DOI: https://doi.org/10.37817/ikraith-informatika.v9i3
P-ISSN : 2580-4316
E-ISSN : 2654-8054

Pengembangan RSA Web Cryptosystem Berbasis Python untuk Pengamanan Data dan Evaluasi Kinerja Kriptografi

Yulita Laurentz[1], Nizirwan Anwar[2] , Alivia Yulfitri[3]

[1,2] Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Esa Unggul , [3]Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Esa Unggul

Surel yulitalrntz@gmail.com, nizirwan.anwar@esaunggul.ac.id , alivia@esaunggul.ac.id

Abstrak

Perkembangan teknologi informasi dan komunikasi menuntut adanya sistem keamanan data yang mampu melindungi kerahasiaan dan integritas informasi dalam proses pertukaran data digital. Salah satu metode yang широко digunakan dalam kriptografi modern adalah algoritma RSA yang termasuk dalam kategori kriptografi kunci publik. Penelitian ini bertujuan untuk mengembangkan dan mengimplementasikan sistem RSA Web Cryptosystem berbasis Python untuk pengamanan pesan teks serta melakukan evaluasi kinerja algoritma kriptografi berdasarkan ukuran kunci yang digunakan. Sistem dikembangkan menggunakan bahasa pemrograman Python dengan framework Flask serta memanfaatkan pustaka kriptografi Python yang mendukung mekanisme enkripsi dan dekripsi menggunakan skema padding OAEP. Aplikasi yang dibangun memungkinkan pengguna melakukan pembangkitan pasangan kunci publik dan kunci privat, proses enkripsi plaintext menjadi ciphertext, serta proses dekripsi ciphertext kembali menjadi plaintext melalui antarmuka berbasis web. Selain itu, sistem juga mencatat waktu proses enkripsi dan dekripsi untuk melakukan analisis performa algoritma RSA. Pengujian dilakukan dengan menggunakan ukuran kunci RSA 1024-bit dan 2048-bit untuk membandingkan waktu komputasi serta ukuran ciphertext yang dihasilkan. Hasil eksperimen menunjukkan bahwa algoritma RSA dapat diimplementasikan dengan baik dalam sistem berbasis web dan mampu melakukan proses enkripsi serta dekripsi secara konsisten. Namun demikian, peningkatan ukuran kunci memberikan tingkat keamanan yang lebih tinggi tetapi juga meningkatkan kompleksitas komputasi dan waktu proses enkripsi maupun dekripsi.

Kata kunci: RSA, kriptografi kunci publik, enkripsi, dekripsi, keamanan data.

1 Pendahuluan

Perkembangan teknologi informasi dan komunikasi telah membawa perubahan signifikan dalam cara manusia menyimpan, mengelola, dan mendistribusikan informasi. Berbagai aktivitas digital seperti komunikasi daring, pertukaran dokumen elektronik, transaksi keuangan, serta layanan berbasis internet semakin meningkat seiring dengan kemajuan infrastruktur jaringan global. Kondisi ini menimbulkan tantangan baru dalam menjaga keamanan informasi, terutama terkait dengan kerahasiaan data, integritas pesan, serta autentikasi pihak yang berkomunikasi. Data yang dikirim melalui jaringan publik berpotensi mengalami penyadapan, manipulasi, maupun akses oleh pihak yang tidak berwenang. Oleh karena itu, diperlukan mekanisme pengamanan yang mampu melindungi informasi dari berbagai ancaman keamanan digital.

Kriptografi merupakan salah satu teknik utama yang digunakan untuk menjaga keamanan informasi dalam sistem komputer dan jaringan. Secara umum, kriptografi bekerja dengan mengubah data asli yang dapat dibaca manusia (plaintext) menjadi bentuk tersandi (ciphertext) sehingga hanya pihak yang memiliki kunci tertentu yang dapat mengembalikannya ke bentuk semula. Penerapan kriptografi telah menjadi fondasi penting dalam berbagai sistem keamanan digital modern, termasuk protokol keamanan internet, sistem autentikasi, serta mekanisme tanda tangan digital (Stallings, 2017). Dalam perkembangan kriptografi modern, metode kriptografi dibedakan menjadi dua kategori utama, yaitu kriptografi simetris dan kriptografi asimetris. Kriptografi simetris menggunakan satu kunci yang sama untuk proses enkripsi dan dekripsi, sedangkan kriptografi asimetris menggunakan pasangan kunci yang berbeda, yaitu kunci publik dan kunci privat.

