

# Integrasi Sistem Komunikasi *Modbus* TCP/IP pada PLC Siemens S7-1200, ESP32, dan HMI

Ananda Sakinata Prastiwi<sup>1</sup>, Ii Munadhif<sup>2</sup>, Isa Rachman<sup>3</sup>, Ryan Yudha Adhitya<sup>4</sup>, Rini Indarti<sup>5</sup>  
e-mail: [anandasakinata14@student.ppns.ac.id](mailto:anandasakinata14@student.ppns.ac.id), [iimunadhif@ppns.ac.id](mailto:iimunadhif@ppns.ac.id), [isarachman@ppns.ac.id](mailto:isarachman@ppns.ac.id),  
[ryanyudhaadhitya@ppns.ac.id](mailto:ryanyudhaadhitya@ppns.ac.id), [riniindarti@ppns.ac.id](mailto:riniindarti@ppns.ac.id)

<sup>1,2,3,4</sup>Program Studi Teknik Otomasi, Jurusan Teknik Kelistrikan Kapal, Politeknik Perkapalan Negeri Surabaya  
Kampus ITS Sukolilo Jl. Teknik Kimia, Keputih, Sukolilo, Surabaya, Indonesia

<sup>5</sup>Program Studi Teknik Kelistrikan Kapal, Jurusan Teknik Kelistrikan Kapal, Politeknik Perkapalan Negeri Surabaya  
Kampus ITS Sukolilo Jl. Teknik Kimia, Keputih, Sukolilo, Surabaya, Indonesia

## Informasi Artikel

### Riwayat Artikel

Diterima 10 Juli 2023

Direvisi 31 Juli 2023

Diterbitkan 31 Juli 2023

### Kata kunci:

ESP32

HMI

*Modbus* TCP/IP

PLC

### Keywords:

ESP32

HMI

*Modbus* TCP/IP

PLC

## ABSTRAK

Inovasi di dunia industri kini berkembang dengan pesat seiring dengan berkembangnya zaman. Protokol komunikasi *Modbus* TCP/IP telah digunakan secara luas dalam industri untuk menghubungkan perangkat-perangkat suatu sistem kontrol. Dengan peningkatan sistem ini, diyakini dapat membuat dan memberikan kontribusi yang lebih besar dalam lingkungan industri yang cakap *Programmable Logic Controller* (PLC) merupakan pengendali khusus pengganti *relay* untuk mengendalikan proses industri. PLC terdiri dari *Central Processing Unit* (CPU), modul *input output*, memori PLC (*Program Memory & Data Memory*), serta unit catu daya. Sementara ESP32 merupakan mikrokontroler yang populer dalam bidang otomasi industri karena fitur-fitur yang canggih dan dukungan komunikasi yang kuat. Untuk tampilan antarmuka *Human Machine Interface* (HMI) masih menjadi yang utama. Dengan HMI dapat memudahkan manusia untuk berinteraksi dengan mesin atau sistem komputer. Dalam komunikasi *Modbus* TCP/IP, perangkat yang berperan sebagai "*Client*" mengirim permintaan ke perangkat "*Server*" untuk mengakses atau mengontrol data tertentu. Pada penelitian ini akan mengkomunikasikan ketiga perangkat tersebut menggunakan protokol komunikasi *Modbus* TCP/IP. Tujuan dari penelitian ini untuk melakukan analisis terhadap implementasi komunikasi *Modbus* TCP/IP pada berbagai sektor. Untuk pengujian menggunakan data pembacaan sensor ultrasonik dan dihasilkan rata-rata waktu pada proses pengiriman data antara PLC dan ESP32 adalah 0.97 detik. Sedangkan saat proses pengiriman data pada pengujian keseluruhan sistem tidak terjadi *error*.

## ABSTRACT

*Innovation in the industrial world is now growing rapidly along with the times. The Modbus TCP/IP communication protocol has been widely used in industry to connect the devices of a control system. With the improvement of this system, it is believed to be able to make and make a greater contribution in a capable industrial environment Programmable Logic Controller (PLC) is a special controller to replace relays to control industrial processes. PLC consists of Central Processing Unit (CPU), input output module, PLC memory (Program Memory & Data Memory), and power supply unit. While ESP32 is a microcontroller that is popular in the field of industrial automation because of its advanced features and strong communication support. For the Human Machine Interface (HMI) display is still the main one. With HMI, it can make it easier for humans to interact with machines or computer systems. In Modbus TCP/IP communication, devices that act as "Client" send requests to "Server" devices to access or control certain data. This research will communicate the three devices using the Modbus TCP/IP communication protocol. The purpose of this research is to analyze the implementation of Modbus TCP/IP communication in various sectors. For testing using*



*ultrasonic sensor reading data and the resulting average time in the process of sending data between PLC and ESP32 is 0.97 seconds. While during the data transmission process in the overall system test no errors occurred.*

**Penulis Korespondensi:**

Ananda Sakinata Prastiwi,  
Jurusan Teknik Kelistrikan Kapal Program Studi Teknik Otomasi,  
Politeknik Perkapalan Negeri Surabaya,  
Kampus ITS Sukolilo Jl. Teknik Kimia, Keputih, Sukolilo, Surabaya, Indonesia, Kode Pos 60111  
Email: [anandasakinata14@student.ppns.ac.id](mailto:anandasakinata14@student.ppns.ac.id)  
Nomor HP/WA aktif: +62 822-5716-7669

