

Optimizing Digital and Multimedia File Security Using Hybrid RSA and Extended Vigenère Cipher Schemes

Emanuel Pinesthi Widi Utomo ¹, Eko Hari Rachmawanto ^{2*}, Christy Atika Sari ³

Informatics Engineering, Universitas Dian Nuswantoro, Indonesia

111202314898@mhs.dinus.ac.id ¹, eko.hari@dsn.dinus.ac.id ², christy.atika.sari@dsn.dinus.ac.id ³

Article Info

Article history:

Received 2026-07-07

Revised 2026-08-10

Accepted 2026-10-03

Keyword:

*Extended Vigenere Cipher,
Hybrid Cryptography,
Multimedia,
RSA-OAEP,
Shannon Entropy.*

ABSTRACT

The increasing use of digital and multimedia files requires cryptographic schemes that can process various file formats efficiently. This study aims to implement and analyze the performance of a hybrid cryptographic scheme combining RSA-OAEP and Extended Vigenère Cipher for multimedia file encryption. The test data consist of JPG, MP4, TXT, PDF, and ZIP files with different sizes. In the developed system, each file is read directly as a byte array. The file content is encrypted using Extended Vigenère Cipher with modulo 256 operation, while the Extended Vigenère key is encrypted using RSA-OAEP with SHA-256. The encryption output is stored in .enc format, containing metadata, RSA public key, encrypted key, and ciphertext, while the RSA private key is stored separately in .pem format. The evaluation was conducted based on encryption and decryption success, processing time, throughput, encrypted file size, and Shannon Entropy. The results show that all test files were successfully encrypted and decrypted without changing the size of the decrypted files. The largest file, an 82 MB ZIP file, was encrypted in 0.314 seconds and decrypted in 0.251 seconds. The encryption throughput ranged from 28.16 MB/s to 337.62 MB/s, while the decryption throughput ranged from 28.11 MB/s to 337.73 MB/s. The Shannon Entropy values of the encrypted files ranged from 7.8 to 8, indicating a more uniform byte distribution in the ciphertext. However, Shannon Entropy is used only as a supporting metric to observe ciphertext characteristics, not as a sole indicator of cryptographic security. Therefore, this study positions the proposed scheme as a performance analysis of hybrid RSA-OAEP and byte-based Extended Vigenère Cipher for multimedia file encryption.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. PENDAHULUAN

Meningkatnya penggunaan data digital di era modern di zaman ini adalah dampak dari perkembangan teknologi informasi yang masif di berbagai aktivitas, baik pada lingkup individu, organisasi, bisnis, maupun institusi pemerintahan. Data digital disini tidak hanya berbentuk teks, tetapi juga dalam bentuk file multimedia seperti gambar, audio, video, dokumen digital, dan file biner lainnya. Kemudahan dalam menyimpan, mengirim, dan mengakses, data digital memberikan banyak manfaat, namun juga menimbulkan risiko keamanan seperti pencurian, manipulasi, penyalahgunaan, dan akses oleh pihak yang tidak berwenang [1], [2]. Maka keamanan data menjadi aspek yang penting

untuk menjaga kerahasiaan, keutuhan, autentikasi, dan non-repudiation dari informasi yang di proses [3], [4].

Kriptografi merupakan metode yang banyak digunakan untuk menjaga keamanan data digital dengan mengubah plaintext menjadi ciphertext melalui proses enkripsi, kemudian mengembalikannya ke bentuk semula melalui proses dekripsi [1], [4]. Berdasarkan penggunaan kuncinya, algoritma kriptografi terbagi menjadi algoritma simetris dan asimetris. Algoritma simetris memiliki keunggulan dari sisi kecepatan karena menggunakan satu kunci yang sama tetapi memiliki kelemahan pada distribusi kunci. Sebaliknya, algoritma asimetris menggunakan public key dan private key sehingga distribusi kunci lebih aman, namun proses

komputasinya lebih kompleks dan kurang efisien untuk data berukuran besar [5], [6], [7].

Pendekatan kriptografi hybrid digunakan untuk menggabungkan keunggulan algoritma simetris dan asimetris. Dalam penelitian ini, skema hybrid yang digunakan adalah menggunakan algoritma simetris untuk mengenkripsi data utama karena lebih efisien dalam memproses data berukuran besar. Sedangkan, algoritma asimetris digunakan untuk mengamankan kunci simetris agar proses pengelolaan kunci menjadi lebih aman. [4], [5], [6]. Pendekatan ini relevan untuk pengamanan file multimedia yang umumnya memiliki ukuran besar dan struktur data yang lebih kompleks dibandingkan teks biasa. Dengan demikian, kriptografi hybrid dapat menjadi pendekatan implementatif untuk menganalisis bagaimana kombinasi algoritma asimetris dan algoritma simetris bekerja dalam proses enkripsi dan dekripsi file digital.

Beberapa penelitian sebelumnya telah membahas penerapan kriptografi dalam pengamanan data digital. Penelitian [5] menerapkan Vigenère Cipher untuk pengamanan citra digital, namun masih berfokus pada data citra dan belum menggunakan skema hybrid untuk pengamanan kunci. Penelitian [6] mengembangkan aplikasi pengamanan dokumen digital dengan menggabungkan Vernam Cipher, RSA, dan kompresi Huffman. Penelitian tersebut menjadi acuan penting karena telah menerapkan pendekatan hybrid, tetapi masih terbatas pada dokumen digital dan belum secara khusus membahas penerapan Extended Vigenère Cipher pada berbagai file multimedia berbasis byte. Penelitian [7] dan [8] juga menunjukkan bahwa hybrid cryptography dapat diterapkan pada sistem penyimpanan data, namun belum secara khusus membahas analisis performa skema RSA-OAEP dan Extended Vigenère Cipher pada berbagai format file multimedia.

Vigenère Cipher klasik pada dasarnya bekerja pada karakter alfabet dengan operasi modulo 26 sehingga lebih sesuai untuk data teks [5], [9]. Untuk memperluas penggunaannya pada data biner, Extended Vigenère Cipher menggunakan operasi modulo 256 sehingga setiap byte pada file dapat diproses secara langsung [10], [11], [12]. Perluasan ini memungkinkan algoritma Vigenère Cipher klasik dimodifikasi agar dapat digunakan pada berbagai jenis file digital, termasuk gambar, video, dokumen, arsip, dan file multimedia lainnya. Dengan demikian, Extended Vigenère Cipher dalam penelitian ini diposisikan sebagai algoritma simetris klasik yang diperluas cakupan operasinya dari karakter alfabet menjadi byte 0-255.

