Analisis Keamanan Pesan Teks Menggunakan Kriptografi Hybrid CBC dan RC4

Aslamadin Alvian Haz [a,1], Mushlihudin [b,2.*]

[a] Program Studi Informatika, Universitas Ahmad Dahlan, Jln. Ring Road Selatan, Daerah Istimewa Yogyakarta 55191, Indonesia
[b] Program Studi Informatika, Universitas Ahmad Dahlan, Jln. Ring Road Selatan, Daerah Istimewa Yogyakarta 55191, Indonesia
[1] aslamadin1600018092@webmail.uad.ac.id ; [2] mushlihudin@tif.uad.ac.id
* Penulis Korespondensi

ABSTRAK

Saat ini pertukaran pesan meningkat sangat bagus terutama kualitas pada kemudahan dan kecepatan. Disisi lain kerentanan terhadap penyadapan dan pencurian merupakan ancaman signifikan. Keamanan pesan teks menjadi sangat penting, terutama dalam aplikasi berbasis web yang rentan terhadap ancaman keamanan. Penelitian ini bertujuan untuk mengkaji implementasi kombinasi algoritma *Cipher Block Chaining* (CBC) dan *Rivest Cipher* 4 (RC4) untuk mengamankan pesan teks dalam format .txt. Pengujian *avalanche effect* menunjukkan bahwa algoritma ini memiliki tingkat keamanan yang baik, dengan nilai yang berkisar antara 45% hingga 48%, yang memenuhi standar keamanan kriptografi. Meskipun demikian, terdapat keterbatasan dalam dekripsi karakter extended ASCII (nilai desimal 128-255), yang tidak dapat dikembalikan ke bentuk aslinya setelah dienkripsi. Pengujian menunjukkan bahwa jumlah karakter dalam file berpengaruh signifikan terhadap waktu eksekusi, dengan semakin banyak karakter, waktu enkripsi dan dekripsi semakin lama. Selain itu, rata-rata waktu proses dekripsi lebih cepat dibandingkan dengan enkripsi, terutama ketika menggunakan kunci yang lebih pendek seperti "Uad". Hasil penelitian ini menunjukkan bahwa kombinasi CBC dan RC4 dapat diimplementasikan secara efektif untuk pengamanan pesan dalam aplikasi berbasis web, dengan catatan adanya optimisasi lebih lanjut terkait kinerja dan jenis karakter yang digunakan.

Riwayat Artikel
Diterima 3 Juni 2024
Diperbaiki 28 Juni 2024
Diterbitkan 30 Juni 2024

Kata Kunci
Keamanan Pesan
Algoritma CBC
Algoritma RC4
Algoritma kriptografi *hybrid avalanche effect*

1. Pendahuluan

Dalam era teknologi informasi dan komunikasi saat ini, pertukaran data digital melalui internet telah menjadi bagian integral dari kehidupan sehari-hari, terutama dalam konteks aplikasi berbasis web. Internet menyediakan informasi yang lebih cepat serta pengguna dapat dengan mudah mendapatkannya sepanjang waktu menggunakan *smartphone*, tablet, *notebook* dan komputer [1][2][3]. Pasar perangkat digital yang berkembang pesat telah meningkatkan kebutuhan akan keamanan pesan, termasuk keamanan pesan teks yang sering dipertukarkan melalui berbagai platform digital [4]. Meskipun pertukaran pesan ini menawarkan kemudahan dan kecepatan, kerentanannya terhadap penyadapan dan pencurian data menimbulkan kekhawatiran yang signifikan.

Pertukaran data digital merupakan proses transfer data yang terstruktur antar sistem komputer dengan format standar yang telah disepakati [5].Salah satu bentuk pertukaran data dan informasi berupa pesan *text* dapat dilakukan melalui media sosial, menggunakan *platforms* media sosial mendorong seseorang dapat membangun komunitas dan berkomunikasi secara *online* kemudian membagikan informasi pribadi, berita, opini dan berbisnis [6]. Pesan *text* merupakan data atau informasi yang dapat dibaca dan dipahami maknanya, pesan *text* dapat berupa data atau informasi yang dikirim dan diterima ataupun berupa data atau informasi yang disimpan [7]. Namun, pertukaran data dan informasi digital memiliki kerentanan terhadap penyadapan, oleh karena itu penting untuk melakukan pengamanan terhadap data dan informasi yang bersifat pribadi dan rahasia. Perkembangan teknik untuk mengambil data secara illegal sering mengakibatkan pencurian data [8].

Keamanan merupakan aspek penting dalam transaksi pertukaran data dan informasi yang bersifat rahasia melalui jaringan internet, keamanan menjamin kerahasiaan pesan *text* agar tetap terjaga ketika proses pertukaran data dan informasi [9]. Penyadapan terhadap data dan informasi dapat diatasi dengan melakukan enkripsi, enkripsi merupakan metode untuk mengamankan data dan informasi dengan mengacak pesan sehingga tidak dapat dibaca atau secara sederhananya enkripsi merupakan proses merubah *plaintext*(pesan asli) menjadi *ciphertext*(pesan teracak) [10][11].

Kritografi adalah upaya yang melibatkan teknik dan prinsip matematika untuk memastikan keamanan informasi, termasuk kerahasiaan, integritas data, dan otentikasi [12]. Untuk melakukan proses enkripsi, telah dikembangkan berbagai algoritma, salah satunya adalah *Cipher Block Chaining* (CBC), sebuah algoritma kriptografi modern yang menggunakan operasi *block cipher* dengan meng-XOR setiap blok *plaintext*. Algoritma CBC memiliki keunggulan setiap blok *plaintext* yang sama tidak akan menghasilkan blok *cipherteks* yang sama. Namun kunci yang digunakan setiap blok sama, sehingga memungkinkan terjadinya cipherteks yang berulang [13]. Algoritma *Rivest Cipher* 4 (RC4) merupakan salah satu *stream cipher*, yaitu memproses inputan data, pesan atau informasi dalam satu waktu. RC4 memiliki dua fase yaitu proses inisiasi kunci (KSA dan PRGA) dan enkripsi [14]. Algoritma RC4 memiliki kelebihan dari sisi performa dikarenakan proses enkripsinya cukup sederhana. Algoritma RC4 memiliki kelemahan pada *array* S yang dapat berulang jika kunci yang dimasukan memiliki karakter yang berulang sehingga menghasilkan nilai permutasi yang sama [15] [16].

