

EVALUASI KINERJA SISTEM WSN BERBASIS ESP32 DENGAN TOPOLOGI *STAR* DAN *MESH MULTI-HOP*

Ahmad Ikhyari¹, Rahmawati Hasanah², Sultan Arya Kandi³

^{1,2,3}Jurusan Teknik Elektro, Politeknik Negeri Bandung

Email: ahmad.ikhyari@polban.ac.id¹, rahmawati@polban.ac.id², sultan.arya.tkom422@polban.ac.id³

Corresponding Author : Ahmad Ikhyari

Jurusan Teknik Elektro, Politeknik Negeri Bandung

Email: ahmad.ikhyari@polban.ac.id

Abstrak – Salah satu solusi infrastruktur *Internet of Things* (IoT) yang efisien dan berbiaya rendah adalah *Wireless Sensor Network* (WSN), yang memungkinkan komunikasi antar-*node* secara mandiri dan fleksibel. Metodologi penelitian meliputi perancangan arsitektur sistem, implementasi perangkat keras dan perangkat lunak, serta pengujian kinerja jaringan. Sistem diterapkan menggunakan enam unit ESP32 yang terdiri dari lima *node* sensor dan satu *gateway*. Topologi *mesh* direalisasikan menggunakan *painlessMesh* dengan mekanisme *multi-hop* dan *self-healing*, sedangkan topologi *star* menggunakan protokol ESP-NOW untuk komunikasi langsung berlatensi rendah. Topologi *star* mengonfigurasi *node* dengan *MAC address gateway* tujuan, sedangkan topologi *mesh* tidak memerlukan *MAC address gateway*, terkait mekanisme *multi-hop* dan *broadcast* ke sejumlah *node* tetangganya. Khusus untuk topologi *mesh*, ditinjau juga ketahanan jaringan, yakni kinerja jaringan ketika satu atau dua *node* relai dimatikan. Pengujian dilakukan dengan membandingkan performa kedua topologi menggunakan parameter *Packet Delivery Ratio* (PDR), Latensi, dan *Received Signal Strength Indicator* (RSSI) dalam kondisi jaringan yang sama. Hasil penelitian menunjukkan bahwa masing-masing topologi memiliki karakteristik yang berbeda, di mana topologi *star* unggul dalam latensi rendah, terutama di kondisi tanpa halangan, yakni mencapai 1,114 ms pada interval pengiriman 5 detik, berbeda dengan topologi *mesh* yang memiliki latensi lebih tinggi, yaitu 80,098 ms. Di sisi lain, topologi *mesh* lebih andal dalam jangkauan dan toleransi terhadap kegagalan *node*. Pada kondisi dengan halangan pada *node* 1-5, topologi *mesh* mampu mengirimkan paket sampai ke tujuan dengan PDR konsisten, sekitar 96,77%, dengan RSSI yang cenderung lebih stabil, yaitu di kisaran -71 sampai -67 dBm. Topologi *star* cenderung memiliki PDR yang tidak konsisten yakni bernilai 100% di *node* 1 dan *node* 2, namun turun drastis menjadi 81,08% di *node* 3, serta mengalami kegagalan total di *node* 4 dan 5, dengan RSSI yang cenderung lebih rendah hingga menyentuh -96,23 dBm di *node* 3. Pemilihan topologi jaringan WSN mesti disesuaikan dengan kebutuhan aplikasi, sehingga hasil analisis dapat digunakan sebagai rekomendasi dalam implementasi IoT di lingkungan luas atau terhalang.

Kata-kata kunci: WSN, ESP32, Topologi Mesh, Topologi Star, Evaluasi Kinerja IoT.

Abstract – Among possible solutions of efficient, low-cost, and wide coverage *Internet of Things* (IoT) infrastructure is a *Wireless Sensor Network* (WSN), which enables autonomous and flexible communication between sensor nodes. The research methodology includes system architecture design, hardware and software implementation, and network performance testing. The system consists of six ESP32 units, including five sensor nodes and one gateway. The mesh topology is implemented using *painlessMesh* with *multi-hop* and *self-healing* capabilities, while the star topology uses the ESP-NOW protocol for low-latency direct communication. It configures each node with targeted MAC address gateway, whereas the mesh topology does not need any MAC address gateway, which is related to *multi-hop* and *broadcast* mechanism towards some adjoined nodes. Particularly for mesh topology, network reliability, which considers events of one or two switched-off relay nodes, is also inspected. Performance evaluation is conducted using parameters such as *Packet Delivery Ratio* (PDR), Latency, and *Received Signal Strength Indicator* (RSSI), under identical conditions. The results indicate that each topology has distinct characteristics, where the star topology excels in low latency, of 1.114 ms with the sending interval of 5 seconds, in contradiction with the mesh one, which suffers higher latency, of 80.098 ms. Apart from that, the mesh topology is more reliable in coverage and tolerance of nodes' failure. At node 1-5 during non-line-of-sight condition, it is capable to send packets to each destination with consistent value of PDR, of around 96.77% with stable RSSI interval of -71 and -67 dBm. Conversely, the star one tends to have inconsistent PDR value; it has PDR of 100% at node 1 and 2, and drops to be 81.08% at node 3, together with total failure at node 4 and 5, with low RSSI value of -96.23 dBm at node 3. In summary, the selection of WSN topology should be tailored to application requirements, and the findings can serve as a reference for IoT implementation in wide or obstacle-rich environments.

Keywords: WSN, ESP32, Mesh Topology, Star Topology, IoT Performance Evaluation.

I. PENDAHULUAN

Integrasi jaringan sensor nirkabel atau *Wireless Sensor Network* (WSN) dalam ekosistem *Internet of Things* (IoT) sangat penting untuk pemantauan lingkungan secara *real-time* [1]. Namun, tantangan teknis seperti skalabilitas, latensi, konsumsi daya, dan hambatan fisik (*Non-Line of Sight/NLOS*) menghambat konektivitas masif, terutama di lingkungan industri yang memiliki redaman sinyal tinggi [1]. Arsitektur jaringan ESP32 yang tepat mendesak untuk menjamin keandalan [2]. Topologi *Star* unggul dalam kesederhanaan dan rendahnya *latency*, namun terbatas pada jangkauan komunikasi langsung ke *gateway* [3]. Sebaliknya, topologi *Mesh Multi-Hop* menawarkan jangkauan luas dan fitur *self-healing* yang meningkatkan ketahanan operasional jaringan, meskipun mekanisme *multi-hop*-nya berdampak pada *latency* yang justru lebih tinggi [4].

Protokol ESP-NOW pada ESP32 mampu menjaga keandalan tinggi dan *latency* rendah pada jangkauan menengah atau cukup jauh tanpa ketergantungan pada *router* Wi-Fi eksternal [5]. Namun, penggunaan topologi *Mesh* pada perangkat hemat daya terhambat oleh batasan memori *buffer*, yang berdampak langsung pada stabilitas pengiriman data, sehingga membutuhkan penempatan jaringan yang cermat [4]. Meskipun ESP-NOW efisien secara *point-to-point* [5] dan WSN membutuhkan arsitektur hemat daya [1], studi literatur yang secara spesifik menganalisis perbandingan performa antara topologi *Star* berbasis ESP-NOW dan topologi *Mesh* berbasis *multi-hop* dinamis pada kondisi lingkungan yang sama masih terbatas. Integrasi evaluasi *latency* rendah ESP-NOW [5] dengan taksonomi tantangan WSN [1] melahirkan gagasan untuk membandingkan performa efisiensi *point-to-point* dengan ketahanan topologi jaringan lokal.

Berdasarkan urgensi tersebut, penelitian ini bertujuan melakukan analisis komparatif performa topologi *Star* berbasis ESP-NOW dan topologi *Mesh* berbasis *painlessMesh* menggunakan mikrokontroler ESP32. Pengujian akan membandingkan kinerja kedua topologi melalui parameter *Packet Delivery Ratio* (PDR), *latency*, dan *Received Signal Strength Indicator* (RSSI), baik pada kondisi tanpa halangan (*Line of Sight/LOS*) maupun terhalang hambatan fisik (NLOS). Penelitian diharapkan memberikan rekomendasi praktis bagi pemilihan topologi WSN yang tepat dan efisien sesuai karakteristik kebutuhan implementasinya.

Dalam penelitian ini, jumlah *node* yang digunakan dalam sistem adalah enam unit yang terdiri dari lima *node* sensor dan satu *gateway*. Topologi jaringan yang diuji hanya terbatas pada topologi *mesh* dan topologi *star*. Protokol komunikasi yang digunakan adalah *painlessMesh* untuk topologi *mesh* dan ESP-NOW untuk topologi *star*. Selain itu, pengujian ini dilakukan dalam lingkungan yang terbatas, sehingga belum mencakup seluruh variasi kondisi jaringan yang mungkin terjadi. Sistem tidak membahas aspek keamanan jaringan secara mendalam seperti enkripsi tingkat lanjut. Aktivasi *node* sensor dilakukan secara bertahap, sehingga pengambilan data didasarkan pada paket yang berhasil diterima pada

setiap *node*. Pengujian sistem pada penelitian ini dilakukan menggunakan penyangga (*tripod*) dengan nilai ketinggian dari permukaan tanah sekitar 1 meter. Untuk pengujian pada lingkungan yang berbeda, kondisi penyangga dan lingkungan pengujian dapat berbeda karena menyesuaikan kondisi lingkungan setempat.

Meskipun protokol ESP-NOW pada jaringan *star* efisien secara *point-to-point*, dan jaringan *mesh* menawarkan fitur *self-healing* yang andal, penelitian terdahulu umumnya mengevaluasi kedua pendekatan ini secara terpisah, atau hanya sebatas pada simulasi dalam ranah perangkat lunak. Oleh karena itu, *research gap* dalam penelitian ini adalah belum adanya kajian empiris yang membandingkan secara langsung batas kinerja topologi *star* (berbasis ESP-NOW) dan topologi *mesh* (berbasis *painlessMesh*) di bawah kondisi lingkungan dan spesifikasi perangkat keras yang identik. Sebagai *novelty* (kebaruan) serta kontribusi utama, penelitian ini merumuskan evaluasi kinerja secara komprehensif berdasarkan kombinasi pengujian kondisi bebas tanpa halangan (LOS) dan terhalang vegetasi (NLOS), variasi interval transmisi data, perbandingan metrik PDR, latensi, dan RSSI, serta uji ketahanan jaringan secara mandiri (*self-healing*) pada topologi *mesh* secara spesifik saat terjadi kegagalan (*node failure*) pada satu hingga dua *node* relai.

II. TINJAUAN PUSTAKA

Kajian ini membahas perkembangan *Wireless Sensor Network* (WSN) berbasis *Internet of Things* (IoT), dengan fokus pada aspek topologi jaringan, mekanisme komunikasi, serta parameter yang digunakan untuk mengevaluasi kinerja jaringan.

