

RANCANG BANGUN SISTEM KOMUNIKASI OFFLINE BERBASIS LORA SEBAGAI SOLUSI DI WILAYAH MINIM SINYAL

Maulana Aullaur Rahman; Fajar Suryawan, S.T., M.Eng.Sc., Ph.D
Program Studi Teknik Elektro, Fakultas Teknik,
Universitas Muhammadiyah Surakarta

Abstrak

Penelitian ini berfokus pada perancangan dan pengembangan sistem komunikasi offline berbasis *Long Range* (LoRa) menggunakan mikrokontroler ESP32 sebagai solusi komunikasi pada wilayah minim sinyal maupun daerah terdampak bencana. Sistem dikembangkan dengan metode komunikasi *peer-to-peer* sehingga memungkinkan pertukaran pesan teks tanpa bergantung pada jaringan internet maupun jaringan seluler. Aplikasi Flutter digunakan sebagai antarmuka pengguna yang dilengkapi mini server untuk mendukung penyimpanan dan sinkronisasi riwayat percakapan. Pengujian sistem dilakukan berdasarkan fungsionalitas komponen, pengiriman pesan, sinkronisasi riwayat chat, serta kualitas komunikasi menggunakan parameter *Received Signal Strength Indicator* (RSSI) dan *Signal-to-Noise Ratio* (SNR). Hasil pengujian menunjukkan bahwa sistem mampu melakukan pengiriman pesan dengan baik hingga jarak 150 meter pada kondisi *Line of Sight* (LOS), sedangkan pada jarak 175 hingga 200 meter komunikasi mulai mengalami kegagalan akibat penurunan kualitas sinyal. Selain itu, fitur sinkronisasi riwayat chat berhasil berjalan dengan tingkat keberhasilan 100% hingga jarak 150 meter, sehingga node yang baru bergabung dapat memperoleh seluruh riwayat percakapan sebelum melanjutkan komunikasi. Hasil penelitian menunjukkan bahwa sistem mampu menjadi alternatif komunikasi yang sederhana, efisien, dan andal pada wilayah dengan keterbatasan jaringan internet maupun seluler.

Kata Kunci: LoRa, ESP32, komunikasi *offline*, *peer-to-peer*, komunikasi radio, RSSI, Flutter.

Abstract

This research focuses on the design and development of an offline communication system based on Long Range (LoRa) technology using the ESP32 microcontroller as an alternative communication solution for areas with limited cellular and internet network coverage, including remote and disaster-affected regions. The system was developed using a peer-to-peer communication method, enabling users to exchange text messages without relying on conventional communication infrastructure. A Flutter application was implemented as the user interface and integrated with a mini server to support local message storage and chat history synchronization. System performance was evaluated through component functionality testing, message transmission testing, chat history synchronization testing, and communication quality analysis using Received Signal Strength Indicator (RSSI) and Signal-to-Noise Ratio (SNR) parameters. The results show that the system successfully transmitted messages up to a distance of 150 meters under Line of Sight (LOS) conditions, while communication began to fail at distances between 175 and 200 meters due to signal degradation. In addition, the chat history synchronization feature achieved a 100% success rate up to 150 meters, allowing newly joined nodes to retrieve previous conversations before continuing communication. These results indicate

that the proposed system can serve as a simple, efficient, and reliable communication alternative for areas with limited or unavailable internet and cellular network coverage.

Keywords: LoRa, ESP32, offline communication, peer-to-peer, radio communication, RSSI, Flutter.

1. PENDAHULUAN

Komunikasi merupakan kebutuhan dasar manusia untuk menyampaikan informasi, baik dalam kegiatan sehari-hari maupun dalam kondisi darurat. Perkembangan teknologi komunikasi saat ini sangat bergantung pada jaringan internet dan infrastruktur seluler. Namun, tidak semua wilayah memiliki akses sinyal yang memadai, terutama di daerah terpencil, pegunungan, maupun wilayah terdampak bencana. Menurut Badan Nasional Penanggulangan Bencana (BNPB), kerusakan infrastruktur telekomunikasi akibat gempa bumi, banjir, tanah longsor, maupun bencana lainnya dapat menyebabkan terputusnya akses komunikasi sehingga menghambat proses koordinasi, dan penyebaran informasi (Badan Nasional Penanggulangan Bencana, 2025). Kondisi tersebut menunjukkan pentingnya pengembangan sistem komunikasi alternatif yang mampu beroperasi tanpa ketergantungan pada jaringan operator maupun internet.

Salah satu teknologi yang berpotensi digunakan sebagai solusi adalah *Long Range* (LoRa). LoRa merupakan teknologi komunikasi nirkabel berdaya rendah yang mampu melakukan transmisi data pada jarak jauh dengan konsumsi daya yang relatif kecil (Augustin et al., 2016). Teknologi ini banyak diterapkan pada sistem *Internet of Things* (IoT), pemantauan jarak jauh, dan aplikasi komunikasi di wilayah dengan keterbatasan infrastruktur jaringan (Adelantado et al., 2017). Karakteristik tersebut menjadikan LoRa sebagai teknologi yang sesuai untuk mendukung sistem komunikasi darurat dan komunikasi pada daerah minim sinyal (paul S, 2024).

Beberapa penelitian sebelumnya telah menunjukkan bahwa teknologi LoRa mampu digunakan sebagai media komunikasi data jarak jauh dengan tingkat konsumsi daya yang rendah dan jangkauan yang luas (Ramadhani et al., 2025). Pada penelitian yang dilakukan oleh (Fadillah et al., 2022), berhasil mengembangkan prototipe komunikasi antar smartphone menggunakan modul LoRa SX1278 dan Bluetooth HC-06 sehingga pengguna dapat bertukar pesan tanpa memanfaatkan jaringan internet maupun jaringan seluler. Penelitian tersebut juga menunjukkan bahwa teknologi LoRa dapat dimanfaatkan sebagai media komunikasi alternatif pada kondisi bencana maupun wilayah yang sulit memperoleh akses internet. Selain itu, penerapan LoRa pada sistem komunikasi *peer-to-peer* memungkinkan pertukaran informasi secara langsung antar perangkat tanpa memerlukan jaringan seluler maupun akses internet. Meskipun demikian, sebagian besar penelitian masih berfokus pada pengiriman data sensor dan belum banyak mengembangkan sistem komunikasi teks darurat

(*messaging system*) yang mudah digunakan oleh masyarakat melalui perangkat *smartphone* (Februari et al., 2026).

Penelitian lain yang dilakukan oleh (Farras Fauzan et al., 2021). mengembangkan alat komunikasi darurat berbasis ESP8266 dan LoRa yang ditujukan bagi pendaki gunung. Sistem tersebut memanfaatkan jaringan WiFi yang dibentuk oleh ESP8266 sehingga pengguna dapat mengakses antarmuka komunikasi melalui browser pada *smartphone* tanpa memerlukan koneksi internet. Selain mendukung komunikasi berbasis LoRa, sistem juga dilengkapi dengan modul GPS untuk menampilkan lokasi pengguna sebagai pendukung koordinasi pada kondisi darurat.

Berdasarkan kondisi tersebut, permasalahan yang diangkat dalam penelitian ini adalah bagaimana merancang dan mengembangkan sistem komunikasi *offline* berbasis LoRa yang mampu digunakan untuk pertukaran pesan teks antar pengguna tanpa bergantung pada infrastruktur komunikasi konvensional. Selain itu, diperlukan evaluasi terhadap kualitas komunikasi yang dihasilkan untuk mengetahui tingkat keandalan sistem pada berbagai kondisi pengujian.

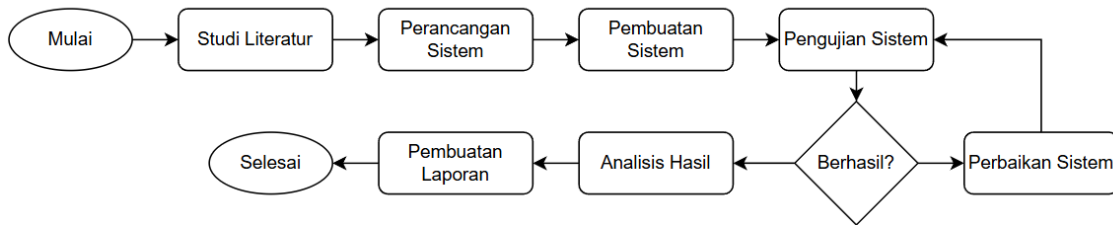
Penelitian ini bertujuan untuk merancang dan membangun sistem komunikasi *offline* berbasis LoRa menggunakan mikrokontroler ESP32 dan aplikasi Flutter sebagai antarmuka pengguna. Sistem dikembangkan menggunakan metode komunikasi *peer-to-peer* sehingga memungkinkan pengiriman dan penerimaan pesan teks secara langsung antar *node*. Kinerja sistem dievaluasi berdasarkan parameter jarak komunikasi, *Received Signal Strength Indicator* (RSSI), dan *Signal-to-Noise Ratio* (SNR).