Perkembangan ilmu kriptografi memberikan peluang para kriptanalis dapat memecahkan pesan rahasia dengan berbagai macam teknik yang ada. Kriptanalis merupakan person yang mendalami tentang *cipher*, *ciphertext*, *cryptosystem* dengan tujuan untuk menemukan celah atau kelemahan dari suatu algoritma penyandian dan memecahkan *ciphertext* tanpa menggunakan kunci yang legal [17]. Algoritma kriptografi dapat diketahui tingkat keamanannya berdasarkan pada nilai *avalanche effect*. *Avalanche effect* merupakan jumlah perubahan bit pada *plaintext* ketika telah dienkripsi menjadi *ciphertext* [18]. Nilai *avalanche effect* dapat dikatakan bagus apabila bernilai antara 45-60% (50% merupakan hasil yang sangat bagus). Hal ini berpengaruh terhadap tingkat kesulitan kriptanalis untuk melakukan pemecahan *ciphertext* [13].

$$xAvalanche\ Effect\ (AE) = \frac{\sum bit\_change}{\sum bit\_totals} \times 100\%$$ [13].

Upaya mengatasi penyadapan dan meningkatkan keamanan terhadap kriptanalisis maka dilakukan penerapan algoritma kriptografi secara *hybrid* menggunakan algoritma CBC dan RC4. Kriptografi *hybrid* merupakan pemanfaatan sandi dari algoritma yang berbeda secara bersama dengan memanfaatkan keunggulan dari tiap algoritma tersebut [19]. Secara terpisah, kriptografi *hybrid* juga dapat disebut sebagai penggabungan antara dua tingkatan kunci yaitu kunci rahasia (simetris), yang disebut juga session key (kunci sesi), untuk enkripsi data dan pasangan kunci rahasia (kunci publik) untuk pemberian tanda tangan digital serta melidungi kunci simetris [20].

2. Metode

Objek penelitian ini adalah algoritma CBC yang dikombinasikan dengan algoritma RC4, yang kemudian diimplementasikan untuk mengenkripsi pesan dalam bentuk file teks dengan format *.txt. Fokus penelitian ini adalah pada implementasi kombinasi algoritma CBC dan RC4 guna meningkatkan keamanan dan kinerja enkripsi. Studi pustaka dilakukan untuk mempelajari dan menganalisis jurnal, literatur, atau buku dengan tujuan memperoleh referensi yang dapat membantu penulis dalam melaksanakan penelitian. Untuk memahami kelebihan dan kelemahan dari objek penelitian serta memperkuat latar belakang masalah, studi pustaka dilakukan melalui review jurnal, buku, dan media online. Studi pustaka ini merujuk pada topik-topik yang berkaitan dengan algoritma CBC, algoritma RC4, dan kriptografi. Proses observasi dilakukan untuk memahami lebih dalam proses enkripsi dan dekripsi. Selain itu, percobaan manual juga dilakukan untuk memperdalam pemahaman tentang proses enkripsi dan dekripsi sebelum penerapannya dalam program.

Jurnal Sarjana Teknik Informatika
Vol. 12., No. 2, Juni 2024, pp. 70-80

Gambar 1. Tahapan Penelitian

Penelitian ini diawali dengan melakukan metode pengumpulan data berupa studi pustaka dan observasi, kemudian dilanjutkan dengan melakukan tahapan penelitian. Tahapan penelitian meliputi analisis algoritma, perancangan algoritma, implementasi dan pengujian. Gambar 1 merupakan tahapan yang dilakukan pada penelitian ini.

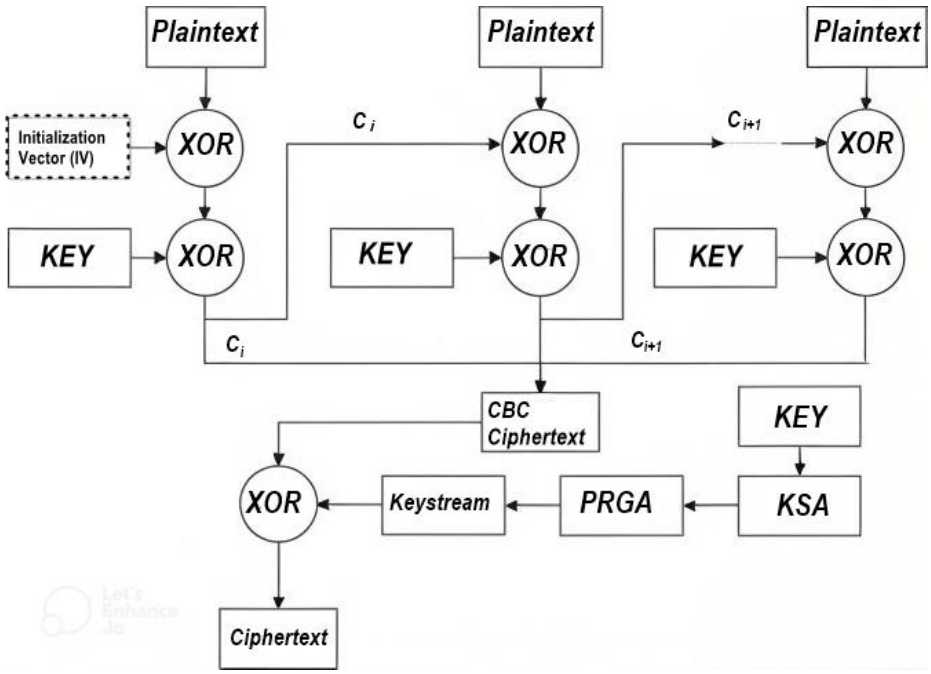
Proses enkripsi pesan dengan menggunakan dua tahap enkripsi bertingkat yaitu CBC dan RC4 dapat dilihat pada gambar 2. *Plaintext* adalah pesan asli yang belum mengalami proses enkripsi. *Plaintext* ini merupakan input awal yang akan dienkripsi untuk melindungi informasi di dalamnya. Proses Enkripsi pertama menggunakan metode CBC dan memberikan hasil blok *ciphertext* pertama. Setelah enkripsi CBC, hasilnya kemudian dienkripsi lagi menggunakan algoritma RC4. Setelah melalui dua tahap enkripsi, yaitu CBC dan RC4, pesan asli (*plaintext*) berubah menjadi *ciphertext*.

