

**PENGAMANAN KARTU TAG RFID (*RADIO FREQUENCY
IDENTIFICATION*) UNTUK PEMINJAMAN BUKU PERPUSTAKAAN**

SKRIPSI

SITI JUBAIDAH MUNTHE

201401014



PROGRAM STUDI ILMU KOMPUTER

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS SUMATERA UTARA

MEDAN

2024

**PENGAMANAN KARTU TAG RFID (*RADIO FREQUENCY
IDENTIFICATION*) UNTUK PEMINJAMAN BUKU PERPUSTAKAAN**

SKRIPSI

**Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah
Sarjana Ilmu Komputer**

SITI JUBAIDAH MUNTHE

201401014



PROGRAM STUDI ILMU KOMPUTER

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UNIVERSITAS SUMATERA UTARA

MEDAN

2024

UNIVERSITAS SUMATERA UTARA

PERSETUJUAN

Judul : PENGAMANAN KARTU TAG RFID (*RADIO
FREQUENCY IDENTIFICATION*) UNTUK
PEMINJAMAN BUKU PERPUSTAKAAN

Kategori : SKRIPSI

Nama : SITI JUBAIDAH MUNTHER

Nomor Induk Mahasiswa : 201401014

Program Studi : SARJANA (S1) ILMU KOMPUTER

Fakultas : ILMU KOMPUTER DAN TEKNOLOGI
INFORMASI UNIVERSITAS SUMATERA
UTARA

Telah diuji dan dinyatakan lulus di Medan, 15 Oktober 2024

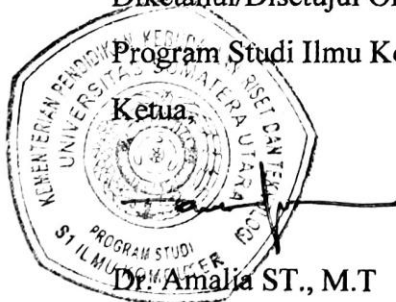
Pembimbing II

Seniman S.Kom., M.Kom
NIP. 196203171991031001

Pembimbing I

Prof. Dr. Syahril Efendi S.Si., M.IT
NIP. 196711101996021001

Diketahui/Disetujui Oleh
Program Studi Ilmu Komputer



Ketua,
Dr. Amalia ST., M.T
NIP. 197812212014042001

PERNYATAAN**PENGAMANAN KARTU TAG RFID (*RADIO FREQUENCY
IDENTIFICATION*) UNTUK PEMINJAMAN BUKU PERPUSTAKAAN****SKRIPSI**

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 30 Mei 2024



Siti Jubaidah Munthe

201401014

PENGHARGAAN

Bismillahirrahmanirrahim, Dengan menyebut nama Allah Yang Maha Pengasih lagi Maha Penyayang, segala puji hanya milik Allah, Tuhan semesta alam. Tiada kata yang pantas terucap selain kata syukur kepada Allah Swt, berkat limpahan dan rahmat-Nya penulis mampu menyelesaikan skripsi ini sebagai syarat gelar Sarjana Komputer, pada Program Studi S1 Ilmu Komputer Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara. Shalawat serta salam tak lupa penulis haturkan kepada junjungan kita Nabi Muhammad SAW yang mana berkat rahmat Beliau kita mampu merasakan dunia yang penuh dengan ilmu pengetahuan ini.

Penulis ingin mengungkapkan penghargaan dan terima kasih yang besar kepada:

1. Kedua orang tua tercinta, Ibunda Irma Suriani dan Ayahanda M. Ayunan Munthe. Terima kasih atas kasih sayang, doa, semangat, dukungan, serta pengorbanan yang tak terhingga. Semoga kalian selalu sehat, panjang umur, dan bahagia.
2. Bapak Dr. Muryanto Amin, S.Sos, M.Si, selaku Rektor Universitas Sumatera Utara.
3. Ibu Dr. Maya Silvi Lidya, B.Sc, M.Sc, selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi.
4. Ibu Dr. Amalia, S.T, M.T, selaku Ketua Program Studi S-1 Ilmu Komputer Universitas Sumatera Utara.
5. Ibu Desilia Selvida S.Kom., M.Kom, selaku Dosen Pembimbing Akademik.
6. Bapak Prof. Dr. Syahril Efendi S.Si., M.IT, selaku Dosen Pembimbing I. Terima kasih atas bimbingan, kritik, saran, dan waktu yang diberikan di tengah kesibukan.

7. Bapak Seniman S.Kom., M.Kom, selaku Dosen Pembimbing II. Terima kasih atas bimbingan, kritik, saran, dan waktu yang diberikan di tengah kesibukan..
8. Seluruh staf pengajar dan pegawai Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara yang telah membantu dalam proses penyusunan skripsi ini.
9. Saudara kandungku, Kakak Nur Aini Br Munthe, Rizki Ananda Munthe, Sri Atika Munthe, dan saudara ipar Abang Gundong Siregar, yang selalu memberikan dorongan dan motivasi hingga mencapai tahap ini. Semoga kalian selalu sehat, panjang umur, dan bahagia.
10. Seluruh keluarga yang selalu memberikan dorongan dan motivasi.
11. Seluruh teman-teman Kom A dan Stambuk 2020 yang telah memberikan semangat baik secara langsung maupun tidak langsung.
12. Semua pihak yang terlibat, baik secara langsung maupun tidak langsung, yang tidak dapat disebutkan satu per satu.

Medan, 30 Mei 2024



Siti Jubaidah Munthe

201401014

PENGAMANAN KARTU TAG RFID (*RADIO FREQUENCY IDENTIFICATION*) UNTUK PEMINJAMAN BUKU PERPUSTAKAAN

ABSTRAK

Dalam era digital saat ini, teknologi RFID (*Radio Frequency Identification*) telah banyak diterapkan dalam berbagai sistem otomatisasi, termasuk sistem peminjaman buku di perpustakaan. Penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan sistem keamanan berbasis RFID guna meningkatkan keamanan dan efisiensi proses peminjaman buku di perpustakaan. Fokus utama dari penelitian ini adalah mengatasi masalah keamanan yang sering timbul, seperti duplikat kartu dan manipulasi, dengan menggunakan teknologi enkripsi pada kartu tag RFID. Hasil penelitian menunjukkan bahwa sistem RFID dengan fitur keamanan tambahan mampu secara signifikan mengurangi risiko duplikat kartu dan manipulasi. Implementasi sistem ini juga terbukti meningkatkan efisiensi proses peminjaman dan pengembalian buku, serta mempermudah manajemen inventaris perpustakaan. Dengan demikian, penelitian ini menyimpulkan bahwa penggunaan teknologi RFID yang dilengkapi dengan protokol keamanan dapat menjadi solusi efektif untuk mengatasi masalah keamanan dalam sistem peminjaman buku perpustakaan menggunakan kartu tag RFID.

Kata Kunci : RFID, Perpustakaan, Pengamanan, Kartu RFID, Enkripsi, Dekripsi, Algoritma AES.

SECURITY OF RFID (RADIO FREQUENCY IDENTIFICATION) TAG CARDS FOR LIBRARY BOOK BORROWING

Abstract

In the current digital era, RFID (Radio Frequency Identification) technology has been widely applied in various automation systems, including book lending systems in libraries. This research aims to develop and implement an RFID-based security system to improve the security and efficiency of the book borrowing process in libraries. The main focus of this research is to overcome security problems that often arise, such as duplicate cards and manipulation, by using encryption technology on RFID tag cards. The research results show that an RFID system with additional security features can significantly reduce the risk of duplicate cards and manipulation. Implementation of this system has also been proven to increase the efficiency of the book borrowing and returning process, as well as simplifying library inventory management. Thus, this research concludes that the use of RFID technology equipped with security protocols can be an effective solution to overcome security problems in library book lending systems using RFID tag cards.

Keywords: *RFID, Library, Security, RFID Card, Encryption, Decryption, AES Algorithm.*

DAFTAR ISI

PERSETUJUAN	i
PERNYATAAN	ii
PENGHARGAAN	iii
ABSTRAK	v
ABSTRACT	Error! Bookmark not defined.
DAFTAR ISI	vii
DAFTAR GAMBAR	x
DAFTAR TABEL	xii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Metodologi Penelitian	3
1.7 Penelitian Relevan	4
1.8 Sistematika Penulisan	6
BAB 2 LANDASAN TEORI	8
2.1 Mikrokontroler	8
2.2 Internet of Things (IoT)	8
2.3 RFID (Radio Frequency Identification)	8
2.3.1 RFID Tag	9
2.3.2 RFID Reader	10
2.4 RFID RC 522	10
2.5 Algoritma Advanced Encryption Standard (AES)	11
2.6 Arduino Uno	12
2.7 MIFARE Classic 1K	13
2.7.1 Struktur Memory MIFARE Classic 1 K.....	13
2.8 Node Js	14
2.9 Express Js	15
2.10 Kabel Jumper	15

2.11 Database	16
BAB 3 ANALISIS DAN PERANCANGAN	18
3.1 Analisis Sistem	18
3.1.1 Analisis Masalah	18
3.1.2 Analisis Kebutuhan	18
3.2 Arsitektur Umum	19
3.3 Pemodelan Sistem	20
3.3.1 Use Case Diagram	21
3.3.2 Activity Diagram	22
3.4 Flowchart	23
3.5 Perancangan <i>Interface</i> Sistem	25
3.5.1 Halaman <i>Registration</i>	25
3.5.2 Halaman Peminjaman Buku	26
3.6 Perancangan Alat	27
BAB 4 IMPLENTASI DAN PENGUJIAN SISTEM	30
4.1 Implementasi Sistem	30
4.1.1 Perangkat Keras (<i>Hardware</i>)	30
4.1.2 Perangkat Lunak (<i>Software</i>)	30
4.1.3 Desain Alat	31
4.2 Proses Pada Program	31
4.2.1 Mengimport Module dan <i>Library</i>	32
4.2.2 Mengimport dan Mengkonfigurasi Enkripsi dan Dekripsi AES ...	32
4.2.3 Membuat <i>Instance Express</i>	33
4.2.4 Membuat HTTP <i>Server</i>	33
4.2.5 Mengimpor dan Mengkonfigurasi <i>Socket.io</i>	34
4.2.6 Mengatur Direktori dan <i>View Engine</i>	34
4.2.7 Mengatur Sesi dan <i>Flash Message</i>	35
4.2.8 Koneksi <i>Serial</i>	36
4.2.9 <i>Endpoint</i> Untuk Mengirim Data Ke Arduino	37
4.3 Implementasi Model Kedalam Web	38
4.3.1 Halaman <i>Registration</i>	38
4.3.2 Halaman Peminjaman Buku dan Tambah Buku	39
BAB 5 KESIMPULAN DAN SARAN	52
5.1 Kesimpulan	52

5.2	Saran.....	52
DAFTAR PUSTAKA		53

DAFTAR GAMBAR

Gambar 2.1 Kartu Tag RFID	10
Gambar 2.2 RFID RC 522	11
Gambar 2.10 Gambar Struktur Memori pada Kartu RFID.....	13
Gambar 2.11 Kabel Jumper	16
Gambar 3.1 Arsitektur Umum	20
Gambar 3.2 Use Case Diagram	21
Gambar 3.3 Activity Diagram	22
Gambar 3.4 Flowchart	24
Gambar 3.5 Desain Halaman Registration	25
Gambar 3.6 Desain Halaman Pinjaman Buku	26
Gambar 3.7 Desain Tampilan Tambah Buku	27
Gambar 3.9 Gambar Rangkaian	28
Gambar 4.1 Desain Alat Sistem	32
Gambar 4.2 Mengimpor Module dan Library	32
Gambar 4.3 Mengimpor dan Mengkonfigurasi Enkripsi dan Dekripsi AES	33
Gambar 4.4 Memebuat Instance <i>Express</i>	33
Gambar 4.5 Membuat HTTP Server.....	34
Gambar 4.6 Mengimpor dan Mengkonfigurasi <i>Socket.io</i>	34
Gambar 4.7 Mengatur Direktori dan <i>View Engine</i>	35
Gambar 4.8 Mengatur Sesi dan <i>Flash Message</i>	36
Gambar 4.9 Koneksi Serial.....	37
Gambar 4.10 <i>Endpoint</i> Untuk Mengirim Data Ke Arduino.....	37
Gambar 4.11 Halaman <i>Registration</i>	38
Gambar 4.12 Halaman Peminjaman Buku.....	39
Gambar 4.13 Halaman Tambah Buku.....	39
Gambar 4.14 Hasil Uji Read Kartu (1) Pada RFID Reader.....	44
Gambar 4.15 Hasil Uji Read Kartu (1) Pada MIFARE Classic Tool.....	45
Gambar 4.16 Hasil Uji Read Kartu (2) Pada RFID Reader.....	46
Gambar 4.17 Hasil Uji Read Kartu (2) Pada MIFARE Classic Tool.....	47

Gambar 4.18 Hasil Uji Read Kartu (3) Pada RFID Reader.....	48
Gambar 4.19 Hasil Uji Read Kartu (3) Pada MIFARE Classic Tool.....	49
Gambar 4.20 Hasil Uji Read Kartu (4) Pada RFID Reader.....	50
Gambar 4.21 Hasil Uji Read Kartu (4) Pada MIFARE Classic Tool.....	51
Gambar 4.22 Hasil Uji Read Kartu (5) Pada RFID Reader.....	52
Gambar 4.23 Hasil Uji Read Kartu (5) Pada MIFARE Classic Tool.....	53

DAFTAR TABEL

Tabel 3.1 Koneksi RFID RC 522 dengan Arduino Uno	28
Tabel 4. 1 Perangkat Keras (<i>Hardware</i>)	30
Tabel 4.2 Perangkat Lunak (<i>Software</i>).....	31
Tabel 4.3 Hasil Pengujian Jarak <i>Reader</i>	40
Tabel 4.4 Hasil Pengujian Waktu <i>Reader</i>	40

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perpustakaan adalah institusi atau tempat yang mengumpulkan, mengelola, dan menyediakan akses ke berbagai sumber informasi, seperti buku, majalah, surat kabar, jurnal, peta, manuskrip, dan bahan multimedia lainnya. Perpustakaan bertujuan untuk mendukung pendidikan, penelitian, dan rekreasi dengan menyediakan bahan bacaan dan referensi kepada masyarakat umum, pelajar, peneliti, dan profesional (Nugraha, 2014).