Salah satu algoritma kriptografi asimetris yang paling dikenal dan banyak digunakan adalah algoritma RSA. Algoritma ini diperkenalkan oleh Rivest, Shamir, dan Adleman pada tahun 1977 dan hingga saat ini masih menjadi salah satu standar dalam sistem keamanan digital. Keamanan algoritma RSA didasarkan pada

P-ISSN : 2580-4316
E-ISSN : 2654-8054 DOI: https://doi.org/10.37817/ikraith-informatika.v9i3

kesulitan komputasional dalam memfaktorkan bilangan bulat besar yang merupakan hasil perkalian dua bilangan prima besar. Proses perkalian dua bilangan prima dapat dilakukan dengan mudah, namun proses kebalikannya, yaitu menemukan faktor prima dari bilangan hasil perkalian tersebut, merupakan masalah matematika yang sangat sulit untuk diselesaikan dalam waktu yang wajar menggunakan kemampuan komputasi saat ini. Karakteristik ini menjadikan RSA sebagai salah satu algoritma kriptografi yang kuat dan banyak digunakan dalam berbagai protokol keamanan seperti SSL/TLS, sistem pertukaran kunci, serta mekanisme tanda tangan digital (Rivest et al., 1978; Menezes et al., 1996).

Meskipun secara teoritis algoritma RSA memiliki tingkat keamanan yang tinggi, implementasi algoritma kriptografi dalam sistem perangkat lunak tetap memerlukan perhatian khusus. Keamanan suatu sistem kriptografi tidak hanya ditentukan oleh kekuatan algoritma yang digunakan, tetapi juga oleh bagaimana algoritma tersebut diimplementasikan dalam perangkat lunak. Kesalahan implementasi dapat menimbulkan kerentanan keamanan meskipun algoritma yang digunakan telah terbukti kuat secara matematis. Oleh karena itu, pengembangan sistem kriptografi perlu memperhatikan aspek implementasi yang benar, penggunaan pustaka kriptografi yang aman, serta evaluasi performa komputasi dari algoritma yang diterapkan (Ferguson et al., 2010).

Dalam konteks pengembangan perangkat lunak modern, bahasa pemrograman Python menjadi salah satu pilihan yang populer untuk implementasi algoritma kriptografi. Python memiliki kemampuan komputasi bilangan besar yang baik serta didukung oleh berbagai pustaka kriptografi yang telah teruji keamanannya. Selain itu, Python juga dapat digunakan untuk membangun aplikasi berbasis web melalui framework ringan seperti Flask yang memungkinkan pengembangan sistem interaktif dengan arsitektur client–server. Dengan memanfaatkan kombinasi Python, Flask, dan pustaka kriptografi yang tersedia, algoritma RSA dapat diimplementasikan dalam bentuk aplikasi berbasis web yang memungkinkan pengguna melakukan pembangkitan kunci, enkripsi pesan, serta dekripsi data secara langsung melalui antarmuka web (Grinberg, 2018).

Berdasarkan latar belakang tersebut, penelitian ini berfokus pada pengembangan sebuah sistem RSA Web Cryptosystem berbasis Python yang mampu melakukan proses pembangkitan kunci publik dan kunci privat, enkripsi pesan teks menjadi ciphertext, serta dekripsi ciphertext kembali menjadi plaintext. Sistem yang dikembangkan memanfaatkan framework Flask sebagai penghubung antara antarmuka pengguna dan modul pemrosesan kriptografi di sisi server. Selain menyediakan fungsi enkripsi dan dekripsi, sistem juga dirancang untuk menampilkan parameter kriptografi serta mencatat waktu proses enkripsi dan dekripsi sebagai bagian dari evaluasi performa algoritma RSA.

Lingkup penelitian ini difokuskan pada implementasi algoritma RSA dalam aplikasi berbasis web menggunakan bahasa pemrograman Python. Proses yang dianalisis meliputi pembangkitan pasangan kunci RSA, proses enkripsi plaintext menjadi ciphertext menggunakan kunci publik, serta proses dekripsi ciphertext kembali menjadi plaintext menggunakan kunci privat. Evaluasi sistem dilakukan dengan membandingkan performa algoritma RSA berdasarkan variasi ukuran kunci, khususnya RSA 1024-bit dan RSA 2048-bit, untuk melihat pengaruh ukuran kunci terhadap waktu komputasi serta ukuran ciphertext yang dihasilkan.