Gambar 2. Alur Enkripsi

Proses dekripsi pesan dengan menggunakan dua tahap dekripsi bertingkat yaitu RC4 dan CBC dapat dilihat pada gambar 3. *Cipherteks* ini adalah hasil dari proses enkripsi bertahap sebelumnya, di mana *plaintext* awal telah dienkripsi menggunakan dua algoritma yang berbeda. Langkah pertama dalam proses dekripsi adalah menggunakan algoritma dekripsi RC4. Setelah proses dekripsi RC4, hasilnya berupa *ciphertext* CBC, kemudian diproses melalui dekripsi menggunakan algoritma CBC. Setelah melalui kedua proses dekripsi, pesan asli (*plaintext*) berhasil dipulihkan.

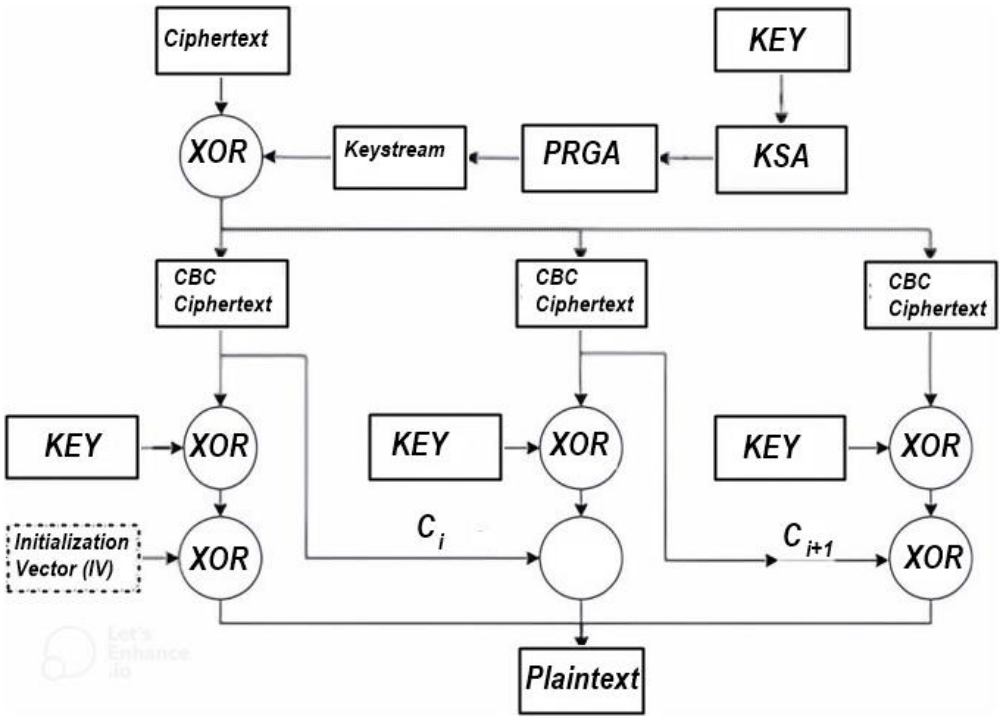
Gambar 3. Alur Dekripsi

Gambar 4 merupakan skema algoritma untuk enkripsi, *plaintext* akan dienkripsi menggunakan algoritma CBC terlebih dahulu kemudian hasilnya akan dienkripsi menggunakan algoritma RC4. Proses enkripsi dengan mengintegrasikan dua algoritma kriptografi, yakni CBC dan RC4, untuk meningkatkan keamanan pesan. Dimulai dengan *plaintext*, setiap blok pesan diolah menggunakan mode CBC, di mana *plaintext* pertama di-XOR dengan *Initialization Vector* (IV), kemudian dienkripsi menggunakan kunci tertentu untuk menghasilkan *ciphertext*. Hasil dari blok pertama ini di-XOR dengan blok *plaintext* berikutnya dan kemudian dienkripsi. Proses ini berlanjut hingga semua blok *plaintext* diproses dan menciptakan *chaining effect* (efek rantai) yang bergantung pada *ciphertext* sebelumnya untuk setiap blok baru. Setelah seluruh blok diolah oleh CBC, hasilnya kemudian diproses oleh algoritma RC4. Dalam RC4, *Key Scheduling Algorithm* (KSA) digunakan untuk menghasilkan keystream melalui *Pseudo-Random Generation Algorithm* (PRGA). *Keystream* ini kemudian di-XOR dengan *output* dari CBC, menghasilkan *ciphertext* final yang merupakan pesan terenkripsi. Kombinasi dari dua metode ini memastikan bahwa pesan terlindungi dengan baik dan membuatnya lebih sulit untuk dipecahkan tanpa mengetahui kunci yang tepat dan memahami proses enkripsi yang kompleks.



Gambar 4. Skema Algoritma Enkripsi

Proses dekripsi seperti pada gambar 5, merupakan proses dekripsi pesan yang telah dienkripsi menggunakan kombinasi RC4 dan CBC, dimulai dari *ciphertext* hingga kembali menjadi *plaintext*. Proses dekripsi diawali dengan penguraian *ciphertext* menggunakan XOR dengan *keystream* yang dihasilkan oleh RC4, yang sebelumnya diinisialisasi melalui *Key Scheduling Algorithm* (KSA) dan *Pseudo-Random Generation Algorithm* (PRGA).