Perkembangan teknologi informasi telah membawa perubahan signifikan dalam berbagai aspek kehidupan, termasuk dalam manajemen perpustakaan. Salah satu inovasi penting dalam pengelolaan perpustakaan adalah penggunaan teknologi RFID (*Radio Frequency Identification*) untuk sistem peminjaman buku. Teknologi RFID menawarkan berbagai keuntungan seperti efisiensi, kecepatan dalam proses peminjaman dan pengembalian buku, serta pengelolaan inventaris yang lebih baik. Namun, seiring dengan meningkatnya penggunaan RFID, muncul juga tantangan baru terkait keamanan data (Fahrizandi, 2020)

Teknologi *Radio Frequency Identification* (RFID) telah mengalami kemajuan yang pesat dalam beberapa tahun terakhir. Beberapa faktor mendukung kemajuan ini, terutama kebutuhan yang meningkat akan aplikasi konsumen yang menggunakan teknologi tersebut. RFID mampu mengidentifikasi objek secara otomatis dan diprediksi akan menggantikan barcode yang lebih dikenal (Mutiarwati, 2004). Selain itu, biaya RFID telah menjadi lebih terjangkau karena perkembangan industri manufaktur silikon. Sistem RFID terdiri dari tag radio atau transponder dan pembaca *tag* atau penerima. Pembaca *tag* membaca data yang dipindai melalui frekuensi radio (Purnomo, 2021).

Kartu *tag* RFID bekerja dengan memancarkan sinyal radio yang dapat dibaca oleh perangkat pembaca RFID. Proses ini membuat informasi yang tersimpan dalam *tag* RFID dapat diakses secara nirkabel, sehingga memperbesar risiko data tersebut jatuh ke tangan yang tidak berwenang. Oleh karena itu,

diperlukan mekanisme pengamanan yang handal untuk melindungi informasi yang tersimpan di dalam kartu tag RFID.

Ancaman keamanan ini menuntut adanya mekanisme perlindungan yang dapat menjamin kerahasiaan dan integritas data yang disimpan dan ditransmisikan oleh *tag* RFID. Salah satu solusi yang menjanjikan adalah penerapan algoritma enkripsi untuk melindungi data. *Advanced Encryption Standard* (AES) adalah salah satu algoritma enkripsi simetris yang paling banyak digunakan untuk sistem keamanan. AES dipilih karena kemampuannya dalam memberikan perlindungan kuat terhadap data sekaligus efisiensi dalam proses enkripsi dan dekripsi, yang sangat penting mengingat keterbatasan sumber daya pada perangkat RFID (Bhaskoro et al., 2023).

Algoritma AES bekerja dengan mengenkripsi data menggunakan kunci rahasia sehingga data yang disimpan pada *tag* RFID tidak dapat dibaca oleh pihak yang tidak berwenang tanpa kunci tersebut. Dengan demikian, meskipun *tag* RFID jatuh ke tangan yang salah atau dibaca oleh pembaca yang tidak sah, data di dalamnya tetap terlindungi. Implementasi AES dalam sistem RFID untuk perpustakaan diharapkan dapat mengurangi risiko pencurian dan penyalahgunaan data, serta meningkatkan kepercayaan pengguna terhadap sistem perpustakaan yang telah dibuat (Fachrozi & Fahmi, 2021).

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, dapat dirumuskan masalah penelitian pada sistem pengamanan kartu tag RFID (*Radio Frequency Identification*) dengan menggunakan algoritma AES (*Advanced Encryption Standard*)

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini mencakup:

1. Penelitian ini hanya akan menggunakan algoritma *Advanced Encryption Standard* (AES) untuk enkripsi dan dekripsi data.
2. Bahasa pemrograman menggunakan *javascript*.
3. Kartu yang digunakan yaitu kartu MIFARE 13.56 Mhz 1K.
4. Data yang terdapat pada kartu MIFARE *Clasic* 1k, adalah *hexadecimal*.

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah mengembangkan sistem keamanan untuk kartu *tag* RFID dalam konteks peminjaman buku perpustakaan dengan menggunakan algoritma AES serta mengurangi risiko penyalahgunaan kartu *tag* RFID oleh pihak yang tidak berwenang, serta melindungi privasi pengguna.

1.5 Manfaat Penelitian

1. penggunaan AES (*Advanced Encryption Standard*) memastikan keamanan data yang tinggi dengan mengenkripsi informasi yang dikirim antara kartu tag dan sistem perpustakaan. Hal ini mengurangi risiko duplikat dan manipulasi data.
2. Menggunakan AES, sistem perpustakaan dapat mematuhi standar keamanan data yang lebih tinggi.
3. penelitian ini tidak hanya meningkatkan keamanan, tetapi juga meningkatkan efisiensi operasional dan keandalan sistem peminjaman buku perpustakaan berbasis RFID.

1.6 Metodologi Penelitian

Beberapa metode yang digunakan dalam penelitian ini mencakup:

1. Studi Pustaka

Pada tahap awal penelitian ini, pencarian referensi dilakukan melalui studi dokumen, karya ilmiah, makalah, jurnal, serta bahan tertulis lainnya yang mendukung dan relevan dengan topik penelitian ini. Langkah ini bertujuan untuk mengumpulkan informasi dan pemahaman yang mendalam mengenai konsep, teknologi, dan metodologi yang akan digunakan.

2. Analisis dan Perancangan Sistem

Berdasarkan cakupan penelitian, pada tahapan ini akan dilakukan analisis hal-hal yang diperlukan untuk memenuhi kebutuhan pada penelitian dan merancangnya dalam bentuk diagram alir untuk menggambarkan alur kerja penelitian.

3. Implementasi Sistem

Pada tahap ini melakukan proses pembangunan sistem berdasarkan diagram alir yang sudah dirancang serta menghubungkan sistem dengan Arduino, RFID RC-522, algoritma AES dan menjadikan sebuah sistem yang berlandaskan konsep *Internet of Things* (IoT).

4. Pengujian Sistem

Tahap ini melakukan proses pengujian serta percobaan terhadap sistem yang telah dibuat untuk mengetahui apakah telah memenuhi persyaratan dan tujuan yang telah ditetapkan sebelumnya serta memastikan bahwa perancangan sistem yang dihasilkan berfungsi sesuai dengan harapan.

5. Dokumentasi

Tahap ini dilakukan dengan mendokumentasikan setiap langkah penelitian dan menarik kesimpulan akhir berupa laporan penelitian.

1.7 Penelitian Relevan

Penelitian yang dilakukan dengan melihat beberapa penelitian yang relevan terlebih dahulu antara lain:

1. Penelitian oleh (Bhaskoro et al., 2023) dengan judul “Sistem Keamanan Kartu NFC Menggunakan Metode AES Pada Sistem Pembayaran Elektronik” bertujuan untuk mengembangkan sistem keamanan yang efektif dan andal untuk kartu *Near Field Communication* (NFC) dalam konteks sistem pembayaran elektronik. Metode yang digunakan adalah *Advanced Encryption Standard* (AES) yang terkenal dengan keamanannya. Dengan mengimplementasikan AES pada kartu NFC, penelitian ini bertujuan untuk meningkatkan tingkat perlindungan terhadap informasi sensitif seperti data transaksi dan informasi keuangan pengguna. Dengan demikian, tujuan utama penelitian ini adalah untuk menyediakan solusi keamanan yang handal dan dapat diandalkan bagi sistem pembayaran elektronik menggunakan kartu NFC.

2. Penelitian oleh (Afiani, 2020) dengan judul “Desain E-Tol Dengan *Radio Frequency Identification* (RFID) Menggunakan Algoritma Kriptografi Blowfish” bertujuan untuk merancang dan mengimplementasikan sistem E-Tol berbasis teknologi *Radio Frequency Identification* (RFID) dengan menggunakan algoritma kriptografi *Blowfish* guna meningkatkan keamanan data transaksi tol. Dalam sistem E-Tol, data yang dikirimkan antara tag RFID dan pembaca sangat rentan terhadap berbagai ancaman keamanan seperti penyadapan dan pemalsuan data. Dengan menerapkan algoritma *Blowfish*, yang dikenal sebagai salah satu algoritma kriptografi simetris yang kuat dan efisien, diharapkan data yang disimpan dan ditransmisikan melalui tag RFID dapat dienkripsi dengan tingkat keamanan yang tinggi. Tujuan utama dari penelitian ini adalah untuk mengembangkan desain sistem yang memastikan bahwa data pengguna dan transaksi tetap aman dari akses dan modifikasi yang tidak sah, serta untuk menguji efektivitas dan kinerja algoritma *Blowfish* dalam lingkungan operasional E-Tol.
3. Penelitian oleh (NATALIANA et al., 2019) dengan judul “Rancang Bangun Sistem Keamanan RFID Tag Menggunakan *Metode Caesar Cipher* Pada Sistem Pembayaran Elektronik” bertujuan untuk merancang dan membangun sistem keamanan bagi *tag* RFID dalam konteks sistem pembayaran elektronik dengan menggunakan metode *Caesar Cipher*. Teknologi RFID, meskipun menawarkan banyak keuntungan dalam hal efisiensi dan kemudahan transaksi, menghadapi tantangan serius terkait keamanan data yang ditransmisikan antara *tag* RFID dan pembaca. Informasi sensitif seperti data pengguna dan rincian transaksi dapat menjadi target serangan pihak yang tidak berwenang. Untuk mengatasi masalah ini, penelitian ini berfokus pada penerapan metode enkripsi *Caesar Cipher*, sebuah teknik kriptografi klasik yang sederhana namun efektif dalam menggeser karakter data untuk mengaburkan informasi aslinya.
4. Penelitian oleh (Suryam Dora, 2022) dengan judul “*Smart Gate System* Untuk Akses Kontrol Keamanan Kampus” bertujuan untuk mengembangkan sebuah sistem keamanan informasi yang memanfaatkan algoritma *Caesar Cipher* untuk

mengkripsi data yang dikendalikan oleh *Smart Gate*. Algoritma *Caesar Cipher*, sebagai salah satu teknik kriptografi klasik, bekerja dengan menggeser setiap karakter dalam data dengan jumlah tertentu, sehingga informasi menjadi tidak dapat dibaca oleh pihak yang tidak berwenang. Dalam penelitian ini, *Visual Basic* akan digunakan untuk mengimplementasikan algoritma tersebut, serta untuk merancang antarmuka pengguna yang intuitif dan mekanisme enkripsi-dekripsi yang efisien dan untuk menguji dan mengevaluasi kinerja sistem dalam berbagai skenario operasional guna memastikan keandalan dan keamanannya dalam kondisi nyata. Pengujian ini mencakup analisis efektivitas enkripsi dalam melindungi data dan evaluasi performa sistem dalam lingkungan yang sebenarnya. Dengan demikian, diharapkan hasil dari penelitian ini dapat memberikan solusi praktis dan efektif untuk meningkatkan keamanan informasi pada *Smart Gate*, serta memberikan kontribusi signifikan terhadap pengembangan teknologi akses kontrol yang lebih aman dan andal.

1.8 Sistematika Penulisan

Struktur skripsi ini terdiri dari lima bab yang mencakup:

BAB 1 PENDAHULUAN

Bab ini menjelaskan tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan dan manfaat penelitian, metodologi penelitian, serta sistematika penulisan skripsi ini.

BAB 2 LANDASAN TEORI

Bab ini menguraikan beberapa konsep yang memiliki hubungan dengan penelitian, seperti mikrokontroler, IoT, RFID, RFID RC 522, algoritma AES.

BAB 3 ANALISIS DAN PERANCANGAN

Bab ini menguraikan hal-hal yang dianalisis serta membuat perancangan dari rancangan sistem yang akan dibangun. Selanjutnya membuat rancangan diagram alir terkait sistem tersebut.

BAB 4 IMPLEMENTASI DAN PENGUJIAN

Bab ini menguraikan tahap penerapan yang didapat dari sistem yang sudah dibangun serta dilakukan uji pada sistem untuk mendapatkan hasil dari analisis kinerja sistem tersebut.

BAB 5 KESIMPULAN DAN SARAN

Bab ini menguraikan beberapa simpulan berupa rangkuman berdasarkan pemaparan beberapa bab sebelumnya dan juga berupa saran yang dapat peneliti gunakan sebagai masukan untuk penelitian selanjutnya.

BAB 2

LANDASAN TEORI

2.1 Mikrokontroler

Mikrokontroler adalah perangkat kecil yang menggabungkan berbagai komponen penting ke dalam satu chip IC (*Integrated Circuit*), sehingga sering disebut sebagai komputer mikro tunggal. Fungsinya adalah mengontrol berbagai komponen elektronik dan biasanya dapat menyimpan program. Secara umum, *Mikrokontroler* terdiri dari *Central Processing Unit* (CPU), memori, Input/Output (I/O) khusus, dan unit pendukung seperti *Analog-to-Digital Converter* (ADC) yang terintegrasi didalamnya. Semua komponen ini disusun dalam satu *chip*, memungkinkan prosesor, memori dan I/O bekerja bersama sebagai bagian dari sistem kontrol. Dengan demikian, mikrokontroler bisa dianggap sebagai versi dari komputer yang dapat beroperasi secara fleksibel sesuai dengan kebutuhan sistem. (Nugroho & Djaksana, 2022)

2.2 Internet of Things (IoT)

Internet of Things (IoT) adalah konsep yang bertujuan untuk memperluas manfaat dan konektivitas internet yang terus-menerus. Dengan kemajuan infrastruktur internet yang terus berkembang, kita menuju masa dimana tidak hanya perangkat seperti ponsel pintar atau komputer yang dapat terhubung dengan internet, tetapi juga berbagai objek fisik lainnya. Objek-objek ini dapat mencakup peralatan produksi, kendaraan, perangkat elektronik, produk yang dapat dipakai (*wearables*), dan berbagai objek fisik lainnya yang terhubung ke jaringan lokal maupun global melalui sensor dan aktuator yang terintegrasi. (Mehta, 2021)

2.3 RFID (Radio Frequency Identification)

RFID (*Radio Frequency Identification*) adalah suatu teknologi yang memungkinkan identifikasi dan pemantauan objek secara otomatis menggunakan gelombang radio. Sistem RFID terdiri dari beberapa komponen utama, yaitu *tag* RFID, pembaca RFID, dan sistem manajemen RFID (Ruslan, 2020).

Prinsip kerja RFID (*Radio Frequency Identification*) didasarkan pada komunikasi nirkabel antara *tag* RFID (*Radio Frequency Identification*) dan

pembaca RFID. Inisiasi komunikasi, pembaca RFID mengirimkan sinyal radio berupa permintaan baca kepada *tag* RFID (Manurung et al., 2021). Permintaan baca ini berfungsi untuk mengaktifkan *tag* RFID yang berada dalam jangkauan pembaca dan respon *tag* RFID, Setelah diaktifkan *tag* RFID merespons dengan mengirimkan informasi yang disimpan di dalamnya kembali ke pembaca. Informasi yang dikirimkan dapat berupa nomor seri, kode identifikasi unik, atau data lain yang ditanamkan di dalam *tag* (Hamdani et al., 2019).

