

## Implementasi Jaringan Lora *Multi-Node* dengan Konfigurasi Dinamis *Master-Slave* berbasis *Local Server*

Muhammad Abdurrohim<sup>1</sup>✉, Dedi Ary Prasetya<sup>1</sup>, Muhammad Kusban<sup>1</sup>, Umi  
Fadlilah<sup>1</sup>

<sup>1</sup> Universitas Muhammadiyah Surakarta, Jl. A. Yani Tromol Pos 1, Pabelan, Kartasura,  
Sukoharjo, Indonesia

✉ Email korespondensi: muhammadainna@gmail.com

**Abstrak.** *Internet of Things* (IoT) menawarkan potensi besar dalam mendukung sistem pemantauan dan pengendalian jarak jauh. Namun, masih terdapat keterbatasan berupa ketergantungan terhadap koneksi internet yang dapat menimbulkan latensi, risiko privasi, serta kegagalan sistem saat jaringan terputus. Selain itu, protokol standar *Long Range Wide Area Network* (LoRaWAN) yang menggunakan metode akses *Additive Links On-line Hawaii Area* (ALOHA) tidak terkoordinasi, sehingga rentan terhadap tabrakan data, terutama pada jaringan yang padat. Penelitian ini bertujuan untuk merancang arsitektur jaringan monitoring *multi-node* berbasis LoRa dengan sistem *master-slave* dinamis yang terintegrasi dengan *server* lokal, agar sistem dapat beroperasi secara mandiri dan meminimalkan risiko tabrakan paket data. Sistem ini dibangun menggunakan modul ESP32 dan LoRa E220-900T22D, di mana *node slave* membaca data dari sensor suhu, kelembapan, arus, tegangan, serta intensitas cahaya, kemudian mengirimkannya ke *node master* untuk disimpan dan ditampilkan melalui *server* lokal berbasis XAMPP. Pengujian dilakukan di lingkungan kampus Universitas Muhammadiyah Surakarta dalam skenario horizontal dan vertikal, baik dengan maupun tanpa penghalang, menggunakan antena penguat berspesifikasi 5 dBi 915 MHz. Hasil pengujian menunjukkan bahwa sistem bekerja dengan baik hingga jarak 100 meter secara horizontal dengan *packet loss* 0% dan *delay* rata-rata di bawah 7 detik tanpa penghalang. Untuk pengujian respons tombol, sistem masih berfungsi baik hingga kurang dari 50 meter dengan *delay* rata-rata 4,9 detik; *delay* meningkat menjadi rata-rata 14,5 detik pada jarak 100 meter dan mencapai rata-rata 15 detik pada 200 meter, dipengaruhi oleh hambatan seperti pepohonan, kendaraan, dan bangunan. Pengujian secara vertikal menunjukkan penurunan performa signifikan akibat hambatan beton, dengan *delay* tertinggi mencapai 181 detik dan *packet loss* hingga 80% pada ketinggian empat lantai. Penelitian ini menyimpulkan bahwa konsep *master-slave* berbasis *server* lokal layak dijadikan solusi *monitoring* yang fleksibel dan mandiri,



meskipun tetap perlu mempertimbangkan kondisi lingkungan fisik dalam desain jaringan LoRa.

**Kata kunci:** *LoRa, multi-node; master-slave; local server; delay; packet loss*

## PENDAHULUAN

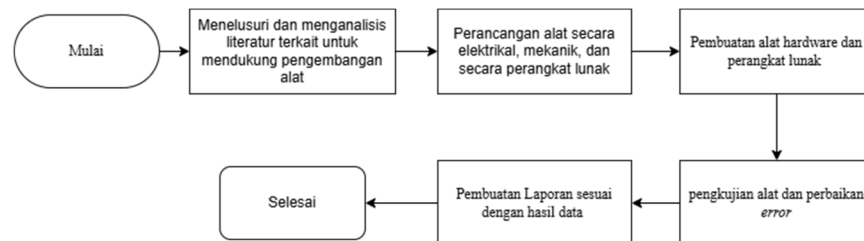
Meskipun *Internet of Things* (IoT) memiliki potensi besar di berbagai sektor, arsitektur konvensional yang bergantung pada *cloud computing* masih menghadapi masalah latensi, *bandwidth*, risiko privasi, serta ketergantungan koneksi internet [1]. Hambatan ini semakin nyata pada aplikasi yang memerlukan respon cepat dan stabil di area terpencil atau lingkungan industri [2]. Sebagai solusi, konsep *edge computing* hadir dengan memproses data secara lokal guna menekan latensi dan menjaga kemandirian sistem. Salah satu teknologi konektivitas yang mendukung IoT adalah *Low Power Wide Area Network* LPWAN, di mana LoRa menjadi unggulan berkat operasi di spektrum ISM tanpa lisensi, jangkauan luas, serta efisiensi dayanya [3]. LoRa memanfaatkan modulasi *Chirp Spread Spectrum* (CSS) dan parameter seperti *Spreading Factor* untuk menyeimbangkan jarak dan kecepatan data [4]. Namun, protokol standar *Long Range Wide Area Network* (LoRaWAN) masih memiliki kelemahan, terutama karena menggunakan *topologi star* dan protokol *Additive Links On-line Hawaii Area ALOHA* tanpa koordinasi yang rawan tabrakan paket pada jaringan padat [5]. Berbagai penelitian telah berupaya mengembangkan algoritma untuk mengurangi dampak tabrakan ini [6]. Lebih lanjut, ketergantungan *Long Range Wide Area Network* (LoRaWAN) pada *Network Server* pusat membuatnya kurang ideal untuk aplikasi yang memerlukan otonomi penuh. Untuk mengatasi kelemahan ganda tersebut, penelitian ini mengajukan sebuah arsitektur jaringan baru yang dibangun di atas tiga pilar konseptual. Pertama, penerapan topologi *Master-Slave* untuk menciptakan skema komunikasi terkelola yang secara efektif menghilangkan tabrakan data. Kedua, mekanisme konfigurasi dinamis yang memungkinkan jaringan untuk menyesuaikan parameter transmisi secara adaptif terhadap perubahan kondisi tautan, termasuk saat perangkat bergerak [7]. Ketiga, pemanfaatan *local server* sebagai simpul *edge* yang bertanggung jawab atas pemrosesan data dan manajemen jaringan, sehingga sistem dapat beroperasi sepenuhnya secara mandiri. Melalui sintesis dari ketiga pendekatan tersebut, tujuan utama penelitian ini adalah mengembangkan protokol komunikasi kustom di atas lapisan fisik LoRa yang memiliki tingkat keandalan, skalabilitas, dan efisiensi yang tinggi. Sistem yang diusulkan dirancang untuk mencapai *Packet Delivery Ratio* (PDR) yang superior dan latensi yang dapat diprediksi, dengan tetap menjaga konsumsi daya yang minimal pada perangkat *slave*. Arsitektur semacam ini memiliki relevansi tinggi untuk aplikasi-aplikasi krusial di Indonesia, seperti di sektor pertanian, mitigasi bencana, dan pemantauan infrastruktur,



