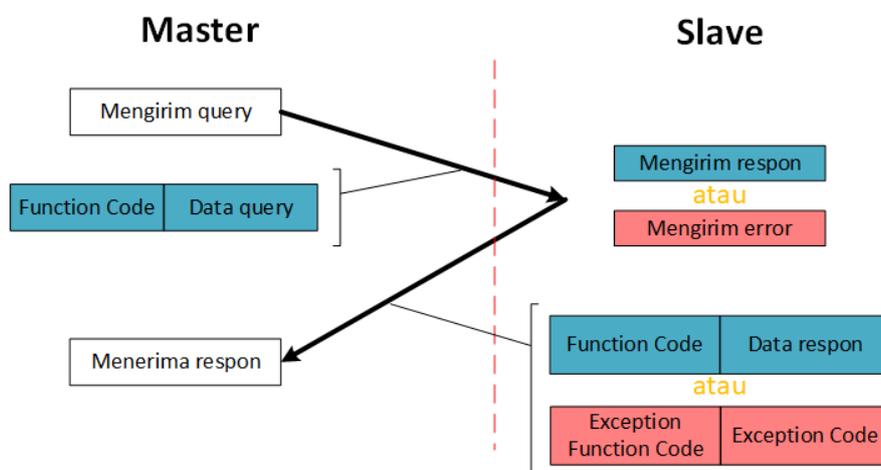
Sistem *pick-by-light* terdiri dari dua bagian, yaitu aplikasi perangkat lunak sebagai pengirim perintah dan perangkat keras sebagai pemberi tanda. Pada perangkat keras terdiri dari pengolah data, lampu LED, dan tombol sebagai konfirmasi barang tersebut sudah diambil[3]. Jumlah lampu LED dan tombol dapat bervariasi sesuai dengan tempat penyimpanan barang. Penelitian ini berfokus pada perancangan perangkat keras dengan jumlah satu buah perangkat pengolah data yang disebut dengan konsentrator serta empat buah lampu LED dan tombol yang disimpan pada dalam sebuah *box* yang disebut dengan *control box*. Pada sistem ini menerapkan sistem *close loop* atau sistem kendali tertutup[4].

Perangkat antarmuka yang digunakan dalam menghubungkan perangkat lunak dan perangkat keras adalah RS-485 dan TCP/IP. Dalam mengatur pertukaran data, sistem ini menggunakan protokol Modbus. Protokol Modbus yang berjalan pada komunikasi serial RS-485 adalah Modbus RTU sedangkan Modbus TCP berjalan pada komunikasi TCP/IP.

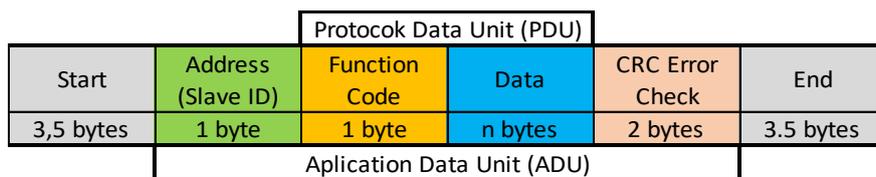
RS-485 mempunyai nama lengkap *EIA/TIA-485*

Standard for Electrical Characteristics of Generators and Receivers for use in a Balanced Digital Multipoint System adalah sebuah standar komunikasi serial asinkron berstandar industri yang ditetapkan oleh *Electronics Industries Association (EIA)* pada tahun 1983[5]. RS-485 berjalan pada *layer* fisik yang dapat mengirimkan data sejauh 1,2 km dan menghubungkan *slave* secara *one to many* dengan metode *multiple point* sampai dengan 32 perangkat dengan dua kabel tanpa referensi *ground* yang sama[6]-[9]. Komunikasi RS-485 menggunakan transmisi *diferensial balanced transmission* yaitu mengubah tegangan TTL menjadi selisih tegangan antara output A dan B sehingga meminimalkan efek dari *noise*[10]. Selisih tegangan antara output A dan B akan tetap karena *interferensi noise* akan terjadi sekaligus pada jalur *output A* dan jalur *complementary output B*[8]. Pada penelitian ini menggunakan RS-485 karena mempunyai beberapa kelebihan, diantaranya yaitu biaya pembuatan yang rendah, dapat menghubungkan banyak perangkat, memiliki *anti-interference*, dan dapat mencapai jarak yang jauh[11]. Topologi yang digunakan pada RS-485 adalah *daisy chain* dimana setiap *slave* terhubung ke *master* dalam satu jalur.