Penelitian ini mengacu pada hasil penelitian terdahulu yang telah dipublikasikan oleh Armanda [6] di lingkungan institusi pengusul. Penelitian tersebut berfokus pada rancangan dan implementasi WSN menggunakan mikrokontroler berbasis ESP32 dengan menerapkan topologi *mesh*, guna memantau kondisi lingkungan fisik secara aktual. Pendekatan rancang bangun *hardware* serta kerangka kerja penataan *node* sensor pada penelitian tersebut diadopsi sebagai basis dasar penentuan tata letak sistem pada penelitian ini. Relevansi metodologi pengujian keterhubungan antar-*node* sensor dari penelitian tersebut memberikan panduan teknis yang signifikan dalam memitigasi kendala transmisi data berskala lokal di lingkungan institusi pengusul.

Acuan metodologis dalam penelitian ini mengacu pada karya ilmiah oleh Wedyanti dan Wagya [7]. Penelitian tersebut menguji dan menganalisis performa protokol ESP-NOW sebagai jalur komunikasi data berlatensi rendah yang efisien untuk menghubungkan arsitektur *multi-node* dengan *gateway* pusat pada ekosistem IoT. Bagian dari arsitektur *multi-node* dalam publikasi tersebut akan dimodifikasi pada penelitian ini melalui optimasi algoritma pada lima kali pengiriman data periodik untuk meningkatkan efisiensi daya serap energi pada *node* sensor.

Selain itu, penelitian oleh Ramadhan [8] menyajikan implementasi jaringan *mesh* dengan protokol *routing* OLSRv2 pada perangkat berorientasi *resource-constrained*. Mekanisme manajemen perutean dinamis dari tesis tersebut disempurnakan dan disesuaikan dalam penelitian ini agar dapat diaplikasikan secara optimal pada klaster perangkat berspesifikasi *hardware* yang terbatas.

Penelitian terkait penerapan WSN dengan topologi *star* pun, oleh Candra dan Busran [9], untuk sistem deteksi dini banjir, terdiri atas tiga *node* pengirim dengan sensor suhu, ketinggian udara, curah hujan, dan kelembapan, serta satu *node* penerima, yang saling tersusun dalam topologi *star*. Penelitian tersebut menghasilkan pengoperasian sistem yang stabil pada jarak 110 meter dengan nisbah kehilangan data di bawah 2%. Antarmuka web pada sistem tersebut memungkinkan pantauan parameter lingkungan dan data historis dari tiap *node* secara *remote*. Integrasi multi-sensor tersebut mendukung pemantauan kondisi banjir pada sejumlah titik.

Lebih lanjut, Ghazali dkk. [10] melakukan simulasi sistem *Wireless Sensor Network* (WSN) berbasis *Internet of Things* (IoT) untuk pemantauan kualitas air dengan menggunakan lima topologi jaringan, yaitu *Hybrid Star-Mesh*, *Cluster Tree*, *Full Mesh*, *Ring*, dan *ZigBee Star*, yang masing-masing melibatkan 20 *node*. Hasil penelitian menunjukkan bahwa topologi *Hybrid Star-Mesh* meminimalkan latensi dan kehilangan paket sekaligus memaksimalkan *throughput*; topologi *Ring* secara konsisten menunjukkan kinerja terendah; *Cluster Tree* dan *ZigBee Star* berada di tingkat menengah; *Full Mesh* mengorbankan redundansi demi latensi dan *bandwidth*.

Penelitian oleh Urazayev dkk. (2023) [4] mengevaluasi kinerja protokol ESP-NOW pada skenario dalam ruangan melalui pengujian eksperimental terhadap 15 *node* penerima (ESP32 WROOM) pada kondisi *line-of-sight* (LOS), non-LOS, dan posisi jauh, dengan mengukur RSS dan PSR, dibandingkan WiFi TCP (802.11n). Hasil menunjukkan PSR 100% pada LOS (5–55 m), turun ke 93,8–99,9% pada non-LOS, serta jangkauan maksimum mode *Long-Range* mencapai 280–330 meter pada target PSR 90%. ESP-NOW mengonsumsi daya 600–660 mW, lebih hemat energi >3% dibanding WiFi TCP, dengan PDR *indoor* 15% lebih baik. Relevansi bagi penelitian ini adalah bukti kuantitatif kelayakan ESP-NOW sebagai protokol komunikasi utama antar-*node* karena efisiensi daya dan jangkauannya tanpa bergantung pada infrastruktur *router* WiFi.

Penelitian oleh Fu dkk. (2023) [11] memetakan arsitektur, tantangan, dan arah pengembangan JSN dalam ekosistem IoT melalui tinjauan literatur. Relevansi bagi penelitian ini adalah landasan taksonomi konseptual untuk merumuskan arsitektur sistem yang mempertimbangkan efisiensi daya dan skalabilitas *node*.

Penelitian oleh Trigka dan Dritsas (2025) [12] memetakan perkembangan JSN secara menyeluruh—arsitektur (*single-tier*, *multi-tier*, *duty-cycled*, dll.),

protokol MAC/*routing*, hingga *energy harvesting* dan integrasi AI—melalui tinjauan sistematis di IEEE Access. Hasil kajian menegaskan arsitektur *duty-cycled* dan *energy harvesting* multi-sumber sebagai pendekatan paling menjanjikan untuk memperpanjang usia jaringan berdaya terbatas. Relevansi bagi penelitian ini adalah pemutakhiran kerangka teoretis untuk validasi rancangan arsitektur jaringan hemat daya.

Penelitian oleh Irmayani dkk. (2025) [13] merancang purwarupa komunikasi teks nirkabel bertopologi *mesh* berbasis ESP32 untuk memperluas cakupan sinyal di daerah pelosok, diuji dengan protokol ESP-NOW dan skema *routing* bertingkat (SP1–SP3) pada kondisi LOS menggunakan empat modul ESP32. Hasil menunjukkan mode *point-to-point* tanpa relai menjangkau hingga 545 meter (durasi kirim stabil 3 ms di bawah 455 m, memburuk menjadi 8–25 ms hingga gagal total setelah 547 m), sedangkan penambahan dua *node* relai memperluas jangkauan hingga 1270 meter dengan keberhasilan pengiriman 100%. Kapasitas *payload* maksimum tercatat 244 byte. Relevansi bagi penelitian ini adalah bukti kuantitatif bahwa relai bertingkat mampu menambah cakupan area dibanding *point-to-point* murni, menjadi acuan jumlah dan posisi *node* relai dalam arsitektur yang dikembangkan.

Penelitian oleh Santos dkk. (2022) [2] mengevaluasi performa jaringan *mesh* berbasis ESP8266 (*library* *painlessMesh*) untuk memantau barang mudah rusak di kontainer berpendingin, menguji *delivery ratio* dan *one-way delay* pada variasi jumlah *node* (1–5), *traffic load* (1–10 pesan/detik), *payload* (10–2000 byte), dan dua konfigurasi MSS. Penambahan *node* kedua-ketiga menurunkan performa lebih signifikan dibanding *node* keempat-kelima, dan *delay* menunjukkan volatilitas tinggi (10–150 detik) yang meningkat seiring beban jaringan. Relevansi bagi penelitian ini adalah bukti kuantitatif adanya ambang optimal jumlah *node* dan ukuran *payload* terhadap *delivery ratio* dan latensi pada topologi *mesh*.

Penelitian oleh Afifie dkk. [3] berfokus pada metode evaluasi komparatif implementasi protokol *mesh* standar di atas platform komputasi ESP32 dan ESP8266 untuk melihat perbandingan stabilitas pengiriman paket. Dokumen teknis Espressif Systems [5] memuat manual referensi teknis arsitektur *hardware* ESP32, yang dipakai untuk mengonfigurasi interupsi *register* dan manajemen daya rendah secara presisi pada mikrokomputer.

Studi oleh Mardamsyah dkk. [14] membahas penerapan komunikasi protokol ESP-NOW berbasis topologi *mesh* demi mewujudkan aspek interoperabilitas data tinggi dalam studi konsep *Network Centric Warfare*. Penelitian oleh Fitriani dkk. [15] menganalisis komparasi performa antara karakteristik jaringan topologi *star* dan topologi *mesh* pada implementasi jaringan lokal, memberikan pemahaman perilaku transmisi data statis.

Dokumen rilis resmi dari Espressif Systems [1] menyediakan panduan teknis repositori perangkat lunak *master release* sebagai referensi fungsi pemrograman pustaka komponen ekosistem mikrokontroler Espressif terbaru. Penelitian oleh Arofah dkk. [16] memaparkan

implementasi protokol ESP-NOW pada sistem *portable traffic light*, yang membuktikan kelayakan protokol tersebut dalam skenario komunikasi nirkabel yang membutuhkan respons *real-time* pada perangkat mikrokontroler berdaya rendah. Penelitian oleh Prasetya dkk. [17] memperkenalkan mekanisme sinkronisasi waktu pada jaringan sensor nirkabel berbasis ESP-NOW menggunakan *Flooding Time Synchronization Protocol*, yang memberikan landasan bagi penelitian ini dalam mempertimbangkan aspek konsistensi temporal pengiriman data periodik antar-*node* sensor.

Berdasarkan tinjauan pustaka-pustaka tersebut, penelitian terdahulu dapat diklasifikasikan ke dalam tiga fokus utama, yaitu:

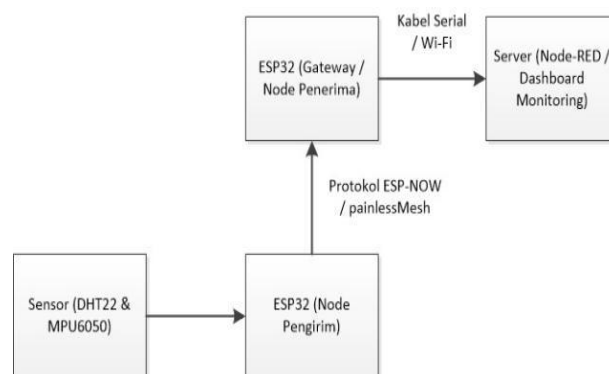
- Fokus 1 (Kinerja ESP-NOW/Star): Penelitian yang mengeksplorasi efisiensi latensi dan jangkauan ESP-NOW secara spesifik, seperti pada implementasi protokol antar *multi-node* dan *gateway* [18], maupun evaluasi kinerja di dalam ruangan (*indoor*) [17].
- Fokus 2 (Implementasi Topologi Mesh): Penelitian yang mengandalkan topologi *Mesh* untuk perluasan area cakupan pemantauan [2], komunikasi data berbasis teks [10], serta penilaian performa jaringan *wireless mesh* secara umum [15].
- Fokus 3 (Perbandingan Topologi WSN): Evaluasi komparatif berbagai topologi WSN, seperti perbandingan kinerja topologi *star* dan *mesh*, baik pada lingkungan perkantoran [7] maupun sistem pemantauan berbasis IoT [9].

