***Performance Evaluation of Video and Image Encryption and Decryption using AES and RSA Algorithms with Python***

Evaluasi Performa Enkripsi dan Dekripsi Video dan Gambar dengan Algoritma AES dan RSA dengan Python

**Dea Reigina1), Naufal Rafid Muhammad Faddila2)**

1,2) Informatika, Universitas Pembangunan Nasional Veteran Yogyakarta, Indonesia

*Email:* 1*123220020@student.upnyk.ac.id,* 2*123220052@student.upnyk.ac.id*

| ***Informasi Artikel*** | ***Abstract*** |
| --- | --- |
| *Received: October 2024*  *Revised: -*  *Accepted: October 2024*  *Published: October 2024* | *This study aims to compare the performance of two algorithms, namely AES (Advanced Encryption Standard) and RSA (Rivest Shamir Adleman), in encrypting and decrypting image and video files. In this research, we tested both algorithms using the Python programming language. We measured the time needed for encryption and decryption, memory usage, throughput, and CPU usage. We evaluated which algorithm is more efficient for different sizes of image and video files. AES is proven to be faster than RSA for large file encryption and decryption, which means this algorithm is good for projects needing high speed and processing of large data. On the other hand, RSA, although slower, is more stable in CPU and memory usage, making it better for smaller data. AES also shows higher throughput and more efficient CPU usage compared to RSA. This study provides a performance evaluation of AES and RSA in securing image and video files. Hope this is helpful for people looking for references to choose the encryption algorithm that suits their application needs, whether they prioritize speed or resource stability.* |
|  | **Abstrak** |
| *Keywords:cryptography; AES; RSA; python; algorithm*  Kata kunci: kriptografi; AES; RSA; python; algoritma | Tujuan penelitian ini adalah untuk membandingkan performa dua algoritma enkripsi, yaitu AES (Advanced Encryption Standard) dan RSA (Rivest-Shamir-Adleman), dalam mengenkripsi dan mendekripsi gambar dan video. Kami menguji kedua algoritma dengan menggunakan bahasa pemrograman Python. Kami menghitung waktu yang dibutuhkan untuk program enkripsi dan dekripsi, juga penggunaan memori, throughput, serta penggunaan CPU. Evaluasi dilakukan untuk mengetahui algoritma mana yang lebih efisien untuk berbagai ukuran inputan berjenis gambar dan video. AES terbukti lebih cepat daripada RSA dalam enkripsi dan dekripsi file yang besar, artinya algoritma ini baik untuk projek yang membutuhkan kecepatan tinggi dan pengolahan data yang besar. Lain halnya dengan RSA, meskipun lebih lambat, hasil evaluasinya dari penggunaan CPU dan memori, lebih cocok untuk data berukuran kecil. AES juga menunjukkan throughput yang lebih tinggi dan penggunaan CPU yang lebih efisien dibandingkan RSA. Penelitian ini memberikan evaluasi performa antara AES dan RSA dalam mengamankan file gambar dan video. Harapannya bermanfaat bagi orang yang sedang mencari referensi untuk memilih algoritma enkripsi yang sesuai dengan kebutuhan aplikasi mereka, baik yang mengutamakan kecepatan maupun stabilitas penggunaan sumber daya perangkat. |
|  |  |

# Pendahuluan

Keamanan sebuah data adalah salah satu aspek yang penting dalam dunia digital saat ini, terutama dengan meningkatnya ancaman terhadap privasi dan integritas informasi. Dalam penelitian ini, kriptografi menjadi salah satu metode untuk melindungi data dari pihak yang tidak diinginkan.

Kriptografi itu sendiri adalah ilmu yang mempelajari dan menjaga kerahasiaan dan keamanan informasi saat dikirim, dan membuat pesan menjadi bentuk yang tidak bisa dibaca oleh pihak lain. Dalam kriptografi, ada tiga konsep dasar yang penting, yaitu enkripsi, dekripsi, dan kunci. Tujuan utamanya adalah memastikan kunci tetap rahasia dan mengubah teks asli (plaintext) menjadi teks terenkripsi (ciphertext), inilah yang menyebabkan pihak yang tidak diinginkan tidak bisa membaca pesan atau informasi terenkripsi.

Dalam penelitian ini, kami melakukan evaluasi performa enkripsi dan dekripsi gambar serta video pada algoritma AES (Advanced Encryption Standard) dan RSA (Rivest-Shamir-Adleman) dengan pemrograman Python. Tujuan utama dari evaluasi ini adalah untuk mengukur kinerja algoritma dalam proses enkripsi dan dekripsi serta sumber daya komputasi yang digunakan selama proses tersebut. Dengan membandingkan kedua algoritma ini, diharapkan dapat memberikan wawasan mengenai keunggulan masing-masing algoritma dalam konteks kecepatan dan keamanan, serta memberikan rekomendasi penggunaan yang sesuai berdasarkan hasil evaluasi performa yang dilakukan.

