# IMPLEMENTASI HYBRID ENCRYPTION TWOFISH DAN KYBER KEY ENCAPSULATION MECHANISM PADA KEAMANAN FILE DOKUMEN

1st Revaldy Krisna Putra
Program Studi Informatika
Universitas Telkom, Kampus Surabaya
Surabaya 60231, Jawa Timur, Indonesia
revaldykrisna@student.telkomuniversity.ac.id

2nd Rizky Fenaldo Maulana, S.Kom., M.Kom
Program Studi Informatika
Universitas Telkom, Kampus Surabaya
Surabaya 60231, Jawa Timur, Indonesia
rizkyfenaldo@telkomuniversity.ac.id

3rd Fandisya Rahman, S.Kom., M.Kom
Program Studi Informatika
Universitas Telkom, Kampus Surabaya
Surabaya 60231, Jawa Timur, Indonesia
fandisya@telkomuniversity.ac.id

***Abstrak* — Perkembangan teknologi kuantum membawa ancaman baru terhadap sistem kriptografi klasik, seperti RSA dan AES. Untuk mengatasi risiko ini, penelitian ini mengusulkan implementasi hybrid encryption yang menggabungkan Twofish sebagai algoritma simetris dan Kyber Key Encapsulation Mechanism (KEM) sebagai algoritma asimetris post-kuantum. Sistem ini dirancang untuk mengamankan file dokumen dengan ukuran hingga 10MB, serta diuji melalui skenario komunikasi antar virtual machine. Hasil pengujian menunjukkan waktu proses enkripsi-dekripsi yang efisien dan konsistensi hasil hash, menandakan integritas data tetap terjaga. Pengujian sniffing menggunakan Wireshark juga membuktikan bahwa data tidak dapat diakses pihak ketiga. Dengan demikian, sistem terbukti aman dan layak digunakan dalam menghadapi tantangan kriptografi masa depan**

***Kata kunci*— Twofish, Kyber KEM, Hybrid Encryption, Post-Quantum Cryptography, Keamanan Dokumen.**

## I. PENDAHULUAN

Kemajuan teknologi memungkinkan manusia untuk saling terhubung dan berbagi data tanpa batasan ruang dan waktu. Namun, peningkatan arus informasi ini juga membuka peluang terhadap serangan siber seperti pencurian data, pemalsuan, dan akses ilegal. Oleh karena itu, kebutuhan akan sistem keamanan informasi yang handal menjadi semakin mendesak.[1] Salah satu teknologi yang menjadi kunci utama dalam menjaga kerahasiaan dan integritas data adalah kriptografi. Kriptografi mengubah data yang mudah dibaca (*plaintext*) menjadi bentuk yang tidak dapat dipahami (*ciphertext*) untuk mencegah akses oleh pihak tidak berwenang.[2] Namun, munculnya komputasi kuantum menjadi ancaman bagi algoritma kriptografi klasik seperti RSA dan AES, karena algoritma seperti Shor dapat memecahkan kunci publik secara efisien. Oleh sebab itu, pengembangan sistem post-quantum cryptography (PQC) menjadi penting.[3] Twofish merupakan algoritma simetris berbasis *Feistel network* yang menawarkan efisiensi dan keamanan tinggi. Dengan panjang blok tetap 128-bit dan dukungan kunci hingga 256-bit, Twofish cocok digunakan dalam perangkat lunak maupun perangkat keras dengan sumber daya terbatas.[4][5] Sedangkan Kyber KEM adalah algoritma PQC berbasis lattice yang aman terhadap serangan kuantum serta mendukung keamanan IND-CCA dan ANO-CCA.[6] Penelitian ini bertujuan untuk mengembangkan dan mengevaluasi implementasi sistem hybrid encryption dengan menggabungkan Twofish dan Kyber KEM dalam konteks keamanan file dokumen digital.

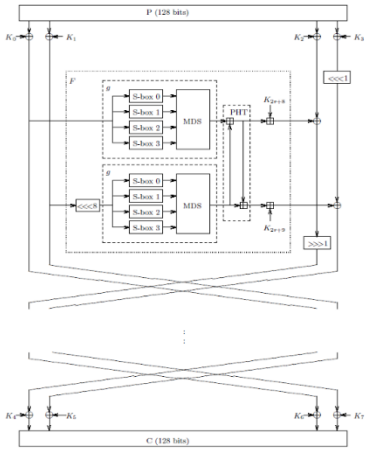
## II. KAJIAN TEORI

### A. Kriptografi

Kriptografi merupakan ilmu yang mempelajari cara mengamankan komunikasi agar isi pesan hanya dapat dipahami oleh pihak yang berhak. Awalnya bersifat sebagai seni menyembunyikan pesan, kriptografi mulai berkembang menjadi disiplin ilmu pada pertengahan abad ke-20.[7] Dalam kriptografi, enkripsi mengubah plaintext menjadi ciphertext, sedangkan dekripsi mengembalikan ciphertext menjadi plaintext. Proses ini dilakukan menggunakan fungsi sistematis untuk menjaga kerahasiaan informasi.[8]

### B. Twofish

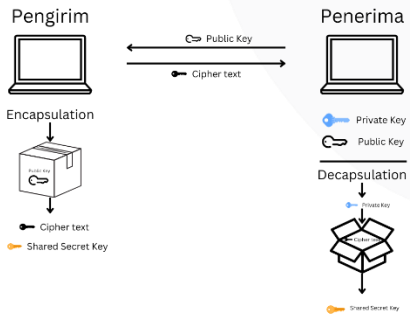
Twofish adalah algoritma blok cipher simetris yang dikembangkan oleh Bruce Schneier sebagai penyempurnaan dari Blowfish. Dirancang sesuai kriteria NIST dalam kompetisi AES, Twofish menggunakan blok berukuran 128 bit dan mendukung kunci sepanjang 128, 192, atau 256 bit.[9] Twofish menggunakan struktur mirip Feistel dengan 16 putaran dan menerapkan teknik whitening pada input dan output melalui operasi XOR dengan kunci. Salah satu ciri khasnya adalah rotasi 1 bit, yang meskipun berada di luar struktur Feistel standar, dapat dimasukkan ke dalam fungsi F dengan tambahan langkah rotasi sebelum whitening output.[10]