Hasil penelitian ini diharapkan dapat menghasilkan sistem komunikasi alternatif yang sederhana, efisien, dan andal untuk digunakan pada kondisi darurat maupun wilayah dengan keterbatasan akses jaringan. Selain itu, penelitian ini diharapkan dapat menjadi referensi dalam pengembangan teknologi komunikasi berbasis LoRa yang efektif untuk mitigasi bencana di masa depan.

2. METODE

2.1 Metode Penelitian

Penelitian ini menggunakan metode *Research and Development* yang bertujuan untuk merancang, membangun, dan menguji sistem komunikasi *offline* berbasis LoRa sebagai alternatif komunikasi pada wilayah yang tidak memiliki akses jaringan seluler maupun internet. Metode ini dipilih karena penelitian tidak hanya menghasilkan kajian teoritis, tetapi juga menghasilkan sebuah prototipe yang dapat diimplementasikan dan diuji secara langsung. Tahapan penelitian yang dilakukan meliputi studi literatur, perancangan sistem, pembuatan sistem, pengujian sistem, perbaikan sistem, analisis hasil, dan pembuatan laporan. Alur metode penelitian yang digunakan dapat dilihat pada Gambar 1.



Gambar 1. Alur Metode Penelitian

A. Studi literatur

Tahap pertama yang dilakukan adalah studi literatur. Pada tahap ini dilakukan pengumpulan berbagai referensi yang berkaitan dengan topik penelitian, baik dari jurnal ilmiah, buku, maupun dokumentasi resmi. Literatur yang dipelajari meliputi teknologi LoRa, ESP32, Flutter, komunikasi nirkabel, serta penelitian terdahulu yang relevan. Tahap ini bertujuan untuk memperoleh pemahaman mengenai teknologi yang digunakan sekaligus menjadi dasar dalam proses perancangan sistem.

B. Perancangan sistem

Setelah memperoleh landasan teori yang cukup, tahap berikutnya adalah perancangan sistem. Pada tahap ini dilakukan perancangan arsitektur sistem, diagram blok, wiring diagram, serta rancangan perangkat lunak yang akan digunakan. Perancangan dilakukan untuk menggambarkan cara kerja sistem dan hubungan antar komponen sehingga proses implementasi dapat dilakukan dengan lebih terarah.

C. Pembuatan sistem

Tahap pembuatan sistem merupakan proses implementasi dari rancangan yang telah dibuat sebelumnya. Pada tahap ini dilakukan perakitan perangkat keras yang terdiri dari ESP32, modul LoRa SX1278, OLED, LED, buzzer, dan rangkaian catu daya. Selain itu, dilakukan juga pengembangan perangkat lunak pada ESP32 dan aplikasi Flutter yang berfungsi sebagai antarmuka pengguna dalam mengirim maupun menerima pesan.

D. Pengujian sistem

Setelah sistem berhasil dibangun, dilakukan pengujian untuk memastikan seluruh fungsi dapat berjalan sesuai dengan rancangan. Pengujian dilakukan dengan mengirim pesan menggunakan aplikasi flutter dan melihat jarak yang bisa dijangkau oleh LoRa

untuk mengirim pesan, dan juga memantau nilai kekuatan sinyal LoRa seperti RSSI, dan SNR.

E. Perbaikan sistem

Apabila selama proses pengujian ditemukan kendala atau fungsi yang belum berjalan sesuai harapan, maka dilakukan perbaikan pada sistem. Perbaikan dapat berupa penyesuaian perangkat keras maupun penyempurnaan program hingga sistem dapat beroperasi dengan baik.

F. Analisis hasil

Data yang diperoleh dari hasil pengujian kemudian dianalisis untuk mengetahui kinerja sistem yang telah dibangun. Analisis dilakukan terhadap keberhasilan komunikasi, kualitas sinyal yang diterima, waktu pengiriman data, serta kemampuan sistem dalam melakukan sinkronisasi riwayat percakapan antar *node*.

G. Pembuatan laporan

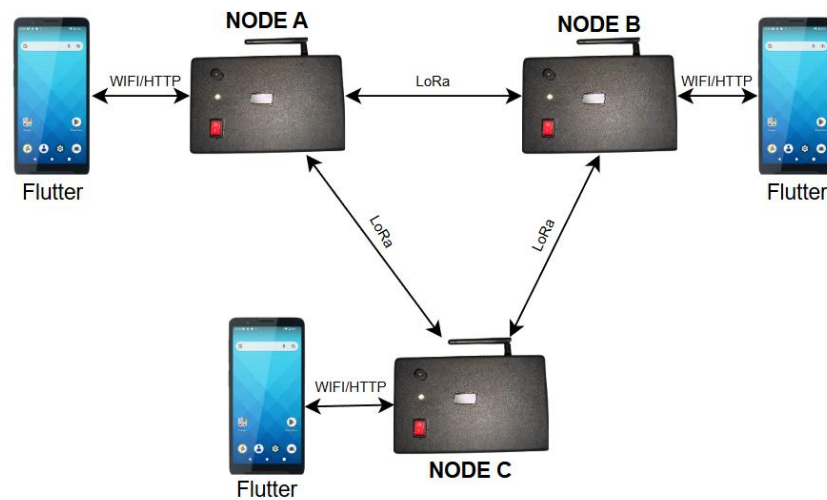
Tahap terakhir adalah penyusunan laporan tugas akhir. Seluruh proses penelitian mulai dari studi literatur hingga analisis hasil didokumentasikan secara sistematis sesuai dengan format penulisan yang berlaku. Laporan ini disusun sebagai bentuk pertanggungjawaban ilmiah sekaligus dokumentasi dari penelitian yang telah dilakukan.

2.2 Teknologi LoRa

LoRa (Long Range) merupakan teknologi komunikasi nirkabel yang dirancang untuk mendukung transmisi data jarak jauh dengan konsumsi daya yang rendah melalui teknik modulasi *Chirp Spread Spectrum* (CSS). Teknologi ini banyak digunakan pada berbagai aplikasi Internet of Things (IoT) karena mampu menyediakan jangkauan komunikasi yang luas dengan kebutuhan energi yang efisien. Pada penelitian ini digunakan modul LoRa SX1278 yang beroperasi pada frekuensi 433 MHz. Pemilihan frekuensi tersebut didasarkan pada kompatibilitas modul yang digunakan serta ketersediaan perangkat dengan harga yang relatif lebih terjangkau dibandingkan modul LoRa pada pita frekuensi lainnya, sehingga sesuai untuk pengembangan prototipe penelitian. Meskipun demikian, penggunaan frekuensi radio harus memperhatikan regulasi spektrum yang berlaku pada masing-masing negara, termasuk ketentuan mengenai rentang frekuensi dan batas daya pancar yang diizinkan. Oleh karena itu, implementasi sistem pada penelitian ini difokuskan untuk keperluan penelitian dan pengujian performa komunikasi.

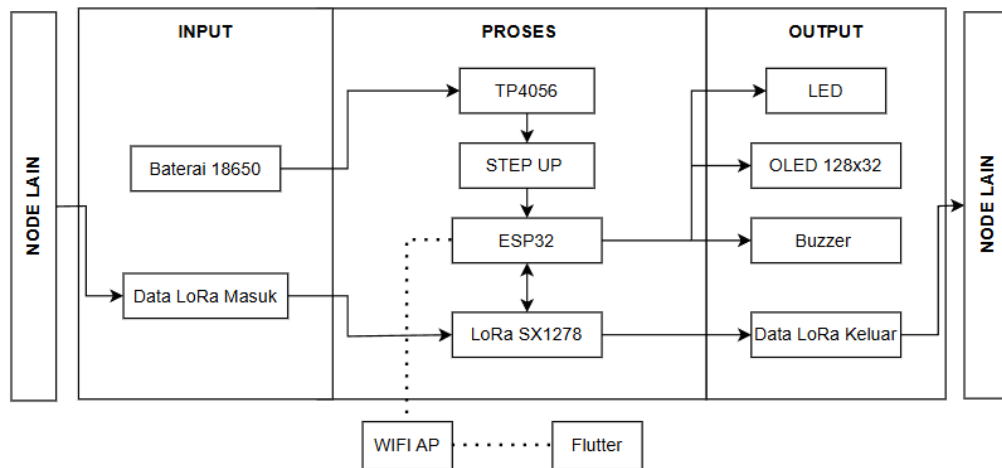
2.3 Perancangan *Hardware* Sistem

Tahap perancangan *hardware* sistem bertujuan untuk menggambarkan struktur dan mekanisme kerja sistem yang dibangun. Perancangan ini meliputi penyusunan arsitektur sistem, diagram sistem, serta wiring diagram yang menunjukkan hubungan antar komponen perangkat keras. Melalui tahap perancangan ini diharapkan seluruh komponen dapat terintegrasi dengan baik sehingga sistem komunikasi *offline* berbasis LoRa dapat beroperasi sesuai dengan kebutuhan penelitian.