Tujuan utama dari penelitian ini adalah mengimplementasikan algoritma RSA dalam sistem kriptografi berbasis web menggunakan Python serta menganalisis kinerja algoritma tersebut dalam proses enkripsi dan dekripsi data. Selain itu, penelitian ini juga bertujuan untuk memberikan gambaran implementasi praktis kriptografi kunci publik dalam pengamanan data digital serta menunjukkan bagaimana perubahan ukuran kunci mempengaruhi tingkat keamanan dan performa komputasi sistem.

Hasil penelitian ini diharapkan dapat memberikan manfaat baik secara akademik maupun praktis. Dari sisi akademik, penelitian ini dapat memberikan pemahaman mengenai konsep dan implementasi algoritma kriptografi RSA dalam sistem berbasis web. Dari sisi praktis, aplikasi yang dikembangkan dapat digunakan sebagai media pembelajaran dalam memahami mekanisme kerja kriptografi kunci publik serta sebagai prototipe awal bagi pengembangan sistem keamanan data digital berbasis kriptografi di masa mendatang.

DOI: https://doi.org/10.37817/ikraith-informatika.v9i3
P-ISSN : 2580-4316
E-ISSN : 2654-8054

2 Studi Kajian Literasi

Dalam beberapa dekade terakhir, penelitian mengenai kriptografi kunci publik telah berkembang pesat seiring dengan meningkatnya kebutuhan keamanan data dalam sistem informasi modern. Berbagai studi telah membahas implementasi algoritma RSA dalam berbagai konteks aplikasi, mulai dari sistem keamanan jaringan, pertukaran kunci dalam protokol komunikasi, hingga mekanisme autentikasi digital. Penelitian yang dilakukan oleh Rivest, Shamir, dan Adleman (1978) memperkenalkan konsep kriptografi kunci publik RSA yang menjadi salah satu fondasi utama dalam sistem keamanan digital. Selanjutnya, berbagai penelitian lanjutan mengembangkan metode implementasi RSA yang lebih efisien dan aman melalui optimasi komputasi modular serta penggunaan skema padding kriptografi yang lebih kuat (Menezes et al., 1996; Stallings, 2017).

Dalam implementasi praktis, banyak sistem keamanan modern menggunakan pendekatan hybrid cryptosystem, yaitu menggabungkan kriptografi asimetris seperti RSA dengan kriptografi simetris seperti AES untuk meningkatkan efisiensi pengolahan data dalam jumlah besar. RSA biasanya digunakan untuk proses pertukaran kunci, sementara algoritma simetris digunakan untuk enkripsi data utama karena memiliki performa komputasi yang lebih cepat (Ferguson et al., 2010). Selain itu, perkembangan pustaka kriptografi dalam berbagai bahasa pemrograman telah mempermudah implementasi algoritma RSA dalam sistem perangkat lunak, termasuk pada platform berbasis web yang memungkinkan akses sistem secara lebih luas.

Meskipun demikian, sebagian besar penelitian yang membahas implementasi RSA lebih berfokus pada aspek teoritis atau pengembangan algoritma kriptografi itu sendiri, sedangkan penelitian yang mengintegrasikan implementasi RSA dalam aplikasi berbasis web yang bersifat edukatif dan sekaligus melakukan evaluasi kinerja algoritma masih relatif terbatas. Banyak implementasi kriptografi yang tersedia hanya berfokus pada fungsi enkripsi dan dekripsi tanpa memberikan visualisasi proses atau analisis performa algoritma yang digunakan. Padahal, pemahaman terhadap performa komputasi algoritma kriptografi, khususnya dalam kaitannya dengan ukuran kunci yang digunakan, merupakan aspek penting dalam pengembangan sistem keamanan informasi.

Berdasarkan kondisi tersebut, terdapat kesenjangan penelitian (research gap) yang berkaitan dengan kebutuhan akan sistem implementasi kriptografi yang tidak hanya mampu menjalankan fungsi enkripsi dan dekripsi, tetapi juga mampu memberikan gambaran mengenai kinerja algoritma kriptografi dalam lingkungan aplikasi nyata. Selain itu, diperlukan sistem yang dapat digunakan sebagai media pembelajaran yang menjelaskan proses kerja algoritma RSA secara lebih transparan melalui antarmuka yang mudah digunakan.