di mana keandalan operasional dan kemandirian dari jaringan internet menjadi prasyarat utama.

## METODE

Langkah-langkah pelaksanaan penelitian ini digambarkan secara sistematis dalam Gambar 1, yang menunjukkan alur mulai dari kajian teori hingga pelaporan.



**Gambar 1.** Alur Pelaksanaan

Berdasarkan pada gambar diatas dapat dipaparkan sebagai berikut Langkah awal berupa pencarian dan kajian teori yang relevan terhadap topik atau masalah yang akan diselesaikan dengan menyesuaikan dengan parameter pembuatan alat. Merancang desain alat atau sistem yang akan dibuat. Pada tahapan ini mencakup skema kerja, pemilihan komponen, serta rancangan fisik dan fungsional prototipe. Tahap pengumpulan dan penyiapan semua kebutuhan fisik seperti komponen elektronik, sensor, bahan material, serta alat bantu untuk perakitan dan pengujian. Pengujian dilakukan untuk memastikan alat berfungsi dengan baik. Jika ditemukan kesalahan atau alat tidak bekerja sebagaimana mestinya, maka akan dilakukan perbaikan guna mengatasi error yang muncul sehingga proses penyempurnaan alat menjadi lebih efisien. Penyusunan laporan dilakukan berdasarkan hasil pengujian alat. Tahap ini bertujuan untuk menjelaskan dan menganalisis data sesuai dengan temuan yang diperoleh selama proses penelitian.

## Perancangan Sistem

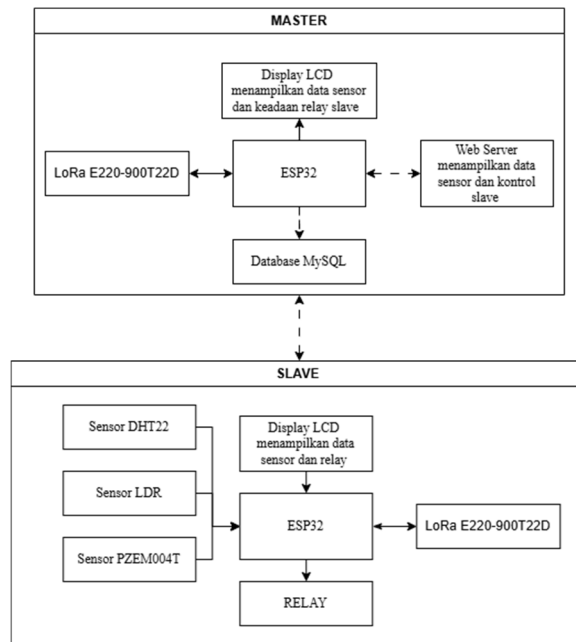
Tahapan perancangan sistem ini mencakup pemrograman modul ESP32 sebagai *Master* dan *Slave* serta modul LoRa untuk komunikasi data antar *node*, di mana data sensor dari setiap *Slave* dikirim secara berkala ke *Master*. Selanjutnya, data tersebut diteruskan secara *real-time* ke *server* lokal. Pada sisi *server*, perancangan ini secara spesifik memanfaatkan PHP sebagai bahasa pemrograman *backend*; sebuah skrip PHP berfungsi sebagai *Application Programming Interface (API) endpoint* untuk menerima, memproses, dan menyimpan data kiriman dari ESP32 ke *database* MySQL yang dikelola oleh XAMPP. Selain itu, PHP juga berperan krusial dalam mengambil data dari *database* untuk menampilkannya secara dinamis pada antarmuka web. Perancangan sistem yang detail dari perangkat keras hingga *software server* ini bertujuan untuk mengidentifikasi



kelemahan dan keunggulan alat, serta memastikan integrasi sistem berjalan secara optimal.

### Perancangan Umum Sistem

Sistem yang dirancang terdiri dari beberapa *node* yang dapat dikonfigurasi sebagai *Master* atau *Slave*. Komunikasi antar *node* menggunakan modul LoRa E220-900T22D tanpa memerlukan koneksi internet. Setiap *node* memiliki struktur perangkat keras dan perangkat lunak yang sama, namun perannya ditentukan melalui konfigurasi dalam kode program. *Node Master* bertugas mengatur komunikasi, menerima data, menampilkan data pada LCD, dan mengirimkannya ke *server* lokal. *Node Slave* bertugas membaca sensor dan merespons perintah dari *Master*. Visualisasi dari rancangan sistem *monitoring* dan kontrol ini dapat dilihat pada Gambar 2.



