Jurnal Quancom
Vol.2, No.2, Desember 2024, pp. 23~30
e-ISSN: 2988-0998
P-ISSN: 2988-1005

Perbandingan *Performance* Kriptografi RSA, RSA-CRT, Rabin dalam Proses Pengamanan Pesan Berbasis Teks

Deby Cynthia Rohara Manalu [1]*, Mutiara Enjelina [2], Johannes Bastian Jasa Sipayung [3]
[1*23] Sarjana Terapan Teknologi Rekayasa Perangkat Lunak, Institut Teknologi Del
debymanalu2607@gmail.com *

Article Info

Article history:

Received 5 Desember 2024
Revised 17 Desember 2024
Accepted 23 Desember 2024

Keyword:

RSA, RSA-CRT, Rabin, Cryptography, Algorithm

ABSTRACT

Data security is a critical aspect in the digital era, especially in securing text-based messages. This research compares the performance of three asymmetric cryptography algorithms-RSA, RSA-CRT, and Rabin-in terms of the speed and efficiency of the encryption and decryption process. The RSA (Rivest-Shamir-Adleman) algorithm, one of the most popular cryptographic methods, uses two different keys (public and private) to encrypt and decrypt data, but has a weakness in decryption time efficiency on large data. RSA-CRT, as a modification of RSA with the Chinese Remainder Theorem approach, speeds up the decryption process by dividing large operations into smaller ones. Rabin's algorithm, which is based on prime factorization, shows high time efficiency for both encryption and decryption processes, although it produces four possible plaintexts that require additional selection. All three algorithms were tested using text data with varying character sizes, ranging from 100 to 10,000 characters. The test results show that RSA has the fastest encryption time, RSA-CRT excels in decryption, and Rabin offers the best overall efficiency for securing text-based data. This research provides algorithm recommendations according to system needs, taking into account the balance between efficiency and security.

.

PENDAHULUAN

Keamanan informasi menjadi aspek krusial dalam era digital, di mana informasi tidak hanya berperan sebagai aset strategis, tetapi juga sebagai fondasi kepercayaan dan keberlanjutan bisnis. Keamanan informasi mencakup tiga pilar utama, yaitu kerahasiaan (confidentiality), integritas (integrity), dan ketersediaan (availability). Ketiga aspek ini berfungsi untuk melindungi informasi dari ancaman, baik yang disengaja maupun tidak disengaja [1]. Maka mengatasi tantangan ini, diperlukan penerapan metode kriptografi yang andal dalam menjaga keamanan pesan berbasis teks.

kriptografi telah menjadi salah satu metode utama dalam melindungi data. Dengan akar kata dari bahasa Yunani, "crypto" yang berarti rahasia, dan "graphia" yang berarti tulisan, kriptografi merupakan ilmu dan seni untuk memastikan keamanan pesan. Prinsip dasarnya melibatkan proses enkripsi untuk mengacak informasi sehingga tidak dapat dibaca oleh pihak yang tidak memiliki otoritas, serta proses dekripsi untuk mengembalikan pesan ke bentuk aslinya [2]. Ketiga fungsi utama kriptografi, yaitu enkripsi, dekripsi, dan kunci, berperan penting dalam menciptakan sistem pengamanan yang andal untuk mencegah kebocoran dan penyalahgunaan informasi. Kriptografi asimetris merupakan salah satu metode yang berkembang dalam kriptografi modern untuk meningkatkan keamanan data. Berbeda dengan kriptografi simetris yang menggunakan satu kunci yang sama untuk proses enkripsi dan dekripsi, kriptografi asimetris memanfaatkan dua kunci yang berbeda, yaitu kunci publik untuk enkripsi dan kunci privat untuk dekripsi. Pemisahan peran antara kunci publik dan privat ini membuat kriptografi asimetris lebih aman karena kunci privat tidak perlu didistribusikan, sehingga risiko kebocoran kunci dapat diminimalkan [3]. Kriptografi asimetris mencakup

berbagai algoritma yang dirancang untuk memanfaatkan pasangan kunci publik dan privat. Beberapa algoritma terkenal dalam kriptografi asimetris antara lain RSA (Rivest-Shamir-Adleman), RSA-CRT (Chinese Remainder Theorem), dan Rabin.

Algoritma RSA (Rivest-Shamir-Adleman) merupakan salah satu metode kriptografi yang paling banyak digunakan dalam pengamanan data digital. Kelebihan utama dari algoritma ini terletak pada kemampuannya untuk menyediakan keamanan tinggi melalui penggunaan dua kunci yang berbeda, yaitu kunci publik dan kunci privat. Keamanan RSA berasal dari kesulitan dalam memfaktorkan bilangan besar menjadi faktor prima, sehingga membuatnya sulit untuk diretas [4]. Selain itu, algoritma ini juga dapat diterapkan pada berbagai jenis data, termasuk teks dan citra digital, sehingga menawarkan fleksibilitas dalam penggunaannya di berbagai aplikasi, seperti pengamanan password dan enkripsi data citra [5]. Dengan demikian, RSA menjadi pilihan yang tepat untuk sistem informasi yang memerlukan tingkat keamanan yang tinggi.

