Accredited SINTA 5
72/E/KPT/2024

Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)
Vol. 8 No. 4, Agustus 2025
P-ISSN 2620-8342
E-ISSN 2621-3052

Analisis Kriptografi pada Keamanan E-mail di Dinas Kominfotik Provinsi Gorontalo mengggunakan Algoritma *Advanced Encryption Standard* (*AES*) 256 Bit

Fatmawati Meilan Harun[1*], Wahyudin Hasyim[2], Alter Lasarudin[3]
[1,2,3]Prodi Sistem Informasi, Universitas Muhammadiyah Gorontalo
Jl. Prof. Dr. H. Mansoer Pateda, Desa Pentadio Timur, Kec. Telaga Biru, Kab. Gorontalo, Gorontalo 96181, Indonesia
E-mail: fatmawatyharun02@gmail.com

Submitted Date : ***Accepted Date :***

Abstrak - Dalam konteks perkembangan teknologi yang pesat, keamanan informasi menjadi krusial, terutama dalam komunikasi resmi di instansi pemerintah. Dinas Komunikasi dan Informatika (Kominfotik) Provinsi Gorontalo menghadapi tantangan dalam melindungi data sensitif yang dikirim melalui email. Penggunaan metode kriptografi, khususnya algoritma Advanced Encryption Standard (AES), sangat penting untuk menjaga kerahasiaan dan integritas informasi. Penelitian ini bertujuan untuk menganalisis penerapan AES dalam sistem email dan tantangan yang dihadapi dalam implementasi Tanda Tangan Elektronik (TTE). Melalui metode AES-256, peneliti menganalisis bahwa meskipun AES efektif, terdapat kelemahan dalam implementasinya yang dapat membahayakan keamanan data. Rekomendasi diberikan untuk mengintegrasikan metode kriptografi tambahan dan melaksanakan pelatihan rutin, serta evaluasi sistem secara berkala. Hasil penelitian ini diharapkan dapat menjadi acuan bagi instansi pemerintah lain dalam meningkatkan keamanan informasi.
Kata kunci : Kriptografi, Algoritma Advanced Encryption Standard (AES) 256 Bit, Tanda Tangan Elektronik (TTE).

Abstract - *In the context of rapid technological developments, information security has become crucial, especially in official communications within government agencies. The Gorontalo Province Communication and Informatics Office (Kominfotik) faces challenges in protecting sensitive data sent via email. The use of cryptographic methods, specifically the Advanced Encryption Standard (AES) algorithm, is crucial for maintaining the confidentiality and integrity of information. This study aims to analyse the application of AES in email systems and the challenges faced in implementing Electronic Signatures (TTE). Using the AES-256 method, the researcher analyses that although AES is effective, there are weaknesses in its implementation that can compromise data security. Recommendations are given for integrating additional cryptographic methods and conducting routine training and periodic system evaluations. The results of this research are expected to serve as a reference for other government agencies in improving information security.*
Keywords: Cryptography, Advanced Encryption Standard (AES) 256-bit Algorithm, Electronic Signature (TTE).

1. Pendahuluan

Dalam era digital yang semakin berkembang, keamanan informasi menjadi salah satu aspek yang sangat penting, terutama dalam konteks komunikasi resmi di instansi pemerintah Sebagai lembaga yang bertanggung jawab atas pengelolaan informasi publik, Dinas Komunikasi dan Informatika (Kominfotik) Provinsi Gorontalo, penggunaan email sebagai sarana komunikasi sehari-hari membawa tantangan tersendiri terkait perlindungan data. Informasi yang dikirimkan melalui email sering kali mengandung data sensitif yang jika jatuh ke tangan yang salah dapat menimbulkan dampak negatif, baik bagi instansi maupun masyarakat. Oleh karena itu, menggunakan metode kriptografi adalah langkah penting untuk menjaga kerahasiaan dan integritas data yang dikirimkan.

Kriptografi merupakan bidang yang sangat penting dalam menjaga keamanan informasi Di era digital saat ini, perlindungan informasi menjadi semakin penting. Seiring dengan pesatnya perkembangan teknologi informasi dan komunikasi, ancaman terhadap data sensitif juga turut meningkat, baik akibat serangan siber maupun kebocoran data. Untuk mengatasi hal ini, kriptografi memainkan peran vital dalam menjaga kerahasiaan, integritas, dan keaslian informasi dengan cara mengenkripsi dan mendekripsi data. Proses enkripsi mengubah data asli (plaintext) menjadi bentuk terenkripsi (ciphertext) yang tidak dapat dibaca tanpa kunci yang sesuai. Dengan demikian, hanya pihak yang berwenang yang dapat mengakses informasi tersebut. Perlindungan semacam ini sangat penting terutama dalam komunikasi resmi, transaksi finansial, dan penyimpanan data yang bersifat rahasia [1]. Kriptografi menggunakan metode Advanced Encryption Standard

Copyright © Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)

Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)
Vol. 8 No. 4, Agustus 2025
P-ISSN 2620-8342
E-ISSN 2621-3052

(AES) terbukti sangat efektif dalam melindungi data dari akses tidak sah. AES, yang merupakan algoritma enkripsi simetris, mampu mengenkripsi data dengan ukuran blok 128 bit dan mendukung kunci dengan panjang 128, 192, dan 256 bit. Advanced Encryption Standard (AES) 256 bit adalah algoritma enkripsi simetris yang dirancang untuk memberikan tingkat keamanan tinggi dalam pengolahan data. Dengan panjang kunci 256-bit, AES mampu melindungi informasi sensitif dari berbagai ancaman, termasuk serangan yang mencoba menebak kunci secara berulang [2]. Algoritma ini telah diadopsi secara luas dalam berbagai aplikasi, mulai dari komunikasi pemerintah hingga transaksi keuangan, berkat kemampuannya dalam menjaga kerahasiaan dan integritas data. Dengan demikian, AES 256-bit menjadi pilihan utama untuk memastikan keamanan informasi di era digital saat ini.