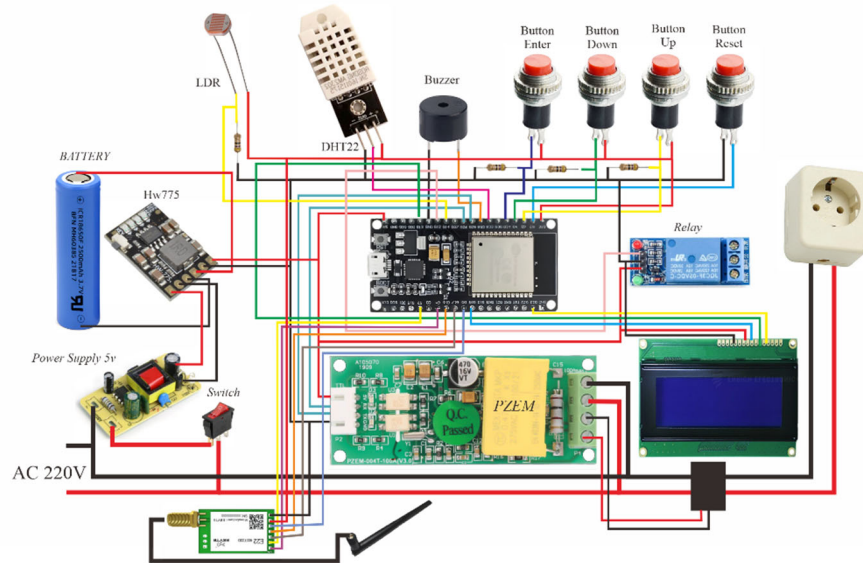
**Gambar 2.** Diagram Blok Perancangan *Master-Slave*. Keterangan: Garis putus-putus menunjukkan koneksi nirkabel (*wireless*), dan garis tebal menunjukkan koneksi kabel (*wired*)

### Perancangan Rangkaian Elektrikal

Tahap ini merupakan perancangan sistem instalasi komponen secara dua dimensi yang digunakan dalam pengembangan alat. Pada proyek ini, perangkat dirancang agar dapat berfungsi sebagai *Master* maupun *Slave* secara fleksibel, sehingga rangkaian *wiring* untuk setiap *node* dibuat seragam. Perancangan elektrikal mencakup pemasangan komponen utama seperti ESP32, modul LoRa E220-900T22D yang dilengkapi antena *omnidirectional* eksternal untuk optimasi jangkauan dan stabilitas sinyal, sensor DHT11,



sensor arus dan tegangan PZEM-004T v3.0, sensor cahaya LDR, modul relay, serta LCD I2C 20x4 sebagai tampilan data. Rangkaian ini juga dilengkapi dengan indikator LED untuk menandakan status sistem atau kondisi pengendalian. *Wiring diagram* ini berfungsi sebagai acuan pemasangan *hardware* agar integrasi antar-komponen dapat berjalan sesuai fungsi yang diharapkan. Detail koneksi antar komponen yang membentuk sistem dapat dilihat pada Gambar 3 yang menunjukkan wiring diagram perangkat.



Gambar 3. Wiring Diagram

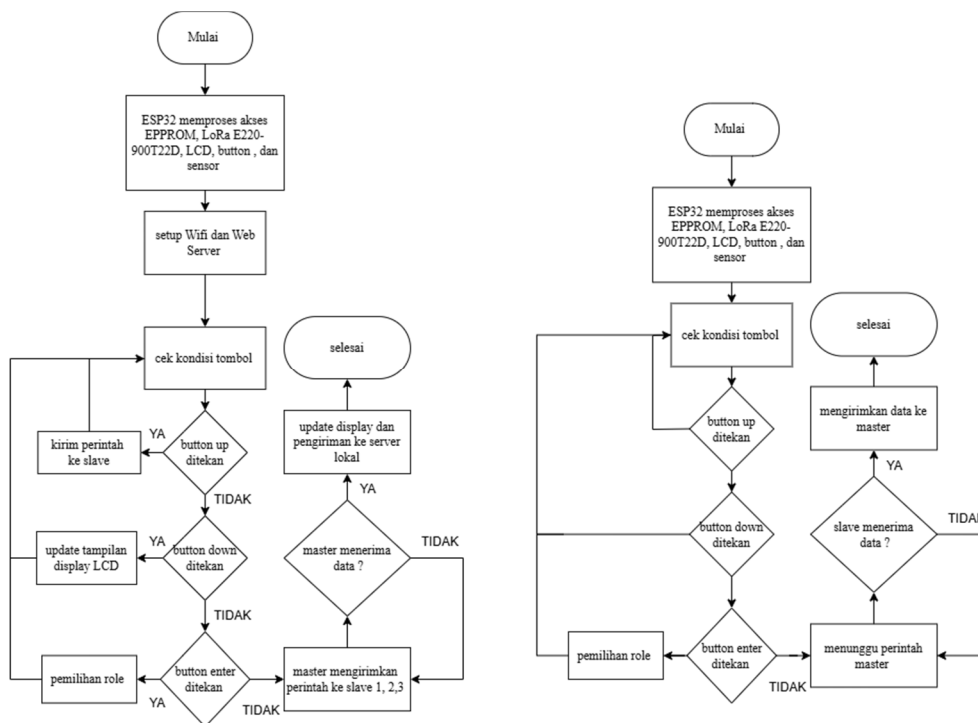
### Perancangan Perangkat Lunak

Tahapan ini merupakan proses perancangan perangkat lunak dengan menyusun program pada *mikrokontroler* ESP32 yang berperan sebagai pusat pengendali dalam sistem. Dalam sistem ini, setiap perangkat memiliki susunan wiring dan program dasar yang sama, namun peran sebagai *Master* atau *Slave* ditentukan berdasarkan konfigurasi di dalam kode program. Hal ini memberikan fleksibilitas agar perangkat dapat saling bertukar peran sesuai kebutuhan lapangan.

