Jurnal Teknik Informatika (JUTIF)
Vol. 5, No. 4, October 2024, pp. 1375-1384

DOI: https://doi.org/10.52436/1.jutif.2024.5.5.2665
p-ISSN: 2723-3863
e-ISSN: 2723-3871

IMPLEMENTATION OF A COMBINATION OF ADVANCED ENCRYPTION STANDARD CRYPTOGRAPHY WITH SUBBYTES MODIFICATION AND STEGANOGRAPHY BASED ON A WEBSITE

Muhammad Ilham Kurniawan[*1], Eddy Maryanto[2], Swahesti Puspita Rahayu[3]

[1,2,3]Informatics, Engineering Faculty, Universitas Jenderal Soedirman, Indonesia
Email: [1]ilham.kurniawan@mhs.unsoed.ac.id, [2]eddy.maryanto@unsoed.ac.id, [3]swahesti.rahayu@unsoed.ac.id

(Article received: August 15, 2024; Revision: September 23, 2024; published: October 25, 2024)

Abstract

The Advanced Encryption Standard (AES) is a symmetric encryption algorithm commonly used to protect digital data. However, concerns about potential attacks on cryptographic keys and the development of cryptanalysis methods further reinforce the need for security enhancement. This study aims to combine two technologies: the Advanced Encryption Standard (AES) cryptography with modifications to the SubBytes, and steganography using the Least Significant Bit (LSB) method in images, to enhance the security level of encrypted messages in the context of transmission through websites. In this study, modifications were made to the AES algorithm by replacing the S-box in the SubBytes process with a perfect SAC S-box with an average SAC value of 0.5. This testing is divided into two types: algorithm testing and system testing. Algorithm testing involves performance testing methods that show longer decryption times with an average difference of 80.27 milliseconds, cryptanalysis testing showing increased ciphertext security based on cryptanalysis time estimates using brute force, and randomness testing to demonstrate improvements in Frequency and Poker tests. System testing using the Black Box method shows results that are valid as expected.

Keywords: *Advanced Encryption Standard, Least Significant Bit, SubBytes, S-box*

IMPLEMENTASI KOMBINASI KRIPTOGRAFI ADVANCED ENCRYPTION STANDARD DENGAN MODIFIKASI SUBBYTES DAN STEGANOGRAFI BERBASIS WEBSITE

Abstrak

Advanced Encryption Standard (AES) merupakan algoritma enkripsi simetris yang umumnya digunakan untuk melindungi data digital. Namun, kekhawatiran akan potensi serangan terhadap kunci kriptografi serta pengembangan metode kriptanalisis semakin memperkuat kebutuhan akan peningkatan keamanan. Penelitian ini bertujuan untuk menggabungkan dua teknologi, yaitu kriptografi *Advanced Encryption Standard* (AES) dengan modifikasi pada *SubBytes*, dan steganografi menggunakan metode *Least Significant Bit* (LSB) pada gambar, untuk meningkatkan tingkat keamanan pesan terenkripsi dalam konteks pengiriman melalui *website*. Dalam penelitian ini, dilakukan modifikasi terhadap algoritma AES dengan mengganti S-box pada proses *SubBytes* dengan perfect SAC S-box yang memiliki nilai rata-rata SAC sebesar 0,5. Pengujian ini terbagi menjadi dua jenis, yaitu pengujian algoritma dan pengujian sistem. Pengujian algoritma melibatkan metode uji performa yang menunjukkan waktu dekripsi yang lebih lama dengan rata-rata selisih sebesar 80.27 milidetik, uji *cryptanalysis* yang menunjukkan peningkatan keamanan *ciphertext* berdasarkan estimasi waktu *cryptanalysis* menggunakan *brute force*, dan uji *randomness* untuk menunjukkan peningkatan pada tes *Frequency* dan *Poker*. Pengujian sistem menggunakan metode *Blackbox* menunjukkan hasil yang valid sesuai harapan.

Kata kunci: *Advanced Encryption Standard, Least Significant Bit, SubBytes, S-box*

1. PENDAHULUAN

Pada era digital yang terus berkembang, keamanan informasi menjadi isu krusial yang membutuhkan solusi inovatif. Pengiriman pesan terenkripsi melalui *website* menjadi semakin penting dalam berbagai aspek, seperti komunikasi bisnis, pertukaran data pribadi, dan interaksi *online*. Salah satu algoritma enkripsi data yang terkenal dan sering digunakan dalam berbagai aplikasi dan sistem keamanan data adalah *Advanced Encryption Standard* (AES). AES adalah suatu metode enkripsi simetris yang digunakan untuk menjaga keamanan

data dalam bentuk digital [1]. Awalnya AES bertujuan menggantikan *Data Encryption Standard* (DES) karena DES menggunakan kunci enkripsi yang terbatas dan algoritmanya memiliki kecepatan yang lambat [2].

Pemilihan metode AES pada penelitian ini didasarkan pada keunggulan AES dalam hal keamanan data, kecepatan enkripsi dan dekripsi, serta efisiensi penggunaan sumber daya komputasi dibandingkan dengan algoritma lainnya. Pada penelitian sebelumnya, seperti yang dilakukan oleh Rahmaniah [3], telah menunjukkan bahwa AES menawarkan fleksibilitas dalam memilih tingkat keamanan berdasarkan panjang kunci yang digunakan.

Di sisi lain, penggunaan kriptografi simetris dalam penelitian ini dipertimbangkan berdasarkan penelitian terdahulu. Penelitian yang dilakukan oleh Arif dan Nurokhman [4] menunjukkan bahwa metode simetris standar menawarkan kecepatan komputasi yang lebih baik, menjadikannya pilihan yang diinginkan untuk distribusi data yang cepat.

Namun, kendati AES telah memberikan tingkat keamanan yang tinggi, masih terdapat tantangan terkait dengan penyimpanan dan pengiriman pesan terenkripsi secara efisien melalui *platform website*. Selain itu, karena AES sudah ditemukan dan secara luas digunakan sejak diterbitkannya pada tahun 2001 [5], [6], algoritma ini dianggap sudah lama dan usang, terdapat kekhawatiran terhadap potensi serangan terhadap kunci kriptografi dan pengembangan metode kriptanalisis semakin memicu kebutuhan akan peningkatan keamanan.

Penelitian ini menjawab panggilan tersebut dengan menggabungkan kekuatan dua teknologi, yaitu kriptografi *Advanced Encryption Standard* (AES) dengan modifikasi *SubBytes*, dan steganografi menggunakan metode *Least Significant Bit* (LSB) pada gambar. Implementasi modifikasi *SubBytes* pada AES bertujuan untuk meningkatkan kompleksitas enkripsi, sementara steganografi LSB diharapkan memberikan lapisan tambahan untuk penyimpanan pesan terenkripsi dalam gambar tanpa mengorbankan kualitas visualnya.

Metode LSB adalah salah satu metode yang sangat umum digunakan karena merupakan dasar bagi pengembangan metode steganografi yang lebih kompleks. Dalam metode ini, bit yang memiliki nilai paling kecil atau yang paling tidak berarti digunakan untuk menyisipkan bit pesan tanpa mempengaruhi secara signifikan citra atau data asli [7].

Penggunaan metode LSB dipilih karena cenderung sederhana, kecepatan ekstraksi yang tinggi, dan kapasitas penyisipan yang cukup besar [8]. Di sisi lain titik berat pada penelitian ini adalah modifikasi enkripsi, penggunaan steganografi metode LSB hanya sebagai tambahan proses untuk merubah *ciphertext* ke dalam bentuk lain yang dalam hal ini adalah gambar.

Sistem yang dikembangkan menggunakan *Streamlit Python* untuk mendukung keberlanjutan penggunaan kriptografi dan steganografi pada lingkungan *website* menggunakan metode *waterfall*. Dinamakan *waterfall* karena setiap tahap harus menunggu penyelesaian tahap sebelumnya dan dilaksanakan secara berurutan [9]. *Python* digunakan pada penelitian ini karena penggunaan bahasa pemrograman seperti PHP atau C++ dalam implementasi aplikasi seperti pada penelitian Angraini dan Suryanto [10] dianggap kurang optimal, sedangkan *Python* dianggap lebih handal karena didesain untuk kemudahan penggunaan (ease of use), *open source*, ekosistem perpustakaan yang kaya, serta dukungan untuk *machine learning* [11], [12], [13].

Implementasi pada lingkungan *website* dilakukan karena implementasi dalam bentuk aplikasi mandiri seperti *mobile* atau *desktop* yang seperti pada penelitian Cristy dan Riandari [14] menyebabkan kurangnya fleksibilitas, karena aplikasi hanya dapat diakses oleh sejumlah perangkat tertentu dan memerlukan proses instalasi.

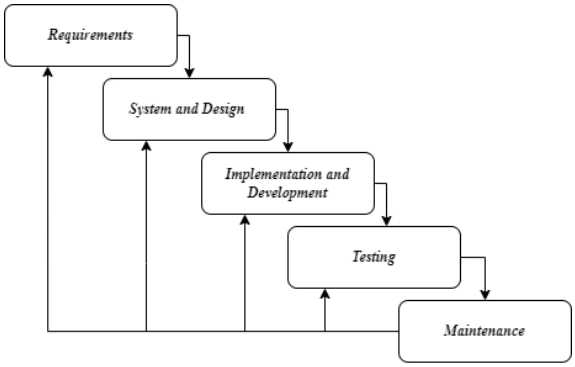
Penelitian ini bertujuan untuk meningkatkan keamanan AES yang dianggap sudah lama dan usang dengan modifikasi *SubBytes* dan penambahan steganografi LSB yang dapat diintegrasikan dengan mudah pada *platform web* yang dapat diakses secara bersama-sama [15].

Melalui penelitian ini, diharapkan dapat terbentuk landasan yang kuat untuk pengembangan sistem keamanan informasi yang lebih tangguh, memberikan manfaat tidak hanya bagi keamanan data, tetapi juga bagi perkembangan teknologi dan aplikasi *web* secara umum.

2. METODE PENELITIAN

Subjek penelitian adalah data valid yang akan diamati. Subjek dari penelitian ini adalah pesan terenkripsi dalam format gambar (jpg, png, jpeg). Sedangkan Objek penelitian ini melibatkan implementasi modifikasi *SubByte* pada AES Kriptografi dan Steganografi LSB.

Langkah-langkah penelitian yang akan dilakukan akan diintegrasikan ke dalam metodologi *waterfall*. Pendekatan ini dikenal melakukan proses secara sistematis dan berurutan. Berikut adalah detail metode waterfall dapat dilihat pada Gambar 1:



Gambar 1. Metode Waterfall

Tahapan dalam metode *waterfall* adalah sebagai berikut:

a) *Requirements*

Memahami kebutuhan *software* melalui wawancara, survei, atau diskusi, serta menganalisis informasi untuk mendapatkan data lengkap tentang kebutuhan pengguna.

b) *System and Design*

Melakukan desain sebelum pengkodean, memberikan gambaran lengkap tentang tampilan dan fungsi sistem yang diinginkan.

c) *Implementation*

Menulis kode program, memecahnya menjadi modul-modul kecil, dan melakukan pemeriksaan mendalam terhadap setiap modul.

d) *Testing*

Memastikan bahwa *software* sesuai dengan desain dan bebas dari kesalahan atau bug signifikan.

e) *Maintenance*

Mengoperasikan aplikasi setelah pengujian, serta melakukan perbaikan kesalahan yang ditemukan setelah aplikasi digunakan oleh pengguna.

2.1. Modifikasi *Advanced Encryption Standard* (AES)

Advanced Encryption Standard (AES) adalah salah satu algoritma kriptografi simetris yang umum digunakan untuk melindungi keamanan data. AES berfungsi sebagai *block cipher* simetris dan resmi diperkenalkan oleh Institut Standar dan Teknologi Nasional AS pada tahun 2001.

Algoritma AES dipilih karena kemampuannya dalam mengenkripsi dan mendekripsi data dengan panjang kunci yang dapat divariasikan, yaitu 128 bit, 192 bit, dan 256 bit. Perbedaan dalam panjang kunci ini berdampak pada jumlah putaran dalam algoritma AES, sesuai dengan penelitian oleh Azhari, dkk. [16]. Oleh karena itu, dalam penelitian ini, AES yang diadopsi adalah AES-256, dengan panjang kunci 256 bit.

Proses enkripsi algoritma AES akan melakukan 4 proses, yaitu *SubBytes*, *ShiftRows*, *MixColumn* dan *addArroundKey*. Tahapan-tahapan yang dilakukan seperti :

a) *SubBytes*: Substitusi setiap byte dalam blok data menggunakan tabel substitusi S-*box*.

b) *ShiftRows*: Pergeseran baris dalam blok data untuk menyebarluaskan data.

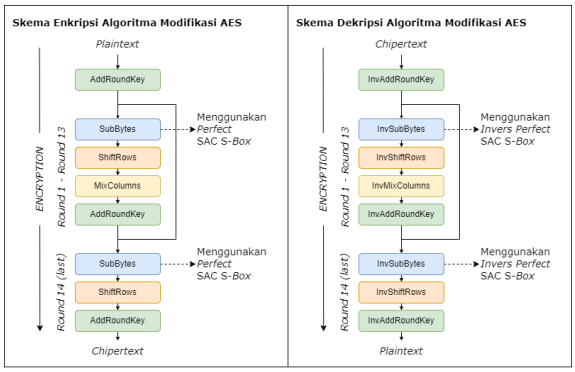
c) *MixColumns*: Transformasi linier terhadap setiap kolom dalam blok data.

d) *AddRoundKey*: Melakukan operasi XOR antara setiap byte blok data dengan kunci putaran.

Tahapan-tahapan ini diulang dalam sejumlah putaran tertentu, tergantung pada panjang kunci yang digunakan, untuk mencapai enkripsi yang kuat dan kompleks.

Ciphertext pada algoritma modifikasi AES dihasilkan melalui transformasi dari blok input. Transformasi ini dilakukan dalam 14 putaran. Pada putaran terakhir, terdapat perbedaan dengan putaran sebelumnya, yaitu Nr-1 (putaran ketiga belas), di mana tidak ada transformasi *MixColumns*().

Proses enkripsi dan dekripsi untuk algoritma modifikasi AES dapat dilihat pada Gambar 2.



Gambar 2. Skema Enkripsi dan Dekripsi AES Modifikasi

Algoritma modifikasi AES memiliki struktur yang disebut *Substitution and Permutation Network* (SPN). Sesuai dengan nama strukturnya, desain algoritma AES terdiri dari lapisan substitusi dan permutasi. Lapisan substitusi dipenuhi dari proses *SubByte* yang menggunakan S-*box*, sedangkan permutasi dari proses *Shiftrows* dan *Mixcolumn*. Proses *Mixcolumn* menggunakan perkalian matriks MDS yang merupakan proses multi permutasi.

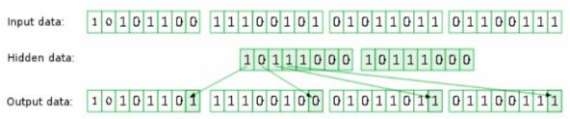
Sebelum memasuki proses AES, *Plaintext* dan *Chiper Key* yang sudah didapatkan sebelumnya diubah menjadi matriks lalu diubah formatnya menjadi *HexaDecimal* berukuran 4x4 untuk *plaintext* dan 8x8 untuk *chiperkey*. Lalu setelah proses AES dijalankan, matriks kembali diubah kembali menjadi *Decimal* lalu dikembalikan lagi menjadi teks. Cara yang sama juga digunakan ketika melakukan dekripsi.

2.2. *Least Significant Bit* (LSB)

Least Significant Bit (LSB) merupakan teknik steganografi yang paling sederhana dan mudah diaplikasikan dalam berbagai aplikasi. Teknik ini memanfaatkan gambar digital sebagai media penyimpanan pesan. Pada setiap byte data yang

terdiri dari 8 bit, *Least Significant Bit* (LSB) digunakan untuk menyimpan informasi rahasia [17].

Ada dua teknik yang dapat digunakan pada LSB, yaitu penyisipan secara sekuensial dan secara acak. Penyisipan sekuensial dilakukan berurutan sedangkan penyisipan acak dilakukan dengan memasukan kata kunci (stego key) [18]. Berikut adalah ilustrasi cara kerja algoritma LSB yang dapat dilihat pada Gambar 3.



Gambar 3. Ilustrasi Cara Kerja Algoritma LSB

3. HASIL DAN PEMBAHASAN

Aplikasi ini memiliki nama yaitu ”CryptoGuard” yang dikembangkan dan dibangun dengan menggunakan Bahasa pemrograman *Python*, menggunakan *framework Streamlit*, dan *database* MySQL.

3.1. *Requirements*

Langkah pertama yaitu mengumpulkan data yang akan digunakan sebagai subjek penelitian. Kemudian mengidentifikasi pengguna yang akan menggunakan sistem penyimpanan data menggunakan kriptografi dan steganografi ini.

3.1.1. Pengumpulan Data

Data ini bersumber dari *website* PDDikti (Pangkalan Data Pendidikan Tinggi) untuk pengambilan informasi akademik dan Sista Unsoed untuk pengambilan foto mahasiswa. Subjek pada data ini adalah mahasiswa Informatika Universitas Jenderal Soedirman. Jumlah data yang diperoleh dan akan diuji dalam penelitian ini sebanyak 10 mahasiswa.

Data yang dikumpulkan hanya akan digunakan sebagai data sampel dalam implementasi sistem. Data ini dapat dilihat pada Tabel 1 berikut:

Tabel 1. Data Mahasiswa

No	Foto	Nama	NIM	Program Studi	Semester	Status
1		Firman Kunia Jati	H1D020001	Informatika	8	Belum Lulus
2		Hisyam Adelio Pradipta	H1D020088	Informatika	8	Belum Lulus
3		Rizki Hasan Maulana	H1D020005	Informatika	8	Belum Lulus
4		Ali Murtadho	H1D020054	Informatika	8	Belum Lulus
5		Hafizh Trisnindito	H1D020063	Informatika	8	Belum Lulus
6		Fardan Maula Azizi	H1D020053	Informatika	8	Belum Lulus
7		Pinggang Taruna Andalan	H1D020067	Informatika	8	Belum Lulus
8		Lintang Adi Bagaskara	H1D020039	Informatika	8	Belum Lulus
9		Safa Muazam	H1D020048	Informatika	8	Belum Lulus
10		Anin Ammbya Soulani	H1D020055	Informatika	8	Belum Lulus

3.1.2. Inisialisasi Metode

Pada bagian ini dijelaskan lebih lanjut terkait bagaimana penggunaan metode yang digunakan pada penelitian ini.

a) Inisialisasi AES

Dalam penelitian ini, dilakukan modifikasi terhadap algoritma AES dengan mengganti S-*box* yang digunakan pada proses SubBytes. S-box yang dipilih adalah S-*box* yang dihasilkan oleh Alamsyah [19] yang memiliki nilai rata-rata SAC (Strict Avalanche Criterion) sebesar 0,5, sehingga disebut sebagai *perfect* SAC S-*box*. Adapun *Perfect* SAC S-*box* yang digunakan pada penelitian ini ditunjukkan pada Gambar 4 di bawah ini :

63	7C	D7	44	2	81	F0	F3	E8	13	12	24	91	74	10	C2
9D	2E	60	28	E0	F4	FB	6E	1A	DA	D3	61	E1	A1	B3	7F
27	45	FE	9	E2	C3	C6	F	99	CE	A8	26	14	B0	DE	A
E4	CF	BF	58	3B	A5	62	1C	19	B5	39	46	30	90	56	3C
7A	A9	70	35	AD	7B	6D	32	98	41	33	3	8A	52	55	C9
1E	D6	8E	F8	BD	A7	FA	88	D8	64	B1	6C	86	67	EC	21
A0	50	E	53	D	BA	C5	6A	4F	47	0	1D	E3	FD	DC	FC
65	BB	8	E5	4E	57	F1	FF	CA	48	9A	2A	F9	72	F7	84
EF	3E	3D	7	EA	2F	73	93	4	AF	6F	85	5F	76	CB	23
9E	1F	49	D4	4B	CC	68	69	97	17	C0	A3	78	D1	36	A2
DD	D9	82	8D	AE	8C	95	3F	C	9B	1	4A	94	8B	96	6
BE	16	DB	BC	31	92	DF	C4	AA	89	5A	80	A4	B6	42	C8
B9	F6	C1	25	D5	51	40	77	54	7E	B4	9C	B	1B	E7	6B
75	5	71	D0	E9	2B	5C	5E	18	D2	2C	7D	87	43	AC	37
5B	5D	34	A6	ED	83	20	4D	F5	8F	79	4C	11	66	2D	E6
B7	59	CD	22	9F	38	C7	B2	15	3A	EB	EE	29	B8	AB	F2

Gambar 4. *Perfect* SAC S-*Box*

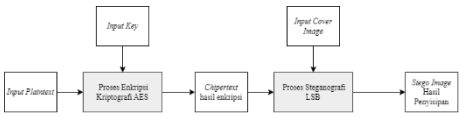
b) Inisialisasi LSB

Proses inisialisasi *Least Significant Bit* (LSB) melibatkan penyesuaian algoritma penyisipan pesan terenkripsi ke dalam bit-bit yang paling tidak signifikan pada gambar berekstensi jpg, png, dan jpeg. Pemilihan bit yang sesuai dan penyesuaian format gambar menjadi bagian krusial dari inisialisasi ini. Proses inisialisasi ini dilakukan untuk memastikan keefektifan steganografi LSB yang akan diimplementasikan.

1) Konversi Pesan ke Format *Biner*
2) Hitung Kapasitas Carrier
3) Inisialisasi Pointer pada *File Cover*
4) Iterasi Melalui Pesan Biner
5) Simpan File Hasil

c) Alur Enkripsi Teks dan Penyisipan Gambar

Berikut alur enkripsi teks dan penyisipan gambar dapat dilihat pada Gambar 5:

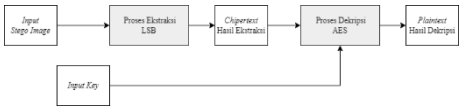


Gambar 5. Alur Enkripsi Teks dan Penyisipan Gambar

Proses dilakukan ketika *user* ingin merubah *plaintext* ke dalam bentuk *stego image* yang tidak dapat diketahui makna yang tersembunyi di dalamnya.

d) Alur Ekstraksi *Stego Image* dan Dekripsi *Ciphertext*

Berikut ekstraksi stego image dan dekripsi ciphertext dapat dilihat pada Gambar 6:



Gambar 6. Alur Ekstraksi *Stego Image* dan Dekripsi *Ciphertext*

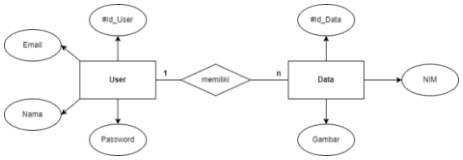
Proses dilakukan ketika user ingin merubah stego image ke dalam bentuk plaintext agar dapat diketahui makna yang tersembunyi di dalamnya.

3.2. *System and Design*

Tahapan system and design terbagi menjadi 5, yaitu *Entity Relationship Diagram* (ERD), *Use Case Diagram*, *Flowchart*, dan rancangan *user interface*.

3.2.1. *Entity Relationship Diagram* (ERD)

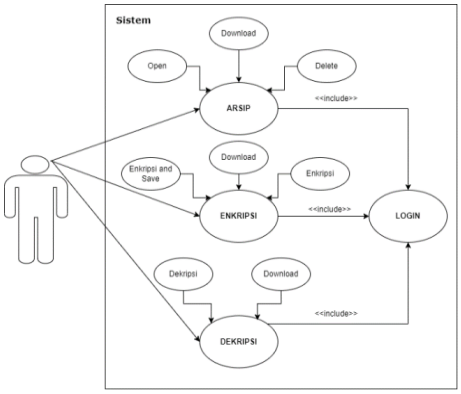
Entity Relationship Diagram (ERD) adalah suatu representasi visual yang digunakan untuk merancang struktur *database* dan mengilustrasikan hubungan antara entitas. ERD dari sistem yang sedang disusun dalam penelitian ini dapat disimak melalui Gambar 7.



Gambar 7. *Entity Relationship Diagram* Sistem

3.2.3. *Use Case Diagram*

Use Case Diagram adalah visualisasi dari alur informasi atau data yang mengalir dari *input* menuju *output*. Berikut ini adalah *Use Case Diagram* dari sistem yang sedang dikembangkan pada penelitian ini, yang dapat dilihat pada Gambar 8.



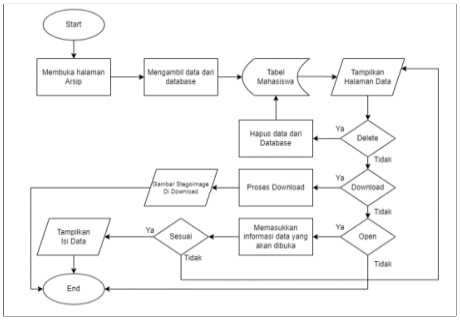
Gambar 8. *Use Case Diagram* Sistem

3.2.4. *Flowchart*

Flowchart adalah representasi visual dari langkah-langkah serta urutan prosedur suatu program. Berikut adalah *flowchart* dari sistem yang dikembangkan pada penelitian ini.

1. *Flowchart* Arsip

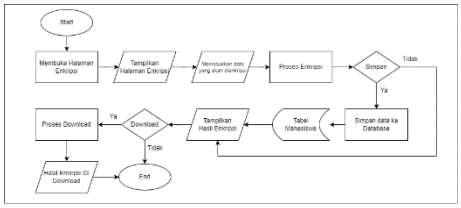
Gambaran *flowchart* pada halaman Arsip sistem yang dikembangkan pada penelitian ini dapat dilihat pada Gambar 9.



Gambar 9. Flowchart Halaman Arsip

2. *Flowchart* Enkripsi

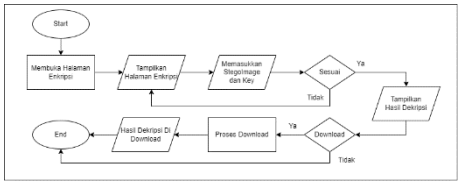
Gambaran *flowchart* pada halaman Enkripsi sistem yang dikembangkan pada penelitian ini dapat dilihat pada Gambar 10.



Gambar 10. Flowchart Halaman Enkripsi

3. *Flowchart* Dekripsi

Gambaran *flowchart* pada halaman Dekripsi sistem yang dikembangkan pada penelitian ini dapat dilihat pada Gambar 11.



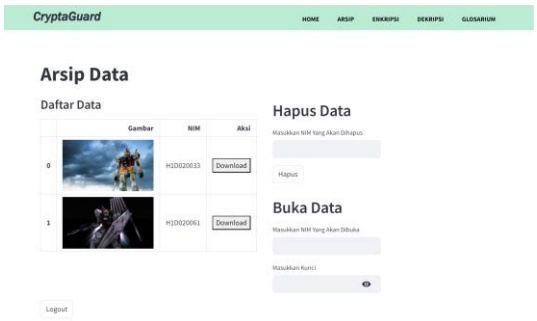
Gambar 11. *Flowchart* Halaman Dekripsi

3.3. *Implementation and Development*

Pada tahap ini, rancangan antarmuka yang sudah dibuat kemudian diimplementasikan ke dalam aplikasi berbentuk *website*. Berikut adalah beberapa halaman yang ada pada aplikasi.

3.3.3. Halaman Arsip

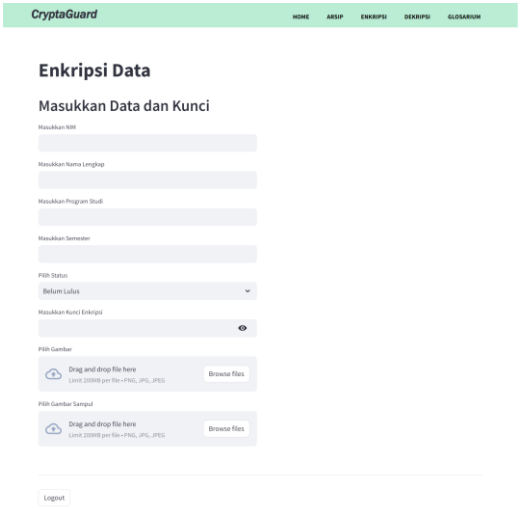
Pada halaman ini pengguna dapat melihat, menghapus, dan mendownload *stego image* serta foto di dalam *stego image* tersebut. Gambar 12 merupakan implementasi dari halaman Enkripsi.



Gambar 12. Halaman Arsip

3.3.4. Halaman Enkripsi

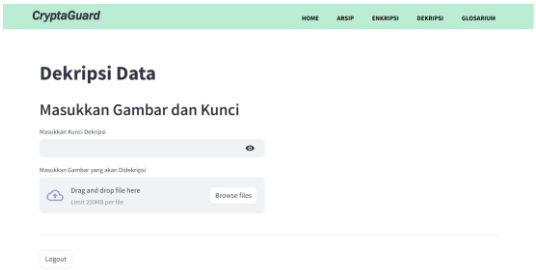
Pada halaman ini pengguna dapat menggunakan fungsi enkripsi yang tersedia dengan memasukkan informasi termasuk di dalamnya kata sandi, foto, dan *cover image* yang ingin digunakan. Gambar 13 merupakan implementasi dari halaman Enkripsi.



Gambar 13. Halaman Enkripsi

3.3.5. Halaman Dekripsi

Pada halaman ini pengguna dapat menggunakan fungsi dekripsi yang tersedia dengan memasukkan *stego image* dan kata sandi yang sesuai. Gambar 14 merupakan implementasi dari halaman Dekripsi.



Gambar 14. Halaman Dekripsi

3.4. *Testing*

Dalam fase pengujian ini, akan dijelaskan proses pengujian sistem yang dirancang untuk menilai sejauh mana kesesuaian implementasi dengan desain yang telah disiapkan sebelumnya.

3.4.1. Hasil Uji Performa

Pengujian performa bertujuan untuk mengevaluasi kecepatan enkripsi dan dekripsi pada sistem dengan modifikasi *SubByte*, dibandingkan dengan versi sebelum modifikasi. Perbandingan kecepatan enkripsi dan dekripsi menggunakan aplikasi ini terdokumentasi pada Tabel 3.

Tabel 2. Hasil Uji Performa

No	Sebelum Modifikasi		Setelah Modifikasi	
	Enkripsi	Dekripsi	Enkripsi	Dekripsi
1	684.66 milidetik	691.37 milidetik	665.41 milidetik	763.68 milidetik
2	796.74 milidetik	730.91 milidetik	748.26 milidetik	882.14 milidetik
3	695.12 milidetik	677.79 milidetik	651.23 milidetik	698.63 milidetik
4	713.16 milidetik	762.63 milidetik	683.30 milidetik	811.68 milidetik
5	699.57 milidetik	696.36 milidetik	657.86 milidetik	835.59 milidetik
6	741.88 milidetik	683.07 milidetik	664.87 milidetik	852.58 milidetik
7	785.64 milidetik	735.14 milidetik	780.29 milidetik	853.48 milidetik
8	673.11 milidetik	676.06 milidetik	647.11 milidetik	716.57 milidetik
9	826.00 milidetik	700.07 milidetik	699.27 milidetik	716.12 milidetik
10	682.25 milidetik	682.20 milidetik	665.51 milidetik	707.81 milidetik
Rata -rata	729.81 milidetik	703.56 milidetik	686.31 milidetik	783.83 milidetik

3.4.2. Hasil Uji *Cryptanalysis*

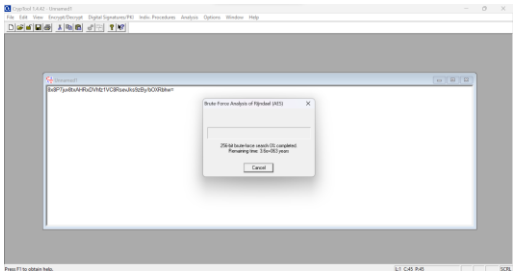
Pengujian *Cryptanalysis* bertujuan untuk mengevaluasi tingkat keamanan hasil enkripsi pada sistem yang mengalami modifikasi *SubByte*. Dalam penelitian ini, uji *cryptanalysis* dilakukan menggunakan metode *brute force* yang dibagi menjadi dua pengujian utama.

a) *Worst Case Brute Force*

Pengujian ini dilakukan dengan menghitung kemungkinan kasus terburuk dalam penggunaan *brute force* pada proses c*ryptanalysis*. Dengan penggunaan kunci sebesar 256-bit atau sebesar 32 byte pada proses enkripsi dan dekripsi kriptografi menjadikan terdapat 2^{256} kemungkinan. Dalam notasi ilmiah, jumlah kemungkinan ini dapat ditulis sebagai $1.15792089 \times 10^{77.}$

b) *Automation Brute Force*

Pengujian dilakukan menggunakan bantuan *software CrypTool* di mana digunakan *menu analysis* untuk AES pada *Symmetric Encryption* (*modern*) pada *ciphertext* yang dihasilkan pada sistem. Digunakan *key* yang sama untuk serangkaian pengujian. Pengujian menggunakan *CrypTool* dapat dilihat pada Gambar 22.



Gambar 15. Pengujian keamanan menggunakan *CrypTool*

Hal yang dibandingkan melalui aplikasi ini adalah estimasi waktu yang diperlukan oleh *software* untuk mendekripsikan *ciphertext* yang dimasukkan. Hasil uji *Cryptanalysis* dapat dilihat pada Tabel 4 berikut:

Tabel 3. Hasil Uji Automation Brute Force

No	Estimasi Waktu	
	Sebelum Modifikasi	**Setelah Modifikasi**
1	*3.4e+063 years*	*3.6e+063 years*
2	*3.4e+063 years*	*3.6e+063 years*
3	*3.4e+063 years*	*3.6e+063 years*
4	*3.4e+063 years*	*3.6e+063 years*
5	*3.4e+063 years*	*3.6e+063 years*
6	*3.4e+063 years*	*3.6e+063 years*
7	*3.4e+063 years*	*3.6e+063 years*
8	*3.4e+063 years*	*3.6e+063 years*
9	*3.4e+063 years*	*3.6e+063 years*
10	*3.4e+063 years*	*3.6e+063 years*

3.4.3. Hasil Uji *Randomness*

Tes Keacakan (Randomness) adalah metode untuk menguji seberapa acak atau kebetulan serangkaian data. Tes keacakan digunakan untuk memeriksa apakah serangkaian data mengikuti pola yang dapat diprediksi atau jika data tersebut benar-benar acak.

Terdapat beberapa jenis keacakan yang akan digunakan diantaranya:

a) *Frequency Test*: Tes sederhana yang menguji apakah frekuensi kemunculan simbol dalam serangkaian data mengikuti distribusi yang diharapkan secara acak.

b) *Poker Test*: Melibatkan pembagian data menjadi sub-blok dan penghitungan frekuensi munculnya kombinasi tertentu dari simbol-simbol dalam sub-blok tersebut.

c) *Run Test*: Menguji kecenderungan munculnya *run* (barisan simbol berturut-turut yang sama) dalam serangkaian data.

d) *Serial Test*: Menguji apakah ada korelasi antara simbol-simbol yang berdekatan dalam serangkaian data

Hasil uji perbandingan nilai randomness dari tiap tes dapat dilihat pada Tabel 5 berikut:

Tabel 4. Perbandingan Nilai *Randomness*

No	*Frequency*		*Poker*		*Run*		*Serial*	
	Sebelum	**Sesudah**	**Sebelum**	**Sesudah**	**Sebelum**	**Sesudah**	**Sebelum**	**Sesudah**
1	5.460227	5.818182	26.923077	28.837607	27.794375	42.685398	28.421707	30.248675
2	0.363636	2.051136	23.846154	14.136752	20.428292	18.618468	21.102935	9.019688
3	0.142045	3.551136	17.350427	17.28051	26.865927	18.545473	21.102709	17.522436
4	1.454545	1.642045	11.948718	18.376068	9.802990	16.570627	9.713307	14.299366
5	1.454545	6.187500	20.632479	16.324786	25.009638	30.800815	20.097124	21.298986
6	0.204545	0.960227	5.726496	14.820513	16.729245	16.187793	9.421255	10.765235
7	3.272727	0.818182	22.205128	17.487179	19.825983	40.178560	18.438130	29.911548
8	1.278409	2.505682	28.358974	21.589744	38.778023	25.654606	32.540702	18.954869
9	3.272727	0.000000	17.145299	14.957265	19.408472	30.111328	15.854916	20.146515
10	0.272727	8.642045	23.094017	24.735043	30.205188	31.678899	21.529169	31.469708
Rata-rata	1.72	3.22	19.72	18.85	23.48	27.10	19.82	20.36

3.5. *Maintenance*

Sistem penyimpanan data menggunakan kriptografi dan steganografi yang telah selesai dirancang sudah dapat digunakan untuk menyimpan data mahasiswa dengan menggunakan algoritma kriptografi AES modifikasi *SubBytes* serta Steganografi LSB untuk keamanan datanya. Pada penelitian ini, *maintenance* tidak dilakukan karena tahapan pembuatan aplikasi pada penelitian ini hanya sampai tahap *testing*.

4. DISKUSI

Hasil penelitian dan pengujian yang telah dilakukan pada *Perfect* SAC S-*box* dan algoritma modifikasi kemudian dianalisis dan dibandingkan dengan S-*box* AES dan algoritma AES asli.

4.1. Analisis Uji Performa

Berbeda dari penelitian yang sudah dilakukan oleh Rachmat dan Samsuryadi [20] serta Kuntal [21] yang melakukan uji performa dengan menghitung

lama pemrosesan enkripsi *file*, pada penelitian ini uji performa dilakukan dengan menghitung lama enkripsi *string* variabel yang didapatkan melalui masukan pengguna sehingga secara keseluruhan lama waktu pemrosesan lebih cepat.

Hasil pengujian performa yang diperlihatkan dalam Tabel 3 mengindikasikan bahwa algoritma modifikasi AES menunjukkan waktu enkripsi yang lebih cepat (rata-rata 686.31 milidetik) dibandingkan dengan AES asli (rata-rata 729.81 milidetik), dengan selisih sebesar 43.50 milidetik. Namun, terdapat perlambatan pada waktu dekripsi (rata-rata 783.83 milidetik dibandingkan dengan 703.56 milidetik untuk AES asli, dengan selisih 80.27 milidetik), menggambarkan adanya peningkatan dalam kecepatan enkripsi dan perlambatan dalam proses dekripsi. Hal ini menunjukkan bahwa algoritma modifikasi AES memiliki tingkat kesulitan yang lebih tinggi dalam proses dekripsi, yang tercermin dari lamanya waktu yang diperlukan.

4.2. Analisis Uji *Cryptanalysis*

a) *Worst Case Brute Force*

Dari uji performa yang sudah dilakukan sebelumnya diketahui sebelum modifikasi, rerata waktu yang dibutuhkan adalah 703.56 milidetik per percobaan, sedangkan setelah modifikasi rerata waktu menjadi 783.83 milidetik per percobaan.

Dengan demikian, estimasi waktu total untuk mencoba semua kemungkinan kunci adalah sekitar 9.4*e*+069 tahun sebelum modifikasi dan 10.4*e*+069 tahun setelah modifikasi yang dipresentasikan dengan notasi ilmiah.

Penghitungan ini menunjukkan bahwa peningkatan rerata waktu proses *brute force* secara signifikan berdampak pada estimasi total waktu yang diperlukan dalam proses c*ryptanalysis*. Temuan ini juga menunjukkan bahwa durasi yang dibutuhkan untuk proses c*ryptanalysis* menggunakan metode *brute force* pada algoritma AES yang telah dimodifikasi jauh lebih besar dibandingkan dengan sebelum dimodifikasi

b) *Automation Brute Force*

Berdasarkan hasil pengujian yang ditunjukan oleh Tabel 4, *CrypTool* hanya memberikan perkiraan estimasi waktu yang diperlukan oleh aplikasi untuk mendekripsikan *ciphertext* yang dimasukkan. Hal ini disebabkan oleh kompleksitas tinggi dari algoritma AES.

Hasil yang sama juga muncul pada penelitian yang dilakukan Nursalman, dkk [22] yang menghasilkan estimasi waktu dengan satuan tahun ketika melakukan penghitungan menggunakan *Cryptool*.

Berikut adalah perbandingan antara hasil pengujian menggunakan *Worst Case Brute Force* dan *Automation Brute Force* pada Tabel 10 berikut:

Tabel 5. Perbandingan Hasil *Uji Worst Case* dan *Automation Brute Force*

Metode	*Worst Case*	*Automation*
Sebelum Modifikasi	*9.4e+069 years*	*3.4e+063 years*
Setelah Modifikasi	*10.4e+069 years*	*3.6e+063 years*

Berdasarkan tabel tersebut, dapat disimpulkan bahwa implementasi modifikasi *SubByte* pada AES telah meningkatkan keamanan *ciphertext* yang dihasilkan. Hal ini terbukti dengan fakta bahwa *ciphertext* hasil modifikasi membutuhkan waktu yang lebih lama bagi aplikasi untuk melakukan *Cryptanalysis* daripada *ciphertext* sebelum modifikasi.

4.3. Analisis Uji *Randomness*

Secara umum, dalam tes keacakan, data akan dianggap lebih acak jika nilainya mendekati nilai yang diharapkan atau biasa disebut sebagai *max value* yang berlaku dalam distribusi acak. Nilai dari *max value* yang digunakan pada penelitian ini pada masing-masing tes dapat dilihat pada Tabel 8 berikut:

Tabel 6. *Max Value* dari Masing-masing Tes *Randomness*

Uji	Max Value
Frequency	3.841000
Poker	14.070000
Run	9.488000
Serial	5.991000

Nilai *max value* dapat berbeda-beda sesuai metode, aplikasi, serta kondisi ketika dilakukan penghitungan. Seperti nilai *max value* pada penelitian ini yang memiliki nilai max value yang berbeda dari penelitian yang dilakukan oleh Zagi dan Maolood [23] karena tidak menggunakan bantuan *software Cryptool* ataupun penelitian Ali, dkk [24] yang menggunakan *Cryptool* namun memiliki kondisi pengujian yang berbeda.

Dari pengujian yang sudah dilakukan diketahui bahwa *ciphertext* hasil metode AES setelah dimodifikasi memiliki nilai rata-rata yang lebih baik pada tes *Frequency* dan *Poker*. Di sisi lain nilai rata-rata dari metode AES sebelum modifikasi lebih baik pada tes *Run* dan *Serial*.

5. KESIMPULAN

Berdasarkan analisis yang sudah dilakukan pada penelitian ini, dapat disimpulkan bahwa Implementasi modifikasi *SubBytes* pada metode AES menunjukkan keamanan yang lebih baik dibandingkan dengan metode AES sebelum dimodifikasi pada uji Performa, uji *Cryptanalysis*, dan uji *Randomness*.

DAFTAR PUSTAKA

[1] N. Wachid Hidayatulloh, M. Tahir, H. Amalia, N. Afdlolul Basyar, A. F. Prianggara, and M. Yasin, “Mengenal

Advance Encrytion Standard (AES) Sebagai Algoritma Kriptografi Dalam Mengamankan Data," *Digital Transformation Technology (Digitech) | e*, vol. 3, no. 1, pp. 1–10, 2023, doi: 10.47709/digitech.v3i1.2293.

[2] M. B. Aryanto, M. Tahir, S. I. Devita, Z. N. Mustofa, Q. Ainiyah, and S. Sundoro, "Implementasi Enkrip Dan Dekrip File Menggunakan Metode Advance Encryption Standard (AES-128)," *JUISIK*, vol. 3, no. 1, pp. 89–104, 2023, [Online]. Available: http://journal.sinov.id/index.php/juisik/index HalamanUTAMAJurnal:https://journal.sinov.id/index.php

[3] Rahmaniah, M. Firman Aditya, W. Arfanda, V. Ndika purnama, and Cicilia, "Studi Algoritma Kriptografi Kunci Simetris pada Keamanan Data dengan Metode Komparasi," *JURNAL SITEBA*, vol. 2, no. 1, pp. 7–14, 2023, [Online]. Available: https://journal.iteba.ac.id/index.php/jurnalsiteba/index

[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," *JTSI*, vol. 4, no. 2, pp. 394–405, 2023.

[5] Y. Putra, Y. Yuhandri, and S. Sumijan, "Meningkatkan Keamanan Web Menggunakan Algoritma Advanced Encryption Standard (AES) terhadap Seragan Cross Site Scripting," *Jurnal Sistim Informasi dan Teknologi*, vol. 3, no. 2, pp. 56–63, Jun. 2021, doi: 10.37034/jsisfotek.v3i2.44.

[6] M. Adharis Adlani and R. E. Putra, "Pengamanan Mnemonic Phrase Menggunakan Modified Advanced Encryption Standart," *Journal of Informatics and Computer Science*, vol. 03, no. 4, pp. 425–434, 2022.

[7] Siaulhak and S. Kasma, "Sistem Pengiriman File Menggunakan Steganografi Pengolahan Citra Digital Berbasis Matriks Laboratory," *BANDWIDTH: Journal of Informatics and Computer Engineering*, vol. 01, no. 02, pp. 75–81, 2023.

[8] F. Yanti and K. Budayawan, "Implementation Steganografi Menggunakan Metode Least Significant Bit (LSB) dalam Pengamanan Informasi pada Citra Digital," *Jurnal Vocational Teknik Elektronika dan Informatika*, vol. 11, no. 1, pp. 63–70, 2023, [Online]. Available: http://ejournal.unp.ac.id/index.php/voteknika/index

[9] A. A. Wahid, "Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi," *Jurnal Ilmu-ilmu Informatika dan Manajemen STMIK* , pp. 1–5, 2020, [Online]. Available: https://www.researchgate.net/publication/346397070

[10] N. Angraini and Y. Suryanto, "MODIFICATION ADVANCED ENCRYPTION STANDARD (AES) ALGORITHM WITH PERFECT STRICT AVALANCHE CRITERION S-BOX," *Jurnal Teknik Informatika (Jutif)*, vol. 3, no. 4, pp. 897–906, Aug. 2022, doi: 10.20884/1.jutif.2022.3.4.352.

[11] N. Maulida Surbakti *et al.*, "Penggunaan Bahasa Pemrograman Python dalam Pembelajaran Kalkulus Fungsi Dua Variabel," *Algoritma : Jurnal Matematika, Ilmu pengetahuan Alam, Kebumian dan Angkasa*, vol. 2, no. 3, pp. 98–107, 2024, doi: 10.62383/algoritma.v2i3.67.

[12] Y. P. Pratama, W. Prastiwinarti, L. Ahmad, and Z. S. Mahmuda, "Perancangan Aplikasi Konversi RGB CMYK berbasis Python," *JOURNAL OF APPLIED ELECTRICAL ENGINEERING*, vol. 7, no. 2, pp. 102–105, 2023.

[13] M. Shakila, P. Akmal Aoulia, P. S. Harson, F. M. Kurniawan, and P. Rosyani, "Analasis Penerapan Python Dengan Perhitungan Pertidaksamaan," *NEWTON : Jurnal Matematika, Fisika, Algoritma dan Sains*, vol. 1, no. 1, pp. 29–33, 2023, [Online]. Available: https://ojs.jurnalmahasiswa.com/ojs/index.php/newtoon

[14] N. Cristy and F. Riandari, "Implementasi Metode Advanced Encryption Standard (AES 128 Bit) Untuk Mengamankan Data Keuangan," *JIKOMSI*, vol. 4, no. 2, pp. 75–85, 2021.

[15] K. Hasna, M. Defriani, and M. H. Totohendarto, "Redesign User Interface Dan User Experience Pada Website Eclinic Menggunakan Metode Design Thinking," *KLIK: Kajian Ilmiah Informatika dan Komputer*, vol. 4, no. 1, pp. 84–92, 2023, doi: 10.30865/klik.v4i1.1072.

[16] 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. 163–171, 2022, doi: 10.47709/jpsk.v2i1.1390.

[17] A. Davy Wiranata and R. T. Aldisa, "Aplikasi Steganografi Menggunakan Least Significant Bit (LSB) dengan Enkripsi Caesar Chipper dan Rivest Code 4 (RC4)

Menggunakan Bahasa Pemrograman JAVA," *Jurnal Teknologi Informasi dan Komunikasi)*, vol. 5, no. 3, p. 2021, 2021, doi: 10.35870/jti.

[18] A. E. Setiawan and A. Pasaribu, "Penerapan Steganografi Pada Citra Digital Menggunakan Metode Least Significant Bit (LSB) Kombinasi RC4 Berbasis Mobile Android," *Aisyah Journal of Informatics and Electrical Engineering*, vol. 2, no. 1, pp. 18–28, 2020, [Online]. Available: http://jti.aisyahuniversity.ac.id/index.php/AJIEE

[19] Alamsyah, "A Novel Construction of Perfect Strict Avalanche Criterion S-box using Simple Irreducible Polynomials," *Scientific Journal of Informatics*, vol. 7, no. 1, pp. 10–22, 2020, [Online]. Available: http://journal.unnes.ac.id/nju/index.php/sji

[20] N. Rachmat and Samsuryadi, "Performance Analysis of 256-bit AES Encryption Algorithm on Android Smartphone," *J Phys Conf Ser*, vol. 1196, no. 1, pp. 1–6, Apr. 2019, doi: 10.1088/1742-6596/1196/1/012049.

[21] P. Kuntal, "Performance Analysis of AES, DES and Blowfish Cryptographic Algorithms on Small and Large Data Files," *International Journal of Information Technology (Singapore)*, vol. 11, no. 4, pp. 813–819, Dec. 2019, doi: 10.1007/s41870-018-0271-4.

[22] M. Nursalman, E. P. Nugroho, and F. R. A. Nur, "Implementation of Rivest Chiper Cryptography (RC6) with One Time Password (OTP) and Two Central Facilities Protocol in Complaint Service System," in *Proceedings of the 7th Mathematics, Science, and Computer Science Education International Seminar, MSCEIS 2019*, European Alliance for Innovation, 2020. doi: 10.4108/eai.12-10-2019.2296270.

[23] H. R. Zagi and A. A. T. Maolood, "A New Key Generation to Greate Enhanced Security Version of AES Encryption Method," *Journal of College of Education*, vol. 2, no. 1, pp. 1–16, 2021.

[24] H. J. Ali, T. M. Jawad, and H. Zuhair, "Data Security Using Random Dynamic Salting and AES Based on Master-Slave Keys for Iraqi Dam Danagement Dystem," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 2, pp. 1018–1029, Aug. 2021, doi: 10.11591/ijeecs.v23.i2.pp1018-1029.