# Metode

Metode dan tahapan pada penelitian ini adalah yang pertama mengidentifikasi masalah, kemudian pengumpulan data, melakukan enkripsi, hasil enkripsi, kemudian melakukan dekripsi, hasil dekripsi dan mengambil kesimpulan. Berikut alur penelitiannya dibawah ini.



**Gambar 1**. Alur Penelitian

Berikut penjelasan dan alur mengenai algoritma AES dan RSA yang diterapkan pada penelitian , yaitu:

1. Algoritma AES

AES yang digunakan pada penelitian ini adalah algoritma enkripsi blok yang bekerja pada blok data sebesar 16 byte. Program ini menggunakan AES dengan mode CBC (Cipher Block Chaining), di mana setiap blok hasil enkripsi tergantung pada blok sebelumnya, membuat hasil lebih aman. Dalam mode CBC, setiap blok dienkripsi setelah di-XOR dengan blok sebelumnya, sehingga mencegah pengulangan pola pada data yang dienkripsi.

Pada setiap enkripsi, IV (Initialization Vector) digunakan untuk memulai proses enkripsi sehingga output menjadi unik, bahkan jika input dan kunci yang digunakan sama. Program ini menyimpan IV di awal file terenkripsi agar proses dekripsi bisa mengaksesnya.Jika ukuran data asli tidak merupakan kelipatan 16 byte, program menggunakan padding (penambahan data ke akhir blok) agar data dapat dibagi menjadi blok 16 byte. Program ini menggunakan pad dari pustaka Crypto.Util.Padding.

Untuk proses enkripsi, program membaca file gambar dan video sebagai data biner, lalu mengenkripsi data tersebut menggunakan AES dengan kunci 32 byte (AES-256). Kunci ini dihasilkan secara acak menggunakan get\_random\_bytes. Setelah data dienkripsi, program menyimpan IV di awal file terenkripsi diikuti oleh hasil enkripsi (ciphertext). Program juga mengukur waktu enkripsi, penggunaan memori, throughput (jumlah data terenkripsi per detik), dan penggunaan CPU.

Kemudian untuk proses dekripsi, program membaca IV dari awal file terenkripsi, lalu mendekripsi sisa file menggunakan kunci yang sama. Setelah proses dekripsi, program melakukan unpadding pada hasil dekripsi untuk menghapus padding yang ditambahkan saat enkripsi. Program juga mencatat waktu dekripsi, penggunaan memori, throughput, dan penggunaan CPU. Kemudian untuk menilai kinerja algoritma, program menggunakan beberapa metrik, termasuk waktu eksekusi, penggunaan memori maksimum, throughput (MB/s), dan penggunaan CPU, untuk menganalisis performa enkripsi dan dekripsi pada data.

1. Algoritma RSA

Algoritma RSA yang digunakan dalam program ini adalah algoritma enkripsi asimetris, yang berarti menggunakan pasangan kunci yang berbeda untuk enkripsi dan dekripsi. kunci publik untuk enkripsi dan kunci privat untuk dekripsi.

Untuk alur atau cara kerja dari algoritmanya sendiri yaitu, yang pertama adalah pembangkitan kunci RSA dengan fungsi generate\_rsa\_keys untuk membuat pasangan kunci RSA 2048-bit. Kunci publik (public\_key) digunakan untuk enkripsi, sedangkan kunci privat (private\_key) digunakan untuk dekripsi.

Kemudian proses enkripsi dilakukan pada fungsi encrypt\_image, program membaca data gambar dan video dalam bentuk biner. Karena RSA memiliki batasan panjang data yang bisa dienkripsi dalam satu kali operasi (disebabkan oleh padding PKCS1\_OAEP), program membagi data menjadi chunks atau potongan-potongan berukuran 190 byte, yang merupakan ukuran maksimum untuk kunci RSA 2048-bit dengan padding PKCS1\_OAEP. Setiap potongan dienkripsi menggunakan kunci publik dengan metode PKCS1\_OAEP dan ditambahkan ke ciphertext, yang merupakan hasil enkripsi keseluruhan gambar dan video. Program menyimpan hasil enkripsi dalam file, dan mencatat waktu enkripsi, penggunaan memori maksimum, throughput (jumlah data yang diproses per detik), dan penggunaan CPU.

Kemudian untuk proses dekripsi dilakukan pada fungsi decrypt\_image, program membaca file gambar dan video yang telah terenkripsi. Data enkripsi dibagi menjadi potongan-potongan berukuran 256 byte (ukuran yang dapat di-dekripsi dengan RSA 2048-bit dan padding PKCS1\_OAEP) untuk setiap kali dekripsi. Setiap potongan didekripsi menggunakan kunci privat RSA, dan hasilnya disatukan kembali untuk mendapatkan data gambar dan video asli. Program kemudian menyimpan data yang telah didekripsi kembali menjadi file gambar asli. Selain itu, program mencatat waktu dekripsi, penggunaan memori maksimum, throughput, dan penggunaan CPU.