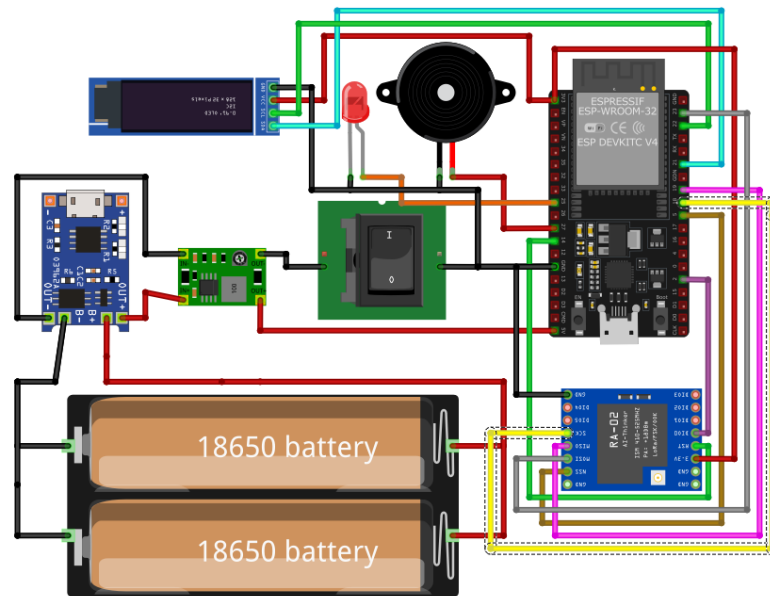
Gambar 2. *Arsitektur Sistem*

Gambar 2 menunjukkan arsitektur sistem komunikasi *offline* berbasis LoRa yang terdiri dari tiga *node*, yaitu *Node A*, *Node B*, dan *Node C*. Setiap *node* terdiri dari sebuah perangkat ESP32 yang terintegrasi dengan modul LoRa SX1278 serta aplikasi Flutter yang berjalan pada smartphone sebagai antarmuka pengguna. Komunikasi antara aplikasi Flutter dan ESP32 dilakukan melalui jaringan WiFi *Access Point* yang dibentuk oleh ESP32 dengan menggunakan protokol HTTP untuk proses pengiriman dan penerimaan data. Komunikasi antar *node* dilakukan secara peer-to-peer menggunakan teknologi LoRa tanpa memerlukan gateway, server, maupun koneksi internet. Dengan mekanisme tersebut, setiap *node* dapat mengirim dan menerima pesan secara langsung dari *node* lain yang berada dalam jangkauan komunikasi LoRa. Arsitektur ini memungkinkan sistem tetap beroperasi pada wilayah yang tidak memiliki akses jaringan seluler maupun internet sehingga sesuai untuk digunakan pada daerah terpencil maupun kondisi darurat akibat bencana.



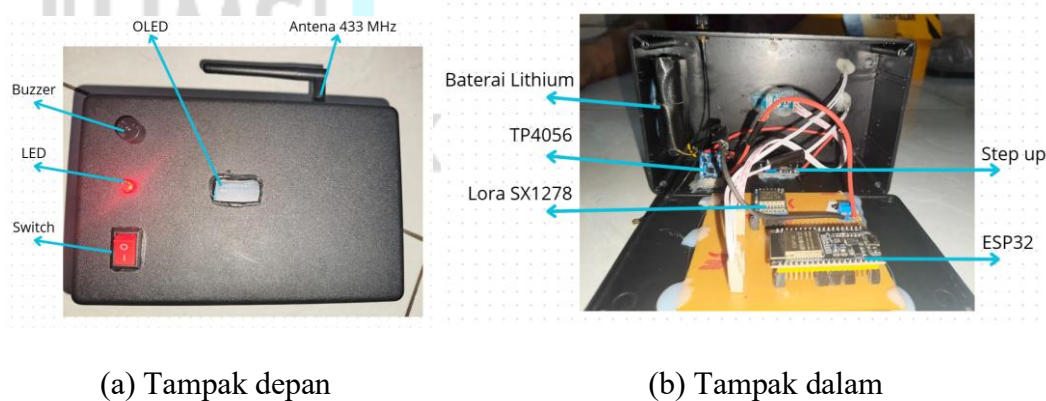
Gambar 3. Diagram Sistem

Diagram sistem pada Gambar 3 menggambarkan alur kerja sistem komunikasi *offline* berbasis LoRa yang terdiri atas tiga bagian utama, yaitu input, proses, dan output. Pada bagian input, sistem memperoleh sumber daya dari baterai 18650 yang digunakan untuk menyuplai kebutuhan listrik seluruh perangkat, serta menerima data yang berasal dari *node* lain melalui komunikasi LoRa. Aplikasi Flutter berperan sebagai antarmuka pengguna yang terhubung ke ESP32 melalui jaringan WiFi *Access Point* untuk mengirim dan menerima pesan. Pada bagian proses, energi dari baterai dikelola oleh modul TP4056 dan *step-up converter* untuk menghasilkan tegangan yang sesuai dengan kebutuhan sistem. Selanjutnya, ESP32 berfungsi sebagai pusat pengendali yang mengolah data dari aplikasi Flutter dan modul LoRa SX1278 serta mengatur komunikasi melalui jaringan WiFi dan LoRa. Hasil pengolahan tersebut kemudian ditampilkan pada bagian output berupa informasi pada OLED 128×32, indikator LED, dan buzzer sebagai notifikasi ketika pesan diterima, serta data yang dikirimkan ke *node* lain melalui komunikasi LoRa. Melalui integrasi seluruh komponen tersebut, sistem mampu mendukung komunikasi teks secara nirkabel antar *node* tanpa bergantung pada jaringan internet maupun layanan operator seluler.



Gambar 4. *Wiring Diagram*

Gambar 4 menunjukkan wiring diagram sistem komunikasi *offline* berbasis LoRa yang dirancang sebagai panduan dalam proses perakitan perangkat. Wiring diagram ini memberikan informasi mengenai hubungan antar komponen yang digunakan pada sistem, mulai dari sumber daya, perangkat pemrosesan, perangkat input dan output, hingga modul komunikasi yang mendukung pertukaran data antar *node*. Melalui diagram tersebut, proses integrasi setiap komponen dapat dipahami dengan lebih mudah sehingga meminimalkan kesalahan pada tahap implementasi.



Gambar 5. *Realisasi Hardware*

Gambar 5 merupakan realisasi *hardware* yang menggunakan box elektronika x6 yang memiliki ukuran 18,5 cm x 11,5 cm x 6,5 cm yang pada gambar (a) tampak depan terdapat komponen Switch, LED, Buzzer, OLED yang semuanya diletakkan di depan boxnya sedangkan untuk antena diatas. Sedangkan pada gambar (b) tampak dalam terdapat komponen untuk proses yaitu ada Baterai

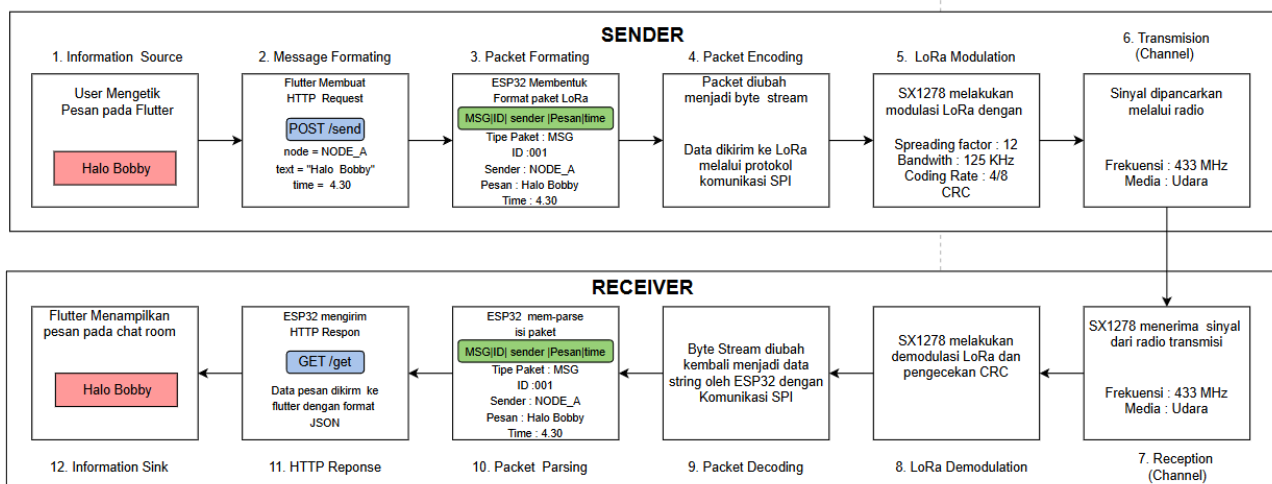
Lithium, Modul TP4056, dan Step-up yang digunakan untuk daya ESP32, selanjutnya pada papan PCB terdapat Lora SX1278 dan ESP32 sebagai komponen utama dari sistem ini.

