

Perancangan dan Analisis Kinerja Sistem Gateway IoT Berbasis RS485 Menggunakan ESP32 untuk *Monitoring* Intensitas Cahaya secara *Real-Time*

Akmal Fauzi Rahman ^{1*)}

Program Studi Teknik Elektro
Universitas Negeri Semarang
Jalan Raya Banaran, Sekaran, Kota Semarang, Jawa Tengah 50229, Indonesia

^{*)}Korespondensi : akmalrahman@students.unnes.ac.id

Abstrak

Perkembangan Internet of Things (IoT) mendorong penerapan sistem monitoring real-time pada berbagai sektor industri. Penelitian ini bertujuan merancang sistem monitoring intensitas cahaya berbasis IoT dengan komunikasi RS485 menggunakan protokol Modbus RTU serta integrasi MQTT. Sistem menggunakan ESP32 sebagai gateway yang menghubungkan sensor BH1750 dengan platform monitoring. Metode meliputi perancangan perangkat keras, pengembangan perangkat lunak berbasis FreeRTOS, dan pengujian dengan interval satu menit. Evaluasi dilakukan menggunakan Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), rata-rata delay, dan Packet Delivery Ratio (PDR). Hasil menunjukkan sistem mampu bekerja secara real-time dengan akurasi yang baik, ditunjukkan oleh nilai MAE dan MAPE yang rendah. Selain itu, rata-rata delay sekitar 120 ms dan nilai PDR 100% menunjukkan komunikasi data berjalan stabil. Sistem ini memenuhi aspek akurasi, responsivitas, dan keandalan, sehingga berpotensi diterapkan pada monitoring industri skala kecil hingga menengah.

Kata kunci : Internet of Things (IoT), ESP32, RS485, Modbus RTU, MQTT, Monitoring Intensitas Cahaya.

Abstract

The development of the Internet of Things (IoT) has enabled the implementation of real-time monitoring systems in various industrial sectors. This study aims to design an IoT-based light intensity monitoring system using RS485 communication with the Modbus RTU protocol and MQTT integration. The system utilizes ESP32 as a gateway to connect the BH1750 sensor with a monitoring platform. The methodology includes hardware design, FreeRTOS-based software development, and testing with one-minute data intervals. Evaluation is conducted using Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), average delay, and Packet Delivery Ratio (PDR). The results indicate that the system operates in real-time with good accuracy, as shown by low MAE and MAPE values. In addition, the average delay of approximately 120 ms and PDR 100% indicate stable data communication. The system meets the requirements of accuracy, responsiveness, and reliability.

Keywords : Internet of Things (IoT), ESP32, RS485, Modbus RTU, MQTT, Light Intensity Monitoring

I. PENDAHULUAN

Perkembangan teknologi Internet of Things (IoT) telah mendorong transformasi signifikan dalam sistem monitoring dan otomasi di berbagai sektor industri. Integrasi antara sensor, mikrokontroler, dan jaringan komunikasi memungkinkan proses akuisisi data dilakukan secara real-time, sehingga meningkatkan efisiensi dan akurasi dalam pengambilan keputusan. Berbagai penelitian menunjukkan bahwa implementasi sistem monitoring berbasis IoT mampu mendukung pengawasan parameter lingkungan maupun energi secara lebih efektif dibandingkan sistem konvensional, khususnya dalam konteks industri dan energi terbarukan [1], [2]. Selain itu, pendekatan modular dalam sistem IoT memungkinkan pengembangan sistem yang lebih fleksibel dan scalable [3]

Info Makalah:

Dikirim : 04-21-2026;
Revisi 1 : 06-29-2026;
Revisi 2 : -
Diterima : 06-30-2026.

Penulis Korespondensi:

Telp : +62-859-2484-1384
e-mail :
akmalrahman@student.unnes.ac.id

Dalam lingkungan industri modern, kebutuhan akan sistem monitoring yang andal semakin meningkat seiring dengan kompleksitas sistem dan tuntutan efisiensi operasional. Sistem berbasis *Programmable Logic Controller* (PLC) masih banyak digunakan karena keandalannya, namun biaya implementasi yang relatif tinggi menjadi kendala pada skala kecil hingga menengah [4]. Oleh karena itu, mikrokontroler seperti Arduino dan ESP32 mulai dikembangkan sebagai alternatif berbiaya rendah dengan performa yang kompetitif, terutama ketika dikombinasikan dengan teknologi komunikasi dan protokol IoT modern [5], [6]. Integrasi dengan platform berbasis MQTT juga memungkinkan visualisasi data secara *real-time* melalui dashboard digital (Revelivan et al., 2019).

Di sisi lain, aspek komunikasi data menjadi tantangan utama dalam sistem monitoring industri, terutama pada lingkungan dengan gangguan elektromagnetik tinggi dan kebutuhan transmisi jarak jauh. Teknologi komunikasi konvensional seperti I2C dan UART memiliki keterbatasan dalam hal jarak dan ketahanan terhadap noise. Oleh karena itu, RS485 banyak digunakan karena mampu mendukung komunikasi jarak jauh, tahan terhadap gangguan, serta memungkinkan komunikasi multi-node [6], [7]. Namun, integrasi komunikasi industri tersebut dengan sistem IoT berbasis cloud masih belum optimal dalam banyak implementasi.

Meskipun sistem monitoring berbasis IoT telah banyak dikembangkan, sebagian besar penelitian masih berfokus pada integrasi sensor dan platform cloud tanpa menekankan keandalan komunikasi pada lingkungan industri [1], [2]. Di sisi lain, implementasi RS485 dan Modbus RTU umumnya masih terbatas pada sistem lokal dan belum terintegrasi secara menyeluruh dengan teknologi IoT [6], [7]. Selain itu, integrasi antara komunikasi industri dan protokol IoT seperti MQTT dalam satu arsitektur terpadu masih belum banyak dikaji secara komprehensif [3], [8].

