MOTEKAR: Jurnal Multidisiplin Teknologi dan Arsitektur
E-ISSN: 3025-227X P-ISSN: 3025-2288
Vol. 2 No. 2 November 2024

Implementasi Algoritma Kriptografi Modern melalui Google Colab: Studi Kasus AES dan RSA

Adzkia Nur Nasution[1] Ardilla Syahfitri[2] Zulfahmi Indra[3]
Program Studi Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Medan, Kota Medan, Provinsi Sumatera Utara, Indonesia[1,2,3]
Email: adzkiaaanur41@gmail.com[1] ardillasyahfitrilbs@gmail.com[2]
zulfahmi.indra@unimed.ac.id[3]

Abstrak

Di era digital yang semakin cepat, enkripsi modern memainkan peran penting dalam menjaga keamanan dan privasi data, terutama di lingkungan teknologi yang semakin saling terhubung dan kompleks. Dengan semakin banyaknya ancaman dunia maya, termasuk serangan malware, phishing, ransomware, dan upaya peretasan data yang semakin canggih, keamanan informasi telah menjadi prioritas utama. Algoritma enkripsi modern, seperti Advanced Encryption Standard (AES) dan Rivest-Shamir-Adleman (RSA) memastikan data terlindungi baik dari segi kerahasiaan, integritas, dan otentikasi. AES dikenal luas karena kecepatan dan efisiensinya dalam mengenkripsi data dengan ukuran blok tetap, sedangkan RSA, salah satu algoritma kriptografi asimetris paling terkenal, menggunakan kunci publik dan pribadi serta memberikan keamanan (Anwar, et al, 2018). Keduanya telah terbukti menggagalkan berbagai ancaman dan banyak digunakan dalam aplikasi sehari-hari seperti komunikasi yang aman, transaksi keuangan online, dan melindungi data sensitif. Seiring berkembangnya teknologi, platform seperti Google Colab memberikan solusi yang efisien dan mudah diakses untuk menerapkan dan menguji algoritma kriptografi ini. Google Colab memungkinkan peneliti, pengembang, dan profesional melakukan eksperimen kriptografi tanpa perangkat keras yang mahal, cukup dengan memanfaatkan kekuatan komputasi yang disediakan oleh Google Cloud. Artikel ini menjelaskan cara mengimplementasikan algoritma AES dan RSA menggunakan Google Colab, mulai dari menginstal perpustakaan yang diperlukan hingga penjelasan kode mendetail dan menganalisis hasil enkripsi dan dekripsi data. Selain itu, diskusi ini mencakup manfaat utama penggunaan platform cloud untuk pengembangan dan pengujian kriptografi seperti skalabilitas, fleksibilitas, dan ketersediaan sumber daya komputasi yang besar. Namun, hal ini tidak menutup kemungkinan adanya tantangan, seperti risiko perlindungan data di google cloud dan kemungkinan ketergantungan pada infrastruktur pihak ketiga. Selain itu, artikel ini mengeksplorasi kemungkinan masa depan kriptografi dalam platform cloud termasuk integrasi teknologi blockchain dan komputasi kuantum yang diharapkan dapat merevolusi keamanan digital di masa depan. Pendekatan komprehensif ini memungkinkan artikel ini memberikan pemahaman yang lebih luas tentang pentingnya enkripsi dalam menjaga keamanan data di era digital dan bagaimana komputasi awan dapat mendukung perkembangannya secara efektif dan efisien.

Kata Kunci: Kriptografi Modern, AES, RSA, Google Colab, Cloud Computing, Python, Enkripsi, Dekripsi

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

PENDAHULUAN

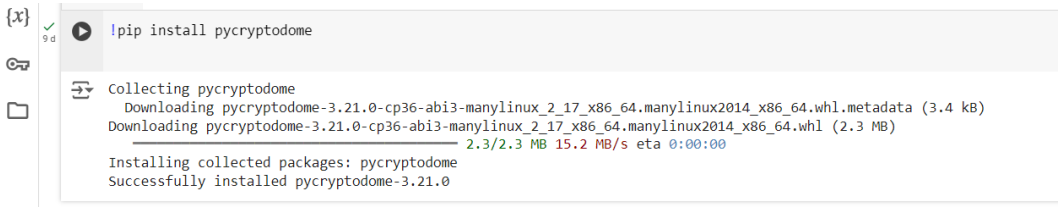
Di era digital saat ini, informasi digital sering menjadi sasaran serangan siber, dan kesadaran akan pentingnya keamanan data semakin meningkat. Cara efektif untuk melindungi data Anda adalah dengan menggunakan enkripsi algoritmik. Algoritma seperti AES simetris (Standar Enkripsi Lanjutan) dan RSA asimetris (Rivest-Shamir-Adleman) adalah dua teknik enkripsi yang paling umum digunakan untuk menjaga keamanan data. AES atau Advanced Encryption Standard adalah algoritma simetris yang menggunakan kunci yang sama untuk enkripsi dan dekripsi. Artinya data yang dienkripsi dengan AES hanya dapat didekripsi oleh

pemilik kunci. Algoritma ini banyak digunakan untuk melindungi data dalam jumlah besar karena kecepatan dan efisiensinya. RSA di sisi lain adalah algoritma kriptografi kunci publik yang menggunakan pasangan kunci privat dan publik. Dengan RSA, data yang dienkripsi dengan kunci publik hanya dapat didekripsi dengan kunci privat, dan sebaliknya, data yang dienkripsi dengan kunci privat dapat diverifikasi dengan kunci publik. Hal ini menjadikan RSA ideal untuk aplikasi yang memerlukan tingkat keamanan tinggi, seperti mengirim pesan rahasia atau tanda tangan digital. Keandalan kedua algoritma ini telah terbukti dalam menjaga kerahasiaan dan integritas data, yang merupakan hal penting di era digital. Namun, seiring dengan semakin kompleksnya serangan siber, penting untuk terus meningkatkan metode dan pendekatan keamanan, termasuk penerapan enkripsi. Selain itu, keberadaan platform berbasis cloud seperti Google Colab membuat penerapan enkripsi menjadi lebih mudah dan efisien. Google Colab menyediakan lingkungan pemrograman berbasis Python di cloud yang sangat berguna bagi peneliti dan praktisi untuk mengembangkan aplikasi kriptografi tanpa memasang perangkat keras yang rumit. Google Colab membuat eksperimen dan pengembangan kriptografi menjadi cepat dan mudah, bahkan bagi pengguna tanpa infrastruktur komputasi canggih, dengan menyediakan akses ke sumber daya komputasi yang luas dan mendukung berbagai perpustakaan kriptografi. Secara keseluruhan, teknologi enkripsi yang kuat seperti AES dan RSA, dikombinasikan dengan platform pengembangan modern seperti Google Colab, membantu menjaga keamanan data di era digital yang semakin rentan terhadap ancaman dunia maya (Ananda & Lukman, 2022).

METODE PENELITIAN

Penelitian ini bertujuan untuk mengimplementasikan dua algoritma enkripsi yaitu AES (Advanced Encryption Standard) dan RSA (Rivest-Shamir-Adleman) menggunakan platform Google Collab. Tahap Penelitian ini mengikuti beberapa langkah sebagai berikut (Utomo, et al, 2023):

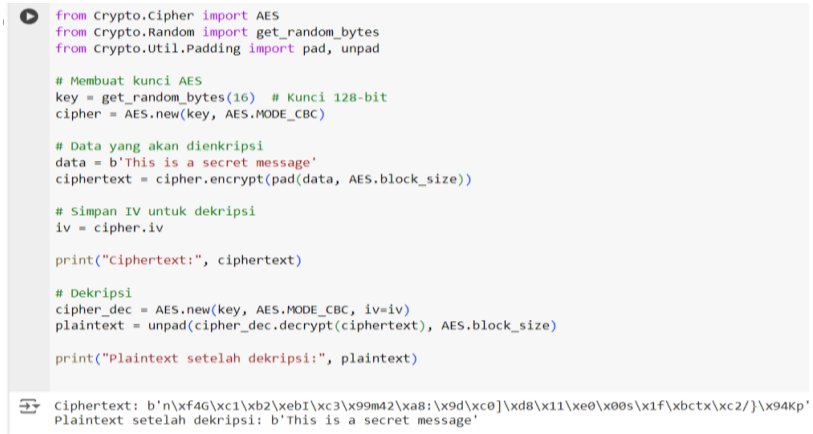
1. Instalasi Pustaka Kriptografi. Penelitian ini menggunakan perpustakaan Pycryptodome yang menyediakan implementasi lengkap AES dan RSA, perpustakaan diinstal menggunakan perintah berikut:



```
!pip install pycryptodome

Collecting pycryptodome
  Downloading pycryptodome-3.21.0-cp36-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (3.4 kB)
Downloading pycryptodome-3.21.0-cp36-abi3-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.3/2.3 MB 15.2 MB/s eta 0:00:00
Installing collected packages: pycryptodome
Successfully installed pycryptodome-3.21.0
```

2. Implementasi AES : AES adalah algoritma enkripsi simetris yang menggunakan kunci untuk proses enkripsi dan dekripsi. Contoh penerapan AES di Google Colab adalah:



```
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes
from Crypto.Util.Padding import pad, unpad

# Membuat kunci AES
key = get_random_bytes(16)  # Kunci 128-bit
cipher = AES.new(key, AES.MODE_CBC)

# Data yang akan dienkripsi
data = b'This is a secret message'
ciphertext = cipher.encrypt(pad(data, AES.block_size))

# Simpan IV untuk dekripsi
iv = cipher.iv

print("Ciphertext:", ciphertext)

# Dekripsi
cipher_dec = AES.new(key, AES.MODE_CBC, iv=iv)
plaintext = unpad(cipher_dec.decrypt(ciphertext), AES.block_size)

print("Plaintext setelah dekripsi:", plaintext)

Ciphertext: b'n\xf4G\xc1\xb2\xebI\xc3\x99m42\xa8:\x9d\xc0]\xd8\x11\xe0\x00s\x1f\xbctx\xc2/}\x94Kp'
Plaintext setelah dekripsi: b'This is a secret message'
```

3. Implementasi RSA : RSA adalah algoritma asimetris yang menggunakan pasangan kunci publik-pribadi. Berikut contoh implementasi RSA di Google Colab:

```
from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP
from Crypto.Random import get_random_bytes

# Membuat pasangan kunci RSA
key = RSA.generate(2048)
public_key = key.publickey()

# Membuat cipher dengan kunci publik
cipher_rsa = PKCS1_OAEP.new(public_key)

# Data yang akan dienkripsi
data = b'This is a secret message'
ciphertext = cipher_rsa.encrypt(data)

print("Ciphertext:", ciphertext)

# Dekripsi dengan kunci privat
cipher_rsa_dec = PKCS1_OAEP.new(key)
plaintext = cipher_rsa_dec.decrypt(ciphertext)

print("Plaintext setelah dekripsi:", plaintext)
```

```
Ciphertext: b'\xa3\xbc3\x04\x90L\xe0\x98\xa4\x1f\xfa9\x8b\x07A\x05D!\xdd\xe0\x8e\x80[\xb9\xbe\xac\xce\x9a\xc8\x1d\xf9\x91\xaaw\x1f\xf9L\xa1J\xc8\xcf\xf6\xce\xc6\x01\xde6\xceg\x13\x17\
Plaintext setelah dekripsi: b'This is a secret message'
```

HASIL PENELITIAN DAN PEMBAHASAN

Hasil Implementasi AES

Saat menjalankan kode di atas penulis akan mendapatkan ciphertext sebagai hasilnya. Ini adalah bentuk pesan rahasia terenkripsi yang dihasilkan oleh algoritma Advanced Encryption Standard (AES). Ciphertext pada dasarnya adalah pesan yang telah dienkripsi sedemikian rupa sehingga tidak dapat dibaca atau dipahami tanpa kunci dekripsi yang sesuai. Tujuan metode enkripsi ini adalah untuk melindungi kerahasiaan pesan dari akses tidak sah, terutama dalam situasi di mana data dikirimkan melalui jaringan yang tidak aman. Operasi dekripsi kemudian dilakukan menggunakan kunci yang sama. Hal ini karena AES merupakan algoritma simetris. Artinya kunci yang digunakan untuk enkripsi juga digunakan untuk dekripsi (Arieka & Mukti, 2023). Proses dekripsi berhasil mengubah ciphertext kembali menjadi plaintext atau pesan asli sebelum enkripsi, tanpa kehilangan atau distorsi data. Hal ini menunjukkan keandalan dan keakuratan AES dalam menjaga integritas data selama proses enkripsi dan dekripsi (Alfani, et al, 2024).

Hasil Implementasi RSA

Implementasi RSA menunjukkan cara kerja mekanisme kriptografi kunci publik dalam melindungi data. Algoritma ini menggunakan kunci publik untuk mengenkripsi pesan dan kunci privat untuk mendekripsinya. Artinya, siapapun yang memiliki kunci publik dapat mengenkripsi pesan, namun hanya seseorang yang memiliki kunci privat yang dapat mendekripsinya. Hal ini membuat RSA sangat berguna dalam memastikan bahwa hanya penerima yang berwenang yang dapat membaca pesan, dan dalam situasi di mana distribusi kunci harus dilakukan melalui saluran yang mungkin tidak sepenuhnya aman. RSA menawarkan keuntungan signifikan dalam autentikasi dan pertukaran kunci (Pratama & Zakaria, 2022). Dalam hal otentikasi, RSA memungkinkan penerima memverifikasi keaslian pesan atau dokumen yang mereka terima. Misalnya, pengirim menggunakan kunci pribadinya untuk mengenkripsi pesan dan penerima memverifikasi keaslian pesan dengan mendekripsinya menggunakan kunci publik pengirim, sehingga memverifikasi bahwa pesan tersebut benar-benar. Penulis dapat membuktikan bahwa pesan tersebut berasal dari pengirim yang sah. Meskipun RSA sangat kuat dalam hal keamanan, algoritma ini umumnya lambat dibandingkan dengan algoritma simetris seperti AES. Hal ini dikarenakan memerlukan operasi matematika yang sangat kompleks, terutama dalam mengelola bilangan prima besar yang digunakan untuk membentuk kunci publik dan privat. Operasi pembagian eksponensial dan pembagian modulo pada proses enkripsi dan dekripsi RSA memerlukan waktu komputasi yang lebih lama, terutama bila digunakan untuk mengenkripsi atau mendekripsi data dalam

jumlah besar (Saputra, et al, 2022). Namun, RSA masih menawarkan keuntungan strategis yang penting. RSA sering digunakan bersama dengan algoritma simetris seperti AES melalui skema hybrid. Misalnya, RSA digunakan untuk mengenkripsi dan mendistribusikan kunci AES, dan AES kemudian digunakan untuk mengenkripsi dan mendekripsi pesan. Skema ini memanfaatkan kecepatan AES dan keamanan RSA dalam satu proses. RSA juga memberikan solusi ideal dalam situasi di mana pertukaran kunci harus terjadi melalui jaringan publik, atau dalam sistem yang memerlukan otentikasi digital. Sertifikat Digital dan Tanda Tangan Digital. Meskipun RSA kurang efisien dalam menangani data dalam jumlah besar, keunggulannya dalam otentikasi, pertukaran kunci yang aman, dan tingkat keamanan yang tinggi menjadikannya komponen penting dalam banyak aplikasi keamanan digital modern (Wahyudi & Ristian, 2024).

Pembahasan

Ada banyak manfaat menggunakan Google Collab sebagai platform untuk mengimplementasikan algoritma kriptografi. Pertama yaitu kemudahan akses memberikan kemampuan untuk menjalankan kode arbitrer tanpa memerlukan konfigurasi perangkat keras lokal atau instalasi perangkat lunak khusus cukup koneksi Internet dan navigator. Selain itu karena Collab berbasis cloud, Colab memiliki efisiensi komputasi yang tinggi dan dapat menangani proses enkripsi dan dekripsi tingkat lanjut seperti algoritme RSA, sehingga memungkinkan penggunaan GPU dengan kecepatan lebih tinggi (Wahdini, et al, 2021). Collab juga memungkinkan banyak orang mengerjakan proyek yang sama pada waktu yang sama, sehingga mempermudah kolaborasi dengan orang yang berbeda. Namun, ada beberapa tantangan yang perlu dipertimbangkan, termasuk risiko keamanan data, karena menyimpan kunci enkripsi di cloud dapat membuat Anda rentan terhadap serangan siber. Selain itu ketergantungan pada konektivitas internet merupakan hambatan, karena pengguna tidak akan dapat mengaksesnya jika terjadi kegagalan jaringan atau server. Dan collab memiliki batasan penggunaan berikut untuk layanan gratisnya Waktu penggunaan dan akses GPU. Ini bisa menjadi kendala untuk proyek-proyek yang memerlukan banyak pekerjaan komputasi (Hidayat, 2024).

KESIMPULAN

Penelitian ini mengimplementasikan algoritma AES dan RSA di Google Collab. Hasil yang ditampilkan di Google Collab adalah platform efisien dan mudah digunakan untuk mengembangkan dan menguji algoritma AES telah terbukti efisien dalam enkripsi simetris dan memberikan kinerja yang cepat dan aman saat memproses data dalam jumlah besar. RSA sebaliknya, lebih lambat karena operasi matematika yang rumit, namun menawarkan keunggulan dalam pertukaran kunci dan autentikasi yang penting untuk keamanan komunikasi. Tantangan utamanya adalah risiko keamanan data di Google cloud, terutama dalam hal penyimpanan kunci kriptografi, yang memerlukan perhatian khusus untuk menjaga kerahasiaan dan keamanan kunci tersebut.

DAFTAR PUSTAKA

Alfani, M. R., Furqan, M., & Nasution, Y. R. (2024). Pengamanan Data Teks Menggunakan Metode Digital Signature Algorithm (Dsa) Dan Advanced Encryption Standard (Aes). Journal Of Science And Social Research, 7(1), 301-306.

Ananda, S. P., & Lukman, S. (2022). Analisa Metode Kriptografi Modern Advance Encryption Standard (AES) 128 Bit dalam Mengenkripsi dan Mendekripsi File Dokumen Digital: Array. Jurnal Ilmiah Komputasi, 21(3), 333-344.

Anwar, N., Munawwar, M., Abduh, M., & Santosa, N. B. (2018). Komparatif performance model keamanan menggunakan metode Algoritma AES 256 bit dan RSA. Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 2(3), 783-791.

Arieska, A. E. B., & Mukti, F. S. (2023). Pemanfaatan One-Time Password dan Algoritma Advanced Encryption Standard dalam Sistem Login Internet Kampus. G-Tech: Jurnal Teknologi Terapan, 7(4), 1262-1271.

Hidayat, R. (2024). Algoritma kriptografi modern: RSA, AES, dan ECC. Pusat Pengembangan Data dan Teknologi. https://p2dpt.uma.ac.id/2024/07/23/algoritma-kriptografi-modern-rsa-aes-dan-ecc/

Pratama, S. A., & Zakaria, H. (2022). Penerapan Ilmu Kriptografi untuk Keamanan Informasi Konsumen Menggunakan Algoritma Vigenere Cipher dan RC6 Berbasis Android:(Studi Kasus: PT BFI Finance Indonesia Tbk). Scientia Sacra: Jurnal Sains, Teknologi dan Masyarakat, 2(2), 604-619.

Saputra, R. P., Wahyudi, J., & Jumadi, J. (2022). Comparative analysis of the blowfish algorithm and the des algorithm in the document file encryption and decryption process. Jurnal Komputer, Informasi dan Teknologi, 2(2), 605-612.

Utomo, N. P., Fahriani, N., & Amirul, M. (2023). Implementasi Kriptografi Dengan Metode RSA Untuk Keamanan Data Pada Email Berbasis PHP. In SEMASTER: Seminar Nasional Teknologi Informasi & Ilmu Komputer (Vol. 2, No. 1, pp. 97-105).

Wahdini, S. V., Hartama, D., & Kirana, I. O. (2021). Pengamanan Data Pelanggan dan Penjualan Menggunakan Implementasi Algoritma Kriptografi. Journal of Informatics Management and Information Technology, 1(3), 101-107.

Wahyudi, R., & Ristian, U. (2024). Pengamanan Tanda Tangan Digital Dalam QR Code Berbasis Website Menggunakan Metode RSA (Studi Kasus: Kantor Desa Parit Baru). JUPITER: Jurnal Penelitian Ilmu dan Teknologi Komputer, 16(1), 181-193.