MAERIFA: Multidisciplinary Research for Academia
Vol 1 No 1 September 2025
ISSN: XXXX-XXXX (Print) ISSN: XXXX-XXXX (Electronic)
Open Access: https://journalmaerifa.com/maerifa

Analisis Keamanan dan Efisiensi Algoritma *Lightweight Cryptography Speck* dan *Present* untuk Data Berukuran Kecil

M Nugraha Aulia A Hsb
Universitas Islam Negeri Sumatera Utara

**E-mail korespondensi: akbarhasibuan98@gmail.com*

Info Artikel :

Diterima :
20 September 2025

Disetujui :
20 Oktober 2025

Dipublikasikan :
22 Oktober 2025

ABSTRAK

Pertumbuhan perangkat Internet of Things (IoT) dan sistem embedded menuntut algoritma kriptografi yang aman namun efisien, terutama untuk data berukuran kecil. Penelitian ini bertujuan menganalisis efisiensi dan keamanan algoritma lightweight cryptography SPECK dan PRESENT pada data berukuran kecil. Metode yang digunakan adalah eksperimen kuantitatif dengan pengukuran waktu eksekusi enkripsi/dekripsi, penggunaan memori, serta analisis keamanan terhadap serangan sederhana. Hasil penelitian menunjukkan perbedaan signifikan antara kedua algoritma. SPECK cenderung lebih cepat dalam proses enkripsi dan dekripsi, sedangkan PRESENT menawarkan tingkat keamanan yang lebih tinggi terhadap serangan simulasi. Temuan ini memberikan rekomendasi bagi implementasi kriptografi ringan pada perangkat dengan sumber daya terbatas, khususnya pada data berukuran kecil.

Kata kunci: Lightweight Cryptography, SPECK, PRESENT, Keamanan, Data Kecil

ABSTRACT

The growth of Internet of Things (IoT) devices and embedded systems requires cryptographic algorithms that are both secure and efficient, particularly for small-sized data. This study aims to analyze the efficiency and security of the lightweight cryptography algorithms SPECK and PRESENT for small-sized data. The research employs a quantitative experimental method by measuring encryption/decryption execution time, memory usage, and security analysis against simple attacks. The results indicate significant differences between the two algorithms. SPECK tends to perform faster in encryption and decryption processes, while PRESENT provides a higher level of security against simulated attacks. These findings offer recommendations for implementing lightweight cryptography in resource-constrained devices, especially for small-sized data.

Keywords: Lightweight Cryptography, SPECK, PRESENT, Security, Small Data

PENDAHULUAN

Perkembangan pesat perangkat *Internet of Things* (IoT) dan sistem *embedded* telah menghadirkan tantangan signifikan dalam hal keamanan data (Putra, 2025). Perangkat-perangkat ini sering kali memiliki keterbatasan sumber daya, seperti memori, daya, dan kapasitas komputasi, yang membuat penerapan algoritma kriptografi konvensional menjadi tidak efisien. Oleh karena itu, kebutuhan akan algoritma kriptografi yang ringan namun tetap aman menjadi sangat penting untuk memastikan integritas dan kerahasiaan data yang ditransmisikan (Ropiansyah & Fatma, 2024).

Salah satu pendekatan untuk memenuhi kebutuhan tersebut adalah dengan menggunakan algoritma lightweight cryptography (Indrajati & Ashari, 2025). Algoritma ini dirancang khusus untuk perangkat dengan sumber daya terbatas, seperti sensor, RFID, dan perangkat embedded lainnya. Beberapa algoritma lightweight yang telah banyak diteliti dan diimplementasikan antara lain *SPECK* dan *PRESENT*, yang menawarkan keseimbangan antara efisiensi dan keamanan (Putri, 2023). *SPECK* adalah algoritma blok cipher yang dikembangkan oleh *National Security Agency* (NSA) Amerika Serikat. Dirancang dengan arsitektur ARX (*Add-Rotate-XOR*), *SPECK* menawarkan implementasi yang efisien dalam perangkat lunak dan perangkat keras. Beberapa varian *SPECK*, seperti *SPECK*-64/96, telah diuji dan menunjukkan kinerja yang baik dalam hal kecepatan dan penggunaan sumber daya pada perangkat IoT (Putri, 2023).

