

# Integrasi LabVIEW Dengan Arduino Melalui Modbus RTU Untuk Sistem Monitoring dan Kontrol

Syafriyadi Nor<sup>1</sup>, Sarifudin<sup>2</sup>

Jurusan Teknik Elektro, Program Studi D3 Elektronika, Politeknik Negeri Banjarmasin, Banjarmasin  
Jl. Brigjen H. Basri (Komplek ULM) Kayutangi, Banjarmasin 70123

<sup>1</sup>syafriyadi.nor@poliban.ac.id

<sup>2</sup>sarif@poliban.ac.id

**Intisari** — Penelitian ini bertujuan untuk mengintegrasikan perangkat lunak LabVIEW dengan mikrokontroler Arduino menggunakan protokol Modbus RTU untuk pengembangan sistem monitoring dan kontrol. LabVIEW digunakan sebagai antarmuka pengguna yang memungkinkan pemantauan *real-time* dan pengendalian perangkat keras Arduino, seperti sensor DHT11 dan Relay, secara efisien melalui jaringan Modbus. Metodologi penelitian mencakup pengaturan dan konfigurasi komunikasi Modbus RTU antara LabVIEW dan Arduino, pengembangan antarmuka pengguna untuk visualisasi data sensor DHT11. Tujuan dari penelitian ini adalah menguji fungsi input register pada Modbus RTU untuk membaca suhu dan kelembaban dari sensor DHT11, menguji fungsi *coil* untuk mengaktifkan dan menonaktifkan relay, serta menguji fungsi input status untuk membaca status tombol saat ditekan. Hasil penelitian ini menunjukkan bahwa data dari sensor DHT11 dapat ditransmisikan secara akurat ke LabVIEW, dengan pembacaan sensor yang konsisten dan akurat baik untuk suhu maupun kelembaban. Status tombol berubah sesuai dengan interaksi fisik yang dilakukan, dan indikator lampu pada antarmuka LabVIEW merekam perubahan status dengan akurat. Meskipun terdapat fluktuasi dalam waktu respon yang bisa dipengaruhi oleh beban relay, kondisi jaringan komunikasi, atau performa perangkat keras, respons yang tercatat masih dalam rentang yang dapat diterima. Tidak ada anomali atau *delay* yang signifikan dalam pembacaan status *input*, menunjukkan keandalan sistem dalam merespons interaksi pengguna dengan tombol dan *feedback* yang diberikan melalui antarmuka LabVIEW. Sistem yang dikembangkan mampu memberikan monitoring dan kontrol yang efisien, serta dapat diandalkan untuk aplikasi industri maupun penelitian.

**Kata kunci** — LabVIEW, Arduino, Modbus, RTU, Monitoring, Kontrol.

**Abstract** — This study aims to integrate LabVIEW software with an Arduino microcontroller using the Modbus RTU protocol for monitoring and control system development. LabVIEW is used as a user interface that allows real-time monitoring and control of Arduino hardware, such as DHT11 sensors and relays, efficiently through a Modbus network. The research methodology includes setting up and configuring Modbus RTU communication between LabVIEW and Arduino, and developing a user interface for visualization of DHT11 sensor data. The purpose of this study is to test the function of the input register on Modbus RTU to read temperature and humidity from the DHT11 sensor, test the function of the coil to activate and deactivate the relay, and test the function of the status input to read the status of the push button. The results of this study indicate that data from the DHT11 sensor can be transmitted accurately to LabVIEW, with consistent and accurate sensor readings for both temperature and humidity. The button status changes according to the physical interaction performed, and the light indicator on the LabVIEW interface records status changes accurately. Although there are fluctuations in response time that can be influenced by relay load, communication network conditions, or hardware performance, the recorded response is still within an acceptable range. There are no significant anomalies or delays in reading the input status, indicating the reliability of the system in responding to user interaction with buttons and feedback provided through the LabVIEW interface. The developed system is able to provide efficient monitoring and control, and is reliable for industrial and research applications.

**Keywords**— LabVIEW, Arduino, Modbus, RTU, Monitoring, Control.

## I. PENDAHULUAN

Dalam era industri 4.0, kebutuhan akan sistem monitoring dan kontrol yang efisien dan terintegrasi semakin meningkat. Teknologi yang mampu memberikan pemantauan dan pengendalian *real-time* terhadap berbagai parameter proses industri menjadi sangat penting untuk meningkatkan efisiensi operasional, mengurangi *downtime*, dan memastikan kualitas produk yang konsisten.

LabVIEW, sebagai *platform* pengembangan sistem instrumentasi dan kontrol berbasis grafis yang populer [1], menawarkan berbagai keunggulan dalam hal fleksibilitas dan kemudahan penggunaan. Namun, penggunaannya seringkali memerlukan integrasi dengan perangkat keras eksternal untuk mendapatkan data dari sensor dan mengendalikan aktuator. Salah satu perangkat keras yang banyak digunakan dalam berbagai aplikasi industri dan penelitian adalah Arduino. Arduino dikenal karena sifatnya yang *open-source*, murah, dan mudah diprogram, sehingga menjadi pilihan yang tepat untuk prototipe dan aplikasi skala kecil hingga menengah.