Untuk menjawab kesenjangan tersebut, penelitian ini mengembangkan sebuah RSA Web Cryptosystem berbasis Python yang mengintegrasikan algoritma RSA ke dalam aplikasi berbasis web dengan memanfaatkan framework Flask serta pustaka kriptografi Python yang telah teruji keamanannya. Sistem yang dikembangkan tidak hanya menyediakan fungsi pembangkitan kunci, enkripsi, dan dekripsi pesan teks, tetapi juga menampilkan parameter kriptografi serta melakukan pencatatan waktu proses enkripsi dan dekripsi. Dengan demikian, sistem ini memungkinkan dilakukan analisis kinerja algoritma RSA berdasarkan variasi ukuran kunci yang digunakan.

Kontribusi utama penelitian ini terletak pada pengembangan sistem kriptografi berbasis web yang menggabungkan implementasi algoritma RSA dengan evaluasi kinerja komputasi dalam satu platform aplikasi. Selain itu, penelitian ini juga memberikan gambaran praktis mengenai hubungan antara ukuran kunci RSA, tingkat keamanan yang dihasilkan, serta performa komputasi yang diperlukan dalam proses enkripsi dan dekripsi data. Dengan adanya sistem ini, diharapkan dapat memberikan kontribusi dalam pengembangan media pembelajaran kriptografi sekaligus menjadi referensi awal bagi pengembangan aplikasi keamanan data berbasis kriptografi pada sistem informasi modern.

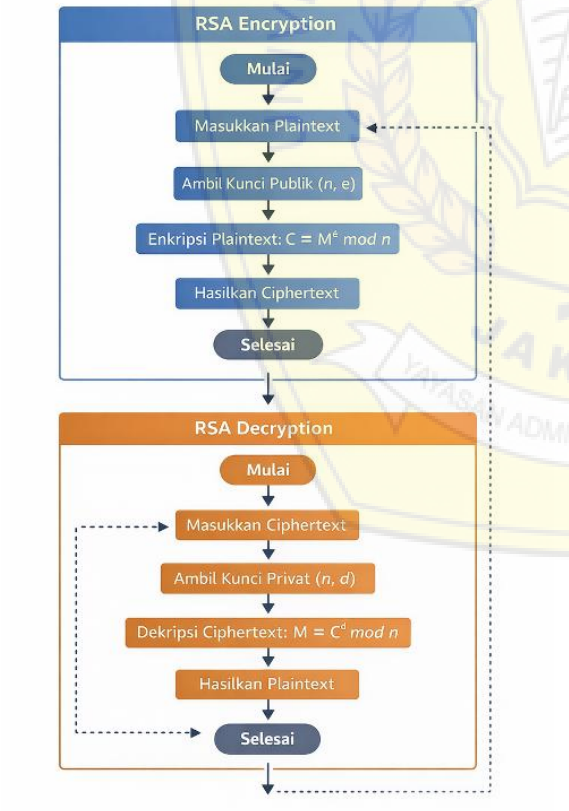
3 Metodologi

Penelitian ini menggunakan pendekatan perancangan dan implementasi sistem (design and implementation approach) yang bertujuan untuk mengembangkan sebuah aplikasi

P-ISSN : 2580-4316
E-ISSN : 2654-8054 DOI: https://doi.org/10.37817/ikraith-informatika.v9i3

kriptografi berbasis web yang menerapkan algoritma RSA sebagai mekanisme pengamanan data. Metodologi penelitian dilakukan melalui beberapa tahapan utama yang meliputi analisis kebutuhan sistem, perancangan arsitektur aplikasi, implementasi algoritma kriptografi, serta pengujian dan evaluasi performa sistem yang dikembangkan.

Tahap pertama dalam penelitian ini adalah analisis kebutuhan sistem, yaitu mengidentifikasi kebutuhan fungsional dan nonfungsional dari aplikasi kriptografi yang akan dikembangkan. Pada tahap ini ditentukan bahwa sistem harus mampu melakukan beberapa fungsi utama, yaitu pembangkitan pasangan kunci RSA (public key dan private key), proses enkripsi pesan teks (plaintext) menjadi ciphertext, serta proses dekripsi ciphertext kembali menjadi plaintext. Selain itu, sistem juga dirancang untuk menampilkan parameter kriptografi yang dihasilkan serta mencatat waktu proses enkripsi dan dekripsi sebagai bagian dari evaluasi kinerja algoritma.



