MIFORTEKH (Jurnal Manajemen Informatika & Teknologi)
p-ISSN : 2808-7550 (print) e-ISSN : 2798-0235 (online)
Vol. 5, No. 1, Mei 2025
https://https//journal.stiestekom.ac.id/index.php/mifortekh

Pengamanan Data Transmisi Aplikasi Web Menggunakan Algoritma Kriptografi RSA: Studi Kasus dan Analisis

Hermalia Putri[1], Lurya Virna[2], Tia Febrianti[3], Tata Sutabri[4]
Jurusan Teknik Informatika, Universitas Bina Darma, Palembang
email:,[1]hermaliap01@gmail.com, [2]lyravirna811@gmail.com, [3]tyafebriantu2204@gmail.com, [4]tatasutabri@gmail.com

Article Info

Article history:
Received Maret 28, 2025
Revised April 13, 2025
Accepted April 28, 2025

Keywords:
RSA Cryptography
Web Security
Encryption
Public Key
Data Transmission
Web Applications

ABSTRACT

In the rapidly developing digital era, web applications have become the main means of exchanging information, ranging from personal data, financial transactions, to business communications. However, the process of transmitting data over the internet network is very vulnerable to various security threats, such as eavesdropping, data tampering, and man-in-the-middle attacks. Therefore, a security mechanism is needed that can guarantee the confidentiality, integrity, and authentication of the data sent.

One method that is widely used to secure data transmission is cryptography. This study focuses on the use of the RSA (Rivest–Shamir–Adleman) asymmetric cryptography algorithm, which uses a pair of public and private keys in the data encryption and decryption process. RSA provides advantages in terms of key distribution and security based on the mathematical complexity of large prime number factorization.

The case study was conducted on a simple web application in the form of a personal data entry form that is usually sent in plain text. In this implementation, data from the client side is encrypted using the RSA public key, then sent to the server, and decrypted using the private key to be stored in the database. The test results show that RSA is able to secure data well and prevent data from being read directly even if wiretapping occurs. In addition, an analysis of the algorithm's performance in terms of encryption and decryption speed for various data sizes was also carried out.

Although effective in terms of security, the RSA algorithm has limitations in handling large data, which causes a significant decrease in performance. Therefore, RSA is more suitable for use in hybrid systems with symmetric algorithms such as AES to encrypt message contents. This study provides a practical overview of how RSA can be applied in the context of the modern web and an analysis of its advantages and limitations in real practice..

Corresponding Author:
Hermalia Putri,
Universitas Bina Darma
Jl. Jenderal Ahmad Yani No.12, Sei Kambing C II, 20 Ilir D. III, Kec. Ilir Timur I, Kota Palembang, Sumatera Selatan 30113, Indonesia.
Email: hermaliap01@gmail.com

p-ISSN: 2808-7550
e-ISSN: : 2798-0235

ABSTRAK

Dalam era digital yang semakin berkembang pesat, aplikasi web telah menjadi sarana utama dalam pertukaran informasi, mulai dari data pribadi, transaksi keuangan, hingga komunikasi bisnis. Namun, proses transmisi data melalui jaringan internet sangat rentan terhadap berbagai ancaman keamanan, seperti penyadapan (eavesdropping), modifikasi data (data tampering), dan serangan man-in-the-middle. Oleh karena itu, diperlukan suatu mekanisme pengamanan yang mampu menjamin kerahasiaan, integritas, dan autentikasi data yang dikirim.

Salah satu metode yang banyak digunakan untuk mengamankan transmisi data adalah kriptografi. Penelitian ini memfokuskan pada penggunaan algoritma kriptografi asimetris RSA (Rivest–Shamir–Adleman), yang menggunakan pasangan kunci publik dan kunci privat dalam proses enkripsi dan dekripsi data. RSA memberikan keunggulan dalam hal distribusi kunci dan keamanan berbasis kompleksitas matematis faktorisasi bilangan prima besar.

Studi kasus dilakukan pada aplikasi web sederhana berupa formulir pengisian data pribadi yang biasanya dikirim dalam bentuk teks biasa (plaintext). Dalam implementasi ini, data dari sisi klien dienkripsi menggunakan kunci publik RSA, kemudian dikirim ke server, dan didekripsi menggunakan kunci privat untuk disimpan di database. Hasil pengujian menunjukkan bahwa RSA mampu mengamankan data dengan baik dan mencegah data terbaca secara langsung meskipun terjadi penyadapan. Selain itu, dilakukan juga analisis terhadap performa algoritma dalam hal kecepatan enkripsi dan dekripsi terhadap berbagai ukuran data.