Gambar 1. Blok Diagram Twofish

C. Kyber KEM

Kyber adalah algoritma kriptografi pasca-kuantum berbasis module-lattice yang dirancang untuk menghadapi ancaman komputer kuantum. Keamanannya bergantung pada masalah Learning With Errors (LWE) yang sulit dipecahkan, bahkan oleh komputer kuantum. Kyber hadir dalam tiga varian: Kyber-512 (setara AES-128), Kyber-768 (AES-192), dan Kyber-1024 (AES-256), masing-masing dengan ukuran kunci dan ciphertext yang berbeda, disesuaikan dengan kebutuhan efisiensi dan tingkat keamanan.[11] Algoritma ini menggunakan parameter seperti modulus q=3329q = 3329q=3329, dimensi n=256n = 256n=256, dan distribusi binomial. Tiga proses utama dalam Kyber KEM meliputi *key generation*, *encryption* (encapsulation), dan *decryption* (decapsulation). Pada tahap *key generation*, penerima menghasilkan pasangan kunci berupa public key dan private key, di mana public key kemudian dikirim ke pengirim. Selanjutnya, pada tahap *encryption*, pengirim menggunakan public key tersebut untuk menghasilkan ciphertext dan shared secret key. Terakhir, dalam proses *decryption*, penerima menggunakan private key untuk mendekapsulasi ciphertext dan memperoleh kembali shared secret key yang sama.[12]




Gambar 2. *Encapsulation Mechanism*

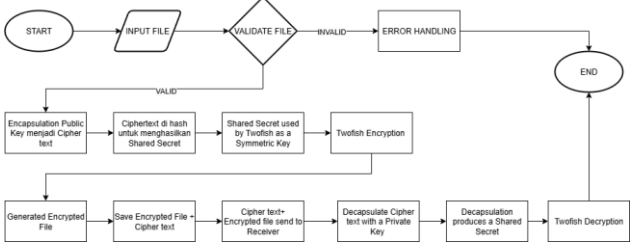
D. Hybrid Encryption

Hybrid Encryption menggabungkan algoritma simetris dan asimetris untuk mengatasi kelemahan masing-masing. Metode ini memungkinkan pertukaran kunci yang aman (via algoritma asimetris) sekaligus enkripsi data yang efisien (via algoritma simetris).[13][14]

## III. METODE

A. Alur Penelitian

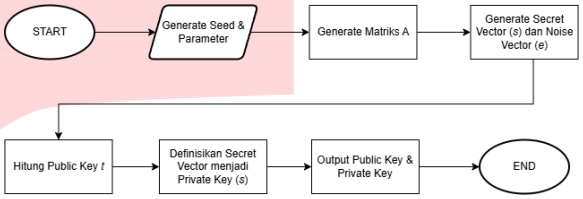
Keamanan file dokumen dilakukan menggunakan metode Hybrid encryption dengan mengintegrasikan algoritma Kyber KEM untuk membuat kunci dengan algoritma Twofish untuk enkripsi dan dekripsi file. Alur penelitian dapat dilihat pada gambar berikut.




Gambar 3. Alur Penelitian

B. Key Generation Kyber KEM

Proses pembuatan kunci simetris yang akan digunakan oleh twofish untuk enkripsi *file* menggunakan algoritma Kyber KEM.




Gambar 4. Key Generation Kyber KEM

Proses *Key Generation* pada Kyber KEM dimulai dengan input berupa seed dan parameter kriptografi untuk membentuk matriks publik A secara deterministik melalui algoritma seperti hash-based expansion. Vektor rahasia s dan vektor noise e dihasilkan menggunakan distribusi binomial terpusat guna menjamin keamanan.

$$t = A \cdot s + e \bmod q \quad (1)$$

Public key t dihitung dari operasi antara A, s, dan e dalam modulus q. Private key didefinisikan sebagai vektor s, sedangkan public key terdiri dari t dan seed untuk merekonstruksi matriks A. Public key digunakan pengirim dalam proses *encapsulation*, sementara private key disimpan oleh penerima untuk *decapsulation*. Dalam proses ini, *shared secret* dihasilkan oleh pengirim menggunakan public key dan direkonstruksi kembali oleh penerima menggunakan private key. Shared secret ini kemudian digunakan sebagai kunci simetris dalam algoritma Twofish.

C. *Encapsulation Mechanism*

Encapsulation mechanism adalah proses untuk menghasilkan *shared secret* secara aman tanpa mengirimkan kunci simetris secara langsung melalui jaringan. Dalam Kyber KEM, mekanisme ini memanfaatkan kriptografi berbasis *module-lattice* yang tahan terhadap serangan komputer kuantum. Proses dilakukan menggunakan *public key* penerima untuk menghasilkan dua keluaran utama, yaitu *ciphertext* dan *shared secret*. Langkah pertama adalah membangkitkan vektor polinomial acak *r* dari distribusi binomial terpusat, yang digunakan sebagai dasar enkripsi. Dengan menggunakan *public key* penerima, pengirim menghitung *ciphertext c*.