Gambar 5. Skema Algoritma Dekripsi

Hasil XOR ini merupakan CBC *ciphertext* yang kemudian diuraikan menggunakan mode CBC. Dalam CBC, setiap blok *ciphertext* di-XOR dengan blok *ciphertext* sebelumnya setelah dienkripsi dengan kunci yang sama, dimulai dari blok pertama yang menggunakan *Initialization Vector* (IV).

Jurnal Sarjana Teknik Informatika
Vol. 12., No. 2, Juni 2024, pp. 70-80

Proses ini dilakukan berulang kali hingga seluruh blok pesan berhasil diuraikan, menghasilkan *plaintext* asli. Setiap langkah dalam proses dekripsi ini membutuhkan kunci yang tepat dan pemahaman mendalam tentang algoritma yang digunakan untuk memastikan pesan dapat dikembalikan ke bentuk aslinya tanpa kehilangan integritas.

3. Hasil dan Pembahasan

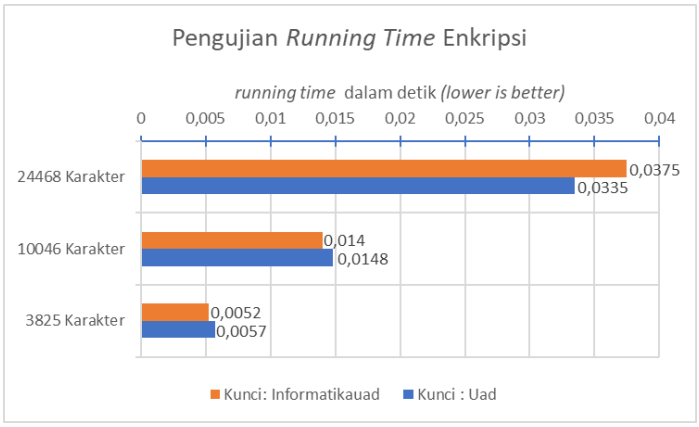
Beberapa mekanisme pengujian dilakukan yaitu pengujian enkripsi dan dekripsi untuk mengetahui tingkat keberhasilannya, pengujian waktu eksekusi (*running time*) untuk menilai performa dari segi kecepatan program yang diimplementasikan, pengujian perhitungan *avalanche effect* untuk mengukur tingkat keamanan algoritma, dan yang terakhir pengujian perbandingan untuk melihat apakah *ciphertext* yang dihasilkan masih dapat dikenali secara visual. Pengujian dilakukan terhadap aplikasi yang sudah dihosting dan dijalankan secara online.

Pengujian enkripsi dan dekripsi dilakukan untuk memastikan apakah rancangan kombinasi algoritma yang diimplementasikan dapat berjalan dengan baik, baik dari segi fungsi enkripsi dan dekripsi maupun dari segi performa. Indikator yang menentukan kelancaran proses enkripsi dan dekripsi dapat dilihat melalui perbandingan antara file asli, file hasil enkripsi, dan file hasil dekripsi, yang mencakup jumlah karakter, ukuran file, dan keberhasilan dalam mengenkripsi atau mendekripsi file.

Tabel 1 Pengujian Enkripsi dan Dekripsi

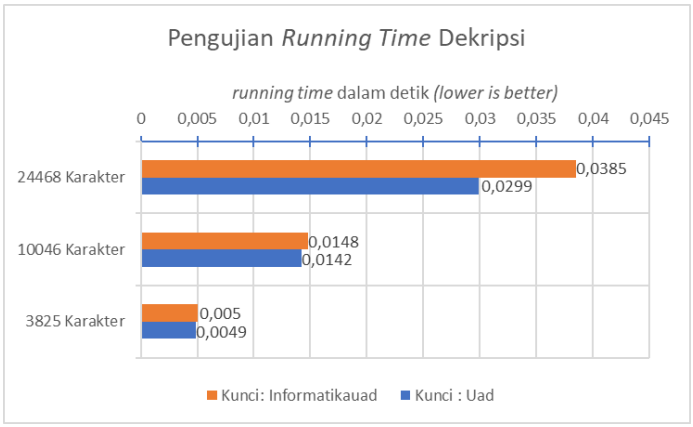
No	Nama File	Ukuran File	Ukuran File Setelah Enkripsi/Dekripsi	**Status Proses** enkripsi dan dekrirpsi
1	Pengujian1.txt	3,73 KB (3.825 bytes)	Ukuran tidak berubah	Proses berhasil
2	Pengujian2.txt	9,81 KB (10.046 bytes)	Ukuran tidak berubah	Proses berhasil
3	Pengujian3.txt	23,8 KB (24.468 bytes)	Ukuran tidak berubah	Proses berhasil
4	Pengujian4.txt	37,4 KB (38.347 bytes)	Ukuran tidak berubah	Proses berhasil
5	Pengujian5.txt	112 KB (115.049 bytes)	Ukuran tidak berubah	Proses berhasil
6	Pengujian6.txt	125 KB (128.638 bytes)	Ukuran tidak berubah	Proses berhasil
7	Pengujian7.txt	237 KB (243.691 bytes)	Ukuran tidak berubah	Proses berhasil
8	Pengujian8.txt	475 KB (487.386 bytes)	Ukuran tidak berubah	Proses berhasil
9	Pengujian9.txt	512 KB (524.288 bytes)	Ukuran tidak berubah	Proses berhasil
10	Pengujian10.txt	513 KB (524.289 bytes)	Tidak dapat dienkripsi /dekripsi	Proses gagal (batas memori tercapai)

