Implementasi SHA-256 dalam Program Verifikasi Originalitas Video Sebelum dan Sesudah Proses Kriptografi

Daniel Surya Wijaya[a1], I Ketut Gede Suhartana[a2]

[a]Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana
Jalan Raya Kampus UNUD, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia
[1]daniel999.dd@gmail.com
[2]ikg.suhartana@unud.ac.id

Abstract

This research aims to develop a computer program utilizing the SHA-256 algorithm to compare the authenticity between the original video and the video that has undergone cryptographic processes, particularly during the decryption phase. The program is designed to provide additional verification regarding the success of the decryption process in restoring the video to its original condition. The program development is conducted using the Python programming language. The SHA-256 algorithm is employed to generate hash values for both the original video and the decrypted video. The resulting hash values of the two videos are then compared to evaluate their similarity. The developed program successfully compares the authenticity between the original video and the decrypted video. Through the analysis of hash values using SHA-256, the program concludes whether the decryption process successfully restores the video to its original state or not.

Keywords: *SHA-256, Cryptography,Video,Hash,Python*

1. Pendahuluan

Di era digital saat ini, memastikan keamanan dan integritas konten multimedia, seperti video, menjadi sangat penting, terutama saat mentransmisikan informasi sensitif melalui jaringan atau menyimpan data di lingkungan cloud. Kriptografi memainkan peran penting dalam melindungi konten multimedia dengan mengenkripsi, sehingga mencegah akses tidak sah dan memastikan kerahasiaan. Namun, sementara enkripsi memberikan tingkat keamanan yang tinggi, hal itu memperkenalkan tantangan untuk memverifikasi keaslian konten yang sudah didekripsi.Proses dekripsi, khususnya dalam skema enkripsi asimetris di mana kunci yang berbeda digunakan untuk enkripsi dan dekripsi, memerlukan mekanisme yang kuat untuk memvalidasi apakah konten yang sudah didekripsi tetap tidak berubah dari keadaan aslinya. Validasi ini penting untuk aplikasi seperti transmisi video yang aman, forensik digital, dan verifikasi integritas data.Salah satu metode yang banyak diadopsi untuk memverifikasi keaslian konten adalah melalui penggunaan fungsi hash kriptografi. Fungsi-fungsi ini menghasilkan nilai hash berukuran tetap, atau "sidik jari," dari data yang secara unik mewakili isinya. Di antara berbagai fungsi hash kriptografi yang tersedia, algoritma SHA-256 menonjol karena penerapannya yang luas dan sifat kriptografinya yang kuat.[1] Penelitian ini berfokus pada implementasi algoritma SHA-256 dalam sebuah program komputer untuk membandingkan keaslian video sebelum dan sesudah proses kriptografi, khususnya pada tahap dekripsi. Dengan menghasilkan dan membandingkan nilai hash SHA-256 dari video asli dan video yang telah didekripsi, program ini bertujuan untuk memberikan lapisan verifikasi tambahan untuk memastikan pemulihan yang berhasil dari video ke keadaan aslinya setelah proses dekripsi.Hasil dari penelitian ini diharapkan dapat berkontribusi pada pengembangan sistem kriptografi yang lebih kokoh untuk konten multimedia, meningkatkan keamanan dan keandalannya dalam berbagai aplikasi. Selain itu, wawasan yang diperoleh dari studi ini dapat memberikan informasi untuk kemajuan masa depan dalam teknik verifikasi konten dan protokol kriptografi.

2. Metode Penelitian

2.1 Kajian Pustaka

a. Kriptografi

Kriptografi, yang berasal dari bahasa Yunani, terdiri dari dua kata yaitu "kripto" yang berarti menyembunyikan, dan "graphia" yang berarti tulisan. Ini adalah studi yang mempelajari teknik-teknik matematika yang berkaitan dengan keamanan informasi, seperti kerahasiaan, keabsahan, integritas, dan autentikasi data. Namun, tidak semua aspek keamanan informasi dapat diatasi oleh kriptografi. Ini juga bisa dianggap sebagai seni untuk menjaga keamanan pesan. Saat pesan dikirim, kemungkinan disadap oleh pihak tidak berwenang ada. Untuk mengatasi ini, pesan bisa diubah menjadi kode yang tidak bisa dimengerti oleh orang lain.Enkripsi adalah proses mengubah pesan dari yang dapat dimengerti menjadi kode yang tidak bisa dimengerti. Proses kebalikannya, mengubah kode yang tidak bisa dimengerti menjadi pesan yang dapat dimengerti, disebut dekripsi. Kedua proses ini memerlukan mekanisme dan kunci tertentu. Kriptografi adalah ilmu tentang teknik enkripsi di mana data diacak menggunakan kunci enkripsi sehingga sulit dibaca tanpa kunci dekripsi yang sesuai. Kunci dekripsi digunakan untuk mendapatkan kembali data asli. Proses enkripsi menggunakan algoritma dengan beberapa parameter. Biasanya, algoritma tidak dirahasiakan karena mengandalkan kerahasiaan algoritma dianggap tidak aman. Keamanan terletak pada parameter yang digunakan, sehingga kunci ditentukan oleh parameter tersebut.[2]