Modbus adalah protokol komunikasi jaringan berstandar internasional yang diterapkan pada industri dan bersifat *open source*, dapat berjalan pada berbagai media antarmuka serta sederhana dan efisien[12]. Modbus pertama kali dipublikasikan oleh Modicon pada tahun 1979 yang digunakan pada *Programmable Logic Controllers (PLC)*[13]. Perangkat yang mengirimkan perintah disebut dengan *master* dan penerima perintah disebut *slave*. *Master* bersifat aktif dengan mengirimkan permintaan atau *query* yang terdiri dari *function code* dan data. Sedangkan *slave* bersifat pasif yang hanya merespon jika ada permintaan dari *master* dengan mengirimkan pesan *data response* saat kondisi normal maupun *exception code* saat terjadi *error*. Proses transaksi dari *master* ke *slave* dapat dilihat pada gambar 1.



Gambar 1. Proses Transaksi Master dan Slave pada Modbus



Gambar 2. Format Frame Modbus RTU

Function code adalah perintah yang harus dikerjakan oleh slave. Setiap function code mempunyai fungsi yang berbeda sesuai dengan tipe data dan jenis perintah. Tipe data dapat berupa bilangan diskrit maupun analog, sedangkan jenis perintah dapat menulis atau membaca data. Setiap data disimpan pada register dan coil dengan alamat yang berbeda. Function code pada Modbus dapat dilihat pada tabel 1.

Tabel 1. Function Code Modbus

Function Code	Perintah	Jenis Data
01 (01 hex)	Read	Discrete Output
05 (05 hex)	Write Single	Discrete Output
15 (0F hex)	Write Multiple	Discrete Output
02 (02 hex)	Read	Discrete Input
04 (04 hex)	Read	Analog Input
03 (03 hex)	Read	Analog Output
06 (06 hex)	Write Single	Analog Output
16 (10 hex)	Write Multiple	Analog Output

Modbus RTU (Remote Terminal Unit) merupakan varian Modbus yang digunakan pada komunikasi serial. Modbus RTU berjalan pada layer Data Link OSI Model sedangkan pada layer fisik menggunakan RS-485 dan RS-232[14]. Format frame pada Modbus RTU terdiri dari start bit, slave ID, function code, data, Cyclic Redundant Check (CRC), dan end bit. Frame Modbus RTU dapat dilihat pada gambar 2. Keuntungan dari Modbus RTU adalah lebih efisien dalam komunikasi karena dapat mengirim lebih banyak data dalam baud rate yang sama[15]. Setiap data yang dikirim memiliki waktu tunda antara 1,5 sampai dengan 3,5 karakter pada awal dan akhir pesan dari baud rate yang dipakai[16].

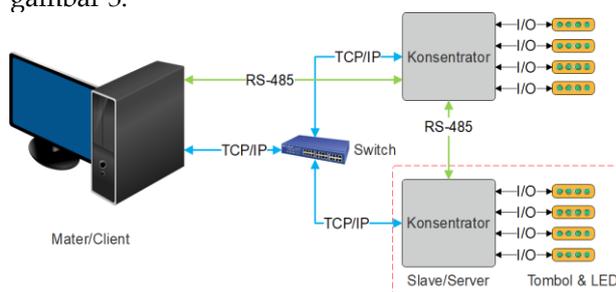
Modbus TCP adalah protokol Modbus yang menggunakan komunikasi TCP/IP dengan media Ethernet. Modbus TCP berjalan pada layer aplikasi TCP/IP sebagai metode untuk merepresentasikan data[17]. Modbus TCP menanamkan frame data Modbus standar ke dalam frame TCP tanpa checksum yang berisi Modbus Application Protocol (MBAP) dan Protocol Data Unit (PDU)[18]. Pada Modbus TCP menggunakan model komunikasi client/server, dimana master bertindak sebagai client dan slave sebagai server.