Kemudian untuk menganalisis kinerja algoritma, program mengukur beberapa metrik kinerja selama enkripsi dan dekripsi, yaitu waktu eksekusi untuk mengetahui seberapa cepat proses enkripsi dan dekripsi dilakukan. Penggunaan memori maksimum selama proses. Throughput untuk menghitung kecepatan enkripsi/dekripsi per detik. Dan terakhir yaitu penggunaan CPU sebelum dan sesudah operasi.

## Pengumpulan Data

Penelitian ini menggunakan data yang terdiri dari gambar dan video dengan berbagai ukuran file yang bervariasi. Variasi ukuran ini bertujuan untuk mengevaluasi performa algoritma enkripsi dan dekripsi AES dan RSA dalam berbagai kondisi, serta memastikan keandalan algoritma dalam menangani beragam ukuran file. Data dikumpulkan dengan harapan dapat mengukur kecepatan proses, penggunaan memori, throughput, dan penggunaan CPU saat algoritma diterapkan pada file multimedia yang berbeda.

**2.1.1 Data Uji**

Data uji mencakup beberapa file gambar dan video dengan ukuran berbeda, yang digunakan untuk mengukur performa algoritma enkripsi dan dekripsi. Adapun rincian ukuran file yang digunakan adalah sebagai berikut:

| **No** | **Nama File** | **Ukuran File (KB)** | **Tampilan** |
| --- | --- | --- | --- |
| 1 | 1.png | 152 |  |
| 2 | 2.png | 290 |  |
| 3 | 3.png | 987 |  |
| 4 | 4.png | 1.384 |  |
| 5 | 5.png | 2.393 |  |
| 6 | 1.mp4 | 1.996 |  |
| 7 | 2.mp4 | 2.299 |  |
| 8 | 3.mp4 | 3.900 |  |
| 9 | 4.mp4 | 5.135 |  |
| 10 | 5.mp4 | 8.632 |  |

**Tabel 1**. Data Uji

Dengan variasi ukuran ini, diharapkan algoritma enkripsi dan dekripsi dapat diuji secara komprehensif terhadap kecepatan dan efisiensi sumber daya saat memproses file yang berbeda.