Kontribusi utama penelitian ini adalah merancang dan mengimplementasikan sistem monitoring terintegrasi yang menggabungkan komunikasi industri dan IoT dalam satu arsitektur yang efisien dan berbiaya rendah. Sistem ini memungkinkan akuisisi data yang stabil melalui RS485 serta distribusi data secara *real-time* melalui MQTT, sehingga dapat menjadi solusi alternatif untuk sistem monitoring industri skala kecil hingga menengah.

Penelitian ini bertujuan untuk mengembangkan sistem monitoring berbasis Internet of Things (IoT) yang mampu mengintegrasikan komunikasi industri dengan platform digital secara *real-time*. Pengembangan sistem ini difokuskan pada peningkatan keandalan komunikasi data dalam lingkungan industri melalui pemanfaatan protokol komunikasi yang robust serta arsitektur sistem yang terintegrasi [4], [9]. Secara lebih spesifik, penelitian ini bertujuan untuk merancang sistem monitoring berbasis IoT dengan menggunakan mikrokontroler ESP32 sebagai gateway, mengimplementasikan komunikasi data menggunakan RS485 dengan protokol Modbus RTU untuk meningkatkan stabilitas transmisi data, serta mengintegrasikan sistem tersebut dengan protokol MQTT untuk memungkinkan pengiriman data ke platform monitoring berbasis cloud secara *real-time*. Selain itu, penelitian ini juga bertujuan untuk mengevaluasi kinerja sistem dalam hal keandalan komunikasi dan keberhasilan integrasi antar protokol dalam satu arsitektur sistem [3].

Berdasarkan latar belakang dan celah penelitian yang telah diidentifikasi, rumusan masalah dalam penelitian ini difokuskan pada bagaimana merancang sistem monitoring berbasis IoT yang mampu mengintegrasikan komunikasi industri dan platform digital secara efektif. Selain itu, penelitian ini juga mengkaji bagaimana implementasi komunikasi RS485 dengan protokol Modbus RTU dapat bekerja secara stabil dalam sistem monitoring, serta bagaimana proses integrasi antara komunikasi berbasis Modbus RTU dengan protokol MQTT dapat dilakukan untuk mendukung pengiriman data secara *real-time*. Permasalahan lain yang turut dikaji adalah bagaimana kinerja sistem yang dikembangkan dalam hal keandalan komunikasi serta integrasi data antar perangkat dalam satu jaringan [6], [7].

Ruang lingkup penelitian dibatasi pada pengembangan sistem monitoring berbasis IoT yang menggunakan mikrokontroler ESP32 sebagai gateway utama dalam sistem. Komunikasi antar perangkat dibatasi menggunakan protokol Modbus RTU berbasis RS485, sedangkan pengiriman data ke platform monitoring dilakukan melalui protokol MQTT. Parameter yang dimonitor dalam penelitian ini terbatas

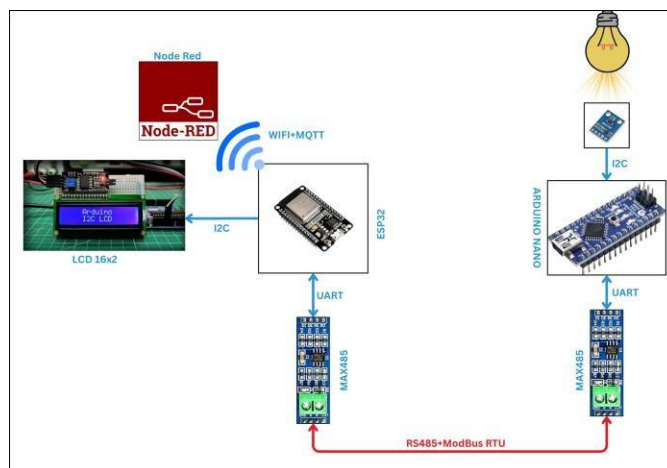
Implementasi Sistem Monitoring Ketinggian Air pada Tandon Menggunakan ESP32 dengan Komunikasi RS485 dan Integrasi IoT ThingSpeak (Akmal Fauzi Rahman: Halaman 40 - 50)

pada data sensor yang terhubung dalam sistem, seperti parameter lingkungan atau intensitas cahaya. Selain itu, pengujian sistem difokuskan pada aspek keandalan komunikasi dan integrasi data, tanpa mencakup analisis keamanan jaringan secara mendalam. Implementasi sistem juga difokuskan pada skala kecil hingga menengah sesuai dengan karakteristik penggunaan mikrokontroler sebagai solusi monitoring yang ekonomis dan fleksibel [5], [8].

II. METODE

A. Arsitektur Sistem

Sistem yang dikembangkan dalam penelitian ini merupakan sistem *monitoring* intensitas cahaya berbasis Internet of Things (IoT) yang mengintegrasikan komunikasi industri menggunakan RS485 dengan protokol Modbus RTU dalam arsitektur *master-slave*. Sistem terdiri dari node *slave* berupa Arduino Nano yang berfungsi membaca data sensor BH1750 melalui komunikasi I2C, kemudian menyimpan dan mengirimkan data tersebut dalam bentuk register Modbus melalui modul MAX485 sebagai konverter sinyal UART ke RS485. Data selanjutnya dikirim melalui jaringan RS485 dengan topologi multi-drop dan mode half-duplex menuju node *master*, yaitu ESP32, yang berperan sebagai gateway IoT[10].