2. METODE DAN BAHAN

Tahapan penelitian dalam penelitian ini yaitu melakukan observasi ke lapangan yang bertujuan untuk menganalisis kebutuhan sistem yang terdiri dari analisis kebutuhan fungsional dan non-fungsional serta kebutuhan perangkat keras dan perangkat lunak. Tahapan selanjutnya yaitu melakukan perancangan sistem yang meliputi pembuatan arsitektur sistem, diagram blok, skematik rangkaian, dan desain alat.

Bahan-bahan yang digunakan dalam penelitian ini diantaranya yaitu Arduino Mega Pro Mini yang berbasis ATmega2560, modul Ethernet Wiznet W5500 untuk mengkonversi komunikasi serial ke Ethernet, modul RS-485 untuk mengkonversi komunikasi serial ke RS-485, dan Illuminated Push Button yaitu perangkat tombol sekaligus lampu LED. Adapun perangkat lunak yang digunakan adalah Arduino IDE untuk memprogram Arduino, Modbus Poll sebagai aplikasi simulator Modbus, dan Eagle untuk membuat PCB.

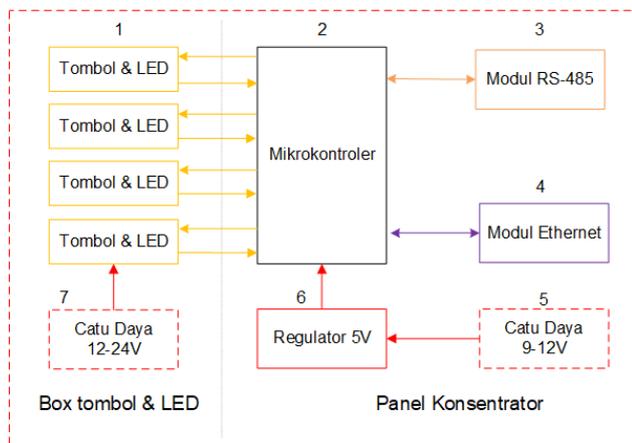
Arsitektur sistem yang akan dibuat terdiri dari aplikasi desktop sebagai master/client yang berfungsi mengolah data barang yang akan diambil, perangkat konsentrator sebagai slave/server, dan control box yang terdiri dari tombol dan LED. Pada penelitian ini hanya berfokus pada perancangan dan pembuatan konsentrator dan control box yang dapat dilihat pada gambar 3.



Gambar 3. Arsitektur Sistem

Pada bagian konsentrator terdapat mikrokontroler yang berfungsi membaca data dari master/client kemudian mengatur LED dan membaca penekanan tombol, modul RS-485, modul Ethernet, rangkaian regulator tegangan, rangkaian digital isolated input, dan rangkaian digital output. Tegangan yang dibutuhkan dalam menghidupkan perangkat konsentrator adalah 9-12 Volt DC dan 12-24 Volt DC untuk menghidupkan LED. Untuk menghubungkan

konsentrator dan *control box* menggunakan kabel UTP. Diagram blok sistem yang akan dibuat dapat dilihat pada gambar 4.



Gambar 4. Diagram Blok Sistem

Berdasarkan diagram blok yang sudah dibuat, langkah selanjutnya yaitu pembuatan skematik. Dari skematik tersebut akan dibuat PCB yang berfungsi sebagai tempat untuk menghubungkan semua komponen. Skematik rangkaian konsentrator dapat dilihat pada gambar 5.

Pada program Arduino akan membaca *query* dari *master/client*. Jika terdapat *error*, maka akan

mengirimkan pesan *exception error*. Sedangkan jika tidak ada *error*, maka *query* tersebut akan dieksekusi sesuai fungsi dan data yang dikirim.

