



**ANALISIS MEKANISME *FIRMWARE OVER-THE-AIR*  
PADA JARINGAN SENSOR NIRKABEL  
MENGUNAKAN ARSITEKTUR *GATEWAY*  
BLUETOOTH LOW ENERGY**

**SKRIPSI**

**RAFI RASHYA HUSEIN PANJAITAN**

**2207421038**

**PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN  
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER  
POLITEKNIK NEGERI JAKARTA**

**2026**



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**ANALISIS MEKANISME *FIRMWARE OVER-THE-AIR*  
PADA JARINGAN SENSOR NIRKABEL  
MENGUNAKAN ARSITEKTUR *GATEWAY*  
BLUETOOTH LOW ENERGY**

**SKRIPSI**

**Dibuat Untuk Melengkapi Syarat Syarat yang Diperlukan Untuk Memperoleh  
Diploma Empat Politeknik**

**RAFI RASHYA HUSEIN PANJAITAN**

**2207421038**

**POLITEKNIK  
NEGERI  
JAKARTA**

**PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN  
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER  
POLITEKNIK NEGERI JAKARTA**

**2026**



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini:

Nama : Rafi Rashya Husein Panjaitan  
NIM : 2207421038  
Jurusan/Program Studi : Teknik Informatika dan Komputer / Teknik Multimedia dan Jaringan  
Judul skripsi : Analisis Mekanisme Firmware Over-the-Air pada Jaringan Sensor Nirkabel Menggunakan Arsitektur Gateway Bluetooth Low Energy

Menyatakan dengan sebenarnya bahwa skripsi ini benar-benar merupakan hasil karya saya sendiri, bebas dari peniruan terhadap karya dari orang lain. Kutipan pendapat dan tulisan orang lain ditunjuk sesuai dengan cara-cara penulisan karya ilmiah yang berlaku.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa dalam skripsi ini terkandung ciri-ciri plagiat dan bentuk-bentuk peniruan lain yang dianggap melanggar peraturan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Depok, 16 Juni 2026

Yang membuat pernyataan

(Rafi Rashya Husein Panjaitan)

NIM 2207421038



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LEMBAR PENGESAHAN

Skripsi diajukan oleh :

Nama : Rafi Rashya Husein Panjaitan  
NIM : 2207421038  
Program Studi : Teknik Multimedia dan Jaringan  
Judul Skripsi : Analisis Mekanisme Firmware Over-the-Air pada Jaringan Sensor Nirkabel Menggunakan Arsitektur Gateway Bluetooth Low Energy

Telah diuji oleh tim penguji dalam Sidang Skripsi pada hari Rabu, Tanggal 1, Bulan Juli, Tahun 2026 dan dinyatakan LULUS.

Disahkan oleh

Pembimbing I : Dr. Prihatin Oktivasari, S.Si., M.Si.

Penguji I : Ayu Rosyida Zain, S.ST, M.T

Penguji II : Ariawan Andi Suhandana, S.Kom., M.T.I.

Penguji III : Chandra Wirawan, S.Kom, M.Kom.

Mengetahui :

Jurusan Teknik Informatika dan Komputer

Ketua



Dr. Anisa Hidayati, S.Kom., M.Kom.

NIP. 197908032003122003



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Allah SWT, karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan skripsi yang berjudul "Analisis Mekanisme Firmware Over-the-Air pada Jaringan Sensor Nirkabel Menggunakan Arsitektur Gateway Bluetooth Low Energy" yang disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Terapan Komputer di Politeknik Negeri Jakarta. Penyusunan skripsi ini tidak lepas dari bimbingan, dukungan, dan bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Allah SWT atas kemudahan, petunjuk serta karunia-Nya sehingga penulis dapat menyelesaikan penelitian.
2. Ibu Dr. Prihatin Oktivasari, S.Si., M.Si selaku Dosen Pembimbing yang telah memberikan banyak arahan, masukan, dan bimbingan yang sangat berharga selama proses penyusunan skripsi ini.
3. Orang tua dan keluarga tercinta yang senantiasa memberikan doa, restu, serta dukungan moral dan material kepada penulis.
4. Teman-teman mahasiswa Program Studi Teknik Multimedia dan Jaringan angkatan 2022 yang telah membantu dalam penyelesaian skripsi ini yang tidak dapat disebutkan satu per satu.
5. Zahra Aulia Choerunnisa yang senantiasa mendampingi dengan penuh kesabaran, memberikan semangat, kebahagiaan, serta dukungan emosional yang tiada henti di setiap tahapan penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan. Oleh karena itu, kritik dan saran yang membangun sangat penulis harapkan demi perbaikan di masa mendatang. Semoga skripsi ini dapat memberikan manfaat bagi perkembangan ilmu pengetahuan, khususnya di bidang teknologi informasi dan komunikasi.

Depok, 25 Juni 2026

Rafi Rashya Husein Panjaitan



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Politeknik Negeri Jakarta, saya bertanda tangan di bawah ini:

Nama : Rafi Rashya Husein Panjaitan  
NIM : 2207421038  
Jurusan/Program Studi : Teknik Informatika dan Komputer /  
Teknik Multimedia dan Jaringan

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Politeknik Negeri Jakarta Hak Bebas Royalti Non-Eksklusif atas karya ilmiah saya yang berjudul:

### ANALISIS MEKANISME FIRMWARE OVER-THE-AIR PADA JARINGAN SENSOR NIRKABEL MENGGUNAKAN ARSITEKTUR GATEWAY BLUETOOTH LOW ENERGY

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksklusif ini Politeknik Negeri Jakarta berhak menyimpan, mengalihmediakan/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan skripsi saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai Penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Depok, 21 Juli 2026

Yang menyatakan,



(Rafi Rashya Husein Panjaitan)

NIM.2207421038



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## **Analisis Mekanisme *Firmware Over-the-Air* pada Jaringan Sensor Nirkabel Menggunakan Arsitektur *Gateway Bluetooth Low Energy***

### **ABSTRAK**

*Pengelolaan pemeliharaan Jaringan Sensor Nirkabel secara manual tidak efisien, sehingga pembaruan firmware nirkabel sangat dibutuhkan. Penelitian ini bertujuan merancang, mengimplementasikan, dan menganalisis kinerja transmisi data FOTA berbasis arsitektur gateway Bluetooth Low Energy 5.0 pada WSN dibandingkan protokol ESP-NOW sebagai pembanding. Metode penelitian adalah rekayasa eksperimental menggunakan gateway ESP32-S3 dan node ESP32-C3 dengan variasi jarak 1 hingga 15 meter, hambatan LOS dan NLOS, serta ukuran berkas. Hasil pengujian menunjukkan FOTA BLE sukses 100% pada jarak 1-5 meter. Namun, BLE sensitif terhadap hambatan beton, di mana pada 10m NLOS, rasio retransmisi BLE melonjak tajam hingga 1432,19% pada berkas kecil dan 869,63% pada berkas sedang, sedangkan ESP-NOW tetap stabil di kisaran 100-102%. Dari segi kecepatan, ESP-NOW mengungguli BLE dengan throughput rata-rata mencapai 165,19 Kbps pada berkas kecil dan 185,93 Kbps sedang pada 1m LOS, berbanding BLE yang hanya mencapai 138,72 Kbps dan 105,84 Kbps. Pada 15m NLOS, kedua protokol gagal transmisi dengan Success Rate 0%. Disimpulkan bahwa meskipun FOTA BLE stabil pada jarak dekat, protokol ini memiliki keterbatasan keandalan dan throughput yang signifikan pada kondisi berpenghalang dibandingkan ESP-NOW.*

**Kata Kunci:** *Bluetooth Low Energy, ESP-NOW, ESP32, Firmware Over-the-Air, Jaringan Sensor Nirkabel, Retransmisi, Throughput.*



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

**DAFTAR ISI**

|                                                     |                              |
|-----------------------------------------------------|------------------------------|
| SURAT PERNYATAAN BEBAS PLAGIARISME ..               | Error! Bookmark not defined. |
| LEMBAR PENGESAHAN .....                             | iii                          |
| KATA PENGANTAR.....                                 | iv                           |
| ABSTRAK .....                                       | vii                          |
| DAFTAR ISI .....                                    | viii                         |
| DAFTAR GAMBAR .....                                 | xi                           |
| DAFTAR TABEL.....                                   | xiii                         |
| BAB I .....                                         | 1                            |
| PENDAHULUAN .....                                   | 1                            |
| 1.1 Latar Belakang .....                            | 1                            |
| 1.2 Rumusan Masalah .....                           | 3                            |
| 1.3 Batasan Masalah.....                            | 3                            |
| 1.4 Tujuan dan Manfaat.....                         | 4                            |
| 1.4.1 Tujuan .....                                  | 4                            |
| 1.4.2 Manfaat .....                                 | 4                            |
| 1.5 Sistematika Penulisan.....                      | 4                            |
| BAB II.....                                         | 6                            |
| TINJAUAN PUSTAKA.....                               | 6                            |
| 2.1 Penelitian Sejenis .....                        | 6                            |
| 2.2 ESP32-S3.....                                   | 9                            |
| 2.3 ESP32-C3 .....                                  | 9                            |
| 2.4 Bluetooth Low Energy .....                      | 10                           |
| 2.5 Firmware pada Jaringan Sensor Nirkabel .....    | 10                           |
| 2.6 Mekanisme Firmware Over-the-Air.....            | 11                           |
| 2.7 Sensor Akselerometer ADXL345.....               | 11                           |
| 2.8 Message Queuing Telemetry Transport (MQTT)..... | 11                           |
| 2.9 Sistem Multitasking FreeRTOS.....               | 12                           |
| 2.10 NimBLE <i>Stack</i> .....                      | 12                           |
| 2.11 Google Cloud Storage .....                     | 13                           |
| 2.12 TimescaleDB .....                              | 13                           |
| 2.13 Pengujian Sistem .....                         | 14                           |
| BAB III .....                                       | 16                           |
| METODE PENELITIAN.....                              | 16                           |
| 3.1 Rancangan Penelitian .....                      | 16                           |
| 3.2 Tahapan Penelitian.....                         | 16                           |
| 3.3 Objek Penelitian .....                          | 17                           |
| BAB IV .....                                        | 17                           |
| HASIL DAN PEMBAHASAN .....                          | 17                           |
| 4.1 Analisis Kebutuhan.....                         | 17                           |
| 4.1.1 Analisis Kebutuhan Perangkat Keras.....       | 17                           |
| 4.1.1.1 Node Sensor .....                           | 18                           |



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|                                                                           |    |
|---------------------------------------------------------------------------|----|
| 4.1.1.2 Gateway.....                                                      | 19 |
| 4.1.2 Analisis Kebutuhan Perangkat Lunak.....                             | 21 |
| 4.1.3 Analisis Kebutuhan Fungsional dan Non-Fungsional .....              | 22 |
| 4.1.3.1 Kebutuhan Fungsional .....                                        | 22 |
| 4.1.3.2 Kebutuhan Non-Fungsional .....                                    | 23 |
| 4.2 Perancangan Sistem.....                                               | 24 |
| 4.2.1 Blok Diagram.....                                                   | 24 |
| 4.2.2 Arsitektur Sistem.....                                              | 25 |
| 4.2.3 Diagram Alir .....                                                  | 26 |
| 4.2.4 Rancangan Skematik Perangkat.....                                   | 30 |
| 4.2.4.1 Skematik <i>Node</i> Sensor .....                                 | 30 |
| 4.2.4.2 Skematik Gateway .....                                            | 33 |
| 4.2.5 Rancangan Komunikasi Data.....                                      | 35 |
| 4.2.5.1 Rancangan Protokol MQTT .....                                     | 35 |
| 4.2.5.2 Rancangan Profil GATT BLE .....                                   | 36 |
| 4.2.6 Rancangan Partisi Memori.....                                       | 37 |
| 4.2.7 Rancangan Basis Data .....                                          | 39 |
| 4.2.7.1 Entity Relationship Diagram.....                                  | 39 |
| 4.2.7.2 Deskripsi Tabel Basis Data .....                                  | 40 |
| 4.2.7.3 Pengelolaan Data <i>Time-Series</i> Menggunakan TimescaleDB ..... | 41 |
| 4.3 Implementasi Sistem .....                                             | 42 |
| 4.3.1 Implementasi Perangkat Keras .....                                  | 42 |
| 4.3.1.1 Implementasi <i>Node</i> Sensor.....                              | 42 |
| 4.3.1.2 Implementasi <i>Gateway</i> .....                                 | 43 |
| 4.3.2 Implementasi Perangkat Lunak Mikrokontroler.....                    | 44 |
| 4.3.2.1 Implementasi <i>Firmware</i> <i>Node</i> Sensor.....              | 44 |
| 4.3.2.2 Implementasi <i>Firmware</i> <i>Gateway</i> .....                 | 48 |
| 4.3.3 Implementasi Broker MQTT .....                                      | 52 |
| 4.3.4 Implementasi Backend dan Basis Data.....                            | 55 |
| 4.3.4.1 Implementasi Layanan Backend .....                                | 55 |
| 4.3.4.2 Implementasi Komunikasi <i>Message</i> Broker .....               | 57 |
| 4.3.5 Implementasi Antarmuka Pengguna .....                               | 59 |
| 4.3.5.1 Visualisasi Data Telemetri Pemantauan Kesehatan Struktur .....    | 59 |
| 4.3.5.2 Panel Pemicu dan Riwayat Pembaruan FOTA.....                      | 60 |
| 4.3.5.3 Halaman Pustaka Berkas <i>Firmware</i> .....                      | 61 |
| 4.4 Pengujian Sistem .....                                                | 61 |
| 4.4.1 Deskripsi Pengujian .....                                           | 61 |
| 4.4.2 Prosedur Pengujian .....                                            | 62 |
| 4.4.2.1 Prosedur Pengujian Fungsionalitas Sistem .....                    | 62 |
| 4.4.2.2 Prosedur Pengujian Kinerja Jaringan Komparatif.....               | 63 |
| 4.4.3 Data Hasil Pengujian .....                                          | 64 |
| 4.4.3.1 Hasil Pengujian Fungsionalitas Sistem .....                       | 65 |
| 4.4.3.2 Hasil Pengujian Kinerja Transmisi Berkas Kecil.....               | 69 |
| 4.4.3.3 Hasil Pengujian Kinerja Transmisi Berkas Sedang.....              | 71 |
| 4.4.3.4 Hasil Pengujian Kinerja Transmisi Berkas Besar .....              | 72 |



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|                                                                  |    |
|------------------------------------------------------------------|----|
| 4.4.4 Analisis Data dan Evaluasi Pengujian .....                 | 73 |
| 4.4.4.1 Analisis Pengujian Fungsionalitas Sistem.....            | 73 |
| 4.4.4.2 Analisis Pengujian Kinerja Transmisi Berkas Kecil .....  | 74 |
| 4.4.4.3 Analisis Pengujian Kinerja Transmisi Berkas Sedang ..... | 76 |
| 4.4.4.4 Analisis Pengujian Kinerja Transmisi Berkas Besar .....  | 77 |
| BAB V .....                                                      | 80 |
| PENUTUP.....                                                     | 80 |
| 5.1 Simpulan.....                                                | 80 |
| 5.2 Saran .....                                                  | 80 |
| DAFTAR PUSTAKA .....                                             | 82 |
| DAFTAR RIWAYAT HIDUP.....                                        | 86 |
| LAMPIRAN .....                                                   | 87 |





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## DAFTAR GAMBAR

|                                                                                                        |    |
|--------------------------------------------------------------------------------------------------------|----|
| Gambar 2.1 Perbandingan Metrik Teknologi Nirkabel .....                                                | 10 |
| Gambar 2.2 Prinsip Kerja Broker MQTT dan <i>Publisher Subscriber</i> .....                             | 12 |
| Gambar 3.1 Tahapan Penelitian .....                                                                    | 16 |
| Gambar 4.1 Blok Diagram Sistem .....                                                                   | 24 |
| Gambar 4.2 Arsitektur Sistem .....                                                                     | 25 |
| Gambar 4.3 Diagram Sekuensial Proses FOTA .....                                                        | 27 |
| Gambar 4.4 Diagram Alir Fragmentasi dan Transmisi Data .....                                           | 28 |
| Gambar 4.5 Diagram Alir Penerimaan Data dan Manajemen Partisi .....                                    | 29 |
| Gambar 4.6 Skematik Node Sensor .....                                                                  | 30 |
| Gambar 4.7 Skematik <i>Gateway</i> .....                                                               | 33 |
| Gambar 4.8 Entity Relationship Diagram Basis Data .....                                                | 40 |
| Gambar 4.10 Implementasi Node Sensor .....                                                             | 42 |
| Gambar 4.11 Implementasi <i>Gateway</i> .....                                                          | 43 |
| Gambar 4.12 Karakteristik BLE GATT OTA .....                                                           | 45 |
| Gambar 4.13 Implementasi Pembacaan Sensor ADXL345 .....                                                | 46 |
| Gambar 4.14 Implementasi Fase Inisialisasi OTA .....                                                   | 47 |
| Gambar 4.15 Implementasi Fase Penulisan OTA .....                                                      | 47 |
| Gambar 4.16 Implementasi Fase Finalisasi Perintah OTA .....                                            | 48 |
| Gambar 4.17 Implementasi <i>Event Handler</i> dan Pemrosesan <i>Payload</i> JSON .....                 | 49 |
| Gambar 4.18 Implementasi Penerimaan Notifikasi Telemetri dan Negosiasi MTU ..                          | 50 |
| Gambar 4.19 Algoritma Fragmentasi FOTA .....                                                           | 51 |
| Gambar 4.20 Implementasi Validasi Versi Pasca-Pembaruan .....                                          | 52 |
| Gambar 4.21 Tampilan Dasbor Utama Deployment EMQX Serverless .....                                     | 52 |
| Gambar 4.22 Daftar Klien yang Terhubung pada Broker EMQX .....                                         | 53 |
| Gambar 4.23 Konfigurasi Otorisasi Akses Topik MQTT .....                                               | 53 |
| Gambar 4.24 Format <i>Payload</i> Instruksi Pembaruan pada Topik shm/ota/trigger .....                 | 54 |
| Gambar 4.25 Implementasi Pesan Status Pembaruan pada Topik shm/ota/status .....                        | 55 |
| Gambar 4.26 Implementasi <i>Bucket</i> GCS .....                                                       | 55 |
| Gambar 4.27 Implementasi Worker Membuat Signed URL dan Payload MQTT .....                              | 56 |
| Gambar 4.28 Implementasi Kode Menangani Status Pembaruan .....                                         | 57 |
| Gambar 4.29 Implementasi <i>Timeout</i> pada <i>Worker</i> FOTA .....                                  | 57 |
| Gambar 4.30 Implementasi Koneksi Klien MQTT dan Pengaturan TLS .....                                   | 58 |
| Gambar 4.31 Implementasi Kanal Langganan pada Topik Status .....                                       | 59 |
| Gambar 4.32 Antarmuka Visualisasi Real-Time Telemetri Sensor SHM .....                                 | 60 |
| Gambar 4.33 Antarmuka Panel Orkestrasi dan Tabel Riwayat Pembaruan FOTA ....                           | 60 |
| Gambar 4.34 Halaman Pustaka Berkas <i>Firmware</i> .....                                               | 61 |
| Gambar 4.35 Respons Log <i>Gateway</i> saat Inisialisasi dan Penerimaan Data Telemetri .....           | 67 |
| Gambar 4.36 Fluktuasi Grafik Telemetri Berdasarkan Simulasi Gangguan Fisik ....                        | 67 |
| Gambar 4.37 Inisiasi Perintah FOTA pada Antarmuka Web .....                                            | 68 |
| Gambar 4.38 Proses Transmisi <i>Firmware</i> dari <i>Gateway</i> menuju <i>Node</i> .....              | 68 |
| Gambar 4.39 Verifikasi Integritas Versi <i>Firmware</i> Pasca- <i>Reboot</i> oleh <i>Gateway</i> ..... | 69 |
| Gambar 4.40 Sinkronisasi Status Keberhasilan OTA pada Tabel Dashboard .....                            | 69 |



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|                                                                                   |    |
|-----------------------------------------------------------------------------------|----|
| Gambar 4.41 Grafik Hasil Pengujian Fungsionalitas Sistem.....                     | 74 |
| Gambar 4.42 Grafik Komparasi Throughput Berkas Kecil BLE vs ESP-NOW .....         | 74 |
| Gambar 4.43 Grafik Komparasi Rasio Retransmisi Berkas Kecil BLE vs ESP-NOW .....  | 75 |
| Gambar 4.44 Grafik Komparasi Throughput Berkas Sedang BLE vs ESP-NOW ....         | 76 |
| Gambar 4.45 Grafik Komparasi Rasio Retransmisi Berkas Sedang BLE vs ESP-NOW ..... | 77 |
| Gambar 4.46 Grafik Komparasi Throughput Berkas Besar BLE vs ESP-NOW .....         | 78 |
| Gambar 4.47 Grafik Komparasi Rasio Retransmisi Berkas Besar BLE vs ESP-NOW .....  | 79 |





**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

**DAFTAR TABEL**

|                                                                        |    |
|------------------------------------------------------------------------|----|
| Tabel 2.1 Tabel Penelitian Terdahulu .....                             | 6  |
| Tabel 4.1 Spesifikasi Node Sensor.....                                 | 19 |
| Tabel 4.2 Spesifikasi Gateway .....                                    | 20 |
| Tabel 4.3 Kebutuhan Perangkat Lunak .....                              | 21 |
| Tabel 4.4 Kebutuhan Fungsional Sistem.....                             | 22 |
| Tabel 4.5 Kebutuhan Non-Fungsional .....                               | 23 |
| Tabel 4.6 Pemetaan Jalur Subsistem Manajemen Daya .....                | 31 |
| Tabel 4.7 Pemetaan Jalur Subsistem Pengolahan Data .....               | 31 |
| Tabel 4.8 Pemetaan Jalur Subsistem Akuisisi Data Getaran .....         | 32 |
| Tabel 4.9 Pemetaan Jalur Subsistem Pemantauan Tegangan Baterai.....    | 33 |
| Tabel 4.10 Pemetaan Jalur Subsistem Gateway .....                      | 34 |
| Tabel 4.11 Rancangan Topik dan Payload MQTT .....                      | 36 |
| Tabel 4.12 Rancangan Profil GATT BLE Kustom.....                       | 37 |
| Tabel 4.13 Tabel Partisi Memori Flash ESP32-C3 .....                   | 38 |
| Tabel 4.14 Deskripsi Tabel Basis Data.....                             | 41 |
| Tabel 4.15 Prosedur Pengujian Fungsionalitas Sistem .....              | 62 |
| Tabel 4.16 Skenario Pengujian Kinerja Jaringan dan Mekanisme FOTA..... | 63 |
| Tabel 4.17 Hasil Pengujian Fungsionalitas Sistem .....                 | 65 |
| Tabel 4.18 Data Pengujian Transmisi Berkas Kecil.....                  | 69 |
| Tabel 4.19 Data Pengujian Transmisi Berkas Sedang.....                 | 71 |
| Tabel 4.20 Data Pengujian Transmisi Berkas Besar .....                 | 72 |

**POLITEKNIK  
NEGERI  
JAKARTA**

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## BAB I PENDAHULUAN

### 1.1 Latar Belakang

*Internet of Things* (IoT) telah mengalami pertumbuhan yang pesat dalam beberapa tahun terakhir, dengan adopsi yang meluas di berbagai sektor industri, mulai dari pertanian cerdas, pemantauan infrastruktur, hingga logistik. Dalam implementasi skala besar, teknologi ini sering berwujud Jaringan Sensor Nirkabel atau *Wireless Sensor Network* (WSN) yang terdiri dari ratusan hingga ribuan *node* sensor. Perangkat-perangkat ini seringkali beroperasi tanpa pengawasan di lokasi yang beragam dan untuk jangka waktu yang lama, sehingga metode pemeliharaan perangkat secara manual tidak lagi efektif untuk implementasi skala besar yang heterogen (Arakadakis et al., 2022; Braten et al., 2020).

Pengelolaan perangkat dalam skala masif menghadirkan tantangan besar, terutama ketika perangkat mengalami *bug* perangkat lunak pasca-perilisan, membutuhkan penambalan celah keamanan, atau memerlukan penyesuaian logika bisnis baru (Bauwens et al., 2020). Dari berbagai aspek pemeliharaan, pembaruan *firmware* merupakan aktivitas yang paling kritis sekaligus paling menantang. Berbeda dengan perubahan konfigurasi parameter yang ringan, perbaikan *bug* logika atau penambahan fitur membutuhkan penggantian seluruh blok kode program pada memori mikrokontroler. Tanpa mekanisme nirkabel, teknisi harus mengakses fisik perangkat satu per satu untuk melakukan *flashing* ulang via kabel. Hal ini menjadi tidak efektif secara biaya dan waktu, bahkan mustahil dilakukan pada infrastruktur WSN yang *node*-nya tertanam permanen di lokasi yang sulit dijangkau, seperti pada struktur bangunan tinggi atau area tertutup (M. A. Mostafa et al., 2025; Peter et al., 2023).

Oleh karena itu, mekanisme *Firmware Over-the-Air* (FOTA) menjadi kebutuhan yang relevan untuk diimplementasikan, bukan sekadar fitur tambahan. Namun, implementasi FOTA pada WSN memiliki kendala ketat terkait efisiensi energi. *Node* sensor umumnya beroperasi menggunakan sumber daya baterai terbatas dan diharapkan bertahan dalam hitungan tahun. Penggunaan protokol hemat energi seperti Bluetooth Low Energy (BLE) kini menjadi standar industri untuk komunikasi jarak



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

dekat, menggantikan *Wi-Fi* yang dinilai boros daya. Hal ini divalidasi oleh studi komparatif terbaru yang menunjukkan kesenjangan konsumsi energi yang signifikan, pada pengujian aktivitas pemindaian berdurasi 60 detik protokol *Wi-Fi* mencatat konsumsi daya rata-rata sebesar 218 mW, hampir dua kali lipat lebih tinggi dibandingkan BLE yang hanya mengonsumsi 134 mW (Chakraborty et al., 2024).

Meskipun BLE ideal untuk operasional WSN berdaya rendah, penerapannya untuk mekanisme FOTA menghadirkan kompleksitas tersendiri. Berbeda dengan *Wi-Fi* yang memiliki *throughput* tinggi sebesar 6-30 Mbps, kecepatan BLE yang terbatas pada 1 Mbps dapat memperpanjang durasi transfer *firmware* secara signifikan (Jaouhari & Bouvet, 2022). Hal ini menciptakan kontradiksi, proses FOTA yang lambat berpotensi menguras energi baterai karena radio harus aktif lebih lama, serta meningkatkan risiko kegagalan transfer di tengah jalan. Mekanisme OTA konvensional yang mengirimkan *firmware* secara utuh tanpa manajemen partisi yang baik memungkinkan proses mengalami kegagalan pada protokol berdaya rendah ini.

Strategi arsitektur sistem yang memanfaatkan perangkat perantara seperti *gateway* untuk mengelola distribusi pembaruan diperlukan mengatasi permasalahan tersebut. *Gateway* bertugas menjembatani komunikasi antara *server* pusat dengan *node* sensor yang terisolasi, serta mengatur fragmentasi data agar sesuai dengan karakteristik paket BLE. Tantangan utamanya adalah merancang mekanisme transfer yang mampu menyeimbangkan kebutuhan pembaruan perangkat lunak dengan keandalan kanal transmisi BLE. Untuk mengukur batas performa dan keandalan FOTA berbasis BLE tersebut, diperlukan perbandingan langsung dengan protokol nirkabel berkarakteristik berbeda. Protokol ESP-NOW, yang merupakan teknologi komunikasi nirkabel berbasis Wi-Fi fisik dari Espressif, dapat digunakan sebagai parameter pembanding karena menawarkan kecepatan tinggi dan jangkauan luas tanpa *overhead* koneksi.

Berdasarkan permasalahan di atas, penelitian ini diarahkan untuk melakukan analisis kinerja mekanisme *Firmware Over-the-Air* pada jaringan sensor nirkabel berbasis arsitektur *gateway* BLE. Penelitian ini berfokus pada analisis mendalam aspek *throughput*, waktu pengiriman, tingkat keberhasilan, dan rasio retransmisi BLE pada variasi jarak dan penghalang fisik, dengan menyandingkan kinerjanya terhadap protokol ESP-NOW sebagai pembanding evaluasi.



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dalam penelitian ini adalah:

- a. Bagaimana merancang dan mengimplementasikan mekanisme pembaruan FOTA pada jaringan sensor nirkabel dengan arsitektur *gateway* berbasis protokol BLE?
- b. Bagaimana performa kecepatan transmisi dan durasi waktu FOTA pada arsitektur berbasis BLE yang dibangun jika dibandingkan dengan protokol ESP-NOW?
- c. Bagaimana dampak redaman jarak transmisi dan adanya hambatan fisik terhadap rasio retransmisi dan tingkat keberhasilan FOTA berbasis BLE?

## 1.3 Batasan Masalah

Mengingat luasnya cakupan pengembangan sistem ini, serta untuk menjaga fokus penelitian pada implementasi mekanisme pembaruan *firmware* berbasis BLE, penulis menetapkan batasan masalah sebagai berikut:

- a. Implementasi sistem dilakukan pada mikrokontroler jenis ESP32-S3 sebagai *gateway* dan ESP32-C3 sebagai *node* sensor dengan protokol BLE versi 5.0.
- b. Fokus utama penelitian adalah mekanisme transmisi data *firmware* dari perangkat *gateway* menuju *node* menggunakan protokol BLE. Mekanisme komunikasi antara *server* ke *gateway* melalui *Wi-Fi* atau Internet hanya dibahas sebatas fungsionalitas pengunduhan *file* dan tidak dianalisis secara mendalam.
- c. Proses pembaruan *firmware* ke banyak *node* dilakukan secara sekuensial oleh mekanisme antrian pada *backend*, bukan secara paralel langsung dari *gateway* untuk menjaga stabilitas koneksi dan integritas data pada kanal BLE.
- d. Analisis kinerja sistem dibatasi pada pengukuran parameter *throughput*, efisiensi konsumsi daya selama proses OTA, jumlah transmisi ulang paket pada proses OTA, dan tingkat keberhasilan proses pembaruan.
- e. Pembacaan data sensor getaran ADXL345 dan tegangan baterai pada *node* sensor hanya berfungsi sebagai data telemetri prototipe untuk mensimulasikan kondisi operasional jaringan sensor nirkabel. Penelitian ini tidak membahas kalibrasi presisi, validasi akurasi pembacaan fisik sensor getaran, maupun analisis kelayakan struktural dari atap baja itu sendiri.



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## 1.4 Tujuan dan Manfaat

### 1.4.1 Tujuan

Tujuan utama dari penelitian ini adalah :

- a. Mengembangkan mekanisme pembaruan *firmware* nirkabel pada Jaringan Sensor Nirkabel berbasis ESP32 dengan menerapkan arsitektur *gateway* yang memanfaatkan protokol BLE.
- b. Menganalisis performa kecepatan transmisi dan durasi FOTA berbasis BLE dengan membandingkannya terhadap protokol ESP-NOW sebagai tolok ukur pada berbagai ukuran berkas.
- c. Menganalisis dampak jarak dan hambatan fisik terhadap rasio retransmisi dan tingkat keberhasilan FOTA berbasis BLE untuk memetakan batas toleransi keandalannya.

### 1.4.2 Manfaat

Manfaat dari hasil perancangan sistem ini antara lain:

- a. Meningkatkan efisiensi operasional teknis dalam melakukan pemeliharaan perangkat lunak pada *node* sensor yang sulit dijangkau.
- b. Memperpanjang usia pakai baterai pada *node* sensor melalui penerapan metode transmisi BLE yang dioptimalkan.
- c. Menyediakan data analisis empiris mengenai karakteristik dan kinerja transfer *firmware* via BLE yang dapat dimanfaatkan sebagai referensi akademis untuk pengembangan sistem IoT *low-power* di masa depan.

## 1.5 Sistematika Penulisan

Sistematika penulisan penelitian ini dibagi menjadi 5 bab dengan sistematika sebagai berikut:

### a. BAB I PENDAHULUAN

Bab ini membahas dasar pemikiran dilakukannya penelitian yang mencakup latar belakang masalah mengenai urgensi efisiensi pembaruan perangkat lunak pada jaringan sensor nirkabel berbasis *gateway*. Selanjutnya, bab ini menjabarkan rumusan masalah, batasan masalah untuk menjaga fokus penelitian, tujuan yang



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

ingin dicapai, manfaat penelitian bagi pengembangan ilmu dan industri, serta diakhiri dengan sistematika penulisan laporan.

#### b. BAB II TINJAUAN PUSTAKA

Bab ini menguraikan landasan teori yang dapat mendukung penelitian, meliputi spesifikasi teknis mikrokontroler ESP32-S3 dan ESP32-C3, arsitektur protokol BLE, serta mekanisme *Over-the-Air (OTA) Update*.