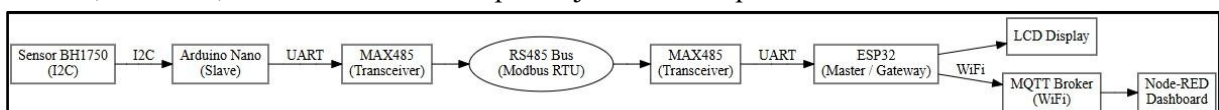


Gambar 1. Arsitektur sistem

ESP32 melakukan proses *polling* terhadap *slave* untuk memperoleh data intensitas cahaya, kemudian memproses dan menampilkannya pada media lokal LCD serta indikator LED, sekaligus meneruskan data ke platform IoT melalui protokol MQTT menggunakan koneksi WiFi. Data yang dikirimkan akan divisualisasikan pada dashboard berbasis Node-RED sehingga memungkinkan *monitoring* secara real-time. Dengan arsitektur ini, sistem mampu menggabungkan keandalan komunikasi industri berbasis RS485 dengan fleksibilitas teknologi IoT untuk menghasilkan sistem *monitoring* yang efisien, scalable, dan berbiaya relatif rendah[11].

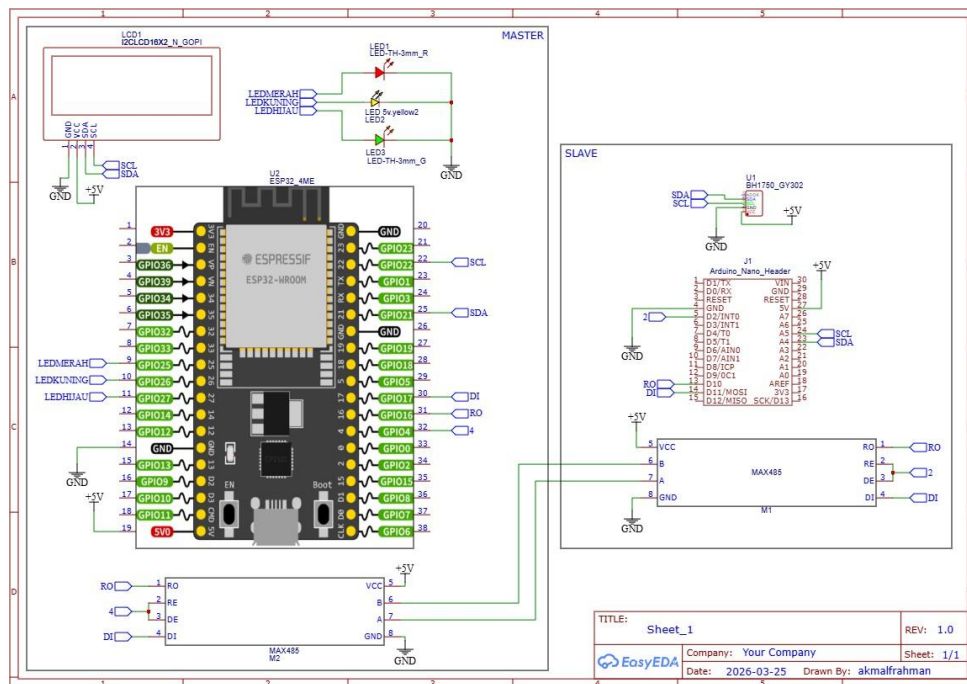
B. Perancangan Perangkat Keras

Perancangan perangkat keras pada penelitian ini dilakukan untuk membangun sistem *monitoring* berbasis Internet of Things (IoT) yang mampu mengintegrasikan sensor, mikrokontroler, serta komunikasi industri menggunakan RS485[12]. Sistem dirancang dengan pendekatan modular yang terdiri dari node sensor sebagai *slave* dan gateway sebagai *master*, sehingga memudahkan proses pengembangan dan skalabilitas sistem. Setiap komponen memiliki fungsi spesifik yang saling terhubung untuk memastikan proses akuisisi, transmisi, dan visualisasi data dapat berjalan secara optimal.



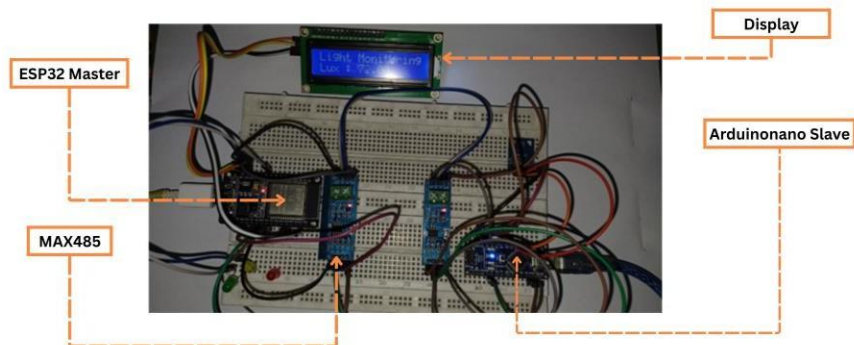
Gambar 2. Diagram blok arsitektur sistem monitoring intensitas cahaya berbasis IoT menggunakan komunikasi RS485 dan MQTT

Gambar 2 menunjukkan blok diagram perangkat keras yang menggambarkan alur kerja sistem secara umum. Sensor BH1750 digunakan untuk mengukur intensitas cahaya yang kemudian diproses oleh Arduino Nano sebagai node *slave*. Data hasil pembacaan sensor dikirimkan melalui komunikasi serial UART ke modul MAX485 untuk dikonversi menjadi sinyal diferensial RS485. Selanjutnya, data ditransmisikan melalui jalur RS485 menuju ESP32 sebagai node *master* atau gateway. ESP32 menerima dan mengolah data sebelum menampilkannya pada perangkat output lokal serta mengirimkannya ke server melalui jaringan WiFi menggunakan protokol MQTT untuk keperluan *monitoring* berbasis Node-RED.