Modbus merupakan salah satu protokol komunikasi yang sering digunakan dalam industri untuk menghubungkan berbagai perangkat. Varian Modbus RTU (*Remote Terminal Unit*) adalah protokol komunikasi serial yang umum digunakan dalam industri untuk menghubungkan berbagai perangkat dan memungkinkan komunikasi data yang andal [2]. Modbus RTU menawarkan komunikasi yang dapat diandalkan melalui jalur serial seperti RS-232 dan RS-485, yang dapat mentransmisikan data jarak jauh.

Penelitian ini bertujuan untuk mengintegrasikan LabVIEW dengan Arduino Nano melalui protokol Modbus RTU. Integrasi ini diharapkan dapat memanfaatkan keunggulan masing-masing teknologi untuk menciptakan sistem monitoring dan kontrol yang efisien dan handal. Dengan menggunakan LabVIEW, pengguna dapat membuat antarmuka pengguna yang intuitif dan kaya fitur untuk memantau dan mengendalikan sistem. Sementara itu, Arduino Nano akan bertindak sebagai

perangkat keras yang mengumpulkan data dari sensor dan mengontrol aktuator sesuai dengan perintah yang diterima dari LabVIEW melalui Modbus RTU. Adapun pengujian yang dilakukan adalah sebagai berikut:

1. menguji fungsi input register pada Modbus RTU, yaitu membaca suhu dan kelembaban dari sensor DHT11.
2. Menguji fungsi coil pada Modbus RTU, yaitu mengaktifkan dan menonaktifkan relay.
3. Menguji fungsi input status pada Modbus RTU, yaitu membaca status tombol saat ditekan

Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan sistem monitoring dan kontrol yang lebih terintegrasi, efisien, dan mudah digunakan, serta dapat diaplikasikan dalam berbagai bidang industri dan penelitian.

## II. PENELITIAN TERDAHULU

Banyak penelitian telah dilakukan terkait integrasi sistem monitoring dan kontrol menggunakan berbagai *platform* dan protokol komunikasi. Penelitian oleh [3] menggunakan LabVIEW untuk mengembangkan antarmuka pengguna yang memantau ketinggian air menggunakan sensor ultrasonik SRF05 yang terhubung ke Arduino Uno R3. Penelitian ini menunjukkan bahwa LabVIEW dapat secara efektif digunakan untuk membuat sistem monitoring yang *user-friendly* dengan memanfaatkan Arduino Uno R3 sebagai perangkat pengumpulan data. Penelitian oleh [4] membahas kegunaan perangkat lunak LabVIEW dalam aplikasi DAQ yang berkaitan dengan pengukuran suhu. Dalam penelitian oleh [5], LabVIEW digunakan untuk mengintegrasikan berbagai data dari berbagai sensor dan perangkat ke dalam satu sistem yang terpusat. Penelitian ini menunjukkan bahwa penggunaan LabVIEW untuk monitoring dan kontrol smart home menjadi lebih terintegrasi, efisien, dan mudah diakses. Penelitian oleh [6] mengembangkan sistem pemantauan menggunakan LabVIEW melalui perangkat DAQ USB. Penelitian ini

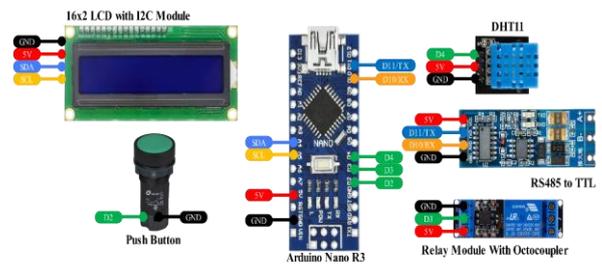
melakukan analisis singkat terhadap kinerja pemanas udara tenaga surya dengan bantuan LabVIEW, serta mengusulkan perubahan konstruktif untuk meningkatkan efisiensi termalnya. Penelitian oleh [7] mengintegrasikan Arduino dan LabVIEW untuk pemantauan arus dan tegangan menggunakan sensor INA219. Hasil penelitian menunjukkan bahwa data arus dan tegangan ditampilkan secara langsung pada antarmuka LabVIEW dan dapat diakses melalui web browser menggunakan jaringan lokal.

Penelitian-penelitian tersebut menunjukkan bahwa integrasi LabVIEW dengan Arduino melalui memiliki potensi besar dalam pengembangan sistem monitoring dan kontrol yang efektif dan efisien. Namun, belum banyak penelitian yang secara spesifik mengeksplorasi penggunaan Modbus RTU sebagai penghubung antara LabVIEW dan Arduino dalam konteks aplikasi monitoring dan kontrol yang lebih kompleks. Oleh karena itu, penelitian ini bertujuan untuk mengisi celah tersebut dengan mengembangkan dan menguji sistem yang menggabungkan keunggulan ketiga teknologi ini.