Tujuan penelitian ini adalah untuk menganalisis penerapan metode kriptografi, khususnya algoritma Advanced Encryption Standard (AES), dalam sistem email di Dinas Komunikasi dan Informatika (Kominfotik) Provinsi Gorontalo, serta mengevaluasi dampaknya terhadap kerahasiaan dan integritas informasi yang dikirimkan. Selain itu, penelitian ini bertujuan untuk mengidentifikasi tantangan yang dihadapi dalam melindungi data sensitif dan memberikan rekomendasi untuk meningkatkan sistem keamanan informasi mereka berdasarkan hasil analisis penerapan AES.

2. Tinjauan Pustaka

2.1. Analisis

Analisis adalah proses penyelidikan yang bertujuan untuk memahami suatu fenomena atau peristiwa dengan cara menguraikan komponen – komponen. Proses ini melibatkan pengumpulan dan evaluasi data untuk mengidentifikasi pola, hubungan, dan penyebab yang mendasari suatu masalah [3].

2.2. Sistem

Sistem didefinisikan sebagai sekumpulan dua atau lebih komponen yang saling terhubung dan berinteraksi untuk mencapai suatu tujuan. Biasanya, sistem ini dibagi menjadi sub-sistem yang lebih kecil yang mendukung keseluruhan sistem. Setiap sistem memiliki karakteristik atau sifat tertentu [4].

2.3. Informasi

Informasi merupakan hasil pengolahan data, namun tidak semua hasil pengolahan tersebut dapat dianggap sebagai informasi. Hasil pengolahan data yang tidak memiliki arti atau makna, serta tidak bermanfaat bagi seseorang, tidak dapat disebut informasi untuk orang tersebut [5].

2.4. Sistem Informasi

Sistem informasi merupakan sebuah sistem yang terdiri dari berbagai komponen yang saling terhubung dan bekerja sama dalam mengumpulkan, mengolah, menyimpan, serta menyebarkan informasi yang bermanfaat bagi para penggunanya [6].

2.5. Kriptografi

Kriptografi adalah bidang ilmu yang mempelajari teknik-teknik matematis untuk melindungi berbagai aspek keamanan informasi, termasuk kerahasiaan, integritas data, dan otentikasi. Tujuan kriptografi adalah menyediakan layanan keamanan informasi, yang mencakup kerahasiaan, integritas data, otentikasi, dan non-repudiasi [7]. Dalam menjaga kerahasiaan data dengan kriptografi, data sederhana yang dikirim (plaintext) diubah ke dalam bentuk data sandi (chipertext), kemudian data sandi tersebut hanya dapat dikembalikan kebentuk data sebenarnya hanya dengan menggunakan kunci (key) tertentu yang dimiliki oleh pihak yang sah saja, ini menyebabkan pihak lain yang tidak memiliki kunci tidak akan dapat membaca data yang sebenarnya sehingga data akan tetap terjaga kerahasiannya [8].

2.6. Enkripsi

Enkripsi adalah proses mengubah informasi atau data menjadi bentuk yang tidak dapat dibaca oleh pihak yang tidak berwenang, dengan tujuan melindungi kerahasiaan dan integritas data tersebut. Proses ini menggunakan algoritma kriptografi yang kompleks untuk mengacak data asli (plaintext) menjadi format yang tidak dapat dipahami (chipertext), sehingga hanya individu atau sistem yang memiliki kunci dekripsi yang tepat yang dapat mengembalikannya ke bentuk aslinya [9].

2.7. Deskripsi

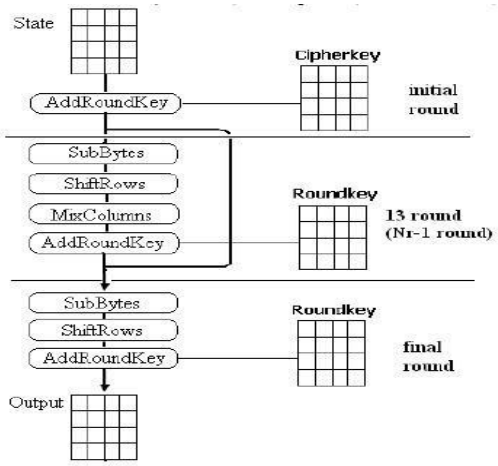
Dekripsi adalah proses yang berlawanan dengan enkripsi, di mana data yang telah dienkripsi (chipertext) diubah kembali menjadi bentuk aslinya (plaintext) sehingga dapat dibaca dan dipahami. Proses ini memerlukan kunci dekripsi yang sesuai, yang memungkinkan pihak yang berwenang untuk mengakses informasi yang sebelumnya telah dilindungi [10].