Perancangan perangkat lunak dibagi menjadi dua bagian utama. Pertama, perancangan program untuk perangkat yang dikonfigurasi sebagai *Master*, yang meliputi pengendalian komunikasi LoRa, pengiriman perintah kepada node *Slave*, penerimaan data dari *Slave*, dan pengiriman data sensor ke *server* lokal XAMPP melalui komunikasi serial. Selain itu, perangkat *Master* juga menangani tampilan data pada LCD 20x4 I2C serta antarmuka kontrol melalui *webserver* lokal. Pada sisi *server*, perancangan ini secara spesifik memanfaatkan PHP sebagai bahasa pemrograman *backend*; sebuah skrip PHP berfungsi sebagai *Application Programming Interface (API) endpoint* untuk menerima, memproses, dan menyimpan data kiriman dari ESP32 ke *database* MySQL yang dikelola oleh XAMPP



Kedua, perancangan program untuk perangkat yang dikonfigurasi sebagai *Slave*, yang bertugas membaca data dari sensor seperti DHT11, PZEM-004T v3.0, dan LDR, serta merespon perintah relay yang dikirim oleh *Master*. Data hasil pembacaan sensor akan dikirimkan kembali ke *Master* melalui komunikasi LoRa. Karena semua *node* memiliki *wiring* dan struktur program yang sama, perbedaan hanya terletak pada konfigurasi alamat dan peran (*Master/Slave*) yang diatur dalam kode. Rangkaian logika alur perangkat lunak untuk masing-masing peran dapat dilihat pada Gambar 4.(a) untuk *Node Master* dan Gambar 4.(b) untuk *Node Slave*.



**Gambar 4.** Rangkaian Logika Alur Perangkat Lunak (a) *Master* dan (b) *Slave*

### Pembuatan Sistem

Pembuatan alat pada proyek ini terdiri dari beberapa tahapan utama, mulai dari pemrograman *mikrokontroler* menggunakan ESP32 hingga integrasi sistem *monitoring* dan kontrol berbasis LoRa. Proses ini bertujuan agar data yang diperoleh dari sensor pada masing-masing *node* dapat diproses, dikendalikan, dan dikirimkan ke *server* lokal (XAMPP) secara *real-time*. Perangkat ESP32 pada sistem ini dirancang agar dapat dikonfigurasi sebagai *Master* ataupun *Slave*, sehingga *wiring* dan struktur perangkat keras pada setiap unit dibuat seragam.



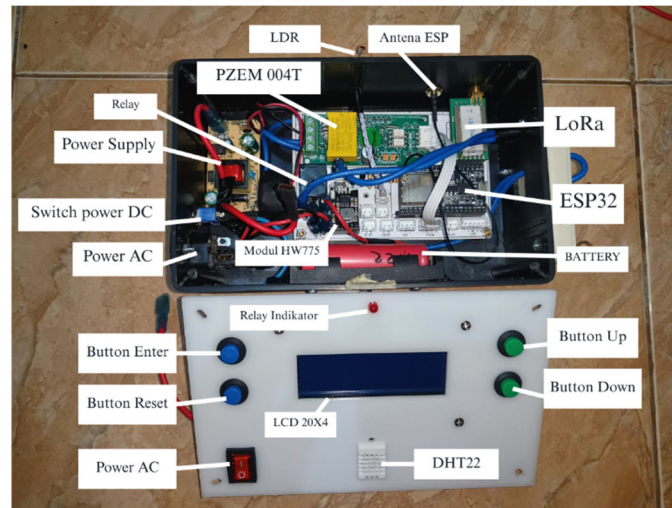
### **Pembuatan Rangkaian Elektrikal**

Pembuatan rangkaian elektrikal dilakukan melalui beberapa tahapan yang terstruktur. Tahap pertama dimulai dengan mengidentifikasi komponen-komponen yang diperlukan, seperti ESP32, modul LoRa E220-900T22D, sensor suhu dan kelembapan DHT11, sensor energi PZEM-004T, sensor cahaya LDR, modul relay, dan LCD I2C 20x4. Selanjutnya, seluruh komponen tersebut dirakit ke dalam box dengan ukuran yang seragam, mencakup tiga unit ESP32 yang dirancang agar dapat difungsikan baik sebagai *Master* maupun *Slave*. Proses perakitan dilakukan dengan mengikuti *wiring diagram* yang telah dirancang sebelumnya, memastikan setiap koneksi terpasang dengan benar. Setelah itu, dilakukan pengujian terhadap fungsi masing-masing komponen, termasuk sensor suhu dan kelembapan, sensor listrik, tampilan pada LCD, serta proses komunikasi data melalui modul LoRa untuk memastikan semua sistem berjalan sesuai harapan. Setiap perangkat, baik yang berperan sebagai *Master* maupun *Slave*, menggunakan susunan wiring yang identik, sementara perbedaan peran ditentukan melalui konfigurasi dalam program. Untuk mempermudah pengaturan koneksi antar komponen, dilakukan pula perancangan skematik dan layout PCB menggunakan software EasyEDA. Rangkaian PCB ini dirancang agar jalur koneksi lebih ringkas, efisien, dan meminimalkan kesalahan sambungan kabel selama proses perakitan. Desain ini memungkinkan pemasangan perangkat secara praktis di lapangan, dengan penempatan komponen yang rapi dan sistematis.

### **Pembuatan Rangkaian Mekanik**

Setelah perakitan rangkaian elektrikal selesai, perangkat dimasukkan ke dalam box tertutup untuk memberikan perlindungan dari faktor lingkungan seperti debu dan kelembapan. Proses ini bertujuan agar alat dapat digunakan dalam jangka panjang dan dapat dipasang secara fleksibel di berbagai lokasi. Rangkaian komponen yang telah dirakit ke dalam box pelindung ditunjukkan pada Gambar 5.





