Terjemahan dari 10.22146/jnteti.v13i3.11824

Mengamankan RFID di Jaringan IoT dengan Pendekatan Kriptografi AES dan ECDH Ringan

Robby Kurniawan Harahap[1], Alief Vickry Thaha Maulidzart[1], Antonius Irianto Sukowati[2], Dyah Nur'ainingsih[1], Widyastuti[1], Desy Kristyawati[1]

[1] Program Studi Teknik Elektro, Universitas Gunadarma, Depok, Jawa Barat 16424, Indonesia
[2] Program Studi Teknik Elektro, Universitas Cendekia Abditama, Tangerang, Banten 15810, Indonesia

[Diserahkan: 7 Februari 2024, Direvisi: 9 April 2024, Diterima: 5 Juli 2024]
Penulis Korespondensi: Robby Kurniawan Harahap (email: robby_kurniawan@staff.gunadarma.ac.id)

INTISARI — *Radio frequency identification* (RFID) yang diintegrasikan ke dalam *internet of things* (IoT) sering kali menimbulkan permasalahan keamanan dan privasi karena kerentanannya terhadap serangan. Penelitian ini mengusulkan model kriptografi ringan yang dirancang untuk diimplementasikan pada jaringan dengan sumber daya yang terbatas. Tujuannya adalah untuk mengatasi ancaman keamanan sekaligus mengakomodasi kebutuhan memori, daya, dan ukuran yang terbatas. Gabungan algoritma *Advanced Encryption Standard* (AES) 126 bit yang dimodifikasi dengan kunci kriptografi *elliptic curve Diffie-Hellman* (ECDH) 256 bit digunakan untuk mengembangkan kriptografi ringan guna mengamankan data RFID. Implementasinya dilakukan menggunakan bahasa pemrograman Python di Jupyter Notebook, dengan RFID yang beroperasi pada 13,56 MHz. Metodologi yang digunakan adalah mengambil data RFID melalui program tambahan dan menyamakan kunci ECDH untuk enkripsi dan dekripsi. Pengujian enkripsi dan dekripsi menunjukkan tingkat keberhasilan yang tinggi dengan akurasi mencapai 99,9%. Upaya enkripsi pertama memerlukan waktu 85,125 ms, sedangkan enkripsi kedua selesai lebih cepat, dengan waktu 65,95 ms, yang menunjukkan peningkatan efisiensi. Ukuran enkripsi *file* rata-rata 29,875 bita (*byte*) untuk percobaan pertama dan 30,1 bita untuk percobaan berikutnya. Penelitian ini terbatas pada evaluasi algoritma dan belum diimplementasikan pada perangkat keras. Namun, kriptografi hibrida yang diusulkan menawarkan manfaat yang signifikan untuk menjaga kerahasiaan data RFID di lingkungan IoT. Enkripsi data *unique identifier* (UID) yang cepat, efisien, dan ringkas memastikan keamanan yang lebih baik, sehingga mengatasi masalah kritis yang terkait dengan jaringan IoT berkemampuan RFID.

KATA KUNCI — *Advanced Encryption Standard* (AES), Kriptografi *Elliptic Curve*, Jaringan IoT, Kriptografi Ringan, Keamanan RFID.

I. PENDAHULUAN

Dalam dunia digital yang serba cepat, teknologi *radio frequency identification* (RFID) berperan penting dalam berbagai sektor, seperti manajemen rantai pasok dan kontrol akses. Namun, terlepas dari berbagai manfaatnya, muncul masalah terkait kerentanannya terhadap serangan siber dan potensi ancaman terhadap privasi pengguna. RFID, komponen kunci dari *automatic identification and data capture* (AIDC), memungkinkan pembacaan beberapa *tag* RFID secara simultan melalui transmisi data nirkabel [1], [2], yang memfasilitasi pendeteksian, identifikasi, pelacakan, dan penelusuran berbagai objek [2]. *Tag* RFID mengidentifikasi dirinya sendirinya menggunakan pembaca RFID ketika dipicu oleh sinyal perangkat yang kompatibel [3]. Mengingat peran RFID dalam mentransmisikan informasi, keamanan informasi di dalamnya menjadi sangat penting. Keamanan ini mencakup aspek-aspek seperti komunikasi (enkripsi, autentikasi, dan otorisasi), protokol yang aman, perutean, dan keamanan jaringan [4].

Teknologi RFID telah menjadi bagian terpadu dari *internet of things* (IoT), mendorong konektivitas tanpa batas di antara objek sehari-hari dan memfasilitasi pertukaran informasi tanpa campur tangan manusia. Kolaborasi ini tidak hanya meningkatkan efisiensi operasional, tetapi juga memungkinkan pemahaman yang lebih baik terhadap aplikasi tertentu, seperti sistem pelacakan inventaris, dengan mengumpulkan dan menganalisis data [5]. Namun, mengamankan RFID dalam sistem IoT tetap menjadi tantangan yang signifikan. Kunci keamanan, yang sangat penting untuk mengendalikan operasi terenkripsi seperti enkripsi dan dekripsi data, berperan penting dalam menentukan tingkat keamanan sistem atau perangkat. Model enkripsi ringan yang dirancang untuk perangkat dengan sumber daya terbatas, seperti *tag* RFID, didesain untuk beroperasi secara efisien di lingkungan dengan sumber daya terbatas seperti memori, daya, dan ukuran. Selain itu *soft password* harus dapat menyeimbangkan antara keamanan, biaya, dan kinerja [6].

Dalam IoT, yang ditandai dengan tingginya mobilitas dan rendahnya keterbatasan penyimpanan, kecepatan eksekusi dan efisiensi penyimpanan sangat penting untuk mendukung fungsionalitas perangkat. Oleh karena itu, enkripsi yang ringan menjadi sangat penting. Sebuah penelitian telah menggarisbawahi pentingnya keamanan terhadap serangan yang membahayakan infrastruktur sistem IoT seperti sistem *supervisory control and data acquisition* (SCADA) [7]. Algoritma enkripsi yang ringan bertujuan untuk meminimalkan beban komputasi serta memastikan proses enkripsi dan dekripsi yang cepat tanpa perlu menambah sumber daya perangkat keras. Selain itu, kriptografi ringan dapat menghasilkan *file* enkripsi yang lebih kecil, sehingga mengatasi kendala penyimpanan perangkat IoT dan memungkinkan pengoperasian yang efisien di tengah mobilitas dan keterbatasan sumber daya.

Penelitian sebelumnya tentang kriptografi ringan pada perangkat RFID yang menggunakan *Advanced Encryption Standard* (AES) dan *elliptic curve Diffie-Hellman* (ECDH) menunjukkan adanya ketertarikan yang makin besar untuk mengembangkan solusi yang efisien dan aman bagi jaringan IoT [8]. Kerentanan perangkat IoT terhadap berbagai serangan

menekankan perlunya protokol keamanan yang ringan [9]. Kriptografi ringan, ditandai dengan skema enkripsi yang sederhana dengan kompleksitas komputasi yang rendah, menawarkan solusi yang tepat untuk perangkat yang memiliki sumber daya terbatas seperti *tag* RFID [10]. Meskipun AES telah banyak digunakan untuk komunikasi yang aman di antara perangkat IoT, keterbatasannya di lingkungan yang sangat terbatas telah mendorong para peneliti untuk mengeksplorasi *block cipher* baru yang dioptimalkan [11]. Selain itu, penggunaan ECDH dan mekanisme autentikasi ringan telah diusulkan untuk meningkatkan keamanan perangkat IoT yang terbatas [12]. Secara keseluruhan, penelitian ini menunjukkan adanya pergeseran dalam pengembangan algoritma kriptografi ringan yang disesuaikan dengan batasan spesifik perangkat IoT dan persyaratan keamanan, terutama dalam sistem RFID.