Meskipun efektif dari sisi keamanan, algoritma RSA memiliki keterbatasan dalam menangani data berukuran besar, yang menyebabkan penurunan performa secara signifikan. Oleh karena itu, RSA lebih cocok digunakan dalam sistem hybrid bersama algoritma simetris seperti AES untuk mengenkripsi isi pesan. Penelitian ini memberikan gambaran praktis tentang bagaimana RSA dapat diterapkan dalam konteks web modern serta analisis kelebihan dan keterbatasannya dalam praktik nyata.

Kata Kunci: *Kriptografi RSA, Keamanan Web, Enkripsi, Kunci Publik, Transmisi Data, Aplikasi Web.*

1. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi yang pesat telah membawa dampak signifikan terhadap cara manusia berinteraksi dan bertukar informasi. Aplikasi web menjadi salah satu media utama dalam menunjang berbagai aktivitas digital, mulai dari sistem informasi akademik, layanan e-commerce, layanan publik, hingga aplikasi berbasis keuangan. Dalam proses operasionalnya, aplikasi web banyak berinteraksi dengan data pengguna, baik dalam bentuk input, pemrosesan, maupun penyimpanan. Oleh karena itu, data yang ditransmisikan antara klien (pengguna) dan server memiliki potensi untuk menjadi target serangan siber apabila tidak dilindungi secara memadai.

Salah satu risiko utama dalam transmisi data adalah terjadinya penyadapan (*eavesdropping*) oleh pihak yang tidak berwenang, yang dapat menyebabkan kebocoran data pribadi, informasi login, hingga data transaksi yang bersifat

sensitif. Selain itu, serangan man-in-the-middle (MITM) dapat dilakukan untuk memodifikasi atau menyisipkan data palsu selama proses komunikasi berlangsung. Ancaman-ancaman ini menjadi perhatian penting dalam pengembangan sistem aplikasi web yang aman dan andal.

Dalam konteks pengamanan transmisi data, kriptografi merupakan teknik yang banyak digunakan untuk menjaga kerahasiaan dan integritas informasi. Kriptografi modern terbagi menjadi dua kategori utama, yaitu kriptografi simetris dan kriptografi asimetris. Kriptografi simetris menggunakan satu kunci yang sama untuk enkripsi dan dekripsi, sedangkan kriptografi asimetris menggunakan sepasang kunci, yaitu kunci publik dan kunci privat. Salah satu algoritma kriptografi asimetris yang paling terkenal dan banyak digunakan adalah algoritma RSA (Rivest–Shamir–Adleman).

RSA dikembangkan pada tahun 1977 dan menjadi dasar dari banyak protokol keamanan modern seperti SSL/TLS yang digunakan dalam komunikasi HTTPS. RSA menawarkan tingkat keamanan tinggi karena proses dekripsinya hanya dapat dilakukan dengan kunci privat yang bersifat rahasia, sementara proses enkripsinya menggunakan kunci publik yang dapat dibagikan secara terbuka. Dengan demikian, komunikasi data dapat diamankan tanpa perlu mendistribusikan kunci rahasia melalui jalur komunikasi yang rawan.

Penelitian ini bertujuan untuk mengimplementasikan dan menganalisis algoritma RSA dalam konteks aplikasi web, dengan fokus pada bagaimana RSA dapat melindungi data yang dikirimkan dari sisi klien ke sisi server. Studi kasus dilakukan pada sistem formulir online, di mana data identitas pengguna seperti nama dan nomor telepon dikirim melalui internet. Penelitian ini juga menganalisis aspek performa algoritma, termasuk waktu yang dibutuhkan untuk proses enkripsi dan dekripsi, serta batasan dalam penggunaan RSA untuk data berukuran besar.

Dengan memahami dan menguji langsung implementasi RSA dalam skenario web, diharapkan penelitian ini dapat memberikan kontribusi dalam meningkatkan kesadaran dan praktik pengamanan data pada aplikasi web yang semakin banyak digunakan dalam kehidupan sehari-hari, terutama di tengah meningkatnya serangan siber dan kebutuhan akan perlindungan data pribadi.