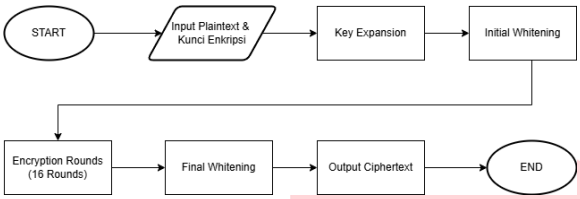
$$c = A \cdot r + e \quad (2)$$

Selanjutnya, *shared secret s* dihasilkan melalui proses hashing terhadap kombinasi *r* dan *c*, memastikan bahwa *shared secret* tergantung pada nilai acak dan *ciphertext* yang terbentuk.

$$s\,s = Hash(r||c) \quad (3)$$

D. Twofish *Encryption*

Setelah melewati proses *encapsulation*, maka *shared secret* akan digunakan untuk kunci enkripsi twofish.




Gambar 5. Enkripsi Twofish

Proses enkripsi dengan algoritma Twofish dimulai dengan memasukkan plaintext dan kunci utama berdimensi 128, 192, atau 256 bit. Kunci ini berasal dari *shared secret* hasil Kyber KEM dan diperluas menjadi subkunci melalui proses *key schedule*. Enkripsi diawali dengan *initial whitening*, yaitu pembagian plaintext menjadi empat blok 32-bit yang masing-masing di-XOR dengan subkunci awal. Selanjutnya, data melalui 16 ronde enkripsi yang melibatkan substitusi menggunakan S-box kunci-dependen, permutasi oleh matriks MDS, pencampuran kunci dengan XOR, dan pertukaran posisi blok. Setelah ronde selesai, dilakukan *final whitening* dengan XOR terhadap subkunci terakhir untuk menghasilkan ciphertext.

E. *Decapsulation*

Decapsulation adalah proses yang dilakukan oleh penerima untuk merekonstruksi *shared secret* yang identik dengan hasil dari proses *encapsulation* oleh pengirim. Proses ini menggunakan *ciphertext* yang diterima dan *private key* milik penerima. Dalam Kyber KEM, decapsulation melibatkan dekripsi *ciphertext* dan rekonstruksi *shared secret* menggunakan fungsi hashing yang sama seperti pada *encapsulation*. Penerima memanfaatkan *private key s* dan *ciphertext c* untuk menghitung kembali vektor acak *r* yang digunakan selama *encapsulation*.

$$\tilde{r} = c - A \cdot s \quad (4)$$

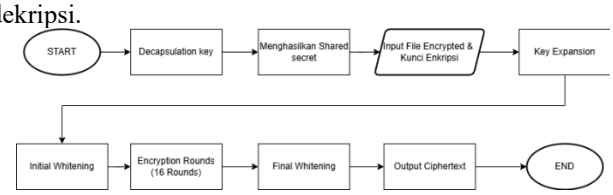
Setelah vektor $\tilde{r}$ berhasil direkonstruksi, *shared secret s* dihitung melalui proses hashing yang identik dengan pihak pengirim.

$$s\,s = Hash(\tilde{r}||c) \quad (5)$$

Jika tidak ada gangguan dalam komunikasi atau proses, *shared secret* yang dihasilkan akan sama persis antara pengirim dan penerima.

F. Dekripsi Twofish

Proses dekripsi pada algoritma Twofish menggunakan parameter dan kunci yang sama seperti saat enkripsi. Tahap pertama adalah mendekapsulasi *shared secret key* dari ciphertext menggunakan Kyber KEM. Setelah itu, subkunci yang diperoleh dari *key schedule* digunakan kembali, namun dalam urutan terbalik. Dekripsi diawali dengan membagi ciphertext menjadi empat blok 32-bit, kemudian dilakukan *final whitening* dengan operasi XOR terhadap subkunci akhir untuk memperoleh data awal sebelum memasuki ronde dekripsi.

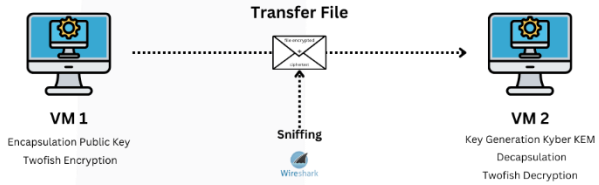



Gambar 6. Dekripsi Twofish

Selanjutnya, ciphertext diproses melalui 16 ronde dekripsi yang merupakan kebalikan dari proses enkripsi. Pada setiap ronde, efek substitusi, permutasi, dan pencampuran kunci dibalik menggunakan operasi invers, termasuk membalik fungsi F dengan substitusi, XOR, dan transformasi matriks MDS secara terbalik. Setelah seluruh ronde selesai, proses ditutup dengan *initial whitening*, yaitu XOR terhadap subkunci awal untuk menghasilkan plaintext asli.

G. Pengujian Sistem

Penelitian ini mengimplementasikan *hybrid encryption* menggunakan algoritma Twofish dan Kyber KEM untuk mengamankan *file* dokumen berformat .docx dan .pdf. Pengujian dilakukan terhadap berbagai jenis *file* dengan ukuran berbeda, yaitu 100 KB, 1 MB, 5 MB, 10 MB, serta *file* yang berisi teks, gambar, dan kombinasi keduanya. Setiap *file* dienkripsi menggunakan Twofish, dengan kunci enkripsi yang diperoleh dari hasil enkapsulasi menggunakan Kyber KEM untuk menghasilkan *shared secret key*.