Tabel 1 memaparkan hasil dari 10 kali pengujian yang dilakukan terhadap proses enkripsi dan dekripsi file teks menggunakan kombinasi algoritma CBC dan RC4. Setiap pengujian dilakukan dengan file teks berukuran berbeda, untuk mengevaluasi efektivitas dan kinerja algoritma kombinasi ini dalam menjaga integritas pesan selama proses enkripsi dan dekripsi. Dari 10 pengujian sebanyak 9 di antaranya berhasil menjalankan proses enkripsi dan dekripsi dengan ukuran file yang tetap sama sebelum dan sesudah proses dilakukan. Hal ini menunjukkan bahwa algoritma kombinasi CBC dan RC4 mampu mengamankan pesan tanpa mengubah ukuran atau jumlah karakter dalam file. Pengujian pada file ke-5 dan ke-6 mengungkap adanya kelemahan dalam menangani karakter dengan nilai desimal 128-255, yang dikenal sebagai extended ASCII. Karakter-karakter ini tidak dapat didekripsi kembali ke bentuk aslinya setelah dienkripsi, ada batasan dalam penanganan karakter khusus yang memerlukan perhatian lebih lanjut. Pada pengujian ke-10, proses enkripsi dan dekripsi gagal diselesaikan akibat keterbatasan memori pada server yang digunakan. Hal ini mengindikasikan bahwa, meskipun algoritma kombinasi CBC dan RC4 mampu bekerja dengan baik pada file berukuran sedang hingga besar, terdapat batasan teknis yang perlu diatasi, terutama ketika bekerja dengan file yang sangat besar dalam lingkungan dengan sumber daya memori yang terbatas.



Gambar 6. Pengujian *Running Time* Enkripsi

Pada Gambar 6, dilakukan pengujian waktu eksekusi enkripsi dengan tiga set pesan berbeda, masing-masing menggunakan kunci "Uad" dan "Informatikauad". Pengujian pertama dilakukan dengan file yang berisi 3.825 karakter. Hasilnya menunjukkan bahwa menggunakan kunci "Uad", waktu rata-rata yang diperlukan adalah 0,0057 detik setelah 5 kali percobaan, sedangkan dengan kunci "Informatikauad", waktu rata-rata yang dicatat adalah 0,0052 detik. Pengujian kedua dilakukan dengan file yang berisi 10.046 karakter. Hasilnya menunjukkan bahwa waktu rata-rata untuk kunci "Uad" adalah 0,0148 detik setelah 5 kali percobaan, sementara kunci "Informatikauad" mencatat waktu rata-rata 0,014 detik. Pada pengujian terakhir, menggunakan file dengan 24.468 karakter, waktu rata-rata yang diperlukan dengan kunci "Uad" adalah 0,0335 detik, dan dengan kunci "Informatikauad", waktu rata-rata meningkat menjadi 0,0375 detik. Hasil-hasil ini menunjukkan bahwa perbedaan dalam panjang kunci mempengaruhi waktu eksekusi, meskipun secara umum, selisihnya kecil. Kunci yang lebih panjang (Informatikauad) cenderung membutuhkan sedikit lebih banyak waktu untuk proses enkripsi dibandingkan dengan kunci yang lebih pendek (Uad), terutama pada file yang lebih besar.

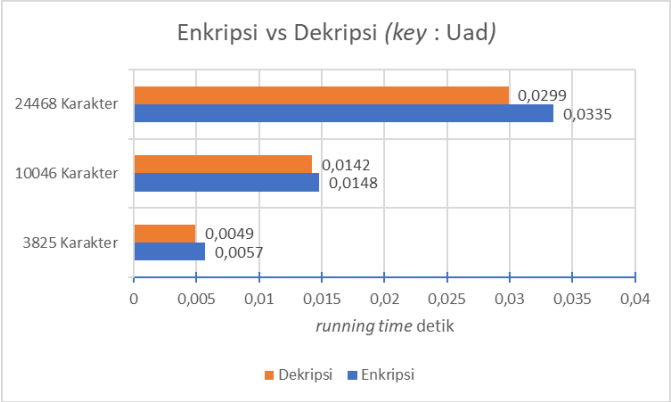


Gambar 7. Pengujian *Running Time* Dekripsi

Setelah melakukan pengujian waktu eksekusi (*running time*) untuk proses enkripsi, pengujian dilanjutkan dengan mengukur waktu eksekusi untuk proses dekripsi. Hasil pengujian waktu eksekusi dekripsi ditampilkan pada Gambar 7. Pada pengujian pertama, file yang berisi 3.825 karakter di-dekripsi menggunakan kunci "Uad". Waktu rata-rata yang diperoleh adalah 0,0049 detik dari 5 kali percobaan. Ketika menggunakan kunci "Informatikauad" untuk file yang sama, waktu rata-rata yang diperoleh sedikit lebih tinggi, yaitu 0,0050 detik. Pengujian kedua dilakukan dengan file yang berisi 10.046 karakter. Hasilnya menunjukkan bahwa waktu rata-rata untuk dekripsi menggunakan kunci "Uad" adalah 0,0142 detik dari 5 kali percobaan. Sementara itu, ketika kunci "Informatikauad" digunakan, waktu rata-rata yang diperlukan meningkat menjadi 0,0148 detik. Pada pengujian terakhir, file yang berisi 24.468 karakter di-dekripsi. Dengan menggunakan kunci "Uad", waktu rata-

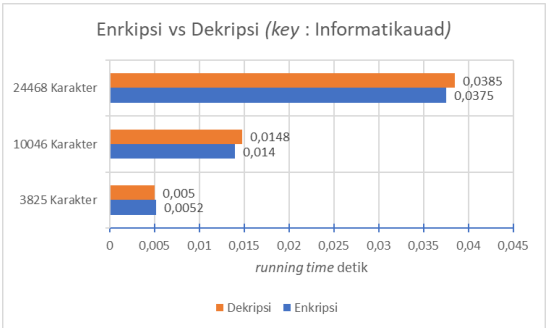
rata yang dicatat adalah 0,0299 detik. Namun, ketika menggunakan kunci yang lebih panjang, yaitu "Informatikauad", waktu rata-rata dekripsi meningkat secara signifikan menjadi 0,0385 detik.