Penelitian yang berfokus pada penerapan kriptografi untuk perangkat RFID menekankan tantangan dalam mencapai keseimbangan yang tepat antara biaya, kinerja, dan keamanan dalam kriptografi ringan [13]. Selain itu, sumber daya komputasi dan memori yang terbatas, terutama dalam konteks teknologi RFID pasif, menjadi kendala bagi sistem keamanan RFID pada IoT [14]. Tantangan-tantangan ini menekankan perlunya penelitian lebih lanjut tentang autentikasi, enkripsi, dan protokol keamanan untuk mengatasi sumber daya yang terbatas yang menghambat fungsi keamanan standar seperti enkripsi AES. Penerapan metode enkripsi tradisional dapat menjadi tantangan karena adanya keterbatasan sumber daya, sehingga membutuhkan protokol autentikasi yang ringan dan sangat ringan yang cocok untuk penerapan RFID berbiaya rendah. Penerapan algoritma AES pada sistem RFID menjamin keamanan transmisi data.

Terlepas dari kendala yang ada, masih ada peluang untuk mendesain dan mengimplementasikan kriptografi ringan pada perangkat RFID. Makalah ini menggunakan perangkat lunak dan metode simulasi untuk mendesain dan mengimplementasikan kriptografi ringan di perangkat RFID untuk jaringan IoT. Proses ini melibatkan pengintegrasian algoritma kriptografi ringan ke dalam perangkat keras dan perangkat lunak untuk *tag* RFID [15]. Hasil yang ditargetkan berfokus pada akurasi rata-rata, kecepatan, dan ukuran data yang signifikan. Penelitian yang dilakukan diharapkan dapat memberikan kontribusi berharga bagi pengimplementasian kriptografi sebagai perangkat RFID yang aman dan berkualitas tinggi di jaringan IoT.

Makalah ini disusun sebagai berikut: Bagian II, Bahan dan Metodologi, membahas konsep kriptografi ringan dan metode yang digunakan. Bagian III memaparkan hasil yang dicapai melalui simulasi dan pengujian. Diskusi tentang hasil dan perbandingan dengan literatur yang ada disediakan di Bagian IV. Selanjutnya, makalah ini diakhiri dengan rangkuman dari hasil penelitian dan implikasinya.

II. BAHAN DAN METODOLOGI

A. GAMBARAN UMUM IoT

IoT telah mengantarkan dunia pada era baru dominasi penelitian dengan aplikasi meliputi logistik dan transportasi cerdas, kesehatan cerdas, tata kelola lingkungan cerdas (*smart environment*), infrastruktur cerdas (termasuk kota, rumah, kantor, dan mal cerdas serta Industri 4.0), dan pertanian pintar. IoT merupakan jaringan objek yang saling terhubung, masing-masing dengan kode unik yang memfasilitasi pengumpulan dan berbagi data melalui internet, secara mandiri ataupun dengan interaksi manusia [13]. Perangkat IoT dapat dikategorikan secara luas ke dalam dua kelompok: perangkat yang memiliki sumber daya yang melimpah, seperti server, komputer pribadi, tablet, dan ponsel pintar; dan perangkat yang memiliki sumber daya terbatas, seperti sensor industri, *node* sensor, *tag* RFID, dan aktuator.

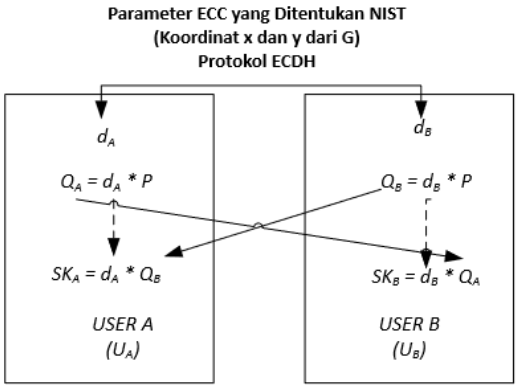
Di berbagai domain, berbagai implementasi IoT telah dilakukan, termasuk di antaranya adalah *maritime IoT* (MIoT). MIoT mencakup jaringan perangkat, sensor, dan sistem yang saling terhubung yang digunakan dalam industri maritim untuk aplikasi seperti pelacakan kapal, logistik, manajemen rantai pasokan, dan operasi maritim [14]. Menggabungkan IoT dan RFID meningkatkan keamanan kendaraan dan membantu mencegah pencurian dalam sistem keamanan kendaraan [16]. Fokus utama IoT terletak pada aktivitas pemantauan, seperti pemantauan pemberian makan hewan [17]. Namun, keamanan siber menjadi tantangan yang signifikan dalam penelitian sistem IoT, yang mencakup kerahasiaan, integritas data, autentikasi, dan otorisasi. Sistem keamanan IoT sering kali menggunakan pendekatan kriptografi untuk menangani aspek-aspek keamanan siber ini.

B. KRIPTOGRAFI RINGAN

Kriptografi ringan berperan penting dalam memastikan komunikasi yang aman, terutama untuk perangkat dengan sumber daya terbatas seperti yang ditemukan pada IoT. Sebagaimana diidentifikasi dalam [13], tantangan penelitian dalam keamanan IoT meliputi terbatasnya memori (register, RAM, ROM), berkurangnya daya komputasi, kecilnya area perakitan fisik, rendahnya daya baterai, dan kebutuhan respons secara *real-time*. Keterbatasan ini bersumber dari kecilnya ukuran dan terbatasnya sumber daya pada perangkat IoT. Standar kriptografi konvensional, ketika diterapkan pada perangkat IoT, terutama pada aplikasi *real-time* seperti RFID, sering kali kesulitan dalam memberikan respons yang cepat dan akurat dengan tetap memberikan keamanan menggunakan sumber daya yang tersedia. Namun, kriptografi ringan menawarkan solusi dengan memanfaatkan fitur-fitur seperti *small memory footprint*, daya pemrosesan rendah, konsumsi energi minimal, dan respons secara *real-time*, bahkan pada perangkat dengan sumber daya yang terbatas.

Dalam IoT, kriptografi ringan primitif mencakup empat jenis: *lightweight block cipher* (LWBC), *lightweight stream ciphers* (LWSC), *lightweight hash functions* (LWHF), dan *elliptic curve cryptography* (ECC) [18]. ECC, sebuah bentuk kriptografi kunci publik yang lebih baru, menawarkan *key agreement*, *signature*, dan pembuatan kunci yang cepat, meskipun tidak secara eksplisit menyediakan mekanisme enkripsi. Namun demikian, ECC menawarkan keuntungan seperti penggunaan memori yang minimal, konsumsi energi yang lebih rendah, faktor daya yang lebih optimal, dan kecepatan yang lebih baik.

AES, yang terkenal dengan tingkat keamanannya yang tinggi, merupakan metode enkripsi yang populer karena keefektifan, kesederhanaan, dan dukungan platformnya yang luas. AES menawarkan panjang kunci 128, 192, dan 256 bit [8], sehingga menjadi pilihan serbaguna untuk enkripsi. Namun, penelitian ini menggunakan kunci 128 bit dan 16 proses perulangan. Setiap proses enkripsi dan dekripsi membutuhkan sebuah kunci yang dapat berupa kata atau frasa; kunci-kunci ini merupakan bagian integral dari metode kriptografi untuk keamanan data. Diadopsi sebagai *Federal Information Processing Standard* (FIPS) oleh National Institute of Standards and Technology (NIST) pada tahun 2001 [19],



Gambar 1. Konsep protokol ECDH.

enkripsi AES diakui secara luas karena reliabilitasnya. Protokol ECDH memfasilitasi skema *key agreement*, yang memungkinkan pihak A dan B untuk membuat kunci rahasia bersama untuk algoritma kunci privat. Melalui pertukaran informasi publik, kedua belah pihak dapat menghasilkan kunci rahasia bersama menggunakan informasi publik dan privat masing-masing. Pihak ketiga tidak dapat mengetahui kunci rahasia bersama dari informasi yang tersedia untuk umum tanpa mengetahui detail pribadi kedua belah pihak [20]. Sebuah penelitian melakukan aplikasi sederhana dari kriptografi ringan, mengimplementasikan *cipher* ringan pada perangkat *application-specific integrated circuit* (ASIC) dan *field-programmable gate array* (FPGA) untuk mengoptimalkan penggunaan energi [21].