Gambar 3. Skematik rangkaian *slave* dan *master*

Gambar 3 menunjukkan skematik rangkaian perangkat keras yang digunakan dalam sistem. Sensor BH1750 dihubungkan ke Arduino Nano melalui antarmuka I2C, sedangkan komunikasi antara Arduino Nano dan modul MAX485 menggunakan antarmuka UART. Modul MAX485 berfungsi sebagai transceiver yang mengubah sinyal UART menjadi sinyal diferensial RS485 untuk mendukung komunikasi jarak jauh yang lebih stabil. Pada sisi penerima, ESP32 dihubungkan dengan modul MAX485 untuk menerima data dari jalur RS485. Koneksi antar perangkat dirancang dengan mempertimbangkan kestabilan komunikasi serta efisiensi penggunaan pin pada masing-masing mikrokontroler.



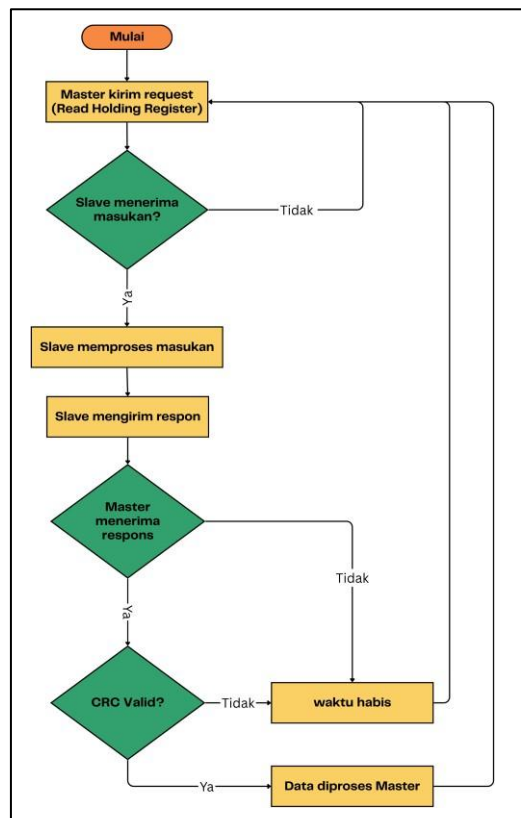
Gambar 4. Implementasi perangkat keras sistem monitoring intensitas cahaya berbasis IoT menggunakan ESP32, Arduino Nano, dan komunikasi RS485

Implementasi Sistem Monitoring Ketinggian Air pada Tandon Menggunakan ESP32 dengan Komunikasi RS485 dan Integrasi IoT ThingSpeak (Akmal Fauzi Rahman: Halaman 40 - 50)

Gambar 4 menunjukkan implementasi fisik dari sistem yang telah dirancang. Seluruh komponen perangkat keras, termasuk Arduino Nano, ESP32, modul MAX485, serta sensor BH1750, dirakit dalam satu sistem yang terintegrasi. Koneksi antar modul menggunakan kabel *jumper* dengan konfigurasi sesuai dengan skematik rangkaian yang telah dirancang sebelumnya. Prototipe ini digunakan untuk melakukan pengujian sistem secara langsung guna memastikan bahwa proses akuisisi data, komunikasi RS485, serta pengiriman data ke platform IoT dapat berjalan dengan baik.

C. Komunikasi RS485 Berbasis Modbus RTU

Komunikasi RS485 merupakan standar komunikasi serial diferensial yang banyak digunakan pada sistem industri karena memiliki ketahanan tinggi terhadap gangguan noise dan mampu mentransmisikan data dalam jarak yang relatif jauh. Sistem ini menggunakan dua jalur sinyal, yaitu A dan B, yang bekerja secara diferensial sehingga meningkatkan keandalan transmisi dibandingkan komunikasi serial konvensional. Dalam penelitian ini, RS485 diimplementasikan menggunakan protokol Modbus RTU, yaitu protokol komunikasi berbasis *master-slave* yang memungkinkan pertukaran data secara terstruktur. Modbus RTU menggunakan frame data yang terdiri dari alamat perangkat, kode fungsi, data, serta mekanisme pengecekan error menggunakan cyclic redundancy check (CRC), sehingga komunikasi antar perangkat menjadi lebih stabil dan minim kesalahan.



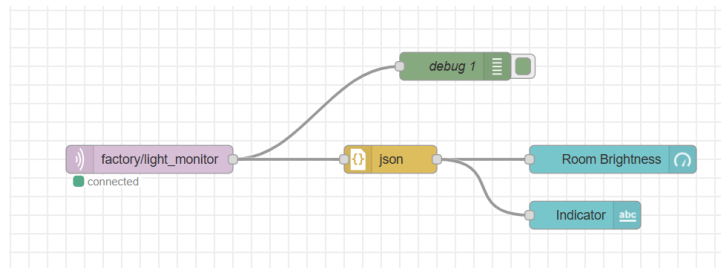
Gambar 5. Diagram alir komunikasi Modbus RTU antara *master* dan *slave*

Implementasi komunikasi dilakukan dengan menggunakan ESP32 sebagai *master* dan Arduino Nano sebagai *slave* yang terhubung melalui modul MAX485 sebagai transceiver. Arduino Nano bertugas membaca data dari sensor BH1750 dan menyimpannya dalam register Modbus yang dapat diakses oleh *master*. ESP32 kemudian melakukan proses polling secara periodik untuk meminta data dari *slave* melalui jaringan RS485. Setelah menerima permintaan, *slave* akan mengirimkan respons berupa data sensor dalam format frame Modbus RTU. Data yang diterima oleh ESP32 selanjutnya diproses untuk ditampilkan pada LCD maupun dikirim ke platform IoT untuk keperluan *monitoring*. Pendekatan ini

dipilih karena mampu memberikan komunikasi yang stabil, efisien, serta mudah dikembangkan untuk sistem dengan jumlah node yang lebih banyak.