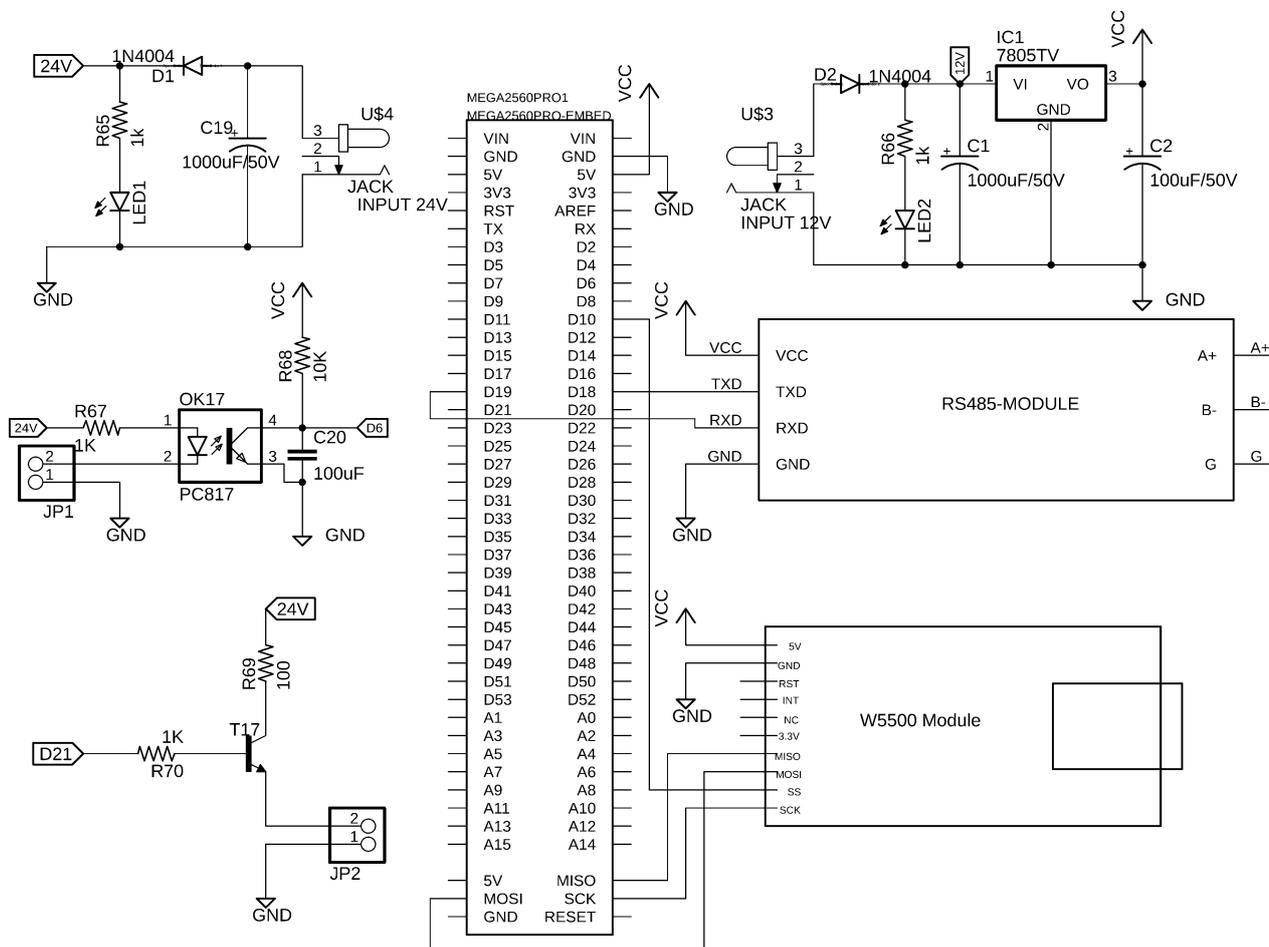
Pengujian yang akan dilakukan meliputi pengujian komunikasi menggunakan fungsi *single read* dan *single write* yaitu membaca dan menulis satu data serta *multiple read* dan *multiple write* yaitu membaca dan menulis banyak data sekaligus.

3. HASIL DAN PEMBAHASAN

Pengujian dilakukan dengan empat metode, yaitu komunikasi serial RS-485 menggunakan protokol Modbus RTU, komunikasi TCP/IP menggunakan protokol Modbus TCP, komunikasi *multiple point*, dan pengujian integrasi.