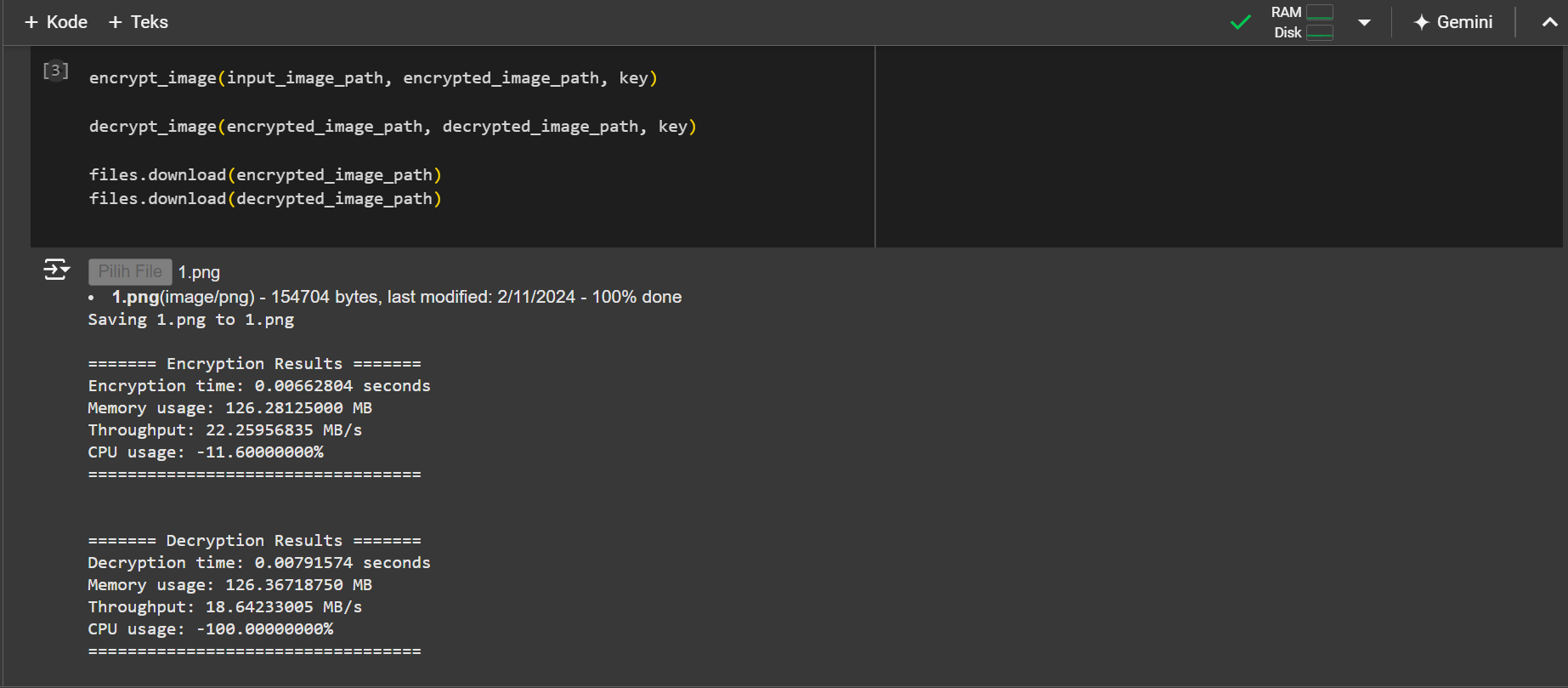
# Hasil dan Pembahasan

Program yang digunakan untuk menganalisis kinerja algoritma AES dan RSA pada penelitian ini menggunakan bahasa pemrograman python yang dikerjakan di platform Google Colab. Program ini dijalan pada perangkat dengan sistem operasi Windows 11 dengan prosesor Intel i7-1255U, 12 CPUs dan 1.7GHz. Dalam program ini analisis kinerja algoritma proses enkripsi dan dekripsi didasarkan pada 4 hal yaitu, waktu eksekusi, penggunaan memori, throughput, dan penggunaan CPU. Hasil yang didapatkan dari pengujian ini bisa berbeda, tergantung dari spesifikasi perangkat yang digunakan.

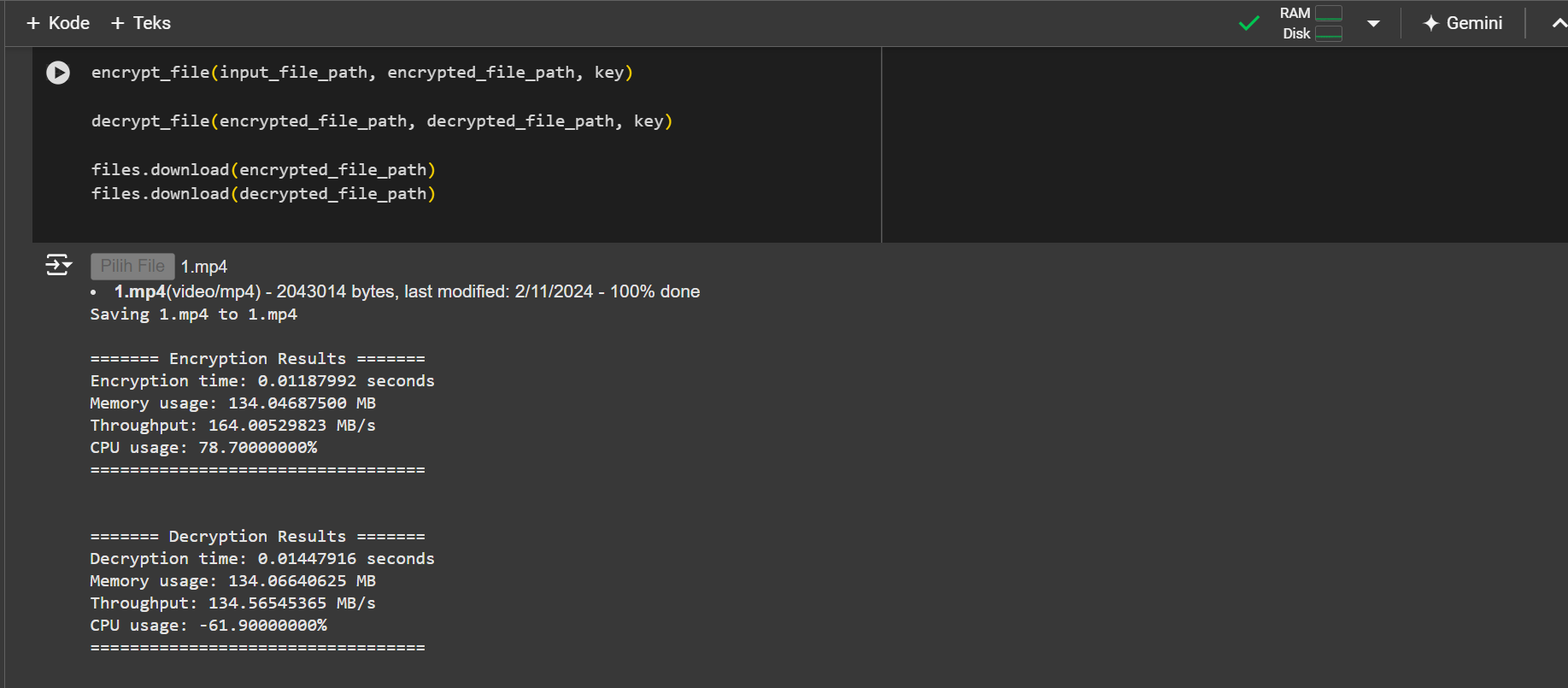
Berikut langkah - langkah dalam pengujian kinerja algoritma AES dan RSA:

1. Untuk sampel gambar, menggunakan 5 file dengan format PNG dengan ukuran yang berbeda. Kemudian untuk sampel video menggunakan 5 file dengan format MP4 dengan ukuran yang berbeda.
2. Melakukan enkripsi dan dekripsi pada setiap file.
3. Mengukur kinerja algoritma pada setiap file dalam proses enkripsi dan dekripsi.

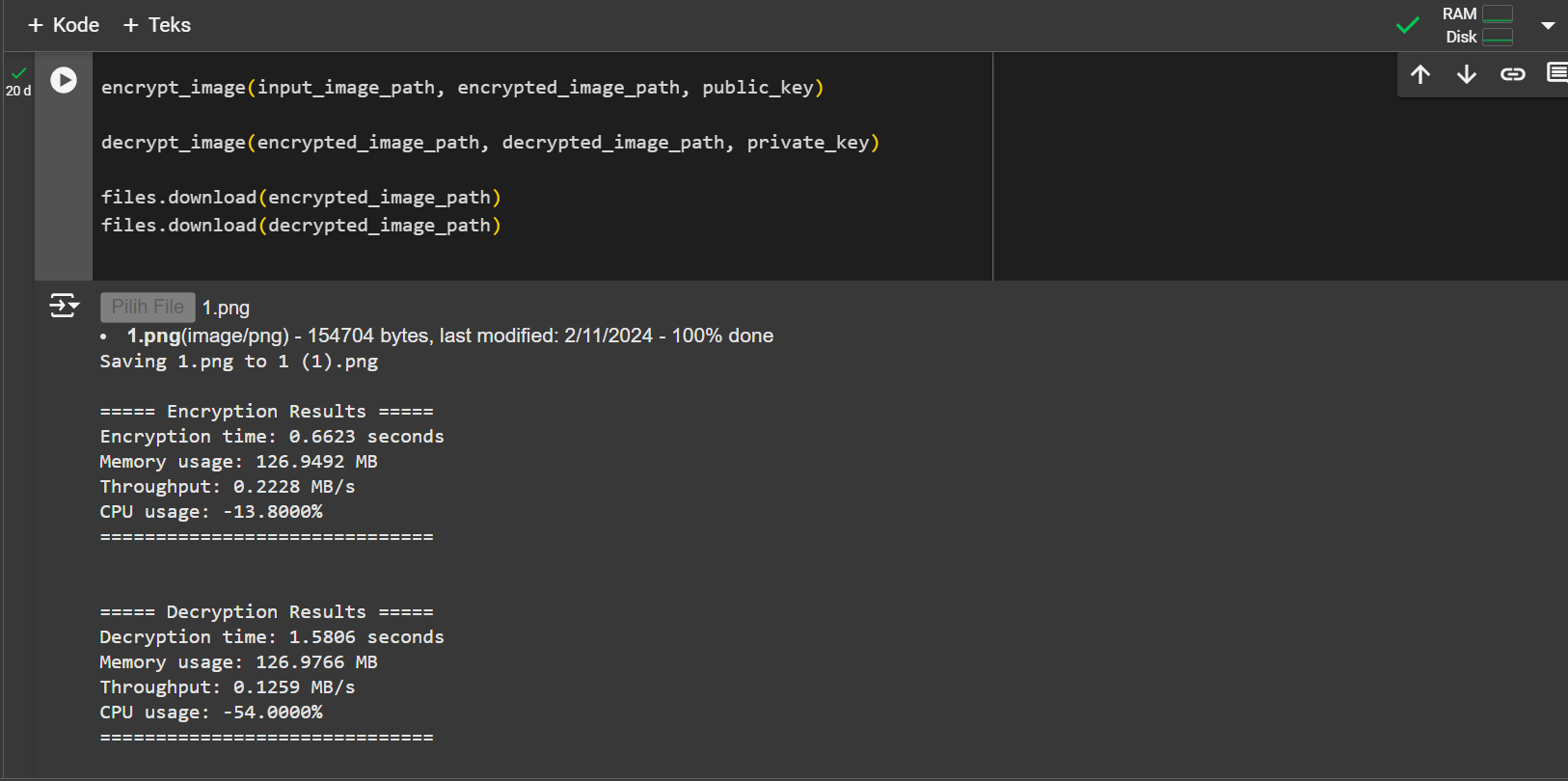
Gambaran program Enkripsi dan Dekripsi yaitu sebagai berikut:



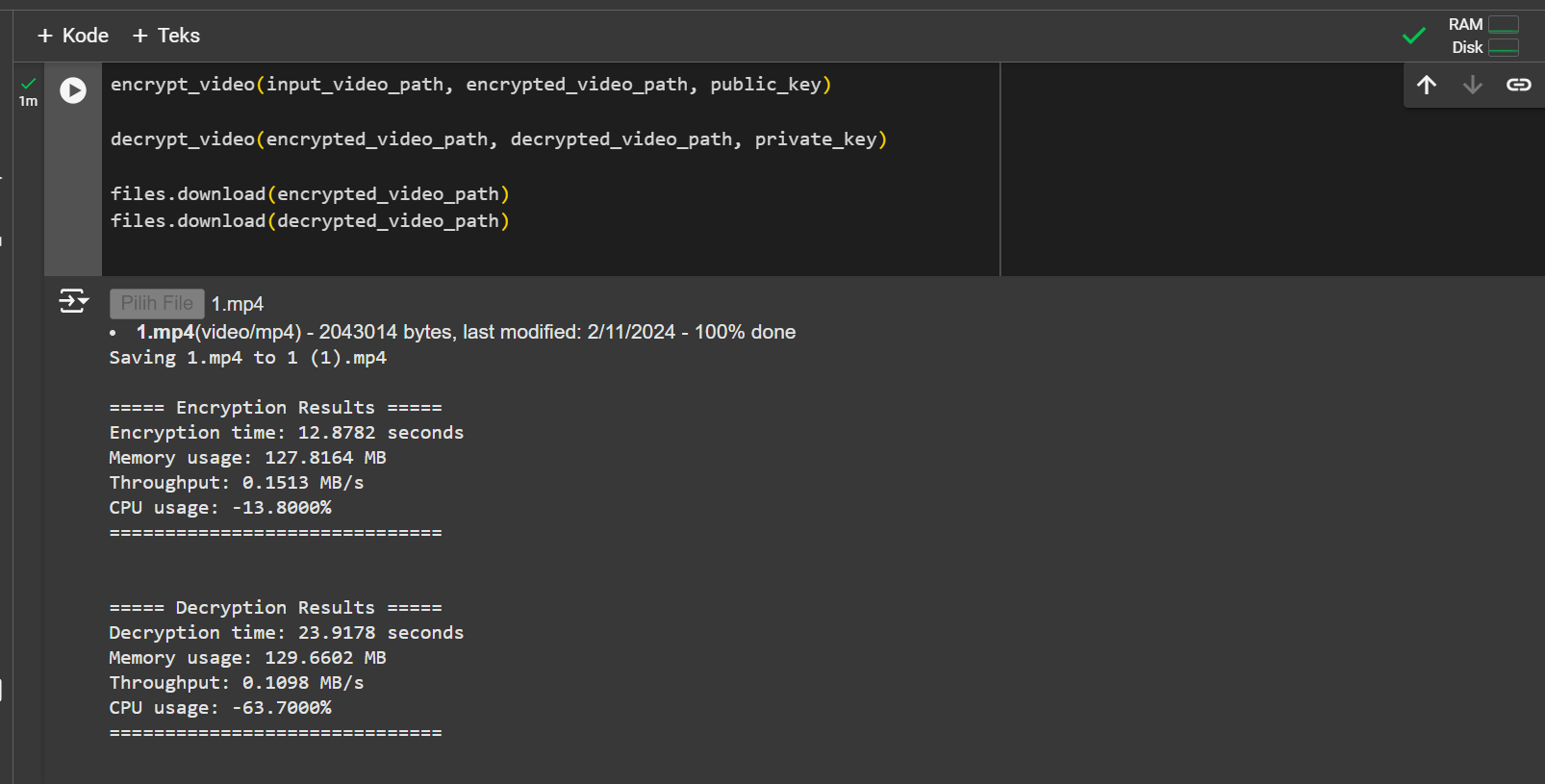
**Gambar 2.** Program Enkripsi dan Dekripsi Gambar dengan Algoritma AES



**Gambar 3.** Program Enkripsi dan Dekripsi Video dengan Algoritma AES



**Gambar 4.** Program Enkripsi dan Dekripsi Gambar dengan Algoritma RSA



**Gambar 5.** Program Enkripsi dan Dekripsi Video dengan Algoritma RSA

Algoritma AES:

| Nama File | Ukuran File (KB) | Enkripsi AES | | | | Dekripsi AES | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Encryption time(s) | Memory usage (MB) | Thoutghput (MB/s) | CPU usage (%) | Encryption time(ms) | Memory usage (MB) | Thoutghput (KB/s) | CPU usage (%) |
| 1,png | 152 | 0,003 | 129,6992 | 49,0617 | 89,8 | 0,0012 | 129,8046 | 116,1899 | -66,7 |
| 2,png | 290 | 0,0033 | 129,8203 | 83,4371 | -11 | 0,0031 | 129,8203 | 90,2967 | 0 |
| 3,png | 987 | 0,0068 | 129,8281 | 139,9905 | -11 | 0,0094 | 129,8281 | 102,103 | -64,7 |
| 4,png | 1,384 | 0,0079 | 129,832 | 170,7645 | 88,8 | 0,0075 | 129,832 | 178,5129 | 41,2 |
| 5,png | 2,393 | 0,0225 | 139,2773 | 103,6121 | 87,9 | 0,021 | 136,9375 | 110,8762 | -35,7 |
| Rata-rata Gambar | | 0,0087 | 131,69138 | 109,37318 | 48,9 | 0,00844 | 131,2445 | 119,59574 | -25,18 |
| 1,mp4 | 1,996 | 0,0119 | 137,3476 | 162,3856 | 37 | 0,0147 | 135,0117 | 132,3683 | 3,5 |
| 2,mp4 | 2,299 | 0,0151 | 138,7539 | 148,3833 | 88,2 | 0,0176 | 136,5195 | 127,4697 | -62,5 |
| 3,mp4 | 3,9 | 0,0401 | 152,0234 | 94,747 | 85,9 | 0,04267 | 152,0195 | 89,2384 | 3,2 |
| 4,mp4 | 5,135 | 0,0388 | 164,2851 | 129,0284 | 37,7 | 0,03718 | 160,4335 | 134,8389 | -7,7 |
| 5,mp4 | 8,632 | 0,0697 | 184,4375 | 120,8811 | 36,4 | 0,0611 | 184,3906 | 137,7512 | -0,4 |
| Rata-rata Video | | 0,03512 | 155,3695 | 131,08508 | 57,04 | 0,03465 | 153,67496 | 124,3333 | -12,78 |