Dari hasil pengujian dekripsi ini, terlihat bahwa panjang kunci mempengaruhi waktu eksekusi dekripsi. Kunci yang lebih panjang, seperti "Informatikauad", cenderung membutuhkan waktu yang lebih lama untuk menyelesaikan proses dekripsi dibandingkan dengan kunci yang lebih pendek, terutama pada file dengan jumlah karakter yang lebih besar. Hal ini menunjukkan adanya *trade-off* antara panjang kunci dan efisiensi waktu dalam proses dekripsi.



Gambar 8. Perbandingan *Running Time* Enkripsi dan Dekripsi *key*:Uad

Gambar 8 menampilkan perbandingan antara waktu eksekusi rata-rata (*running time*) untuk proses enkripsi dan dekripsi menggunakan kunci "Uad" pada file dengan jumlah karakter yang berbeda. Dari hasil pengujian, dapat disimpulkan bahwa waktu rata-rata untuk proses dekripsi umumnya lebih cepat dibandingkan dengan waktu rata-rata untuk proses enkripsi. Pada file dengan 3.825 karakter, waktu dekripsi rata-rata tercatat 0,0049 detik, sementara waktu enkripsi rata-rata adalah 0,0057 detik. Untuk file yang berisi 10.046 karakter, waktu dekripsi rata-rata adalah 0,0142 detik, sedikit lebih cepat dibandingkan waktu enkripsi rata-rata sebesar 0,0148 detik. Pada file terbesar dengan 24.468 karakter, waktu dekripsi rata-rata adalah 0,0299 detik, sementara waktu enkripsi rata-rata adalah 0,0335 detik. Hasil ini menunjukkan bahwa proses dekripsi cenderung lebih efisien dalam hal waktu dibandingkan dengan proses enkripsi ketika menggunakan kunci yang sama. Perbedaan ini dapat diakibatkan oleh kompleksitas tambahan yang diperlukan selama proses enkripsi, yang tidak sepenuhnya ada dalam proses dekripsi. Kesimpulan ini penting untuk mempertimbangkan efisiensi dalam implementasi algoritma enkripsi dan dekripsi di lingkungan yang memerlukan kinerja tinggi.



Gambar 9. Perbandingan *Running Time* Enkripsi dan Dekripsi *key*:Informatikauad

Gambar 9 menunjukkan perbandingan waktu eksekusi rata-rata antara proses enkripsi dan dekripsi menggunakan kunci "Informatikauad" pada berbagai ukuran file. Dari grafik tersebut, terlihat bahwa proses enkripsi umumnya memakan waktu yang lebih lama dibandingkan dengan proses dekripsi. Misalnya, untuk file dengan 24.468 karakter, waktu enkripsi rata-rata adalah 0,0375 detik, sementara waktu dekripsi rata-rata adalah 0,0385 detik. Pada file yang lebih kecil, seperti yang

memiliki 3.825 karakter, perbedaan waktu antara enkripsi dan dekripsi lebih tipis, dengan waktu enkripsi 0,0052 detik dan dekripsi 0,0050 detik. Analisis ini menunjukkan bahwa meskipun perbedaan waktu antara enkripsi dan dekripsi relatif kecil, kunci yang lebih panjang seperti "Informatikauad" dapat menyebabkan sedikit peningkatan waktu eksekusi, terutama pada file yang lebih besar. Hal ini menunjukkan adanya perbaikan dalam implementasi algoritma enkripsi dan dekripsi, terutama ketika efisiensi waktu menjadi faktor kritis.

Tabel 2. Pengujian *Avalanche Effect*

No	*Plaintext*	*Ciphertext*	AE %	Keterangan
1	INFORMATIKA	`ÃÀûRÚ%&cþ~	45.5 %	Bagus
2	PRODI INFORMATIKA	RÉØõj8×à».^V ¼5Ô	48.5 %	Bagus
3	PRODI INFORMATIKA UNIVERSITAS	RÉØõj8×à».^V ¼5Ô «ÄÁBÍ»'ßpE	48.7 %	Bagus
4	PRODI NFORMATIKA UNIVERSITAS AHMAD	RÉØõj8×à».^V ¼5Ô «ÄÁBÍ»'ßpE,®W +Ö•	46.8 %	Bagus
5	PRODI INFORMATIKA UNIVERSITAS AHMAD DAHLAN	RÉØõj8×à».^V¼5Ô «ÄÁBÍ»'ßpE,®W +Ö• túãì€	48.5 %	Bagus

Tabel 2 menunjukkan hasil pengujian *avalanche effect* yang dilakukan pada lima sampel *plaintext* dengan menggunakan kombinasi algoritma CBC dan RC4. Dari pengujian tersebut, didapatkan nilai *avalanche effect* berkisar antara 45% hingga 48%. Rentang nilai ini menunjukkan bahwa kombinasi algoritma CBC dan RC4 memiliki tingkat keamanan yang dapat dikategorikan baik, karena sesuai dengan standar yang mengindikasikan bahwa perubahan kecil pada *plaintext* (seperti perubahan satu bit) menyebabkan perubahan signifikan pada *ciphertext*.

Selain itu, pengujian juga mencakup perbandingan antara dua *ciphertext* yang dihasilkan dari dua kali pemrosesan *plaintext* yang sama menggunakan kunci "Uad". Proses pertama menghasilkan *ciphertext* C1, sedangkan proses kedua menghasilkan *ciphertext* C2, yang merupakan *ciphertext* akhir. Perbedaan yang signifikan antara C1 dan C2 menunjukkan bahwa algoritma kombinasi ini memiliki ketahanan yang baik terhadap serangan diferensial, karena dua hasil enkripsi yang berbeda dapat diperoleh meskipun *plaintext* dan kunci yang digunakan tetap sama.

Berdasar pada hasil dari *avalanche effect* dan perbandingan *ciphertext* mendukung kesimpulan bahwa algoritma kombinasi CBC dan RC4 yang digunakan dalam pengujian ini mampu memberikan keamanan yang tinggi dalam proses enkripsi pesan.