Salah satu algoritma kriptografi yang banyak digunakan adalah RSA (Rivest Shamir Adleman), yang dikenal dengan keamanannya dalam melindungi data melalui sistem kunci publik. Namun, RSA juga memiliki kelemahan dalam hal efisiensi waktu, terutama ketika dihadapkan pada data berukuran besar. Untuk mengatasi masalah ini, RSA-CRT (RSA dengan Chinese Remainder Theorem) diperkenalkan sebagai modifikasi yang signifikan. Algoritma RSA-CRT tidak hanya menjaga keamanan data dengan tingkat yang sama, tetapi juga meningkatkan efisiensi proses dekripsi secara signifikan, sehingga mengurangi waktu pemrosesan. Penelitian oleh Sulistiyorini dan Prihanto menunjukkan bahwa RSA-CRT dapat mempercepat proses dekripsi hingga 50% dibandingkan dengan RSA standar [6]. Selain itu, penelitian oleh Sutrimo et al. menekankan bahwa RSA-CRT mampu melindungi data penjualan yang sensitif, seperti file .csv, dari penyalahgunaan, memastikan integritas dan kerahasiaan informasi [7]. Dengan demikian, RSA-CRT menjadi solusi yang lebih efektif untuk aplikasi yang memerlukan keamanan tinggi dan efisiensi dalam pengolahan data.

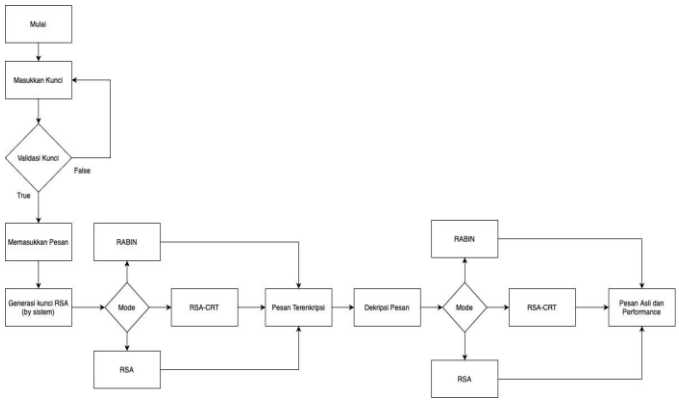
Kriptografi Rabin adalah varian dari algoritma RSA yang menggunakan pendekatan unik dalam enkripsi dan dekripsi data. Kriptografi Rabin menawarkan tingkat keamanan yang tinggi dengan mengandalkan kompleksitas faktorisasi bilangan besar, mirip dengan RSA. Penggunaan Teorema CRT (Chinese Remainder Theorem) dalam proses dekripsi meningkatkan efisiensi perhitungan, menjadikannya lebih cepat dan cocok untuk perangkat dengan sumber daya terbatas, seperti smart card. Selain itu, Rabin juga unggul dalam menghadapi serangan pasif seperti factorization attack, sehingga memberikan lapisan perlindungan tambahan terhadap ancaman tertentu [8]. Metode kriptografi Rabin efektif untuk menjaga kerahasiaan data, seperti pada penyandian file audio. Dengan menggunakan kunci publik, file audio terenkripsi dan tidak dapat diputar, sementara kunci privat memungkinkan dekripsi dan mengembalikannya ke bentuk semula. Metode ini memberikan keamanan tinggi untuk melindungi data sensitif [9].

Berdasarkan penelitian terdahulu yang telah dijelaskan, penelitian ini bertujuan untuk membandingkan kinerja tiga algoritma yaitu kriptografi RSA, RSA-CRT, dan Rabin dalam pengamanan pesan berbasis teks dengan mengevaluasi algoritma mana yang paling efektif dalam menjaga keamanan informasi.

METODE

Penelitian ini merupakan penelitian kuantitatif dengan pendekatan deskriptif. Penelitian kuantitatif merupakan penelitian yang berpangkal dari pola pikir deduktif, yang menggunakan data numerik untuk menguji hipotesis dan menganalisis hubungan antar variabel dengan metode statistik [10].

Pada metode ini terdapat rancangan sistem. Rancangan sistem ini bertujuan untuk mengembangkan aplikasi kriptografi berbasis web yang menggunakan algoritma RSA dan optimisasi Teorema CRT untuk meningkatkan efisiensi proses dekripsi. Flowchart aplikasi kriptografi berbasis web ini menggambarkan alur proses enkripsi dan dekripsi pesan dengan jelas dan terstruktur. Flowchart sangat dibutuhkan dalam penggunaannya karena digunakan sebagai bukti dokumentasi untuk menjelaskan gambaran logis sebuah sistem yang akan dibangun kepada programmer [11]. Pengguna memulai dengan memilih opsi enkripsi atau dekripsi, kemudian memasukkan dua bilangan prima p dan q untuk menghasilkan kunci RSA. Setelah itu, pengguna dapat memasukkan pesan yang ingin dienkripsi, dan aplikasi akan menampilkan hasil enkripsi. Untuk dekripsi, pengguna memasukkan pesan terenkripsi, dan hasil dekripsi akan ditampilkan, memungkinkan pengguna untuk melihat kembali pesan asli. Desain flowchart ini memastikan setiap langkah mudah diikuti, memberikan pengalaman pengguna yang intuitif dan efisien dalam menjaga keamanan data.