#### c. BAB III METODE PENELITIAN

Bab ini menguraikan metode penelitian dengan pendekatan rekayasa sistem, mencakup rancangan visualisasi dan alur logika sistem IoT pada mikrokontroler mulai dari analisis hingga pengujian. Pembahasan meliputi spesifikasi sumber dan teknik pengumpulan data melalui eksperimen perangkat keras, teknik analisis kinerja teknis, serta dilengkapi dengan jadwal pelaksanaan dan rincian anggaran biaya.

#### d. BAB IV HASIL DAN PEMBAHASAN

Bab ini memaparkan data hasil analisis kebutuhan, perancangan arsitektur sistem, realisasi implementasi perangkat keras, serta pengembangan perangkat lunak mikrokontroler dan antarmuka pemantauan berbasis web. Bab ini juga membahas evaluasi hasil pengujian fungsionalitas, analisis performa jaringan, serta analisis komparatif konsumsi daya *node* sensor.

#### e. BAB V PENUTUP

Bab ini menyajikan simpulan akhir dari seluruh rangkaian pengujian dan evaluasi kinerja sistem yang telah dirancang, serta saran-saran teoretis dan praktis yang diajukan untuk pengembangan penelitian selanjutnya di masa depan.

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Sejenis

Tinjauan terhadap studi terdahulu diperlukan sebagai landasan teoretis sekaligus pembandingan untuk memastikan posisi kebaruan riset ini.

Tabel 2.1 Tabel Penelitian Terdahulu

| Jurnal                  | Hasil                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Catatan                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (Hakim et al., 2021)    | <p><b>Tujuan</b> : Penerapan over the air <i>update firmware</i> dengan protokol MQTT dengan aplikasi berbasis website sebagai interface pengguna</p> <p><b>Teknologi</b> : ESP8266-12E, DHT11, MQTT</p> <p><b>Cara Kerja</b> : Perangkat target melakukan <i>subscribe</i> ke topik tertentu pada broker MQTT yang nantinya akan menerima pesan <i>update</i> jika ada pengguna yang mengunggah file <i>.bin firmware</i> yang baru, setelah menerima pesan proses <i>update</i> dilakukan pada sisi perangkat.</p> | <p>Penggunaan MQTT sebagai protokol adalah standar industri yang sangat umum sehingga mudah diintegrasikan dengan banyak platform, serta penelitian ini secara detail menganalisis dampak level QoS (0, 1, 2) terhadap keberhasilan <i>update</i>.</p> <p>Namun, penggunaan MQTT bergantung pada protokol komunikasi Wi-Fi yang tinggi dalam hal konsumsi daya sehingga tidak cocok pada sistem jaringan sensor nirkabel.</p> |
| (Formanek et al., 2025) | <p><b>Tujuan</b> : Membangun arsitektur remote update multi-protokol komunikasi untuk IoT berbasis ESP32</p> <p><b>Teknologi</b> : ESP32, Wi-Fi, BLE, Zigbee, Mobile app</p>                                                                                                                                                                                                                                                                                                                                         | <p>Arsitektur pembaruan multi-protokol yang diajukan pada penelitian berhasil mengeksplorasi keterbatasan penggunaan protokol komunikasi</p>                                                                                                                                                                                                                                                                                  |



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                                                                                                                                                                                                                             |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                          | <p><b>Cara Kerja :</b> Aplikasi secara otomatis memverifikasi versi firmware ke server dan segera mengunduh pembaruan ke penyimpanan sementara jika terdeteksi adanya perbedaan versi. Selanjutnya, firmware ditransmisikan ke perangkat target via BLE menggunakan paket data kecil yang divalidasi integritasnya melalui checksum, kemudian file sementara tersebut akan dihapus secara otomatis dari penyimpanan lokal setelah proses instalasi dan restart berhasil diselesaikan.</p> | <p>tunggal dari <i>framework</i> OTA yang sudah ada seperti OTA Library dari Esspresif dan Zigbee2MQTT OTA.</p> <p>Namun, implementasi OTA berbasis BLE yang dirancang pada penelitian ini memerlukan <i>smartphone</i> dengan aplikasi khusus sebagai <i>gateway</i>, hal ini tidak cocok untuk Jaringan Sensor Nirkabel yang tersebar.</p>                |
| (Mahfoudhi et al., 2022) | <p><b>Tujuan :</b> Mengimplementasikan mekanisme <i>over-the-air firmware update</i> untuk Narrowband IoT (NB-IoT).</p> <p><b>Teknologi :</b> STM32 Nucleo, NB-IoT Module, Flash Memory</p> <p><b>Cara Kerja :</b> Server memberikan notifikasi bahwa ada <i>firmware</i> versi baru kepada semua perangkat yang memenuhi syarat. Setelah dilakukan cek integritas menggunakan token perangkat akan mendownload <i>firmware</i> dari server dan melakukan <i>restart</i>.</p>             | <p>Penggunaan NB-IoT menggunakan koneksi seluler memungkinkan <i>firmware</i> dikirim dalam segmen-segmen yang cukup besar (1280 <i>bytes</i>).</p> <p>Berbeda dengan NB-IoT yang menurut penelitian ini mengalami latensi hingga 10 detik pada RSSI -110 dBm, BLE 5.0 memprioritaskan integritas layanan dengan membatasi koneksi hingga RSSI -95 dBm.</p> |



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                              |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Mekanisme ini justru menjamin <i>throughput</i> yang stabil selama perangkat terhubung, menjadikannya opsi yang lebih andal untuk eksekusi FOTA dibandingkan NB-IoT pada kondisi sinyal batas.                                                                                                                                               |
| (Nguyen et al., 2024)  | <p><b>Tujuan</b> : Membuat rancangan mekanisme <i>firmware update over-the-air</i> pada jaringan LoRaWAN yang memiliki bandwidth sangat terbatas</p> <p><b>Teknologi</b> : STM32WLE5JC, LoRaWAN, MQTT</p> <p><b>Cara Kerja</b> : Dari awal <i>firmware</i> tidak dibuat sebagai satu file besar melainkan dipisah menjadi <i>Micro-system</i> dan <i>Modules</i>. Ketika ada perubahan kode, server hanya mengkompilasi modul yang berubah, modul biner tersebut dipecah menjadi fragmen-fragmen kecil agar muat dalam payload paket LoRaWAN. Fragmen dikirim ke perangkat via MQTT yang nantinya akan disusun kembali di buffer menjadi modul dan memuat modul baru tersebut ke memori</p> | <p>Pendekatan <i>update</i> modular yang dikembangkan berhasil membuat ukuran paket menjadi sangat kecil hingga hanya sekitar 2 KB sehingga sangat menghemat <i>bandwidth</i> LoRa yang kecil.</p> <p>Penerapan modular manager memakan RAM dan Flash tambahan untuk sistem manajemennya, kurang efisien untuk Jaringan Sensor Nirkabel.</p> |
| (Cahyono et al., 2023) | <p><b>Tujuan</b> : Mengembangkan <i>gateway</i> untuk mendukung proses <i>over-the-</i></p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | <p>Penggunaan <i>gateway</i> berhasil mengatasi</p>                                                                                                                                                                                                                                                                                          |



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                   |
|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | <p><i>air firmware</i> pada perangkat WSN berbasis BLE</p> <p><b>Teknologi :</b> Raspberry Pi 3, ESP32, BLE, Firebase</p> <p><b>Cara Kerja :</b> Pengguna mengunggah file berformat .bin melalui aplikasi web, setelah sukses status pembaruan pada Firebase akan berubah menjadi aktif, Raspberry Pi 3 sebagai <i>gateway</i> mengecek secara berkala status tersebut, lalu mengirim data file <i>firmware byte per byte</i> ke ESP32 untuk dilakukan pembaruan.</p> | <p>masalah pada perangkat WSN yang tidak memiliki konektivitas internet secara langsung</p> <p>Proses pengiriman data dengan pendekatan <i>byte per byte</i> membuat durasi transfer sangat lama. Hasilnya, untuk mengirim data sekecil 227 kB pada jarak 5 meter, dibutuhkan waktu rata-rata 29 menit 52 detik. Untuk data 954 kB, butuh waktu hampir 2 jam.</p> |
|--|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### 2.2 ESP32-S3

ESP32-S3 merupakan mikrokontroler rendah daya yang terintegrasi penuh dengan konektivitas *Wi-Fi* 2.4 GHz dan Bluetooth Low Energy. Arsitektur perangkat ini ditenagai oleh mikroprosesor *dual-core* Xtensa® 32-bit LX7 berkinerja tinggi yang dilengkapi dengan *co-processor* rendah daya, modul RF, serta beragam periferal pendukung. Secara spesifik, ESP32-S3 dirancang untuk mendukung aplikasi *Artificial Intelligence of Things* (AIoT) karena kapabilitas *Digital Signal Processing* (DSP) yang mampu mengakselerasi pemrosesan *Convolutional Neural Network* (CNN) serta memiliki kapasitas daya yang memadai untuk mengolah informasi audio-visual secara efisien (Nemlaha et al., 2023).

### 2.3 ESP32-C3

ESP32-C3 merupakan mikrokontroler *single-core* yang dibangun di atas arsitektur RISC-V, perangkat ini dirancang sebagai solusi yang hemat biaya dan memiliki konsumsi daya rendah namun tetap menawarkan kemampuan pemrosesan yang memadai untuk menangani aplikasi kompleks seperti sistem *smart home* dan otomasi



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

industri. Keunggulan utama ESP32-C3 terletak pada integrasi modul komunikasi Wi-Fi dan BLE secara *built-in* tanpa memerlukan perangkat keras tambahan, serta kebutuhan daya yang kecil tanpa mempertaruhkan kestabilan uptime membuat mikrokontroler ini sangat relevan untuk sistem yang beroperasi secara kontinu (Kurniawan & Budjianto, 2025).

## 2.4 Bluetooth Low Energy

*Bluetooth Low Energy* merupakan teknologi komunikasi nirkabel yang dominan untuk aplikasi *Internet of Things* berkat karakteristik utama seperti konsumsi daya yang sangat rendah, memungkinkan perangkat beroperasi selama bertahun-tahun dengan baterai kecil, serta biaya implementasi yang efisien dan kemudahan integrasi. Arsitektur BLE memungkinkan perangkat untuk sebagian besar waktu dalam mode tidur yang hanya memakan daya sekitar 1  $\mu$ A dan hanya aktif saat transmisi data diperlukan, sebuah prinsip yang secara signifikan mengurangi konsumsi energi dibandingkan dengan Bluetooth klasik (Liu et al., 2021). Meskipun BLE menawarkan keunggulan dalam efisiensi energi dan adaptasi di lingkungan dinamis, teknologi ini memiliki keterbatasan jangkauan hingga 30 meter dan kecepatan transfer data yang lebih rendah (hingga 1 Mbps) dibandingkan Wi-Fi, yang dapat memperpanjang waktu pembaruan dan meningkatkan kemungkinan kesalahan (Nakakaze et al., 2024). Perbandingan karakteristik teknis antara BLE dengan teknologi transmisi nirkabel lainnya dapat dilihat pada Gambar 2.1.

| Technology (Sensors) | Usage          | Sensor Range [m] | Position Accuracy [m] | Latency of IPS [ms] | Tracking Indoors | Multiple Positioning |
|----------------------|----------------|------------------|-----------------------|---------------------|------------------|----------------------|
| GPS                  | Outdoor        | -                | 5–10                  | 200–1000            | Yes              | Yes                  |
| Magnetic             | Indoor/outdoor | -                | 1–2                   | 1–10                | No               | Yes                  |
| Inertial             | Indoor/outdoor | -                | -                     | 1–20                | No               | No                   |
| Ultrasound           | Indoor         | 10               | 0.01–0.05             | 5–100               | No               | Yes                  |
| Infrared             | Indoor         | 5                | 1–2                   | 10–100              | Yes              | Yes                  |
| VLC                  | Indoor/outdoor | -                | 0.01–0.1              | 1–10                | Yes              | Yes                  |
| RFID                 | Indoor         | 50               | 1–3                   | 10–200              | No               | Yes                  |
| NFC                  | Indoor         | 0.04–0.1         | 0.01–0.04             | 1–10                | No               | No                   |
| Wi-Fi                | Indoor/outdoor | 50               | 1–5                   | 30–100              | No               | No                   |
| Bluetooth 5.0        | Indoor/outdoor | 40               | 1–5                   | 10–100              | Yes              | Yes                  |
| BLE                  | Indoor/outdoor | 25               | 1–5                   | 1–6                 | Yes              | Yes                  |
| UWB                  | Indoor/outdoor | 30               | 0.06–0.1              | 0.00001             | Yes              | Yes                  |
| VSS                  | Indoor/outdoor | 50               | -                     | 50–500              | Yes              | Yes                  |
| QR codes             | Indoor/outdoor | 0.25–7.6         | -                     | 50–200              | No               | No                   |

Gambar 2.1 Perbandingan Metrik Teknologi Nirkabel

Sumber : (Skýpalová et al., 2025)

## 2.5 Firmware pada Jaringan Sensor Nirkabel



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Dalam konteks *embedded system* dan IoT, *firmware* merupakan perangkat lunak tingkat rendah yang ditanamkan ke dalam memori *non-volatile* seperti *flash memory* pada sebuah mikrokontroler. *Firmware* berada di antara perangkat keras dan perangkat lunak lapisan aplikasi dengan tugas sebagai program antarmuka untuk lapisan perangkat lunak dengan mewujudkan perintah perangkat keras. *Firmware* mencakup kombinasi berbagai *file* biner, termasuk *bootloader*, kernel sistem operasi, sistem *file*, dan berbagai *header*, yang seringkali dikompresi mengingat keterbatasan komputasi dan ruang penyimpanan pada perangkat IoT (Ul Haq et al., 2023).

## 2.6 Mekanisme Firmware Over-the-Air

Mekanisme *Firmware Over-the-Air* (FOTA) merupakan pendekatan krusial dalam manajemen siklus hidup perangkat IoT yang memungkinkan pembaruan perangkat lunak secara nirkabel, esensial untuk mengatasi *bug*, menambahkan fungsionalitas baru, dan memperbaiki kerentanan keamanan tanpa intervensi fisik (Arakadakis et al., 2022). Implementasi FOTA umumnya mengadopsi arsitektur memori partisi ganda di mana satu partisi berfungsi sebagai sistem aktif, sementara partisi lain menjadi wadah sementara untuk *firmware* baru. Proses ini melibatkan pengunduhan *firmware*, verifikasi integritas dan otentikasi, serta tanda tangan digital, sebelum *bootloader* melakukan penukaran partisi. Pendekatan ini memastikan pembaruan yang aman dalam rangka mencegah *firmware* yang korup karena masalah seperti kehilangan daya atau koneksi (Jaouhari & Bouvet, 2022).

## 2.7 Sensor Akselerometer ADXL345

Sensor ADXL345 merupakan sensor akselerometer digital tiga sumbu (X, Y, dan Z) berbasis *Micro-Electro-Mechanical Systems* (MEMS) yang dirancang untuk mendeteksi percepatan dinamis akibat getaran atau guncangan, serta percepatan statis akibat gravitasi bumi. Sensor ini bekerja pada tegangan 2,0 V hingga 3,6 V pada mode pengukuran (Ferrina et al., 2022), serta mendukung interkoneksi digital standar melalui antarmuka komunikasi I2C dan SPI.

## 2.8 Message Queuing Telemetry Transport (MQTT)

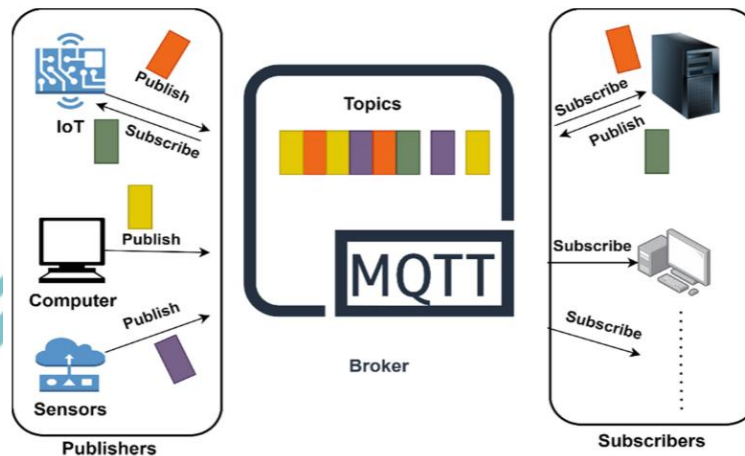
*Message Queuing Telemetry Transport* (MQTT) adalah protokol komunikasi data ringan berbasis *publish-subscribe* yang berjalan di atas arsitektur TCP/IP dan dirancang secara spesifik untuk perangkat dengan keterbatasan komputasi serta



#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

jaringan dengan *bandwidth* terbatas. Arsitektur protokol ini bertumpu pada peran *Broker* yang bertindak sebagai pengelola pesan lalu lintas data dari *Publisher* menuju *Subscriber* berdasarkan filter topik tertentu (Nemlaha et al., 2023), prinsip kerja MQTT tersebut diperlihatkan pada Gambar 2.2.



Gambar 2.2 Prinsip Kerja Broker MQTT dan *Publisher Subscriber*

Sumber: (Nemlaha et al., 2023)

## 2.9 Sistem Multitasking FreeRTOS

FreeRTOS berfungsi sebagai kernel sistem operasi real-time dan terukur yang dirancang khusus untuk mikrokontroler tertanam, menyediakan penjadwalan tugas penting dan mekanisme komunikasi antar-proses. Dengan beroperasi di bawah lisensi MIT, ia menawarkan solusi hemat biaya bagi pengembang yang memprioritaskan portabilitas dan pemeliharaan di lingkungan dengan sumber daya terbatas (Behnke, 2024). Arsitektur kernel yang ringan mendukung lebih dari 40 arsitektur mikrokontroler yang berbeda termasuk RISC-V dan ARMv8-M (Mazzi et al., 2021), memungkinkan efisiensi energi yang tinggi melalui mode hemat daya yang sangat cepat. Dengan menekan interupsi pengatur waktu berkala yang biasanya membangunkan prosesor, mekanisme idle tanpa tick memungkinkan mikrokontroler untuk tetap berada dalam kondisi tidur nyenyak untuk waktu yang lama, hanya melanjutkan operasi ketika terjadi interupsi eksternal atau tugas terjadwal sudah siap.

## 2.10 NimBLE Stack

NimBLE merupakan *stack* protokol BLE *open-source* yang dikembangkan dan telah terintegrasi ke dalam kerangka kerja Espressif IoT Development Framework (ESP-



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

IDF) sebagai alternatif dari *stack* Bluedroid bawaan. Dirancang khusus untuk sistem tertanam yang memiliki keterbatasan sumber daya, NimBLE menawarkan efisiensi penggunaan memori RAM dan *flash* yang jauh lebih tinggi ~37% dibandingkan dengan Bluedroid sehingga sangat membantu dalam menekan konsumsi daya operasional mikrokontroler (Milić et al., 2025). Pustaka ini mendukung penuh seluruh spesifikasi BLE 5.0, termasuk fitur kustomisasi nilai MTU hingga kapasitas maksimal 512 bytes serta opsi transmisi tanpa konfirmasi balik yaitu *Write Without Response*.

## 2.11 Google Cloud Storage

Google Cloud Storage (GCS) merupakan layanan penyimpanan data tidak terstruktur berbasis *cloud* yang dikembangkan oleh Google Cloud Platform dengan fokus pada skalabilitas tinggi, keandalan data, serta sistem keamanan yang ketat (Aprianto & Ramayanti, 2024). Berbeda dengan sistem berkas hierarkis tradisional, GCS mengorganisasikan data sebagai objek mandiri di dalam wadah penyimpanan yang disebut *bucket*, di mana setiap objek memiliki kunci pengenalan unik beserta metadata pendukung. Dalam arsitektur sistem FOTA, GCS dimanfaatkan sebagai repositori terpusat untuk menyimpan berkas biner *firmware* secara aman. GCS menyediakan fitur *Signed URL* yang memungkinkan sistem menghasilkan tautan unduhan sementara terenkripsi dengan parameter waktu kedaluwarsa tertentu. Melalui mekanisme ini, perangkat perantara seperti *gateway* dapat melakukan pengunduhan aliran data secara langsung melalui protokol HTTP *Client* secara aman tanpa harus membuka hak akses *bucket* secara permanen kepada publik.

## 2.12 TimescaleDB

TimescaleDB merupakan basis data relasional yang dibangun di atas PostgreSQL, dioptimalkan khusus untuk menyimpan dan menganalisis data deret waktu (*time-series*) dalam skala besar. Basis data ini memanfaatkan konsep *hypertables* untuk secara otomatis mempartisi data berdasarkan dimensi waktu, sehingga memungkinkan eksekusi kueri agregasi yang kompleks dengan latensi rendah pada set data berskala besar (J. Mostafa et al., 2022). Selain itu, fitur kompresi data berbasis umur dengan mengubah baris menjadi format kolom pada TimescaleDB mampu menekan kebutuhan ruang penyimpanan secara signifikan, yang sangat krusial untuk menangani beban data telemetri yang terus meningkat dari perangkat IoT (J. Mostafa et al., 2022).



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

### 2.13 Pengujian Sistem

Pengujian sistem merupakan proses evaluasi terhadap integrasi seluruh komponen perangkat keras dan perangkat lunak untuk memastikan bahwa sistem mampu beroperasi sesuai dengan spesifikasi kebutuhan yang telah ditetapkan. Beberapa pengujian yang dilakukan terhadap sistem pada penelitian ini antara lain:

#### 1. Pengujian Fungsionalitas

Pengujian fungsional menerapkan metode *black-box testing*, yaitu pengujian yang berfokus pada verifikasi fungsionalitas aplikasi dengan mengamati kesesuaian antara data *input* dan *output* tanpa melibatkan analisis terhadap alur logika internal kode program. Metrik evaluasi pengujian fungsional dihitung menggunakan persentase tingkat keberhasilan uji dengan persamaan (1) (Ayuningtyas et al., 2023):

$$\text{Persentase Error} = \frac{\text{Jumlah Skenario Gagal}}{\text{Total Pengujian}} \times 100\%$$

#### 2. Pengujian Kinerja Jaringan

Pengujian kinerja jaringan berfokus pada analisis transmisi nirkabel berkas *firmware* FOTA berbasis protokol BLE. Pengujian ini diukur menggunakan tiga metrik jaringan, yaitu *throughput*, *success rate*, dan *retransmission ratio*. Nilai *throughput* Mengukur laju kecepatan transfer data aplikasi per satuan waktu dalam satuan Kbps dengan persamaan (2) (Putra et al., 2021):

$$\text{Throughput (Kbps)} = \frac{\text{Ukuran Berkas (Bytes)} \times 8}{\text{Waktu Transmisi (detik)} \times 1000}$$

*Success rate* mengukur tingkat keandalan koneksi FOTA hingga selesai tanpa putus di tengah jalan dengan persamaan (3) (Hakim et al., 2021):

$$\text{Success Rate (\%)} = \left( \frac{\text{Transmisi Sukses}}{\text{Total Pengujian}} \right) \times 100\%$$

Persentase retransmisi: Mengukur rasio beban pengulangan paket data di udara (overhead) akibat terjadinya *packet loss* dengan persamaan (4) (Baazaoui et al., 2023):

$$\text{Retransmission Rate} = \left( \frac{\text{Total Paket Diulang}}{\text{Total Paket Ideal (Expected Chunks)}} \right) \times 100\%$$



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

### 3. Pengujian Efisiensi Konsumsi Daya

Pengujian efisiensi konsumsi daya dihitung berdasarkan arus searah yang dikonsumsi pada berbagai skenario pengujian untuk memvalidasi perangkat dan evaluasi kinerja FOTA berbasis BLE yang dibangun.

Nilai efisiensi penurunan arus pada berbagai skenario menggunakan persamaan (5) (Hariyadi et al., n.d.):

$$\eta = (X_{(sebelum)} - X_{(sesudah)}) / X_{(sebelum)} \times 100\%$$

Perhitungan konsumsi energi kumulatif (6) dirumuskan sebagai berikut (Subagiyo et al., 2023):

$$E = V \times I_{rata-rata} \times t$$

Keterangan:

- $E$  = Energi kumulatif yang dikonsumsi oleh *node* sensor (Joule)
- $V$  = Tegangan catu daya nominal baterai LiPo, yaitu 3,7 V
- $I_{rata-rata}$  = Arus rata-rata yang mengalir selama fase FOTA
- $t$  = Durasi penyelesaian transfer *firmware* (detik).

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## BAB III

### METODE PENELITIAN

#### 3.1 Rancangan Penelitian

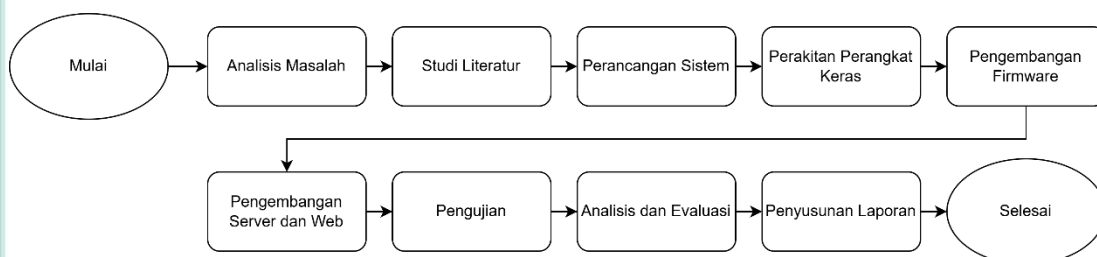
Penelitian ini menggunakan pendekatan penelitian kuantitatif eksperimental dengan evaluasi terhadap sistem yang dibangun menitikberatkan pada pengukuran kinerja secara objektif yang menghasilkan data numerik yang dilakukan dengan cara merancang, membangun, dan mengimplementasikan prototipe sistem komunikasi nirkabel berbasis mikrokontroler ESP32-S3 sebagai *gateway* dan ESP32-C3 sebagai *node* sensor. Setelah prototipe berhasil dibangun, eksperimen dilakukan di dalam lingkungan terkendali dengan memberikan berbagai skenario perlakuan.

Pengumpulan data pada penelitian ini dilakukan melalui pengukuran observasi perangkat keras dan perangkat lunak secara langsung saat sistem beroperasi.

- a. Pengujian fungsionalitas dengan *black-box testing* untuk memastikan sistem yang telah dibangun berjalan sesuai dengan rancangan.
- b. Data metrik kinerja jaringan, seperti *throughput*, *success rate*, dan *retransmission rate* dikumpulkan dengan cara merekam serial monitor yang dihasilkan oleh *gateway* dan *node* saat proses FOTA berlangsung baik dengan protokol BLE ataupun ESP-NOW.

Seluruh data hasil pengukuran waktu tempuh pembaruan, *throughput*, retransmisi, dan persentase keberhasilan akan dikumpulkan, dianalisis, dan divisualisasikan dalam bentuk grafik serta tabel komparasi. Melalui analisis ini, akan dievaluasi secara komprehensif batas toleransi jangkauan, keandalan, dan efisiensi kecepatan transmisi arsitektur *gateway* FOTA BLE jika dibandingkan dengan protokol ESP-NOW.

#### 3.2 Tahapan Penelitian



Gambar 3.1 Tahapan Penelitian



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Gambar 3.1 memperlihatkan proses diawali dengan tahap analisis masalah untuk mengidentifikasi permasalahan yang ada pada proses pembaruan *firmware*, studi literatur untuk memperkuat landasan teori dengan melibatkan berbagai referensi seperti jurnal, buku, ataupun artikel ilmiah, kemudian dilanjutkan dengan perancangan sistem serta perakitan perangkat keras sebagai tahap awal realisasi. Pengembangan *firmware* dilakukan setelahnya agar perangkat keras dapat bekerja sesuai dengan rancangan. Pengembangan server atau *backend* dan antarmuka web dilakukan untuk mengintegrasikan sistem yang telah dibangun dan sebagai pusat kontrol sistem oleh pengguna. Kinerja sistem lalu divalidasi melalui serangkaian pengujian serta data hasil pengujian dianalisis dan dievaluasi secara mendalam, sebelum akhirnya seluruh rangkaian kegiatan didokumentasikan dalam tahap penulisan laporan penelitian.

### 3.3 Objek Penelitian

Objek utama yang dikaji dalam penelitian ini adalah kinerja transmisi data pada mekanisme pembaruan *firmware* nirkabel berbasis protokol Bluetooth Low Energy yang dibandingkan terhadap protokol ESP-NOW pada arsitektur jaringan sensor nirkabel heterogen. Mekanisme tersebut diterapkan pada prototipe pemantauan kesehatan struktur rangka atap baja menggunakan dua varian mikrokontroler dari keluarga Espressif, yaitu ESP32-S3 sebagai \*gateway\* dan ESP32-C3 sebagai \*node\* sensor, yang dipilih berdasarkan kesesuaian karakteristik teknis masing-masing dalam arsitektur sistem FOTA.

## BAB IV HASIL DAN PEMBAHASAN

### 4.1 Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk mengidentifikasi spesifikasi sistem yang diperlukan dalam penelitian mekanisme Firmware Over-The-Air (FOTA) pada jaringan sensor nirkabel dengan arsitektur *gateway* BLE.

#### 4.1.1 Analisis Kebutuhan Perangkat Keras

Kebutuhan perangkat keras dalam penelitian ini ditentukan berdasarkan arsitektur sistem yang dikembangkan, sistem terdiri dari dua komponen utama, yaitu *sensor node* dan *gateway* BLE.



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

#### 4.1.1.1 Node Sensor

Perangkat sensor berfungsi sebagai unit akuisisi data sekaligus target pembaruan firmware OTA. Komponen yang digunakan meliputi:

- a) ESP32-C3 SuperMini sebagai mikrokontroler utama pada sensor node yang berfungsi untuk memproses data sensor, mengelola komunikasi BLE, serta menjalankan mekanisme OTA. Mikrokontroler ini dipilih sebab konsumsi daya yang rendah, dukungan Bluetooth Low Energy (BLE), ukuran fisik yang kecil, serta kemampuan pemrosesan yang memadai untuk aplikasi jaringan sensor nirkabel.
- b) Sensor ADXL345 sebagai sensor akselerometer yang berfungsi untuk mendeteksi getaran atau perubahan percepatan pada objek yang dipantau. Sensor ini dipilih karena konsumsi daya yang rendah, resolusi pengukuran yang tinggi, serta dukungan antarmuka komunikasi digital (I2C/SPI) yang mudah diintegrasikan dengan mikrokontroler.
- c) Baterai Lithium Polymer 3.7v 1100mAh sebagai sumber daya utama sensor node yang memungkinkan perangkat beroperasi secara mandiri tanpa koneksi catu daya eksternal. Penggunaan baterai dipilih untuk merepresentasikan karakteristik sistem IoT serta memungkinkan evaluasi konsumsi energi selama operasi normal dan proses OTA.
- d) Rangkaian Pembagi Tegangan (Voltage Divider) yang terdiri dari dua Resistor 100 k $\Omega$  sebagai komponen pendukung yang berfungsi untuk menyesuaikan tegangan baterai agar berada dalam batas aman pembacaan Analog-to-Digital-Converter (ADC) mikrokontroler. Penggunaan rangkaian pembagi tegangan dipilih karena tegangan baterai Li-ion/Li-Po dapat mencapai 4.2 V pada kondisi penuh, sedangkan batas aman input ADC mikrokontroler adalah sekitar 3.3 V
- e) Modul Pengisian Baterai TP4056 sebagai komponen manajemen daya yang berfungsi untuk mengatur proses pengisian ulang baterai Li-ion/Li-Po pada sensor node. Komponen ini dipilih sebab mampu menyediakan mekanisme pengisian baterai dengan metode *constant current/constant voltage* (CC/CV), yang merupakan standar pengisian baterai Li-Po.