Gambar 7. Simulasi *Virtual Machine* (VM)

Uji coba dilakukan secara lokal menggunakan satu perangkat dengan dua *Virtual Machine* (VM) yang mensimulasikan komunikasi antara pengirim dan penerima. VM pertama berperan sebagai pengirim yang melakukan *key generation*, menghasilkan *public key* dan *private key*, lalu melakukan *encapsulation* untuk memperoleh *shared secret*. *File* kemudian dienkripsi menggunakan Twofish dan *shared secret* tersebut, lalu dikirim ke VM kedua.

Selama proses transfer, dilakukan pengujian *sniffing* menggunakan Wireshark untuk menganalisis keamanan lalu lintas data. Wireshark digunakan sebagai alat *network packet analyzer* untuk menangkap dan menampilkan informasi detail setiap paket data yang melewati jaringan. Evaluasi ini bertujuan untuk mengetahui apakah data yang dikirim dalam bentuk terenkripsi benar-benar aman dari intersepsi pihak ketiga.

Di sisi penerima, VM kedua menerima file terenkripsi dan hasil *encapsulation*. Kemudian dilakukan *decapsulation* menggunakan private key untuk memperoleh kembali *shared secret*, yang selanjutnya digunakan dalam proses dekripsi file. Hasil dekripsi dibandingkan dengan file asli untuk menguji integritas dan keberhasilan sistem enkripsi yang diimplementasikan.

H. Evaluasi Performa

Pengujian sistem ini bertujuan untuk mengevaluasi performa implementasi algoritma *hybrid encryption* menggunakan Twofish dan Kyber KEM dengan berbagai ukuran file, yaitu 100 KB, 1 MB, 5 MB, dan 10 MB.Seperti Tabel 1(A).

Tabel 1

(A)File Pengujian

| File Size | Format | Keterangan |
|---|---|---|
| 100KB | PDF dan DOCX | Berisi teks dan gambar |
| 1MB | PDF dan DOCX | Berisi teks dan tabel |
| 5MB | PDF dan DOCX | Berisi teks dan tabel |
| 10MB | PDF dan DOCX | Berisi teks dan tabel |

Parameter yang diuji mencakup:

1. Waktu Proses Key Generation oleh Kyber KEM

   Mengukur seberapa cepat sistem menghasilkan pasangan *public key* dan *private key* menggunakan algoritma Kyber KEM. Karena Kyber berbasis kriptografi lattice yang kompleks, pengujian ini penting untuk menilai efisiensi pembangkitan kunci dalam konteks komunikasi aman yang memerlukan waktu respons cepat.
2. Waktu Proses Enkripsi

   Mengukur durasi yang dibutuhkan untuk mengenkripsi data menggunakan Twofish dan *shared secret key* dari Kyber. Parameter ini penting untuk menilai efisiensi enkripsi, terutama dalam aplikasi yang menangani data besar atau membutuhkan transmisi real-time.
3. Waktu Proses Dekripsi

   Mengukur durasi dekripsi data terenkripsi menggunakan Twofish dan *shared secret key* yang diperoleh dari proses *decapsulation*. Pengujian ini menunjukkan seberapa cepat sistem dapat mengembalikan data ke bentuk aslinya, yang penting dalam aplikasi dengan kebutuhan akses cepat terhadap informasi.
4. Kecepatan Enkripsi

   Merupakan rasio antara ukuran file (dalam byte) dengan waktu enkripsi (dalam detik), yang merepresentasikan efisiensi sistem dalam mengubah data menjadi ciphertext.

$$Kecepatan\ enkripsi = \frac{Ukuran\ file\ (byte)}{Waktu\ enrkipsi\ (detik)} \quad (6)$$

5. Kecepatan Dekripsi

   Mengukur efisiensi sistem dalam mengembalikan ciphertext menjadi plaintext, dihitung berdasarkan ukuran file terenkripsi dibagi waktu dekripsi.

$$Kecepatan\ dekripsi = \frac{Ukuran\ file\ (byte)}{Waktu\ dekripsi\ (detik)} \quad (7)$$

## IV. HASIL DAN PEMBAHASAN

A. Hasil Performa Sistem

Pengujian dilakukan terhadap file dokumen (.pdf dan .docx) dengan ukuran 100KB hingga 10MB. Parameter evaluasi mencakup waktu proses key generation, enkripsi, dekripsi, serta kecepatan pemrosesan. Hasil yang didapat pada format pdf seperti Gambar 8.

| No | *File* Size | Key Generation (s) | Enkripsi (s) | Dekripsi (s) | Kecepatan Enkripsi (KB/s) | Kecepatan Dekripsi (KB/s) |
|---|---|---|---|---|---|---|
| 1 | 100KB | 0.2523 s | 0.0179 s | 0.0258 s | 9586.70 KB/s | 4178.49 KB/s |
| 2 | 1MB | 0.2406 s | 0.1600 s | 0.2788 s | 9725.46 KB/s | 4543.18 KB/s |
| 3 | 5MB | 0.2433 s | 0.8066 s | 1.3507 s | 9498.99 KB/s | 4849.93 KB/s |
| 4 | 10MB | 0.2392 s | 1.5071 s | 2.5186 s | 9796.23 KB/s | 4599.80 KB/s |

Gambar 8. Hasil Pengujian Format File .pdf

Hasil pengujian menunjukkan bahwa waktu *key generation* untuk seluruh file berkisar antara 0,2392 hingga 0,2523 detik, dengan variasi yang sangat kecil. Hal ini menandakan bahwa proses *key generation* Kyber KEM tidak dipengaruhi oleh ukuran file karena hanya dilakukan satu kali di awal dan digunakan secara konsisten dalam proses enkripsi dan dekripsi.