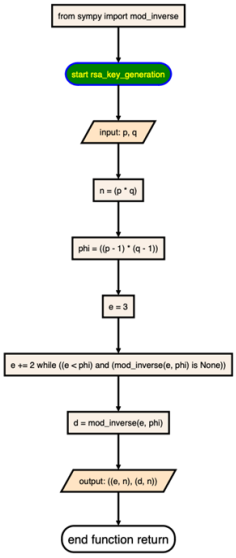
Gambar 1. Rancangan sistem secara umum

Pada Gambar 1 menggambarkan alur proses aplikasi kriptografi yang dirancang. Proses dimulai dengan input "Mulai", lalu pengguna menginputkan kunci Selanjutnya, aplikasi akan memvalidasi apakah kunci valid atau tidak. Jika tidak valid, maka akan menampilkan "False", sedangkan jika valid, maka akan menampilkan "True" dan melanjutkan ke proses selanjutnya. Setelah itu, aplikasi akan melakukan proses enkripsi pesan menggunakan algoritma Rabin. Kemudian, aplikasi akan memberikan opsi mode, yaitu RSA, RSA-CRT, dan Rabin. Pengguna dapat memilih mode yang diinginkan, dan aplikasi akan memproses pesan sesuai dengan mode yang dipilih. Setelah proses enkripsi selesai, aplikasi akan menampilkan pesan yang telah dienkripsi dan memberikan pilihan untuk melakukan dekripsi pesan.

A. Algoritma RSA

Pada algortima RSA terdapat 3 langkah yaitu pembangkitan kunci, enkripsi, dan dekripsi [12].

Pembangkitan Kunci: Langkah awal yang dapat dilakukan adalah membangkitkan kunci publik dan privat dengan memilih dua bilangan prima besar, menghitung hasil perkaliannya, dan menentukan eksponen yang sesuai untuk proses enkripsi dan dekripsi.



Gambar 2. Flowchart rsa

Pada Gambar 2. menggambarkan proses pembangkitan kunci dan enkripsi RSA. Flowchart ini menggambarkan proses pembangkitan kunci RSA. Proses dimulai dengan memanggil fungsi "start_rsa_key_generation" yang membutuhkan input nilai p dan q. Selanjutnya, sistem akan menghitung nilai yang diperlukan untuk membangkitkan kunci publik dan kunci privat. Setelah itu, sistem akan mencari nilai e yang relatif prima dengan phi (nilai yang dihitung sebelumnya) dan nilai d yang merupakan invers modulo dari e. Pada akhir proses, fungsi akan mengembalikan output berupa sepasang kunci, yaitu kunci publik (e, n) dan kunci privat (d, n), di mana n merupakan hasil perkalian p dan q. Flowchart ini menggambarkan langkah-langkah utama dalam pembangkitan kunci RSA, yang merupakan salah satu algoritma kriptografi kunci publik yang penting.

Enkripsi: Setelah tahap pembangkitan kunci selesai dilakukan, langkah selanjutnya adalah proses enkripsi menggunakan kunci publik yang telah dihasilkan, sesuai dengan rumus berikut:

$$C = P^e \bmod n \quad (1)$$

Dekripsi: Dekripsi merupakan langkah untuk mengembalikan ciphertext menjadi plaintext asli. Dalam algoritma RSA, proses ini dilakukan menggunakan persamaan berikut:

$$P = C^d \bmod n \quad (2)$$

B. Algoritma RSA-CRT

Dalam algortima RSA-CRT juga melalui 3 langkah yaitu pembangkitan kunci, enkripsi, dan dekripsi [6].

Pembangkitan Kunci: Langkah awal yang dapat dilakukan adalah membangkitkan kunci publik dan privat dengan memilih dua bilangan prima besar, lalu menghitung komponen tambahan seperti d_p_dan d_q untuk mempercepat proses dekripsi.

Gambar 3. Flowchart rsa-crt

Pada Gambar 3. Flowchart ini menggambarkan sebuah fungsi yang melakukan proses pembangkitan kunci RSA-CRT. Proses ini dimulai dengan menerima input berupa dua bilangan prima, p dan q. Selanjutnya, fungsi ini akan menghitung nilai n, yang merupakan hasil perkalian p dan q. Nilai n ini akan digunakan sebagai modulus untuk kunci publik dan kunci privat. Setelah itu, fungsi akan mencari nilai e, yang merupakan kunci publik. Nilai e harus relatif prima dengan nilai yang dihitung sebelumnya, dan lebih kecil dari nilai tersebut. Setelah mendapatkan nilai e, fungsi akan menghitung nilai d, yang merupakan kunci privat. Nilai d dihitung menggunakan fungsi mod_inverse, yang akan

memberikan nilai invers multiplikatif dari e modulo nilai yang dihitung sebelumnya. Akhirnya, fungsi ini akan mengembalikan output berupa pasangan kunci publik (e, n) dan pasangan kunci privat (d, n), serta nilai p dan q.