Posisi penelitian ini secara eksplisit menjembatani ketiga kelompok tersebut dengan menyajikan komparasi empiris (implementasi *hardware*) antara topologi *star* (berbasis protokol ESP-NOW) dan topologi *mesh* (berbasis *painlessMesh*). Perbedaan mendasar, sekaligus kontribusi utama penelitian ini, dibandingkan literatur sebelumnya, adalah integrasi evaluasi metrik kinerja (PDR, latensi, dan RSSI) terhadap variasi beban interval transmisi (1 hingga 60 detik) pada kondisi propagasi secara nyata di lapangan (LOS dan NLOS), yang secara khusus pada jaringan *mesh*, dipadukan dengan skenario kegagalan titik (*node failure*) untuk menguji dan membuktikan batas ambang toleransi kemampuan pemulihan jaringan secara mandiri (*self-healing*).

III. METODOLOGI

Sistem yang dirancang merupakan *Wireless Sensor Network* (WSN) berbasis ESP32 yang terdiri atas 5 *node* sensor, 1 *gateway* ESP32, dan 1 peladen atau *server* (laptop/PC dengan Node-RED). Node-RED merupakan platform pemrograman berbasis alur (*flow-based programming*) untuk pengembangan aplikasi *Internet of Things* (IoT) secara cepat dan mudah [18][19]. Platform ini menyediakan antarmuka visual berbasis web yang memungkinkan pengguna membina alur pemrosesan data menggunakan *node-node* yang saling terhubung. Pada penelitian ini, Node-RED digunakan sebagai media pemantauan dan visualisasi data hasil pengukuran yang dikirimkan dari *gateway* ESP32, sehingga pengguna bisa

memantau kondisi sistem secara *real-time* melalui *dashboard* berbasis web.



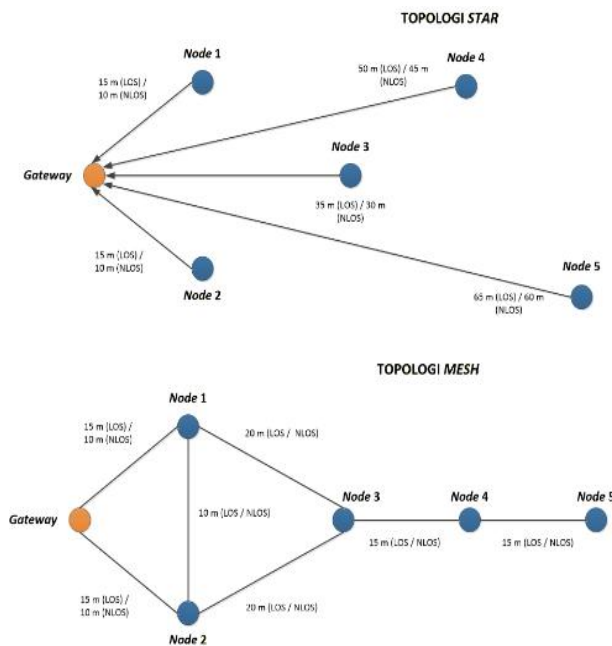
Gbr. 1 Diagram Blok Sistem WSN

Arsitektur sistem terdiri atas *node* sensor yang berfungsi mengakuisisi data lingkungan, *gateway* ESP32 sebagai penerima data, serta *server* berbasis Node-RED untuk penyimpanan dan visualisasi data. Gambaran hubungan antarkomponen tersebut direpresentasikan pada Gambar 1.

Data sensor DHT22 dan MPU6050 dibaca oleh ESP32 sebagai *node* pengirim, kemudian dikirimkan ke *gateway* ESP32 melalui protokol ESP-NOW atau *painlessMesh* sesuai topologi yang digunakan. Selanjutnya, *gateway* meneruskan data ke *server* melalui koneksi serial atau Wi-Fi untuk diproses, disimpan, dan ditampilkan pada *dashboard* Node-RED sebagai media monitoring secara *real-time*, sehingga kondisi lingkungan dan pergerakan objek dapat dipantau dari jarak jauh guna mendukung penentuan keputusan cepat.

Pada perangkat ESP32 dan ESP8266, implementasi jaringan *mesh* didukung oleh pustaka perangkat lunak *painlessMesh*. *Library* ini menyediakan mekanisme pembentukan *mesh* secara otomatis tanpa konfigurasi *routing* kompleks. *painlessMesh* memiliki fitur *self-organizing* dan *self-healing* sebagai alat bantu *node* baru untuk otomatis bergabung ke jaringan, serta membentuk ulang jalur komunikasi apabila terjadi kegagalan pada salah satu *node*. Dengan fitur tersebut, jaringan dapat tetap beroperasi, meskipun terdapat perubahan struktur jaringan selama proses komunikasi berlangsung. Pada penerapannya, setiap *node* dapat berfungsi sebagai pengirim, penerima, sekaligus penerus paket data. Mekanisme ini memungkinkan terbentuknya komunikasi *multi-hop*, sehingga jangkauan jaringan dapat diperluas tanpa peningkatan daya pancar perangkat. Pada penelitian ini, *painlessMesh* digunakan sebagai platform komunikasi utama untuk membangun topologi *mesh multi-hop* berbasis ESP32.

Selanjutnya, pemetaan riil dari kedua arsitektur hubungan *node* tersebut ditunjukkan secara terperinci pada Gambar 2.



Gbr. 2 Pemetaan Posisi dan Kondisi LOS/NLOS Kedua Topologi

A. Metodologi Pengujian

Pengujian sistem dilakukan dengan mengukur tiga parameter utama, yaitu *Packet Delivery Ratio* (PDR), latensi (*end-to-end delay*), dan *Received Signal Strength Indicator* (RSSI), pada dua kondisi lingkungan: *Line of Sight* (LOS) tanpa penghalang dan *Non-Line of Sight* (NLOS) dengan penghalang.

Pengujian kondisi *Non-Line of Sight* (NLOS) dilakukan di lingkungan luar ruangan (*outdoor*), tepatnya di area halaman rumah dalam kompleks perumahan, untuk merepresentasikan kondisi propagasi sinyal dengan hambatan natural. Penghalang fisik utama yang memotong jalur komunikasi antar-node berupa pagar tanaman (vegetasi) berdaun rapat, sebagaimana yang diterapkan pada perumahan secara umum. Karena pengujian dilakukan pada skenario dunia nyata (*real-world scenario*), dimensi pasti dari vegetasi bervariasi. Namun, secara umum pagar tanaman ini diestimasikan memiliki ketebalan sekitar 30 hingga 50 cm, dan cukup padat untuk memblokir garis pandang visual sepenuhnya (*visual Line of Sight*) antarperangkat. Penggunaan vegetasi sebagai penghalang fisik ini bertujuan untuk menguji redaman sinyal (*signal attenuation*) akibat penyerapan gelombang radio oleh dedaunan, sehingga merepresentasikan tantangan nyata pada implementasi jaringan di area hijau.

Setiap *node* menerima pengiriman dengan variasi interval pengiriman (1, 5, 10, 30, dan 60 detik) pada rentang jarak 15–65 meter untuk LOS dan 10–60 meter untuk NLOS. Pengujian dilakukan pada dua konfigurasi topologi, yaitu *star* (komunikasi langsung) dan *mesh* (*multi-hop*), serta dilengkapi pengujian skenario penonaktifan *node* relai untuk mengevaluasi kemampuan *self-healing* jaringan *mesh*. Hasil dari seluruh skenario tersebut kemudian dianalisis berdasarkan tiga metrik pengujian yang padu, yaitu PDR (%), latensi (ms), dan

RSSI, guna menilai keandalan dan performa sistem pada berbagai kondisi dalam aspek jarak, topologi, dan gangguan jaringan.

B. Implementasi Hardware

Tabulasi spesifikasi *hardware* untuk sistem WSN disajikan pada Tabel I. Tahap ini merupakan proses realisasi fisik dari rancangan sistem yang telah dibuat, meliputi penyolderan komponen, pengujian kontinuitas untuk mencegah korsleting, serta integrasi modul sensor dengan mikrokontroler menjadi satu sistem. Pada tahap ini juga dilakukan pengujian fungsi setiap komponen dan penataan letaknya, guna meminimalkan interferensi, serta memastikan sistem siap dioperasikan. Hasil implementasi ini menjadi dasar keberhasilan uji sistem pada tahap lanjut. Maka, seluruh rangkaian perangkat keras harus dipastikan terpasang kuat dan beroperasi sesuai parameter yang ditentukan. Kegagalan pada tahap ini akan berdampak pada keakuratan data serta respons keseluruhan sistem.

TABEL I
Spesifikasi *Hardware* Sistem WSN

Nama	Spesifikasi	Jumlah
Mikrokontroler ESP32 DevKit	Flash 4 MB, SRAM 520 kB, Processor dual-core 32-bit LX6 240 MHz	6
Sensor DHT22	Tegangan kerja 3,3–5 V, akurasi suhu $\pm 0,5^\circ \text{C}$, kelembaban $\pm 2\%$	5
Sensor MPU6050	Gyroscope + Accelerometer 3-axis, tegangan 3,3–5 V	5
Breadboard + Jumper	Media <i>prototyping</i> rangkaian	6 set
Project Box	Enclosure pelindung perangkat	6
LED Indikator + Resistor	Tegangan 2–3 V sebagai indikator status	6 set
Rocker Switch 2 Pin	Sakelar ON/OFF manual, tipe SPST	5
Modul Step Up DC Boost Converter	Input 1–5 V, output 5 V	5
Lithium-ion Battery 18650 3,7 V	Tegangan nominal 3,7 V, tipe untuk isi ulang	5
PCB Matrix 7 × 9 cm	Media perakitan sirkuit permanen	6

a. Implementasi Node Sensor

Node sensor, sebagaimana yang ditunjukkan pada Gambar 3, terdiri atas mikrokontroler ESP32 sebagai unit pemrosesan utama yang terhubung dengan sensor DHT22 untuk pengukuran suhu dan kelembapan, serta sensor MPU6050 untuk mendeteksi percepatan dan orientasi gerakan. Sumber daya perangkat berasal dari

baterai *lithium-ion* 18650 yang dihubungkan melalui modul *step-up converter* untuk menjaga kestabilan tegangan. Seluruh rangkaian dipasang pada *PCB matrix* dan ditempatkan dalam *project box*. Selain itu, indikator LED dan resistor digunakan sebagai penanda status operasi perangkat serta proses komunikasi data selama pengujian berlangsung.