RFID mengirimkan data dalam bentuk gelombang radio atau nirkabel (*wireless*). Dalam proses komunikasi data ini terdapat dua bagian utama, yaitu:

2.3.1 RFID Tag

Menurut (Mochamad Irvan Fadillah, 2020) Tag RFID merupakan perangkat elektronik yang menggabungkan *antena* dan *memori* menjadi satu kesatuan. Secara umum, bagian elektronik pada *tag* RFID telah dirancang dengan struktur penyimpanan data yang meliputi sel penyimpanan dan pembaca yang hanya mampu membaca (*Read Only*). Nomor seri unik merupakan informasi yang terdapat dalam *tag* RFID sejak awal diproduksi. Berdasarkan informasi tersebut, *tag* RFID dapat diklasifikasikan menjadi dua jenis kategori, yaitu:

1. Menurut (Choerudin, 2021) *Tag* Aktif adalah jenis *tag* RFID yang menurunkan kebutuhan daya pembaca RFID dengan menggunakan baterai *internal*. *Tag* dapat mengirimkan data lebih jauh dengan baterai ini. Namun demikian, kekurangan dari *tag* ini adalah biaya dan kompleksitasnya yang meningkat, yang menyebabkan ukurannya menjadi lebih besar. *Tag* RFID semakin canggih, semakin banyak fungsi yang dimilikinya.
2. Menurut (Daulay & Alamsyah, 2019) *Tag* RFID yang pasif mendapatkan energinya dari medan yang dipancarkan oleh pembaca RFID, bukan dari baterai. Kerugian utama dari *tag* ini adalah bahwa *tag* ini hanya dapat membawa data dalam jarak pendek. Selain itu, agar *tag* RFID dapat bekerja dengan benar, pembaca RFID perlu memasok lebih banyak daya. Gambar 2.1 menunjukkan gambar *tag* RFID.



Gambar 2.1 Kartu *Tag* RFID

2.3.2 RFID Reader

RFID reader adalah perangkat elektronik yang digunakan untuk membaca informasi yang disimpan pada tag RFID (*Radio Frequency Identification*). Alat ini bekerja dengan mengirimkan sinyal radio ke *tag* RFID dan menerima sinyal balasan yang berisi data identifikasi atau informasi lainnya dari *tag* tersebut. RFID reader terdiri dari antena yang memancarkan sinyal radio, modul penerima untuk menangkap sinyal yang dipantulkan kembali oleh *tag*, dan unit pemrosesan yang mengubah sinyal tersebut menjadi data yang dapat dimengerti oleh sistem komputer. RFID reader dapat bekerja pada berbagai frekuensi, termasuk *Low Frequency* (LF), *High Frequency* (HF), dan *Ultra-High Frequency* (UHF), yang masing-masing memiliki jangkauan dan aplikasi yang berbeda. Alat ini digunakan dalam berbagai aplikasi seperti manajemen inventaris, pelacakan barang dalam rantai pasokan, sistem kontrol akses, dan pembayaran tanpa kontak. Dengan kemampuannya untuk membaca banyak tag secara simultan dan dari jarak tertentu tanpa memerlukan kontak fisik, RFID *reader* sangat meningkatkan efisiensi dan akurasi dalam proses pengumpulan data dan pelacakan objek (Mamun & Hasanuzzaman, 2020).

2.4 RFID RC 522

RFID RC522 adalah modul pembaca dan penulis RFID yang beroperasi pada frekuensi 13.56 MHz, sesuai dengan standar ISO/IEC 14443A. Modul ini dirancang untuk bekerja dengan berbagai jenis tag RFID, seperti Mifare *Classic*, yang sering digunakan dalam aplikasi akses kontrol, sistem pembayaran, dan pelacakan inventaris. RC522 menggunakan antarmuka komunikasi SPI (*Serial Peripheral*

Interface) untuk berinteraksi dengan *mikrokontroler* atau komputer, seperti Arduino, memungkinkan integrasi yang mudah dalam berbagai proyek elektronik. Modul ini dapat membaca dan menulis data ke tag RFID, serta mendukung operasi enkripsi dasar untuk keamanan tambahan. Selain itu, RC522 dikenal karena konsumsi dayanya yang rendah dan harganya yang terjangkau, menjadikannya pilihan populer di kalangan bisnis dan pengembang perangkat keras. Dalam praktiknya, modul ini memungkinkan pengembangan aplikasi yang membutuhkan identifikasi dan autentikasi nirkabel dengan cepat dan efisien, memfasilitasi berbagai solusi teknologi modern (Wulandaru et al., 2017).



Gambar 2.2 RFID RC 522

2.5 Algoritma *Advanced Encryption Standard* (AES)

Algoritma *Advanced Encryption Standard* (AES) adalah sebuah algoritma kriptografi simetris yang digunakan untuk mengamankan data. Dikembangkan oleh dua kriptografer Belgia, *Vincent Rijmen* dan *Joan Daemen*, algoritma ini terpilih sebagai standar enkripsi oleh *National Institute of Standards and Technology* (NIST) Amerika Serikat pada tahun 2001, menggantikan algoritma *Data Encryption Standard* (DES). AES bekerja dengan menggunakan kunci simetris yang berarti kunci yang sama digunakan untuk enkripsi dan dekripsi data. Algoritma ini tersedia dalam tiga ukuran kunci yang berbeda: 128-bit, 192-bit, dan 256-bit, yang masing-masing mempengaruhi tingkat keamanan dan performa. Proses enkripsi dan dekripsi pada AES melibatkan beberapa tahap operasi termasuk substitusi *byte*, pergantian baris, pencampuran kolom, dan penambahan kunci ronde, yang diulang dalam beberapa putaran tergantung pada panjang kunci. AES

dikenal karena keamanan yang tinggi serta efisiensi dalam implementasi perangkat keras dan perangkat lunak, menjadikannya pilihan utama dalam berbagai aplikasi keamanan data modern, seperti komunikasi elektronik, penyimpanan data, dan jaringan

2.6 Arduino Uno

Arduino Uno adalah *mikrokontroler* berbasis ATmega328P yang populer digunakan untuk berbagai proyek elektronik dan otomasi. Cara kerjanya dimulai dengan pengunggahan program atau sketsa ke papan Arduino melalui koneksi USB menggunakan *Integrated Development Environment* (IDE) Arduino (Yulita et al., 2022). Program tersebut ditulis dalam bahasa pemrograman yang mirip dengan C/C++. Setelah program diunggah, ATmega328P menjalankan instruksi yang tertulis dalam sketsa, yang dapat mencakup membaca input dari berbagai sensor yang terhubung ke pin digital atau analog, dan mengendalikan output seperti LED, motor, atau perangkat lain. Arduino Uno memiliki 14 pin input/output digital (6 di antaranya dapat digunakan sebagai output PWM) dan 6 input analog, yang memungkinkan interaksi dengan berbagai komponen elektronik. Papan ini juga memiliki pin untuk daya 5V dan 3.3V, serta pin *ground* untuk menyediakan sumber daya ke sensor dan aktuator. Arduino Uno bekerja dengan membaca input, memproses data tersebut sesuai dengan logika program, dan menghasilkan output yang diinginkan, membuatnya sangat berguna untuk pengembangan *prototipe* dan proyek DIY yang memerlukan kontrol elektronik yang fleksibel dan mudah diimplementasikan atau mengendalikan output seperti LED, motor, atau perangkat lain sesuai dengan kebutuhan pengguna (Yusuf Nur & Asep Saepuloh, 2021).

Selama proses pengembangan, Arduino IDE menyediakan berbagai fitur seperti penyorotan sintaks, saran kode (*autocomplete*), dan fungsi pemecahan masalah (*debugging*) yang memudahkan pengguna dalam mengembangkan dan menguji kode program. IDE juga memungkinkan pengguna untuk melihat *output* dari program melalui *monitor serial*, memantau *variabel* yang digunakan dalam program, dan melakukan penyesuaian jika diperlukan. Dengan demikian, Arduino IDE menjadi alat yang penting dan efisien dalam pengembangan proyek elektronik menggunakan papan Arduino (Adfry et al., 2023).

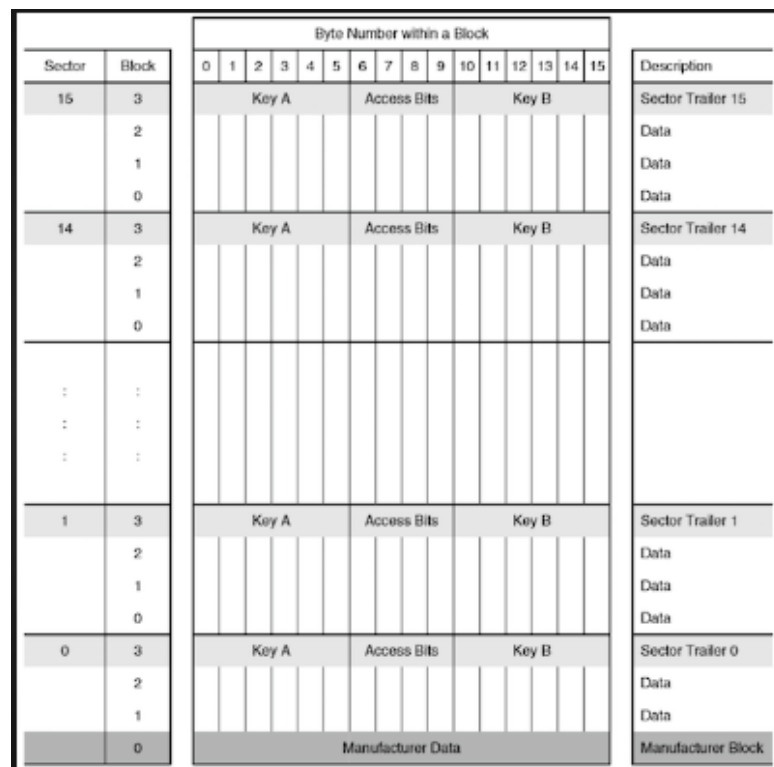
2.7 MIFARE Classic 1K

MIFARE Classic 1K adalah jenis kartu RFID (*Radio-Frequency Identification*) yang dikembangkan oleh NXP Semiconductors, terkenal karena penggunaannya yang luas dalam aplikasi tiket transportasi, kontrol akses, dan sistem pembayaran. Kartu ini memiliki kapasitas memori sebesar 1 *kilobyte* (1024 *bytes*), yang dibagi menjadi 16 sektor, masing-masing terdiri dari 4 blok dengan ukuran 16 bytes per blok. Setiap sektor dapat diakses dan dikelola secara independen dengan menggunakan kunci keamanan (Key A dan Key B) dan bit akses yang mengatur hak baca/tulis. Data kartu dienkripsi menggunakan kunci 48-bit dan disimpan di sektor-sektor pada kartu. Kombinasi dari kapasitas memori yang cukup untuk aplikasi sederhana dan fitur keamanan yang solid membuat MIFARE Classic 1K menjadi pilihan populer dalam berbagai sistem yang memerlukan identifikasi dan penyimpanan data yang aman.

2.7.1 Struktur Memory MIFARE Classic 1 K

Jika kita menghitungnya, kita dapat mengetahui seperti apa struktur memorynya :

$$16 \text{ byte (1 blok) } * 4 \text{ blok } * 16 \text{ sektor } = 1024 \text{ byte}$$



Gambar 2.3 Gambar Struktur Memori pada Kartu RFID

Gambar tersebut menunjukkan struktur memori dari kartu RFID MIFARE *Classic*. Struktur memori kartu ini dibagi menjadi beberapa sektor, dan setiap sektor dibagi lagi menjadi empat blok. Setiap blok berisi 16 *byte* data pada bagian kiri gambar, terdapat kolom "*Sector*" dan "*Block*" yang menunjukkan nomor sektor dan blok. Setiap sektor terdiri dari blok-blok bernomor 0 hingga 3. Blok 3 dari setiap sektor disebut "*Sector Trailer*" yang menyimpan informasi keamanan penting yaitu Key A, Key B, dan *Access Bits*. Key A dan Key B digunakan untuk otentikasi akses ke data dalam sektor tersebut, sementara *Access Bits* menentukan hak akses terhadap data. Blok-blok lainnya dalam setiap sektor digunakan untuk menyimpan data pengguna. Khusus untuk Sektor 0, Blok 0, berisi *Manufacturer Data* yang biasanya tetap dan tidak dapat diubah, berfungsi untuk menyimpan informasi pabrikan.

Di bagian atas gambar, ditampilkan "*Byte Number within a Block*" yang menunjukkan nomor *byte* dalam satu blok, mulai dari 0 hingga 15. Untuk *Sector Trailer*, *byte-byte* ini terbagi untuk Key A (*byte* 0-5), *Access Bits* (*byte* 6-8), dan Key B (*byte* 10-15), dengan *byte* 9 yang tidak digunakan (kosong). Deskripsi di sebelah kanan memberikan penjelasan singkat tentang isi dari setiap blok dalam sektor. Untuk sektor terakhir (sektor 15), blok terakhir (blok 3) adalah *Sector Trailer* yang mengandung Key A, *Access Bits*, dan Key B, sementara blok-blok sebelumnya (blok 0, 1, 2) digunakan untuk data. Pola ini berulang untuk setiap sektor hingga sektor 0.

2.8 Node Js

Node.js adalah *platform runtime* yang memungkinkan eksekusi *JavaScript* di sisi server, dibangun di atas mesin *JavaScript V8* milik *Google Chrome*. Dikembangkan oleh *Ryan Dahl* pada tahun 2009, *Node.js* dirancang untuk membangun aplikasi jaringan yang cepat dan skalabel dengan memanfaatkan arsitektur *non-blocking I/O* yang berbasis *event-driven*. Ini berarti *Node.js* dapat menangani banyak koneksi simultan dengan efisiensi tinggi tanpa menunggu operasi I/O selesai, membuatnya ideal untuk aplikasi *real-time* seperti chat, game online, dan *server* web yang membutuhkan performa tinggi. *Node.js* juga menyertakan *npm (Node Package Manager)*, yang merupakan ekosistem pustaka

terbesar untuk *JavaScript*, memudahkan pengembang untuk mengelola dan berbagi kode. Dengan kemampuannya yang *cross-platform*, *Node.js* dapat dijalankan pada berbagai sistem operasi, memberikan fleksibilitas dalam pengembangan dan *deployment* aplikasi. Dengan keunggulan dalam skalabilitas, kecepatan, dan ekosistem yang luas, *Node.js* telah menjadi pilihan populer untuk pengembangan backend dan aplikasi jaringan modern.

