JIKUM
JURNAL ILMU KOMPUTER

E-ISSN: 3090-739X
Volume 1, **Number** 2, Nopember 2025
https://doi.org/10.62671/jikum.v1i2.40

Pengembangan Aplikasi Kriptografi RSA dan SHA-256 Berbasis Web Menggunakan Flask

Nur Azila Tarigan[1], Anastasya Cut Nabila Anggreni[2], Alfia Balqis[3], Indah Nurfadilah[4], Eri Setia Bakti[5], Fadli Mahyudin[6]
[1,2,3,4,5,6]Universitas Malikussaleh, Indonesia
[1]azilatarigan@gmail.com, [2]anastasya.220180136@mhs.unimal.ac.id, [3]alfiyabalqis98@gmail.com, [4]indah.220180039@mhs.unimal.ac.id

ABSTRACT

In today's digital era, data security has become a crucial issue due to the increasing volume of information exchanged over the internet. One of the main approaches to protecting data is through the use of cryptographic techniques. This research presents the development of a modern cryptographic application based on web technology, implementing the RSA (Rivest-Shamir-Adleman) algorithm for encryption and decryption processes, as well as the SHA-256 (Secure Hash Algorithm) for generating hash values to verify data integrity. The application is built using the Flask framework and Python programming language, complemented by a responsive interface developed with Bootstrap. Features include the ability to encrypt and decrypt messages, generate hash values, and download keys and processing results in text file formats. The development methodology follows a prototyping approach, with a series of functional tests conducted to ensure encryption accuracy, hashing consistency, and file export reliability. The results show that the application successfully performs all features and provides an intuitive and efficient user experience in understanding basic cryptographic concepts.

Kata Kunci: *RSA, SHA-256, Kriptografi, Enkripsi, Dekripsi, Flask, Python, Keamanan Data.*

PENDAHULUAN

Dalam era digital saat ini, perlindungan informasi menjadi prioritas utama dalam menjaga privasi dan keamanan data. Ancaman terhadap keamanan data, seperti penyadapan, manipulasi informasi, dan pencurian identitas, semakin berkembang seiring dengan kemajuan teknologi informasi. Salah satu cara paling efektif untuk melindungi informasi adalah melalui penerapan teknik kriptografi.

RSA (Rivest-Shamir-Adleman) adalah algoritma kriptografi kunci publik yang memungkinkan proses enkripsi dan dekripsi data dengan menggunakan sepasang kunci yang berbeda, yaitu kunci publik dan kunci privat. RSA banyak diterapkan dalam berbagai bidang, seperti keamanan transaksi online, tanda tangan digital, dan autentikasi pengguna.

Selain enkripsi, verifikasi integritas data juga menjadi aspek penting dalam keamanan informasi. SHA-256 (Secure Hash Algorithm 256-bit) merupakan algoritma hash yang mampu menghasilkan representasi unik dan tetap dari data input, sehingga setiap perubahan sekecil apapun pada data dapat terdeteksi dengan mudah. Fungsi hash ini digunakan secara luas dalam pengamanan password, blockchain, dan sistem verifikasi file.

Pengembangan aplikasi kriptografi berbasis web ini bertujuan untuk memperkenalkan penggunaan RSA dan SHA-256 secara praktis kepada pengguna, sekaligus memberikan pengalaman langsung dalam memahami proses enkripsi, dekripsi, dan hashing. Dengan tampilan antarmuka berbasis Flask dan Bootstrap, aplikasi ini diharapkan dapat meningkatkan pemahaman masyarakat umum terhadap pentingnya perlindungan data di era digital.