2. KAJIAN TEORI

2.1 Keamanan Data dalam Aplikasi Web

Keamanan data merupakan aspek fundamental dalam pengembangan aplikasi web, khususnya ketika aplikasi tersebut menangani informasi sensitif seperti data pribadi, kredensial login, transaksi keuangan, atau dokumen rahasia. Dalam lingkungan web, data ditransmisikan melalui jaringan terbuka seperti internet, yang berarti informasi tersebut dapat dengan mudah diakses, dimodifikasi, atau dimanipulasi oleh pihak yang tidak berwenang jika tidak diberikan perlindungan yang memadai.

Beberapa ancaman utama terhadap keamanan data dalam aplikasi web antara lain:

- Penyadapan (Eavesdropping): Penyerang dapat menangkap data yang ditransmisikan antara pengguna dan server.
- Man-in-the-Middle (MITM): Penyerang menyisipkan dirinya di antara komunikasi dua pihak dan mengakses atau memanipulasi data secara diam-diam.
- Data Tampering: Data yang dikirim dapat diubah sebelum sampai ke tujuan.
- Phishing & Spoofing: Data pengguna dikumpulkan melalui halaman web palsu yang menyerupai halaman asli.

Untuk mengatasi ancaman-ancaman ini, penerapan protokol keamanan seperti HTTPS, penggunaan SSL/TLS, dan pengamanan di tingkat aplikasi melalui teknik enkripsi menjadi sangat penting.

2.2 Pengantar Kriptografi

Kriptografi adalah teknik dan ilmu yang berkaitan dengan pengamanan informasi melalui proses transformasi data agar tidak dapat dibaca oleh pihak yang tidak berwenang. Tujuan utama dari kriptografi meliputi:

- Kerahasiaan (Confidentiality): Hanya pihak yang berwenang yang dapat membaca data.
- Integritas (Integrity): Data tidak mengalami perubahan tanpa terdeteksi.

- Autentikasi (Authentication): Identitas pengirim atau penerima dapat diverifikasi.
- Non-repudiation: Pengirim tidak dapat menyangkal bahwa ia telah mengirimkan data.

2.3 Kriptografi Simetris dan Asimetris

2.3.1 Kriptografi Simetris

Kriptografi simetris menggunakan satu kunci yang sama untuk proses enkripsi dan dekripsi. Contoh algoritma simetris yang populer antara lain:

- AES (*Advanced Encryption Standard*)
- DES (*Data Encryption Standard*)
- Blowfish

Kelebihan dari kriptografi simetris adalah kecepatan enkripsi yang tinggi. Namun, kelemahannya terletak pada distribusi kunci yang aman, karena jika kunci berhasil didapatkan oleh pihak ketiga, seluruh sistem akan terancam.

2.3.2 Kriptografi Asimetris

Berbeda dengan simetris, kriptografi asimetris menggunakan dua buah kunci:

- Kunci publik (*public key*): Dapat dibagikan ke siapa saja dan digunakan untuk mengenkripsi pesan.
- Kunci privat (*private key*): Bersifat rahasia dan hanya digunakan oleh pemiliknya untuk mendekripsi pesan.

Keunggulan kriptografi asimetris terletak pada kemudahan distribusi kunci dan keamanan tingkat tinggi terhadap penyadapan. Namun, proses enkripsi-dekripsi lebih lambat dibandingkan dengan kriptografi simetris.

2.4 Algoritma RSA (Rivest–Shamir–Adleman)

RSA adalah algoritma kriptografi asimetris yang paling terkenal dan digunakan luas dalam komunikasi aman berbasis internet. Algoritma ini ditemukan oleh Ron

p-ISSN: 2808-7550
e-ISSN: : 2798-0235

Rivest, Adi Shamir, dan Leonard Adleman pada tahun 1977. RSA didasarkan pada kompleksitas faktorisasi bilangan bulat besar yang merupakan hasil perkalian dua bilangan prima.

Langkah-langkah algoritma RSA:

1. Pembuatan Kunci:
   - Pilih dua bilangan prima besar: p dan q.
   - Hitung n = p × q.
   - Hitung φ(n) = (p−1)(q−1).
   - Pilih bilangan e (public exponent), dengan 1 < e < φ(n), dan gcd(e, φ(n)) = 1.
   - Hitung d sebagai invers dari e modulo φ(n), yaitu $d \equiv e^{-1} \bmod \varphi(n)$.
   - Pasangan kunci:
   - Kunci publik (e, n) digunakan untuk enkripsi.
   - Kunci privat (d, n) digunakan untuk dekripsi.
2. Proses Enkripsi:
   - Plaintext (dinyatakan sebagai bilangan M): C = M^e mod n
3. Proses Dekripsi:
   - Ciphertext C : M = C^d mod n

Keunggulan RSA:

- Kunci publik dapat disebarluaskan tanpa membahayakan keamanan data.
- Sangat cocok digunakan untuk otentikasi dan pengamanan pertukaran kunci (misalnya dalam protokol SSL/TLS).
- RSA memungkinkan penerapan tanda tangan digital (digital signature).

Kelemahan RSA:

- Lambat untuk data berukuran besar.
- Tidak efisien jika digunakan untuk enkripsi seluruh dokumen atau file media.
- Membutuhkan bilangan prima sangat besar untuk menjaga tingkat keamanan yang tinggi.

2.5 RSA dalam Pengamanan Web

RSA merupakan komponen utama dalam berbagai protokol keamanan web modern. Contoh paling nyata adalah pada protokol HTTPS, di mana RSA digunakan saat proses *handshake* untuk pertukaran kunci simetris yang nantinya digunakan untuk sesi komunikasi aman.

Selain dalam HTTPS, RSA juga bisa diimplementasikan langsung dalam skenario pengiriman data manual, misalnya ketika pengguna mengisi form login, lalu data dienkripsi di sisi klien menggunakan kunci publik server dan dikirim untuk didekripsi di sisi server menggunakan kunci privat.

Implementasi RSA di tingkat aplikasi web dapat menjadi solusi ketika pengembang tidak memiliki akses terhadap SSL/TLS atau memerlukan lapisan keamanan tambahan. Namun, praktik terbaik adalah menggabungkan RSA dengan algoritma simetris seperti AES dalam sistem hybrid, di mana RSA hanya digunakan untuk enkripsi kunci AES yang digunakan untuk data.

2.6 Studi Terkait

Berbagai studi sebelumnya telah menunjukkan efektivitas algoritma RSA dalam mengamankan transmisi data. Beberapa penelitian menggabungkan RSA dengan protokol keamanan lainnya untuk meningkatkan efisiensi dan skalabilitas. Studi yang dilakukan oleh Aji (2022) menunjukkan bahwa penggunaan RSA pada form login memberikan perlindungan terhadap penyadapan, meskipun mengalami peningkatan waktu proses dibanding form tanpa enkripsi. Sementara itu, penelitian dari Wulandari et al. (2021) menunjukkan bahwa kombinasi RSA dan AES (hybrid) mampu menyeimbangkan antara kecepatan dan keamanan.

3. METODE PENELITIAN

3.1 Pendekatan Penelitian

Penelitian ini menggunakan pendekatan **kualitatif dan kuantitatif** dengan metode studi kasus dan eksperimen untuk menganalisis implementasi algoritma RSA dalam pengamanan transmisi data aplikasi web. Pendekatan kualitatif digunakan untuk memahami konsep dan teori kriptografi serta pengaplikasiannya pada sistem web, sementara pendekatan kuantitatif digunakan untuk mengukur performa algoritma RSA dalam proses enkripsi dan dekripsi data.

3.2 Tahapan Penelitian

Penelitian ini terdiri dari beberapa tahapan utama sebagai berikut:

3.2.1 Studi Literatur dan Kajian Teori

Melakukan telaah pustaka mendalam terkait kriptografi, khususnya algoritma RSA, dan penerapannya dalam pengamanan data transmisi web. Literatur yang dikaji meliputi jurnal ilmiah, buku, artikel teknis, dan sumber terpercaya lainnya.

3.2.2 Perancangan Sistem

Membuat rancangan sistem aplikasi web sederhana yang akan digunakan sebagai objek penelitian. Sistem ini berupa formulir pengisian data identitas pengguna yang akan dienkripsi menggunakan algoritma RSA pada sisi klien sebelum dikirim ke server. Rancangan sistem mencakup:

- Desain antarmuka pengguna (UI)
- Alur proses enkripsi dan dekripsi
- Penyimpanan data di server

3.2.3 Implementasi Algoritma RSA

Mengimplementasikan algoritma RSA dalam aplikasi web menggunakan bahasa pemrograman PHP untuk sisi server dan JavaScript untuk enkripsi di sisi klien. Proses ini meliputi:

- Pembuatan kunci publik dan privat RSA
- Penambahan fungsi enkripsi data dengan kunci publik di browser pengguna
- Fungsi dekripsi data dengan kunci privat di server
- Penanganan error dan validasi data

3.2.4 Pengujian Sistem

Pengujian dilakukan untuk mengukur:

- **Keberhasilan enkripsi dan dekripsi:** memastikan data terenkripsi dapat berhasil didekripsi dan valid.
- **Kecepatan proses:** pengukuran waktu enkripsi dan dekripsi dengan variasi ukuran data input.
- **Keamanan:** pengujian simulasi serangan penyadapan untuk melihat apakah data yang ditransmisikan dapat dibaca tanpa kunci privat.

Pengujian dilakukan dengan mengirimkan data berbagai ukuran mulai dari teks pendek (nama) hingga teks yang lebih panjang (alamat lengkap).

3.2.5 Analisis Data

Melakukan analisis hasil pengujian dari segi performa dan keamanan. Data yang diperoleh dianalisis secara statistik sederhana untuk melihat hubungan antara ukuran data dan waktu proses enkripsi/dekripsi. Analisis kualitatif juga dilakukan untuk menilai efektivitas pengamanan berdasarkan skenario simulasi ancaman.

3.3 Tools dan Teknologi yang Digunakan

- **Bahasa Pemrograman:** PHP untuk backend, JavaScript untuk frontend (enkripsi sisi klien)
- **Web Server:** Apache yang berjalan di lingkungan lokal (XAMPP/Laragon)

- **Database:** MySQL untuk penyimpanan data hasil dekripsi
- **Library/Framework:**
  - PHP OpenSSL untuk operasi kriptografi RSA
  - JavaScript CryptoJS (atau library RSA khusus) untuk enkripsi di sisi klien
- **Alat Pengukur Waktu:** Fungsi *microtime()* di PHP dan *performance.now()* di JavaScript untuk akurasi pengukuran proses.

3.4 Desain Sistem dan Alur Kerja

Sistem yang dirancang memiliki dua komponen utama, yaitu:

1. **Client-Side:**
   - User mengisi formulir data
   - Data dienkripsi menggunakan kunci publik RSA melalui JavaScript
   - Data terenkripsi dikirim ke server melalui HTTP POST
2. **Server-Side:**
   - Menerima data terenkripsi
   - Melakukan dekripsi menggunakan kunci privat RSA
   - Menyimpan data hasil dekripsi ke database
   - Memberikan respons kepada client sebagai tanda sukses

Diagram alur sistem dapat digambarkan sebagai berikut:

3.5 Variabel dan Parameter yang Diukur

- **Variabel Independen:** Ukuran data yang dienkripsi (jumlah karakter)
- **Variabel Dependen:**
  - Waktu proses enkripsi (ms)
  - Waktu proses dekripsi (ms)
  - Keberhasilan pengamanan (hasil pengujian simulasi serangan)

3.6 Batasan Penelitian

- RSA digunakan untuk mengenkripsi data teks dengan ukuran terbatas (tidak lebih dari 1 KB) untuk menjaga performa.
- Pengujian dilakukan di lingkungan lokal sehingga variabel jaringan publik tidak diperhitungkan.
- Fokus penelitian pada algoritma RSA tanpa kombinasi dengan algoritma lain seperti AES (hybrid).

4. HASIL DAN PEMBAHASAN

4.1 Implementasi Algoritma RSA pada Aplikasi Web

Pada tahap implementasi, algoritma RSA berhasil diterapkan dalam sebuah aplikasi web sederhana yang terdiri dari formulir pengisian data

pengguna. Data yang dimasukkan oleh pengguna dienkripsi menggunakan kunci publik RSA di sisi klien (browser) dengan JavaScript sebelum dikirim ke server. Server kemudian melakukan dekripsi menggunakan kunci privat RSA dengan PHP OpenSSL.

Proses ini memastikan bahwa data yang melewati jaringan internet tidak dalam bentuk teks asli (plaintext), melainkan sudah dalam bentuk ciphertext yang tidak dapat dibaca oleh pihak ketiga yang mencoba melakukan penyadapan. Implementasi ini menunjukkan bahwa algoritma RSA dapat diintegrasikan ke dalam aplikasi web tanpa memerlukan protokol keamanan tambahan seperti HTTPS, meskipun penggunaan HTTPS tetap sangat dianjurkan untuk keamanan menyeluruh.

4.2 Pengujian Performa Enkripsi dan Dekripsi

Pengujian dilakukan dengan berbagai ukuran data input, mulai dari 10 karakter hingga 512 karakter. Masing-masing ukuran data diukur waktu yang dibutuhkan untuk proses enkripsi di sisi klien dan proses dekripsi di sisi server.

Ukuran Data (Karakter)	Waktu Enkripsi (ms)	Waktu Dekripsi (ms)
10	12	8
50	15	10
100	20	13
256	38	26
512	75	50

Analisis:

- Waktu proses enkripsi dan dekripsi meningkat secara linier seiring bertambahnya ukuran data.
- Enkripsi di sisi klien relatif lebih cepat dibandingkan dekripsi di server, kemungkinan karena implementasi JavaScript yang lebih ringan dan overhead server.

- Meski waktu bertambah, rentang waktu masih dalam kategori wajar untuk aplikasi web interaktif.

4.3 Analisis Keamanan

Simulasi serangan dilakukan dengan mencoba menangkap data yang dikirim melalui network sniffing menggunakan tools seperti Wireshark. Hasilnya, data yang diterima oleh server dalam bentuk terenkripsi sehingga tidak dapat dibaca atau dimanipulasi tanpa kunci privat.

Selain itu, pengujian mencoba memasukkan ciphertext yang diubah secara acak, dan sistem menolak data tersebut saat proses dekripsi, menunjukkan bahwa integritas data terjaga. Ini mengindikasikan bahwa RSA juga membantu dalam menjaga validitas data selama transmisi.

4.4 Kelebihan Implementasi RSA pada Aplikasi Web

- **Keamanan Tinggi:** RSA menggunakan kunci publik dan privat yang berbeda sehingga data tetap aman meskipun kunci publik diketahui banyak pihak.
- **Distribusi Kunci Mudah:** Tidak perlu distribusi kunci privat yang rahasia karena hanya kunci publik yang disebarkan.
- **Integrasi Mudah:** RSA dapat diimplementasikan menggunakan library yang sudah ada di berbagai bahasa pemrograman, memudahkan integrasi di sisi klien dan server.
- **Mencegah Penyadapan:** Data yang ditransmisikan tidak dapat dibaca oleh pihak ketiga.

4.5 Keterbatasan dan Tantangan

- **Kecepatan Proses:** RSA memiliki proses yang lebih lambat dibanding algoritma simetris, sehingga kurang efisien untuk data berukuran besar.
- **Ukuran Data Terbatas:** RSA idealnya digunakan untuk data berukuran kecil, seperti kunci sesi atau informasi singkat. Untuk data besar, diperlukan kombinasi dengan algoritma lain.

- **Penggunaan Resource:** Proses enkripsi dan dekripsi RSA memerlukan resource komputasi cukup besar, yang dapat berdampak pada performa server jika banyak request simultan.
- **Pengamanan Tambahan:** Implementasi RSA tidak menggantikan kebutuhan penggunaan HTTPS yang memberikan perlindungan lapisan transport yang lengkap.

4.6 Studi Perbandingan dengan Algoritma Lain

Dalam literatur, algoritma seperti AES memiliki kecepatan proses enkripsi dan dekripsi yang jauh lebih cepat dibanding RSA, namun memiliki kelemahan pada distribusi kunci. Oleh karena itu, sistem keamanan modern sering menggunakan pendekatan **hybrid cryptosystem** yang menggabungkan kekuatan RSA (untuk distribusi kunci) dan AES (untuk enkripsi data).

4.7 Implikasi dan Rekomendasi

Implementasi RSA dalam aplikasi web sangat berguna untuk pengamanan data pada sistem yang belum memiliki protokol HTTPS. Namun, untuk aplikasi berskala besar dan data sensitif yang besar, disarankan untuk menggunakan RSA dalam kombinasi dengan algoritma simetris serta protokol komunikasi aman.

Pengembang aplikasi juga harus memperhatikan manajemen kunci secara ketat, memastikan kunci privat disimpan dengan aman agar tidak terjadi kebocoran yang dapat membahayakan keamanan sistem.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil implementasi dan analisis terhadap pengamanan data transmisi aplikasi web menggunakan algoritma kriptografi RSA, dapat disimpulkan bahwa RSA merupakan metode yang sangat efektif dalam menjaga kerahasiaan dan integritas data yang dikirim dari klien ke server. Enkripsi data di sisi klien menggunakan kunci publik RSA dan dekripsi di sisi server menggunakan kunci privat RSA memberikan jaminan bahwa data yang dikirim tidak dapat diakses oleh pihak yang tidak memiliki otoritas, meskipun data tersebut melewati jaringan publik seperti internet.

RSA sangat cocok digunakan untuk data yang bersifat sensitif namun berukuran kecil, seperti kredensial login (username dan password), token autentikasi, dan kunci simetris dalam sistem hybrid. Dari sisi keamanan, RSA dengan panjang kunci minimal 2048 bit masih dianggap aman terhadap berbagai serangan kriptografi modern, termasuk brute force dan analisis faktor bilangan prima.

Namun, algoritma RSA juga memiliki sejumlah keterbatasan, terutama dalam hal performa dan efisiensi. Proses dekripsi dengan kunci privat memerlukan waktu komputasi yang cukup besar, terutama jika panjang kunci tinggi. RSA tidak efisien untuk mengenkripsi data dalam jumlah besar atau file besar secara langsung, karena keterbatasan ukuran blok dan kecepatan. Dalam implementasi web modern, RSA lebih disarankan digunakan sebagai bagian dari sistem kriptografi hybrid, di mana RSA digunakan untuk mengenkripsi kunci simetris, sedangkan algoritma seperti AES digunakan untuk enkripsi data utama.

Secara keseluruhan, implementasi RSA pada aplikasi web memberikan peningkatan keamanan signifikan terhadap risiko pencurian data selama transmisi, selama digunakan dengan konfigurasi yang tepat dan batasan yang dipahami dengan baik.

5.2 Saran

Untuk pengembangan lebih lanjut dan penerapan sistem pengamanan berbasis RSA yang lebih optimal, beberapa saran yang dapat diberikan adalah sebagai berikut:

1. **Penggunaan Kriptografi Hybrid (RSA + AES):**

   Mengingat RSA tidak cocok untuk mengenkripsi data besar, maka sangat disarankan untuk menggabungkannya dengan algoritma simetris seperti AES. RSA digunakan untuk mengenkripsi kunci AES, sedangkan data utama dienkripsi menggunakan AES. Pendekatan ini banyak digunakan dalam protokol HTTPS dan terbukti efisien dan aman.

2. **Peningkatan Panjang Kunci RSA:**

   Untuk menjaga keamanan jangka panjang terhadap ancaman komputasi modern, terutama dengan berkembangnya komputasi kuantum, disarankan untuk menggunakan panjang kunci RSA minimal 3072 bit. Namun, hal ini perlu dibarengi dengan evaluasi beban komputasi di sisi server.

3. **Optimasi Performa Server:**

   Karena proses dekripsi RSA cenderung lambat, terutama dengan panjang kunci yang tinggi, maka optimasi server seperti penggunaan cache, load balancing, dan pengelolaan thread secara efisien menjadi penting, agar performa tidak menurun saat menangani banyak permintaan klien secara simultan.

4. **Validasi Keamanan dari Pihak Ketiga:**

   Implementasi sistem keamanan sebaiknya diverifikasi oleh pihak ketiga melalui audit keamanan atau penetration testing, untuk memastikan tidak terdapat celah atau kesalahan konfigurasi dalam penggunaan RSA yang bisa dimanfaatkan oleh pihak yang tidak bertanggung jawab.

5. **Edukasi Pengguna dan Developer:**

   Selain teknologi, faktor manusia juga penting. Pengembang harus memahami cara kerja dan batasan RSA, sedangkan pengguna akhir perlu diedukasi agar tidak membagikan kunci atau data penting secara sembarangan.

6. **Pemantauan dan Logging:**

   Sistem harus dilengkapi dengan fitur pemantauan dan pencatatan log atas aktivitas enkripsi dan dekripsi, guna mendeteksi aktivitas mencurigakan atau serangan yang mungkin terjadi.

DAFTAR PUSTAKA

Rivest, R. L., 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* (7th ed.). Pearson Education.

Paar, C., & Pelzl, J. (2010). *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer. https://doi.org/10.1007/978-3-642-04101-3

OpenSSL Project. (2023). *OpenSSL Cryptography and SSL/TLS Toolkit*. https://www.openssl.org/

Mozilla Developer Network (MDN). (2023). *HTTPS and Public Key Encryption*. https://developer.mozilla.org/en-US/docs/Web/Security

JSEncrypt Project. (2023). *JSEncrypt: JavaScript Library for RSA Encryption*. https://github.com/travist/jsencrypt

Diffie, W., & Hellman, M. (1976). *New Directions in Cryptography*. IEEE Transactions on Information Theory, 22(6), 644–654. https://doi.org/10.1109/TIT.1976.1055638

Kaur, P., & Singh, H. (2020). *Performance Analysis of RSA and AES Encryption Algorithms for Web Applications*. *International Journal of Computer Applications*, 175(15), 1–6. https://doi.org/10.5120/ijca2020920712

Rouse, M. (2022). *What is RSA (Rivest-Shamir-Adleman)?* TechTarget - SearchSecurity. https://www.techtarget.com/searchsecurity/definition/RSA

Sutabri, T., Enjelika, D., Mujiranda, S., & Virna, L. (2023). Transformasi digital di puskesmas menuju pelayanan kesehatan yang lebih efisien dan berkualitas. *IJM: Indonesian Journal of Multidisciplinary*, *1*(5), 1–11. https://journal.csspublishing.com/index.php/ijm/article/view/389

Sutabri, T., Enjelika, D., Virna, L., & Mujiranda, S. (2023). "Mengoptimalkan Konsumsi Energi di Rumah Pintar Menggunakan Sistem Pendukung Keputusan Cerdas." IJM: Indonesian Journal of Multidisciplinary, 1(6). Diakses dari https://journal.csspublishing.com/index.php/ijm/article/view/533

Sutabri, T., & Napitupulu, D. (2019). Sistem Informasi Bisnis. Yogyakarta: Penerbit Andi.

Sutabri, T. (2012). Konsep Sistem Informasi. Yogyakarta: Penerbit Andi.

Nabila, P. H., Hermalia, P., & Tia, F. (2023). Transformasi pendidikan di Indonesia selama pandemi. *IJM: Indonesian Journal of Multidisciplinary, 2*(1). https://ojs.csspublishing.com/index.php/ijm/article/view/132

Putri, H., & Sutabri, T. (2025). Peran chatbot dalam meningkatkan layanan pelanggan di era digital. *IJM: Journal of Multidisciplinary, 3*(3). https://ojs.csspublishing.com/index.php/ijm/article/view/132

Virna, L., & Sutabri, T. (2025, April 21). *Eksplorasi pemanfaatan teknologi robotika dalam dunia industri dan pendidikan di Indonesia*. IJM: Journal of Multidisciplinary, 3(3).

https://ojs.csspublishing.com/index.php/ijm/article/view/133

Febrianti, T., & Sutabri, T. (2025, April 21). *Analisis pengaruh teknologi dan automasi pada tenaga kerja manusia*. IJM: Journal of Multidisciplinary, 3(3). https://ojs.csspublishing.com/index.php/ijm/article/view/134

Julian, D., & Sutabri, T. (2023). *Analisa kinerja aplikasi digital forensik Autopsy untuk pengembalian data menggunakan metode NIST SP 800-86*. Jurnal Informatika Terpadu, 9(2), 136–142. https://journal.nurulfikri.ac.id/index.php/JIT

Pratama, Y., & Sutabri, T. (2023). *Analisis kriptografi algoritma Blowfish pada keamanan data menggunakan Dart*. Jurnal Informatika Terpadu, 9(2), 126–135. https://journal.nurulfikri.ac.id/index.php/JIT