2.8. Enkripsi *Advanced Encryption System (AES)*

AES (Advanced Encryption System), adalah salah satu algoritma block cipher yang menggunakan kunci simetris untuk enkripsi dan dekripsi pesan. Algoritma AES memiliki variasi kunci yang berbeda, yaitu 128-bit, 192-bit, dan 256-bit. Perbedaan antara ketiga kunci tersebut terletak pada panjang kunci dan jumlah putaran (round) yang digunakan [11]. AES 256-bit banyak diterapkan dalam berbagai bidang, termasuk komunikasi, penyimpanan data, dan transaksi keuangan, di mana perlindungan data menjadi prioritas utama. Proses enkripsi

Copyright © Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)

dan dekripsi yang dilakukan oleh AES 256-bit melibatkan beberapa putaran transformasi, yang masing-masing bertujuan untuk meningkatkan kerahasiaan data yang diproses [12]. Ada empat jenis transformasi yang digunakan pada proses enkripsi algoritma AES yaitu SubBytes, ShiftRows, Mixcolumns, dan AddRoundKey.



Gambar 1. *flowchart* algoritma AES 256 bit

Proses enkripsi yang dilakukan menggunakan algoritma AES yaitu:

a. SubBytes

SubByte merupakan transformasi byte dimana setiap elemen pada state akan dipetakan dengan menggunakan sebuah tabel substitusi (S-Box). Untuk setiap byte pada state dinyatakan dengan S'[r, c]. S'[r, c] adalah elemen di dalam tabel subtitusi yang merupakan perpotongan baris (x) dengan kolom (y).

b. Shiftrows

Shiftrows pada dasarnya adalah proses pergeseran bit dimana bit paling kiri akan dipindahkan menjadi bit paling kanan (rotasi bit). Namun jumlah pergeseran yang dilakukan berbeda, tergantung untuk setiap barisnya. Baris pertama tidak terjadi pergeseran. Setiap byte dari baris kedua pada matriks state digeser satu byte ke kiri. Selanjutnya baris ketiga digeser ke kiri sebanyak dua byte dan pada baris ke empat digeser ke kiri sebanyak tiga byte. Proses ini bertujuan untuk menghasilkan diffusion yakni dengan menyebarkan pengaruh transformasi nonlinear pada baris-baris matriks state untuk putaran selanjutnya

c. MixColumns

Pada proses MixColumns, tiap kolom dari matriks state dilakukan operasi perkalian. Hal ini bertujuan untuk menyebarkan pengaruh setiap bit plaintext dan chiperkey terhadap chipertext yang dihasilkan, pada arah kolom matriks state. Setiap kolom matriks state diperlakukan sebagai polinomial empat suku dalam Galois field, kemudian dikalikan dengan modulo (X 8 + X4 + X + X + 1). Operasi MixColumns juga dapat dipandang sebagai perkalian matriks, dengan mengalikan empat bilangan di dalam Galois field MixColumns juga disebut sebagai proses mengalikan setiap kolom dengan matriks berikut:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

Gambar 2. MixColumns

Atau : $A(x) = \{03\}x^2 + \{01\}x^2 + \{01\}x + \{02\}$

d. AddRoundKey

Dalam tahap AddRoundKey ini, cipherkey yang telah ada di ekspansikan terlebih dahulu maka akan di dapat roundkey yang akan digunakan untuk proses selanjutnya. Kemudian setiap byte dari matriks state keluaran proses MixColumns dilakukan operasi XOR dengan setiap byte dari roundkey. Proses round atau proses SubBytes, ShiftRows, MixColumns, dan AddRoundKey dilakukan hingga putaran ke - n dengan cara yang sama. Sedangkan untuk putaran terakhir atau disebut juga final round proses SubBytes, ShiftRows, dan AddRoundKey tetap dilakukan tetapi proses MixColumns tidak dilakukan [13].

Copyright © Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)

Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)
Vol. 8 No. 4, Agustus 2025
P-ISSN 2620-8342
E-ISSN 2621-3052

3. Metode Penelitian

Penelitian ini menggunakan pendekatan kualitatif deskriptif dengan studi kasus untuk menganalisis penerapan algoritma Advanced Encryption Standard (AES) 256-bit dalam keamanan email di Dinas Komunikasi dan Informatika (Kominfotik) Provinsi Gorontalo. Penelitian ini menerapkan algoritma Advanced Encryption Standard (AES) dengan panjang kunci 256-bit sebagai metode untuk mengamankan passphrase yang digunakan dalam sistem tanda tangan elektronik (TTE) di Dinas Komunikasi, Informatika, dan Statistik Provinsi Gorontalo. Passphrase berfungsi sebagai elemen autentikasi yang membuka akses terhadap kunci privat dalam sertifikat elektronik, sehingga perlindungan terhadap passphrase menjadi hal yang krusial. Dalam simulasi ini, digunakan contoh passphrase "GORONTALOPROV" untuk menunjukkan proses enkripsi dan dekripsi menggunakan AES-256. Seluruh proses diimplementasikan dalam bahasa pemrograman Python.Tahapan Implementasi AES 256 Bit sebagai berikut:

1. Persiapan Plainteks

Passphrase awal, yaitu string “GORONTALOPROV”, pertama-tama dikonversi ke bentuk biner (bytes). Hal ini diperlukan karena algoritma AES bekerja dengan data dalam bentuk byte, bukan string biasa. Namun, AES memiliki ukuran blok tetap yaitu 16 byte. Oleh karena itu, passphrase yang panjangnya kurang dari 16 byte harus ditambah dengan padding agar memenuhi ukuran blok tersebut.

Padding yang digunakan dalam implementasi ini adalah metode PKCS7, yang secara otomatis menambahkan byte tertentu pada akhir data agar panjangnya menjadi kelipatan dari ukuran blok AES. Proses padding ini penting untuk menjaga konsistensi dan mencegah kesalahan saat enkripsi.

2. Penentuan Kunci AES-256.

AES-256 mensyaratkan kunci sepanjang 32 byte. Pada implementasi ini digunakan kunci simulasi yang telah ditentukan, yaitu b'1234567890ABCDEF1234567890ABCDEF'. Kunci ini bersifat statis hanya untuk keperluan simulasi, namun dalam penerapan nyata kunci harus dijaga kerahasiaannya dan dikelola secara aman.

3. Enkripsi Plainteks

Setelah passphrase siap (dalam bentuk byte dan telah dipadding), proses enkripsi dilakukan menggunakan mode ECB (Electronic Codebook) dari AES. Mode ECB adalah mode enkripsi dasar yang memproses blok data secara terpisah.

Enkripsi ini menghasilkan ciphertext — data terenkripsi yang tidak dapat dibaca secara langsung dan menjamin kerahasiaan passphrase selama proses penyimpanan atau transmisi.

4. Penyajian Hasil Ciphertext

Ciphertext yang dihasilkan berupa rangkaian byte yang tidak terbaca. Agar mudah dianalisis dan dipresentasikan, ciphertext diubah ke dalam format heksadesimal (hex). Data hex ini dapat ditampilkan dalam format 4 byte per baris untuk memperjelas struktur blok hasil enkripsi.

5. Dekripsi Ciphertext

Untuk memastikan bahwa proses enkripsi berjalan benar, dilakukan proses dekripsi ciphertext menggunakan kunci yang sama. Setelah didekripsi, hasilnya masih berupa data yang dipadding.

6. Penghilangan Padding

Setelah dekripsi, padding yang sebelumnya ditambahkan pada langkah awal harus dihapus agar data kembali ke bentuk aslinya. Proses unpadding ini menggunakan metode yang sama (PKCS7), sehingga diperoleh kembali passphrase asli “GORONTALOPROV”.

7. Validasi Keaslian Data

Hasil unpadding ini kemudian dibandingkan dengan passphrase awal sebagai validasi bahwa proses enkripsi dan dekripsi berhasil tanpa perubahan data. Jika hasilnya sama persis, maka proses pengamanan dengan AES-256 berhasil diterapkan.

4. Hasil dan Pembahasan

4.1. Implementasi Enkripsi AES-256 Bit

Tahapan awal dari proses enkripsi dengan algoritma AES-256 Bit meliputi persiapan plainteks dan kunci, konversi ke blok 128-bit, serta penambahan padding untuk memastikan data dapat diproses oleh algoritma.

1. Persiapan Plainteks dan Kunci

- Plainteks

“GORONTALOPROV”

- Kunci

"GORONTALOPROV123456789012345678 9"

(Diperpanjang menjadi 32 karakter untuk memenuhi panjang kunci AES-256).

2. Konversi Plainteks ke Blok 128-bit

Copyright © Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)

Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)
Vol. 8 No. 4, Agustus 2025
P-ISSN 2620-8342
E-ISSN 2621-3052

a. Pengkodean Plainteks:
Plainteks akan diubah menjadi representasi heksadesimal. Setiap karakter dalam plainteks akan dikonversi menjadi kode ASCII dan kemudian ke heksadesimal.
Contoh Konversi :
•"G" = 0x47
•"O" = 0x4f
•"R" = 0x52
•"O" = 0x4f
•"N" = 0x4e
•"T" = 0x54
•"A" = 0x41
•"L" = 0x4c
•"O" = 0x4f
•"P" = 0x50
•"R" = 0x52
•"O" = 0x4f
•"V" = 0x56
b. Padding:
AES memerlukan panjang data yang merupakan kelipatan dari 128 bit (16 byte). Karena panjang "GORONTALOPROV" adalah 13 byte, maka perlu ditambahkan 3 byte padding. Padding menggunakan metode PKCS#7, di mana byte padding diisi dengan jumlah byte yang ditambahkan, yaitu 0x03, 0x03, 0x03.
Hasil akhir plainteks setelah ditambahkan padding:
[0x47, 0x4F, 0x52, 0x4F,
0x4E, 0x54, 0x41, 0x4C,
0x4F, 0x50, 0x52, 0x4F,
0x56, 0x03, 0x03, 0x03]
3. Tahap Awal AddRoundKey
Tahapan AddRoundKey adalah proses pertama dalam algoritma AES sebelum memasuki putaran enkripsi utama. Pada tahap ini, blok plainteks yang telah melalui konversi dan penambahan padding akan digabungkan dengan kunci enkripsi menggunakan operasi XOR (Exclusive OR). Tujuannya adalah untuk mengaitkan data dengan kunci, sehingga setiap bit dari blok plainteks terpengaruh oleh kunci yang digunakan.
a. Round Key ke-0
Kunci AES-256 memiliki panjang 256-bit (32 byte). Untuk proses AddRoundKey awal, kita menggunakan 16 byte pertama dari kunci:
•Kunci asli: "GORONTALOPROV1234567890123456789"
•16 karakter pertama: "GORONTALOPROV123"
[0x47, 0x4F, 0x52, 0x4F,
0x4E, 0x54, 0x41, 0x4C,
0x4F, 0x50, 0x52, 0x4F,
0x56, 0x31, 0x32, 0x33]
b. Proses XOR dengan Plainteks
Pada tahap ini, setiap byte dari plainteks akan di-XOR dengan byte yang sesuai dari Round Key ke-0. Operasi XOR antara dua byte menghasilkan 0 jika keduanya sama, dan 1 jika keduanya berbeda. XOR bekerja dengan aturan dasar:
•Jika dua bit bernilai sama (misalnya 0 XOR 0 atau 1 XOR 1), maka hasilnya adalah 0.
•Jika dua bit bernilai berbeda (misalnya 0 XOR 1 atau 1 XOR 0), maka hasilnya adalah 1.
Jadi, hasil dari operasi XOR antara plainteks dan Round Key ke-0 adalah sebagai berikut:
[0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
0x32, 0x31, 0x30, 0x30]
4. Proses Enkripsi Ronde Pertama
Setelah tahap AddRoundKey, kita akan memasuki proses enkripsi yang terdiri dari 14 ronde. Setiap ronde mencakup beberapa langkah penting:
a. SubBytes (ronde pertama)
Tahap SubBytes merupakan proses substitusi non-linear pada algoritma AES yang bertujuan untuk meningkatkan kompleksitas data sehingga lebih sulit dianalisis atau diprediksi. Pada langkah ini, setiap byte dari blok data hasil XOR awal akan digantikan dengan nilai baru berdasarkan tabel substitusi yang disebut S-Box.

Copyright © Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)

Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)
Vol. 8 No. 4, Agustus 2025
P-ISSN 2620-8342
E-ISSN 2621-3052

Tabel 1. S-Box 16x16

HEX	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	OD	0E	Of
00	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
01	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
02	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
03	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
04	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
05	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
06	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
07	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
08	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
09	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5F	0B	DB
0A	E0	32	3A	0A	F4	C6	E8	DD	74	1E	1F	6D	6C	56	57	A9
0B	86	0E	73	9B	83	9E	8D	8B	70	3E	B5	66	48	03	F6	0B
0C	6F	79	80	75	01	57	89	8E	7F	5D	1D	40	D4	3D	8A	CA
0D	3C	63	8F	5C	2B	1A	09	12	6E	53	7F	CB	2A	3F	21	C0
0E	60	92	08	5B	D5	9F	4C	3E	2F	8C	3B	05	7A	9E	1F	6A
0F	9A	18	5F	E0	D8	B4	6F	94	87	1D	60	7E	9B	3F	0E	1A

Hasil dari operasi XOR antara plainteks dan Round Key ke-0
[0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00,
0x32, 0x31, 0x30, 0x30]
Proses penggantian dilakukan dengan mencari nilai heksadesimal setiap byte pada baris dan kolom tabel S-Box.
Langkah 1 – Konversi ke biner
0x00 = 0000 0000
•4 bit atas (baris): 0000 = 0x0
•4 bit bawah (kolom): 0000 = 0x0
Langkah 2 – Subtitusi di Tabel S-box
Cari baris 0, kolom 0 pada tabel S-box:
-0x00 setelah proses SubBytes akan menjadi 0x63.Jadi semua nilai untuk 0x00 hasil setelah subtitusi yaitu 0x63.
-0x32 Cari Baris 0x3, Kolom 0x2 hasilnya 0x23
-0x31 Cari Baris 0x3, Kolom 0x1 hasilnya 0xC7
-0x30 Cari Baris 0x3, Kolom 0x0 hasilnya 0x04
Setelah dilakukan substitusi dengan S-Box, blok data tersebut berubah menjadi:
[0x63, 0x63, 0x63, 0x63,
0x63, 0x63, 0x63, 0x63,
0x63, 0x63, 0x63, 0x63,
0x23, 0xC7, 0x04, 0x04]
Perubahan ini bersifat tetap karena S-Box bersifat statis, tetapi hasil akhirnya tergantung dari isi awal plainteks dan kunci yang digunakan. Transformasi SubBytes ini berperan penting dalam memberikan keamanan pada AES karena mampu menyembunyikan pola dari data asli sebelum dilanjutkan ke proses berikutnya.
b. ShiftRows(ronde pertama)
Tahap ShiftRows adalah proses permutasi baris pada matriks 4×4 hasil dari SubBytes. Proses ini dilakukan dengan menggeser setiap baris ke kiri sebanyak jumlah tertentu, tergantung pada indeks barisnya:
•Baris ke-0: tidak digeser

Copyright © Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)

Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)
Vol. 8 No. 4, Agustus 2025
P-ISSN 2620-8342
E-ISSN 2621-3052

•Baris ke-1: digeser 1 byte ke kiri
•Baris ke-2: digeser 2 byte ke kiri
•Baris ke-3: digeser 3 byte ke kiri
Sebelum dilakukan pergeseran, hasil dari SubBytes disusun ke dalam bentuk matriks 4×4 secara kolom-major order, artinya data diisi per kolom dari atas ke bawah.
Matriks hasil SubBytes:
0x63	0x63	0x63	0x63
0x63	0x63	0x63	0x63
0x63	0x63	0x63	0x63
0x23	0xC7	0x04	0x04
Pergeseran ShiftRows:
•Baris 0 (tidak digeser):
o[0x63, 0x63, 0x63, 0x63]
•Baris 1 (geser 1 ke kiri):
o[0x63, 0x63, 0x63, 0x63] → tetap sama karena nilainya seragam
•Baris 2 (geser 2 ke kiri):
o[0x63, 0x63, 0x63, 0x63] → juga tetap sama
•Baris 3 (geser 3 ke kiri):
oSebelumnya : [0x23, 0xC7, 0x04, 0x04]
oSetelah geser kiri 3 posisi: [0x04, 0x04, 0x23, 0xC7]
Hasil setelah Shifrows:
0x63	0x63	0x63	0x63
0x63	0x63	0x63	0x63
0x63	0x63	0x63	0x63
0x04	0x04	0x23	0xC7
Setelah tahap ini, hasil dari ShiftRows akan digunakan pada proses selanjutnya, yaitu MixColumns. Proses pergeseran ini penting karena memberikan difusi pada data, artinya memperluas pengaruh tiap byte agar perubahan kecil pada masukan berdampak besar pada hasil akhirnya.

c. Mix colums (ronde pertama)

MixColumns adalah salah satu transformasi penting dalam algoritma AES yang berfungsi untuk mencampurkan setiap kolom dalam matriks blok data (State). Transformasi ini meningkatkan difusi, yaitu penyebaran pengaruh dari setiap byte input ke seluruh blok, sehingga menambah kekuatan kriptografi terhadap serangan analisis statistik.
Struktur Matriks State Setelah ShiftRows
Setelah tahap ShiftRows, data plainteks yang telah diproses memiliki bentuk sebagai berikut (disusun secara kolom dalam bentuk matriks 4x4):
0x63 0x63 0x63 0x63
0x63 0x63 0x63 0x63
0x63 0x63 0x63 0x63
0x04 0x04 0x23 0xc7
Matriks Transformasi MixColumns
Dalam AES, setiap kolom dari state akan dikalikan (dalam Galois Field GF(2^8)) dengan matriks tetap sebagai berikut:
| 02 03 01 01 |
| 01 02 03 01 |
| 01 01 02 03 |
| 03 01 01 02 |
Operasi ini dilakukan secara per kolom, bukan per baris.HasilMixColumns akan menjadi:
Baris 1 = (0x02 * 0x63) ^ (0x03 * 0x63) ^ (0x01 * 0x63) ^ (0x01 * 0x04)
Baris 2 = (0x01 * 0x63) ^ (0x02 * 0x63) ^ (0x03 * 0x63) ^ (0x01 * 0x04)
Baris 3 = (0x01 * 0x63) ^ (0x01 * 0x63) ^ (0x02 * 0x63) ^ (0x03 * 0x23)
Baris 4 = (0x03 * 0x63) ^ (0x01 * 0x63) ^ (0x01 * 0x63) ^ (0x02 * 0xc7)
Untuk proses perhitungan pada tahapan MixColumns, penulis menggunakan bahasa pemrograman Python. Dengan memanfaatkan operasi pada Galois Field (GF(2^8)), proses perkalian matriks dan penggabungan XOR dapat dihitung secara otomatis dan akurat. Pendekatan ini digunakan agar perhitungan tidak dilakukan secara manual dan menghindari kemungkinan kesalahan perhitungan.
Hasil ShiftRows ronde pertama
[0x63, 0x63, 0x63, 0x63],
[0x63, 0x63, 0x63, 0x63],

Copyright © Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)

[0x63, 0x63, 0x63, 0x63],
[0x04, 0x04, 0x23, 0xC7]
Hasil MixColumns setelah dijalankan divscode menggunakan bahasa python:
Berdasarkan perhitungan menggunakan standar operasi dalam GF(2^8), hasil dari transformasi MixColumns untuk masing-masing kolom adalah sebagai berikut:
0x04 0x04 0x23 0xC7
0x04 0x04 0x23 0xC7
0xCA 0Xca 0xE3 0x94
0xAD 0xAD 0xE3 0x30
d. AddRoundKey (ronde pertama)
Tahapan selanjutnya setelah proses MixColumns adalah AddRoundKey. AddRoundKey merupakan proses di mana setiap byte pada matriks state akan di-XOR (Exclusive OR) dengan byte yang bersesuaian dari round key yang sesuai dengan rondenya. Pada putaran ini, round key yang digunakan adalah round key pertama dari ekspansi kunci 256-bit, yaitu 16 byte pertama. Proses XOR dilakukan byte demi byte sesuai posisi dalam matriks 4x4.
Proses ini sangat penting karena AddRoundKey menjadi satu-satunya tahap dalam algoritma AES yang secara langsung memasukkan unsur kunci rahasia ke dalam proses enkripsi. Tanpa tahapan ini, cipher tidak akan mampu memberikan keamanan karena tidak mengandung elemen rahasia yang membedakan satu enkripsi dari yang lain.
Untuk perhitungan XOR antara state hasil MixColumns dan round key ronde pertama, penulis menggunakan bahasa pemrograman Python yang dijalankan melalui aplikasi Visual Studio Code (VSCode).
Hasil MixColumns
[0x04, 0x04, 0x23, 0xC7],
[0x04, 0x04, 0x23, 0xC7],
[0xCA, 0xCA, 0xE3, 0x94],
[0xAD, 0xAD, 0xE3, 0x30]

Hasil AddRoundkey setelah dijalankan divscode menggunakan bahasa python :
['0x43', '0x4b', '0x71', '0x88']
['0x4a', '0x50', '0x62', '0x8b']
['0x85', '0x9a', '0xb1', '0xdb']
['0xfb', '0x9c', '0xd1', '0x3']
Output AddRoundKey merupakan representasi akhir dari matriks state setelah dilakukan proses XOR dengan round key. Setiap byte dalam hasil ini adalah hasil dari operasi state_byte $\oplus$ round_key_byte berdasarkan posisi baris dan kolom yang sama dalam kedua matriks.