3.1. Komunikasi Serial RS-485

Pengujian pertama dilakukan menggunakan protokol Modbus RTU dengan fungsi *single read*. Perangkat *master* menggunakan aplikasi Modbus Poll, konsentrator sebagai *slave*, dan aplikasi Serial Port Monitor untuk melihat data yang dikirim. Pada fungsi *single read*, *master* akan mengirim *query* ke *slave* untuk meminta data dari *slave*. Kemudian *slave* akan mengirim data yang diminta. Proses pengujian fungsi *single read* dapat dilihat pada gambar 6.



Gambar 5. Skematik Konsentrator

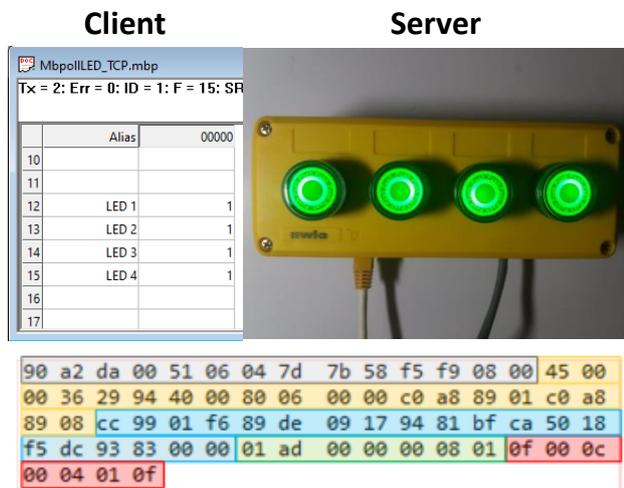


Gambar 6. Pengujian Single Read Modbus RTU

Berdasarkan gambar 6, master mengirimkan query berupa perintah untuk membaca satu buah data pada coil 0 dengan alamat slave 01. Saat tombol pada control box ditekan, konsentrator sebagai slave akan membaca penekanan tombol tersebut dan mengirimkan data ke master. Data yang dikirim dari slave berjumlah enam byte angka heksa desimal yang disebut dengan Application Data Unit (ADU). ADU terdiri dari slave ID, Protocol Data Unit (PDU) yang terdiri dari function code dan data, serta CRC. Byte pertama yaitu 01 menunjukkan slave ID, 02 adalah function code read input status, 01 alamat coil, 01 status coil, dan 60 48 adalah CRC.

3.2. Komunikasi TCP/IP

Pengujian selanjutnya yaitu menggunakan komunikasi TCP/IP dengan protokol Modbus TCP. Perangkat client menggunakan aplikasi Modbus Poll, konsentrator sebagai server, dan aplikasi Wire Shark untuk melihat data yang dikirim. Fungsi yang digunakan dalam pengujian ini yaitu multiple write dimana client akan mengirim perintah untuk menghidupkan empat buah LED. Proses pengujian Modbus TCP dapat dilihat pada gambar 7.



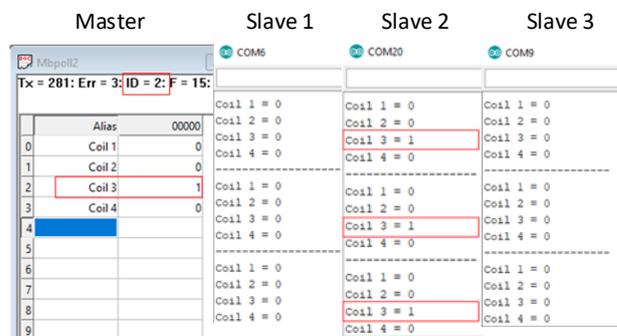
Gambar 7. Pengujian Multiple Write Modbus TCP

Berdasarkan gambar 7, client mengirimkan query

yang berisi perintah untuk menghidupkan empat buah LED dengan mengirim fungsi multiple write dan empat bit data yang masing-masing berada pada coil 12,13,14, dan 15. Data tersebut kemudian diterima oleh konsentrator sebagai server dengan menghidupkan LED. Data yang dikirim melalui proses enkapsulasi data pada setiap layer yang dilewatinya yang ditunjukkan dengan warna yang berbeda. Blok warna merah menunjukkan PDU Modbus yang terdiri dari function code dan. Blok warna hijau merupakan frame Modbus TCP dengan menambahkan Unit Identifier data. Blok warna biru merupakan frame TCP dengan menambahkan port asal dan tujuan data. Secara default, port server adalah 502. Blok warna jingga merupakan frame IP dengan menambahkan alamat IP asal dan tujuan data. Pada pengujian ini alamat server adalah 192.168.137.8. Blok warna abu-abu adalah frame Ethernet dengan menambahkan alamat Ethernet asal dan tujuan.