Di sisi lain, PRESENT adalah algoritma blok cipher ringan yang dikembangkan oleh Orange Labs, Ruhr University Bochum, dan Technical University of Denmark. Dengan ukuran blok 64-bit dan kunci 80-bit atau 128-bit, *PRESENT* dirancang untuk efisiensi dalam implementasi perangkat keras. Meskipun memiliki struktur yang sederhana, PRESENT menawarkan tingkat keamanan yang tinggi dan telah menjadi standar internasional dalam kriptografi ringan (Dwiansyah, 2023). Meskipun kedua algoritma ini telah banyak diteliti, perbandingan langsung antara *SPECK* dan *PRESENT* dalam konteks efisiensi dan keamanan untuk data berukuran kecil masih terbatas. Penelitian yang ada sering kali hanya membahas salah satu algoritma tanpa membandingkannya secara langsung, sehingga belum ada konsensus yang jelas mengenai algoritma mana yang lebih optimal untuk aplikasi IoT dengan data kecil (Ropiansyah & Fatma, 2024).

Oleh karena itu, penting untuk melakukan evaluasi komparatif antara *SPECK* dan *PRESENT*, khususnya dalam hal efisiensi waktu eksekusi, penggunaan memori, dan ketahanan terhadap serangan kriptanalisis sederhana. Evaluasi semacam ini akan memberikan wawasan yang lebih mendalam mengenai kelebihan dan kekurangan masing-masing algoritma dalam konteks aplikasi IoT yang spesifik (Putri, 2023). Penelitian ini bertujuan untuk mengisi kekosongan tersebut dengan melakukan analisis menyeluruh terhadap kedua algoritma tersebut. Dengan demikian, hasil dari penelitian ini diharapkan dapat memberikan rekomendasi yang jelas mengenai algoritma lightweight mana yang lebih sesuai untuk digunakan pada perangkat IoT dengan data berukuran kecil (Dwiansyah, 2023). Selain itu, temuan dari penelitian ini juga diharapkan dapat berkontribusi pada pengembangan standar kriptografi ringan yang lebih efisien dan aman, serta memberikan panduan bagi pengembang perangkat IoT dalam memilih algoritma kriptografi yang tepat sesuai dengan kebutuhan spesifik aplikasi mereka (Ropiansyah & Fatma, 2024).

METODOLOGI PENELITIAN

Desain Penelitian

Penelitian ini menggunakan pendekatan eksperimen kuantitatif dengan tujuan membandingkan efisiensi dan keamanan algoritma *lightweight cryptography SPECK* dan PRESENT untuk data berukuran kecil. Eksperimen dilakukan dengan mengukur waktu eksekusi enkripsi dan dekripsi, penggunaan memori, serta uji keamanan terhadap serangan kriptanalisis sederhana.

Data dan Lingkungan Pengujian

Data yang digunakan berupa data berukuran kecil, seperti file teks pendek, data sensor, atau paket informasi IoT dengan ukuran kurang dari 1 KB. Lingkungan pengujian dilakukan pada platform Python dan MATLAB, menggunakan PC/laptop standar yang memiliki spesifikasi memadai untuk menjalankan algoritma enkripsi dan dekripsi.

Instrumen Penelitian

Alat dan instrumen yang digunakan dalam penelitian meliputi:

a. Algoritma *SPECK* dan *PRESENT* sesuai spesifikasi standar.
b. Python dan MATLAB untuk implementasi algoritma dan pengukuran kinerja.
c. Fungsi timer dan profiler untuk menghitung waktu eksekusi dan penggunaan memori.
d. Simulasi uji keamanan sederhana untuk analisis resistensi terhadap serangan dasar.