TINJAUAN PUSTAKA

Kriptografi merupakan salah satu pilar utama dalam keamanan data digital. Teknik ini berfungsi untuk melindungi informasi dengan cara mengubah data menjadi bentuk yang tidak dapat dibaca tanpa proses dekripsi yang benar. Dalam dunia kriptografi, terdapat dua pendekatan utama, yaitu kriptografi simetris yang menggunakan satu kunci untuk enkripsi dan dekripsi, serta kriptografi asimetris yang menggunakan pasangan kunci berbeda. RSA merupakan salah satu algoritma kriptografi asimetris paling populer yang digunakan untuk mengamankan komunikasi digital. Selain itu, untuk menjamin integritas data, algoritma hashing seperti SHA-256 juga sangat penting karena dapat menghasilkan sidik jari digital dari data tanpa kemungkinan dikembalikan ke bentuk aslinya.

RSA (Rivest-Shamir-Adleman)

RSA adalah algoritma kriptografi asimetris yang menggunakan dua kunci berbeda untuk proses enkripsi dan dekripsi. Keamanan RSA bergantung pada kesulitan faktorisasi bilangan besar, sebuah masalah matematika yang memerlukan waktu komputasi sangat besar untuk diselesaikan. Algoritma ini banyak digunakan dalam protokol keamanan seperti SSL/TLS dan sistem tanda tangan digital (Stallings, 2017).

Menurut Rivest, Shamir, dan Adleman (1978), proses enkripsi dalam RSA dilakukan dengan mengubah pesan

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

JIKUM
JURNAL ILMU KOMPUTER
E-ISSN: 3090-739X
Volume 1, **Number** 2, Nopember 2025
https://doi.org/10.62671/jikum.v1i2.40

menjadi bilangan numerik yang kemudian dipangkatkan dengan kunci publik dan dimodulo-kan dengan bilangan besar hasil perkalian dua bilangan prima. Dekripsi dilakukan dengan kunci privat yang hanya diketahui oleh penerima.

SHA-256 (Secure Hash Algorithm 256-bit)

SHA-256 adalah bagian dari keluarga algoritma SHA-2 yang dirancang untuk menghasilkan nilai hash sepanjang 256-bit. Algoritma ini sangat penting dalam menjaga integritas data karena setiap perubahan kecil pada input akan menghasilkan hash yang sangat berbeda. SHA-256 tidak dapat dikembalikan ke bentuk aslinya, sehingga sangat aman untuk kebutuhan verifikasi data dan penyimpanan password (Eastlake & Jones, 2001).

Dalam dunia blockchain, SHA-256 digunakan untuk membangun keamanan jaringan Bitcoin dan memastikan bahwa setiap blok dalam rantai tidak dapat dimodifikasi tanpa mempengaruhi seluruh sistem.

Flask dan Bootstrap

Flask merupakan microframework berbasis Python yang sederhana namun powerful untuk membangun aplikasi web. Flask mendukung pengembangan modular dan memungkinkan integrasi berbagai pustaka tambahan dengan mudah. Sedangkan Bootstrap adalah framework CSS yang membantu dalam membuat tampilan web yang responsif dan menarik. Kombinasi Flask dan Bootstrap memberikan keunggulan dalam membangun aplikasi web yang ringan, cepat, dan user-friendly (Grinberg, 2018).

METODE

Penelitian ini menggunakan metode pengembangan berbasis prototyping, di mana aplikasi dirancang, diuji, dan disempurnakan secara iteratif berdasarkan hasil pengujian awal dan feedback pengguna.

Rancangan Sistem

Rancangan aplikasi melibatkan tiga komponen utama: sistem enkripsi dan dekripsi berbasis RSA, sistem hashing berbasis SHA-256, dan antarmuka pengguna berbasis web. Arsitektur sistem dibuat modular, memisahkan fungsi logika kriptografi dengan komponen antarmuka pengguna, untuk memudahkan pengembangan dan pemeliharaan aplikasi di masa depan.

Pengguna dapat memilih antara operasi enkripsi, dekripsi, atau hashing melalui tampilan web. Setelah memasukkan teks, sistem akan menampilkan hasil proses secara real-time. Selain itu, pengguna dapat mengunduh kunci publik, kunci privat, serta hasil enkripsi, dekripsi, dan hashing dalam bentuk file teks.