### III. METODE PENELITIAN

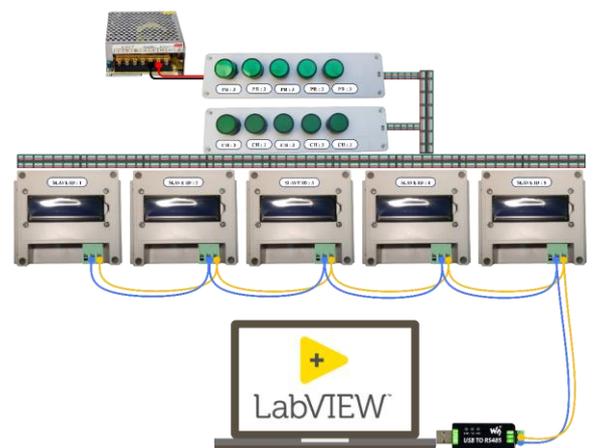
#### A. Perancangan Sistem

Penelitian ini menggunakan metode eksperimen untuk mengintegrasikan LabVIEW dengan Arduino melalui protokol Modbus RTU dalam sistem monitoring dan kontrol. Dalam penelitian ini, konfigurasi antar komponen diilustrasikan pada Gbr 1. Dari gambar tersebut, dapat dilihat bahwa terdapat Arduino Nano R3 sebagai mikrokontroler, modul *RS485 to TTL (Transistor-Transistor Logic)* untuk komunikasi serial, sensor DHT11 untuk mengukur suhu dan kelembaban, *Liquid Crystal Display (LCD)* untuk menampilkan data, modul relay untuk mengontrol lampu pilot 12V, dan *push button* sebagai *input* manual. Arduino Nano R3 mengumpulkan data dari sensor DHT11 dan mengirimkannya ke LabVIEW melalui modul *RS485 to TTL* menggunakan protokol Modbus RTU.



Gbr. 1 Konfigurasi Perangkat Keras (*Hardware*)

Arduino Nano R3 dihubungkan dengan modul RS485 to TTL melalui pin TX (*Transmit*) dan RX (*Receive*). Modul *RS485 to TTL (Transistor-Transistor Logic)* kemudian terhubung ke jaringan RS485 menggunakan konfigurasi *daisy chain* bersama dengan lima perangkat lain yang juga menggunakan modul RS485 to TTL pada Gbr 2. *USB to RS485 Converter* digunakan untuk menghubungkan komputer (yang menjalankan LabVIEW) dengan jaringan RS485. Konverter ini memastikan bahwa komputer dapat berkomunikasi dengan Arduino melalui protokol RS485. Sensor DHT11 dihubungkan ke Arduino Nano R3 untuk mengukur suhu dan kelembaban. Selain itu, LCD dihubungkan untuk menampilkan data secara lokal, dan relay terhubung untuk mengontrol lampu pilot 12V berdasarkan perintah dari LabVIEW.



Gbr. 2 Komunikasi Antar Perangkat dengan RS-485

Program Arduino Nano dirancang untuk melakukan beberapa fungsi utama dalam sistem monitoring dan kontrol. Pertama, Arduino akan membaca data suhu dan kelembaban dari sensor DHT11 yang terhubung.

Data yang terbaca kemudian akan ditampilkan pada LCD 16x2 untuk memudahkan pemantauan oleh pengguna. Selain itu, Arduino juga akan mengontrol relay berdasarkan perintah yang diterima melalui Modbus RTU menggunakan *library* ModbusRTUSlave [8]. Status relay yang dikendalikan (*ON* atau *OFF*) akan direkam dan disimpan dalam *register* Modbus untuk pelaporan dan kontrol lebih lanjut. Arduino juga dilengkapi dengan fungsi untuk membaca status *push button*, yang dapat digunakan sebagai input manual untuk mengambil keputusan tertentu dalam sistem

### B. Konfigurasi Modbus RTU

Tabel 1. Konfigurasi Modbus RTU antara Arduino dan LabVIEW

Parameter	Konfigurasi pada Arduino dan LabVIEW
<i>Baud Rate</i>	57600
<i>Data Bits</i>	8
<i>Parity</i>	<i>None</i>
<i>Stop Bits</i>	1
<i>Mode</i>	<i>RS-485 Half Duplex</i>

Tabel 1 menunjukkan parameter-parameter penting yang harus disesuaikan baik pada sisi Arduino maupun LabVIEW untuk memastikan komunikasi Modbus RTU berjalan dengan lancar dan efektif. Memastikan bahwa kedua sistem menggunakan *baud rate*, *data bits*, *parity*, *stop bits*, untuk menghindari kesalahan komunikasi dan memastikan data dikirim dan diterima dengan benar. *Mode RS-485 Half Duplex* dipilih karena mendukung komunikasi yang reliabel di lingkungan industri dan jarak yang lebih jauh dibandingkan dengan mode serial standar.