**Tabel 2.** Hasil performa Algoritma AES untuk Enkripsi dan Dekripsi

Algoritma RSA:

| Nama File | Ukuran File (KB) | Enkripsi RSA | | | | Dekripsi RSA | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Encryption time(s) | Memory usage (MB) | Thoutghput (MB/s) | CPU usage (%) | Encryption time(ms) | Memory usage (MB) | Thoutghput (KB/s) | CPU usage (%) |
| 1,png | 152 | 0,6623 | 126,9492 | 0,2228 | -13,8 | 1,5806 | 126,9766 | 0,1259 | -54 |
| 2,png | 290 | 1,8543 | 127,2852 | 0,1525 | 86,2 | 3,4147 | 127,2969 | 0,1116 | -65 |
| 3,png | 987 | 5,1827 | 125,1797 | 0,1858 | -14,1 | 11,1667 | 125,1836 | 0,1162 | 38,5 |
| 4,png | 1,384 | 6,9974 | 126,9648 | 0,1931 | 84,6 | 16,0063 | 126,9844 | 0,1137 | -61,3 |
| 5,png | 2,393 | 12,2319 | 137,0273 | 0,191 | 87,8 | 29,9335 | 137,0273 | 0,1052 | -65,1 |
| Rata-rata Gambar | | 5,38572 | 128,68124 | 0,18904 | 46,14 | 12,42036 | 128,69376 | 0,11452 | -41,38 |
| 1,mp4 | 1,996 | 12,8782 | 127,8164 | 0,1513 | -13,8 | 23,9178 | 129,6602 | 0,1098 | -63,7 |
| 2,mp4 | 2,299 | 17,042 | 130,9844 | 0,1317 | -19,9 | 27,863 | 130,2461 | 0,1086 | -62,1 |
| 3,mp4 | 3,9 | 36,7308 | 146,6484 | 0,1037 | 83,2 | 51,8781 | 149,1602 | 0,0989 | -65,2 |
| 4,mp4 | 5,135 | 57,4802 | 148,7031 | 0,0872 | -25 | 72,4321 | 151,9766 | 0,0933 | -63,8 |
| 5,mp4 | 8,632 | 96,7403 | 175,5586 | 0,0871 | 75,9 | 138,1142 | 164,2031 | 0,0822 | 35,5 |
| Rata-rata Video | | 31,0328 | 138,538075 | 0,118475 | 20,08 | 44,02275 | 140,260775 | 0,10265 | -43,86 |

**Tabel 3.** Hasil performa Algoritma RSA untuk Enkripsi dan Dekripsi

## Evaluasi performa Algoritma AES dan RSA

Hasil evaluasi performa algoritma AES dan RSA untuk enkripsi dan dekripsi video dan gambar dengan menggunakan bahasa pemrograman Python adalah sebagai berikut:

1. Waktu Enkripsi dan Dekripsi:
   1. AES memiliki waktu eksekusi untuk enkripsi dan dekripsi yang lebih cepat dibandingkan dengan RSA, menjadikannya lebih efisien untuk memproses file dengan ukuran besar seperti gambar dan video. Waktu yang cepat ini sangat menguntungkan dalam aplikasi yang membutuhkan waktu respons cepat.
   2. Sedangkan RSA memiliki waktu eksekusi untuk enkripsi dan dekripsi yang lebih lambat, terutama untuk file berukuran besar. Hal ini membuat RSA lebih sesuai untuk data berukuran kecil atau aplikasi yang tidak terlalu mengutamakan kecepatan.
2. Penggunaan Memori:
   1. AES menggunakan lebih banyak memori saat menangani file berukuran besar. Meski begitu, rata-rata penggunaan memori masih terkendali di sekitar 143 MB untuk enkripsi.
   2. RSA memiliki penggunaan memori yang lebih stabil dan lebih rendah dibandingkan AES, dengan rata-rata sekitar 133 MB. Perbedaan ukuran file tidak mempengaruhi penggunaan memori RSA secara signifikan.
3. Throughput (MB/s):
   1. AES menunjukkan throughput yang lebih tinggi, terutama pada file besar, yang mengindikasikan kemampuan AES dalam memproses data dalam jumlah lebih banyak per satuan waktu. Ini membuat AES lebih unggul untuk aplikasi yang memerlukan kecepatan transfer data tinggi.
   2. RSA memiliki throughput yang lebih rendah dibandingkan AES, disebabkan oleh waktu yang dibutuhkan untuk proses enkripsi dan dekripsi lebih lama.
4. Penggunaan CPU:
   1. AES menunjukkan penggunaan CPU yang lebih konsisten dengan rata-rata penggunaan CPU positif, yang menunjukkan efisiensi dalam memanfaatkan sumber daya pemrosesan.
   2. RSA memiliki variasi penggunaan CPU yang lebih besar, termasuk beberapa nilai negatif, menunjukkan bahwa RSA dapat membebani sumber daya saat menangani file besar.