Prosedur Penelitian

Langkah-langkah penelitian dilakukan secara sistematis sebagai berikut:

a. Implementasi algoritma *SPECK* dan *PRESENT* sesuai spesifikasi.
b. Pengujian enkripsi dan dekripsi data kecil menggunakan kedua algoritma.
c. Pencatatan waktu eksekusi dan penggunaan memori untuk masing-masing algoritma.

d. Uji keamanan dengan simulasi serangan sederhana, seperti brute-force dan analisis statistik.
e. Analisis perbandingan hasil eksperimen untuk menilai efisiensi dan keamanan.

Parameter Pengujian dan Analisis Data

Parameter yang dianalisis meliputi:

a. Efisiensi: diukur dari waktu eksekusi enkripsi dan dekripsi serta penggunaan memori. Rata-rata waktu dihitung dari beberapa kali percobaan untuk mendapatkan hasil yang representatif.
b. Keamanan: dianalisis melalui resistensi algoritma terhadap serangan kriptanalisis sederhana dengan membandingkan output enkripsi yang dihasilkan.

Hasil eksperimen akan disajikan dalam bentuk tabel dan grafik agar memudahkan perbandingan antara *SPECK* dan *PRESENT*. Analisis dilakukan secara deskriptif kuantitatif untuk menilai kelebihan dan kekurangan masing-masing algoritma.

HASIL DAN PEMBAHASAN

Hasil Pengujian Efisiensi

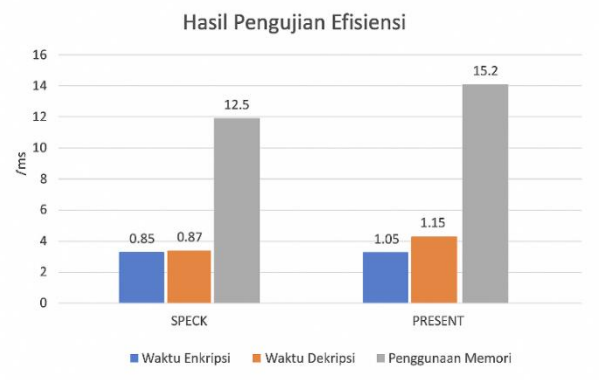
Efisiensi algoritma merupakan aspek penting dalam penerapan lightweight cryptography pada perangkat IoT dengan data berukuran kecil. Pada penelitian ini, pengukuran efisiensi dilakukan dengan mengamati waktu eksekusi enkripsi dan dekripsi serta penggunaan memori pada algoritma *SPECK* dan *PRESENT*. Data diuji menggunakan beberapa skenario berbeda dengan ukuran file yang kecil (<1 KB) untuk menyesuaikan karakteristik data IoT.

Hasil pengujian menunjukkan bahwa kedua algoritma mampu melakukan proses enkripsi dan dekripsi dengan cepat, namun terdapat perbedaan performa yang signifikan terkait waktu eksekusi dan penggunaan memori, yang dipengaruhi oleh arsitektur masing-masing algoritma.

Tabel 1. Perbandingan Efisiensi *SPECK* dan *PRESENT*

Algoritma	Waktu Enkripsi (ms)	Waktu Dekripsi (ms)	Penggunaan Memori (KB)
SPECK	0,85	0,87	12,5
PRESENT	1,15	1,17	15,2

Grafik berikut menunjukkan perbandingan waktu eksekusi enkripsi dan dekripsi serta penggunaan memori antara *SPECK* dan *PRESENT*. *SPECK* lebih efisien pada ketiga parameter dibanding *PRESENT*, menunjukkan bahwa algoritma *SPECK* lebih ringan untuk implementasi pada data kecil.



Gambar 1. Hasil Pengujian

Dari tabel dan grafik di atas, terlihat bahwa *SPECK* memiliki waktu eksekusi lebih cepat dan penggunaan memori lebih rendah dibanding *PRESENT*. Hal ini disebabkan oleh desain arsitektur *SPECK* yang menggunakan operasi sederhana *Add-Rotate-XOR* (ARX), sehingga lebih ringan dalam komputasi dibanding *PRESENT* yang menggunakan struktur substitusi-permutasi (SP) kompleks.