Varian ECC dari protokol Diffie-Hellman adalah ECDH, yang didesain untuk *key agreement* (atau pembuatan kunci bersama) antara dua pengguna. Penelitian sebelumnya telah menyoroti bahwa standar algoritma ECDH rentan terhadap serangan *man-in-the-middle*, yaitu penyerang dapat membaca dan memodifikasi semua pesan yang dikirim tanpa menargetkan pengguna yang sah [22], [23]. Dua solusi diusulkan untuk memperkuat algoritma ECDH terhadap serangan ini. Pertama, autentikasi kunci publik pengguna memastikan validasi kunci publik statis pengguna. Kedua, kedua belah pihak menghasilkan kunci publik sementara untuk setiap sesi komunikasi. Pendekatan ini memungkinkan adanya *perfect forward secrecy* (PFS) dan mengurangi kompleksitas algoritma, sehingga tidak memerlukan perhitungan autentikasi tambahan. Komputasi kriptografi untuk setiap lapisan pada Gambar 1 melibatkan beberapa algoritma atau protokol [24], [25]. Setiap pengguna U_A dan U_B mulai membuat kunci bersama menggunakan parameter ECC standar. Selanjutnya, setiap pengguna U_A dan U_B menghasilkan kunci publiknya menggunakan titik dasar generator (G) dan kunci privat. Dalam [26], algoritma untuk pertukaran kunci ECDC diperiksa dan dievaluasi menggunakan pustaka PyCryptodome dan skema enkripsi terintegrasi *elliptic curve*.

C. METODE

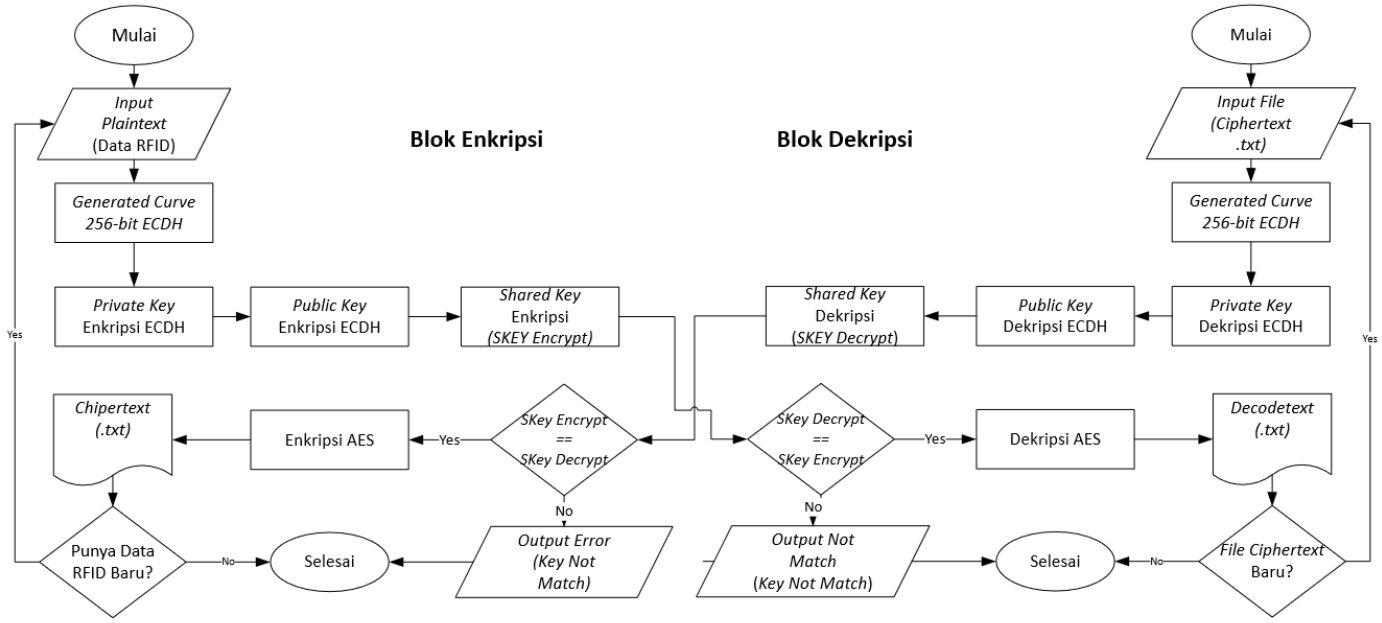
Penelitian ini menggunakan konsep kriptografi hibrida yang telah diperkenalkan sebelumnya [27]. Konsep ini menggabungkan modifikasi algoritma AES dengan kunci ECDH. Pendekatan ini bertujuan untuk mengembangkan kriptografi yang ringan dalam rangka meningkatkan keamanan data RFID. Proses enkripsi dan dekripsi menggunakan AES sebagai kriptografi simetris dengan panjang kunci 128 bit, 192 bit, dan 256 bit serta ukuran paket 128 bit, membuat sistem enkripsi AES sangat mudah beradaptasi. Oleh sebab itu, teknologi enkripsi AES diimplementasikan secara luas pada perangkat keras dan perangkat lunak, sementara ECDH berfungsi sebagai kriptografi asimetris [28]. ECDH menghasilkan kunci menggunakan *elliptic curve* [29]. Pada skema *master agreement*, semua pihak yang terlibat dalam suatu komunikasi harus memberikan kontribusi data atau informasi untuk menghasilkan kunci sesi bersama [30]. Penelitian ini menggunakan bahasa pemrograman Python melalui Jupyter Notebook di Anaconda Navigator, dengan menggunakan dua buah laptop. Satu laptop didedikasikan untuk menguji blok enkripsi dan satu laptop lainnya untuk menguji blok dekripsi. Digunakan pula skema enkripsi hibrida dengan skema pertukaran kunci ECDH.

Gambar 2 mengilustrasikan metode untuk enkripsi dan dekripsi menggunakan AES dengan kunci ECDH. Metode ini terdiri atas dua blok, yaitu blok enkripsi dan blok dekripsi. Langkah awal adalah membuat kunci privat untuk setiap blok menggunakan pustaka kurva 256 bit "brainpoolP256r1", diikuti dengan membuat kunci privat. Kunci privat yang dihasilkan pada blok enkripsi disimpan untuk kunci publik ECDH enkripsi dan untuk proses dekripsi AES pada blok dekripsi, dan sebaliknya. Hasil kunci privat dalam blok dekripsi disimpan untuk kunci publik ECDH dekripsi dan proses enkripsi AES dalam blok enkripsi. Menurut aturan ECDH, jika dua angka rahasia, "a" dan "b" (mewakili kunci privat U_A dan U_B), digabungkan dengan *elliptic curve* ECC yang memiliki titik generator (G), U_A PrivKey G, dan UB PrivKey G, nilai-nilai dapat bertukar melalui saluran yang tidak aman. Dengan demikian, kunci publik U_A dan U_B digunakan dalam (1).

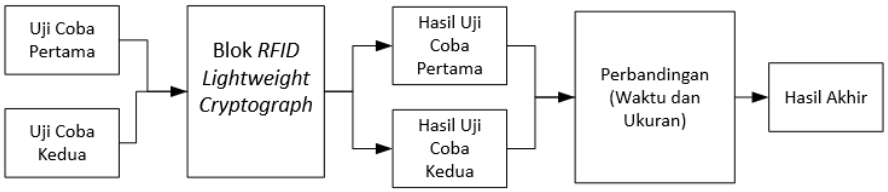
$$U_A\, PubKey \times U_B PrivKey = U_B PubKey \times U_A PrivKey. \quad (1)$$

Proses enkripsi AES melibatkan modifikasi prosedur standar dengan memasukkan kunci ECDH. Setiap putaran enkripsi terdiri atas empat langkah fungsional, yaitu *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey*. Langkah-langkah ini dioperasikan pada blok standar data AES, dengan masing-masing blok diwakili oleh matriks yang berisi 16 atau 128 bit. Setelah enkripsi berhasil, hasilnya disimpan dalam sebuah *file*. Proses enkripsi hibrida yang menggunakan kunci ECDH juga menghasilkan data terenkripsi yang disimpan dalam sebuah *file*. Selanjutnya, tahap analisis dilakukan, lalu hasil enkripsi (*ciphertext*) dan hasil dekripsinya disimpan dalam format *.txt*. Analisis ini dilakukan untuk mengevaluasi parameter seperti lama proses enkripsi, ukuran *file*, dan *runtime*. Selain itu, hasil enkripsi dan dekripsi dibandingkan untuk menjelaskan lebih lanjut keunggulan metode enkripsi hibrida, sebagaimana yang ditunjukkan pada Gambar 3.