Enkripsi: Setelah kunci berhasil dihasilkan, tahap berikutnya adalah melakukan enkripsi dengan memanfaatkan kunci publik yang telah dibuat sebelumnya. Proses ini dilakukan berdasarkan persamaan berikut:

$$C = P^e \bmod n \quad (3)$$

Dekripsi: RSA-CRT memanfaatkan kunci d untuk menghasilkan nilai $d_p$dan d_q, di mana kedua nilai tersebut lebih kecil dari $d$karena diperoleh melalui operasi modulus d terhadap p dan q. Perhitungan $d_p$dan d_q dilakukan menggunakan persamaan berikut.

$$d \bmod (p-1) = e - 1 \bmod (p-1) \quad (4)$$
$$d \bmod (q-1) = e - 1 \bmod (q-1) \quad (5)$$
$$d_p = e - 1 \bmod (p-1) = d \bmod (p-1) \quad (6)$$
$$d_q = e - 1 \bmod (q-1) = d \bmod (q-1) \quad (7)$$

Dari perhitungan sebelumnya, diperoleh nilai d_p dan d_q yang lebih kecil dibandingkan d. Selanjutnya, nilai representasi pesan m_1 dan m_2 akan dihitung sebagai bagian dari proses akhir dekripsi menggunakan persamaan berikut:

$$m_1 = c^{dp} \bmod p \quad (8)$$
$$m_2 = c^{dq} \bmod p \quad (9)$$

Nilai m_1 dan m_2 yang telah diperoleh akan dimasukkan ke dalam persamaan Garner (6) untuk menentukan hasil akhir dekripsi, seperti pada langkah berikut:

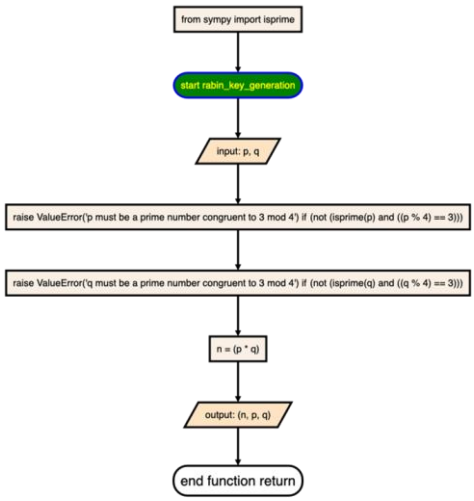
$$qlnv = \left(\frac{1}{q}\right) \bmod p = 1 + \frac{kp}{q} \quad (10)$$

$$h = qlnv(m_1 - m_2) \bmod p \quad (11)$$
$$m = m_2 + h.q \quad (12)$$

C. Algoritma Rabin

Pada algoritma Rabin juga melalui 3 langkah yaitu pembangkitan kunci, enkripsi, dan dekripsi [13].

Pembangkitan Kunci: Langkah awal yang dapat dilakukan adalah membangkitkan kunci publik dan privat dengan memilih dua bilangan prima besar yang memenuhi syarat khusus untuk proses enkripsi dan dekripsi.



Gambar 4. Flowchart rabin

Pada Gambar 4 menggambarkan proses pembuatan kunci Rabin. Proses ini dimulai dengan memanggil fungsi rabin_key_generation yang membutuhkan dua input, yaitu p dan q. Fungsi ini akan memeriksa apakah p dan q merupakan bilangan prima yang kongruen dengan 3 modulo 4. Jika tidak, maka akan muncul pesan kesalahan yang memberitahukan bahwa p dan q harus memenuhi kondisi tersebut. Jika kondisi terpenuhi, maka akan dihitung nilai n yang merupakan hasil perkalian antara p dan q. Setelah nilai n dihitung, fungsi akan mengembalikan output berupa tuple yang terdiri dari n, p, dan q. Kemudian, fungsi akan selesai dan mengembalikan nilai tersebut. Flowchart ini menunjukkan bahwa proses pembuatan kunci Rabin membutuhkan dua bilangan prima yang kongruen dengan 3 modulo 4, dan menghasilkan nilai n yang akan digunakan dalam proses enkripsi dan dekripsi pesan.

Enkripsi: Proses enkripsi pada teknik Rabin dapat dirumuskan sebagai berikut:

$$C = P^2 \bmod n \quad (13)$$

C : Cipherteks

P : Plainteks

n : kunci publik

Dekripsi: Proses Dekripsi pada teknik Rabin dilakukan dengan menggunakan teorema Chinese remainder. Teorema ini digunakan untuk mendapatkan teks asli yang benar. Rumus umum yang digunakan sebagai berikut:

$$x \equiv m_1 \,(mod\ p) \quad (14)$$

$$x \equiv m_2 \,(mod\ q) \quad (15)$$

Dengan langkah-langkah:

$$m_1 = C^{(p+1)/4} \,(mod\ p) \quad (16)$$

$$m_2 = C^{(q+1)/4} \,(mod\ q) \quad (17)$$

Hasil dekripsi berupa empat kemungkinan nilai, dan algoritma Rabin akan memilih salah satu nilai tersebut sebagai plaintext yang benar berdasarkan konteks pesan

HASIL DAN PEMBAHASAN

Dalam pengembangan ini menggunakan bahasa pemrograman Python dikarenakan pada kemampuannya dalam menangani operasi matematika yang kompleks serta dukungan pustaka kriptografi yang lengkap, yang memudahkan dalam mengembangkan solusi pengamanan pesan.

