Implementasi dan Analisis Performa Algoritma Enkripsi ChaCha20 Berbasis Protokol Komunikasi ESP-NOW Pada Wireless Sensor Network

Disusun oleh:

Naufal Farras Trikusuma

NIM: 215150301111047



**PROGRAM STUDI TEKNIK KOMPUTER**

**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS BRAWIJAYA**

**MALANG**

**2024**

DAFTAR ISI

[DAFTAR ISI ii](#_Toc181220062)

[DAFTAR TABEL iv](#_Toc181220063)

[DAFTAR GAMBAR v](#_Toc181220064)

[BAB 1 PENDAHULUAN 6](#_Toc181220065)

[1.1 Latar Belakang 6](#_Toc181220066)

[1.2 Rumusan Masalah 7](#_Toc181220067)

[1.3 Tujuan 7](#_Toc181220068)

[1.4 Manfaat 7](#_Toc181220069)

[1.5 Batasan Masalah 7](#_Toc181220070)

[1.6 Sistematika Pembahasan 8](#_Toc181220071)

[BAB 2 LANDASAN KEPUSTAKAAN 9](#_Toc181220072)

[2.1 Tinjauan Kepustakaan 9](#_Toc181220073)

[2.1.1 Implementation and Performance Analysis of AES-128 CBC algorithm in WSNs 1](#_Toc181220074)

[2.1.2 Implementasi Algoritme Clefia 128-Bit untuk Pengamanan Modul Komunikasi Lora. 1](#_Toc181220075)

[2.1.3 Implementasi Algoritma Enkripsi Snow-V pada Wireless Sensor Network (WSN) 1](#_Toc181220076)

[2.2 Dasar Teori 2](#_Toc181220077)

[2.2.1 Wireless Sensor Network 2](#_Toc181220078)

[2.2.2 Kriptografi 2](#_Toc181220079)

[2.2.3 Algoritma ChaCha20 2](#_Toc181220080)

[2.2.4 Protokol ESP-NOW 3](#_Toc181220081)

[BAB 3 METODOLOGI 5](#_Toc181220082)

[3.1 Tipe Penelitian 5](#_Toc181220083)

[3.2 Metode Penelitian 5](#_Toc181220084)

[3.2.1 Studi Literatur 6](#_Toc181220085)

[3.2.2 Rekayasa Kebutuhan 7](#_Toc181220086)

[3.2.3 Perancangan dan Implementasi Sistem 7](#_Toc181220087)

[3.2.4 Pengujian 7](#_Toc181220088)

[3.2.5 Analisis Hasil Pengujian 8](#_Toc181220089)

[3.2.6 Kesimpulan dan Saran 8](#_Toc181220090)

[BAB 4 REKAYASA KEBUTUHAN 9](#_Toc181220091)

[4.1 Kajian Masalah 9](#_Toc181220092)

[4.2 Identifikasi Stakeholder 9](#_Toc181220093)

[4.3 Kebutuhan Fungsional 9](#_Toc181220094)

[4.4 Spesifikasi Sistem 10](#_Toc181220095)

[4.5 Analisis Kebutuhan Perangkat Keras dan Perangkat Lunak 10](#_Toc181220096)

[4.5.1 Kebutuhan Perangkat Keras 10](#_Toc181220097)

[BAB 5 PERANCANGAN DAN IMPLEMENTASI 12](#_Toc181220098)

[DAFTAR REFERENSI 13](#_Toc181220099)

DAFTAR TABEL

[Tabel 2.1 Daftar Tinjauan Pustaka 9](#_Toc175987526)

DAFTAR GAMBAR

[Gambar 2.1 Wireless Sensor Network 2](#_Toc175988313)

[Gambar 2.2 Cara Kerja ChaCha20 3](#_Toc175988314)

[Gambar 2.3 ESP-NOW Layer 4](#_Toc175988315)

[Gambar 3.1 Diagram Alir Penelitian 6](#_Toc175988316)

# PENDAHULUAN

Pada bab pendahuluan ini berisi penelitian yang berisikan uraian mengenai latar belakang penelitian, rumusan masalah yang diangkat penulis, tujuan dan manfaat penelitian, serta batasan masalah dari lingkup penelitian ini. Pada bab ini terdapat sistematika penulisan mengenai isi secara umum dari setiap bab yang terdapat dalam penelitian ini.

## Latar Belakang

Perkembangan teknologi pada bidang wireless sensor network dalam beberapa tahun terakhir memiliki fungsionalitas yang beragam untuk membantu proses automasi pekerjaan manusia seperti proses untuk pengumpulan data dari sensor, pengendalian sistem secara nirkabel, dan monitoring/pengawasan. Menurut Astuti dan Wibisono (2017), Jaringan Sensor Nirkabel (Wireless sensor network/WSN) merupakan kumpulan jaringan node sensor yang saling berkomunikasi untuk melakukan pemindaian dan pengiriman/penerimaan data secara nirkabel yang memiliki keterbatasan pada sumber daya dan kemampuan komunikasi.

Pada aplikasi WSN, keamanan data merupakan aspek penting yang perlu diperhatikan khususnya pada komunikasi yang dilakukan pada jaringan lokal ataupun internet. Data atau informasi yang ditransmisikan pada beberapa kasus merupakan data privat dan sensitif yang perlu diamankan karena dapat mengakibatkan data breach dan unauthorized access oleh orang yang tidak bertanggung jawab (Sarker et al, 2020). Oleh karena itu, diperlukan penggunaan metode kriptografi seperti menggunakan algoritma enkripsi pada data yang akan ditransmisikan untuk meningkatkan keamanan pada aspek konfidensialitas dan integritas.

Menurut Gunathilake et al (2019) perangkat pada sistem tertanam dan/atau WSN memiliki kapasitas komputasi dan sumber daya yang rendah seperti terbatasnya random access memory (RAM), penyimpanan internal, daya komputasi pada prosesor, dan energi/sumber daya seperti baterai. Perangkat seperti ini tidak dapat mengalokasikan secara besar penggunaan sumber daya komputasi hanya untuk aspek keamanan. Oleh karena itu, diperlukan lightweight cryptography (LWC) yang diharapkan dapat mengeksekusi algoritma kriptografi yang lebih ringan atau dengan penggunaan sumber daya komputasi yang lebih rendah dibandingkan teknik kriptografi konvensional dan masih dapat menyediakan fungsi keamanan yang kuat untuk menanggulangi *security attacks.*

Algoritma enkripsi ChaCha20 adalah stream cipher berkecepatan tinggi yang dirancang oleh D. J. Bernstein pada tahun 2008 sebagai penyempurnaan dari stream cipher Salsa20. ChaCha20 merupakan alternatif algoritma pada Transport Layer Security (TLS) protokol yang bertujuan untuk meningkatkan batas keamanan tanpa mengorbankan kinerja pada platform perangkat lunak namun menghasilkan performa high-throughput stream cipher. (Santis et al, 2017).

Penelitian ini mengangkat masalah keamanan data dan keterbatasan sumber daya pada perangkat dalam aplikasi Wireless Sensor Network (WSN). Solusi yang ditawarkan adalah menganalisis performa algoritma enkripsi ChaCha20 dengan algoritma lainnya pada protokol komunikasi ESP-NOW berbasis ESP8266. Pengujian meliputi kecepatan enkripsi, dekripsi, serta uji penetrasi seperti sniffing dan *known-plaintext attack* (KPA). Selain itu, penelitian ini juga menilai penggunaan sumber daya pada mikrokontroler, untuk menentukan apakah algoritma ChaCha20 efektif dalam memberikan keamanan pada WSN yang memiliki keterbatasan sumber daya.

## Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan diatas, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana performa algoritma ChaCha20 dalam hal waktu komputasi pada perangkat ESP8266?

2. Bagaimana pengaruh penggunaan algoritma enkripsi ChaCha20 terhadap penggunaan sumber daya komputasi pada ESP8266?

3. Bagaimana implementasi algoritma enkripsi ChaCha20 pada protokol komunikasi ESP-NOW dan hasil pengujian algoritma dalam aspek keamanan?

## Tujuan

Adapun beberapa tujuan yang ingin dicapai oleh penulis dari penelitian ini adalah sebagai berikut:

1. Menganalisis dan evaluasi performa algoritma ChaCha20 dalam hal waktu komputasi pada perangkat ESP8266.
2. Menganalisis pengaruh penggunaan algoritma enkripsi ChaCha20 terhadap penggunaan sumber daya komputasi pada ESP8266.
3. Mengetahui implementasi dan evaluasi aspek keamanan algoritma enkripsi ChaCha20 pada protokol komunikasi ESP-NOW.

## Manfaat

Penelitian ini diharapkan dapat bermanfaat bagi para peneliti atau masyarakat terkhusus dalam mengimplementasikan dan menggunakan algoritma keamanan pada sistem WSN. Sehingga pembaca dapat memilih lebih tepat algoritma enkripsi yang lebih hemat sumber daya, cepat, dan aman pada sistem atau penelitian selanjutnya. Selain itu, penulis berharap agar masyarakat yang membaca skripsi ini dapat meningkatkan pemahaman mengenai aspek keamanan yang efisien pada *wireless sensor network*.

## Batasan Masalah

Agar lingkup penelitian ini lebih spesifik dan keterbatasan pengujian maka perlu adanya beberapa batasan masalah, yaitu:

1. Implementasi sistem berupa prototipe menggunakan 3 perangkat ESP8266.
2. Subjek pengujian dalam perbandingan performa algoritma sebanyak 3.

## Sistematika Pembahasan

Pada sistematika pembahasan terdapat penjelasan umum tiap bagian bab yang terdapat dalam penelitian ini. Penelitian ini dibagi pada beberapa bagian bab yang sesuai dengan alur penelitian. Sistematika penulisan skripsi dibagi menjadi beberapa bab, yaitu:

**BAB I PENDAHULUAN**

Bab ini memiliki beberapa subbab berisikan latar belakang dari penelitian, rumusan masalah dari latar belakang penelitian, tujuan, manfaat, batasan masalah, dan sistematika pembahasan skripsi.