Integrasi kunci AES dan ECDH dalam kerangka kriptografi hibrida menghasilkan solusi yang unggul untuk mengatasi masalah keamanan data RFID, dengan memanfaatkan keunggulan kriptografi simetris dan asimetris. Metode ini membentuk saluran komunikasi yang aman dengan menggabungkan pertukaran kunci ECDH selama enkripsi, sehingga mengurangi kemungkinan akses yang tidak sah atau penyadapan data. Selain itu, pengujian dan analisis yang ketat, yang mencakup penilaian durasi proses enkripsi, ukuran *file*, dan perbandingan *runtime*, menegaskan kemampuan dan efisiensi metode ini dalam meningkatkan langkah-langkah keamanan sekaligus mempertahankan efektivitas operasional. Evaluasi komprehensif ini, disertai dengan penggambaran metodologi penelitian secara terperinci pada Gambar 3, memberikan validasi yang meyakinkan tentang keampuhan dan integritas pendekatan yang diusulkan.



Gambar 2. Diagram alur kriptografi ringan AES dan ECDH.



Gambar 3. Metode penelitian.

III. HASIL

Pada tahap ini, temuan penelitian dipresentasikan, berbagai parameter dijelaskan, termasuk pengumpulan *unique identifier* (UID) RFID, hasil pengujian enkripsi dan dekripsi, waktu pemrosesan untuk enkripsi dan dekripsi, serta ukuran *file plaintext* asli, file *ciphertext* terenkripsi, dan file *decodetext*. Selain itu, perbandingan waktu pemrosesan dan ukuran *file* juga dianalisis. Penelitian ini melakukan dua percobaan yang berbeda, masing-masing menggunakan kunci ECDH yang berbeda. Percobaan awal melibatkan delapan item UID RFID (lima kartu RFID dan tiga gantungan kunci RFID), sedangkan percobaan kedua menggunakan sepuluh kartu UID RFID.

A. PENGUMPULAN DATA RFID

Pengumpulan data RFID melibatkan pengumpulan data UID dari *chip* RFID, yang kemudian digunakan sebagai *plaintext* untuk proses enkripsi. Data yang terkumpul disimpan dalam *file* dengan format *.txt*. Percobaan pertama, seperti yang dirinci pada Tabel I, menggunakan delapan *tag* RFID yang terdiri atas lima kartu RFID dan tiga kunci RFID. Setiap *tag* diberi delapan digit UID yang berbeda, yang berfungsi sebagai pengenal unik untuk pembaca data. Demikian pula, percobaan berikutnya, yang diuraikan dalam Tabel II, melibatkan sepuluh kartu RFID, dengan mempertahankan karakteristik *tag* dan UID yang sama dengan percobaan awal.

B. MENGUJI HASIL ENKRIPSI DAN RUNTIME ENKRIPSI

Selama pengujian enkripsi, data UID RFID (*plaintext*) diubah menjadi karakter yang tidak dapat dibaca, yang dikenal sebagai *ciphertext*. Penelitian ini menggunakan kurva 256 bit untuk kunci ECDH dan 128 bit untuk AES. Selanjutnya, lama proses enkripsi dihitung guna menentukan durasi yang diperlukan untuk mengubah *plaintext* ke *ciphertext*. Hasil untuk kunci enkripsi (ECDH) adalah 697831462505796649579499205152035665873002698782338187339985812442062651678 33, yang mewakili kunci ECDH dan digunakan dalam proses enkripsi pada percobaan awal. Setiap perbedaan antara kunci ECDH yang dihasilkan dalam enkripsi dan dekripsi menyebabkan adanya variasi pada *plaintext*.

Tabel III menyajikan hasil uji coba enkripsi hibrida awal yang dilakukan pada delapan RFID. Terlihat jelas bahwa enkripsi menghasilkan serangkaian karakter acak yang tidak dapat dibaca oleh manusia. Semua data UID RFID berhasil dienkripsi tanpa mengalami kesalahan. Pada percobaan pertama, proses enkripsi tercepat adalah 85 ms, yaitu enkripsi Kunci RFID 3 dengan UID B61D62AF. Sebaliknya, proses enkripsi yang paling lama berlangsung selama 106 ms, yaitu pada enkripsi RFID Card 3 dengan UID A8972327. Waktu enkripsi kumulatif pada percobaan pertama adalah 726 ms. Hasil kunci enkripsi (ECDH) adalah 580638978007529434362635958133229218956681109252703267608179494212712481002 7, yang mewakili kunci ECDH yang digunakan dalam proses enkripsi selama percobaan berikutnya. Setiap perbedaan antara kunci ECDH yang dihasilkan yang digunakan untuk enkripsi dan dekripsi dapat menyebabkan adanya variasi dalam *plaintext*.

Tabel IV menunjukkan hasil dari enkripsi hibrida awal yang dilakukan pada sepuluh RFID. Terbukti bahwa enkripsi menghasilkan urutan karakter acak yang tidak dapat dibaca oleh manusia. Semua data UID RFID dienkripsi secara efektif tanpa mengalami kesalahan. Pada percobaan kedua, proses enkripsi tercepat berlangsung selama 41 ms, yaitu pada

Jurnal Nasional Teknik Elektro dan Teknologi Informasi
Volume 13 Nomor 3 Agustus 2024

TABEL I
HASIL EKSPERIMEN PERTAMA PENGUMPULAN DATA UID RFID

ID RFID	UID RFID (*Plaintext*)
Kartu RFID 1	638FE118
Kartu RFID 2	1499C649
Kartu RFID 3	A8972327
Kartu RFID 4	83139205
Kartu RFID 5	E3670B04
Kunci RFID 1	A7C6C128
Kunci RFID 2	4C832949
Kunci RFID 3	B61D62AF

TABEL II
HASIL EKSPERIMEN KEDUA PENGUMPULAN DATA UID RFID

ID RFID	UID RFID (*Plaintext*)
Kartu RFID 6	F3F2BADF
Kartu RFID 7	0DD60D04
Kartu RFID 8	AB95B9DF
Kartu RFID 9	41C755FF
Kartu RFID 10	979ABBDF
Kartu RFID 11	271ABADF
Kartu RFID 12	D9CDBBDF
Kartu RFID 13	A7E6B8DF
Kartu RFID 14	00A8BADF
Kartu RFID 15	22F7BBDF

enkripsi Kartu RFID 7 dengan UID 0DD60D04. Sebaliknya, proses enkripsi terlama berlangsung hingga 80 ms, yang melibatkan Kartu RFID 13 dengan UID A7E6B8DF dan Kartu RFID 15 dengan UID 22F7BBDF. Waktu enkripsi kumulatif pada upaya awal mencapai 669 ms.

C. MENGUJI HASIL DEKRIPSI DAN RUNTIME DEKRIPSI

Uji dekripsi mengubah *file ciphertext* yang terdiri atas karakter yang tidak dapat dibaca menjadi *decodetext*, sehingga dapat dibaca oleh manusia. Dengan menggunakan kurva 256 bit untuk kunci ECDH dan 128 bit untuk AES, proses dekripsi bertujuan untuk memastikan waktu yang diperlukan untuk mengubah *ciphertext* menjadi teks yang dapat diuraikan. *File* ini mengalami dekripsi dengan memasukkan file *.txt* terenkripsi, dengan teks yang dihasilkan disimpan dalam format *.txt*. Kunci dekripsi (ECDH) yang digunakan dalam percobaan pertama adalah 48821956027739608857925916805825085142072912 266785442189318763110805727647718. Jika kunci ECDH tidak sama dengan kunci enkripsi, teks hasil dekripsi akan berbeda dengan teks asli. Tabel V menunjukkan status waktu dekripsi yang berhasil pada percobaan pertama. Waktu dekripsi tercepat tercatat selama 66 ms pada Kunci RFID 3, sedangkan Kartu RFID 1 memiliki waktu dekripsi terlama, yaitu 90 ms. Waktu pemrosesan dekripsi kumulatif dalam percobaan awal mencapai 636 ms. Proses dekripsi secara otomatis menyimpan teks yang dihasilkan dalam format *decrypted [date of decryption].txt*.