Komponen-komponen perangkat keras pada *node* sensor ini dapat dilihat spesifikasinya pada tabel 4.1 berikut:



Tabel 4.1 Spesifikasi Node Sensor

| No | Nama Komponen                                          | Spesifikasi                                                                                                                                                                                                                                                    |
|----|--------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | ESP32-C3 SuperMini                                     | <ul style="list-style-type: none"> <li>- Arsitektur RISC-V 32-bit</li> <li>- BLE 5.0</li> <li>- Tegangan kerja 3.3 V</li> <li>- Konsumsi daya rendah</li> <li>- Memori flash internal</li> </ul>                                                               |
| 2  | Sensor ADXL345                                         | <ul style="list-style-type: none"> <li>- Tipe: Digital accelerometer</li> <li>- Resolusi hingga 13-bit</li> <li>- Rentang pengukuran <math>\pm 2g</math> hingga <math>\pm 16g</math></li> <li>- Antarmuka I2C / SPI</li> <li>- Konsumsi daya rendah</li> </ul> |
| 3  | Baterai Lithium Polymer 3.7 V 1100 mAh                 | <ul style="list-style-type: none"> <li>- Kapasitas: 1100mAh</li> <li>- Ukuran: 3CM X 3.5CM</li> <li>- Tebal: 1cm</li> <li>- Tegangan: 3.7V</li> <li>- <i>Life cycle</i>: 1000 kali pengisian</li> <li>- Dilengkapi IC <i>protection board</i></li> </ul>       |
| 4  | Voltage Divider ( $2 \times$ Resistor 100 k $\Omega$ ) | <ul style="list-style-type: none"> <li>- Resistor R1: 100 k<math>\Omega</math></li> <li>- Resistor R2: 100 k<math>\Omega</math></li> <li>- Rasio pembagian: 1:2</li> </ul>                                                                                     |
| 5  | Modul TP4056                                           | <ul style="list-style-type: none"> <li>- Metode pengisian: CC/CV</li> <li>- Tegangan input: 5 V</li> <li>- Tegangan pengisian: 4.2 V</li> <li>- Fitur proteksi baterai</li> </ul>                                                                              |
| 6  | Slide Switch SS12D10                                   | <ul style="list-style-type: none"> <li>- Single Pole Double Throw (SPDT)</li> <li>- 3 kaki Pin</li> <li>- Rating 0.5A 50V DC / 0.3A 4V DC</li> </ul>                                                                                                           |

#### 4.1.1.2 Gateway

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Gateway BLE berfungsi sebagai pusat komunikasi, pengelola OTA, serta penghubung antara sensor node dan sistem monitoring. Komponen yang digunakan meliputi:

- a) ESP32-S3 DevKitC-1 WROOM-1 (Lexin N16R8) sebagai mikrokontroler utama *gateway* yang berfungsi untuk mengelola koneksi BLE, memproses *firmware* OTA, serta menangani komunikasi dengan sistem pemantauan. Mikrokontroler ini dipilih sebab performanya yang tinggi, kapasitas memorinya besar, serta kemampuan multitasking yang mendukung proses OTA dan komunikasi simultan.
- b) Antena Eksternal IPEX sebagai komponen komunikasi tambahan yang berfungsi untuk meningkatkan jangkauan dan stabilitas koneksi BLE antara *gateway* dan sensor *node*.
- c) LED dioda 5mm digunakan sebagai indikator visual dari status sistem yang dapat diamati secara langsung di lapangan. Terdapat 3 jenis LED yang digunakan yaitu berwarna biru, kuning, dan RGB yang mempunyai makna statusnya masing-masing.
- d) Resistor 220  $\Omega$  digunakan untuk penahan arus agar LED tidak putus.
- e) Catu daya *gateway* melalui USB atau adaptor daya yang berfungsi untuk memastikan *gateway* beroperasi secara stabil dan kontinu tanpa keterbatasan energi seperti sensor *node*.

Komponen-komponen perangkat keras pada *gateway* ini dapat dilihat spesifikasinya pada tabel 4.2 berikut:

Tabel 4.2 Spesifikasi Gateway

| No | Nama Komponen                            | Spesifikasi                                                                                                                                                                                                        |
|----|------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | ESP32-S3 DevKitC-1 WROOM-1 (Lexin N16R8) | <ul style="list-style-type: none"> <li>- Mikrokontroler Xtensa LX7 dual-core</li> <li>- BLE 5.0</li> <li>- Tegangan kerja 3.3 V</li> <li>- Flash memory internal</li> <li>- PSRAM (sesuai varian modul)</li> </ul> |
| 2  | Antena Eksternal IPEX                    | <ul style="list-style-type: none"> <li>- Tipe konektor IPEX/U.FL</li> </ul>                                                                                                                                        |



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|   |                                           |                                                                                                                                                                                       |
|---|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   |                                           | <ul style="list-style-type: none"> <li>- Frekuensi kerja 2.4 GHz</li> <li>- Mendukung BLE/Wi-Fi</li> <li>- Kabel IPEX</li> </ul>                                                      |
| 3 | Kabel USB Type-C & Adaptor Daya 5V        | <ul style="list-style-type: none"> <li>- Sumber daya USB / adaptor eksternal</li> <li>- Tegangan input 5 V</li> <li>- Regulasi tegangan on-board</li> </ul>                           |
| 4 | 3 Buah LED 5mm (Biru, kuning, RGB 4 kaki) | <ul style="list-style-type: none"> <li>- Sudut pandang 120 Derajat</li> <li>- Diameter 5mm</li> <li>- Intensitas Cahaya 1000-5000mcd</li> <li>- Tegangan 3V-3.2V atau 20mA</li> </ul> |
| 5 | 3 Buah Resistor 220 $\Omega$              | <ul style="list-style-type: none"> <li>- Nilai resistansi 220 Ohm</li> <li>- Toleransi <math>\pm 5\%</math></li> </ul>                                                                |

### 4.1.2 Analisis Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak dalam penelitian ini mencakup seluruh komponen sistem yang mendukung implementasi mekanisme Firmware Over-The-Air (FOTA), komunikasi Bluetooth Low Energy (BLE), pengelolaan data sensor, serta sistem monitoring berbasis web. Lebih lanjut, perangkat lunak yang dibutuhkan dalam penelitian ini dijelaskan pada tabel 4.3 berikut:

Tabel 4.3 Kebutuhan Perangkat Lunak

| No. | Perangkat Lunak                               | Fungsi                                                                                                                                                      |
|-----|-----------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1   | Espressif IoT Development Framework (ESP-IDF) | Framework utama pengembangan <i>firmware</i> pada sensor node dan <i>gateway</i> , termasuk pengelolaan perangkat keras, komunikasi BLE, dan mekanisme OTA. |
| 2   | NimBLE Stack                                  | Protokol komunikasi BLE yang digunakan dalam sistem yang berfungsi mengelola koneksi, pertukaran data, serta transfer <i>firmware</i> OTA.                  |



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

| No. | Perangkat Lunak                  | Fungsi                                                                                                                                                                |
|-----|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3   | Bahasa Pemrograman C/C++         | Digunakan sebagai bahasa pemrograman pengembangan <i>firmware embedded</i> pada sensor node dan gateway.                                                              |
| 4   | Go Programming Language (Golang) | Sebagai Bahasa pengembangan <i>backend</i> server menangani logika aplikasi, pemrosesan data sensor, serta integrasi dengan MQTT sebagai publisher dan basis data.    |
| 5   | EMQX MQTT Broker                 | Layanan <i>message broker</i> MQTT untuk mengelola komunikasi data agar server dapat memberikan notifikasi <i>firmware</i> baru ke <i>gateway</i> agar dapat diunduh. |
| 6   | PostgreSQL Database              | Digunakan untuk menyimpan data sensor, status perangkat, log OTA, serta informasi monitoring sistem                                                                   |
| 7   | Web Monitoring                   | Antarmuka pengguna untuk menampilkan data sensor, status perangkat, kapasitas baterai, serta halaman manajemen <i>firmware</i> .                                      |

#### 4.1.3 Analisis Kebutuhan Fungsional dan Non-Fungsional

Analisis kebutuhan sistem bertujuan untuk mengidentifikasi kebutuhan yang harus dipenuhi agar sistem yang dikembangkan dapat beroperasi sesuai dengan tujuan penelitian. Kebutuhan sistem pada penelitian ini dibagi menjadi kebutuhan fungsional dan kebutuhan non-fungsional.

##### 4.1.3.1 Kebutuhan Fungsional

Kebutuhan fungsional mendefinisikan layanan atau fungsi yang harus dimiliki oleh sistem. Pada penelitian ini, kebutuhan fungsional difokuskan pada fungsi monitoring sensor serta implementasi mekanisme FOTA, seperti pada tabel 4.4 berikut:

Tabel 4.4 Kebutuhan Fungsional Sistem



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

| No | Kebutuhan Fungsional                                                                                                                                                                             |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | <i>Node</i> sensor mampu melakukan akuisisi data sensor ADXL345 dan kapasitas baterai.                                                                                                           |
| 2  | <i>Node</i> sensor mampu mentransmisikan data sensor dan baterai ke <i>gateway</i> menggunakan BLE.                                                                                              |
| 3  | <i>Gateway</i> dapat meneruskan data ke server melalui protokol MQTT.                                                                                                                            |
| 4  | Sistem mampu menyimpan data sensor dan baterai ke dalam <i>database</i> .                                                                                                                        |
| 5  | Pengguna mampu melihat data sensor dan penggunaan baterai melalui web <i>monitoring</i> .                                                                                                        |
| 6  | Pengguna dapat mengelola versi <i>firmware</i> <i>node</i> sensor melalui <i>website</i> .                                                                                                       |
| 7  | <i>Gateway</i> mampu menerima perintah pembaruan dari server dan mengunduh berkas <i>firmware</i> secara utuh.                                                                                   |
| 8  | <i>Gateway</i> mampu memecah berkas <i>firmware</i> menjadi paket-paket kecil dan mengirimkannya secara berurutan ke <i>node</i> sensor melalui <i>Characteristic</i> BLE yang telah ditentukan. |
| 9  | <i>Node</i> sensor mampu menerima paket data <i>firmware</i> dari <i>gateway</i> dan menuliskannya secara utuh ke dalam partisi OTA pada memori <i>flash</i> .                                   |
| 10 | <i>Node</i> dapat mengatur <i>boot partition</i> ke partisi OTA yang baru dan melakukan proses restart secara otomatis untuk menjalankan <i>firmware</i> terbaru.                                |

#### 4.1.3.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan atribut kualitas yang menentukan perilaku, kinerja, keamanan, dan keandalan sistem, hal ini mencakup kecepatan, skalabilitas, dan kenyamanan pengguna, serta memastikan sistem beroperasi dengan baik. Kebutuhan non-fungsional dari penelitian ini dijelaskan pada tabel 4.5 berikut:

Tabel 4.5 Kebutuhan Non-Fungsional

| No. | Kebutuhan Non-Fungsional                                                                                                                                |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1   | <i>Node</i> sensor harus memiliki konsumsi daya rendah untuk mendukung operasi berbasis baterai                                                         |
| 2   | <i>Node</i> sensor dapat tetap berjalan seperti biasa jika proses pembaruan terputus di tengah jalan dengan cara melakukan booting ke versi sebelumnya. |



#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

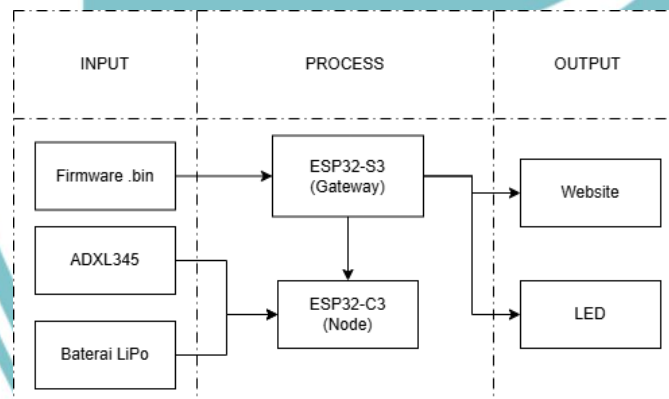
| No. | Kebutuhan Non-Fungsional                                                                                                            |
|-----|-------------------------------------------------------------------------------------------------------------------------------------|
| 3   | <i>Node</i> sensor dapat melakukan <i>checksum</i> sebelum memutuskan untuk mengeksekusi <i>firmware</i> baru demi integritas data. |
| 4   | Sistem dapat melakukan penambahan <i>node</i> sensor untuk skalabilitas.                                                            |
| 5   | <i>Web monitoring</i> mampu menampilkan data sensor dan baterai secara <i>real-time</i> .                                           |

## 4.2 Perancangan Sistem

Tahap perancangan sistem dalam penelitian ini bertujuan untuk memodelkan integrasi arsitektur perangkat keras dan perangkat lunak dalam membangun mekanisme pembaruan OTA pada jaringan sensor nirkabel berbasis BLE.

### 4.2.1 Blok Diagram

Blok diagram adalah representasi atau gambaran akan suatu sistem yang kompleks namun disederhanakan untuk mempermudah analisis. Membantu memvisualisasikan bagaimana sebuah sistem bekerja secara garis besar.



Gambar 4.1 Blok Diagram Sistem

Rancangan perangkat keras dan perangkat lunak dalam penelitian ini direpresentasikan melalui diagram blok dengan *input*, proses, *output*. Gambar 4.1 menunjukkan alur kerja sistem di mana pada bagian *input* terdapat *server* sebagai penyedia berkas *firmware* terbaru, sensor ADXL345 untuk membaca data pergerakan rangka atap baja, serta Baterai Li-Po yang bertindak sebagai sumber catu daya utama bagi *node* sensor. Data dan daya dari bagian masukan ini kemudian dikelola pada bagian proses, yang



#### Hak Cipta :

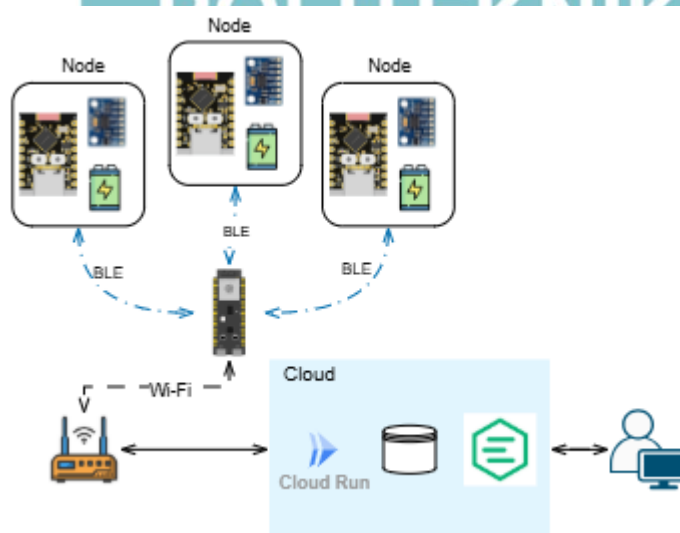
1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

merupakan inti komunikasi dua arah berbasis BLE. Pada tahap ini, mikrokontroler ESP32-S3 yang berperan sebagai *gateway* bertugas mengunduh dan mengirimkan paket *firmware* ke ESP32-C3 sebagai *node*, sementara *node* secara berkesinambungan mengakuisisi dan mengirimkan kembali data pembacaan sensor ADXL345 ke *gateway*.

Setelah melalui tahap pemrosesan dan pertukaran data antar mikrokontroler, hasil akhir dari operasional sistem akan diteruskan ke bagian *output*. Informasi yang berhasil dihimpun oleh *gateway* akan dikirimkan ke *website* yang berfungsi sebagai antarmuka pengguna. *Website* ini bertugas menampilkan visualisasi data dari sensor ADXL345 sekaligus memantau status keberhasilan proses pembaruan OTA pada tiap *node*. Selain antarmuka digital, sistem juga memanfaatkan komponen LED sebagai luaran fisik yang memberikan indikator visual di lapangan terkait status operasional perangkat, seperti indikasi berlangsungnya proses transmisi BLE maupun status konektivitas jaringan.

#### 4.2.2 Arsitektur Sistem

Arsitektur sistem adalah cetak biru dari struktur sebuah sistem yang mendefinisikan komponen-komponen utama, serta bagaimana komponen tersebut saling berinteraksi.



Gambar 4.2 Arsitektur Sistem

Arsitektur jaringan sistem pemantauan dan pembaruan *firmware* secara keseluruhan diilustrasikan pada Gambar 4.2. Setiap *node* beroperasi di dalam jaringan sensor nirkabel sebagai perangkat BLE *peripheral* yang bertugas mentransmisikan data



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

pembacaan sensor menuju satu perangkat pusat, yakni *Gateway*. Komunikasi pada lapisan lokal ini secara eksklusif memanfaatkan protokol BLE yang direpresentasikan melalui garis putus-putus berwarna biru, memungkinkan pertukaran data dua arah secara efisien tanpa menguras daya baterai *node*.

Pada lapisan jaringan yang lebih luas, *gateway* bertindak sebagai jembatan penghubung antara ekosistem BLE lokal dengan *server cloud* melalui koneksi internet berbasis *Wi-Fi*. Di dalam arsitektur awan tersebut, data sensor yang diterima akan diproses, disimpan ke dalam basis data, dan divisualisasikan kepada pengguna akhir melalui sebuah *dashboard*. Selain menangani aliran data sensor ke atas, arsitektur infrastruktur ini juga dirancang untuk mengelola aliran perintah ke bawah dalam proses pembaruan sistem. Pengguna dapat mengunggah berkas *firmware* terbaru melalui *cloud*, yang selanjutnya akan diunduh oleh *gateway* via *Wi-Fi* dan didistribusikan secara bergiliran kepada tiap *node* sensor target menggunakan mekanisme OTA berbasis koneksi BLE.

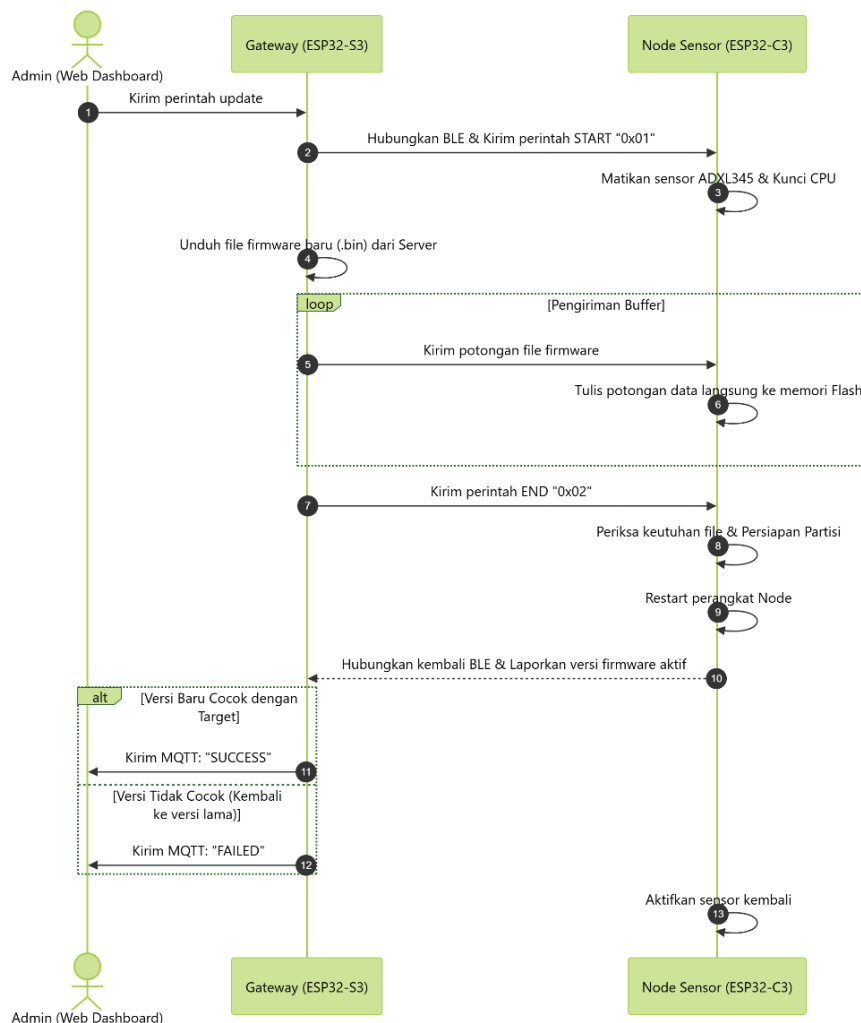
### 4.2.3 Diagram Alir

Diagram alir menggambarkan urutan langkah suatu proses, cara berpikir, atau algoritma dari suatu sistem yang berfungsi untuk memetakan proses yang rumit menjadi urutan yang lebih ringkas dan mudah dipahami



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 4.3 Diagram Sekuensial Proses FOTA

Proses pembaruan *firmware* diperlihatkan pada Gambar 4.3 diawali dengan pengiriman perintah dari admin melalui web menuju *gateway*, yang kemudian ditindaklanjuti dengan menginisiasi koneksi BLE dan mengirimkan perintah mulai dengan *byte* "0x01" ke *Node Sensor*. Setelah menerima sinyal inisiasi tersebut, *node sensor* segera menonaktifkan sensor akselerometer ADXL345 dan mengunci kerja CPU agar tetap aktif dalam performa penuh guna mencegah masuk ke mode daya rendah selama FOTA berlangsung. Secara paralel, *Gateway* mengunduh berkas biner dari *server* dan memancarkannya kembali secara bertahap kepada *node sensor* di dalam struktur *looping* pengiriman *buffer*. Setiap potongan data *firmware* yang diterima melalui karakteristik BLE akan dituliskan langsung oleh *node* ke dalam memori *flash* internalnya, hingga akhirnya *gateway* mengirimkan perintah selesai

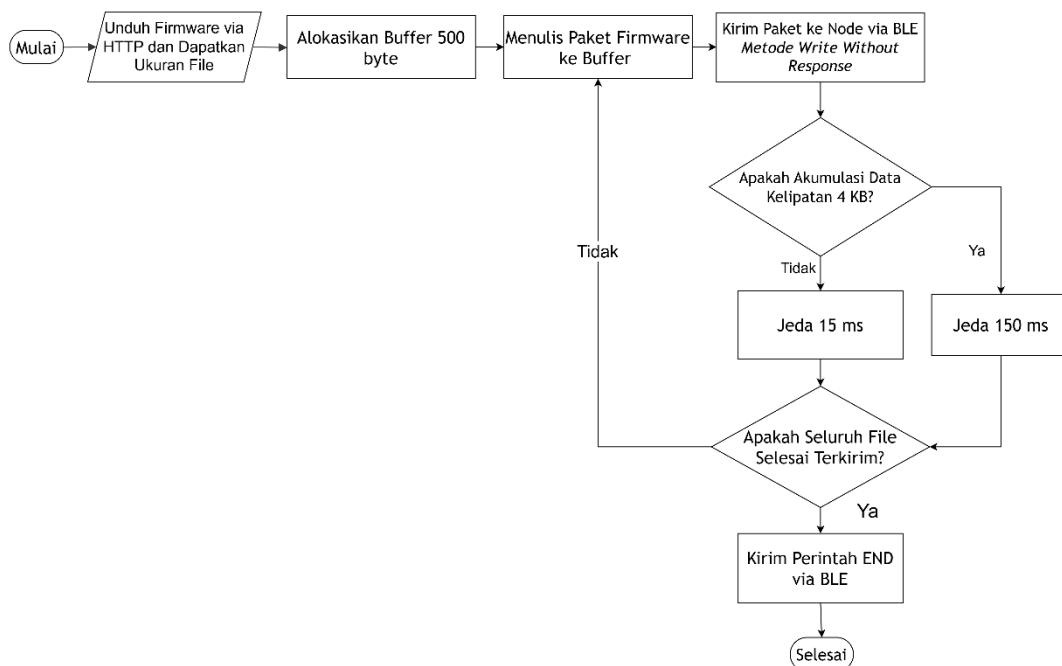


### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

berstatus *byte* “0x02” sebagai penanda bahwa seluruh paket data biner telah selesai ditransfer.

Setelah perintah selesai diterima, *node* sensor melakukan validasi mandiri untuk memeriksa keutuhan berkas dan mempersiapkan tabel partisi memori sebelum melakukan proses *restart*. Begitu sistem menyala kembali, *node* Sensor akan secara otomatis menghubungkan kembali jaringan BLE-nya dengan *gateway* dan melaporkan versi *firmware* yang saat ini aktif. *Gateway* kemudian mencocokkan versi tersebut dengan target pembaruan, apabila versi baru telah sesuai, *gateway* mengirimkan notifikasi MQTT berstatus “SUCCESS” ke *web dashboard*, sedangkan jika terjadi kegagalan atau ketidakcocokan versi, *gateway* akan mengirimkan status “FAILED” yang menandakan bahwa sistem melakukan *rollback* ke versi *firmware* lama. Proses FOTA ini diakhiri dengan aktivitas *node* sensor yang menyalakan kembali sensor ADXL345 untuk beroperasi normal.



Gambar 4.4 Diagram Alir Fragmentasi dan Transmisi Data

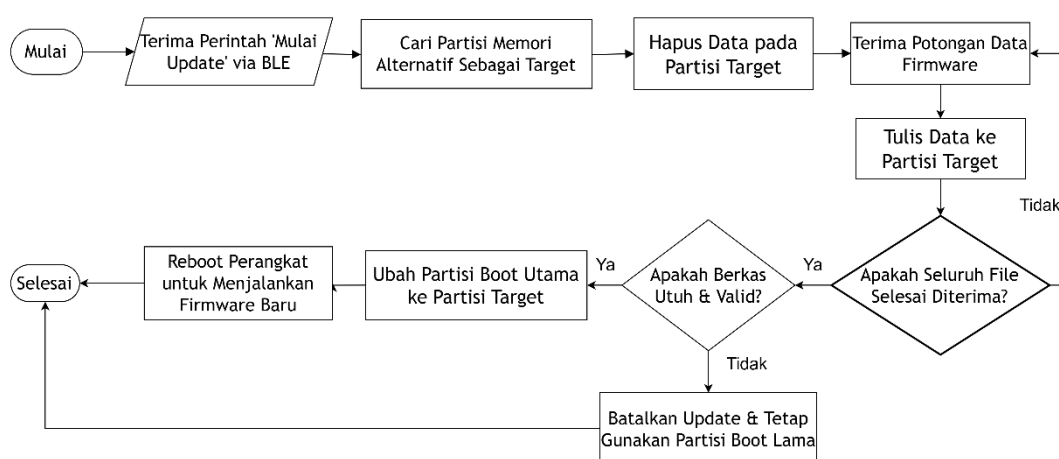
Proses fragmentasi dan transmisi data *firmware* pada *gateway* seperti yang terlihat pada Gambar 4.4 diawali dengan pengunduhan berkas biner melalui protokol HTTP guna memperoleh informasi ukuran total, yang kemudian dilanjutkan dengan pemecahan berkas menjadi potongan paket kecil di dalam *buffer* memori dengan



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

kapasitas maksimal 500 *byte* sesuai dengan negosiasi batas ukuran MTU BLE. Paket-paket tersebut kemudian dialirkan ke *node* penerima secara beruntun menggunakan metode transmisi *Write Without Response* untuk memangkas latensi komunikasi, di mana setelah setiap pengiriman paket diberikan jeda singkat selama 15 ms untuk stabilisasi frekuensi radio BLE. Sistem menerapkan mekanisme *flow control* berbasis akumulasi data untuk mencegah terjadinya luapan *buffer* pada sisi penerima, jika total data yang terkirim mencapai kelipatan 4 KB, transmisi akan ditangguhkan dengan jeda selama 150 ms untuk memberikan waktu bagi *node* menuliskan data *buffer* RAM ke dalam sektor memori fisik. Siklus pengiriman dan penundaan ini terus berulang hingga seluruh bagian berkas selesai dikirimkan secara utuh, yang diakhiri dengan pengiriman perintah selesai berupa kode *byte* 0x02 via BLE sebagai penanda akhir sesi transfer data *firmware*.



Gambar 4.5 Diagram Alir Penerimaan Data dan Manajemen Partisi

Prosedur penerimaan data FOTA dan manajemen partisi memori pada sisi *node* sensor ditunjukkan pada Gambar 4.5. Alur kerja diawali dengan penerimaan sinyal inisiasi pembaruan via BLE, yang memicu *node* untuk mendeteksi dan mencari partisi memori cadangan yang sedang tidak aktif untuk kemudian dilakukan penghapusan sektor memori. Selama fase transfer data berlangsung, *node* sensor menerima potongan berkas *firmware* secara bertahap dan menyimpannya secara sekuensial ke dalam memori *flash* pada partisi target hingga seluruh berkas selesai ditransmisikan. Setelah transmisi selesai, *node* sensor melakukan pengujian integritas berkas dan jika terdeteksi kerusakan berkas, sistem akan membatalkan pembaruan dan tetap



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

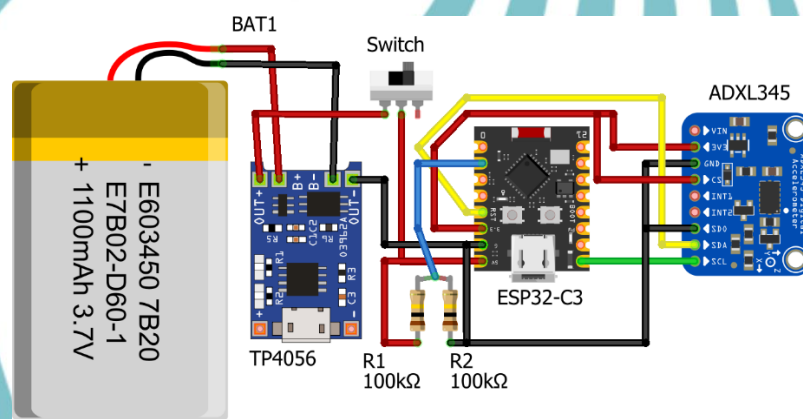
menggunakan partisi *boot* lama, namun jika berkas terbukti valid, penunjuk *boot* utama akan dialihkan ke partisi target baru dan diikuti dengan proses *boot* ulang perangkat guna memuat versi *firmware* yang baru.

#### 4.2.4 Rancangan Skematik Perangkat

Rancangan skematik perangkat adalah diagram dari suatu sistem perangkat keras yang menggunakan simbol-simbol atau ikon komponen untuk menunjukkan bagaimana komponen-komponen tersebut saling terhubung.

##### 4.2.4.1 Skematik *Node* Sensor

Skematik *node* sensor dibuat untuk merancang jalur kabel rangkaian setiap komponen dan mendapat gambaran realisasi perangkat keras *node* nantinya.



Gambar 4.6 Skematik *Node* Sensor

Gambar 4.6 mengilustrasikan rancangan arsitektur perangkat keras untuk *node* sensor pada sistem jaringan sensor nirkabel. Perangkat ini dirancang agar dapat beroperasi secara nirkabel dan mandiri dengan efisiensi daya yang optimal. Sistem ini berpusat pada mikrokontroler ESP32-C3 SuperMini yang bertugas mengakuisisi data getaran dari lingkungan sekitar, memantau kapasitas daya baterai, serta mengirimkan data tersebut ke *gateway* menggunakan protokol BLE. Secara garis besar, rancangan perangkat keras ini terdiri dari empat subsistem utama:

##### a) Subsistem Manajemen Daya

Sumber energi utama pada *node* sensor menggunakan baterai Lithium-Polymer 3.7V berkapasitas 1100mAh. Untuk menjaga siklus hidup baterai, digunakan modul pengisi daya TP4056 yang telah terintegrasi dengan sirkuit proteksi



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

tegangan. Jalur keluaran daya dari baterai melalui pin OUT+ TP4056 dihubungkan secara seri dengan sebuah sakelar *Single Pole Double Throw* (SPDT) yang berfungsi sebagai pemutus arus utama menuju mikrokontroler. Konfigurasi ini memungkinkan pengisian daya baterai tetap dapat dilakukan meskipun *node* sensor dalam keadaan mati. Tabel 4.6 memperlihatkan jalur subsistem ini

Tabel 4.6 Pemetaan Jalur Subsistem Manajemen Daya

| Titik Asal        | Pin Asal          | Titik Tujuan   | Pin Tujuan |
|-------------------|-------------------|----------------|------------|
| Baterai LiPo 3.7V | Kutub Positif (+) | Modul TP4056   | B+         |
| Baterai LiPo 3.7v | Kutub Negatif (-) | Modul TP4056   | B-         |
| Modul TP4056      | OUT+              | Sakelar (SPDT) | Pin 1      |

b) Subsistem Pengolahan Data dan Komunikasi

Sebagai unit pemrosesan utama, digunakan ESP32-C3 SuperMini. Jalur daya positif yang telah melewati sakelar dihubungkan ke pin 5V pada ESP32-C3. Mikrokontroler ini dilengkapi dengan *Low Dropout Regulator* (LDO) internal yang akan meregulasi tegangan menjadi 3.3V, tegangan operasi standar yang aman bagi *chip* dan sensor. Mikrokontroler ini juga bertanggung jawab untuk mengeksekusi mekanisme pembaruan *firmware* nirkabel atau FOTA yang dikirimkan melalui jaringan BLE. Tabel 4.7 memperlihatkan jalur subsistem ini

Tabel 4.7 Pemetaan Jalur Subsistem Pengolahan Data

| Titik Asal     | Pin Asal  | Titik Tujuan       | Pin Tujuan |
|----------------|-----------|--------------------|------------|
| Sakelar (SPDT) | Kaki ke-2 | ESP32-C3 SuperMini | 5V         |
| Modul TP4056   | OUT-      | ESP32-C3 SuperMini | GND        |

c) Subsistem Akuisisi Data Getaran



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Pemantauan getaran struktural dilakukan menggunakan sensor akselerometer ADXL345. Sensor ini dikonfigurasi untuk beroperasi menggunakan antarmuka komunikasi I2C (*Inter-Integrated Circuit*) guna meminimalisir penggunaan pin GPIO pada mikrokontroler. Tegangan kerja sensor disuplai langsung dari pin 3V3 ESP32-C3. Untuk mengaktifkan mode I2C, pin CS (*Chip Select*) pada sensor dihubungkan ke 3.3V, sedangkan pin SDO dihubungkan ke GND untuk menetapkan alamat perangkat I2C secara perangkat keras pada nilai 0x53. Jalur data SDA dan SCL dihubungkan masing-masing ke pin GPIO mikrokontroler yang telah ditentukan. Tabel 4.8 memperlihatkan jalur subsistem ini

Tabel 4.8 Pemetaan Jalur Subsistem Akuisisi Data Getaran

| Titik Asal         | Pin Asal | Titik Tujuan  | Pin Tujuan |
|--------------------|----------|---------------|------------|
| ESP32-C3 SuperMini | 3V3      | Modul ADXL345 | VCC        |
| ESP32-C3 SuperMini | GND      | Modul ADXL345 | GND        |
| ESP32-C3 SuperMini | GPIO 4   | Modul ADXL345 | SDA        |
| ESP32-C3 SuperMini | GPIO 5   | Modul ADXL345 | SCL        |
| ESP32-C3 SuperMini | 3V3      | Modul ADXL345 | CS         |
| ESP32-C3 SuperMini | GND      | Modul ADXL345 | SDO        |

d) Subsistem Pemantauan Tegangan Baterai

Untuk mengetahui status sisa daya baterai, sistem dilengkapi dengan rangkaian pembagi tegangan atau *voltage divider* yang tersusun dari dua buah resistor 100k $\Omega$ . Rangkaian ini mengambil titik referensi tegangan langsung dari jalur daya baterai sebelum masuk ke mikrokontroler. Pembagian tegangan ini diperlukan karena pin *Analog-to-Digital Converter* (ADC) pada ESP32-C3 memiliki batas maksimal toleransi tegangan, sehingga tegangan baterai maksimal yang mencapai 4.2V dapat diturunkan hingga setengahnya agar



#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

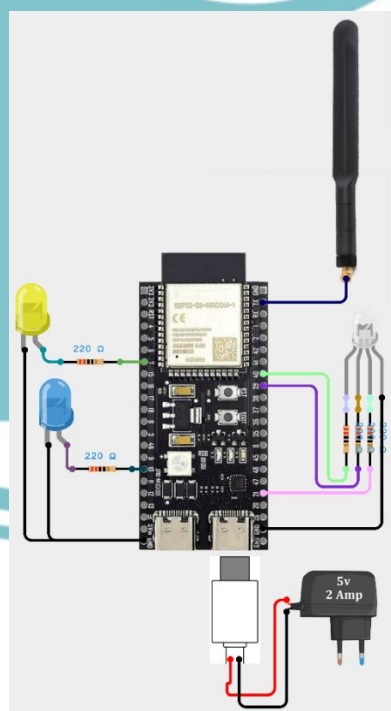
aman untuk dibaca oleh pin ADC. Hasil pembacaan nilai digital ini kemudian dikonversi kembali menjadi persentase baterai di dalam perangkat lunak. Tabel 4.9 memperlihatkan jalur subsistem ini.