Gambar 1. Flowchart Proses Enkripsi dan Dekripsi RSA

Tahap selanjutnya adalah perancangan arsitektur sistem. Aplikasi yang dikembangkan menggunakan arsitektur client–server berbasis web. Pada arsitektur ini, pengguna berinteraksi dengan sistem melalui antarmuka web yang dijalankan pada browser. Permintaan pengguna, seperti pembangkitan kunci atau proses enkripsi dan dekripsi, dikirimkan ke server melalui mekanisme HTTP request. Server kemudian memproses permintaan tersebut menggunakan modul kriptografi yang telah diimplementasikan dalam bahasa pemrograman Python. Framework Flask digunakan sebagai komponen utama yang menghubungkan antarmuka pengguna dengan modul pemrosesan kriptografi pada sisi server.

Tahap berikutnya adalah implementasi algoritma RSA dalam sistem aplikasi. Algoritma RSA merupakan algoritma kriptografi kunci publik yang bekerja dengan menggunakan pasangan kunci yang saling berkaitan secara matematis. Proses implementasi algoritma RSA dalam sistem meliputi tiga tahap utama, yaitu pembangkitan kunci (key generation), proses enkripsi (encryption), dan proses dekripsi (decryption). Pada tahap pembangkitan kunci, sistem menghasilkan dua bilangan prima besar yang kemudian digunakan untuk menghitung nilai modulus dan parameter kunci publik serta kunci privat. Proses enkripsi dilakukan dengan menggunakan kunci publik RSA untuk mengubah plaintext menjadi ciphertext melalui operasi perpangkatan modular. Sebaliknya, proses dekripsi menggunakan kunci privat untuk mengembalikan ciphertext ke bentuk plaintext semula. Dalam implementasi sistem ini, proses kriptografi dilakukan dengan memanfaatkan pustaka kriptografi Python yang telah menyediakan fungsi pembangkitan kunci RSA serta mekanisme enkripsi dan dekripsi dengan skema padding OAEP (Optimal Asymmetric Encryption Padding) yang direkomendasikan dalam implementasi RSA modern.

Setelah proses implementasi sistem selesai dilakukan, tahap berikutnya adalah pengujian sistem. Pengujian bertujuan untuk memastikan bahwa seluruh fungsi utama aplikasi dapat berjalan dengan baik sesuai dengan perancangan yang telah dilakukan. Pengujian meliputi beberapa aspek, yaitu pengujian pembangkitan kunci RSA, pengujian proses enkripsi pesan teks, serta pengujian proses dekripsi ciphertext. Selain itu, pengujian juga dilakukan untuk memastikan bahwa plaintext yang telah dienkripsi dapat dikembalikan ke

DOI: https://doi.org/10.37817/ikraith-informatika.v9i3
P-ISSN : 2580-4316
E-ISSN : 2654-8054

bentuk semula setelah melalui proses dekripsi menggunakan kunci privat yang sesuai.

Tahap terakhir dalam metodologi penelitian ini adalah evaluasi kinerja algoritma RSA yang diimplementasikan dalam sistem. Evaluasi dilakukan dengan membandingkan performa algoritma RSA berdasarkan variasi ukuran kunci yang digunakan, yaitu RSA 1024-bit dan RSA 2048-bit. Parameter yang dianalisis dalam evaluasi ini meliputi waktu proses enkripsi, waktu proses dekripsi, serta ukuran ciphertext yang dihasilkan. Pengukuran waktu proses dilakukan menggunakan modul waktu pada Python untuk memperoleh nilai waktu eksekusi dalam satuan milidetik. Hasil pengukuran tersebut kemudian dianalisis untuk melihat pengaruh ukuran kunci terhadap performa komputasi algoritma RSA.

Melalui tahapan metodologi tersebut, penelitian ini diharapkan dapat menghasilkan sebuah sistem implementasi kriptografi RSA berbasis web yang tidak hanya mampu menjalankan fungsi enkripsi dan dekripsi data, tetapi juga dapat digunakan sebagai media untuk menganalisis kinerja algoritma kriptografi dalam lingkungan aplikasi nyata.

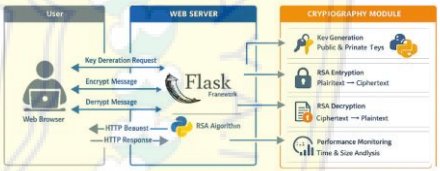
4 Perancangan Sistem

Perancangan sistem bertujuan untuk menggambarkan struktur dan alur kerja aplikasi RSA Web Cryptosystem berbasis Python yang dikembangkan dalam penelitian ini. Sistem dirancang menggunakan pendekatan arsitektur client–server berbasis web, di mana pengguna berinteraksi dengan sistem melalui antarmuka web pada browser, sementara proses komputasi kriptografi dijalankan pada sisi server.

Pada sistem yang dikembangkan, pengguna dapat melakukan beberapa operasi utama, yaitu pembangkitan pasangan kunci RSA, enkripsi pesan teks, serta dekripsi ciphertext. Permintaan dari pengguna dikirimkan melalui antarmuka web ke server menggunakan protokol HTTP. Server kemudian memproses permintaan tersebut dengan memanfaatkan framework Flask yang berfungsi sebagai penghubung antara antarmuka pengguna dan modul pemrosesan kriptografi yang ditulis menggunakan bahasa pemrograman Python.

Arsitektur sistem secara umum terdiri dari tiga komponen utama, yaitu komponen input, modul pemrosesan kriptografi, dan komponen output. Komponen input berfungsi menerima data dari pengguna, seperti plaintext yang akan dienkripsi atau ciphertext yang akan didekripsi. Modul pemrosesan kriptografi merupakan inti dari sistem yang menjalankan algoritma RSA, termasuk pembangkitan kunci, proses enkripsi, dan proses dekripsi data. Sementara itu, komponen output bertugas menampilkan hasil proses kriptografi kepada pengguna, seperti ciphertext hasil enkripsi, plaintext hasil dekripsi, serta informasi waktu proses komputasi.

Dalam implementasi algoritma RSA, sistem memanfaatkan pustaka kriptografi Python yang menyediakan fungsi pembangkitan kunci RSA serta mekanisme enkripsi dan dekripsi menggunakan skema padding OAEP (Optimal Asymmetric Encryption Padding). Skema padding ini digunakan untuk meningkatkan keamanan RSA terhadap berbagai serangan kriptografi modern. Selain itu, sistem juga mencatat waktu proses enkripsi dan dekripsi untuk melakukan evaluasi performa algoritma berdasarkan ukuran kunci yang digunakan.



Gambar 2. Arsitektur Sistem RSA Web Cryptosystem Berbasis Python

5 Hasil dan Pembahasan

5.1 Hasil Implementasi Sistem

Hasil dari penelitian ini adalah sebuah aplikasi RSA Web Cryptosystem berbasis Python yang mampu menjalankan proses pembangkitan kunci, enkripsi, dan dekripsi pesan teks melalui antarmuka web. Sistem memungkinkan pengguna memilih ukuran kunci RSA yang digunakan, yaitu 1024-bit atau 2048-bit. Setelah pasangan kunci berhasil dibangkitkan, pengguna dapat memasukkan plaintext untuk dienkripsi menggunakan kunci publik yang dihasilkan. Sistem kemudian menghasilkan ciphertext yang ditampilkan dalam bentuk karakter terenkripsi.

Selain proses enkripsi, sistem juga menyediakan fitur dekripsi yang memungkinkan ciphertext dikembalikan ke bentuk plaintext menggunakan kunci privat yang sesuai. Hasil pengujian menunjukkan

P-ISSN : 2580-4316
E-ISSN : 2654-8054 DOI: https://doi.org/10.37817/ikraith-informatika.v9i3

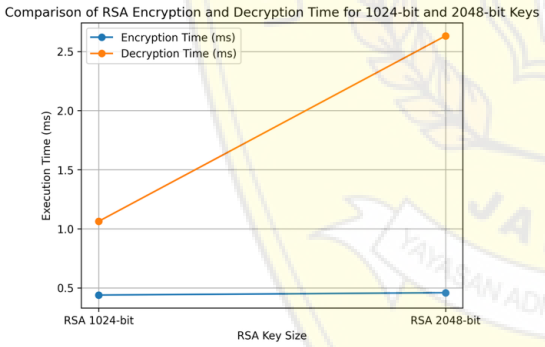
bahwa sistem mampu menjalankan proses enkripsi dan dekripsi dengan benar sesuai dengan prinsip kerja algoritma RSA. Pesan yang telah dienkripsi dapat dikembalikan ke bentuk semula tanpa perubahan isi ketika didekripsi menggunakan kunci privat yang tepat.