2.9 Express Js

Express.js adalah kerangka kerja (*framework*) web aplikasi untuk *Node.js* yang dirancang untuk membangun aplikasi web dan API dengan cepat dan mudah. Dikembangkan oleh TJ Holowaychuk dan pertama kali dirilis pada tahun 2010, *Express.js* menyediakan serangkaian fitur minimalis dan fleksibel untuk pengembangan *server-side*, seperti *routing* yang kuat, *middleware* yang dapat dihubungkan, dan penanganan permintaan HTTP yang efisien. Dengan arsitektur yang ringan dan tidak mengikat pengembang pada pola tertentu, *Express.js* memungkinkan pengembang untuk membangun aplikasi web modular dan terukur. Salah satu keunggulan utama *Express.js* adalah ekosistemnya yang luas, dengan banyak *middleware* dan modul yang tersedia untuk menambah fungsionalitas, seperti autentikasi, logging, dan validasi data. *Express.js* juga menjadi fondasi untuk banyak kerangka kerja populer lainnya, seperti *Sails.js* dan *Kraken.js*, dan sering digunakan bersama dengan pustaka *front-end* seperti *React*, *Angular*, dan *Vue.js* untuk membangun aplikasi *full-stack*. Dengan dokumentasi yang baik dan komunitas yang aktif, *Express.js* tetap menjadi pilihan utama bagi banyak pengembang yang bekerja dengan *Node.js* untuk membangun aplikasi web yang cepat, efisien, dan dapat diandalkan (Linardi et al., 2023).

2.10 Kabel Jumper

Kabel *jumper* adalah kabel listrik kecil yang sering digunakan dalam elektronik untuk membuat koneksi sementara antara dua titik dalam suatu rangkaian. Kabel ini biasanya memiliki ujung yang dilengkapi dengan konektor atau pin, yang dapat dengan mudah disisipkan ke dalam header atau soket pada papan sirkuit, seperti *breadboard*. Tersedia dalam berbagai jenis konektor seperti *male-to-male*, *female-to-female*, dan *male-to-female* kabel *jumper* memberikan *fleksibilitas* dalam menghubungkan komponen elektronik tanpa perlu menyolder. Mereka datang

dalam berbagai panjang dan warna, membantu dalam pengelolaan dan identifikasi jalur koneksi pada rangkaian yang kompleks. Penggunaannya sangat populer dalam pendidikan dan hobi elektronik, terutama untuk eksperimen dan pengembangan proyek-proyek kecil menggunakan platform seperti Arduino dan *Raspberry Pi*. Dengan demikian, kabel *jumper* memainkan peran penting dalam fase *prototipe*, memungkinkan perakitan dan pengujian rangkaian secara mudah dan efisien sebelum membuat koneksi yang lebih permanen (Radice, 2022).



Gambar 2.4 Kabel *Jumper*

2.11 Database

Database adalah kumpulan data yang terorganisir dan disimpan secara elektronik untuk memudahkan pengelolaan, akses, dan manipulasi informasi. Dalam database, data disusun dalam tabel yang terdiri dari baris (rekaman) dan kolom (atribut), dimana setiap tabel menyimpan informasi tentang entitas tertentu seperti pelanggan, produk, atau transaksi. Tabel-tabel ini dapat dihubungkan melalui kunci utama (*primary key*) dan kunci asing (*foreign key*) untuk menjaga integritas data dan mengurangi redundansi. Pengelolaan database dilakukan oleh sistem manajemen basis data (DBMS) seperti *MySQL*, *PostgreSQL*, dan *Oracle Database*, yang menyediakan berbagai fitur untuk membuat, mengelola, dan mengakses data. Pengguna dapat mengakses dan memanipulasi data menggunakan bahasa query, dengan SQL (*Structured Query Language*) sebagai yang paling umum digunakan. Proses normalisasi diterapkan untuk mengatur data secara efisien dan memastikan integritasnya, sementara fitur backup dan *recovery* digunakan untuk melindungi data dari kehilangan atau kerusakan. Dalam berbagai bidang seperti perbankan, pendidikan, dan bisnis, database menjadi alat penting

untuk menyimpan dan mengelola informasi secara efisien dan terstruktur (Fadhliil Khaliq, 2021).

BAB 3

ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Analisis sistem merupakan tahapan pada penelitian yang berfokus pada merinci komponen-komponen yang diperlukan agar suatu sistem berfungsi. Analisis sistem dibagi menjadi dua kategori, yaitu analisis masalah dan analisis kebutuhan. Analisis masalah digunakan untuk mengidentifikasi sumber serta dampak dari suatu masalah, sedangkan analisis kebutuhan berperan dalam menentukan data dan proses yang diperlukan dalam mendesain suatu rangkaian.

3.1.1 Analisis Masalah

Penelitian tentang pengamanan kartu *tag* RFID untuk peminjaman buku perpustakaan menggunakan algoritma AES mengidentifikasi beberapa masalah kritis yang perlu dianalisis. Pertama, keamanan kartu yang rentan terhadap serangan seperti duplikat dan manipulasi. Implementasi algoritma AES bertujuan untuk mengenkripsi data, sehingga meningkatkan tingkat keamanan dan mencegah akses tidak sah. Namun, tantangan teknis dalam mengintegrasikan AES ke dalam sistem RFID perlu dieksplorasi, termasuk kompatibilitas perangkat keras dan efisiensi proses enkripsi-dekripsi yang mungkin mempengaruhi kinerja sistem secara keseluruhan.

Selain itu, faktor-faktor seperti kecepatan pemrosesan dan konsumsi daya juga menjadi pertimbangan penting, mengingat RFID sering digunakan dalam lingkungan dengan sumber daya terbatas. Penelitian ini juga harus mempertimbangkan aspek kemudahan penggunaan dan pengalaman pengguna, memastikan bahwa solusi keamanan yang diterapkan tidak menghambat kelancaran proses peminjaman buku. Dengan demikian, analisis mendalam diperlukan untuk mengoptimalkan keseimbangan antara keamanan, kinerja, dan kenyamanan pengguna dalam sistem perpustakaan berbasis RFID.

3.1.2 Analisis Kebutuhan

Analisis kebutuhan menjelaskan tahapan dalam prosedur pemahaman dan pengenalan yang dihasilkan dari keperluan yang dibutuhkan sistem untuk

mencapai tujuannya. Analisis kebutuhan dibagi menjadi dua bagian utama, yaitu fungsional dan non-fungsional.

3.1.2.1 Kebutuhan Fungsional

Analisis kebutuhan fungsional melibatkan penjelasan mengenai prosedur yang harus dijalankan oleh sistem untuk memenuhi persyaratan. Persyaratan fungsional yang diperlukan dalam sistem ini mencakup:

1. Sistem harus dapat mengenkripsi UID yang dimasukkan ke memori kartu *tag* RFID menggunakan algoritma AES untuk melindungi informasi sensitif.
2. Sistem harus memiliki mekanisme yang aman untuk menghasilkan, mendistribusikan, dan menyimpan kunci enkripsi AES yang digunakan dalam proses enkripsi dan dekripsi.
3. Sistem harus memastikan bahwa komunikasi antara kartu RFID dan pembaca RFID terenkripsi dan aman dari serangan manipulasi dan duplikat.

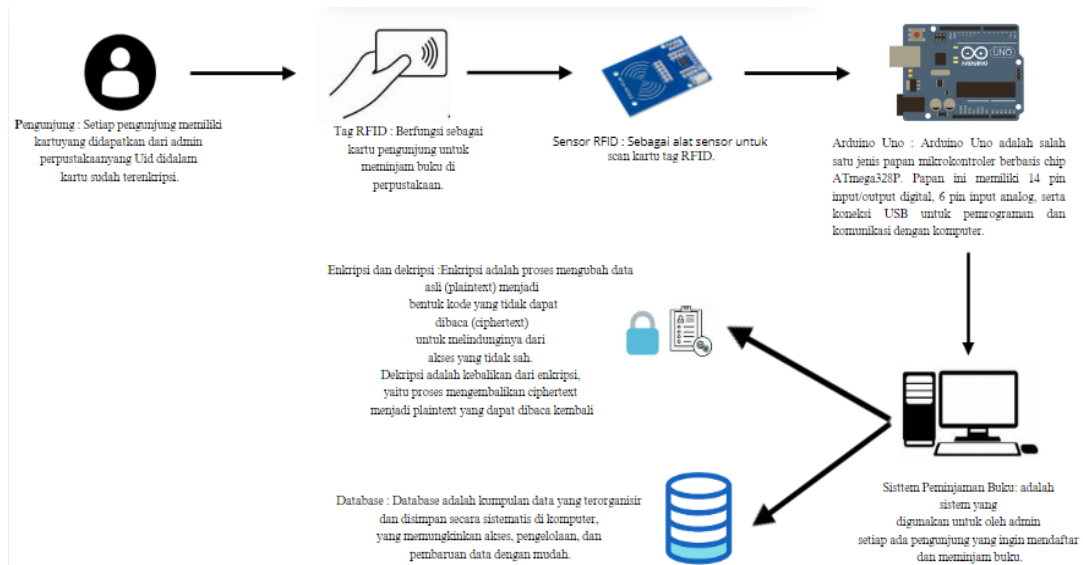
3.1.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional mengacu pada fitur, karakteristik, atau batasan yang terkait dengan fungsi atau pelayanan yang disediakan oleh sistem. Berikut adalah persyaratan non-fungsional yang diperlukan untuk sistem ini:

1. Sistem harus memenuhi standar keamanan yang sedang untuk melindungi data sensitif pengguna dan informasi buku dari akses tidak sah dan manipulasi.
2. Sistem harus berkinerja tinggi dan responsif, memungkinkan proses enkripsi-dekripsi data pada kartu *tag* RFID tanpa mengalami penundaan yang signifikan.

3.2 Arsitektur Umum

Arsitektur umum adalah struktur suatu sistem dirancang dan bagaimana komponen-komponen didalamnya saling berinteraksi untuk mencapai tujuan. Berikut adalah desain arsitektur umum dari sistem.



Gambar 3.1 Arsitektur Umum

Gambar di atas menggambarkan alur kerja sistem peminjaman buku menggunakan teknologi RFID yang terintegrasi dengan Arduino Uno. Dimulai dari pengunjung yang memiliki kartu RFID yang telah terdaftar oleh admin perpustakaan. Kartu RFID berfungsi sebagai identifikasi pengunjung untuk meminjam buku. Ketika kartu di-scan oleh sensor RFID, informasi dari kartu dibaca dan diteruskan ke Arduino Uno, yang bertugas memproses data tersebut. Arduino Uno kemudian menghubungkan data dengan sistem peminjaman buku di komputer, di mana data akan dienkripsi untuk keamanan. Sistem ini memverifikasi pengguna dan mencatat buku yang dipinjam ke dalam database perpustakaan, yang mengorganisir dan menyimpan semua data transaksi. Proses ini mencakup penggunaan enkripsi untuk menjaga kerahasiaan data dan dekripsi untuk mengembalikan data ke bentuk aslinya saat diperlukan, sehingga memastikan sistem aman dan efisien dalam mengelola peminjaman buku di perpustakaan.

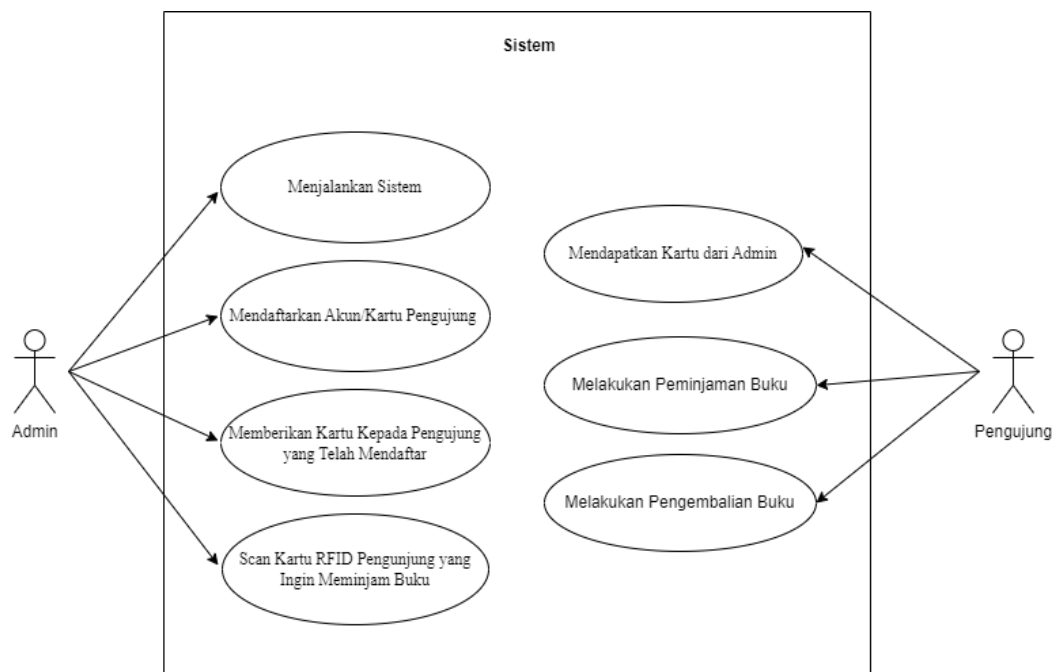
3.3 Pemodelan Sistem

Pemodelan sistem melibatkan penjelasan tahapan interaksi antara pengguna dengan aplikasi yang dibangun untuk memastikan sistem beroperasi secara optimal. Biasanya, pemodelan sistem digambarkan menggunakan UML (*Unified Modeling Language*), bahasa pemodelan yang umum digunakan untuk menggambarkan hubungan antar komponen dalam sistem agar dapat berinteraksi melalui pengguna.

Dalam penelitian ini, kerangka UML yang dimanfaatkan terdiri dari *Use Case Diagram* dan *Activity Diagram*.

3.3.1 Use Case Diagram

Use case diagram adalah cara pemodelan yang digunakan untuk memaparkan hubungan antara pengguna dan sistem yang telah dibuat.