2.4 Perancangan *Software* Sistem

Perancangan *Software* dilakukan untuk mengatur seluruh proses komunikasi dan pertukaran data pada sistem komunikasi *offline* berbasis LoRa. Perangkat lunak yang dikembangkan terdiri dari program pada mikrokontroler ESP32 dan aplikasi Flutter sebagai antarmuka pengguna. Program pada ESP32 dikembangkan menggunakan Arduino IDE dengan bahasa pemrograman C++, sedangkan aplikasi Flutter dikembangkan menggunakan Visual Studio Code menggunakan bahasa Dart.

2.3.1. Arsitektur Komunikasi Sistem

Arsitektur komunikasi sistem dirancang untuk menggambarkan proses pertukaran data dari pengguna sebagai pengirim (*sender*) hingga diterima oleh pengguna lain sebagai penerima (*receiver*). Sistem memanfaatkan dua jenis komunikasi, yaitu komunikasi HTTP melalui jaringan WiFi Access Point antara aplikasi Flutter dan ESP32, serta komunikasi nirkabel menggunakan teknologi LoRa antar node. ESP32 berperan sebagai penghubung (*gateway*) yang menerjemahkan data dari aplikasi Flutter menjadi paket komunikasi LoRa, kemudian mengubah kembali paket yang diterima menjadi data yang dapat ditampilkan pada aplikasi Flutter.



Gambar 6. Arsitektur Komunikasi Sistem

Berdasarkan Gambar 6, proses komunikasi pada sistem terdiri dari sebelas tahapan yang saling terintegrasi mulai dari proses pembentukan pesan hingga pesan diterima oleh pengguna pada node tujuan. Setiap tahapan memiliki fungsi yang berbeda dalam memastikan data dapat dikirim dan diterima dengan baik melalui komunikasi LoRa. Adapun penjelasan setiap tahapan adalah sebagai berikut.

1. *Information Source*

Tahap pertama merupakan proses pembentukan informasi oleh pengguna. Pada tahap ini pengguna mengetikkan pesan melalui aplikasi Flutter, kemudian pesan tersebut menjadi data awal yang akan diproses oleh sistem sebelum dikirim ke node lain.

2. *Message Formatting*

Setelah pengguna menekan tombol kirim, aplikasi Flutter membentuk HTTP Request menuju ESP32 melalui endpoint */send*. Request tersebut memuat informasi berupa identitas node pengirim, isi pesan, serta waktu pengiriman yang nantinya digunakan sebagai data masukan oleh ESP32.

3. *Packet Formatting*

ESP32 menerima HTTP Request dari aplikasi Flutter kemudian menyusun data menjadi format paket komunikasi LoRa. Paket yang digunakan pada penelitian ini terdiri dari tipe paket (MSG), ID pesan, identitas pengirim, isi pesan, dan waktu pengiriman. Penyusunan format paket bertujuan agar setiap node dapat mengenali struktur data yang diterima sehingga proses komunikasi berlangsung secara konsisten.

4. *Packet Encoding*

Paket komunikasi yang telah disusun selanjutnya diubah menjadi *byte stream* menggunakan library LoRa pada ESP32. Data dalam bentuk *byte* kemudian dikirim menuju modul LoRa SX1278 melalui antarmuka *Serial Peripheral Interface* (SPI) sehingga siap diproses untuk transmisi.

5. *LoRa Modulation*

Modul SX1278 melakukan proses modulasi menggunakan teknik *Chirp Spread Spectrum* (CSS) sesuai parameter komunikasi yang telah ditentukan, yaitu frekuensi 433 MHz, *bandwidth* 125 kHz, *spreading factor* 12, *coding rate* 4/8, serta dilengkapi pemeriksaan CRC (*Cyclic Redundancy Check*) untuk meningkatkan keandalan komunikasi.

6. *Transmission (Channel)*

Setelah proses modulasi selesai, paket data dipancarkan melalui media udara menggunakan frekuensi 433 MHz. Pada tahap ini sinyal radio merambat menuju node tujuan dan kualitas komunikasi dipengaruhi oleh kondisi lingkungan, jarak antar node, serta adanya hambatan yang dapat memengaruhi nilai RSSI dan SNR.

7. Reception (Channel)

Node tujuan menerima sinyal radio melalui modul LoRa SX1278. Modul kemudian menangkap sinyal yang dipancarkan oleh node pengirim sebelum diproses pada tahap demodulasi.

8. LoRa Demodulation

SX1278 melakukan proses demodulasi untuk mengubah sinyal radio menjadi data digital kembali. Pada tahap ini juga dilakukan pemeriksaan CRC untuk memastikan paket yang diterima tidak mengalami kerusakan selama proses transmisi.

9. Packet Decoding

Data dalam bentuk byte stream selanjutnya dikirim dari modul SX1278 menuju ESP32 melalui komunikasi SPI. ESP32 kemudian mengubah byte stream tersebut kembali menjadi data string sehingga dapat diproses lebih lanjut.

10. Packet Parsing

ESP32 melakukan proses parsing terhadap paket komunikasi dengan memisahkan setiap informasi yang terdapat di dalamnya, seperti tipe paket, ID pesan, identitas pengirim, isi pesan, dan waktu pengiriman. Hasil parsing digunakan sebagai dasar dalam menentukan proses selanjutnya, seperti menampilkan pesan, melakukan sinkronisasi riwayat, maupun menjalankan fungsi komunikasi lainnya.

11. HTTP Respon

Setelah proses parsing selesai, ESP32 membentuk HTTP Response yang berisi data hasil pemrosesan paket komunikasi. Data tersebut dikirimkan kepada aplikasi Flutter melalui endpoint /get dalam format JSON. Penggunaan format JSON bertujuan untuk memudahkan aplikasi Flutter dalam membaca dan mengolah data sehingga informasi yang diterima dapat ditampilkan sesuai dengan struktur pesan yang dikirimkan.

12. Information Sink

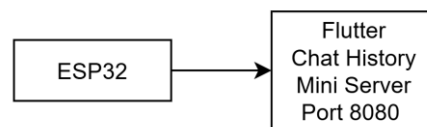
Tahap terakhir merupakan *Information Sink*, yaitu proses ketika aplikasi Flutter menampilkan pesan yang telah diterima pada *chat room*. Selain ditampilkan kepada pengguna, pesan juga disimpan ke dalam riwayat percakapan (*chat history*) sehingga dapat diakses kembali maupun digunakan pada proses sinkronisasi riwayat apabila terdapat node yang baru bergabung ke dalam jaringan. Pada tahap ini proses komunikasi dinyatakan selesai karena informasi yang dikirim oleh node pengirim telah berhasil diterima dan dipahami oleh pengguna pada node tujuan.

2.3.1. Perancangan Mini Server Flutter

Mini server pada aplikasi Flutter digunakan sebagai media pertukaran data antara smartphone dan ESP32. Penggunaan mini server memungkinkan ESP32 mengakses riwayat percakapan yang tersimpan pada aplikasi Flutter melalui jaringan WiFi lokal tanpa memerlukan internet. Perancangan mini server yang digunakan pada sistem meliputi arsitektur, mekanisme kerja, konfigurasi, dan *endpoint* komunikasi.

- Arsitektur Mini Server

Untuk mendukung proses pertukaran data antara aplikasi Flutter dan ESP32, sistem dilengkapi dengan mini server yang berjalan secara lokal pada perangkat pengguna. Mini server ini berfungsi sebagai penyedia layanan data yang dapat diakses oleh ESP32 melalui jaringan WiFi lokal tanpa memerlukan koneksi internet maupun server eksternal. Arsitektur mini server yang digunakan pada sistem ditunjukkan pada Gambar 6.