Tabel 3 Pengujian Perbandingan *Plaintext*, C1 dan C2

No	*Plaintext*	C1	C2
1	Fakultas Teknologi Industri	ddzz`@hLĆ`` ``````"r f DlB@f|	~ƒÚ•@ºHß|´&9 [®oÂŒÅvê‹h¥• Âõ
2	Universitas Ahmad Dahlan	B\F@HdhxXpT`` ````-||dV VB	X»æ¯hž|CŠnj+G °mÆ›xè™R¥
3	Informatika – FTI – UAD	zd`Tr`HbxDD` |```` `pR	`ƒÀ»Rš%fc¾~zã- ¾ ø$wü×V

Tabel 3 menyajikan perbandingan antara *plaintext*, C1, dan C2 yang dihasilkan dari enkripsi menggunakan kombinasi algoritma CBC dan RC4. *Plaintext* yang ditampilkan pada kolom pertama adalah teks asli yang akan dienkripsi. Kolom kedua (C1) menunjukkan hasil enkripsi pertama, sementara kolom ketiga (C2) menampilkan hasil enkripsi setelah proses enkripsi ulang. Dari tabel ini, dapat diamati bahwa pada C1 masih terdapat beberapa karakter yang dapat dikenali, seperti huruf alfabet dan tanda baca tertentu. Hal ini menunjukkan bahwa pada enkripsi pertama, meskipun sebagian besar teks telah teracak, masih ada sejumlah karakter yang tetap terlihat dalam bentuk yang dapat dikenali. Namun, setelah dilakukan enkripsi ulang dan menghasilkan C2, karakter-karakter tersebut menjadi jauh lebih teracak dan sulit dikenali. Hal ini menunjukkan bahwa proses enkripsi ulang dengan kombinasi algoritma CBC dan RC4 berhasil meningkatkan tingkat kerumitan *ciphertext*, membuatnya lebih sulit untuk dibaca atau diuraikan tanpa kunci yang tepat.Perubahan

signifikan dari C1 ke C2 ini menegaskan keefektifan kombinasi algoritma dalam menghasilkan *ciphertext* yang aman dan tidak dapat dengan mudah diidentifikasi, yang merupakan indikator penting dalam menilai kekuatan suatu algoritma enkripsi.

Berdasarkan hasil pengujian yang telah dilakukan, kombinasi algoritma CBC dan RC4 menunjukkan potensi yang kuat sebagai metode pengamanan untuk pesan teks dalam format .TXT. Pengujian *avalanche effect* mengindikasikan bahwa tingkat keamanan *ciphertext* yang dihasilkan tergolong baik, dengan nilai yang berkisar antara 45% hingga 48%. Nilai ini menunjukkan bahwa perubahan kecil pada *plaintext* menyebabkan perubahan signifikan pada *ciphertext*, yang meningkatkan kompleksitas dan kesulitan bagi kriptanalis dalam upaya memecahkan atau menguraikan *ciphertext* tanpa mengetahui kunci enkripsi yang tepat.

Tabel 4 Pembanding nilai *Avalanche Effect* penelitian lain.

No	Penelitian	Algoritma yang Digunakan	Nilai *Avalanche Effect* (%)	Kesimpulan
1	Penelitian Ini	CBC dan RC4	45% - 48%	Nilai *avalanche* effect menunjukkan keamanan yang baik, meskipun sedikit lebih rendah dibandingkan penelitian lain.
2	"*Analysis of the Security level of modified CBC algorithm cryptography using avalanche effect*" [13].	CBC dan Vigenere Cipher	51% - 54%	Modifikasi CBC dengan Vigenere Cipher menghasilkan nilai *avalanche effect* yang lebih tinggi.
3	"*Analyse On Avalanche Effect In Cryptography Algorithm*" [21]	Algoritma Cipher Block	50%	Algoritma cipher block yang diteliti memenuhi kriteria avalanche *effect* dengan hasil mencapai 50%.

Tabel 4 membandingkan hasil nilai *avalanche effect* dari penelitian yang dilakukan dengan dua penelitian lainnya yang menggunakan algoritma kriptografi yang berbeda. Nilai *avalanche effect* dari penelitian ini berkisar antara 45% hingga 48%, yang sedikit lebih rendah dibandingkan dengan penelitian sebelumnya, yaitu penelitian oleh [13] yang mencapai nilai antara 51% hingga 54% setelah memodifikasi algoritma CBC dengan Vigenere Cipher, dan penelitian oleh [21] yang mencatat nilai *avalanche effect* sebesar 50% pada algoritma cipher block. Meskipun nilai *avalanche effect* dalam penelitian ini lebih rendah, algoritma kombinasi CBC dan RC4 tetap dikategorikan aman karena hasil tersebut sesuai dengan standar keamanan yang diharapkan, yaitu antara 45% hingga 60%. Perbandingan ini menekankan bahwa meskipun terdapat perbedaan dalam nilai *avalanche effect*, kombinasi CBC dan RC4 masih memenuhi kriteria yang baik untuk melindungi pesan dari serangan kriptanalisis.

4. Kesimpulan

Algoritma kombinasi CBC dan RC4 yang diimplementasikan terbukti efektif dalam menjaga keamanan pesan, dengan nilai *avalanche effect* antara 45% hingga 48%, dan menunjukkan perubahan signifikan pada *ciphertext* saat terjadi perubahan kecil pada *plaintext*. Algoritma ini berfungsi dengan baik dalam aplikasi berbasis web, meskipun terdapat keterbatasan pada dekripsi karakter extended ASCII (nilai desimal 128-255). Jumlah karakter dalam file berpengaruh pada waktu eksekusi, dengan waktu dekripsi yang umumnya lebih cepat dibandingkan enkripsi, terutama saat menggunakan kunci yang lebih pendek. Hasil ini menegaskan bahwa kombinasi algoritma CBC dan RC4 dapat diandalkan untuk mengamankan data, dengan perhatian khusus pada optimisasi kinerja dan karakter yang digunakan.

Daftar Pustaka

[1] M. Danuri, "Development and transformation of digital technology," *Infokam*, vol. XV, no. II, pp. 116–123, 2019.