b. SHA-256

SHA (Secure Hash Algorithm) adalah salah satu algoritma hash yang relatif baru, yang dikembangkan oleh The National Institute of Standards and Technology (NIST) pada tahun 2002. Versi dari algoritma ini, yaitu SHA-256, menghasilkan pesan digest dengan panjang 256 bit. Keamanan SHA-256 didasarkan pada desainnya yang memastikan bahwa pesan yang berbeda tidak akan menghasilkan pesan digest yang sama. Ini berlaku untuk berbagai kasus, seperti ketika kita memproses dokumen citra digital hasil pemindaian ijazah dan transkrip nilai. Proses untuk menghasilkan pesan digest dalam algoritma ini melibatkan lima tahapan:

- Pengisian Pesan (Message Padding): Tahap ini melibatkan penambahan bit-bit ke pesan agar panjang pesan memenuhi syarat tertentu sebelum proses hash dimulai. Ini memastikan bahwa pesan yang diproses memiliki panjang yang sesuai dengan algoritma.
- Pengaturan Panjang Bit (Bit Length Padding): Langkah ini melibatkan penambahan informasi tentang panjang pesan asli ke dalam pesan yang telah di-padding sebelumnya. Ini penting untuk memastikan integritas data dan menghindari serangan pengubahan data yang memanfaatkan panjang pesan yang sama.
- Inisialisasi Nilai Hash Awal (Initial Hash Value Initialization): Nilai hash awal ditentukan sebelum proses hash dimulai. Nilai-nilai ini biasanya ditetapkan oleh algoritma dan menjadi titik awal dari proses hash.
- Pemrosesan (Processing): Pada tahap ini, pesan yang telah di-padding dan diatur panjang bitnya diproses menggunakan fungsi hash yang kompleks. Proses ini melibatkan pengulangan serangkaian operasi yang dirancang untuk mengacak data dan menghasilkan pesan digest.
- Output: Setelah proses pemrosesan selesai, pesan digest dihasilkan sebagai output. Pesan digest ini adalah representasi unik dari pesan asli dan digunakan untuk verifikasi integritas dan otentikasi data.[3]

c. Nilai Hash

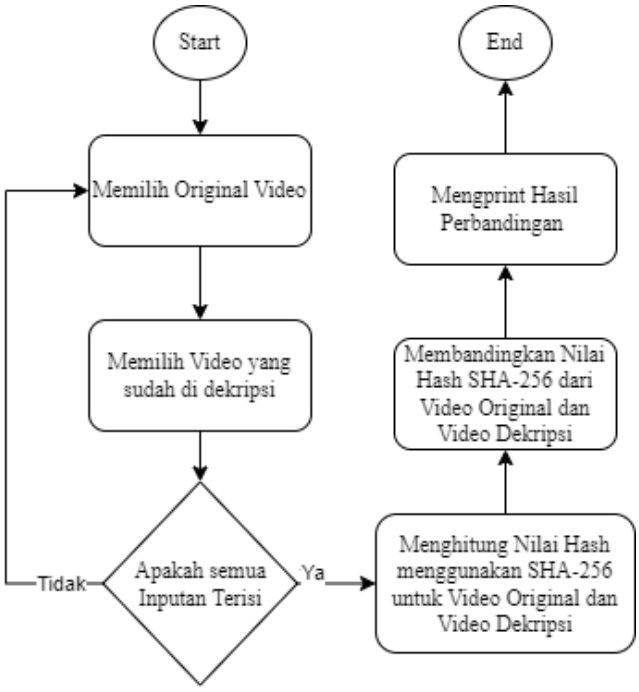
Nilai hash, dalam konteks kriptografi dan keamanan informasi, adalah hasil dari proses penghitungan yang disebut fungsi hash yang diterapkan pada data input. Fungsi hash ini bertanggung jawab untuk mengonversi data input, seperti teks atau file, menjadi representasi numerik yang unik dan memiliki panjang tetap, yang sering kali diekspresikan dalam bentuk heksadesimal. Nilai hash memiliki sifat yang sangat penting dalam