### C. Pengaturan LabVIEW

Membuat VI (*Virtual Instrument*) yang menggunakan LabVIEW Modbus API (*Application Programming Interface*) untuk mengakses data dari Arduino. API ini menyediakan antarmuka yang memungkinkan pengguna LabVIEW untuk berkomunikasi dengan perangkat yang mendukung Modbus, baik sebagai *Master* (pemantau dan pengendali) atau sebagai

*Slave* (perangkat yang diatur dan dimonitor) [9].

Menampilkan data suhu (*temperature*) dan kelembaban (*humidity*) yang dibaca dari sensor DHT11 pada Arduino melalui *Read Input Registers*, kemudian menampilkannya secara *real-time* dalam bentuk grafik gelombang (*waveform chart*) pada antarmuka LabVIEW. Ini memungkinkan pemantauan kontinu terhadap perubahan suhu dan kelembaban secara visual. Sedangkan status *input* digital yang dibaca dari *push button* pada Arduino melalui *Read Discrete Inputs*. Informasi ini ditampilkan pada LabVIEW untuk memberikan indikasi visual apakah input tersebut dalam keadaan aktif (*high*) atau tidak aktif (*low*). Selain itu, status *coil* (*Write Single Coil*) pada antarmuka LabVIEW berupa *toggle switch* yang digunakan untuk mengaktifkan atau menonaktifkan *coil* (Modul Relay) pada Arduino, sehingga memungkinkan kontrol terhadap perangkat yang terhubung ke *coil* tersebut.

## IV. HASIL DAN PEMBAHASAN

### A. Hasil Pengujian Modbus RTU

Pengujian komunikasi Modbus RTU dari Arduino ke komputer dilakukan dengan menggunakan aplikasi Modbus Poll [10] yang terhubung melalui konverter USB to RS485, memastikan integrasi yang efektif antara perangkat keras Arduino dan perangkat lunak komputer untuk pertukaran data yang stabil dan termonitor dengan baik.

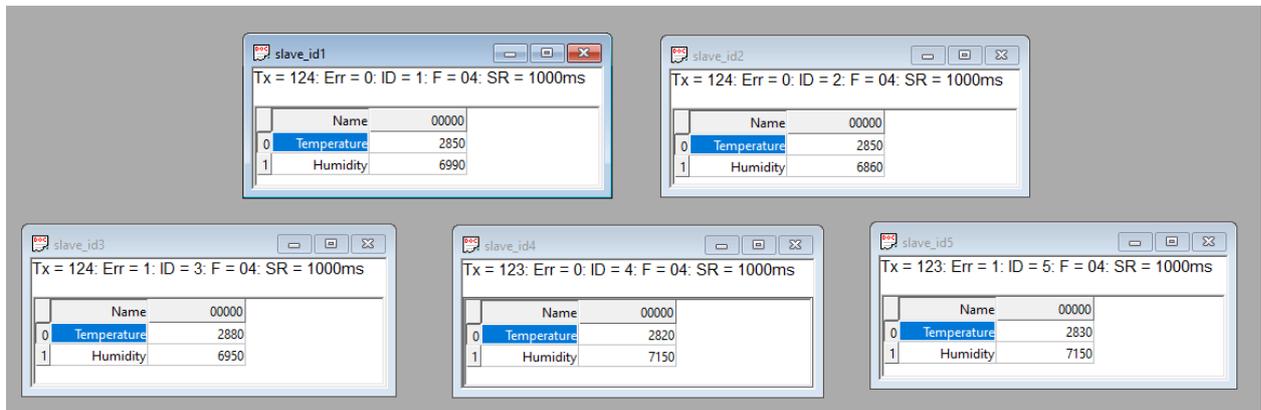
Pengujian komunikasi antara Modbus Poll dan Arduino untuk memastikan data dapat ditransfer dengan benar melalui Modbus RTU. Memastikan data yang ditampilkan di LCD sesuai dengan yang dibaca oleh sensor DHT11 di Arduino. Modbus pada dasarnya adalah protokol berbasis register yang dirancang untuk mentransmisikan bilangan bulat 16-bit (angka bulat dari 0 hingga 65535). Ini berarti bahwa untuk mengirim nilai desimal pada data "*Temperature*" dan "*Humidity*", harus diubah menjadi bentuk bilangan bulat dengan dikali dengan faktor 100. Konversi nilai desimal ke bilangan bulat sebelum pengiriman melalui Modbus dan kemudian mengonversinya kembali setelah

penerimaan adalah praktik umum untuk menjaga kompatibilitas, kesederhanaan, dan efisiensi protokol Modbus yang dirancang untuk bekerja dengan bilangan bulat 16-bit.