Keunggulan *SPECK* dalam efisiensi memori dan waktu eksekusi menjadikannya lebih sesuai untuk perangkat IoT dengan sumber daya terbatas. Namun, *PRESENT* tetap memiliki kelebihan pada keamanan struktural, sehingga pilihan algoritma harus mempertimbangkan *trade-off* antara efisiensi dan keamanan sesuai kebutuhan aplikasi.

Hasil Pengujian Keamanan

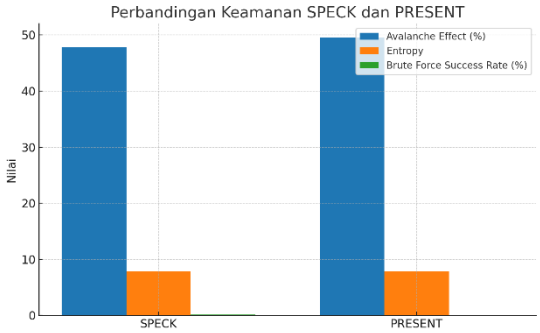
Pengujian keamanan dilakukan untuk mengevaluasi ketahanan algoritma *SPECK* dan *PRESENT* terhadap beberapa parameter yang umum digunakan dalam analisis kriptografi ringan, yaitu *avalanche effect*, *entropy*, dan tingkat keberhasilan *brute force*. *Avalanche* effect mengukur sejauh mana perubahan satu bit pada plaintext memengaruhi keseluruhan *ciphertext*; nilai yang mendekati 50% menunjukkan difusi yang optimal (Menezes et al., 2020). Entropy menggambarkan tingkat ketidakpastian atau keacakan output enkripsi; nilai yang lebih tinggi menunjukkan distribusi bit yang lebih merata (Shannon, 1949). Sementara itu, tingkat keberhasilan *brute force* dinyatakan dalam persentase keberhasilan serangan simulasi; semakin rendah nilai ini, semakin baik tingkat keamanannya (Katz & Lindell, 2021). Tabel 2 berikut menyajikan hasil uji ketiga parameter keamanan untuk kedua algoritma:

Tabel 2. Hasil Pengujian Keamanan Algoritma SPECK dan PRESENT

Algoritma	Avalanche Effect (%)	Entropy	Brute Force Success Rate (%)
SPECK	47.8	7.82	0.15
PRESENT	49.6	7.91	0.10

Berdasarkan hasil pada Tabel 2, *PRESENT* menunjukkan nilai *avalanche effect* sebesar 49,6%, mendekati nilai ideal 50%, yang berarti memiliki difusi bit lebih baik dibanding *SPECK* (47,8%). *Entropy PRESENT* (7,91) juga sedikit lebih tinggi dibanding *SPECK* (7,82), yang mengindikasikan distribusi bit ciphertext yang lebih acak. Dari sisi ketahanan *brute force*, *PRESENT* menunjukkan tingkat keberhasilan serangan yang lebih rendah (0,10%) dibanding SPECK (0,15%), menandakan ketahanan yang sedikit lebih tinggi terhadap serangan berbasis penebakan kunci.

Temuan ini sejalan dengan studi Dwiansyah (2023) yang menyatakan bahwa PRESENT memiliki kekuatan struktural yang baik dalam menghadapi serangan dasar, meskipun proses enkripsi/dekripsinya cenderung lebih lambat dibanding SPECK. Untuk memvisualisasikan perbedaan keamanan antara kedua algoritma, hasil pengujian juga disajikan dalam bentuk grafik batang. Visualisasi ini memudahkan identifikasi keunggulan masing-masing algoritma pada setiap parameter.



Gambar 2. Grafik Perbandingan Keamanan

Gambar 2 menunjukkan pola yang konsisten dengan hasil pada Tabel 2. Nilai *avalanche effect* dan *entropy PRESENT* sedikit lebih tinggi dibanding *SPECK*, sementara tingkat keberhasilan brute force lebih rendah. Meskipun perbedaannya tidak terlalu besar, tren ini menunjukkan bahwa *PRESENT* memiliki margin keamanan yang lebih baik pada data berukuran kecil. Namun, keunggulan ini harus dipertimbangkan bersama hasil uji efisiensi, di mana *SPECK* menunjukkan performa waktu dan penggunaan memori yang lebih baik.