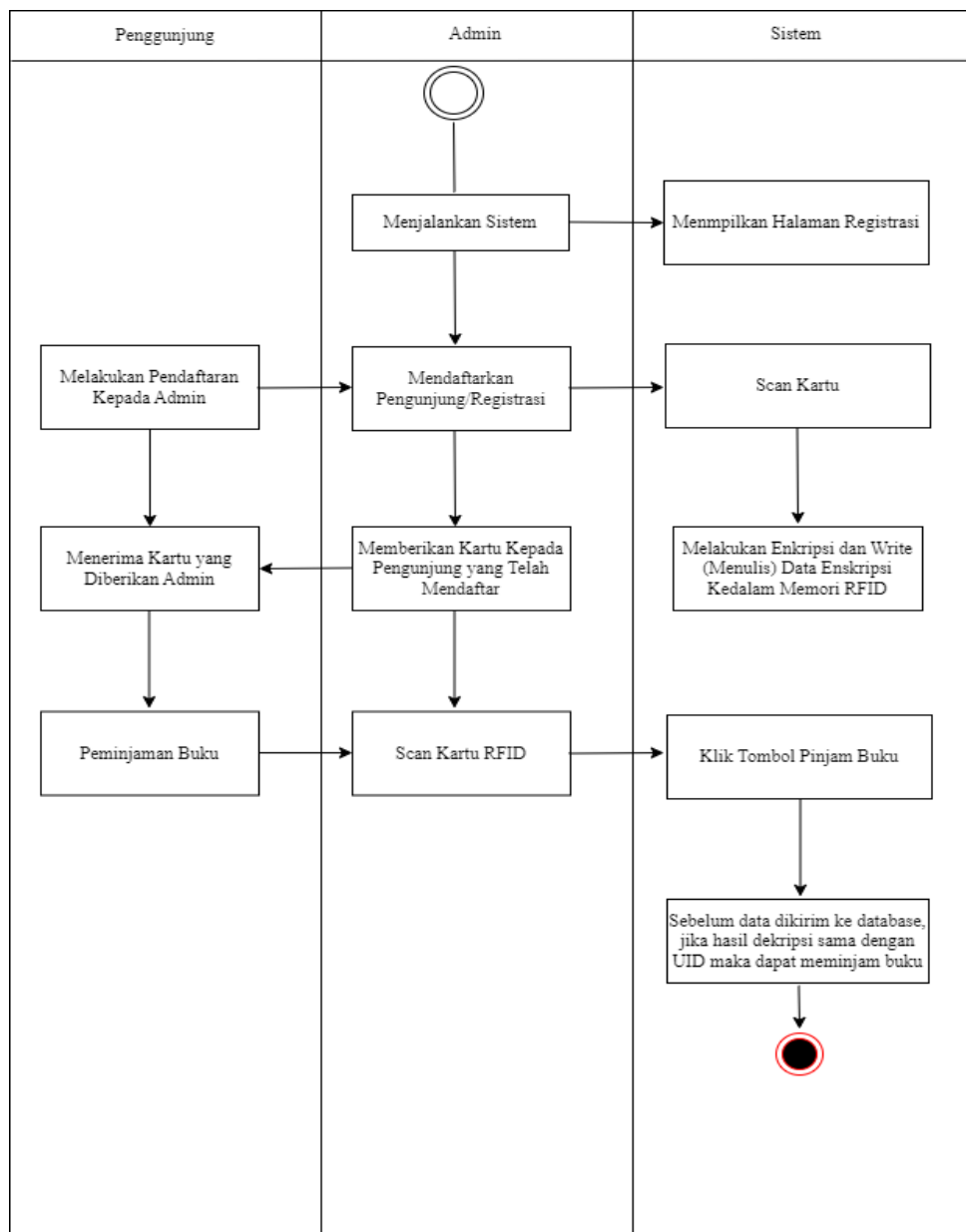
Gambar 3.2 *Use Case Diagram*

Gambar diatas menggambarkan alur interaksi antara aktor (pengunjung dan admin) dengan sistem peminjaman buku berbasis RFID. Di sisi kiri, terdapat aktor admin yang bertanggung jawab untuk menjalankan sistem, mendaftarkan akun kartu pengunjung, dan memberikan kartu kepada pengunjung yang sudah terdaftar. Admin juga bertugas mempersiapkan sistem agar siap digunakan oleh pengunjung.

Di sisi kanan, aktor pengunjung dapat melakukan berbagai aktivitas seperti mendapatkan kartu RFID dari admin, kemudian menggunakan kartu tersebut untuk meminjam atau mengembalikan buku. Pengunjung akan *scan* kartu RFID mereka untuk melakukan peminjaman buku dan juga menggunakannya saat melakukan pengembalian buku.

3.3.2 Activity Diagram

Activity Diagram merupakan alat visual yang dipakai untuk mendeskripsikan alur kerja dan aktivitas dalam proses atau sistem. Ini mencatat berbagai langkah dan kegiatan yang terjadi dari awal hingga akhir suatu proses atau aktivitas. Diagram ini memperlihatkan urutan langkah-langkah, keputusan, garis waktu, dan aliran kontrol dalam proses tersebut. Dengan demikian, *activity diagram* membantu dalam memahami secara visual bagaimana suatu sistem atau proses berjalan dan bagaimana berbagai elemen terkait berinteraksi satu sama lain.

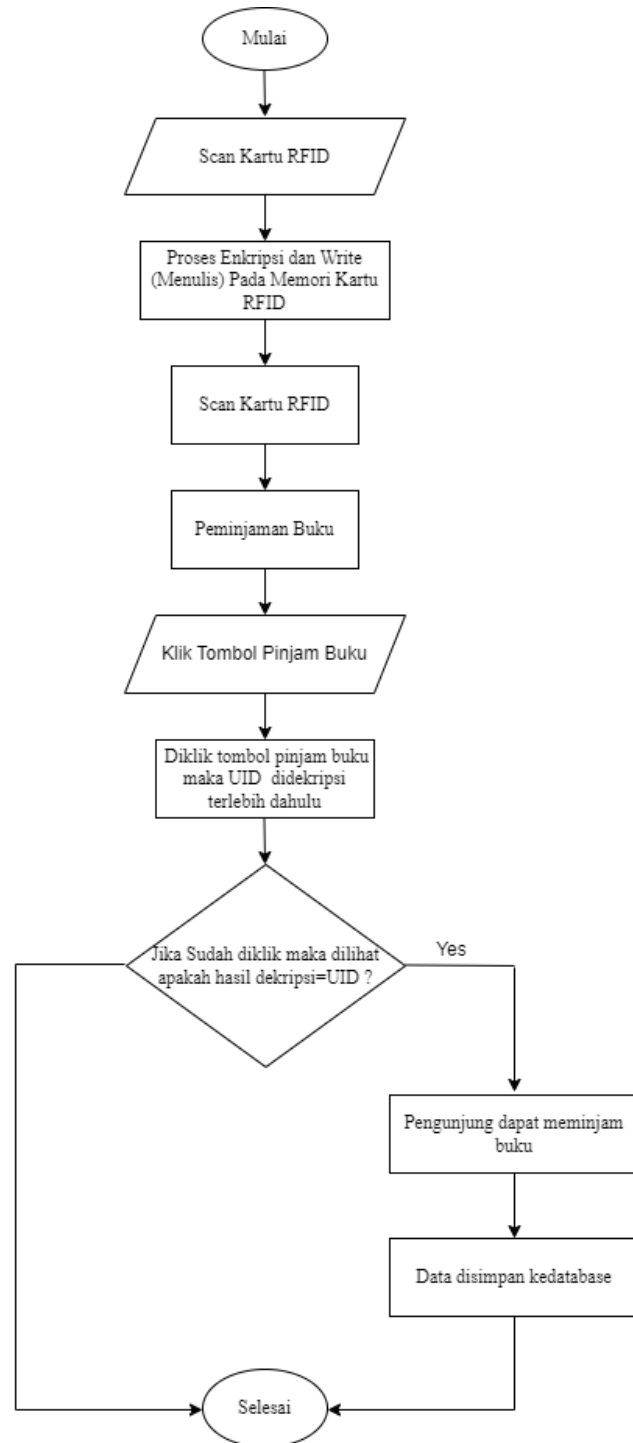


Gambar 3.3 Activity Diagram

Gambar tersebut menggambarkan alur proses interaksi antara pengguna (*User*) dan *administrator* (Admin) dengan sistem dalam konteks peminjaman buku menggunakan teknologi RFID. Untuk pengguna, proses dimulai dengan memindai kartu RFID. Sistem kemudian mengenkripsi data dan menuliskannya ke memori RFID. Setelah itu, pengguna memasukkan data peminjaman buku, dan sistem melakukan pengecekan data serta menampilkan hasil pinjaman. Untuk *administrator*, proses dimulai dengan menjalankan sistem, yang kemudian menampilkan tampilan awal sistem. Setiap proses ini diilustrasikan dengan diagram alur yang menunjukkan langkah-langkah yang dilakukan oleh pengguna dan *administrator* beserta respon dari sistem.

3.4 Flowchart

Flowchart adalah diagram yang menggunakan simbol-simbol grafis untuk menjelaskan tahapan dalam suatu proses atau alur kerja. Simbol-simbol tersebut dihubungkan dengan panah untuk menunjukkan urutan dan arah aliran. *Flowchart* membantu memvisualisasikan proses secara sederhana, memudahkan pemahaman, dan digunakan untuk menganalisis atau mendesain sistem.



Gambar 3.4 *Flowchart*

Gambar ini adalah sebuah diagram alir yang menjelaskan alur proses peminjaman buku menggunakan teknologi kartu RFID. Proses dimulai dengan pengguna yang memindai kartu RFID mereka. Setelah kartu RFID dipindai, sistem akan melakukan proses enkripsi dan menulis data ke dalam memori kartu RFID.

Kartu RFID kemudian dipindai lagi untuk verifikasi. Setelah verifikasi, proses peminjaman buku dapat dilanjutkan. Pengguna kemudian mengklik tombol untuk meminjam buku. Ketika tombol ini diklik, UID (Unique Identifier) dari kartu RFID akan didekripsi untuk memastikan keaslian data. Selanjutnya, sistem akan memeriksa apakah hasil dekripsi UID tersebut valid. Jika hasil dekripsi sesuai dengan UID yang terdaftar, pengunjung diperbolehkan meminjam buku. Setelah itu, data peminjaman akan disimpan ke dalam database. Proses ini kemudian diakhiri dengan tahap "Selesai", yang menandakan bahwa peminjaman buku berhasil dilakukan.

3.5 Perancangan *Interface* Sistem

Perancangan *interface* adalah proses merancang tampilan dan interaksi pengguna dalam sebuah aplikasi, situs web, atau perangkat lunak. Ini melibatkan pemikiran tentang bagaimana pengguna akan berinteraksi dengan sistem, termasuk tata letak elemen-elemen antarmuka, pengaturan warna dan desain, serta *navigasi interface*. *Interface* harus dirancang sesuai kebutuhan dan preferensi pengguna untuk mencapai tujuan yang diinginkan.

3.5.1 Halaman *Registration*

The image shows a registration form titled "Registration". It contains the following elements from top to bottom:

- A text input field labeled "NIM".
- A text input field labeled "NAMA".
- A text label "Scan RFID Terlebih Dahulu".
- A button labeled "Aktivasi".
- A text input field labeled "Alamat".
- A text input field labeled "No Hp".
- A text input field labeled "Create Password".
- A text input field labeled "Confirm Password".
- A button labeled "Tambah Akun".

Gambar 3.5 Desain Halaman *Registration*

Halaman *registration* diatas merupakan halaman yang mana pengunjung mengisi form data diri. Dibagian bacaan “ Scan RFID terlebih dahulu” merupakan proses untuk membaca UID dari kartu pengunjung, setelah UID terbaca maka pengunjung mengklik tombol “Aktivasi”. Tombol aktivasi ini merupakan tombol untuk memproses hasil enkripsi yang mana maka hasil enkripsi akan muncul di teminal aplikasi *visual studio code*. Ketika setelah diklik tombol aktivasi maka sistem akan memerintahkan untuk men-scan ulang kartu RFID yang apabila discan kembali maka sistem akan menulis (*write*) hasil enkripsi kedalam memori kartu RFID. Selanjutnya pengunjung mengisi *password* yang diinginkan agar tidak dapat diketahui oleh orang lain. Setelah mengisi form selesai maka pengunjung mengklik tombol tambah akun, maka akan terkirim ke database, maka data-data tersebut akan tersimpan di database.

3.5.2 Halaman Peminjaman Buku

Pinjam Buku

NIM

NAMA

Scan Kartu

Alamat

No Hp

Password

Buku Tambah Buku

dd/mm/yyyy

dd/mm/yyyy

Pinjam Buku

Gambar 3.6 Desain Halaman Pinjaman Buku

Pada gambar 3.6 merupakan desain halaman peminjaman buku, yang mana pada halaman ini merupakan proses untuk pengunjung melakukan peminjaman buku. Pada halaman ini pengunjung juga diperintahkan untuk mengisi data diri pada form peminjaman ini. Selanjutnya pada tabel buku ada pilihan untuk tambah buku, jika ada pengunjung yang melakukan peminjaman buku lebih dari satu maka dapat mengklik tombol tambah buku. Setelah pengunjung telah mengisi form peminjaman maka akan diperintahkan untuk klik tombol pinjam buku.



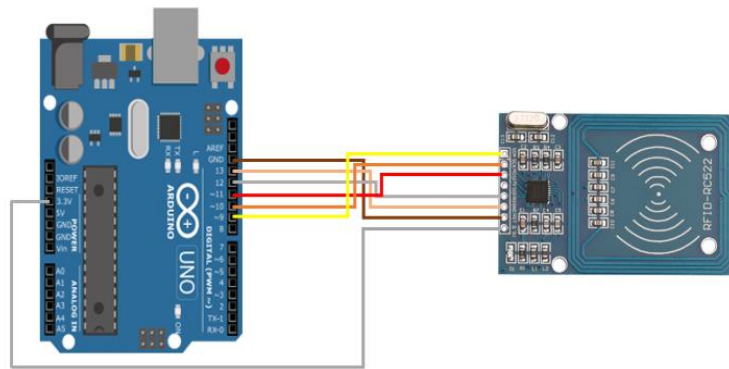
The image shows a web form titled "Tambah Buku". It contains two input fields: "ID Buku" and "Nama Buku". Below these fields is a button labeled "Tambah Buku".

Gambar 3.7 Desain Tampilan Tambah Buku

Pada gambar diatas merupakan desain tampilan tambah buku yang mana apabila pengunjung ingin melakukan peminjaman buku lebih dari satu maka pengunjung dapat masuk kehalaman ini. Pada halaman ini pengunjung diperintahkan untuk mengisi id buku dan nama buku.

3.6 Perancangan Alat

Dibawah ini merupakan perancangan keseluruhan alat yang di implementasikan pada sistem.



Gambar 3.8 Gambar Rangkaian

Berikut adalah menunjukkan koneksi antara Arduino Uno dan RFID RC-522 :

Tabel 3.1 Koneksi RFID RC 522 dengan Arduino Uno

RFID RC 522	Arduino Uno
VCC	3.3V
GND	GND
RST	9
MISO	12
MOSI	11
SCK	13
SDA (SS)	10

Berikut penjelasan hubungan pin antara RFID RC 522 dengan Arduino Uno:

1. VCC (RFID RC 522) ke 3.3V (Arduino Uno), Modul RFID RC522 bekerja pada tegangan 3.3V.
2. GND (RFID RC 522) ke GND (Arduino Uno), *Ground* bersama dua prangkat.
3. RST (RFID RC 522) ke 9 (Arduino Uno), Pin ini digunakan untuk me-*reset* modul RFID.
4. MISO (RFID RC 522) ke 12 (Arduino Uno), Pin ini digunakan untuk mentransfer data dari modul RFID ke Arduino (*Master In Slave Out*).