Pengembangan ini menggunakan Python versi 3.12.2, bahasa pemrograman Python telah banyak digunakan karena sifatnya yang open source, yang memungkinkan siapa saja untuk memanfaatkan dan mengembangkan bahasa ini [14]. Ini merupakan Python versi terbaru sehingga banyak pustaka yang digunakan tentunya. Dalam pengembangan ini Flask adalah framework web yang digunakan karena fleksibel. Hal tersebut yang memungkinkan pengembangan aplikasi dengan mudah tanpa ketergantungan pihak ketiga, serta menyediakan layanan seperti server HTTP bawaan dan dukungan untuk pengujian unit [15].

Pengembangan ini memanfaatkan pustaka Python seperti time untuk mengukur waktu eksekusi algoritma, random untuk menghasilkan nilai acak dalam pembuatan kunci, mod_inverse untuk menghitung invers modular pada algoritma seperti RSA dan Rabin, serta io untuk menangani input-output data, memastikan aplikasi berfungsi optimal saat berinteraksi dengan sistem eksternal.

A. Data Penelitian

Penelitian ini menggunakan data teks untuk proses enkripsi dan dekripsi dengan fokus pada tiga algoritma kriptografi, yaitu RSA, RSA-CRT, dan Rabin. Perangkat lunak yang dikembangkan memfasilitasi enkripsi dan dekripsi menggunakan variasi ukuran karakter, seperti 100, 200, 500, 1000, dan 10000. Penelitian ini bertujuan untuk menganalisis performansi masing-masing algoritma dalam mengamankan pesan berbasis teks.

B. Desain dan Hasil Tampilan

Penelitian ini menyajikan desain dan hasil akhir dari tampilan website yang telah dikembangkan. Tampilan mencerminkan implementasi dari desain tersebut, yang bertujuan untuk memenuhi kebutuhan pengguna dan tujuan proyek secara keseluruhan.

Tampilan Informasi Algoritma: Aplikasi Kriptografi ini dirancang sebagai sistem berbasis web menggunakan bahasa pemrograman Python. Berikut hasil dan pembahasan dari penelitian ini. Berikut di bawah ini menunjukkan tampilan awal dari sistem yang telah dibangun.

Gambar 5. Tampilan informasi algoritma rsa

Tampilan awal aplikasi pada Gambar 5 menunjukkan antarmuka yang intuitif, memungkinkan pengguna memilih menu enkripsi atau dekripsi. Pada proses enkripsi, pengguna memasukkan pesan dan kunci publik untuk menghasilkan hasil enkripsi yang dapat disimpan atau dikirim. Aplikasi juga menyediakan menu RSA, RSA-CRT, dan Rabin, dengan page RSA berisi informasi tentang algoritma RSA.

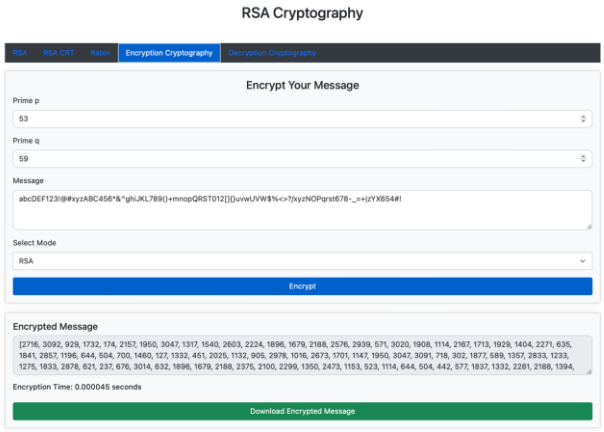
Gambar 6. Tampilan informasi rsa-crt

Pada Gambar 6 menampilkan antarmuka aplikasi web yang menggunakan algoritma RSA dengan optimisasi CRT (Chinese Remainder Theorem) untuk mempercepat dekripsi. Halaman ini menjelaskan langkah-langkah penggunaan RSA CRT dan manfaatnya dalam meningkatkan efisiensi dengan membagi operasi besar menjadi lebih kecil dan cepat, mendukung keamanan data dalam aplikasi kriptografi berbasis web.

Gambar 7. Tampilan informasi rabin

Pada Gambar 7 menampilkan antarmuka web yang menjelaskan algoritma kriptografi Rabin, metode asimetris berbasis faktorisasi bilangan besar. Halaman ini menguraikan langkah-langkah enkripsi Rabin Cryptosystem, menyoroti keunikan dan kompleksitasnya dalam mengamankan data secara efisien.