Pada parameter waktu enkripsi, terlihat adanya peningkatan yang signifikan seiring bertambahnya ukuran file. Peningkatan ini bersifat linear dan mulai meningkat tajam pada file berukuran di atas 5 MB. Hal ini wajar karena Twofish merupakan algoritma *block cipher* yang mengenkripsi data per blok, sehingga semakin besar ukuran file, semakin banyak blok yang perlu diproses.

Waktu dekripsi juga menunjukkan pola peningkatan yang serupa dengan enkripsi, namun dengan nilai yang umumnya lebih tinggi, terutama pada file berukuran 5 MB dan 10 MB. Hal ini kemungkinan disebabkan oleh adanya overhead tambahan dalam proses *unpadding* dan penulisan ulang file hasil dekripsi ke dalam disk.

Dari segi performa, kecepatan enkripsi berada dalam rentang yang stabil, yaitu antara 9500 hingga 9800 KB/s. Bahkan, pada file berukuran 10 MB, kecepatan enkripsi meningkat hingga 9796,23 KB/s, menunjukkan efisiensi sistem yang baik. Kecepatan dekripsi berkisar antara 4178,49 hingga 4849,93 KB/s. Meskipun sedikit lebih rendah

dibandingkan kecepatan enkripsi, performanya tetap stabil dan menunjukkan kemampuan sistem dalam mendekripsi data dengan kecepatan yang dapat diterima. Secara keseluruhan, hasil pengujian menunjukkan bahwa implementasi hybrid encryption menggunakan Kyber KEM dan Twofish memberikan kinerja yang efisien dan konsisten.

Hasil pengujian pada format .docx didapatkan seperti pada Gambar berikut.

| No | *File* Size | Key Generation (s) | Enkripsi (s) | Dekripsi (s) | Kecepatan Enkripsi (KB/s) | Kecepatan Dekripsi (KB/s) |
|---|---|---|---|---|---|---|
| 1 | 100KB | 0.2500 s | 0.0215 s | 0.0333 s | 9309.12 KB/s | 4362.59 KB/s |
| 2 | 1MB | 0.2436 s | 0.1589 s | 0.2551 s | 9628.19 KB/s | 4680.87 KB/s |
| 3 | 5MB | 0.2348 s | 0.7615 s | 1.4097 s | 9488.22 KB/s | 4626.18 KB/s |
| 4 | 10MB | 0.2588 s | 1.5831 s | 2.6045 s | 9494.68 KB/s | 4696.62 KB/s |

Gambar 9. Hasil Pengujian Format File .docx

Pengujian terhadap file berformat DOCX dilakukan dengan tahapan dan ukuran file yang sama seperti pada file PDF, namun ditemukan beberapa perbedaan signifikan, khususnya pada file berukuran besar. Pada proses *key generation*, tidak terdapat perbedaan waktu yang mencolok karena proses ini bersifat independen dan hanya dilakukan sekali di awal, tanpa bergantung pada jenis atau ukuran file.

Waktu enkripsi pada file DOCX cenderung sedikit lebih tinggi dibandingkan PDF. Selisih waktu ini relatif kecil pada file berukuran kecil, namun pada ukuran besar seperti 10 MB, perbedaannya dapat mencapai lebih dari 0,0760 detik. Perbedaan ini diduga disebabkan oleh kompleksitas struktur internal file DOCX yang berbasis XML terkompresi, sehingga pola blok datanya lebih rumit dibandingkan struktur file PDF yang lebih sederhana.

Waktu dekripsi pada file DOCX juga lebih lama daripada PDF. Hal ini kemungkinan besar disebabkan oleh banyaknya komponen dalam file DOCX, seperti gambar, teks, dan elemen styling dalam bentuk XML, yang memerlukan proses penulisan ulang ke disk secara lebih kompleks saat dilakukan dekripsi.

Meski terdapat perbedaan waktu proses, sistem hybrid encryption yang menggabungkan Kyber KEM dan Twofish tetap menunjukkan kecepatan enkripsi dan dekripsi yang relatif stabil. Kecepatan enkripsi berkisar antara 9300 hingga 9500 KB/s, sementara kecepatan dekripsi berada pada kisaran 4600 hingga 4700 KB/s. Stabilitas performa ini menunjukkan bahwa sistem mampu menangani berbagai ukuran dan format file tanpa mengalami penurunan kinerja yang signifikan, sehingga cocok untuk diterapkan dalam berbagai skenario pengamanan dokumen digital.

Perlu dicatat bahwa dalam pengujian ini, sisi pengirim dijalankan pada komputer lokal (host) dengan spesifikasi RAM sebesar 8GB dan 6 Cores, sedangkan sisi penerima dijalankan di dalam virtual machine (VM) dengan alokasi RAM hanya 4GB dan 2 Cores. Perbedaan spesifikasi perangkat ini dapat memberikan pengaruh terhadap hasil pengujian, khususnya pada proses dekripsi yang dilakukan di VM dengan sumber daya yang lebih terbatas.

Dalam pengujian tambahan ini, sistem diuji dengan menempatkan sisi pengirim dan penerima pada dua lingkungan *virtual machine* (VM) yang terpisah, masing-masing dialokasikan sumber daya terbatas, yaitu 4GB RAM dan 2 CPU cores.

| File Size | Key Generation (s) | Enkripsi (s) | Dekripsi (s) | Kecepatan Enkripsi (KB/s) | Kecepatan Dekripsi (KB/s) |
|---|---|---|---|---|---|
| 100KB | 0.2509 s | 0.0259 s | 0.0360 s | 4671.25 KB/s | 3722.22 KB/s |
| 1MB | 0.2456 s | 0.2291 s | 0.2869 s | 4690.17 KB/s | 3745.26 KB/s |
| 5MB | 0.2545 s | 1.1473 s | 1.2770 s | 4686.31 KB/s | 4210.33 KB/s |
| 10MB | 0.2482 s | 2.2563 s | 2.4813 s | 4540.33 KB/s | 4128.68 KB/s |

Gambar 10. Hasil Pengujian VM to VM

Hasil pengujian sistem *hybrid encryption* Kyber KEM dan Twofish dalam skenario komunikasi antar Virtual Machine (VM to VM) menunjukkan performa yang cukup stabil dan efisien meskipun dijalankan dalam lingkungan dengan spesifikasi terbatas. Pengujian dilakukan terhadap empat ukuran file, yaitu 100KB, 1MB, 5MB, dan 10MB.