keamanan informasi karena merupakan representasi digital dari data yang diberikan, yang memungkinkan verifikasi integritas, autentikasi, dan pencocokan kata sandi. Nilai hash sering digunakan dalam berbagai aspek keamanan informasi, termasuk dalam penyimpanan dan transmisi data sensitif. Dengan menggunakan nilai hash dari suatu data, kita dapat memverifikasi apakah data tersebut telah diubah secara tidak sah atau tetap utuh selama penyimpanan atau transmisi. Ini dilakukan dengan membandingkan nilai hash data yang baru dengan nilai hash data yang disimpan sebelumnya. Jika nilai hash sama, maka data dianggap utuh; namun, jika nilai hash berbeda, itu menandakan bahwa data telah diubah atau rusak. Selain itu, nilai hash juga digunakan dalam proses autentikasi, di mana nilai hash dari kata sandi pengguna disimpan secara aman dan hanya nilai hashnya yang dibandingkan saat pengguna mencoba masuk. Ini membantu mencegah akses tidak sah ke sistem oleh pihak yang tidak berwenang. Selain itu, nilai hash juga digunakan dalam pencocokan kata sandi, di mana nilai hash dari kata sandi yang dimasukkan oleh pengguna dibandingkan dengan nilai hash yang disimpan dalam basis data. Ini memastikan bahwa kata sandi tidak pernah disimpan dalam teks biasa, yang dapat menjadi target potensial bagi serangan peretas [4].

d. Video

video merupakan salah satu jenis media audio-visual dan dapat menggambarkan suatu objek yang bergerak bersama-sama dengan suara alamiah atau suara yang sesuai. Video menyajikan informasi, memaparkan proses, menjelaskan konsep yang rumit, mengajarkan keterampilan, menyingkat atau memperpanjang waktu, dan mempengaruhi sikap [5].

2.2 Flowchart Program



Gambar 1. Gambar Flowchart Aplikasi

Dalam program ini, langkah-langkah untuk membandingkan video asli dengan video yang telah dideskripsi telah dioptimalkan dengan menggabungkan proses penghitungan nilai hash SHA-256 dari kedua video tersebut. Setelah pengguna memilih kedua file video, program membaca keduanya secara bersamaan. Selama proses membaca, program juga menghitung nilai hash SHA-256 untuk kedua video tersebut secara langsung. Penggunaan algoritma hashing SHA-256 memungkinkan program untuk menghasilkan nilai hash unik yang mewakili konten video dengan cepat dan efisien. Setelah nilai hash dari kedua video dihitung, program membandingkan keduanya. Jika nilai hash dari video yang telah dideskripsi sama dengan nilai hash dari video asli,

program menyimpulkan bahwa video yang telah dideskripsi adalah identik dengan video asli. Namun, jika nilai hashnya berbeda, itu menunjukkan bahwa ada perbedaan antara video yang telah dideskripsi dan video asli. Hasil perbandingan kemudian dicetak atau ditampilkan kepada pengguna, dan program berakhir.

2.3 Pembuatan Aplikasi

Program yang Saya buat adalah sebuah alat sederhana yang bertujuan untuk membandingkan dua file video. Namun, yang menarik dari alat ini adalah bagaimana ia melakukan perbandingan. Istilah "nilai hash SHA-256" mungkin terdengar rumit, tetapi pada dasarnya, itu adalah cara untuk mewakili konten dari setiap file video dalam bentuk kode unik.Dengan menggunakan nilai hash ini, program Saya mampu melakukan perbandingan antara kedua video dan menentukan apakah keduanya identik atau tidak.Untuk membuat program ini, Saya mengikuti langkah-langkah berikut:

a. Instalasi Python: Memastikan Saya memiliki Python terinstal di komputer saya. Jika belum, saya bisa mengunduh dan menginstalnya dari situs web resminya.
b. Membuat Fungsi untuk Menghitung SHA-256: Saya perlu membuat fungsi untuk menghitung nilai hash SHA-256 dari file. Saya juga dapat menggunakan modul hashlib yang sudah disediakan oleh Python. Fungsi ini membuka file, membaca isinya dalam potongan (chunks), dan mengupdate nilai hashnya secara bertahap.
c. Membandingkan Nilai Hash: Setelah saya memiliki nilai hash untuk kedua file video, saya bisa membandingkannya. Jika keduanya sama, artinya video tersebut identik, sedangkan jika berbeda, artinya ada perbedaan di antara keduanya.
d. Menampilkan Hasil: Terakhir, saya bisa menampilkan hasil perbandingan ke layar, misalnya dengan mencetak pesan yang sesuai seperti "Video yang sudah dideskripsi sama dengan video asli."

Setelah saya menulis kode programnya, saya bisa menyimpannya dalam sebuah file Python dengan ekstensi .py, dan kemudian menjalankannya dari terminal atau lingkungan pengembangan Python seperti PyCharm atau VSCode.

3. Hasil dan Diskusi

3.1. Uji Coba Aplikasi

Tabel 1. Hasil Uji Coba Aplikasi

Video Original	Video Terdekripsi	Hasil dari Program
Video 1	Video 1d	Identik/Sama
Video 2	Video 2d	Identik/Sama
Video 3	Video 3d	Identik/Sama
Video 4	Video 4d	Identik/Sama
Video 5	Video 5d	Identik/Sama
Video 6	Video 6d	Identik/Sama
Video 7	Video 7d	Identik/Sama
Video 8	Video 8d	Identik/Sama
Video 9	Video 9d	Identik/Sama
Video 10	Video 10d	Identik/Sama
Video 11	Video 11d	Identik/Sama
Video 12	Video 12d	Identik/Sama

Video Original	Video Terdekripsi	Hasil dari Program
Video 13	Video 13d	Identik/Sama
Video 14	Video 14d	Identik/Sama
Video 15	Video 15d	Identik/Sama
Video 16	Video 1d	Berbeda
Video 17	Video 2d	Berbeda
Video 18	Video 3d	Berbeda
Video 19	Video 4d	Berbeda
Video 20	Video 5d	Berbeda
Video 21	Video 6d	Berbeda
Video 22	Video 7d	Berbeda
Video 23	Video 8d	Berbeda
Video 24	Video 9d	Berbeda
Video 25	Video 10d	Berbeda
Video 26	Video 11d	Berbeda
Video 27	Video 12d	Berbeda
Video 28	Video 13d	Berbeda
Video 29	Video 14d	Berbeda
Video 30	Video 15d	Berbeda
Video 31	Video 16d	Berbeda

Dalam analisis yang Saya lakukan terhadap 31 sampel video, terdapat dua hasil yang diharapkan:

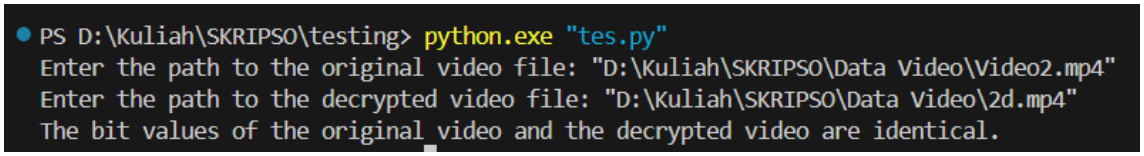
a. Video Asli dan Video yang Telah Didekripsi Identik (16 Sampel):
Dalam hasil ini, program saya seharusnya menemukan bahwa nilai hash SHA-256 dari video asli dan video yang telah didekripsi adalah sama. Ini menunjukkan bahwa proses enkripsi dan dekripsi tidak mengubah konten dari video tersebut. Hasil ini sesuai dengan harapan, karena video asli dan video yang telah didekripsi seharusnya memiliki konten yang identik. Konsistensi dalam menemukan kesamaan nilai hash di antara 16 sampel ini menunjukkan bahwa program saya efektif dalam memverifikasi identitas video sebelum dan setelah proses kriptografi.

b. Video Asli dan Video yang Telah Didekripsi Berbeda (15 Sampel):
Dalam hasil ini, program saya seharusnya menemukan bahwa nilai hash SHA-256 dari video asli dan video yang telah didekripsi berbeda. Ini menunjukkan bahwa terdapat perbedaan dalam konten antara video asli dan video yang telah didekripsi. Hal ini juga sesuai dengan harapan, karena video yang di bandingkan merupakan video yang berbeda sehingga hasil dari nilai hash nya juga akan berbeda yang menyebabkan program mengatakan ini adalah 2 video yang berbeda.