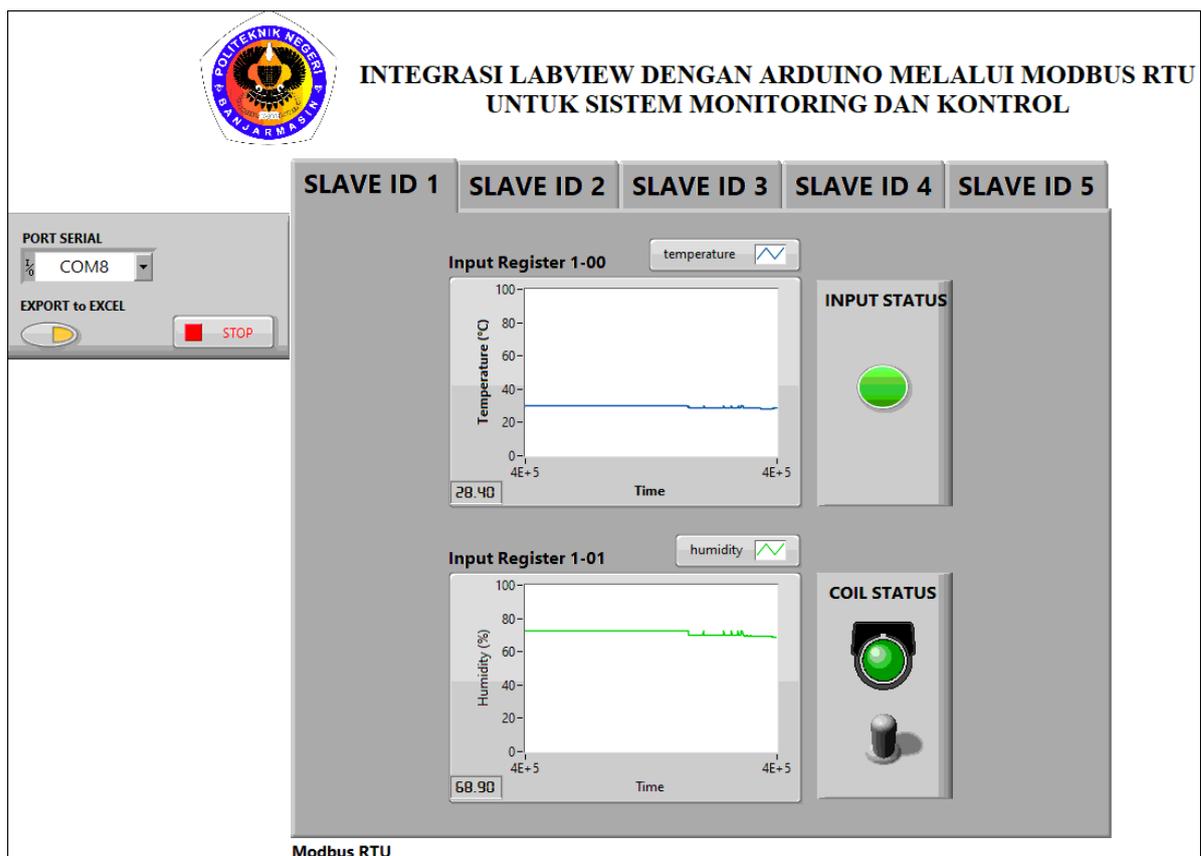
### B. Hasil Tampilan Antarmuka LabVIEW

Tampilan front panel LabVIEW untuk menampilkan data dari setiap *slave*

menggunakan *Tab Control* pada LabVIEW. Kontainer tersebut adalah sebuah elemen antarmuka grafis yang memungkinkan pengguna untuk mengatur dan mengelompokkan berbagai komponen atau indikator ke dalam beberapa tab yang berbeda. Setiap tab dalam *Tab Control*