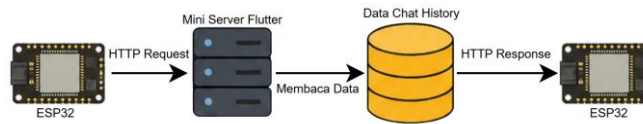


Gambar 7. Diagram Arsitektur Mini Server

Berdasarkan Gambar 7, mini server dijalankan pada aplikasi Flutter dan berfungsi sebagai penyedia data yang tersimpan pada perangkat pengguna. ESP32 bertindak sebagai client yang dapat mengirimkan permintaan data melalui protokol HTTP ke mini server yang berjalan pada port 8080. Dengan mekanisme ini, ESP32 dapat mengakses data yang dibutuhkan tanpa harus menyimpan seluruh informasi pada memori internalnya, sehingga penggunaannya tidak membebani ESP32.

- Mekanisme Kerja Mini Server

Setelah mini server dijalankan pada aplikasi Flutter, ESP32 dapat berkomunikasi secara langsung untuk memperoleh data yang diperlukan. Proses pertukaran data dilakukan menggunakan metode request dan response melalui protokol HTTP. Mekanisme kerja mini server yang diterapkan pada sistem dapat dilihat pada Gambar 7.



Gambar 8. Mekanisme Mini Server

Berdasarkan Gambar 8, proses diawali ketika ESP32 mengirimkan permintaan (*HTTP request*) kepada mini server Flutter. Setelah menerima permintaan tersebut, mini server akan membaca data yang tersimpan pada penyimpanan lokal aplikasi, seperti riwayat percakapan pengguna. Selanjutnya, data yang diminta akan dikirimkan kembali ke ESP32 dalam bentuk *HTTP response*. Mekanisme ini memungkinkan pertukaran data dilakukan secara langsung melalui jaringan lokal tanpa memerlukan layanan cloud maupun akses internet.

- Konfigurasi Mini Server

Konfigurasi diperlukan untuk memastikan mini server pada aplikasi Flutter dapat beroperasi dengan baik dan menerima permintaan data dari ESP32 melalui jaringan WiFi lokal. Parameter yang digunakan meliputi protokol komunikasi, alamat server, dan port yang berfungsi sebagai identitas layanan yang akan diakses oleh ESP32. Konfigurasi mini server yang digunakan pada sistem ditunjukkan pada Tabel 3.

Tabel 1. Konfigurasi Pembentukan Mini Server

Parameter	Nilai
Protocol	HTTP
Addres	0.0.0.0
Port	8080

Berdasarkan Tabel 1, mini server menggunakan protokol HTTP sebagai media komunikasi antara aplikasi Flutter dan ESP32 karena implementasinya sederhana serta mudah diterapkan pada jaringan lokal. Alamat 0.0.0.0 digunakan agar mini server dapat menerima permintaan dari perangkat lain yang berada pada jaringan yang sama. Sementara itu, port 8080 digunakan sebagai jalur komunikasi untuk melayani permintaan data yang dikirim oleh ESP32. Dengan konfigurasi tersebut, ESP32 dapat mengakses layanan yang disediakan oleh mini server secara langsung tanpa memerlukan koneksi internet maupun server eksternal.

- **Endpoint Mini Server**

Untuk menyediakan akses data kepada ESP32, mini server menyediakan *endpoint* yang dapat dipanggil melalui protokol HTTP. *Endpoint* yang digunakan ditunjukkan pada Tabel 2.

Tabel 2. *Endpoint History*

Endpoint	Fungsi
<i>/history</i>	Menyediakan data riwayat percakapan

Endpoint /history digunakan untuk mengakses data riwayat percakapan yang tersimpan pada aplikasi Flutter. Ketika *endpoint* ini dipanggil oleh ESP32, mini server akan membaca seluruh data *history* yang tersimpan kemudian mengirimkannya kembali sebagai respons HTTP. Data tersebut selanjutnya dapat diproses oleh ESP32 sesuai kebutuhan sistem.

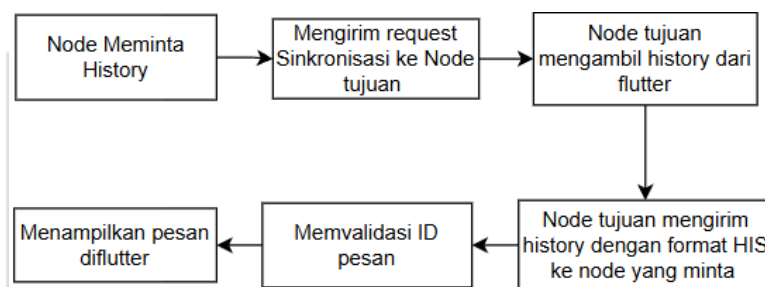
- **Peran Smartphone Dalam Sistem**

Smartphone pada penelitian ini tidak hanya berfungsi sebagai antarmuka pengguna (*user interface*), tetapi juga sebagai media penyimpanan riwayat pesan serta menjalankan mini server yang menyediakan layanan pertukaran data dengan ESP32 melalui jaringan WiFi lokal. Pendekatan ini dipilih karena smartphone memiliki kapasitas penyimpanan yang lebih besar dibandingkan memori internal ESP32, sehingga penyimpanan riwayat percakapan dapat dilakukan tanpa membebani sumber daya mikrokontroler. Selain itu, penggunaan smartphone memungkinkan pengguna mengakses, membaca, dan mengelola riwayat percakapan secara langsung melalui aplikasi Flutter.

Dengan arsitektur tersebut, proses sinkronisasi riwayat percakapan bergantung pada ketersediaan smartphone yang menjalankan aplikasi Flutter. Apabila smartphone dimatikan, aplikasi dihentikan, atau tidak terhubung ke jaringan WiFi lokal ESP32, maka mini server tidak dapat memberikan layanan data sehingga proses sinkronisasi riwayat tidak dapat dilakukan. Tetapi, komunikasi pesan antar node melalui LoRa tetap dapat berlangsung karena proses pengiriman pesan dilakukan secara langsung oleh ESP32 dan modul LoRa tanpa bergantung pada mini server. Setelah smartphone kembali aktif dan terhubung ke jaringan, proses sinkronisasi riwayat dapat dilakukan kembali sehingga data percakapan pada setiap node tetap konsisten.

2.3.2. Perancangan Sinkronisasi Riwayat Chat

Pada sistem komunikasi berbasis LoRa, setiap *node* tidak selalu berada dalam kondisi aktif secara bersamaan. Dalam kondisi tertentu, suatu *node* dapat mengalami gangguan komunikasi, berada di luar jangkauan jaringan, atau kembali aktif setelah perangkat dimatikan. Kondisi tersebut menyebabkan pesan yang dikirim selama periode tertentu tidak dapat diterima secara langsung sehingga riwayat percakapan yang dimiliki setiap *node* menjadi berbeda. Perbedaan data percakapan dapat mengurangi kelengkapan informasi yang diterima pengguna, terutama pada kondisi komunikasi darurat yang memerlukan penyampaian informasi secara berkelanjutan. Oleh karena itu, diperlukan mekanisme sinkronisasi riwayat *chat* untuk memastikan setiap *node* dapat memperoleh kembali pesan yang belum diterima sebelumnya sehingga konsistensi data percakapan antar *node* tetap terjaga.

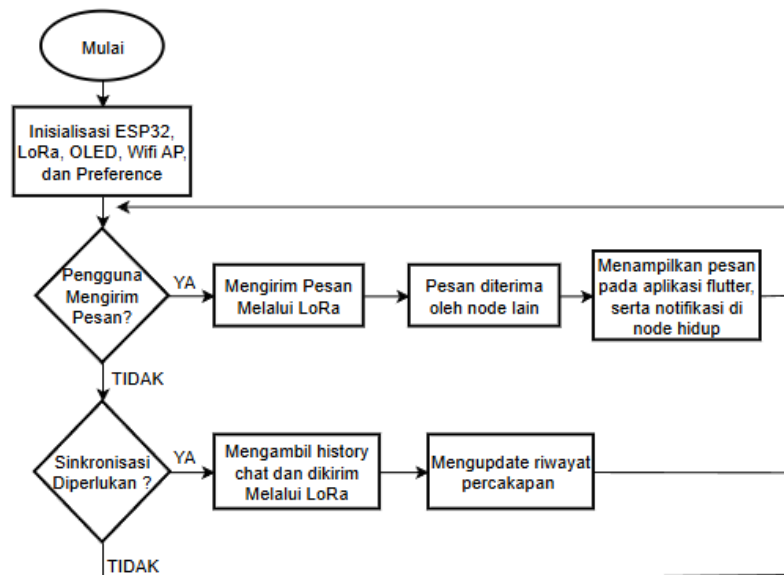


Gambar 9. Diagram Mekanisme Sinkronisasi

Pada gambar 9 proses sinkronisasi diawali ketika *node* yang membutuhkan riwayat percakapan mengirimkan permintaan sinkronisasi kepada *node* tujuan. Setelah menerima permintaan tersebut, *node* tujuan mengakses data *history* yang tersimpan pada aplikasi Flutter melalui mini server. Data riwayat percakapan kemudian dikirimkan kembali menggunakan format paket HIS melalui jaringan LoRa. Pada sisi penerima, setiap pesan yang diterima akan divalidasi berdasarkan ID pesan untuk mencegah penyimpanan data yang sama secara berulang. Apabila ID pesan belum tersedia, data akan disimpan dan ditampilkan pada aplikasi Flutter. Proses ini berlangsung hingga seluruh data riwayat percakapan berhasil diterima oleh *node* yang melakukan sinkronisasi.

2.5 Flowchart Sistem

Flowchart sistem digunakan untuk menggambarkan alur kerja keseluruhan sistem komunikasi berbasis LoRa yang dikembangkan pada penelitian ini. Diagram tersebut menunjukkan proses yang terjadi mulai dari inisialisasi perangkat, pengiriman dan penerimaan pesan antar *node*, hingga mekanisme sinkronisasi riwayat percakapan.



Gambar 10. Flowchart Sistem

Pada gambar 10 proses sistem diawali dengan inisialisasi seluruh komponen yang digunakan, meliputi ESP32, modul LoRa, OLED, Wi-Fi *Access Point*, dan parameter konfigurasi yang tersimpan pada memori perangkat. Setelah proses inisialisasi selesai, sistem memasuki kondisi siaga untuk menunggu aktivitas komunikasi dari pengguna.

Apabila pengguna mengirimkan pesan melalui aplikasi Flutter, data pesan akan diteruskan ke ESP32 dan dikirimkan melalui jaringan LoRa menuju *node* tujuan. Selanjutnya, pesan yang diterima oleh *node* lain akan ditampilkan pada aplikasi Flutter dan disertai dengan notifikasi sebagai penanda adanya pesan baru. Setelah proses tersebut selesai, sistem kembali ke kondisi siaga untuk menunggu aktivitas berikutnya.

Selain melayani pengiriman pesan, sistem juga mendukung mekanisme sinkronisasi riwayat percakapan. Ketika sinkronisasi diperlukan, *node* akan mengambil data riwayat *chat* yang tersedia dan mengirimkannya melalui jaringan LoRa kepada *node* yang membutuhkan. Data yang diterima kemudian digunakan untuk memperbarui riwayat percakapan sehingga informasi yang dimiliki oleh setiap *node* tetap konsisten. Setelah proses sinkronisasi selesai, sistem kembali ke kondisi siaga dan siap melayani proses komunikasi berikutnya.

3. HASIL DAN PEMBAHASAN

3.1. Pengujian Fungsionalitas Komponen

Pengujian fungsionalitas komponen dilakukan untuk memastikan setiap perangkat keras yang digunakan pada sistem dapat beroperasi sesuai dengan fungsi yang telah dirancang. Pengujian dilakukan secara individual pada masing-masing komponen dengan

mengamati respon dan keluaran yang dihasilkan. Hasil pengujian fungsionalitas komponen ditunjukkan pada Tabel 3.

Tabel 3. Fungsionalitas Komponen

Komponen	Kondisi	Keterangan
ESP32	Normal	Menjalankan fungsi kontrol sistem
LoRa SX1278	Normal	Mengirim pesan <i>kenode</i> lain
LED notifikasi	Normal	Hidup ketika ada pesan masuk
Buzzer	Normal	Memberikan notifikasi suara saat pesan diterima
Modul TP4056	Normal	Mengisi baterai dengan baik
Step-up MT3608	Normal	Menaikkan tegangan untuk ESP32
Baterai	Normal	Menyuplai daya untuk ESP32
OLED Display	Normal	Menampilkan pesan yang dikirim
Switch	Normal	Menghidupkan sistem

Pada tabel 3 semua komponen berfungsi dengan normal LoRa dapat mengirim pesan dengan normal walaupun memiliki keterbatasan jarak, kemudian untuk komponen notifikasi juga berfungsi dengan normal, serta komponen untuk catu daya juga berfungsi dengan baik.

3.2. Lokasi Pengujian

Pengujian dilakukan di jalan yang bersampingan dengan perumahan *Lor inn Residence* dan kampus 4 kedokteran Universitas Muhammadiyah Surakarta. Untuk detail lokasinya bisa dilihat pada gambar 11.



Gambar 11. Lokasi Pengujian

Pada gambar 11 merupakan lokasi pengujian yang memiliki panjang jalur sekitar 200 m meter yang digunakan sebagai jalur pengukuran performa komunikasi LoRa. Lokasi ini dipilih karena memiliki kondisi *Line of Sight* (LOS) dengan hambatan yang relatif minim, sehingga sesuai untuk mengevaluasi kemampuan sistem dalam mengirim dan menerima pesan pada berbagai variasi jarak.

3.3. Pengujian Skenario Riwayat Pesan

Pengujian sinkronisasi riwayat chat dilakukan dengan melibatkan tiga pengguna, yaitu Ayu (Node A), Bobby (Node B), dan Chandra (Node C). Pada awal pengujian hanya Ayu yang aktif, sedangkan Bobby dan Chandra berada dalam kondisi *offline*. Ayu kemudian mengirimkan beberapa pesan ke dalam *chat room*. Setelah beberapa saat, Bobby diaktifkan dan secara otomatis melakukan sinkronisasi terhadap seluruh riwayat percakapan yang sebelumnya dikirim oleh Ayu. Selanjutnya Ayu dan Bobby melanjutkan percakapan. Setelah beberapa pesan terkirim, Chandra *online* dan kembali melakukan sinkronisasi sehingga seluruh riwayat percakapan dari Ayu dan Bobby dapat diterima sebelum ikut bergabung dalam percakapan. Skenario percakapan ditunjukkan pada Tabel 4.

Tabel 4. Skenario percakapan

NO	Waktu	Pengguna	Kondisi	Pesan
1	4.36	Ayu	Bobby & Chandra <i>Offline</i>	Halo semuanya, apakah ada yang online?
2	4.36	Ayu	Bobby & Chandra <i>Offline</i>	Saya akan mengirim beberapa informasi penting.
3	4.37	Ayu	Bobby & Chandra <i>Offline</i>	Lokasi pengamatan berada di Pos 1.
4	4.37	Ayu	Bobby & Chandra <i>Offline</i>	Mohon konfirmasi jika sudah bergabung
5	4.40	Bobby	Bobby & Ayu <i>Online</i>	Halo Ayu, saya baru bergabung. Semua pesan sudah saya terima
6	4.40	Ayu	Bobby & Ayu <i>Online</i>	Baik Bobby, silakan lanjutkan pemantauan
7	4.41	Bobby	Bobby & Ayu <i>Online</i>	Siap, kondisi di Pos 1 masih aman
8	4.42	Ayu	Bobby & Ayu <i>Online</i>	Baik, tetap pantau perkembangan di lokasi

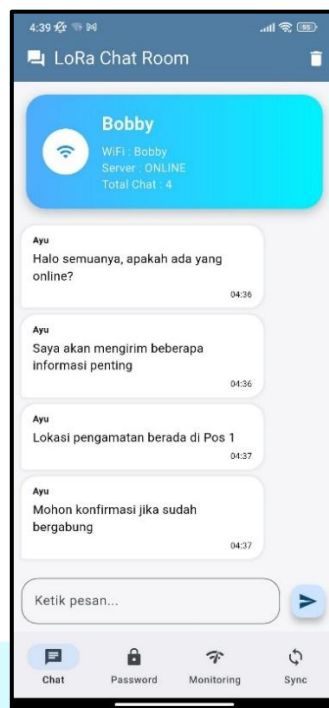
9	4.45	Chandra	Semua <i>Online</i>	Halo semuanya, saya baru masuk dan seluruh riwayat chat sudah diterima
10	4.46	Ayu	Semua <i>Online</i>	Selamat datang Chandra
11	4.47	Bobby	Semua <i>Online</i>	Kondisi komunikasi berjalan normal
12	4.48	Chandra	Semua <i>Online</i>	Baik, saya sudah menerima seluruh informasi sebelumnya
13	4.48	Ayu	Semua <i>Online</i>	Selanjutnya kita gunakan grup ini untuk koordinasi
14	4.48	Bobby	Semua <i>Online</i>	Siap
15	4.49	Chandra	Semua <i>Online</i>	Siap, komunikasi berjalan dengan baik

- Tampilan Chat Room Flutter



Gambar 12. Tampilan Chat Ayu Sebelum yang Lain Bergabung

Pada gambar 12 menunjukkan kondisi awal ketika hanya Ayu yang aktif. Pada tahap ini Bobby dan Chandra masih berada dalam keadaan *offline*, sehingga seluruh pesan yang dikirim oleh Ayu hanya tersimpan pada aplikasi Flutter milik Ayu.