Hybrid encryption dengan menggunakan algoritma asimetris untuk mengamankan kunci simetris memang telah banyak digunakan pada penelitian dan implementasi kriptografi [4], [6], [7], [13], [14]. Sehingga, kontribusi penelitian ini tidak terletak pada konsep dasar hybrid encryption, melainkan pada implementasi dan analisis performa skema hybrid RSA-OAEP dan Extended Vigenère Cipher berbasis byte pada file multimedia [6], [13], [15], [16]. Extended Vigenère Cipher dipilih karena mampu memproses data dalam rentang byte 0-255 melalui operasi modulo 256,

sehingga dapat diterapkan pada berbagai jenis file digital, tidak terbatas pada teks alfabet seperti Vigenère klasik [10], [11], [17]. Selanjutnya, kunci dari proses enkripsi Extended Vigenère Cipher akan di enkripsi menggunakan algoritma RSA-OAEP. Dengan pembagian peran tersebut, algoritma RSA tidak digunakan untuk mengenkripsi seluruh isi file, tetapi hanya digunakan untuk mengamankan kunci dari algoritma Extended Vigenère Cipher yang telah dipakai pada proses enkripsi data utama.

Pemilihan Extended Vigenère Cipher dalam penelitian ini didasarkan pada tujuan untuk menganalisis performa simetris klasik yang diperluas agar dapat memproses file digital berbasis byte. Dalam penelitian ini, peneliti tidak bertujuan untuk membuktikan bahwa algoritma Extended Vigenère Cipher lebih unggul dibandingkan dengan algoritma simetris modern seperti AES, AES-GCM, atau ChaCha20. Beberapa contoh penelitian terdahulu telah membahas performa algoritma simetris modern pada file multimedia maupun dokumen digital [15], [16], [18], [19]. Tetapi, pada penelitian ini memiliki ruang lingkup yang berbeda, yakni menganalisis performa skema hybrid antara RSA-OAEP dan Extended Vigenère Cipher sebagai representasi kombinasi algoritma asimetris modern dengan algoritma simetris klasik berbasis byte.

Berdasarkan uraian tersebut, penelitian ini berfokus pada analisis performa skema Hybrid RSA-OAEP dan Extended Vigenère Cipher pada enkripsi file multimedia. Data uji yang digunakan mencakup beberapa format file, yakni JPG, MP4, TXT, PDF, dan ZIP dengan variasi ukuran file. Evaluasi dilakukan berdasarkan keberhasilan proses enkripsi dan dekripsi, waktu enkripsi, waktu dekripsi, throughput, ukuran file hasil enkripsi, serta Shannon Entropy sebagai metrik pendukung untuk melihat distribusi byte pada ciphertext [13], [15], [16], [20]. Shannon Entropy dalam penelitian ini digunakan sebagai indikator karakteristik keacakan ciphertext [12], [20], bukan sebagai satu-satunya dasar untuk menyatakan tingkat keamanan kriptografi. Dengan demikian, penelitian ini diharapkan dapat memberikan gambaran performa penerapan skema hybrid RSA-OAEP dan Extended Vigenère Cipher pada proses enkripsi file multimedia berbasis byte.

II. METODE

A. Analisis Kebutuhan

Tahapan analisis kebutuhan dilakukan untuk menentukan kebutuhan sistem, baik dari sisi *input*, proses, maupun *output*. Sistem pada penelitian ini dikembangkan untuk melakukan proses enkripsi dan dekripsi file multimedia menggunakan skema *hybrid RSA-OAEP* dan *Extended Vigenère Cipher*. File yang digunakan sebagai data uji pada penelitian ini adalah file berformat *JPG*, *MP4*, *TXT*, *PDF*, dan *ZIP*. Sistem dikembangkan supaya dapat membaca file sebagai *byte array*, sehingga proses enkripsi tidak bergantung pada jenis file atau format file tertentu.

- Sistem dapat menerima input file *JPG*, *MP4*, *TXT*, *PDF*, dan *ZIP*.

- Sistem dapat membaca file sebagai byte array
- Sistem dapat membangkitkan RSA Key Pair secara otomatis.
- Sistem dapat membangkitkan kunci Extended Vigenère secara otomatis.
- Sistem dapat mengenkripsi isi file menggunakan Extended Vigenère Cipher berbasis modulo 256.
- Sistem dapat mengenkripsi kunci Extended Vigenère Cipher menggunakan RSA Public Key dengan skema RSA-OAEP SHA-256.
- Sistem dapat menyimpan hasil enkripsi dalam format .enc.
- Sistem dapat melakukan proses dekripsi untuk mengembalikan file asli.

B. Tahapan Penelitian

Metode penelitian ini disusun berdasarkan tahapan pengembangan dan pengujian sistem kriptografi hybrid. Alur penelitian dimulai dari studi literatur, analisis kebutuhan, desain dan perancangan sistem, implementasi algoritma,

pengujian dan analisis hasil, hingga penarikan kesimpulan. Tahapan ini digunakan sebagai panduan dalam membangun dan mengevaluasi performa skema hybrid RSA-OAEP dan Extended Vigenère Cipher pada proses enkripsi file multimedia. Gambar 1 menunjukkan alur penelitian yang digunakan dalam penelitian ini. Setiap tahapan dilakukan secara berurutan agar proses perancangan, implementasi, dan pengujian sistem dapat berjalan sistematis. Studi literatur dilakukan untuk memahami konsep kriptografi *hybrid*, *RSA-OAEP*, *Extended Vigenère Cipher*, dan evaluasi performa enkripsi file. Analisis kebutuhan dilakukan untuk menentukan input, proses, dan output sistem. Tahap desain dan perancangan digunakan untuk menyusun alur enkripsi, alur dekripsi, serta struktur file hasil enkripsi. Selanjutnya, implementasi dilakukan dalam bentuk sistem berbasis desktop. Setelah sistem selesai dikembangkan, dilakukan pengujian terhadap file uji untuk memperoleh data waktu enkripsi, waktu dekripsi, ukuran file, *throughput*, dan *Shannon Entropy*.

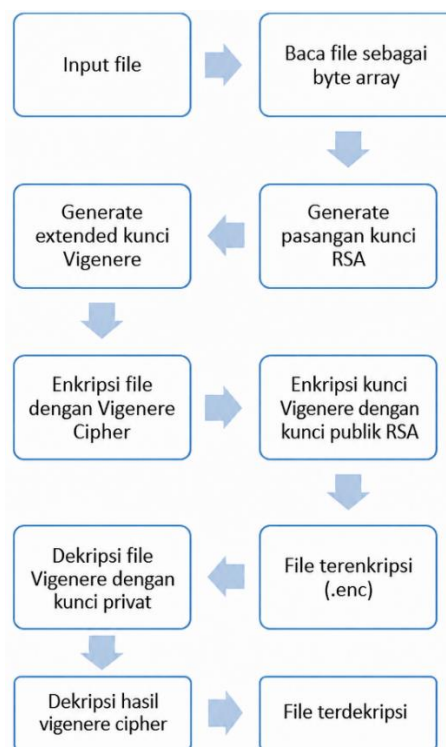


Gambar 1. Tahapan penelitian

C. Desain dan Perancangan

Tahapan desain dan perancangan ini dilakukan untuk menyusun sistem yang berbasis desktop. Sistem dirancang untuk membaca file sebagai *byte array*, mengenkripsi isi file menggunakan *Extended Vigenère Cipher*, mengamankan kunci *Extended Vigenère* menggunakan *RSA-OAEP*, dan menyimpan hasil enkripsi dalam file berformat .enc.

Berdasarkan gambar 2, sistem kriptografi dalam penelitian ini bekerja dengan membagi peran antara algoritma simetris dan algoritma asimetris. *Extended Vigenère Cipher* digunakan untuk mengenkripsi isi file karena algoritma ini bekerja langsung pada data berbasis *byte*. Sedangkan, *RSA-OAEP* digunakan untuk mengenkripsi kunci *Extended Vigenère Cipher*. Maka, *RSA* tidak digunakan untuk mengenkripsi seluruh isi file, tetapi hanya digunakan untuk mengamankan kunci simetris. Pada proses dekripsi, sistem membaca file .enc, mengambil *encrypted key* dan *ciphertext*, mendekripsi kunci *Extended Vigenère* menggunakan *RSA private key*, kemudian menggunakan kunci tersebut untuk mengembalikan *ciphertext* menjadi file asli.



Gambar 2. Alur sistem enkripsi dekripsi

D. Data Uji Penelitian

Pada penelitian ini, peneliti menggunakan data uji berupa file multimedia dengan format *JPG*, *MP4*, *TXT*, *PDF*, dan *ZIP*. Pemilihat format tersebut dilakukan untuk

merepresentasikan beberapa jenis file yang umum digunakan dalam aktivitas penyimpanan data pertukaran data digital. File *JPG* digunakan untuk mewakili data gambar, *MP4* digunakan untuk mewakili data video, *TXT* digunakan untuk mewakili data text digital, *PDF* digunakan untuk mewakili dokumen digital, dan *ZIP* digunakan untuk mewakili arsip terkompresi.

TABEL I
DAFTAR DATA PENGUJIAN

Nama File	Ekstensi File	Kategori File	Ukuran File
JPG-01	JPG	Picture	0.169 MB
JPG-02	JPG	Picture	1.14 MB
JPG-03	JPG	Picture	9.23 MB
MP4-01	MP4	Video	10 MB
MP4-02	MP4	Video	33.3 MB
MP4-03	MP4	Video	58.2 MB
TXT-01	TXT	Text	0.3 MB
TXT-02	TXT	Text	2 MB
TXT-03	TXT	Text	6 MB
PDF-01	PDF	Document	0.167 MB
PDF-02	PDF	Document	0.945 MB
PDF-03	PDF	Document	17.9 MB
ZIP-01	ZIP	Compressed archive	10.3 MB
ZIP-02	ZIP	Compressed archive	35.1 MB
ZIP-03	ZIP	Compressed archive	82 MB

Pada penelitian ini, setiap format file diuji menggunakan tiga variasi ukuran, yakni ukuran kecil, sedang, dan besar. Variasi ukuran file digunakan untuk mengetahui pengaruh ukuran file terhadap waktu enkripsi, waktu dekripsi, ukuran file hasil enkripsi, *throughput*, dan keberhasilan proses pengembalian file setelah dekripsi. Seluruh file uji dibaca sebagai *byte array*, sehingga sistem tidak melakukan konversi file ke format teks sebelum proses enkripsi. Berdasarkan Tabel I, data uji terdiri dari 15 file dengan 5 jenis ekstensi yang berbeda.

E. Parameter Kriptografi

Parameter kriptografi digunakan untuk memperjelas konfigurasi algoritma yang diterapkan dalam sistem. Parameter ini penting agar proses implementasi dan pengujian dapat direproduksi secara lebih jelas. *RSA-OAEP* digunakan untuk mengenkripsi kunci *Extended Vigenère Cipher* karena *RSA* lebih sesuai digunakan untuk data berukuran kecil, seperti kunci simetris. Sementara itu, *Extended Vigenère Cipher* digunakan untuk mengenkripsi isi file karena prosesnya dilakukan pada level *byte*. Kunci *Extended Vigenère* dibangkitkan secara acak sebagai *session key* menggunakan pembangkit bilangan acak kriptografis, sehingga setiap proses enkripsi menghasilkan kunci yang berbeda.

TABEL II
DAFTAR PARAMETER PENELITIAN

Parameter	Nilai yang Digunakan
Skema kriptografi	<i>Hybrid cryptography</i>
Algoritma asimetris	<i>RSA-OAEP</i>
Ukuran kunci <i>RSA</i>	2048-bit
Hash <i>OAEP</i>	<i>SHA-256</i>

Algoritma simetris	<i>Extended Vigenère Cipher</i>
Ruang operasi <i>Extended Vigenère</i>	Byte 0–255
Operasi modulo	Modulo 256
Panjang kunci <i>Extended Vigenère</i>	32 byte
Pembangkitan kunci <i>Extended Vigenère</i>	CSPRNG melalui modul <i>crypto</i>
Format <i>private key</i>	.pem
Format file hasil enkripsi	.enc

F. Extended Vigenère Cipher

Pada penelitian ini, *Extended Vigenère Cipher* digunakan sebagai algoritma utama untuk mengenkripsi isi file. Algoritma ini diterapkan pada data biner dengan membaca file sebagai *byte array*, sehingga setiap *byte* file diproses dalam rentang nilai 0 sampai 255. Berbeda dengan *Vigenère Cipher* klasik yang menggunakan modulo 26 pada karakter alfabet, *Extended Vigenère Cipher* pada sistem ini menggunakan operasi modulo 256 agar dapat diterapkan pada file *JPG*, *MP4*, *PDF*, dan *ZIP*.

Kunci yang digunakan dalam proses *Extended Vigenère* dibangkitkan secara acak sebagai *session key*. Proses enkripsi dilakukan dengan menjumlahkan *byte plaintext* dengan *byte kunci* [17]. Secara matematis, proses enkripsi *Extended Vigenère* mengikuti Persamaan (1), dimana *Ci* adalah nilai *Ciphertext*, *Pi* adalah nilai Kunci, *mod 256* adalah modulus 256 karakter *ASCII*. Sedangkan proses dekripsi dilakukan dengan mengurangi *byte ciphertext* dengan *byte kunci* [17]. Proses dekripsi *Extended Vigenère Cipher* mengikuti Persamaan (2), dimana *Ci* adalah nilai *Ciphertext*, *Pi* adalah nilai Kunci, *mod 256* adalah modulus 256 karakter *ASCII*. Penambahan nilai 256 digunakan untuk menghindari hasil negatif ketika nilai *byte ciphertext* lebih kecil daripada nilai *byte kunci*.

$$Ci = (Pi + Ki) \bmod 256 \quad (1)$$

$$Pi = (Ci - Ki + 256) \bmod 256 \quad (2)$$

Setelah kunci *Extended Vigenère* digunakan untuk mengenkripsi isi file, kunci tersebut perlu diamankan agar tidak dapat digunakan oleh pihak yang tidak berwenang. Oleh karena itu, penelitian ini menggunakan algoritma *RSA* untuk mengenkripsi kunci *Extended Vigenère* melalui *public key* dan mendekripsinya kembali menggunakan *private key*.

G. RSA Private dan Public Key

Pada penelitian ini, Algoritma *RSA* digunakan untuk mengamankan kunci *Extended Vigenère Cipher*. Sistem membangkitkan pasangan kunci *RSA* secara otomatis ketika proses enkripsi sedang dijalankan menggunakan modul bawaan dari *Node.js*, yaitu *crypto*. Modul ini digunakan untuk menghasilkan pasangan kunci *RSA* yang terdiri dari *Public Key* yang digunakan untuk mengenkripsi kunci *Extended Vigenère Cipher* menggunakan skema *RSA-OAEP* dengan *SHA-256*, sedangkan *Private Key* yang digunakan untuk mendekripsi kembali kunci *Extended Vigenère Cipher*. Kedua kunci yang dihasilkan dalam proses pembangkitan kunci *RSA*

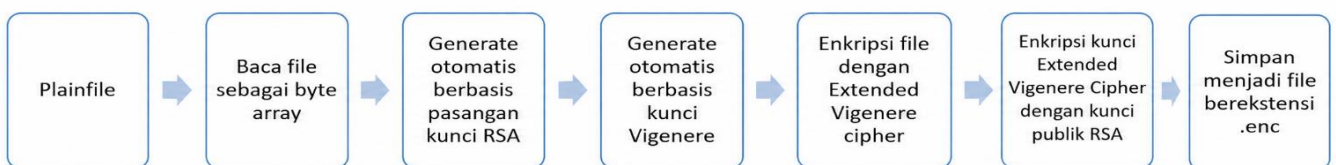
ini disimpan dalam format *.pem*. Karena sifatnya yang tidak rahasia, *Public Key* yang telah digunakan untuk mengenkripsi kunci *Extended Vigenère Cipher* akan disimpan bersama dengan metadata file, encrypted key, dan ciphertext ke dalam file terenkripsi berformat *.enc*. Sedangkan *Private Key* yang memiliki sifat rahasia akan disimpan dalam format yang sama yakni *.pem* namun kunci ini akan disimpan terpisah dari file terenkripsi karena *Private Key* ini akan digunakan dalam proses dekripsi kunci *Extended Vigenère Cipher*.

H. Struktur File *.enc*

Hasil akhir proses enkripsi disimpan dalam file berformat *.enc*. File ini tidak hanya berisi *ciphertext*, tetapi juga memuat beberapa komponen pendukung yang diperlukan dalam proses dekripsi. Berdasarkan Tabel III struktur *.enc* digunakan agar hasil enkripsi tersimpan dalam satu file yang terorganisasi. *Public key* disimpan di dalam file *.enc* karena tidak bersifat rahasia, sedangkan *private key* tidak disimpan di dalam file tersebut karena sifatnya yang rahasia dan akan digunakan saat proses dekripsi file. Dengan struktur ini, proses dekripsi dapat mengambil *encrypted key* dan *ciphertext* dari file *.enc*, kemudian menggunakan *private key* yang sesuai untuk mengembalikan kunci *Extended Vigenère*.

TABEL III
STRUKTUR FILE *.ENC*

Komponen	Fungsi
Metadata file	Menyimpan nama, ekstensi, ukuran file asli, dan waktu enkripsi
Public key RSA	Menyimpan <i>public key</i> yang digunakan saat enkripsi
Encrypted Extended Vigenère key	Menyimpan kunci <i>Extended Vigenère</i> yang telah dienkripsi <i>RSA-OAEP</i>
Ciphertext	Menyimpan isi file yang telah dienkripsi <i>Extended Vigenère</i>



Gambar 3. Diagram Alur Enkripsi

J. Dekripsi

Proses dekripsi merupakan tahap untuk mengembalikan file terenkripsi menjadi file asli kembali. Pada penelitian ini, proses dekripsi dilakukan terhadap file hasil enkripsi yang memiliki format *.enc*. File tersebut berisi metadata file asli, *public key RSA*, *encrypted Extended Vigenère key*, dan *ciphertext* hasil enkripsi file. Tahapan proses dekripsi dalam sistem ini dapat dilihat pada Gambar 4. Pada tahap awal, pengguna memilih file *.enc* yang didekripsi. Sistem kemudian membaca struktur file *.enc* untuk mengambil informasi yang diperlukan dalam proses dekripsi, seperti metadata file, *encrypted Extended Vigenère key*, dan *ciphertext*. Metadata digunakan untuk mengetahui informasi file asli, seperti nama file, ekstensi, dan ukuran file. Setelah file *.enc* berhasil dibaca,

I. Enkripsi

Proses enkripsi merupakan tahap utama dalam sistem kriptografi *Hybrid* yang dikembangkan pada penelitian ini. Setiap file uji dibaca secara langsung dalam bentuk *byte array*, sehingga sistem tidak perlu melakukan konversi file ke format teks sebelum proses enkripsi dilakukan. Alur proses enkripsi dapat dilihat pada Gambar 3. Pada tahap awal, pengguna memilih file yang dienkripsi. Sistem kemudian membaca file tersebut sebagai *byte array*. Selanjutnya sistem otomatis membangkitkan pasangan kunci *RSA* menggunakan modul *crypto* pada *Node.js*. Selanjutnya, sistem membangkitkan kunci *Extended Vigenère Cipher* yang digunakan untuk mengenkripsi file. Selanjutnya proses enkripsi file yang telah terbaca sebagai *byte array* menggunakan *Extended Vigenère Cipher*. Setelah file berhasil dienkripsi, kunci *Extended Vigenère Cipher* diamankan menggunakan *Public Key RSA* dengan skema *RSA-OAEP SHA 256*.

Hasil akhir dari proses enkripsi disimpan dalam file berformat *.enc*. File tersebut berisi metadata file asli, *Public Key RSA*, *encrypted Extended Vigenère Key*, dan *ciphertext* hasil enkripsi file. Sedangkan, *Private Key RSA* tidak disimpan dalam file terenkripsi *.enc*, tetapi disimpan secara terpisah untuk menjaga kerahasiaan kunci tersebut. Dengan pendekatan ini, *RSA* tidak digunakan secara langsung untuk mengenkripsi seluruh isi file, tetapi hanya digunakan untuk mengamankan kunci *Extended Vigenère*. Hal ini membuat proses enkripsi lebih efisien.

sistem mencari *private key RSA* yang sesuai. *Private key* tersebut digunakan untuk mendekripsi *encrypted Extended Vigenère key*. Proses dekripsi kunci dilakukan menggunakan skema *RSA-OAEP SHA-256*. Setelah kunci *Extended Vigenère* berhasil diperoleh kembali, kunci tersebut digunakan untuk mendekripsi *ciphertext* file. Proses dekripsi *ciphertext* dilakukan terhadap data biner atau *byte array* menggunakan mekanisme *Extended Vigenère Cipher* yang telah dijelaskan pada subbab *Extended Vigenère Cipher* sebelumnya. Hasil akhir dari proses ini adalah file asli yang direkonstruksi kembali sesuai dengan ekstensi awalnya, yaitu *JPG*, *MP4*, *TXT*, *PDF*, atau *ZIP*. Pemisahan *private key* dari file *.enc* bertujuan untuk menjaga keamanan, karena file terenkripsi tidak menyimpan kunci rahasia yang dapat digunakan langsung untuk membuka *ciphertext*.