## 1. PENDAHULUAN

Sistem kontrol sektor industri akhir-akhir ini berkembang dengan pesat. Inovasi dan teknologi terbaru terus muncul untuk meningkatkan efisiensi, keamanan, dan produktivitas proses industri. Integrasi dan komunikasi yang efektif tentunya menjadi salah satu bentuk kemajuan dalam sistem kontrol industri *modern*. Konsep pertukaran data dapat membuat beberapa *device* dapat saling mengirim, memproses, serta menerima data sensor. Kemampuan ini dapat dilakukan dengan berbagai media komunikasi seperti *Wi-Fi*, *LoRa*, *bluetooth*, dll [1]. Dalam konteks ini, protokol komunikasi *Modbus* TCP/IP telah digunakan secara luas dalam industri untuk menghubungkan perangkat-perangkat suatu sistem kontrol. Dengan peningkatan sistem ini, diyakini dapat membuat dan memberikan kontribusi yang lebih besar dalam lingkungan industri yang cakup.

PLC sering digunakan dalam berbagai industri seperti manufaktur, otomasi industri, serta kontrol proses. Beberapa metode pemrograman yang umum digunakan oleh PLC diantaranya *ladder diagram*, dan *function block diagram*. PLC juga sudah dilengkapi dengan modul-modul komunikasi seperti RS232, RS 485, Ethernet, serta *port* USB untuk berkomunikasi dengan perangkat lainnya [2]. ESP32 lebih umum digunakan dalam pengembangan kontrol berbasis TCP/IP yang dapat mengirim dan menerima data secara *real-time*. Penggunaan ESP32 pada industrial sistem dapat dipertimbangkan mengingat harga yang lebih murah dengan efisiensi yang cukup tinggi. Sebagai tampilan antarmuka *Human Machine Interface* (HMI) menjadi pilihan yang utama. HMI merupakan perangkat dan sarana yang sangat penting untuk suatu sistem pengendalian dan *monitoring* secara *real-time* [3]. HMI memungkinkan interaksi antara manusia dan mesin atau sistem komputer. Untuk memungkinkan komunikasi yang efektif diperlukan jembatan interoperabilitas antara PLC, ESP32, dan HMI. Komunikasi *Modbus* dapat dilakukan dengan perantara kabel Ethernet ataupun jaringan *Wi-Fi* dengan memanfaatkan komunikasi *Modbus* TCP/IP. Pengiriman Data *Modbus* dapat dilakukan di dalam jaringan lokal maupun internet [4]. Pada rancang bangun pengendalian tegangan simulator *input* dan *output* komunikasi PLC-ESP32 menggunakan protokol komunikasi *modbus* RTU dengan hasil pembacaan data analog *input* terbaca kurang lebih 1 detik [5]. Dalam hal lain, komunikasi antara PLC dengan ESP32 juga diterapkan pada rancang bangun *greenhouse* otomatis dengan kinerja sistem yang cukup baik, tetapi pada penelitian tersebut tidak membahas secara mendalam mengenai komunikasi antara PLC dan ESP32 [6].

Tujuan dari penelitian ini adalah untuk mengintegrasikan komunikasi *Modbus* TCP/IP sebagai penghubung antara PLC, ESP32, dan HMI dalam sistem kontrol industri. Penelitian ini akan membahas mengenai implementasi dan pengujian kinerja komunikasi antara PLC, ESP32, dan HMI menggunakan protokol *Modbus* TCP/IP. Selain itu akan dilakukan evaluasi terhadap latensi (waktu respon) komunikasi antara ketiga perangkat tersebut. Dengan demikian, penelitian ini dapat menambah wawasan yang lebih mendalam dan solusi yang lebih baik dalam mengoptimalkan integrasi PLC, ESP32, dan HMI dalam sistem kontrol industri yang efektif dan efisien.

## 2. METODE PENELITIAN

PLC merupakan sebuah peralatan yang *user firendly*, berbasis *microprocessor*, yaitu suatu komputer khusus dengan berbagai fungsi kontrol dan fleksibilitasnya [6]. Umumnya pemrograman kontrol logika untuk PLC dilakukan dengan perangkat lunak yang berjalan di komputer (PC). PLC digunakan sebagai pengganti relay. Dengan adanya PLC, sistem pengkabelan menjadi lebih mudah diatur dan lebih rapi, serta memudahkan dalam membuat



instruksi seperti aritmatika, sekuensial, logika, *timer*, *counter*. Didalam PLC juga terdapat analog I/O untuk menerima maupun mengirim sinyal analog. Kelebihan menggunakan PLC yaitu mudah dalam pengoperasian, mudah jika melakukan pengeditan ketika proses berjalan, serta efektif dalam mengawasi kontrol yang sedang besar. Sebagian besar industri, menggunakan PLC untuk dijadikan sistem kontrol. PLC yang digunakan peneliti adalah PLC Siemens S7-1200.