Alat dan Bahan

- Bahasa Pemrograman: Python 3.x
- Framework Web: Flask
- Library Tambahan: PyCryptodome (untuk RSA), hashlib (untuk SHA-256), io dan send_file (untuk ekspor file)
- Framework Frontend: Bootstrap 5
- Perangkat Keras: Laptop dengan spesifikasi minimal Intel Core i5, RAM 8 GB, sistem operasi Windows 10
-

Implementasi Algoritma

- RSA: Sistem menghasilkan pasangan kunci RSA (publik dan privat). Pesan yang dimasukkan dienkripsi menggunakan kunci publik dan didekripsi menggunakan kunci privat.
- SHA-256: Algoritma hashing menghasilkan output tetap sepanjang 64 karakter dalam format heksadesimal untuk setiap input yang diberikan. Proses ini satu arah dan tidak memungkinkan dekripsi.

Teknik Pengumpulan Data

Data dikumpulkan melalui pengujian fungsional aplikasi menggunakan berbagai jenis input, termasuk teks pendek, teks panjang, teks dengan karakter khusus, dan kombinasi alfanumerik. Proses pengujian mencatat akurasi hasil enkripsi, dekripsi, dan konsistensi hasil hashing.

Definisi Operasional Variabel

- Keakuratan Enkripsi dan Dekripsi: Dihitung dari tingkat kesamaan antara teks asli dengan hasil dekripsi.
- Konsistensi Hash: Dihitung berdasarkan kesamaan output SHA-256 untuk input yang sama.
- Kecepatan Proses: Dicatat waktu yang dibutuhkan untuk setiap proses enkripsi, dekripsi, dan hashing.

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

JIKUM
JURNAL ILMU KOMPUTER
E-ISSN: 3090-739X
Volume 1, **Number** 2, Nopember 2025
https://doi.org/10.62671/jikum.v1i2.40

Teknik Pengujian dan Analisis

Pengujian dilakukan dengan menggunakan lima variasi input yang berbeda untuk mengevaluasi kinerja aplikasi dan keakuratan hasil kriptografi. Setiap input diuji dengan algoritma RSA dan SHA-256, serta hasilnya dicatat dalam tabel analisis. Pengujian bertujuan untuk memastikan:

- Proses enkripsi menghasilkan ciphertext yang valid dan dapat didekripsi kembali ke pesan asli.
- Proses hashing menghasilkan output tetap dan konsisten untuk input yang sama.
- Fungsi ekspor file dapat menghasilkan file .txt dan .pem yang valid.

HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan sebuah aplikasi web berbasis Python menggunakan Flask, yang mampu melakukan enkripsi dan dekripsi dengan algoritma RSA serta hashing dengan SHA-256. Aplikasi juga mendukung fitur ekspor hasil ke dalam bentuk file. Berikut hasil dan pembahasan dari implementasi dan pengujian aplikasi:

- **Tampilan Antarmuka Pengguna**

Antarmuka aplikasi dirancang sederhana dan responsif menggunakan kombinasi Flask dan Bootstrap. Komponen utama terdiri atas area input pesan, pilihan mode operasi (enkripsi/dekripsi), tombol untuk menjalankan proses, serta fitur unduhan untuk kunci RSA.

Antarmuka ini memudahkan pengguna untuk berinteraksi langsung dengan sistem, mulai dari memasukkan pesan, memilih mode operasi, hingga mengunduh hasil proses kriptografi.