**BAB II LANDASAN KEPUSTAKAAN**

Bab ini memuat kajian atau tinjauan kepustakaan dan landasan teori dari penelitian sebelumnya. Tinjauan dan landasan teori yang digunakan akan memiliki keterkaitan dengan teori dan implementasi dari penelitian ini sebagai landasan metode dan referensi.

**BAB III METODOLOGI PENELITIAN**

Bab ini berisikan alur proses dan metode yang akan digunakan dalam penelitian ini. Pada bab III secara lebih spesifik akan dimuat mengenai metode yang akan digunakan oleh penulis seperti diagram alir dan penjelasan prosesnya.

**BAB IV REKAYASA KEBUTUHAN**

Bab ini memuat pengkajian masalah secara umum, kebutuhan fungsional yang dimiliki sistem meliputi perangkat lunak, perangkat keras, dan alat yang digunakan dalam pengujian.

**BAB V PERANCANGAN DAN IMPLEMENTASI**

Bab ini berisikan proses perancangan dan implementasi sistem, yaitu bagaimana merancang sistem dari perangkat lunak dan perangkat keras berdasarkan rekayasa kebutuhan seperti membuat blok diagram, skematik, konfigurasi sistem, dan diagram alir hingga proses implementasi sistem.

**BAB VI PENGUJIAN DAN ANALISIS**

Bab ini berisikan hasil pengujian untuk menganalisis dan mengevaluasi kinerja sistem secara keseluruhan. Pada bab ini juga akan terdapat jawaban daripada rumusan masalah yang ada.

**BAB VII PENUTUP**

Bab ini memuat kesimpulan dan saran yang didapat setelah penelitian dilakukan berdasarkan hasil pengujian dan analisis. Saran dan kesimpulan penelitian diharapkan dapat meningkatkan penelitian selanjutnya dari hasil penelitian penulis.

Performance Evaluation of IoT Encryption Algorithms: Memory, Timing, and Energy

# LANDASAN KEPUSTAKAAN

Landasan kepustakaan berisi uraian dan pembahasan tentang teori, konsep, model, metode, atau sistem dari pustaka ilmiah, yang berkaitan dengan tema, masalah, atau pertanyaan penelitian. Dalam landasan kepustakaan terdapat landasan teori dari berbagai sumber pustaka yang terkait dengan teori dan metode yang digunakan dalam penelitian. Jika dibutuhkan sesuai dengan karakteristik penelitiannya dan syarat kecukupan khusus keminatan tertentu, bisa juga terdapat kajian pustaka yang menjelaskan secara umum penelitian-penelitian terdahulu yang berhubungan dengan topik skripsi dan menunjukkan persamaan dan perbedaan skripsi tersebut terhadap penelitian terdahulu yang dituliskan.

## Tinjauan Kepustakaan

Pada saat dimulai penelitian, penulis meninjau beberapa literatur dan jurnal yang berhubungan/berkaitan dengan penelitian untuk mendapatkan informasi dan referensi dalam penelitian ini. Berikut adalah isi dari kajian pustaka yang berisikan penelitian-penelitian sebelumnya, dapat dilihat pada Tabel.2.1.

1 Tabel 2.1 Daftar Tinjauan Pustaka

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No. | Nama Penulis (Tahun),  Judul Penelitian | Persamaan Penelitian | Perbedaan | |
| Penelitian Sebelumnya | Penelitian Sekarang |
| 1 | Maitra et al (2019) “Performance Evaluation of IoT Encryption Algorithms: Memory, Timing, and Energy.” | Menggunakan algoritma enkripsi pada WSN dalam pengujian performa sebagai subjek penelitian. | Menggunakan PIC18F27K40 8-bit, pengujian performa algoritma AES pada kecepatan komputasi, penggunaan memori, dan energi. | Menggunakan  ESP8266, pengujian performa algoritma chacha20 pada penggunaan kecepatan dan sumber daya komputasi. |
| 2 | Iman et al (2022) “Implementasi Algoritme Clefia 128-Bit untuk Pengamanan Modul Komunikasi Lora. ” | Menggunakan algoritma enkripsi pada WSN dan pengujian keamanan algoritma sebagai subjek penelitian. | Menggunakan  arduino uno, modul LoRa, pengujian algoritma clefia 128-bit. | Menggunakan  ESP8266 dan protokol ESP-NOW, pengujian performa algoritma chacha20. |
| 3 | Pratama et al (2021),  “Implementasi Algoritma Enkripsi Snow-V pada Wireless Sensor Network (WSN)” | Menggunakan algoritma enkripsi pada WSN, pengujian performa dan keamanan algoritma sebagai subjek penelitian. | Menggunakan  arduino uno, modul nRF24L01, pengujian performa algoritma snow-v. | Menggunakan  ESP8266 dan protokol ESP-NOW, pengujian performa algoritma chacha20 |

### Implementation and Performance Analysis of AES-128 CBC algorithm in WSNs

Penelitian ini dilakukan oleh Maitra et al (2019,) yang berjudul ”Performance Evaluation of IoT Encryption Algorithms: Memory, Timing, and Energy” merupakan penelitian mengenai evaluasi performa algoritma enkripsi pada IoT khususnya pada aspek kecepatan, penggunaan memori, dan energi. Hasil pengujian pada riset ini menghasilkan kesimpulan bahwa penggunaan algoritma AES pada perangkat mikrokontroler yang memiliki keterbatasan sumber daya tidak *feasible* atau kurang cocok jika dibandingkan dengan algoritma yang lebih ringan seperti XTEA yang dari sisi pengujian yang lebih efisien pada aspek daya. Namun algoritma XTEA tidak dapat dibandingkan secara level keamanan dengan AES yang merupakan algoritma modern.

### Implementasi Algoritme Clefia 128-Bit untuk Pengamanan Modul Komunikasi Lora.

Penelitian yang dilakukan Muhammad Fadhil Iman et al (2022), dengan judul ”Implementasi Algoritme Clefia 128-Bit Untuk Pengamanan Modul Komunikasi LoRa” adalah implementasi algoritma enkripsi clefia 128-bit pada modul LoRa yang tidak memiliki sistem keamanan untuk melindungi aspek konfidesialitas pada suatu data yang di transmisikan. Pada implementasi ini sistem diuji dengan berbagai pengujian seperti tes vektor, uji serangan aktif seperti *known-plaintext-attack (*KPA), dan uji serangan pasif seperti *sniffing.* Secara umum implementasi sistem dari algoritma ini mampu melewati pengujian yang dilakukan dengan baik. Pengujian pada riset ini hanya terkhusus pada aspek keamanan data, aspek lain seperti kecepatan komputasi dan penggunaan sumber daya tidak di uji. Oleh karena itu, diperlukan pengujian lebih lanjut mengenai performa algoritma ini karena salah satu tantangan utama dari sistem WSN adalah terbatasnya sumber daya yang memengaruhi *lifetime* dari sistem.

### Implementasi Algoritma Enkripsi Snow-V pada Wireless Sensor Network (WSN)

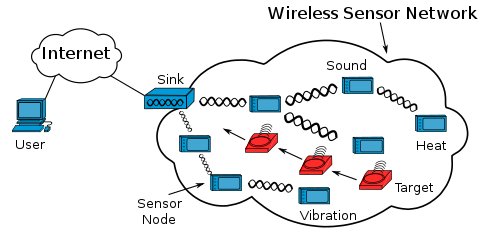
Penelitian selanjutnya yang dilakukan Yulius Adi Pratama et al (2021), dengan judul ”Implementasi Algoritma Enkripsi Snow-V pada Wireless Sensor Network (WSN)”. Penelitian ini dilakukan untuk mengimplementasikan salah satu algoritma enkripsi snow-v yang berbasis Arduino Uno dan modul komunikasi nRF24L01. Pada penelitian disebutkan bahwa selain implementasi dilakukan juga proses pengujian seperti tes vektor untuk memastikan aspek *availability* data, uji *sniffing* saat data berjalan di jalur komunikasi, dan menghitung waktu komputasi enkripsi dan dekripsi yang berturut-turut rata-rata waktunya 241ms dan 185ms. Dalam konteks kecepatan komputasi, waktu yang didapatkan algoritma ini dalam keseluruhan proses enkripsi dapat mencapai ±426ms yang dapat memberikan delay yang cukup signifikan yang dimana delay lain seperti proses komunikasi, proses akuisisi data sensor belum termasuk. Selanjutnya, pada penelitian ini peneliti belum menguji penggunaan sumber daya komputasi yang digunakan saat proses enkripsi, yang dimana aspek efisiensi daya menjadi hal utama dalam konsep wireless sensor network.

## Dasar Teori

Pada dasar teori akan dijelaskan teori-teori yang digunakan dalam mengimplementasikan penelitian ini. Dengan adanya pengertian, persamaan, dan cara kerja yang diharapkan dapat sebagai penunjang pemahaman dalam penelitian.

### Wireless Sensor Network

Jaringan Sensor Nirkabel(Wireless sensor network)  merupakan kumpulan jaringan *node* sensor yang saling berkomunikasi untuk melakukan pemindaian dan pengiriman/penerimaan data secara nirkabel yang memiliki keterbatasan pada sumber daya dan kemampuan komunikasi (Astuti dan Wibisono, 2017).



1 Gambar 2.1 Wireless Sensor Network

(Sumber: commons.wikimedia.org)

### Kriptografi

Kriptografi adalah konsep penyandian yang digunakan untuk menjaga kerahasiaan data, sehingga hanya pihak yang berwenang yang dapat mengetahui informasi dari data tersebut. Dalam kriptografi, data asli (plaintext) akan dienkripsi menjadi ciphertext dan kemudian didekripsi kembali menjadi plaintext ketika diterima oleh pihak yang berhak. Saat ini, algoritma modern banyak digunakan untuk mengamankan data, yang secara umum dibagi menjadi dua jenis, yaitu stream ciphers dan block ciphers. Stream ciphers bekerja dengan melakukan operasi XOR antara keystream dan plaintext untuk menghasilkan ciphertext. (Qadir dan Varol, 2019).

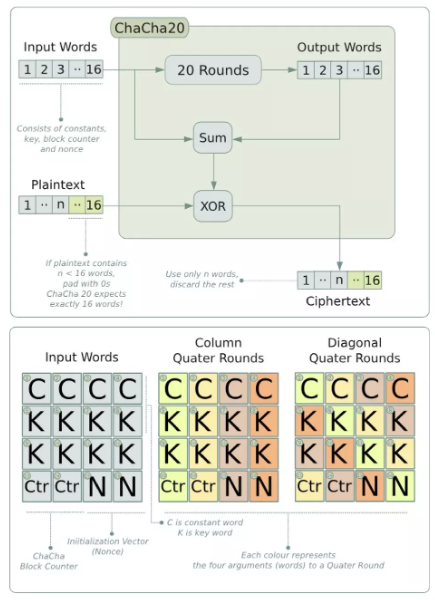
### Algoritma ChaCha20

ChaCha20 adalah stream cipher berkecepatan tinggi yang dirancang oleh D. J. Bernstein pada tahun 2008 sebagai penyempurnaan dari stream cipher Salsa20. ChaCha20 merupakan alternatif algoritma pada *Transport Layer Security* (TLS) protokol yang bertujuan untuk meningkatkan batas keamanan tanpa mengorbankan kinerja pada platform perangkat lunak namun menghasilkan performa *high-throughput stream cipher*. (Santis et al, 2017).

Menurut Procter (2014), ChaCha akan membuat *keystream* menggunakan blok fungsi ChaCha20 pada kunci, *nonce,* dan *counter block.* Persamaan blok fungsi ChaCha20 adalah berikut:

 (2.1)

Diatas adalah blok fungsi ChaCha20 yang memiliki input 32-byte kunci, 4-byte blok angka, 12-byte *nonce,* dan 64 *pseudo-random* bytes output.



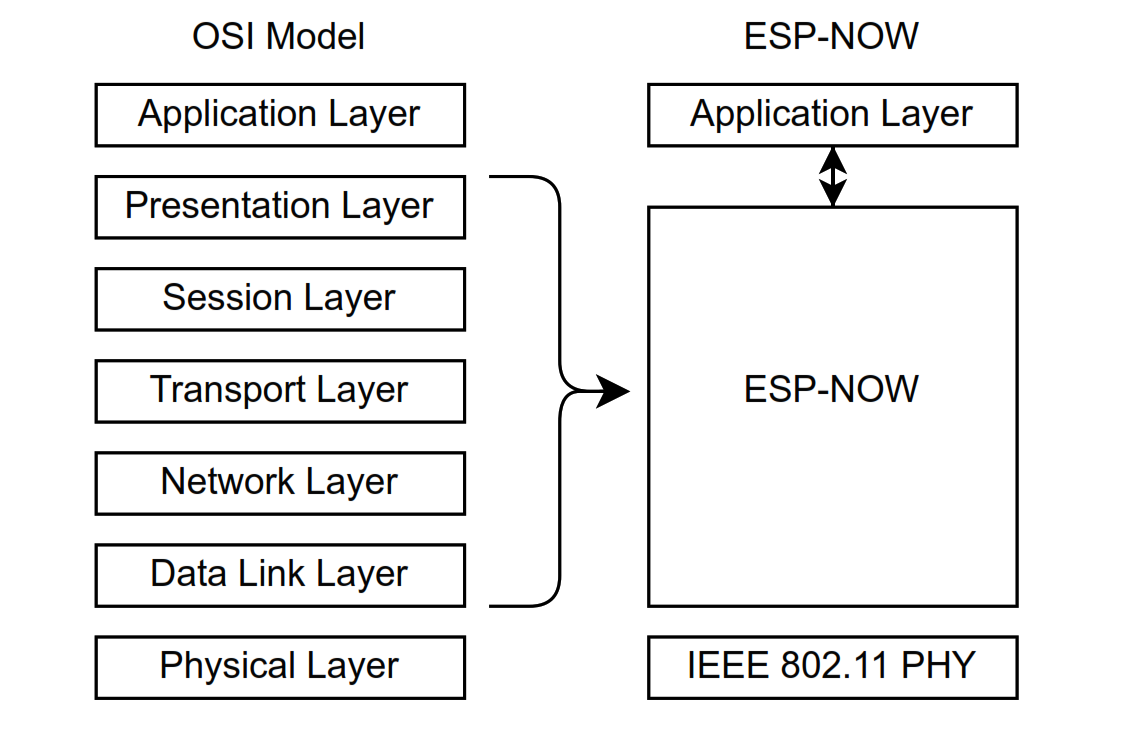
2 Gambar 2.2 Cara Kerja ChaCha20

(Sumber: datatracker.ietf.org)

### Protokol ESP-NOW

ESP-NOW merupakan protokol komunikasi nirkabel sebagai solusi untuk *low cost* dan *low power* protokol untuk perangkat IoT yang berjalan pada frekuensi 2.4 GHz *Industrial, Scientific, and Medical* (ISM) spektrum. ESP-NOW memungkinkan banyak perangkat berkomunikasi secara 2 arah, yang secara teoritis jangkauan sinyal dapat lebih baik hingga 15 kali daripada *Bluetooth Low Energy* (BLE).

Pada *physical layer* protokol ini berjalan secara *native* pada standar IEEE802.11 ESP-NOW memerlukan alamat *Medium Access Protocol* (MAC) dari tiap perangkat untuk melakukan komunikasi, pada MAC layer digunakan metode *Carrier Sense Multiple Access/Collision Avoidance* (CSMA/CA) untuk mencegah terjadinya tabrakan ataupun data yang tidak terkirim karena adanya komunikasi lain pada jalur atau satu waktu yang sama. (Urazayev et al, 2023).



3 Gambar 2.3 ESP-NOW Layer

(Sumber: (Urazayev et al, 2023))

# METODOLOGI

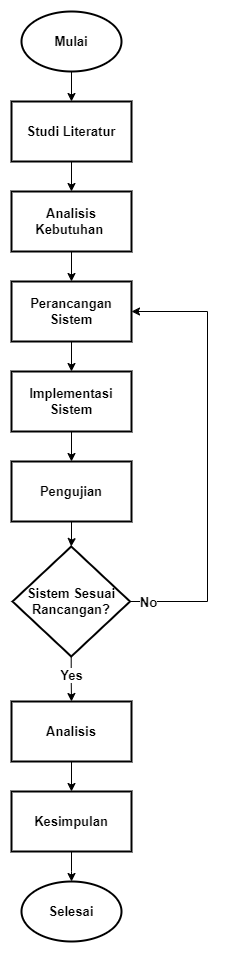
Pada bab ketiga atau bab metodologi dalam penelitian ini, penulis akan membahas beberapa sub-bab. Pertama, akan dijelaskan tentang tipe penelitian yang digunakan dalam penelitian ini. Selanjutnya, akan terdapat metode penelitian melalui sebuah diagram alir. Terakhir, penulis akan mencantumkan peralatan yang mendukung jalannya penelitian.

## Tipe Penelitian

Penelitian ini dilakukan dengan tipe penelitian implementatif - pengembangan pada bidang rekayasa sistem komputer (RSK). Tipe penelitian ini berfokus pada implementasi analisis performa algoritma enkripsi ChaCha20 pada WSN dengan pengembangan jenis pengujian dan perbandingan dengan algoritma lain yang menghasilkan prototipe dan hasil analisis sistem secara komperehensif.

## Metode Penelitian

Metode penelitian yang digunakan penulis akan dijelaskan dalam bentuk diagram alir yaitu pada gambar 3.1 dan penjelasan tiap langkahnya secara spesifik untuk mencapai tujuan penelitian.



4 Gambar 3.1 Diagram Alir Penelitian

### Studi Literatur

Tahap studi literatur oleh penulis dilakukan dengan *literature review* dan meninjau terhadap kepustakaan yang memiliki keterkaitan dengan topik untuk menyusun dasar teori dalam memenuhi kebutuhan penelitian, lalu untuk mengimplementasikan beberapa metode berdasarkan jurnal-jurnal yang dicatut. Terdapat beberapa literatur berupa jurnal, buku, ataupun sumber lainnya mengenai aspek-aspek seperti *wireless sensor network,* kriptografi, algoritma ChaCha20, dan protokol komunikasi ESP-NOW yang akan digunakan sebagai dasar dalam penelitian ini.

### Rekayasa Kebutuhan

Tahap selanjutnya adalah melakukan analisis terhadap kebutuhan sistem yang akan diimplementasikan agar penelitian ini dapat dilakukan. Penulis melakukan perencanaan kebutuhan perangkat keras dan perangkat lunak yang akan digunakan selama proses implementasi, pengujian, hingga analisis sistem. Berikut merupakan kebutuhan yang akan digunakan dalam penelitian.

Kebutuhan perangkat keras:

1. ESP8266

2. Sensor arus dan daya INA219

3. Sensor suhu dan kelembapan DHT22

4. Baterai 9V

5. Kabel USB dan Jumper

6. Power Supply DC

Kebutuhan perangkat lunak:

1. Arduino IDE

2. Library algoritma enkripsi ChaCha20

3. Library sensor INA219 dan DHT

### Perancangan dan Implementasi Sistem

Pada tahap ini dilakukan perancangan perangkat keras maupun lunak sistem berdasarkan rekayasa kebutuhan. Pada perancangan perangkat keras pembuatan diagram skematik yang tersusun dari ESP8266, sensor DHT22 dan komponen pengujian sumber daya komputasi seperti seperti sensor INA219 dan power supply dc. Selanjutnya, perancangan perangkat lunak adalah dengan menyusun kode program untuk menjalankan dan pengujian sistem. Setelah dilakukan perancangan, dilakukan implementasi dengan mengintegrasikan perangkat lunak dan perangkat keras yang telah disusun seperti pengunggahan kode program dan *wiring* pada komponen perangkat keras.

### Pengujian

Tahap pengujian akan implementasi sistem berfungsi untuk mengetahui fungsionalitas kesuluruhan sistem dan mengumpulkan data guna menyelesaikan rumusan masalah yang telah disusun. Pengujian yang akan dilakukan adalah uji tes vektor pada algoritma enkripsi untuk memvalidasi kerja algoritma, uji performa algoritma khususnya pada kecepatan komputasi dan penggunaan sumber daya komputasi seperti penggunaan memori, daya/energi yang digunakan saat melakukan proses enkripsi dan dekripsi. Lalu dilakukan pengujian pada aspek keamanan dengan melakukan uji serangan pasif (sniffing) dan uji serangan aktif (known-plaintext attack) untuk melihat kerentanan pada algoritma enkripsi. Pengujian akan dilakukan pada beberapa algoritma sekaligus sebagai komparasi, yaitu algoritma ChaCha20, Clefia, AES, dan Snow-V yang dimana semua menggunakan 256-bit key.

### Analisis Hasil Pengujian

Selanjutnya pada tahap analisis akan dilakukan pengolahan data hasil uji dan menampilkan hasil pengujian sistem yang telah dikerjakan sebelumnya dengan analisis dan perbandingan terhadap pengujian dengan algoritma lain untuk melihat keseluruhan performa pada masing-masing algoritma.

### Kesimpulan dan Saran

Pada akhir penelitian ini, penulis akan menyimpulkan temuan yang menjawab pertanyaan pada rumusan masalah. Kesimpulan ini didasarkan pada hasil pengujian yang telah diperoleh. Selain itu, penulis juga memberikan beberapa saran agar penelitian dengan topik serupa dapat dikembangkan lebih lanjut di masa depan.

# REKAYASA KEBUTUHAN

Rekayasa kebutuhan merupakan bab cepat dalam penelitian ini yang berisi uraian terkait meninjau masalah secara umum, mengidentifikasi stakeholder, mengembangkan kebutuhan fungsional dan spesifikasi yang harus dimiliki oleh sistem, juga menganalisis kebutuhan perangkat keras dan perangkat lunak yang menyusun sistem.

## Kajian Masalah

Penelitian ini berfokus untuk menganalisis dan melakukan evaluasi pada performa algoritma enkripsi pada WSN yang memiliki kerentanan pada hal keamanan saat transmisi data dan karakteristik *limited resources-*nya. Penelitian ini utamanya melakukan komparasi performa algoritma enkripsi ChaCha20 dengan beberapa algoritma seperti Clefia, AES, dan Snow-V dengan parameter pengujian tingkat keamanan algoritma, tingkat efisiensi penggunaan sumber daya, dan kecepatan komputasi, parameter tersebut menjadi pertimbangan utama dalam kehandalan dan keamanan sistem WSN.

Algoritma ChaCha20 dipilih penulis dikarenakan kecepatan nya dalam proses enkripsi yang sangat cepat untuk diharapkan dapat lebih efisien dibandingkan algoritma lainnya, namun perlu dilakukan pengujian secara komperehensif untuk menentukan algoritma yang paling cocok dengan karakteristik WSN.

## Identifikasi Stakeholder

Analisis dan evaluasi hasil pengujian algoritma enkripsi pada studi kasus WSN memiliki peran penting dalam proses perancangan dan implementasi sistem WSN kedepan. Hasil penelitian yang diperoleh dapat menjadi dasar pengambilan keputusan dalam menentukan algoritma enkripsi yang tepat untuk diterapkan pada sistem WSN. Beberapa stakeholders dari penelitian ini adalah Perekayasa dan Pengembang Sistem dalam perancangan sistemnya, pengguna sistem WSN dimana sistemnya dapat lebih aman dan efisien, lalu peneliti atau akademisi yang dapat melakukan studi lanjutan mengenai topik ini.

## 4.3 Kebutuhan Fungsional

Kebutuhan fungsional merinci fungsi yang harus disediakan agar sistem dapat beroperasi sesuai dengan tujuan yang telah ditetapkan. Kebutuhan fungsional yang harus dipenuhi dalam sistem ini adalah sebagai berikut:

* + 1. Sistem yaitu ESP8266 dapat melakukan enkripsi dan dekripsi dengan algoritma enkripsi ChaCha20, Clefia, AES, dan Snow-V.
    2. Sistem yaitu ESP8266 dapat mengirimkan dan/atau menerima data terenkripsi dengan protokol ESP-NOW.
    3. Sistem pengujian yaitu ESP32 dan sensor INA219 dapat melakukan pengukuran arus, tegangan, dan daya pada beban dengan *sampling* *rate* konstan2 ms (milisekon) untuk dapat melihat perubahan daya dengan akurat.
    4. Sistem pengujian dapat mengukur waktu komputasi setiap algoritma dengan tingkat presisi hingga mikrodetik, guna memastikan evaluasi kinerja yang akurat.
    5. Sistem pengujian dapat melakukan uji keamanan setiap algoritma dalam uji serangan aktif dan pasif untuk memastikan aspek *availability* dan *confidentiality*.

## 4.4 Spesifikasi Sistem

Sistem akan dibagi menjadi sistem yang diuji dan sistem penguji. Pada sistem yang diuji akan diimplementasikan Algoritma enkripsi di 2 mikrokontroler ESP8266 yang digunakan sebagai enkriptor dan dekriptor yang saling berkomunikasi dengan protokol ESP-NOW, dimana data uji yang digunakan berasal dari sensor suhu dan kelembapan DHT22. Sistem dirancang dengan baterai 5 Volt untuk berjalan. Lalu pada sistem penguji, setiap algoritma akan diuji kecepatan komputasinya dalam melakukan enkripsi dan dekripsi. Dalam pengujian penggunaan sumber daya atau energi yang digunakan tiap algoritma, digunakan ESP32 dan sensor INA219 yang dapat mengukur arus dan voltase pada beban yaitu ESP8266 yang akan menghasilkan daya dan total energi yang diperlukan setiap algoritma melakukan enkripsi.

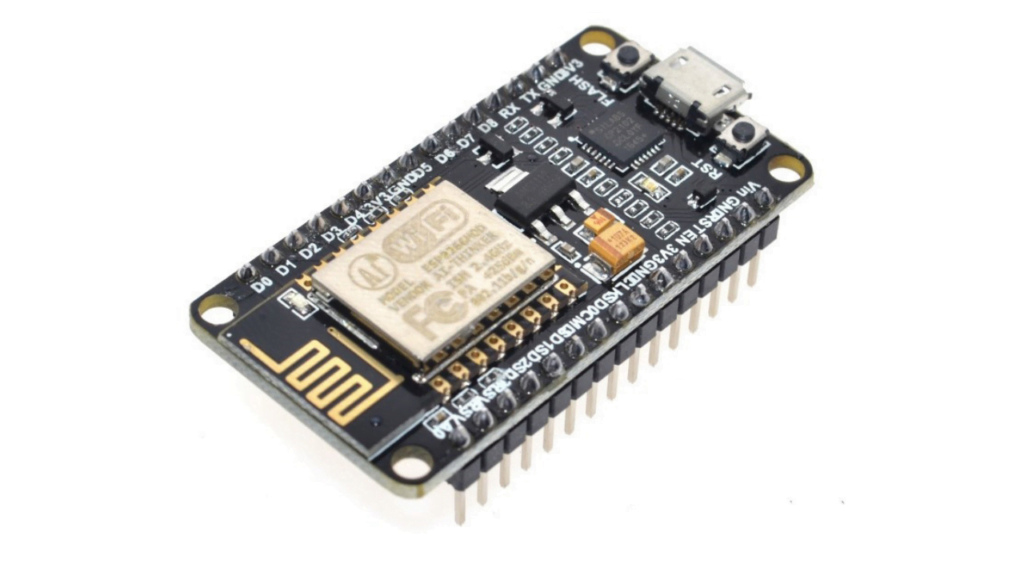
## Analisis Kebutuhan Perangkat Keras dan Perangkat Lunak

Kebutuhan non-fungsional menjelaskan persyaratan yang harus dipenuhi agar sistem dapat beroperasi dengan optimal. Kebutuhan ini mencakup dua aspek utama, yaitu kebutuhan perangkat keras dan kebutuhan perangkat lunak.

### Kebutuhan Perangkat Keras

Kebutuhan perangkat keras harus disesuaikan dengan setiap komponen yang digunakan dalam sistem agar kompatibel dengan spesifikasinya. Penyesuaian ini bertujuan untuk memastikan semua komponen dapat berfungsi secara optimal. Berikut adalah daftar kebutuhan perangkat keras yang akan digunakan oleh penulis:

#### 4.5.1.1 ESP 8266



Gambar 4.1 ESP8266

(Sumber: blog.indobot.co.id, 2022)

ESP8266 adalah mikrokontroler dengan modul Wi-Fi terintegrasi yang umum digunakan dalam proyek IoT (Internet of Things) ataupun WSN (Wireless Sensor Network). Mikrokontroler ini memiliki kapabilitas jaringan Wi-Fi dan menjalankan program secara mandiri tanpa memerlukan mikrokontroler eksternal. Dengan prosesor berbasis Tensilica L106 32-bit prosesor berkecepatan 80-160 MHz, ESP8266 mendukung protokol seperti TCP/IP dan ESP-NOW yang dapat digunakan untuk komunikasi nirkabel. Selain itu, modul ini memiliki GPIO (General Purpose Input Output) yang memungkinkan interaksi dengan berbagai sensor dan aktuator, membuatnya populer untuk implementasi seperti *smart home*, *monitoring*, dan wireless sensor network.

Tabel 2.1 Spesifikasi ESP8266

|  |  |
| --- | --- |
| Mikrokontroler | ESP8266 |
| Prosesor Utama | Tensilica L106 32-bit processor |
| Frekuensi | 80-160MHz |
| Wi-Fi | 802.11 b/g/n (HT20) (802.11n @ 2,4 GHz-2,5GHz hingga 72,2 Mbps) |
| SRAM | 50 KiB |
| Flash Memory | 4 MiB |
| Operating Voltage | 2.5 V ~ 3.6 V |

#### 4.5.1.2 ESP 32



Gambar 4.2 ESP32

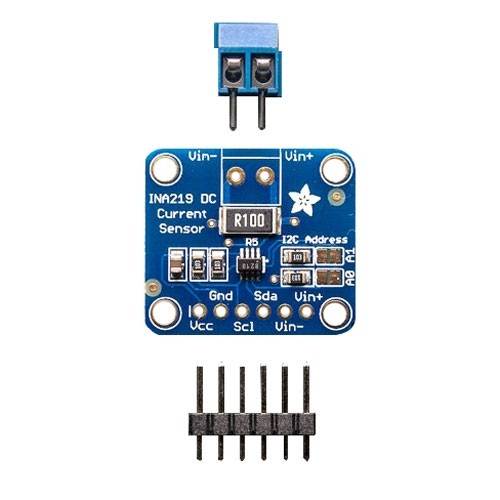
(Sumber: ecadio.com, 2021)

ESP32 adalah mikrokontroler dengan modul Wi-Fi dan Bluetooth terintegrasi Dengan prosesor dual-core Xtensa LX6 berkecepatan hingga 240 MHz,. Modul ini mendukung berbagai protokol seperti TCP/IP, BLE (Bluetooth Low Energy), dan Bluetooth, menjadikannya ideal untuk aplikasi IoT, smart home, dan wireless sensor network. ESP32 memiliki GPIO untuk berinteraksi dengan sensor/actuator, sensor sentuh kapasitif, modul enkripsi hardware, dan dukungan untuk komunikasi seperti SPI, I2C, serta UART.

Tabel 2.2 Spesifikasi ESP32

|  |  |
| --- | --- |
| Mikrokontroler | ESP32 |
| Prosesor Utama | Tensilica Xtensa 32-bit LX6 microprocessor |
| Inti | 2 atau 1 |
| Frekuensi | Hingga 240MHz |
| Wi-Fi | 802.11 b/g/n/e/i (802.11n @ 2,4 GHz hingga 150 Mbit/detik) |
| Bluetooth | v4.2 BR/EDR dan Bluetooth Low Energy (BLE) |
| FLASH | 4 MiB |
| SRAM | 520 KiB |
| Operating Voltage | 3.0 V ~ 3.6 V |

#### 4.5.1.3 Sensor INA219



Gambar 4.3 Sensor INA219

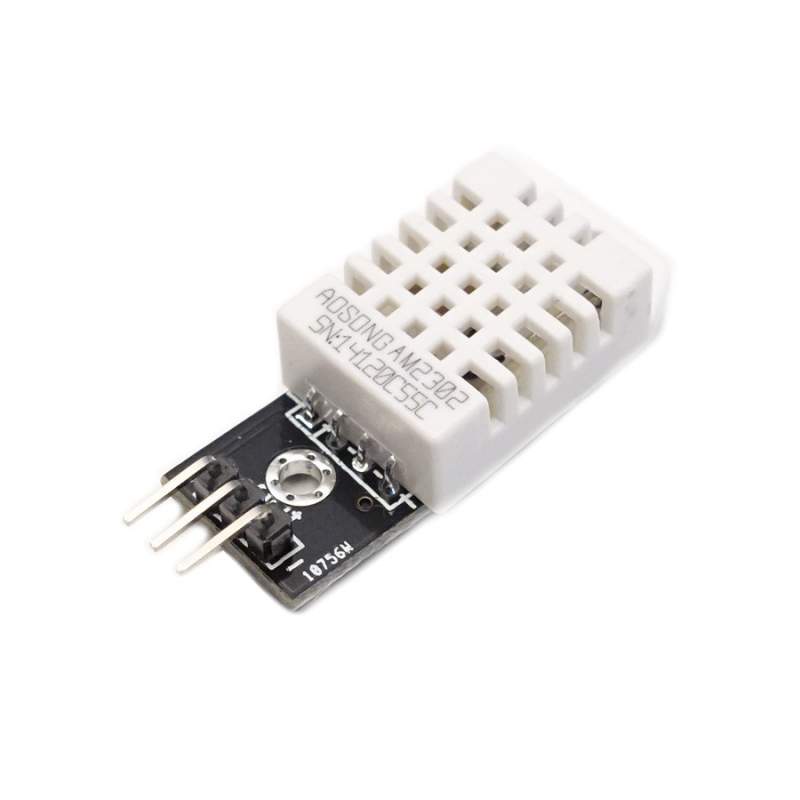
(Sumber: breakrow.com, 2020)

INA219 adalah sensor arus dan tegangan yang dapat digunakan untuk memantau konsumsi daya pada berbagai perangkat. Sensor ini bekerja dengan mengukur tegangan pada resistor shunt dan menghitung arus yang mengalir melalui beban. INA219 mendukung pengukuran arus hingga 3.2A dan tegangan hingga 26V, dengan resolusi tinggi.

Tabel 2.3 Spesifikasi Sensor INA219

|  |  |
| --- | --- |
| Sensor | INA219 |
| Tegangan Kerja | 3-5V |
| Tegangan Bus/Beban | 0-26V |
| Arus Maksimal | 3,2 A |
| Jumlah Pin | 6 |

#### 4.5.1.4 Sensor DHT22



Gambar 4.4 Sensor DHT22

(Sumber: mail.tokoteknologi.co.id, 2024)

DHT22 adalah sensor suhu dan kelembapan yang umum digunakan dalam sistem monitoring. Sensor ini mampu mengukur suhu dalam rentang -40 hingga 80 °C dengan akurasi sekitar ±0,5 °C, serta kelembapan antara 0% hingga 100% dengan akurasi ±2–5%.

Tabel 2.4 Spesifikasi Sensor DHT22

|  |  |
| --- | --- |
| Sensor | DHT22 |
| Tegangan Kerja | 3,3-6 V |
| Sistem Komunikasi | Serial (single – Wire Two way) |
| Range Suhu | 400C – 800C |
| Range Kelembapan | 0% – 100% RH |
| Akurasi | ±20C (suhu), ±5% RH (kelembapan) |
| Jumlah Pin | 4 |

#### 4.5.1.5 Pelindung Baterai 18650 V3



Gambar 4.5 Pelindung Baterai 18650 V3

(Sumber: mail.tokoteknologi.co.id, 2024)

Pelindung Baterai 18650 V3 adalah modul baterai yang dapat digunakan untuk memberikan daya listrik dengan berbagai keluaran. Modul ini dapat mengakomodasi baterai Li-Ion 18650 dan menyediakan port USB-A untuk memberikan daya 5V, serta 3 port 5V (hingga 4A) dan 3 port 3.3V (hingga 1A) sebagai sumber daya. Dilengkapi dengan indikator LED untuk status pengisian baterai dan mekanisme perlindungan baterai.

Tabel 2.5 Spesifikasi Pelindung Baterai 18650 V3

|  |  |
| --- | --- |
| Pelindung Baterai | 18650 V3 |
| Input Port | 5V/500mA Micro USB |
| Output Catu Daya | 5VDC USB tipe A port, 3x 5VDC (hingga 4A), 3x 3VDC (hingga 1A) |
| Konverter Tegangan | Chip konverter DC-DC boost FP6298 dan Regulator tegangan XC6206 |

#### 4.5.1.6 Baterai 18650



Gambar 4.6 Baterai 18650

(Sumber: digiwarestore.com, 2024)

Tabel 2.6 Spesifikasi Pelindung Baterai 18650 V3

|  |  |
| --- | --- |
| Baterai | 18650 |
| Jenis Baterai | Lithium Ion / Li-Ion |
| Kapasitas | 3000mAh / 3Ah |
| Resistansi | <40mΩ |

#### 4.5.1.7 Modul Micro SD



Gambar 4.7 Modul Micro SD

(Sumber: jogjarobotika.com, 2024)

Pemrosesan dan penyimpanan data pada kartu Micro SD dilakukan melalui Modul Micro SD. Modul ini menggunakan antarmuka Serial Peripheral Interface (SPI) untuk mengakses kartu Micro SD dalam operasi baca dan tulis data. Agar Modul Micro SD dapat berfungsi, perlu ditambahkan akses ke pustaka SD.h dan SPI.h pada Arduino IDE. Modul ini juga dilengkapi dengan konektor yang sesuai untuk kartu Micro SD, sehingga memudahkan pemasangan kartu ke dalam modul.

Tabel 2.7 *Modul Micro SD*

|  |  |
| --- | --- |
| Modul | Micro SD |
| Tegangan Kerja | 4,5 -5,5V |
| Jenis SD Card | Micro SD Card (hingga 2GB), Micro SDHC Card hingga 32GB) |
| Resistansi | <40mΩ |
| Jumlah Pin | 6 |

#### 4.5.1.8 Micro SD



Gambar 4.7 Micro SD

(Sumber: ie.mobilefun.com, 2024)

Tabel 2.8 Micro SD

|  |  |
| --- | --- |
| Jenis Penyimpanam | Micro SD |
| Ukuran Memori | 4 GB |
| Kecepatan | 14,8 MB/S |

### Kebutuhan Perangkat Lunak

Perangkat lunak memiliki peran penting dalam pembuatan sistem, digunakan untuk mengoperasikan berbagai komponen sistem melalui kode program. Berikut adalah perangkat lunak yang digunakan oleh penulis.