Gbr. 3 Tampilan Antarmuka Modbus Poll



Gbr. 4 Tampilan Front panel pada LabVIEW

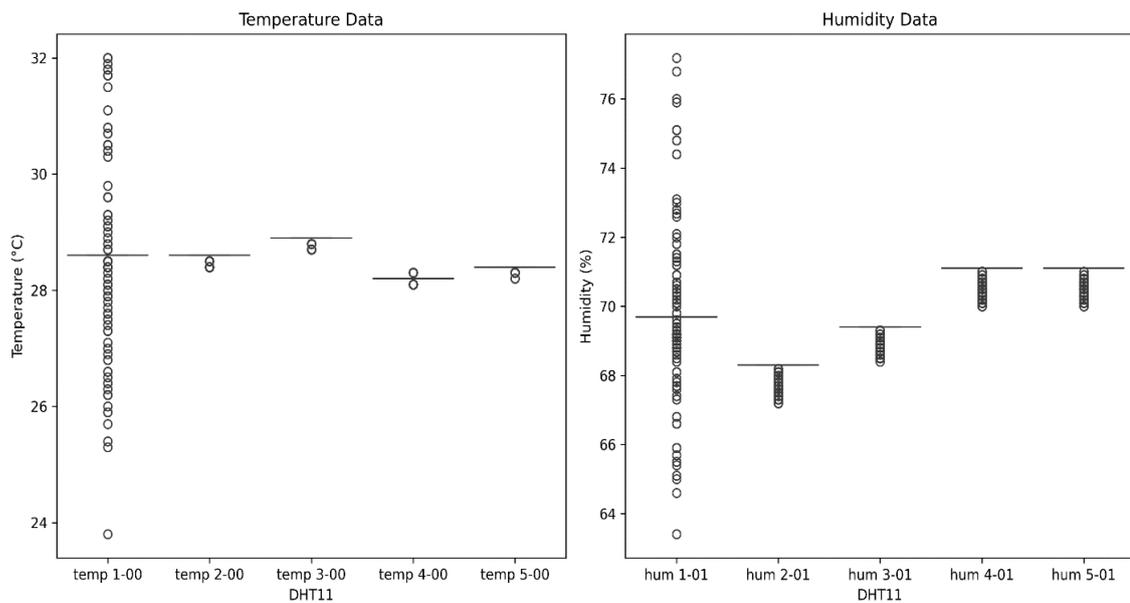
digunakan untuk satu Slave ID, menampilkan data *temperature* dan *humidity* menggunakan *waveform chart*, menampilkan *input status* dengan *Read Discrete Inputs*, dan mengontrol status *coil* dengan *Write Single*

*Coil*. Ini membantu dalam menyederhanakan antarmuka pengguna dan meningkatkan navigasi serta efisiensi visual.

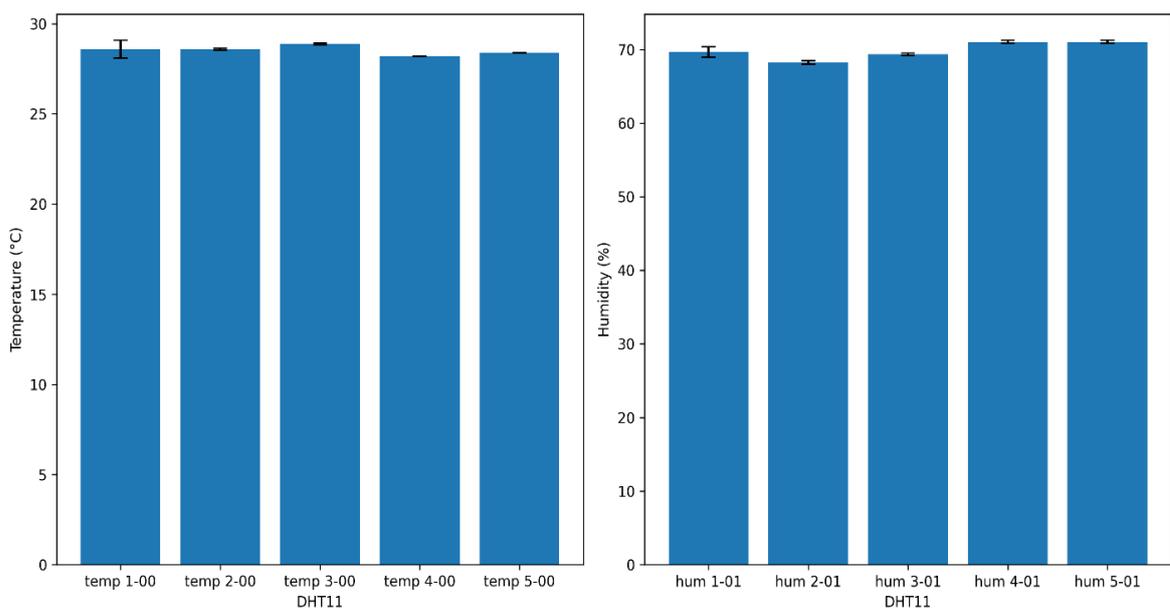
### C. Hasil Pembacaan Sensor DHT11

Data berasal dari lima sensor DHT11 pada setiap perangkat yang dihubungkan melalui Arduino dan LabVIEW melalui Modbus RTU. Setiap sensor memberikan pembacaan suhu (temp 1-00, temp 2-00, temp 3-00, temp 4-00, temp 5-00) dan kelembapan (hum 1-01, hum 2-01, hum 3-01, hum 4-01, hum 5-01) pada interval waktu yang sama. Angka yang disebutkan di depan kata "temp" dan "hum" mengacu pada label ID slave dan alamat memori pada protokol Modbus RTU. Ini

mengindikasikan bahwa setiap sensor memiliki ID slave tertentu dan alamat memori yang ditetapkan di dalam fungsi kode Modbus *Read Input Register* (04), dimulai dari alamat awal 0 dengan jumlah 2 alamat. Dataset ini terdiri dari 10 kolom yang masing-masing mewakili pembacaan dari lima sensor DHT11 pada masing-masing perangkat. Data dikumpulkan dan disimpan dalam format excel menggunakan fungsi *"Write To Measurement File"*.