5. MOSI (RFID RC 522) ke 11 (Arduino Uno), Pin ini digunakan untuk mentransfer data dari Arduino ke modul RFID (*Master Out Slave In*).
6. SCK (RFID RC 522) ke 13 (Arduino Uno), Pin ini digunakan sebagai clock untuk sinkronisasi data antara Arduino dan modul RFID.
7. SDA (RFID RC 522) ke 10 (Arduino Uno), Pin ini digunakan untuk memilih modul RFID ketika ada beberapa perangkat SPI yang terhubung.

BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Studi ini akan meneliti rancangan pengamanan kartu tag RFID untuk peminjaman buku perpustakaan dan alat-alat diperlukan yang dalam membangun algoritma yang diperlukan untuk memastikan sistem yang dibuat telah bekerja sesuai dengan rencana.

Berikut adalah alat yang digunakan dalam perncangan penelitian pengamanan kartu *tag* RFID untuk peminjaman buku perpustakaan, adalah sebagai berikut:

4.1.1 Perangkat Keras (*Hardware*)

Spesifikasi perangkat keras yang digunakan dalam penelitian ini dapat dilihat pada tabel 4.1 berikut:

Tabel 4.1 Perangkat Keras (*Hardware*)

No	Alat yang Digunakan
1.	Processor Intel(R) Core(TM) i5-3340M CPU @ 2.70GHz 2.70 GHz
2.	Memory 4,00 GB (3,90 GB usable)
3.	SSD dengan kapasitas 500 GB
4.	RFID RC 522
5.	Arduino Uno
6.	Kabel <i>Jumper</i>
7.	Kartu MIFARE 13.56 Mhz 1K

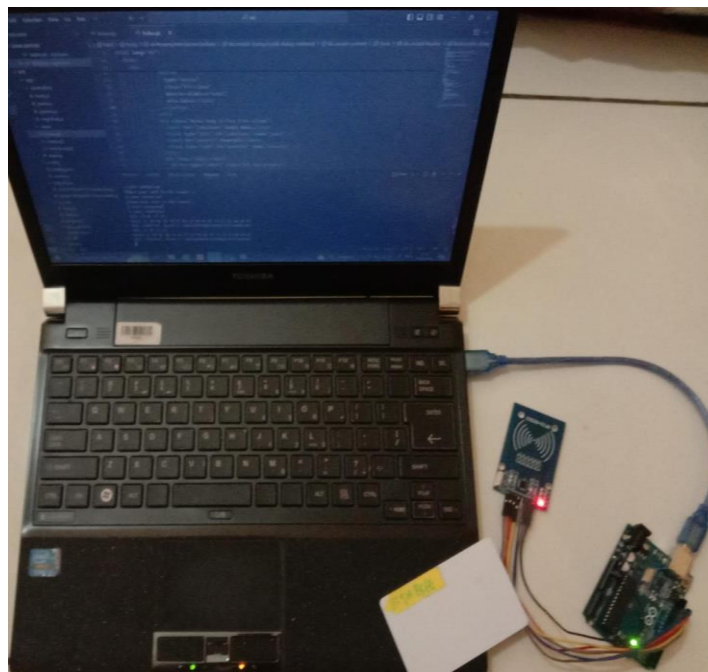
4.1.2 Perangkat Lunak (*Software*)

Macam-macam perangkat lunak yang dibutuhkan dalam pembuatan sistem dapat dilihat pada tabel 4.2 berikut:

Tabel 4.2 Perangkat Lunak (*Software*)

No	Alat yang Digunakan
1.	Arduino IDE
2.	Node Js
3.	Express Js
4.	XAMPP

4.1.3 Desain Alat

**Gambar 4.1** Desain Alat Sistem

Pada gambar 4.1 merupakan simulasi dari rangkaian pengamanan kartu tag RFID (*Radio Frequency Identification*). Didalam rangkaian pengamanan kartu tag RFID untuk peminjaman buku perpustakaan yang dibangun berisikan modul dan sensor, yaitu Arduino Uno dan RFID RC-522, kartu RFID, dan kabel *jumper*.

4.2 Proses Pada Program

Berikut adalah beberapa proses yang terdapat didalam program yang membangun sistem ini, sebagai berikut:

4.2.1 Mengimport Module dan *Library*

```
/**Mengimpor Modul dan Library: */
const { SerialPort } = require("serialport");
const { ReadlineParser } = require("@serialport/parser-readline");
const session = require("express-session");
const flash = require("connect-flash");
const path = require("path");
const router = require("../config/routes");
const bodyParser = require("body-parser");
```

Gambar 4.2 Mengimport Modul dan *Library*

Pada gambar 4.1 diatas merupakan proses mengimpor modul dan *library* proses ini memungkinkan programmer untuk mengakses fungsi, kelas, dan objek yang sudah ada dan dapat digunakan untuk menyelesaikan berbagai tugas tanpa harus menulis kode dari awal. Seperti *SerialPort* dan *ReadlineParser* yang diimpor dari *library* 'serialport' untuk berkomunikasi dengan *port serial* dan memarsing data dari *port serial*. *Session* diimpor dari 'express-session' untuk mengelola sesi pengguna. *Flash* diimpor dari 'connect-flash' untuk pesan *flash* (seperti pesan kesalahan atau sukses) di aplikasi web. *Path* adalah modul bawaan Node.js yang digunakan untuk bekerja dengan *path* file dan direktori. Router diimpor dari ../config/routes yang mungkin berisi rute-rute untuk aplikasi web. *BodyParser* diimpor dari *body-parser* untuk mem-parsing *body* dari *request* HTTP.

4.2.2 Mengimport dan Mengonfigurasi Enkripsi dan Dekripsi AES

```
/**Mengimpor dan Mengonfigurasi Enkripsi AES */
const { Aes128EcbEncrypt, Aes128EcbDecrypt } = require("crypto-aes-ecb");
const key = "secretkey16bytes";
```

Gambar 4.3 Mengimpor dan Mengonfigurasi Enkripsi AES

Pada gambar 4.2 merupakan proses mengimpor dan mengonfigurasi enkripsi AES, Dalam konteks *JavaScript* atau Node.js, untuk mengimplementasikan enkripsi dan dekripsi AES menggunakan mode ECB dengan menggunakan *library crypto-aes-ecb*, mengimpor fungsi *Aes128EcbEncrypt* dan *Aes128EcbDecrypt* dari *library crypto-aes-ecb* menggunakan sintaks `const {Aes128EcbEncrypt, Aes128EcbDecrypt} = require("crypto-aes-ecb");`. Modul ini menyediakan fungsi-fungsi yang diperlukan untuk melakukan enkripsi dan dekripsi dengan AES dalam mode ECB. Selanjutnya, mendefinisikan kunci enkripsi AES. Dalam contoh ini, kunci

didefinisikan sebagai string `"secretkey16bytes"`. Penting untuk dicatat bahwa untuk menggunakan AES-128, kunci harus tepat 16 *byte* panjangnya. Setelah mengimpor dan mendefinisikan kunci, Anda dapat menggunakan fungsi `Aes128EcbEncrypt` untuk mengenkripsi data dengan AES ECB menggunakan kunci yang telah ditentukan. Begitu juga dengan fungsi `Aes128EcbDecrypt` untuk mendekripsi data yang telah dienkripsi sebelumnya menggunakan kunci yang sama.

4.2.3 Membuat *Instance Express*

```
const express = require("express"); //Memuat modul Express
const app = express(); //Membuat instance aplikasi Express
```

Gambar 4.4 Membuat *Instance Express*

Express adalah sebuah *framework* web yang populer untuk Node.js, yang mempermudah pembuatan dan pengelolaan aplikasi web. Untuk memulai menggunakan *Express*, langkah pertama yang perlu dilakukan adalah membuat sebuah *instance* dari aplikasi *Express*. *Instance* ini sering disimpan dalam sebuah variabel bernama *app*. Dengan menggunakan *instance app*, kita dapat mendefinisikan berbagai rute, middleware, dan konfigurasi lainnya untuk aplikasi web kita.

4.2.4 Membuat HTTP Server

```
const http = require("http");
const server = http.createServer(app);
```

Gambar 4.5 Membuat HTTP Server

Gambar 4.4 ini merupakan Membuat HTTP server adalah langkah dasar dalam pengembangan aplikasi web, di mana *server* ini bertanggung jawab untuk menangani permintaan (*request*) dari *klien* dan mengirimkan tanggapan (*response*) kembali ke *klien*. Dalam konteks Node.js, HTTP server dapat dibuat dengan menggunakan modul bawaan bernama *http*. Proses ini dimulai dengan memuat modul *http* dan menggunakan fungsi `'createServer'` untuk membuat *instance server*. Fungsi ini menerima *callback* yang berisi dua parameter, yaitu request dan response. Parameter request berisi informasi tentang permintaan yang

diterima dari klien, seperti URL dan metode HTTP yang digunakan, sementara *parameter response* digunakan untuk mengirimkan tanggapan kembali ke *klien*. Setelah *server* dibuat, kita harus menetapkan *port* di mana *server* akan mendengarkan permintaan masuk menggunakan metode *listen*.

4.2.5 Mengimpor dan Mengkonfigurasi *Socket.io*

```
io.on("connection", (socket) => {
  console.log("a user connected");
  socket.on("disconnect", () => {
    console.log("user disconnected");
  });
});

const publicDir = path.join(__dirname, "../public");
const viewsDir = path.join(__dirname, "../views");
```

Gambar 4.6 Mengimpor dan Mengkonfigurasi *Socket.io*

Pada gambar diatas Kode ini mengimpor dan mengkonfigurasi *Socket.io* untuk menangani koneksi *real-time* antara *server* dan *klien*. Pertama, modul *Server* diimpor dari *socket.io* dan *instance io* dibuat dengan menghubungkannya ke HTTP *server (server)*. *Event listener* dipasang pada *instance io* untuk menangani *event connection* dan *disconnect*. Ketika klien terhubung, pesan "*a user connected*" akan dicetak di konsol, dan ketika klien terputus, pesan "*user disconnected*" akan dicetak.

4.2.6 Mengatur Direktori dan *View Engine*

```
/** Install View Engine */
app.set("views", viewsDir);
app.set("view engine", "ejs");

// Setup express-session
app.use(
  session({
    secret: "your-secret-key",
    resave: false,
    saveUninitialized: true,
  })
);
```

Gambar 4.7 Mengatur *View Engine*

Pada gambar diatas menjelaskan pengaturan penting terkait direktori publik dan *view engine* untuk aplikasi *Express*. Pertama, direktori publik diatur menggunakan *express.static()*, yang menetapkan *folder public* sebagai tempat untuk menyimpan file statis seperti gambar, CSS, dan *JavaScript*. Direktori ini diakses melalui *path* yang ditentukan oleh *path.join(dirname, "/public")*. Selanjutnya, direktori tampilan (*views*) diatur dengan *app.set("views", viewsDir)*,

yang menunjukkan folder *views* sebagai tempat menyimpan template tampilan aplikasi, ditentukan oleh `path.join(dirname, "./views")`. View engine yang digunakan adalah *ejs* (*Embedded JavaScript*), diatur melalui `app.set("view engine", "ejs")`, memungkinkan penggunaan template HTML dengan logika *JavaScript* yang disematkan di dalamnya. Kombinasi dari pengaturan ini memastikan bahwa aplikasi dapat melayani file statis dengan benar dan *render* halaman dinamis menggunakan template EJS.

4.2.7 Mengatur Sesi dan *Flash Message*

```
// Setup connect-flash
app.use(flash());
// Middleware untuk flash message
app.use((req, res, next) => {
  res.locals.successMessage = req.flash("successMessage");
  next();
});
```

Gambar 4.8 Mengatur Sesi dan *Flash Message*

Gambar diatas menjelaskan untuk mengatur penggunaan *connect-flash* untuk menangani pesan *flash* dalam aplikasi *Express*. Pertama, *middleware connect-flash* diinisialisasi dengan `app.use(flash());`, memungkinkan penyimpanan pesan sementara yang dapat bertahan hingga *request* berikutnya. Pesan *flash* ini sering digunakan untuk memberikan umpan balik kepada pengguna, seperti pesan keberhasilan atau kesalahan. Selanjutnya, *middleware* khusus ditambahkan untuk menangani pesan *flash* tersebut. Dalam *middleware* ini, pesan *flash* dengan kunci "*successMessage*" diambil dari sesi menggunakan `req.flash("successMessage")` dan disimpan ke dalam `res.locals.successMessage`. Dengan menyimpan pesan ini di `res.locals`, pesan tersebut menjadi tersedia untuk template *view* yang *render*, sehingga dapat ditampilkan kepada pengguna. Fungsi `next()` dipanggil untuk memastikan *request* diproses lebih lanjut oleh *middleware* berikutnya. Kombinasi dari pengaturan ini memungkinkan aplikasi untuk menyimpan dan menampilkan pesan sementara dengan mudah, meningkatkan pengalaman pengguna dengan memberikan umpan balik yang relevan.

4.2.8 Koneksi Serial

```
const port = new SerialPort({
  path: "COM9",
  baudRate: 9600,
});

const parser = port.pipe(new ReadlineParser({ delimiter: "\n" }));

let uid = null;
let block1 = null;

parser.on("data", (line) => {
  console.log(line);
  line = line.trim();

  if (line.startsWith("UID:")) {
    let uidLine = line.slice(4).trim();
    let uidHex = uidLine.replace(/\\s/g, "");
    uid = uidHex;
  } else if (line.startsWith("Block 1:")) {
    let block1Line = line.slice(8).trim();
    let block1Hex = block1Line.replace(/\\s/g, "");
    block1 = block1Hex;
  }

  if (uid && block1) {
    console.log(`UID: ${uid}, Block 1: ${block1}`);
    io.emit("data", { uid: uid, block4: block1 });
  }
});
```

Gambar 4.9 Koneksi Serial

Gambar diatas menunjukkan bagaimana mengatur dan menangani koneksi serial menggunakan pustaka serialport di *Node.js*. Koneksi serial diatur untuk berkomunikasi melalui port COM9 dengan *baud* rate 9600, yang merupakan kecepatan komunikasi data. Objek *SerialPort* diinisialisasi dengan menentukan *path* ke *port serial* dan *baud rate* yang diinginkan. Data yang diterima dari *port serial* diproses oleh *parser ReadlineParser*, yang memisahkan data berdasarkan *delimiter* baris baru (\n). Dalam *callback parser.on("data", ...)*, data yang diterima diproses baris demi baris. Jika baris data dimulai dengan "UID:", program mengambil UID tersebut, menghilangkan spasi, dan menyimpannya dalam variabel uid. Jika baris data dimulai dengan "Block 1:", program mengambil data blok tersebut, menghilangkan spasi, dan menyimpannya dalam variabel block1. Ketika kedua variabel uid dan block1 sudah diisi, nilai-nilai ini dicetak ke konsol dan dikirim ke *klien* melalui *io.emit* dengan *event* data. Proses ini memungkinkan komunikasi data yang efektif antara perangkat yang terhubung melalui *port serial* dan aplikasi *Node.js*.