# Kesimpulan dan Saran

Dalam evaluasi performa enkripsi dan dekripsi dengan algoritma AES dan RSA, hasil menunjukkan bahwa kedua algoritma memiliki kelebihan dan kekurangan tergantung pada jenis file dan kebutuhan penggunaan.

* AES unggul dalam hal kecepatan pemrosesan untuk enkripsi dan dekripsi, menjadikannya pilihan yang lebih efisien untuk file berukuran besar, seperti video dan gambar. Rata-rata waktu eksekusi proses enkripsi dan dekripsi oleh AES sangat cepat, meskipun memori yang digunakan lebih besar pada file besar. Dengan throughput yang tinggi, AES cocok untuk data yang intensif dan aplikasi yang membutuhkan pemrosesan cepat.
* RSA, di sisi lain, membutuhkan waktu eksekusi lebih lama untuk enkripsi dan dekripsi jika dibandingkan dengan AES. Namun, RSA menunjukkan stabilitas dalam penggunaan CPU dan memori meskipun throughput-nya lebih rendah. RSA lebih cocok untuk file berukuran kecil atau situasi di mana keamanan lebih diutamakan daripada kecepatan.

Untuk aplikasi yang membutuhkan pemrosesan data dalam jumlah besar dan kecepatan tinggi, AES adalah pilihan yang lebih baik. Sebaliknya, RSA lebih sesuai untuk data kecil yang membutuhkan keamanan ekstra tanpa kebutuhan respons yang cepat.

# Daftar Pustaka

[1] D. I. Gunawan, M. R. Aldizar, I. F. Nasution, and M. F. Nasution, “Comparison of Symmetric and Asymmetric Cryptographic Algorithms,” *UNES Journal of Information System*, vol. 8, no. 1, pp. 1-10, Jun. 2023. P-ISSN: 2528-3502, E-ISSN: 2528-5955.

[2] F. Muharram, H. Azis, and A. R. Manga’, “Analisis Algoritma pada Proses Enkripsi dan Dekripsi File Menggunakan Advanced Encryption Standard (AES),” in *Prosiding Seminar Nasional Ilmu Komputer dan Teknologi Informasi*, vol. 3, no. 2, Dec. 2018. e-ISSN: 2540-7902, p-ISSN: 2541-366X.

[3] B. O. P. Irawan, M. Tahir, N. A. Windrastuti, D. Y. Cholili, D. Mulaikah, and A. B. M. S. Wachid, “Implementasi Kriptografi pada Keamanan Data Menggunakan Algoritma Advanced Encryption Standard (AES),” *Jurnal SimanteC*, vol. 11, no. 2, pp. 100-110, Jun. 2023. P-ISSN: 2088-2130, E-ISSN: 2502-4884.

[4] A. Hermawan and E. I. H. Ujianto, “Implementasi Enkripsi Data Menggunakan Kombinasi AES dan RSA,” *INFOTEKJAR: Jurnal Nasional Informatika dan Teknologi Jaringan*, vol. 5, no. 2, pp. 75-85, Mar. 2021. ISSN (Print): 2540-7597, ISSN (Online): 2540-7600.

[5] D. Sari, H. Prabowo, and S. Utami, “An Analysis of RSA Algorithm in Secure Data Communication,” *International Journal of Computer Applications*, vol. 175, no. 5, pp. 10-15, 2021.

[6] RCDevs, “Cryptography: What is RSA Encryption and How it Works?” Accessed 2024.

[7] Simplilearn, “RSA Algorithm: Secure Your Data with Public-Key Encryption,” Accessed 2024.

[8] R. S. Mahmoud and H. Abdelwahab, “Combining RSA and Traditional Ciphers to Improve Security in Data Transmission,” *Journal of Communications and Network Security*, vol. 15, no. 3, pp. 210-218, 2022.

[9] D. Lex, “RSA Encryption & Decryption In Python: Key Creation, Storage, Algorithm, and Forward Secrecy,” Apr. 24, 2022.

[10] Basile, “AES Encryption & Decryption In Python: Implementation, Modes & Key Management,” Apr. 13, 2022.

[11] H. Simangunsong and M. A. Raharja, “Penerapan Algoritma Advanced Encryption Standard (AES128) Dengan Mode ECB Dalam Pengamanan File,” *JNATIA: Jurnal Nasional Teknologi Informasi dan Aplikasinya*, vol. 1, no. 1, pp. 743-750, Nov. 2022.