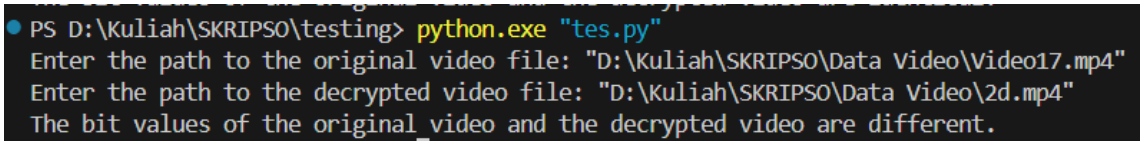
Dalam analisis tersebut, dapat ditambahkan bahwa hasil yang diperoleh dari program sesuai dengan harapan yang diinginkan. Artinya, program berhasil bekerja dengan baik dalam membandingkan video asli dan video yang telah didekripsi, serta menghasilkan output yang konsisten dengan ekspektasi. Hal ini memberikan keyakinan bahwa program saya dapat diandalkan dalam memverifikasi identitas video sebelum dan setelah proses enkripsi, serta membedakan antara video yang identik dan yang berbeda. Dengan demikian, keberhasilan

p-ISSN: 2986-3929
e-ISSN: 3032-1948

program ini memberikan kontribusi penting dalam memastikan keamanan dan integritas data dalam konteks penggunaan video.



Gambar 2. Gambar Contoh Uji Coba yang Berhasil



Gambar 3. Gambar Contoh Uji Coba yang Tidak Berhasil

4. Kesimpulan

Kesimpulan dari penelitian ini adalah bahwa penggunaan algoritma hash SHA-256 dalam program verifikasi originalitas antara video asli dan video setelah proses dekripsi memberikan hasil yang memuaskan. Dengan menggunakan nilai hash SHA-256 sebagai representasi unik dari setiap video, program ini mampu membandingkan kedua versi video dengan akurasi tinggi. Melalui implementasi ini, ditemukan bahwa program berhasil mengidentifikasi video yang identik dengan nilai hash yang sama sebelum dan sesudah proses dekripsi. Hal ini menunjukkan bahwa proses dekripsi tidak mengubah konten dari video tersebut dan video yang telah dienkripsi berhasil dipulihkan ke keadaan aslinya.Selain itu, program juga berhasil membedakan video yang berbeda dengan nilai hash yang berbeda. Ini menunjukkan kemampuan program dalam mendeteksi perbedaan antara video asli dan video yang telah didekripsi.Dengan demikian, kesimpulan dari penelitian ini adalah bahwa penggunaan SHA-256 dalam program verifikasi originalitas video asli dan video setelah proses dekripsi efektif dalam memastikan keaslian dan integritas data video.

Daftar Pustaka

[1] S. Sulastri and R. D. M. Putri, “Implementasi Enkripsi Data Secure Hash Algorithm (SHA-256) dan Message Digest Algorithm (MD5) pada Proses Pengamanan Kata Sandi Sistem Penjadwalan Karyawan,” *J. Tek. Elektro*, vol. 10, no. 2, pp. 70–74, 2018, doi: 10.15294/jte.v10i2.18628.

[2] M. M. Amin, “Implementasi Kriptografi Klasik Pada Komunikasi Berbasis Teks,” *Pseudocode*, vol. 3, no. 2, pp. 129–136, 2017, doi: 10.33369/pseudocode.3.2.129-136.

[3] S. Nainggolan, “RESOLUSI : Rekayasa Teknik Informatika dan Informasi Implementasi Algoritma SHA-256 Pada Aplikasi Duplicate Document Scanner,” *Media Online*, vol. 2, no. 5, pp. 201–213, 2022, [Online]. Available: https://djournals.com/resolusi.

[4] W. Stallings, *Cryptography and Network Secuirty*. 2017.

[5] R. Aboe, “Penggunaan Media Video Dalam Pembelajaran Speaking,” *Hum. J. Penelit.*, vol. 11, no. 1, pp. 33–38, 2020, doi: 10.33387/humano.v11i1.1937.