D. Perancangan Perangkat Lunak

Perancangan perangkat lunak pada penelitian ini mencakup pengembangan sistem komunikasi antara node *slave* dan *master* serta integrasi dengan platform Internet of Things untuk keperluan *monitoring*. Arduino Nano diprogram sebagai *slave* menggunakan protokol Modbus RTU untuk membaca data dari sensor BH1750 dan menyimpannya dalam register yang dapat diakses oleh *master*. ESP32 berperan sebagai *master* yang secara periodik melakukan proses polling untuk mengambil data dari *slave* melalui komunikasi RS485 menggunakan modul MAX485. Data yang diterima kemudian diproses dan ditampilkan pada perangkat output lokal sebagai bentuk *monitoring* langsung. ESP32 juga berfungsi sebagai gateway IoT dengan mengirimkan data sensor ke broker MQTT melalui koneksi WiFi. Data yang telah dipublish kemudian diterima dan diolah menggunakan platform Node-RED untuk divisualisasikan dalam bentuk *dashboard monitoring* secara real-time. Perancangan perangkat lunak ini memungkinkan integrasi antara sistem komunikasi industri berbasis RS485 dengan teknologi IoT, sehingga data dapat diakses secara fleksibel dan efisien baik secara lokal maupun jarak jauh



Gambar 6. Node program yang digunakan pada Node-RED

E. Implementasi FreeRTOS

Implementasi FreeRTOS pada penelitian ini dilakukan untuk mendukung mekanisme multitasking dalam pengolahan sistem secara real-time, khususnya dalam menangani komunikasi RS485, pemrosesan data, serta pengiriman data ke platform IoT secara bersamaan. Sistem dirancang dengan membagi proses ke dalam beberapa task yang berjalan secara paralel, meliputi task komunikasi Modbus RTU untuk melakukan polling data dari Arduino Nano, task pemrosesan data untuk mengolah hasil pembacaan sensor, serta tugas (*task*) komunikasi jaringan untuk mengirimkan data ke broker MQTT yang selanjutnya divisualisasikan melalui Node-RED. Dengan pendekatan ini, sistem mampu mengurangi blocking process dan meningkatkan efisiensi serta responsivitas dalam pengolahan data, sehingga kinerja sistem menjadi lebih stabil dan optimal untuk aplikasi *monitoring* berbasis IoT.

F. Alur Kerja Sistem

Alur kerja sistem dimulai dari proses pembacaan data intensitas cahaya oleh sensor BH1750 yang terhubung dengan Arduino Nano sebagai node *slave*. Data sensor yang diperoleh kemudian diproses dan disimpan dalam register Modbus untuk selanjutnya dikirimkan melalui komunikasi RS485 menggunakan modul MAX485. ESP32 sebagai node *master* melakukan proses polling secara berkala untuk mengambil data dari *slave*, kemudian memproses data tersebut untuk ditampilkan pada perangkat output lokal. Selain itu, ESP32 juga mengirimkan data ke broker MQTT melalui jaringan WiFi, sehingga data dapat diterima dan divisualisasikan secara real-time menggunakan platform Node-RED. Alur kerja ini menunjukkan integrasi antara sistem komunikasi industri dan teknologi IoT dalam mendukung proses *monitoring* yang efisien dan terstruktur.

G. Metode Pengujian

Metode pengujian pada penelitian ini dilakukan untuk mengevaluasi kinerja sistem dari aspek akurasi sensor, waktu respons, serta keandalan komunikasi data. Pengujian akurasi sensor dilakukan dengan membandingkan nilai intensitas cahaya yang diukur oleh sensor BH1750 dengan nilai referensi yang diperoleh dari aplikasi lux meter pada smartphone, sehingga pengujian bersifat komparatif. Selain itu, pengujian delay sistem dilakukan dengan mengukur selisih waktu antara data dikirim dari node *slave*

hingga diterima dan diproses oleh ESP32 sebagai master. Sementara itu, keandalan komunikasi RS485 diuji dengan menghitung jumlah data yang berhasil dikirim dan diterima dibandingkan dengan total data yang dikirim selama periode pengujian. Data hasil pengujian kemudian disajikan dalam bentuk tabel untuk memudahkan proses analisis lebih lanjut.

H. Metode Analisis Data

Analisis data pada penelitian ini dilakukan menggunakan beberapa parameter evaluasi yang umum digunakan dalam sistem monitoring dan komunikasi data[13].

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - y_i| \quad (1)$$

Tingkat akurasi sensor dianalisis menggunakan metode Mean Absolute Error (MAE) dan Mean Absolute Percentage Error (MAPE) dengan membandingkan nilai sensor terhadap nilai referensi[14].