Tabel 4.9 Pemetaan Jalur Subsistem Pemantauan Tegangan Baterai

| Titik Asal     | Pin Asal     | Titik Tujuan                  | Pin Tujuan |
|----------------|--------------|-------------------------------|------------|
| Resistor R1    | Kaki Bawah   | Resistor R2 (100 k $\Omega$ ) | Kaki Atas  |
| Simpul R1 & R2 | Titik Tengah | ESP32-C3 SuperMini            | GPIO 1     |
| Resistor R2    | Kaki Bawah   | ESP32-C3 SuperMini            | GND        |

#### 4.2.4.2 Skematik Gateway

Skematik *gateway* dibuat untuk merancang jalur kabel rangkaian setiap komponen dan mendapat gambaran realisasi perangkat keras *gateway* nantinya.



Gambar 4.7 Skematik *Gateway*

Gambar 4.7 memperlihatkan rancangan perangkat keras untuk *gateway* pada jaringan sensor nirkabel. Berbeda dengan *node* Sensor yang beroperasi menggunakan sumber



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

daya baterai secara terbatas, *gateway* dirancang untuk beroperasi secara kontinu sebagai pusat kendali lalu lintas data. Mikrokontroler utama yang digunakan adalah ESP32-S3. Perangkat ini berfungsi ganda sebagai jembatan jaringan yang mengoordinasikan komunikasi BLE dengan *node* Sensor dan komunikasi *Wi-Fi* atau MQTT dengan server *cloud*.

Secara garis besar, perancangan perangkat keras Gateway difokuskan pada dua subsistem utama:

a) Subsistem Daya dan Komunikasi Inti

Sistem *gateway* tidak mengimplementasikan manajemen baterai, melainkan ditenagai secara langsung menggunakan adaptor daya eksternal yang menyuplai tegangan 5V DC melalui konektor USB. Tegangan masukan ini dihubungkan langsung ke pin 5V pada mikrokontroler dan diregulasi secara internal oleh *Low Dropout Regulator* (LDO) menjadi 3.3V. *Gateway* dihubungkan dengan antena eksternal melalui konektor antarmuka IPEX/U.FL untuk memaksimalkan stabilitas jangkauan transmisi data nirkabel saat proses FOTA berlangsung.

b) Subsistem Indikator Visual

*Light Emitting Diode* (LED) digunakan untuk memberikan umpan balik visual kepada pengguna di lapangan tanpa harus bergantung pada *dashboard web*. Subsistem ini memanfaatkan satu buah LED berwarna Kuning, satu buah LED berwarna Biru, dan satu buah LED RGB 4-kaki bertipe *Common Cathode*. Setiap pin anoda pada LED dihubungkan dengan resistor pembatas arus sebesar 220Ω menuju pin *General-Purpose Input/Output* (GPIO) untuk mencegah terjadinya kelebihan beban arus. Secara perangkat lunak, LED Kuning dan Biru dikonfigurasi untuk merepresentasikan aktivitas jaringan *Wi-Fi* dan BLE, sedangkan kombinasi warna pada LED RGB dimanfaatkan untuk memvisualisasikan status keberhasilan, proses transfer, atau indikasi kegagalan pada saat mekanisme pembaruan FOTA dieksekusi. Tabel 4.10 memperlihatkan jalur subsistem *gateway*.

Tabel 4.10 Pemetaan Jalur Subsistem Gateway



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

| Titik Asal         | Pin/Jalur Asal   | Titik Tujuan       | Pin/Jalur Tujuan     |
|--------------------|------------------|--------------------|----------------------|
| ESP32-S3 WROOM-1   | Port U.FL / IPEX | Antena Eksternal   | Konektor Antena      |
| ESP32-S3 WROOM-1   | GPIO 7           | Resistor R1 (220Ω) | Kaki Atas            |
| Resistor R1 (220Ω) | Kaki Bawah       | LED Kuning 5mm     | Anoda (+)            |
| ESP32-S3 WROOM-1   | GPIO 10          | Resistor R2 (220Ω) | Kaki Atas            |
| Resistor R2 (220Ω) | Kaki Bawah       | LED Biru 5mm       | Anoda (+)            |
| ESP32-S3 WROOM-1   | GPIO 39          | Resistor R3 (220Ω) | Kaki Atas            |
| Resistor R3 (220Ω) | Kaki Bawah       | LED RGB 4-Kaki     | Pin <i>Red</i> (R)   |
| ESP32-S3 WROOM-1   | GPIO 40          | Resistor R4 (220Ω) | Kaki Atas            |
| Resistor R4 (220Ω) | Kaki Bawah       | LED RGB 4-Kaki     | Pin <i>Green</i> (G) |
| ESP32-S3 WROOM-1   | GPIO 42          | Resistor R5 (220Ω) | Kaki Atas            |
| Resistor R5 (220Ω) | Kaki Bawah       | LED RGB 4-Kaki     | Pin <i>Blue</i> (B)  |
| ESP32-S3 WROOM-1   | GND              | Semua LED          | Katoda (-)           |

#### 4.2.5 Rancangan Komunikasi Data

Sistem pada penelitian ini menggunakan arsitektur komunikasi dua tingkat untuk menjembatani perangkat *node* sensor dengan *server*. Komunikasi tingkat pertama menggunakan protokol BLE antara *node* sensor dan *gateway*, sedangkan komunikasi tingkat kedua menggunakan protokol *Message Queuing Telemetry Transport* (MQTT) antara *gateway* dan *server*.

##### 4.2.5.1 Rancangan Protokol MQTT

Protokol MQTT dipilih karena bersifat ringan dan efisien untuk pengiriman data telemetri secara *real-time*. Pada sistem ini, *gateway* bertindak sebagai MQTT *client* yang terhubung ke EMQX *Broker* secara aman menggunakan enkripsi TLS.

Komunikasi data MQTT pada sistem ini dirancang menggunakan format *JavaScript Object Notation* (JSON) agar mudah diproses oleh *backend*. Topik dan muatan yang dirancang dibagi berdasarkan fungsinya, sebagaimana ditunjukkan pada Tabel 4.11 berikut:

Tabel 4.11 Rancangan Topik dan Payload MQTT

| Topik MQTT         | Arah Publikasi   | Deskripsi dan Contoh Payload JSON                                                                                                                                                                                                                                              |
|--------------------|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| shm/gateway/status | Gateway → Server | Digunakan untuk melaporkan status online/offline gateway melalui mekanisme Last Will and Testament (LWT) dan Keepalive.<br><br>Payload: {"status":"online", "gateway_mac":"XX:XX:XX:XX:XX:XX"}                                                                                 |
| shm/telemetry      | Gateway → Server | Berisi aliran data telemetry pembacaan akselerometer (sumbu X, Y, Z) dan tegangan baterai dari node sensor.<br><br>Payload: {"node_mac":"...", "ax":-105.99, "ay":65.89, "az":0.00, "vbatt":3.81}                                                                              |
| Shm/ota/status     | Gateway → Server | Digunakan untuk melaporkan proses berjalannya OTA pada node tertentu, mulai dari status IN_PROGRESS, REBOOTING, SUCCESS, hingga FAILED.<br><br>Payload: {"cmd":"ota_status", "node_mac":"...", "status":"REBOOTING"}                                                           |
| shm/ota/trigger    | Server → Gateway | Topik yang di-subscribe oleh Gateway untuk menerima perintah inisiasi OTA dari admin web dashboard. Berisi URL firmware terenkripsi (Signed URL GCS) dan versi target.<br><br>Payload: {"cmd":"start_ota", "target_mac":"...", "url":"https://...", "target_version":"v1.0.1"} |

#### 4.2.5.2 Rancangan Profil GATT BLE

Komunikasi tingkat lokal antara *gateway* dan *node* dirancang menggunakan arsitektur *Generic Attribute Profile* (GATT) dari protokol BLE. Pada rancangan ini, *node* sensor bertindak sebagai *GATT Server* yang menyediakan layanan data, sedangkan *gateway* bertindak sebagai *GATT Client* yang membaca dan menulis data ke *node*.

Untuk menghindari tabrakan GATT dan pemotongan paket data secara diam-diam akibat keterbatasan ukuran paket bawaan BLE, sistem ini dirancang dengan mekanisme antrian atau *callback chaining* dan negosiasi *Maximum Transmission Unit* (MTU) sebesar 512 *bytes*. Hal ini memungkinkan *firmware* dikirim dalam ukuran chunk 500 *bytes* per paket.



#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Struktur layanan dan karakteristik BLE yang dibangun secara kustom dengan *Universally Unique Identifier* dapat dilihat pada Tabel 4.12 berikut:

Tabel 4.12 Rancangan Profil GATT BLE Kustom

| UUID Kustom (128-bit)                | Tipe Atribut          | Properti (Hak Akses)          | Fungsi dan Deskripsi                                                                                                                                                                |
|--------------------------------------|-----------------------|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 12345678-1234-1234-1234-1234567890ab | <i>Service</i>        | -                             | Layanan utama Structure Health Monitoring                                                                                                                                           |
| abcd1234-5678-90ab-cdef-1234567890ab | <i>Characteristic</i> | <i>Notify</i>                 | Node mengirimkan stream data akselerometer (AX, AY, AZ) dan Vbatt ke Gateway setiap 1 detik tanpa diminta secara terus-menerus.                                                     |
| 12345678-0000-0000-0000-1234567890ab | <i>Service</i>        | -                             | Layanan pembaruan perangkat lunak (FOTA)                                                                                                                                            |
| abcd1234-0001-0000-0000-1234567890ab | <i>Characteristic</i> | <i>Write</i>                  | Karakteristik Kontrol. Gateway mengirimkan instruksi 0x01 untuk memulai proses penulisan partisi baru, dan 0x02 untuk mengakhiri.                                                   |
| abcd1234-0002-0000-0000-1234567890ab | <i>Characteristic</i> | <i>Write without Response</i> | Pipa Data Firmware. Menerima potongan file binari firmware sebesar 500 bytes secara cepat (dikirim setiap 15ms) tanpa kewajiban node memberikan sinyal acknowledge.                 |
| abcd1234-0003-0000-0000-1234567890ab | <i>Characteristic</i> | <i>Read</i>                   | Karakteristik Validasi Versi. Gateway akan membaca karakteristik ini setelah proses reboot untuk memastikan bahwa Node telah menjalankan firmware dengan versi terbaru yang dituju. |

#### 4.2.6 Rancangan Partisi Memori

Implementasi mekanisme FOTA memerlukan modifikasi pada struktur penyimpanan memori *flash* mikrokontroler. Pada penelitian ini, sistem menggunakan arsitektur partisi ganda atau *dual-bank partition* yang mengalokasikan dua ruang terpisah untuk menyimpan aplikasi. Arsitektur ini menjamin keandalan sistem sehingga apabila proses pembaruan gagal atau terputus di tengah jalan, perangkat tidak akan mengalami



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

keadaan mati total karena sistem selalu memiliki satu partisi aplikasi cadangan yang utuh untuk dimulai ulang.

Mengingat terbatasnya kapasitas memori *flash* internal pada ESP32-C3 SuperMini yaitu sebesar 4MB, tabel partisi standar bawaan ESP-IDF dimodifikasi untuk menghilangkan partisi bawaan. Penghapusan partisi bawaan bertujuan untuk menghemat ruang memori, sehingga partisi OTA dapat diperbesar menjadi masing-masing 1.5 MB guna mengakomodasi *firmware* dengan fitur BLE dan komunikasi sensor yang ukurannya cukup masif.

Rancangan struktur memori tersebut didefinisikan secara khusus melalui *file* konfigurasi kustom dengan alokasi yang ditunjukkan pada Tabel 4.13 berikut:

Tabel 4.13 Tabel Partisi Memori Flash ESP32-C3

| Nama Partisi | Tipe | Sub-Tipe | Ukuran (Bytes)   | Fungsi dan Keterangan                                                                                                                                                                           |
|--------------|------|----------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| nvs          | data | nvs      | 0x4000<br>(16KB) | Area penyimpanan data persisten untuk sistem dan pengguna, seperti parameter kalibrasi jaringan dan penyimpanan kredensial sementara.                                                           |
| otadata      | data | ota      | 0x2000<br>(8KB)  | Partisi paling krusial dalam mekanisme FOTA yang berfungsi sebagai penunjuk bagi <i>bootloader</i> untuk menentukan partisi aplikasi mana yang harus dieksekusi saat mikrokontroler dinyalakan. |
| phy_init     | data | phy      | 0x1000<br>(4KB)  | Partisi untuk menyimpan konfigurasi fisik antena RF ( <i>Radio Frequency</i> ), berfungsi untuk stabilitas pancaran sinyal BLE perangkat.                                                       |
| ota_0        | app  | ota_0    | 1500K<br>(1.5MB) | Tempat penyimpanan kode program berbentuk binari yang dapat dieksekusi.                                                                                                                         |
| ota_1        | app  | ota_1    | 1500K<br>(1.5MB) | Tempat penyimpanan kode program (binari) alternatif yang ukurannya simetris dengan ota_0                                                                                                        |



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Proses pembaruan *firmware* pada arsitektur di atas dioperasikan melalui mekanisme partisi aktif dan pasif. Algoritma penulisan memori dirancang sebagai berikut:

1. Ketika *gateway* mulai mengirimkan *firmware* baru, fungsi API NimBLE OTA pada *node* akan mengecek partisi mana yang sedang aktif berjalan.
2. Jika sistem saat ini berjalan di *ota\_0*, maka paket *chunk* 500 bytes dari protokol BLE akan ditulis secara sekuensial ke dalam partisi pasif *ota\_1*. Partisi *ota\_0* tidak disentuh, sehingga jika di tengah proses pembaruan terjadi kehilangan daya, perangkat tetap dapat beroperasi normal menggunakan *ota\_0*.
3. Setelah proses penulisan seluruh paket selesai dan tervalidasi keutuhannya, *bootloader* akan memodifikasi isi dari partisi *otadata*.
4. Mikrokontroler melakukan *reboot* otomatis, *bootloader* akan membaca instruksi baru dari *otadata* yang kini menunjuk ke *ota\_1* sebagai partisi aktif. Proses FOTA dinyatakan sukses dan *firmware* baru mulai dieksekusi.

#### 4.2.7 Rancangan Basis Data

Rancangan basis data pada sistem ini menggunakan *Relational Database Management System* (RDBMS) PostgreSQL. Basis data berfungsi untuk menyimpan data telemetri dari *node* sensor secara persisten, mengelola inventarisasi perangkat, merekam pustaka *firmware*, serta mencatat log riwayat eksekusi FOTA. Struktur relasi antar-tabel dirancang untuk menjamin integritas data melalui penerapan *Primary Key* (PK) dan *Foreign Key* (FK).

##### 4.2.7.1 Entity Relationship Diagram

Hubungan logika antar-entitas di dalam sistem dikelompokkan ke dalam 5 tabel utama yang saling berelasi. Alur relasi tersebut dirancang agar satu versi *firmware* dapat diimplementasikan ke banyak *node*, dan satu *node* dapat memiliki banyak rekam jejak pembaruan serta riwayat data sensor. Visualisasi relasi antar-tabel tersebut dapat dilihat pada Gambar 4.8.



Hubungan logika antar-entitas di dalam sistem dikelompokkan ke dalam 5 tabel utama: users, nodes, firmwares, ota\_logs, dan sensor\_data. Penjelasan rinci mengenai relasi dari ERD tersebut adalah sebagai berikut:

- #### 4.2.7.2 Deskripsi Tabel Basis Data

Untuk mendukung arsitektur relasional pada Gambar 4.8 di atas, fungsionalitas dari masing-masing tabel didefinisikan pada Tabel 4.14 berikut:



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Tabel 4.14 Deskripsi Tabel Basis Data

| Nama Tabel      | Deskripsi                                                                                                                                                                           |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| users           | Menyimpan data kredensial akun administrator seperti <i>email</i> dan kata sandi terenkripsi yang memiliki hak akses untuk masuk ke <i>dashboard</i> web dan memicu perintah FOTA.  |
| nodes           | Mencatat inventaris perangkat keras <i>node sensor</i> ESP32-C3 yang terdaftar di dalam jaringan sensor nirkabel beserta informasi alamat fisik MAC Address perangkat.              |
| firmwares       | Berfungsi sebagai katalog pustaka <i>firmware</i> yang memuat informasi mengenai versi, ukuran file biner, catatan rilis, dan waktu unggah <i>firmware</i> ke sistem.               |
| ota_logs        | Mencatat riwayat transaksi proses pembaruan <i>firmware</i> secara nirkabel, mencakup target <i>node</i> , versi <i>firmware</i> , admin yang memicu, serta status akhir.           |
| shm_telemetries | Menampung aliran data telemetri yang dikirimkan oleh <i>node</i> via <i>gateway</i> , meliputi data percepatan getaran 3-sumbu dari sensor ADXL345 dan nilai tegangan sisa baterai. |

#### 4.2.7.3 Pengelolaan Data *Time-Series* Menggunakan TimescaleDB

Perancangan basis data pada sistem ini dioptimalkan menggunakan TimescaleDB, sebuah ekstensi *open-source* pada PostgreSQL yang dirancang secara spesifik untuk beban kerja *time-series*. Pada implementasinya, tabel *shm\_telemetries* dikonversi dari tabel standar menjadi sebuah arsitektur *Hypertable*. Secara konseptual, *Hypertable* dipresentasikan sebagai tabel tunggal pada lapisan antarmuka aplikasi atau *backend*. Namun di belakang layar, arsitektur ini secara otomatis mempartisi atau memecah data ke dalam bongkahan-bongkahan fisik yang lebih kecil berdasarkan dimensi interval waktu pada kolom *created\_at*.

Pendekatan pemecahan otomatis ini memberikan tiga keunggulan komputasional utama bagi sistem yang dibangun:

1. Dengan memecah data menjadi *chunks*, TimescaleDB memastikan bahwa indeks waktu dari data terbaru selalu termuat di dalam memori utama. Hal ini memungkinkan *gateway* untuk melakukan insersi ribuan paket data sensor secara terus-menerus tanpa mengalami hambatan pada penyimpanan fisik.



#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

2. Saat antarmuka *web* memuat grafik visualisasi fluktuasi getaran rangka atap, eksekutor kueri basis data hanya akan memindai memori pada *chunk* waktu yang relevan, alih-alih melakukan pemindaian menyeluruh terhadap seluruh riwayat data perangkat.
3. Arsitektur ini memudahkan penerapan kebijakan retensi di masa mendatang. Data historis pemantauan struktur yang sudah usang dapat dihapus dengan melepaskan *chunk* secara utuh, yang jauh lebih ringan dan cepat dibandingkan proses *delete* baris per baris pada tabel relasional konvensional.

### 4.3 Implementasi Sistem

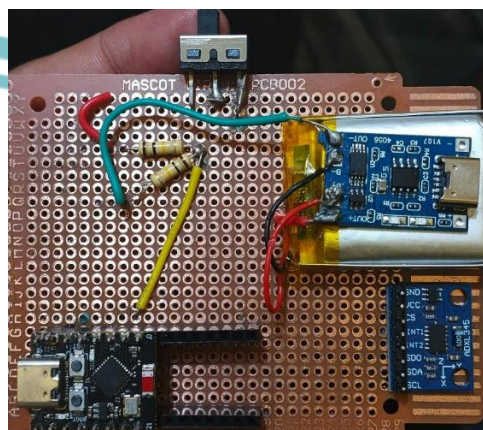
Implementasi sistem adalah tahapan realisasi atau penerapan dari rancangan yang telah dibuat pada tahap sebelumnya. Implementasi pada penelitian ini meliputi implementasi perangkat keras dan perangkat lunak mikrokontroler, implementasi broker MQTT, implementasi *backend* dan basis data, dan implementasi antarmuka atau *web*.

#### 4.3.1 Implementasi Perangkat Keras

Tahap implementasi perangkat keras merupakan realisasi fisik dari rancangan skematik yang telah dijabarkan pada sub-bab sebelumnya. Implementasi perangkat keras ini dibagi menjadi dua unit fungsional utama, yaitu *node* sensor dan *gateway*.

##### 4.3.1.1 Implementasi Node Sensor

Implementasi perangkat keras *node* sensor dilakukan dengan cara menyolder secara permanen komponen-komponen yang dibutuhkan secara terintegrasi pada sebuah papan PCB berukuran 7x9 cm.



Gambar 4.10 Implementasi Node Sensor



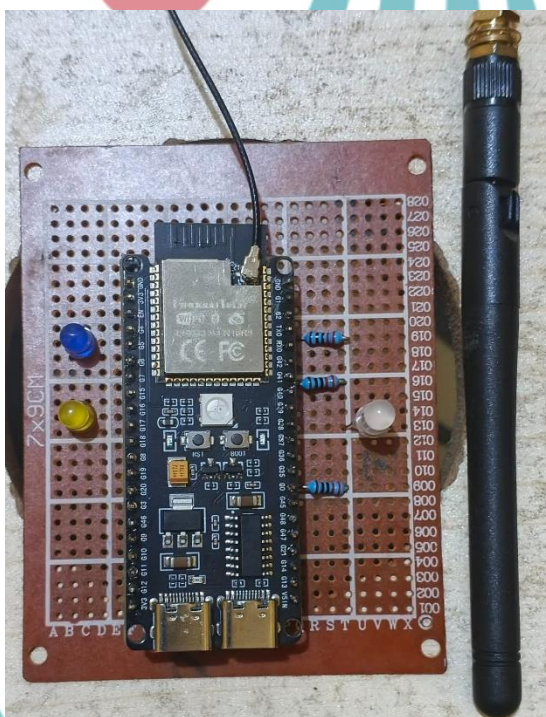
#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Berdasarkan realisasi fisik pada Gambar 4.10, perakitan perangkat keras *node* sensor dilakukan dengan menyolder secara permanen seluruh komponen di atas *Printed Circuit Board* atau PCB jenis *dot matrix*. Modul sensor ADXL345 ditancapkan menggunakan *male header* yang disolder menembus PCB, menjamin posisi sensor tetap kaku dan mencegah anomali pembacaan data akibat efek ayunan komponen. Jalur komunikasi I2C antara pin SDA/SCL sensor dan GPIO mikrokontroler direalisasikan menggunakan teknik *point-to-point wiring* di sisi bawah PCB.

#### 4.3.1.2 Implementasi Gateway

Perangkat ini dibangun menggunakan ESP32-S3 WROOM-1 dan dirancang untuk ditempatkan pada titik yang memiliki akses sumber daya listrik kontinu.



Gambar 4.11 Implementasi Gateway

Realisasi fisik *gateway* sebagaimana ditunjukkan pada Gambar 4.11 memperlihatkan penggunaan antenna omnidireksional eksternal yang dihubungkan melalui *port* IPEX/U.FL untuk meminimalisir redaman sinyal pada jarak yang bervariasi mengingat proses transmisi FOTA menuntut kestabilan paket data yang tinggi. Antarmuka pemantauan lapangan dirakit menggunakan tiga buah LED, yakni warna kuning, biru, dan modul RGB. Masing-masing jalur anoda dipasang resistor pembatas arus sebesar 220  $\Omega$  menuju pin GPIO. Indikator ini merespons status instruksi logika untuk



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

memberikan umpan balik langsung terkait konektivitas *Wi-Fi*, transmisi BLE, dan kode status kegagalan atau keberhasilan proses pembaruan memori *flash*. Tidak seperti *node* sensor, perangkat *gateway* tidak menggunakan modul manajemen baterai. Daya dipasok secara konstan melalui kabel daya USB yang menyalurkan tegangan stabil 5V DC langsung ke regulator internal mikrokontroler.

#### 4.3.2 Implementasi Perangkat Lunak Mikrokontroler

Pengembangan perangkat lunak mikrokontroler atau *firmware* pada sistem ini dibangun di atas kerangka kerja Espressif IoT Development Framework (ESP-IDF) yang berjalan pada sistem operasi FreeRTOS.

##### 4.3.2.1 Implementasi *Firmware* Node Sensor

*Firmware* pada ESP32-C3 bertindak sebagai perangkat periferi atau BLE *Server* yang pasif menunggu koneksi, mengumpulkan data getaran, dan menangani penulisan memori saat pembaruan nirkabel berlangsung. Implementasi kodenya dijabarkan pada tiga fungsionalitas berikut:

##### 1. Profil BLE GATT dan Optimasi Kapasitas Transmisi

Untuk mengatasi masalah pemotongan data bawaan protokol BLE yang sangat membatasi kecepatan FOTA, *firmware* secara eksplisit menegosiasikan ukuran MTU menjadi 512 *bytes* sesaat sebelum *stack* BLE diinisialisasi. Profil GATT dikonfigurasi menggunakan UUID kustom 128-bit untuk memisahkan layanan data sensor dan layanan pembaruan FOTA.

Gambar 4.12 memperlihatkan potongan kode karakteristik aliran data FOTA yang diatur dengan *flag* “BLE\_GATT\_CHR\_F\_WRITE\_NO\_RSP” agar *gateway* dapat mengirimkan potongan memori secara sekuensial tanpa harus menunggu sinyal konfirmasi dari *node*.

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

1 {
2     .type = BLE_GATT_SVC_TYPE_PRIMARY,
3     .uuid = &ota_svc_uuid.u,
4     .characteristics = (struct ble_gatt_chr_def[]) {
5         {
6             .uuid = &ota_chr_ctrl_uuid.u,
7             .access_cb = ota_gatt_ctrl_cb,
8             .flags = BLE_GATT_CHR_F_WRITE | BLE_GATT_CHR_F_READ,
9         },
10        {
11            .uuid = &ota_chr_data_uuid.u,
12            .access_cb = ota_gatt_data_cb,
13            .flags = BLE_GATT_CHR_F_WRITE_NO_RSP,
14        },
15        {
16            .uuid = &ota_chr_ver_uuid.u,
17            .access_cb = ota_gatt_ver_cb,
18            .flags = BLE_GATT_CHR_F_READ,
19        },
20        { 0 }
21    }
22 }

```

Gambar 4.12 Karakteristik BLE GATT OTA

## 2. Akuisisi Data Multitasking dengan FreeRTOS

Proses pembacaan sensor berjalan di dalam *task* tersendiri yaitu “shm\_data\_task” yang dialokasikan ke dalam memori sebesar 4096 *bytes*. Di dalam siklus *loop* tak terbatas, mikrokontroler membaca register ADXL345\_REG\_DATA\_X0 via protokol I2C dan membaca nilai *Analog-to-Digital Converter* untuk menghitung tegangan baterai.

Apabila koneksi BLE aktif dan proses FOTA tidak sedang berjalan (!ota\_is\_running), data tersebut akan dikonversi ke satuan *g-force* dan memicu fungsi “ble\_gatts\_chr\_updated()” untuk langsung menembakkan notifikasi data ke *gateway* setiap 1 detik. Potongan kode fungsi pembacaan data sensor dapat dilihat pada Gambar 4.13 di bawah.



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
void shm_data_task(void *pvParameter) {
    uint8_t data_raw[6];

    while(1) {
        if (target_conn_handle != BLE_HS_CONN_HANDLE_NONE && !ota_is_running) {
            int raw_val = 0;
            ESP_ERROR_CHECK(adc_oneshot_read(adc1_handle, ADC_CHANNEL_1, &raw_val));
            float voltage = (raw_val / 4095.0) * 3.3 * 1.83;
            uint8_t reg_addr = ADXL345_REG_DATA0;
            esp_err_t err = i2c_master_write_read_device(I2C_MASTER_NUM, ADXL345_ADDR,
                                                         &reg_addr, 1, data_raw, 6,
                                                         1000 / portTICK_PERIOD_MS);

            if (err == ESP_OK) {
                int16_t x_int = (data_raw[1] << 8) | data_raw[0];
                int16_t y_int = (data_raw[3] << 8) | data_raw[2];
                int16_t z_int = (data_raw[5] << 8) | data_raw[4];

                current_shm_data.ax = x_int * 0.0039f;
                current_shm_data.ay = y_int * 0.0039f;
                current_shm_data.az = z_int * 0.0039f;
            } else {
                printf("[HW ERROR] Gagal membaca ADXL345! Cek Kabel (Status: %s)\n", esp_err_to_name(err));
                current_shm_data.ax = 0.0f;
                current_shm_data.ay = 0.0f;
                current_shm_data.az = 0.0f;
            }
            current_shm_data.vbatt = voltage;
            ble_gatts_chr_updated(notify_char_val_handle);
            printf("[NODE] AX:%.2f AY:%.2f AZ:%.2f | Vbatt: %.2fV\n",
                  current_shm_data.ax, current_shm_data.ay, current_shm_data.az, current_shm_data.vbatt);
        }
        vTaskDelay(pdMS_TO_TICKS(1000));
    }
}
```

Gambar 4.13 Implementasi Pembacaan Sensor ADXL345

### 3. Rutinitas Pembaruan Partisi Memori

Mekanisme pertukaran *firmware* dijalankan melalui fungsi *callback* saat data masuk dari jalur BLE. Terdapat tiga fase utama yang dieksekusi menggunakan API bawaan `esp_ota_ops.h`:

- a. Fase Inisialisasi: Saat *node* menerima perintah mulai dengan kode “0x01” pada *Characteristic Control*, sistem akan memanggil “`esp_ota_get_next_update_partition(NULL)`” untuk mencari blok partisi pasif yang sedang tidak digunakan, lalu membuka jalur penulisan menggunakan “`esp_ota_begin()`”. Gambar 4.14 menunjukkan implementasi fase inisialisasi pada *Characteristic Control*.



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

1 if (cmd == 0x01) {
2     printf("\n[OTA] Menerima perintah START (0x01)\n");
3     update_partition = esp_ota_get_next_update_partition(NULL);
4     if (update_partition == NULL) {
5         printf("[OTA] Error: Partisi OTA tidak ditemukan!\n");
6         return BLE_ATT_ERR_UNLIKELY;
7     }
8
9     printf("[OTA] Menulis ke partisi: %s\n", update_partition->label);
10    esp_err_t err = esp_ota_begin(update_partition,
11                                OTA_WITH_SEQUENTIAL_WRITES,
12                                &ota_handle);
13
14    if (err != ESP_OK) {
15        printf("[OTA] Error esp_ota_begin: %s\n", esp_err_to_name(err));
16        return BLE_ATT_ERR_UNLIKELY;
17    }
18
19    ota_is_running = true;
20    ota_total_bytes = 0;
21    printf("[OTA] OTA Dimulai! Menyiapkan pipa data...\n");
22 }

```

Gambar 4.14 Implementasi Fase Inisialisasi OTA

- b. Fase Penulisan: Setiap kali paket data 500 *bytes* masuk ke fungsi “ota\_gatt\_data\_cb()”, paket tersebut disalin ke *buffer* lokal dan langsung ditulis ke dalam memori *flash* menggunakan “esp\_ota\_write()”. Gambar 4.15 memperlihatkan implementasi kode *callback* untuk fase penulisan paket *firmware*.