K. Pengujian dan Evaluasi

Tahap pengujian dilakukan dengan menjalankan proses enkripsi pada masing-masing file uji. Pada proses ini, sistem akan mencatat ukuran file asli, waktu enkripsi, ukuran file hasil enkripsi, serta keberhasilan pembentukan file *.enc*. Setelah itu, file *.enc* didekripsi kembali untuk mengetahui apakah file hasil dekripsi dapat dikembalikan sesuai dengan file aslinya.

Evaluasi dilakukan dengan memperhatikan beberapa parameter, yaitu waktu enkripsi, waktu dekripsi, ukuran file hasil enkripsi, *throughput*, dan keutuhan file hasil dekripsi. Parameter tersebut digunakan untuk mengetahui pengaruh jenis dan ukuran file terhadap performa sistem. Pada penelitian ini, peneliti juga menilai keacakan data hasil enkripsi menggunakan penilaian *Shannon Entropy* [20]. Perhitungan *entropy* dilakukan terhadap data *byte* dengan rentang nilai 0 sampai 255. Setiap *byte* dihitung frekuensi kemunculannya, kemudian dihitung probabilitasnya terhadap total *byte* file. Perhitungan nilai *entropy* diperoleh menggunakan Persamaan (3), dimana $H(X)$ adalah nilai *Shannon Entropy*, $p(x_i)$ adalah probabilitas kemunculan *byte* ke- i .

$$H(X) = - \sum p(x_i) (\log_2 p(x_i)) \quad (3)$$

Dengan pengujian ini, sistem dapat dievaluasi dari sisi performa, efisiensi ukuran file, tingkat keacakan *ciphertext*, serta kemampuan sistem dalam mengembalikan file terenkripsi ke bentuk aslinya kembali. Hasil evaluasi tersebut menjadi dasar untuk menganalisis performa skema *hybrid RSA-OAEP* dan *Extended Vigenère Cipher* pada proses enkripsi dan dekripsi file multimedia.

III. HASIL DAN PEMBAHASAN

Pengujian dilakukan untuk mengetahui keberhasilan proses enkripsi dan dekripsi, performa sistem berdasarkan waktu proses dan *throughput*, perubahan ukuran file hasil enkripsi, serta karakteristik distribusi *byte ciphertext* berdasarkan nilai *Shannon Entropy*. Pembahasan pada bab ini difokuskan pada analisis performa skema *hybrid RSA-OAEP* dan *Extended Vigenère Cipher* dalam scenario pengujian yang digunakan, bukan pada pembuktian keunggulan keamanan dibandingkan algoritma kriptografi modern lainnya.

A. Hasil Pengujian Enkripsi dan Dekripsi

Pengujian enkripsi dan dekripsi dilakukan terhadap 15 file uji yang terdiri dari lima jenis ekstensi, yaitu *JPG*, *MP4*, *TXT*, *PDF*, dan *ZIP*. Setiap jenis file memiliki tiga variasi ukuran untuk melihat kemampuan sistem dalam memproses file berukuran kecil, sedang, dan besar. Seluruh file uji dibaca langsung sebagai *byte array* sehingga proses enkripsi tidak bergantung pada format file tertentu.

Berdasarkan Tabel IV, seluruh file uji berhasil melalui proses enkripsi dan dekripsi. File dengan format *JPG*, *MP4*, *TXT*, *PDF*, dan *ZIP* dapat dienkripsi menjadi file berformat *.enc*, kemudian berhasil didekripsi kembali sesuai dengan ukuran file aslinya. Hasil ini menunjukkan bahwa sistem mampu memproses berbagai jenis file, baik file multimedia, dokumen digital, file teks, maupun arsip terkompresi.

Pada file *JPG-01*, *PDF-01*, dan *PDF-02* terdapat sedikit penambahan ukuran pada file terenkripsi. Hal ini terjadi karena file *.enc* tidak hanya menyimpan *ciphertext*, tetapi juga menyimpan metadata, *public key RSA*, dan *encrypted Extended Vigenère key*. tetapi, setelah proses dekripsi dilakukan, ukuran file hasil dekripsi kembali sama dengan ukuran file asli.

TABEL IV
HASIL UJI ENKRIPSI DAN DEKRIPSI

Nama File	Ekstensi File	Ukuran Asli	Ukuran File Enkripsi	Ukuran File Dekripsi	Status Enkripsi	Status Dekripsi
JPG-01	JPG	0.169 MB	0.171 MB	0.169 MB	Berhasil	Berhasil
JPG-02	JPG	1.14 MB	1.14 MB	1.14 mb	Berhasil	Berhasil
JPG-03	JPG	9.23 MB	9.23 MB	9.23 MB	Berhasil	Berhasil
MP4-01	MP4	10 MB	10 MB	10 MB	Berhasil	Berhasil
MP4-02	MP4	33.3 MB	33.3 MB	33.3 MB	Berhasil	Berhasil
MP4-03	MP4	58.2 MB	58.2 MB	58.2 MB	Berhasil	Berhasil
TXT-01	TXT	0.3 MB	0.3 MB	0.3 MB	Berhasil	Berhasil
TXT-02	TXT	2 MB	2 MB	2 MB	Berhasil	Berhasil
TXT-03	TXT	6 MB	6 MB	6 MB	Berhasil	Berhasil
PDF-01	PDF	0.167 MB	0.169 MB	0.167 MB	Berhasil	Berhasil
PDF-02	PDF	0.945 MB	0.946 MB	0.945 MB	Berhasil	Berhasil
PDF-03	PDF	17.9 MB	17.9 MB	17.9 MB	Berhasil	Berhasil
ZIP-01	ZIP	10.3 MB	10.3 MB	10.3 MB	Berhasil	Berhasil
ZIP-02	ZIP	35.1 MB	35.1 MB	35.1 MB	Berhasil	Berhasil
ZIP-03	ZIP	82 MB	82 MB	82 MB	Berhasil	Berhasil

B. Hasil Evaluasi Performa Sistem

Evaluasi performa sistem pada penelitian ini dilakukan untuk mengetahui kecepatan proses enkripsi dan dekripsi

pada masing-masing file uji. Parameter yang digunakan pada pengujian ini adalah waktu enkripsi, waktu dekripsi,

throughput enkripsi, dan *throughput* dekripsi. *Throughput* digunakan untuk mengukur efisiensi algoritma dalam memproses file yang dihitung berdasarkan besarnya ukuran file uji terhadap waktu yang dibutuhkan [15]. Rumus perhitungan *throughput* dapat diperhatikan pada Persamaan (4).

$$\text{Throughput} = \frac{\text{Ukuran Data (MB)}}{\text{Waktu Enkripsi atau Dekripsi (s)}} \quad (4)$$

Karena ukuran file dinyatakan dalam satuan MB dan waktu proses sistemnya dinyatakan dalam satuan detik (s), maka hasil perhitungan *throughput* dinyatakan dalam satuan MB/s.

TABEL V
HASIL UJI PERFORMA SISTEM

Nama File	Ekstensi File	Ukuran File	Waktu Enkripsi (s)	Waktu Dekripsi (s)	Throughput Enkripsi (MB/S)	Throughput Dekripsi (MB/S)
JPG-01	JPG	0.169 MB	0.006 s	0.006 s	28.16 MB/s	28.11 MB/s
JPG-02	JPG	1.14 MB	0.007 s	0.007 s	162.85 MB/s	162.85 MB/s
JPG-03	JPG	9.23 MB	0.028 s	0.037 s	337.62 MB/s	255.49 MB/s
MP4-01	MP4	10 MB	0.032 s	0.049 s	321.51 MB/s	209.97 MB/s
MP4-02	MP4	33.3 MB	0.122 S	0.115 S	279.54 MB/s	296.55 MB/s
MP4-03	MP4	58.2 MB	0.232 s	0.197 s	250.86 MB/s	295.43 MB/s
TXT-01	TXT	0.3 MB	0.003 s	0.006 s	100 MB/s	50 MB/s
TXT-02	TXT	2 MB	0.007 s	0.007 s	285.71 MB/s	285.71 MB/s
TXT-03	TXT	6 MB	0.019 s	0.018 s	315.78 MB/s	333.33 MB/s
PDF-01	PDF	0.167 MB	0.002 s	0.003 s	83.5 MB/s	55.67 MB/s
PDF-02	PDF	0.945 MB	0.004 s	0.007 s	236.25 MB/s	135 MB/s
PDF-03	PDF	17.9 MB	0.065 s	0.053 s	275.38 MB/s	337.73 MB/s
ZIP-01	ZIP	10.3 MB	0.044 s	0.032 s	234 MB/s	321.87 MB/s
ZIP-02	ZIP	35.1 MB	0.133 s	0.107 s	263.9 MB/s	328 MB/s
ZIP-03	ZIP	82 MB	0.314 s	0.251 s	261.14 MB/s	326.69 MB/s

Berdasarkan Tabel V, waktu enkripsi tercepat diperoleh pada file PDF-01 dengan ukuran 0,167 MB, yakni memakan waktu selama 0,002 detik. Sedangkan, waktu enkripsi terlama diperoleh pada file ZIP-03 dengan ukuran 82 MB, yakni memakan waktu selama 0,314 detik. Pada proses dekripsi, waktu tercepat diperoleh pada file PDF-01, yakni memakan waktu selama 0,003 detik. Sedangkan, waktu dekripsi terlama diperoleh pada file ZIP-03, yakni memakan waktu selama 0,251 detik.

Hasil tersebut menunjukkan bahwa ukuran file berpengaruh terhadap waktu proses enkripsi dan dekripsi. Semakin besar ukuran filenya, semakin banyak juga byte yang harus diproses oleh *Extended Vigenère Cipher*, sehingga waktu proses cenderung meningkat. Hal ini sesuai dengan karakteristik *Extended Vigenère Cipher* yang memproses data secara *byte-per-byte* menggunakan operasi modulo 256. Tetapi, dalam scenario pengujian ini, file terbesar yang diuji, yakni ZIP-03 berukuran 82 MB, masih dapat diproses dalam waktu kurang dari satu detik.

Dari sisi *throughput*, nilai *throughput* enkripsi berada pada rentang 28,16 MB/s hingga 337,62 MB/s. *Throughput* enkripsi tertinggi diperoleh pada file JPG-03 sebesar 337,62 MB/s. Sementara itu, *throughput* dekripsi berada pada rentang 28,11 MB/s hingga 337,73 MB/s, dengan nilai tertinggi diperoleh pada file PDF-03 sebesar 337,73 MB/s. Perbedaan *throughput* pada masing-masing file dipengaruhi oleh ukuran file, waktu proses, serta karakteristik data yang diproses. Berdasarkan pernyataan sebelumnya, hasil pengujian ini menunjukkan bahwa skema *hybrid RSA-OAEP* dan *Extended Vigenère Cipher* dapat memproses file uji dengan *throughput* yang bervariasi sesuai ukuran dan jenis file.

Perlu ditegaskan kembali bahwa *RSA-OAEP* dalam sistem ini tidak digunakan untuk mengenkripsi seluruh isi file. *RSA-OAEP* hanya digunakan untuk mengenkripsi kunci *Extended Vigenère Cipher*, sedangkan isi file diproses menggunakan *Extended Vigenère Cipher*. Pembagian peran ini membuat beban proses *RSA* tidak bertambah secara langsung mengikuti ukuran file, karena proses utama terhadap isi file dilakukan oleh *Extended Vigenère Cipher*.

C. Hasil Shannon Entropy

Pengujian *Shannon Entropy* pada penelitian ini dilakukan untuk mengetahui tingkat keacakan data yang diuji sebelum dan sesudah proses enkripsi. *Entropy* digunakan untuk melihat tingkat keacakan *byte* pada file. Semakin tinggi nilai *entropy* file tersebut, maka semakin acak distribusi *byte* pada file tersebut. Pada data berbasis *byte*, nilai maksimum *Shannon Entropy* yang dihasilkan pada pengujian *entropy* ini adalah 8 bit [21]. Semakin mendekati nilai 8, maka distribusi *byte* pada file cenderung semakin merata.

Berdasarkan Tabel VI, seluruh file hasil enkripsi memiliki nilai *entropy* yang tinggi, yakni berada dalam rentang 7.8 hingga 8. Nilai tersebut menunjukkan bahwa file hasil enkripsi memiliki distribusi *byte* yang lebih acak dibandingkan file aslinya. Peningkatan *entropy* paling terlihat pada *TXT* dan *PDF*. File *TXT* memiliki *entropy* awal sebesar 5.8 hingga 5.9, kemudian meningkat hingga rentang 7.8 hingga 7.9 setelah proses enkripsi. File *PDF* memiliki *entropy* awal sebesar 6.4 dan meningkat menjadi 7.9 setelah proses enkripsi.

Hal ini menunjukkan bahwa proses enkripsi mampu menyamarkan pola *byte* pada file yang sebelumnya memiliki distribusi data lebih rendah menjadi lebih sulit dikenali

setelah proses enkripsi. Tetapi, interpretasi nilai *entropy* perlu dilakukan secara hati-hati karena *entropy* hanya menunjukkan karakteristik distribusi byte, bukan membuktikan ketahanan algoritma terhadap serangan kriptanalisis.

TABEL VI

HASIL PENGUJIAN SHANNON ENTROPY

Nama File	Ekstensi File	Entropy File Asli	Entropy File Enkripsi
JPG-01	JPG	7.9	7.9
JPG-02	JPG	7.9	7.9
JPG-03	JPG	7.9	7.9
MP4-01	MP4	7.9	8
MP4-02	MP4	7.9	8
MP4-03	MP4	7.9	8
TXT-01	TXT	5.8	7.9
TXT-02	TXT	5.9	7.9
TXT-03	TXT	5.9	7.8
PDF-01	PDF	6.4	7.9
PDF-02	PDF	6.4	7.9
PDF-03	PDF	6.4	7.9
ZIP-01	ZIP	7.9	8
ZIP-02	ZIP	7.9	8
ZIP-03	ZIP	7.9	8

Pada file JPG, MP4, dan ZIP, nilai *entropy* file asli sudah relatif tinggi sebelum proses enkripsi. File JPG memiliki rentang nilai *entropy* awal sekitar 7.95 hingga 7.97, file MP4 memiliki nilai *entropy* awal sebesar 7.99, dan file ZIP memiliki nilai *entropy* awal sebesar 7.9 juga. Hal ini dapat terjadi karena format JPG, MP4, dan ZIP umumnya sudah memiliki struktur data yang padat atau terkompresi atau *encoding* sehingga distribusi *byte* awalnya sudah mendekati acak. Setelah proses enkripsi, nilai *entropy* pada file-file tersebut tetap berapa pada nilai tinggi dan mendekati nilai maksimum 8.

Namun, nilai *Entropy* tidak dapat digunakan sebagai satu-satunya dasar untuk menyatakan tingkat keamanan kriptografi. *Entropy* yang tinggi hanya untuk menunjukkan bahwa distribusi *byte ciphertext* terlihat lebih acak, tetapi tidak membuktikan bahwa algoritma tersebut tahan terhadap serangan seperti *known-plaintext attack*. Maka, dalam penelitian ini *Shannon Entropy* digunakan sebagai metrik pendukung untuk menganalisis karakteristik *ciphertext*, bukan sebagai indikator tunggal keamanan algoritma.

D. Pembahasan

Berdasarkan hasil pengujian, skema *hybrid RSA-OAEP* dan *Extended Vigenère Cipher* berhasil mengenkripsi dan mendekripsi seluruh file uji berformat JPG, MP4, TXT, PDF, dan ZIP. Seluruh file dapat dikembalikan ke ukuran awalnya setelah proses dekripsi, sehingga pembacaan file sebagai *byte array* dapat diterapkan pada berbagai jenis file dalam skenario pengujian ini.

Dari sisi performa, ukuran file memengaruhi waktu enkripsi dan dekripsi karena *Extended Vigenère Cipher* memproses data pada level *byte*. File terbesar, yaitu ZIP-03 berukuran 82 MB, dapat dienkripsi dalam 0,314 detik dan didekripsi dalam 0,251 detik. Hal ini menunjukkan bahwa

sistem mampu memproses file uji dengan waktu yang relatif singkat pada skenario pengujian yang digunakan.

RSA-OAEP dalam skema ini tidak digunakan untuk mengenkripsi seluruh isi file, tetapi hanya untuk mengenkripsi kunci *Extended Vigenère Cipher*. Isi file tetap dienkripsi menggunakan *Extended Vigenère Cipher* berbasis *modulo 256*. Pembagian peran ini membuat proses utama dilakukan oleh algoritma simetris, sedangkan *RSA-OAEP* berfungsi sebagai pengaman kunci.

Hasil *Shannon Entropy* menunjukkan bahwa *ciphertext* memiliki nilai tinggi, yaitu 7,8 hingga 8, yang menandakan distribusi *byte* lebih merata. Namun, *entropy* tidak dapat dijadikan satu-satunya dasar untuk menyatakan keamanan kriptografi. Dengan demikian, penelitian ini lebih tepat dipahami sebagai analisis performa skema *hybrid RSA-OAEP* dan *Extended Vigenère Cipher*, bukan sebagai pembuktian bahwa *Extended Vigenère Cipher* dapat menggantikan algoritma simetris modern seperti AES atau *ChaCha20*.

IV. KESIMPULAN

Berdasarkan hasil implementasi dan pengujian, skema *hybrid RSA-OAEP* dan *Extended Vigenère Cipher* berhasil diterapkan pada proses enkripsi dan dekripsi file multimedia berformat JPG, MP4, TXT, PDF, dan ZIP. Seluruh file uji berhasil dienkripsi menjadi file berformat *.enc* dan didekripsi kembali ke bentuk semula tanpa perubahan ukuran pada file hasil dekripsi. Hal ini menunjukkan bahwa pembacaan file sebagai *byte array* dapat diterapkan pada berbagai jenis file digital dalam skenario pengujian yang digunakan.

Dari sisi performa, ukuran file berpengaruh terhadap waktu enkripsi dan dekripsi. File terbesar dalam pengujian, yaitu ZIP-03 berukuran 82 MB, dapat dienkripsi dalam waktu 0,314 detik dan didekripsi dalam waktu 0,251 detik. Hasil *throughput* juga menunjukkan bahwa sistem mampu memproses file uji dengan kecepatan yang bervariasi, yaitu 28,16 MB/s hingga 337,62 MB/s untuk enkripsi dan 28,11 MB/s hingga 337,73 MB/s untuk dekripsi. Performa tersebut dipengaruhi oleh pembagian peran dalam skema *hybrid*, yaitu *Extended Vigenère Cipher* digunakan untuk mengenkripsi isi file pada level *byte*, sedangkan *RSA-OAEP* digunakan untuk mengenkripsi kunci *Extended Vigenère Cipher*.

Hasil pengujian *Shannon Entropy* menunjukkan bahwa file hasil enkripsi memiliki nilai *entropy* pada rentang 7,8 hingga 8. Nilai tersebut menunjukkan bahwa distribusi *byte* pada *ciphertext* cenderung lebih merata, terutama pada file TXT dan PDF yang memiliki peningkatan *entropy* paling terlihat. Namun, *Shannon Entropy* dalam penelitian ini hanya digunakan sebagai metrik pendukung untuk melihat karakteristik distribusi *byte ciphertext*, bukan sebagai satu-satunya dasar untuk menyatakan tingkat keamanan kriptografi.

Dengan demikian, penelitian ini menunjukkan bahwa skema *hybrid RSA-OAEP* dan *Extended Vigenère Cipher* berhasil dianalisis performanya pada proses enkripsi file multimedia. *Extended Vigenère Cipher* dalam penelitian ini

diposisikan sebagai algoritma simetris klasik berbasis byte, bukan sebagai pengganti algoritma simetris modern seperti AES atau ChaCha20. Penelitian selanjutnya dapat memperluas jumlah dan variasi data uji, menambahkan pengujian pada file berukuran lebih besar, mengevaluasi penggunaan memori dan skalabilitas sistem, serta melakukan analisis keamanan kriptografi yang lebih mendalam