Gbr. 3 Implementasi *Hardware: Node Sensor* (a) Tampak Dalam; (b) Tampak Luar

b. Implementasi *Gateway*

Gateway berbasis ESP32 berfungsi menerima data dari seluruh *node* sensor dan meneruskannya ke laptop melalui koneksi serial USB untuk pemantauan dan pengolahan data. *Gateway* dilengkapi LED indikator sebagai penanda status sistem, sedangkan koneksi USB juga berfungsi sebagai jalur komunikasi serial dan catu daya. Gambar 4 merupakan dokumentasi tampak dalam dari *box* yang digunakan untuk *gateway*.



Gbr. 4 Tampak Dalam *Box* untuk *Gateway*

C. Implementasi *Software*

Visualisasi hasil pengukuran sensor merupakan capaian utama tahap implementasi *software*. Tahap ini merupakan penentuan kesinambungan kerja perangkat-perangkat keras (*hardware*) dan unsur visual yang dibutuhkan, demi keterbacaan oleh pengguna secara umum.

a. *Firmware Node Sensor*

Firmware node sensor diinstal pada mikrokontroler ESP32 untuk mengendalikan seluruh proses operasional *node* sensor. *Firmware* ini bertugas melakukan inisialisasi perangkat, membaca data dari sensor DHT22 dan MPU6050, melakukan pengolahan data awal, serta mengemas data ke dalam format JSON. *Payload* JSON juga dilengkapi nomor urut paket (*sequence number/seq*) yang digunakan *gateway* untuk mendeteksi rugi-rugi pengiriman paket atau *packet loss* dan menghitung nilai *Packet Delivery Ratio* (PDR) secara *real-time*.

b. *Firmware Gateway*

Firmware gateway berfungsi menerima data dari *node* sensor, melakukan pemrosesan data yang diterima, kemudian meneruskannya ke *server* untuk visualisasi dan penyimpanan data.

```
===== TOPOLOGI MESH ==
>>> DATA MASUK DARI NODE [ 1 ] <<<
Sequence No : 2
-----
Suhu Ruangan : 28.40 °C
Kelembaban : 70.00 %
Sudut Roll : -1.00 °
Sudut Pitch : -2.11 °
-----
RSSI Sinyal : -52 dBm
Latency Link : 2.80 ms
PDR (QoS) : 50.00 %
=====
>>> DATA MASUK DARI NODE [ 3 ] <<<
Sequence No : 2

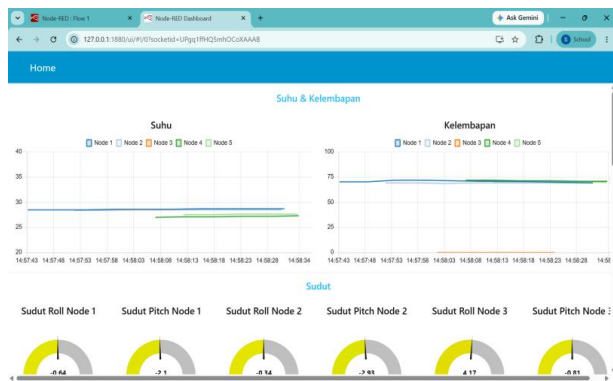
===== TOPOLOGI MESH =====
>>> DATA MASUK DARI NODE [ 2 ] <<<
Sequence No : 2
-----
Suhu Ruangan : 28.30 °C
Kelembaban : 69.10 %
Sudut Roll : -0.22 °
Sudut Pitch : -3.15 °
-----
RSSI Sinyal : -37 dBm
Latency Link : 5.85 ms
PDR (QoS) : 50.00 %
=====
>>> DATA MASUK DARI NODE [ 5 ] <<<
Sequence No : 2
```

Gbr. 5 Monitor Serial *Gateway* (Penerimaan Parameter *seq*)

Gambar 5 menunjukkan pengujian aliran data mentah yang masuk pada *Serial Monitor* di sisi *gateway*. Pada tahap ini, terlihat bahwa setiap paket data yang dikirimkan oleh *node* Sensor membawa parameter *seq* (*Sequence Number*). Parameter *seq* ini diproses secara *real-time* dan berfungsi sebagai identitas nomor urut paket yang krusial untuk melacak total jumlah paket yang dikirimkan (Tx) oleh *Node*. Data *seq* inilah yang digunakan dalam perhitungan program untuk mendapatkan persentase *Packet Delivery Ratio* (PDR) sebelum akhirnya sisa data lainnya diteruskan dan disimpan ke dalam *database* CSV *Node-RED*.

c. Penerapan *Dashboard* *Node-RED*

Gambar 6 menunjukkan jepretan layar *dashboard* *Node-RED* berbasis peramban *web* atau *web browser* yang digunakan untuk memantau sistem WSN secara *real-time*. *Dashboard* menampilkan data sensor DHT22 dan MPU6050, serta parameter QoS berupa PDR, latensi, dan RSSI. Selain itu, tersedia tampilan monitor serial untuk memantau komunikasi antarperangkat. Seluruh data yang diterima disimpan otomatis ke berkas CSV untuk keperluan pencatatan dan analisis lanjut.



Gbr. 6 Tampilan Dashboard Node-RED

D. Implementasi Komunikasi

Inti dari sistem WSN adalah protokol komunikasi itu sendiri.

a. Protokol ESP-NOW

Pada mode *star*, komunikasi data dilakukan secara langsung antara *node* sensor dan *gateway* menggunakan protokol ESP-NOW. Setiap *node* mengirimkan data tanpa melalui *node* perantara, sehingga menghasilkan proses komunikasi sederhana dan latensi rendah. Pun, nilai *MAC address gateway* pada setiap *node* dikonfigurasi, agar setiap *node* memiliki arah pengiriman paket.

b. Protokol *painlessMesh*

Pada mode *mesh*, komunikasi data dibangun menggunakan library *painlessMesh*. Data dapat diteruskan melalui *node* lain (*multi-hop*), tanpa konfigurasi *MAC address gateway* secara manual, serta bekerja secara *broadcast* sebelum mencapai *gateway*, sehingga memungkinkan perluasan jarak komunikasi pada kawasan pengujian.

IV. HASIL DAN PEMBAHASAN

Pengujian dilakukan untuk mengetahui performa sistem berdasarkan parameter *Packet Delivery Ratio* (PDR), *delay* komunikasi, serta stabilitas jaringan pada berbagai kondisi pengujian. Selain itu, dilakukan juga pengujian fungsionalitas untuk memastikan seluruh komponen perangkat keras dan perangkat lunak bekerja sesuai dengan perancangan sistem.

A. Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan untuk memastikan seluruh bagian sistem dapat bekerja sesuai dengan fungsi yang telah dirancang. Pengujian meliputi *node* sensor, komunikasi jaringan, *gateway*, serta *dashboard* monitoring Node-RED.

a. Pengujian *Node* Sensor

Berdasarkan hasil pengujian pada Tabel II, seluruh fungsi utama *node* sensor berjalan sesuai dengan perancangan sistem. Sensor DHT22 dan MPU6050

mampu membaca data lingkungan dan orientasi secara stabil. Selain itu, komunikasi menggunakan ESP-NOW maupun *mesh multi-hop* berhasil dilakukan, sehingga *node* sensor dapat mengirimkan data menuju *gateway*. Keberhasilan fungsionalitas ini memastikan bahwa seluruh infrastruktur *hardware* pada *node* sensor siap digunakan dalam pengujian performa lanjutan jaringan.

TABEL II
Hasil Uji Fungsionalitas *Node* Sensor

Fitur	Fungsionalitas	Realisasi
Sensor DHT22	Membaca data suhu dan kelembaban	Data suhu dan kelembaban berhasil terbaca
Sensor MPU6050	Membaca nilai <i>roll</i> dan <i>pitch</i>	Nilai <i>roll</i> dan <i>pitch</i> berhasil terbaca
ESP-NOW	Mengirim data langsung ke <i>gateway</i>	Data berhasil diterima <i>gateway</i>
<i>Mesh Multi-Hop</i>	Mengirim data melalui <i>node</i> relai	Data berhasil diteruskan menuju <i>gateway</i>
Interval pengiriman	Mengirim data sesuai interval	Pengiriman stabil

b. Pengujian *Gateway*

Berdasarkan Tabel III, *gateway* berhasil menerima paket data dari seluruh *node* sensor, baik pada topologi *star* maupun *mesh*, serta mampu *parsing* data serta meneruskan data menuju *dashboard* monitoring secara *real-time*. Fungsionalitas sistem komunikasi dan integrasi *hardware* maupun *software* telah berjalan sesuai dengan rancangan. Keberhasilan ini mencakup kemampuan penerimaan data ESP-NOW dari *node* sensor dan penerimaan data *Mesh* dari jaringan *mesh*, di mana seluruh paket diterima dengan baik. Selain itu, *gateway* secara akurat mampu melakukan *parsing payload* untuk memisahkan isi data sensor, menambahkan *timestamp* pada setiap waktu penerimaan paket, hingga mengeksekusi proses pengiriman data ke Node-RED, sehingga visualisasi pada *dashboard* monitoring dapat ditampilkan secara *real-time*.

TABEL III
Hasil Uji Fungsionalitas *Gateway*

Fitur	Fungsionalitas	Realisasi
Penerimaan data ESP-NOW	Menerima paket dari <i>node</i> sensor	Paket berhasil diterima
Penerimaan data <i>Mesh</i>	Menerima data dari jaringan <i>mesh</i>	Paket berhasil diterima
<i>Parsing payload</i>	Memisahkan isi data sensor	Data berhasil di- <i>parsing</i>
<i>Timestamp data</i>	Menambahkan info waktu penerimaan paket	Timestamp berhasil ditambahkan
Pengiriman ke Node-RED	Mengirim data monitoring ke <i>dashboard</i>	Data berhasil tampil real-time

c. Uji Dashboard Monitoring

Dashboard monitoring, sebagaimana ditunjukkan pada hasil pengujian yang disajikan dalam Tabel IV, berhasil menampilkan data sensor secara *real-time* dan stabil selama proses pengujian berlangsung. Sistem juga mampu menyimpan data hasil pengujian untuk keperluan analisis performa jaringan. Visualisasi data yang disajikan melalui antarmuka grafik memudahkan pengguna dalam memantau fluktuasi parameter lingkungan dari waktu ke waktu secara intuitif. Keandalan dalam pembaruan status data secara *real-time* ini membuktikan, bahwa jalur komunikasi data dari perangkat *node* menuju *server* melalui protokol nirkabel telah berjalan dengan efisiensi tinggi tanpa adanya kendala keterlambatan pengiriman data (latensi) yang berarti. Berdasarkan keseluruhan hasil pengujian fungsionalitas ini, dapat disimpulkan bahwa *dashboard* telah memenuhi spesifikasi kebutuhan sistem monitoring secara *real-time* dan siap digunakan untuk mendukung operasional pemantauan jaringan sensor secara berlanjut.