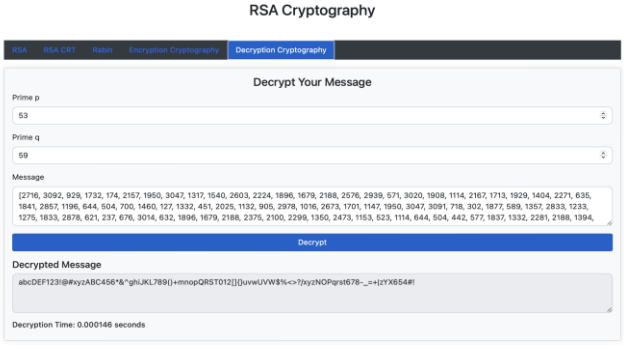
Tampilan Enkripsi: Enkripsi merupakan salah satu teknik utama dalam kriptografi yang bertujuan untuk mengamankan pesan dengan cara mengubahnya menjadi format yang tidak dapat dibaca oleh pihak yang tidak berwenang. Sub-bab ini membahas implementasi proses enkripsi menggunakan algoritma RSA, meliputi antarmuka, parameter yang diperlukan, dan hasil yang dihasilkan oleh aplikasi.



Gambar 8. Tampilan enkripsi kriptografi

Bagian Encryption Cryptography menyediakan kolom input untuk memasukkan parameter enkripsi, termasuk bilangan prima p dan q, pesan teks, serta menu dropdown "Select Mode" yang disetel ke "RSA". Tombol biru "Encrypt" digunakan untuk memproses enkripsi, dan hasilnya ditampilkan di bagian "Encrypted Message" dalam bentuk angka, disertai waktu proses, seperti 0.00045 detik. Terdapat juga tombol hijau "Download Encrypted Message" untuk mengunduh hasil enkripsi. Desain antarmuka yang sederhana dan minimalis memudahkan pengguna fokus pada parameter utama, cocok untuk mereka yang memahami dasar-dasar kriptografi RSA.

Tampilan Dekripsi: Dekripsi mengembalikan pesan terenkripsi ke bentuk aslinya agar dapat dipahami oleh pihak berwenang. Sub-bab ini membahas implementasi dekripsi RSA, meliputi antarmuka, input parameter, dan hasil dekripsi yang ditampilkan aplikasi.



Gambar 9. Tampilan dekripsi kriptografi

Antarmuka ini menyediakan formulir dengan kolom input untuk bilangan prima p dan q (contoh: 53 dan 59) serta pesan terenkripsi. Pengguna dapat menekan tombol biru "Decrypt" untuk memulai proses dekripsi. Hasilnya ditampilkan di kolom "Decrypted Message", seperti "Keamanan Perangkat Lunak". Desain minimalis dan intuitif ini memudahkan pengguna mengisi parameter utama, ditujukan untuk mereka yang memahami dasar kriptografi RSA.

C. Hasil Pengujian

Dalam proses mengenkripsi pesan, digunakan beberapa mode algoritma kriptografi yang saling berhubungan, yaitu RSA, RSA-CRT, dan Rabin. Masing-masing algoritma ini memiliki cara kerja yang berbeda dalam mengamankan pesan, namun tetap berada dalam lingkup metode kriptografi berbasis kunci publik. RSA merupakan algoritma dasar yang banyak digunakan, sementara RSA-CRT menawarkan optimasi untuk meningkatkan kecepatan, dan Rabin mengandalkan konsep matematis yang sedikit berbeda namun tetap efektif dalam hal keamanan, berikut ini adalah tabel performansi dari setiap algoritma.

TABEL I
PERFORMANSI ENKRIPSI RSA

Algoritma	Karakter	Waktu Eksekusi	Rata-rata waktu
RSA	100	0.000054	0.0009684
	200	0.000105	
	500	0.000125	
	1000	0.000638	
	10000	0.003920	

TABEL II
PERFORMANSI DEKRIPSI RSA

Algoritma	Karakter	Waktu Eksekusi	Rata-rata waktu
RSA	100	0.000148	0.0047552
	200	0.000292	
	500	0.000900	
	1000	0.002676	
	10000	0.019760	

TABEL III
PERFORMANSI ENKRIPSI RSA-CRT

Algoritma	Karakter	Waktu Eksekusi	Rata-rata waktu
RSA	100	0.000049	0.0010434
	200	0.000066	
	500	0.000145	
	1000	0.000642	
	10000	0.004315	

TABEL IV
PERFORMANSI DEKRIPSI RSA-CRT

Algoritma	Karakter	Waktu Eksekusi	Rata-rata waktu
RSA	100	0.000202	0.003403
	200	0.000235	
	500	0.000941	
	1000	0.002621	
	10000	0.014006	

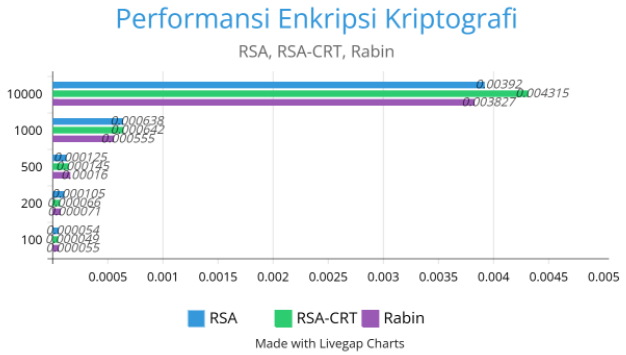
TABEL V
PERFORMANSI ENKRIPSI RABIN

Algoritma	Karakter	Waktu Eksekusi	Rata-rata waktu
RSA	100	0.000055	0.000934
	200	0.000071	
	500	0.000160	
	1000	0.000555	
	10000	0.003827	

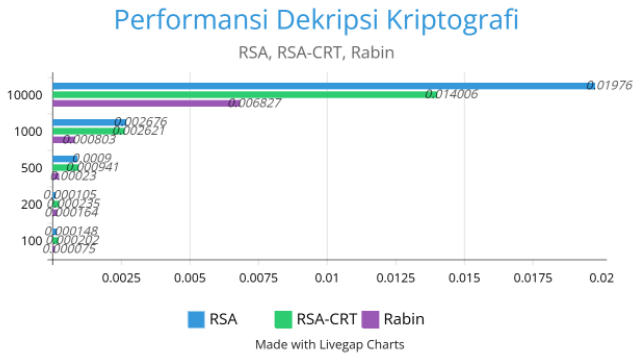
TABEL VI
PERFORMANSI DEKRIPSI RABIN

Algoritma	Karakter	Waktu Eksekusi	Rata-rata waktu
RSA	100	0.000075	0.00159028
	200	0.000164	
	500	0.000230	
	1000	0.000803	
	10000	0.006827	

Untuk lebih mempermudah mengetahui informasi perbandingan dari RSA, RSA-CRT dan Rabin dapat dilihat dari visualisasi berikut.



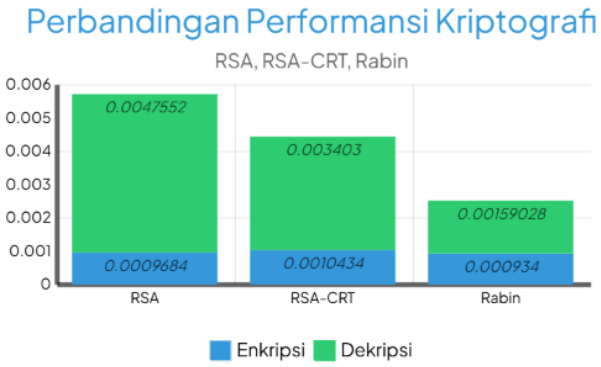
Gambar 10. Performansi enkripsi kriptografi



Gambar 11. Performasi deksripsi kriptografi

Berdasarkan data yang disajikan dalam tabel, dapat dilihat bahwa algoritma RSA memiliki waktu enkripsi yang lebih cepat dibandingkan RSA-CRT dan Rabin, terutama untuk kunci yang lebih pendek (100-1000 karakter). Algoritma RSA memiliki waktu enkripsi yang paling cepat di antara ketiga algoritma tersebut. Di sisi lain, algoritma Rabin memiliki waktu enkripsi yang paling lama di antara ketiganya.

Untuk proses dekripsi, algoritma RSA-CRT memiliki waktu dekripsi yang lebih cepat dibandingkan RSA dan Rabin, terutama untuk kunci yang lebih panjang (1000-10000 karakter). Algoritma RSA-CRT memiliki kinerja yang lebih baik untuk dekripsi pesan yang menggunakan kunci yang lebih panjang. Sementara itu, algoritma Rabin memiliki waktu dekripsi yang paling lama di antara ketiga algoritma. Secara keseluruhan, algoritma RSA dapat dianggap sebagai yang terbaik di antara ketiga algoritma tersebut, karena memiliki waktu enkripsi dan dekripsi yang paling cepat.



Gambar 12. Perbandingan performansi kriptografi secara keseluruhan

Hasil pengujian menunjukkan bahwa algoritma Rabin memiliki waktu enkripsi yang paling cepat, yaitu 0.000934 detik, meskipun waktu dekripsi sedikit lebih tinggi dibandingkan RSA-CRT, yaitu 0.00159028 detik. Algoritma RSA membutuhkan waktu enkripsi 0.0009684 detik, namun dekripsi lebih lambat, yakni 0.0047552 detik. Sementara itu, RSA-CRT menunjukkan performa yang lebih baik dalam proses dekripsi dengan waktu 0.003403 detik, meskipun waktu enkripsinya sedikit lebih lama dibandingkan RSA, yaitu 0.0010434 detik.

Secara keseluruhan, pemilihan algoritma tergantung pada kebutuhan sistem. Jika prioritasnya adalah kecepatan enkripsi, Rabin adalah pilihan terbaik. Namun, jika fokus pada kecepatan dekripsi, RSA-CRT lebih efisien. RSA tetap relevan, meskipun membutuhkan waktu lebih lama untuk proses dekripsi. Hasil ini menunjukkan trade-off antara waktu enkripsi dan dekripsi yang perlu dipertimbangkan dalam merancang sistem pengamanan pesan berbasis teks.

SIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa meskipun algoritma RSA-CRT menunjukkan peningkatan kecepatan dibandingkan dengan algoritma RSA, algoritma Rabin tetap lebih unggul dalam hal efisiensi waktu eksekusi enkripsi dan dekripsi. Tabel menunjukkan bahwa Rabin memiliki waktu eksekusi yang lebih singkat untuk semua jumlah karakter yang diuji, meskipun RSA-CRT memberikan performa yang lebih baik dibandingkan RSA. Kenaikan waktu eksekusi seiring bertambahnya jumlah karakter lebih signifikan pada RSA dan RSA-CRT dibandingkan Rabin. Oleh karena itu, untuk aplikasi yang memerlukan efisiensi tinggi, algoritma Rabin

lebih direkomendasikan. Namun, pemilihan algoritma tetap harus mempertimbangkan faktor keamanan dan kebutuhan spesifik dari aplikasi, mengingat bahwa RSA-CRT juga memiliki keunggulan dalam beberapa aspek.