**Gambar 5.** Rangkaian dalam *box*

### **Pembuatan Perangkat Lunak (Software)**

Perangkat lunak dibuat menggunakan Arduino IDE untuk semua unit ESP32. Karena setiap perangkat memiliki potensi untuk berperan sebagai *Master* maupun *Slave*, maka struktur program dasarnya dibuat seragam, dengan perbedaan peran ditentukan oleh konfigurasi variabel di awal program. Pada mode *Slave*, ESP32 akan membaca data dari sensor-sensor (DHT11, PZEM, LDR) dan menunggu perintah dari *Master* sebelum mengirimkan data melalui LoRa. Pada mode *Master*, ESP32 bertugas mengirimkan perintah ke *node Slave*, menerima data, menampilkannya pada LCD, serta meneruskan data ke *server* lokal (XAMPP) melalui komunikasi serial. Komunikasi antar perangkat menggunakan protokol string sederhana, dengan sistem alamat tetap yang ditentukan dalam kode. Seluruh data yang diterima oleh *Master* dikirim ke *server* lokal menggunakan PHP dan MySQL, dan dapat diakses melalui web browser secara *real-time*. Perangkat lunak ini juga memperhitungkan stabilitas sinyal LoRa dan memastikan pengiriman data dilakukan secara teratur dan efisien, tanpa tergantung pada koneksi internet.

### **Pengujian Alat dan Perbaikan**

Tahapan pengujian sistem dilakukan untuk mengevaluasi performa dan keandalan perangkat dalam melakukan proses *monitoring* dan kontrol berbasis komunikasi LoRa. Pengujian ini mencakup aspek perangkat keras, perangkat lunak, serta komunikasi antar *node*. Dalam sistem ini, pengujian dilakukan dengan mengoperasikan dua *node* ESP32 sebagai *Slave* dan satu perangkat sebagai *Master*, yang semuanya terhubung melalui protokol LoRa dan data dikirimkan ke *server* lokal berbasis XAMPP. Perangkat diuji pada dua lokasi berbeda di dalam area kampus untuk mensimulasikan lingkungan yang



memiliki jarak fisik terpisah antar *node*. *Master* ditempatkan di lokasi pusat (dekat laptop *server*), sedangkan masing-masing *Slave* diposisikan pada titik yang berbeda, guna mengevaluasi kestabilan komunikasi LoRa dalam kondisi nyata. *Node Slave* akan membaca data dari sensor dan mengirimkan data ke *Master* secara berkala, kemudian data yang diterima ditampilkan pada LCD dan dikirimkan ke *server* lokal.

### Analisa Data

Tahap analisis data dilakukan berdasarkan hasil pengujian sistem yang telah dilaksanakan, dengan fokus khusus pada aspek komunikasi data antara perangkat *Master* dan *Slave*. Analisis ini menekankan dua parameter utama, yaitu nilai *delay* komunikasi dan tingkat *packet loss*. Hasil analisis ini digunakan untuk mengevaluasi kinerja sistem dalam berbagai kondisi lingkungan dan jarak komunikasi.

*Delay* komunikasi diukur sebagai selisih waktu antara saat perintah dikirim dari *Master* dan saat respon diterima kembali dari *Slave*. Pengukuran *delay* ini dilakukan untuk mengevaluasi seberapa cepat sistem dapat merespons instruksi dan mengembalikan data. *Delay* diukur dalam satuan detik, dan pengujian dilakukan pada beberapa skenario jarak serta kondisi lingkungan berbeda. Rumus perhitungan *delay* adalah sebagai berikut:

$$Delay = t_1 - t_0 \quad (1)$$

Nilai rata-rata *delay* juga dihitung untuk memberikan gambaran umum tentang performa respon sistem. *Packet loss* dianalisis untuk mengetahui seberapa handal sistem dalam mentransmisikan data antar *node*. Pengujian dilakukan dengan mengirim sejumlah paket data secara berulang, kemudian dihitung berapa banyak paket yang gagal diterima. Rumus perhitungan *packet loss* sebagai berikut:

$$Packet Loss (\%) = \frac{|jumlah\ paket\ hilang|}{jumlah\ paket\ Dikirim} \times 100\% \quad (2)$$

Evaluasi dilakukan dengan variasi jarak serta kondisi lingkungan baik tanpa penghalang (*non-obstacle*) maupun dengan penghalang (*obstacle*).

## HASIL

### Hasil Pengujian Vertikal

#### Hasil Pengujian Vertikal Slave 1

Pengujian jarak vertikal ini dilakukan untuk mengetahui sejauh mana kemampuan modul komunikasi LoRa dalam mengirimkan dan menerima data antar lantai di dalam bangunan bertingkat. Dalam pengujian ini, pengirim (*master*) berada di lantai 1, sedangkan penerima (*slave 1*) diletakkan secara vertikal di lantai 2, lantai 3, dan lantai 4.



Lingkungan pengujian dilakukan di gedung H Universitas Muhammadiyah Surakarta (UMS) dengan struktur bangunan berupa beton yang berpotensi melemahkan sinyal LoRa. Data yang dikumpulkan meliputi jumlah paket yang dikirim dan diterima, waktu tunda (*delay*) menggunakan rumus persamaan 1, serta persentase kehilangan paket (*packet loss*) menggunakan rumus persamaan 2 untuk masing-masing jarak vertikal.

Tabel 1. Pengujian Vertikal slave 1 dan 2

Jarak Vertikal (m)	Jumlah paket dikirim	paket diterima slave 1	paket diterima slave 2	Delay slave 1(s)	Delay slave 2(s)	Packet Loss 1 (%)	Packet Loss 2 (%)	Halangan
5	20	18	17	6,55	6,55	10	10	BETON
10	20	11	5	26,15	181,7	45	75	BETON
15	20	7	3	7,125	7	65	80	BETON

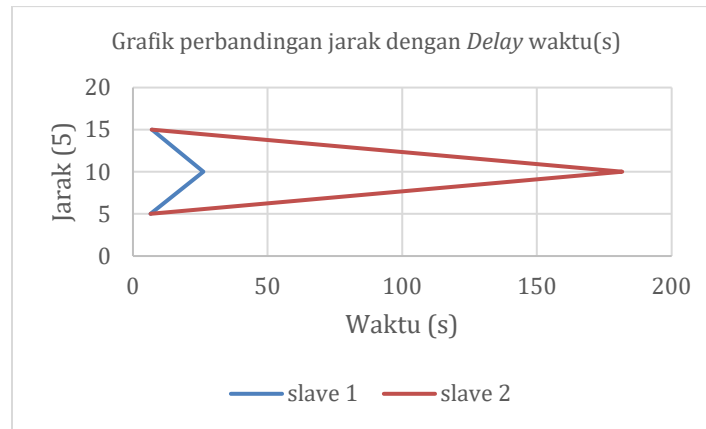
Tabel 1 menyajikan hasil pengujian komunikasi vertikal menggunakan teknologi LoRa antara perangkat *master* dan *slave* pada beberapa ketinggian berbeda.



Gambar 6. Grafik Perbandingan Jarak dengan Jumlah Paket yang Diterima

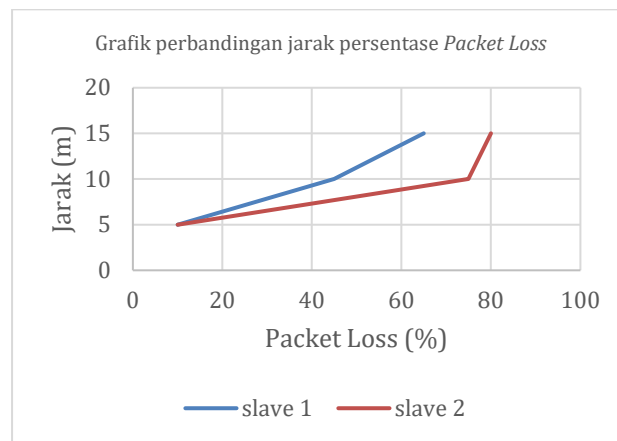
Grafik gambar 6 menunjukkan grafik hubungan antara jarak vertikal dan jumlah paket yang diterima yang menunjukkan bahwa semakin jauh jarak vertikal, jumlah paket yang diterima oleh kedua perangkat dari total 20 paket yang dikirim semakin menurun. Pada jarak awal 5 meter, kinerja keduanya hampir serupa, di mana *slave 1* menerima 18 paket dan *slave 2* menerima 17 paket. Namun, performa *slave 1* terbukti jauh lebih unggul pada jarak 10 meter saat hanya menerima 11 paket, sementara *slave 2* anjlok ke 5 paket. Pada akhirnya di jarak 15 meter, *slave 1* yang lebih superior pun hanya menerima 7 paket, dan *slave 2* hanya 3 paket.





**Gambar 7.** Grafik Perbandingan Jarak dengan *Delay* Waktu(S)

Grafik pada Gambar 7 yang diambil dari data Tabel 1 menunjukkan bahwa pada jarak vertikal 5 meter, *delay* komunikasi di kedua slave relatif stabil sekitar 6,5 detik dengan *packet loss* 10%. Namun di ketinggian 10 meter, *delay* terutama pada *Slave 2* meningkat tajam hingga 181 detik, diikuti *packet loss* yang juga melonjak signifikan. Pada jarak 15 meter, *delay* turun kembali mendekati 7 detik, meskipun *packet loss* masih tinggi.



**Gambar 8.** Grafik Perbandingan Jarak Persentase *Packet Loss*

Gambar 8 memperlihatkan grafik perbandingan jarak vertikal dengan persentase *packet loss*, di mana paket dianggap diterima jika *delay* di bawah 10 detik. Pada ketinggian 5 meter, sistem masih berjalan baik dengan *slave 1* menerima 18 paket dan *slave 2* menerima 17 paket, *delay* rata-rata sekitar 6,5 detik, serta *packet loss* 10%. Ketika jarak meningkat ke 10 meter, kinerja menurun signifikan dengan hanya 11 paket diterima oleh *slave 1* dan 5 oleh *slave 2*, serta *delay* sangat tinggi hingga 181 detik pada *slave 2*. Pada 15 meter, kualitas komunikasi semakin merosot dengan *packet loss* mencapai 65%.



### Hasil Pengujian Horizontal

Pengujian horizontal dilakukan di area sekitar Kampus 2 Universitas Muhammadiyah Surakarta dengan konfigurasi jarak 5 meter, 10 meter, 50 meter, dan 100 meter.

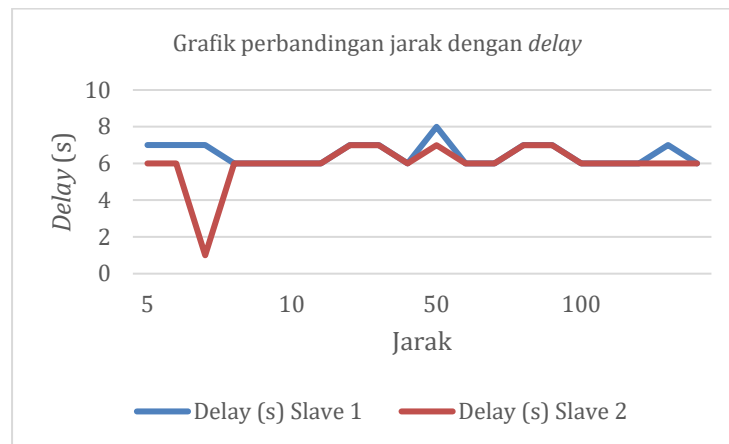
Tabel 2. Pengujian Horizontal Tanpa Obstacle

Jarak (m)	Waktu Kirim (t0) Slave 1	Waktu Terima (t1) Slave 1	Delay (s) Slave 1	Waktu Kirim (t0) Slave 2	Waktu Terima (t1) Slave 2	Delay (s) Slave 2
5	17:08:03	17:08:10	7	17:10:03	17:10:09	6
	17:08:33	17:08:40	7	17:10:33	17:10:39	6
	17:09:03	17:09:10	7	17:11:14	17:11:15	1
	17:09:33	17:09:39	6	17:11:38	17:11:44	6
	17:10:03	17:10:09	6	17:12:08	17:12:14	6
10	17:33:28	17:33:34	6	17:34:37	17:34:43	6
	17:33:58	17:34:04	6	17:35:07	17:35:13	6
	17:34:27	17:34:34	7	17:35:36	17:35:43	7
	17:34:57	17:35:04	7	17:36:06	17:36:13	7
	17:35:27	17:35:33	6	17:36:36	17:36:42	6
50	12:06:16	12:06:24	8	12:06:55	12:07:02	7
	12:06:47	12:06:53	6	12:07:25	12:07:31	6
	12:07:16	12:07:22	6	12:07:55	12:08:01	6
	12:07:45	12:07:52	7	12:08:24	12:08:31	7
	12:08:15	12:08:22	7	12:08:54	12:09:01	7
100	12:36:31	12:36:37	6	12:40:46	12:41:51	6
	12:37:01	12:37:07	6	12:41:15	12:42:21	6
	12:37:31	12:37:37	6	12:41:45	12:42:51	6



12:38:02	12:38:09	7	12:42:15	12:43:21	6
12:38:34	12:38:40	6	12:42:45	12:43:51	6

Tabel 2 menampilkan hasil pengujian delay dalam berbagai skenario jarak horizontal tanpa hambatan. Data ini memperlihatkan bahwa hasil pengujian modul komunikasi LoRa pada jarak 5 sampai 100 meter tanpa hambatan fisik antara perangkat *master* dengan dua perangkat penerima, yaitu *slave 1* dan *slave 2*.



**Gambar 9.** Grafik Perbandingan Jarak dengan *Delay*

Gambar 9 menunjukkan grafik hubungan perbandingan rata-rata *delay* antara *Slave 1* dan *Slave 2* dalam uji komunikasi horizontal tanpa adanya hambatan di lingkungan kampus UMS. Terlihat bahwa *Slave 1* mengalami variasi *delay* yang sedikit naik terutama pada jarak 50 meter, dengan nilai tertinggi mendekati 8 detik, sementara pada jarak 5, 10, dan 100 meter *delay*-nya berada di rentang 6–7 detik. Berbeda dengan *Slave 1*, nilai *delay* *Slave 2* tampak lebih stabil di seluruh jarak uji, berkisar antara 6 hingga 7 detik tanpa perubahan yang berarti. Hal ini menandakan bahwa performa komunikasi *Slave 2* lebih konsisten dibandingkan *Slave 1*, meskipun secara keseluruhan kedua perangkat masih memiliki waktu respon yang wajar dan dapat diterima pada sistem komunikasi LoRa *multi-node*. Dengan demikian, grafik ini mengonfirmasi bahwa jarak horizontal tanpa *obstacle* tidak memengaruhi kestabilan *delay* secara signifikan dalam jaringan LoRa yang diimplementasikan.

**Tabel 3.** Pengujian Respons Button

Jarak (m)	Delay (s) Slave 1	Delay (s) Slave 2	Rata-rata Slave 1	Rata-rata slave 2	Obstacle
-----------	-------------------	-------------------	-------------------	-------------------	----------



	3,55	3,37			
	2,96	3,64			
50	3,53	3,25	3.38	4.9	
	3,2	3,70			
	3,74	3,47			
	3,31	11,97			
	24,19	4,9			Pohon, motor,
100	21,94	11,47	14.59	7.08	bangunan
	20,88	3,54			
	3,45	3,28			
	4,47	12,31			
	23,4	7,74			
200	23,94	12,76	15.3	10.75	
	8,42	5,45			
	5,44	17,08			
250					

Tabel 3 menunjukkan hasil uji respon tombol dalam berbagai sekenario jarak. Pada jarak 50 meter, rata-rata *delay Slave 1* dan *Slave 2* masih stabil di sekitar 3,38 detik dan 4,9 detik. Namun, ketika jarak diperluas hingga 100 meter, rata-rata *delay* meningkat, terutama pada *Slave 1* yang mencapai 14,59 detik, kemungkinan dipengaruhi gangguan kendaraan yang lewat. Di jarak 200 meter, *delay* rata-rata juga naik hingga 15,3 detik untuk *Slave 1* dan 10,75 detik untuk *Slave 2*, karena adanya hambatan seperti pohon, kendaraan, atau bangunan. Di jarak 250 meter *slave 1* dan *slave 2* sudah tidak bisa menerima perintah dari master. Secara umum, hasil ini menegaskan bahwa respon tombol tetap baik pada jarak pendek, tetapi performanya menurun di jarak jauh akibat halangan di lingkungan uji dan pada jarak 250 meter *slave* tidak bisa menerima data dari *master*.



## PEMBAHASAN

Pengujian komunikasi LoRa pada penelitian ini dilakukan di gedung H Fakultas Teknik Universitas Muhammadiyah Surakarta untuk menguji performa komunikasi secara vertikal antar lantai, sehingga dapat mensimulasikan kondisi sebenarnya di lingkungan bertingkat dengan adanya sekat atau dinding bangunan. Selain itu, pengujian juga dilakukan pada jarak horizontal 5, 10, dan 15 meter di berbagai lokasi, baik di area terbuka tanpa penghalang maupun area dengan obstacle seperti dinding atau peralatan besar. Uji coba ini berfokus pada parameter *delay* komunikasi dan tingkat *packet loss* antara perangkat *Master* dan *Slave*, agar dapat memberikan gambaran kehandalan sistem *monitoring* dan kontrol dalam situasi nyata.