TABEL IV
Hasil Uji Fungsionalitas *Dashboard* Monitoring

Fitur	Fungsionalitas	Realisasi
Monitoring <i>real-time</i>	Menampilkan data sensor <i>real-time</i>	Data berhasil ditampilkan <i>real-time</i>
Grafik monitoring	Menampilkan grafik suhu dan kelembaban	Grafik berhasil ditampilkan
Penyimpanan data	Menyimpan hasil uji ke <i>database</i> /log	Data berhasil tersimpan
Status <i>node</i>	Menampilkan status <i>node</i> aktif	Status <i>node</i> berhasil ditampilkan

B. Pengujian Packet Delivery Ratio (PDR)

Pada pengujian *Packet Delivery Ratio* (PDR), digunakan salah satu sampel interval pengiriman data, yaitu 5 detik, dengan jumlah penerimaan data sebanyak 30 kali. Sampel tersebut digunakan sebagai representasi untuk menganalisis kinerja sistem, di mana sampel tersebut berfungsi sebagai representasi untuk menganalisis kinerja sistem dalam mempertahankan keandalan komunikasi nirkabel antar-*node*.

```

if (id > 0 && id <= 5) {
    stats[id].totalReceived++;

    if (incomingData.seq > stats[id].expectedSeq) {
        stats[id].lostPackets += (incomingData.seq - stats[id].expectedSeq);
    }

    if (incomingData.seq < stats[id].expectedSeq - 10) {
        stats[id].lostPackets = 0;
        stats[id].totalReceived = 1;
    }

    stats[id].expectedSeq = incomingData.seq + 1;
    stats[id].pdr = ((float)stats[id].totalReceived / (float)(stats[id].totalReceived + stats[id].lostPackets)) * 100.0;
}

```

(a)

```

if (seq != stats[id].lastSeq) {
    stats[id].totalReceived++;
    if (stats[id].lastSeq > 0 && seq > (stats[id].lastSeq + 1)) {
        stats[id].lostPackets += (seq - stats[id].lastSeq - 1);
    }
    stats[id].lastSeq = seq;
}

```

(b)

Gbr. 7 Potongan Kode Logika Perhitungan PDR:

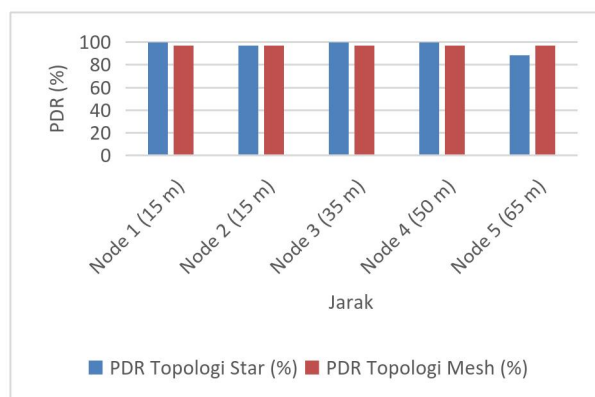
(a) pada Topologi *Star*; (b) pada *Mesh*

Sistem WSN, yang disusun berdasarkan pengodean seperti Gambar 7, menghitung PDR menggunakan logika pembagian dinamis. Pada topologi *star*, PDR dihitung dengan membagi *totalReceived* dengan hasil jumlah *totalReceived* dan *lostPackets*. Sedangkan pada topologi *mesh*, *totalReceived* dibandingkan langsung dengan nilai *seq* (nomor urut terakhir) yang berhasil masuk. Melalui implementasi logika pengodean, nilai PDR pada tabel-tabel hasil pengujian tetap dapat disajikan secara presisi, meskipun data log pada berkas CSV hanya merekam baris paket yang sukses diterima saja.

a. Pengujian PDR Kondisi LOS

Berdasarkan hasil pengujian pada kondisi tanpa halangan (LOS) yang disajikan pada Gambar 8, kedua topologi menunjukkan performa pengiriman data yang sangat baik, namun dengan karakteristik yang berbeda. Topologi *star*, berdasarkan sajian data tersebut, unggul pada pengiriman jarak dekat hingga menengah (15 sampai 50 meter) dengan tingkat keberhasilan pengiriman (PDR) mencapai angka sempurna (100%). Namun, saat jarak diperpanjang menjadi 65 meter (*node* 5), performa topologi *star* mulai mengalami penurunan, yakni nilai PDR menjadi 88,24%.

Di sisi lain, topologi *mesh* menunjukkan tingkat kestabilan yang sangat tinggi. Meskipun tidak menyentuh angka 100%, topologi *mesh* mampu mempertahankan nilai PDR secara konstan, di angka 96,77%, pada seluruh titik pengujian, mulai dari jarak 15 hingga 65 meter.

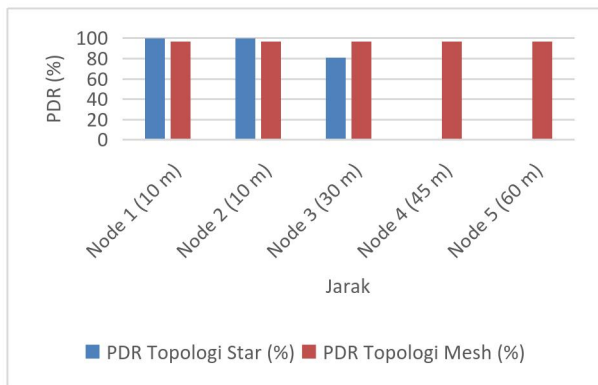


Gbr. 8 Grafik PDR *Star* vs *Mesh* (LOS)

b. Pengujian PDR Kondisi NLOS

Hasil pengujian pada kondisi dengan halangan (NLOS) yang divisualisasikan pada Gambar 9. Uji pada kondisi non-LOS dilakukan untuk melihat dampak redaman sinyal akibat halangan fisik (*obstacle*) semisal tembok terhadap keberhasilan pengiriman paket data. Pada pengujian ini, topologi *star* menunjukkan kelemahan yang fatal. Meskipun masih mampu mengirimkan data dengan PDR 100% pada jarak 10 meter, performanya turun drastis menjadi 81,08% pada jarak 30 meter. Puncaknya, pada jarak 45 meter (*Node* 4) dan 60 meter (*Node* 5), topologi *star* mengalami kegagalan komunikasi total (*link failure*) yang ditandai dengan nilai PDR sebesar 0 %. Hal ini terjadi karena

sinyal dari *node* jarak jauh tidak mampu menembus hambatan fisik untuk mencapai *gateway* secara langsung.



Gbr. 9 Grafik PDR Star vs Mesh (NLOS)

Sebaliknya, topologi *mesh* mengungguli *star* dalam menangani hambatan fisik. Meski terhalang oleh *obstacle* dan berada pada jarak 60 meter, nilai PDR topologi *mesh* tetap stabil di angka 96,77%. Kestabilan ini tercapai berkat mekanisme *multi-hop* pada jaringan *mesh*; *node* 4 dan *node* 5 tidak perlu memaksakan diri mengirim data langsung ke *gateway*, tetapi meneruskan paket datanya melalui *node* terdekat.

Dilakukan audit terhadap data mentah pengujian (jumlah paket terkirim atau *Tx*, diterima atau *Rx*, dan yang hilang atau *Lost* per *node*) pada sampel representatif interval 5 detik. Hasil audit menunjukkan bahwa pada topologi *mesh*, hampir seluruh *node*, baik kondisi LOS maupun NLOS, secara konsisten mencatat *Tx* = 31 paket, *Rx* = 30 paket, dan *Lost* = 1 paket (PDR = $30/31 = 96,77\%$). Nilai ini bersifat konstan pada hampir seluruh skenario pengujian *mesh* (termasuk interval 10, 30, dan 60 detik, serta skenario kematian *node*), sehingga diduga kuat merupakan artefak formula perhitungan PDR berbasis nomor urut (*seq*), yaitu paket pertama pada penomoran *seq* tidak tercatat sebagai *totalReceived* namun tetap menambah nilai *seq* acuan, bukan murni cerminan kehilangan paket akibat kondisi kanal. Penyimpangan dari nilai 96,77% pada beberapa titik (misalnya 81,08% di *node* 2 skenario *node* 1 mati, dan 71,43% di *node* 5 interval 30 detik) mengindikasikan kehilangan paket tambahan yang riil, terutama pada kondisi beban jaringan tinggi atau redundansi jalur yang berkurang akibat kegagalan *node*.

Sebaliknya, audit pada topologi *star* menunjukkan nilai *Tx* dan *Lost* yang bervariasi konsisten dengan kondisi jarak dan hambatan, misalnya 0 paket hilang dari 30 paket terkirim pada *node* 1-4 kondisi LOS, namun meningkat menjadi 7 paket hilang dari 37 paket terkirim (PDR = 81,08%) pada *node* 3 kondisi NLOS. Perbedaan mekanisme pencatatan ini—*loss counter* eksplisit pada *star* versus rasio *seq* pada *mesh*—menegaskan bahwa nilai PDR kedua topologi tidak sepenuhnya dapat dibandingkan secara langsung tanpa mempertimbangkan basis perhitungannya masing-masing. *Firmware* direkomendasikan menambahkan pencacah *Tx* eksplisit pada *node mesh* untuk audit PDR yang lebih presisi pada penelitian lanjutan.

C. Pengujian Latensi

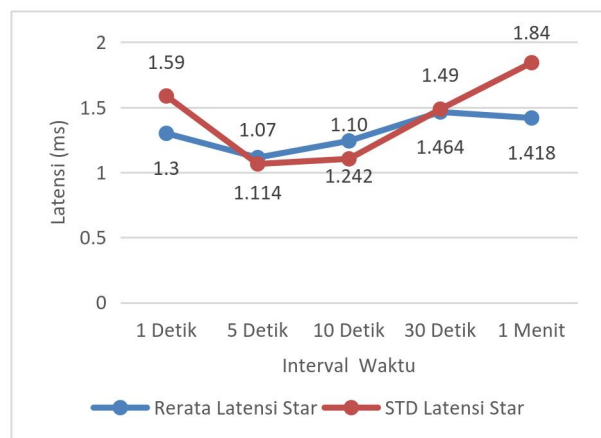
Nilai yang dianalisis pada pengujian waktu tunggu atau latensi diperoleh dari rata-rata hasil pengukuran pada seluruh variasi interval pengiriman data, yaitu 1 detik, 5 detik, 10 detik, 30 detik, dan 1 menit. Waktu tempuh data (latensi) direkam secara langsung pada setiap paket yang berhasil diterima. Latensi dihitung sistem menggunakan selisih, antara stempel waktu (*timestamp*) saat paket dikirim oleh *node* sensor (t_{send}), dengan waktu saat paket tersebut tiba di *gateway* (t_{receive}), melalui persamaan matematis:

$$\text{Latency} = t_{\text{receive}} - t_{\text{send}} \quad (1)$$

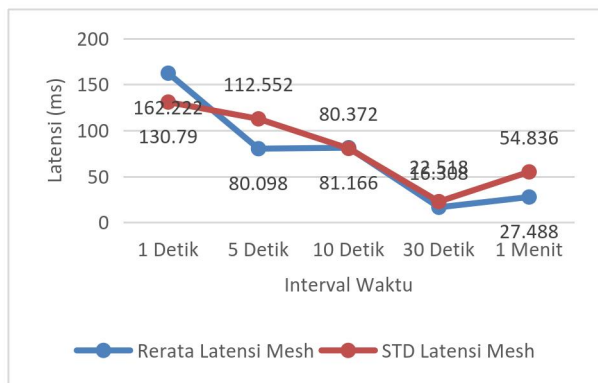
Hasil selisih tersebut dikonversi ke satuan milidetik (ms). Pengambilan sampel dilakukan hingga 30 kali penerimaan data yang berhasil pada setiap variasi interval untuk memperoleh nilai rerata (*mean*), serta meminimalkan pengaruh bias atau anomali selama proses transmisi paket data. Namun, pada pengujian topologi *Star* kondisi *Non-Line-of-Sight* (NLOS), beberapa *node* tidak mampu mengirimkan hingga 30 paket berhasil diterima, bahkan terdapat *node* yang mengalami kegagalan komunikasi. Maka, perhitungan rata-rata dilakukan berdasarkan jumlah paket yang berhasil diterima pada masing-masing *node*.