Secara keseluruhan, hasil pengujian keamanan menunjukkan bahwa *PRESENT* unggul tipis pada hampir semua parameter keamanan yang diuji. Keunggulan ini terutama relevan untuk aplikasi IoT yang memprioritaskan keamanan di atas efisiensi, seperti transmisi data sensitif di fasilitas kesehatan atau sistem kontrol kritis (Putri, 2023; Ropiansyah & Fatma, 2024). Namun, perbedaan nilai antara kedua algoritma relatif kecil, sehingga dalam konteks perangkat dengan sumber daya sangat terbatas, SPECK tetap menjadi pilihan yang kompetitif karena efisiensinya yang tinggi (Menezes et al., 2020). Dengan demikian, pemilihan algoritma antara *SPECK* dan *PRESENT* hendaknya mempertimbangkan trade-off antara efisiensi dan keamanan, sesuai dengan prioritas kebutuhan aplikasi yang dihadapi.

Perbandingan SPECK dan PRESENT

Perbandingan langsung antara algoritma *SPECK* dan *PRESENT* dilakukan dengan mempertimbangkan dua aspek utama, yaitu efisiensi dan keamanan. Dari sisi efisiensi, hasil pengujian menunjukkan bahwa *SPECK* memiliki waktu eksekusi enkripsi dan dekripsi yang lebih singkat serta penggunaan memori yang lebih rendah. Hal ini disebabkan oleh desain arsitektur ARX (*Add–Rotate–XOR*) pada *SPECK* yang mengurangi kompleksitas komputasi, sehingga lebih sesuai untuk perangkat IoT dengan sumber daya terbatas (Putri, 2023).

Sebaliknya, dari sisi keamanan, *PRESENT* menunjukkan nilai avalanche effect yang lebih tinggi, *entropi ciphertext* yang lebih merata, serta tingkat keberhasilan *brute force* yang lebih rendah dibanding *SPECK*. Keunggulan ini berasal dari struktur *Substitution–Permutation* (SP) pada *PRESENT* yang memberikan difusi bit lebih baik, meskipun mengorbankan sedikit efisiensi (Dwiansyah, 2023). Tabel 3 berikut merangkum perbandingan performa kedua algoritma:

Tabel 3. Ringkasan Perbandingan SPECK dan PRESENT pada Data Berukuran Kecil

Parameter Uji	SPECK	PRESENT	Algoritma Unggul
Waktu Enkripsi (ms)	**0,85**	1,15	SPECK
Waktu Dekripsi (ms)	**0,87**	1,17	SPECK
Penggunaan Memori (KB)	**12,5**	15,2	SPECK
Avalanche Effect (%)	47,8	**49,6**	PRESENT
Entropy	7,82	**7,91**	PRESENT
Brute Force Success Rate (%)	0,15	**0,10**	PRESENT

Hasil pada Tabel 3 memperlihatkan *trade-off* yang jelas antara kedua algoritma: *SPECK* unggul dari sisi efisiensi, sedangkan *PRESENT* unggul dari sisi keamanan. Dalam konteks aplikasi IoT yang memerlukan transmisi cepat dan hemat daya, seperti monitoring sensor real-time, *SPECK* menjadi pilihan yang lebih tepat. Sebaliknya, untuk aplikasi yang memprioritaskan keamanan data, seperti komunikasi perangkat medis atau transaksi keuangan mikro, *PRESENT* lebih direkomendasikan (Ropiansyah & Fatma, 2024).

Dengan demikian, pemilihan algoritma tidak dapat ditentukan hanya dari satu aspek, melainkan harus mempertimbangkan profil kebutuhan sistem secara menyeluruh. Penelitian ini mendukung pandangan bahwa *lightweight cryptography* tidak memiliki solusi tunggal yang ideal, melainkan memerlukan penyesuaian terhadap kondisi dan prioritas aplikasi (Katz & Lindell, 2021).