#### 4.5.2.1 Arduino IDE

Arduino IDE adalah IDE (Integrated Development Environment) yang digunakan sebagai teks editor dan mengunggah kode program ke perangkat keras seperti ESP8266 dan IDE ini juga mendukung berbagai jenis perangkat Arduino lainnya, seperti Arduino Nano, Arduino Uno, ESP32, dan lainnya.

#### 4.5.2.2 Pustaka espnow.h

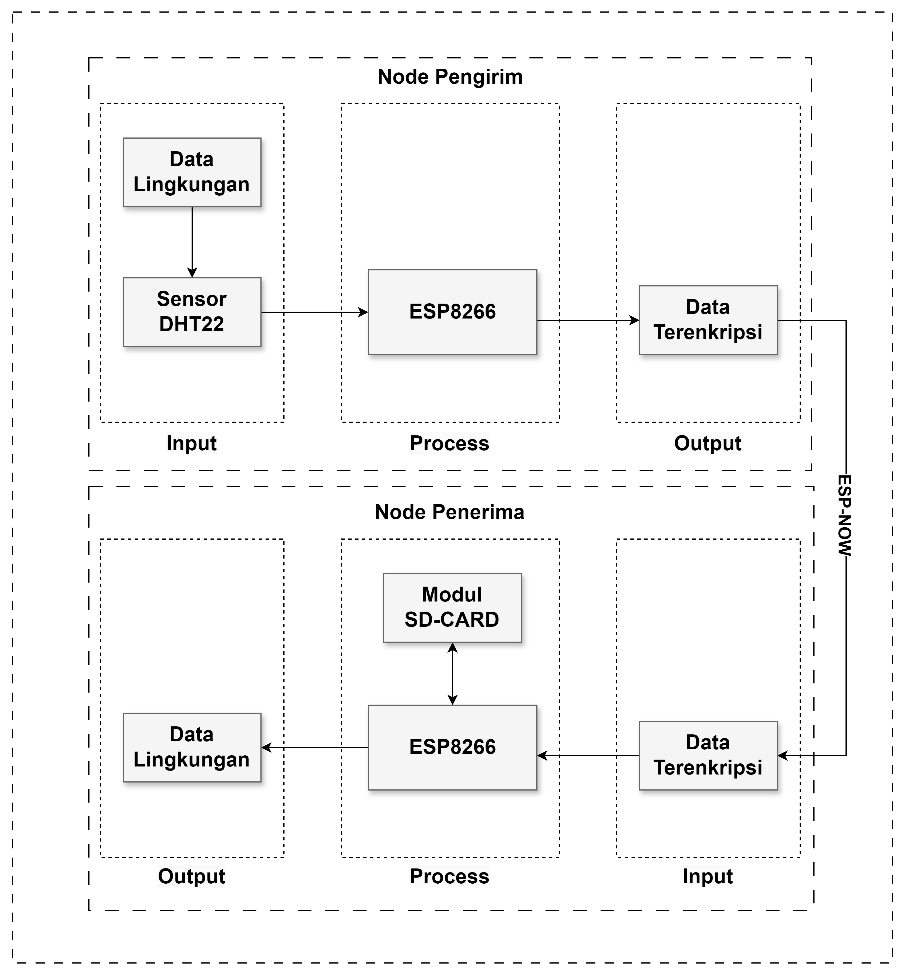
Library ini digunakan untuk mengakses program pada protokol komunikasi ESP-NOW yang dirancang khusus untuk perangkat ESP. Dengan menggunakan library ini, perangkat ESP dapat berkomunikasi secara langsung satu sama lain, memungkinkan data dikirimkan secara efisien.

# PERANCANGAN DAN IMPLEMENTASI

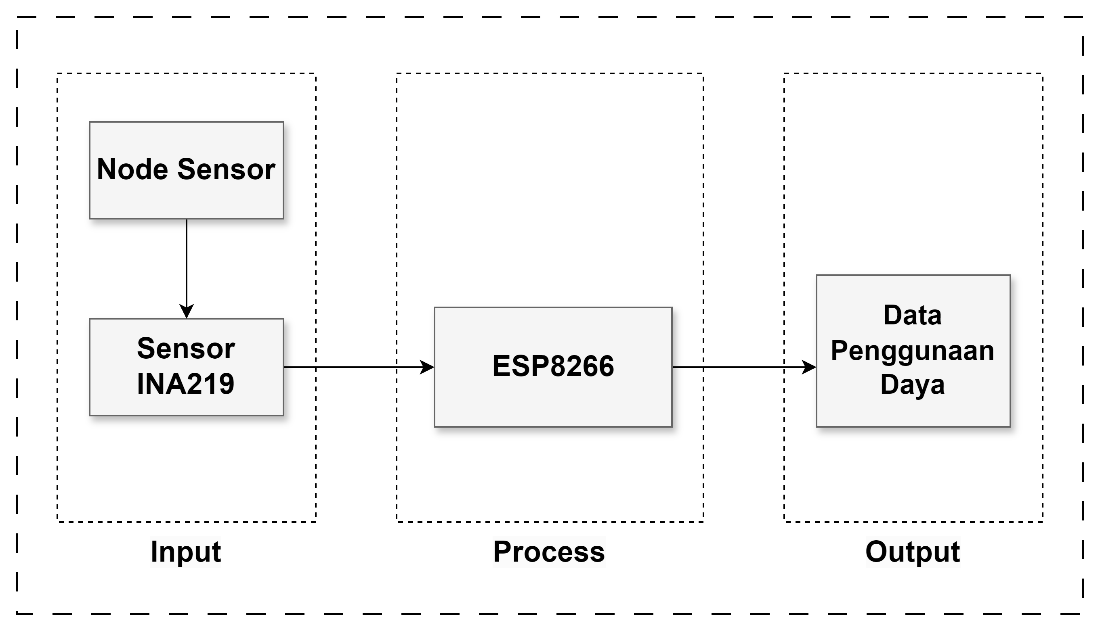
Bab ini menjelaskan proses perancangan sistem yang didasarkan pada hasil rekayasa kebutuhan. Setiap keputusan diambil dengan mempertimbangkan pemahaman yang komprehensif mengenai kebutuhan pengguna, batasan teknis, dan persyaratan sistem. Perancangan dilakukan secara sistematis dan terdokumentasi agar dapat digunakan sebagai panduan selama tahap implementasi. Rancangan tersebut mencakup aspek perangkat keras (hardware) dan perangkat lunak (software). Selain itu, bab ini juga membahas penerapan sistem yang telah dirancang guna memastikan bahwa solusi yang dihasilkan dapat memenuhi tujuan penelitian.\*

## Perancangan

Sistem ini terdiri dari dua bagian utama: sistem yang diuji dan sistem penguji. Pada sistem yang diuji, akan diimplementasikan algoritma enkripsi pada 2 mikrokontroler ESP8266 yang berfungsi sebagai enkriptor dan dekriptor, yang saling berkomunikasi menggunakan protokol ESP-NOW. Data yang digunakan untuk pengujian berasal dari sensor suhu dan kelembapan DHT22.



Gambar 5.1 Diagram Blok Perancangan Sistem

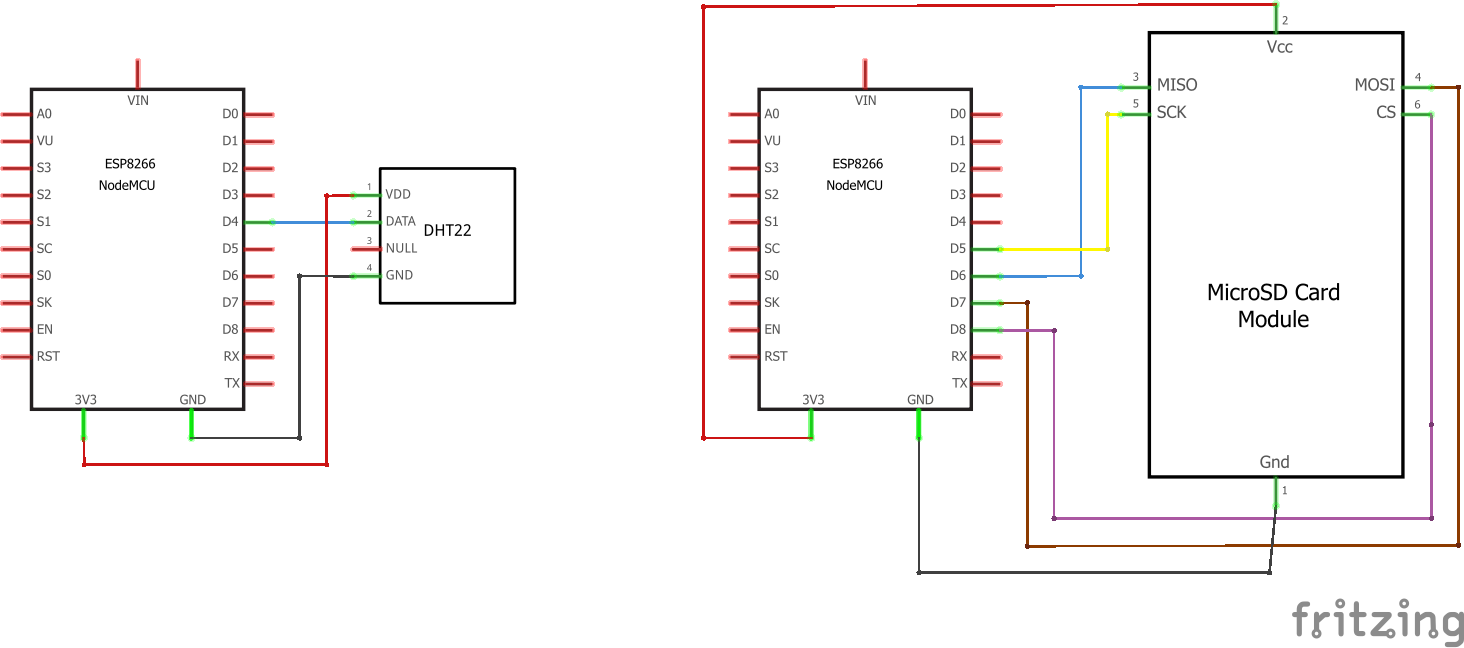


Gambar 5.2 Diagram Blok Perancangan Sistem Penguji

Sementara itu, pada sistem penguji, algoritma akan diuji dari segi kecepatan komputasi saat melakukan enkripsi dan dekripsi. Untuk pengujian efisiensi energi yang digunakan oleh tiap algoritma, sistem penguji menggunakan ESP32 dan sensor INA219, yang dapat mengukur arus dan tegangan pada ESP8266 sebagai beban. Hasil pengujian akan menunjukkan daya dan total energi yang dibutuhkan oleh tiap algoritma saat melakukan enkripsi.