a. Pengujian Latensi Kondisi LOS

Data grafik pada Gambar 10 dan Gambar 11 menunjukkan perbandingan kinerja antara topologi *star* dan *mesh* pada kondisi *Line of Sight* (LOS). Pada grafik tersebut terlihat perbedaan yang mencolok dalam kemampuan kedua jaringan menangani pengiriman data. Topologi *star* menunjukkan stabilitas dan kecepatan tinggi, dengan rata-rata latensi yang konstan di angka 1 ms (terendah 1,114 ms pada interval 5 detik dan tertinggi 1,464 ms pada 30 detik). Nilai deviasi standar (STD) latensi pada topologi *star* juga kecil, berkisar antara 1,065 hingga 1,843, yang membuktikan bahwa jaringan ini hampir tidak mengalami fluktuasi atau penundaan waktu sama sekali, terlepas dari seberapa padat interval pengirimannya.



Gbr. 10 Kinerja Latensi Topologi Star (LOS)

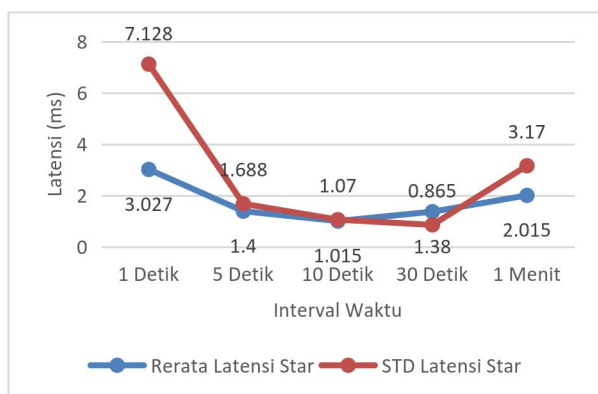


Gbr. 11 Kinerja Latensi Topologi Mesh (LOS)

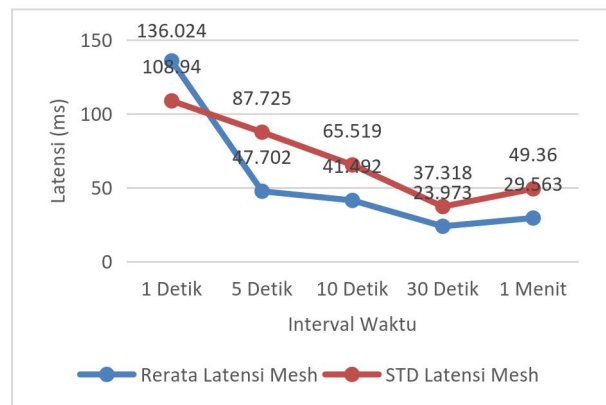
Sebaliknya, topologi *mesh* menunjukkan kinerja yang buruk dan tidak stabil pada interval waktu yang padat. Pada pengiriman berinterval 1 detik, rata-rata latensi topologi *mesh* melonjak tajam hingga 162,222 ms dengan sebaran data yang kacau, dibuktikan oleh nilai deviasi standar yang tinggi yaitu 130,79. Hal ini mengindikasikan terjadinya antrean data yang parah atau tabrakan paket (kolisi) di dalam jaringan saat node dipaksa mengirim data secara beruntun. Kinerja jaringan *mesh* baru mulai menunjukkan perbaikan dan penurunan latensi yang signifikan (16,308 ms) ketika beban pengiriman diregangkan menjadi setiap 30 detik. Secara keseluruhan, data ini mengonfirmasi bahwa arsitektur perutean pada topologi *mesh* kurang efisien untuk lalu lintas data berkecepatan tinggi, sedangkan topologi *star* mutlak jauh lebih andal untuk skenario tersebut.

b. Pengujian Latensi Kondisi NLOS

Berdasarkan data gambar grafik pengujian pada Gambar 12 dan Gambar 13 kondisi terhalang atau *Non-Line of Sight* (NLOS), topologi *star* terbukti tetap lebih tangguh dan konsisten dibandingkan topologi *mesh*. Meskipun sinyal harus melewati halangan fisik, rata-rata latensi pada jaringan *star* masih rendah dan beroperasi optimal di kisaran 1 hingga 2 ms. Memang teramati sedikit riak ketidakstabilan pada interval pengiriman paling ekstrem, yakni 1 detik, di mana latensinya naik menjadi rata-rata 3,027 ms dengan deviasi standar bernilai 7,128. Namun, nilai tersebut masih tergolong cepat dan andal.



Gbr. 12 Kinerja Latensi Topologi Star (NLOS)



Gbr. 13 Kinerja Latensi Topologi Mesh (NLOS)

Sebaliknya, topologi *mesh* kewalahan dan rentan terhadap gangguan saat dihadapkan pada lalu lintas data yang padat di kondisi NLOS. Pada pengiriman berinterval 1 detik, latensi rata-rata *mesh* melonjak drastis hingga 136,024 ms dengan fluktuasi yang liar (deviasi standar 108,94), menandakan antrean data yang parah akibat kombinasi halangan fisik dan lambatnya algoritma perutean antar-node. Kinerja jaringan *mesh* ini baru mulai "bernapas" dan menunjukkan perbaikan yang lumayan stabil ketika beban pengiriman diregangkan menjadi setiap 30 detik, di mana latensinya berhasil ditekan turun ke angka 23,973 ms. Secara keseluruhan, narasi ini semakin mempertegas kesimpulan sebelumnya: topologi *star* mutlak lebih stabil dan imun terhadap hambatan fisik maupun kepadatan interval waktu, sementara topologi *mesh* membutuhkan interval yang lambat (seperti 30 detik) agar data bisa diterima node dengan selamat tanpa jeda yang lama.

Terkait metode pencatatan waktu, nilai t_{send} diperoleh dari waktu lokal ($millis()$) pada node sensor saat paket dibentuk, sedangkan $t_{receive}$ dicatat oleh gateway saat paket diterima; node dan gateway tidak menjalankan sinkronisasi waktu absolut (seumpama *Network Time Protocol* atau NTP), melainkan mengandalkan selisih waktu relatif terhadap referensi awal (*uptime*) masing-masing perangkat. Pendekatan ini berpotensi menimbulkan bias kecil akibat perbedaan waktu mulai (*offset*) antarperangkat, sebagaimana tercermin dari sejumlah kecil pembacaan latensi bernilai negatif (misalnya -1 ms) pada data *star*, yang secara fisik kontradiktif dengan konsep bilangan/nilai mutlak/absolut pada definisi latensi itu sendiri, dan diperlakukan sebagai artefak pembulatan/*offset*, bukan representasi latensi negatif sesungguhnya.

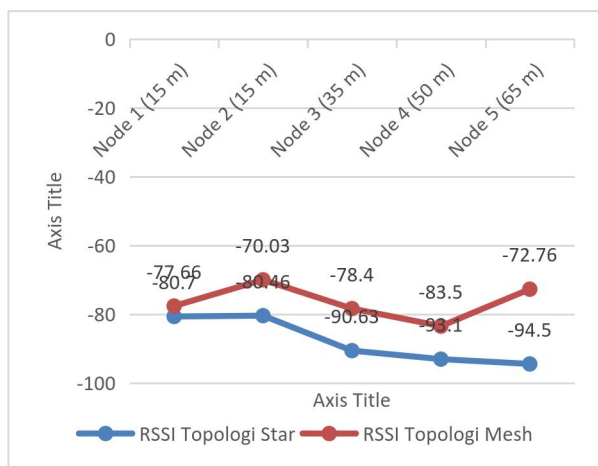
D. Pengujian Received Signal Strength Indicator

Pada uji *Received Signal Strength Indicator* (RSSI), digunakan salah satu sampel interval pengiriman data, yaitu 5 detik, dengan jumlah penerimaan data sebanyak 30 kali. Sampel tersebut digunakan sebagai representasi untuk analisis kinerja sistem. Sebagai klarifikasi, nilai RSSI tidak dihitung melalui kalkulasi matematis *software*, melainkan diperoleh secara langsung dari pembacaan level daya *hardware* radio frekuensi (modul

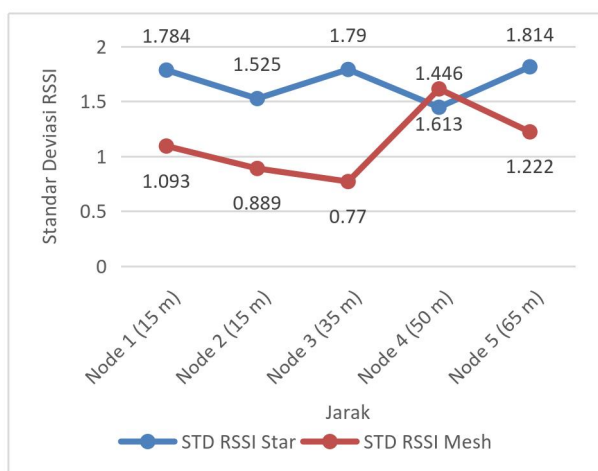
transceiver) pada ESP32 *gateway* setiap kali sebuah paket data berhasil ditangkap (dalam satuan dBm). Nilai ukur akhir RSSI yang dianalisis adalah hasil kalkulasi nilai rata-rata (*mean*) dari seluruh paket data yang berhasil diterima *node* selama durasi pengujian untuk meminimalisir anomali fluktuasi sinyal radio di medium udara.

a. Pengujian RSSI Kondisi LOS

Berdasarkan hasil data pengujian kuat sinyal (RSSI) pada Gambar 14 dan Gambar 15 kondisi *Line of Sight* (LOS), terlihat perbedaan karakteristik yang sangat mencolok antara arsitektur jaringan *star* dan *mesh* seiring bertambahnya jarak titik sensor. Pada topologi *star*, kualitas penerimaan sinyal mengalami pelemahan yang linier dan konsisten sejalan dengan semakin jauhnya posisi jarak *node* dari titik pusat (*gateway*). Hal ini terbukti dari nilai rata-rata RSSI yang berangsur melemah dari -80,7 dBm pada jarak 15 meter (*node* 1) hingga mencapai titik paling kritis yaitu -94,5 dBm pada jarak 65 meter (*node* 5). Meskipun demikian, tingkat fluktuasi sinyal pada jaringan *star* tetap tergolong stabil pada setiap jarak, ditunjukkan oleh nilai deviasi standar yang konstan berada di kisaran 1,4 hingga 1,8.