```

1 static int ota_gatt_data_cb(uint16_t conn_handle, uint16_t attr_handle,
2                             struct ble_gatt_access_ctxt *ctxt, void *arg) {
3     if (ctxt->op == BLE_GATT_ACCESS_OP_WRITE_CHR) {
4         if (!ota_is_running) return BLE_ATT_ERR_UNLIKELY;
5
6         uint16_t len = OS_MBUF_PKTLEN(ctxt->om);
7         uint8_t data_buf[512];
8         if (len > sizeof(data_buf)) return BLE_ATT_ERR_INSUFFICIENT_RES;
9
10        os_mbuf_copydata(ctxt->om, 0, len, data_buf);
11
12        esp_err_t err = esp_ota_write(ota_handle, data_buf, len);
13        if (err != ESP_OK) {
14            printf("[OTA] Gagal menulis ke Flash: %s\n", esp_err_to_name(err));
15            esp_ota_abort(ota_handle);
16            ota_is_running = false;
17            return BLE_ATT_ERR_UNLIKELY;
18        }
19
20        ota_total_bytes += len;
21
22        if (ota_total_bytes % 10240 < len) {
23            printf("[OTA] Menerima data: %d bytes...\n", ota_total_bytes);
24        }
25    }
26    return 0;
27 }

```

Gambar 4.15 Implementasi Fase Penulisan OTA



#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- c. Fase Finalisasi: Setelah seluruh *byte* dikirim, *gateway* memberikan perintah selesai dengan kode “0x02”. *Node* mengeksekusi “*esp\_ota\_end()*” untuk memvalidasi integritas file, kemudian mengubah rute *bootloader* dengan perintah “*esp\_ota\_set\_boot\_partition()*”. Setelah berhasil, mikrokontroler dipaksa memuat ulang sistem dengan “*esp\_restart()*” agar masuk ke sistem operasi yang baru. Gambar 4.16 menunjukkan kode hasil implementasi fase finalisasi.

```

1 else if (cmd == 0x02) {
2     if (!ota_is_running) return BLE_ATT_ERR_UNLIKELY;
3     printf("\n[OTA] Menerima perintah END (0x02). Total data: %d bytes\n",
4           ota_total_bytes);
5
6     esp_err_t err = esp_ota_end(ota_handle);
7     if (err != ESP_OK) {
8         printf("[OTA] Error esp_ota_end: %s\n", esp_err_to_name(err));
9         ota_is_running = false;
10        return BLE_ATT_ERR_UNLIKELY;
11    }
12
13    err = esp_ota_set_boot_partition(update_partition);
14    if (err != ESP_OK) {
15        printf("[OTA] Error mengatur partisi boot: %s\n", esp_err_to_name(err));
16        ota_is_running = false;
17        return BLE_ATT_ERR_UNLIKELY;
18    }
19
20    printf("[OTA] FIRMWARE BERHASIL DIPERBARUI! Restarting dalam 3 detik...\n");
21    vTaskDelay(pdMS_TO_TICKS(3000));
22    esp_restart();
23 }
```

Gambar 4.16 Implementasi Fase Finalisasi Perintah OTA

#### 4.3.2.2 Implementasi *Firmware Gateway*

Perangkat lunak pada *gateway* memiliki peran krusial sebagai jembatan jaringan sekaligus pusat kendali distribusi pembaruan. *Firmware gateway* dikembangkan menggunakan kerangka kerja ESP-IDF dan pustaka Bluetooth NimBLE. Mengingat perannya yang padat komputasi, implementasi logika pada *gateway* dibagi ke dalam beberapa fungsionalitas utama yang dijalankan sebagai task FreeRTOS independen:

##### 1. Manajemen Konektivitas *Wi-Fi* dan MQTT

Sebagai penghubung ke infrastruktur *cloud*, *gateway* pertama-kali menginisialisasi protokol *Wi-Fi* dan setelah memperoleh alamat IP, *gateway* mengonfigurasi klien MQTT untuk terhubung secara aman ke EMQX Broker menggunakan sertifikat TLS.



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Gambar 4.17 menunjukkan implementasi penanganan pesan MQTT pada *gateway* dengan melakukan *subscribe* pada topik “shm/ota/trigger” untuk mendengarkan instruksi pembaruan dari administrator web. Ketika *payload* JSON berisi URL *firmware* diterima, sistem akan memproses data tersebut dan memicu *task* “ota\_download\_and\_send\_task”.

```

1 case MQTT_EVENT_DATA:
2     if (strcmp(event->topic, "shm/ota/trigger", event->topic_len) == 0) {
3         char *json_str = malloc(event->data_len + 1);
4         memcpy(json_str, event->data, event->data_len);
5         json_str[event->data_len] = '\0';
6
7         cJSON *root = cJSON_Parse(json_str);
8         if (root != NULL) {
9             cJSON *cmd = cJSON_GetObjectItem(root, "cmd");
10            cJSON *target_mac = cJSON_GetObjectItem(root, "target_mac");
11            cJSON *url = cJSON_GetObjectItem(root, "url");
12            cJSON *target_ver = cJSON_GetObjectItem(root, "target_version");
13
14            if (cmd && cJSON_IsString(cmd) && strcmp(cmd->valuestring, "start_ota") == 0) {
15                if (target_mac && url && target_ver) {
16                    int target_idx = -1;
17                    for (int i = 0; i < MAX_NODES; i++) {
18                        if (nodes[i].is_active && strcmp(nodes[i].mac_address, target_mac->valuestring) == 0) {
19                            target_idx = i;
20                            break;
21                        }
22                    }
23
24                    if (target_idx != -1) {
25                        if (!nodes[target_idx].ota_in_progress) {
26                            strncpy(current_ota_url, url->valuestring, sizeof(current_ota_url)-1);
27                            strncpy(target_ota_version, target_ver->valuestring, sizeof(target_ota_version)-1);
28                            printf("\n[MQTT] TRIGGER OTA DITERIMA! NODE: %s\n", nodes[target_idx].mac_address);
29
30                            int *arg_idx = malloc(sizeof(int));
31                            *arg_idx = target_idx;
32                            xTaskCreate(ota_download_and_send_task, "ota_task", 8192, arg_idx, 5, NULL);
33                        }
34                    }
35                }
36            }
37            cJSON_Delete(root);
38            free(json_str);
39        }
40    }
41    break;

```

Gambar 4.17 Implementasi *Event Handler* dan Pemrosesan *Payload* JSON

### c) Pemindaian BLE, Negosiasi MTU, dan Penerimaan Telemetri

Di sisi jaringan lokal, *gateway* beroperasi sebagai *GATT Client*. Agar tidak menguras kinerja antenna dan meminimalisir tabrakan sinyal dengan *Wi-Fi*, siklus kerja pemindaian BLE dibatasi menjadi 50% dengan jendela pemindaian 50ms dalam interval 100ms. Gambar 4.18 memperlihatkan implementasi rutin “ble\_gap\_event\_cb” saat koneksi berhasil dibangun. *Gateway* tidak langsung meminta data, melainkan mengutamakan negosiasi MTU hingga 512 *bytes* guna membuka lebar pipa transmisi. Setelah *node* mengirimkan notifikasi data getaran,



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

*gateway* membongkar struktur data biner tersebut, mengemasnya kembali menjadi format JSON, dan menerbitkannya ke broker MQTT.

```

1 case BLE_GAP_EVENT_CONNECT:
2 ble_gattc_exchange_mtu(event->connect.conn_handle, on_mtu_exchanged, NULL);
3 break;
4
5 case BLE_GAP_EVENT_NOTIFY_RX: {
6     int n_idx = find_node_by_handle(event->notify_rx.conn_handle);
7     if (n_idx != -1 && event->notify_rx.attr_handle == nodes[n_idx].notify_handle) {
8         SHMData shm;
9         if (OS_MBUF_PKTLEN(event->notify_rx.om) == sizeof(SHMData)) {
10             os_mbuf_copydata(event->notify_rx.om, 0, sizeof(SHMData), &shm);
11
12             printf("[SHM] Node %s | AX:%.2f AY:%.2f AZ:%.2f | Vbatt: %.2fV\n",
13                 nodes[n_idx].mac_address, shm.ax, shm.ay, shm.az, shm.vbatt);
14
15             if (mqtt_connected) {
16                 char json_payload[128];
17                 snprintf(json_payload, sizeof(json_payload),
18                     "{\"node_mac\":\"%s\", \"ax\":%.2f, \"ay\":%.2f, \"az\":%.2f, \"vbatt\":%.2f}\n",
19                     nodes[n_idx].mac_address, shm.ax, shm.ay, shm.az, shm.vbatt);
20                 esp_mqtt_client_publish(mqtt_client, MQTT_TOPIC, json_payload, 0, 1, 0);
21             }
22         }
23     }
24     break;

```

Gambar 4.18 Implementasi Penerimaan Notifikasi Telemetri dan Negosiasi MTU

#### 4. Fragmentasi *Streaming* FOTA via HTTP

Ini adalah inti dari penyelesaian masalah keterbatasan memori dan kecepatan pada FOTA. Daripada mengunduh *file* biner utuh ke dalam memori internal yang terbatas, *gateway* mengimplementasikan metode *streaming* FOTA.

Seperti yang ditunjukkan pada Gambar 4.19 di bawah, pertama-tama *gateway* membuka koneksi HTTP *Client* langsung ke tautan *firmware* target. Sistem membaca aliran paket dari internet sebesar 500 *bytes* per siklus, dan seketika itu juga meneruskannya ke *node* via BLE menggunakan metode *Write Without Response*. Mekanisme ini dilengkapi dengan manajemen jeda dinamis setiap 4096 *bytes* terkirim, berfungsi untuk mencegah *buffer overflow* atau penumpukan antrean pada modul radio BLE *node*.



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

1 esp_http_client_config_t config = {
2     .url = current_ota_url,
3     .method = HTTP_METHOD_GET,
4     .cert_bundle_attach = esp_cert_bundle_attach,
5     .buffer_size = 8192,
6     .buffer_size_tx = 2048,
7     .timeout_ms = 30000,
8     .keep_alive_enable = true,
9 };
10 esp_http_client_handle_t client = esp_http_client_init(&config);
11 esp_err_t err = esp_http_client_open(client, 0);
12
13 if (err == ESP_OK) {
14     uint8_t buffer[500];
15     int data_read = 0;
16     int total_sent = 0;
17     bool is_failed = false;
18     int bytes_since_last_pause = 0;
19
20     while (1) {
21         data_read = esp_http_client_read(client, (char *)buffer, sizeof(buffer));
22
23         int rc;
24         do {
25             rc = ble_gattc_write_no_rsp_flat(nodes[idx].conn_handle,
26                                             nodes[idx].ota_data_handle,
27                                             buffer,
28                                             data_read);
29             if (rc == 6 || rc == 130) vTaskDelay(pdMS_TO_TICKS(10));
30         } while (rc == 6 || rc == 130);
31
32         total_sent += data_read;
33         bytes_since_last_pause += data_read;
34
35         if (bytes_since_last_pause >= 4096) {
36             vTaskDelay(pdMS_TO_TICKS(150));
37             bytes_since_last_pause -= 4096;
38         } else {
39             vTaskDelay(pdMS_TO_TICKS(15));
40         }
41     }
42     /** PERINTAH END **/
43 } else {
44     printf("[GATEWAY OTA] Gagal terhubung ke URL Server!\n");
45     publish_ota_status(nodes[idx].mac_address, "FAILED");
46 }

```

Gambar 4.19 Algoritma Fragmentasi FOTA

#### 4. Mekanisme Verifikasi Integritas dan *Rollback*

Untuk menjamin keamanan pembaruan, sistem harus memvalidasi bahwa *firmware* baru telah benar-benar dieksekusi. Gambar 4.20 menunjukkan implementasi rutin verifikasi yang dijalankan pasca-*reboot*. Setelah *node* menyala kembali dan terkoneksi, *gateway* secara otomatis memanggil karakteristik pembaca versi. Jika versi yang dibaca cocok dengan versi target di awal, *gateway* mengirimkan log “SUCCESS” ke sistem web. Namun, apabila terjadi kegagalan sistem dan *node* terpaksa melakukan *rollback*, *gateway* akan mendeteksinya dan mempublikasikan log “FAILED\_ROLLBACK” untuk memberi tahu administrator.



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

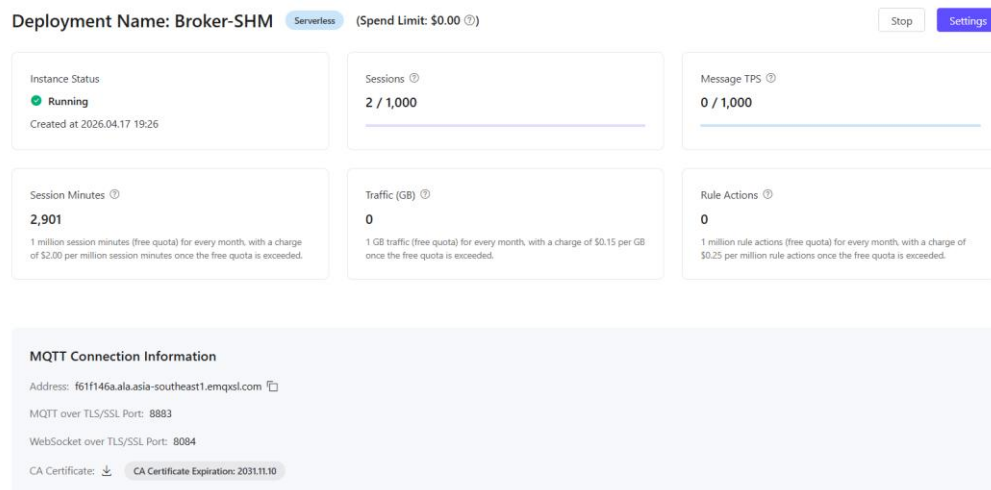
1 static int read_version_cb(uint16_t conn_handle, const struct ble_gatt_error *error,
2                          struct ble_gatt_attr *attr, void *arg) {
3     if (error->status == 0) {
4         char node_ver[20] = {0};
5         int len = OS_MBUF_PKTLEN(attr->om);
6
7         if (strlen(verifying_mac) > 0 && strcmp(nodes[idx].mac_address, verifying_mac) == 0) {
8             printf("\n[VERIFIKASI] Node MAC %s melapor versi: '%s'\n", nodes[idx].mac_address, node_ver);
9             char *n_ver = node_ver;
10            char *t_ver = target_ota_version;
11
12            if (n_ver[0] == 'v' || n_ver[0] == 'V') n_ver++;
13            if (t_ver[0] == 'v' || t_ver[0] == 'V') t_ver++;
14
15            if (strcmp(n_ver, t_ver) == 0) {
16                printf("[VERIFIKASI] SUKSES! Versi cocok.\n");
17                publish_ota_status(nodes[idx].mac_address, "SUCCESS");
18            } else {
19                printf("[VERIFIKASI] GAGAL! Versi tidak sama.\n");
20                publish_ota_status(nodes[idx].mac_address, "FAILED_ROLLBACK");
21            }
22            memset(verifying_mac, 0, sizeof(verifying_mac));
23        }
24    } else {
25        printf("[VERIFIKASI] Gagal membaca versi GATT! Error: %d\n", error->status);
26        int idx = find_node_by_handle(conn_handle);
27        if (idx != -1) publish_ota_status(nodes[idx].mac_address, "FAILED_ROLLBACK");
28        memset(verifying_mac, 0, sizeof(verifying_mac));
29    }
30    return 0;
31 }

```

Gambar 4.20 Implementasi Validasi Versi Pasca-Pembaruan

### 4.3.3 Implementasi Broker MQTT

Komunikasi data antara perangkat keras *gateway* dan sistem API *Server* diimplementasikan menggunakan protokol komunikasi MQTT. Penelitian ini memanfaatkan layanan *EMQX Serverless* sebagai broker MQTT.



Gambar 4.21 Tampilan Dasbor Utama Deployment EMQX Serverless

Gambar 4.21 menunjukkan dasbor utama dari EMQX *Serverless* yang mengonfirmasi bahwa broker MQTT telah beroperasi secara aktif, dasbor tersebut menginformasikan



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

parameter esensial yang dibutuhkan oleh klien perangkat keras seperti *gateway* maupun API Server untuk menjalin komunikasi, yaitu “Address” yang bertindak sebagai *endpoint* utama jaringan.

Selain itu, sistem menyediakan berbagai opsi *port* koneksi untuk memfasilitasi kebutuhan klien, mencakup *port* 8084 untuk protokol Websocket dan *port* 8883 untuk komunikasi MQTT yang diamankan dengan enkripsi TLS.

| Client ID           | Username   | IP Address         | Keepalive | Protocol Type | Clean Start | Expiry Interval (s) | Status    |
|---------------------|------------|--------------------|-----------|---------------|-------------|---------------------|-----------|
| ESP32_d0D0B0        | rafirashya | 103.136.56.74:4186 | 10        | MQTT          | true        | 0                   | Connected |
| fota_backend_server | rafirashya | 103.136.56.74:4334 | 30        | MQTT          | true        | 0                   | Connected |

Gambar 4.22 Daftar Klien yang Terhubung pada Broker EMQX

Gambar 4.22 memperlihatkan antarmuka pemantauan klien pada dasbor EMQX *Serverless*, yang memvalidasi keberhasilan integrasi antara perangkat keras dan backend atau API Server. Pada daftar tersebut, terlihat perangkat klien yakni *gateway* dengan *client id* “ESP32\_d0D0B0” dan sistem backend dengan *client id* “fota\_backend\_server” berhasil menjalin dan mempertahankan sesi koneksi MQTT ke broker secara bersamaan dengan status aktif. Kehadiran kedua entitas klien pada daftar ini memperlihatkan bahwa konfigurasi jaringan dan kredensial autentikasi telah diimplementasikan dengan benar dan siap melakukan pertukaran data.

| Username           | Number of permissions | Actions    |
|--------------------|-----------------------|------------|
| rafirashya         | 4                     |            |
| Topic              | Action                | Permission |
| shm/gateway/status | Publish & Subscribe   | Allow      |
| shm/ota/status     | Publish & Subscribe   | Allow      |
| shm/telemetry      | Publish & Subscribe   | Allow      |
| shm/ota/trigger    | Publish & Subscribe   | Allow      |

Gambar 4.23 Konfigurasi Otorisasi Akses Topik MQTT

Gambar 4.23 menampilkan konfigurasi *Access Control List* (ACL) atau daftar kontrol akses pada broker EMQX *Serverless*. Tangkapan layar ini memperlihatkan penerapan lapisan keamanan otorisasi lanjutan bagi klien yang terhubung menggunakan



#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

kredensial *username* “rafirashya”. Selain dibatasi oleh kata sandi untuk autentikasi masuk, klien juga dibatasi secara ketat mengenai jalur komunikasi mana saja yang dapat diaksesnya. Sistem memberikan hak akses untuk melakukan aksi publikasi maupun berlangganan secara eksklusif hanya pada empat topik spesifik, yaitu “shm/ota/trigger”, “shm/ota/status”, “shm/telemetry”, serta “shm/gateway/status”.

```

1  {
2    "cmd": "start_ota",
3    "target_mac": "1C:DB:D4:C6:3C:56",
4    "target_version": "1.0.0",
5    "url": "https://storage.googleapis.com/shm-firmware-bucket/"
6  }
  
```

Gambar 4.24 Format *Payload* Instruksi Pembaruan pada Topik shm/ota/trigger

Gambar 4.24 memperlihatkan hasil implementasi lalu lintas pesan saat API Server memublikasikan instruksi inisiasi pembaruan jarak jauh. Pesan tersebut dikirimkan melalui topik “shm/ota/trigger” dengan menggunakan format data terstruktur JSON untuk proses ekstraksi data di sisi mikrokontroler *gateway*.

Struktur data tersebut memuat empat parameter operasional utama dijelaskan sebagai berikut:

1. Parameter “cmd” berisi perintah “start\_ota” untuk memicu status operasi pada perangkat keras.
2. Parameter “target\_mac” menspesifikasikan alamat fisik perangkat sasaran, yang berfungsi sebagai filter keamanan agar pembaruan hanya diproses oleh *node* yang dituju.
3. Parameter “target\_version” memastikan kontrol versi *firmware* yang diimplementasikan.
4. Parameter “url” menyediakan tautan sementara yang mengarah ke *cloud storage bucket* sebagai penyimpanan tempat berkas biner berada.

Keempat parameter terintegrasi ini membuktikan bahwa API *Server* mampu mendistribusikan instruksi perutean berkas yang presisi kepada *gateway* sebelum proses transmisi BLE dilakukan.



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

|                |                                                                              |              |
|----------------|------------------------------------------------------------------------------|--------------|
| shm/ota/status | {"cmd":"ota_status", "node_mac":"1C:DB:D4:C6:3C:56", "status":"SUCCESS"}     | 20:50:33.170 |
| shm/ota/status | {"cmd":"ota_status", "node_mac":"1C:DB:D4:C6:3C:56", "status":"REBOOTING"}   | 20:50:25.930 |
| shm/ota/status | {"cmd":"ota_status", "node_mac":"1C:DB:D4:C6:3C:56", "status":"IN_PROGRESS"} | 20:49:40.457 |

Gambar 4.25 Implementasi Pesan Status Pembaruan pada Topik shm/ota/status

Gambar 4.25 menampilkan implementasi pesan status pembaruan yang dipublikasikan oleh *gateway* ke *Server* melalui topik “shm/ota/status”. Saluran komunikasi ini bertindak sebagai mekanisme pelaporan agar arsitektur *backend* dapat memantau perkembangan sistem jarak jauh. Muatan pesan dikirimkan dalam format JSON yang memuat parameter cmd, mac address perangkat, dan status proses.

### 4.3.4 Implementasi Backend dan Basis Data

Implementasi *backend* dilakukan untuk membangun sistem yang memproses logika aplikasi di balik layar dan menjadi penghubung antara perangkat keras di lapangan, antarmuka *web*, dan basis data.

#### 4.3.4.1 Implementasi Layanan Backend

##### a) Manajemen Penyimpanan Berkas Firmware

Sistem *backend* tidak menyimpan berkas biner *firmware* secara lokal, melainkan bertindak sebagai jembatan menuju layanan penyimpanan objek. Pada berkas *backend*, fungsi *Upload* diimplementasikan untuk menerima berkas biner berukuran maksimal 10 MB. Sebelum berkas dilempar ke *bucket* Google Cloud Storage (GCS) yang dapat dilihat pada Gambar 4.26, *backend* menghitung nilai *hash* SHA-256 dari berkas tersebut secara dinamis. Nilai ini kemudian disimpan ke dalam basis data PostgreSQL sebagai parameter verifikasi integritas data untuk mencegah korupsi pembaruan.

| shm-firmware-bucket         |               |               |             |  |  |  |  |
|-----------------------------|---------------|---------------|-------------|--|--|--|--|
| Location                    | Storage class | Public access | Protection  |  |  |  |  |
| asia-southeast1 (Singapore) | Standard      | Not public    | Soft Delete |  |  |  |  |

| Objects                                                                             |         |      |        |         |               |               |     |
|-------------------------------------------------------------------------------------|---------|------|--------|---------|---------------|---------------|-----|
| Folder browser                                                                      |         |      |        |         |               |               |     |
| Buckets > shm-firmware-bucket > firmware                                            |         |      |        |         |               |               |     |
| Create folder Upload Transfer data Create batch operations New Other services       |         |      |        |         |               |               |     |
| Filter by name prefix only Filter Filter objects and folders Show Live objects only |         |      |        |         |               |               |     |
|                                                                                     | Name    | Size | Type   | Created | Storage class | Last modified | Cus |
| <input type="checkbox"/>                                                            | v1.0.0/ | —    | Folder | —       | —             | —             | —   |
| <input type="checkbox"/>                                                            | v1.0.1/ | —    | Folder | —       | —             | —             | —   |
| <input type="checkbox"/>                                                            | v1.0.2/ | —    | Folder | —       | —             | —             | —   |

Gambar 4.26 Implementasi *Bucket* GCS

##### b) Orkestrasi Antrean FOTA



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Karena proses transmisi FOTA melalui jaringan BLE memakan waktu hingga beberapa detik hingga menit, *backend* tidak boleh memblokir API yang sedang menunggu proses selesai. Oleh karena itu, sistem mengimplementasikan *Background Worker* menggunakan *Goroutine* yang berjalan secara asinkron di belakang layar. *Worker* tersebut bertugas memindai basis data setiap 2 detik untuk mencari tiket pembaruan yang berstatus “PENDING”. Saat menemukan antrean, *worker* ini akan membuat *Signed URL* ke GCS yang hanya berlaku selama 30 menit. Tautan ini bersama dengan *MAC Address* target akan dikonversi menjadi format JSON dan dipublikasikan ke *gateway* melalui protokol MQTT. Gambar 4.27 menunjukkan implementasi *worker* dalam membuat *Signed URL* dan mengonversi muatan MQTT

```

1 objectKey := fmt.Sprintf("firmware/v%s/nimble-shm-ota.bin", otaLog.TargetFirmware.Version)
2 opts := &storage.SignedURLOptions{
3     GoogleAccessID: h.GoogleAccessID,
4     PrivateKey:      h.PrivateKey,
5     Method:          "GET",
6     Expires:         time.Now().Add(30 * time.Minute),
7 }
8 signedURL, err := storage.SignedURL(h.BucketName, objectKey, opts)
9
10 triggerMsg := map[string]string{
11     "cmd":      "start_ota",
12     "target_mac": otaLog.Node.MacAddress,
13     "url":       signedURL,
14     "target_version": otaLog.TargetFirmware.Version,
15 }
16 payload, _ := json.Marshal(triggerMsg)
17 h.MQTTClient.Publish("shm/ota/trigger", 1, false, payload)

```

Gambar 4.27 Implementasi Worker Membuat Signed URL dan Payload MQTT

### c) Sinkronisasi Status Asinkron Menggunakan *Channel*

Tantangan terbesar dalam komunikasi asinkron IoT adalah bagaimana *Worker* mengetahui bahwa FOTA telah selesai. Implementasi ini menggunakan fitur *Channel* pada Golang sebagai jembatan komunikasi antar-*Thread*. Ketika *gateway* mengirimkan laporan sukses atau gagal ke topik “shm/ota/status”, fungsi yang menangani status akan menangkap pesan tersebut, memperbarui tabel basis data, dan menyuntikkan *MAC Address* ke dalam *Channel* sebagaimana ditunjukkan pada Gambar 4.28.



#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

1 // Potongan fungsi HandleStatusUpdate (ota.go)
2 if payload.Status == "SUCCESS" {
3     h.DB.Model(&lastLog).Update("completed_at", time.Now())
4     h.DB.Model(&node).Update("current_firmware_id", lastLog.TargetFirmwareID)
5 }
6
7 if payload.Status == "SUCCESS" || payload.Status == "FAILED" {
8     select {
9         case OTAUpdateChan <- payload.NodeMAC: // Mengirim sinyal ke Channel
10         default:
11     }
12 }

```

Gambar 4.28 Implementasi Kode Menangani Status Pembaruan

Di sisi lain, *Worker* yang sebelumnya memicu FOTA sedang dalam kondisi diblokir menunggu kiriman sinyal dari *Channel* tersebut. Sistem dilengkapi dengan fungsi *timeout* absolut selama 3 menit. Jika dalam waktu 3 menit *Gateway* tidak memberikan respons balik, *Worker* akan menganggap proses tersebut kedaluwarsa dan secara otomatis mengubah statusnya di basis data menjadi “FAILED”, fungsi *timeout* dapat dilihat pada Gambar 4.29 di bawah ini.

```

1 timeout := time.After(5 * time.Minute)
2 completed := false
3
4 for !completed {
5     select {
6         case mac := <-OTAUpdateChan:
7             if mac == otaLog.Node.MacAddress {
8                 log.Printf("[OTA-WORKER] Node %s selesai memproses update", mac)
9                 completed = true
10            } else {
11                log.Printf("[OTA-WORKER] Mengabaikan status update node lain: %s", mac)
12            }
13        case <-timeout:
14            var checkLog models.OtaLog
15            if err := h.Database.First(&checkLog, "id = ?", otaLog.ID).Error; err == nil {
16                if checkLog.Status == "IN_PROGRESS" || checkLog.Status == "PENDING" {
17                    h.Database.Model(&checkLog).Update("status", "FAILED")
18                }
19            }
20            completed = true
21        }

```

Gambar 4.29 Implementasi *Timeout* pada *Worker* FOTA

Pendekatan sinkronisasi menggunakan metode *Channel* ini mencegah terjadinya pemborosan penggunaan CPU di sisi *gateway* dan memastikan stabilitas *backend* saat menangani proses pembaruan *firmware* secara massal.

#### 4.3.4.2 Implementasi Komunikasi *Message Broker*



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

#### a) Inisialisasi Klien dan Konfigurasi Keamanan

Implementasi keamanan data telemetri struktur dan berkas biner *firmware* yang ditransmisikan melalui jaringan publik menggunakan koneksi MQTT yang dikonfigurasi untuk menggunakan lapisan enkripsi *Transport Layer Security* (TLS) melalui pustaka standar Go. Fungsi “NewMQTTClient” mengonfigurasi parameter esensial meliputi kredensial autentikasi, waktu tunggu batas koneksi selama 10 detik, serta mengaktifkan fitur *SetAutoReconnect* bernilai *true*. Fitur reaktivasi otomatis ini krusial untuk memastikan *server* dapat langsung memulihkan jalur komunikasi tanpa perlu melakukan *restart* apabila broker EMQX mengalami gangguan jaringan sementara. Realisasi konfigurasi inisialisasi klien MQTT dan pengaturan TLS ditunjukkan pada Gambar 4.30.

```

1 func NewMQTTClient(broker , clientID, username, password string) *MQTTClient {
2     opts := mqtt.NewClientOptions()
3     opts.AddBroker(broker)
4     opts.SetClientID(clientID)
5     opts.SetUsername(username)
6     opts.SetPassword(password)
7     opts.SetAutoReconnect(true)
8     opts.SetConnectTimeout(10 * time.Second)
9
10    tlsconfig := &tls.Config{
11        InsecureSkipVerify: false,
12    }
13    opts.SetTLSConfig(tlsconfig)
14
15    opts.OnConnect = func(c mqtt.Client) {
16        log.Printf("Berhasil terhubung ke MQTT Broker: %s", broker)
17    }
18    opts.OnConnectionLost = func(c mqtt.Client, err error) {
19        log.Printf("Koneksi ke MQTT Broker terputus: %v", err)
20    }
21
22    client := mqtt.NewClient(opts)
23    return &MQTTClient{client: client}
24 }

```

Gambar 4.30 Implementasi Koneksi Klien MQTT dan Pengaturan TLS

#### b) Arsitektur Alur Topik

Implementasi subsistem MQTT pada *backend* membagi fungsionalitas jaringan ke dalam dua kanal utama untuk memisahkan lalu lintas telemetri data sensor dengan lalu lintas kendali *firmware*:

1. Kanal Publikasi dengan topik “shm/ota/trigger” digunakan oleh *background worker* yang implementasinya dapat dilihat pada Gambar 4.23. Ketika proses antrean valid dicapai, objek *Signed URL* dari GCS dikemas menjadi JSON dan



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

dipublikasikan dengan parameter tingkat keandalan Quality of Service (QoS) disetel pada nilai 1.

2. Kanal Langganan dengan topik “shm/ota/status” digunakan oleh *backend* untuk mendengarkan umpan balik dari *gateway*. Fungsi penanganan status didaftarkan sebagai *callback* pada topik ini dengan implementasi yang dapat dilihat pada Gambar 4.31. Setiap kali *Gateway* mengirimkan pembaruan tahapan eksekusi FOTA dari lapangan seperti status “IN\_PROGRESS”, “REBOOTING”, “SUCCESS”, ataupun “FAILED”, data tersebut langsung diolah untuk disinkronkan ke basis data.

```
1 err := mqttClient.Subscribe("shm/ota/status", 1,
   otaHandler.HandleStatusUpdate)
2 if err != nil {
3     log.Printf("Gagal subscribe ke topik status OTA: %v", err)
4 }
```

Gambar 4.31 Implementasi Kanal Langganan pada Topik Status

Pemisahan tata kelola kanal MQTT ini membuat arsitektur jaringan *backend* menjadi sangat modular, adaptif terhadap fluktuasi sinyal internet *gateway*, dan menjamin reliabilitas alur validasi status FOTA dari hulu ke hilir.