### Perancangan Perangkat Keras

#### Perancangan Sistem

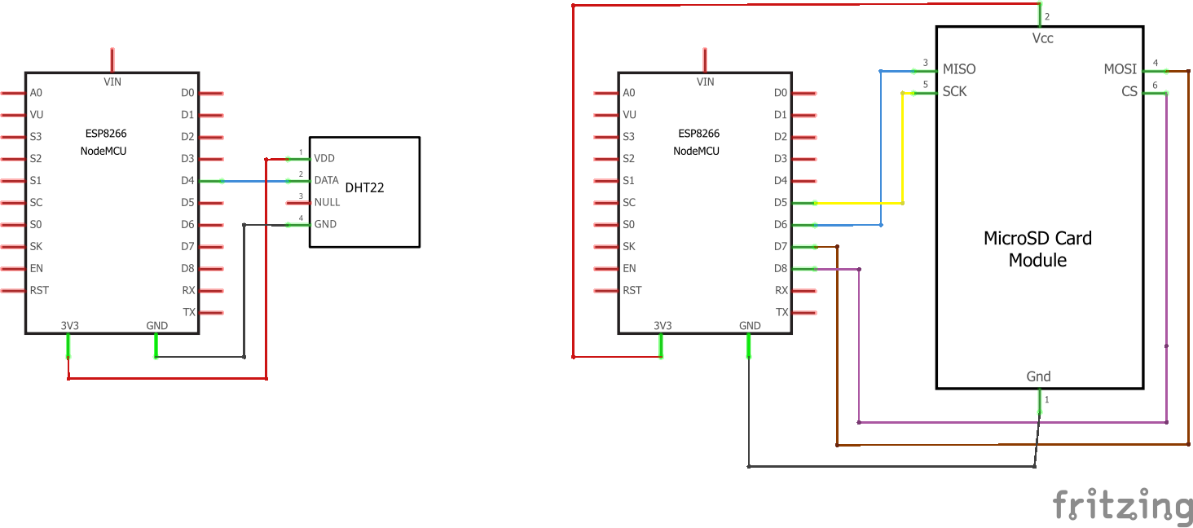


Gambar 5.3 Skematik Perancangan Sistem (Node Pengirim)

Dalam perancangan diagram skematik seperti diatas, penulis menuliskan jalur kabel dengan tabel konfigurasi pin pada Tabel 5.1 dibawah ini.

##### Tabel 5.1 Konfigurasi Pin Sistem Node Pengirim

|  |  |  |  |
| --- | --- | --- | --- |
| **Pin ESP8266** | **Pin Komponen** | **Warna Kabel** | **Komponen** |
| 3.3 V | VCC | Merah | Sensor  DHT22 |
| GND | GND | Hitam |
| D4 | Data | Biru |



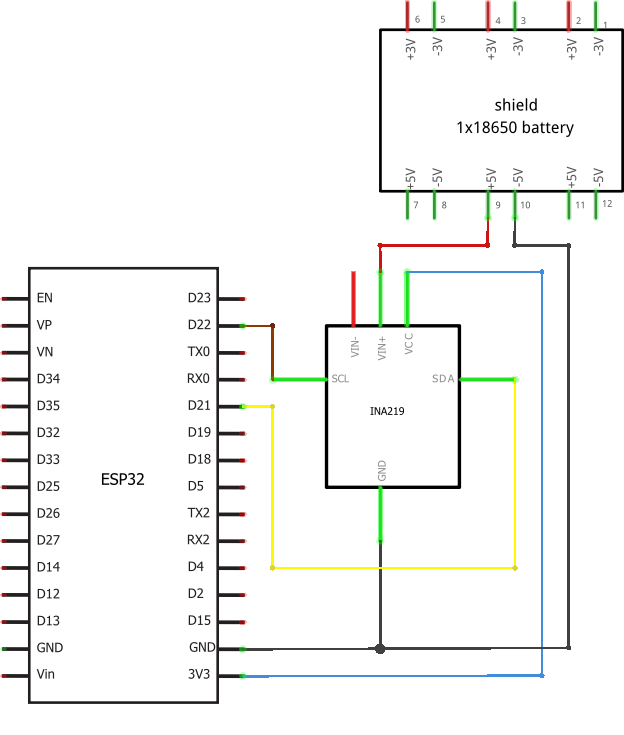
Gambar 5.4 Skematik Perancangan Sistem (Node Penerima)

Dalam perancangan diagram skematik seperti diatas, penulis menuliskan jalur kabel dengan tabel konfigurasi pin pada Tabel 5.2 dibawah ini.

##### Tabel 5.2 Konfigurasi Pin Sistem Node Penerima

|  |  |  |  |
| --- | --- | --- | --- |
| **Pin ESP8266** | **Pin Komponen** | **Warna Kabel** | **Komponen** |
| 3.3 V | VCC | Merah | Module Micro SD |
| GND | GND | Hitam |
| D6 | MISO | Biru |
| D5 | SCK | Kuning |
| D7 | MOSI | Coklat |
| D8 | CS | Ungu |

#### Perancangan Sistem Pengukur Daya



Gambar 5.5 Skematik Perancangan Sistem Pengukur Daya

Dalam perancangan diagram skematik seperti diatas, penulis menuliskan jalur kabel dengan tabel konfigurasi pin pada Tabel 5.3 dibawah ini.

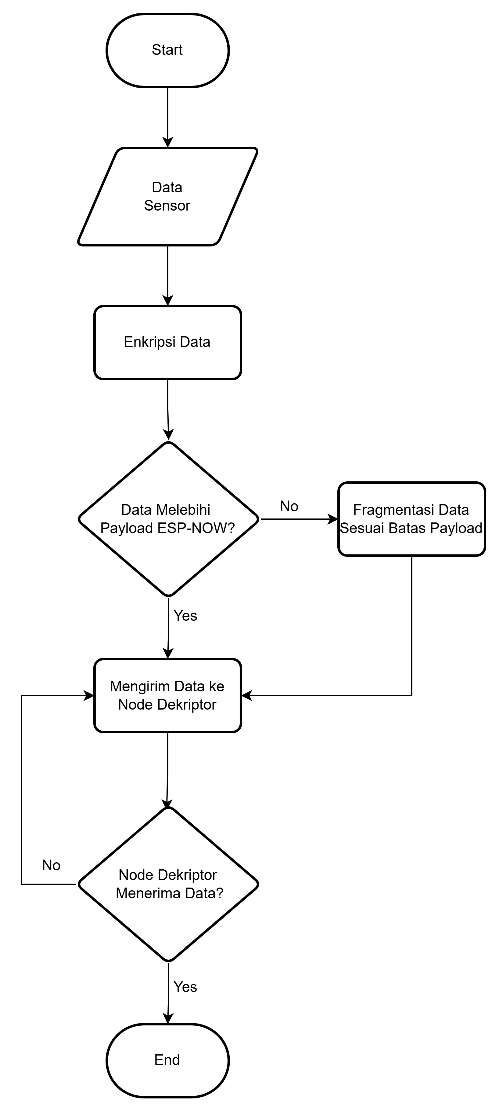
##### Tabel 5.3 Konfigurasi Pin Sistem Pengukur Daya

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Pin ESP32** | **Pin Battery Shield** | **Pin Komponen** | **Warna Kabel** | **Komponen** |
| 3.3 V | - | VCC | Biru | Sensor INA219 |
| GND | - | GND | Hitam |
| GPIO22 | - | SCL | Coklat |
| GPIO21 | - | SDA | Kuning |
| - | 5V- | GND | Hitam |
| - | 5V+ | VIN+ | Merah |
| - | - | VIN- | - |

### Perancangan Perangkat Lunak

Perangkat lunak berperan penting dalam sistem, mengendalikan perangkat keras agar berfungsi sesuai tujuan dan memungkinkan komunikasi antar komponen. Perancangan perangkat lunak mencakup alur pemrograman dan mekanisme komunikasi dalam sistem.

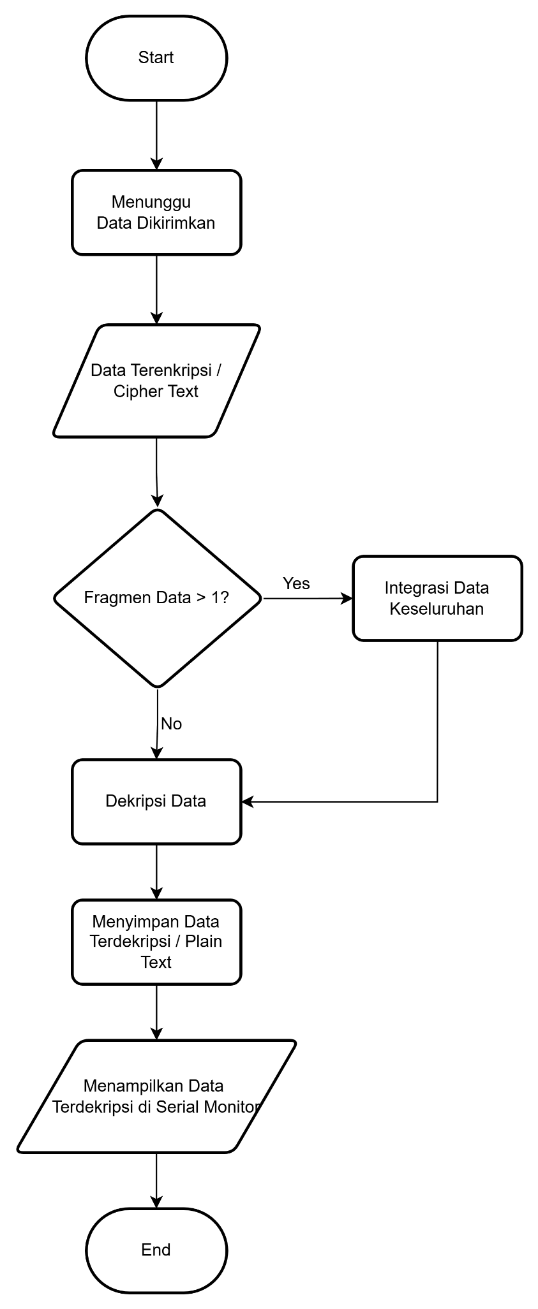
Pada fase perancangan perangkat lunak, secara umum dibagi menjadi 3 sistem yaitu sistem untuk node pengirim, sistem untuk node penerima, dan sistem pengukur daya. Ketiga sistem tersebut berada pada node yang terpisah dikarenakan akan diuji performanya sehingga diharapkan dapat melakukan komputasi secara maksimal tanpa interferensi proses lain. Dibawah merupakan diagram alir sistem tersebut.



Gambar 5.6 Diagram Alir Sistem Node Pengirim

Pada proses ini dimulai dengan pengumpulan data dari sensor. Setelah data dikumpulkan, data tersebut dienkripsi untuk menjaga keamanan informasi. Selanjutnya, sistem memeriksa apakah ukuran data yang telah dienkripsi melebihi batas payload dari protokol ESP-NOW. Jika ukuran data melebihi batas, data akan dipecah atau di-fragmentasi agar sesuai dengan batas payload yang ditentukan. Jika data sudah dalam batas yang diperbolehkan, proses fragmentasi dilewati.

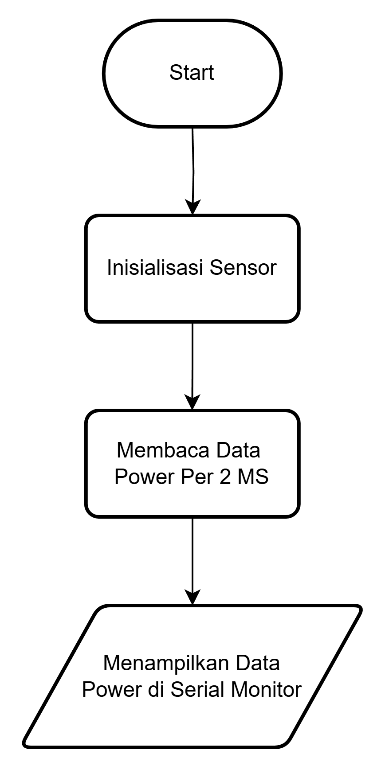
Setelah data siap dalam bentuk payload yang sesuai, data tersebut dikirim ke node penerima (node dekriptor). Sistem kemudian memeriksa apakah node penerima telah menerima data tersebut. Jika data belum diterima, proses pengiriman diulang hingga data berhasil diterima. Jika data telah diterima oleh node dekriptor, proses selesai.



Gambar 5.7 Diagram Alir Sistem Node Penerima

Pada proses ini dimulai dengan sistem berada dalam keadaan menunggu hingga data terenkripsi dikirim dan diterima. Data yang diterima dalam bentuk cipher text atau teks terenkripsi ini kemudian diperiksa untuk melihat apakah data tersebut terdiri dari lebih dari satu fragmen. Jika data memiliki lebih dari satu fragmen, sistem akan menggabungkan semua fragmen tersebut melalui proses integrasi untuk mendapatkan data utuh. Jika data yang diterima hanya terdiri dari satu fragmen, maka proses penggabungan tidak diperlukan, dan sistem dapat langsung melanjutkan ke tahap berikutnya.

Setelah data utuh diperoleh, baik melalui integrasi atau langsung dari satu fragmen, data tersebut kemudian didekripsi untuk dikonversi kembali menjadi plain text atau teks asli. Data yang telah berhasil didekripsi ini kemudian disimpan untuk referensi lebih lanjut. Sebagai langkah akhir, data yang telah didekripsi ditampilkan pada serial monitor untuk tujuan pemantauan atau debugging, sehingga dapat diperiksa dan divalidasi oleh pengguna.



Gambar 5.8 Diagram Alir Sistem Pengukur Daya

Proses dimulai dengan menginisialisasi sensor INA219. Setelah sensor siap, sistem akan membaca data daya yang dihasilkan setiap 2 milidetik (ms) secara berulang. Pembacaan data daya pada interval waktu yang sangat singkat ini memungkinkan sistem untuk memantau perubahan daya secara *real-time* atau hampir *real-time.*

Setelah data daya berhasil dikumpulkan, data tersebut ditampilkan pada serial monitor. Tampilan ini memungkinkan pengguna untuk melihat dan memantau data daya yang terbaca secara langsung pada perangkat, yang berguna untuk analisis mengenai jumlah energi yang digunakan tiap algoritma.

## Implementasi Sistem

Pada bagian implementasi, akan dijelaskan secara lebih spesifik mengenai proses penerapan perangkat keras dan perangkat lunak pada sistem yang telah dirancang sebelumnya dalam subbab perancangan.

### Implementasi Perangkat Keras

### Implementasi Perangkat Lunak

Pada bagian implementasi, akan dijelaskan implementasi bagian kode dari algoritma ChaCha20, AES, Snow-V, dan Clefia proses penerapan perangkat keras dan perangkat lunak pada sistem yang telah dirancang sebelumnya dalam subbab perancangan.

#### 4.5.1.1 Implementasi Program ESP-NOW

|  |  |
| --- | --- |
| Program 5.2.2.1: Impor library yang dibutuhkan | |
| 1  2  4  5  6  7  8  9  10  11  12 | #include <ESP8266WiFi.h>  #include <espnow.h>  #include <chrono>  using namespace std::chrono;  size\_t totalChunks = 0;  size\_t chunksAcked = 0;  bool allChunksSent = false;  bool status;  const size\_t MAX\_CHUNK\_SIZE = 250;  const size\_t MAX\_INPUT\_SIZE = 16384;  uint8\_t receiverMAC[] = {0x84, 0xF3, 0xEB, 0x05, 0x50, 0xB7};  void onSend(uint8\_t \*mac\_addr, uint8\_t deliveryStatus) {    bool sendStatus;      if (deliveryStatus == 0) {          status = true;          chunksAcked++;      } else {        status = false;      }      if (chunksAcked == totalChunks && deliveryStatus == 0) {          allChunksSent = true;          Serial.println("All chunks sent successfully");      }  }  bool initESPNow() {      if (esp\_now\_init() != 0) {          Serial.println("Error initializing ESP-NOW");          return false;      }      esp\_now\_set\_self\_role(ESP\_NOW\_ROLE\_CONTROLLER);      esp\_now\_register\_send\_cb(onSend);      return true;  } |

### 

DAFTAR REFERENSI

Astuti, L.D. and Wibisono, W., 2017. Peningkatan Networklifetimepada Wireless Sensor Network Menggunakan Clustered Shortest Geopath Routing (C-SGP) Protocol. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), 4(3), pp.148-15.

De Santis, F., Schauer, A. and Sigl, G., 2017, March. ChaCha20-Poly1305 authenticated encryption for high-speed embedded IoT applications. In Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017 (pp. 692-697). IEEE.

Gunathilake, N.A., Buchanan, W.J. and Asif, R., 2019, April. Next generation lightweight cryptography for smart IoT devices:: implementation, challenges and applications. In 2019 IEEE 5th World Forum on Internet of Things (WF-IoT) (pp. 707-710). IEEE.

Iman, M.F., Kusyanti, A. and Primananda, R., 2022. Implementasi Algoritme Clefia 128-Bit untuk Pengamanan Modul Komunikasi Lora. Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), 9(7).

Kane, L. E., Chen, J. J., Thomas, R., Liu, V., & Mckague, M. (2020). Security and performance in IoT: A balancing act. *IEEE access*, *8*, 121969-121986.

Lee, Hyeopgeon, Kyounghwa Lee, and Yongtae Shin. "Implementation and Performance Analysis of AES-128 CBC algorithm in WSNs." In 2010 The 12th International Conference on Advanced Communication Technology (ICACT), vol. 1, pp. 243-248. IEEE, 2010.

Maitra, S., Richards, D., Abdelgawad, A., & Yelamarthi, K. (2019, March). Performance evaluation of IoT encryption algorithms: memory, timing, and energy. In *2019 IEEE sensors applications symposium (SAS)* (pp. 1-6). IEEE.

Pratama, Y. A., Budi, A. S., & Kusyanti, A. (2021). Implementasi Algoritma Enkripsi Snow-V pada Wireless Sensor Network (WSN). Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 5(10), 4689-4697.

Qadir, A.M. and Varol, N., 2019, June. A review paper on cryptography. In 2019 7th international symposium on digital forensics and security (ISDFS) (pp. 1-6). IEEE.

Sarker, V.K., Gia, T.N., Tenhunen, H. and Westerlund, T., 2020, June. Lightweight security algorithms for resource-constrained IoT-based sensor nodes. In ICC 2020-2020 IEEE International Conference on Communications (ICC) (pp. 1-7). IEEE.

Urazayev, D., Eduard, A., Ahsan, M., & Zorbas, D. (2023, May). Indoor performance evaluation of ESP-NOW. In 2023 IEEE International Conference on Smart Information Systems and Technologies (SIST) (pp. 1-6). IEEE