DAFTAR PUSTAKA

- [1] Dola Ramalinda, Jayadi, and Agung Rachmat Raharja, "Strategi Perlindungan Data Menggunakan Sistem Kriptografi Dalam Keamanan Informasi," *Journal of International Multidisciplinary Research*, vol. 2, no. 6, pp. 665–671, Jun. 2024, doi: 10.62504/jimr679.
- [2] N. A. Nanda, S. M. S. Silalahi, D. Patricia Nasution, M. Sari, and I. Gunawan, "Kriptografi dan Penerapannya Dalam Sistem Keamanan Data," *Jurnal Media Informatika*, vol. 4, no. 2, pp. 90–93, Jun. 2023, doi: 10.55338/jumin.v4i2.428.
- [3] A. Kurniawan and R. Novrian Kahar, "Sosialisasi Cyber Security Dan Kriptografi Dasar Dalam Menyongsong Era Society 5.0 Di Smk Ricardo Auto Machine," 2023. [Online]. Available: <http://pijarpemikiran.com/>
- [4] Permana J, Hendrana, and Munir R, "Mengamankan File Rahasia Menggunakan Hybrid Kriptografi," *JRPP (Journal Review Pendidikan dan Pengajaran)*, vol. 7, no. 1, pp. 2449–2460, Feb. 2024, doi: 10.31004/jrpp.v7i1.26156.
- [5] Imam Riadi, Abdul Fadlil, and Fahmi Auliya Tsani, "Pengamanan Citra Digital Berbasis Kriptografi Menggunakan Algoritma Vigenere Cipher," *JISKA (Jurnal Informatika Sunan Kalijaga)*, vol. 7, no. 1, pp. 33–45, Jan. 2022, doi: 10.14421/jiska.2022.7.1.33-45.
- [6] F. D. Yonathan, H. Nasution, and H. Priyanto, "Aplikasi Pengaman Dokumen Digital Menggunakan Algoritma Kriptografi Hybrid dan Algoritma Kompresi Huffman," *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, vol. 7, no. 2, p. 181, Aug. 2021, doi: 10.26418/jp.v7i2.47077.
- [7] J.-F. Lai and S.-H. Heng, "Secure File Storage On Cloud Using Hybrid Cryptography," *Journal of Informatics and Web Engineering*, vol. 1, no. 2, pp. 1–18, Sep. 2022, doi: 10.33093/jiwe.2022.1.2.1.
- [8] A. K. Bermami, T. A. K. Murshedi, and Z. A. Abod, "A hybrid cryptography technique for data storage on cloud computing," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 24, no. 6, pp. 1613–1624, 2021, doi: 10.1080/09720529.2020.1859799.
- [9] D. Arfandy, M. Simanjuntak, and T. Pasaribu, "Penerapan Metode Vigenere Cipher untuk Mengamankan Data Text," *JUKI: Jurnal Komputer dan Informatika*, vol. 4, no. 1, pp. 60–66, Jun. 2022, doi: 10.53842/juki.v4i1.103.
- [10] E. Lestariningsih, E. Ardianto, and W. Tri Handoko, "Adopsi Pembangkit Kunci Extended Vigenere Menggunakan Fungsi Random Dan Blum Blum Shub," *Jurnal Informatika dan Rekayasa Elektronik*, vol. 5, no. 2, pp. 263–271, Nov. 2022, doi: 10.36595/jire.v5i2.706.
- [11] E. Ardianto, W. T. Handoko, E. Supriyanto, and H. Murti, "Evolusi Cipher Vigenere dalam Peningkatan Pengamanan Informasi," *Jurnal Informatika Upgris*, vol. 7, no. 2, Dec. 2021, doi: 10.26877/jiu.v7i2.9333.
- [12] E. Ardianto, W. T. Handoko, E. Lestariningsih, and F. A. Sutanto, "Adopsi Pembangkit Kunci Blum Blum Shub Dan Bilangan Euler Pada Algoritma Extended Vigenere," *JOINTECS (Journal of Information Technology and Computer Science)*, vol. 8, no. 2, p. 41, Jul. 2023, doi: 10.31328/jointecs.v8i2.4326.
- [13] A. F. Pranata, A. F. Saragih, R. Fikri, K. Rahmadani, and N. P. Aulia, "Analisis Perbandingan Enkripsi dan Dekripsi Menggunakan Algoritma Simetris AES dan Hybrid (AES + RSA) Pada File Video," *JIKUM: Jurnal Ilmu Komputer*, vol. 2, no. 1, pp. 102–107, Jan. 2026, doi: 10.62671/jikum.v2i1.143.
- [14] R. Akter, Md. A. R. Khan, F. Rahman, S. J. Soheli, and N. J. Suha, "RSA and AES Based Hybrid Encryption Technique for Enhancing Data Security in Cloud Computing," *International Journal of Computational and Applied Mathematics & Computer Science*, vol. 3, pp. 60–71, Oct. 2023, doi: 10.37394/232028.2023.3.8.
- [15] S. Sugiyatno and Hafidz Maulana Rahman, "Analisis Kinerja AES dan RC6 pada File Multimedia," *SATESI: Jurnal Sains Teknologi dan Sistem Informasi*, vol. 6, no. 1, pp. 46–51, Apr. 2026, doi: 10.54259/satesi.v6i1.7523.
- [16] M. Patria and D. A. Andriati, "Analisis Komparatif Performa AES-GCM dan ChaCha20-Poly1305 dalam Enkripsi Dokumen PDF Berbasis AEAD," *Arcitech: Journal of Computer Science and Artificial Intelligence*, vol. 5, no. 1, pp. 49–69, Jun. 2025, doi: 10.29240/arcitech.v5i1.13645.
- [17] M. Boru, A. A. Sultani, A. Y. Mauko, S. A. S. Mola, K. Letelay, and D. M. Sihotang, "Kombinasi Cipher Substitusi (Beaufort dan Vigenere) Menggunakan Pembangkit Kunci RC4 Pada Kriptografi Video Audio Video Interlaced (AVI)," *TELEKONTRAN: Jurnal Ilmiah Telekomunikasi, Kendali dan Elektronika Terapan*, vol. 12, no. 2, pp. 187–195, Dec. 2024, doi: 10.34010/telekontran.v12i2.14167.
- [18] A. Susanti, B. A. Prasetya, O. D. Pangesti, L. D. Suryawati, and I. A. Saputro, "Perbandingan Kinerja dan Keamanan Algoritma Kriptografi Modern AES-GCM dengan CHACHA20-POLY1305," *Infomatek*, vol. 26, no. 2, pp. 253–264, Dec. 2024, doi: 10.23969/infomatek.v26i2.19255.
- [19] R. K. Muhammed et al., "Comparative Analysis of AES, Blowfish, Twofish, Salsa20, and ChaCha20 for Image Encryption," *Kurdistan Journal of Applied Research*, vol. 9, no. 1, pp. 52–65, May 2024, doi: 10.24017/science.2024.1.5.
- [20] T. D. Ferdillah and A. Fauzi, "Perbandingan Teknik Algoritma Rsa Dengan Algoritma Elgamal Dalam Meningkatkan Nilai Entropy Dalam Chiphertext," *JAINS (Journal of Artificial Intelligence and Network Systems)*, vol. 1, no. 1, pp. 25–30, Sep. 2025.
- [21] C. Paramita and Usman Sudibyo, "Kriptografi Audio MP3 Menggunakan RSA dan Transposisi Kolom," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 3, pp. 483–488, Jun. 2021, doi: 10.29207/resti.v5i3.2996.