#### 4.3.5 Implementasi Antarmuka Pengguna

Implementasi antarmuka pengguna dilakukan dengan penerapan fungsionalitas nyata sebuah *website* agar admin yang ingin melakukan pembaruan *firmware* atau pengguna yang ingin memantau data sensor dapat dengan mudah melakukan tugasnya melalui *website* tersebut.

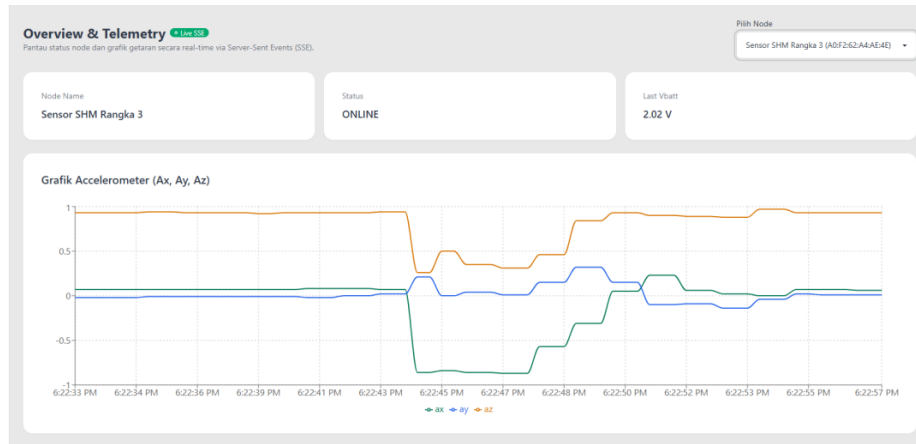
##### 4.3.5.1 Visualisasi Data Telemetri Pemantauan Kesehatan Struktur

Halaman ini berfungsi sebagai fitur pemantauan utama yang mengonsumsi data dari *endpoint* API *backend*. Antarmuka visualisasi data telemetri ini ditunjukkan pada Gambar 4.32.



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 4.32 Antarmuka Visualisasi Real-Time Telemetry Sensor SHM

Seperti yang dapat dilihat pada Gambar 4.32, halaman ini memberikan informasi telemetry yang terdiri dari sensor ADXL345 dan status tegangan baterai. Garis pada grafik yang semula lurus tiba-tiba mengalami fluktuasi di tengah-tengah menunjukkan aktivitas getaran yang terjadi pada *node*. Pada halaman ini juga terdapat nama *node* yang sedang dipantau beserta status jaringannya.

#### 4.3.5.2 Panel Pemicu dan Riwayat Pembaruan FOTA

halaman pemicu bertindak sebagai antarmuka eksekusi sentral yang memfasilitasi administrator untuk menginisiasi proses pembaruan perangkat lunak secara nirkabel, serta statusnya dapat dipantau pada halaman riwayat.

The figure shows a web interface for deploying firmware. It has a left sidebar with 'Deploy Firmware v1.0.0' and a 'Pilih Node Target' section. The main area is a table with columns: Status, Node, Firmware, Dimulai, Selesai, Durasi, and Admin. The table lists several nodes and their update progress.

| Status      | Node                                     | Firmware | Dimulai             | Selesai             | Durasi | Admin                    |
|-------------|------------------------------------------|----------|---------------------|---------------------|--------|--------------------------|
| ONLINE      | Sensor SHM Rangka 3<br>AB:F2:62:AA:AE:4E | v1.0.0   | 12/6/2026, 01:41:26 | 12/6/2026, 01:42:25 | 5%     | rafi.rashya.22@gmail.com |
| ONLINE      | Sensor SHM Rangka 3<br>AB:F2:62:AA:AE:4E | v1.0.0   | 12/6/2026, 01:38:49 | 12/6/2026, 01:39:37 | 4%     | rafi.rashya.22@gmail.com |
| IN PROGRESS | Sensor SHM Rangka 3<br>AB:F2:62:AA:AE:4E | v1.0.0   | 12/6/2026, 01:38:43 | -                   | -      | rafi.rashya.22@gmail.com |
| IN PROGRESS | Sensor SHM Rangka 3<br>AB:F2:62:AA:AE:4E | v1.0.1   | 12/6/2026, 01:38:42 | -                   | -      | rafi.rashya.22@gmail.com |
| IN PROGRESS | Sensor SHM Rangka 1<br>3C:08:04:C6:3C:55 | v1.0.1   | 12/6/2026, 01:33:43 | -                   | -      | rafi.rashya.22@gmail.com |
| FAILED      | Sensor SHM Rangka 1<br>3C:08:04:C6:3C:55 | v1.0.1   | 12/6/2026, 01:33:43 | -                   | -      | rafi.rashya.22@gmail.com |
| IN PROGRESS | Sensor SHM Rangka 3<br>AB:F2:62:AA:AE:4E | v1.0.1   | 12/6/2026, 01:33:42 | -                   | -      | rafi.rashya.22@gmail.com |
| FAILED      | Sensor SHM Rangka 1<br>3C:08:04:C6:3C:55 | v1.0.1   | 12/6/2026, 01:28:43 | -                   | -      | rafi.rashya.22@gmail.com |
| FAILED      | Sensor SHM Rangka 3<br>AB:F2:62:AA:AE:4E | v1.0.1   | 12/6/2026, 01:28:42 | -                   | -      | rafi.rashya.22@gmail.com |
| FAILED      | Sensor SHM Rangka 2                      | v1.0.1   | 12/6/2026, 01:28:42 | -                   | -      | rafi.rashya.22@gmail.com |

Gambar 4.33 Antarmuka Panel Orkestrasi dan Tabel Riwayat Pembaruan FOTA

Diperlihatkan pada Gambar 4.33, proses pembaruan perangkat lunak secara nirkabel baik untuk *node* tunggal maupun secara serentak ke berbagai perangkat melalui fitur pemilihan target interaktif. Pada halaman lain, sistem menyajikan tabel riwayat



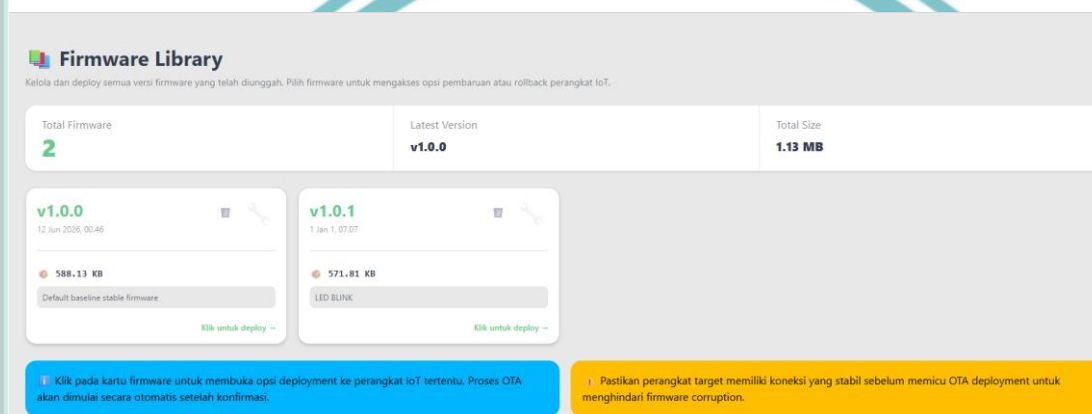
### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

pembaruan yang berfungsi sebagai acuan admin untuk melacak status pembaruan. Tabel ini secara visual memperlihatkan informasi penting seperti status akhir, identitas *node* sasaran, versi *firmware*, waktu eksekusi, serta admin yang memicu pembaruan.

#### 4.3.5.3 Halaman Pustaka Berkas *Firmware*

Berkas *firmware* yang diunggah oleh admin dapat dikelola pada halaman pustaka ini, admin dapat melihat versi, mengubah catatan rilis, dan menghapus versi *firmware* jika dibutuhkan.



Gambar 4.34 Halaman Pustaka Berkas *Firmware*

Gambar 4.34 memperlihatkan antarmuka halaman pustaka berkas *firmware* yang berfungsi sebagai pusat manajemen repositori pembaruan perangkat lunak. Halaman ini menyajikan daftar *firmware* yang telah tersimpan di *server* dalam bentuk tabel terstruktur. Tabel repositori tersebut menampilkan *metadata* dari setiap *firmware*, meliputi nomor versi rilis, nama dan ukuran berkas, catatan rilis, serta riwayat waktu unggah.

### 4.4 Pengujian Sistem

Pengujian dilakukan untuk memastikan bahwa seluruh subsistem perangkat keras dan perangkat lunak dapat terintegrasi serta beroperasi sesuai dengan spesifikasi rancangan yang telah ditetapkan. Rincian lebih lanjut mengenai parameter, perangkat kerja, dan skenario evaluasi dari tahap ini akan diuraikan secara menyeluruh pada deskripsi pengujian.

#### 4.4.1 Deskripsi Pengujian



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Pengujian merupakan tahapan krusial untuk mengevaluasi dan memvalidasi kinerja sistem purwarupa yang telah diimplementasikan. Pengujian dalam penelitian ini dirancang berdasarkan parameter yang selaras dengan rumusan masalah, tujuan penelitian, dan kebutuhan arsitektur jaringan sensor nirkabel. Pengujian yang akan dilakukan meliputi:

1. Pengujian fungsionalitas sistem, untuk memastikan perangkat keras, perangkat lunak mikrokontroler, dan *web dashboard* dapat terintegrasi serta beroperasi sesuai dengan rancangan.
2. Pengujian kinerja jaringan komparatif FOTA BLE dengan protokol ESP-NOW, meliputi metrik *success rate*, *throughput* aktual, dan *retransmission ratio* untuk mengevaluasi keandalan dan skalabilitas distribusi pembaruan pada 3 kategori ukuran berkas *firmware* (Kecil, Sedang, Besar) pada variasi jarak 1 s.d. 15 meter dalam kondisi *Line of Sight* (LOS) dan *Non-Line of Sight* (NLOS).

#### 4.4.2 Prosedur Pengujian

Sub-bab ini menguraikan langkah-langkah teknis dan sistematis yang dilakukan pada setiap tahap pengujian.

##### 4.4.2.1 Prosedur Pengujian Fungsionalitas Sistem

Pengujian fungsionalitas sistem dilakukan menggunakan pendekatan kotak hitam atau *black-box testing* untuk memverifikasi bahwa seluruh komponen perangkat keras, perangkat lunak, dan antarmuka web dapat beroperasi serta berintegrasi secara normal tanpa mengalami kegagalan. Rincian prosedur pengujian fungsionalitas dijabarkan pada Tabel 4.15 berikut:

Tabel 4.15 Prosedur Pengujian Fungsionalitas Sistem

| No | Metode Pengujian                         | Tujuan Pengujian                                                                                                                                                                                                                                               |
|----|------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Inisialisasi dan Verifikasi Koneksi Awal | Memastikan <i>node</i> sensor dan <i>gateway</i> menyala dengan normal, <i>gateway</i> berhasil terhubung ke jaringan <i>Wi-Fi</i> dan <i>broker</i> MQTT, serta kedua perangkat sukses melakukan penyambungan BLE dan negosiasi kapasitas transmisi atau MTU. |
| 2  | Simulasi Input Data Telemetri            | Memvalidasi kemampuan akuisisi data mikrokontroler dengan memberikan gangguan mekanis seperti ayunan                                                                                                                                                           |



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

| No | Metode Pengujian                                  | Tujuan Pengujian                                                                                                                                                                                          |
|----|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    |                                                   | atau getaran fisik secara acak pada modul sensor ADXL345.                                                                                                                                                 |
| 3  | Validasi Antarmuka Web Telemetry                  | Memastikan data telemetry hasil simulasi pada <i>node</i> berhasil dikirimkan secara <i>real-time</i> dan dirender ke dalam grafik <i>time-series</i> pada halaman <i>Overview &amp; Telemetry</i> .      |
| 4  | Inisiasi Pembaruan <i>Firmware</i> (FOTA)         | Memverifikasi keandalan integrasi web dengan memastikan bahwa penekanan tombol <i>Trigger</i> OTA pada <i>dashboard</i> berhasil menerbitkan instruksi perintah berformat JSON menuju <i>broker</i> MQTT. |
| 5  | Observasi Logika Transmisi <i>Gateway</i>         | Memastikan <i>gateway</i> merespons instruksi MQTT dengan mengekstrak URL <i>firmware</i> , mulai mengunduh berkas biner .bin via <i>HTTP Client</i> , dan mengalirkan paket data ke <i>node</i> via BLE. |
| 6  | Validasi Eksekusi Partisi dan Sinkronisasi Status | Memastikan <i>node</i> sensor berhasil menulis partisi pasif dan melakukan <i>reboot</i> ke versi terbaru, serta memastikan antarmuka web menyinkronkan status log OTA secara otomatis.                   |

#### 4.4.2.2 Prosedur Pengujian Kinerja Jaringan Komparatif

Pengujian kinerja jaringan komparatif dilakukan secara eksperimental untuk mengukur tingkat keandalan, kecepatan transfer, dan retransmisi paket data saat Gateway mendistribusikan berkas biner pembaruan ke Node Sensor menggunakan protokol BLE dan ESP-NOW.

Variabel bebas yang diuji meliputi jarak transmisi fisik, kondisi halangan, serta 3 kategori ukuran berkas *firmware*. Untuk meminimalkan kesalahan pencatatan manual, pengambilan data dilakukan secara otomatis menggunakan skrip otomatisasi Python yang membaca keluaran serial *Gateway*. Rincian matriks skenario pengujian dijabarkan pada Tabel 4.16 berikut:

Tabel 4.16 Skenario Pengujian Kinerja Jaringan dan Mekanisme FOTA



#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

| Skenario | Protokol        | Ukuran Berkas   | Jarak Transmisi      | Kondisi Lingkungan | Jumlah Repetisi |
|----------|-----------------|-----------------|----------------------|--------------------|-----------------|
| 1        | BLE dan ESP-NOW | Kecil (150 KB)  | 1 m, 5 m, 10 m, 15 m | LOS dan NLOS       | 10              |
| 2        | BLE dan ESP-NOW | Sedang (585 KB) | 1 m, 5 m, 10 m, 15 m | LOS dan NLOS       | 10              |
| 1        | BLE dan ESP-NOW | Besar (1,2 MB)  | 1 m, 5 m, 10 m, 15 m | LOS dan NLOS       | 10              |

Prosedur pelaksanaan dari skenario pengujian pada Tabel 4.15 dilakukan dengan urutan langkah sebagai berikut:

1. Menempatkan perangkat *node* sensor dan *gateway* pada titik jarak uji secara bertahap. Penempatan ini dilakukan untuk dua kondisi lingkungan, yakni kondisi tanpa hambatan fisik (LOS) di lorong terbuka dan kondisi terhalang dinding beton pembatas ruangan (NLOS).
2. Menjalankan skrip Python otomatisasi pengujian. Skrip akan meminta pengujian memilih protokol BLE atau ESP-NOW, kategori file, jarak, kondisi halangan, dan port *gateway*.
3. Skrip secara otomatis akan mengirimkan instruksi FOTA ke topik MQTT `'shm/ota/trigger'` secara berulang sebanyak 10 kali iterasi. Di setiap iterasi, *gateway* mengunduh biner *firmware* dari server HTTP lokal, melakukan streaming ke *node*, dan mencetak statistik hasil FOTA.
4. Skrip otomatis mem-parsing log keluaran serial *gateway*, menangkap baris hasil pengujian, lalu mencatat status keberhasilan, durasi waktu transmisi, ukuran total *bytes* terkirim, jumlah ekspektasi paket, dan total percobaan transmisi secara langsung ke dalam berkas `'.csv'` tujuan.
5. Mengulangi langkah 1 hingga 5 untuk semua titik jarak, kondisi hambatan, kategori berkas, dan protokol komunikasi yang diuji.

#### 4.4.3 Data Hasil Pengujian

Data hasil pengujian menunjukkan data-data hasil dari setiap pengujian yang telah dilakukan.



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

#### 4.4.3.1 Hasil Pengujian Fungsionalitas Sistem

Pengujian fungsionalitas sistem telah dilaksanakan menggunakan metode *black-box* testing berdasarkan prosedur yang telah ditetapkan. Hasil pengujian menunjukkan bahwa seluruh subsistem, mulai dari modul akuisisi tingkat rendah hingga antarmuka pengguna di tingkat awan atau *cloud*, dapat beroperasi dan terintegrasi secara normal tanpa memicu kegagalan sistem. Ringkasan dari hasil pengujian fungsionalitas ini dijabarkan pada Tabel 4.17 berikut:

Tabel 4.17 Hasil Pengujian Fungsionalitas Sistem

| No | Metode Pengujian                          | Hasil Pengamatan                                                                                                                                                                                                   | Kesimpulan |
|----|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 1  | Inisialisasi dan Verifikasi Koneksi Awal  | <i>Gateway</i> berhasil terhubung ke jaringan Wi-Fi dan <i>broker</i> MQTT secara aman (TLS). <i>Node</i> dan <i>gateway</i> sukses melakukan terhubung via BLE serta menegosiasikan MTU pada 512 <i>bytes</i> .   | Sesuai     |
| 2  | Simulasi Input Data Telemetry             | <i>Node</i> sensor berhasil merespon gangguan mekanis dan membaca nilai fluktuasi register X, Y, dan Z dari ADXL345 via komunikasi I2C secara kontinu.                                                             | Sesuai     |
| 3  | Validasi Antarmuka Web                    | Data telemetry yang dikemas dalam format JSON oleh <i>gateway</i> sukses diteruskan via MQTT dan berhasil dirender menjadi fluktuasi grafik <i>time-series</i> pada <i>web dashboard</i> secara <i>real-time</i> . | Sesuai     |
| 4  | Inisiasi Pembaruan <i>Firmware</i> (FOTA) | Penekanan tombol <i>Trigger</i> OTA pada antarmuka web berhasil mengubah status perangkat menjadi “IN_PROGRESS” dan menerbitkan instruksi yang valid ke <i>broker</i> MQTT.                                        | Sesuai     |
| 5  | Observasi Logika                          | <i>Gateway</i> sukses menangkap interupsi MQTT, mengunduh file .bin dari server                                                                                                                                    | Sesuai     |



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

| No | Metode Pengujian                                           | Hasil Pengamatan                                                                                                                                                                  | Kesimpulan |
|----|------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
|    | Transmisi<br><i>Gateway</i>                                | via protokol HTTP, dan mengalirkan potongan data tersebut ke <i>node</i> via BLE secara stabil.                                                                                   |            |
| 6  | Validasi<br>Eksekusi Partisi<br>dan Sinkronisasi<br>Status | <i>Node</i> sukses melakukan <i>reboot</i> ke partisi memori baru. <i>Gateway</i> berhasil memverifikasi kecocokan versi GATT, dan status web otomatis berubah menjadi “SUCCESS”. | Sesuai     |

Pembuktian secara komprehensif dari hasil ringkasan pada Tabel 4.17 diuraikan lebih lanjut ke dalam dua bagian pengujian fungsional utama, yaitu aliran data telemetri dan mekanisme pembaruan FOTA.

a) Hasil Pengujian Aliran Data Telemetri

Pengujian ini memvalidasi keandalan sistem dalam melakukan akuisisi dan perutean data. Gambar 4.35 menunjukkan tangkapan layar dari *serial monitor* pada perangkat *gateway*. Terlihat bahwa *gateway* berhasil mencetak log “[MQTT] Terhubung secara AMAN (TLS) ke EMQX!”, yang menandakan keberhasilan koneksi ke infrastruktur internet. Sesaat kemudian, perangkat berhasil memindai dan mencetak “KONEKSI BLE BERHASIL!”, diikuti dengan aliran data telemetri sumbu X, Y, Z dan tegangan baterai yang diterima secara periodik dari *node* sensor.



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
[WIFI] Terhubung! IP Address: 10.31.217.157

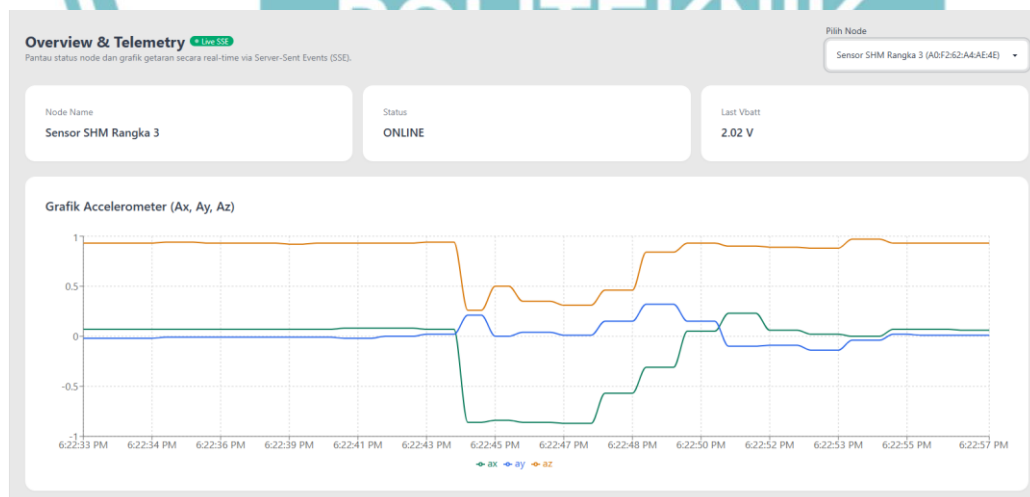
[MQTT] Terhubung secara AMAN (TLS) ke EMQX!
[MQTT] Melaporkan Status Gateway: ONLINE
I (27021) wifi:<ba-addr>idx:0 (ifx:0, 56:36:b6:de:66:7c), tid:1, ssn:16, wi

[GATEWAY] Target 'SHM_Node_C3' Ditemukan! Memulai koneksi...

=== KONEKSI BLE BERHASIL! Node MAC: A0:F2:62:A4:DE:72 di slot 0 ===
[BLE] MTU Sukses Dinegosiasikan! Kapasitas: 512 bytes
>> Mengaktifkan Notifikasi SHM (Subscribe)...
>> Subscribe Sukses!
>> Membuka gerbang pencarian untuk Node selanjutnya...
>> BLE Scanner dinyalakan (Mode Hemat Antena)...
[SHM] Node A0:F2:62:A4:DE:72 | AX:-105.99 AY:65.89 AZ:0.00 | Vbatt: 4.29V
[SHM] Node A0:F2:62:A4:DE:72 | AX:-105.99 AY:65.89 AZ:0.00 | Vbatt: 4.29V
[SHM] Node A0:F2:62:A4:DE:72 | AX:-105.99 AY:65.89 AZ:0.00 | Vbatt: 4.28V
[SHM] Node A0:F2:62:A4:DE:72 | AX:-105.99 AY:65.89 AZ:0.00 | Vbatt: 4.28V
[SHM] Node A0:F2:62:A4:DE:72 | AX:-105.99 AY:65.89 AZ:0.00 | Vbatt: 4.29V
[SHM] Node A0:F2:62:A4:DE:72 | AX:-105.99 AY:65.89 AZ:0.00 | Vbatt: 4.28V
```

Gambar 4.35 Respons Log *Gateway* saat Inisialisasi dan Penerimaan Data Telemetri

Data biner yang telah dikonversi dan dipublikasikan oleh *gateway* kemudian divalidasi pada antarmuka pengguna. Gambar 4.36 menampilkan halaman pemantauan telemetri pada web *dashboard*. Saat modul sensor ADXL345 diberikan gangguan fisik, grafik *time-series* merespons secara *real-time* dengan menampilkan fluktuasi puncak gelombang yang selaras dengan intensitas gangguan. Hal ini membuktikan bahwa tidak ada kemacetan data pada arsitektur jaringan yang dibangun.



Gambar 4.36 Fluktuasi Grafik Telemetri Berdasarkan Simulasi Gangguan Fisik

### b) Hasil Pengujian Mekanisme FOTA

Eksekusi dimulai dari antarmuka web, sebagaimana ditunjukkan pada Gambar 4.37. admin memicu pembaruan melalui tombol *Trigger* OTA, dan tabel pencatatan sistem



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

seketika memberikan umpan balik visual dengan mengubah status log dari kosong menjadi “IN\_PROGRESS”.

Gambar 4.37 Inisiasi Perintah FOTA pada Antarmuka Web

Berdasarkan instruksi tersebut, *gateway* merespons secara instan. Gambar 4.38 menunjukkan log serial monitor *gateway* saat menerima JSON perintah dan memulai proses eksekusi. Terlihat indikator pemrosesan HTTP yang berhasil mengidentifikasi ukuran berkas *firmware* target, diikuti dengan log “[GATEWAY OTA] Progres:” yang terus bertambah. Proses ini berjalan hingga sistem mencetak “[GATEWAY OTA] Streaming UTUH selesai! Mengirim perintah END...”, yang membuktikan bahwa fungsi fragmentasi 500 *bytes* berjalan sukses.

```
[MQTT] TRIGGER OTA DITERIMA! NODE: A0:F2:62:A4:DE:72
[MQTT] Melaporkan Status OTA Node A0:F2:62:A4:DE:72: IN_PROGRESS

[GATEWAY OTA] Memulai proses Streaming dari Server ke Node A0:F2:62:A4:DE:72...
[GATEWAY OTA] Mematikan background scanner sementara...
[GATEWAY OTA] Ukuran Firmware dari header: 585536 bytes. Mulai streaming...
[GATEWAY OTA] Progres: 10000 / 585536 bytes...
[GATEWAY OTA] Progres: 20000 / 585536 bytes...
[GATEWAY OTA] Progres: 30000 / 585536 bytes...
```

Gambar 4.38 Proses Transmisi *Firmware* dari *Gateway* menuju *Node*

Sebagai tahap finalisasi yang menjamin integritas pembaruan, Gambar 4.39 membuktikan keberhasilan sinkronisasi sistem pasca-*reboot*. Setelah *node* sensor menyala kembali dan beroperasi menggunakan partisi *firmware* baru, *gateway* secara



#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

otomatis melakukan pemeriksaan versi silang dan mencetak log “[VERIFIKASI] SUKSES! Versi cocok..”.

```
>> BLE Scanner dinyalakan (Mode Hemat Antena)...

[GATEWAY] Target 'SHM_Node_C3' Ditemukan! Memulai koneksi...

=== KONEKSI BLE BERHASIL! Node MAC: A0:F2:62:A4:DE:72 di slot 0 ===
[BLE] MTU Sukses Dinegosiasikan! Kapasitas: 512 bytes
>> Mengaktifkan Notifikasi SHM (Subscribe)...
>> Subscribe Sukses!
>> Membuka gerbang pencarian untuk Node selanjutnya...
>> BLE Scanner dinyalakan (Mode Hemat Antena)...