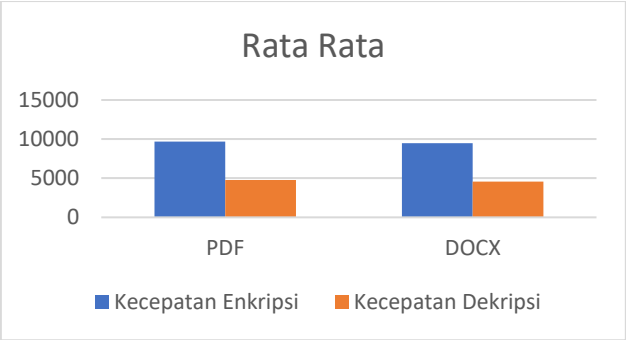
Proses *key generation* menunjukkan waktu yang relatif konstan, berkisar antara 0,24 hingga 0,25 detik, karena hanya dilakukan satu kali dan tidak bergantung pada ukuran file. Sebaliknya, waktu enkripsi dan dekripsi meningkat secara signifikan seiring bertambahnya ukuran file. Sebagai contoh, waktu enkripsi naik dari 0,0259 detik untuk file 100KB menjadi 2,2563 detik untuk file 10MB. Waktu dekripsi juga meningkat dari 0,0360 detik menjadi 2,4813 detik untuk ukuran file yang sama.

Dari sisi kecepatan, proses enkripsi mencatatkan kecepatan tertinggi sebesar 4690,17 KB/s (pada file 1MB), sementara dekripsi tertinggi mencapai 4210,33 KB/s (pada file 5MB). Secara umum, kecepatan enkripsi lebih tinggi dibandingkan dekripsi pada semua ukuran file, dengan nilai kecepatan enkripsi yang relatif stabil di kisaran 4500–4700 KB/s. Kecepatan dekripsi menunjukkan peningkatan signifikan pada file 5MB dan 10MB, mencerminkan efisiensi sistem meskipun terdapat beban pemrosesan lebih besar.

Untuk mengevaluasi efektivitas algoritma simetris dalam sistem *hybrid encryption* ini, dilakukan pengujian perbandingan antara algoritma Twofish dan AES, yang keduanya dipadukan dengan algoritma asimetris Kyber KEM sebagai mekanisme pengelola kunci.

| File Size | Key generation (s) | Enkripsi (s) | Dekripsi (s) | Kecepatan Enkripsi (KB/s) | Kecepatan Dekripsi (KB/s) |
|---|---|---|---|---|---|
| 100KB | 0.4237 s | 0.0041 s | 0.0027 s | 24763.85 KB/s | 37870.36 KB/s |
| 1MB | 0.4330 s | 0.0098 s | 0.0142 s | 109714.92 Kb/s | 75837.52 KB/s |
| 5MB | 0.4447 s | 0.0375 s | 0.0463 s | 143441.44 KB/s | 116217.34 KB/s |
| 10MB | 0.4376 s | 0.0705 s | 0.1376 s | 145244. 04 KB/s | 74468.89 KB/s |

Gambar 11. Hasil Pengujian Kyber dan AES




Gambar 12. Grafik Rata rata Kecepatan

Pengujian performa sistem enkripsi hybrid yang menggabungkan algoritma pascakriptografi Kyber KEM dengan algoritma simetris AES dilakukan terhadap empat variasi ukuran file: 100KB, 1MB, 5MB, dan 10MB. Setiap pengujian mencatat waktu proses untuk key generation, enkripsi, dan dekripsi, serta menghitung kecepatan enkripsi dan dekripsi dalam satuan KB/s.

Hasil pengujian menunjukkan bahwa waktu proses untuk key generation relatif stabil pada semua ukuran file, yakni berada di kisaran 0.24–0.25 detik, serupa dengan hasil pengujian pada kombinasi Kyber KEM + Twofish. Hal ini menunjukkan bahwa proses key generation tidak bergantung pada ukuran file karena hanya dilakukan satu kali di awal komunikasi.

Performa AES terlihat sangat menonjol dalam aspek kecepatan. Pada file berukuran kecil (100KB), waktu enkripsi dan dekripsi masing-masing hanya 0.0041 detik dan 0.0027 detik, dengan kecepatan enkripsi mencapai 24.763,85 KB/s dan dekripsi 37.870,36 KB/s. Kecepatan ini meningkat drastis seiring bertambahnya ukuran file. Misalnya, pada file berukuran 1MB, kecepatan enkripsi mencapai 109.714,92 KB/s, dan dekripsi 75.837,52 KB/s.

Kinerja AES semakin optimal pada file besar, yaitu 5MB dan 10MB. Kecepatan enkripsi mencapai 143.441,44 KB/s (5MB) dan 145.244,04 KB/s (10MB), sedangkan kecepatan dekripsi mencapai 74.668,68 KB/s dan 74.468,78 KB/s pada ukuran yang sama. Waktu proses enkripsi dan dekripsi pun tetap tergolong cepat, yaitu hanya 0.0705 detik dan 0.1376 detik untuk file 10MB.