Kunci ECDH yang digunakan untuk proses dekripsi pada percobaan kedua diwakili oleh nilai 68723788531881136467441998490530808783086550182713 89059513855619855053573996. Sangat penting bagi kunci ini untuk mencocokkan kunci yang dihasilkan selama enkripsi; jika tidak, dapat terjadi perbedaan antara *ciphertext* dan *plaintext*. Tabel VI memberikan informasi tentang lama uji dekripsi selama percobaan kedua. Kartu RFID 11 menunjukkan waktu dekripsi tercepat, yaitu 40 ms; sedangkan Kartu RFID 9

TABEL III
HASIL UJI ENKRIPSI PADA PERCOBAAN PERTAMA

ID RFID	*Ciphertext*	Waktu Enkripsi (ms)
Kartu RFID 1	Encrypting... ChiperText : [VòØõÿðüË□□Ï□ËÊ□	86
Kartu RFID 2	Encrypting... ChiperText : öÕðü □ø□□□□□□È□□	95
Kartu RFID 3	Encrypting... ChiperText : Ü½¨□□E□□ÏÈ□□□□Ï□	106
Kartu RFID 4	ChiperText : 9V§□□□©ŪĮĘ□□□□Į□ Encrypting...	98
Kartu RFID 5	Encrypting... ChiperText : ¯V □G□□ÇÏÈ□□□□Ï□	101
Kunci RFID 1	Encrypting... ChiperText : ÜÀÃRðEü□ÏÈ□□□□Ï□	85
Kunci RFID 2	Encrypting... ChiperText : □Ú¤□ÿÎð□ÏÈ□□□□Ï□	94
Kunci RFID 3	Encrypting... ChiperText : A□§□□□□^ÏÈ□□□□Ï□	61
Keseluruhan waktu enkripsi		726

TABEL IV
HASIL UJI ENKRIPSI PADA PERCOBAAN KEDUA

ID RFID	*Ciphertext*	Waktu Enkripsi (ms)
Kartu RFID 6	Encrypting... ChiperText : □V□□Ù□□F□□□Ê□□□È	65
Kartu RFID 7	Encrypting... ChiperText : Â×□X□Æ□Å□□□Ê□□□È	41
Kartu RFID 8	Encrypting... ChiperText : Ü þ£ü□□F□□□Ê□□□È	60
Kartu RFID 9	Encrypting... ChiperText : □¾□□¥□□§□□□Ê□□□È	69
Kartu RFID 10	Encrypting... ChiperText : ùÀþßD□□F□□□Ê□□□È	75
Kartu RFID 11	Encrypting... ChiperText : □ÀñßÙ□□F□□□Ê□□□È	70
Kartu RFID 12	Encrypting... ChiperText : □±□□D□□F□□□Ê□□□È	70
Kartu RFID 13	Encrypting... ChiperText : ÜÀ¨Xð□□F□□□Ê□□□È	80
Kartu RFID 14	Encrypting... ChiperText : Â□ÛöÙ□□F□□□Ê□□□È	59
Kartu RFID 15	Encrypting... ChiperText : □^□□D□□F□□□Ê□□□È	80
Keseluruhan waktu enkrispsi		669

menunjukkan waktu dekripsi terlama, yaitu waktu 101 ms. Waktu pemrosesan dekripsi kumulatif untuk percobaan kedua mencapai 650 ms.

D. HASIL PENGUKURAN UKURAN FILE ASLI, FILE TERENKRIPSI, DAN FILE YANG DIDESKRIPSIKAN

Hasil pengukuran berfungsi untuk menentukan ukuran *file* asli (*plaintext*), *file* hasil enkripsi (*ciphertext*), dan *file* hasil dekripsi (*decodetext*). Tabel VII menampilkan ukuran *file plaintext*, *ciphertext*, dan *decodetext* dari percobaan awal. Ukuran *file plaintext* terkecil yang tercatat adalah 7 bita (*byte*), yang terdapat pada tiga buah *file* RFID. Sementara itu, ukuran terbesar adalah 8 bita, ditemukan pada lima *file* RFID, sehingga menghasilkan ukuran total rata-rata 7,625 bita. Terkait ukuran *file* enkripsi (*ciphertext*), Kartu RFID 2 dan Kartu RFID 3 memiliki *file* terbesar dengan ukuran masing-masing 31 bita,

TABEL V
HASIL UJI DEKRIPSI PADA PERCOBAAN PERTAMA

ID RFID	Nama *File* Dekripsi	Waktu Dekripsi (ms)
Kartu RFID 1	Deskrip 29-07-2023(40).txt	68
Kartu RFID 2	Deskrip 29-07-2023(24).txt	72
Kartu RFID 3	Deskrip 29-07-2023(09).txt	81
Kartu RFID 4	Deskrip 29-07-2023(17).txt	86
Kartu RFID 5	Deskrip 29-07-2023(23).txt	85
Kunci RFID 1	Deskrip 29-07-2023(31).txt	90
Kunci RFID 2	Deskrip 29-07-2023(39).txt	88
Kunci RFID 3	Deskrip 29-07-2023(36).txt	66
Keseluruhan waktu dekripsi		636

TABEL VI
HASIL UJI DEKRIPSI PADA PERCOBAAN KEDUA

ID RFID	Nama *File* Dekripsi	Waktu Dekripsi (ms)
Kartu RFID 6	Deskrip 30-07-2023(33).txt	60
Kartu RFID 7	Deskrip 30-07-2023(52).txt	50
Kartu RFID 8	Deskrip 30-07-2023(22).txt	71
Kartu RFID 9	Deskrip 30-07-2023(33).txt	101
Kartu RFID 10	Deskrip 30-07-2023(00).txt	75
Kartu RFID 11	Deskrip 30-07-2023(18).txt	40
Kartu RFID 12	Deskrip 30-07-2023(26).txt	55
Kartu RFID 13	Deskrip 30-07-2023(25).txt	65
Kartu RFID 14	Deskrip 30-07-2023(11).txt	60
Kartu RFID 15	Deskrip 30-07-2023(41).txt	73
Keseluruhan waktu dekripsi		650

sedangkan Kunci RFID 3 memiliki ukuran terkecil dengan 28 bita, dengan ukuran rata-rata 29,875 bita. Selain itu, ukuran *decodetext* untuk semua *file* RFID secara konsisten adalah 16 bita. Setelah menjumlahkan ukuran semua *file* yang diproses, ukuran keseluruhannya adalah 61 bita untuk *file* asli, 269 bita untuk file *ciphertext*, dan 128 bita untuk *decodetext*.

Tabel VIII menampilkan ukuran *file plaintext*, *file ciphertext*, dan *decodetext* yang dihasilkan dari percobaan awal. *File plaintext* terkecil berukuran 7 bita, yang tercatat dalam lima *file* RFID. Di sisi lain, yang terbesar, yaitu 8 bita, teramati dalam jumlah *file* RFID yang sama, sehingga menghasilkan ukuran total rata-rata 7,5 bita. Terkait ukuran *file ciphertext*, Kartu RFID 9 memiliki *file* terbesar dengan 32 bita, sedangkan Kartu RFID 15 memiliki ukuran terkecil dengan 28 bita, dengan rata-rata 30,1 bita. Selain itu, ukuran *decodetext* untuk semua file RFID tetap konsisten pada 16 bita. Dari keseluruhan proses, ukuran total adalah 75 bita untuk *file* asli, 301 bita untuk *ciphertext*, dan 160 bita untuk *decodetext*.

IV. PEMBAHASAN

A. PERBANDINGAN WAKTU ENKRIPSI DAN WAKTU DEKRIPSI

Perbandingan antara waktu pemrosesan enkripsi hibrida dan dekripsi hibrida merupakan aspek penting dalam mengevaluasi efisiensi metode kriptografi yang digunakan. Analisis ini meneliti waktu yang diperlukan untuk proses enkripsi dan dekripsi, menjelaskan kompleksitas transformasi UID RFID. Gambar 4 dengan jelas menunjukkan kontras temporal antara hasil enkripsi dan dekripsi dari percobaan awal. Data menunjukkan bahwa proses enkripsi biasanya memakan

TABEL VII
HASIL UKURAN *FILE* ASLI, *FILE* TERENKRIPSI, DAN *FILE* DEKRIPSI PADA EKSPERIMEN PERTAMA

ID RFID	Ukuran *File* Asli (Bita)	Ukuran *File Ciphertext* (Bita)	Ukuran *File Decodetext* (Bita)
Kartu RFID 1	7	30	16
Kartu RFID 2	8	31	16
Kartu RFID 3	8	31	16
Kartu RFID 4	8	29	16
Kartu RFID 5	8	30	16
Kunci RFID 1	7	30	16
Kunci RFID 2	7	30	16
Kunci RFID 3	8	28	16
Ukuran total	61	269	128
Rata-rata	7,625	29,875	16

TABEL VIII
HASIL UKURAN *FILE* ASLI, *FILE* TERENKRIPSI, DAN *FILE* DEKRIPSI PADA EKSPERIMEN KEDUA

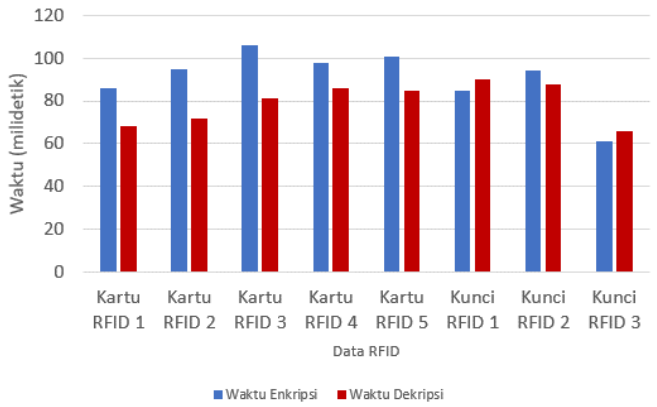
ID RFID	Ukuran *File* Asli (Bita)	Ukuran *File Ciphertext* (Bita)	Ukuran *File Decodetext* (Bita)
Kartu RFID 6	7	29	16
Kartu RFID 7	8	31	16
Kartu RFID 8	8	30	16
Kartu RFID 9	7	32	16
Kartu RFID 10	8	30	16
Kartu RFID 11	7	30	16
Kartu RFID 12	7	30	16
Kartu RFID 13	8	30	16
Kartu RFID 14	7	31	16
Kartu RFID 15	8	28	16
Ukuran total	75	301	160
Rata-rata	7,5	30,1	16

waktu lebih lama daripada dekripsi, seperti yang ditunjukkan oleh garis batang biru dan merah.

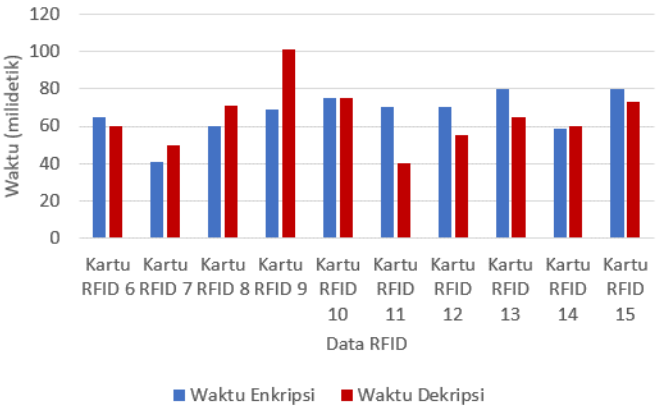
Waktu pemrosesan rata-rata yang diperoleh dari analisis ini memberikan bukti terukur tentang perbedaan temporal antara enkripsi dan dekripsi hibrida. Dengan rata-rata enkripsi 90,75 ms dan dekripsi 79,5 ms, terlihat jelas bahwa proses dekripsi secara umum berlangsung lebih cepat. Hal ini menggarisbawahi pentingnya pertimbangan temporal dalam menilai teknik kriptografi dan meningkatkan optimasi potensial untuk meningkatkan kinerja sistem. Gambar 5 secara visual mewakili kontras temporal antara enkripsi dan dekripsi pada percobaan kedua. Garis batang biru menggambarkan periode enkripsi, sedangkan garis batang merah menggambarkan waktu dekripsi. Tidak seperti percobaan pertama, perbedaan waktu antara enkripsi dan dekripsi pada percobaan kedua ini tidak terlalu mencolok, dengan kata lain waktu antara enkripsi dan dekripsi lebih merata. Hal ini menunjukkan adanya potensi keuntungan efisiensi dalam operasi kriptografi. Terdapat sedikit perbedaan antara kedua prosedur tersebut, dengan rata-rata waktu enkripsi 66,9 ms dan rata-rata waktu dekripsi 65 ms. Konvergensi durasi pemrosesan menunjukkan peluang untuk menyempurnakan metode kriptografi, yang mengarah pada proses enkripsi-dekripsi yang lebih seimbang dan efisien.

B. PERBANDINGAN UKURAN FILE ANTARA PLAINTEXT, CIPHERTEXT, DAN DECODETEXT

Uji perbandingan antara *plaintext*, *ciphertext*, dan *decodetext* berfungsi untuk mengevaluasi ukuran *file*, yang menunjukaan efisiensi dan efektivitas prosedur kriptografi.



Gambar 4. Perbandingan waktu enkripsi dan waktu dekripsi pada percobaan pertama.



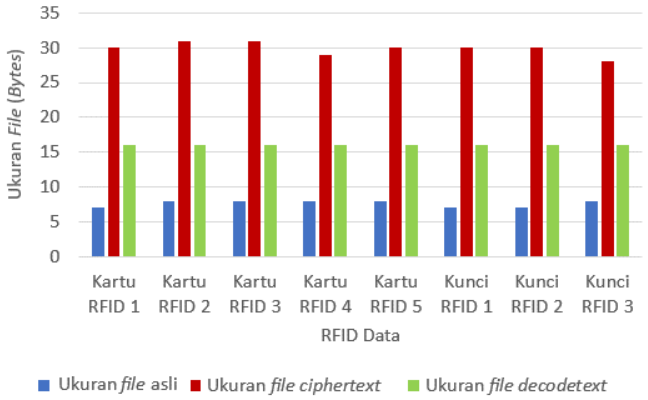
Gambar 5. Perbandingan waktu enkripsi dan waktu dekripsi pada percobaan kedua.

Penelitian ini meneliti dimensi relatif dari *file* awal, *file* yang dienkripsi (*ciphertext*), dan teks hasil dekripsi (*decodetext*). Gambar 6 menunjukkan ilustrasi terperinci mengenai perbandingan ukuran, dengan warna biru, merah, dan hijau, masing-masing mewakili *file* asli, *file* terenkripsi, dan teks hasil dekripsi.

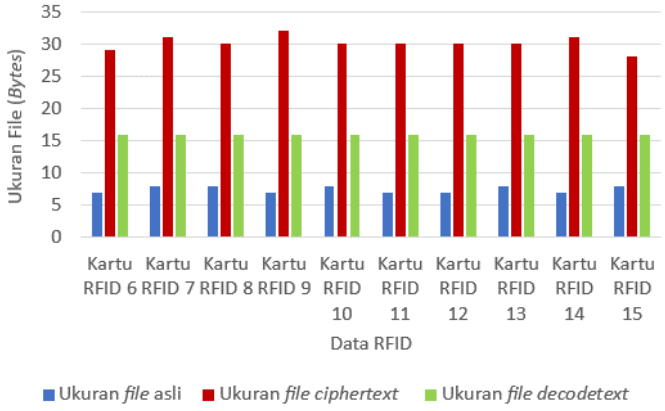
Setelah menganalisis perbandingan ukuran yang ditunjukkan pada Gambar 6, dapat terlihat pola pada ukuran *file* asli, terenkripsi, dan *file* hasil dekripsi. *File* terenkripsi menunjukkan peningkatan ukuran yang cukup besar karena penggunaan karakter yang kompleks dan acak dalam *ciphertext*, sehingga tidak dapat dimengerti oleh manusia. Sebaliknya, *file* asli menunjukkan ukuran terkecil di antara ketiga variabel tersebut. Perbedaan ini menggarisbawahi dampak signifikan dari enkripsi pada ukuran *file*, menekankan pertukaran antara keamanan informasi dan volume data.

Uji coba kedua menghasilkan analisis kuantitatif yang menunjukkan perbedaan rata-rata sebesar 22,25 bita antara ukuran *file* awal dan terenkripsi. Perbedaan ini menekankan pengaruh enkripsi pada volume data dan pentingnya mempertimbangkan ukuran *file* dalam proses kriptografi. Hasil dari uji komparatif ini memberikan wawasan yang signifikan untuk mengoptimalkan protokol kriptografi demi menyeimbangkan persyaratan keamanan dan faktor praktis, seperti batasan ukuran *file*.

Gambar 7 menunjukkan analisis komprehensif ukuran *file*, yang mencakup *file* asli, *file* terenkripsi (*ciphertext*), dan file teks hasil dekripsi (*decodetext*) dari percobaan awal. Setiap batang pada grafik menunjukkan ukuran *file*, dengan batang biru menunjukkan *file* asli, merah menunjukkan *file* terenkripsi, dan batang hijau menunjukkan *file* teks hasil dekripsi. *File* yang



Gambar 6. Perbedaan ukuran antara *plaintext, ciphertext,* dan *decodetext* pada percobaan pertama.



Gambar 7. Perbedaan ukuran antara *plaintext, ciphertext,* dan *decodetext* pada percobaan kedua.

dienkripsi menunjukkan peningkatan ukuran yang signifikan karena adanya karakter yang rumit dan acak, sehingga tidak mungkin dimengerti oleh manusia. Selain itu, ukuran *file* ini menunjukkan dampak prosedur kriptografi pada volume data karena penggabungan karakter yang rumit berkontribusi pada ukuran *file* yang semakin besar. Sebaliknya, *file* asli memiliki ukuran terkecil di antara ketiga variabel, menyoroti dampak substansial enkripsi pada dimensi *file*. Oleh karena itu, ukuran *file* ini berfungsi sebagai ukuran standar untuk perbandingan. Di sisi lain, prosedur dekripsi menghasilkan ukuran *file* yang mewakili hubungan antara *file* asli dan *file* terenkripsi, yang berfungsi sebagai representasi fisik dari transformasi kriptografi.

Setelah penelitian tambahan, yaitu percobaan kedua, dilakukan, ditemukan bahwa terdapat perbedaan rata-rata sebesar 22,6 bita antara *file* asli dan ukuran *file* terenkripsi. Hasil ini menekankan pengaruh enkripsi yang terus-menerus terhadap jumlah data, terlepas dari seringnya enkripsi dilakukan, sehingga menyoroti pentingnya mempertimbangkan ukuran *file* dalam kriptografi. Hasil ini juga menekankan pentingnya meningkatkan protokol kriptografi untuk mencapai keseimbangan yang harmonis antara prasyarat keamanan dan batasan praktis yang terkait dengan pertimbangan ukuran *file*.

Terdapat beberapa aspek utama yang dapat dibandingkan antara hasil penelitian ini dengan penelitian sebelumnya [8] untuk memastikan efektivitas dan keunggulan masing-masing teknik. Pertama, mengenai tujuan, meskipun kedua penelitian tersebut bertujuan untuk meningkatkan teknik kriptografi, penelitian ini secara eksplisit menargetkan kriptografi ringan untuk jaringan IoT. Fokus ini secara langsung menanggapi peningkatan permintaan untuk implementasi IoT yang aman

dalam lingkungan dengan sumber daya terbatas, menyarankan solusi yang lebih sesuai dan relevan dibandingkan dengan tujuan yang lebih luas untuk meningkatkan kekuatan AES [8].

Kedua, dalam hal pendekatan dan teknik yang digunakan, penelitian ini memperkenalkan pendekatan kriptografi hibrida baru dengan menggabungkan AES yang dimodifikasi dengan kunci ECDH yang dirancang khusus untuk lingkungan IoT. Sebaliknya, penelitian sebelumnya berfokus pada modifikasi transformasi *SubBytes* dan *ShiftRows* AES untuk meningkatkan kekuatannya [8]. Meskipun kedua pendekatan tersebut menunjukkan inovasi, pendekatan hibrida dalam penelitian ini menunjukkan solusi yang lebih holistik dengan mengatasi kendala dan persyaratan unik jaringan IoT, yang berpotensi menawarkan kemampuan beradaptasi dan efisiensi yang lebih baik dalam implementasi di dunia nyata.

Terakhir, terlihat dari metrik kinerja pada Tabel I bahwa penelitian ini mencapai tingkat keberhasilan yang tinggi, yaitu 99,9%, dalam proses enkripsi dan dekripsi, dengan waktu enkripsi berkisar antara 85,125 ms hingga 65,95 ms (Tabel I). Meskipun penelitian sebelumnya menunjukkan efek penurunan sebesar 57,81% untuk algoritma AES yang dimodifikasi [8], perbandingan kinerja waktu enkripsi menunjukkan bahwa penelitian ini menawarkan kecepatan pemrosesan yang lebih cepat. Selain itu, ukuran enkripsi *file* pada kedua penelitian tersebut sebanding, menunjukkan efisiensi yang sama dalam penanganan data. Oleh karena itu, berdasarkan aspek-aspek ini, penelitian ini menyajikan solusi yang lebih sesuai, efisien, dan berpotensi lebih unggul untuk mengamankan jaringan IoT dibandingkan dengan temuan dari penelitian sebelumnya [8].

V. KESIMPULAN

Dari percobaan yang telah dilakukan, dapat disimpulkan bahwa evaluasi prosedur enkripsi dan dekripsi memberikan hasil yang menjanjikan dalam memajukan kriptografi ringan. Keampuhan penggunaan kunci AES dan ECDH ditunjukkan dengan tingkat keberhasilan yang mengesankan, yaitu 99,9%, baik dalam pengujian enkripsi maupun dekripsi. Berkurangnya waktu pemrosesan dari percobaan awal, yang membutuhkan 85,125 ms, ke iterasi berikutnya, yang membutuhkan 65,95 ms, menandakan adanya peningkatan efisiensi. Meskipun menggunakan metodologi enkripsi dan dekripsi manual, temuan ini menggarisbawahi potensi kriptografi ringan untuk jaringan IoT, yang menawarkan pengambilan data yang cepat dan ukuran *file* yang ringkas.

Di masa depan, terdapat peluang yang signifikan untuk meningkatkan penelitian ini, yaitu dengan mengimplementasikan kriptografi ringan pada platform perangkat keras, terutama dengan memanfaatkan mikrokontroler. Integrasi implementasi berbasis perangkat keras secara substansial dapat meningkatkan efisiensi dan penerapan algoritma kriptografi dalam skenario praktis. Selain itu, pengembangan penelitian di masa depan dapat mengeksplorasi cara untuk mengoptimalkan efisiensi waktu pemrosesan dan meminimalkan ukuran *file*. Para peneliti dapat mempelajari lebih dalam tentang kriptografi ringan dan relevansinya dalam lanskap teknologi yang sedang berkembang, dengan memanfaatkan wawasan yang diperoleh dari penelitian ini. Kemajuan tersebut pada akhirnya akan meningkatkan transmisi data yang aman dan efisien dalam jaringan IoT.

KONFLIK KEPENTINGAN

Penulis menyatakan bahwa tidak terdapat konflik kepentingan.

KONTRIBUSI PENULIS

Konseptualisasi, Robby Kurniawan Harahap; metodologi, Robby Kurniawan Harahap; perangkat lunak, Alief Vickry Thaha Maulidzart; penulisan—penyiapan draf awal, Antonius Irianto Sukowati dan Dyah Nur'ainingsih; penulisan—penelaahan dan penyuntingan, Widyastuti dan Desy Kristyawati.

REFERENSI

[1] A. Haibi dkk., "Systematic mapping study on RFID technology," *IEEE Access*, vol. 10, hal. 6363–6380, Jan. 2022, doi: 10.1109/ACCESS.2022.3140475.

[2] W.C. Tan dan M.S. Sidhu, "Review of RFID and IoT integration in supply chain management," *Oper. Res. Perspect.*, vol. 9, hal. 1–17, Feb. 2022, doi: 10.1016/j.orp.2022.100229.

[3] R. Hassan, A.A. Majeed, dan Muqorobin, "Fingerprint data security system using AES algorithm on radio frequency identification (RFID) based population system," *Int. J. Inf. Technol. (INJIT)*, vol. 1, no. 1, hal. 14–20, Jan.-Apr. 2023.