Gambar 13. Tampilan sinkronisasi pada chat Bobby

Pada gambar 13 menunjukkan pesan Ayu yang sudah disinkronisasi pada aplikasi flutter milik Bobby yang terdapat 4 pesan dari Ayu beserta waktu pengirimannya.



Gambar 14. Tampilan Ayu dan Bobby *Chatting* bersama

Pada gambar 14 setelah proses sinkronisasi selesai, Ayu dan Bobby dapat melanjutkan percakapan seperti biasa. Hal ini menunjukkan bahwa mekanisme sinkronisasi tidak mengganggu proses komunikasi yang sedang berlangsung.



Gambar 15. Tampilan sinkronisasi pada chat Chandra

Pada gambar 15 menunjukkan pesan Ayu dan Bobby yang sudah disinkronisasi pada aplikasi flutter milik Chandra yang terdapat 8 pesan dari Ayu dan Bobby beserta waktu pengirimannya.



Gambar 16. Tampilan Ayu, Bobby dan Chandra *chatting* bersama

Pada gambar 16 menunjukkan kondisi akhir setelah proses sinkronisasi selesai. Seluruh node telah memiliki riwayat percakapan yang sama dan dapat melanjutkan komunikasi secara normal di dalam *chat room*.

3.4. Pengujian Pengiriman Pesan Terhadap Jarak

Pengujian pengiriman pesan dilakukan dengan membawa salah satu node ke setiap jarak pengujian yang telah ditentukan, kemudian node tersebut menerima pesan dari salah satu node lainnya yang berada pada posisi tetap untuk mengetahui keberhasilan pengiriman pesan. Selain mengevaluasi status keberhasilan pengiriman pesan, pada setiap pengujian juga dilakukan pencatatan nilai Received Signal Strength Indicator (RSSI) dan Signal-to-Noise Ratio (SNR) sebagai indikator kualitas sinyal yang diterima. Setiap pengujian dilakukan dengan mengirimkan 10 pesan disetiap jaraknya sehingga diperoleh beberapa data pengukuran. Selanjutnya, data tersebut dianalisis menggunakan statistik deskriptif berupa nilai rata-rata (mean) dan standar deviasi. Untuk mengukur nilai mean dan standar deviasi digunakan persamaan:

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (1)$$

$$S = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}} \quad (2)$$

Dengan rumus persamaan 1 dan 2 didapatkan hasil pengujian yang tertampil pada tabel dibawah

Tabel 5. Pengujian pengiriman pesan dari *Node B* ke *Node A*

Jarak (m)	Kondisi	Status Pesan	Mean RSSI (dBm)	SD RSSI (dBm)	Mean SNR (dB)	SD SNR (dB)
25	LOS	Terkirim	-80	1.15	1.9	0.21
50	LOS	Terkirim	-83	1.15	-1.5	0.28
75	LOS	Terkirim	-85	1.05	-3.5	0.28
100	LOS	Terkirim	-86	1.25	-6.8	0.27
125	LOS	Terkirim	-97	1.49	-8.9	0.26
150	LOS	Terkirim	-115	1.70	-13.2	0.27
175	LOS	Terputus	-125	1.56	-16.5	0.27

200	LOS	Terputus	-131	1.63	-18.0	0.28
-----	-----	----------	------	------	-------	------

Tabel 6. Pengujian pengiriman pesan dari *Node C* ke *Node B*

Jarak (m)	Kondisi	Status Pesan	Mean RSSI (dBm)	SD RSSI (dBm)	Mean SNR (dB)	SD SNR (dB)
25	LOS	Terkirim	-79	1.03	2.1	0.25
50	LOS	Terkirim	-82	1.05	-1.2	0.20
75	LOS	Terkirim	-84	1.05	-3.2	0.19
100	LOS	Terkirim	-88	1.05	-6.4	0.28
125	LOS	Terkirim	-95	1.05	-8.3	0.19
150	LOS	Terkirim	-105	1.25	-12.8	0.19
175	LOS	Terputus	-125	1.15	-16.3	0.21
200	LOS	Terputus	-131	1.15	-17.5	0.19

Tabel 7. Pengujian pengiriman pesan dari *Node A* ke *Node C*

Jarak (m)	Kondisi	Status Pesan	Mean RSSI (dBm)	SD RSSI (dBm)	Mean SNR (dB)	SD SNR (dB)
25	LOS	Terkirim	-81	1.25	2.4	0.20
50	LOS	Terkirim	-84	1.41	-0.9	0.23
75	LOS	Terkirim	-87	1.49	-2.8	0.23
100	LOS	Terkirim	-89	1.25	-5.9	0.24
125	LOS	Terkirim	-98	1.49	-8.1	0.22
150	LOS	Terkirim	-107	1.49	-12.1	0.22
175	LOS	Terputus	-125	1.49	-15.8	0.22
200	LOS	Terputus	-131	1.49	-17.2	0.25

Berdasarkan hasil pengujian pengiriman pesan pada *Node A*, *Node B*, dan *Node C*, kualitas komunikasi LoRa mengalami penurunan seiring bertambahnya jarak antar node. Pada ketiga skenario pengujian, seluruh pesan berhasil dikirim hingga jarak 150 meter, sedangkan pada jarak 175 meter dan 200 meter komunikasi terputus. Nilai rata-rata RSSI

menunjukkan tren menurun dari sekitar -79 dBm hingga -81 dBm pada jarak 25 meter menjadi sekitar -105 dBm hingga -115 dBm pada jarak 150 meter. Demikian pula, nilai SNR menurun dari sekitar 2 dB menjadi sekitar -12 dB, yang menunjukkan kualitas sinyal semakin berkurang seiring bertambahnya jarak. Standar deviasi RSSI yang berada pada kisaran 1,03–1,70 dBm dan standar deviasi SNR pada kisaran 0,19–0,28 dB menunjukkan bahwa hasil pengukuran cukup konsisten dengan variasi yang rendah. Perbedaan nilai RSSI dan SNR antar skenario juga relatif kecil karena seluruh pengujian menggunakan modul LoRa, konfigurasi, dan kondisi Line of Sight (LOS) yang sama. Secara keseluruhan, hasil pengujian menunjukkan bahwa sistem komunikasi LoRa yang dikembangkan mampu melakukan pengiriman pesan dengan baik jarak hingga 150 meter pada kondisi LOS, sedangkan pada jarak yang lebih jauh kualitas sinyal menurun sehingga komunikasi terputus.

3.5. Keterbatasan Alat

Sistem komunikasi offline berbasis LoRa yang dikembangkan telah berhasil mendukung proses pengiriman pesan dan sinkronisasi riwayat percakapan sesuai dengan perancangan. Namun, selama proses implementasi dan pengujian masih ditemukan beberapa keterbatasan yang dapat menjadi bahan evaluasi sekaligus peluang pengembangan pada penelitian berikutnya, di antaranya sebagai berikut:

1. Keterbatasan jangkauan komunikasi. Kinerja sistem sangat dipengaruhi oleh jarak dan kondisi lingkungan. Pada pengujian dengan kondisi Line of Sight (LOS), komunikasi masih berjalan dengan baik hingga jarak tertentu, sedangkan pada jarak yang lebih jauh kualitas sinyal menurun sehingga menyebabkan kegagalan pengiriman maupun sinkronisasi pesan.
2. Sinkronisasi riwayat dilakukan secara bertahap. Proses sinkronisasi mengirimkan riwayat percakapan dalam beberapa paket data secara berurutan. Semakin banyak riwayat yang harus dikirim, semakin lama waktu yang dibutuhkan dan semakin besar kemungkinan terjadinya kehilangan paket ketika kualitas sinyal menurun.
3. Kapasitas penyimpanan bergantung pada perangkat smartphone. Riwayat percakapan disimpan secara lokal pada aplikasi Flutter sehingga ketersediaan data bergantung pada kondisi penyimpanan perangkat pengguna. Apabila data aplikasi terhapus atau perangkat mengalami kerusakan, riwayat percakapan tidak dapat dipulihkan karena sistem tidak menggunakan penyimpanan berbasis cloud.