Kemudian dilanjutkan sampai ke ronde 14. Setelah menganalisis ronde pertama dari AES 256 dengan detail, di temukan bahwa pola yang sama berlanjut hingga ronde ke-14.

4.2. Hasil Akhir Enkripsi (ciphertext)

Pada tahap akhir enkripsi AES-256, proses AddRoundKey ronde ke-14 menghasilkan ciphertext yang merupakan keluaran akhir dari algoritma enkripsi. Ciphertext ini adalah hasil XOR antara state terakhir dan kunci putaran ke-14. Proses ini memastikan bahwa seluruh 14 ronde transformasi telah diterapkan secara lengkap pada data asli, sehingga meningkatkan keamanan terhadap serangan kriptografi.
Ciphertext yang dihasilkan selanjutnya dapat dikirim melalui media komunikasi seperti email secara aman, karena telah terlindungi oleh algoritma enkripsi simetris AES-256 yang kuat.
Berikut adalah hasil akhir enkripsi (ciphertext) dalam format heksadesimal:
['0xd7', '0x47', '0x4e', '0xef']
['0x36', '0x4', '0x78', '0xc3']
['0x15', '0xb6', '0x92', '0xdf']
['0x34', '0x1b', '0x20', '0xf6']
Proses Dekripsi
Tahap dekripsi bertujuan untuk mengembalikan ciphertext yang telah dihasilkan menjadi bentuk plainteks asli. Dalam kasus ini, ciphertext yang diperoleh dari proses enkripsi AES-256 akan diproses kembali agar dapat dibaca sebagai teks asli, yaitu "GORONTALOPROV123".
Proses dekripsi dilakukan dengan menggunakan kunci yang sama seperti pada proses enkripsi. Ciphertext akan didekripsi menggunakan algoritma AES-256 dalam mode ECB, kemudian hasilnya masih mengandung data padding yang ditambahkan saat proses enkripsi. Oleh karena itu, tahap selanjutnya adalah menghapus padding tersebut agar plainteks dapat dikembalikan ke bentuk aslinya secara tepat.

Copyright © Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)

Dengan melalui tahap ini, data terenkripsi yang semula tidak dapat dibaca akan berubah kembali menjadi informasi yang dapat dipahami

Berikut implementasi kode python dari chipertext ke plainteks

```
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad, unpad
# Plainteks asli
plaintext = "GORONTALOPROV"
# Kunci AES 256-bit (harus 32 byte)
key = b'1234567890ABCDEF1234567890ABCDEF'  # kunci tetap
# Panjang blok AES = 16 byte
block_size = 16
# Padding plainteks agar panjangnya kelipatan block_size
padded_plaintext = pad(plaintext.encode(), block_size)
# Membuat objek cipher AES dengan mode ECB
cipher = AES.new(key, AES.MODE_ECB)
# Enkripsi plainteks yang sudah dipadding
ciphertext = cipher.encrypt(padded_plaintext)
# Tampilkan ciphertext dalam format hex (4 byte per baris)
ciphertext_hex = [hex(b) for b in ciphertext]
print("Ciphertext hex:")
for i in range(0, len(ciphertext_hex), 4):
print(ciphertext_hex[i:i+4])
# Dekripsi ciphertext
decrypted = cipher.decrypt(ciphertext)
# Hapus padding hasil dekripsi dan ubah ke string asli
unpadded = unpad(decrypted, block_size).decode()
print("\nHasil dekripsi:", unpadded)
```

Hasil output diatas menampilkan:

```
Ciphertext hex:
['0xd7', '0x47', '0x4e', '0xef']
['0x36', '0x4', '0x78', '0xc3']
['0x15', '0xb6', '0x92', '0xdf']
['0x34', '0x1b', '0x20', '0xf6']
Hasil dekripsi: GORONTALOPROV
```

Setelah menjalankan proses dekripsi, kita dapat melihat bahwa ciphertext berhasil dikembalikan menjadi plainteks asli, yaitu "GORONTALOPROV". Proses ini menegaskan bahwa data yang telah dienkripsi masih dapat dikembalikan ke bentuk plainteks aslinya. Hal ini memastikan bahwa informasi yang awalnya terenkripsi tetap dapat diakses dan dibaca oleh pihak yang berwenang.

5. Kesimpulan

1. Berdasarkan hasil penelitian dengan menggunakan algoritma AES-256, dapat disimpulkan bahwa penerapan algoritma Advanced Encryption Standard (AES) dalam meningkatkan kerahasiaan dan integritas informasi dalam komunikasi email di Dinas Komunikasi dan Informatika Provinsi Gorontalo, terdapat kelemahan yang signifikan. Meskipun algoritma Advanced Encryption Standard (AES) secara teori merupakan metode kriptografi yang kuat dan telah banyak digunakan secara luas, Hasil penelitian ini menunjukkan bahwa dalam implementasi yang diuji, ciphertext masih dapat dikembalikan ke bentuk plaintext. Hal ini mengindikasikan adanya kelemahan. Ini menandakan bahwa AES dalam implementasinya, tidak sepenuhnya aman jika tidak diiringi dengan algoritma tambahan atau penggabungan metode lain.
2. Implementasi Tanda Tangan Elektronik di Dinas Komunikasi dan Informatika Provinsi Gorontalo masih menghadapi tantangan, Yaitu pada aspek pemahaman kepada pengguna. Banyak pegawai belum sepenuhnya memahami fungsi passphrase sebagai pelindung kunci privat, yang merupakan elemen penting dalam menjaga keamanan identitas digital. Selain itu, belum terlihat sistem pengawasan yang secara aktif mencatat dan mengevaluasi penggunaan sertifikat elektronik, sehingga potensi penyalahgunaan atau kelalaian tidak dapat terdeteksi. Hal ini berdampak pada efektivitas TTE dalam menjamin integritas serta keaslian dokumen elektronik yang dikirimkan melalui email kedinasan.