4.2.9 Endpoint Untuk Mengirim Data Ke Arduino

```
app.post("/arduinoApi", (req, res) => {
  const data = req.body.data;
  const encryptData = Aes128EcbEncrypt(data, key);
  const dataString = atob(encryptData);
  function stringToHex(string) {
    let hexString = "";
    for (let i = 0; i < string.length; i++) {
      let hex = string.charCodeAt(i).toString(16);
      hexString += ("00" + hex).slice(-2); // Pastikan setiap byte diwakili oleh dua digit hex
    }
    return hexString;
  }

  let hexString = stringToHex(dataString);

  console.log(hexString);
  port.write(hexString, (err) => {
    if (err) {
      console.log(err);
      res.status(500).json({ error: "write error" });
    }
    console.log("data terkirim", hexString);
    res.render("index");
  });
});
```

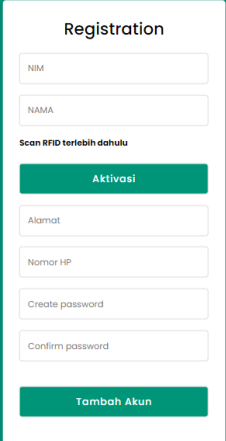
Gambar 4.10 Endpoint Untuk Mengirim Data ke Arduino

Gambar ini menunjukkan cara membuat *endpoint* `/arduinoApi` dalam aplikasi *Express* untuk mengirim data ke Arduino melalui *port serial*. *Endpoint* ini menerima permintaan POST dengan data yang dikirimkan dalam tubuh permintaan (`req.body.data`). Data ini kemudian dienkripsi menggunakan fungsi `Aes128EcbEncrypt` dengan kunci yang telah ditentukan (`key`). Hasil enkripsi, yang berbentuk *base64*, diubah menjadi string menggunakan fungsi `atob`. Fungsi `stringToHex` mengubah *string* ini menjadi representasi *heksadesimal*, di mana setiap karakter string diubah menjadi dua digit *heksadesimal*, memastikan setiap *byte* diwakili dengan benar. Hasil konversi *heksadesimal* ini kemudian dikirim melalui *port serial* ke Arduino menggunakan `port.write`. Jika terjadi kesalahan saat mengirim data, respons kesalahan dikembalikan dengan status 500. Jika berhasil, pesan yang dikirim dicetak ke konsol, dan halaman *index* dirender sebagai respons. *Endpoint* ini memungkinkan pengiriman data terenkripsi ke Arduino melalui koneksi *serial* dengan memastikan format data sesuai untuk pengiriman.

4.3 Implementasi Model Kedalam Web

Dalam penelitian ini, sistem dikembangkan menggunakan algoritma AES (*Advanced Encryption Standard*). Aplikasi ini dirancang dengan dua bagian utama, yaitu bagian *frontend* dan *backend*. Bagian *backend* dikembangkan dengan node js sementara bagian *frontend* dengan express js. Kemudian sistem dijalankan menggunakan server lokal.

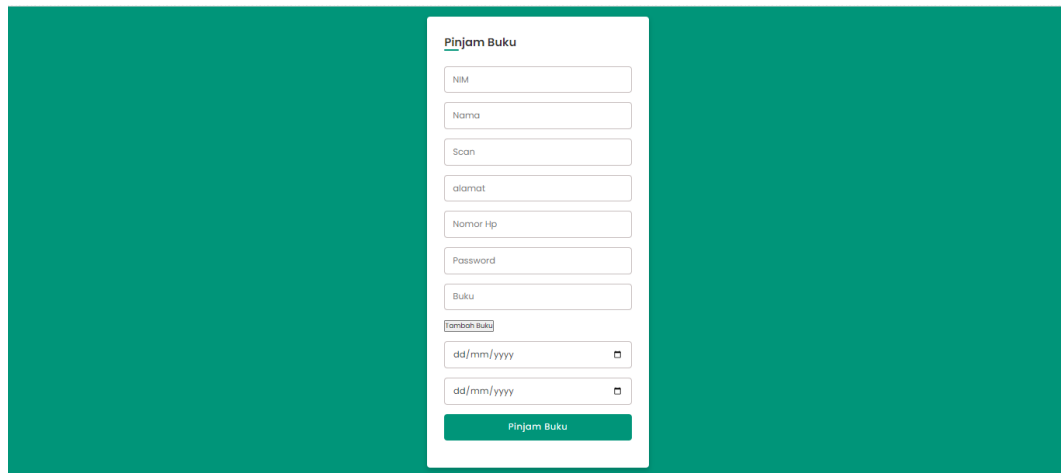
4.3.1 Halaman *Registration*

The image shows a registration form titled "Registration" centered on a teal background. The form is a white rectangle with rounded corners. It contains several input fields: "NIM", "NAMA", "Alamat", "Nomor HP", "Create password", and "Confirm password". There are two green buttons: "Aktivasi" (activation) located below the "Scan RFID terlebih dahulu" (scan RFID first) instruction, and "Tambah Akun" (add account) at the bottom. The text "Scan RFID terlebih dahulu" is in a small, light blue font.

Gambar 4.11 Halaman *Registration*

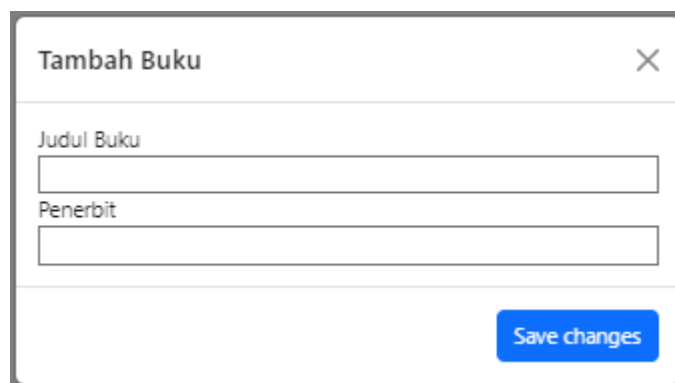
Halaman *registration* adalah halaman yang wajib di isi oleh penunjang untuk dapat melakukan peminjaman buku. Halaman ini memiliki kolom untuk data diri, scan kartu, dan memasukkan password. Scan kartu berguna untuk membaca UID yang terdapat pada kartu RFID. Terdapat tombol ‘aktivasi’ dan ‘tambah akun’, tombol aktivasi berguna untuk proses enkripsi dan *write* (menulis) pada memori kartu, dan tombol tambah akun berguna untuk menyimpan data pada database.

4.3.2 Halaman Peminjaman Buku dan Tambah Buku



Gambar 4.12 Halaman Peminjaman Buku

Pada gambar 4.12 diatas merupakan halaman untuk peminjam buku, yang mana jika kartu di scan maka nama dan nim otomatis terisi, selanjutnya mengisi kolom alamat, nomor hp, *password*, buku, dan tanggal peminjaman dan pengembalian. Pada kolom *password* diisi sesuai dengan *password* apa yang telah dimasukkan pada saat *registration*. Dan tombol tambah buku dapat dilihat penjelasannya dibawah ini.



Gambar 4.13 Halaman Tambah Buku

Gambar diatas merupakan halaman tambah buku, jika pada halaman *registration* tombol tambah buku di klik maka akan masuk kehalaman tambah buku ini. Pada halaman tambah buku ini pengunjung dapat mengisi kolom judul buku dan penerbit. Pada kolom judul buku pengunjung memasukkan buku apa yang telah

ingin dipinjam, dan pada halaman penerbit pengunjung mengisi siapa yang telah menulis buku yang dipinjam. Setelah kolom judul buku dan penerbit diisi, maka pengunjung dapat mengklik tombol '*save change*' maka data tambah buku akan tersimpan pada halaman tabel.

4.4 Pengujian

Pengujian dilakukan untuk melihat kinerja dari sistem dan alat yang telah dirancang. Untuk mengetahui sistem dapat berjalan dengan baik maka dilakukan percobaan. Berikut adalah hasil dari pengujian sistem dan alat yang dijelaskan dibawah ini, yaitu sebagai berikut:

4.4.1 Pengujian RFID Reader

Pengujian reader dilakukan dengan mengukur jarak baca dan mengukur lama baca data. Pengujian dari jarak reader membaca *tag* dapat dilihat pada tabel 4.3.

Tabel 4.3 Hasil Pengujian Jarak *Reader*

No Id	Rata-Rata Jarak Baca (cm)
53417954	0,68
B3DAB3FE	0,70
23FBB354	0,71
33BE5F54	0,69
33A4DA54	0,75

Kemudian untuk pengujian waktu *reader* membaca RFID tag dapat dilihat tabel 4.4.

Tabel 4.4 Hasil Pengujian Waktu *Reader*

No Id	Rata-Rata Waktu Baca (S)
53417954	0-1,5
B3DAB3FE	0,1,5
23FBB354	0,1,5
33BE5F54	0,1,5
33A4DA54	0,1,5

Dari tabel 4.3 dan 4.4 dapat dinyatakan bahwa *reader* dapat membaca RFID tag dengan baik. Kemampuan jarak baca *reader* juga cukup baik dan kemampuan jarak waktu *reader* baik dapat dilihat pada tabel.

4.4.2 Pengujian Enkripsi, *Write* (Menulis) dan *Read* (Membaca)

Pengujian enkripsi, *write* dan *read* ini dilakukan untuk memastikan apakah proses enkripsi, *write* dan *read* berhasil dilakukan. Proses enkripsi merupakan proses yang dilakukan untuk mengubah uid asli menjadi acak. Proses *write* ini dilakukan untuk memasukkan hasil enkripsi kedalam memori kartu RFID dan proses *read* ini lakukan untuk membaca hasil enkripsi yang telah di *write* kedalam memori kartu RFID. Berikut adalah hasil pengujian yang dihasilkan..

4.4.2.1 Hasil Pengujian Enkripsi

Hasil enkripsi adalah hasil dari uid asli yang di proses menggunakan algoritma AES, berikut ini adalah tabel hasil enkripsi:

Tabel 4.5 Hasil Pengujian Enkripsi

No.	UID Asli	Hasil Enkripsi
1.	53 41 79 54	56 02 7F 49 11 A9 45 17 E6 BF DE 56 27 D7 68 9C
2.	B3 DA B3 FE	4A F2 71 D3 76 43 AC 07 D5 2B 63 C2 12 A2 EE BB
3.	23 FB B3 54	4F 2A 8C9E 1B 3D 7F 6A 5C 1E 3B 2D 8A 9F 0C 4E
4.	33 A4 DA 54	B9 5C 6D 24 C8 68 18 0E A0 17 9A FD D8 F8 4B C7
5.	33 BE 5F 54	B6 0C E0 70 FB 85 12 D4 4B D7 F4 E3 02 8A 54 B7

4.4.2.2 Hasil Pengujian *Read* (Membaca) Menggunakan RFID Reader Aplikasi MIFARE Classic Tool

Hasil *read* merupakan proses membaca isi memori kartu RFID yang sebelum *diread* maka sistem akan *write* kedalam memori kartu RFID. Berikut adalah tabel hasil *read* dari 2 aplikasi, yaitu.

Berikut adalah beberapa hasil percobaan dari beberapa kartu tag RFID. Setiap kartu memiliki UID sebagai identitas kartu. Kertu RFID ini merupakan kartu yang digunakan pengujung untuk meminjaman buku diperpustakaan.

1. UID 53 41 79 54

a. RFID Reader

5	23	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	22	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	21	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	20	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
4	19	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	18	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	17	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	16	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
3	15	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	14	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	13	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	12	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
2	11	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	10	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	9	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	8	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
1	7	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	6	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	5	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	4	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
0	3	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	2	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	1	56 02 7F 49	11 A9 45 17	E6 BF DE 56	27 D7 68 9C
	0	53 41 79 54	3F 08 04 00	62 63 64 65	66 67 68 69

Gambar 4.14 Hasil Uji Read Kartu (1) Pada RFID Reader

Gambar diatas merupakan tampilan data dalam format *hex dump*, di mana data biner diwakili dalam bentuk heksadesimal (basis 16). Dalam *hex dump*, biasanya data biner yang diambil dari memori atau file ditampilkan dalam dua kolom: satu kolom berisi data heksadesimal dan kolom lainnya adalah representasi karakter dari data tersebut (jika karakter tersebut bisa direpresentasikan dalam bentuk ASCII). Pada sector 0 dan blok 1 terdapat hasil enkripsi dari UID kartu tersebut. Gambar diatas ini untuk read ini dilakukan menggunakan RFID Reader.

b. MIFARE Classic Tool



Gambar 4.15 Hasil Uji Read Kartu (1) Pada MIFARE Classic Tool

Gambar diatas adalah *hexdump*, yang merupakan representasi data biner dalam format *hexadecimal*. Data ini menunjukkan *byte-by-byte* informasi yang disusun berdasarkan *offset* atau alamat memori, dalam gambar ini terdapat pada sektor 0 serta blok 1 yang merupakan hasil enkripsi dari UID kartu. Kartu ini dijalankan menggunakan MIFARE Classic Tool.