[2] K. Nassif Jassim *et al.*, "Hybrid cryptography and steganography method to embed encrypted text message within image," in *Journal of Physics: Conference Series*, 2019, vol. 1339, no. 1, doi: 10.1088/1742-6596/1339/1/012061.

[3] D. D. A. Yani, H. S. Pratiwi, and H. Muhardi, "Implementasi Web Scraping untuk Pengambilan Data pada Situs Marketplace," *J. Sist. dan Teknol. Inf.*, vol. 7, no. 4, p. 257, 2019, doi: 10.26418/justin.v7i4.30930.

[4] F. Cochoy, C. Licoppe, M. P. McIntyre, and N. Sörum, "Digitalizing consumer society: equipment and devices of digital consumption," *J. Cult. Econ.*, vol. 13, no. 1, pp. 1–11, 2020, doi: 10.1080/17530350.2019.1702576.

[5] J. Rengamani, F. A. James, R. Srinivasan, S. Poongavanam, and R. Vettriselvan, "Impact on the usage of electronic data interchange (EDI) on the international shipping business in Chennai," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 2, pp. 418–422, 2019.

[6] K. Thakur, T. Hayajneh, and J. Tseng, "Cyber Security in Social Media: Challenges and the Way Forward," *IT Prof.*, vol. 21, no. 2, pp. 41–49, 2019, doi: 10.1109/MITP.2018.2881373.

[7] O. K. Sulaiman, K. Nasution, and S. Y. Prayogi, "Enkripsi Surat Elektronik Menggunakan Metode XXTEA," *Comput. Eng. Sci. Syst. J.*, vol. 4, no. 1, p. 99, 2019, doi: 10.24114/cess.v4i1.12354.

[8] F. A, F. Yanto, I. Iskandar, and Pizaini, "Jurnal Computer Science and Information Technology (CoSciTech) Kombinasi algoritma kriptografi vigenere cipher dengan metode zig-zag dalam pengamanan pesan teks," vol. 4, no. 1, pp. 182–192, 2023, doi: https://doi.org/10.37859/coscitech.v4i1.4787.

[9] S. N. Siregar, "Pengamanan Pesan Teks Menggunakan Algoritma FEAL dan RSA Pada Aplikasi Android," *J. Comput. Syst. Informatics …*, vol. 1, no. 4, pp. 328–336, 2020, [Online]. Available: https://ejurnal.seminar-id.com/index.php/josyc/article/view/351.

[10] F. Nova Hulu, M. Putri, and K. Kunci, "Metode Analitis Enkripsi Dan Dekripsi Dengan Penerapan Algoritma Kriptografi Klasik Ke Dalam Cipher," *J. Elektro dan Telkomunikasi*, pp. 26–34, 2022.

[11] E. Sutanty, M. B. Siregar, and E. Setiyaningsih, "Implementasi Feistel Block Cipher Dalam Enkripsi File Berbentuk Teks," *J. Ilm. Inform. Komput.*, vol. 26, no. 2, pp. 136–148, 2021, doi: 10.35760/ik.2021.v26i2.4238.

[12] R. Munir, "IF4020 Kriptografi - Semester I Tahun 2021/2022," *https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2021-2022/kripto21-22.htm*, 2021. https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2021-2022/kripto21-22.htm (accessed Oct. 21, 2022).

[13] N. R. D. P. Astuti, I. Arfiani, and E. Aribowo, "Analysis of the security level of modified CBC algorithm cryptography using avalanche effect," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 674, no. 1, 2019, doi: 10.1088/1757-899X/674/1/012056.

[14] A. F. Doni, O. A. H. Maria, and S. Hanif, "Implementation of RC4 Cryptography Algorithm for Data File Security," *J. Phys. Conf. Ser.*, vol. 1569, no. 2, pp. 2–8, 2020, doi: 10.1088/1742-6596/1569/2/022080.

[15] R. Mohammed and L. M. Jawad, "Secure Image Encryption Scheme Using Chaotic Maps and RC4 Algorithm," *Solid State Technology*, vol. 63, no. 3, pp. 3450-3465, Nov. 2020

[16] K. N. Suryani, "Algoritma Rc4 Sebagai Metode Enkripsi," *Enkripsi*, vol. 4, p. 5, 2009.

[17] S. Purba, "Kriptanalisis Kunci Publik Algoritma Rabin Menggunakan Metode Kraitchik," *Jurnal Sains dan teknologi*, vol. 11, no. 2. pp. 205–213, 2019, [Online]. Available: https://ejurnal.istp.ac.id/index.php/jsti/article/view/25.

[18] M. Karuppiah, S. Ramanujam, and A. Professor, "Designing an algorithm with high Avalanche Effect," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 11, no. 1, p. 106, 2011, [Online]. Available: https://www.researchgate.net/publication/266468045.

[19] S. Suhandinata, R. A. Rizal, D. O. Wijaya, P. Warren, and S. Srinjiwi, "ANALISIS PERFORMA KRIPTOGRAFI HYBRID ALGORITMA BLOWFISH DAN ALGORITMA RSA," *JURTEKSI (Jurnal Teknol. dan Sist. Informasi)*, vol. 6, no. 1, pp. 1–10, 2019, doi: 10.33330/jurteksi.v6i1.395.

[20] J. Jamaludin and R. Romindo, "Rancang Bangun Pengamanan Teks Menggunakan Kombinasi Vigenere Cipher dan RSA dalam Hybrid Cryptosystem," *Pros. Semin. Nas. Ris. ...*, vol. 2, pp. 105–116, 2020, [Online]. Available: http://tunasbangsa.ac.id/seminar/index.php/senaris/article/view/150/0.

[21] K. Mohamed, "Analyse On Avalanche Effect In Cryptography Algorithm," in *Proceedings of the International Conference on Sustainable Practices, Development and Urbanisation (IConsPADU 2021), 16 November 2021, Universiti Selangor (UNISEL), Malaysia*, Oct. 2022, vol. 3, pp. 610–618, doi: 10.15405/epms.2022.10.57.