3.3. Komunikasi Multiple Point

Pada pengujian multiple point menggunakan komunikasi RS-485 dengan sebuah master dan tiga buah slave. Aplikasi Modbus Poll sebagai master dan Arduino sebagai slave. Jarak antar slave sejauh 10 meter, sehingga total jarak dari master ke slave terjauh adalah 30 meter. Serial Monitor pada Arduino IDE digunakan untuk dapat melihat data yang diterima. Hasil dari pengujian multiple point dapat dilihat pada gambar 8.



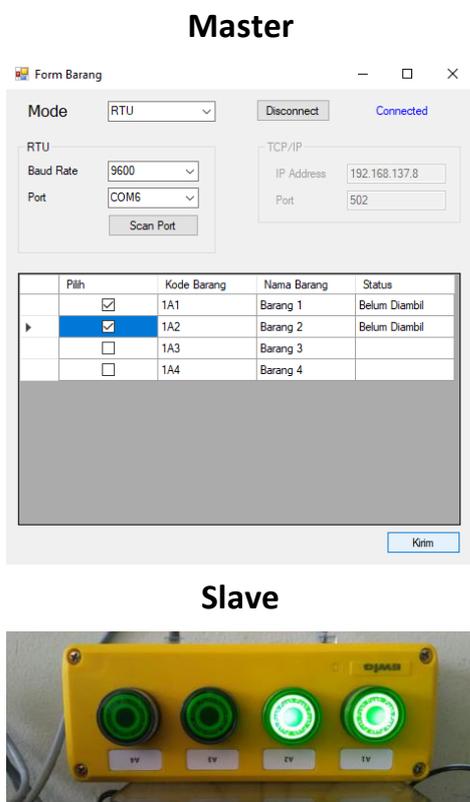
Gambar 8. Pengujian Multiple Point

Berdasarkan gambar 8, master mengirimkan query berisi data 1 pada coil 3 dengan slave ID 2. Perangkat slave dengan ID 2 dapat menerima data yang kemudian ditampilkan pada Serial Monitor. Sedangkan slave dengan ID 1 dan 3 tidak menerima data tersebut karena ID-nya berbeda walaupun terhubung dalam satu jaringan. Hal tersebut menunjukkan komunikasi multiple point telah berhasil.

3.4. Pengujian Integrasi

Pada pengujian integrasi dilakukan menggunakan aplikasi desktop sebagai master yang berfungsi untuk mengolah data barang yang akan

diambil, konsentrator sebagai *slave*, dan *control box* untuk menampilkan LED dan tombol. Pengujian ini menggunakan komunikasi serial RS-485 dengan protokol Modbus RTU yang dapat dilihat pada gambar 9.



Gambar 9. Pengujian Integrasi

Dalam pengujian ini, *master* akan mengirim perintah untuk menghidupkan LED 1 dan 2 dimana posisi barang akan diambil. Pengujian ini berhasil menghidupkan LED sesuai dengan lokasi barang yang dimaksud dan menampilkan status barang yang belum diambil pada aplikasi *desktop*. Selanjutnya operator akan menuju ke LED yang menyala. Setelah mengambil barang, operator menekan tombol sebagai konfirmasi bahwa barang tersebut sudah diambil. Setelah tombol ditekan, *slave* akan mengubah status barang pada *coil* kemudian *master* akan membaca status tersebut dan mengirim perintah untuk mematikan LED yang berarti barang tersebut sudah diambil. Status barang akan berubah menjadi sudah diambil.

4. KESIMPULAN

Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa penelitian ini sudah berhasil merancang dan menerapkan komunikasi serial RS-485 menggunakan protokol Modbus RTU dan Modbus TCP. Hasil tersebut didapat dari pengujian yang sudah dilakukan berdasarkan tujuan penelitian yaitu memberi tanda kepada operator berupa LED yang menyala sebagai tanda barang tersebut harus diambil. Untuk

pengembangan selanjutnya dapat menerapkan mode komunikasi lain misalnya menggunakan LoRa dengan protokol Modbus.