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{x_i - y_i}{y_i} \right| \quad (2)$$

Kinerja sistem dari sisi waktu respons dianalisis menggunakan parameter rata-rata delay (average delay) yang dihitung dari selisih waktu pengiriman dan penerimaan data[15].

$$Avg\ Delay = \frac{1}{n} \sum_{i=1}^n d_i \quad (3)$$

Selain itu, keandalan komunikasi RS485 dievaluasi menggunakan parameter Packet Delivery Ratio (PDR), yaitu perbandingan antara jumlah data yang berhasil diterima dengan jumlah data yang dikirim. Penggunaan parameter-parameter ini memungkinkan evaluasi sistem dilakukan secara kuantitatif dan terukur[16].

$$PDR = \frac{Packet\ diterima}{Packet\ dikirm} \times 100\% \quad (3)$$

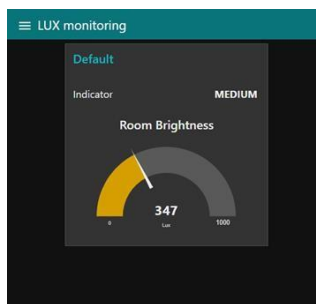
III. HASIL DAN DISKUSI

A. Implementasi Sistem

Implementasi sistem dilakukan berdasarkan perancangan yang telah dijelaskan pada tahap metode, dengan merealisasikan seluruh komponen menjadi suatu sistem monitoring berbasis *Internet of Things* (IoT) yang terintegrasi. Sistem ini terdiri dari perangkat keras berupa mikrokontroler sebagai pusat kendali, sensor untuk pengambilan data, serta modul komunikasi yang mendukung pertukaran data antar perangkat. Seluruh komponen dirangkai sesuai dengan arsitektur sistem yang telah dirancang sehingga mampu bekerja secara terkoordinasi dalam melakukan proses akuisisi dan transmisi data.

Pada tahap implementasi perangkat keras, seluruh komponen diintegrasikan dalam satu sistem yang mampu melakukan pembacaan data sensor secara langsung dari lingkungan. Data yang diperoleh kemudian diproses oleh mikrokontroler sebelum dikirimkan melalui jalur komunikasi yang telah ditentukan. Implementasi ini memastikan bahwa setiap komponen dapat berfungsi sesuai dengan perannya masing-masing, baik dalam proses pengambilan data maupun dalam proses komunikasi antar perangkat.

Implementasi perangkat lunak dilakukan dengan mengembangkan program yang berjalan pada mikrokontroler untuk mengatur alur kerja sistem secara keseluruhan. Program ini mencakup proses pembacaan data sensor, pengolahan data, serta pengiriman data ke sistem monitoring. Selain itu, sistem juga dilengkapi dengan mekanisme komunikasi yang memungkinkan pertukaran data secara terstruktur dan berkelanjutan, sehingga data yang dihasilkan dapat diakses secara real-time.



Gambar 7. Tampilan hasil monitoring pada sistem

Secara keseluruhan, hasil implementasi menunjukkan bahwa sistem yang dikembangkan telah mampu bekerja sesuai dengan perancangan awal, dimana proses akuisisi data, komunikasi, serta visualisasi data dapat berjalan dengan baik dalam satu kesatuan sistem yang terintegrasi. Hal ini menunjukkan bahwa pendekatan yang digunakan dalam pengembangan sistem monitoring berbasis IoT dapat diimplementasikan secara efektif untuk mendukung kebutuhan pemantauan secara real-time.

B. Hasil Pengujian

Hasil pengujian sistem diperoleh berdasarkan skenario yang telah dirancang sebelumnya, dengan fokus pada pengamatan terhadap data sensor, kestabilan komunikasi, serta keberhasilan sistem dalam menampilkan data secara real-time. Proses pengambilan data dilakukan dalam interval waktu satu menit untuk melihat konsistensi pembacaan sensor serta performa sistem dalam kondisi operasional.

Tabel 1 Hasil Pembacaan Sensor

No	Waktu	BH1750 (Lux)	HP Lux Meter (Lux)	Error (Lux)	Error (%)
1	menit ke-1	58	60	2	3,33
2	menit ke-2	135	130	5	3,85
3	menit ke-3	310	320	10	3,13
4	menit ke-4	545	530	15	2,83
5	menit ke-5	820	850	30	3,53
6	menit ke-6	920	900	20	2,22
7	menit ke-7	970	1000	30	3,00
8	menit ke-8	1230	1200	30	2,50
9	menit ke-9	1350	1400	50	3,57
10	menit ke-10	1650	1600	50	3,13
11	menit ke-11	1750	1800	50	2,78
12	menit ke-12	1950	1900	50	2,63
13	menit ke-13	1920	2000	80	4,00
14	menit ke-14	2100	2200	100	4,55
15	menit ke-15	2350	2400	50	2,08

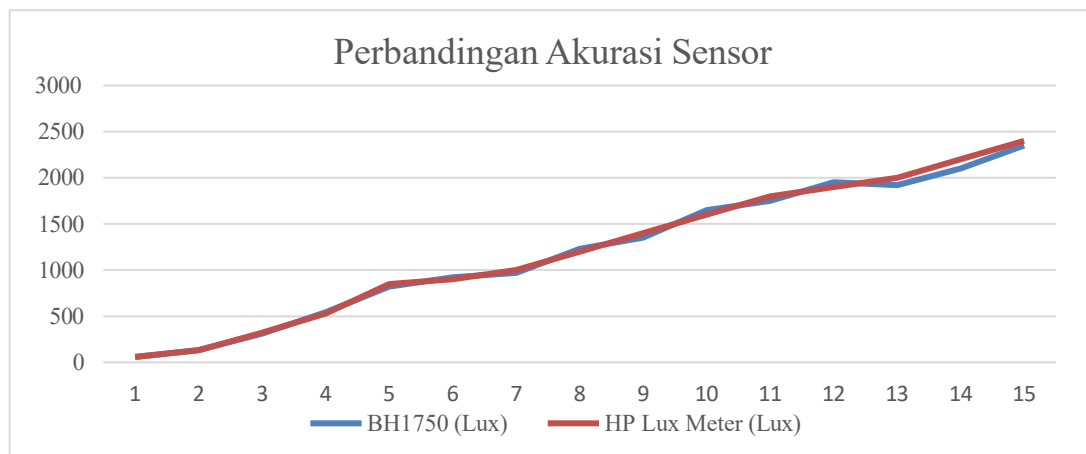
Tabel 2 Hasil delay system

No	Pengujian data ke-	Delay	Status
1	Data ke 1	118	OK
2	Data ke 2	120	OK
3	Data ke 3	121	OK