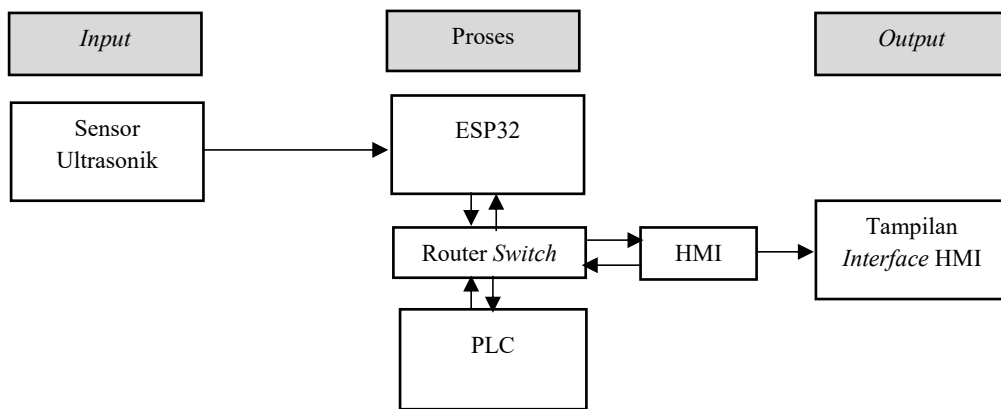
ESP32 merupakan mikrokontroler populer dan dikembangkan oleh Espressif System. ESP32 merupakan pengembangan dari ESP8266. Terdapat modul *Wi-Fi* dalam *chip* pada mikrokontroler sehingga dapat diaplikasikan untuk sistem *Internet of Things* (IoT) [7]. Pemrograman ESP32 dapat dilakukan pada berbagai macam environment seperti Arduino IDE.

HMI digunakan untuk memudahkan kontrol operator di *site*. HMI berisi tampilan antarmuka (*interface*) sebagai perantara antara manusia dengan mesin maupun sistem. Selain itu, HMI dapat diakses kapan saja untuk berbagai tujuan, seperti kegagalan mesin, hingga menampilkan suatu proses. Penggunaan HMI harus terintegrasi dengan *controller* yang digunakan seperti pada PLC. Pada penelitian ini HMI yang digunakan yaitu Weintek MT-8072IP.

Komunikasi Modbus TCP/IP adalah sebuah protokol komunikasi yang digunakan untuk mentransfer data antara perangkat elektronik melalui jaringan TCP/IP. *Modbus* TCP/IP menggunakan model komunikasi *client/server*, dimana *master* bertindak sebagai *client* sementara *slave* bertindak sebagai *server* [8]. Protokol Modbus TCP/IP mendukung beberapa jenis fungsi, seperti membaca dan menulis data (input dan output), membaca status register, mengendalikan perangkat, dan lainnya. Dengan sistem *Modbus* protokol TCP/IP dapat memungkinkan untuk komunikasi antar *device* dan menunjukkan data yang *real time* [9]. Dalam pengembangan aplikasi Modbus TCP/IP, perlu untuk memperhatikan konfigurasi perangkat, pengaturan alamat IP dan port, serta pemrograman untuk mengimplementasikan fungsi Modbus yang sesuai.

## 2.1 Perancangan Sistem

Perancangan sistem digunakan sebagai landasan dan juga untuk mempermudah dalam proses integrasi perangkat yang telah direncanakan.



Gambar 1 : Perancangan Sistem

Gambar 1 adalah model rancangan sistem pada penelitian ini. Untuk pengambilan data digunakan sensor ultrasonik yang nantinya akan dibaca oleh ESP32. Dari ESP32 akan mengirimkan data pembacaan sensor ke PLC Siemens S7-1200. Antara PLC Siemens S7-1200 dan ESP32 digunakan protokol komunikasi modbus TCP/IP. Dimana dalam hal ini ESP32 bertindak sebagai client sedangkan PLC sebagai server. Dari PLC akan diintegrasikan dengan HMI sehingga data pembacaan sensor yang ditampilkan HMI nantinya akan dibandingkan dengan data pembacaan sensor yang dibaca oleh ESP32. Dari sinilah akan dilakukan pengamatan waktu respon pengiriman data dan juga keberhasilan dalam pengiriman data.



## 2.2 Perancangan Program Komunikasi *Modbus TCP/IP* PLC dan *ESP32*

Pada penelitian ini, digunakan 1 alamat register untuk ultrasonik sebagai media pembacaan data sensor yang nantinya akan dikirim dari *ESP32* ke *PLC*.



Gambar 2 : (a) Inisialisasi Library, (b) Setup Koneksi WiFi

Langkah pertama yaitu inisialisasi *library* dari *wifi esp32* dan *Modbus TCP/IP* yang akan digunakan. Setelah itu menentukan variabel register yang dibutuhkan dengan tipe data *integer*. Lalu menambahkan *IP address* sesuai dengan *IP address* dari *PLC* yang bertindak sebagai *server*. Kemudian mengaktifkan variabel "mb" yang digunakan sebagai objek *modbus IP*. Kemudian mendefinisikan variabel data yang akan dikirim atau yang akan menerima data dari register. Data yang dikomunikasikan menggunakan tipe data *unsigned integer* dengan panjang 16 bit.

Pada *void setup*, dilakukan *setup* koneksi *wifi* dengan mengaktifkan komunikasi serial terlebih dahulu dengan *baudrate* 115200 seperti yang ditunjukkan pada Gambar 6. Lalu menghubungkan *ESP32* dengan *access point*. Apabila *Wi-Fi* belum tersambung dengan *access point* maka akan menampilkan "NOT CONNECTED" dan jika tersambung akan menampilkan "WiFi Connected" dan *IP* lokal yang dimiliki *ESP32*. Langkah terakhir yaitu mendefinisikan posisi *modbus* sebagai *client* dengan variabel "mb".