[VERIFIKASI] Node MAC A0:F2:62:A4:DE:72 melapor versi: 'v1.0.1'
[VERIFIKASI] SUKSES! Versi cocok.
[MQTT] Melaporkan Status OTA Node A0:F2:62:A4:DE:72: SUCCESS
[SHM] Node A0:F2:62:A4:DE:72 | AX: -105.99 AY: 65.89 AZ: 0.00 | Vbatt: 4.19V
```

Gambar 4.39 Verifikasi Integritas Versi *Firmware* Pasca-*Reboot* oleh *Gateway*

Keberhasilan validasi tingkat mikrokontroler ini dilaporkan kembali ke *server*. Seperti yang terlihat pada Gambar 4.40, antarmuka web berhasil menerjemahkan paket pelaporan MQTT dari *gateway* dan secara otomatis mengubah log status dari yang sebelumnya “IN\_PROGRESS” menjadi “SUCCESS”. Urutan keberhasilan fungsional dari ujung ke ujung ini mengonfirmasi bahwa rancangan arsitektur perutean *streaming* FOTA sangat layak dan siap untuk diuji kinerja efisiensinya.

|           |                                            |        |                     |
|-----------|--------------------------------------------|--------|---------------------|
| ✓ SUCCESS | Node Ruang Kamar Rafi<br>1C:DB:D4:C6:3C:56 | v1.0.1 | 11/5/2026, 11.14.27 |
|-----------|--------------------------------------------|--------|---------------------|

Gambar 4.40 Sinkronisasi Status Keberhasilan OTA pada Tabel Dashboard

#### 4.4.3.2 Hasil Pengujian Kinerja Transmisi Berkas Kecil

Tabel 4.18 di bawah ini merangkum tingkat keberhasilan, rata-rata waktu, throughput aktual, dan rasio retransmisi berkas biner *firmware* kategori kecil.

Tabel 4.18 Data Pengujian Transmisi Berkas Kecil

| Protokol | Jarak | Hambatan | Success Rate | Rata-Rata Waktu (s) | Throughput  | Total Paket (Retry) | Rasio Retransmisi (%) |
|----------|-------|----------|--------------|---------------------|-------------|---------------------|-----------------------|
| BLE      | 1m    | LOS      | 100%         | 8.65 s              | 138.72 Kbps | 304.9               | 101.63 %              |



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

| Protokol | Jarak | Hambatan | Success Rate | Rata-Rata Waktu (s) | Throughput  | Total Paket (Retry) | Rasio Retransmisi (%) |
|----------|-------|----------|--------------|---------------------|-------------|---------------------|-----------------------|
|          | 5m    | NLOS     | 100%         | 9.31 s              | 129.10 Kbps | 375.8               | 125.27 %              |
|          |       | LOS      | 100%         | 9.82 s              | 123.44 Kbps | 416.3               | 138.77 %              |
|          |       | NLOS     | 100%         | 18.35 s             | 67.14 Kbps  | 1273.9              | 424.63 %              |
|          | 10m   | LOS      | 100%         | 12.46 s             | 97.02 Kbps  | 678.6               | 226.20 %              |
|          |       | NLOS     | 90%          | 48.59 s             | 35.27 Kbps  | 4296.6              | 1432.19 %             |
|          | 15m   | LOS      | 100%         | 16.30 s             | 75.55 Kbps  | 1002.2              | 334.07 %              |
|          |       | NLOS     | 0%           | -                   | -           | -                   | -                     |
|          | 1m    | LOS      | 100%         | 7.27 s              | 165.19 Kbps | 683.0               | 100.15 %              |
|          |       | NLOS     | 100%         | 8.61 s              | 148.19 Kbps | 685.7               | 100.54 %              |
|          | 5m    | LOS      | 100%         | 11.09 s             | 114.89 Kbps | 683.7               | 100.25 %              |
|          |       | NLOS     | 90%          | 8.78 s              | 137.89 Kbps | 684.0               | 100.29 %              |
| ESP-NOW  | 10m   | LOS      | 100%         | 21.51 s             | 79.75 Kbps  | 694.1               | 101.77 %              |
|          |       | NLOS     | 90%          | 19.03 s             | 82.89 Kbps  | 699.2               | 102.53 %              |
|          | 15m   | LOS      | 90%          | 17.94 s             | 91.36 Kbps  | 694.0               | 101.76 %              |
|          |       | NLOS     | 0%           | -                   | -           | -                   | -                     |

Berdasarkan data pengujian berkas Kecil di atas, protokol ESP-NOW menunjukkan keunggulan performa kecepatan transmisi yang signifikan dibandingkan BLE. Pada kondisi ideal 1 meter LOS, ESP-NOW mencapai throughput 165.19 Kbps berbanding BLE sebesar 138.72 Kbps. Ketika diuji dengan hambatan fisik 10 meter NLOS, performa BLE terdegradasi tajam dengan throughput drop ke 35.27 Kbps dan lonjakan rasio retransmisi hingga 1432.19 % akibat sensitivitas link layer BLE terhadap redaman beton, sedangkan ESP-NOW tetap mempertahankan stabilitas throughput 82.89 Kbps. Pada batas jangkauan terjauh 15 meter NLOS kedua protokol mengalami kegagalan transfer total dengan success rate 0%, namun pada kondisi tanpa halangan



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

15 meter LOS ESP-NOW tetap mampu menjaga keandalan pengiriman dengan tingkat keberhasilan sebesar 90% berbanding BLE sebesar 100%.

#### 4.4.3.3 Hasil Pengujian Kinerja Transmisi Berkas Sedang

Tabel 4.19 di bawah ini merangkum tingkat keberhasilan, rata-rata waktu, throughput aktual, dan rasio retransmisi berkas biner firmware kategori Sedang:

Tabel 4.19 Data Pengujian Transmisi Berkas Sedang

| Protokol | Jarak | Hambatan | Success Rate | Rata-Rata Waktu (s) | Throughput Aktual (Kbps) | Total Paket (Retry) | Rasio Retransmisi (%) |
|----------|-------|----------|--------------|---------------------|--------------------------|---------------------|-----------------------|
| BLE      | 1m    | LOS      | 100%         | 44.29 s             | 105.84 Kbps              | 1218.4              | 104.05 %              |
|          |       | NLOS     | 100%         | 47.24 s             | 99.94 Kbps               | 1165.3              | 99.51 %               |
|          | 5m    | LOS      | 100%         | 54.54 s             | 89.85 Kbps               | 2174.8              | 185.72 %              |
|          |       | NLOS     | 100%         | 80.30 s             | 62.08 Kbps               | 4828.0              | 412.30 %              |
|          | 10m   | LOS      | 100%         | 88.37 s             | 54.65 Kbps               | 5616.5              | 479.63 %              |
|          |       | NLOS     | 80%          | 134.11 s            | 37.37 Kbps               | 10183.4             | 869.63 %              |
|          | 15m   | LOS      | 80%          | 94.91 s             | 54.33 Kbps               | 6259.9              | 534.58 %              |
|          |       | NLOS     | 0%           | -                   | -                        | -                   | -                     |
| ESP-NOW  | 1m    | LOS      | 100%         | 25.28 s             | 185.93 Kbps              | 2663.0              | 100.04 %              |
|          |       | NLOS     | 100%         | 37.64 s             | 126.37 Kbps              | 2663.9              | 100.07 %              |
|          | 5m    | LOS      | 100%         | 25.10 s             | 186.79 Kbps              | 2663.0              | 100.04 %              |
|          |       | NLOS     | 100%         | 55.21 s             | 96.52 Kbps               | 2686.2              | 100.91 %              |
|          | 10m   | LOS      | 100%         | 30.33 s             | 156.76 Kbps              | 2666.9              | 100.18 %              |
|          |       | NLOS     | 70%          | 58.32 s             | 90.13 Kbps               | 2487.3              | 101.56 %              |
|          | 15m   | LOS      | 100%         | 29.31 s             | 161.43 Kbps              | 2666.0              | 100.15 %              |
|          |       | NLOS     | 0%           | -                   | -                        | -                   | -                     |

Berdasarkan data pengujian berkas Sedang di atas, protokol ESP-NOW menunjukkan keunggulan performa kecepatan transmisi yang signifikan dibandingkan BLE. Pada



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

kondisi ideal 1 meter LOS, ESP-NOW mencapai throughput 185.93 Kbps berbanding BLE sebesar 105.84 Kbps. Ketika diuji dengan hambatan fisik 10 meter NLOS, performa BLE terdegradasi tajam dengan throughput drop ke 37.37 Kbps dan lonjakan rasio retransmisi hingga 869.63 % akibat sensitivitas link layer BLE terhadap redaman beton, sedangkan ESP-NOW tetap mempertahankan stabilitas throughput 90.13 Kbps. Pada batas jangkauan terjauh 15 meter NLOS kedua protokol mengalami kegagalan transfer total, namun pada kondisi tanpa halangan 15 meter LOS ESP-NOW tetap mampu menjaga keandalan pengiriman dengan tingkat keberhasilan sebesar 100% berbanding BLE sebesar 80%.

#### 4.4.3.4 Hasil Pengujian Kinerja Transmisi Berkas Besar

Tabel 4.20 di bawah ini merangkum tingkat keberhasilan, rata-rata waktu, throughput aktual, dan rasio retransmisi berkas biner firmware kategori besar:

Tabel 4.20 Data Pengujian Transmisi Berkas Besar

| Protokol | Jarak | Hambatan | Success Rate | Rata-Rata Waktu (s) | Throughput Aktual (Kbps) | Total Paket (Retry) | Rasio Retransmisi (%) |
|----------|-------|----------|--------------|---------------------|--------------------------|---------------------|-----------------------|
| BLE      | 1m    | LOS      | 100%         | 69.78 s             | 137.78 Kbps              | 2593.8              | 108.08 %              |
|          |       | NLOS     | 100%         | 79.47 s             | 121.41 Kbps              | 3162.9              | 131.79 %              |
|          | 5m    | LOS      | 100%         | 80.25 s             | 120.14 Kbps              | 2890.0              | 120.42 %              |
|          |       | NLOS     | 100%         | 160.33 s            | 60.61 Kbps               | 11227.4             | 467.81 %              |
|          | 10m   | LOS      | 100%         | 100.24 s            | 101.00 Kbps              | 5625.3              | 234.39 %              |
|          |       | NLOS     | 100%         | 236.23 s            | 42.47 Kbps               | 19485.1             | 811.88 %              |
|          | 15m   | LOS      | 100%         | 91.88 s             | 105.97 Kbps              | 4892.3              | 203.85 %              |
|          |       | NLOS     | 0%           | -                   | -                        | -                   | -                     |
| ESP-NOW  | 1m    | LOS      | 100%         | 70.56 s             | 138.49 Kbps              | 5456.1              | 100.02 %              |
|          |       | NLOS     | 100%         | 81.18 s             | 131.24 Kbps              | 5486.5              | 100.58 %              |
|          | 5m    | LOS      | 100%         | 62.61 s             | 159.22 Kbps              | 5458.8              | 100.07 %              |
|          |       | NLOS     | 90%          | 128.26 s            | 79.70 Kbps               | 5546.3              | 101.67 %              |



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

| Protokol | Jarak | Hambatan | Success Rate | Rata-Rata Waktu (s) | Throughput Aktual (Kbps) | Total Paket (Retry) | Rasio Retransmisi (%) |
|----------|-------|----------|--------------|---------------------|--------------------------|---------------------|-----------------------|
|          | 10m   | LOS      | 100%         | 94.46 s             | 116.36 Kbps              | 5498.9              | 100.80 %              |
|          |       | NLOS     | 100%         | 98.25 s             | 101.10 Kbps              | 5510.2              | 101.01 %              |
|          | 15m   | LOS      | 100%         | 100.71 s            | 104.78 Kbps              | 5499.6              | 100.82 %              |
|          |       | NLOS     | 0%           | -                   | -                        | -                   | -                     |

Berdasarkan data pengujian berkas Besar di atas, protokol ESP-NOW menunjukkan keunggulan performa kecepatan transmisi yang signifikan dibandingkan BLE. Pada kondisi ideal 1 meter LOS, ESP-NOW mencapai throughput 138.49 Kbps berbanding BLE sebesar 137.78 Kbps. Ketika diuji dengan hambatan fisik, performa BLE terdegradasi tajam dengan throughput drop ke 42.47 Kbps dan lonjakan rasio retransmisi hingga 811.88 % akibat sensitivitas link layer BLE terhadap redaman beton, sedangkan ESP-NOW tetap mempertahankan stabilitas throughput 101.10 Kbps. Pada batas jangkauan terjauh 15 meter NLOS kedua protokol mengalami kegagalan transfer total, namun pada kondisi tanpa halangan 15 meter LOS ESP-NOW tetap mampu menjaga keandalan pengiriman dengan tingkat keberhasilan sebesar 100% berbanding BLE sebesar 100%.

#### 4.4.4 Analisis Data dan Evaluasi Pengujian

Analisis data dilakukan berdasarkan data observasi hasil pengujian yang telah dilakukan. Pengujian dilaksanakan untuk mengevaluasi sistem pembaruan ESP32 berbasis Bluetooth Low Energi dengan arsitektur *gateway*.

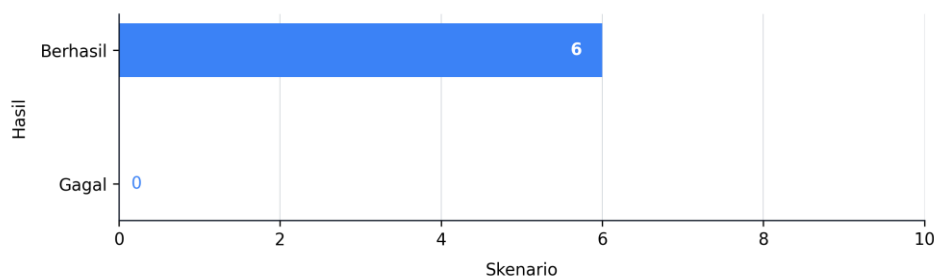
##### 4.4.4.1 Analisis Pengujian Fungsionalitas Sistem

Berdasarkan hasil eksekusi pengujian fungsionalitas yang telah dilakukan, tingkat keberhasilan fungsi sistem divisualisasikan pada Gambar 4.41 di bawah ini.



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 4.41 Grafik Hasil Pengujian Fungsionalitas Sistem

Tingkat kegagalan dari pengujian fungsionalitas sistem dihitung menggunakan rumus persentase (1):

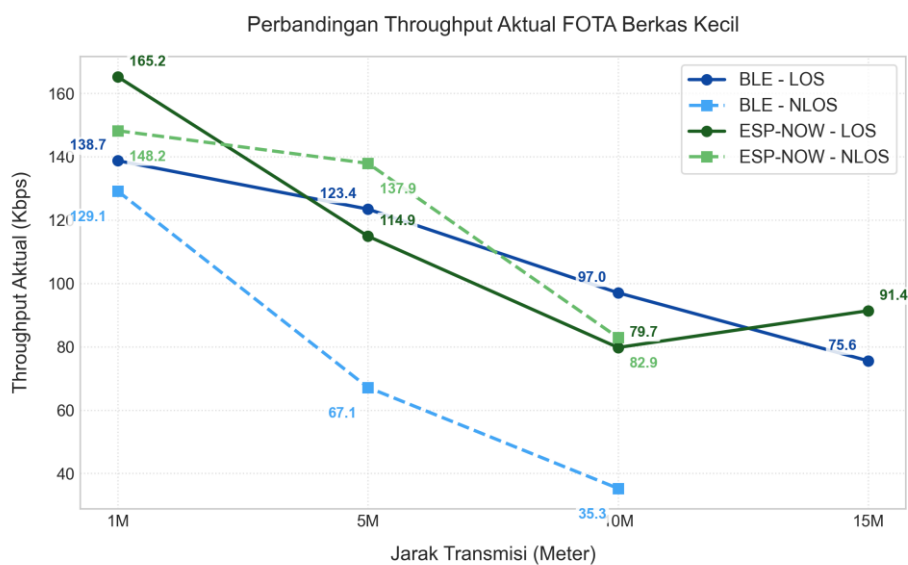
$$\text{Persentase Error} = \frac{\text{Jumlah Skenario Gagal}}{\text{Total Pengujian}} \times 100\%$$

$$\text{Persentase Error} = \frac{0}{6} \times 100\% = 0\%$$

Hasil perhitungan di atas membuktikan bahwa sistem yang dikembangkan memiliki tingkat kegagalan sebesar 0% yang berarti keberhasilan fungsional sistem sebesar 100%.

#### 4.4.4.2 Analisis Pengujian Kinerja Transmisi Berkas Kecil

Data pengujian kinerja jaringan pada berkas kecil 150KB yang telah didapat dianalisis dengan grafik visual komparasi.



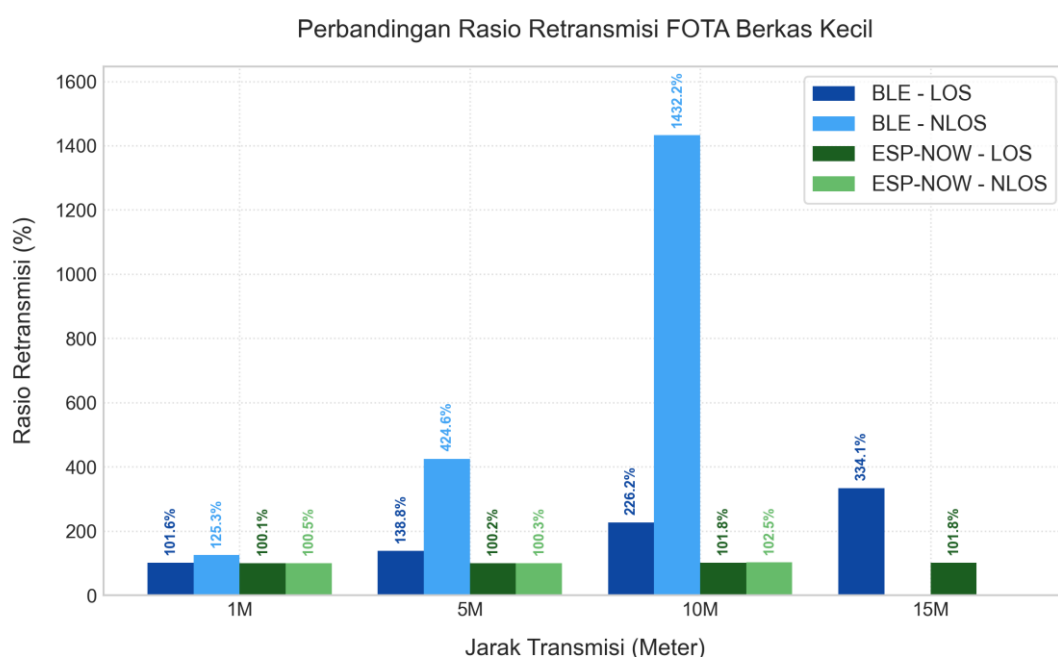
Gambar 4.42 Grafik Komparasi Throughput Berkas Kecil BLE vs ESP-NOW



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Gambar 4.42 memperlihatkan perbandingan throughput antara protokol BLE dan ESP-NOW. Pada kondisi ideal 1 meter LOS, ESP-NOW mencatatkan throughput tertinggi sebesar 165.19 Kbps berbanding BLE yang mencapai 138.72 Kbps. Seiring bertambahnya jarak dan adanya hambatan pada 10 meter NLOS, throughput BLE mengalami penurunan drastis hingga menyusut ke 35.27 Kbps, sedangkan ESP-NOW tetap menunjukkan performa transmisi nirkabel yang stabil dengan mempertahankan throughput sebesar 82.89 Kbps. Pada batas ekstrim 15M NLOS kedua protokol mengalami pemutusan komunikasi total, namun pada kondisi tanpa hambatan 15 meter LOS ESP-NOW tetap andal menjaga throughput 91.36 Kbps dibanding BLE sebesar 75.55 Kbps.



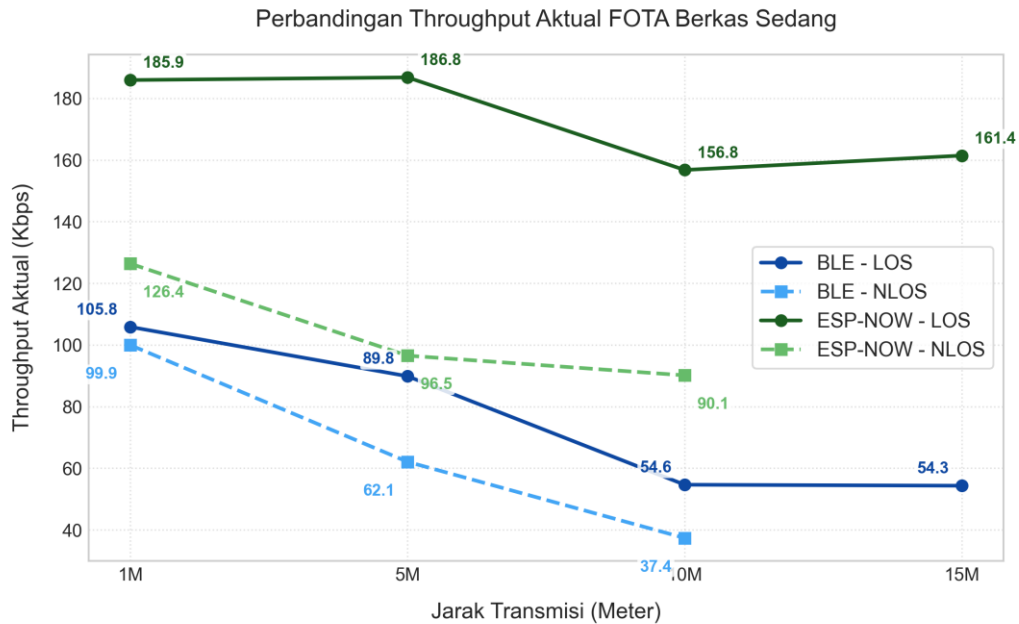
Gambar 4.43 Grafik Komparasi Rasio Retransmisi Berkas Kecil BLE vs ESP-NOW

Gambar 4.43 memaparkan perbandingan tingkat kestabilan pengiriman paket data melalui metrik rasio retransmisi. Protokol ESP-NOW menunjukkan rasio retransmisi aplikasi yang sangat stabil, berkisar konstan antara 100.15 % pada 1 meter LOS hingga 102.53% pada 10M NLOS. Sebaliknya, protokol BLE sangat sensitif terhadap hambatan beton di mana rasionya melonjak secara dramatis dari 101.63 % pada 1 meter LOS menjadi 1432.19% pada 10 meter NLOS dengan total pengiriman membengkak rata-rata hingga 4296.6 paket.



#### 4.4.4.3 Analisis Pengujian Kinerja Transmisi Berkas Sedang

Data pengujian kinerja jaringan pada berkas sedang 585KB yang telah didapat dianalisis dengan grafik visual komparasi.



Gambar 4.44 Grafik Komparasi Throughput Berkas Sedang BLE vs ESP-NOW

Gambar 4.44 memperlihatkan perbandingan throughput antara protokol BLE dan ESP-NOW. Pada kondisi ideal 5 meter LOS, ESP-NOW mencatatkan throughput tertinggi sebesar 186.8 Kbps dibanding BLE yang mencapai 105.84 Kbps. Seiring bertambahnya jarak dan adanya hambatan seperti pada 10 meter NLOS, throughput BLE mengalami degradasi drastis hingga menyusut ke 37.37 Kbps, sedangkan ESP-NOW tetap menunjukkan performa transmisi nirkabel yang stabil dengan mempertahankan throughput sebesar 90.13 Kbps. Pada batas ekstrim 15M NLOS kedua protokol mengalami pemutusan komunikasi total, namun pada kondisi tanpa hambatan 15 meter LOS ESP-NOW tetap andal menjaga throughput 161.43 Kbps dibanding BLE sebesar 54.33 Kbps.

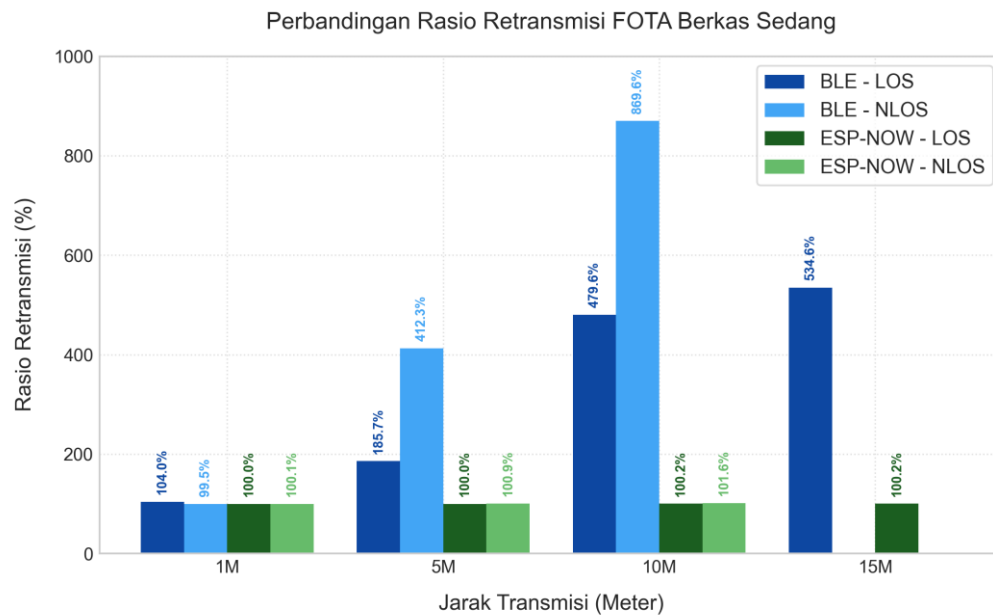
#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 4.45 Grafik Komparasi Rasio Retransmisi Berkas Sedang BLE vs ESP-NOW

Gambar 4.45 memaparkan perbandingan tingkat kestabilan pengiriman paket data melalui metrik rasio retransmisi. Protokol ESP-NOW menunjukkan rasio retransmisi aplikasi yang sangat stabil, berkisar konstan antara 100.04 % pada 1 meter LOS hingga 101.56% pada 10 meter NLOS. Sebaliknya, protokol BLE sangat sensitif terhadap hambatan beton di mana rasionya melonjak secara dramatis dari 104.05% pada 1 meter LOS menjadi 869.63% pada 10 meter NLOS menyebabkan total pengiriman membengkak rata-rata hingga 10183.4 paket.

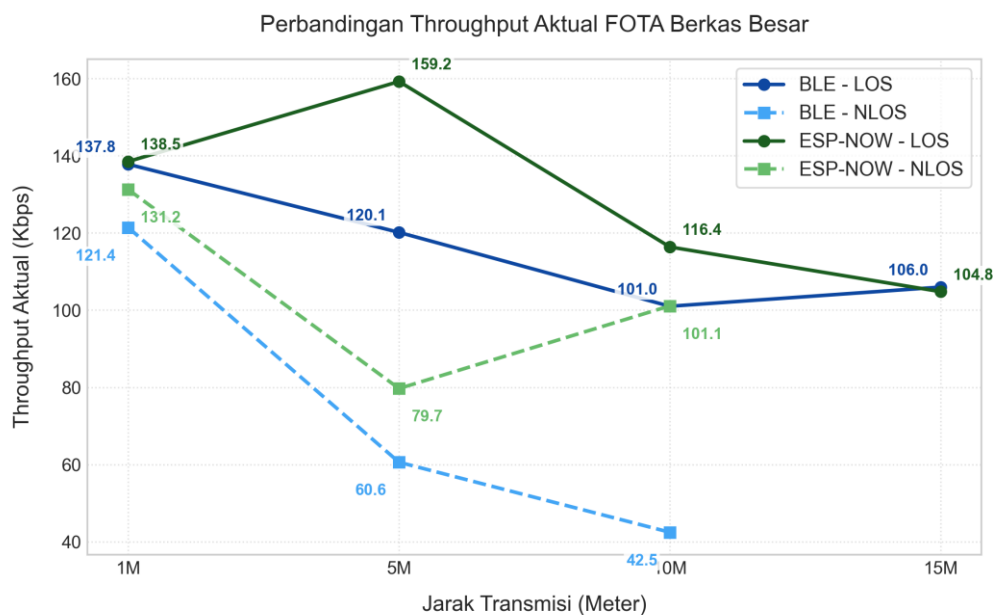
#### 4.4.4.4 Analisis Pengujian Kinerja Transmisi Berkas Besar

Data pengujian kinerja jaringan pada berkas besar 1,2 MB yang telah didapat dianalisis dengan grafik visual komparasi.



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



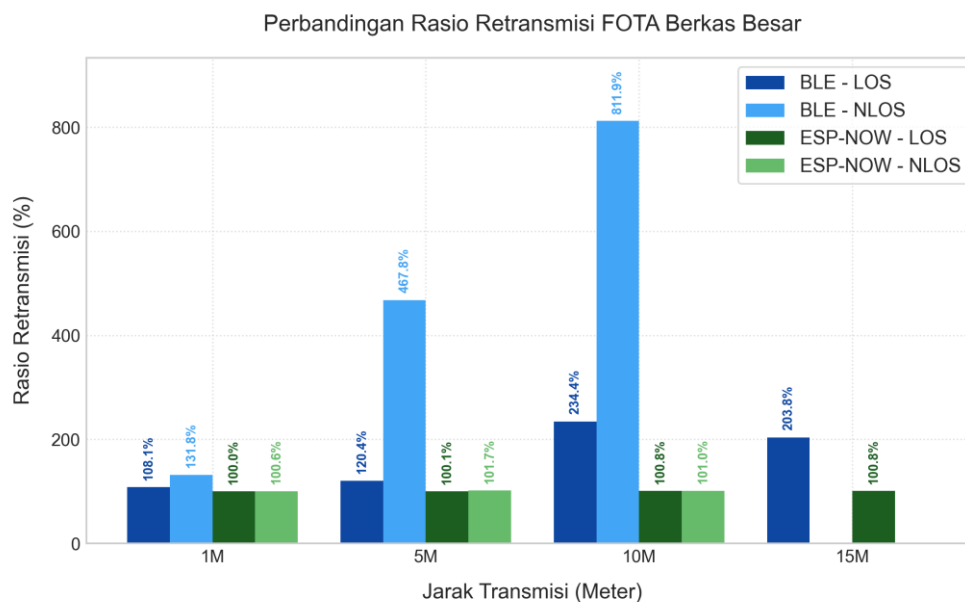
Gambar 4.46 Grafik Komparasi Throughput Berkas Besar BLE vs ESP-NOW

Gambar 4.46 memperlihatkan perbandingan throughput aktual antara protokol BLE dan ESP-NOW. Pada kondisi ideal 1 meter LOS, ESP-NOW mencatatkan throughput tertinggi sebesar 138.49 Kbps berbanding BLE yang mencapai 137.78 Kbps. Seiring bertambahnya jarak dan adanya hambatan pada 10 meter NLOS, throughput BLE mengalami degradasi drastis hingga menyusut ke 42.47 Kbps, sedangkan ESP-NOW tetap menunjukkan performa transmisi nirkabel yang stabil dengan mempertahankan throughput sebesar 101.10 Kbps. Pada batas ekstrim 15 meter NLOS kedua protokol mengalami pemutusan komunikasi total, namun pada kondisi tanpa hambatan 15M LOS ESP-NOW tetap andal menjaga throughput 104.78 Kbps dibanding BLE sebesar 105.97 Kbps.



### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



Gambar 4.47 Grafik Komparasi Rasio Retransmisi Berkas Besar BLE vs ESP-NOW

Gambar 4.47 memaparkan perbandingan tingkat kestabilan pengiriman paket data melalui metrik rasio retransmisi. Protokol ESP-NOW menunjukkan rasio retransmisi aplikasi yang sangat stabil, berkisar konstan antara 100.02% pada 1 meter LOS hingga 101.01% pada 10 meter NLOS. Sebaliknya, protokol BLE sangat sensitif terhadap hambatan beton di mana rasionya melonjak secara dramatis dari 108.08% pada 1 meter LOS menjadi 811.88 % pada 10 meter NLOS, hal ini menyebabkan total pengiriman membengkak rata-rata hingga 19485.1 paket.

**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## BAB V PENUTUP

### 5.1 Simpulan

Berdasarkan hasil perancangan, implementasi, dan analisis pengujian sistem *Firmware Over-the-Air* pada jaringan sensor nirkabel menggunakan arsitektur *Gateway* BLE, dapat diambil simpulan sebagai berikut:

1. Mekanisme FOTA dengan arsitektur *gateway* berbasis BLE 5.0 berhasil diimplementasikan secara asinkron mulai dari *cloud storage*, *backend* Go, broker MQTT EMQX, hingga transmisi data nirkabel dari *gateway* ESP32-S3 ke *node* sensor ESP32-C3 menggunakan protokol BLE secara stabil.
2. Pengujian throughput membuktikan keunggulan protokol ESP-NOW secara konsisten di semua ukuran berkas dibandingkan dengan BLE. Pada kondisi ideal pada 1 meter LOS, ESP-NOW mencatatkan throughput rata-rata hingga 165,19 Kbps pada berkas 150 KB, 185,93 Kbps pada berkas 585 KB, dan 138,49 Kbps pada berkas 1,2 MB berbanding BLE yang hanya mencapai 138,72 Kbps pada berkas 150 KB, 105,84 pada berkas 585 KB, dan 137,78 Kbps pada berkas sebesar 1,2 MB.
3. Protokol BLE sangat sensitif terhadap hambatan beton, di mana rasio retransmisi BLE melonjak tajam hingga 1432,19% pada berkas 150 KB dan 869,63% pada berkas 585 KB di jarak 10 meter NLOS, sedangkan ESP-NOW tetap mempertahankan kestabilan rasio retransmisi di kisaran 100% s.d. 102%. Pada jarak ekstrim 15 meter NLOS kedua protokol mengalami kegagalan transfer total, namun pada kondisi tanpa halangan yaitu 15 meter LOS ESP-NOW menunjukkan keandalan transfer yang lebih baik dibandingkan dengan BLE.

### 5.2 Saran

Saran yang dapat diimplementasikan pada penelitian selanjutnya, antara lain:

1. Sistem dapat dikembangkan menerapkan algoritma penyesuaian parameter interval koneksi secara dinamis yang digabungkan dengan BLE *mesh* untuk memperkuat ketahanan sinyal di lingkungan berpenghalang.



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

2. Perlu diuji coba menggunakan lebih banyak node yang terintegrasi dengan satu *gateway* yang sama untuk melihat kemampuan koneksi dan transmisi data protokol Bluetooth Low Energy pada ESP32-S3 secara bersamaan.
3. Pengujian FOTA komparatif selanjutnya disarankan menggunakan variasi protokol nirkabel berdaya rendah lainnya seperti Zigbee atau LoRaWAN untuk memetakan performa transmisi pada cakupan topologi jaringan sensor yang lebih luas.





**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

**DAFTAR PUSTAKA**

- Aprianto, A. D., & Ramayanti, D. (2024). Penggunaan Klasifikasi Objek dalam Aplikasi Android untuk Melestarikan Kuliner Khas Indonesia. *Jurnal Ilmiah FIFO*, 16(1), 25. <https://doi.org/10.22441/fifo.2024.v16i1.003>
- Arakadakis, K., Charalampidis, P., Makrogiannakis, A., & Fragkiadakis, A. (2022). Firmware Over-the-air Programming Techniques for IoT Networks- A Survey. *ACM Computing Surveys*, 54(9). <https://doi.org/10.1145/3472292>
- Ayuningtyas, P. K., Atmodjo, D., & Rachmadi, P. (2023). Performance And Functional Testing With The Black Box Testing Method. *International Journal of Progressive Sciences and Technologies (IJPSAT)*, 39(2), 212–218. <http://portal.perbanas.id>.
- Baazaoui, M. K., Ketata, I., Fakhfakh, A., & Derbel, F. (2023). Modeling of Packet Error Rate Distribution Based on Received Signal Strength Indications in OMNeT++ for Wake-Up Receivers. *Sensors*, 23(5). <https://doi.org/10.3390/s23052394>
- Bauwens, J., Ruckebusch, P., Giannoulis, S., Moerman, I., & De Poorter, E. (2020). Over-the-Air Software Updates in the Internet-of-Things: An Overview of Key Principles. *IEEE Communications Magazine*. <https://doi.org/10.1109/MCOM.001.1900125>
- Behnke, I. (2024). *Real-Time Aware IP-Networking for Resource-Constrained Embedded Devices*. <https://doi.org/10.14279/depositonce-20667>
- Braten, A. E., Kraemer, F. A., & Palma, D. (2020). Autonomous IoT Device Management Systems: Structured Review and Generalized Cognitive Model. *IEEE Internet of Things Journal*. <https://doi.org/10.1109/JIOT.2020.3035389>
- Cahyono, E. B., Setia Budi, A., & Akbar, S. R. (2023). Pengembangan Gateway untuk Mendukung Proses Over The Air Firmware Update pada Perangkat WSN berbasis Bluetooth Low Energy. *Jurnal Pengembangan Teknologi*



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

*Informasi Dan Ilmu Komputer*, 7(2), 953–959. <https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/12354>

Chakraborty, A., Sadhukhan, P., Das, P. K., & Roy, N. (2024). *Statistical Real-time Power Consumption Analysis of Wi-Fi and BLE Scan Operation: A Comparative Study*. <https://doi.org/10.21203/rs.3.rs-4696654/v1>

Dodik Kurniawan, M., & Budjianto, A. (2025). IMPLEMENTASI SISTEM PROGRAMABLE TIMER MENGGUNAKAN ESP32-C3 SUPERMINI DENGAN TAMPILAN OLED DAN INDIKATOR BUZZER. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 9, Number 6). <https://doi.org/10.36040/jati.v9i6.15785>

Ferrina, Q., Ratna Sulistiyanti, S., & Junaidi, dan. (2022). Hasil Pengujian Kalibrasi Sensor Akselerometer ADXL345. In *Jurnal Teori dan Aplikasi Fisika* (Vol. 10, Number 02).

Formanek, L., Kubascik, M., Karpis, O., & Kolok, P. (2025). Advanced System for Remote Updates on ESP32-Based Devices Using Over-the-Air Update Technology. *Computers*, 14(12). <https://doi.org/10.3390/computers14120531>

Hakim, L., Kusuma, W. A., Faiqurahman, M., & Supriyanto. (2021). Over The Air Update Firmware pada Perangkat IoT Dengan Protokol MQTT. *Jurnal Sistem Dan Informatika (JSI)*, 14(2), 99–105. <https://doi.org/10.30864/jsi.v14i2.244>

Hariyadi, A., Taufik, M., Hidayati, N., Eka Rakhmania, A., & Hendra Yoga Perdana, R. (n.d.). *Efisiensi Daya Perangkat Wireless Sensor Network Pada Penerangan Jalan Umum (PJU) Berbasis Algoritma Leach*.

Jaouhari, S. El, & Bouvet, E. (2022, January 26). *Toward a generic and secure bootloader for IoT device firmware OTA update*. <https://doi.org/10.1109/ICOIN53446.2022.9687242>



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

- Liu, C., Zhang, Y., & Zhou, H. (2021). A comprehensive study of bluetooth low energy. *Journal of Physics: Conference Series*, 2093(1). <https://doi.org/10.1088/1742-6596/2093/1/012021>
- Mahfoudhi, F., Sultania, A. K., & Famaey, J. (2022). Over-the-Air Firmware Updates for Constrained NB-IoT Devices. *Sensors*, 22(19). <https://doi.org/10.3390/s22197572>
- Mazzi, Y., Gaga, A., & Errahimi, F. (2021). Benchmarking and Comparison of Two Open-source RTOSs for Embedded Systems Based on ARM Cortex-M4 MCU. *Indian Journal of Science and Technology*, 14(16), 1261–1273. <https://doi.org/10.17485/IJST/v14i16.387>
- Milić, L., Jelenković, L., & Magdalenic, I. (2025). Minimizing IoT Energy Consumption Using BLE and Metaprotocol Architecture. *International Journal of Distributed Sensor Networks*, 2025(1). <https://doi.org/10.1155/dsn/7859748>
- Mostafa, J., Wehbi, S., Chilingaryan, S., & Kopmann, A. (2022, August 23). SciTS: A Benchmark for Time-Series Databases in Scientific Experiments and Industrial Internet of Things. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/3538712.3538723>
- Mostafa, M. A., Mohamed, M. K., & Ezzat, R. W. (2025). Enhancing AUTOSAR-Based Firmware Over-the-Air Updates in the Automotive Industry with a Practical Implementation on a Steering System. <http://arxiv.org/abs/2503.05839>
- Nakakaze, O., Koren, I., Brillowski, F., & Klamma, R. (2024). Adaptive retrofitting for industrial machines: utilizing webassembly and peer-to-peer connectivity on the edge. *World Wide Web*, 27(1). <https://doi.org/10.1007/s11280-024-01237-8>
- Nemlaha, E., Střelec, P., Horák, T., Kováč, S., & Tanuška, P. (2023). Suitability of MQTT and REST Communication Protocols for AIoT or IIoT Devices Based on ESP32 S3. *Lecture Notes in Networks and Systems*, 596 LNNS, 225–233. [https://doi.org/10.1007/978-3-031-21435-6\\_19](https://doi.org/10.1007/978-3-031-21435-6_19)



**Hak Cipta :**

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Nguyen, H. D., Sommer, N. Le, & Mahéo, Y. (2024). Over-the-Air Firmware Update in LoRaWAN Networks: A New Module-based Approach. *Procedia Computer Science*, 241, 154–161.

<https://doi.org/10.1016/j.procs.2024.08.022>

Peter, C. S., Penning, L., Zimpeck, A., Marques, F., & Yamin, A. (2023). An Approach to Remote Update Embedded Systems in the Internet of Things. *Journal of Internet Services and Applications*, 14(1), 151–159. <https://doi.org/10.5753/jisa.2023.3078>

Putra, I. B. A. E. M., Adnyana, M. S. I. D., & Jasa, L. (2021). Analisis Quality of Service Pada Jaringan Komputer. *Majalah Ilmiah Teknologi Elektro*, 20(1), 95. <https://doi.org/10.24843/mite.2021.v20i01.p11>

Skýpalová, E., Boroš, M., Loveček, T., & Veľas, A. (2025). Innovative Indoor Positioning: BLE Beacons for Healthcare Tracking. *Electronics (Switzerland)*, 14(10). <https://doi.org/10.3390/electronics14102018>

Subagiyo, H., Rhamadani, R., & Wahyuni, R. T. (2023). Penghematan Daya Pada Sensor Node Sistem Monitoring Kualitas Udara. *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, 13(2). <https://doi.org/10.22146/ijeis.81886>

Ul Haq, S., Singh, Y., Sharma, A., Gupta, R., & Gupta, D. (2023). A survey on IoT & embedded device firmware security: architecture, extraction techniques, and vulnerability analysis frameworks. In *Discover Internet of Things* (Vol. 3, Number 1). Springer Nature. <https://doi.org/10.1007/s43926-023-00045-2>

## DAFTAR RIWAYAT HIDUP

### Rafi Rashya Husein Panjaitan



Lahir pada tahun 2004, saat ini berdomisili di Jakarta. Putra kedua dari tiga bersaudara. Lulus dari pendidikan Sekolah Dasar pada tahun 2016, Sekolah Menengah Pertama pada tahun 2019, dan Sekolah Menengah Atas dengan jurusan pada tahun 2022. Berkesempatan untuk melanjutkan pendidikan Diploma IV di Program Studi Teknik Multimedia dan Jaringan, Jurusan Teknik Informatika dan Komputer, Politeknik Negeri Jakarta.

© Hak Cipta milik Politeknik Negeri Jakarta

#### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## LAMPIRAN