## KESIMPULAN

Dari serangkaian pengujian yang dilakukan, sistem komunikasi *multi-node* berbasis LoRa dengan konfigurasi *master-slave* dinamis yang terhubung ke *server* lokal berhasil dirancang dan diimplementasikan untuk mendukung proses *monitoring* serta kontrol data secara mandiri. Pada skenario horizontal tanpa penghalang hingga jarak 10 meter, sistem menunjukkan performa optimal dengan rata-rata *delay* di bawah 7 detik dan *packet loss* 0%. Saat pengujian dilakukan dengan *obstacle* di jalur horizontal, performa komunikasi masih terjaga meskipun terdapat sedikit kenaikan *delay* akibat aktivitas kendaraan di sekitar lokasi dan ketika mencapai jarak 250 meter *node slave* tidak bisa menerima data dari *master*. Pada uji vertikal antar lantai, performa sistem menurun cukup tajam. Pada jarak vertikal 5 meter, komunikasi berjalan stabil dengan rata-rata *delay* 6–7 detik dan *packet loss* di bawah 10%. Namun, ketika diuji pada 10–15 meter (hingga empat lantai), terjadi lonjakan *delay* signifikan terutama pada *Slave 2* hingga 181 detik, dan *packet loss* meningkat sampai 65% untuk *Slave 1* serta 80% pada *Slave 2*, akibat penghalang material beton. Untuk keseluruhan pengujian menggunakan antena penguat berspesifikasi 5 dBi 915 MHz. Secara keseluruhan, pendekatan *master-slave* dengan *server* lokal ini terbukti layak sebagai alternatif sistem *monitoring* dan kontrol yang fleksibel serta mandiri tanpa internet. Meskipun demikian, desain jaringan LoRa perlu mempertimbangkan kondisi lingkungan fisik seperti tembok beton dan potensi gangguan dinamis dari kendaraan yang lewat, karena dapat mempengaruhi kualitas komunikasi terutama dalam hal *delay* dan *packet loss*.

## Saran

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa saran untuk pengembangan Sistem Monitoring dan Kontrol Multi-Node Berbasis LoRa dengan Integrasi Local Server, antara lain sebagai berikut: Optimize sistem komunikasi antara perangkat master dan slave agar transmisi data lebih stabil dan responsif. Perancangan Elektrikal yang lebih matang dan lebih baik. Meningkatkan integrasi antara



web server, perangkat (*device*), dan lokal server untuk memastikan sinkronisasi data berjalan secara real-time dan tanpa gangguan.

## UCAPAN TERIMA KASIH

Dengan memanjatkan puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, hidayah dan kesehatan sehingga penulis bisa menyelesaikan Tugas Akhir ini. Dengan bantuan dan bimbingan dari berbagai pihak penulis ingin menyampaikan banyak terimakasih kepada:

1. kepada Allah SWT yang telah memberikan kesehatan, kekuatan, dan kemudahan dalam setiap langkah hingga Tugas Akhir ini dapat diselesaikan dengan baik.
2. Kedua orang tua yang senantiasa memberi do'a dan dukungan materi kepada penulis sehingga dapat menyelesaikan Tugas Akhir ini.
3. Bapak Dedy Ary Prasetya, S.T., M.Eng. selaku dosen pembimbing yang telah memberikan arahan dan bimbingan.
4. Farid Ahmad Fauzan dan Hasby Asshodiq yang telah memberikan arahan, bantuan, dan bimbingan.
5. Kepada Iqbal Reza Pradana, Dzaky Novika Ramadhan, Muhammad Afin, Catur Putra Aprilianto, Abdillah Khalid, Soraya Komala Firdaus, Nadia Intani Zalbila, Naufal Gian Syafi, Farel Nugroho, Hafiz Atsal Febriyanto, Mohammad Addo P, dan Rozan Iqbal Haq karena telah memberikan semangat, bantuan, dan dukungan untuk mengerjakan Tugas Akhir ini.
6. Teman-teman Asisten Teknik Elektro yang selalu membantu dan memberikan dukungan dalam pengerjaan Tugas Akhir.
7. Kepada teman-teman yang telah kebersamai dari awal perkuliahan hingga menyelesaikan Tugas Akhir ini.
- 8.

## DAFTAR PUSTAKA

- [1] D. A. Verde Romero, E. Villalvazo Laureano, R. O. Jiménez Betancourt, and E. Navarro Álvarez, "An open source IoT edge-computing system for monitoring energy consumption in buildings," *Results Eng.*, vol. 21, no. February, 2024, doi: 10.1016/j.rineng.2024.101875.
- [2] I. Lopes *et al.*, "LoRa-based IoT platform for remote soil parameter monitoring," *Dyna*, vol. 91, no. 231, pp. 86–93, 2024, doi: 10.15446/dyna.v91n231.111612.
- [3] M. H. Widiyanto, "Systematic Literature Review: LoRa and LoraWAN projects in practice," *J. Softw. Eng. Inf. Commun. Technol.*, vol. 3, no. 1, pp. 25–34, 2022, doi: 10.17509/seict.v3i1.47780.
- [4] M. O. Farooq, "Multi-hop communication protocol for LoRa with software-defined



- networking extension," *Internet of Things (Netherlands)*, vol. 14, no. February, 2021, doi: 10.1016/j.iot.2021.100379.
- [5] N. Chinchilla-Romero, J. Navarro-Ortiz, P. Muñoz, and P. Ameigeiras, "Collision avoidance resource allocation for LoRaWAN," *Sensors (Switzerland)*, vol. 21, no. 4, pp. 1–19, 2021, doi: 10.3390/s21041218.
- [6] R. Kufakunesu, G. P. Hancke, and A. M. Abu-Mahfouz, "Collision Avoidance Adaptive Data Rate Algorithm for LoRaWAN," *Futur. Internet*, vol. 16, no. 10, pp. 1–19, 2024, doi: 10.3390/fi16100380.
- [7] A. Nouar, M. T. Abbas, S. Boumerdassi, and M. Chaib, "Impact of Mobility Model on LoRaWAN Performance," *J. Commun.*, vol. 19, no. 1, pp. 7–18, 2024, doi: 10.12720/jcm.19.1.7-18.