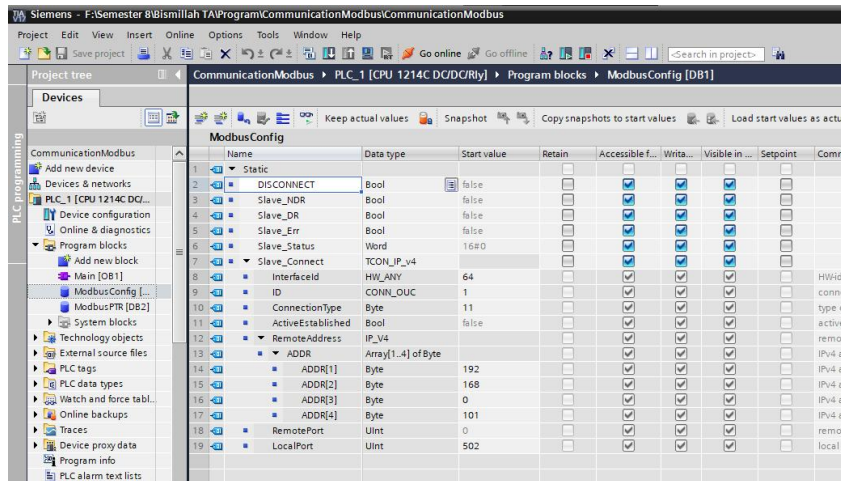


Gambar 3 : Program Transfer Data

Pada *void loop* dilakukan pengujian koneksi ke *PLC*. Jika belum terhubung maka program akan mencoba menghubungkan ke *device server*. Apabila sudah terhubung maka akan mengirimkan data ke *PLC*. Urutan program pada pengiriman data ini adalah *remote (IP address server)*, *REG (data register)*, *res (data yang dikirim)*. Sedangkan

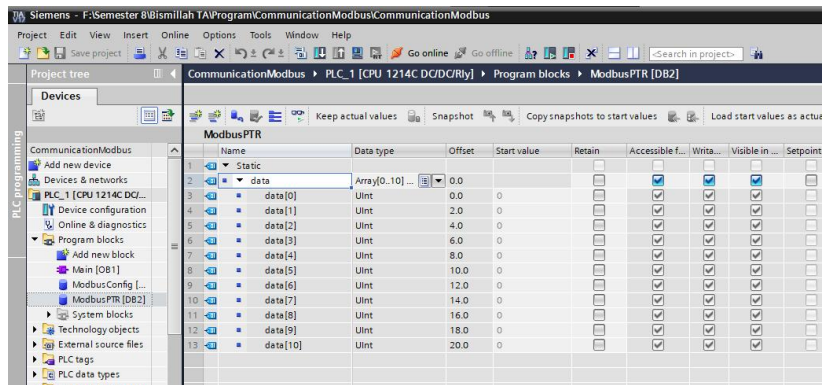


data pembacaan dari sisi PLC memiliki urutan yang sama dengan urutan program pengiriman. Langkah terakhir yaitu menampilkan data pada komunikasi serial untuk dapat dimonitor.



Gambar 4 : Data Block Config

Pada sisi PLC yang bertindak sebagai *server*, langkah pertama yaitu membuat *data block config*. *Data block* ini digunakan untuk menyimpan beberapa informasi mengenai koneksi modbus yang digunakan. *data block config* ini dibuat dengan menyesuaikan kebutuhan komunikasi modbus. Pada tag “Slave\_Connect” menggunakan tipe data TCON\_IP\_v4 yang nantinya akan otomatis membuat tag – tag yang dibutuhkan untuk komunikasi modbus. Pada bagian data *array* ADDR[1] hingga ADDR[4] merupakan IP *address* yang dimiliki ESP32.

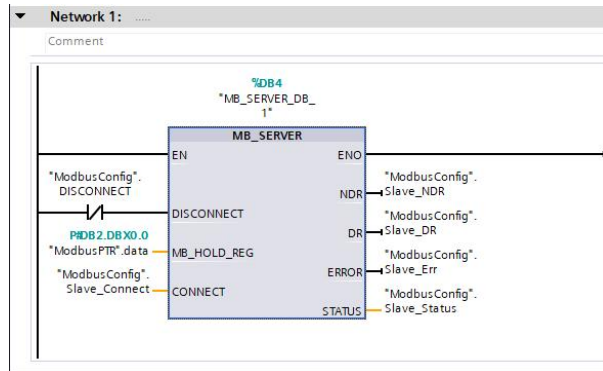


Gambar 5 : Holding Register Modbus

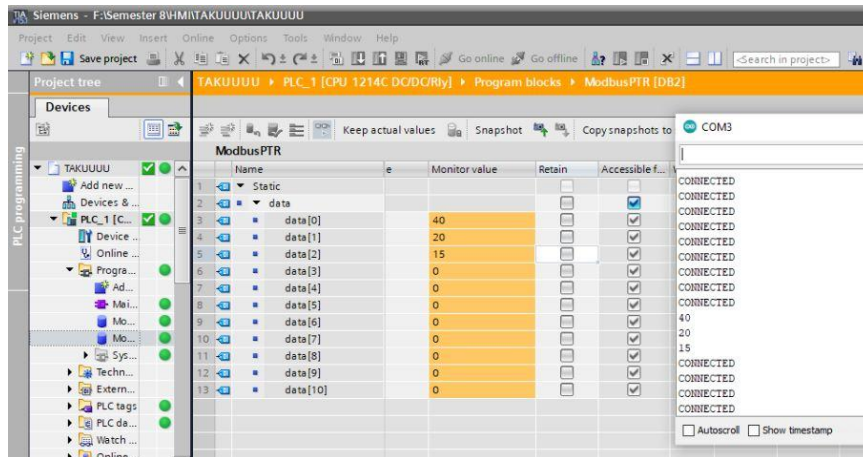
Langkah selanjutnya kembali membuat *data block* yang digunakan untuk menyimpan data sesuai dengan urutan register seperti yang terlihat pada Gambar 9. Data tersebut berupa data *array* dengan tipe data *unsigned* integer 16 bit.