DAFTAR PUSTAKA

- [1] J. De Vries, R. De Koster, and D. Stam, "Exploring the role of picker personality in predicting picking performance with pick by voice, pick to light and RF-terminal picking," *Int. J. Prod. Res.*, vol. 54, no. 8, pp. 2260–2274, 2016.
- [2] G. Dukić, V. Česnik, and T. Opetuk, "Order-picking methods and technologies for greener warehousing," *Strojarstvo*, vol. 52, no. 1, pp. 23–31, 2010.
- [3] Merari, "Metode Pick To Light Pergudangan," *Excelogic*, 2014. [Online]. Available: <http://excelogic.info/metode-pick-to-light-pergudangan.html>. [Accessed: 30-Jan-2020].
- [4] S. Supatmi, W. Zarman, and A. Mulyana, "Kendali Jarak Jauh Lampu Gedung Menggunakan Frekuensi Radio," *J. Tek. Komput. Unikom – Komputika*, vol. 3, no. 1, pp. 14–20, 2014.
- [5] T. Kugelstadt, "The RS-485 Design Guide Application Report The RS-485 Design Guide," no. October, pp. 1–10, 2008.
- [6] J. Bhatt and H. Verma, "RS-485/MODBUS based Intelligent Building Automation System using LabVIEW," *4th Int. Conf. Comput. Appl. Electr. Eng. Adv.*, p. 5, 2010.
- [7] J. K. R. Sastry, A. Suresh, and S. J. S. Bhanu, "Building heterogeneous distributed embedded systems through rs485 communication protocol," *ARPN J. Eng. Appl. Sci.*, vol. 10, no. 16, pp. 6793–6803, 2015.
- [8] A. Salam and T. Sucita, "Rancang Bangun Sistem Jaringan Multidrop Menggunakan Rs485," *electrans*, vol. 11, no. 2, pp. 1–11, 2012.
- [9] A. Zainuri, "Aplikasi Sistem Komunikasi Serial Multipoint RS-485 Pada Kontrol Crane Barang," *Univ. Brawijaya, Malang*, 2010.
- [10] L. Zhao, R. Liang, and J. Zhang, "The Solving of Bias Resistor and Its Effect on the RS485 Fieldbus," *J. Adv. Comput. Networks*, vol. 2, no. 1, pp. 71–75, 2014.
- [11] L. Zhao, R. Liang, and J. Zhang, "Solving for the Best Value of Bias Resistor to Promote Stability of Rs485 Fieldbus," *Int. J. Futur. Gener. Commun. Netw.*, vol. 8, no. 3, pp. 89–96, 2015.
- [12] L. Hui, Z. Hao, and P. Daogang, "Design and Application of Communication Gateway of EPA and MODBUS on Electric Power System," *Energy Procedia*, vol. 17, pp. 286–292, 2012.
- [13] Nurpadmi, "Studi Tentang Modbus Protokol pada Sistem Kontrol," *Swara Patra*, vol. 01, no. 2, 2010.
- [14] G. Jakaboczki and E. Adamko, "Vulnerabilities

- of Modbus Rtu Protocol - a Case Study," *Ann. ORADEA Univ. Fascicle Manag. Technol. Eng.*, vol. XXIV (XIV), no. 1, 2015.
- [15] J. F. Li and S. Cao, "Remote monitoring and management system of CNG flow based on modbus RTU protocol," *Int. J. Interact. Mob. Technol.*, vol. 10, no. 5, pp. 52-56, 2014.
- [16] I. A. S. MODICON, Inc., "Modicon Modbus Protocol Reference Guide Modicon Modbus Protocol Reference Guide," *Int. Bus.*, 1996.
- [17] Acromag Incorporated, *introduction to Modbus TCP/IP*. Michigan, USA, 2005.
- [18] L. Xuan and L. Yongzhong, "Research and Implementation of Modbus TCP Security Enhancement Protocol," *J. Phys. Conf. Ser.*, vol. 1213, no. 5, 2019.