### Lampiran 1. Datasheet ESP32-S3

[https://documentation.espressif.com/esp32-s3\\_datasheet\\_en.pdf](https://documentation.espressif.com/esp32-s3_datasheet_en.pdf)

### Lampiran 2. Datasheet ESP32-C3

[https://documentation.espressif.com/esp32-c3\\_datasheet\\_en.pdf](https://documentation.espressif.com/esp32-c3_datasheet_en.pdf)

### Lampiran 3. Panduan API BLE ESP32-C3

<https://docs.espressif.com/projects/esp-idf/en/stable/esp32c3/api-guides/ble/overview.html>

### Lampiran 4. Panduan API BLE ESP32-S3

<https://docs.espressif.com/projects/esp-idf/en/v5.5.2/esp32s3/api-guides/ble/overview.html>

### Lampiran 5. Source Code Sistem

<https://github.com/RafiRashya/SHM-OTA-NODE.git>

<https://github.com/RafiRashya/SHM-OTA-GATEWAY.git>

<https://github.com/RafiRashya/fota-server.git>

[https://github.com/RafiRashya/fota-web\\_admin.git](https://github.com/RafiRashya/fota-web_admin.git)

### Lampiran 6. Data Pengujian BLE Berkas Kecil

| TABEL: 10M_NLOS.csv |       |          |               |            | TABEL: 10M_NLOS.csv |       |          |          |            | TABEL: 15M_NLOS.csv |       |          |          |            | TABEL: 1M_NLOS.csv |       |          |          |            |
|---------------------|-------|----------|---------------|------------|---------------------|-------|----------|----------|------------|---------------------|-------|----------|----------|------------|--------------------|-------|----------|----------|------------|
| Status              | Waktu | Totality | ExpectedChunk | TotalRetry | Status              | Waktu | Totality | Expected | TotalRetry | Status              | Waktu | Totality | Expected | TotalRetry | Status             | Waktu | Totality | Expected | TotalRetry |
| SUCCESS             | 13.08 | 150000   | 300           | 753        | SUCCESS             | 17.09 | 150000   | 300      | 1084       | SUCCESS             | 12.13 | 150000   | 300      | 660        | SUCCESS            | 8.58  | 150000   | 300      | 303        |
| SUCCESS             | 10.82 | 150000   | 300           | 517        | SUCCESS             | 14.58 | 150000   | 300      | 926        | SUCCESS             | 14.42 | 150000   | 300      | 890        | SUCCESS            | 8.45  | 150000   | 300      | 306        |
| SUCCESS             | 11.79 | 150000   | 300           | 623        | SUCCESS             | 57.71 | 150000   | 300      | 5216       | SUCCESS             | 12.76 | 150000   | 300      | 720        | SUCCESS            | 8.57  | 150000   | 300      | 303        |
| SUCCESS             | 13.45 | 150000   | 300           | 791        | SUCCESS             | 51.72 | 150000   | 300      | 4619       | SUCCESS             | 16.66 | 150000   | 300      | 1073       | SUCCESS            | 8.60  | 150000   | 300      | 307        |
| SUCCESS             | 13.39 | 150000   | 300           | 786        | SUCCESS             | 53.88 | 150000   | 300      | 4835       | SUCCESS             | 14.72 | 150000   | 300      | 896        | SUCCESS            | 8.69  | 150000   | 300      | 304        |
| SUCCESS             | 12.88 | 150000   | 300           | 689        | SUCCESS             | 84.16 | 150000   | 300      | 7863       | SUCCESS             | 19.09 | 150000   | 300      | 1343       | SUCCESS            | 8.69  | 150000   | 300      | 317        |
| SUCCESS             | 13.26 | 150000   | 300           | 729        | SUCCESS             | 58.10 | 150000   | 300      | 5257       | SUCCESS             | 16.92 | 150000   | 300      | 1054       | SUCCESS            | 8.57  | 150000   | 300      | 303        |
| SUCCESS             | 10.49 | 150000   | 300           | 489        | SUCCESS             | 75.61 | 150000   | 300      | 7007       | SUCCESS             | 20.00 | 150000   | 300      | 980        | SUCCESS            | 8.62  | 150000   | 300      | 301        |
| SUCCESS             | 13.27 | 150000   | 300           | 746        | FAILED              | 38.30 | 65000    | 130      | 3680       | SUCCESS             | 18.85 | 150000   | 300      | 1225       | SUCCESS            | 8.58  | 150000   | 300      | 304        |
| SUCCESS             | 12.19 | 150000   | 300           | 661        | SUCCESS             | 24.14 | 150000   | 300      | 1860       | SUCCESS             | 17.53 | 150000   | 300      | 1181       | SUCCESS            | 8.57  | 150000   | 300      | 301        |
| TABEL: 1M_NLOS.csv  |       |          |               |            | TABEL: 1M_NLOS.csv  |       |          |          |            | TABEL: 1M_NLOS.csv  |       |          |          |            | TABEL: 1M_NLOS.csv |       |          |          |            |
| Status              | Waktu | Totality | ExpectedChunk | TotalRetry | Status              | Waktu | Totality | Expected | TotalRetry | Status              | Waktu | Totality | Expected | TotalRetry | Status             | Waktu | Totality | Expected | TotalRetry |
| SUCCESS             | 9.43  | 150000   | 300           | 390        | SUCCESS             | 9.75  | 150000   | 300      | 321        | SUCCESS             | 16.30 | 150000   | 300      | 1256       |                    |       |          |          |            |
| SUCCESS             | 9.30  | 150000   | 300           | 376        | SUCCESS             | 9.18  | 150000   | 300      | 365        | SUCCESS             | 17.73 | 150000   | 300      | 1220       |                    |       |          |          |            |
| SUCCESS             | 9.17  | 150000   | 300           | 363        | SUCCESS             | 9.24  | 150000   | 300      | 364        | SUCCESS             | 17.27 | 150000   | 300      | 1171       |                    |       |          |          |            |
| SUCCESS             | 9.52  | 150000   | 300           | 395        | SUCCESS             | 9.45  | 150000   | 300      | 392        | SUCCESS             | 17.08 | 150000   | 300      | 1156       |                    |       |          |          |            |
| SUCCESS             | 9.50  | 150000   | 300           | 398        | SUCCESS             | 9.02  | 150000   | 300      | 349        | SUCCESS             | 18.58 | 150000   | 300      | 1306       |                    |       |          |          |            |
| SUCCESS             | 8.73  | 150000   | 300           | 321        | SUCCESS             | 10.09 | 150000   | 300      | 442        | SUCCESS             | 15.96 | 150000   | 300      | 1043       |                    |       |          |          |            |
| SUCCESS             | 9.33  | 150000   | 300           | 371        | SUCCESS             | 9.17  | 150000   | 300      | 363        | SUCCESS             | 18.42 | 150000   | 300      | 1288       |                    |       |          |          |            |
| SUCCESS             | 9.28  | 150000   | 300           | 376        | SUCCESS             | 9.83  | 150000   | 300      | 425        | SUCCESS             | 16.44 | 150000   | 300      | 1091       |                    |       |          |          |            |
| SUCCESS             | 10.00 | 150000   | 300           | 445        | SUCCESS             | 11.72 | 150000   | 300      | 614        | SUCCESS             | 15.13 | 150000   | 300      | 896        |                    |       |          |          |            |
| SUCCESS             | 8.91  | 150000   | 300           | 323        | SUCCESS             | 11.72 | 150000   | 300      | 526        | SUCCESS             | 18.70 | 150000   | 300      | 1301       |                    |       |          |          |            |



## © Hak Cipta milik Politeknik Negeri Jakarta

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

## Lampiran 7. Data Pengujian BLE Berkas Sedang

| TABEL: 1SM_IOS.csv |        |           |               |            | TABEL: 1SM_IOS.csv |        |           |               |            | TABEL: 1SM_IOS.csv |        |        |          |            | TABEL: 1SM_IOS.csv |       |       |          |            |
|--------------------|--------|-----------|---------------|------------|--------------------|--------|-----------|---------------|------------|--------------------|--------|--------|----------|------------|--------------------|-------|-------|----------|------------|
| Status             | Waktu  | TotalByte | ExpectedChunk | TotalRetry | Status             | Waktu  | TotalByte | ExpectedChunk | TotalRetry | Status             | Waktu  | Total  | Expected | TotalRetry | Status             | Waktu | Total | Expected | TotalRetry |
| SUCCESS            | 58.97  | 58536     | 1171          | 2726       | SUCCESS            | 159.02 | 58536     | 1171          | 1208       | SUCCESS            | 161.22 | 58536  | 1171     | 12880      | SUCCESS            | 44.16 | 58536 | 1171     | 1236       |
| SUCCESS            | 82.51  | 58536     | 1171          | 5087       | SUCCESS            | 172.31 | 58536     | 1171          | 14012      | FAILED             | 188.77 | 321000 | 642      | 17123      | SUCCESS            | 43.72 | 58536 | 1171     | 1215       |
| SUCCESS            | 79.85  | 58536     | 1171          | 4799       | SUCCESS            | 183.28 | 58536     | 1171          | 13128      | SUCCESS            | 168.16 | 315000 | 1030     | 13980      | SUCCESS            | 44.44 | 58536 | 1171     | 1215       |
| SUCCESS            | 101.52 | 58536     | 1171          | 6929       | SUCCESS            | 83.16  | 58536     | 1171          | 5024       | SUCCESS            | 128.73 | 58536  | 1171     | 9658       | SUCCESS            | 45.35 | 58536 | 1171     | 1305       |
| SUCCESS            | 108.15 | 58536     | 1171          | 7645       | SUCCESS            | 117.57 | 58536     | 1171          | 8453       | SUCCESS            | 91.70  | 58536  | 1171     | 5934       | SUCCESS            | 43.81 | 58536 | 1171     | 1136       |
| SUCCESS            | 100.53 | 58536     | 1171          | 6869       | FAILED             | 70.13  | 185000    | 370           | 5520       | SUCCESS            | 60.80  | 58536  | 1171     | 2832       | SUCCESS            | 47.50 | 58536 | 1171     | 1530       |
| SUCCESS            | 88.40  | 58536     | 1171          | 5674       | FAILED             | 123.53 | 330000    | 660           | 10551      | SUCCESS            | 70.89  | 58536  | 1171     | 3846       | SUCCESS            | 43.44 | 58536 | 1171     | 1130       |
| SUCCESS            | 79.50  | 58536     | 1171          | 4743       | SUCCESS            | 141.17 | 58536     | 1171          | 10950      | SUCCESS            | 91.76  | 58536  | 1171     | 5980       | SUCCESS            | 44.17 | 58536 | 1171     | 1230       |
| SUCCESS            | 79.49  | 58536     | 1171          | 4429       | SUCCESS            | 118.02 | 58536     | 1171          | 8633       | SUCCESS            | 87.77  | 58536  | 1171     | 5555       | SUCCESS            | 43.03 | 58536 | 1171     | 1128       |
| SUCCESS            | 104.81 | 58536     | 1171          | 7264       | SUCCESS            | 98.22  | 58536     | 1171          | 6659       | SUCCESS            | 66.40  | 58536  | 1171     | 3394       | SUCCESS            | 43.29 | 58536 | 1171     | 1069       |

## Lampiran 8. Data Pengujian BLE Berkas Besar

| TABEL: 1SM_IOS.csv |        |           |               |            | TABEL: 1SM_IOS.csv |        |           |               |            | TABEL: 1SM_IOS.csv |        |         |          |            | TABEL: 1SM_IOS.csv |       |         |          |            |
|--------------------|--------|-----------|---------------|------------|--------------------|--------|-----------|---------------|------------|--------------------|--------|---------|----------|------------|--------------------|-------|---------|----------|------------|
| Status             | Waktu  | TotalByte | ExpectedChunk | TotalRetry | Status             | Waktu  | TotalByte | ExpectedChunk | TotalRetry | Status             | Waktu  | Total   | Expected | TotalRetry | Status             | Waktu | Total   | Expected | TotalRetry |
| SUCCESS            | 132.76 | 1200000   | 2400          | 10206      | SUCCESS            | 277.91 | 1200000   | 2400          | 23654      | SUCCESS            | 113.13 | 1200000 | 2400     | 6902       | SUCCESS            | 67.83 | 1200000 | 2400     | 2582       |
| SUCCESS            | 115.46 | 1200000   | 2400          | 7380       | SUCCESS            | 268.22 | 1200000   | 2400          | 22685      | SUCCESS            | 92.85  | 1200000 | 2400     | 5050       | SUCCESS            | 73.95 | 1200000 | 2400     | 2708       |
| SUCCESS            | 139.04 | 1200000   | 2400          | 9283       | SUCCESS            | 254.17 | 1200000   | 2400          | 22280      | SUCCESS            | 82.5   | 1200000 | 2400     | 3976       | SUCCESS            | 68.42 | 1200000 | 2400     | 2577       |
| SUCCESS            | 96.51  | 1200000   | 2400          | 4881       | SUCCESS            | 234.85 | 1200000   | 2400          | 19346      | SUCCESS            | 91.15  | 1200000 | 2400     | 4851       | SUCCESS            | 66.23 | 1200000 | 2400     | 2472       |
| SUCCESS            | 81.14  | 1200000   | 2400          | 3958       | SUCCESS            | 255.13 | 1200000   | 2400          | 21378      | SUCCESS            | 85.55  | 1200000 | 2400     | 4582       | SUCCESS            | 66.8  | 1200000 | 2400     | 2482       |
| SUCCESS            | 86.8   | 1200000   | 2400          | 4500       | SUCCESS            | 272.42 | 1200000   | 2400          | 23105      | SUCCESS            | 90.14  | 1200000 | 2400     | 4767       | SUCCESS            | 68.98 | 1200000 | 2400     | 2578       |
| SUCCESS            | 85.8   | 1200000   | 2400          | 4357       | SUCCESS            | 275.69 | 1200000   | 2400          | 23426      | SUCCESS            | 78.59  | 1200000 | 2400     | 3644       | SUCCESS            | 74.4  | 1200000 | 2400     | 2702       |
| SUCCESS            | 96.05  | 1200000   | 2400          | 4646       | SUCCESS            | 192.89 | 1200000   | 2400          | 15149      | SUCCESS            | 81.48  | 1200000 | 2400     | 3881       | SUCCESS            | 71.13 | 1200000 | 2400     | 2752       |
| SUCCESS            | 76.03  | 1200000   | 2400          | 3368       | SUCCESS            | 166.16 | 1200000   | 2400          | 13377      | SUCCESS            | 87.31  | 1200000 | 2400     | 4526       | SUCCESS            | 71.18 | 1200000 | 2400     | 2982       |
| SUCCESS            | 78.75  | 1200000   | 2400          | 3673       | SUCCESS            | 155.91 | 1200000   | 2400          | 11453      | SUCCESS            | 73.31  | 1200000 | 2400     | 4526       | SUCCESS            | 68.52 | 1200000 | 2400     | 2483       |

## Lampiran 9. Data Pengujian ESP-NOW Berkas Kecil

| TABEL: 1SM_IOS.csv |       |           |               |            | TABEL: 1SM_IOS.csv |       |           |               |            | TABEL: 1SM_IOS.csv |       |        |          |            | TABEL: 1SM_IOS.csv |       |        |          |            |
|--------------------|-------|-----------|---------------|------------|--------------------|-------|-----------|---------------|------------|--------------------|-------|--------|----------|------------|--------------------|-------|--------|----------|------------|
| Status             | Waktu | TotalByte | ExpectedChunk | TotalRetry | Status             | Waktu | TotalByte | ExpectedChunk | TotalRetry | Status             | Waktu | Total  | Expected | TotalRetry | Status             | Waktu | Total  | Expected | TotalRetry |
| SUCCESS            | 15.33 | 150000    | 682           | 690        | SUCCESS            | 24.57 | 150000    | 682           | 707        | SUCCESS            | 11.75 | 150000 | 682      | 687        | SUCCESS            | 7.16  | 150000 | 682      | 683        |
| SUCCESS            | 11.88 | 150000    | 682           | 688        | SUCCESS            | 21.67 | 150000    | 682           | 704        | SUCCESS            | 15.15 | 150000 | 682      | 729        | SUCCESS            | 7.16  | 150000 | 682      | 683        |
| SUCCESS            | 12.00 | 150000    | 682           | 686        | SUCCESS            | 25.07 | 150000    | 682           | 709        | SUCCESS            | 26.49 | 150000 | 682      | 705        | SUCCESS            | 7.10  | 150000 | 682      | 683        |
| SUCCESS            | 68.49 | 150000    | 682           | 729        | SUCCESS            | 11.55 | 150000    | 682           | 689        | SUCCESS            | 11.55 | 150000 | 682      | 687        | SUCCESS            | 7.12  | 150000 | 682      | 683        |
| SUCCESS            | 17.71 | 150000    | 682           | 690        | SUCCESS            | 8.50  | 150000    | 682           | 685        | SUCCESS            | 15.17 | 150000 | 682      | 688        | SUCCESS            | 7.65  | 150000 | 682      | 683        |
| SUCCESS            | 18.64 | 150000    | 682           | 712        | SUCCESS            | 16.54 | 150000    | 682           | 694        | SUCCESS            | 10.02 | 150000 | 682      | 684        | SUCCESS            | 7.55  | 150000 | 682      | 683        |
| SUCCESS            | 17.11 | 150000    | 682           | 690        | FAILED             | 8.02  | 1760      | 8             | 20         | SUCCESS            | 8.74  | 150000 | 682      | 683        | SUCCESS            | 7.13  | 150000 | 682      | 683        |
| SUCCESS            | 10.47 | 150000    | 682           | 685        | SUCCESS            | 43.53 | 150000    | 682           | 732        | FAILED             | 23.08 | 150000 | 682      | 691        | SUCCESS            | 7.28  | 150000 | 682      | 683        |
| SUCCESS            | 8.72  | 150000    | 682           | 684        | SUCCESS            | 8.48  | 150000    | 682           | 685        | SUCCESS            | 11.68 | 150000 | 682      | 687        | SUCCESS            | 7.05  | 150000 | 682      | 683        |
| SUCCESS            | 12.76 | 150000    | 682           | 687        | SUCCESS            | 11.14 | 150000    | 682           | 688        | SUCCESS            | 10.88 | 150000 | 682      | 686        | SUCCESS            | 7.50  | 150000 | 682      | 683        |

## Lampiran 10. Data Pengujian ESP-NOW Berkas Sedang

| TABEL: 1SM_IOS.csv |       |           |               |            | TABEL: 1SM_IOS.csv |        |           |               |            | TABEL: 1SM_IOS.csv |       |       |          |            | TABEL: 1SM_IOS.csv |       |       |          |            |
|--------------------|-------|-----------|---------------|------------|--------------------|--------|-----------|---------------|------------|--------------------|-------|-------|----------|------------|--------------------|-------|-------|----------|------------|
| Status             | Waktu | TotalByte | ExpectedChunk | TotalRetry | Status             | Waktu  | TotalByte | ExpectedChunk | TotalRetry | Status             | Waktu | Total | Expected | TotalRetry | Status             | Waktu | Total | Expected | TotalRetry |
| SUCCESS            | 32.06 | 58536     | 2662          | 2669       | SUCCESS            | 48.89  | 538608    | 2449          | 2483       | SUCCESS            | 29.68 | 58536 | 2662     | 2667       | SUCCESS            | 23.70 | 58536 | 2662     | 2663       |
| SUCCESS            | 25.25 | 58536     | 2662          | 2663       | SUCCESS            | 32.80  | 538608    | 2449          | 2460       | SUCCESS            | 30.96 | 58536 | 2662     | 2665       | SUCCESS            | 24.79 | 58536 | 2662     | 2663       |
| SUCCESS            | 27.19 | 58536     | 2662          | 2664       | SUCCESS            | 36.04  | 538608    | 2449          | 2459       | SUCCESS            | 30.56 | 58536 | 2662     | 2665       | SUCCESS            | 25.06 | 58536 | 2662     | 2663       |
| SUCCESS            | 27.14 | 58536     | 2662          | 2663       | SUCCESS            | 37.84  | 538608    | 2449          | 2465       | SUCCESS            | 28.40 | 58536 | 2662     | 2663       | SUCCESS            | 24.71 | 58536 | 2662     | 2663       |
| SUCCESS            | 27.45 | 58536     | 2662          | 2665       | SUCCESS            | 45.59  | 538608    | 2449          | 2474       | SUCCESS            | 28.32 | 58536 | 2662     | 2665       | SUCCESS            | 25.10 | 58536 | 2662     | 2663       |
| SUCCESS            | 28.93 | 58536     | 2662          | 2664       | FAILED             | 15.46  | 28380     | 129           | 149        | SUCCESS            | 37.33 | 58536 | 2662     | 2678       | SUCCESS            | 27.56 | 58536 | 2662     | 2663       |
| SUCCESS            | 29.10 | 58536     | 2662          | 2663       | FAILED             | 51.59  | 807120    | 1396          | 1445       | SUCCESS            | 28.58 | 58536 | 2662     | 2667       | SUCCESS            | 28.73 | 58536 | 2662     | 2663       |
| SUCCESS            | 35.74 | 58536     | 2662          | 2676       | SUCCESS            | 97.75  | 538608    | 2449          | 2529       | SUCCESS            | 25.56 | 58536 | 2662     | 2663       | SUCCESS            | 24.59 | 58536 | 2662     | 2663       |
| SUCCESS            | 36.54 | 58536     | 2662          | 2678       | FAILED             | 50.99  | 237600    | 1080          | 1127       | SUCCESS            | 27.60 | 58536 | 2662     | 2664       | SUCCESS            | 23.84 | 58536 | 2662     | 2663       |
| SUCCESS            | 33.87 | 58536     | 2662          | 2669       | SUCCESS            | 109.33 | 538608    | 2449          | 2541       | SUCCESS            | 26.14 | 58536 | 2662     | 2663       | SUCCESS            | 24.71 | 58536 | 2662     | 2663       |



© Hak Cipta milik Politeknik Negeri Jakarta

## Lampiran 11. Data Pengujian ESP-NOW Berkas Besar

| TABEL: 10M_LOS.csv |        |           |               |            | TABEL: 10M_NLOS.csv |        |           |          |            | TABEL: 15M_LOS.csv |        |         |          |            | TABEL: 15M_NLOS.csv |       |         |          |            |
|--------------------|--------|-----------|---------------|------------|---------------------|--------|-----------|----------|------------|--------------------|--------|---------|----------|------------|---------------------|-------|---------|----------|------------|
| Status             | Waktu  | TotalByte | ExpectedChunk | TotalRetri | Status              | Waktu  | TotalByte | Expected | TotalRetri | Status             | Waktu  | Total   | Expected | TotalRetri | Status              | Waktu | Total   | Expected | TotalRetri |
| SUCCESS            | 72.93  | 1200000   | 5455          | 5473       | SUCCESS             | 135.81 | 1200000   | 5455     | 5550       | SUCCESS            | 96.85  | 1200000 | 5455     | 5493       | SUCCESS             | 64.53 | 1200000 | 5455     | 5456       |
| SUCCESS            | 82.98  | 1200000   | 5455          | 5483       | SUCCESS             | 114.62 | 1200000   | 5455     | 5520       | SUCCESS            | 90.94  | 1200000 | 5455     | 5477       | SUCCESS             | 54.38 | 1200000 | 5455     | 5456       |
| SUCCESS            | 250.64 | 1200000   | 5455          | 5726       | SUCCESS             | 93.57  | 1200000   | 5455     | 5503       | SUCCESS            | 92.14  | 1200000 | 5455     | 5488       | SUCCESS             | 60.26 | 1200000 | 5455     | 5456       |
| SUCCESS            | 83.30  | 1200000   | 5455          | 5483       | SUCCESS             | 103.23 | 1200000   | 5455     | 5516       | SUCCESS            | 132.51 | 1200000 | 5455     | 5531       | SUCCESS             | 80.45 | 1200000 | 5455     | 5457       |
| SUCCESS            | 78.71  | 1200000   | 5455          | 5469       | SUCCESS             | 92.34  | 1200000   | 5455     | 5504       | SUCCESS            | 83.29  | 1200000 | 5455     | 5475       | SUCCESS             | 78.00 | 1200000 | 5455     | 5456       |
| SUCCESS            | 70.61  | 1200000   | 5455          | 5469       | SUCCESS             | 99.91  | 1200000   | 5455     | 5517       | SUCCESS            | 197.03 | 1200000 | 5455     | 5638       | SUCCESS             | 65.30 | 1200000 | 5455     | 5456       |
| SUCCESS            | 69.33  | 1200000   | 5455          | 5467       | SUCCESS             | 107.53 | 1200000   | 5455     | 5528       | SUCCESS            | 71.86  | 1200000 | 5455     | 5463       | SUCCESS             | 77.79 | 1200000 | 5455     | 5456       |
| SUCCESS            | 73.91  | 1200000   | 5455          | 5471       | SUCCESS             | 84.35  | 1200000   | 5455     | 5495       | SUCCESS            | 101.40 | 1200000 | 5455     | 5505       | SUCCESS             | 83.07 | 1200000 | 5455     | 5456       |
| SUCCESS            | 82.67  | 1200000   | 5455          | 5488       | SUCCESS             | 66.44  | 1200000   | 5455     | 5473       | SUCCESS            | 70.40  | 1200000 | 5455     | 5463       | SUCCESS             | 69.93 | 1200000 | 5455     | 5456       |
| SUCCESS            | 77.56  | 1200000   | 5455          | 5470       | SUCCESS             | 84.71  | 1200000   | 5455     | 5496       | SUCCESS            | 68.72  | 1200000 | 5455     | 5463       | SUCCESS             | 69.93 | 1200000 | 5455     | 5456       |
| TABEL: 1M_NLOS.csv |        |           |               |            | TABEL: 5M_LOS.csv   |        |           |          |            | TABEL: 5M_NLOS.csv |        |         |          |            |                     |       |         |          |            |
| Status             | Waktu  | TotalByte | ExpectedChunk | TotalRetri | Status              | Waktu  | TotalByte | Expected | TotalRetri | Status             | Waktu  | Total   | Expected | TotalRetri |                     |       |         |          |            |
| SUCCESS            | 76.85  | 1200000   | 5455          | 5485       | SUCCESS             | 56.78  | 1200000   | 5455     | 5456       | SUCCESS            | 111.00 | 1200000 | 5455     | 5538       |                     |       |         |          |            |
| SUCCESS            | 65.36  | 1200000   | 5455          | 5468       | SUCCESS             | 58.52  | 1200000   | 5455     | 5456       | FAILED             | 102.98 | 386760  | 1758     | 1880       |                     |       |         |          |            |
| SUCCESS            | 73.00  | 1200000   | 5455          | 5471       | SUCCESS             | 76.40  | 1200000   | 5455     | 5468       | SUCCESS            | 99.04  | 1200000 | 5455     | 5502       |                     |       |         |          |            |
| SUCCESS            | 192.45 | 1200000   | 5455          | 5690       | SUCCESS             | 101.12 | 1200000   | 5455     | 5470       | SUCCESS            | 127.90 | 1200000 | 5455     | 5550       |                     |       |         |          |            |
| SUCCESS            | 64.17  | 1200000   | 5455          | 5457       | SUCCESS             | 60.01  | 1200000   | 5455     | 5457       | SUCCESS            | 162.61 | 1200000 | 5455     | 5589       |                     |       |         |          |            |
| SUCCESS            | 64.99  | 1200000   | 5455          | 5458       | SUCCESS             | 55.95  | 1200000   | 5455     | 5457       | SUCCESS            | 132.94 | 1200000 | 5455     | 5556       |                     |       |         |          |            |
| SUCCESS            | 64.42  | 1200000   | 5455          | 5464       | SUCCESS             | 55.10  | 1200000   | 5455     | 5458       | SUCCESS            | 140.31 | 1200000 | 5455     | 5638       |                     |       |         |          |            |
| SUCCESS            | 62.81  | 1200000   | 5455          | 5457       | SUCCESS             | 54.18  | 1200000   | 5455     | 5456       | SUCCESS            | 134.65 | 1200000 | 5455     | 5554       |                     |       |         |          |            |
| SUCCESS            | 76.06  | 1200000   | 5455          | 5458       | SUCCESS             | 53.82  | 1200000   | 5455     | 5456       | SUCCESS            | 96.09  | 1200000 | 5455     | 5503       |                     |       |         |          |            |
| SUCCESS            | 71.74  | 1200000   | 5455          | 5457       | SUCCESS             | 54.20  | 1200000   | 5455     | 5456       | SUCCESS            | 80.75  | 1200000 | 5455     | 5487       |                     |       |         |          |            |

### Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
  - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
  - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta