5th Seminar Nasional Mahasiswa Fakultas Teknologi Informasi (SENAFTI)
7 September 2024 – Jakarta, Indonesia
Volume 3, Nomor 2, September 2024 - ISSN 2962-8628 (*online*)

PENGAMANAN *FILE* DOKUMEN RESEP BERBASIS WEB MENGGUNAKAN METODE AES 128 BIT PADA ICHIYO CREPES

Felix Adi Pratama[1*], Gunawan Pria Utama[2]

[1,2] Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, Jakarta, Indonesia

Email: [1*]felixprtm@gmail.com, [2]gunawan.priautama@budiluhur.ac.id
(* : corresponding author)

Abstrak-ICHIYO CREPES didirikan pada tahun 2011 merupakan toko penjualan makanan yang mempunyai menu utama yaitu Crepes yang mengadaptasi asal makanan dari Perancis. ICHIYO CREPES yang sekarang mempunyai masalah karena tidak adanya sistem kemanan yang memadai untuk menyimpan informasi seperti data resep perusahaan. Hal ini memungkinkan data tersebut dapat disalahgunakan oleh pihak yang tidak bertanggung jawab untuk kepentingan pribadi atau organisasi. Saat ini, ICHIYO CREPES hanya memiliki hasil catatan resep yang di import kedalam Microsoft Word untuk digitalisasi sehingga pemilik memiliki kekhawatiran terkait data resep tersebut bisa kapan saja rusak, hilang atau diakses pihak lain. Maka dari itu diperlukan sistem enkripsi-dekripsi menggunakan algoritma AES-128(*Advanced Encryption Standard* 128) untuk meminimalisir risiko kehilangan dan dipakai orang yang bukan berwenang. Tujuan dari penelitian ini adalah untuk melindungi file dokumen resep perusahaan di ICHIYO CREPES agar tidak bisa diakses oleh pihak yang tidak berwenang. Dengan memodifikasi kunci acak(*random key*) pada algoritma AES-128, diharapkan output yang dihasilkan selama proses enkripsi selalu berubah secara acak setiap kali data diamankan menggunakan kunci acak yang telah disimpan di database. Hasil pengujian menunjukkan bahwa sistem enkripsi-dekripsi ini berhasil mengamankan data resep dari akses pihak yang tidak berwenang dan juga dapat memodifikasi metode AES-128 dengan *random key* serta dapat mengenkripsi semua *file* objek yang ada.
Kata Kunci: AES-128, kriptografi, enkripsi, *random key*

SECURING WEB-BASED RECIPE DOCUMENT FILES USING AES 128 BIT METHOD AT ICHIYO CREPES

Abstract-*ICHIYO CREPES established in 2011 is a food sales shop that has a main menu, namely Crepes which adapts food origin from France. The current ICHIYO CREPES has a problem because there is no adequate security system to store information such as company recipe data. This allows the data to be misused by irresponsible parties for personal or organisational purposes. Currently, ICHIYO CREPES only has the results of recipe records imported into Microsoft Word for digitisation so that the owner has concerns regarding recipe data that can be damaged, lost or accessed by other parties at any time. Therefore, an encryption-decryption system using the AES-128 (Advanced Encryption Standard 128) algorithm is needed to minimise the risk of loss and use by unauthorised people. The purpose of this research is to protect the company's recipe document files at ICHIYO CREPES from being accessed by unauthorised parties. By modifying the random key in the AES-128 algorithm, it is expected that the output generated during the encryption process always changes randomly each time the data is secured using a random key that has been stored in the database. The test results show that this encryption-decryption system can successfully secure prescription data from unauthorised access and can also modify the AES-128 method with a random key and can encrypt all prescription data.*
Keywords*: AES-128, cryptography, encryption, random key*

1. PENDAHULUAN

ICHIYO CREPES, didirikan pada tahun 2011, merupakan toko yang secara konsisten menjual Crêpe, makanan khas dari Bretagne, Perancis. Sebagai bagian dari industri *Food and Beverage*, ICHIYO CREPES mempunyai data yang mencakup penunjang kesuksesan dari sebuah perusahaan. Data ini bersifat rahasia dan sangat penting, termasuk resep perusahaan yang merupakan aset berharga. Oleh karena itu, resep tersebut tidak boleh digunakan secara sembarangan dan harus diamankan dengan baik oleh ICHIYO CREPES untuk melindungi kepentingan bisnis dan integritas perusahaan.

Perlindungan data khusunya resep perusahaan menjadi isu yang sangat penting terutama dalam sektor penjualan. Tujuan penjualan adalah ketika kedua belah pihak mendapatkan keuntungan dan kepuasan bersama[1]. Berdasarkan isu terkait perlindungan data, diperlukan suatu aplikasi yang bisa mengamankan data[2]. Salah satu teknik untuk mengamankan resep adalah menggunakan teknik kriptografi.

Kriptografi merupakan teknik menjaga keamanan pesan yang dikirim dari suatu tempat ke tempat yang lain. Dengan teknik kriptografi pesan asli yang ingin dikirimkan (*plaintext*) diubah atau dienkripsi dengan suatu kunci menjadi suatu informasi acak yang tidak bermakna (*ciphertext*)[3].

Kriptografi mempunyai dua jenis metode yaitu simetris dan asimetris[4]. Pada algoritma kriptografi simetris menggunakan kunci yang sama untuk enkripsi dan dekripsi data. Keunggulan utama dari kriptografi simetris adalah cepat dan efisien. Namun, kelemahannya terletak pada pada distribusi kunci yang memerlukan jalur aman untuk distribusi kunci antara pihak satu dengan yang lainnya[5]. Sebaliknya, proses kriptografi asimetris menggunakan kunci yang terdiri dari *public key* untuk enkripsi dan *private key* untuk dekripsi. Keunggulan utama dari kriptografi asimetris adalah kemampuannya untuk mengatasi masalah distribusi kunci, karena *public key* dapat dibagikan secara bebas[6]. Namun, kelemahannya pada proses enkripsi dan dekripsi cenderung lebih lambat dan membutuhkan lebih banyak sumber daya.

AES (*Advance Encryption Standard*) atau algoritma Rijndael adalah algoritma yang dicetuskan oleh Vincent Rijmen dan Joan Daemen [7] merupakan salah satu teknik kriptografi dan juga termasuk algoritma simetris yang digunakan dalam enkripsi dan dekripsi yang sangat efektif dalam melindungi data dokumen. Metode ini memiliki kemampuan untuk mengenkripsi dan mendeskripsikan data dengan menggunakan kunci (*key*) dengan panjang yang berbeda-beda, yaitu 128 bit, 192 bit, dan 256 bit. Perbedaan panjang kunci ini akan mempengaruhi jumlah putaran pada algoritma AES[8].

Penelitian sebelumnya telah mengaplikasikan kriptografi untuk mengamankan *file*. Salah satu studi yang dilakukan Eviyanti(2024) menggunakan AES-128 untuk mengenkripsi dokumen di Android dan autentikasi sidik jari untuk keamanan[9]. Hasilnya, kombinasi ini efektif melindungi data dari akses tidak sah. Sementara itu pada penelitian lainnya, dikembangkan juga aplikasi enkripsi-dekripsi pengamanan dokumen menggunakan algoritma ECR menggunakan modifikasi *random key* berbasis teks[10].

Penelitian ini menggunakan algoritma AES-128 dengan modifikasi *random key* karena keunggulannya dalam keamanan, ketahanan terhadap serangan, serta efisiensi dan kecepatan proses enkripsi dan dekripsi. Tujuan dari penelitian ini adalah untuk melindungi *file* dokumen resep perusahaan di ICHIYO CREPES agar tidak bisa diakses oleh pihak yang tidak berwenang.

2. METODE PENELITIAN

2.1 Studi Literatur dan Analisa

Peninjauan studi literatur dimulai dengan mencari artikel jurnal yang berkaitan dengan pengembangan aplikasi keamanan dokumen menggunakan enkripsi AES dan *random key* melalui *Publish or Perish*. Tahap berikutnya melibatkan riset dan identifikasi cakupan area penelitian yang dieksplorasi.

2.2 Implementasi

Pada implementasi ini, apa yang dirancang dalam tahap desain diwujudkan menggunakan bahasa pemrograman tertentu. Dalam hal ini, aplikasi ini menggunakan:

a. Perangkat lunak yang dipakai untuk mengamankan *file* data dikembangkan dengan bahasa pemrograman PHP, dan phpMy*Admin* digunakan sebagai basis datanya.
b. Perangkat keras yang digunakan adalah Windows 11 Home, dengan prosesor AMD Ryzen 5 5500U, RAM 8 GB DDR4, dan SSD 512 GB.

2.3 Data Penelitian

Data penelitian yang akan digunakan untuk menguji program yang akan dibuat adalah *file* dokumen yang berisi resep dan *file* dokumen pendukung lainnya pada ICHIYO CREPES. Berikut adalah tabel 1 terkait data penelitian yang akan digunakan.

Table 1. Data Penelitian

Nama *file*	Jenis *file*	Ukuran *file*
resepichiyo	.docx	14KB
ichiyosales2023	.xlsx	28KB
vidioichiyo1	.mp4	3500KB

2.4 Penerapan Metode

Pada penjelasan penerapan metode dalam penelitian ini adalah bahwa modifikasi yang diusulkan pada AES-128 terletak pada penggunaan kunci acak (*random key*) yang dihasilkan setiap kali sesi enkripsi dimulai. Meskipun proses enkripsi mengikuti aturan standar AES-128, penerapan *random key* ini diharapkan dapat meningkatkan keamanan enkripsi dengan mengubah kunci enkripsi di setiap sesi, sehingga membuat hasil enkripsi lebih sulit diprediksi.

a. Proses Enkripsi

Proses enkripsi dengan algoritma AES-128 terdiri dari beberapa tahapan transformasi yang sistematis. Berikut adalah tahapan proses enkripsi diawali dengan menentukan *plaintext* = "ICHIYOCREPESJAYA" dan *cipherkey* = "SUKSESJAYASELALU".

1) Pada gambar 1 dilakukan proses perubahan kata *plaintext* dan *cipherkey* menjadi heksadesimal.

Plainteks

49	59	45	4A
43	4F	50	41
48	43	45	59
49	52	53	41

Cipherkey

53	45	59	4C
55	53	41	41
4B	4A	53	4C
53	41	45	55

Gambar 1. Hasil Heksadesimal dari *Plaintext* dan *Cipherkey*

2) Selanjutnya dilakukan proses *key expansion* atau pembangkitan kunci pada *cipherkey*. Pada proses ini berfungsi sebagai acuan perhitungan *AddRoundKey* pada enkripsi. Pada gambar 2 ini menggunakan proses RotWord setelah itu dilakukan *SubBytes* untuk mengubah heksadesimal kemudian dilakukan perhitungan XOR dengan bantuan tabel Rcon sehingga didapat hasil seperti dibawah ini lakukan hingga round 10.

Proses Key Expansion

Round 0 (*Initial Round*)

53	45	59	4C
55	53	41	41
4B	4A	53	4C
53	41	45	55

Round 1

D1	94	CD	81
7C	2F	6E	2F
B4	FE	AD	E1
7A	3B	7E	2B

Gambar 2. *Key Expansion*

3) Pada langkah selanjutnya dilakukan proses enkripsi dengan menggunakan heksadesimal dari *plaintext* dan *cipherkey* yang sebelumnya sudah dijabarkan pada poin 1. Selanjutnya gambar 3 dilakukan proses XOR heksadesimal *plaintext* dengan *cipherkey* kemudian didapatkan hasil seperti pada Gambar 3 berikut ini.

Hasil XOR *plainteks* dengan *cipherkey*

1A	1C	1C	06
16	1C	11	00
03	09	16	15
1A	13	16	14

Gambar 3. Hasil XOR *Plaintext* Dan *Cipherkey*

4) Setelah mendapatkan hasil XOR diatas, maka pada gambar 4 dilakukan proses transformasi *SubBytes*. Sehingga didapatkan hasil seperti pada Gambar 4 berikut.