No	Pengujian data ke-	Delay	Status
4	Data ke 4	119	OK
5	Data ke 5	122	OK
6	Data ke 6	120	OK
7	Data ke 7	119	OK
8	Data ke 8	121	OK
9	Data ke 9	120	OK
10	Data ke 10	118	OK
11	Data ke 11	122	OK
12	Data ke 12	121	OK
13	Data ke 13	119	OK
14	Data ke 14	120	OK
15	Data ke 15	121	OK

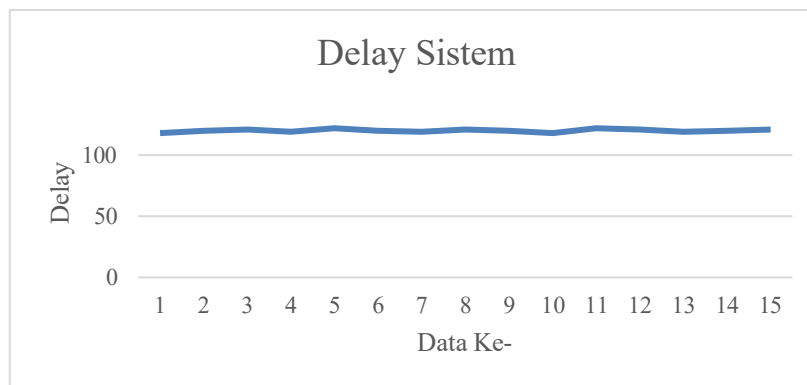
C. Analisis dan Diskusi

Berdasarkan hasil pengujian, sistem monitoring mampu melakukan pembacaan intensitas cahaya menggunakan sensor BH1750 dengan hasil yang relatif mendekati nilai referensi yang diperoleh dari aplikasi lux meter pada perangkat smartphone. Pengambilan data dilakukan setiap satu menit untuk mengamati konsistensi pembacaan sensor dalam berbagai kondisi pencahayaan. Secara umum, nilai yang dihasilkan oleh sensor menunjukkan pola yang sebanding dengan alat referensi, di mana peningkatan intensitas cahaya pada lingkungan diikuti oleh kenaikan nilai pembacaan sensor. Hal ini menunjukkan bahwa sensor mampu merespons perubahan kondisi cahaya dengan baik.



Gambar 8. Perbandingan hasil pengukuran intensitas cahaya antara sensor BH1750 dan HP *Lux meter*

Selisih antara nilai sensor dan nilai referensi ditunjukkan dalam bentuk error absolut dan error persentase. Berdasarkan data yang diperoleh, nilai error berada pada rentang yang relatif kecil, yaitu sekitar 2% hingga 4,55%. Nilai error terbesar terjadi pada pengujian menit ke-14 dengan selisih 100 lux atau sekitar 4,55%, sedangkan error terkecil terjadi pada menit ke-15 sebesar 2,08%. Secara keseluruhan, nilai error yang relatif rendah ini menunjukkan bahwa sensor BH1750 memiliki tingkat akurasi yang cukup baik untuk digunakan dalam sistem monitoring intensitas cahaya.



Gambar 9. hasil delay sistem

Berdasarkan hasil pengujian kinerja sistem dari sisi waktu respons, nilai delay sistem berada pada kisaran 118 ms hingga 122 ms untuk setiap pengiriman data. Nilai delay yang relatif stabil ini menunjukkan bahwa sistem memiliki performa komunikasi yang konsisten dalam mengirimkan data dari perangkat ESP32 ke broker MQTT. Selain itu, seluruh data yang dikirimkan memiliki status “OK”, yang menunjukkan bahwa tidak terdapat kegagalan pengiriman selama proses pengujian berlangsung.

Dengan demikian, sistem yang dikembangkan mampu bekerja dengan baik baik dari sisi akurasi sensor maupun performa komunikasi. Sensor BH1750 menunjukkan tingkat kesalahan yang rendah terhadap nilai referensi, sementara sistem komunikasi berbasis RS485 dan MQTT mampu mengirimkan data secara stabil dan real-time tanpa kehilangan paket data. Hasil ini menunjukkan bahwa sistem monitoring yang dirancang telah memenuhi kriteria dasar dalam hal keandalan, akurasi, dan responsivitas.

IV. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, sistem monitoring intensitas cahaya berbasis Internet of Things (IoT) menggunakan ESP32, komunikasi RS485 dengan protokol Modbus RTU, serta integrasi MQTT telah berhasil dikembangkan dan mampu beroperasi dengan baik. Sistem ini dapat melakukan akuisisi data dari sensor BH1750, menampilkan informasi secara lokal melalui LCD, serta mengirimkan data secara real-time ke platform monitoring. Hasil pengujian menunjukkan bahwa sensor memiliki tingkat akurasi yang baik dengan nilai kesalahan yang relatif rendah, yang ditunjukkan melalui parameter MAE dan MAPE. Selain itu, kinerja sistem komunikasi juga menunjukkan hasil yang stabil dengan nilai delay yang rendah, sehingga mendukung kebutuhan monitoring secara responsif.