Tren ini menunjukkan bahwa AES sangat efisien dalam menangani data dalam blok besar. Overhead awal yang kecil menjadikan kecepatan relatif meningkat seiring bertambahnya ukuran file, bukan menurun. Selain itu, selisih antara kecepatan enkripsi dan dekripsi juga cukup konsisten, menandakan kestabilan performa dalam kedua proses tersebut.

Dibandingkan dengan kombinasi Kyber KEM + Twofish, yang sebelumnya menunjukkan kecepatan enkripsi-dekripsi rata-rata hanya sekitar 4.000–9.800 KB/s, penggunaan AES memberikan keunggulan signifikan dari sisi kecepatan dan efisiensi proses. Meski begitu, Twofish masih relevan untuk digunakan dalam skenario dengan keterbatasan sumber daya, seperti dalam lingkungan virtual machine, karena performanya yang stabil dan penggunaan memori yang lebih ringan.

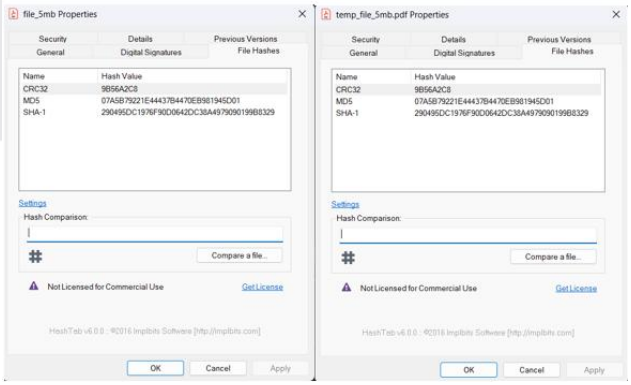
Gambar 12 menunjukkan grafik rata rata dari kecepatan enkripsi dan dekripsi berdasarkan format file yang di uji.

Berdasarkan hasil perhitungan rata-rata yang ditampilkan pada Gambar IV.24, terlihat bahwa kecepatan enkripsi untuk kedua format file—PDF dan DOCX—menunjukkan performa yang sangat tinggi dan hampir setara, dengan nilai mendekati 9500 KB/s. Sementara itu, untuk kecepatan dekripsi, kedua format juga memiliki performa yang relatif seimbang. Namun, file DOCX menunjukkan keunggulan tipis dengan rata-rata kecepatan dekripsi sebesar 4764,85 KB/s, dibandingkan 4562,32 KB/s pada file PDF.

Perbedaan ini menunjukkan bahwa sistem hybrid encryption yang digunakan, yakni kombinasi Kyber KEM dan Twofish, memiliki efisiensi tinggi dalam memproses kedua jenis file meskipun memiliki struktur internal yang berbeda. File DOCX yang berbasis XML cenderung memiliki struktur data yang lebih tersegmentasi dan ringan, sehingga sedikit lebih cepat diproses saat dekripsi. Sebaliknya, file PDF sering kali mengandung elemen visual dan metadata yang kompleks, yang dapat menyebabkan proses dekripsi sedikit lebih lambat. Kendati demikian, selisih kecepatan ini tidak signifikan secara performa keseluruhan, sehingga sistem tetap dinilai stabil dan efisien untuk berbagai format dokumen.

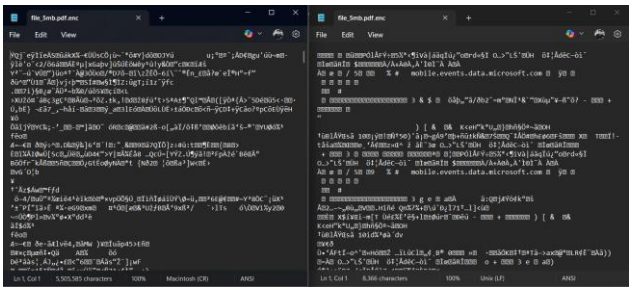
## B. Hasil Integritas File

Setelah mengimplementasikan setiap algoritma yang dibahas sebelumnya. Pengujian selanjutnya yaitu menguji integritas data dari penerapan integrasi Kyber KEM dan Twofish. Hasil implementasi menunjukkan bahwa integritas file asli dan file terdekripsi CRC32, MD5, dan SHA-1 adalah sama. Detail dapat dilihat pada gambar berikut.




Gambar 13. Hasil Hash Values

File juga terenkripsi dengan baik berupa format .enc seperti gambar berikut.



Gambar 14. Potongan Data Terenkripsi

## V. KESIMPULAN

Berdasarkan hasil implementasi dan pengujian yang telah dilakukan, dapat disimpulkan bahwa sistem Hybrid Encryption yang menggabungkan algoritma Twofish dan Kyber Key Encapsulation Mechanism (KEM) berhasil diimplementasikan dengan baik. Algoritma Twofish digunakan sebagai kriptografi simetris untuk proses enkripsi data atau file, sedangkan Kyber KEM berfungsi sebagai algoritma kriptografi asimetris untuk pertukaran kunci secara aman. Proses integrasi dilakukan melalui mekanisme di mana file terlebih dahulu dienkripsi menggunakan Twofish, kemudian kunci Twofish tersebut dienkapsulasi menggunakan Kyber. Sistem ini mampu menjalankan seluruh rangkaian proses—mulai dari enkripsi, enkapsulasi, pengiriman, hingga dekripsi dan dekapsulasi—dengan baik dan menyeluruh.