Pada *software* TIA Portal telah menyediakan *function block* berupa MB\_Server yang digunakan untuk komunikasi modbus seperti pada Gambar 10. Lalu tag data yang telah dibuat pada *data block config* ditempatkan pada masing-masing cabang dari *function block* MB\_Server. Pada Gambar 11 terlihat tampilan *serial monitor* serta TIA Portal yang telah terintegrasi. Pada ujicoba tersebut digunakan nilai 40, 20, dan 15 untuk dikirim dari ESP32 ke PLC. Dapat terlihat bahwa nilai yang dikirim dari ESP32 dapat terbaca pada tampilan monitor TIA Portal.





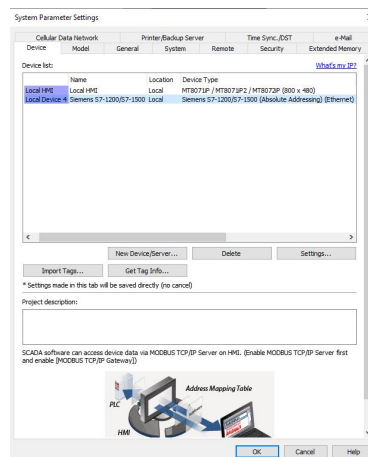
Gambar 6 : Function Block "MB\_Server"



Gambar 7 : Tampilan Komunikasi ESP32 dengan PLC Siemens S7-1200

### 2.3 Perancangan Pembuatan HMI

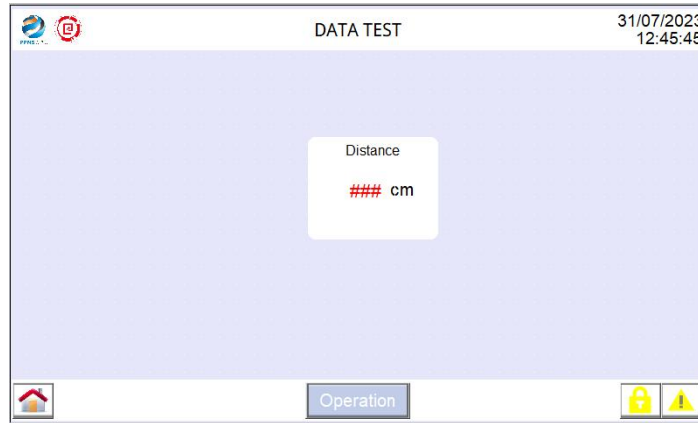
Dalam pembuatan tampilan *interface* pada HMI dapat dilakukan pada *software* Easy Builder Pro. *Software* ini dikembangkan oleh Weintek sendiri dan bersifat *open source*. Easy builder merupakan pemrograman berbagai jenis grafis *interface* mulai dari *function key*, *button*, *alarm log*, dan masih banyak lagi. Disamping itu, easy builder juga dapat terintegrasi dengan berbagai macam PLC seperti *siemens*, *allen-bradley*, *mitsubishi*, *ABB*, dan masih banyak lagi.



Gambar 8 : Konfigurasi HMI



Pada Gambar 8 dapat dilihat konfigurasi pada HMI Weintek MT8071-IP. Untuk menambahkan *device* pilih *new device* pada *system parameters* dan tentukan *device* yang sesuai. *Device* yang digunakan pada penelitian kali ini yaitu Siemens S7-1200/S7-1500. Pada opsi *system parameters* juga dilakukan pengaturan IP yang sesuai dengan IP *address* dari PLC yang kita gunakan. Nantinya, HMI akan terhubung dengan PLC Siemens S7-1200 menggunakan *ethernet*. Setelah dilakukan konfigurasi pada *software* HMI, selanjutnya adalah membuat *interface* yang digunakan untuk monitoring hasil pembacaan sensor ultrasonik. Gambar 9 merupakan tampilan yang digunakan pada penelitian kali ini.



Gambar 9 : Software Easy Builder V6.08

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Pengujian Sensor Ultrasonik HC-SR04

Pengujian dilakukan untuk memastikan tingkat keakuratan dari sensor ultrasonik. Sebagai media pembanding digunakan mistar untuk validasi dari nilai pembacaan yang dilakukan oleh ultrasonik. Hasil pengujian sensor ultrasonik dilakukan dengan sampel pengujian sebanyak 10 sampel. Dari pengujian tersebut akan didapatkan persentase *error* yang dapat dihitung melalui persamaan (1). Hasil pengujian sensor ultrasonik HC-SR04 ditunjukkan pada Tabel I.

$$\%Error = \left| \frac{\text{Nilai Sebenarnya} - \text{Nilai Sensor}}{\text{Nilai Sebenarnya}} \right| \times 100\% \tag{1}$$

TABEL I : PENGUJIAN ULTRASONIK HC-SR04

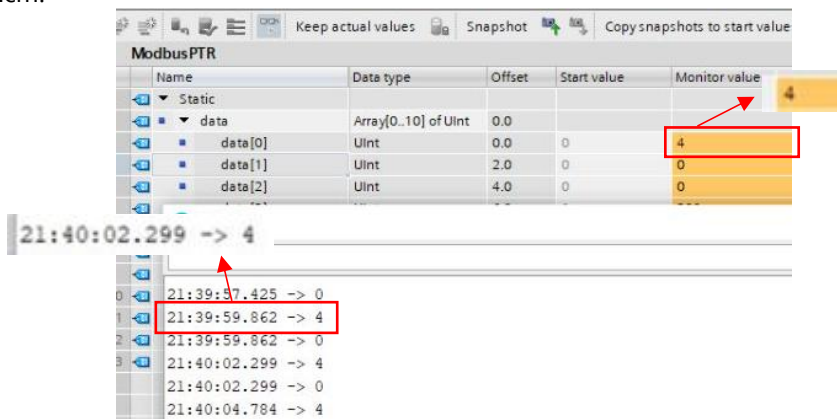
No	Ultrasonik HC-SR04 (cm)	Mistar (cm)	Persentase Error (%)
1	21	22	4.54
2	22	22	0
3	22	23	4.34
4	23	23	0
5	23	24	4.16
6	24	24	0
7	24	25	4
8	25	25	0
9	25	26	3.84
10	26	26	0
Persentase <i>error</i> rata-rata			2.088



Dapat dilihat pada Tabel I pengujian ultrasonik didapatkan presentase error rata-rata 2.088% yang terbilang cukup kecil. Untuk memastikan data yang dibaca valid maka diperlukan validasi serta kalibrasi sensor yang hasilnya telah ditunjukkan pada Tabel I. Pembacaan ultrasonik nantinya yang akan dijadikan media ujicoba dalam komunikasi *modbus tcp/ip*. Data yang terbaca pada ultrasonik akan dikirim dari ESP32 menuju PLC Siemens S7-1200 dan akan ditampilkan pada layar *interface* HMI.

**3.2 Komunikasi ESP32, PLC Siemens S7-1200, dan Weintek MT-8072IP**

Selanjutnya akan dilakukan integrasi ketiga perangkat yaitu ESP32, PLC dan HMI untuk menguji apakah komunikasi data yang dilakukan berhasil dikirim dan dapat ditampilkan pada HMI. Untuk data yang akan dikirim adalah pembacaan data sensor ultrasonik HC-SR04. Pembacaan data dilakukan dengan membandingkan pembacaan dari serial monitor Arduino IDE dengan data *block* dari Tia Portal sendiri, serta mengamati di sisi HMI apakah data dapat terkirim dengan baik. Sampel data yang dikirim berupa 5 data random dengan data 4cm, 6cm, 10cm, 21cm, dan 42cm.



Gambar 10 : Pengujian Komunikasi Modbus TCP/IP

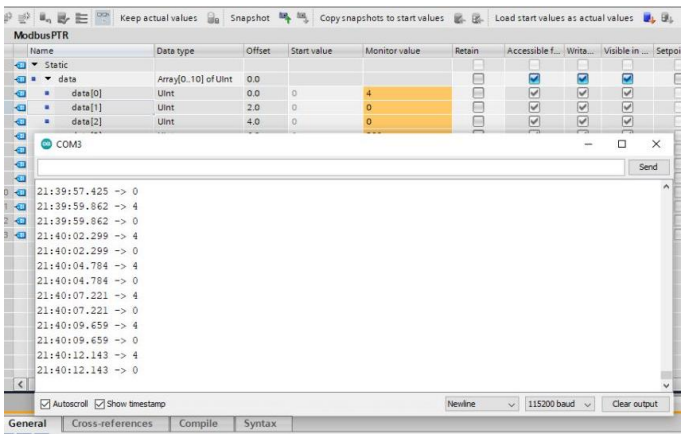
TABEL II : PENGUJIAN JEDA WAKTU KOMUNIKASI MODBUS TCP/IP

No	Pembacaan Data ESP32	Pembacaan Data PLC	Jeda Waktu (s)
1	4	4	0.97
2	6	6	1.01
3	10	10	0.98
4	21	21	1
5	42	42	1.03
	Rata-rata		0.998

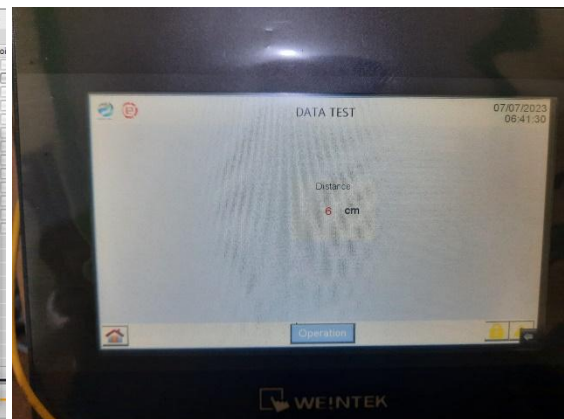
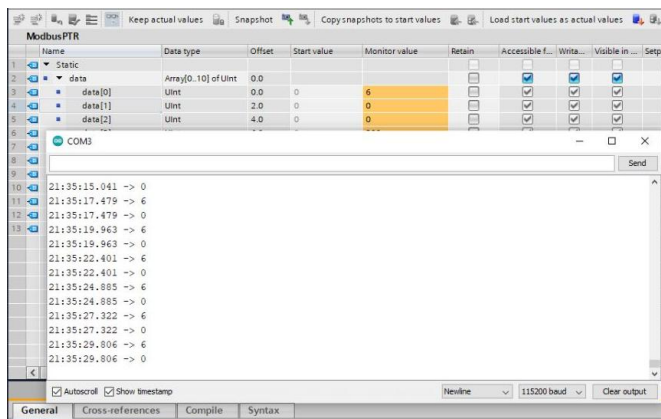
Berdasarkan hasil pengujian yang dilakukan, pengiriman data dari ESP32 ke PLC berhasil terkirim dengan baik namun terdapat jeda waktu (*delay*) pengiriman yang dapat dilihat pada Tabel II. Pembacaan data dari ESP32 dikirim melalui protokol *Modbus TCP/IP* kepada PLC telah berhasil dan pembacaan data sensor akan ditampilkan pada HMI yang ditunjukkan pada gambar dibawah ini.