4. PENUTUP

4.1. Kesimpulan

Rancang bangun sistem komunikasi offline berbasis LoRa sebagai solusi di wilayah minim sinyal telah berhasil direalisasikan sesuai dengan tujuan penelitian. Sistem mampu mendukung proses pengiriman dan penerimaan pesan teks antar pengguna tanpa bergantung pada jaringan internet maupun jaringan seluler. Integrasi ESP32 sebagai pengendali utama, modul LoRa SX1278 sebagai media komunikasi, serta aplikasi Flutter sebagai antarmuka pengguna memungkinkan komunikasi berlangsung secara lokal melalui jaringan WiFi *Access Point* dan LoRa.

Berdasarkan hasil pengujian, sistem menunjukkan performa komunikasi yang baik pada kondisi *Line of Sight* (LOS). Pengiriman pesan berhasil dilakukan hingga jarak 150 meter, sedangkan pada jarak 175 hingga 200 meter komunikasi tidak dapat dilakukan karena penurunan kualitas sinyal yang ditunjukkan oleh nilai RSSI dan SNR.

Berdasarkan skenario pengujian sinkronisasi yang melibatkan tiga pengguna, yaitu Ayu (Node A), Bobby (Node B), dan Chandra (Node C), sistem mampu melakukan sinkronisasi riwayat percakapan secara otomatis ketika pengguna baru bergabung ke dalam *chat room*. Node B berhasil memperoleh seluruh riwayat percakapan yang sebelumnya dikirim oleh Node A, sedangkan Node C berhasil memperoleh riwayat percakapan dari Node A dan Node B sebelum melanjutkan komunikasi bersama. Hasil tersebut menunjukkan bahwa mekanisme sinkronisasi yang diterapkan mampu menjaga konsistensi riwayat percakapan antar node sehingga pengguna tetap memperoleh informasi yang telah dikirim sebelumnya meskipun sempat berada dalam kondisi *offline*.

Secara keseluruhan, sistem yang dikembangkan mampu menjadi alternatif komunikasi pada wilayah dengan keterbatasan atau tidak tersedianya jaringan seluler maupun internet. Dengan adanya fitur pengiriman pesan dan sinkronisasi riwayat chat, sistem dapat membantu menjaga kontinuitas informasi antar pengguna sehingga berpotensi dimanfaatkan sebagai media komunikasi pada kondisi darurat maupun di wilayah minim sinyal.

4.2. Saran

Berdasarkan hasil penelitian yang telah dilakukan, sistem komunikasi offline berbasis LoRa masih memiliki beberapa aspek yang dapat dikembangkan pada penelitian selanjutnya. Untuk meningkatkan jangkauan komunikasi, disarankan menggunakan modul LoRa yang telah dilengkapi dengan *Power Amplifier* (PA) dan *Low Noise Amplifier* (LNA) seperti modul LoRa dengan seri AB-IOT-433 yang dapat memperkuat sinyal LoRa yang tadinya 20dBm menjadi 43dBm. Selain itu, penggunaan antena dengan spesifikasi yang lebih baik serta

optimasi parameter komunikasi LoRa, seperti *Spreading Factor*, *Bandwidth*, dan *Coding Rate*, diharapkan dapat meningkatkan kualitas sinyal dan memperluas jangkauan komunikasi sistem.

Mekanisme sinkronisasi riwayat chat juga dapat dikembangkan agar lebih efisien dengan menerapkan metode sinkronisasi berdasarkan pesan yang belum diterima serta mekanisme pengiriman ulang terhadap paket yang gagal diterima. Pengembangan tersebut diharapkan dapat mengurangi waktu sinkronisasi sekaligus meningkatkan keberhasilan pengiriman riwayat percakapan, terutama ketika kualitas sinyal mulai menurun.

Selain itu, untuk mengurangi risiko kehilangan riwayat percakapan akibat kerusakan perangkat atau penghapusan data aplikasi, sistem dapat dilengkapi dengan mekanisme pencadangan (*backup*) data. Proses pencadangan dapat dilakukan secara otomatis ketika perangkat memperoleh koneksi internet, sehingga sistem tetap dapat beroperasi secara offline saat dibutuhkan, namun riwayat percakapan tetap tersimpan dan dapat dipulihkan apabila terjadi kehilangan data.

PERSANTUNAN

Alhamdulillahirabbil'alamîn, segala puji dan syukur penulis panjatkan ke hadirat Allah SWT atas segala rahmat, karunia, serta kemudahan yang telah diberikan sehingga artikel ilmiah ini dapat diselesaikan dengan baik. Penulisan dan penyelesaian penelitian ini tidak terlepas dari bantuan, dukungan, serta bimbingan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Orang tua dan keluarga tercinta yang senantiasa memberikan doa, dukungan moral, motivasi, serta semangat kepada penulis selama menempuh pendidikan dan menyelesaikan penelitian ini.
2. Bapak Fajar Suryawan, S.T., M.Eng.Sc., Ph.D. selaku dosen pembimbing yang telah meluangkan waktu, memberikan arahan, masukan, serta bimbingan yang sangat berharga selama proses penelitian hingga penyusunan artikel ilmiah ini.
3. Seluruh dosen dan tenaga kependidikan Program Studi Teknik Elektro Universitas Muhammadiyah Surakarta yang telah memberikan ilmu pengetahuan, pengalaman, dan dukungan selama masa perkuliahan.
4. Seluruh mahasiswa Teknik Elektro angkatan 2022 yang telah menjadi teman belajar, berbagi pengalaman, serta memberikan dukungan selama proses perkuliahan hingga penyelesaian karya ilmiah ini.

Penulis menyadari bahwa penelitian ini masih memiliki berbagai keterbatasan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan demi pengembangan penelitian di masa yang akan datang. Semoga segala bantuan, dukungan, dan kebaikan yang telah diberikan mendapatkan balasan terbaik dari Allah SWT serta menjadi amal yang bermanfaat.

DAFTAR PUSTAKA

- Badan Nasional Penanggulangan Bencana. (2025). *Jaringan Komunikasi Bantu Pulihkan Kondisi Darurat Sumbar*. Badan Nasional Penanggulangan Bencana. <https://bnpb.go.id/berita/jaringan-komunikasi-bantu-pulihkan-kondisi-darurat-sumbar>
- Adelantado, F., Vilajosana, X., Tuset-Peiro, P., Martinez, B., Melia-Segui, J., & Watteyne, T. (2017). Understanding the Limits of LoRaWAN. *IEEE Communications Magazine*, 55(9), 34–40. <https://doi.org/10.1109/MCOM.2017.1600613>
- Augustin, A., Yi, J., Clausen, T., & Townsley, W. M. (2016). A study of Lora: Long range & low power networks for the internet of things. *Sensors (Switzerland)*, 16(9), 1–18. <https://doi.org/10.3390/s16091466>
- Badan Nasional Penanggulangan Bencana. (2025). *Jaringan Komunikasi Bantu Pulihkan Kondisi Darurat Sumbar*. Badan Nasional Penanggulangan Bencana. <https://bnpb.go.id/berita/jaringan-komunikasi-bantu-pulihkan-kondisi-darurat-sumbar>
- Fadillah, A. N., Almazazi, A. D., & Pardede, M. (2022). Rancang Bangun Alat Komunikasi Antar Smartphone Melalui Jaringan Nirkabel Lora Multi-Hop. *Prosiding Konferensi Nasional Social & Engineering Polmed (Konsep)*, 3(1), 908–916. <https://ojs.polmed.ac.id/index.php/KONSEP2021/article/view/927/>
- Farras Fauzan, M., Shubhi Maulana, M., Lintar Balle, J., Febriyanti, T., Ronald Suhada, V., Alif Falah, N., Ardelia Wirastuti, M., Fakhiratunisa, N., Renaissance Al-ars, K., Rifa Kusumah, B., & Siskandar, R. (2021). Alat Komunikasi Darurat dengan ESP8266 dan LoRa untuk Pendaki Gunung Emergency. *Indonesian Journal of Science*, 2(2), 52–60. <http://journal.pusatsains.com/index.php/jsi>
- Februari, V. N., Sunaryo, D. A., Anshori, M., & Widya, A. (2026). *Design And Implementation of a Short Message Delivery System Using LoRa Technology*. 5(3).
- paul S, B. (2024). LoRa Based Emergency Communication System. *International Scientific Journal of Engineering and Management*, 03(04), 1–9. <https://doi.org/10.55041/isjem01628>
- Ramadhani, E. H., Wulandari, A., & Hikmaturokhman, A. (2025). Perancangan Sistem Keamanan Komunikasi Data pada Jaringan LoRa Menggunakan Algoritme PRESENT. *Jurnal Telematika*, 19(2), 72–81. <https://journal.ithb.ac.id/index.php/telematika/article/view/674>