Daftar Pustaka

[1] Y. J. El Anwar, R. Habibi, and N. Riza, “Penerapan Metode Kriptografi Aes Untuk Mengamankan File Dokumen,” *J. Tekno Insentif*, vol. 16, no. 2, pp. 92–104, 2022, doi: 10.36787/jti.v16i2.852.

Copyright © Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)

[2] Dio Rachma Putra and Mochamad Sidqon, "Rancang Bangun Sistem Informasi E-Arsip Berbasis Website Untuk Pengamanan Dokumen Menggunakan Metode Advance Encryption Standard (Aes) Dkpp Kota Surabaya," *J. Rekayasa Sist. Inf. dan Teknol.*, vol. 2, no. 1, pp. 445–454, 2024, doi: 10.59407/jrsit.v2i1.972.

[3] A. Asril, "Analisis Dan Perancangan Sistem Informasi Pengolahan Data Berobat Karyawan," *Simtika*, vol. 3, no. 2, pp. 28–33, 2020, [Online]. Available: http://lppm.undhari.ac.id/ejournal/index.php/simtika/article/view/74%0Ahttp://lppm.undhari.ac.id/ejournal/index.php/simtika/article/download/74/53

[4] K. Nistrina *et al.*, "Khilda Nistrina 1) , Anisa Rahmania 2) Program Studi Sistem Informasi Fakultas Teknologi Informasi Universitas Bale Bandung," vol. 03, 2021.

[5] D. Heryana, L. Setiawati, and B. Suhendar, "Sistem Informasi Dan Potensi Manfaat Big Data Untuk Pendidikan," *Gunahumas*, vol. 2, no. 2, pp. 350–357, 2020, doi: 10.17509/ghm.v2i2.23023.

[6] Angelo Delson and Rasid Ridho, "Rancang Bangun Penjualan Lisence Key Berbasis Web Pada Pt. Gfsoft Indonesia," *J. Comasie*, vol. 06, no. 02, pp. 32–39, 2022.

[7] Fefiana Diny Hermawati *et al.*, "Keamanan E-Voting Di Indonesia Melalui Pemanfaatan Kriptografi Pada Sistem AES (Advance Encryption Standard)," *J. Tek. Mesin, Ind. Elektro dan Inform.*, vol. 2, no. 2, pp. 45–56, 2023, doi: 10.55606/jtmei.v2i2.1625.

[8] S. D. Nurcahya, "Implementasi Aplikasi Kriptografi Metode Kode Geser Berbasis Java," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 5, no. 4, pp. 694–697, 2022, doi: 10.32672/jnkti.v5i4.4690.

[9] A. T. Putra, P. L. L.B., and F. Fattah, "Penerapan Enkripsi dan Deskripsi file menggunakan algoritma Twofish," *Bul. Sist. Inf. dan Teknol. Islam*, vol. 2, no. 2, pp. 72–77, 2021, doi: 10.33096/busiti.v2i2.784.

[10] J. H. Sinaga, M. Pangaribuan, F. Fazly, I. Rivaldo, and I. Gunawan, "Penerapan Enkripsi Dan Deskripsi Menggunakan Algoritma Data Encryption Standart Dengan Pemograman Matlab," *J. Media Inform.*, vol. 4, no. 1, pp. 63–69, 2022, doi: 10.55338/jumin.v4i1.468.

[11] A. A. I. Ramadhan, E. Z. Rivanti, and R. S. Zulva, "Implementasi Kriptografi AES Menggunakan Bahasa Java Programming: Meningkatkan Keamanan Data Melalui Enkripsi & Dekripsi Yang Kuat," *J. Pendidik. Teknol. Inf.*, pp. 20–26, 2023, [Online]. Available: https://jurnal.umj.ac.id/index.php/TripleA/article/view/17513%0Ahttps://jurnal.umj.ac.id/index.php/TripleA/article/download/17513/9646

[12] P. Adam and M. Ali Romli, "Implementasi Sistem Keamanan Dokumen Kepegawaian Menggunakan Metode Aes-256 Dan Vigenere Chiper," *J. Komput. dan Teknol.*, vol. 2, no. 2, pp. 20–26, 2024, doi: 10.58290/jukomtek.v2i2.166.

[13] N. Cristy and F. Riandari, "Implementasi Metode Advanced Encryption Standard (AES 128 Bit) UnCristy, N., & Riandari, F. (2021). Implementasi Metode Advanced Encryption Standard (AES 128 Bit) Untuk Mengamankan Data Keuangan. Jurnal Ilmu Komputer Dan Sistem Informasi (JIKOMSI), 4(2), ," *J. Ilmu Komput. dan Sist. Inf.*, vol. 4, no. 2, pp. 75–85, 2021, [Online]. Available: https://ejournal.sisfokomtek.org/index.php/jikom/article/view/181

Copyright © Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)