Gbr. 14 Rerata RSSI Terhadap Jarak (LOS)

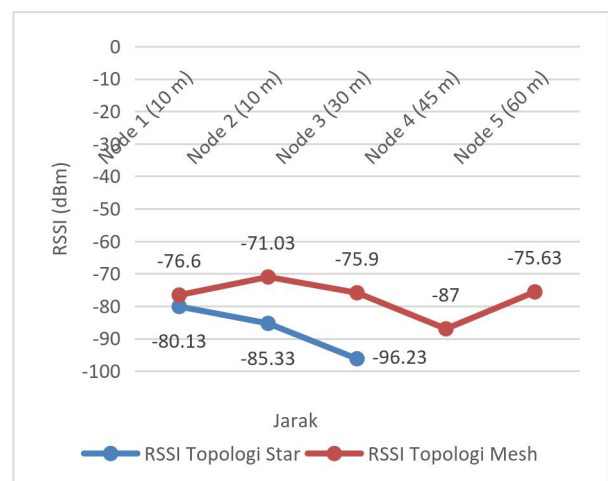


Gbr. 15 Standar Deviasi RSSI terhadap Jarak

Sebaliknya, topologi *mesh* menunjukkan keunggulan mutlak dalam mempertahankan kualitas sinyal di area jangkauan yang lebih luas. Berkat kemampuan *multi-hop* di mana setiap *node* dapat saling bertindak sebagai penguat atau penerus data bagi *node* lainnya, jarak fisik ujung ke ujung tidak lagi menjadi kendala utama pelemahan sinyal. Fakta ini jelas terlihat pada *node* 5 yang berada di jarak terjauh (65 meter) namun tetap mampu mendapatkan kuat sinyal yang prima di angka -72,76 dBm, jauh lebih baik dibandingkan jaringan *star* di titik yang sama. Selain itu, nilai deviasi standar pada jaringan *mesh* juga secara umum lebih rendah (berkisar antara 0,77 hingga 1,613), menandakan bahwa sinyal yang diterima tidak hanya lebih kuat tetapi juga lebih minim fluktuasi. Secara keseluruhan, topologi *mesh* unggul untuk mempertahankan kualitas sinyal pada cakupan jarak yang jauh berkat sistem perutean antar-*nodenya*, sedangkan kinerja topologi *star* dibatasi oleh seberapa dekat jarak fisik *node* terhadap pusat kendalanya.

b. Pengujian RSSI Kondisi NLOS

Berdasarkan hasil data pengujian kuat sinyal (RSSI) pada Gambar 16 dan Gambar 17 kondisi terhalang atau *Non-Line of Sight* (NLOS), kelemahan fatal dari topologi *star* saat menghadapi hambatan fisik menjadi nyata. Pada jarak dekat (10 meter), topologi *star* masih mampu menangkap sinyal dengan rata-rata RSSI -80,13 dBm hingga -85,33 dBm, dan kualitas sinyalnya masih cukup stabil dengan deviasi standar di bawah 1. Namun, pelemahan sinyal terjadi drastis seiring bertambahnya jarak dan rintangan, di mana pada jarak 30 meter (*node* 3) kuat sinyal *star* anjlok ke titik kritis -96,23 dBm. Puncaknya, jaringan *star* benar-benar kehilangan jangkauan (*blank spot*) pada jarak 45 meter (*node* 4) dan 60 meter (*node* 5), terbukti dari ketiadaan data penerimaan sinyal sama sekali pada kedua titik tersebut.



Gbr. 16 Rerata RSSI terhadap Jarak (NLOS)

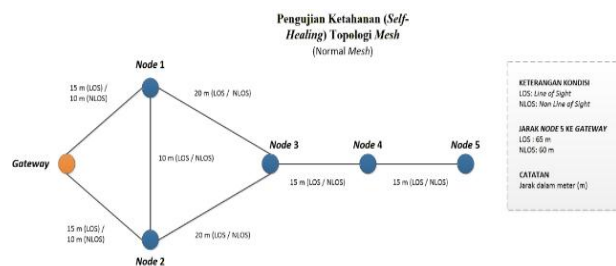


Gbr. 17 Standar Deviasi RSSI terhadap Jarak (NLOS)

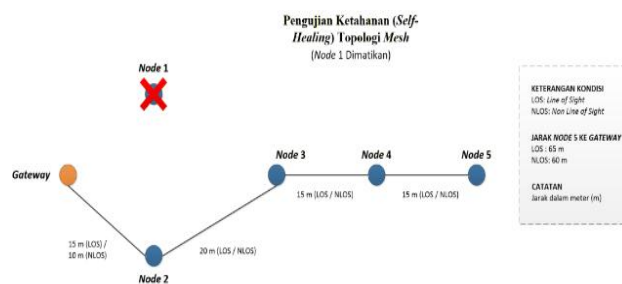
Sebaliknya, kemampuan perutean *multi-hop* pada topologi *mesh* kembali menunjukkan keunggulannya secara absolut di skenario NLOS. Jaringan *mesh* berhasil mengatasi masalah halangan fisik dan pelemahan jarak dengan cara merelai sinyal antar-*node* secara estafet. Hal ini dibuktikan dengan kemampuannya menjangkau seluruh area pengujian hingga titik terjauh di *node 5* (60 meter) dengan kuat sinyal yang prima, yakni -75,63 dBm dan tingkat stabilitas yang tinggi (deviasi standar hanya 0,718). Secara keseluruhan, data ini menyimpulkan bahwa arsitektur jaringan *mesh* adalah solusi tunggal yang andal untuk cakupan sinyal di area luas dengan banyak rintangan (NLOS), sementara jaringan *star*, dalam hal capaian cakupan, tidak direkomendasikan karena gagal mencapai titik-titik terjauh dalam skenario ini.

E. Analisis Ketahanan (Self-Healing) Topologi Mesh

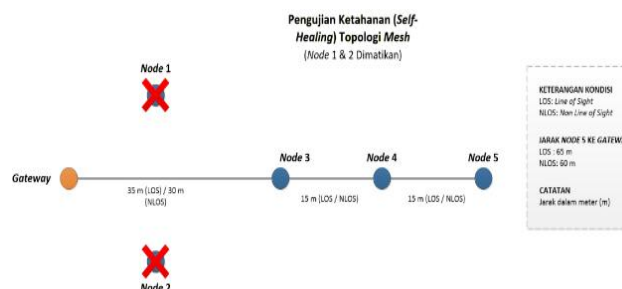
Ketahanan mandiri (*self-healing*) jaringan *mesh* diuji untuk mengukur kemampuan jaringan dalam melakukan komunikasi saat *node* relai gagal. Pada penelitian ini, dipertimbangkan sejumlah skenario, yakni kondisi normal (Gambar 18), pematian *node 1* (Gambar 19), pematian *node 2*, serta pematian *node 2*, serta pematian *node 1* dan *node 2* secara bersamaan (Gambar 20), masing-masing pada kondisi LOS dan non-LOS. Pengujian dilakukan dengan pencatatan 30 kali penerimaan data, menggunakan sampel berinterval 5 detik.



Gbr. 18 Pengujian Ketahanan Mesh Kondisi Normal



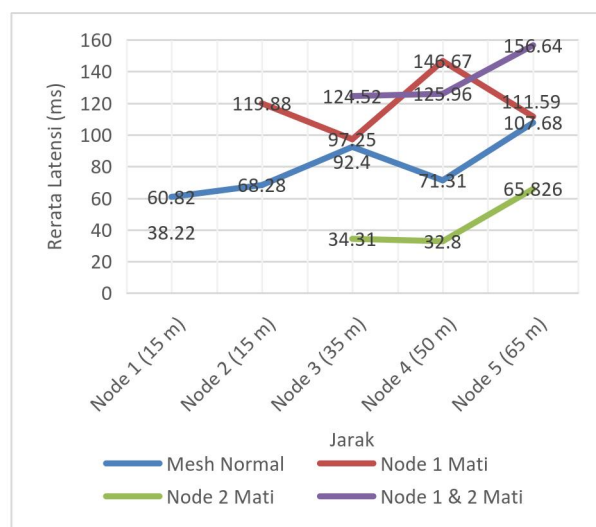
Gbr. 19 Pengujian Ketahanan Mesh Kondisi Pematian Node 1



Gbr. 20 Pengujian Ketahanan Mesh Kondisi Pematian Node 1 dan Node 2

a. Self-healing Kondisi LOS

Berdasarkan Gambar 21, pada kondisi normal tanpa halangan, rerata latensi berada di rentang 60,82 ms hingga 107,68 ms.