UCAPAN TERIMA KASIH

Ucapan terima kasih yang sebesar-besarnya kami sampaikan kepada Tuhan Yang Maha Esa atas berkat dan penyertaan-Nya sehingga penelitian ini dapat diselesaikan dengan baik. Kami juga menyampaikan terima kasih kepada dosen mata kuliah Keamanan Perangkat Lunak yang telah memberikan ilmu dan wawasan yang menjadi dasar bagi terlaksananya penelitian ini. Tak lupa, terima kasih kepada teman-teman dan rekan sejawat atas dukungan, ide, serta kerja sama yang membantu kelancaran penelitian ini. Semoga penelitian ini dapat memberikan manfaat bagi semua pihak.

DAFTAR PUSTAKA

[1] A. H. Harahap, C. Difa Andani, A. Christie, D. Nurhaliza, and A. Fauzi, "Pentingnya Peranan CIA Triad Dalam Keamanan Informasi dan Data Untuk Pemangku Kepentingan atau Stakholder," J. Manaj. dan Pemasar. Digit., vol. 1, no. 2, pp. 73–83, 2023.

[2] M. Sari, H. D. Purnomo, and I. Sembiring, "Review : Algoritma Kriptografi Sistem Keamanan SMS di Android," J. Inf. Technol., vol. 2, no. 1, pp. 11–15, 2022, doi: 10.46229/jifotech.v2i1.292.

[3] R. Siringoringo, "Analisis dan Implementasi Algoritma Rijndael (AES) dan Kriptografi RSA pada Pengamanan File," KAKIFIKOM (Kumpulan Artik. Karya Ilm. Fak. Ilmu Komputer), vol. 02, no. 01, pp. 31–42, 2020, doi: 10.54367/kakifikom.v2i1.666.

[4] Z. Z. Deskiva, P. Studi, T. Informatika, C. Digital, P. Password, and P. Aplikasi, "Implementasi Kriptografi Modern Dengan Metode," pp. 44–49, 2014.

[5] M. S. Dairi, M. Setiani Asih, and corespondent author, "Implementasi Algoritma Kriptografi RSA Dalam Aplikasi Sistem Informasi Perpustakaaan Implementation Of RSA Cryptographic Algorithms in Library Information System Applications," Januari, vol. 2023, no. 2, pp. 214–223, 2022, [Online]. Available: https://jurnal.unity-academy.sch.id/index.php/jirsi/index%0Ahttp://creativecommons.org/licenses/by-sa/4.0/

[6] S. Sulistiyorini and A. Prihanto, "Perbandingan Efisiensi Algoritma RSA dan RSA-CRT Dengan Data Teks Berukuran Besar," J. Informatics Comput. Sci., vol. 1, no. 02, pp. 84–90, 2020, doi: 10.26740/jinacs.v1n02.p84-90.

[7] I. Zulkarnain and R. Mahyuni, "Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD Pengamanan Data Penjualan Menggunakan Algoritma RSA-CRT," Volume, vol. 7, no. 2, pp. 176–183, 2024, [Online]. Available: https://ojs.trigunadharma.ac.id/index.php/jsk/index

[8] A. Widyarnako, "Teknik Kriptografi Rabin, Serangan yang Dapat Dilakukan dan Perbandingannya dengan RSA.," Inst. Teknol. Bandung, vol. 2, 2008.

[9] R. Rusdianto, N. Silalahi, and N. Sitohang, "Penerapan Algoritma Rabin-Public Key Untuk Pengamanan File Audio," Bull. Artif. Intell., vol. 2, no. 1, pp. 100–103, 2023, doi: 10.62866/buai.v2i1.45.

[10] R. Handayani, Metode Penelitian Sosial, no. September. 2020.

[11] R. Rosaly and A. Prasetyo, "Flowchart Beserta Fungsi dan Simbol-Simbol," J. Chem. Inf. Model., vol. 2, no. 3, pp. 5–7, 2020.

[12] Yusfrizal, "Rancang Bangun Aplikasi Kriptografi Pada Teks Menggunakan Metode Reverse Chiper Dan Rsa Berbasis Android," JTIK (Jurnal Tek. Inform. Kaputama), vol. 3, no. 2, pp. 29–37, 2019, [Online]. Available: http://jurnal.kaputama.ac.id/index.php/JTIK/article/view/173

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

[14] R. M. Raihan and S. Yulianto, "Penerapan Pemrograman Python Dalam Menentukan Waktu Overhoul Kondensor Turbin Uap," J. Konversi Energi dan Manufaktur, vol. 8, no. 1, pp. 49–57, 2023, doi: 10.21009/jkem.8.1.6.

[15] N. Idris, C. F. Mohd Foozy, and P. Shamala, "A Generic Review of Web Technology: DJango and Flask," Int. J. Adv. Sci. Comput. Eng., vol. 2, no. 1, pp. 34–40, 2021, doi: 10.62527/ijasce.2.1.29.