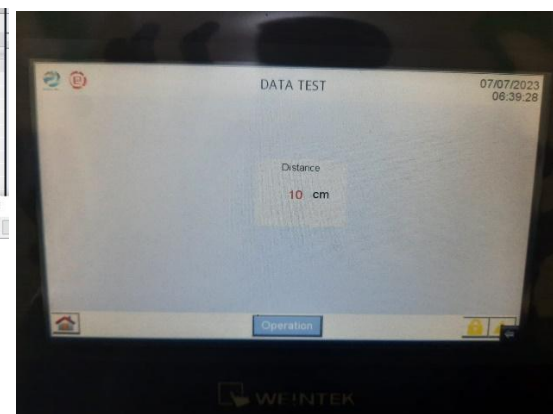
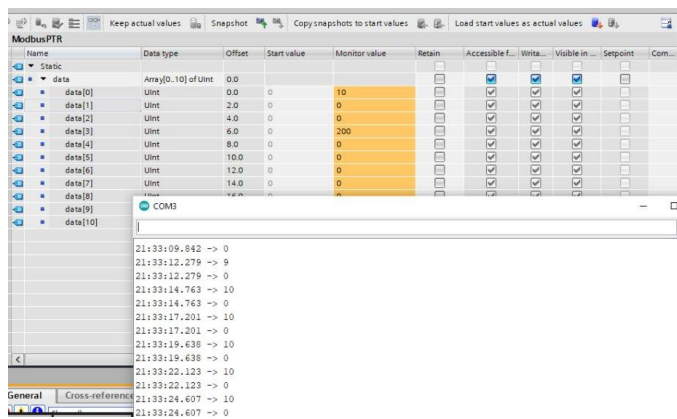