[4] J.R. Naif, G.H. Abdul-Majeed, dan A.K. Farhan, "Secure IoT system based on chaos-modified lightweight AES," dalam *2019 Int. Conf. Adv. Sci. Eng. (ICOASE)*, 2019, hal. 1–6, doi: 10.1109/ICOASE.2019.8723807.

[5] E.B. Setiawan, A. Yunita, dan S.R. Sekarjatiningrum, "Development of automatic real time inventory monitoring system using RFID technology in warehouse," *JOIV, Int. J. Inform. Vis.*, vol. 6, no. 3, hal. 636–642, Sep. 2022, doi: 10.30630/joiv.6.3.1231.

[6] S.Q.A. Al-Rahman, A.M. Sagheer, dan O.A. Dawood, "NVLC: New variant lightweight cryptography algorithm for internet of things," dalam *2018 1st Annu. Int. Conf. Inf. Sci. (AiCIS)*, 2018, hal. 176–181, doi: 10.1109/AiCIS.2018.00042.

[7] L. Han, F. Yuan, dan Y. Jiang, "AES algorithm applied on security protocol of RFID," dalam *2015 3rd AASRI Conf. Comput. Intell. Bioinform. (CIB 2015)*, 2015, doi: 10.12783/dtcse/cib2015/16150.

[8] O.C. Abikoye dkk., "Modified advanced encryption standard algorithm for information security," *Symmetry*, vol. 11, no. 12, hal. 1–16, Des. 2019, doi: 10.3390/sym11121484.

[9] K. Eledlebi, C.Y. Yeun, E. Damiani, dan Y. Al-Hammadi, "Empirical studies of TESLA protocol: Properties, implementations, and replacement of public cryptography using biometric authentication," *IEEE Access*, vol. 10, hal. 21941–21954, Feb. 2022, doi: 10.1109/access.2022.3152895.

[10] R.M.A. Al_Azzawi dan S.S.M. Al-Dabbagh, "Software implementation solutions of a lightweight block cipher to secure restricted IoT environment: A review," *Al-Rafidain J. Comput. Sci. Math.*, vol. 16, no. 2, hal. 77–88, Des. 2022, doi: 10.33899/csmj.2022.176594.

[11] M. Qasaimeh, R.S. Al-Qassas, dan M. Ababneh, "Software design and experimental evaluation of a reduced AES for IoT applications," *Future Internet*, vol. 13, no. 11, hal. 1–21, Nov. 2021, doi: 10.3390/fi13110273.

[12] C. Fathy dan H.M. Ali, "A secure IoT-based irrigation system for precision agriculture using the expeditious cipher," *Sensors*, vol. 23, no. 4, hal. 1–16, Feb. 2023, doi: 10.3390/s23042091.

[13] V.A. Thakor, M.A. Razzaque, dan M.R.A. Khandaker, "Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison and research opportunities," *IEEE Access*, vol. 9, hal. 28177–28193, Jan. 2021, doi: 10.1109/ACCESS.2021.3052867.

[14] G. Mudra, H. Cui, dan M.N. Johnstone, "Survey: An overview of lightweight RFID authentication protocols suitable for the maritime internet of things," *Electronics*, vol. 12, no. 13, hal. 1–20, Jul. 2023, doi: 10.3390/electronics12132990.

[15] A. Sharma dan A. Singh, "Hybrid improved technique for data security and authentication for RFID tags," dalam *2017 4th Int. Conf. Signal Process. Comput. Control (ISPCC)*, 2017, hal. 536–540, doi: 10.1109/ISPCC.2017.8269737.

[16] S. Achmad, R. Adinugroho, N.S. Hendrawan, dan T. Franklin, "IoT based vehicle safety controller using Arduino," *Eng. Math. Comput. Sci. J. (EMACS)*, vol. 5, no. 1, hal. 1–6, Jan. 2023, doi: 10.21512/emacsjournal.v5i1.9251.

[17] R.K. Harahap dkk., "Dogs feed smart system with food scales indicator IoT based," dalam *2022 4th Int. Conf. Cybern. Intell. Syst. (ICORIS)*, 2022, hal. 1–7, doi: 10.1109/ICORIS56080.2022.10031344.

[18] S.S. Dhanda, B. Singh, dan P. Jindal, "Lightweight cryptography: A solution to secure IoT," *Wireless Pers. Commun.*, vol. 112, no. 3, hal. 1947–1980, Jun. 2020, doi: 10.1007/s11277-020-07134-3.

[19] R.H. Prayitno, S.A. Sudiro, dan S. Madenda, "Avoiding lookup table in AES algorithm," dalam *2021 6th Int. Conf. Infor. Comput. (ICIC)*, 2021, hal. 1–6, doi: 10.1109/ICIC54025.2021.9632897.

[20] M.F. Moghadam dkk., "An efficient authentication and key agreement scheme based on ECDH for wireless sensor network," *IEEE Access*, vol. 8, hal. 73182–73192, Apr. 2020, doi: 10.1109/ACCESS.2020.2987764.

[21] B.J. Mohd dan T. Hayajneh, "Lightweight block ciphers for IoT: Energy optimization and survivability techniques," *IEEE Access*, vol. 6, hal. 35966–35978, Jun. 2018, doi: 10.1109/ACCESS.2018.2848586.

[22] K.A. McKay, L. Bassham, M.S. Turan, dan N. Mouha, "Report on lightweight cryptography," Nat. Inst. Stand. Technol., Gaithersburg, MD, AS, NISTIR 8114, Mar. 2017.

[23] M.Sh. Oudah dan A.T. Maolood, "IoT-key agreement protocol based on the lowest work-load versions of the elliptic curve Diffie-Hellman," *Iraqi J. Sci.*, vol. 64, no. 8, hal. 4198–4207, Agu. 2023, doi: 10.24996/ijs.2023.64.8.39.

[24] M. Rashid dkk., "Throughput/area optimized architecture for elliptic-curve Diffie-Hellman protocol," *Appl. Sci.*, vol. 12, no. 8, hal. 1–18, Apr. 2022, doi: 10.3390/app12084091.

[25] S.Z. Khan, S.S. Jamal, A. Sajid, dan M. Rashid, "FPGA implementation of elliptic-curve Diffie Hellman protocol," *Comput. Mater. Continua*, vol. 73, no. 1, hal. 1879–1894, Mei 2022, doi: 10.32604/cmc.2022.028152.

[26] S. Aikins-Bekoe dan J.B. Hayfron-Acquah, "Elliptic curve Diffie-Hellman (ECDH) analogy for secured wireless sensor networks," *Int. J. Comput. Appl.*, vol. 176, hal. 1–8, Apr. 2020, doi: 10.5120/ijca2020920015.

[27] A.V.T. Maulidzart dkk., "The hybrid cryptographic algorithms for secure RFID data protection in the Internet of things," *J. ELTIKOM, J. Tek. Elekt. Teknol. Inf. Komput.*, vol. 7, no. 2, hal. 160–169, Des. 2023, doi: 10.31961/eltikom.v7i2.860.

[28] R.H. Prayitno, S.A. Sudiro, S. Madenda, dan S. Harmanto, "Hardware implementation of Galois field multiplication for mixcolumn and inversemixcolumn process in encryption-decryption algorithms," *J. Theor. Appl. Inf. Technol.*, vol. 100, no. 14, hal. 5358–5367, Jul. 2022.

[29] L. Widyawati, Husain, M. Azwar, dan M.C.S. Girsang, "Analisa perbandingan hybrid cryptography RSA-AES dan ECDH-AES untuk keamanan pesan," *JUTIK (J. Teknol. Inf. Komput.)*, vol. 9, no. 2, hal. 51–62, Jan. 2023, doi: 10.36002/jutik.v9i2.2182.

[30] G. Kanda, A.O.A. Antwi, dan K. Ryoo, "Hardware architecture design of AES cryptosystem with 163-bit elliptic curve," dalam *Adv. Multimed. Ubiquitous Eng.*, J. Park, V. Loia, K.K. Choo, dan G. Yi, Eds., Singapura, Singapura: Springer, 2019, hal. 423–429, doi: 10.1007/978-981-13-1328-8_55.