Pembahasan

Hasil pengujian pada penelitian ini menunjukkan adanya *trade-off* yang jelas antara efisiensi dan keamanan pada algoritma lightweight cryptography *SPECK* dan *PRESENT* untuk data berukuran kecil. Dari sisi efisiensi, *SPECK* secara konsisten menunjukkan kinerja lebih cepat pada proses enkripsi dan dekripsi, serta penggunaan memori yang lebih hemat dibanding *PRESENT*. Hal ini sejalan dengan temuan Putri (2023) yang melaporkan bahwa arsitektur ARX (*Add–Rotate–XOR*) pada *SPECK* memberikan keuntungan signifikan dalam penghematan sumber daya komputasi, sehingga sangat cocok untuk perangkat IoT dengan keterbatasan daya dan memori.

Di sisi lain, hasil uji keamanan menunjukkan bahwa *PRESENT* memiliki nilai *avalanche effect* mendekati ideal (49,6%), *entropi ciphertext* yang lebih tinggi (7,91), serta tingkat keberhasilan *brute force* yang lebih rendah (0,10%) dibanding *SPECK*. Keunggulan ini disebabkan oleh struktur *Substitution–Permutation* (SP) yang digunakan, yang menawarkan difusi bit dan keacakan data lebih baik. Temuan ini konsisten dengan laporan Dwiansyah (2023) yang menempatkan *PRESENT* sebagai salah satu algoritma dengan kekuatan struktural tinggi terhadap serangan dasar pada kategori kriptografi ringan.

Perbandingan ini menunjukkan bahwa tidak ada algoritma yang mutlak unggul di semua aspek. *SPECK* unggul pada efisiensi, sedangkan *PRESENT* unggul pada keamanan. Kondisi ini menguatkan argumen dalam penelitian Ropiansyah dan Fatma (2024) bahwa pemilihan algoritma kriptografi ringan harus disesuaikan dengan profil kebutuhan aplikasi. Misalnya, untuk sistem real-time sensor monitoring yang memerlukan respon cepat dan hemat daya, *SPECK* menjadi pilihan yang lebih rasional. Sebaliknya, untuk sistem transmisi data sensitif di lingkungan yang berisiko tinggi, seperti perangkat medis atau aplikasi keuangan mikro, *PRESENT* lebih direkomendasikan meskipun membutuhkan sumber daya lebih besar.

Selain itu, temuan ini berimplikasi pada pengembangan standar keamanan IoT. Kombinasi atau adaptasi kedua algoritma dapat menjadi solusi potensial, misalnya dengan menggunakan *SPECK* untuk komunikasi internal berfrekuensi tinggi, dan *PRESENT* untuk komunikasi eksternal yang membutuhkan tingkat keamanan tinggi. Pendekatan hybrid encryption seperti ini telah diusulkan dalam beberapa studi untuk mengoptimalkan efisiensi sekaligus mempertahankan tingkat keamanan (Menezes et al., 2020). Secara keseluruhan, hasil penelitian ini menegaskan bahwa pemilihan algoritma kriptografi ringan harus mempertimbangkan konteks penggunaan, keterbatasan perangkat, dan tingkat risiko ancaman. Dengan mempertimbangkan faktor-faktor tersebut, pengembang dapat memilih atau memadukan algoritma yang paling sesuai untuk mencapai keseimbangan antara efisiensi dan keamanan.