Dari sisi keandalan komunikasi, nilai Packet Delivery Ratio (PDR) yang tinggi menunjukkan bahwa seluruh data yang dikirimkan dapat diterima dengan baik tanpa kehilangan paket. Hal ini mengindikasikan bahwa integrasi komunikasi RS485 dan MQTT mampu memberikan performa sistem yang andal dan stabil. Secara keseluruhan, sistem yang dikembangkan telah memenuhi aspek utama dalam sistem IoT, yaitu akurasi, responsivitas, dan keandalan komunikasi. Dengan demikian, sistem ini memiliki potensi untuk dikembangkan lebih lanjut dan diimplementasikan pada skala yang lebih luas, khususnya dalam aplikasi monitoring industri dengan kebutuhan sistem yang efisien dan terintegrasi.

DAFTAR PUSTAKA

- [1] C. M. Nkinyam, C. Oliver, C. O. Asadu, B. Anyaka, and P. A. Olubambi, “Results in Engineering Development of a low-cost monitoring device for solar electric (PV) system using internet of things (IoT),” *Results Eng.*, vol. 28, no. September, p. 107324, 2025, doi: 10.1016/j.rineng.2025.107324.
- [2] H. Andrianto and V. Jonathan, “Design of IoT-Based Electrical Energy Meter,” *2023 IEEE 9th Int. Conf. Comput. Eng. Des.*, pp. 1–4, 2023, doi: 10.1109/ICCED60214.2023.10425406.
- [3] A. Baratta *et al.*, “ScienceDirect Design and Implementation of Modular IoT Solutions for Environmental Monitoring and Smart Automation : Three Case Studies in the Dominican Republic , Panama , and Italy,” *Procedia Comput. Sci.*, vol. 274, pp. 1140–1148, 2025, doi:

- 10.1016/j.procs.2025.12.111.
- [4] F. Moghaddam, "PLC-Based Remote Monitoring and Automation for Energy Applications," *2025 7th Exp. Int. Conf.*, pp. 378–383, 2025, doi: 10.1109/exp.at2565440.2025.11348353.
- [5] A. Abu and A. A. A. Shabaneh, "MethodsX Design of a smart hydroponics monitoring system using an ESP32 microcontroller and the Internet of Things," *MethodsX*, vol. 11, no. September, p. 102401, 2023, doi: 10.1016/j.mex.2023.102401.
- [6] M. Dafare, "LoRa-Enabled Smart RS485 Data Logger and MQTT Gateway for Industrial IoT Applications using ESP32," *2023 Int. Conf. Circuit Power Comput. Technol.*, pp. 1297–1302, 2023, doi: 10.1109/ICCPCT58313.2023.10245760.
- [7] Z. Naufal, "Machine Learning Models Performance for NPK Soil Sensor Calibration," *2025 Int. Conf. ICT Smart Soc.*, no. 2020, pp. 1–6, 2025, doi: 10.1109/ICISS66954.2025.11389538.
- [8] C. Revelivan, H. Wiratama, and N. Hanafiah, "ScienceDirect ScienceDirect Door Security System for Home Monitoring Based on ESP32," *Procedia Comput. Sci.*, vol. 157, pp. 673–682, 2019, doi: 10.1016/j.procs.2019.08.218.
- [9] S. P. S. J, C. M. Iniyar, E. M. Barath, G. M. Thangatamilan, R. Gayathri, and M. Apoorva, "Tank Level Control Using Lora and PLC," *2025 Second Int. Conf. Intell. Technol. Sustain. Electr. Commun. Syst. (iTech SECOM)*, no. 2009, pp. 1–6, 2025, doi: 10.1109/iTechSECOM64750.2025.11307668.
- [10] N. Biswas, M. A. Mahdi, and M. Das Gupta, "Electricity Demand Forecasting for Tetouan, Morocco using Darts in Machine-Learning: A Comparative Study of Forecasting Models," *Results Eng.*, p. 110289, 2026, doi: 10.1016/j.rineng.2026.110289.
- [11] E. Bebronne and S. Limbourg, "Logistics 5.0 for Public Lighting Systems: Research Agenda," *IFAC Pap.*, vol. 59, no. 10, pp. 823–828, 2025, doi: 10.1016/j.ifacol.2025.09.140.
- [12] Z. Wu, "On-Chip ESD Protection for A / B Bus Ports of RS485 Using Asymmetrical dual-direction Silicon Controlled Rectifier," *2025 4th Int. Symp. Semicond. Electron. Technol.*, pp. 148–153, 2025, doi: 10.1109/ISSET66828.2025.11184970.
- [13] A. C. Study, "Hybrid Traditional and Deep Learning Approaches for Network Traffic Forecasting with Real-Time Anomaly Detection," no. May, 2025.
- [14] A. Hossain, Z. Harun, T. M. T. Lei, and M. Kumar, "Computers & Industrial Engineering Development , optimization , and benchmarking of zone-specific deep learning architectures for air quality prediction in end-of-life vehicle recycling zones," *Comput. Ind. Eng.*, vol. 210, no. August, p. 111507, 2025, doi: 10.1016/j.cie.2025.111507.
- [15] J. P. Lujan-leon, R. A. Alvarado-lugo, P. W. Lovon-ramos, and C. Benites-condori, "Datalogger for Low-Cost High Mountain Weather Stations : Design , Optimization and Testing of Power Consumption and Compatibility with Satellite Transmitter," *2023 IEEE XXX Int. Conf. Electron. Electr. Eng. Comput.*, pp. 1–8, doi: 10.1109/INTERCON59652.2023.10326074.
- [16] S. Thakur, "AI-Driven Energy-Efficient Routing in IoT-Based Wireless Sensor Networks : A Comprehensive Review," 2025.