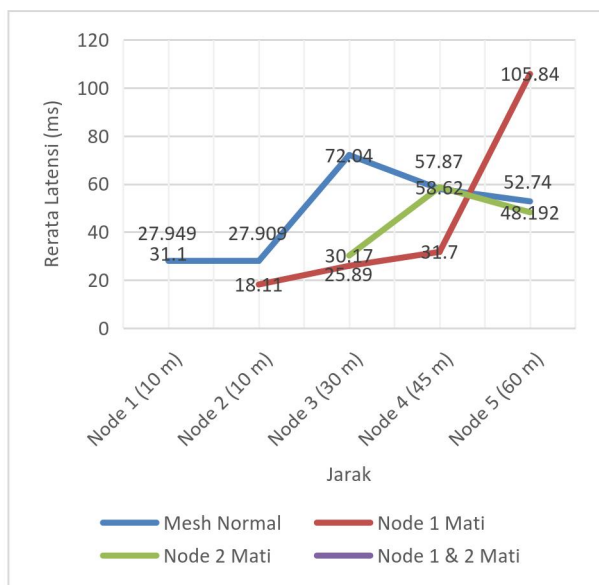
Gbr. 21 Grafik Analisis Ketahanan Topologi Mesh, Kondisi LOS

Ketika skenario kegagalan *node* diterapkan, terjadi perubahan *routing* dinamis yang berdampak pada waktu tunda pengiriman. Saat *node 1* dimatikan, *node* lain harus mencari rute alternatif yang lebih panjang, sehingga terjadi lonjakan latensi, salah satunya pada *node 4* yang latensinya 146,67 ms. Lonjakan latensi tertinggi tercatat saat skenario ekstrem, yakni saat kedua *node* perantara (*node 1* dan *node 2*) dimatikan secara bersamaan. Pada kondisi ini, data *node* terjauh (*node 5*) berusaha mencari jalur lain tanpa bantuan *node* perantara utama, sehingga latensinya melonjak tajam, menjadi

156,64 ms. Meski latensi meningkat signifikan, ketiadaan data yang gagal terkirim membuktikan bahwa algoritma pencarian rute alternatif secara mandiri (*self-healing*) pada *mesh* secara optimal berfungsi di kondisi LOS.

b. Pengujian RSSI Kondisi NLOS

Berdasarkan data yang tersaji pada Gambar 22, pada situasi dengan hambatan fisik (non-LOS), *self-healing* menghadapi tantangan yang jauh lebih berat. Saat jaringan beroperasi normal, latensi masih dapat ditekan di bawah 75 ms berkat *node* perantara yang meneruskan sinyal melewati rintangan. Namun, saat *node* 1 dimatikan, jaringan kehilangan salah satu jembatan utamanya. Akibatnya, paket data dari *node* 5 harus mencari lompatan *hop* memutar yang lebih jauh menembus hambatan, sehingga latensinya melonjak drastis menjadi 105,84 ms dibandingkan kondisi normal, hanya 52,74 ms. Lonjakan latensi yang signifikan ini membuktikan bahwa hilangnya rute utama memaksa sistem melakukan pengalihan jalur *rerouting* yang memakan waktu lebih lama. Meskipun mekanisme *self-healing* berhasil menjaga konektivitas tetap terhubung, efisiensi pengiriman data menurun drastis akibat bertambahnya jumlah lompatan *hop count* dalam kondisi lingkungan yang buruk.



Gbr. 22 Grafik Analisis Ketahanan Topologi *Mesh*, Kondisi NLOS

Temuan paling krusial pada pengujian ini terjadi pada skenario pemutusan kedua *node* (*node* 1 dan *node* 2 dimatikan bersamaan). Seluruh *node* yang tersisa mencatat kegagalan komunikasi total (*link failure*) yang ditandai dengan tidak adanya nilai latensi yang dapat direkam. Hal ini membuktikan bahwa batas maksimal kemampuan *self-healing* topologi *mesh* bergantung pada ketersediaan minimal satu *node* perantara (*gateway* relai) ketika jaringan dihadapkan pada redaman sinyal yang ekstrem akibat halangan.

V. KESIMPULAN

Berdasarkan hasil evaluasi kinerja sistem *Wireless Sensor Network* (WSN) berbasis ESP32 pada rentang jarak pengujian hingga 65 meter untuk kondisi LOS dan 60 meter kondisi NLOS, dapat disimpulkan bahwa penerapan protokol ESP-NOW (topologi *star*) dan *painlessMesh* (topologi *mesh multi-hop*) memiliki *trade-off* yang signifikan, tergantung pada kondisi lingkungan fisik.

Pada kondisi *Line of Sight* (LOS), protokol ESP-NOW dengan topologi *star* memberikan performa terbaik dengan nilai *Packet Delivery Ratio* (PDR) mencapai 100% dan latensi yang rendah karena transmisi data dilakukan secara langsung (*direct-link*) ke *gateway*. Namun, kinerjanya menurun drastis pada kondisi *Non-Line of Sight* (NLOS) dengan hambatan vegetasi, di mana nilai PDR mengalami penurunan hingga 0% pada jarak 45–60 meter akibat pelemahan sinyal (RSSI).

Sementara itu, protokol *painlessMesh* dengan topologi *mesh multi-hop* mampu mempertahankan komunikasi jaringan dengan nilai PDR berkisar antara 96% hingga 98% baik pada kondisi LOS maupun NLOS karena adanya mekanisme *forwarding* antar-*node*. Selain itu, *painlessMesh* terbukti memiliki kemampuan *self-healing* yang dapat mencari rute alternatif saat *node* relai mengalami gangguan. Namun, komunikasi jaringan akan terputus ketika seluruh *node* relai utama mengalami kegagalan secara bersamaan.

Secara keseluruhan, kesesuaian implementasi terikat pada karakteristik lingkungan dan protokol yang digunakan. Konfigurasi ESP-NOW (*star*) sesuai diimplementasikan untuk kebutuhan komunikasi berlatensi rendah pada lingkungan tanpa hambatan fisik. Sebaliknya, konfigurasi *painlessMesh* (*mesh*) lebih ideal diterapkan pada lingkungan dengan hambatan fisik (seperti area bervegetasi), di mana keandalan pengiriman paket data dan ketahanan jaringan lebih diutamakan dibandingkan kecepatan latensi.

REFERENSI

- [1] Afifie, N. A., Khang, A. W. Y., Bin Ja'afar, A. S., Amin, A. F. B. M., Alsayaydeh, J. A. J., Indra, W. A., Herawan, S. G., & Ramli, A. B. (2021). Evaluation Method of Mesh Protocol over ESP32 and ESP8266. *Baghdad Science Journal*, 18(4), 42. [https://doi.org/10.21123/bsj.2021.18.4\(Suppl.\).1397](https://doi.org/10.21123/bsj.2021.18.4(Suppl.).1397).
- [2] Armanda, A. (2025). *Implementasi Jaringan Sensor Nirkabel Berbasis ESP32 dengan Topologi Mesh untuk Pemantauan Lingkungan* [Laporan tugas akhir tidak diterbitkan]. Politeknik Negeri Bandung.
- [3] Arofah, M. F., Mandayatma, E., & Nurcahyo, S. (2023). Penerapan Protokol Komunikasi ESP-Now pada Portable Traffic Light. *Jurnal Elektronika dan Otomasi Industri*, 10(1), 52-59. <http://dx.doi.org/10.33795/elkolind.v10i1.2749>.

- [4] Candra, R., & Busran, B. (2025). Implementasi Jaringan WSN pada Sistem Deteksi Banjir menggunakan Modul NRF24L01 PA+ NA. *Jurnal Informatika dan Teknik Elektro Terapan*, 13(3S1). <https://doi.org/10.23960/jitet.v13i3S1.7624>.
- [5] Espressif Systems. (2026a, Mei). *ESP32 Technical Reference Manual* (Versi 5.7).
- [6] Espressif Systems. (2026b, Mei). *ESP-Techpedia* [Dokumentasi Teknis].
- [7] Fitrian, H. P., Latifah, T. S., Imroatuddin, M., Maulana, I. A., & Al Anshari, F. A. (2025). Analisis Performa Topologi Star dan Mesh dalam Implementasi Jaringan LAN pada Lingkungan Perkantoran. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 9(1), 1399-1403. <https://doi.org/10.36040/jati.v9i1.12539>.
- [8] Zijie, F., Al-Shareeda, M. A., Saare, M. A., Manickam, S., & Karuppayah, S. (2023). Wireless Sensor Networks in The Internet of Things: Review, Techniques, Challenges, and Future Directions. *Indonesian Journal of Electrical Engineering and Computer Science*, 31(2), 1190-1200. <http://doi.org/10.11591/ijeecs.v31.i2.pp1190-1200>.
- [9] Ghozali, M. I., Murti, A. C., Sugiharto, W. H., & Roder, K. (2025). Performance Comparison of WSN Topologies in IoT-Based Water Quality Monitoring Systems. *ZERO: Jurnal Sains, Matematika dan Terapan*, 9(2), 396-406. <http://dx.doi.org/10.30829/zero.v9i2.25706>.
- [10] Irmayani, I., Marwita, F., Newton, I., & Hakim, L. (2025). Prototipe Sistem Komunikasi Data Text Wireless Topologi Mesh untuk Meningkatkan Area Cakupan Berbasis ESP32. *Sinusoida: Jurnal Penelitian dan Pengkajian Elektro*, 27(1), 1-12. <https://doi.org/10.37277/s.v27i1.2417>.
- [11] Mardamsyah, A., Saragih, G. W., Zhafran, A. F., Simanjuntak, I. R., & Riadi, M. F. (2024). Penerapan Komunikasi ESP-Now dengan Topologi Jaringan Mesh sebagai Bentuk Interoperability dalam Konsep Network Centric Warfare. *Journal of Education, Humaniora and Social Sciences*, 7(2), 707-727. <https://doi.org/10.34007/jehss.v7i2.2374>.
- [12] OpenJS Foundation and Contributors. (2026). *Node-RED: Low-code Programming for Event-Driven Applications*. <https://nodered.org>.
- [13] Prasetya, M. F., Budi, A. S., & Akbar, S. R. (2023, October). Time Synchronization on ESP-NOW Based Wireless Sensor Network using Flooding Time Synchronization Protocol. In *Proceedings of the 8th International Conference on Sustainable Information Engineering and Technology* (pp. 608-611). <https://doi.org/10.1145/3626641.3626683>.
- [14] Ramadhan, S. (2025). *Perancangan Jaringan Berbasis Mesh dengan Protokol OLSRv2 pada Resource Constrained Device* [Tesis magister]. Institut Teknologi Bandung.
- [15] Santos, L., Costa, T., Caldeira, J. M., & Soares, V. N. (2022). Performance Assessment of ESP8266 Wireless Mesh Networks. *Information*, 13(5), 210. <https://doi.org/10.3390/info13050210>.
- [16] Trigka, M., & Dritsas, E. (2025). Wireless Sensor Networks: from Fundamentals and Applications to Innovations and Future Trends. *IEEE Access*, 13, 96365-96399. <https://doi.org/10.1109/ACCESS.2025.3572328>.
- [17] Urazayev, D., Eduard, A., Ahsan, M., & Zorbas, D. (2023, May). Indoor Performance Evaluation of ESP-Now. In *2023 IEEE International Conference on Smart Information Systems and Technologies (SIST)* (pp. 1-6). IEEE. <https://doi.org/10.1109/SIST58284.2023.10223585>.
- [18] Puri Wedyanti, F., & Wagya, A. (2025). Analisis Performansi ESP-NOW sebagai Protokol Komunikasi Efisien antar Multi-node dan Gateway dalam Sistem IoT. *Seminar Nasional Inovasi Vokasi*, 4, 607-614. Retrieved from <https://prosiding.pnj.ac.id/index.php/sniv/article/view/3937>.
- [19] Uzougbo, O. I., Olanrewaju, O. A., & Kayode, A. K. (2024). Node-RED and IoT Analytics: A Real-Time Data Processing and Visualization Platform. *Tech-Sphere Journal of Pure and Applied Sciences (TSJPAS)*, 1(1), 1-12. <https://doi.org/10.5281/zenodo.13856860>.