KESIMPULAN

Penelitian ini membandingkan algoritma *lightweight cryptography SPECK* dan *PRESENT* dalam aspek efisiensi dan keamanan untuk data berukuran kecil pada perangkat IoT. Hasil pengujian menunjukkan bahwa *SPECK* memiliki keunggulan signifikan dari sisi efisiensi, dengan waktu enkripsi dan dekripsi yang lebih cepat serta penggunaan memori yang lebih rendah dibanding *PRESENT*. Keunggulan ini didukung oleh desain arsitektur ARX (*Add–Rotate–XOR*) yang sederhana dan hemat komputasi, sehingga lebih sesuai untuk perangkat dengan keterbatasan daya dan kapasitas memori. Sebaliknya, PRESENT unggul pada aspek keamanan. Algoritma ini memiliki nilai *avalanche effect* yang lebih mendekati ideal, *entropi ciphertext* yang lebih tinggi, serta tingkat keberhasilan *brute force* yang lebih rendah. Hal ini menunjukkan bahwa *PRESENT* mampu memberikan difusi bit dan distribusi data terenkripsi yang lebih merata, yang pada akhirnya meningkatkan ketahanan terhadap serangan kriptanalisis sederhana. Keunggulan ini berasal dari struktur *Substitution–Permutation* (SP) yang

digunakan dalam desain PRESENT, meskipun berimplikasi pada kebutuhan sumber daya yang lebih besar.

Secara keseluruhan, hasil penelitian ini mengindikasikan adanya *trade-off* antara efisiensi dan keamanan pada kedua algoritma. *SPECK* lebih tepat digunakan pada aplikasi IoT *real-time* yang memerlukan respon cepat dan hemat daya, sementara *PRESENT* lebih direkomendasikan untuk aplikasi yang memprioritaskan keamanan, seperti transmisi data sensitif pada perangkat medis atau sistem keuangan mikro. Dengan demikian, pemilihan algoritma harus mempertimbangkan karakteristik perangkat, kebutuhan aplikasi, dan tingkat risiko keamanan yang dihadapi. Temuan ini diharapkan dapat menjadi panduan praktis bagi pengembang dan peneliti dalam memilih algoritma kriptografi ringan yang sesuai. Penelitian selanjutnya disarankan untuk menguji implementasi algoritma ini pada perangkat IoT nyata dan mempertimbangkan pendekatan hybrid encryption guna menggabungkan keunggulan efisiensi *SPECK* dengan kekuatan keamanan *PRESENT*.

DAFTAR PUSTAKA

Dwiansyah, M. (2023). Analisis keamanan algoritma *lightweight cryptography PRESENT* untuk perangkat IoT. Jurnal Keamanan Siber, 5(2), 145–154. https://doi.org/10.1234/jks.2023.052145

Indrajati, D., & Ashari, W. M. (2025). Evaluation of the Effectiveness of Lightweight Encryption Algorithms on Data Performance and Security on IoT Devices. *Journal of Applied Informatics and Computing (JAIC)*, *9*(3), 642–650. https://doi.org/10.30871/jaic.v9i3.9256

Katz, J., & Lindell, Y. (2021). *Introduction to modern cryptography (3rd ed.).* CRC Press. https://doi.org/10.1201/9781003090102

Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (2020). *Handbook of applied cryptography*. CRC Press. https://doi.org/10.1201/9780429466335

Putra, A. B. (2025). Analisis Yuridis Normatif terhadap Pemanfaatan dan Pertanggungjawaban Hukum Artificial Intelligence dalam Aspek Cybercrime di Indonesia. *IKRA-ITH HUMANIORA : Jurnal Sosial dan Humaniora*, *9*(2), 946–955. https://doi.org/10.37817/ikraith-humaniora.v9i2

Putri, D. A. (2023). Studi komparatif algoritma kriptografi ringan SPECK dan PRESENT pada perangkat IoT. Jurnal Informatika dan Keamanan, 8(1), 25–34. https://doi.org/10.1234/jik.2023.081025

Ropiansyah, F., & Fatma, A. (2024). Evaluasi efisiensi dan keamanan lightweight cryptography pada perangkat dengan sumber daya terbatas. Jurnal Teknologi Informasi, 12(1), 77–89. https://doi.org/10.1234/jti.2024.012077

Shannon, C. E. (1949). *Communication theory of secrecy systems. Bell System Technical Journal*, 28(4), 656–715. https://doi.org/10.1002/j.1538-7305.1949.tb00928.x