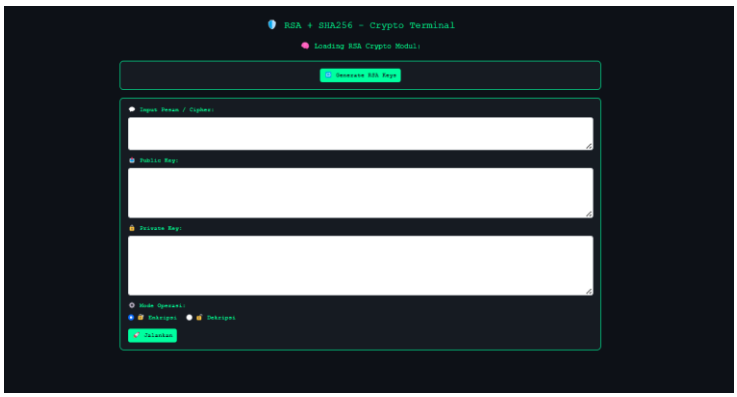

Fig. 1 Tampilan antar muka awal pengguna

- **Proses Enkripsi dan Dekripsi**

Pengguna dapat memilih mode enkripsi untuk mengubah pesan teks menjadi ciphertext menggunakan kunci publik RSA, dan kemudian mendekripsinya kembali menggunakan kunci privat RSA. Hasil dekripsi harus identik dengan input awal untuk memastikan akurasi enkripsi.

Pada contoh, pesan awal "kelompok5" dienkripsi menjadi ciphertext dan kemudian berhasil didekripsi kembali menjadi "kelompok5" tanpa perubahan, menunjukkan keberhasilan fungsi enkripsi dan dekripsi aplikasi.

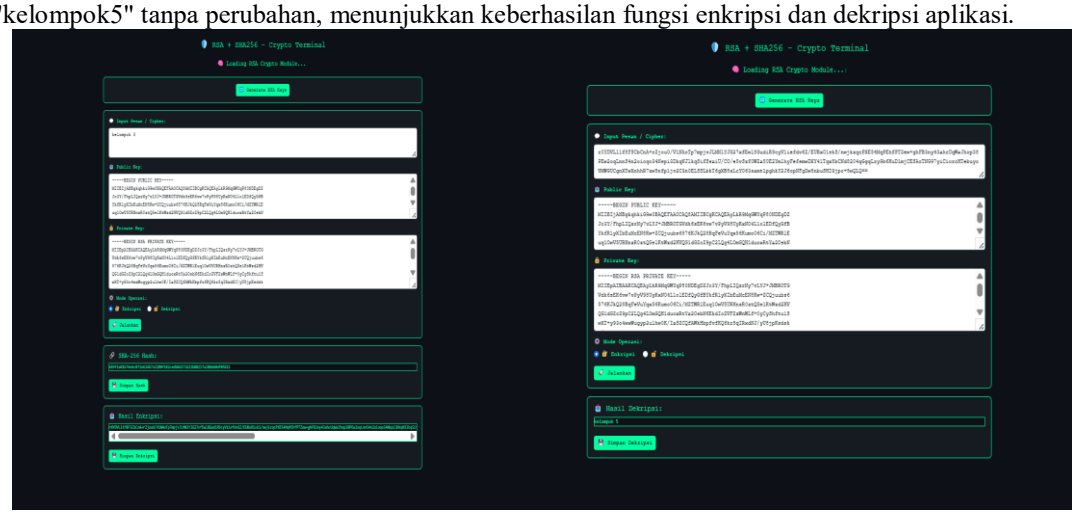

Fig. 2 Prosesn enkripsi pesan dan deskripsi pesan

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

JIKUM
JURNAL ILMU KOMPUTER
E-ISSN: 3090-739X
Volume 1, **Number** 2, Nopember 2025
https://doi.org/10.62671/jikum.v1i2.40

- **Fitur Tambahan**

Aplikasi juga menyediakan fitur untuk menghasilkan nilai hash dari pesan menggunakan algoritma SHA-256. Hash ini berguna untuk verifikasi integritas pesan tanpa perlu mengungkapkan isi aslinya. Sebagai contoh, pesan "kelompok5" menghasilkan nilai hash SHA-256 berupa string heksadesimal sepanjang 64 karakter. Nilai ini tetap konsisten untuk input yang sama, namun akan berubah sepenuhnya jika terjadi sedikit perubahan pada pesan.

Fig. 3 fitu tambahan SHA-256

- **Hasil Pengujian Aplikasi**

Pengujian dilakukan dengan lima variasi input teks untuk menilai akurasi, konsistensi, dan kecepatan dari masing-masing algoritma yang digunakan. Berikut tabel hasil pengujian:

Table 1.Format Pengujian dan Analisis

Input	Algoritma	Hasil Proses	Hasil Dekripsi	Konsistensi
kelompok5	RSA	Ciphertext	kelompok5	Ya
kelompok5	SHA-256	Hash	-	Ya
data123	SHA-256	Hash	-	Ya
Pesan Tes	RSA	Ciphertext	Pesan Tes	Ya

- **Analisis**

Berdasarkan hasil pengujian, aplikasi mampu melakukan enkripsi dan dekripsi dengan RSA secara akurat, serta menghasilkan hash yang konsisten menggunakan SHA-256. Seluruh fungsi berjalan stabil dan file hasil ekspor dapat diunduh dengan format yang sesuai. Antarmuka web juga terbukti responsif dan mudah digunakan.

KESIMPULAN

Penelitian ini berhasil menghasilkan aplikasi web berbasis Python yang mengimplementasikan algoritma RSA untuk enkripsi dan dekripsi, serta SHA-256 untuk hashing. Aplikasi mampu memberikan hasil enkripsi dan dekripsi yang akurat serta konsisten terhadap berbagai variasi input teks, disertai dengan fitur pengunduhan hasil dalam format file. Dengan antarmuka yang sederhana dan responsif, aplikasi ini dapat menjadi alat bantu efektif untuk memahami konsep dasar kriptografi serta meningkatkan kesadaran pentingnya keamanan data di era digital.

REFERENSI

Az Zahra, A., & Rachmawati, R. (2024). Implementasi QR Code dengan Algoritma SHA-256 dan RSA yang Ditingkatkan untuk Autentikasi Dokumen Digital. *Jurnal EurekaMatika, 12*(1), 11-22. https://ejournal.upi.edu/index.php/JEM/article/view/67161

Hutagalung, J., Ramadhan, P. S., & Sihombing, S. J. (2023). Keamanan Data Menggunakan Secure Hashing Algorithm (SHA)-256 dan Rivest Shamir Adleman (RSA) pada Digital Signature. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTI&IK), 10*(6), 1234-1245. https://jtiik.ub.ac.id/index.php/jtiik/article/view/7319

Irawan, C., & Rachmawanto, E. H. (2023). Implementasi Kriptografi dengan Metode RSA untuk Keamanan Data. *Prosiding Seminar Nasional Teknologi Informasi & Ilmu Komputer (SEMASTER), 2*(1), 97-105. https://journal.unilak.ac.id/index.php/Semaster/article/view/18461

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

JIKUM
JURNAL ILMU KOMPUTER

E-ISSN: 3090-739X
Volume 1, **Number** 2, Nopember 2025
https://doi.org/10.62671/jikum.v1i2.40

Mahfud, I., & Utomo, P. H. (2021). Implementasi Sistem Kriptografi RSA Signature dengan SHA-256 pada Mekanisme Autentikasi REST API. *Jurnal Teknoka, 21*(1), 1-10. https://journal.uhamka.ac.id/index.php/teknoka/article/view/10239

Nainggolan, S. (2022). Implementasi Algoritma SHA-256 pada Aplikasi Duplicate Image Checker. *RESOLUSI: Rekayasa Teknik Informatika dan Informasi, 2*(5), 201-213. https://djournals.com/resolusi/article/view/368

This is an Creative Commons License This work is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.