Gbr. 5 Visualisasi Data Suhu dan Kelembaban



Gbr. 6 Perbandingan Suhu dan Kelembaban

Data pada Gbr 6 menunjukkan bahwa pembacaan suhu sangat konsisten di antara semua sensor DHT11, dengan rata-rata pembacaan suhu 28.53°C dan standar deviasi sebesar 0.32°C yang menunjukkan stabilitas suhu di lingkungan yang dipantau. Sementara itu, rata-rata kelembapan mencapai 69.86%. Standar deviasi kelembapan sebesar 1.13% menunjukkan variasi yang sedikit lebih besar dibandingkan dengan suhu, tetapi masih dalam batas yang terukur. Secara keseluruhan, pembacaan suhu dan kelembapan dari semua sensor DHT11 menunjukkan pengukuran yang stabil dengan variasi yang terukur.

#### D. Hasil Pembacaan Status dengan Push Button

Tabel 2. Hasil Pengujian Pembacaan Input Status pada LabVIEW

Slave	Status Input 1	Status Tombol	Status Input 2	Status Tombol
1	1 (Aktif)	Ditekan	0 (Tidak Aktif)	Tidak Ditekan
2	0 (Tidak Aktif)	Tidak Ditekan	1 (Aktif)	Ditekan
3	1 (Aktif)	Ditekan	1 (Aktif)	Ditekan
4	0 (Tidak Aktif)	Tidak Ditekan	0 (Tidak Aktif)	Tidak Ditekan
5	1 (Aktif)	Ditekan	0 (Tidak Aktif)	Tidak Ditekan

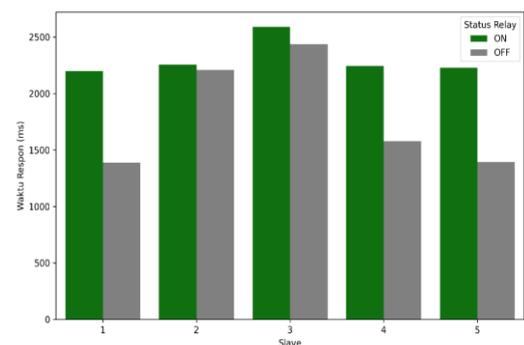
Setiap kolom dalam tabel 1 menunjukkan status dari dua input yang berbeda pada masing-masing *slave*. Status input dapat diinterpretasikan sesuai dengan konteks aplikasi atau sistem yang digunakan. Biasanya, status "Aktif" (1) dan "Tidak Aktif" (0) mengacu pada keadaan atau peristiwa tertentu yang diukur atau terdeteksi oleh perangkat sensor atau kontrol yang terhubung ke *slave* tersebut. Namun, dalam penelitian ini *push button* digunakan untuk menggantikan sensor sebagai input. Status "Aktif" (1) dan "Tidak Aktif" (0) akan menyalakan/mematikan lampu indikator pada antarmuka LabVIEW. Status tombol berubah

sesuai dengan interaksi fisik yang dilakukan dan indikator lampu pada antarmuka LabVIEW, dengan perubahan status yang tercatat dengan akurat. Tidak ada anomali atau delay yang signifikan dalam pembacaan status input.

Tabel 3. Status Coil pada Modul Relay

Data LabVIEW	Perintah Coil	Status Relay
1 (True)	Set Coil	ON
0 (False)	Reset Coil	OFF

Tabel 2 menunjukkan bahwa data boolean yang diterima dari LabVIEW langsung menentukan perintah yang diberikan kepada coil dan, pada gilirannya, menentukan status relay. Kondisi "True" dari LabVIEW menyebabkan coil diaktifkan dan relay berubah ke status "ON". Sebaliknya, kondisi "False" menyebabkan coil dinonaktifkan dan relay berubah ke status "OFF".



Gbr. 7 Waktu Respon Perubahan Status Coil pada Setiap Slave

Gbr 3 menunjukkan waktu respon untuk perubahan status coil pada masing-masing *slave* bervariasi, dengan beberapa *slave* menunjukkan perbedaan signifikan antara status "ON" dan "OFF". Pada Slave 1, 4, dan Slave 5 menunjukkan waktu respon yang lebih cepat untuk status "OFF" dibandingkan dengan status "ON", sementara Slave 2 menunjukkan waktu respon yang hampir sama untuk kedua status namun lebih tinggi dibandingkan *slave* lainnya. Slave 3 menunjukkan waktu respon tertinggi untuk status "ON", yang bisa disebabkan oleh faktor-faktor seperti beban relay, kondisi jaringan komunikasi, atau performa perangkat keras.

Meskipun terdapat variasi dalam waktu respon, sistem secara keseluruhan masih berfungsi dalam batas yang dapat diterima dan menunjukkan keandalan dalam merespons perubahan status *coil*.