Gambar 11 : Tampilan Pengiriman Data 4cm

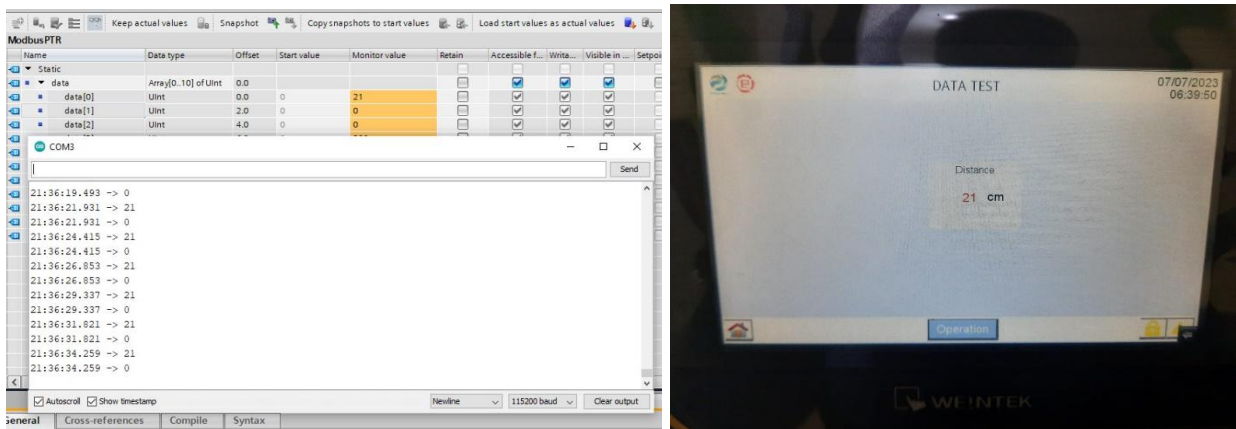


Gambar 12 : Tampilan Pengiriman Data 6cm

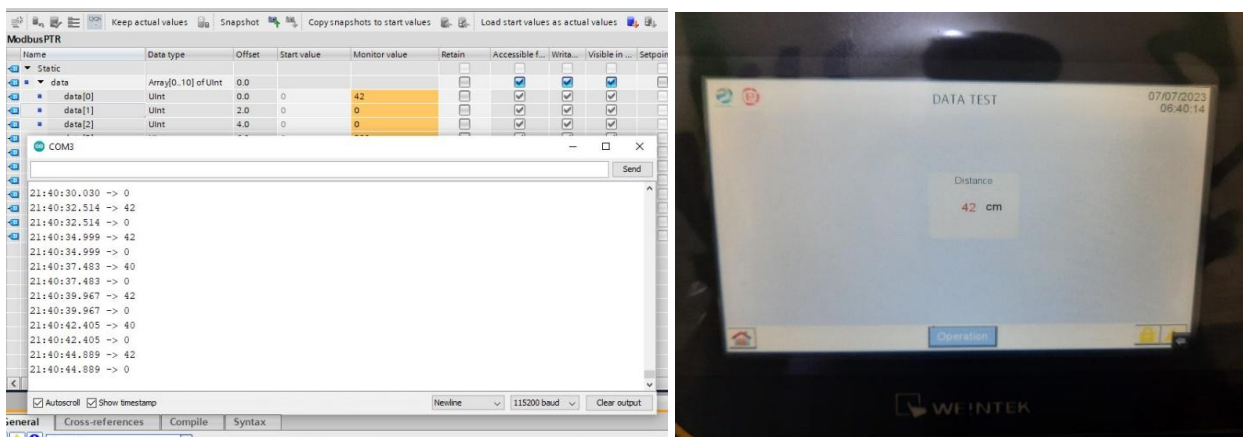


Gambar 13 : Tampilan Pengiriman Data 10cm





Gambar 14 : Tampilan Pengiriman Data 21cm



Gambar 15 : Tampilan Pengiriman Data 42cm

Dari kelima ujicoba pengiriman data, semua data berhasil terkirim dengan baik. Mulai dari pembacaan data sensor yang dibaca oleh ESP32 berhasil dikirim ke PLC Siemens S7-1200 kemudian berhasil ditampilkan pada *interface* HMI. Dari ketiga pengujian yang telah dilakukan terdapat jeda waktu pengiriman data dari ESP32 menuju ke PLC. Jeda waktu bervariasi namun masih dalam rentan waktu yang berdekatan. Jeda waktu terbesar yaitu 1.03 detik dan jeda waktu terkecil yaitu 0.97 detik.

#### 4. KESIMPULAN

- Dari seluruh pengujian yang telah dilakukan untuk membaca data sensor pada ESP32, kemudian pengiriman data dari ESP32 menuju ke PLC Siemens S7-1200, serta menampilkan hasil pembacaan sensor pada HMI tidak terjadi kendala dan semua pembacaan sensor berhasil dibaca dan dikirim dengan baik.
- Berdasarkan hasil pengujian, ada waktu tunda atau jeda waktu pada pembacaan antara data yang dibaca oleh ESP32 dengan data yang tampil pada PLC. Jeda waktu bervariasi namun masih dalam rentang yang berdekatan. Jeda waktu terbesar yaitu 1.03 detik dan yang terkecil yaitu 0.97 detik.

#### DAFTAR PUSTAKA

- [1] M. Rizal, M. S. Hadis, R. Angriawan, and A. Arifin, "EVALUASI KINERJA BLUETOOTH PADA MODUL ESP32 DI LINGKUNGAN LINE OF SIGHT," *JESSI*, vol. 1, May 2020.
- [2] R. Amaliawati, A. Surya Wibowo, and M. A. Murti, "PERANCANGAN SISTEM KOMUNIKASI PLC DENGAN RASPBERRY PI VIA PROTOKOL MODBUS DESIGN OF PLC COMMUNICATION SYSTEMS WITH RASPBERRY PI VIA MODBUS PROTOCOL," *e-Proceeding of Engineering*, vol. 3, Dec. 2020.



- [3] I. Suharto and W. Heryawan, "Rancang Bangun Jaringan Komunikasi PLC Menggunakan Sistem Wlan untuk Memonitor Proses Kontrol Berbasis Human Machine Interface SCADA INFO U," Pontianak, Dec. 2019.
- [4] Teng Jonathan et al, *SISTEM PEMBACAAN DATA POWER METER DENGAN KOMUNIKASI MODBUS SECARA TERPUSAT*. Surabaya, 2019. Accessed: Jul. 05, 2023. [Online]. Available: <https://repository.petra.ac.id/id/eprint/18350>
- [5] F. A. Putri, S. W. Jadmiko, and S. Yahya, "Rancang Bangun Internet of Things (IoT) pada Pengendalian Tegangan Simulator Input Output Berbasis PLC-ESP32," pp. 4–5, Aug. 2021, [Online]. Available: <http://things.ubidots.com/api/v1.6/devices/esp32>,
- [6] D. Yuhendri, "Penggunaan PLC Sebagai Pengontrol Peralatan Building Automatis," *Journal of Electrical Technology*, vol. 3, no. 3, Oct. 2018.
- [7] H. Kusumah and R. A. Pradana, "PENERAPAN TRAINER INTERFACING MIKROKONTROLER DAN INTERNET OF THINGS BERBASIS ESP32 PADA MATA KULIAH INTERFACING," *CERITA*, vol. 5, Aug. 2019.
- [8] T. Tosin, "Perancangan dan Implementasi Komunikasi RS-485 Menggunakan Protokol Modbus RTU dan Modbus TCP Pada Sistem Pick-By-Light," *Komputika: Jurnal Sistem Komputer*, vol. 10, no. 1, pp. 85–91, Mar. 2021, doi: 10.34010/komputika.v10i1.3557.
- [9] B. Yoma Pratama and dan Efi Anisa, "Halaman 43-52, Sekolah Tinggi Teknologi Duta Bangsa," *Kalibaru Medan Satria*, vol. 5, 2023.
- [10] M. I. Hakiki, U. Darusalam, and N. D. Nathasia, "Konfigurasi Arduino IDE Untuk Monitoring Pendeteksi Suhu dan Kelembapan Pada Ruang Data Center Menggunakan Sensor DHT11," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 4, no. 1, p. 150, Jan. 2020, doi: 10.30865/mib.v4i1.1876.