Dari sisi performa, sistem menunjukkan waktu eksekusi yang cepat dan konsisten pada pengujian terhadap file berukuran kecil hingga menengah. Sebagai contoh, untuk file PDF berukuran 100KB, total waktu proses enkripsi dan dekripsi hanya memerlukan 0,4799 detik, dan untuk file berukuran 1MB memerlukan waktu 0,8564 detik. Hal ini menunjukkan bahwa algoritma Twofish cukup efisien dalam menangani file dengan ukuran tersebut. Pada file berukuran menengah, seperti 5MB, waktu proses meningkat menjadi 2,5791 detik, dan mencapai 4,4455 detik pada file berukuran besar (10MB). Peningkatan waktu proses ini menunjukkan bahwa ukuran file memiliki pengaruh langsung terhadap performa sistem, terutama pada tahap enkripsi dan dekripsi yang menggunakan algoritma simetris. Meski demikian, waktu eksekusi secara keseluruhan masih berada dalam kategori layak dan dapat diterima, terutama dalam konteks penggunaan untuk pengiriman data terenkripsi pada file berukuran kecil hingga menengah.

## REFERENSI


[1] M. Azhari, D. I. Mulyana, F. J. Perwitosari, and F. Ali, "Implementasi Pengamanan Data pada Dokumen Menggunakan Algoritma Kriptografi Advanced Encryption Standard (AES)," *J. Pendidik. Sains dan Komput.*, vol. 2, no. 01, pp. 163–171, 2022, doi: 10.47709/jpsk.v2i01.1390.

[2] N. Wachid Hidayatulloh, M. Tahir, H. Amalia, N. Afdlolul Basyar, A. Faizal Prianggara, and M. Yasin, "Mengenal Advance Encrytion Standard (AES) Sebagai Algoritma Kriptografi Dalam Mengamankan Data," *Digit. Transform. Technol.*, vol. Vol.03, no. No.1, pp. 1–10, 2023, [Online]. Available: https://jurnal.itscience.org/index.php/digitech/article/view/2293

[3] M. Anastasova, P. Kampanakis, and J. Massimo, "PQ-HPKE: Post-Quantum Hybrid Public Key Encryption," *Cryptol. ePrint Arch.*, pp. 1–10, 2022.

[4] C. Kurniawan, M. F. Magfur, and F. Fauziah, "Analisis Perbandingan Ruang Dan Waktu Algoritma Enkripsi Blowfish Dan Twofish Pada Enkripsi Dan Deskripsi Berkas Menggunakan Modul Python," *E-Link J. Tek. Elektro dan Inform.*, vol. 19, no. 1, p. 7, 2024, doi: 10.30587/e-link.v19i1.6592.

[5] S. Siswanto, A. Saputro, G. P. Utama, and B. H. Prasetyo, "Penerapan Algoritma Kriptografi Twofish Untuk Mengamankan Data File," *Bit (Fakultas Teknol. Inf. Univ. Budi Luhur)*, vol. 18, no. 1, pp. 9–18, 2021, doi: 10.36080/bit.v18i1.1446.

[6] V. Maram and K. Xagawa, "Post-quantum Anonymity of Kyber," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 13940 LNCS, pp. 3–35, 2023, doi: 10.1007/978-3-031-31368-4_1.

[7] D. Agustina Akmal *et al.*, "Kombinasi Kriptografi Modern Dalam Keamanan Pesan Teks," *Saturnus J. Teknol. dan Sist. Inf.*, vol. 2, no. 4, pp. 11–17, 2024, [Online]. Available: https://doi.org/10.61132/saturnus.v2i3.204

[8] Fachrul Dhika Ardiansyah, Alfina Damayanti, Clarizza Azzahra Mudya Putri, Ayu Fitria Dinda Rany, Syaidin Joyo Biroso, and Muhlis Tahir, "Implementasi Kriptografi Caesar Chiper Pada Aplikasi Enkripsi Dan Dekripsi," *J. Ilm. Sist. Inf. dan Ilmu Komput.*, vol. 3, no. 1, pp. 105–112, 2023, doi: 10.55606/juisik.v3i1.438.

[9] B. P. Aji, "Teknik Penyembunyian Pesan Pdf Terenkripsi Menggunakan Algoritma Twofish Dan Steganografi End Of File Dalam Media Gambar," 2021, [Online]. Available: http://repository.uir.ac.id/id/eprint/8913

[10] J. Banjarnahor, "JURNAL ARMADA INFORMATIKA STMIK Methodist Binjai PENGGUNAAN ALGORITMA TWOFISH UNTUK PENGAMANAN DATA TEKS," 2023, [Online]. Available: https://doi.org/

[11] L. Wan *et al.*, "A Novel High-Performance Implementation of CRYSTALS-Kyber with AI Accelerator," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 13556 LNCS, pp. 514–534, 2022, doi: 10.1007/978-3-031-17143-7_25.

[12] D. T. Dam, T. H. Tran, V. P. Hoang, C. K. Pham, and T. T. Hoang, "A Survey of Post-Quantum Cryptography: Start of a New Race," *Cryptography*, vol. 7, no. 3, pp. 1–18, 2023, doi: 10.3390/cryptography7030040.

[13] P. Kuppuswamy, S. Q. Y. A. K. Al-Maliki, R. John, M. Haseebuddin, and A. A. S. Meeran, "A hybrid encryption system for communication and financial transactions using RSA and a novel symmetric key algorithm," *Bull. Electr. Eng. Informatics*, vol. 12, no. 2, pp. 1148–1158, 2023, doi: 10.11591/eei.v12i2.4967.

[14] Q. Zhang, "An overview and analysis of hybrid encryption: The combination of symmetric encryption and asymmetric encryption," *Proc. - 2021*

*2nd Int. Conf. Comput. Data Sci. CDS 2021*, no. September, pp. 616–622, 2021, doi: 10.1109/CDS52072.2021.00111.