Gbr. 8 Hasil Implementasi Sistem Secara Keseluruhan

Meskipun secara keseluruhan sistem berjalan dengan baik, terdapat beberapa aspek yang masih perlu ditingkatkan. Salah satunya adalah belum adanya implementasi untuk holding register, yang penting untuk menyimpan data set point dalam Modbus. Selain itu, perubahan mode menjadi *full duplex* dapat meningkatkan efisiensi dan kecepatan komunikasi antar sistem secara signifikan.

## V. KESIMPULAN

Hasil penelitian menunjukkan bahwa integrasi LabVIEW dengan Arduino melalui Modbus RTU berhasil dilakukan dengan baik. Data dari sensor DHT11 dapat ditransmisikan secara real-time ke LabVIEW, dan pembacaan sensor yang konsisten dan akurat baik untuk suhu maupun kelembapan. Status tombol berubah sesuai dengan interaksi fisik yang dilakukan dan indikator lampu pada antarmuka LabVIEW merekam perubahan status dengan akurat. Meskipun terdapat fluktuasi dalam waktu respon, seperti yang bisa dipengaruhi oleh beban relay, kondisi jaringan komunikasi, atau performa perangkat keras, respons yang tercatat masih dalam rentang yang dapat diterima. Tidak ada anomali atau delay yang signifikan dalam pembacaan status input, menunjukkan keandalan sistem dalam merespons interaksi pengguna dengan tombol dan feedback yang diberikan melalui antarmuka LabVIEW.

Sistem yang dikembangkan mampu memberikan monitoring dan kontrol yang efisien, serta dapat diandalkan untuk aplikasi industri maupun penelitian. Penelitian ini berhasil menunjukkan bahwa penggunaan Modbus RTU sebagai penghubung antara LabVIEW dan Arduino adalah solusi yang efektif untuk menciptakan sistem monitoring dan kontrol yang terintegrasi.

## UCAPAN TERIMA KASIH

Penulis ingin mengucapkan terima kasih kepada Pusat Penelitian dan Pengabdian kepada Masyarakat (P3M) Politeknik Negeri Banjarmasin atas dukungan finansial yang diberikan melalui hibah penelitian untuk penelitian ini.

## REFERENSI

- [1] C. Elliott, V. Vijayakumar, W. Zink, and R. Hansen, "National Instruments LabVIEW: A Programming Environment for Laboratory Automation and Measurement," *J. Lab. Autom.*, vol. 12, no. 1, pp. 17–24, 2007, doi: 10.1016/j.jala.2006.07.012.
- [2] Modbus Organization, "About the Protocol." Accessed: Jul. 04, 2024. [Online]. Available: <https://modbus.org/faq.php>
- [3] H. Suryantoro, "Prototype Sistem Monitoring Level Air Berbasis Labview dan Arduino Sebagai Sarana Pendukung Praktikum Instrumentasi Sistem Kendali," *Indones. J. Lab.*, vol. 1, no. 3, p. 20, 2019, doi: 10.22146/ijl.v1i3.48718.
- [4] B. N. Mohapatra, N. Barwade, A. Walse, and D. Patil, "Data Acquisition System of Temperature Measurement Using LabVIEW Application," *Andalasian Int. J. Appl. Sci. Eng. Technol.*, vol. 3, no. 01, pp. 26–31, 2023, doi: 10.25077/aijaset.v3i01.70.
- [5] S. S. Tippannavar, S. N, and P. K. M. S, "Smart Home Automation Implemented using LabVIEW and Arduino," in *2022 International Conference on Electronics and Renewable Systems (ICEARS), 2022*, pp. 644–649. doi: 10.1109/ICEARS53579.2022.9752265.
- [6] V.-C. Ifrim, C. Bejenar, L.-D. Milici, and P. Atănăsoae, "LabVIEW-based solar air heater monitoring system," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1256, no. 1, p. 012036, 2022, doi: 10.1088/1757-899x/1256/1/012036.
- [7] S. Dwi Riyanto, P. Purwiyanto, and R. P. Dewi, "Optimalisasi Labview Sebagai

- Kendali dan Monitoring Arus Tegangan pada Modul Solar Cell Menggunakan Jaringan Lokal,” *Infotekmesin*, vol. 14, no. 2, pp. 296–302, 2023, doi: 10.35970/infotekmesin.v14i2.1910.
- [8] C. M. Bulliner, “ModbusRTUSlave.” GitHub, 2023. [Online]. Available: <https://github.com/CMB27/ModbusRTUSlave>
- [9] H.-P. Halvorsen, “Modbus With Practical LabVIEW Examples,” <https://www.halvorsen.blog/>. Accessed: Jul. 03, 2024. [Online]. Available: <https://www.halvorsen.blog>
- [10] modbus tools, “Modbus Poll.” Accessed: Jul. 05, 2024. [Online]. Available: [https://www.modbustools.com/modbus\\_poll.html](https://www.modbustools.com/modbus_poll.html)