Transformasi *SubBytes*

A2	9C	9C	6F
47	9C	82	63
7B	01	47	59
A2	7D	47	FA

Gambar 4. Transformasi *SubBytes*

5) Setelah dilakukan proses transformasi *SubBytes* maka pada gambar 5 dilakukan proses *ShiftRows* yaitu menggeser baris-baris pada byte.

Transformasi *ShiftRows*

A2	9C	9C	6F
9C	82	63	47
47	59	7B	01
FA	A2	7D	47

Gambar 5. Transformasi *ShiftRows*

6) Kemudian pada gambar 6 dilakukan proses transformasi *MixColumns* untuk mengacak baris pada byte. Heksadesimal pada *ShiftRow*s dikalikan dengan matriks Galois *Field* (GF).

Transformasi *MixColumns*

5D	45	80	51
B2	CA	AA	A5
A5	51	8E	E3
C9	3B	5D	79

Gambar 6. Transformasi *MixColumns*

7) Pada proses ini dilakukan proses transformasi *AddRoundKey*. Berikut ini gambar 7 yang menunjukkan proses XOR pada *MixColumns* dengan proses *key expansion*.

Transformasi *AddRoundKey*

8C	D1	4D	D0
CE	E5	C4	8A
12	AC	20	01
B3	00	23	52

Gambar 7. Transformasi *AddRoundKey*

8) Setelah melakukan XOR pada proses diatas yaitu menghasilkan proses enkripsi round 1. Proses yang sudah dijabarkan diatas dilakukan sebanyak 10 putaran dengan catatan pada putaran ke 10 tidak menggunakan transformasi *MixColumns*. Berikut ini gambar 8 adalah hasil proses enkripsi yang sudah dijabarkan sebelumnya.

Proses Enkripsi

Round 0(Initial Round)

1A	1C	1C	06
16	1C	11	00
00	09	16	15
1A	13	16	14

Round 1

8C	D1	4D	D0
CE	E5	C4	8A
12	AC	20	01
B3	00	23	52

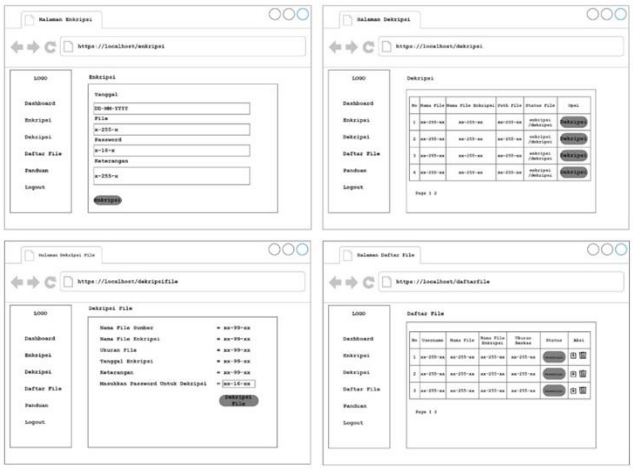
Gambar 8. Proses Enkripsi

b. Proses Dekripsi

Proses dekripsi AES-128 adalah bahwa dekripsi dilakukan dengan menggunakan langkah-langkah yang merupakan kebalikan dari proses enkripsi. Dimulai dengan memasukkan *key password* yang digunakan selama enkripsi, kemudian dilakukan serangkaian transformasi seperti *AddRoundKey*, *InvShiftRows*, *InvSubBytes*, dan *InvMixColumns*. Proses ini diulang hingga mencapai putaran terakhir, di mana hanya dilakukan *InvShiftRows*, *InvSubBytes*, dan *AddRoundKey*. Setelah semua langkah dekripsi selesai, *ciphertext* akan dikembalikan menjadi bentuk asli *plaintext* seperti sebelum dienkripsi.

2.5 Rancangan Layar

Dalam pembuatan program, perancangan layar merupakan bagian yang sangat penting. Oleh karena itu, tujuan dari rancangan layar ini adalah untuk memastikan bahwa *user* dapat dengan mudah memahaminya, sehingga mereka bisa cepat beradaptasi dan tidak mengalami kesulitan saat menggunakannya. Gambar 9 menunjukkan rancangan fitur utama layar dari program yang dibuat.



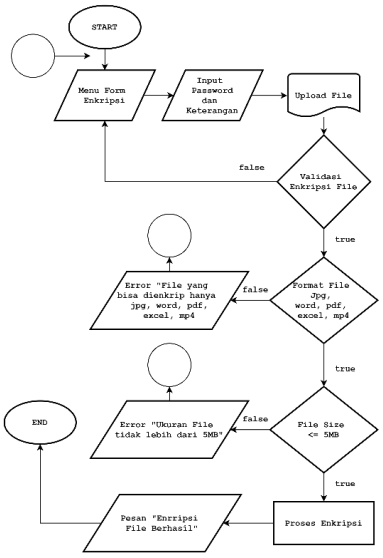
Gambar 9. Rancangan Fitur Utama Layar

3. HASIL DAN PEMBAHASAN

Berdasarkan hasil dari metode penelitian yang sebelumnya telah dibahas pada sub bab 2.4, pada bab ini dijelaskan secara rinci tentang *flowchart* dari proses enkripsi dan dekripsi. Proses enkripsi dan dekripsi dijelaskan mulai dari pemilihan algoritma yang digunakan hingga implementasinya pada data. Selanjutnya, hasil pengujian terhadap proses enkripsi dan dekripsi tersebut ditampilkan, yang meliputi analisis kecepatan, efisiensi, dan tingkat keamanan data yang dihasilkan.

3.1 Flowchart Menu Enkripsi *File*

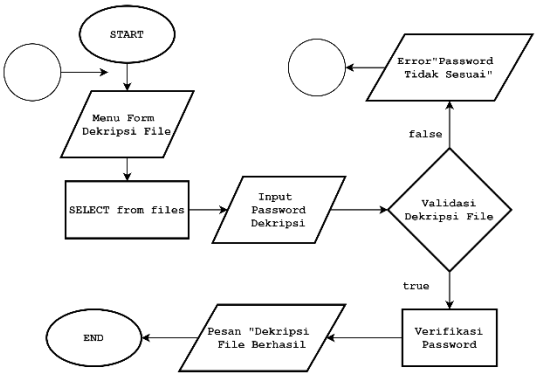
Flowchart untuk menu enkripsi ditampilkan pada gambar 10, yang menjelaskan proses enkripsi *file* oleh *admin*. Langkah pertama yang dilakukan *admin* adalah mengunggah *file* dan memasukkan *password*. *File* yang dapat diunggah harus berformat jpg, png, docx, doc, pdf, xls, xlsx, ppt, pptx, mp4, dengan ukuran tidak melebihi 5MB.



Gambar 10. Flowchart Menu Enkripsi *File*

3.2 *Flowchart* Menu Dekripsi *File*

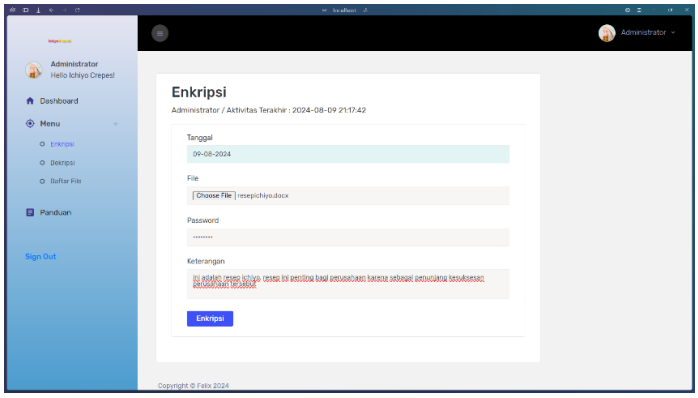
Flowchart untuk menu dekripsi ditampilkan pada gambar 11, yang menunjukkan *file-file* yang akan didekripsi beserta status masing-masing *file*. Setelah *file* dipilih, *admin* dapat memulai proses dekripsi dengan memasukkan *password* yang benar. Jika kata sandi sudah sesuai, *file* akan berhasil didekripsi. Jika kata sandi salah maka akan ada pesan error "*password* tidak sesuai".



Gambar 11. Flowchart Menu Dekripsi *File*

3.3 Tampilan Layar Enkripsi *File*

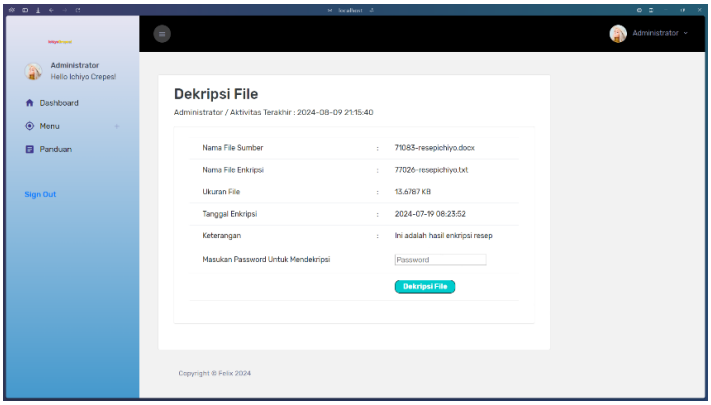
Pada gambar 12 tampilan layar enkripsi menunjukkan bahwa setiap layar telah dirancang dengan fokus pada fungsionalitas dan keamanan. Layar *sign in* menyediakan dua *field* utama, yaitu *username* dan *password*, yang harus diisi oleh pengguna dengan informasi yang terdaftar dalam *database*. Layar enkripsi menampilkan formulir yang mencakup *field* Tanggal, *File*, *Password*, dan Keterangan untuk mendukung proses enkripsi, jika *file* sudah diunggah maka proses selanjutnya adalah "ENKRIPSI *FILE*" untuk mendapatkan hasil *file chipertext*.



Gambar 12. Tampilan Layar Enkripsi *File*

3.4 Tampilan Layar Dekripsi *File*

Pada gambar 13 tampilan layar dekripsi *file* memungkinkan pengguna untuk mengembalikan *file* terdekripsi menjadi *plaintext* dengan memasukkan *password* yang sama yang digunakan dalam proses enkripsi, memastikan integritas dan keamanan data selama proses tersebut dengan menklik tombol "DEKRIPSI *FILE*".



Gambar 13. Tampilan Layar Dekripsi *File*

3.5 Hasil Pengujian *File* Enkripsi dan Dekripsi

Hasil pengujian terhadap *file* dokumen yang sebelumnya diuji, yang datanya berasal dari ICHIYO CREPES khususnya data resep, menunjukkan bahwa semua format *file* yang diuji berhasil diproses dengan baik. *File* yang diuji mencakup berbagai format, termasuk docx, xlxs dan mp4. Proses enkripsi dan dekripsi berjalan lancar pada semua jenis *file*, memastikan bahwa data dalam format visual, teks, dan multimedia terlindungi dengan baik selama proses enkripsi dan tetap utuh setelah dekripsi. Hasil pengujian Tabel 2 ini menegaskan bahwa sistem yang digunakan mampu menangani berbagai format *file* secara efisien dan aman.

Tabel 2. Hasil Pengujian *File* Enkripsi dan Dekripsi

Nama *File*	Ukuran *File* (*Kilobyte*)			Waktu (Detik)	
	Asli	Enkripsi	Dekripsi	Enkripsi	Dekripsi
resepichiyo.docx	14	14	14	0.16	0.16
ichiyosales2023.xlxs	28	28	28	0.83	0.83
vidioichiyo1.mp4	3500	3500	3500	96.5	99.4

4. KESIMPULAN

Berdasarkan pengujian program enkripsi dan dekripsi yang dilakukan untuk mengamankan data pada ICHIYO CREPES, dapat disimpulkan bahwa program ini berhasil mengembangkan dan menguji *file* dokumen resep sehingga terbukti mampu melindungi *file* resep dari akses yang tidak sah. Selain itu, program ini berhasil memodifikasi metode AES-128 dengan *random key*, serta mampu mengenkripsi berbagai format *file* seperti docx, xlxs dan mp4. Hasil pengujian juga menunjukkan bahwa waktu yang dibutuhkan untuk proses enkripsi dan dekripsi meningkat seiring dengan bertambahnya ukuran *file*.

DAFTAR PUSTAKA

[1] K. Andriani and B. H. Hayadi, "Pengamanan Data Penjualan Dengan Kriptografi Algoritma Rivest Shamir Adleman (Rsa) Pada Toko Baju Family," *Journal of Science and Social Research*, 5(3), pp.664-670, 2022. [Online]. Available: http://jurnal.goretanpena.com/index.php/JSSR

[2] A. I. Suranta, D. Virgian, and S. Y. Sakti, "Penerapan Algoritma AES (Advance Encryption Standart) 128 untuk Enkripsi Dokumen di PT. Gunung Geulis Elok Abadi," *SKANIKA: Sistem Komputer dan Teknik Informatika*, vol. 5, no. 1, pp. 1–10, 2022.

[3] A. Eka Putri, A. Kartikadewi, and L. A. Abdul Rosyid, "Implementasi Kriptografi dengan Algoritma Advanced Encryption Standard (AES) 128 Bit dan Steganografi menggunakan Metode End of File (EOF) Berbasis Java Desktop pada Dinas Pendidikan Kabupaten Tangerang," *Applied Information System and Management (AISM)*, vol. 3, no. 2, pp. 69–78, Jan. 2021, doi: 10.15408/aism.v3i2.14722.

[4] Z. Arif and A. Nurokhman, "Analisis Perbandingan Algoritma Kriptografi Simetris Dan Asimetris Dalam Meningkatkan Keamanan Sistem Informasi Comparative Analysis of Symmetric and Asymmetric Cryptographic Algorithms in Improving Information System Security," *Jurnal Teknologi Sistem Informasi*, *4*(2), pp.394-405, 2023.

[5] F. Nova Hulu, M. Putri, and K. Kunci, "Metode Analitis Enkripsi Dan Dekripsi Dengan Penerapan Algoritma Kriptografi Klasik Ke Dalam Cipher." *Jurnal Elektro dan Telkomunikasi*, *8*(1), pp.26-34, 2022.

[6] N. Amalya, S. M. Sopiana Silalahi, D. F. Nasution, M. Sari, and I. Gunawaan, "Kriptografi dan Penerapannya Dalam Sistem Keamanan Data," *Jurnal Media Informatika*, 4(2), pp.90-93, 2023.

[7] S. Setti, I. Gunawan, B. E. Damanik, S. Sumarno, and I. O. Kirana, "Implementasi Algoritma Advanced Encryption Standard dalam Pengamanan Data Penjualan Ramayana Department Store," *JURIKOM (Jurnal Riset Komputer)*, vol. 7, no. 1, p. 182, Feb. 2020, doi: 10.30865/jurikom.v7i1.1960.

[8] M. Azhari, J. Perwitosari, and F. Ali, "Implementasi Pengamanan Data pada Dokumen Menggunakan Algoritma Kriptografi Advanced Encryption Standard (AES)," *Jurnal Pendidikan Sains dan Komputer*, vol. 2, no. 1, pp. 2809–476, 2022, doi: 10.47709/jpsk.v2i1.1390.

[9] H. H. Amirullah and A. Eviyanti, "Android File Security Application with AES Encryption and Fingerprint Authentication [Aplikasi Keamanan Berkas dengan Enkripsi AES dan Biometrik Sidik Jari Berbasis Android].", 2024, DOI: https://doi.org/10.21070/ups.3769

[10] H. Murti, E. Ardhianto, E. Lestariningsih, and W. Tri Handoko, "Desain Baru Covertext dan Encryption Key Generator pada Model Enkripsi ECR," *Jurnal Informatika dan Rekayasa Perangkat Lunak*, vol. 4, no. 2,), pp.92-97, 2022.