5.2 Pengujian Kinerja Algoritma RSA

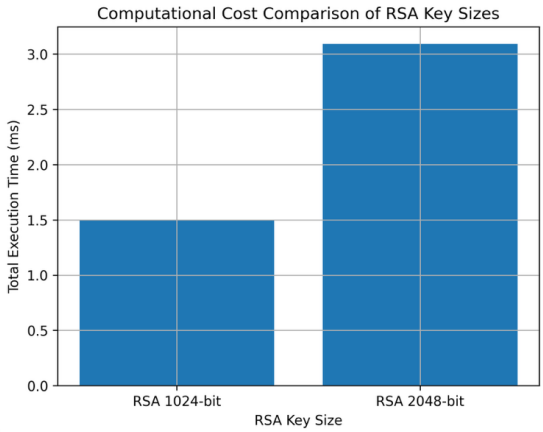
Untuk mengevaluasi kinerja algoritma RSA dalam sistem yang dikembangkan, dilakukan pengujian dengan menggunakan dua ukuran kunci yang berbeda, yaitu RSA 1024-bit dan RSA 2048-bit. Parameter yang dianalisis dalam pengujian ini meliputi waktu proses enkripsi, waktu proses dekripsi, serta ukuran ciphertext yang dihasilkan.

Tabel 1. Perbandingan Hasil Pengujian

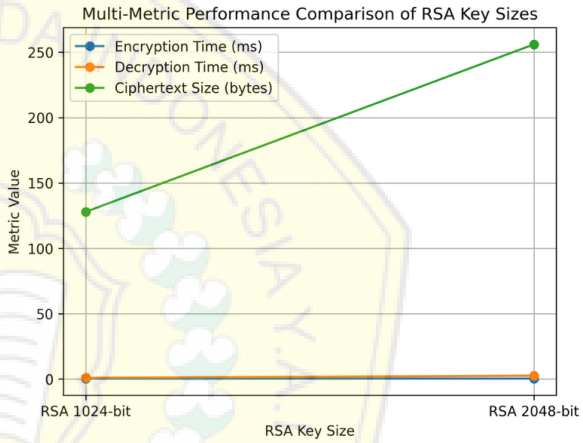
Ukuran Kunci	Waktu Enkripsi	Waktu Dekripsi	Ukuran Ciphertext
RSA 1024-bit	0.438 ms	1.062 ms	128 byte
RSA 2048-bit	0.458 ms	2.632 ms	256 byte



Gambar 3. Perbandingan Waktu Enkripsi dan Dekripsi RSA



Gambar 4. Perbandingan Total Biaya Komputasi RSA Berdasarkan Ukuran Kunci



Gambar 5. Perbandingan Multi-Metrik Performa Algoritma RSA

Berdasarkan hasil pengujian tersebut, dapat diketahui bahwa peningkatan ukuran kunci RSA berdampak langsung terhadap waktu komputasi yang dibutuhkan dalam proses enkripsi dan dekripsi. RSA dengan ukuran kunci yang lebih besar membutuhkan waktu komputasi yang lebih lama dibandingkan dengan ukuran kunci yang lebih kecil. Hal ini terjadi karena operasi matematika pada RSA melibatkan proses perpangkatan modular pada bilangan bulat besar yang kompleksitas komputasinya meningkat seiring dengan bertambahnya panjang kunci.

5.3 Analisis Hasil

Hasil eksperimen menunjukkan adanya hubungan antara ukuran kunci RSA dengan tingkat keamanan serta performa komputasi sistem. Penggunaan kunci yang lebih besar, seperti RSA 2048-bit, memberikan tingkat

keamanan yang lebih tinggi karena lebih sulit untuk dipecahkan melalui proses faktorisasi bilangan besar. Namun demikian, peningkatan ukuran kunci juga menyebabkan meningkatnya waktu proses enkripsi dan dekripsi.

Temuan ini sejalan dengan teori kriptografi modern yang menyatakan bahwa terdapat trade-off antara keamanan dan performa komputasi dalam sistem kriptografi. Sistem dengan tingkat keamanan yang lebih tinggi umumnya memerlukan sumber daya komputasi yang lebih besar untuk menjalankan proses kriptografi. Oleh karena itu, pemilihan ukuran kunci RSA dalam suatu sistem harus mempertimbangkan kebutuhan keamanan serta efisiensi performa sistem yang digunakan.

6 Kesimpulan dan Saran

6.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat disimpulkan bahwa algoritma RSA dapat diimplementasikan dengan baik dalam aplikasi kriptografi berbasis web menggunakan bahasa pemrograman Python dan framework Flask. Sistem yang dikembangkan mampu menjalankan proses pembangkitan kunci, enkripsi plaintext menjadi ciphertext, serta dekripsi ciphertext kembali menjadi plaintext dengan benar sesuai dengan prinsip kerja algoritma RSA.

Hasil pengujian menunjukkan bahwa ukuran kunci RSA memiliki pengaruh signifikan terhadap performa komputasi sistem. RSA dengan ukuran kunci 2048-bit memberikan tingkat keamanan yang lebih tinggi dibandingkan dengan RSA 1024-bit, namun membutuhkan waktu komputasi yang lebih lama dalam proses enkripsi dan dekripsi. Hal ini menunjukkan adanya hubungan antara tingkat keamanan dan efisiensi komputasi dalam implementasi algoritma kriptografi.

Dengan demikian, aplikasi RSA Web Cryptosystem yang dikembangkan dalam penelitian ini dapat berfungsi sebagai media pembelajaran implementasi kriptografi kunci publik serta sebagai prototipe sistem pengamanan data berbasis kriptografi dalam lingkungan aplikasi web.

5.2 Saran

Untuk pengembangan penelitian selanjutnya, beberapa saran yang dapat diberikan antara lain:

1) Mengintegrasikan algoritma RSA dengan algoritma kriptografi simetris seperti AES untuk membangun hybrid cryptosystem yang lebih efisien dalam mengenkripsi data berukuran besar.
2) Mengembangkan sistem dengan fitur keamanan tambahan seperti digital signature dan mekanisme autentikasi pengguna.
3) Melakukan pengujian performa dengan ukuran kunci yang lebih besar, seperti RSA 3072-bit atau RSA 4096-bit, untuk memperoleh analisis keamanan yang lebih komprehensif.
4) Mengembangkan sistem dalam lingkungan komputasi yang lebih luas, seperti integrasi dengan layanan cloud atau aplikasi mobile.

Daftar Pustaka

Barker, E., Chen, L., Roginsky, A., & Vassilev, A. (2023). Recommendation for pair-wise key-establishment using integer factorization cryptography (NIST Special Publication 800-56B Rev. 3). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-56Br3

Bernstein, D. J., Lange, T., & Niederhagen, R. (2022). Post-quantum cryptography. Nature, 549(7671), 188–194. https://doi.org/10.1038/nature23461

Boneh, D., & Shoup, V. (2020). A graduate course in applied cryptography. Draft version. https://crypto.stanford.edu/~dabo/cryptobook/

Ferguson, N., Schneier, B., & Kohno, T. (2010). Cryptography Engineering: Design Principles and Practical Applications. Wiley.

Gordon, D., & Katz, J. (2021). Implementing cryptographic algorithms in modern software systems. IEEE Security & Privacy, 19(4), 78–86. https://doi.org/10.1109/MSEC.2021.3078645

Grinberg, M. (2018). Flask Web Development: Developing Web Applications with Python. O'Reilly Media.

Katz, J., & Lindell, Y. (2020). Introduction to modern cryptography (3rd ed.). CRC Press.

Lenstra, A. K., & Verheul, E. R. (2020). Selecting cryptographic key sizes. Journal of Cryptology, 33(2), 1–21. https://doi.org/10.1007/s00145-019-09337-2

Menezes, A., & Vanstone, S. (2021). Handbook of applied cryptography: Updated perspectives on public-key cryptography. CRC Press.

Menezes, A., van Oorschot, P., & Vanstone, S. (1996). Handbook of Applied Cryptography. CRC Press.

NIST. (2020). Digital signature standard (DSS). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.FIPS.186-4

Rivest, R. L., Shamir, A., & Adleman, L. (2022). Revisiting RSA: Security and performance considerations. Journal of Cryptographic Engineering, 12(3), 257–270. https://doi.org/10.1007/s13389-021-00280-3

Rivest, R., Shamir, A., & Adleman, L. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, 21(2), 120–126. https://doi.org/10.1145/359340.359342

Stallings, W. (2017). Cryptography and Network Security: Principles and Practice. Pearson.

Wang, X., Li, Y., & Zhang, H. (2021). Performance analysis of RSA encryption algorithm in cloud computing environments. Journal of Information Security and Applications, 58, 102706. https://doi.org/10.1016/j.jisa.2021.102706