2. UID B3 DA B3 FE

a. RFID Reader

5	23	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF	[0 0 1]
	22	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
	21	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
	20	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
4	19	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF	[0 0 1]
	18	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
	17	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
	16	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
3	15	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF	[0 0 1]
	14	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
	13	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
	12	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
2	11	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF	[0 0 1]
	10	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
	9	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
	8	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
1	7	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF	[0 0 1]
	6	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
	5	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
	4	16 7A 2F DD	B6 7A DB 3B	13 97 C6 22	9D EF 66 FB	[0 0 0]
0	3	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF	[0 0 1]
	2	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00	[0 0 0]
	1	4A F2 71 D3	76 43 AC 07	D5 2B 63 C2	12 A2 EE BB	[0 0 0]
	0	B3 DA B3 FE	24 08 04 00	62 63 64 65	66 67 68 69	[0 0 0]

Gambar 4.16 Hasil Uji Read Kartu (2) Pada RFID Reader

Gambar diatas merupakan tampilan data dalam format *hex dump*, di mana data biner diwakili dalam bentuk heksadesimal (basis 16). Dalam *hex dump*, biasanya data biner yang diambil dari memori atau file ditampilkan dalam dua kolom: satu kolom berisi data heksadesimal dan kolom lainnya adalah representasi karakter dari data tersebut (jika karakter tersebut bisa direpresentasikan dalam bentuk ASCII). Pada sector 0 dan blok 1 terdapat hasil enkripsi dari UID kartu tersebut. Gambar diatas ini untuk read ini dilakukan menggunakan RFID Reader.

b. MIFARE Classic Tool



Gambar 4.17 Hasil Uji Read Kartu (2) Pada MIFARE Classic Tool

Gambar diatas adalah *hexdump*, yang merupakan representasi data biner dalam format *hexadecimal*. Data ini menunjukkan *byte-by-byte* informasi yang disusun berdasarkan *offset* atau alamat memori, dalam gambar ini terdapat pada sektor 0 serta blok 1 yang merupakan hasil enkripsi dari UID kartu. Kartu ini dijalankan menggunakan MIFARE Classic Tool.

3. UID 23 FB B354

a. RFID Reader

5	23	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	22	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	21	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	20	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
4	19	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	18	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	17	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	16	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
3	15	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	14	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	13	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	12	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
2	11	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	10	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	9	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	8	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
1	7	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	6	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	5	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	4	E6 0B CE 83	B5 FD 76 8B	14 22 13 F0	18 D9 0A 9E
0	3	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	2	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	1	4F 2A 8C 9E	1B 3D 7F 6A	5C 1E 3B 2D	8A 9F 0C 4E
	0	23 FB B3 54	3F 08 04 00	62 63 64 65	66 67 68 69

Gambar 4.18 Hasil Uji Read Kartu (3) Pada RFID Reader

Gambar diatas merupakan tampilan data dalam format *hex dump*, di mana data biner diwakili dalam bentuk heksadesimal (basis 16). Dalam *hex dump*, biasanya data biner yang diambil dari memori atau file ditampilkan dalam dua kolom: satu kolom berisi data heksadesimal dan kolom lainnya adalah representasi karakter dari data tersebut (jika karakter tersebut bisa direpresentasikan dalam bentuk ASCII). Pada sector 0 dan blok 1 terdapat hasil enkripsi dari UID kartu tersebut. Gambar diatas ini untuk read ini dilakukan menggunakan RFID Reader.

b. MIFARE Classic Tool

```

Sector: 0
23FBB3543F0804006263646566676869
4F2A8C9E1B3D7F6A5C1E3B2D8A9F0C4E
00000000000000000000000000000000
FFFFFFFFFFFFFFFFFF078069FFFFFFFFFFFF

Sector: 1
E60BCE83B5FD768B142213F018D90A9E
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFFFFFF078069FFFFFFFFFFFF

Sector: 2
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFFFFFF078069FFFFFFFFFFFF

Sector: 3
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFFFFFF078069FFFFFFFFFFFF

Sector: 4
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFFFFFF078069FFFFFFFFFFFF

```

Gambar 4.19 Hasil Uji Read Kartu (3) Pada MIFARE Classic Tool

Gambar diatas adalah *hexdump*, yang merupakan representasi data biner dalam format *hexadecimal*. Data ini menunjukkan *byte-by-byte* informasi yang disusun berdasarkan *offset* atau alamat memori, dalam gambar ini terdapat pada sektor 0 serta blok 1 yang merupakan hasil enkripsi dari UID kartu. Kartu ini dijalankan menggunakan MIFARE Classic Tool.

4. UID 33 A4 DA 54

a. RFID Reader

5	23	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	22	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	21	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	20	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
4	19	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	18	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	17	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	16	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
3	15	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	14	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	13	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	12	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
2	11	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	10	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	9	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	8	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
1	7	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	6	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	5	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	4	E6 0B CE 83	B5 FD 76 8B	14 22 13 F0	18 D9 0A 9E
0	3	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	2	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	1	4F 2A 8C 9E	1B 3D 7F 6A	5C 1E 3B 2D	8A 9F 0C 4E
	0	23 FB B3 54	3F 08 04 00	62 63 64 65	66 67 68 69

Gambar 4.20 Hasil Uji Read Kartu (4) Pada RFID Reader

Gambar diatas merupakan tampilan data dalam format *hex dump*, di mana data biner diwakili dalam bentuk heksadesimal (basis 16). Dalam *hex dump*, biasanya data biner yang diambil dari memori atau file ditampilkan dalam dua kolom: satu kolom berisi data heksadesimal dan kolom lainnya adalah representasi karakter dari data tersebut (jika karakter tersebut bisa direpresentasikan dalam bentuk ASCII). Pada sector 0 dan blok 1 terdapat hasil enkripsi dari UID kartu tersebut. Gambar diatas ini untuk read ini dilakukan menggunakan RFID Reader.

b. MIFARE Classic Tool

```

Sector: 0
23FBB3543F0804006263646566676869
4F2A8C9E1B3D7F6A5C1E3B2D8A9F0C4E
00000000000000000000000000000000
FFFFFFFFFFFFFFFFFF078069FFFFFFFFFFFF

Sector: 1
E60BCE83B5FD768B142213F018D90A9E
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFFFFFF078069FFFFFFFFFFFF

Sector: 2
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFFFFFF078069FFFFFFFFFFFF

Sector: 3
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFFFFFF078069FFFFFFFFFFFF

Sector: 4
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFFFFFF078069FFFFFFFFFFFF

```

Gambar 4.21 Hasil Uji Read Kartu (4) Pada MIFARE Classic Tool

Gambar diatas adalah *hexdump*, yang merupakan representasi data biner dalam format *hexadecimal*. Data ini menunjukkan *byte-by-byte* informasi yang disusun berdasarkan *offset* atau alamat memori, dalam gambar ini terdapat pada sektor 0 serta blok 1 yang merupakan hasil enkripsi dari UID kartu. Kartu ini dijalankan menggunakan MIFARE Classic Tool.

5. UID 33 BE 5F 54

a. RFID Reader

5	23	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	22	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	21	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	20	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
4	19	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	18	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	17	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	16	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
3	15	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	14	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	13	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	12	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
2	11	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	10	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	9	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	8	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
1	7	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	6	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	5	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	4	90 ED 92 D1	21 FB E4 46	44 AE 9D 25	3E EA CD C2
0	3	00 00 00 00	00 00 FF 07	80 69 FF FF	FF FF FF FF
	2	00 00 00 00	00 00 00 00	00 00 00 00	00 00 00 00
	1	B9 5C 6D 24	C8 68 18 0E	A0 17 9A FD	D8 F8 4B C7
	0	33 A4 DA 54	19 08 04 00	62 63 64 65	66 67 68 69

Gambar 4.22 Hasil Uji Read Kartu (5) Pada RFID Reader

Gambar diatas merupakan tampilan data dalam format *hex dump*, di mana data biner diwakili dalam bentuk heksadesimal (basis 16). Dalam *hex dump*, biasanya data biner yang diambil dari memori atau file ditampilkan dalam dua kolom: satu kolom berisi data heksadesimal dan kolom lainnya adalah representasi karakter dari data tersebut (jika karakter tersebut bisa direpresentasikan dalam bentuk ASCII). Pada sector 0 dan blok 1 terdapat hasil enkripsi dari UID kartu tersebut. Gambar diatas ini untuk read ini dilakukan menggunakan RFID Reader.

b. MIFARE Classic Tool

```

Sector: 0
33A4DA54190804006263646566676869
B95C6D24C868180EA0179AFDD8F84BC7
00000000000000000000000000000000
FFFFFFFFFFFFFFFFFF078069FFFFFFFFFFFF

Sector: 1
90ED92D121FBE44644AE9D253EEACDC2
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFFFFFF078069FFFFFFFFFFFF

Sector: 2
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFFFFFF078069FFFFFFFFFFFF

Sector: 3
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFFFFFF078069FFFFFFFFFFFF

Sector: 4
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
FFFFFFFFFFFFFFFFFF078069FFFFFFFFFFFF

```

Gambar 4.23 Hasil Uji Read Kartu (5) Pada MIFARE Classic Tool

Gambar diatas adalah *hexdump*, yang merupakan representasi data biner dalam format *hexadecimal*. Data ini menunjukkan *byte-by-byte* informasi yang disusun berdasarkan *offset* atau alamat memori, dalam gambar ini terdapat pada sektor 0 serta blok 2 yang merupakan hasil enkripsi dari UID kartu.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berikut merupakan kesimpulan penelitian tentang pengamanan kartu tag RFID dengan algoritma *Advanced Encryption Standard* (AES).

1. AES membantu melindungi kartu tag RFID dari serangan duplikat dan manipulasi.
2. Penggunaan AES dalam mengenkripsi data pada kartu *tag* RFID memberikan tingkat keamanan yang tinggi.
3. Penggunaan enkripsi AES pada kartu tag RFID membantu menjaga privasi pengguna dengan memastikan bahwa data yang disimpan atau ditransfer hanya dapat diakses oleh pihak yang memiliki kunci enkripsi yang sesuai.

5.2 Saran

Berikut adalah beberapa usulan yang dapat dijadikan saran untuk dipertimbangkan dalam penelitian selanjutnya:

1. Implementasi enkripsi *end-to-end* pada data yang disimpan dalam tag RFID sangat disarankan untuk mencegah akses tidak sah.
2. Pengembangan sistem pemantauan *real-time* untuk mendeteksi aktivitas mencurigakan atau upaya akses ilegal pada *tag* RFID akan sangat bermanfaat.
3. Melakukan analisis yang lebih mendalam terhadap potensi serangan terhadap sistem pengamanan yang menggunakan kartu tag RFID dengan enkripsi AES. Ini termasuk pengujian kekuatan algoritma AES terhadap berbagai jenis serangan kriptografi seperti *brute force*, *chosen-plaintext attack*, dan lainnya.

DAFTAR PUSTAKA

- Afiani, D. N. (2014). Desain E-Tol Dengan Radio Frequency Identification (RFID) Menggunakan Algoritma Kriptografi Blowfish. *Dokumen Karya Ilmiah*, 1–8. http://eprints.dinus.ac.id/13200/1/jurnal_13670.pdf
- Bhaskoro, S. B., Anggraeni, P., & Nijam, E. N. (2023). Sistem Keamanan Kartu NFC Menggunakan Metode AES pada Sistem Pembayaran Elektronik. *Jurnal Sistem Cerdas*, 6(2), 168–178. <https://doi.org/10.37396/jsc.v6i2.326>
- Choerudin, A. (2021). Perancangan Sistem Informasi Peminjaman Alat dan Peralatan Laboratorium Berbasis RFID. *Journal of Telecommunication, Electronics, and Control Engineering (JTECE)*, 3(1), 41–47. <https://doi.org/10.20895/jtece.v3i1.251>
- Daulay, N. K., & Alamsyah, M. N. (2019). Monitoring Sistem Keamanan Pintu Menggunakan Rfid Dan Fingerprint Berbasis Web Dan Database. *Jusikom : Jurnal Sistem Komputer Musirawas*, 4(02), 85–92. <https://doi.org/10.32767/jusikom.v4i2.632>
- Fachrozi, M. F., & Fahmi, H. (2021). Penerapan Metode AES-128 Untuk Pengamanan Data Absensi FingerPrint. *JIKOMSI [Jurnal Ilmu Komputer Dan Sistem Informasi]*, 3(3), 1–8.
- Fadhilil Khaliq, K. (2021). Pengamanan Data Akta Dengan Metode Aes Berbasis Cloud Computing. *Jurnal Teknologi Dan Ilmu Komputer Prima (Jutikomp)*, 4(1), 509–512. <https://doi.org/10.34012/jutikomp.v4i1.1555>
- Insan, R. M., Ruuhwan, R., & Rizal, R. (2019). Penerapan Teknologi Radio Frequency Identification (RFID) Pada Data Kunjungan Perpustakaan. *Informatics and Digital Expert (INDEX)*, 1(1), 1–6. <https://doi.org/10.36423/ide.v1i1.281>
- Linardi, W., Aribowo, A., & Yugopuspito, P. (2017). Prototipe Sistem Penguncian Loker Elektronik Dengan Teknologi Identifikasi-Frekuensi Radio. *Linardi, William Aribowo, Arnold Yugopuspito, Pujiyanto*, IV(6), 1–5.

- Mamun, M. A. A., & Hasanuzzaman, M. (2020). IMPLEMENTASI SISTEM MONITORING ABSENSI RFID PROXYMITY. *Energy for Sustainable Development: Demand, Supply, Conversion and Management*, 11(3), 1–14.
- Mochamad Irvan Fadillah. (2020). Aplikasi Informasi Absensi Karyawan Di Pt . Gita Variasi Berbasis RFID. *Jimtek*, 1(2), 80–88.
- NATALIANA, D., HADIATNA, F., & FAUZI, A. (2019). Rancang Bangun Sistem Keamanan RFID Tag menggunakan Metode Caesar Cipher pada Sistem Pembayaran Elektronik. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 7(3), 427. <https://doi.org/10.26760/elkomika.v7i3.427>
- Nugraha, F. (2014). Analisa Dan Perancangan Sistem Informasi Perpustakaan. *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, 5(1), 27–32. <https://doi.org/10.24176/simet.v5i1.132>
- Purnomo, Y. E. (2017). Sistem Peminjaman Buku Berbasis RFID Publikasi Ilmiah. *Universitas Muhammadiyah Surakarta*, 1–20. <http://eprints.ums.ac.id/id/eprint/49216>
- Radice, A. H. (2020). Jumper. *Notes and Queries*, 158(24), 431. <https://doi.org/10.1093/nq/158.24.431c>
- Suryam Dora, D. (2017). Smart Gate System untuk Akses Kontrol Keamanan Kampus. *STUDIES ON VARIATION IN MILK PRODUCTION AND IT'S CONSTITUENTS DURING DIFFERENT SEASON, STAGE OF LACTATION AND PARITY IN GIR COWS M.V.Sc D SURYAM DORA LIVESTOCK*, 6–18.
- Wulandaru, L. A., Supeno, B., & Sumardi, S. (2017). Rancang Bangun Perangkat Rekam Medik Berbasis Teknologi RFID. *Berkala Sainstek*, 5(2), 104. <https://doi.org/10.19184/bst.v5i2.5701>
- Yulita, N., Setyaningsih, D., Eng, M., & Wibowo, B. C. (2022). *SISTEM PRESENSI DENGAN VALIDASI E-KTP MENGGUNAKAN NFC PN532 V3 DAN NOTIFIKASI TELEGRAM Oleh : Rahma Ningsih*. 1–25.

